Įvadas į kiekybinius metodus su R programa

Metodinė medžiaga socialinių mokslų atstovams, siekiantiems pradėti mokytis kiekybinės metodologijos

Dr. Mažvydas Jastramskis
VU TSPMI

Turinys

Įvadas

Lietuvių kalba yra publikuota nemažai vadovėlių, skirtų mokytis statistikos teorijos ir duomenų analizės. Tačiau mokymo priemonių, kurios statistikos nestudijavusį studentą paprastai įvestų į kiekybinės metodologijos pagrindus, vis dar trūksta. Ši metodinė medžiaga skiriasi nuo panašios literatūros tuo, kad ji rengta socialinių mokslų atstovo ir orientuojasi į lengvai suprantamą kalbą.

Reikėtų akcentuoti, kad tai nėra išsamus gidas į kiekybinę metodologiją ir statistikos teoriją (autorius nėra statistikas). Ši priemonė taip pat nėra skirta jau susigaudančiam statistinės analizės programose (*SPSS, Stata, R*) ir norinčiam jas įvaldyti puikiai. Tai „įvado įvadas“, kuriame pristatomi patys paprasčiausi, pagrindiniai kiekybiniai metodai ir parodoma, kaip juos pritaikyti su *R* programa.

Taigi, iš skaitytojo nereikalaujama jokių statistinių ar matematinių žinių, metodinė medžiaga orientuota į pačius pradedančiuosius. Kita vertus, ji tiks ir tiems, kurie prieš kelis metus turėjo kiekybinių metodų kursą ir norėtų atšviežinti bazinius įgūdžius. Tikslinė auditorija yra socialinių mokslų bakalauro ir magistro studentai, tačiau medžiaga ir tinka kiekybinių metodų pradmenis norintiems prisiminti pirmųjų kursų doktorantams. Nors darbui su šia medžiaga statistinių žinių nereikia, pageidautina būti susipažinus su mokslinio tyrimo logika (priklausomas ir nepriklausomas kintamieji, atvejų atranka ir taip toliau) ir turėti bazinius kompiuterinio raštingumo įgūdžius (labiausiai praverstų bent minimali patirtis su *Excel*).

Ši metodinė priemonė Jums bus naudinga, jeigu:

- Norite pasiskaičiuoti paprastus procentus su atsisiųstu apklausos duomenų failu.
- Buvote suinstaliavę *R* programą, tačiau nežinojote, nuo kur pradėti.
- Siekiate kiekybiškai įvertinti ryšį tarp dviejų veiksnių ir jį grafiškai vizualizuoti.
- Jums reikia įgyti bazines kiekybinių metodų žinias ir jas iš karto pritaikyti praktiškai su duomenimis.

Medžiaga suskirstyta į tris pagrindinius skyrius. Pirmajame supažindinama su standartine kiekybinių duomenų struktūra ir įvedama į *R* programos aplinką, pateikiami keli analizės pavyzdžiai. Antrajame skyriuje pristatomi pagrindiniai aprašomosios statistikos metodai ir jų

vizualizacijos. Galiausiai, trečiajame supažindinama su išvadų statistikos esminiais principais ir ryšio tarp dviejų veiksnių matais.

Nors daugelyje statistikos vadovėlių teigiama, kad galima iš karto skaityti vieną ar kitą skyrių, tai nepatartina daryti su šiuo: ypač neturint kiekybinės metodologijos žinių iš anksčiau. Kiekvienas skyrius ir poskyris reikalauja žinių, išdėstytų prieš tai buvusioje dalyje. Kadangi metodinėje medžiagoje akcentuojamas praktinis aspektas, po kiekvieno metodo aprašymo seka pavyzdys, kaip jį pritaikyti, naudojant *R* programą. Dirbant labai rekomenduotina šiuos pavyzdžius iš karto atkartoti savarankiškai – pavyzdžiuose naudojamus duomenis (duomenų failus) galima parsisiųsti kartu su šia medžiaga. Idealiu atveju, mokymosi sesijos neturėtų nutrūkti ir būti tęsiamos poskyrio viduryje. Geriausiai būtų vienai darbo sesijai skirti vieną poskyrį.

Kartu su metodine medžiaga pridedamų failų pavadinimai:

- „*2008 porinkimine.sav*“: apklausos duomenys iš 2008 m. porinkiminio tyrimo, failas sudarytas *SPSS* programa
- „*2008 porinkimine_klausimynas.pdf*“: 2008 m. porinkiminio tyrimo apklausos klausimynas
- „*Gyventojai_nedarbas_2015.xlsx*“: kiekybinių duomenų apie gyventojų skaičių ir nedarbą savivaldybėse pavyzdinis failas, sudarytas *Excel* programa
- „*Partijos_Vilnius_2015.csv*“: duomenų apie 2015 m. savivaldos rinkimų rezultatus Vilniaus apylinkėse failas, sudarytas *Excel* programa

1. Duomenų skalės ir *R* pagrindai

1. 1 Duomenų matavimo skalės

Prieš pradedant gilintis į kiekybinius metodus, reikėtų apsiprasti su standartine kiekybinių duomenų lentele. Kad ir kokie būtų surinkti duomenys (rinkimų rezultatai, ekonominiai rodikliai, apklausos duomenų failas su respondentų atsakymai), juos rekomenduotina suvesti į matricą (dvimatę lentelę) su eilutėmis ir stulpeliais, kur: 1) eilutės atitinka tyrimo stebėjimo atvejus; 2) stulpelių pavadinimai reiškia šių atvejų savybes (tyrimo kintamuosius). Stebėjimo atvejai gali būti įvairūs: respondentas, valstybė, profesija ir taip toliau – tai priklauso nuo jūsų darbo. Ypač pradedantiesiems reikėtų laikytis taisyklės, kad kiekviena eilutė reikštų vis kitą stebėjimo atvejį, o kiekvienas stulpelis atitiktų atskirą atvejo savybę.

Pavyzdžiui, 1 paveikslėlyje yra daugeliui pažįstama *Excel* programa padarytos lentelės su savivaldybių duomenimis fragmentas. Pirmoji eilutė čia nesiskaito kaip stebėjimo atvejis – tai savybių (kintamųjų) pavadinimai (šį dalyką verta atsiminti – jis bus svarbus toliau, nuskaitant *Excel* duomenų failus į *R* programą). Pirmo stulpelio pavadinimas yra „Savivaldybė“, tad aišku, kad kiekviena eilutė (nuo antrosios) atitinka atskirą savivaldybę: pirmoji stebėjimo atvejų eilutė (realiai antroji, tačiau pirma yra kintamųjų pavadinimai) yra Alytaus miesto duomenys, antroji – Alytaus rajono ir taip toliau. Antras stulpelis matuoja, kiek savivaldybėse gyventojų. Taigi, Druskininkų savivaldybėje – 20779, Lazdijų – 20995 ir taip toliau. Trečias stulpelis matuoja nedarbo lygmenį procentais. Turbūt akivaizdu, kad šioje duomenų matricoje stebėjimo atvejis yra savivaldybė.

Duomenys, kurie surašyti stulpeliuose „Gyventojų skaičius“ ir „Nedarbo lygis“, yra tikrieji kiekybiniai duomenys. Tai reiškia, kad su jų atskiromis reikšmėmis galima atlikti aritmetines operacijas, kiekybiškai įvertinti skirtumus. Pavyzdžiui, sudėjus Alytaus miesto ir Alytaus rajono gyventojų skaičių, gausime 83229 gyventojų šiuose dviejuose rajonuose. Galime išvesti vidurkį: pirmųjų penkių savivaldybių nedarbo vidurkis būtų 12,56 procentai. Tokių duomenų lyginimas yra objektyvus, matavimo skalė universali: niekas nesiginčys, kad 50 tūkstančių gyventojų yra objektyviai daugiau, nei 40. Tačiau ne visi duomenys socialiniuose moksluose gali būti matuojami tokiomis skalėmis. Kai kurių kintamųjų, pavyzdžiui, daugelio sociologinės apklausos klausimų, tiesiog neįmanoma išmatuoti tikrąja kiekybine skale.

1 pav. Excel duomenų pavyzdys

1	Savivaldybė	Gyventojų_skaičius	Nedarbas_procentais
2	Alytaus m. sav.	55993	11,4
3	Alytaus r. sav.	27236	15,1
4	Druskininkų sav.	20779	10,4
5	Lazdijų r. sav.	20995	15,4
6	Varėnos r. sav.	23751	10,5
7	Birštono sav.	4353	7,4
8	Jonavos r. sav.	44073	10,3
9	Kaišiadorių r. sav.	32141	7,5
10	Kauno m. sav.	302720	7,3

2 paveikslėlyje pavaizduotas apklausos duomenų failo, sudaryto programa *SPSS* (literatūra, skirta mokytis šia programa, pateikiama knygos gale) fragmentas. Jame kiekviena eilutė atitinka atskiro respondento atsakymus. Šis atvejis kiek sudėtingesnis, kadangi jame nėra stulpelio, kuris nurodytų stebėjimo atvejų tapatybes (normalu, nes apklausos dažniausiai yra anonimiškos). Be to, kintamųjų pavadinimai užkoduoti. Kad sužinotume, kokie duomenys konkrečiame stulpelyje, šį kartą užtenka apklausos klausimyno. Atsidarius jį („2008 porinkimine_klausimynas.pdf“, tarp pridedamų su medžiaga failų) su bet kokia „pdf“ tipo failus skaitančia programa ir susiradę kintamąjį „S1“ matome, kad tai buvo klausimas apie respondento lytį: 1 – vyras, 2 – moteris.

2 pav. Apklausos duomenų failo, sukurto su *SPSS* programa, pavyzdys

	S1	S2	S3	S4
1	1	45	3	10
2	2	38	4	9
3	2	43	3	10
4	1	65	5	7
5	1	57	3	3
6	2	30	5	8
7	1	26	4	9
8	1	60	4	10
9	2	49	5	7
10	1	21	3	10
11	2	77	5	3
12	1	82	5	3

Nepaisant to, kad klausimo apie lytį duomenys yra įvesti skaičiais, pastarųjų interpretavimas yra visiškai kitoks nei kintamųjų „gyventojų skaičius“ ir „nedarbas“ iš ankstesnio pavyzdžio. Čia yra nominalioji duomenų matavimo skalė, pagal kurią objektus galima tik klasifikuoti, priskirti vienai ar kitai grupei. Pavyzdžiui, jeigu sudėsime „1 + 2“, gausime „3“, tačiau šis skaičius nebus prasmingai interpretuojamas. Su tokių duomenų reikšmėmis aritmetinės operacijos, bent jau logiškai interpretuojamos, nėra įmanomos: pavyzdžiui, vidurkis nebus interpretuojamas, negalima surasti skirtumo tarp didžiausios ir mažiausios reikšmės ir taip toliau.

Įsivaizduokite, kad klausiate žmonių, kokiam tikėjimui jie save priskiria. Skirtingus atsakymus patogumo dėlei sužymėsite skaičiais nuo 1 iki 5 (pavyzdžiui, 1 – katalikai, 2 – stačiatikiai, ir taip toliau). Jeigu vienas respondentas yra katalikas (žymėtas „1“), o kitas – evangelikas liuteronas („5“), atėmę vieną reikšmę iš kitos gausime 4. Ką reikš šis skaičius? Nieko. Juk negalėtume iš žodžio “evangelikas liuteronas” atimti žodžio “katalikas”. O štai iš vienos savivaldybės gyventojų skaičiaus galima atimti kitos ir gausime objektyviai interpretuojamą reikšmę – pavyzdžiui, Lazdijų rajone gyventojų mažiau, nei Varėnos rajone. Tas pats galioja ir vidurkio skaičiavimui.

Tai nereiškia, kad dirbant su nominaliais kintamaisiais, neįmanomi apibendrinimai (o su statistine analize įprastai ir siekiame apibendrinti). Pavyzdžiui, įmanoma paskaičiuoti, kiek žmonių apklausoje yra katalikai, kiek yra vyrų ir taip toliau. Tačiau būtina akcentuoti, kad tikro kiekybinio kintamojo atveju reikšmė turės objektyviai matuojamą vienetą (vienas gyventojas, vienas litas, vienas nedirbančių procentas), o nominaliam kintajame skaičius tiesiog reikš simbolį, kuris reprezentuoja tam tikrą klasifikacinę grupę (moterys, katalikai, balsavę už liberalus ir taip toliau).

Tiriant gyventojų pažiūras arba kitus reiškinius (pavyzdžiui, skirstant valstybes pagal demokratijos lygmenį), gali būti poreikis išdėstyti duomenis tam tikra tvarka, nors ir nėra universalių matavimų vienetų. Dažnam yra matyta Likerto skalė, kurioje pateikiamas teiginys ir klausiama, kiek respondentas su juo sutinka. Įprastai būna 4-5 atsakymo variantai, nuo „visiškai sutinku“ iki „visiškai nesutinku“. Tokiu būdu gaunami duomenys, kuriuos galima ne tik klasifikuoti, bet ir palyginti reikšmes (daugiau ar mažiau sutinkama).

Likerto skalė ir į ją panašios skalės bendrai vadinamos „rangų“. Tokia skalė naudojama tada, kai galima nustatyti tiriamo požymio skirtumus ir pagal tai objektus išrikiuoti į eilę (požymio intensyvėjimo tvarka). 2 paveikslėlyje galima matyti kintamąjį „S3“, kuris atitinka klausimą (žiūrėti klausimyne) apie respondento išsilavinimą: „1“ – Pradinis, „2“ – Nebaigtas, „3“ – Vidurinis, „4“ – Aukštesnysis, „5“ – Aukštasis. Tai irgi rangų skalė, nes tvarka yra nuo mažiausio rango iki

didžiausio. Reikėtų akcentuoti, kad rangų skale matuotų kintamųjų reikšmės gali būti tarpusavyje lyginamos tik eiliškumui nustatyti: skirtumai tarp reikšmių negali būti įvertinti kiekybiškai. Mes galime teigti, kad žmogus daugiau sutinka ar yra daugiau išsilavinęs, tačiau kiek tiksliai – šios informacijos rangų skalė nepateikia. Dėl aritmetinių operacijų, socialiniuose moksluose tokiems kintamiesiems daromos išimtys, pavyzdžiui skaičiuojami vidurkiai.

Reikėtų akcentuoti, kad skirtumai tarp šių trijų pagrindinių skalių (kiekybinė, nominali, rangų) nėra tik teoriniai. Kai kurie reiškiniai, pavyzdžiui žmogaus politinės preferencijos, tiesiog negali būti išmatuotos kiekybiškai (nėra universalių „liberalizmo“ matų). Tai, kokia skale galime matuoti reiškinį ir kokius surenkame duomenis, lemia tai, kokius metodus galime naudoti ir kaip interpretuojama analizė. Pavyzdžiui, ryšio vaizdavimas ir kiekybinis įvertinimas labai skiriasi nuo to, kokie duomenys yra naudojami (3 metodinės medžiagos skyrius). Vėliau matysime konkrečiau, kokie skirtumai atsiranda pačioje analizėje, o dabar užtektų žinoti, kad pagal taikomus metodus hierarchija yra tokia: 1) Kiekybiniams duomenims sukurta didžioji dalis statistinių metodų (tikriausiai akivaizdu iš skalės pavadinimo); 2) Dalis kiekybiniams duomenims naudojamų metodų taikomi ir rangų skalei; 3) Turint nominalius duomenis, analizės galimybės yra gana apribotos.

Ką reikėtų žinoti perskaičius šį skyrelį?

- Ką duomenų lentelėje (matricoje) reiškia stulpeliai ir eilutės
- Kuo skiriasi kiekybiniai, nominalūs ir rangų skale matuoti kintamieji (duomenys)
- Kodėl negalima vesti vidurkio iš stalo, kėdės ir šaukšto

1.2 Įvado į *R* įvadas

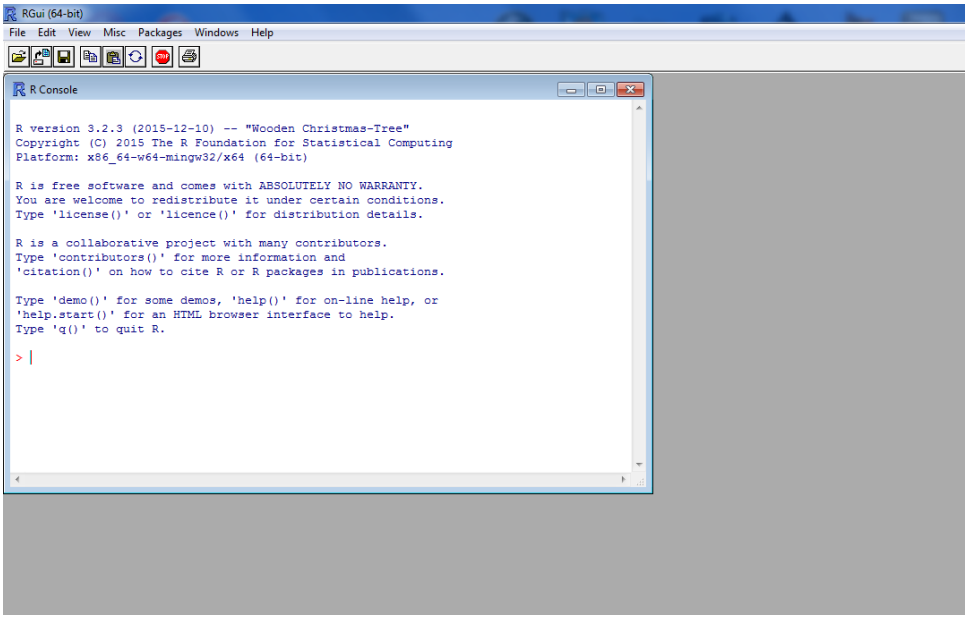
Vienas geriausių dalykų apie *R* yra tai, kad ši programa ir visi jos papildymai yra visiškai nemokami. Egzistuoja keletas *R* versijų. Gana sparčiai populiarėja *R studio*, kurios vartotojo sąsaja yra kiek draugiškesnė. Tačiau pradėjus mokintis nuo jos, didėja rizika neužfiksuoti *R* pagrindų, neišmokti kruopštumo rašant komandas. Tai socialinių mokslų atstovui labai svarbu, nes čia ne vietoje padėtas kablelis ar sumaišytas kintamųjų eiliškumas tiesiog neleis gauti analizės rezultato. Rekomendacija būtų tokia: kiek pasimokykite dirbti su paprasta *R* versija. Jeigu patiks, tęsite darbus ir norėsite supaprastinti kai kuriuos jau išmoktus procesus – pabandykite *R studio*.

Šiame skyriuje nėra dėstomas išsamus įvadas į *R*, tačiau pateikti keli esminiai pagrindai, kuriuos įsisavinus jau galima daryti paprastą analizę su surinktais duomenimis. Todėl skyrius ir pavadintas „įvado į *R* įvadas“. Tiems, kurie norėtų išsamiau gilintis į *R* kalbos subtilybes ir savarankiško kodo rašymą (tai bazinėms statistinėms procedūroms nėra reikalinga), metodinės medžiagos pabaigoje pateikiamos literatūros rekomendacijos. Kaip ir kitus *R* vadovėlius, taip ir šią metodinę medžiagą rekomenduojama skaityti arba paraleliai dirbant su *R*, arba prisėsti prie *R* iš karto po skyriaus perskaitymo. Taip efektyviau formuosis praktiniai įgūdžiai.

R programą *Windows* ar kitai operacinei sistemai galite atsisiųsti iš šio interneto adreso: https://cran.r-project.org/bin/windows/base/. Rekomenduojama siųstis pačią naujausią versiją (metodinės medžiagos rengimo metu tai buvo 3.2.3), nes kai kurie R papildymai su senesnėmis gali tiesiog neveikti. *R* atsisiuntimas ir įdiegimas yra visiškai įprastinis. Kai programa bus įdiegta, atsidarykite ją du kartus paspausdami *R* ikonėlę, atsiradusią ant jūsų *Windows* darbalaukio.

Pirmą kartą (ir antrą, ir visus kitus) atsidarę *R* turėtumėte pamatyti 3 paveikslėlio vaizdą: keleto komandų eilutė viršuje ir konsolės langas (pavadintas *R console*) apačioje. Konsolės lange pateikiama informacijos apie *R*: versija, pranešimas apie garantijos nebuvimą (programa nemokama), keletas galimai naudingų funkcijų (pavyzdžiui, „help()“). Šią informaciją galite laisvai ignoruoti, kaip ir viršuje esančią komandų eilutę. Iš pradžių susipažinkime su *R* veikimo principu – komandų rašymu.

3 pav. *R* programos sąsaja

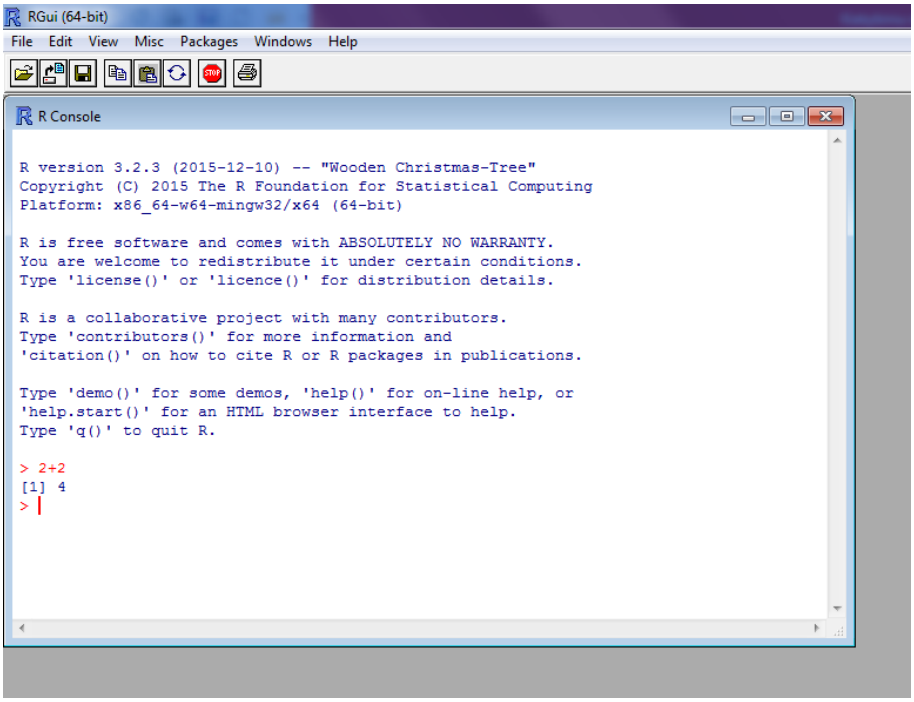


Tikriausiai esate ne kartą rašę tekstą *Word* ar kitoje teksto redaktoriaus programoje. Šiuo požiūriu, darbas su *R* mažai skiriasi nuo teksto rašymo – tik skirtingai nei rašant elektroninį laišką

ar rašto darbą, programa perskaito, ką parašėte ir pateikia tam tikrą rezultatą (arba nepateikia, jeigu įvedėte programai nesuprantamą komandą). Kitaip tariant, su *R* kalbamės būtent jos kalba ir jeigu *R* mus supranta, atsako.

Pavyzdžiui, *R* supranta matematinius skaičiavimus ir gali veikti kaip paprastas kalkuliatorius. Lygiai taip pat, kaip darytumėte naudojant teksto redaktorių, įveskite į konsolę (šalia ženkliuko „>“) paprastą komandą „2+2“. Paspauskite klaviatūros mygtuką *Enter*. Programa turėtų parodyti rezultatą „[1] 4“ (4 paveikslėlis). „[1]“ tiesiog žymi pirmąjį gauto rezultato elementą. Po jo einantis skaičius 4 rodo patį rezultatą (tiek ir turėjome gauti). O dabar įveskite „2+*2“. Turėtumėte gauti įspėjamąją žinutę „Error“: programa nesupranta, ką reiškia iš eilės einantys pliusas ir daugybos ženklas (tai nėra prasminga matematinė operacija). Galite paeksperimentuoti, įvesdami skirtingus skaičiavimus, panaudodami daugybos (*) ir dalybos (/) ženklus.

4 pav. Paprasta *R* komanda ir rezultatas



Būtent taip veikia *R* – įvedame komandą, gauname rezultatą. Tai, ką įvedame kaip komandą, pateikiama raudonu šriftu ir po ženkliuko „>“. Rezultatas pradedamas naujoje eilutėje ir žymimas konsolėje mėlynai. Galima šiuos dalykus (įvedimą ir rezultatą) atskirti, kodą rašant į atskirą failą (vadinamąjį „skriptą“, angliškai *script*), o konsolėje matant rezultatą. Ypač žengiant pirmus žingsnius su *R*, rekomenduojama dirbti būtent taip, nors iš esmės tai patogumo klausimas: pirma, atskiriamas įvedimas ir rezultatas, antra, tai, ką surašėte, galima patogiai išsisaugoti atskirame faile ir vėliau bet kada atsidaryti.

Prieš dirbant toliau, galite ištrinti viską, kas parašyta konsolėje – kartais tai verta padaryti, kad neapsikrautume nebereikalinga informacija ar žinutėmis „error“ apie nepavykusias komandas (jas gauti visiškai normalu ir jos reikalingos, kadangi dažnai parodo, kas negerai mūsų įvedamame tekste). Tai įmanoma pasiekti dviem būdais. Galite paspausti klavišų kombinaciją „Ctrl+L“ arba tiesiog nuveskite pelės žymeklį prie programos lango viršuje esančios komandų grupės „Edit“ ir joje paspauskite „Clear console“ opciją. Gausite švarią lyg tuščias lapas konsolę, visi rezultatai bus ištrinti: beje, to nereikia bijoti. Jeigu turėsite išsisaugoję skriptą su komandomis, juos gauti bus nepaprastai lengva – tai vienas *R* pliusų.

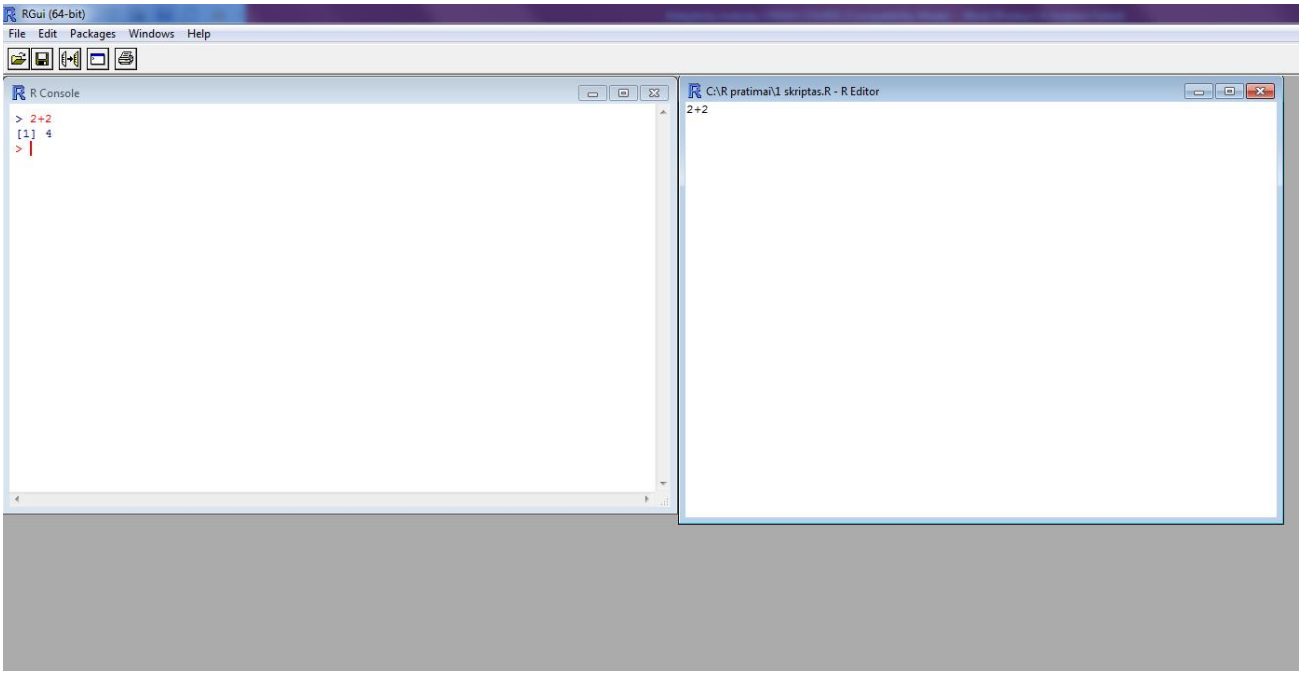
Tam, kad atskirtume įvedimą nuo rezultatų, iš pradžių reikėtų sukurti paprastą tekstinį failą, į kurį rašysime komandas. Šį failą vadinsime skriptu. Nuveskite pelės žymeklį prie programos lango viršuje esančios komandų grupės „File“ ir joje paspauskite „New script“ opciją. Programos aplinkoje atsiras naujas, tuščias lapas, kuriam nuo šiol ir rašysime komandas. Galite išsisaugoti jį sau patogiu pavadinimu – čia veikia ta pati „save“ funkcija, kaip ir kitose programose (*Word, Excel* ir taip toliau). Ją rasite, nuvedę pelės žymeklį prie funkcijų grupės „File“. Rekomenduotina naują skriptą išsisaugoti į aplanką, kurį naudosite tik darbui su *R*. Pavyzdžiui, *C* diske susikurkite aplanką, kurį pavadinsite „R pratimai“ ir jame skriptą išsaugokite pavadinimu „1 skriptas“.

Įrašykite į naują skriptą anksčiau į konsolę vestą komandą „2+2“. Jeigu įvedę paspausite „enter“ – kodas nesuveiks, tiesiog persikelsite į naują teksto eilutę. Taip yra todėl, kad dažnai rašant komandas prireikia daugiau nei vienos eilutės. Tam, kad programa įvykdytų komandą iš skripto, yra keli būdai. Vienas variantas – pastatyti pelės žymeklį komandos (įrašyto teksto) gale ir paspausti klavišų kombinaciją „Ctrl + R“. Antras būdas – pažymėti visą tekstą, kurį norite, kad R įvykdytų, ir paspausti „Ctrl+R“ (jeigu komandos per kelias eilutes, taip patogiau). Galiausiai, komandas paleisti galima ne tik su „Ctrl+R“, bet ir pelės žymeklį nuvedus prie programos funkcijų grupės „ir paspaudus opciją „Run line or selection“. Visgi rekomenduojama įprasti prie klavišų kombinacijos „Ctrl+R“.

Įrašę „2+2“, pažymėję šią komandą ir paspaudę „Ctrl+R“, rezultatą gausite konsolėje (kairėje). 5 paveikslėlyje pateiktas vaizdas to, ką turėtumėte matyti *R*, jeigu atlikote visus aprašytus žingsnius nuo skripto susikūrimo iki paprastos komandos įvykdymo. Nuo šiol kiekvieno dalyko, daromo su *R*, paveikslėliuose nepateiksime, o rašomos komandos bus išskiriamos atskirose pastraipose ir žymimos raudonai (kai vesite jas į savo skriptą, žinoma, jos bus paprasčiausiai juodos).

Rekomenduojama toliau komandas praktikuotis rašyti skriptuose (šiame skyriuje galite toliau naudoti susikurtą „1 skriptas“), o ne konsolėje.

5 pav. *R* komanda skripte ir rezultatas konsolėje



Nors *R* gali atlikti kalkuliatoriaus funkciją, žinoma, tai nėra jos pagrindinis privalumas. Viena iš svarbiausių šios programos savybių ir kartu pliusų yra reikšmių priskyrimas objektams, kuriuos galime pavadinti patys. Sakykime, mums reikės atsiminti skaičiavimo „2+2*“ rezultatą „8“. Įrašykime jį kaip objektą, kurį pavadinsime „A“. Tokiu atveju *R* kalboje naudojamas priskyrimo ženklas „<-“. Paprastai kalbant, jis reiškia „tai, kas yra dešinėje manęs, pavadinkite taip, kaip parašyta kairėje manęs“. Įveskite į skriptą žemiau esančią komandą ir paspauskite „Ctrl+R“ (eilutės gale arba pažymėję visą komandą).

```
A <- 2+2*3
```

Konsolė pakartos tai, ką parašėte: viskas gerai. *R* įvykdė komandą, įrašė dešinėje nuo priskyrimo ženklo esančią informaciją į objektą, pavadintą „A“. Tam, kad sužinotume, kas slypi po objektu „A“, mums tiesiog reikia įrašyti jo pavadinimą į skriptą ir įvykdyti komandą (pažymėkite „A“ ir paspauskite „Ctrl+R“).

```
A
```

Konsolė atkartos komandą „A“ ir kitoje eilutėje parodys, kas įrašyta į objektą „A“. Pirmas objekto „A“ elementas yra „8“ (kitų elementų šiame objekte nėra).

```
> A
[1] 8
```

Šis principas *R* labai svarbus – tai, kad galime priskirti naujiems objektams reikšmes ir pavadinimus, o vėliau lengvai „išsikviesti“ tai, kas buvo priskirta. Pavadinimams reikalavimų nėra daug – pradžioje svarbiausia žinoti, kad jų negalima pradėti skaičiumi ir tai, kad pavadinimas privalo būti vienas žodis. Kitaip tariant, norėdami sukurti objektą „1mano duomenys“, turėtumėte rašyti „mano_duomenys1“. Tokiu pačiu principu *R* galima išsaugoti duomenis, analizės rezultatus (pavyzdžiui, grafiką), funkcijas. Pabandykite įrašyti keletą variantų (pavyzdžiai apačioje), įvykdykite komandas ir panagrinėkite, kada pavadinimas priskiriamas ir reikšmė įrašoma į objektą sėkmingai, o kada gaunate „error“.

```
1objektas <- 5*40
objektas1 <-5*40
antras objektas <- 5*(40+10)
antras_objektas1 <- 5*(40+10)
```

Pasinaudoję komanda „c()“ sukurkime vektorių – to pačio tipo duomenų elementų (jie gali atitikti tam tikrus stebėjimo atvejus) seką. Ši komanda yra kitokia, nei iki šiol rašytos matematinės operacijos. Tai funkcija, kurios skliaustuose rašome tam tikrus argumentus. *R* yra daug įvairių funkcijų – vienos leidžia susirašyti duomenis, o kitos įgalina sukurti sudėtingus grafikus. Kol kas reikėtų įsiminti, kad visų funkcijų bendra išraiška yra tokia: „a(x)“. Paprastai kalbant, „a“ yra unikalus funkcijos pavadinimas, o „x“ – argumentas (argumentai), kurie gali keistis, priklausomai nuo mūsų analizės.

Sukurkime kiekybinių duomenų seką.

```
c(1, 2, 3, 4)
```

Konsolėje gauname rezultatą:

```
> c(1, 2, 3, 4)
```

[1] 1 2 3 4

Įrašykime šią seką kaip objektą „numeris“ . Būtina akcentuoti, kad *R* kalboje yra skirtumas, raidė didžioji, ar mažoji – objektai „numeris“ ir „Numeris“ būtų skirtingi. Todėl įrašę „Numeris“, gausite „error“, o įrašę „numeris“, programa jau parodys keturių skaičių seką. Pasitikrinkime, ką gavome ir pažiūrėkime, kaip *R* traktuoja šį objektą: tam naudojama funkcija „class()“. Gavome tai, kas *R* programos kalboje vadinama *numeric vector* – elementų seka, kurią programa traktuoja kaip skaičius. Pirmojo skyriaus kontekste būtina pabrėžti, kad savo natūra tai gali būti ir kiekybiniai duomenys, bet gali ir žymėti kokybines kategorijas, reikšti nominalų kintamąjį. Šį dalyką kontroliuoja ne programa, o pats tyrėjas.

Nepamirškite, kad norėdami, jog programa parodytų rezultatą, turite savo įrašytas komandas pažymėti ir spausti „Ctrl+R“.

```
numeris <-c(1, 2, 3, 4)
Numeris
numeris
class (numeris)
```

Konsolėje matome tokį rezultatą. Pirma, *R* įrašo skaičių seką ir pavadina ją „numeris“. Antra, bandome išsikviesti objektą „Numeris“. Nepavyksta, kadangi tokio objekto nėra. O štai kai kviečiame objektą „numeris“, programa žino, kad tai keturių skaičių seka ir ją pateikia. Galiausiai, panaudoję funkciją „class()“ pasitikriname, kad tai skaitinė seka (dar galima vadinti skaičių vektoriumi).

```
> numeris <-c(1, 2, 3, 4)
> Numeris
Error: object 'Numeris' not found
> numeris
[1] 1 2 3 4
> class (numeris)
[1] "numeric"
```

Kita vertus, *R* galime kokybinius duomenis įsirašyti žodžiais. Sukurkime elementų seką, turėdami galvoje, kad tai pirmi (pagal gautus balsus) keturi kandidatai 2015 m. Vilniaus mero

rinkimuose. Jeigu seka (kintamasis) sudaromas iš tekstinių įrašų, žodžių, tada reikia naudoti dvigubas kabutes (naudokite būtent tokias, kaip pavaizduota pavyzdyje, tai yra, nelietuviškas – bet kokiu atveju, jeigu patys rašysite *R* komandą, kitokių kabučių programa tiesiog neleis). Pasitikrinkime, kas slypi už objekto „kandidatai“. Gavome tai, kas *R* kalboje vadinama *character vector* – tekstinių reikšmių seka, arba tiesiog tekstinis vektorius. Jeigu jums neleidžia sukurti sekos su lietuviškomis raidėmis, pasikeiskite einamuosius *Windows* kalbos nustatymus iš anglų į lietuvių.

```
kandidatai  <-c ("Šimašius", "Zuokas", "Tomaševskis", "Majauskas")
kandidatai
class(kandidatai)
```

Sukurkime skaičių seką pagal balsus (proc., suapvalintai), kuriuos gavo šie kandidatai pirmame mero rinkimų ture. Pasitikrinkime, kas slypi už objekto „balsai“.

```
balsai <-c(34, 18, 17, 9)
balsai
```

Turime tris kintamuosius (elementų, stebėjimo atvejų sekas): numeris, kandidatai, balsai. Kaip juos sujungti į vieną lentelę? Funkcija „data.frame()“ sukuria vienodo ilgio (tai reiškia, kad elementų skaičius turi būti vienodas) sekų (vektorių) rinkinį – įprastą duomenų lentelę, kurių pavyzdžiai jau buvo pateikti pirmojo skyriaus pradžioje. Pavadinkime šią lentelę „top4“, pasitikrinkime, ar ją gavome ir patikrinkime jos tipą. Įrašykite šias tris eilutes į skriptą ir galite jas visas vienu metu įvykdyti (kaip ir anksčiau, pažymėkite visas tris eilutes ir paspauskite „Ctrl+R“).

```
top4 <- data.frame(numeris, kandidatai, balsai)
top4
class(top4)
```

Viršuje esančių komandų rezultatas konsolėje turėtų atrodyti taip, kaip parodyta po šios pastraipos (jeigu gaunate „error“, tikriausiai nesukūrėte visų trijų elementų sekų – padarykite tai). Pagal pirmą kodo eilutę programa sukuria duomenų lentelę. Pagal antrą kodo eilutę iškviečia lentelę. Matome, kad papildomai programa sunumeruoja stebėjimo atvejus (kaip ir *Excel*). Pirmas stebėjimo atvejis yra Remigijus Šimašius (kintamasis „kandidatai“), kuris pirmajame ture užėmė pirmą vietą („numeris“) ir gavo 34 procentus balsų („balsai“). Atitinkamai galime pažiūrėti

duomenis ir apie kitus tris kandidatus. Galiausiai, pagal funkciją „class()“ programa įvertina, kokio tipo objektas yra „top4“. Žinoma, tai yra duomenų lentelė, R programoje vadinama *data frame*.

```
> top4 <- data.frame(numeris, kandidatai, balsai)
> top4
  numeris  kandidatai balsai
1       1    Šimašius     34
2       2      Zuokas     18
3       3   Tomaševskis     17
4       4   Majauskas      9
> class(top4)
[1] "data.frame"
```

R kalboje (bent jau pradinėse jos stadijose) labai svarbus ženklas yra „$“. Paprastai kalbant, jis reiškia „iš“ tada, kai tam tikrą objektą su konkrečiu pavadinimu norime „ištraukti“ iš kito, kuriam mūsų norimas objektas priklauso. Pavyzdžiui, mums reikia panaudoti konkretų kintamąjį iš duomenų lentelės (dažniausias atvejis, kai naudosime šį ženklą). Norėdami, kad iš duomenų matricos „top4“ mums *R* parodytų tik kintamąjį „balsai“, parašytume tokią komandą.

```
top4$balsai
```

Dar kitas ženklas, kurį būtina išmokti, rašant savo komandas į *R* skriptą, yra „#“. Jo reikšmė yra visiškai kita, nei socialiniuose tinkluose. *R* kalboje šio simbolio reikia tada, kai norime pasakyti programai, kad už „#“ į dešinę yra paprastas tekstas ir jo nereikėtų traktuoti kaip programos kodo. Pavyzdžiui, apačioje pateikiama naudinga funkcija „colnames“, kuri leidžia sužinoti visus konkrečios duomenų lentelės stulpelių (kintamųjų pavadinimus). Už „„#“ yra paaiškinama, kam ta funkcija reikalinga: rašant komandas, kartais labai naudinga šalia pasižymėti komentarus. R paleis kodą iki „#“, o nuo jo į dešinę esantį tekstą tiesiog pakartos konsolėje.

```
colnames(top4) # duomenų lentelės stulpelių (kintamųjų) pavadinimai
```

Pabandykite be „#“ ženklo. Gausite „error“, nes programa galvoja, kad po antro skliausto esantis tekstas yra komanda, kurią ji turėtų skaityti.

```
colnames(top4) duomenų matricos stulpelių (kintamųjų) pavadinimai
```

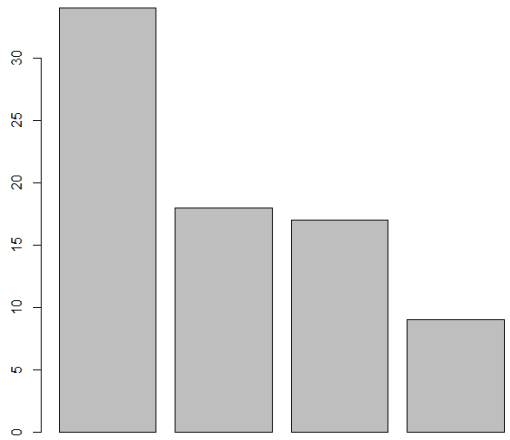
Ženklas „#“ šioje metodinėje medžiagoje dar bus naudojamas ne kartą tais atvejais, kai reikės trumpų paaiškinimų šalia naudojamų komandų. Pavyzdžiui, kaip čia – pateikiamos kelios paprastos funkcijos ir šalia po ženklo #“ paaiškinama, ką jos reiškia. Funkcijoje „head()“ yra du argumentai – pirmasis nurodo duomenų lentelės pavadinimą, antrasis nurodo, kiek pirmų eilučių (stebėjimo atvejų) norime pamatyti. Ji labai naudinga tada, kai naudojamas failas turi labai daug stebėjimo atvejų (pavyzdžiui, 2000 eilučių) ir mes tiesiog norime suprasti, kokius duomenis turime.

```
dim(top4) # eilučių ir stulpelių skaičius matricoje
head(top4, 2) # pirmos dvi duomenų eilutės
head(top4, 3) # pirmos trys duomenų eilutės
```

Šį skyrių pabaigsime analizės pavyzdžiu. Turime keturių kandidatų gautų balsų Vilniaus mero rinkimuose statistiką. Palyginkime kiekvieno kandidato balsų kiekį, naudodami elementarų stulpelinį grafiką (tikriausiai ne kartą matytą). *R* tam yra funkcija „barplot()“. Kaip argumentą įrašome kintamąjį, iš kurio norime, kad programa padarytų grafiką. Prisiminkite, mums reikia nurodyti, kokioje duomenų lentelėje šis kintamasis yra – todėl rašome ne “balsai”, o “top4$balsai“.

```
barplot(top4$balsai)
```

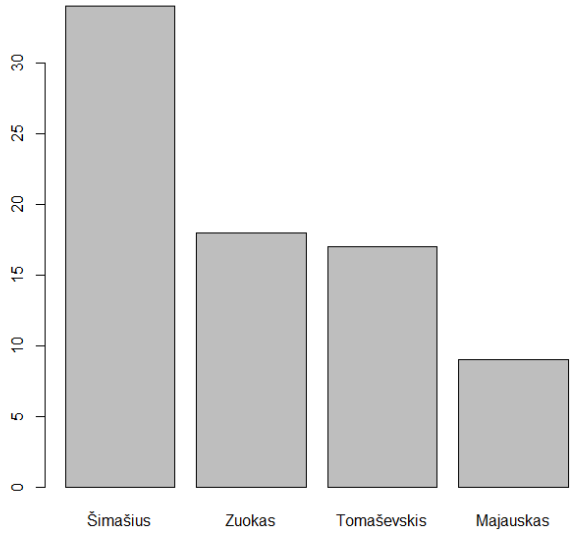
Turėtumėte gauti tokį rezultatą, grafikas atsiranda naujame lange.



Grafike nesimato, koks stulpelis kokio kandidato balsus atitinka. Kodėl? Atsakymas paprastas – mes *R* nenurodėme iš kur imti pavadinimus. Pridėkime į funkciją papildomą argumentą „names“ ir nurodome, kad stulpelių vardai būtų žymimi pagal kintamąjį „kandidatai“. Programa dabar žinos, kokie turėtų būti stulpelių pavadinimai.

```
barplot(top4$balsai, names = top4$kandidatai)
```

Gauname grafiką su kandidatų pavardėmis.



Tikriausiai pastebėjote, kad šį kartą funkcijoje panaudojome lygybės ženklą "=". Jeigu žinome standartinę argumentų seką naudojamoje funkcijoje, jis nėra reikalingas. Visgi dažniausiai naudojame tik kelis argumentus ir tam, kad nesusipainiotume, geriau nurodyti ir argumentų pavadinimus, ir kokios reikšmės jiems priskiriamos. Šiuo atveju „names = top4$kandidatai“ reiškia, kad argumentui „names“ (standartinis funkcijos „barplot()“ argumentas) priskiriame reikšmę „top4$kandidatai“. Programa žinos, kad „names“ nėra tuščias ir ji turėtų pavadinimus įrašyti pagal kintamąjį „kandidatai“.

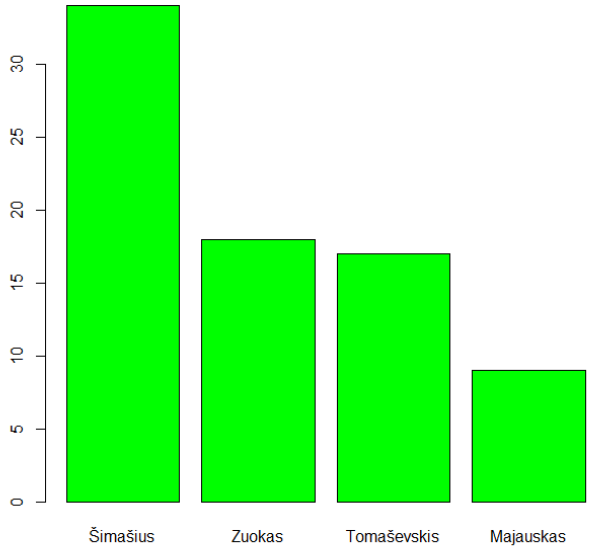
Taigi, *R* kalboje „= „ reiškia priskyrimą (parametro, pavadinimo, objekto). Jį galėtume naudoti ir sukuriant objektus (pabandykite komandas apačioje), tačiau geriau tokioms užduotims naudoti „<-„, kad atskirtume argumentus funkcijose nuo objektų kūrimo už funkcijų ribų.

```
objektas = 2*5
objektas
```

Su *R* kuriant grafikus, gana lengva manipuliuoti įvairiais parametrais. Pavyzdžiui, padarykime, kad mūsų grafiko stulpeliai būtų žali. Tiesiog pridedame papildomą argumentą „col“.

```
barplot(top4$balsai, names = top4$kandidatai, col="green")
```

Gavome žalią grafiką.



Apibendrinant šį skyrelį, reikėtų žinoti šiuos dalykus:

- Kas yra *R* konsolė
- Kaip susikurti ir išsaugoti skriptą, į kurį rašome komandas
- Kaip paleisti į konsolę arba skriptą įrašytas komandas (pasakyti programai, kad duotų rezultatą)
- Ką *R* kalboje reiškia ženklai "#", "<-", "=", "$"
- Ką reiškia R šios funkcijos: c(), class(), data.frame(), colnames(), dim(), head(), barplot ()

1.3 Duomenų įsikėlimas į *R*

R turi daug pliusų, tačiau duomenų parengimas nėra vienas jų. Jeigu dirbate su mažai stebėjimo atvejų, tuomet procesas nebus problemiškas: kaip buvo aprašyta praėjusiame skyriuje, galite su

funkcija „c()“ susikurti kintamuosius (elementų sekas) ir sujungti juos su funkcija „data.frame()“ į paprastą duomenų lentelę. O ką daryti, jeigu norite nusikopijuoti kelių šimtų eilučių duomenis? Arba parsisiuntėte informaciją iš Statistikos departamento, kurie yra *Excel* programos faile? Šiame skyrelyje aptarsime, kaip į *R* persikelti bene dažniausiai socialinių mokslų analizėje pasitaikančius duomenų failų tipus: „.csv“, „.xlsx“ ir „.sav“.

Ruošiant kiekybinius duomenis darbui su *R*, dažniausiai su *Excel* programa susikuriu tekstinį „.csv“ tipo duomenų failą. „Csv“ trumpinys šifruojamas kaip *comma sepparated values* – tokio failo eilutėse duomenys yra atskiriami kableliais, arba kabliataškiais. Būtent pastarasis yra lietuviškas variantas, kadangi pas mus kableliais yra atskiriamos dešimtosios skaičiaus dalys. Tokį failą galima nesunkiai susikurti su *Excel* programa, su funkcija „Save As“ failą išsaugojant naujai, pasirinkus „CSV (Comma delimited)“. Sukuriant „.csv“ failą su *Excel*, būtina, kad duomenys būtų įrašyti tik viename lape (*sheet*).

Kartu su metodine medžiaga galite parsisiųsti failą „Partijos_Vilnius_2015.csv“, kurį dabar ir įsikelsime į *R*. Tačiau prieš tai atsidarykite šį failą su *Excel*. Matysite įprastinę duomenų struktūrą su stulpelių pavadinimais („Apylinke“, „Aktyvumas_proc“). Tai duomenys apie 2015 metų savivaldos rinkimus Vilniuje – koks kiekvienoje iš 151 apylinkių (tiek stebėjimo atvejų ir yra duomenų faile) buvo aktyvumas (procentais) ir kiek gavo balsų keturios didžiausios partijos (procentais). Galite failą atsidaryti ir su *Notepad* programėle, kadangi tai tekstinis formatas. Matysite, kad kiekvienoje eilutėje duomenys yra atskirti kabliataškiu. Esminis klausimas – kaip šiuos duomenis įsikelti į *R*?

Kaip ir daugumai kitų užduočių, *R* turi atskiras funkcijas skirtingo tipo duomenų įsikėlimui. Tačiau prieš panaudojant tokią funkciją, mes būtinai turime programai nurodyti, kur ieškoti duomenų – nusistatyti savo darbinę direktoriją, kitaip tariant, aplanką, iš kurio *R* ims duomenis. Jeigu anksčiau savo kompiuteryje susikūrėte aplanką „R pratimai“, nusikopijuokite į jį failą „Partijos_Vilnius_2015.csv“. Jeigu jūsų darbinio aplanko pavadinimas kitas, viskas gerai, tiesiog nepamirškite to.

Darbinį aplanką su *R* galima nusistatyti dviem būdais. Pirmas paprastesnis – tiesiog nuveskite pelės žymeklį prie funkcijų grupės „File“ ir pasirinkite „Change dir...“. Tada susiraskite aplanką, kuriame yra jūsų duomenys, pažymėkite jį ir paspauskite „ok“. Viskas, direktorija nustatyta (*R* nieko nepraneš konsolėje, kadangi nustatėte darbinį aplanką rankiniu būdu). Kitas būdas – panaudoti funkciją „setwd()“. Skliausteliuose turite įrašyti tikslų aplanko adresą savo kompiuteryje. Paprastumo dėlei naudojame tiesiog *C* diske sukurtą aplanką pavadinimu „R pratimai“. Jeigu jūsų

darbinis aplankas vadinasi kitaip, turite įrašyti būtent jo pavadinimą. Įrašydami aplanko adresą, nepamirškite jo įdėti į kabutes ir naudoti „/“ (simbolį, atskiriant aplankų hierarchiją atkreipkite dėmesį – tai ne „\“, kuris paprastai naudojamas adresuose). Žinoma, kad komanda suveiktų, kaip ir anksčiau, privalote ją pažymėti ir paspausti „Ctrl+R“.

```
setwd("C:/R pratimai")
```

Jeigu negaunate jokių „error“, o programa konsolėje tiesiog pakartoja komandą, reiškia, aplanką nustatėte sėkmingai. Nuo dabar *R* žinos, kad duomenų failų, kuriuos norite įsikelti, reikia ieškoti būtent šioje kompiuterio direktorijoje. Būtina akcentuoti darbinio aplanko (direktorijos) nusistatymo svarbą. Tai viena dažniau pasitaikančių klaidų, pradedant dirbti su *R* – kai prieš bandant duomenis įsikelti programai nenurodoma, kur ji turi ieškoti failų.

Pasitikrinkime – pirma, duomenų failas „Partijos_Vilnius_2015.csv turi būti nusikopijuotas į aplanką “R pratimai“ (ar kitą, su kuriuo dirbate). Antra, *R* programoje turi būti nustatytas būtent šis darbinis aplankas (direktorija). Pagaliau galime prieiti prie trečios dalies, duomenų failo įsikėlimo, arba, kitaip tariant – duomenų nuskaitymo. Jeigu turite „.csv“ duomenų, jiems įsikelti naudojama funkcija „read.csv()“. Nukopijuokite į savo skriptą žemiau esančią komandą ir ją paleiskite.

```
read.csv("Partijos_Vilnius_2015.csv", header=TRUE, sep=";", dec=",")
```

Konsolėje turėtumėte pamatyti duomenis, kuriuos jau matėme duomenų failą atsidarę su *Excel*. Prie jų tuoj grįšime, tačiau reikėtų aptarti kiekvieną iš naudotų funkcijos „read.csv()“ argumentų. Argumente „file“ nurodome duomenų failo pavadinimą. Jis privalo būti tikslus ir kabutėse, vienos raidės netikslumas ves prie „error“. „Header“ argumente nurodome, ar duomenų stulpeliai (kintamieji) turi pavadinimus (jei turi – „TRUE“, jei ne – „FALSE“). „Sep“ – kokiais simboliais atskiri duomenys (anksčiau matėme, kad kabliataškiais). Galiausiai, „dec“ argumente nurodome, kokiu simboliu buvo atskirtos dešimtosios skaičiaus dalys (kableliu).

Tam, kad galėtume dirbti su šiais duomenimis, reikėtų duomenis išsaugoti kaip atskirą objektą. Jau žinome, kaip su R priskirti objektams reikšmes, pavadinti duomenų lentelę. Žemiau esanti komanda leidžia iš duomenų failo „Partijos_Vilnius_2015.csv“ *R* aplinkoje sukurti objektą „failas1“. Patikrinę su funkcija „class()“ sužinome, kad tai yra duomenų lentelė.

```
failas1 <- read.csv("Partijos_Vilnius_2015.csv", header=TRUE, sep=";", dec=",")
```

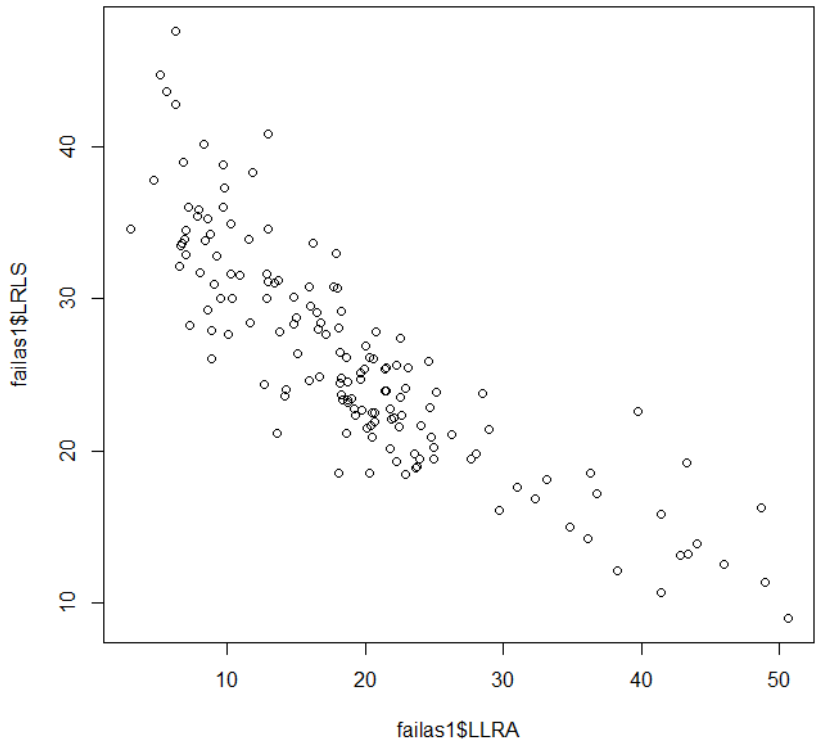
```
class(failas1)
```

Galite pažiūrėti kintamųjų (stulpelių) pavadinimus, kiek yra eilučių ir stulpelių, taip pat (kad nereikėtų konsolėje pateikti visų 151 stebėjimo atvejų) pirmųjų penkių eilučių duomenis.

```
colnames(failas1) # duomenų matricos stulpelių (kintamųjų) pavadinimai
dim(failas1) # eilučių/stulpelių skaičius matricoje
head(failas1, 5) # pirmos kelios duomenų eilutės
```

Ką galima padaryti su šiais duomenimis? Pavyzdžiui, galime sukurti paprastą grafiką, kuriame matysime, kaip susiję LLRA ir LRLS gauti balsai Vilniaus apylinkėse. Žinoma, tolesniuose skyriuose bus pateikta daugiau analizės pavyzdžių. Galite atsiminti funkciją „plot()“.

```
plot(failas1$LLRA, failas1$LRLS)
```



Antras duomenų failo tipas, kurį naudinga žinoti, kaip įsikelti su *R*, yra paprastas „.xlsx“ – standartinis *Excel* duomenų failas. Tačiau šiai užduočiai neužteks standartinės *R* versijos, turėsime ją papildyti. Didelė dalis *R* funkcijų yra prieinamos tik tada, jei suinstaliuotas ir „užkrautas“

atitinkamas paketas (programos papildymas). Tai padaryti nesunku. Pavyzdžiui, tam, kad galėtume nuskaityti „.xlsx“ duomenų failus, mums prireiks paketo „readxl“. Suinstaliuoti labai paprasta, naudojant install.packages() funkciją: skliaustuose kabutėse įrašome norimo paketo pavadinimą.

```
install.packages("readxl")
```

Kai paleisite aukščiau esančią komandą (prisiminkite – pažymime ir „Ctrl+R“), programa jums pateiks sąrašą serverių, iš kurių galima instaliuoti norimą paketą (sąlyga – turite būti prisijungę prie interneto). Realiai didelio skirtumo čia nėra. Galite pasirinkti patį pirmąjį „0 cloud“ ir paspausti „ok“. Programa viskuo pasirūpins, konsolėje turėtumėte pamatyti žinutę apie sėkmingą paketo instaliaciją.

Paketas buvo suinstaliuotas, tačiau tam, kad dabartinėje darbo sesijoje mums būtų prieinamos jo funkcijos, paketą dar reikia atidaryti (užkrauti). Reiktų įsiminti, kad paketą suinstaliuoti užtenka vieną kartą (kaip ir bet kurią kitą programą kompiuteryje), tačiau reikia atidaryti kiekvieną kartą, kai dirbant su *R* mums prireikia jo funkcijų. Paketai užkraunami su „library()“ funkcija. Pasinaudokime ja.

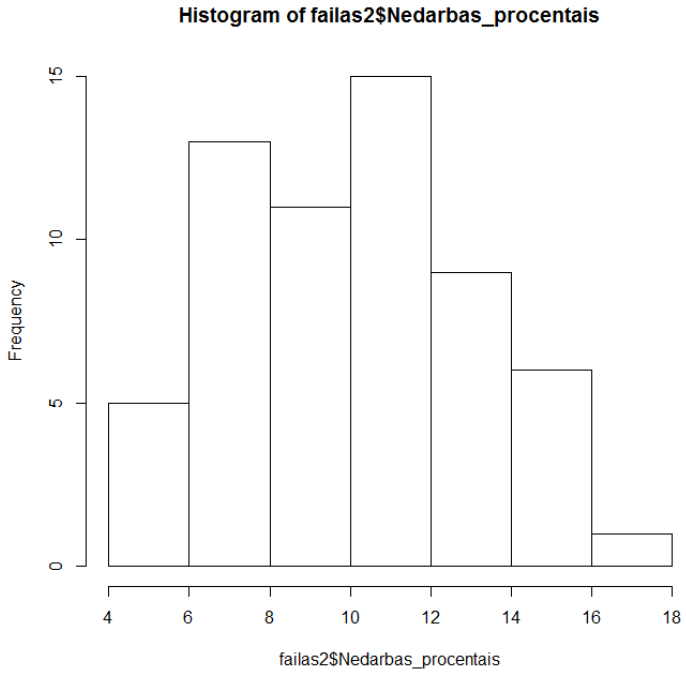
```
library(readxl)
```

Dabar jau galime naudoti funkcijas, esančias šiame pakete. Kartu su metodine medžiaga galite parsisiųsti failą „Gyventojai_nedarbas_2015.xlsx“, kurį dabar ir įsikelsime į *R*. Tam naudojama paketo „readx“ funkcija „read_excel()“. Pirma skliaustuose įrašome failo tikslų pavadinimą, tada argumente „sheet“ nurodome, kurio failo lapo duomenis reikia skaityti ir argumente „col_names“ nurodome, ar yra stulpelių pavadinimai. Galite pastebėti, kad nors kai kurie argumentai reiškia tą patį, kaip funkcijoje „read.csv()“, jie čia vadinami kiek kitaip. Daugiau informacijos apie paketą „readxl“ galima rasti šiuo interneto adresu: https://cran.r-project.org/web/packages/readxl/readxl.pdf.

```
Failas2 <- read_excel("Gyventojai_nedarbas_2015.xlsx", sheet = 1, col_names = TRUE)
```

Galime pasižiūrėti, koks yra nedarbo pasiskirstymas tarp savivaldybių, kokio lygio nedarbas yra dažniausias. Antrajame skyriuje bus aptarta detaliau, ką reiškia šis grafiko tipas – histograma.

```
hist(failas2$Nedarbas_procentais)
```



Galiausiai, paskutinis tipas duomenų failo, kurį gali tekti dažniau sutikti – SPSS programos failas su plėtiniu „.sav“. Dažniausiai šiuose failuose talpinami duomenys iš apklausų. SPSS programa leidžia skaitinėms reikšmėms priskirti tekstinius pavadinimus. Pavyzdžiui, failas gali talpinti duomenis apie domėjimąsi politika skaičiais („1“ arba „2“), tačiau programa žinos, kad „1“ konkrečiame kintamajame reiškia „Domisi“, o „2“ – „Nesidomi“. Nuskaitant failą į tai reikia atsižvelgti: ar į *R* tokius duomenis perkelsime skaičiais, ar tekstu.

Funkcija, leidžianti įsikelti „.sav“ tipo failus, yra pakete „foreign“. Suinstaliuokime ir atidarykime šį paketą.

```
install.packages("foreign")
library(foreign)
```

Kartu su metodine medžiaga galite parsisiųsti failą „2008 porinkimine.sav“, kurį dabar įsikelsime į *R*. Šiame faile yra surašyti respondentų atsakymai iš po 2008 m. Seimo rinkimų vykdytos porinkiminės apklausos. Anksčiau jau buvote atsivertę šios apklausos klausimyną. Dabar mums jo neprireiks, tačiau reikės pakete „foreign“ esančios funkcijos „read.spss()“. Argumentas "use.value.labels" nurodo, ar tam tikriems kintamiesiems imti ne skaičius (pavyzdžiui, „1“), o jų pavadinimus („visiškai pritaria“). Šį kartą pasirinkime „TRUE“ – tai reiškia, kad ten, kur respondento atsakymų kodams yra priskirtos tekstinės reikšmės, *R* skaitys būtent jas. Argumentas „to.data.frame“ reikalingas tam, kad nurodytume programai, jog ji padarytų standartinę duomenų

lentelę. Programa išmes keletą „Warning messages“. Galite jas laisvai ignoruoti, iš praktikos, dirbant su duomenimis problemų neturi kilti.

```
failas3 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = TRUE,  to.data.frame=TRUE)
```

Naudodami išmoktas funkcijas, galite pažiūrėti, kokie yra kintamųjų (stulpelių) pavadinimai, kiek yra eilučių ir stulpelių. Jeigu atsiverstumėte klausimyną „2008 porinkimine_klausimynas.pdf“, pamatytume, kad pirmas klausimas yra apie tai, kiek žmonės domisi politika. Pažiūrėkime, kaip dažnai buvo pasirinktas koks atsakymas su funkcija „table()“.

```
table(failas3$K1)
```

Konsolėje matome, kad labai domėjosi politika 53 apklausos respondentai, visiškai nesidomėjo – 56, ir taip toliau.

```
Labai domiuosi        Domiuosi   Nelabai domiuosi  Visiškai nesidomiu
53                    450        435               56
```

Jeigu nuskaitydami duomenų failą argumente "use.value.labels" nurodysime „FALSE“, vėliau sukuriamoje dažnių lentelėje vietoje reikšmių pavadinimų *R* mums pateiks skaitinius kodus (pažiūrėkite į klausimyną, „1“ reiškia „labai domiuosi“ ir taip toliau). Galite pastebėti, kad iš naujo įrašant informaciją į objektą „failas3“, anksčiau buvę duomenys pakeičiami naujais.

```
failas3 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE)
table(failas3$A1)
```

```
1   2   3   4   9
 53 450 435  56   7
```

Apibendrinant šį skyrelį, reikėtų žinoti šiuos dalykus:

- Kaip su *R* nusistatyti aplanką (direktoriją), kurioje yra jūsų duomenų failai
- Kaip įsikelti „.csv“ tipo duomenų failą
- Kam reikalingi *R* paketai, kaip juos suinstaliuoti ir atidaryti
- Kaip į R įsikelti „.xlsx“ ir „.sav“ tipo duomenis

- R funkcijos: „setwd()“, „read.csv()“, „read_excel()“, „read.spss()“
- R paketai: „readxl“, „foreign“

2. Aprašomosios statistikos pagrindai

Aprašomoji statistika apima duomenų sisteminimo ir grafinio vaizdavimo metodus. Reikėtų akcentuoti, kad šie metodai netikrina hipotezių (ryšių tarp kintamųjų): tuo jie skiriasi nuo išvadų statistikos. Aprašomąja statistika siekiama apibendrinti turimus duomenis apie vieną ar daugiau kintamųjų. Bendrąja prasme kintamasis yra sąvoka, kuri gali turėti daugiau nei vieną reikšmę, o kiekybinėje analizėje kintamasis yra tiesiog stebėjimo atvejų savybė, apie kurią mes turime informaciją. Pirmajame skyriuje turėjom įvairių kintamųjų pavyzdžių – nedarbas, gyventojų skaičius, lytis, tikėjimas, domėjimasis politika.

Taigi, šiame skyriuje pateikiamos bazinės žinios apie tai, kaip kiekybiškai apibendrinti turimus duomenis, priklausomai nuo to, kokia skale jie matuoti (žr. 1.1 skyrelį). Pirmoje skyriaus dalyje aptariami paprasti ir poriniai dažniai (procentai) kintamiesiems, kurie įgyja nedaug reikšmių (apklausos klausimai). Antrame ir trečiame skyreliuose pristatomos duomenų padėties ir sklaidos charakteristikos. Galiausiai, ketvirtajame skyrelyje pristatomi keli standartiniai grafiniai sprendimai, norint vizualizuoti duomenų apibendrinimą. Visi metodai pristatomi ne tik teoriškai – iš karto parodoma, kaip juos atlikti su *R* programa.

Šiame skyriuje dažnai bus naudojama imties sąvoka. Imtis (angl. *sample*) – tai atrinkta stebimos populiacijos (visi objektai, turintys mus dominantį požymį) dalis, apie kurią buvo renkami duomenys, visuma. Raidė „N“ žymi imties dydį. Pavyzdžiui, jeigu surinkome informaciją apie nedarbą 60 savivaldybių, mūsų imties dydis (įprastai žymimas didžiąja „N“ raide) bus lygus 60. Jeigu apklausoje yra 1003 respondentai, tai imties dydis lygus 1003.

Kartais apie dalį respondentų ar kitų stebėjimo atvejų neturėsime informacijos – jie bus neatsakę į klausimą, trūks informacijos apie šalies BVP konkretiems metams, ir taip toliau. Tokie atvejai vadinami „trūkstamomis reikšmėmis“ (angl. *missing values*). Dirbant su jais egzistuoja du pasirinkimai. Pirmas – taikant tokius metodus, kaip įprasti dažniai, juos tiesiog palikti (juk galime suskaičiuoti, kiek žmonių pasakė „nežinau“). Tačiau taikant kitus metodus, pavyzdžiui, skaičiuojant vidurkį (negalime skaičiuoti vidurkio pridedant „nežinau“), analizuojamų stebėjimo atvejų skaičius (patogumo dėlei jį tiesiog vadinsime realiu imties dydžiu) natūraliai sumažės. Taigi, stebėjimo

atvejų skaičius (realus imties dydis(gali kisti priklausomai nuo to, kiek informacijos surinkome ties konkrečiu kintamuoju (savybe).

2.1 Dažniai (procentai)

Paprastas kintamojo dažnis tiesiog parodo, kiek kartų imtyje pasikartojo reikšmė. Pavyzdžiui, 250 respondentų pasakė, kad domisi politika. Santykinis dažnis yra dažnis, padalintas iš imties dydžio: jeigu 250 respondentų iš 1000 sakė, kad domisi politika, reiškia, santykinis dažnis yra 250/100 = 0,25. Padauginę santykinį dažnį iš 100, gausime visiems pažįstamą procentą: 25 procentai respondentų sako, kad domisi politika.

Žinoma, dirbant su realiais duomenimis ne visada gausime tokius gražius, apvalius skaičius. Pavyzdžiui, prezidento galių stiprinimui greičiau pritaria 39 studentai (1 lentelė). Imties dydis yra 105, tad santykinis dažnis lygus 39/105 = 0,371. Padauginę iš 100, gauname, kad 37,1 procentas studentų greičiau pritartų prezidento galių stiprinimui. O kiek ir visiškai pritaria, ir greičiau pritaria? Tai gana lengva sužinoti, pažvelgus į kaupiamąjį dažnį, kuris vis prideda papildomos kategorijos dažnį. Pavyzdžiui, 1 lentelėje pirmųjų dviejų kategorijų bendras procentas yra lygus 42,9 – tiek respondentų pritaria galių stiprinimui.

1 lentelė. Studentų požiūris į prezidento galių stiprinimą, N=105

Kintamojo reikšmė	**Dažnis**	**Santykinis dažnis**	**Procentas**	**Kaupiamasis procentas**
Visiškai pritaria	6	0,057	5,7	5,7
Greičiau pritaria	39	0,371	37,1	42,9
Greičiau nepritaria	47	0,448	44,8	87,6
Visiškai nepritaria	13	0,12	12,4	100

Duomenys yra apibendrinami dažnių lentelėse tada, kai stebimas kintamasis įgyja nedaug reikšmių. Todėl įprastai dažnius ir procentus skaičiuojame nominaliajai ir rangų skalėms (1 lentelės pavyzdyje naudota rangų skalė, kadangi respondentus galima išrikiuoti nuo visiško pritarimo iki visiško nepritarimo). O tikrieji kiekybiniai duomenys dažnių lentelėse įprastai nėra apibendrinami. Jie turi tiesiog per daug unikalių reikšmių: pavyzdžiui, 20000 ir 20001 gyventojų skaičius yra dvi skirtingos reikšmės. Jeigu norima kiekybinės skalės duomenis perteikti dažniais, tuomet kintamąjį reiktų sugrupuoti į kelias kategorijas ar intervalus.

Paskaičiuokime dažnius su *R* programa. Analizei naudosime 2008 metų porinkiminę apklausą. Pasitikrinkite, ar esate įsirašę jos failą „2008 porinkimine.sav“ į savo darbinį aplanką (direktoriją). Pakartodami 1.2 ir 1.3 skyrelio medžiagą, atlikite šiuos veiksmus: 1) nusistatykite darbinį aplanką; 2) Atsidarykite savo pratybų skriptą arba sukurkite naują; 3) užkraukite paketą, skirtą „.sav“ failų įkėlimui į R; 4) įkelkite failą ir pavadinkite jį „duomenys1“. Visas komandas galite pažymėti kartu ir paspausti „Ctrl+R“. Paskutinė komanda (apklausos failo nuskaitymas) yra per dvi eilutes, todėl nepamirškite, kad prieš įvykdydami kodą turite pažymėti ją visą.

```
setwd("C:/R pratimai") # pasirinktinai, galima ir rankiniu būdu per File -> Change dir..
library(foreign)
duomenys1 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = TRUE, to.data.frame=TRUE)
```

Nuskaitę failą, pamatysite keletą įspėjamųjų žinučių „warning messages“. Ignoruokite jas, praktiškai dirbant su duomenimis tai nekels jokių problemų. Su „colnames“ komanda pasižiūrėkite kintamųjų pavadinimus. Jie nieko nesako – K1, K2 ir taip toliau. Kaip buvo aptarta 1.1 skyrelyje, toks kodavimas būdingas apklausų duomenų failams. Šalia *R* programos atsidarykite ir apklausos klausimyną, šio failo pavadinimas yra „2008 porinkimine_klausimynas.pdf“. Duomenų failo kintamųjų (stulpelių) pavadinimai atitinka apklausos klausimus.

```
colnames(duomenys1)
```

Sukurkime antrą duomenų lentelę, kurioje kokybinių (nominalių ir rangų skalės) kintamųjų reikšmės būtų ne žodžiais, o skaičiais. Skirtumas buvo aptartas 1.3 skyrelyje, tačiau greitai aptarsime jį antrą kartą. Kintamųjų pavadinimai išlieka visiškai tokie patys.

```
duomenys2 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE)
colnames(duomenys2)
```

Sakykime, norime sužinoti, kiek respondentų dar prieš rinkimų kampaniją apsisprendė, už kurią partiją balsuos. Šį aspektą apklausoje matuoja klausimas K6, kurį mūsų duomenų lentelėse „duomenys1“ ir „duomenys2“ atitinka kintamasis „K6“. Paskaičiuokime dažnius kintamajam iš pirmos lentelės „duomenys1“(su reikšmių tekstiniais pavadinimais). Funkcija „table()“ parodo

paprasčiausius dažnius. Jos skliaustuose įrašome kintamąjį, kurio dažnius reikia skaičiuoti. Skliaustuose turime nurodyti duomenų lentelę ir kintamąjį (juk programa automatiškai nežinos, iš kur tą kintamąjį paimti). Tai padeda padaryti ženklas „$“: realiai mes sakome programai - iš duomenų lentelės „duomenys1“ naudok kintamąjį „K6“.

```
table(duomenys1$K6)
```

Konsolėje matome komandą ir jos rezultatą. Įdomu tai, kad daugiausia respondentų (474) apsisprendė dar prieš rinkimų kampaniją. Tokių, kurie apsispręstų tik rinkimų dieną, apklausoje buvo gana nedaug, 57.

```
> table(duomenys1$K6)

Dar prieš rinkimų kampaniją  Rinkimų kampanijos eigoje            Rinkimų dieną
                         474                         203                        57
          Nežino/ neatsakė
                         23
```

Dabar pabandykime dažnius paskaičiuoti iš duomenų lentelės, kurioje nėra tekstinių reikšmių, tik reikšmių kodai.

```
table(duomenys2$K6)
```

Atkreipkite dėmesį, kad patys dažniai išlieka tokie patys. O kodėl jie turėtų keistis? „1“ tiesiog reiškia „dar prieš rinkimų kampaniją“ (pažiūrėkite į klausimyną). Tai, kaip mes pasirenkame vaizduoti atsakymo kodą, nekeičia to, kaip dažnai buvo pasirinktas vienas ar kitas atsakymas.

```
  1   2   3   9
474 203  57  23
```

Toliau naudosime kintamąjį iš lentelės „duomenys1“, taigi, su tekstiniais reikšmių pavadinimais. Sakykime, norime paskaičiuoti santykinį dažnį ir procentus. Žinant, kaip tokie dažniai paskaičiuojami, tai galima padaryti gana paprasta. Prisiminkite, *R* palaiko įprastas kalkuliatoriaus funkcijas.

Tam, kad kiekvieną kartą nereikėtų kartoti komandos „table(duomenys1$K6)“, tiesiog priskirkime lentelei pavadinimą. Tai yra, sukurkime objektą „lentele1“, į kurį įrašysime lentelės dažnius.

```
lentele1 <- table(duomenys1$K6) # sukuriame objektą, talpinantį dažnius
lentele1 # pasitikriname, ar pavyko
class(lentele1) # objekto tipas – lentelė
```

R funkcija „sum()“ yra tokia pati, kaip ir panaši funkcija *Excel*. Ji tiesiog parodo elementų sumą. Jeigu pritaikysime ją mūsų dažnių lentelei, gausime skaičių, kuris parodys stebėjimo atvejų skaičių, tai yra, imtį šiame klausime.

```
sum(lentele1)
```

Gauname skaičių 757. Kodėl ne virš 1000, juk tiek dažniausiai apklausiama žmonių viešosios nuomonės tyrimuose? Šiuo atveju imtis mažesnė dėl to, kad ne visi žmonės buvo klausti apie balso apsisprendimą. Juk nėra prasminga klausti rinkimuose nedalyvavusio žmogaus, kada jis apsisprendė, už kurią partiją balsuos.

Prisiminę skyrelio pradžioje minėtas taisykles, galime lengvai suskaičiuoti santykinius dažnius. Kiekviena iš šių penkių komandų duos tokį patį rezultatą. Galite pastebėti *R* lankstumą – galime naudoti ir imties dydį, kaip skaičių, ir jį paskaičiuojančią komandą. Lygiai taip pat galime naudoti ir lentelę, kaip objektą, ir lentelę paskaičiuojančią pirminę komandą. Skirkite laiko tam, kad pasinagrinėtumėte, kaip prie atsakymo prieinama kiekvienoje iš šių penkių komandų.

```
lentele1/751
lentele1/ sum(lentele1)
table(duomenys1$K6)/751
table(duomenys1$K6) / sum(lentele1)
table(duomenys1$K6) / sum(table(duomenys1$K6))
```

Skyrelio pradžioje kalbėjome apie tai, kad procentus galima paskaičiuoti, iš 100 padauginant santykinį dažnį. Tą patį rezultatą gautume, papildomai apskliaudę bet kurią komandą iš viršuje esančių ir padauginę iš 100.

```
(lentele1/ sum(lentele1))*100
```

```
Dar prieš rinkimų kampaniją  Rinkimų kampanijos eigoje  Rinkimų dieną
                   62.615588                26.816380          7.529723
          Nežino/ neatsakė
                   3.038309
```

Konsolėje gauname rezultatą: 62,6 procentai respondentų apsisprendė dar prieš rinkimų kampaniją, 28,8 procentai rinkimų kampanijos eigoje ir taip toliau. Jeigu jums šie skaičiavimai atrodo kiek per daug komplikuoti, galite tuos pačius rezultatus gauti tiesiog naudodamiesi *R* funkcija „prop.table()“ funkcija. Žemiau esančios komandos yra paprasčiausias būdas gauti vieno kintamojo dažnius, santykinius dažnius ir procentus.

```
lentele1 <- table(duomenys1$K6) # sukuriame dažnių lentelę
prop.table(lentele1) # santykinių dažnių lentelė
prop.table(lentele1)*100 # procentų lentelė
```

Iki šiol buvo kalbama apie vieno kintamojo dažnius. Tačiau gali būti naudinga sužinoti, kaip vienas kintamasis susijęs su antru – pavyzdžiui, išsilavinimas su apsisprendimu, už ką balsuoti. Tokiu atveju abiejų kintamųjų dažniai įrašomi į porinių dažnių lenteles (angl. *crosstabs*).

Sakykime, norime sužinoti, kiek žmonių su aukštuoju išsilavinimu apsisprendė, už kurią partiją balsuos, dar prieš rinkimų kampaniją ir palyginti šį rodiklį su kategorija žmonių, kurie turi vidurinį išsilavinimą. Galime padaryti lentelę. Atkreipkite dėmesį, kad pirmas įrašomas kintamasis bus atvaizduojamas porinių dažnių lentelės eilutėse, o antras – stulpeliuose. Ši tvarka niekada nesikeičia, net jeigu naudosime ir kitus kintamuosius.

```
table(duomenys1$K6, duomenys1$S3)
```

Jeigu visa lentelė netelpa, konsolę galite išsididinti (su pelės kairiuoju mygtuku įspausdami ir patempdami jos kraštą) ir pakartoti komandą. Matome, kad žmonių, turinčių aukštąjį išsilavinimą ir nusprendusių dar prieš rinkimų kampaniją, yra 115. Žmonių, turinčių pradinį išsilavinimą ir apsisprendusių dar prieš rinkimų kampaniją, yra 37. Atrodytų, jog aukštasis išsilavinimas padidina šansus, kad žmogus apsispręs anksčiau. Tačiau atkreipkite dėmesį į tai, kad žmonių su pradiniu

išsilavinimu apskritai yra gerokai mažiau. Taigi, reikėtų porinių dažnių lentelės su santykiniais dažniais arba procentais. Kaip juos gauti?

Patogumo dėlei susikurkime objektą „lentele2“, į kurį įrašysime paprastus mus dominančių kintamųjų porinius dažnius.

```
lentele2 <- table(duomenys1$K6, duomenys1$S3)
```

Panaudoję funkciją „prop.table()“, gauname santykinius dažnius. Padauginę komandą iš 100, gauname ir procentus. Būtina atkreipti dėmesį, kad šie procentai (ir santykiniai dažniai) yra paskaičiuoti nuo visų stebėjimo atvejų. Kitaip tariant, imtyje viso yra 15,36 procentai žmonių su viduriniu išsilavinimu, kurie apsisprendė dar prieš rinkimų kampaniją, 2,25 procentų žmonių su aukštuoju išsilavinimu, kurie apsisprendė rinkimų dieną, ir taip toliau. Tai leidžia įvertinti bendras proporcijas.

```
prop.table(lentele2 ) # santykiniai porinių dažnių lentelės dažniai
prop.table(lentele2 )*100 # porinių dažnių lentelės procentai
```

Kita vertus, gautos lentelės nieko nesako apie kintamųjų ryšį – pavyzdžiui, gal aukštojo išsilavinimo grupėje yra santykinai daugiau tų, kurie apsisprendė anksti? Pagal šias dvi komandas paskaičiuokite procentus ir panagrinėkite, kuo skiriasi gaunamos lentelės.

```
prop.table(lentele2, 1)*100 # procentai nuo eilutėje esančio kintamojo (pirmojo)
prop.table(lentele2 , 2)*100 # procentai nuo stulpelyje esančio kintamojo (antrojo)
```

Čia prieiname prie labai svarbaus dalyko – procentai porinių dažnių lentelėje skirsis priklausomai nuo to, kurio kintamojo reikšmes imsime kaip bazę (100 procentų)! Pavyzdžiui, pagal pirmąją lentelę, iš tų, kurie apsisprendė dar prieš rinkimų kampaniją, 24 procentai buvo su aukštuoju išsilavinimu. Šioje lentelėje procentai buvo skaičiuoti nuo eilutėje esančio kintamojo: klausimo apie tai, kada apsispręsta. Tačiau pagal antrąją lentelę, 60 procentų, turinčių aukštąjį išsilavinimą, apsisprendė dar prieš rinkimų kampaniją.

Iš kur tokie skirtumai? Procentai yra santykinis dydis. Nors žmonių, kurie turi aukštąjį ir apsisprendė prieš rinkimų kampaniją yra fiksuotas dažnis (115), gauname skirtingus procentus dėl

to, kad žmonių su aukštuoju yra 191, o apsisprendusių prieš rinkimų kampaniją 473. Jeigu skaičiuosime 115 santykinį dažnį nuo 191, tai bus kitas skaičius, negu nuo 473.

Kaip pasirinkti, nuo kurio kintamojo porinių dažnių lentelėje skaičiuoti procentus? Tai priklauso nuo jūsų tyrimo tikslų, ką norite sužinoti. Pavyzdžiui, kiek pensininkų balsavo už partiją, ir kiek partija turi pensininkų savo elektorate, yra du skirtingi dalykai. Sakykime, kad iš viso kaime A balsavo 50 pensininkų. 10 iš jų balsavo už kandidatą X, o mums įdomu, koks buvo jo populiarumas tarp pensininkų. Taigi, 20 procentų pensininkų (10/50*100) balsavo už kandidatą X. Tačiau mums gali būti įdomu ir kandidato X balsų struktūra. Viso šis kandidatas gavo 20 balsų. Tarp jo rinkėjų pensininkai sudaro 50 procentų (10/20*100).

Jeigu yra aiškus priežastinis ryšys, procentus įprasta skaičiuoti nuo nepriklausomo kintamojo. Sakykime, pagaliau norime išsiaiškinti, ar daro išsilavinimas poveikį tam, kada apsisprendžiama už ką balsuoti, ar ne. Visi žingsniai jau matyti, tačiau pakartoti, dirbant su *R*, visada yra naudinga. Pirmuoju žingsniu sukuriame porinių dažnių lentelę. Atkreipkite dėmesį, kad skliaustuose kintamųjų eilė yra svarbi – pirmas kintamasis (K6, apsisprendimas) eis į eilutes, o antras kintamasis (S3, išsilavinimas) į stulpelius. Antruoju žingsniu pritaikome funkciją, kuri paskaičiuoja dažnių lentelėms procentus. Skliaustuose nurodome lentelės, kuriai skaičiuoti proporcijas, pavadinimą. Antrajame argumente tiesiog nurodome kintamojo numerį, nuo kurio skaičiuoti procentus. Antras kintamasis yra išsilavinimas, jis yra nepriklausomas kintamasis (būtent jis teoriškai daro įtaką tam, kada apsisprendžiama), todėl po lentelės pavadinimo įrašome „2“. Padauginame iš 100, kad gautume procentus, o ne santykinį dažnį.

```
lentele2 <- table(duomenys1$K6, duomenys1$S3) # sukuriama lentelė
prop.table(lentele2 , 2)*100  # paskaičiuojami procentai
```

Palyginkite gautus procentus tarp stulpelių. Kokie žmonės yra linkę kiek vėliau apsispręsti, kada balsuoja? Skirtumai nėra dideli, tačiau procentas žmonių, kurie apsisprendė prieš rinkimų kampaniją, yra didžiausias tarp turinčių pradinį ir vidurinį išsilavinimą. Žmonių, kurie apsisprendė rinkimų kampanijos eigoje, yra (santykinai) daugiau tarp tų, kurie turi vidurinį ar aukštąjį išsilavinimą. Beje, šie skirtumai gali būti atsiradę atsitiktinai. Kokia to tikimybė, aprašomoji statistika, deja, nepasakys: tam reiks išvadų statistikos (apie tai rašoma trečiajame skyriuje).

Apibendrinant šį skyrelį, reikėtų žinoti:

- Kuo skiriasi dažnis, santykinis dažnis ir procentas
- Kas yra poriniai dažniai
- Kuo skiriasi eilutės ir stulpelio procentas
- Kaip su R sukurti dažnių lentelę, įrašyti ją į naują objektą
- R funkcijos: „table()“, „sum()“ „colnames()“, prop.table“

2.2 Duomenų padėties charakteristikos

Procentai ne visada yra patogus būdas apibendrinti kintamojo duomenis, ypač jeigu jie surinkti kiekybine skale. Sakykime, turime nedidelę imtį iš septynių elementų, kiek balsų gavo Tėvynės Sąjunga-Lietuvos krikščionys demokratai savivaldos rinkimuose Kaune. 1995 m. gauta 57641 balsas, 1997 m. – 40167, 2000 m. – 23250, 2002 m. – 22303, 2007 m. – 28175, 2011 m. – 23826, 2015 m. – 27969. Pabandykime susivesti šiuos duomenis ir pasidaryti dažnių lentelę.

```
balsai <- c(57641,40167,23250, 22303, 28175, 23826, 27969) # sukuriame kintamąjį
table(balsai) # paskaičiuojame dažnius
```

Gauname, kad visos reikšmės pasikartojo po vieną kartą – realiai tą pačią informaciją, kurią ir turėjome. Duomenų tinkamai apibendrinti nepavyko. Ši imtis gana maža, tačiau panašiai būtų ir su kiekybiniais duomenimis didelėje imtyje.

```
> table(balsai) # paskaičiuojame dažnius
balsai
22303 23250 23826 27969 28175 40167 57641
    1     1     1     1     1     1     1
```

Tokiu atveju, norint apibendrinti duomenis, galima naudoti kitą visiems pažįstamą statistinį metodą – aritmetinį vidurkį. Tai yra skaičių sekos suma, padalinta iš elementų skaičiaus (imties dydžio). Su *R* galima vidurkį gauti skaičiuojant „kalkuliatoriumi“, galima ir panaudoti funkciją „mean()“. Į skliaustus įrašome kintamąjį, kurio vidurkį skaičiuojame.

```
57641+40167+23250+ 22303+28175+ 23826+ 27969/7 # vidurkis su „kalkuliatoriumi“
mean(balsai) # vidurkis su funkcija „mean()“
```

Gauname, kad Kaune konservatoriai 1995-2015 m. per savivaldos rinkimus vidutiniškai gavo 31904,43 balsų. Matematiniu požiūriu, skaičiuojamas tik kiekybinių duomenų vidurkis: vidutinis gaunamų balsų mieste skaičius, vidutinės gyventojų pajamos, vidutinės vyriausybės išlaidos sveikatos apsaugai ir taip toliau. Ar galimos išimtys iš šios taisyklės? Jos daromos socialiniuose moksluose, dirbant su apklausomis, rangų skale matuotais kintamaisiais. Rangų skale matuotų kintamųjų vidurkis gali suteikti informacijos, bet jo reikšmės interpretacija skirsis nuo kiekybinių kintamųjų vidurkio interpretacijos.

Pavyzdžiams, kaip skaičiuoti ir interpretuoti rangų skalės vidurkį, naudosime tą patį 2008 m. porinkiminės apklausos failą. Tačiau šį kartą nuskaitykime jį taip, kad atsakymai duomenų lentelėje (iš tekstinių reikšmių *R* vidurkio neišves) būtų koduoti ne žodžiais, o skaičiais („use.value.labels = FALSE“).

```
duomenys2 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE)
```

Pavyzdžiui gerai tiks kintamieji, kuriais buvo matuotos žmonių simpatijos konkrečioms partijoms. 2008 m. apklausos klausimyne tai yra K10 klausimas: kalbant tiksliau, K10 klausimų matrica. Duomenų faile yra kintamieji K10_1, K10_2 ir kiti: K10_1 atitiks pirmąjį K10 klausimų matricos klausimą (Lietuvos socialdemokratų partijos vertinimas), K10_2 antrą klausimą (Darbo partijos vertinimas) ir taip toliau. Atkreipkite dėmesį, kad kuo didesnis rangas, tuo partija labiau nepatinka: 1 – Labai patinka, 2 – Patinka, 3 – Nei taip, nei ne, 4 – Nepatinka, 5 – labai nepatinka. Nežinantys ir neatsakę apklausoje buvo koduoti kaip „9“.

Panagrinėkime, kaip po 2008 m. Seimo rinkimų žmonės vertino Lietuvos socialdemokratų partiją (LSDP). Iš pradžių paskaičiuokime dažnius.

```
table(duomenys2$K10_1)
```

Gauname tokį pasiskirstymą. Atkreipkite dėmesį, kad yra 43 respondentai, kurie nežinojo ar neatsakė (šie atsakymai žymėti „9“). Skaičiuojant vidurkį, mes negalime įtraukti šių reikšmių, kadangi jos nepatenka į rangų skalę (nežinoti tai nereiškia, jog daugiau ar mažiau patinka ši partija), dirbtinai padidins vidurkį ir padarys jį neprasmingą. Ką daryti?

```
> table(duomenys2$K10_1)

  1   2   3   4   5   9
 16 122 353 371  96  43
```

Egzistuoja keletas išeičių. Viena jų – nurodyti *R* programai, kad reikšmės „9“ bus pakeistos į į trūkstamas ir vėliau funkcijoje „mean()“ nurodyti, kad trūkstamų reikšmių nereikia įtraukti į vidurkio skaičiavimą. Tai gera proga išmokti keletą naujų *R* kalbos simbolių. Pirmas, kurio reikės šioje operacijoje – „NA“, būtent taip R kalboje žymimos trūkstamos, neišmatuotos reikšmės. Antras simbolis yra „= =“, kuris R kalboje žymi tikrąją lygybę (prisiminkite, ženklas „=“ reiškia reikšmių priskyrimą argumentams). Trečias simbolis – laužtiniai skliaustai, kurie programai nurodo, kad operacijas reikėtų pritaikyti tik tam tikriems elementams (reikšmėms). Sąlyga nurodoma laužtinių skliaustų viduje.

Iš pradžių atsargumo dėlei (kad nesugadintume originalių duomenų) sukurkime naują kintamąjį, kuris turės lygiai tokius pačius duomenis, kaip K10_1. Galite pasitikrinti, ar gauname tokią pačią dažnių lentelę.

```
LSDP <- duomenys2$K10_1 # sukuriame analogišką kintamąjį
table(LSDP) # pasitikriname
```

Žemiau šios pastraipos nurodyta komanda mes pakeičiame reikšmes „9“ į trūkstamas reikšmes „NA“. Apie viską paeiliui. Pirma, iš anksčiau žinome, kad R kalboje ženklas „<-„ reiškia į dešinę nuo jo esančios reikšmės priskyrimą į kairėje esantį objektą. Tačiau objektas neprivalo būti naujas! Taigi, jau egzistuojančiam objektui – skaičių sekai “LSDP” – norime priskirti reikšmes „NA“. Tačiau į „NA“ reikia pakeisti tik objekto „LSDP“ elementus, kurie yra lygūs „9“. Tam reikalingi laužtiniai skliaustai šalia objekto pavadinimo, kuriuos įrašoma ši sąlyga. Šią komandą pavertus į paprastą kalbą, sakytume taip – objekte LSDP tik reikšmėms „9“ reikia priskirti naują reikšmę, „NA“.

```
LSDP[LSDP==9] <- NA
```

Pasitikrinkime, ar pavyko.

```
table(LSDP) # R neberodo “9”, kadangi tai yra trūkstamos reikšmės
```

```
table(LSDP, exclude=NULL) # jeigu visgi norime pamatyti, kiek yra NA
```

Dabar jau galime skaičiuoti vidurkį. Papildomas argumentas „na.rm“ reikalingas tam, kad nurodytume, jog vidurkio skaičiavime pašalinamos trūkstamos reikšmes. Kitu atveju funkcija „mean()“ tiesiog neįvykdys užduoties (aktualu tuo atveju, jeigu elementų sekoje yra trūkstamų reikšmių).

```
mean(LSDP, na.rm=TRUE)
```

Gauname, kad LSDP vertinimo vidurkis yra apie „3,42“. Kaip jį interpretuoti? Prisiminkime, kad rangų skalėje skaičių išsidėliojimas tiesiog reiškia tam tikrą tvarką, tačiau ne kiekybinius skirtumus. Todėl, atsižvelgiant į rangų reikšmę, galime teigti, kad po 2008 m. Seimo rinkimų LSDP partija rinkėjų buvo vertinama blogiau, nei neutraliai – tarp „nei taip, nei ne“ ir „nepatinka“.

```
> mean(LSDP, na.rm=TRUE)
[1] 3.426931
```

Yra kitas būdas, kaip paskaičiuoti vidurkį, nepakeičiant kintamajame originalių reikšmių trūkstamomis. Tai galima padaryti, pridedant laužtinius skliaustus ir juose įrašant sąlygą, kad būtų imamos tik konkrečios kintamojo „duomenys2$K10_1“ reikšmės. Šiuo atveju tai būtų reikšmės, kurios yra lygios „5“ arba mažesnės (prisiminkite, kad nežinantys ir neatsakę buvo koduoti „9“, o partijos vertinimo skalė yra tarp „1“ ir „5“). Konsolėje turėtumėte gauti tokį patį vidurkį, kaip ir ankstesniu būdu.

```
mean(duomenys2$K10_1[duomenys2$K10_1<=5])
```

Galime palyginti kelių partijų vertinimo vidurkius – tik tada geriausiai atsiskleidžia, kuo skiriasi rangų vidurkių interpretacija nuo kiekybinių duomenų. Kintamuoju „K10_2“ matuotas Darbo partijos vertinimas (DP), o „K10_3“ – tuometinės rinkimų laimėtojos Tėvynės Sąjungos-Lietuvos krikščionių demokratų (TS-LKD).

```
mean(duomenys2$K10_1[duomenys2$K10_1<=5])  # LSDP vidurkis
mean(duomenys2$K10_2[duomenys2$K10_2<=5]) # DP vidurkis
mean(duomenys2$K10_3[duomenys2$K10_3<=5]) # TS-LKD vidurkis
```

Konsolėje turėtumėte matyti tokį rezultatą. Vienintelės TS-LKD vidurkis yra mažiau negu 3, taigi, ji vienintelė neturėjo vidutiniškai labiau neigiamo įvertinimo. Galime teigti, kad TS-LKD žmonės vertino geriau, negu LSDP ir DP, o pastarąsias partijas mėgo gana panašiai. Čia reikėtų akcentuoti – kadangi tai rangų skalė, galime pasakyti, ar vertinimas yra geresnis ar blogesnis, tačiau negalime objektyviai įvertinti, kiek jis yra blogesnis. Kiekybinio kintamojo atveju, galėtume pasakyti, kiek balsų partija gavo daugiau. Šiuo atveju skirtumas yra gana subjektyvus, o universalių „partijų vertinimo vienetų“ tiesiog negali būti.

```
> mean(duomenys2$K10_1[duomenys2$K10_1<=5])
[1] 3.426931
> mean(duomenys2$K10_2[duomenys2$K10_2<=5])
[1] 3.408058
> mean(duomenys2$K10_3[duomenys2$K10_3<=5])
[1] 2.917277
```

Vidurkis yra dažniausiai naudojama duomenų padėties charakteristika. Tačiau kartais ji gali būti klaidinanti. Sakykime, kad apklausėme 10 žmonių. 9 jų uždirba po 500 eurų per mėnesį, o 10-tasis uždirba 5500. Aritmetinis vidurkis bus lygus 1000 eurų, tačiau jis gerokai iškreips realybę, nes 9 iš 10 atvejų gauna gerokai mažesnį atlyginimą. Vidurkis išsikreipia, nes yra vienas labai išsiskiriantis atvejis (daug didesnis arba daug mažesnis nei visi kiti) – vadinamoji išskirtis (angl. *outlier*). Kad išvengtume nereprezentatyvaus vidurkio, kai imtyje yra išskirčių, rekomenduojama naudoti medianą.

Mediana (Md) yra skaičius, už kurį 50 procentų variacinės eilutės reikšmių yra ne didesnės ir 50 procentų ne mažesnės. Variacinė eilutė tiesiog reiškia imties elementų seką, elementus išdėliojant nuo mažiausio iki didžiausio. Mediana būtų pats šios variacinės eilutės vidurys. Ankstesniame pavyzdyje variacinė eilutė būtų tokia: 500, 500, 500, **500, 500**, 500, 500, 500, 500, 5500. Kai vidurinės reikšmės nėra (nes imties dydis yra lyginis skaičius), tiesiog imamos dvi (paryškintos) vidurinės reikšmės, sudedamos ir padalinamos iš dviejų. Taigi, šiuo atveju mediana yra (500+500)/2 = 500. Variacinėje eilutėje 1, 2, 2, **3**, 4, 4, 5 mediana būtų tiesiog 3, pati vidurinė reikšmė.

Kaip ir vidurkis, mediana gali būti naudojama rangų skale ir kiekybine skale matuotų kiekybinių kintamųjų aprašymui. R programoje medianą galima apskaičiuoti labai panašiai, kaip vidurkį, su komanda „median“. Pavyzdžiui, suraskime medianą LSDP, DP ir TS-LKD vertinimuose. DP mediana yra „4“. Tai reiškia, kad bent 50 procentų žmonių DP vertino „4“ (nepatinka) ir „5“

(nepatinka), nes „4“ yra pats variacinės eilutės (elementų sekos, išdėliotos nuo mažiausios iki didžiausios reikšmės) vidurys. Galite pastebėti, kad nors LSDP ir DP vertinimo vidurkiai buvo kone identiški (3,42 ir 3,4), medianos skiriasi. Prie to grįšime kiek vėliau.

```
median(duomenys2$K10_1[duomenys2$K10_1<=5])  # LSDP mediana
median(duomenys2$K10_2[duomenys2$K10_2<=5]) # DP mediana
median(duomenys2$K10_3[duomenys2$K10_3<=5]) # TS-LKD mediana
```

Paskutinė duomenų padėties charakteristika, kurią verta žinoti – kvantiliai (angl. *percentiles*). Iš pradžių gali atrodyti, kad tai tik teorinė abstrakcija, tačiau kvantiliai naudingi, norint atskirti didžiausių ar mažiausių reikšmių dalį ar suskirstant duomenis į pagrįstus intervalus. Jie gana panašūs į medianą tuo, kad dalija variacinę eilutę į tam tikras dalis. Pavyzdžiui, 95 procentų kvantilis atskirtų 95 procentus didžiausių reikšmių nuo 5 proc. didžiausių. Mediana yra ne kas kita, o 50 procentų kvantilis.

Geriausiai kvantilius suprasti per pavyzdį su nedidele imtimi. 2011 metais Artūras Zuokas su „Taip“ Šeškinės apylinkėse gavo tiek balsų: 13; 14; 14; 16; 16; 16; 17; 17; 17; 17; 17; **18;** 18; 18. Jeigu norėtume suskaičiuoti 80 procentų kvantilį be statistinės programos, turėtume daryti tokius žingsnius: 1) stebėjimų skaičių (šiuo atveju n = 14) dauginame iš 0.8 (q reikšmės); 2) Randame indekso i (i = q ×n), kurio reiks skaičiuojant kvartilio reikšmę, įvertį 11.2; 3) Imame šio skaičiaus sveikąją dalį; 4) Ieškomas kvantilis yra i + 1 variacinės eilutės narys, šiuo atveju 12-tasis variacinės eilutės narys. Šio nario reikšmė 18, taigi 80 procentus kvantilis yra 18. Už šį skaičių 80 procentų variacinės eilutės narių yra nedidesni (tokie patys arba mažesni), o likę – nemažesni (tokie patys arba didesni).

Kvantilius galima paskaičiuoti su *R*, komanda „quantile“. Tiesa, ši funkcija, skirtingai nei praėjusioje pastraipoje pristatytuose žingsniuose, 80 procentų kvantilio nepateiks kaip sveiko skaičiaus. Pirmu žingsniu sukuriame skaičių seką (A. Zuoko balsus Šeškinės apylinkėse 2011 metais). Antru žingsniu paprašome *R* paskaičiuoti 80 procentų kvantilį: pirmas funkcijos „quantile“ argumentas nurodo kintamąjį, trečias pageidaujamą kvantilį. Gauname skaičių „17,4“. Tokios reikšmės sekoje nėra, tačiau ir jį galime lengvai interpretuoti: 20 procentų apylinkių A. Zuokas gavo daugiau (tiek pat) balsų, 80 procentų mažiau (tiek pat), negu 17,4 procentai. Jeigu pridėsime vienetą ir atmesime skaičių po kablelį, kaip ir hipotetiniame pavyzdyje, gausime 18.

```
zuoko_balsai <- c(13, 14, 14, 16, 16, 16, 17, 17, 17, 17, 17, 18, 18, 18)
```

```
quantile(zuoko_balsai, 0.8)
```

Kartais prireikia sužinoti visas pagrindines duomenų padėties charakteristikas vienu metu. Tam patogiausia naudoti funkciją „summary()“, į skliaustus įrašius atitinkamą kintamąjį. Pavyzdžiui (paleiskite kodo lentelę žemiau), paskaičiuokime pagrindines charakteristikas A. Zuoko balsų Šeškinės apylinkėse imčiai.

```
summary(zuoko_balsai)
```

Turėtumėte gauti žemiau esantį rezultatą. *R* pateikia 6 statistikas. Pirmoji („Min.“) yra mažiausioji imties reikšmė. Reiškia, mažiausiai A. Zuokas gavo 13 procentų balsų kažkurioje apylinkėje. Antra statistika yra pirmasis kvartilis („1st Qu.“). Taip vadinamas kvantilis, kuris atskiria pirmuosius 25 procentus variacinės eilutės narių: kitaip tariant, 25 procentai apylinkių A. Zuokas gavo 16 ir mažiau procentų balsų. Kvartiliai – tai trys skaičiai, kurie dalina variacinę eilutę į keturias lygias dalis, po 25 procentus.

Trečia statistika – jau matyta mediana, ketvirtoji – aritmetinis vidurkis („mean“). Statistikos panašios, taigi, panašu, kad nėra apylinkių, kurios iškreiptų vidurkį (tokių, kuriose A. Zuokas būtų gavęs labai daug ar labai mažai balsų). Mediana, beje, kartu yra ir 50 proc. kvartilis. Penktoji pateikiama statistika yra trečiasis kvartilis – A. Zuokas 75 procentų apylinkių gavo mažiau (tiek pat) balsų, nei 17 proc., o likusiose 25 proc. apylinkių – tiek pat arba daugiau. Galiausiai, šeštoji gaunama statistika („Max.“) yra tiesiog didžiausia reikšmė. Daugiausiai Šeškinės apylinkėse buvo gauta 18 procentų balsų.

```
> summary(zuoko_balsai)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13.00   16.00   17.00   16.29   17.00   18.00 
```

Galima „summary()“ pritaikyti ir ranginiams kintamiesiems. Pavyzdžiui, mūsų naudotoje porinkiminėje apklausoje kintamasis „K10_8“ atitinka Tautos prisikėlimo partijos vertinimą. Tuo pačiu pasikartokime darbinio aplanko nustatymą, failo nuskaitymą ir priskyrimą objektui. Galite nusikopijuoti ir paleisti visą šį kodą (jeigu jūsų darbinė direktorija kita, pakeiskite tai „setwd()“ skliaustuose).

```
setwd("C:/R pratimai") # pasirinktinai, galima ir rankiniu būdu per File -> Change dir..
```

```
library(foreign) # paketas apklausų failų įkelimui
duomenys2 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = TRUE, to.data.frame=TRUE) # įkeliame duomenis
summary(duomenys2$K10_8[duomenys2$K10_8<=5]) # kintamojo duomenų padėties charakteristikos
```

Turėtumėte gauti žemiau esantį rezultatą. Minimali reikšmė yra „1“ – žinoma, buvo žmonių, kuriems TPP labai patiko. Pirmasis kvartilis yra „2“: taigi, matome, kad bent 25 procentams žmonių TPP patiko arba labai patiko. Mediana lygi 3, o vidurkis irgi labai panašus – 2,9. Taigi, visumoje TPP buvo vertinama vidutiniškai: galima sakyti, kad bendrai rinkėjams ji nei patiko, nei nepatiko. Tai matosi ir iš trečiojo kvartilio: 25 procentai žmonių TPP vertino „4“ arba daugiau, taigi, šiam segmentui TPP nepatiko arba labai nepatiko. Galiausiai, maksimali reikšmė yra „5“, kas rodo, jog tikrai buvo tokių žmonių, kuriems TPP labai nepatiko.

```
> summary(duomenys2$K10_8[duomenys2$K10_8<=5]) # kintamojo duomenų padėties charakteristikos
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   2.902   4.000   5.000
```

Šiame poskyryje buvo pateikta daug informacijos ir kai kurios komandos galėjo būti kiek komplikuotos: ypač tos, kuriose reikėjo ir „$“, ir „==“ ženklų. Pastarųjų, norint programai nurodyti, kokias reikšmes imame į analizę, išvengti sunku – taigi, prie jų reikėtų priprasti (ypač dirbant su apklausomis, kur dažnai būna rangų skalė ir trūkstamų reikšmių). O štai „$“ ženklo, kuris rodo, kurį elementą (dažniausiai kintamąjį iš duomenų lentelės) mums reikia paimti, galima ir išvengti. „Attach()“ funkcija leidžia „prisegti“ į atmintį tam tikrą duomenų lentelę ir pasako *R*, kad nuo šiol naudosime kintamuosius tik iš jos. Pavyzdžiui, iš naujo (daroma prielaida, kad jau užkrovėte paketą „foreign“) nuskaitykime duomenis, juos pavadinkime „duomenys3“ ir šiai duomenų lentelei pritaikykime komandą „attach()“.

```
duomenys3 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE)
attach(duomenys3)
```

Konsolė nieko papildomo nepraneša, bet jeigu nėra „error“, komandos įvykdytos. Atrodytų, niekas nepasikeitė: tačiau dabar R žino, kad kintamuosius imsime būtent iš šios duomenų lentelės.

Pabandykite paskaičiuoti LSDP ir TS-LKD vertinimo santraukas, tačiau nenaudodami „$“ ženklo. Žinoma, nurodyti, kad skaičiuojant imti tik „5“ lygias ir mažesnes reikšmes, vis dar reikia.

```
summary(K10_1[K10_1<=5]) # LSDP vertinimo santrauka
summary(K10_3[K10_3<=5]) #  TS-LKD vertinimo santrauka
```

Galima „prisegti“ ir daugiau duomenų lentelių. Visgi patartina dirbti tik su viena, kadangi tarp jų gali sutapti kintamųjų pavadinimai. Po to, kai su konkrečia duomenų lentele baigėte dirbti, rekomenduotina ją „atsegti“ su funkcija „detach()“.

```
detach (duomenys3)
```

Nuo dabar R programa kintamųjų lentelėje „duomenys3“ jau nebeieškos. Apibendrinant šį skyrelį, reikėtų žinoti:

- Kodėl vidurkis ne visada tinka apibendrinti duomenis
- Kas yra mediana ir kuo ji naudinga
- Kas yra kvantiliai ir kvartiliai
- R kalbos ženklai „==“ ir „[ ]“
- R funkcijos „mean()“, „median()“, „quantile()“, „summary ()“, „attach()“, „detach()“

2.3 Duomenų sklaidos charakteristikos

Praėjusiame skyrelyje buvo pateiktas pavyzdys, kad vidurkis gali būti apgaulingas. Vienas toks atvejis – kai analizuojama maža imtis ir aptinkama labai didelė arba labai maža reikšmė, išskirtis. Rekomenduota šalia vidurkio naudoti medianą. Tačiau išskirtis nėra vienintelė situacija, kai vidurkis gali būti nereprezentatyvus. Sakykime, jūs išdėstote žmonių pažiūras nuo „0“, kas reiškia kairė, iki „10“, kas reiškia dešinė. Apklausiate dešimt žmonių, iš kurių aštuoni save priskiria viduriukui, tai yra, „5“, vienas save laiko kiek kairiuoju („4“) o kita save laiko kiek dešiniąja („6“). Įsijunkite R ir sukurkite tokį kintamąjį, paskaičiuokite jo vidurkį ir medianą.

```
seka1 <- c(5,5,5,5,5,5,5,5,4,6)
mean(seka1)
median(seka1)
```

Vidurkis ir mediana lygūs „5“, taigi, sakytume, jog vidutinės pažiūros šioje grupėje yra centristinės, apie vidurį. Tai gana tiksliai atspindi situaciją, nes dauguma save vertina būtent „5“, o ir tie, kurie nevertina, yra gana arti centristinių pažiūrų. O dabar pabandykite paskaičiuoti vidurkį ir medianą tokiu atveju, jeigu pusė save įvardija kaip aiškius kairiuosius („0“), o kita pusė laikosi aiškių dešiniųjų pažiūrų („10“).

```
seka1 <- c(0,0,0,0,0,10,10,10,10,10)
mean(seka1)
median(seka1)
```

Gauname vidurkį ir medianą, kurie lygūs. Tačiau šioje imtyje situacija kardinaliai skirtinga, nei buvusioje prieš tai, kur dauguma žmonių iš tiesų buvo centristinių pažiūrų. Antru atveju pusė yra kairėje, pusė – dešinėje. Centristinių pažiūrų, net artimų joms, tiesiog nėra. Tai yra klasikinis atvejis, kai vidurkis visiškai nereprezentatyvus imties atžvilgiu, o ir mediana ne itin padeda. Taip būna, kai duomenys labai skiriasi tarpusavyje. Kiek jie skiriasi imtyje, padeda išmatuoti duomenų sklaidos charakteristikos.

Pagrindinės kiekybinių kintamųjų duomenų sklaidos charakteristikos yra: dispersija (angl. *variance*) ir standartinis nuokrypis (angl. *standard deviation*). Čia iš karto reikėtų atkreipti dėmesį, kad terminas dispersija dažnai naudojamas ir kaip sinonimas reikšmių sklaidai, ne tik kaip konkretus metodas. Jeigu kalbama apie pačią statistiką, tai dispersija rodo duomenų sklaidą apie vidurkį. Ji skaičiuojama kaip vidutinis skirtumų nuo vidurkio (kiek skiriasi konkreti imties reikšmė nuo imties vidurkio) kvadratas, o žymima „s^2“. Jeigu visos kintamojo reikšmės imtyje vienodos, dispersija lygi 0 – kitaip tariant, reikšmių sklaidos tiesiog nėra, jeigu visi duomenys tokie patys.

Panaudokime ankstesniame skyrelyje naudotą imtį su A. Zuoko gautais balsais (išreikštais procentais) 14-oje Šeškinės apylinkių 2011 metais. Dispersiją skaičiuojame taip (šio kodo į R neveskite): iš kiekvieno elemento reikšmės atimame vidurkį (vidutiniškai gauta 236 balsai), sudedame ir padaliname iš imties dydžio. Įprastai dalinama ne iš „n“, bet „n-1“ (taigi, šiuo atveju jeigu imtis yra iš 14 atvejų, daliname iš 13). Gautas skaičius ir yra kintamojo „A. Zuoko balsai Šeškinės apylinkėse“ dispersija.

((116-236) ² + (180 – 236) ² + (232 - 236) ² + (195 - 236) ² + (254 -236) ² + (311 - 236) ² + (196 - 236) ² + (207 - 236) ² + (267 - 236) ² + (193 - 236) ² + (313 - 236) ² +(273-236) ² + (356 -236) ² + (208 -236) ²)/ 13 = 4070.

Dispersija lygi 4070. Kaip ją interpretuoti? Tai padaryti sunkoka, nes čia vienetai yra „balsai kvadratu“ (atkreipkite dėmesį, kad skirtumus nuo vidurkio kėlėme kvadratu tam, kad jie neišsiprastintų). Kaip ir su dažna duomenų sklaidos statistika, ji tampa prasminga palyginus su kita. Pavyzdžiui, turime dvi imtis – A. Zuoko gautus balsus Šeškinės ir Naujamiesčio apylinkėse. Iš pradžių sukurkime kintamuosius ir palyginkime vidurkius. Jeigu pradedate naują mokymosi sesiją, žinoma, įsijunkite R, nusistatykite darbinį aplanką, atsidarykite skriptą ir nusikopijuokite šį kodą. Tada paleiskite jį (patogiausia pažymėjus visą ir paspaudus „Ctrl+R“).

```
balsai_seskine <- c(116, 180, 232, 195, 254, 311, 196, 207, 267, 193, 313, 273, 356, 208)
balsai_naujamiestis <- c(410, 362, 153, 192, 121, 234, 110, 226, 356, 176, 130, 134, 208)

mean(balsai_seskine)
mean(balsai_naujamiestis)
```

Gauname, kad A. Zuokas Šeškinėje ir Naujamiestyje gavo po panašiai balsų – Šeškinėje vidutiniškai 235,78, o Naujamiestyje 216,3. Tačiau kiekvienas, kuris vaikščiojo po Naujamiestį žino, kad tai gana kontrastingas kvartalas: čia yra ir naujų, atrestauruotų namų, kuriuose gyvena daugiau pasiturintys gyventojų, tačiau vis dar nemažai būstų, kuriuose gyvena pensinio amžiaus, mažiau uždirbantys žmonės. Šeškinė yra kiek vienodesnis, sovietinio tipo miegamasis rajonas, skirtumų jame nedaug. O dabar palyginkime abi imtis, tai yra, balsų skaičių sekas. Net ir plika akimi turėtumėte pamatyti, kad Naujamiestyje kiek daugiau kontrastų ir pagal balsavimą už A. Zuoką – pavyzdžiui, yra pora apylinkių, kuriose balsuota 360-410 balsų ribose, bet ir kelios, kuriose balsai 110-130 ribose. Šeškinėje skirtumų, žinoma, irgi yra (apylinkės juk nevienodo dydžio), tačiau jie mažesni.

Palyginkime dispersijas – reikšmių, tai yra balsų, sklaidą apie vidurkį abiejose imtyse (rajonuose). Tam yra paprasta *R* funkcija „variance()“. Kaip ir kitose panašiose, praėjusiame skyrelyje naudose funkcijose („mean()“, „median()“), skliaustuose įrašome kintamųjų, kurių dispersiją norime gauti, pavadinimą.

```
var(balsai_seskine)
```

```
var(balsai_naujamiestis)
```

Turėtumėte gauti žemiau esantį rezultatą. Pirmos komandos pirmas (ir vienintelis rezultatas) yra 4070 (suapvalinus), antrosios (suapvalinus) – 9964. Matome, kad balsų Naujamiestyje dispersija yra du kartus didesnė, negu Naujamiestyje. Taigi, statistiškai pagrindėme, kad ten balsavimas už A. Zuoką 2011 m. buvo įvairesnis. Kitaip tariant, Naujamiesčio imtyje kintamojo „balsai už A. Zuoką“ duomenys skiriasi daugiau, nei Šeškinės imtyje.

```
> var(balsai_seskine)
[1] 4070.335
> var(balsai_naujamiestis)
[1] 9963.731
```

Kaip buvo minėta anksčiau, dispersija turi vieną trūkumą – jos vienetai yra „kvadratu“, sunku interpretuoti net ir lyginant. Tai galima išspręsti labai paprastai, ištraukiant šaknį iš dispersijos ir gaunant bene dažniausiai naudojamą duomenų sklaidos charakteristiką, standartinį nuokrypį (žymimas tiesiog „s“ arba „sd“). Dirbant su R, standartinį nuokrypį galima paskaičiuoti labai paprastai, su funkcija „sd()“.

```
sd(balsai_seskine)
```

Balsų pasiskirstymo po šeškinės apylinkes standartinis nuokrypis yra 63,8. Skirtingai nei dispersija, standartinis nuokrypis matuojamas tokiais pačiais vienetais kaip ir duomenys. Taigi, ši statistika realiai mums sako, kad standartiškai nuo vidurkio (236 balsai) apylinkės nukrypsta per 63,8 balso. Vėlgi, reikėtų palyginti, kad sužinotume kažką daugiau.

```
sd(balsai_naujamiestis)
```

Naujamiestyje standartinis nuokrypis yra 99,8. Tai, kad duomenų sklaida šioje imtyje didesnė, jau žinojome, tačiau dabar galime tai įvertinti su tiesiogiai interpretuojamais vienetais – čia standartiniai skirtumai nuo vidurkio yra didesni beveik per 40 balsų. Taigi, nors balsų tiek Šeškinėje, tiek Naujamiestyje gauta panašiai, pastarojoje apygardoje yra didesni kontrastai tarp apylinkių, o pats vidurkis (kiek vidutiniškai gaunama balsų) – mažiau reprezentatyvus, nei Šeškinėje. Būtent tai mums ir parodo duomenų sklaidos charakteristikos, standartinis nuokrypis ir dispersija. Ypač lyginant vidurkius imtyse, labai rekomenduotina naudoti šalia vidurkio ir šias

statistikas (bent jau standartinius nuokrypius). Beje, šiuos skirtumus įmanoma ir kartais labai rekomenduotina vizualizuoti – apie tai bus kalbama kitame skyrelyje.

Tačiau prieš tai reikėtų aptarti ne tik kiekybinę, bet ir kitas skales. Įprastai skaičiuojama tik kiekybinių duomenų dispersija ir standartinis nuokrypis. Rangų skale matuotiems kintamiesiems daromos išimtys, kaip ir skaičiuojant vidurkį. Šiaip pageidautina, kad rangų skalėje būtų bent 5 reikšmės. Taip pat reikia prisiminti, kad čia interpretacija bus kita. Lyginant rangų skalės duomenų standartinius nuokrypius galėsime pasakyti, kur duomenys skiriasi daugiau, tačiau ne per kiek daugiau (prisiminkite – rangų skalė subjektyvi ir tiesiog išdėlioja objektus pagal tam tikrą tvarką!).

Pavyzdžiui, galime įvertinti, kaip skiriasi Lietuvos respondentų atsakymai į klausimą apie jų politines pažiūras. Atsiverskite 2008 m. porinkiminės apklausos klausimyną ir susiraskite klausimą „K13“. Jame prašoma respondento į skalėje nuo „1“ iki“ 9“, kur „1“ reiškia kraštutines kairiąsias, o „9“ kraštutines kairiąsias pažiūras. „0“ reiškia, kad respondentas negalėjo savęs skalėje identifikuotis, taigi, logiška, kad šių reikšmių į vidurkio ar standartinio nuokrypio skaičiavimus neįtrauktume.

```
setwd("C:/R pratimai") # priminimas: nusistatome darbinį aplanką
library(foreign)  # užkrauname “sav.” failų nuskaitymo paketą
apklausa <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE)  # nuskaitome failą be reikšmių pavadinimų
```

Padarykime, kad *R* kintamuosius imtų iš duomenų lentelės “apklausa” ir paskaičiuokime kintamojo dažnius.

```
attach(apklausa) # “prisegame” duomenų lentelę su apklausos atsakymais
table(K13) # jeigu duomenų lentelė “prisegta”, ženklo „$“ nereikia
```

Matome, kad nemažai respondentų (viso 288) negalėjo priskirti savęs jokioms politinėms pažiūroms kairės ir dešinės skalėje. Deja, jeigu norime matyti vidurkį ar standartinį nuokrypį, šios svarbios grupės į skaičiavimus negalime įtraukti. Jeigu taip padarysime, vidurkis bus nenatūraliai mažas. Žinoma, pristatant tyrimo rezultatus, būtina paminėti, kad didelė dalis lietuvių tiesiog negali rasti savęs tokioje skalėje.

```
mean(K13)
```

Turėtumėte gauti tokius rezultatus. Vidurkis yra akivaizdžiai iškreipiamas reikšmių “0”. Ką daryti?

```
> mean(K13)
[1] 4.008991
```

Kaip ankstesniame skyriuje, su laužtiniais skliaustais nurodome sąlygą, kad *R* į vidurkio skaičiavimą įtrauktų tik reikšmes, kurios atitinka dvi sąlygas: 1) yra lygios „1“ arba didesnės; 2) lygios „9“ arba mažesnės. Aišku, šį tikslą buvo galima pasiekti ir be antrosios sąlygos (nėra reikšmių, didesnių nei „9“, tą matėme iš dažnių lentelės). Tačiau tai tinkamas momentas parodyti ženklą “&“, kuris reiškia „ir“, panašiai, kaip *Excel* programoje komanda AND. Programai sakome: paskaičiuok vidurkį iš „K13“ kintamojo, tačiau (laužtiniai skliaustai) imk tik tas reikšmes, kurios lygios ir didesnės už „1“ ir tas, kurios lygios ir mažesnės nei „9”.

```
mean(K13[K13>=1 & K13<=9])
```

Gauname vidurkį „5,6“ (suapvalinus). Žinoma, tą patį gautume ir be antrosios sąlygos.

```
mean(K13[K13>=1])
```

Paskaičiuokime daugiau duomenų padėties charakteristikų. Matome, kad mediana gana panaši į vidurkį. Be to, pirmasis kvartilis sutampa su mediana – tai reiškia, kad kairiųjų apklausoje buvo gana mažai, jeigu jau pirmi 25 procentai variacinės eilutės narių turėjo „5“ (pats vidurys, centristinės pažiūros) ir kairesnes pažiūras.

```
summary(K13[K13>=1])
```

Paskaičiuokime ir standartinį nuokrypį. Jis lygus „1,94“, taigi vidutiniškai nuo vidurkio nukrypstama per beveik du rangus.

```
sd(K13[K13>=1])
```

Galime palyginti politinių pažiūrų standartinį nuokrypį tarp vyrų ir moterų. Turėtumėte gauti, kad pažiūrų kintamojo duomenys skiriasi panašiai tiek vyrų, tiek moterų imtyse: duomenys kone identiški.

```
sd(K13[K13>=1 & S1==1]) # kintamasis „S1“ yra lytis, vyrai žymėti „1“
sd(K13[K13>=1 & S1==2]) # kintamasis „S2“ yra lytis, moterys žymėtos „2“
```

Užduotį kiek pasunkinkime. Sakykime, norime išsiaiškinti du dalykus: 1) ar pagal vidutines pažiūras skiriasi žmonės, kurie apsisprendžia seniai prieš rinkimus ir tie, kurie apsisprendžia per rinkimų kampaniją; 2) kurioje iš šių dviejų grupių nuomonės skiriasi. Apsisprendimą matuoja kintamasis „K6“. Jame „1“ žymėti (galite pažiūrėti klausimyną) žmonės, kurie už ką balsuos apsisprendė seniai rinkimuose, o „2“ žymėti tie, kurie apsisprendė rinkimų kampanijos eigoje. Pabandykime paskaičiuoti vidurkius.

```
mean(K13[K13>=1 & K6==1])
mean(K13[K13>=1 & K6==2])
```

Gauname konsolės rezultatą „NA“. Tikriausiai pamenate, kad tai reiškia trūkstamą reikšmę. Tikėtina, kad bandome skaičiuoti vidurkį įtraukdami būtent tokias reikšmes (prisiminkite, juk dalis žmonių apskritai neatsakė klausimo, kada apsisprendė, nes nebalsavo). Tai galime išspręsti gana paprastai, įvedant jau anksčiau matytą argumentą, kad skaičiuojant vidurkį, ignoruoti trūkstamas reikšmes „NA“.

```
mean(K13[K13>=1 & K6==1], na.rm=TRUE) # apsisprendusių prieš kampaniją pažiūrų vidurkis
mean(K13[K13>=1 & K6==2], na.rm=TRUE) # apsisprendusių per kampaniją pažiūrų vidurkis
```

Gauname, kad apsisprendusių dar prieš rinkimų kampaniją vidutinės pažiūros yra dešinesnės, vidurkis lygus 5,9. Tų, kurie apsisprendė per kampaniją, vidurkis yra arčiau tikrojo centro, lygus 5,2. O kokie bus standartiniai nuokrypiai? Skliaustuose komandos mums tinka, reikia tik pakeisti funkcijos pavadinimą.

```
sd(K13[K13>=1 & K6==1], na.rm=TRUE)
sd(K13[K13>=1 & K6==2], na.rm=TRUE)
```

Gauname, kad pirmosios grupės standartinis nuokrypis yra didesnis (2,14), negu antrosios (1,63). Tai logiška, kadangi vėliau apsisprendusieji neturėtų pasižymėti aiškiomis pažiūromis. Mūsų duomenys rodo: pirma, jų vidurkis yra labai arti skalės centro, antra, skirtumai tarp respondentų atsakymų yra gana nedideli (standartinis nuokrypis). Atkreipkite dėmesį, kad abi šios statistikos suteikia naudingos informacijos. O štai tarp žmonių, kurie apsisprendė prieš kampaniją, pažiūrų vidurkis yra dešinesnis, tačiau ir standartinis nuokrypis didesnis – nieko keisto, nes prieš 2008 m. kairiųjų populiarumas buvo gerokai sumažėjęs, bet tai nereiškia, kad Lietuvoje neliko aiškias kairias pažiūras deklaruojančių žmonių. Vėlesniame skyriuje šias dvi grupes vizualizuosime, ir šie skirtumai bus dar akivaizdesni.

Kartais, vertinant mažas rangų skales (pavyzdžiui, tik iš keturių rangų), rekomenduojama naudoti ne standartinį nuokrypį, o skirtumą tarp pirmojo ir trečiojo kvartilių. Tokia statistika parodo vidurinių 50 procentų respondentų (žinoma, išdėliotų pagal atsakymų reikšmes nuo mažiausios iki didžiausios) nuomonių „plotį“. Pavyzdžiui, galime pritaikyti kvartilių skirtumą 2008 m. klausimyno K15 matricos klausimams apie vertybinius teiginius. Čia yra tik keturi rangai, nuo aiškaus pritarimo („1“ = Taip) iki aiškaus nepritarimo („4”=Ne). “9” reiškia, kad respondentas nežino, šių reikšmių mes negalime įtraukti į kvartilių (ar vidurkio, medianos) skaičiavimą.

Paskaičiuokime santraukas dviem kintamiesiems (tikslias formuluotes rasite pačiame klausimyne): 1) pritarimui progresiniams mokesčiams (K15_1); 2) pritarimui abortų uždraudimui (K15_4). Už laužtinių skliaustų įvedame sąlygą

```
summary(K15_1[K15_1<=4]) # pritarimo progresiniams mokesčiams statistika
summary(K15_4[K15_4<=4]) # pritarimo abortų uždraudimui statistika
```

Pirmojo kintamojo atveju pirmas kvartilis yra lygus „1“, o antras „3“. Skirtumas lygus „2“ (atimkime iš trijų vieną), o interpretacija būtų tokia, kad viduriniai 50 procentų respondentai apima ir tuos, kurie progresiniams mokesčiams sako „Taip“, ir tuos, kurie sako „Tikriausiai ne“. Kalbant apie abortus, pirmas kvartilis yra „3“, o trečias – „4“. Taigi, kvartilių skirtumas lygus „1“, kas čia reiškia: viduriniai 50 procentų respondentų apima tik tuos, kurie sako „tikriausiai ne“ ir „ne“. Išvada tokia, kad ties abortais žmonių nuomonės išsiskiria mažiau, negu ties progresiniais mokesčiais.

Apie duomenų sklaidos charakteristikas, taip pat ir kvartilių skirtumą, dar bus kalbama kitame poskyryje – vizualizacijos padės geriau įsigilinti į čia išdėstytus aspektus.

Apibendrinant šį skyrelį, reikėtų žinoti:

- Kam reikalingos duomenų sklaidos charakteristikos
- Kaip skaičiuojama dispersija ir standartinis nuokrypis
- Kaip interpretuoti duomenų sklaidos charakteristikas
- R kalbos ženklas „&“
- R funkcijos „variance()“, „sd()“

2.4 Pagrindiniai grafikai kiekybinėje analizėje

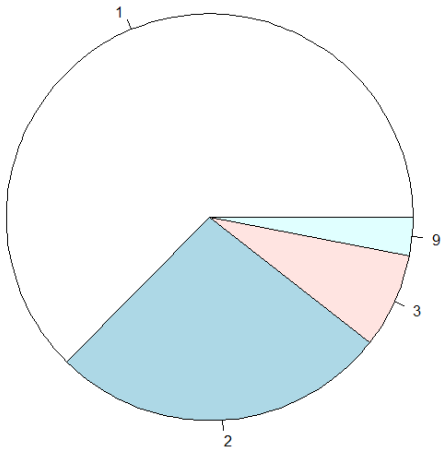
Šiame skyriuje pateikiami keli grafikai, kurie gali būti labai naudingi nagrinėjant duomenis ir suprantant jų struktūrą. Jie visų pirma padeda geriau suprasti anksčiau aptartus metodus ir sąvokas, pavyzdžiui, bendrą reikšmių sklaidą, medianą, kvartilių skirtumą. Šiame skyriuje orientuojamasi į vieno kintamojo duomenų vizualizaciją, o ne ryšio tarp jų perteikimą (tam skirtas paskutinis, trečiasis šios metodinės medžiagos skyrius, kuriame taip pat pateikiama grafikų pavyzdžių).

Pradėkime nuo paprasčiausio pavyzdžio, kaip *R* veikia grafikų kūrimas. Šiame skyrelyje vėl naudosime duomenis iš 2008 m. porinkiminės apklausos. Nusistatykite darbinę direktoriją, nusiskaitykite failą.

```
setwd("C:/R pratimai") # priminimas: nusistatome darbinį aplanką
library(foreign)  # užkrauname “sav.” failų nuskaitymo paketą
apklausa <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE)  # nuskaitome failą be reikšmių pavadinimų
attach(apklausa) # prisegame duomenų lentelę
```

Sukurkime paprastą „pyrago“ formos grafiką, ne kartą matytą įvairiose prezentacijose ar kurtą su Excel. Kaip taisyklė, pyrago formos grafika tinka tik nominaliai skalei ir kai yra nedaug įmanomų reikšmių. Kitu atveju, tokia vizualizacijos forma tampa visiškai nereprezentatyvi, o kalbant apie rangų skalės kintamuosius, jų dažnius geriau vaizduoti stulpelių grafiku : taip geriau matosi kaupiamieji dažniai. Pabandykime padaryti grafiką jau ne kartą analizuotam kintamajam „K6“, apsisprendimo už ką balsuoti laikas.

```
lentele_k6 <-table(K6) # sukuriame dažnių lentelę
pie(lentele_k6) # sukuriame pyrago grafiką pagal dažnių lentelę
```



Gautas grafikas nėra itin išvaizdus – nesimato procentų, spalvos gana blankios. R šiuos dalykus galima itin lengvai koreguoti. Pirma išsiaiškinkime, kokie yra kiekvienos kategorijos procentiniai dažniai.

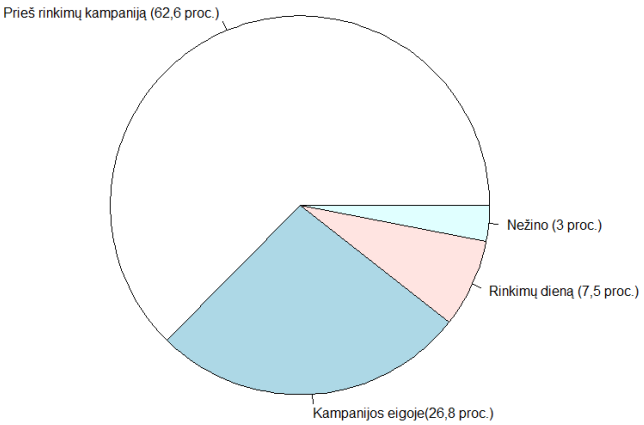
```
prop.table(lentele_k6)
```

Sukurkime tekstinę elementų seką, atitinkančią kategorijų pavadinimus ir su procentais (žr. rezultatą viršuje esančios komandos) skliaustuose.

```
pavadinimai <- c("Prieš rinkimų kampaniją (62,6 proc.)", "Kampanijos eigoje(26,8 proc.)",
"Rinkimų dieną (7,5 proc.)", "Nežino (3 proc.)")
```

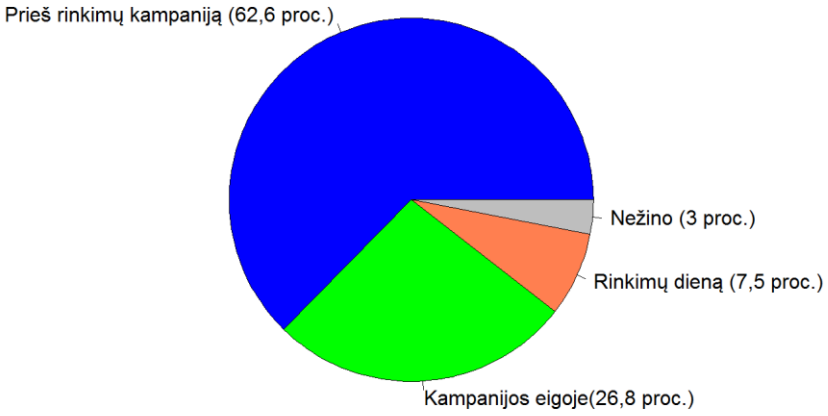
Papildome funkciją argumentu „labels“, kuriam priskiriame reikšmę „pavadinimai“. Tai ką tik mūsų sukurta kategorijų pavadinimų seka.

```
pie(lentele_k6, labels=pavadinimai)
```



Turime pavadinimus, tačiau spalvos vis dar blankios, o patys pavadinimai vos įžiūrimi. Pridėkime du argumentus. Pirmas yra „cex“, jis reguliuoja teksto dydį – galite paeksperimentuoti su šiuo argumentu, įrašykite į jį a „3“, „5“, „0.5“ ir taip toliau. Argumentas „cols“ reguliuoja spalvas – tai elementų seka, sudaryta iš spalvų pavadinimų. Atkreipkite dėmesį, kad spalvų skaičius turi atitikti kategorijų skaičių, o eiliškumas – kategorijų eiliškumą (aišku, jeigu norite, kad spalvos atitiktų jūsų norimas kategorijas. R pateikia labai daug skirtingų spalvų, ne tik bazines, jų pavadinimus galite rasti čia: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf.

```
pie(lentele_k6, labels=pavadinimai, cex=2, col=c("blue", "green", "red", "grey"))
```



Žinoma, kalbant apie grafikus, *R* yra reikalingiausias ne pyrago ar stulpelių grafikams. Pastaruosius, jeigu paprastus, galima lengvai „pasigaminti“ su šios nuorodos pagalba: http://www.statmethods.net/graphs/bar.html. Visgi tokias vizualizacijas žmonės įpratę daryti su *Excel* (jeigu žinomi procentai, o juos paprasta pasidaryti su *R*). Žinoma, įpratus dirbti su R, atsiveria daugiau galimybių ir vaizdo manipuliacijų – metodinės medžiagos pabaigoje pateikiama rekomenduojama literatūra. Tačiau šiame skyriuje koncentruosimės ties dalykais, kurių dažniausiai su *Excel* nedarome.

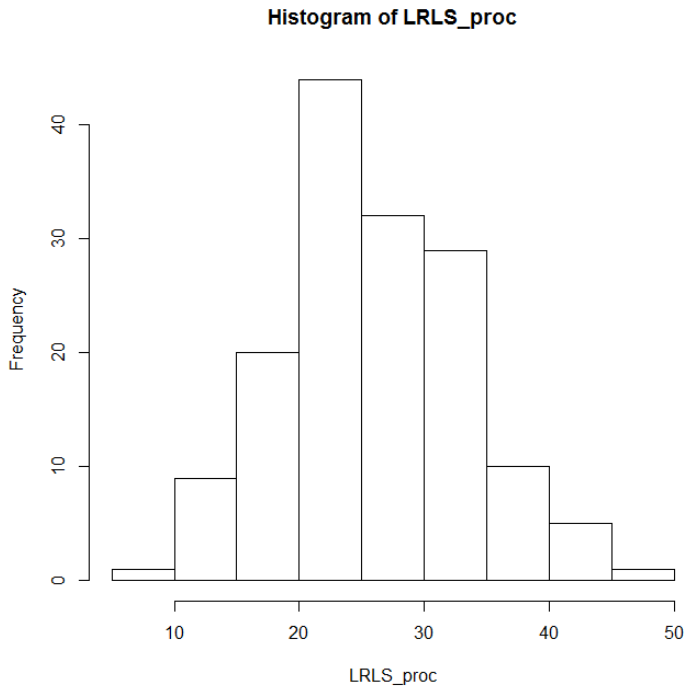
Pirmas labai svarbus, gal net svarbiausias grafikas, kurį reiktų žinoti įvade į kiekybinius metodus, yra histograma. Histograma atskleidžia ranginio ar kiekybinio kintamojo duomenų pasiskirstymo dažnius pagal atitinkamus intervalus ir yra vienas pirmesnių informacijos šaltinių apie reikšmių sklaidą. Paprastai kalbant, histograma rodo reikšmių tankį: kuo dažniau pasikartoja reikšmė (reikšmių intervalas), tuo jos (reikšmių intervalo) stulpelis yra aukštesnis. Pavyzdžiui, pasižiūrėkime, kaip 2015 m. Vilniaus apylinkėse (iš viso 150) pasiskirstė Lietuvos Respublikos liberalų sąjūdžio balsai. Tačiau tam prireiks naujų duomenų.

```
detach(apklausa) # kol kas nusekime apklausą, bet vėliau jos dar prireiks
 vilnius <- read.csv("Partijos_Vilnius_2015.csv", header=TRUE, sep=";", dec=",") # nuskaitome kitą failą su rinkimų rezultatų Vilniuje duomenimis
attach(vilnius) # prisekime Vilniaus rinkimų duomenis
colnames(vilnius) # galite pažiūrėti, kokius turime kintamuosius
```

Dabar jau galime pasinaudoti funkcija „hist()“ ir nubraižyti histogramą.

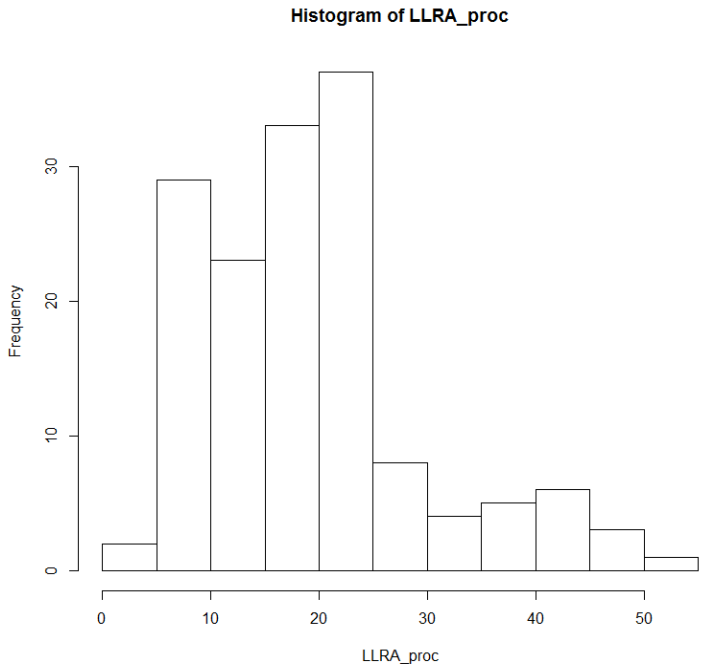
```
hist(LRLS_proc)
```

Horizontali skalė rodo įmanomas kintamojo reikšmes nuo mažiausios iki didžiausios: nuo kiek mažiau nei 10 gautų procentų (buvo ir tokių apylinkių) iki beveik 50 (būta ir net tokių sėkmingų). Stulpelių aukštis ir vertikali skalė rodo, koks yra reikšmių pasikartojimas atitinkame intervale. Galima pastebėti, kad daugiausiai apylinkių yra intervale tarp 20 ir 30 procentų balsų reikšmių, o tokių apylinkių, kuriose gauta arba labai daug, arba labai mažai balsų, tankis mažesnis. Šis paskirstymas gana panašus į normalųjį, simetrišką skirstinį, dar vadinamąjį Gauso pasiskirstymą. Apie jį šioje metodinėje medžiagoje nebus kalbama, bet atkreipkite dėmesį į tai, kad gilinantis į statistiką, apie jį anksčiau ar vėliau išmokti reikės.



O dabar nubraižykime Lietuvos lenkų rinkimų akcijos (LLRA) balsų Vilniaus apylinkėse histogramą.

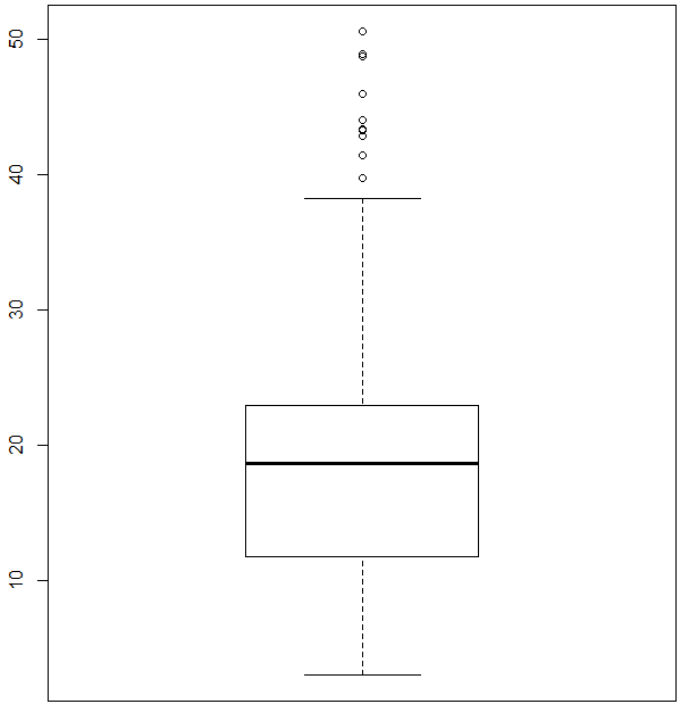
```
hist(LLRA_proc)
```



Pasiskirstymas visiškai kitoks: dauguma reikšmių susispietusios ties 25 ir mažiau riba ir gerokai daugiau, nei LRLS atveju, apylinkių, kuriose gauta apie 10 proc. ir mažiau balsų. Nieko keisto – LLRA elektoratas, nors partija Vilniuje yra gana populiari, yra gana koncentruotas, o lietuviškose apylinkėse partijai gauti daug balsų tiesiog nėra šansų. Reikėtų pastebėti, kad vizualiai reikšmių sklaidos nebūtume galėję taip gerai įsivaizduoti net ir su vidurkiu ir standartiniu nuokrypiu. Histograma iš tiesų yra labai svarbus instrumentas, bandant išsiaiškinti visų pirma kiekybinių duomenų struktūrą.

Dar kitas svarbus grafikas, rečiau (bent jau nei pyrago ar stulpelių grafikai) sutinkamas populiariojoje analitikoje – tai garsaus statistiko Johno Tukey išrastas dėžinis (angl. *boxplot*) arba, dar kitaip, ūselinis (angl. *box and whiskers plot*) grafikas. Šis grafikas yra vienas geriausių grafinės analizės išradimų, nes į jį telpa beveik visos pagrindinės duomenų aprašomosios charakteristikos: galime pamatyti ir didžiausią, ir mažiausią reikšmę, kvartilius, medianą, kvartilių skirtumą, išskirtis. Su R dėžinis grafikas gaunamas labai paprastai, su funkcija „boxplot()“. Pavyzdžiui, pasižiūrėkime LLRA balsų paskirstymą, naudodami būtent tokį grafiką.

boxplot (LLRA_proc)

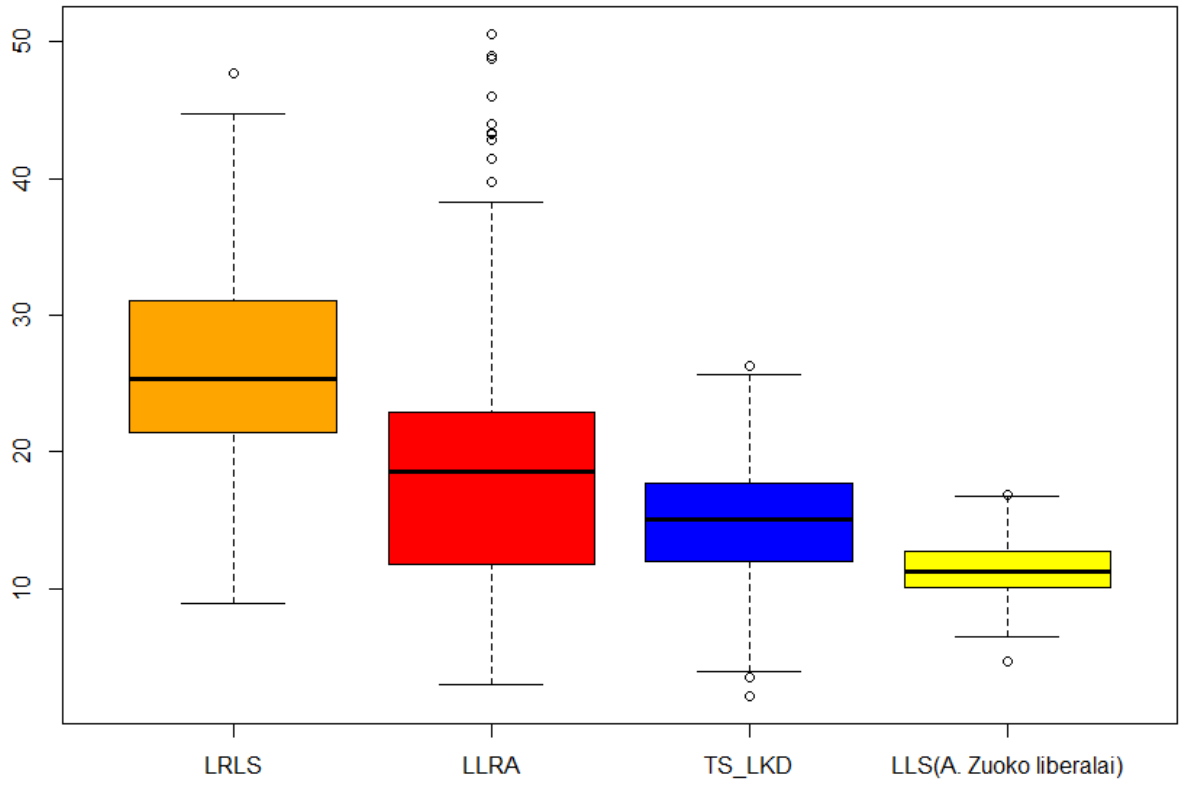


Kaip šį grafiką interpretuoti? Pradėkime nuo apačios. Pačio apatinio ūselio riba yra pati mažiausia reikšmė – LLRA atveju ji yra gerokai žemiau net ir už 10 procentų. Dėžutės apatinė riba yra pirmasis kvartilis, o paryškinta linija dėžutės viduryje (ji gali būti ir labai arti vieno ar kito kvartilio) – mediana. Matome, kad LLRA balsų mediana yra kiek mažiau nei 20 procentų. Viršutinė dėžutės riba yra trečiasis kvartilis – taigi 75 procentai reikšmių patenka žemiau šios ribos. Viršutinio ūselio riba – maksimali reikšmė, o papildomi burbuliukai reiškia išskirtis, nenatūraliai besiskiriančius stebėjimo atvejus. Standartinė taisyklė, kad dėžiniame grafike stebėjimo atvejai vaizduojami atskirai nuo ūselių tada (taigi, laikomi išskirtimis), kai jie yra daugiau nei 1,5 karto didesni negu trečiasis kvartilis.

Itin naudinga sudėti kelių lyginamų kintamųjų (arba imčių) dėžinius grafikus į vieną ir juos palyginti. Pavyzdžiui, apačioje esančios komandos leidžia viename grafike pavaizduoti visų keturių daugiausiai balsų Vilniuje gavusių partijų balsų pasiskirstymus apylinkėse: 1) pirma komanda sukuria pavadinimų kintamąjį/seką (kaip ir ankstesniame pavyzdyje su pyrago grafiku); 2) antra komanda sukuria spalvų seką, kurią galėsime nurodyti funkcijoje; 3) trečia komanda sukuria patį grafiką, kuriame pirma nurodome kintamuosius, kurių dėžiniai grafikai bus atvaizduojami, tada seka pavadinimų ir spalvų argumentai.

```
pavadinimai_vilnius <- c("LRLS", "LLRA", "TS_LKD", "LLS(A. Zuoko liberalai)") # sukuriame seką pagal partijų pavadinimus
spalvos_vilnius <- c("orange", "red", "blue", "yellow") # sukuriame seką spalvoms
boxplot (LRLS_proc, LLRA_proc, TS_LKD_proc, LLS_proc, names=pavadinimai_vilnius, col=spalvos_vilnius) # grafikas
```

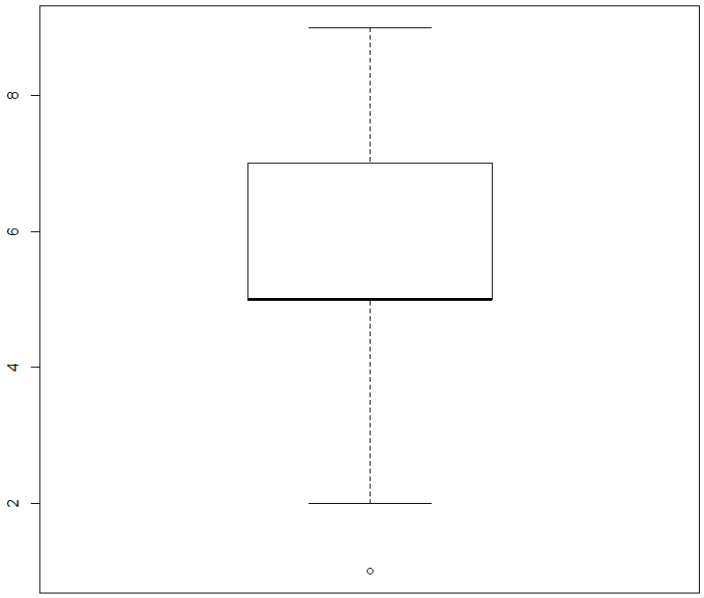
Iš šio grafiko galime pamatyti išties daug. Pirma, medianų ir pačių grafikų dėžučių aukštis parodo partijų rangavimą pagal vidutinę sėkmę rinkimuose. Antra, matome kad A. Zuoko LLS (liberalai) buvo partija, kurios pasirodymų sklaida buvo itin maža (geltonos dėžutės „plotis“). Trečia, LLRA buvo partija, kurios pasirodymuose aptikta daugiausia išskirčių – daugiausiai tautinių mažumų apgyvendintos apylinkės. Galima ir daug kitų pastebėjimų – dėžinis grafikas iš tiesų yra labai informatyvus. Jį galima naudoti ir rangų skale matuotiems kintamiesiems. Tik svarbu, kaip ir skaičiuojant rangų vidurkį, turėti galvoje atitinkamą interpretaciją.



Visų pirma nusekime Vilniaus rinkimų duomenis ir vėl prisekime apklausą. Žinoma, jeigu esate ją nuskaitę iš duomenų failo. Jeigu ne – grįžkite į skyrelio pradžią, ten rasite tam reikalingas komandas. Tada nurodome padaryti dėžinį grafiką iš K13 kintamojo – tai jau anksčiau mūsų analizuota kairės-dešinės pažiūrų skalė. Nurodome, kad imtų tik reikšmes, didesnes nei „1“ (taip neįtraukiam reikšmių „0“, kas reiškia, jog žmogus negali savęs rasti šioje skalėje).

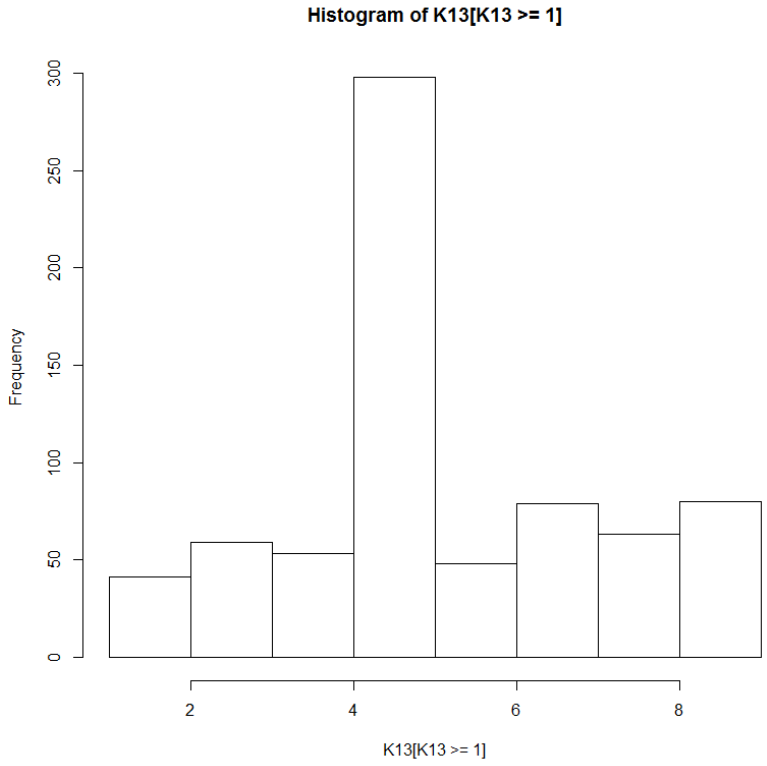
```
detach(vilnius) # nusegame Vilniaus rinkimų rezultatus
attach(apklausa) # vėl prisegame
boxplot(K13[K13>=1]) # kairės-dešinės skalės dėžinis grafikas
```

Tai atvejis (apie tai jau kalbėjome praėjusiuose poskyriuose), kai pirmasis kvartilis sutampa su mediana. Taigi, pirmųjų 50 procentų atsakiusių į šį klausimą pažiūros yra 5 arba mažiau, taigi, centristinės. Kita vertus, tą patį galima pasakyti ir apie pirmus 25 procentus atsakiusiųjų, kas signalizuoja, jog apklausoje tikrai buvo daug tokių, kurie save identifikavo „5“. Įdomu tai, kad kraštutinės kairiosios pažiūros („1“) šioje apklausoje buvo išskirtis: tai reiškia, jog save taip identifikuojančių 2008 m. Lietuvoje beveik nebuvo.



Galime nubraižyti ir šio kintamojo histogramą. Iš jos puikiai matosi, kad Lietuvoje dominavo centristinės pažiūros. Tai praėjusiame skyriuje matėme ir iš gana nedidelio standartinio nuokrypio.

```
hist(K13[K13>=1])
```



Apibendrinant šį skyrelį, reikėtų žinoti:

- Kas yra histograma ir dėžinis grafikas
- Kokios yra sudedamosios dėžinio grafiko dalys
- R funkcijos „pie ()“, „hist()“, „boxplot()“
- Kaip su R pakeisti grafiko (pyrago, dėžinio) spalvą, pavadinimus

3. Įvadas į išvadų statistiką

Rimtas matematikas pasakytų, kad iki šiol statistikos mes nesimokėme – tiesiog pasižaidėme su duomenimis, išmokome juos truputį patyrinėti. Daug statistikos vadovėlių yra skirti išvadų statistikai, mokslui, kuris leidžia iš turimos imties apibendrinti tam tikrus pastebėjimus visai populiacijai. Pavyzdžiui, turime apklausos duomenis apie tai, už ką žmonės balsavo ir koks yra jų išsilavinimas. Išvadų statistika ne tik įgalins mus suprasti, ar tarp šių dviejų dalykų mūsų duomenyse yra ryšys, bet ir pasakys, su kokiu patikimumu apie tai galime (negalime) apibendrinti visam elektoratui (populiacija).

Taigi, skirtingai nei aprašomoji statistika, kuri orientuojasi į imties duomenų sisteminimą, išvadų statistika tikrina hipotezes. Štai pora hipotezių, kurias būtų galima tikrinti statistiniais metodais, tipų: 1) Egzistuoja ryšys tarp veiksnių X ir Y: Zuoko balsai Vilniaus apylinkėse tiesiogiai koreliuoja su konservatorių balsais ; 2) Populiacija 1 ir Populiacija 2 skiriasi pagal požymį Z: Vakarų demokratijos skiriasi nuo pokomunistinių valstybių pagal pasitikėjimą partijomis.

Be imties (surinkti duomenys apie tam tikrą skaičių stebėjimo atvejų) ir populiacijos (stebėjimo atvejų visuma, kuriai norime apibendrinti tyrimo rezultatus), yra dar kelios sąvokos, kurias turime žinoti prieš keldami koją į išvadų statistikos principus ir metodus. Viena jų yra nulinė hipotezė (žymėsime ją H0): ji prieštarauja tyrimo hipotezei ir rodo apie tiriamo efekto (skirtumo, ryšio, įtakos) nebuvimą. Įdomu tai, kad išvadų statistikos metodų taikymo tikslas yra išsiaiškinti, ar galima ją atmesti. Tyrimo hipotezė (H1) yra teiginys, kurį siekiama patvirtinti (pagrįsti) empiriniu tyrimu.

Kitos dvi sąvokos – pirmoj ir antrojo tipo klaidos. Pirmojo tipo klaida yra labai svarbi, ji reiškia, kad nors tyrimo metu hipotezė buvo patvirtinta, iš tikrųjų ši išvada buvo klaidinga ir populiacijoje aptikto ryšio (skirtumo) nėra. Antrojo tipo klaida nėra tokia pavojinga akademiko reputacijai, nes ji

daugiau rodo atsargumą, kai hipotezė atmetama, nors iš tiesų ji populiacijoje teisinga. Būtent dėl pirmojo tipo klaidos pavojaus mums reikia suprasti, kas yra statistinis reikšmingumas, kam skirtas pirmasis skyrelis. Toliau antrajame skyriuje pateikiami keli paprasčiausi išvadų statistikos metodai – ryšio matai. Trečiajame skyrelyje pateikiami populiacijų (vidurkių) lyginimo metodų pavyzdžiai. Pastarasis skyrelis ir baigia šią metodinę medžiagą – aptarus bazinius būdus, kaip galima statistiškai įvertinti ryšį tarp dviejų kintamųjų.

3.1 Statistinis reikšmingumas: kas bendro tarp lošimo kauliuko ir apklausos?

Ar vykdome eksperimentą, ar atliekame apklausą, ar analizuojame duomenis iš rinkimų – vienas esminis klausimas visiems šiems tyrimams yra bendras: ar mūsų gautas rezultatas yra atsitiktinis, ar ne? Pavyzdžiui, pabandykite mesti monetą tris kartus. Sakykime, visus kartus gavote herbą. Ar tai rodo, kad jūs esate puikus herbo metikas? Tikriausiai ne, toks rezultatas gautas tiesiog atsitiktinai – nereikia didelio statistinio pasirengimo, kad padarytume tokią išvadą. Kitas pavyzdys: apklausoje 52 procentai vyrų ir 50 procentų moterų palaiko abortų draudimą. Ar galime apibendrinti, kad vyrai iš esmės yra daugiau konservatyvūs? O gal tokį rezultatą nesunku gauti atsitiktinai (kitoje apklausoje bus 49 procentai vyrų ir 51 procentai moterų), ir tai nerodo esminių skirtumų tarp lyčių? Būtent tokių klausimų centre, sprendžiant tarp apibendrinimo ir atsitiktinumo, yra statistinis reikšmingumas.

Bene garsiausias pavyzdys, kuriame puikiai išdėstoma atsitiktinio ir tikro rezultato principai, yra vieno iš moderniosios statistikos tėvų Ronaldo Fisherio apibūdintas eksperimento pavyzdys apie ponią, geriančia arbatą. Jo sąlygos gana paprastos. Ponia teigia, kad gali atskirti arbatos puodeliu, kuriose pienas buvo įpiltas į arbatą, nuo puodelių, kuriuose pirma buvo įpilta pieno. Apibendrinimas būtų toks: ponia iš tikrųjų geba atskirti arbatas (tai yra, gebės tai padaryti, nepriklausomai nuo ragavimo laiko, puodelių kiekio). O kaip tą išsiaiškinti vienu eksperimentu?

R. Fisheris teigia, kad galėtume daryti taip: pasiūlyti poniai paragauti arbatos iš aštuonių puodelių. Visi puodeliai turi būti vienodi visais kitais atžvilgiais, išskyrus tuo, kad keturiuose arbata įpilta pirma (arbata su pienu), o kituose keturiuose pirmas įpiltas pienas (pienas su arbata). Po paragavimo prašome ponios atskirti arbatas su pienu nuo pieno su arbata, taigi, ji mums turi pateikti keturis puodelius iš aštuonių ir pasakyti, kas juose yra. Eksperimento kriterijus čia bus griežtas: darysime išvadą, kad ponios gebėjimas atskirti arbatas nėra atsitiktinis, jeigu ji teisingai atrinks visus keturis arbatos puodelius. Dabar mums belieka nustatyti, kokia tikimybė tai padaryti atsitiktinai.

Kad nereikėtų vargti prisimenant tikimybių teoriją, įvairios puodelių kombinacijos jau paskaičiuotos pateikiamos 2 lentelėje. Pavyzdžiui, egzistuoja vienintelė kombinacija, kurioje ponia pateiktų visus 4 neteisingus puodelius – nes tik tiek tokių puodelių yra. Kita vertus, atrenkant 2 teisingus, 2 neteisingus puodelius yra net 36 kombinacijos. Iš viso galimų puodelių kombinacijų skaičius, atrenkant 4 iš 8 (jeigu keturi vienos rūšies, o kiti keturi) yra 70. Lygiai taip pat, kaip atrenkant neteisingus, egzistuoja tik viena kombinacija, kaip galima pateikti keturis teisingus puodelius (nurodyti, kad jie visi turi pieną su arbata).

Mes galime lengvai suskaičiuoti tikimybę tai padaryti atsitiktinai: ji lygi 1/70, kitaip tariant 1,4 procento. Palyginkite su monetos metimu: jeigu apsimesite „herbo specialistu“, tikimybė, kad tai pavyks jums padaryti atsitiktinai, yra 1/2, tai yra, 50 procentų. Galima rizikuoti. O štai parodyti, kad esate arbatos su pienu ar pieno su arbata specialistas, jeigu iš tiesų toks nesate, tikimybė R. Fišerio eksperimente yra gerokai mažesnė. Šiuo atveju 1,4 yra statistinio reikšmingumo, kitaip tariant, priėmimo, kad rezultatas nėra atsitiktinis, riba (dažniausiai išreiškiama santykiniu dažniu, 0.014): jeigu tikimybė gauti tokį rezultatą atsitiktinai yra tokia maža (ar mažesnė), darome išvadą, kad ponia iš tiesų sugeba atskirti pieną su arbata nuo pieno be arbatos. R. Fišeris, beje, niekada taip ir neatskleidė, ar toks eksperimentas buvo vykdomas iš tikrųjų ir kokia buvo jo baigtis. Galite pabandyti su keturiais puodeliais Koka-kolos ir keturiais Pepsi-kolos, jeigu turite draugą, kuris sako, kad gali juos atskirti!

2 lentelė. Ponios, geriančios arbatą, eksperimento baigties kombinacijos ir tikimybės

	4 teisingi	4 neteisingi	1 teisingai, 3 neteisingai	2 teisingai, 2 neteisingai	3 teisingai, 1 neteisingai
Galimų kombinacijų skaičius	**1**	1	16	36	16
Tikimybė būtent taip atrinkti puodelius	**1/70**	1/70	16/70	36/70	16/70
Tikimybė, išreikšta proc.	**1,4**	1,4	22,9	51,4	22,9
Statistinio reikšmingumo (*p*) kriterijus	**0.014**	0.014	0.23	0.51	0.23
Galimų kombinacijų skaičius	**1**	1	16	36	16

Statistinį reikšmingumas statistikoje žymimas *p,* o kai kuriose statistinėse programose ir kaip „sig.“ (angl. žodžio *significance* trumpinys). Kaip aptarėme, bendrąja prasme jis parodo, kokia tikimybė mūsų turimą rezultatą (eksperimento baigtį, procentą, ryšį tarp kintamųjų) gauti atsitiktinai. Jeigu ši tikimybė labai maža – patvirtinsime hipotezę ir darysime apibendrinimą. Taigi,

p galima traktuoti kaip pirmojo tipo klaidos tikimybę, patvirtinant hipotezę. Kuo *p* reikšmė didesnė, tuo didesnė jos tikimybė (kad gautas rezultatas yra atsitiktinis ir nerodo tiriamo efekto).

Žinoma, į *p* galima žiūrėti atvirkščiai (būtent taip statistikoje dažniausiai ir žiūrima): kaip į tikimybę, kiek esame tikri, kad nulinė hipotezė gali būti atmesta. Kuo *p* reikšmė didesnė, tuo ši tikimybė mažesnė. Pavyzdžiui, esame 95 proc. įsitikinę ($p<0.05$), kad galime atmesti hipotezę, jog išsilavinimo grupės nesiskiria pagal Darbo partijos vertinimą. Beje, standartinė naudojama statistinio reikšmingumo riba ir yra 0,05. Taigi, jeigu $p<0,05$, nulinė hipotezė atmetama ir daroma išvada, kad aptiktas ryšys (skirtumas ar kitas rezultatas) yra statistiškai reikšmingas.

Galima išskirti tris statistinio reikšmingumo lygmenis 1) $p<0,05$ (tikimybė, kad gautas rezultatas yra atsitiktinis - mažesnė nei 0,05, arba 5 proc.); 2) $p<0,01$(tikimybė, kad gautas rezultatas yra atsitiktinis - mažesnė nei 0,01, arba 1 proc.); 3) $p<0,001$(tikimybė, kad gautas rezultatas yra atsitiktinis - mažesnė nei 0,001, arba 0,1 proc.). Šiuos lygmenis patartina žinoti, kadangi jie rodo, kiek maždaug jūsų rezultatai yra patikimi. Bet kokiu atveju, jeigu *p* reikšmė yra didesnė nei 0,05, bendroje kiekybinių metodų praktikoje yra priimta, kad tikimybė analizės rezultatą gauti atsitiktinai jau yra per didelė.

Galima atkreipti dėmesį, kad R. Fišerio eksperimentas turi gana aiškiai apibrėžtas taisykles ir baigtis. Tačiau jeigu mes norime išsiaiškinti ryšį tarp išsilavinimo ir domėjimosi politika, kaip mus sužinoti, koks procentinis pasiskirstymas yra atsitiktinis, o kuris – ne? Kiekvienos tokios analizės atveju reiktų vargti, skaičiuojant kombinacijas – jeigu nebūtų išvadų statistikos metodų, kurie turi savo reikšmingumo ribas (jas savo ruožtu apskaičiuoja statistinės analizės programa). Vienas dažniausiai naudojamų tokių statistikų yra Chi-kvadratas. Dabar ją pristatysime kaip dar vieną statistinio reikšmingumo kriterijų pavyzdį, o panaudosime su apklausos duomenų pavyzdžiu kitame skyrelyje.

Tikriausiai esate rankose laikę šešių briaunų lošimo kauliuką. Jeigu lošimo kauliukas neforsuotas („sąžiningas“), tikimybė išmesti „1“, „5“ ar „6“ yra tokia pati ir vienoda – vienas iš šešių, 1/6. Taigi, jeigu mestume kauliuką 600 kartų, pagal teorinę tikimybė turėtume 100 kartų išmesti „1“, 100 kartų „2“ ir taip toliau (). Čia yra labai svarbus momentas kiekybinei analizei: tikimybės realizacija nėra tolygu pačiai tikimybei (jeigu patogiau suprasti, teorinei tikimybei)! Dažniai, paskaičiuoti pagal teorinę tikimybę, statistikoje vadinami laukiamais dažniais (angl. *expected counts*): jeigu kauliukas visiškai sąžiningas, laukiami dažniai yra šeši skaičiai išmesti po lygiai 100 kartų. Kitas pavyzdys būtų ryšys tarp išsilavinimo ir domėjimosi politika: jeigu ryšio

visiškai nėra, kiekvienoje išsilavinimo grupėje būtų po lygiai žmonių, besidominčių ir nesidominčių politika.

Grįžkime prie kauliuko. Jeigu turite laiko, galite pabandyti mesti jį 600 kartų. Tačiau tai nėra būtina – tikriausiai akivaizdu, kad nebus taip, jog gausite idealius laukiamus dažnius. O kaip tada nuspręsti, ar kauliukas yra sąžiningas (tai mūsų nulinė hipotezė)? Reikia tam tikros ribos, po kurios jau sakytume, kad vieną ar kitą reikšmę išmetėme tiesiog per dažnai, kad galėtume vis dar laikytis nulinės hipotezės. Taip kauliukas tampa statistine problema – teoriškai lyg ir aišku, tačiau praktiškai pradėjus gilintis prireikia tam tikrų skaičiavimų ir kriterijų.

Apibrėžkime eksperimentą. Nulinė hipotezė: lošimo kauliukas yra sąžiningas. Tyrimo hipotezė: lošimo kauliukas nėra sąžiningas Kitaip tariant, tyrimo hipotezę patvirtintume (ir nulinę atmestume), jeigu lošimo kauliukas reikšmingai skirtųsi nuo to, ką vadintume idealiai sąžiningu kauliuku (jau žinome, kad kažkiek skirsis, bet kiek?). Kitaip tariant, mums reikia sužinoti: ar mūsų duomenys reikšmingai skiriasi nuo tų, kuriuos dar būtų galima gauti atsitiktinai?

Net ir nežinodami statistikos metodų, tikriausiai bandytume išvesti tam tikrą matą, kuris parodytų išmestų dažnių skirtumus nuo tų, kurių tikėtumėmės idealiu atveju. Lentelėje 3 pateikti laukiami ir stebėti (kuriuos gavo medžiagos autorius, kantriai 600 kartų metęs kauliuką) dažniai. Stulpelyje „(O-E)“ (O reiškia *observed*, E reiškia *expected*) pateikiamas rezultatas, jeigu iš stebėto dažnio (ką realiai gavome 600 metę kauliuką) atimtume laukiamą dažnį (idealią teorinę tikimybę). Kyla pagunda šiuos skirtumus sudėti, tačiau to daryti negalime, nes jie išsiprastins tarpusavyje (panašiai, kaip ir skaičiuojant dispersiją). Todėl juos keliame kvadratu („O-E)^2). Galiausiai, mus domina ne patys skirtumai savaime, bet jų santykis su laukiamais dažniais. Sudedame paskutinio stulpelio rezultatus ir gauname statistiką „6,84“, kuri ir vadinama Chi-kvadratu (vėliau bus pademonstruota, kaip jį paskaičiuoti ir su *R* programa).

3 lentelė. Chi-kvadrato skaičiavimo etapai

Skaičius	**Laukiamas dažnis**	**Stebėtas dažnis**	**(O-E)**	**(O-E)^2**	**(O-E)^2/E**
1	100	111	(111-100)=11	11^2=121	121/100=1.21
2	100	90	(90-100)=-10	10^2=100	100/100=1
3	100	85	(85-100)=-15	15^2=225	225/100=2.25
4	100	102	(102-100)=2	2^2=4	4/100=0.04
5	100	115	(115-100)=15	15^2=225	225/100=2.25
6	100	97	(97-100)=-3	-3^2=9	9/100=0.09
Viso (Chi-kvadratas)	600	600			6.84

Pati ji savaime mums nesako nieko. Tačiau kiekviena tokia statistika turi standartinius tikimybių skirstinius, kurie pasako, kokia tikimybė prie tam tikrų sąlygų (imties dydis, kintamųjų skaičius, kintamųjų reikšmių skaičius) gauti konkrečią statistikos realizaciją. Šiuo atveju, jeigu norime 95 procentų patikimumo (standartinė 0,05 riba), kritinė Chi-kvadrato riba būtų apie 11. Mūsų gauta statistika yra lygi 6,84, reiškia, mažesnė nei 11. Taigi, nulinės hipotezės negalime atmesti, rezultatas per mažas (Chi-kvadrato statistika per maža), jis galėjo būti gautas ir atsitiktinai. Kauliukas yra sąžiningas! Tai išsiaiškinome, įvertinę stebėtus dažnius, laukiamus dažnius ir panaudoję tam tikrą statistinį kriterijų.

Beje, gali būti įdomu, iš kur atsiranda tos Chi-kvadrato (ir kitų statistinių kriterijų) kritinės ribos. Tai jau išsamiau statistikos teoriją pristatančio vadovėlio zona. Šioje metodinėje medžiagoje užtektų suprasti, kad daugelis bazinių statistikos metodų veikia būtent taip: išvedama tam tikra statistika ir įvertinama, kokia tikimybė ją buvo gauti atsitiktinai. Tai mums parodo statistinės analizės programos – kitame skyriuje turėsime ne vieną pavyzdį.

Apibendrinant šį skyrelį, reikėtų žinoti:

- Kuo išvadų statistika skiriasi nuo aprašomosios statistikos
- Ką reiškia išvadų statistikoje imtis ir populiacija
- Kas yra nulinė hipotezė
- Kas yra statistinis reikšmingumas ir kaip jį interpretuoti
- Kuo susijęs statistinis reikšmingumas ir pirmojo tipo klaida

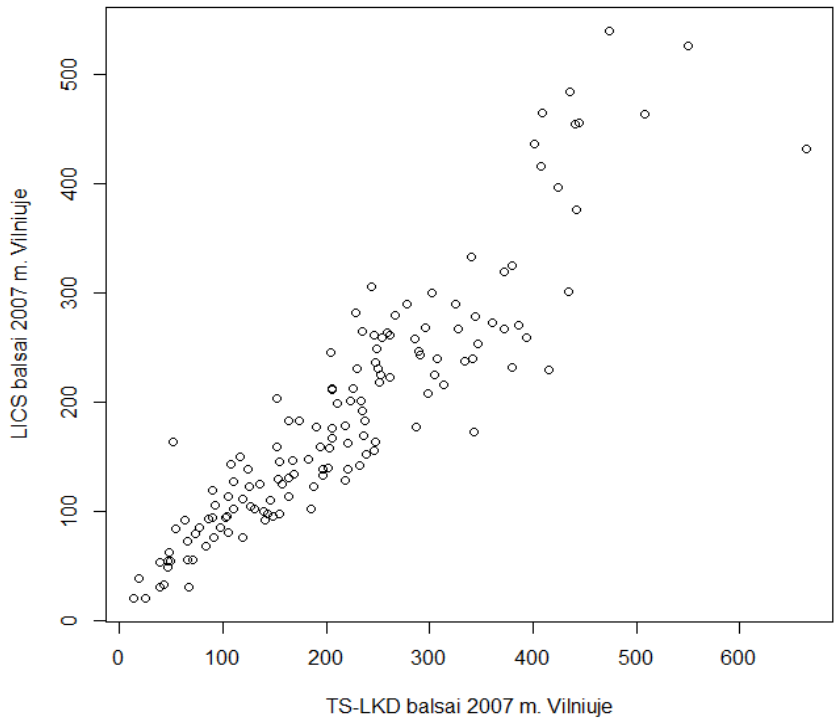
3.2 Ryšio matai: kiekybiniai duomenys

Kalbant apie ryšį tarp dviejų kintamųjų, dažnai naudojamas vadinamasis duomenų sklaidos grafikas (angl. *scatterplot*). Jo pavyzdys yra 6 paveikslėlyje, kuriame pavaizduotas ryšys tarp konservatorių ir liberalų balsų 2007 m. Vilniaus apylinkėse. Galime lengvai pastebėti, kad ryšys yra tiesinis ir teigiamas – ten, kur daugiau balsų gavo konservatoriai, ten sėkmingiau pasirodė ir liberalai (ir atvirkščiai). Beje, čia būtina akcentuoti, kad tai tik koreliacija. Jos interpretacija jau ne grafiko galioje.

Toks ryšio vaizdavimas yra gana patogus ir paprastai interpretuojamas. Deja, ne su visų duomenų tipais standartinis duomenų sklaidos grafikas tinka. Analizuojant duomenis su rangų skale

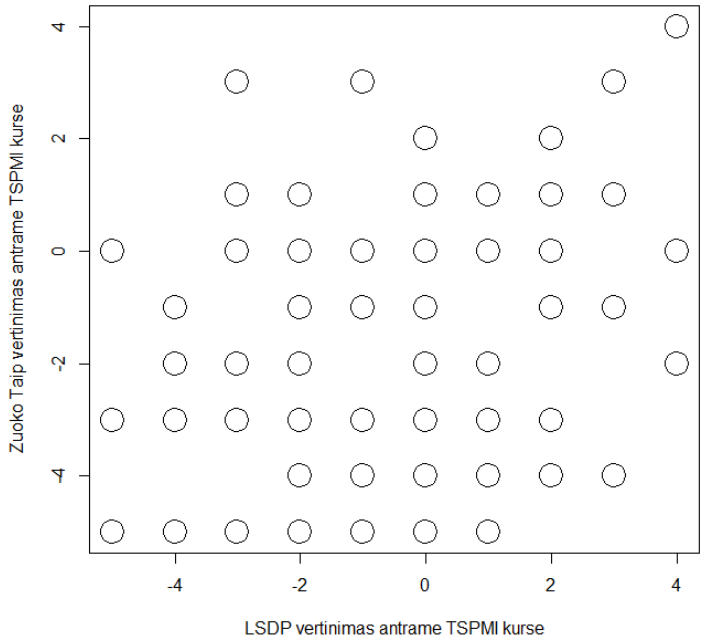
ar nominaliąja, tokia vizualizacija neatskleis nieko geriausiu atveju, o blogiausiu – gali suklaidinti ir lemti blogas išvadas.

6 pav. Ryšys tarp TS-LKD ir LICS balsų 2007 m. Vilniaus apylinkėse



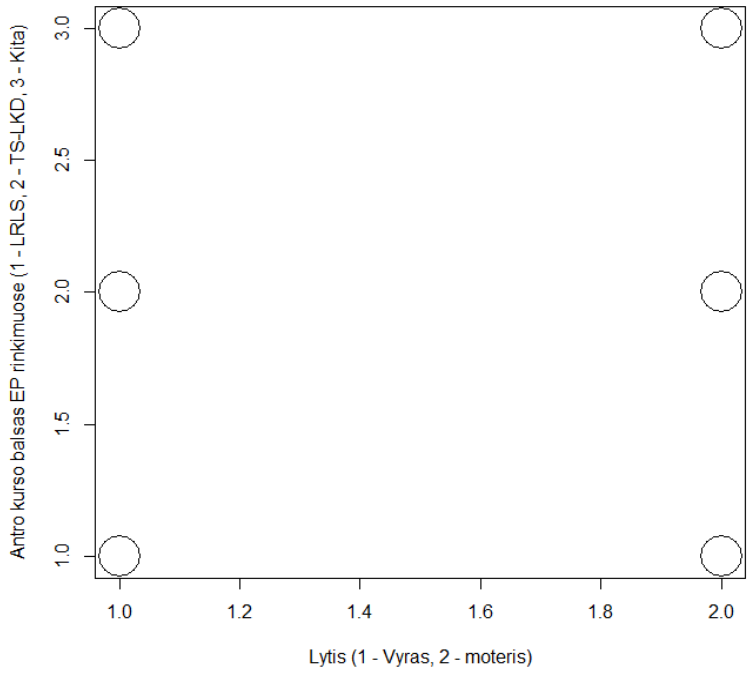
7 paveikslėlyje pavaizduotas ryšys tarp dviejų rangų skale matuotų kintamųjų, klausimų apie LSDP ir A. Zuoko partijos vertinimą (panašius klausimus jau analizavome anksčiau). Jokios aiškesnės tendencijos nematyti. Taip yra dėl to, kad mes nežinome, kiek žmonių yra „pasislėpę“ po apskritimais. Skirtingai nei 6 paveikslėlyje, kur kiekvienas apskritimas reprezentuoja unikalų stebėjimo atvejį, čia yra svarbu yra reikšmių dažniai.

7 pav. Rangų skalės ryšys, vizualizuojant su duomenų sklaidos grafiku



Galiausiai, 8 paveikslėlis rodo ryšį tarp dviejų nominalių kintamųjų – lyties ir balso Europos parlamento rinkimuose. Matome šešias kategorijas, tačiau apie ryšį nieko negalime pasakyti, nes, kaip ir ankstesnio grafiko atveju, nematome, kiek stebėjimo atvejų patenka į kiekvieną iš jų.

8 pav. Nominalios skalės „ryšys“, vizualizuojant su duomenų sklaidos grafiku



Iš šių trijų grafikų palyginimo matosi, kad konkretus ryšio vizualizacijos metodas tinka tik vienai duomenų skalei (gal ir galėtų tikti rangų skalei, tačiau labai specifiniais atvejais – todėl bendrai nerekomenduotina). Panašiai yra ir su kitais ryšio įvertinimo metodais. Šioje medžiagos dalyje bus aptartos visos trys pagrindinės skalės: pademonstruoti grafikai ir baziniai išvadų statistikos metodai, pagal kuriuos galima įvertinti ryšį tarp dviejų veiksnių, turint konkrečius duomenis. Pavyzdžiui, ar yra ryšys tarp išsilavinimo ir demokratijos vertinimo? O aktyvumo ir balsavimo už TS-LKD?

Prieš pradedant nagrinėti tokius klausimus, reikėtų akcentuoti vieną svarbų dalyką. Ryšio matai nenurodo priežastinio ryšio: tai interpretacijos ir teorijos reikalas. Tikriausiai esate girdėję teiginį „koreliacija nereiškia priežastingumo“. Kitaip tariant, jeigu reiškinių tendencijos sutampa (pavyzdžiui, pastebimas Darbo partijos populiarumo didėjimas ir barsukų populiacijos augimas) daryti išvadą, kad vienas lemia kitą, būtų loginė klaida. Tas pats galioja kiekybiniams ryšio matams. Matydami jų statistinę koreliaciją, daugiausiai galime teigti, kad jie tarpusavyje susiję. Kuris lemia kurį, statistika (bent jau aptariama šiame skyriuje) nepasako.

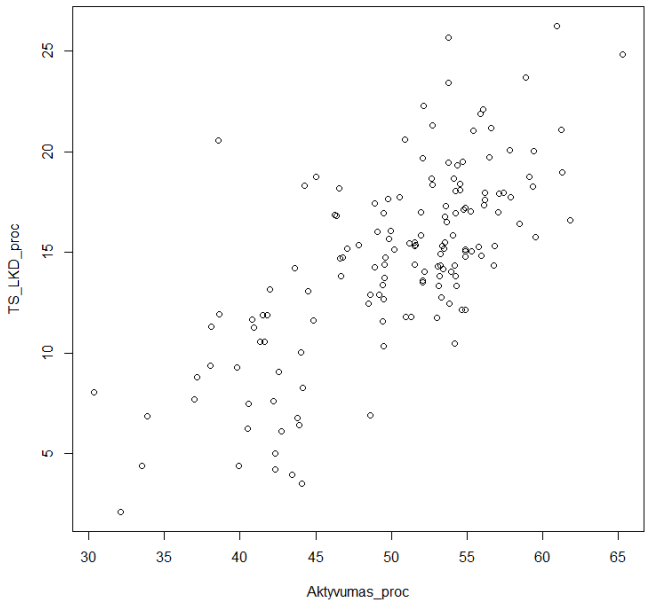
Pradėkime nuo kiekybinės skalės. Ryšys tarp tokios skalės duomenų visų pirma reiškia, kad vieno kintamojo reikšmėms didėjant (mažėjant), kito reikšmės atitinkamai sistemingai didėja (mažėja): prisiminkime, kad kiekybiniai kintamieji turi objektyviai išmatuojamus skirtumus tarp duomenų. Pavyzdžiui čia puikiai tinka savivaldos rinkimų Vilniuje rezultatai: stebėjimo atvejis čia yra apylinkė, o kintamieji – kiek balsų atiduota už konkrečią partiją. Nusiskaitykime duomenų failą, kurį jau naudojote anksčiau (jį galima parsisiųsti kartu su metodine medžiaga). Atkreipkite dėmesį, kad šis failas yra iš *Excel* programos, „.csv“ formato (galite jį atsidaryti ir patyrinėti su *Excel*). Žinoma, prieš tai turėtumėte nusistatyti darbinį aplanką. Tada galite nusikopijuoti visą apačioje esantį kodą ir jį visą paleisti (primename, „Ctrl+R“).

```
vilnius <- read.csv("Partijos_Vilnius_2015.csv", header=TRUE, sep=";", dec=",") # nuskaitome kitą failą su rinkimų rezultatų Vilniuje duomenimis
attach(vilnius) # prisekime Vilniaus rinkimų duomenis
colnames(vilnius) # galite pažiūrėti, kokius turime kintamuosius
```

Sakykime, norime išsiaiškinti, ar balsavimas už TS-LKD buvo susijęs su aktyvumu. Grafiškai, kaip jau buvo aptarta, galime ryšį tarp dviejų kiekybinių kintamųjų išreikšti duomenų sklaidos grafiku. *R* programoje jį sukuria komanda „plot()“ (beje, ši komanda parenka tinkamiausią grafiką

bet kokiems duomenims – galite išbandyti su kitais duomenimis). Skliaustuose pirmą rašome kintamąjį, kurio duomenys bus atvaizduojami horizontalėje. Antrą rašome vertikalės kintamąjį. Galime pastebėti, kad yra bendra tendencija, kad apylinkėse, kur buvo didesnis aktyvumas, TS-LKD gavo daugiau procentų balsų. Ir atvirkščiai.

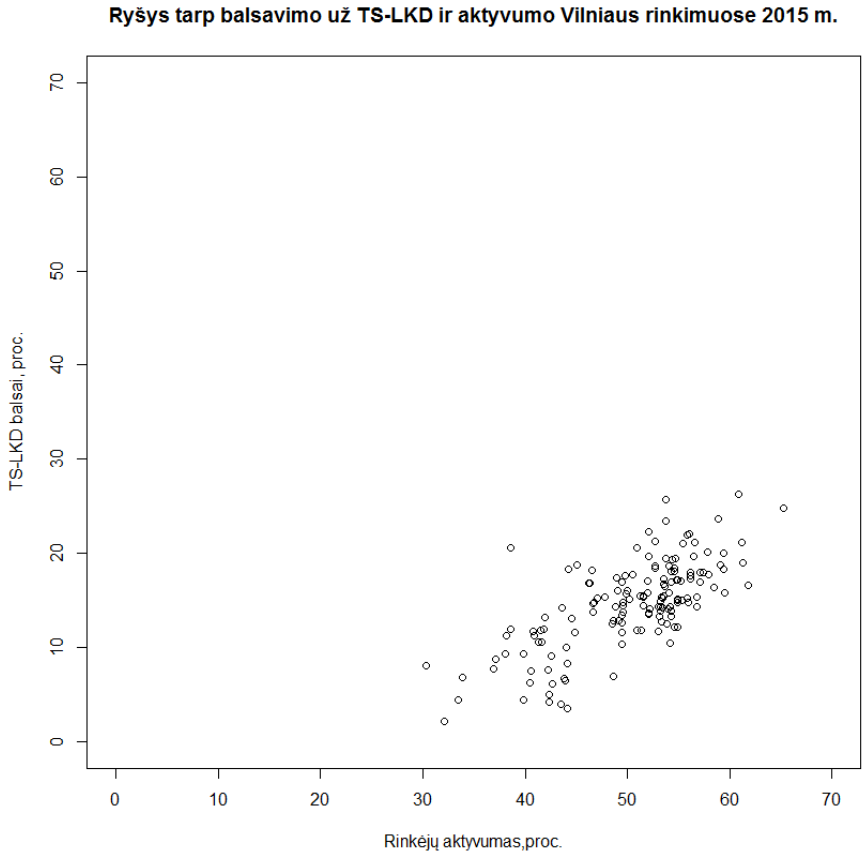
```
plot(Aktyvumas_proc, TS_LKD_proc)
```



Grafiką galima kiek patobulinti, įvesti horizontalės pavadinimą (argumentas „xlab“), vertikalės pavadinimą („ylab“) ir pagrindinį pavadinimą („main“). Jeigu norite suvienodinti x ir y ašies skalių intervalus, galite pasinaudoti argumentais „xlim“ ir „ylim“, kurie nurodomi kaip dviejų skaičių vektoriai (pirmas skaičius nurodo skalės maksimumą, antras – minimumą). Šį kodą turėtumėte nukopijuoti visą – tai viena ilga komanda.

```
plot(Aktyvumas_proc, TS_LKD_proc, xlab="Rinkėjų aktyvumas,proc.",
ylab="TS-LKD balsai, proc.",
main="Ryšys tarp balsavimo už TS-LKD ir aktyvumo Vilniaus rinkimuose 2015 m.",
xlim=c(0,70), ylim=c(0,70))
```

Suvienodinus ašių skalių ribas, grafike atsirado daug tuščios vietos. Nieko keisto – mažiausias aktyvumas buvo apie 30 procentų, o TS-LKD balsai prasideda vos kelių. Galima naudoti ir praėjusiame paveikslėlyje ribas, tik būtina atkreipti dėmesį, kad didesnius aktyvumo skaičius atitinkantys TS-LKD balsai yra mažesni. Pačios tendencijos, žinoma, tai nekeičia.



Grafikai yra labai svarbūs, vizualizuojant ryšį tarp kintamųjų, tačiau tai vis dar išvadų statistika. Sakykime, norime apibendrinti šią koreliaciją viena statistika. Tai padaryti gana nesunku, naudojant Pearsono koreliacijos koeficientą. Jis parodo tiesinį ryšį (koreliaciją) tarp dviejų kiekybinių kintamųjų: vieno reikšmei sistemingai didėjant (mažėjant), kito reikšmė atitinkamai sisteminai didėja (mažėja). Pearsono koeficiento įvertis svyruoja tarp -1 (tobulas atvirkštinis ryšys) ir 1 (tobulas tiesioginis ryšys). Įverčiui artėjant prie 0 iš abiejų pusių, ryšys silpnėja.

Pavyzdžiui, jeigu norėtume gauti Pearsono koreliaciją šiuo konkrečiu atveju, galime tiesiog pasinaudoti komanda „cor()“. Skliaustuose įrašome kintamųjų, tarp kurių norime įvertinti koreliaciją, pavadinimus. Jų eiliškumas čia nėra svarbus.

```
cor(Aktyvumas_proc, TS_LKD_proc)
```

Turėtumėte gauti tokį rezultatą.

```
> cor(Aktyvumas_proc, TS_LKD_proc)
[1] 0.7141686
```

Nėra vienos nuomonės, kokios koreliacijos koeficiento reikšmės turėtų būti traktuojamos kaip stiprus ar silpnas ryšys. Laikantis konservatyvesnio požiūrio, daugiau būdingo statistikos taikymui tiksliuosiuose moksluose, stipriomis reikėtų laikyti koreliacijas tik nuo 0,68 (arba net 0,8), vidutinio – intervale tarp 0,36 ir 0,67. Visgi socialiniuose moksluose vienas reiškinys turi ganėtinai daug galimų paaiškinimų, todėl tikėtis arti idealumo esančių ryšių yra sunku. Galima laikomasi Jacobo Coheno pasiūlytos klasifikacijos (4 lentelė): pagal ją, ryšio tarp kintamųjų nėra, kai Pearsons‘R<0.1 (kintamųjų bendra reikšmių sklaida yra mažesnė nei 1 proc.), o stipri koreliacija aptinkama, kai Pearsons‘R>0.36.

4 lentelė. Pearsons‘R koreliacijos koeficiento įverčių interpretavimas >

	Pearsons‘R įvertis	Kiek reikšmių sklaidos yra bendra kintamiesiems (proc.)
Efekto nėra	0,00–0,09	0–1
Silpna koreliacija	0,1–0,23	1–5
Vidutinio stiprumo koreliacija	0,24–0,36	6–13
Stipri koreliacija	0,37–1	14–100

Mūsų turima koreliacija tarp TS-LKD balsų ir aktyvumo, kaip socialiniams mokslams yra gana stipri ir teigiama (0,71). Tiesa, čia reiktų žinoti, kad kiekybinės koreliacijos įprastai būna stipresnės agreguotiems duomenims (valstybių ar kitų geografinių vienetų, o ne respondentų lygmens). Visgi nežinome svarbiausio dalyko – ar statistiškai reikšmingas ryšys? Tikriausiai taip, nes būtų keista, jeigu toks stiprus ryšys būtų gaunamas atsitiktinai. Tačiau turime įsitikinti, naudodami *p* reikšmę būtent Pearsono koreliacijos koeficiento atveju. Deja, tam funkcija „cor()“ netinka, ji nerodo statistinio reikšmingumo. Koreliacijoms skaičiuoti labai patogi funkcija yra „rcorr“, bet jai reikia paketo „Hmisc“. Instaliuokime jį ir užkraukime.

```
install.packages("Hmisc")
library(Hmisc)
```

Rcorr funkcijoje, kaip ir „cor()“, iš pradžių įvedame kintamųjų pavadinimus. Papildomas argumentas „type“ reikalingas tam, kad nurodytume, kokį ryšio matą naudoti (šioje funkcijoje yra ir rangų skalei skirtas matas, prie jo greitai prieisime).

```
rcorr(Aktyvumas_proc, TS_LKD_proc, type="pearson")
```

Gauname štai tokį rezultatą. *R* pirma pateikia koreliacijų matricą (pirmas įvestas kintamasis tampa „x“, o antras – „y“). Matome, kad x su x ir y su y koreliuoja tobulai. Tai nieko keisto, kiekvienas veiksnys pats su savimi turės idealų ryšį. Tačiau mus domina ryšys tarp atskirų kintamųjų, aktyvumo ir TS-LKD balsų. Kaip ir anksčiau, gauname stiprią koreliaciją: Pearsono koreliacijos koeficientas lygus 0,71. Toliau programa parodo, kad ryšiui skaičiuoti buvo panaudoti 151 stebėjimo atvejis (nieko keisto, turime duomenis apie visas apylinkes, nėra trūkstamų reikšmių). Galiausiai, apatinėje konsolėje rodomo rezultato dalyje prie raidės „P“ nurodomas statistinio reikšmingumo lygmuo, jis yra „0“. P<0,05, taigi, gautas ryšys nėra atsitiktinis. Galime padaryti išvadą (išvadų statistika!), kad tarp TS-LKD gautų balsų ir aktyvumo Vilniaus apylinkėse 2015 m. buvo statistiškai reikšmingas, stiprus ryšys: ten, kur aktyvumas buvo didesnis, konservatoriai sistemingai gavo daugiau balsų.

```
> rcorr(Aktyvumas_proc, TS_LKD_proc, type="pearson")
     x    y
x 1.00 0.71
y 0.71 1.00

n= 151


P
  x  y
x     0
y  0
```

Kartais prireikia paskaičiuoti koreliacijas didesniam kiekiui kintamųjų vienam metu – sudaryti vadinamąją koreliacijų matricą. Kadangi mūsų duomenų lentelėje „vilnius“ yra ne tik kiekybinių duomenų (apylinkių pavadinimai įvesti tekstu), reikėtų atskirti mus dominančius kintamuosius į atskirą lentelę. Tai galime padaryti gana lengvai (prisiminkite, mes „prisegėme“ duomenų lentelę „vilnius“, būtent todėl vesdami kintamųjų pavadinimus nenaudojame „$“ ženklo).

```
koreliacijoms <- data.frame(Aktyvumas_proc, LRLS_proc, LLRA_proc, TS_LKD_proc, LLS_proc)
```

Jeigu norime gauti koreliacijų matricą iš daugiau nei dviejų kintamųjų, „rcorr()“ funkcijoje reikia įterpti papildomą funkciją „as.matrix()“, kurios skliaustuose nurodome lentelę, iš kurios reikėtų imti kiekybinius duomenis. Visa kita išlieka taip pat.

```
rcorr(as.matrix(koreliacijoms), type="pearson")
```

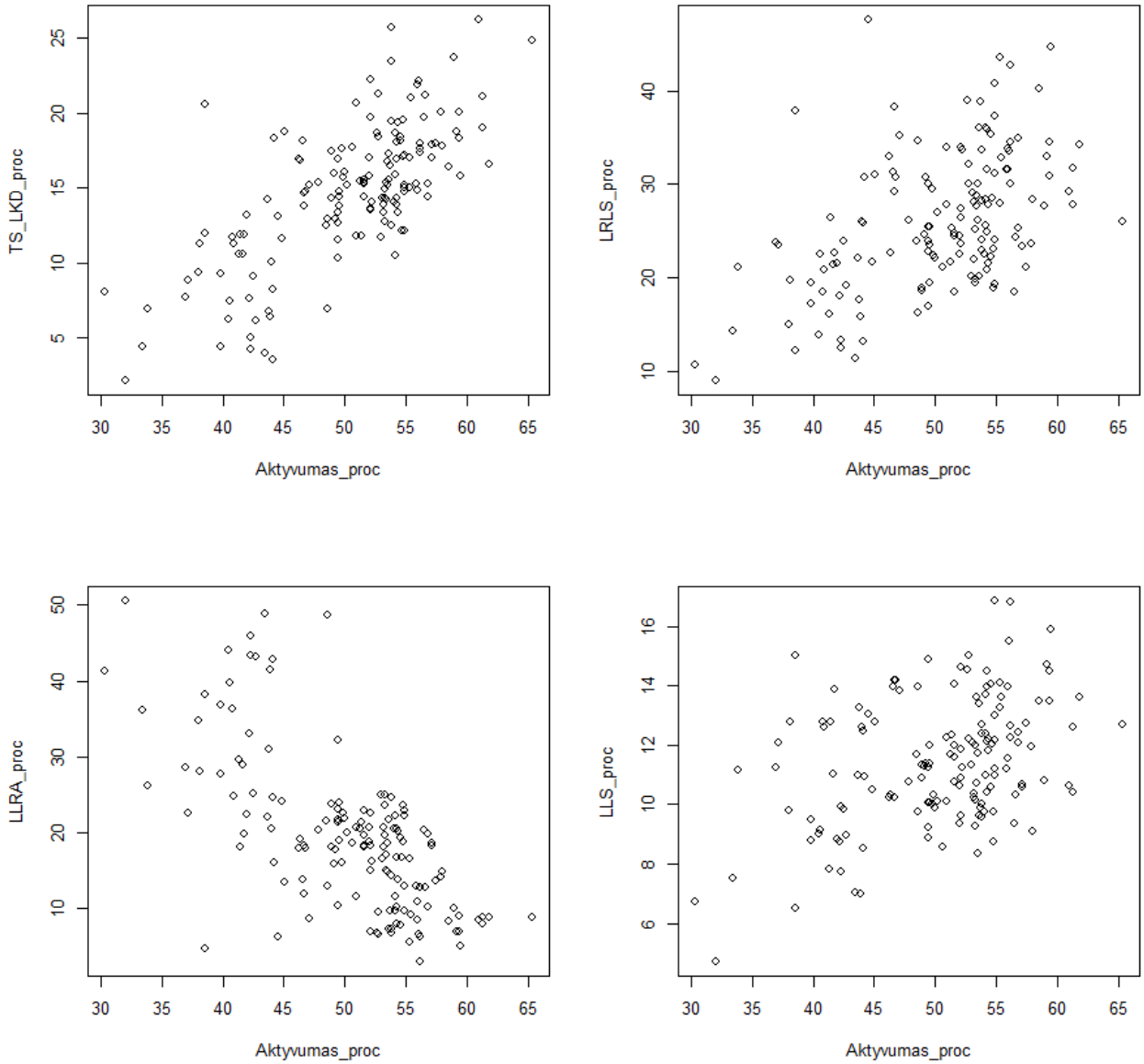
Konsolėje turėtumėte gauti: 1) lentelę koreliacijų; 2) nurodymą, kad imties dydis lygus 151 (n=151); 3) statistinio reikšmingumo lentelę. Pastarojoje vien nuliai – reiškia, visos koreliacijos yra neatsitiktinės! Panagrinėkite pačią koreliacijų lentelę. Ten yra įdomių dalykų. Pavyzdžiui, LLRA yra vienintelė partija, kurios balsai su aktyvumu koreliuoja neigiamai: kitaip tariant, ten, kur aktyvumas buvo didesnis, ten LLRA gavo mažiausiai balsų.

Sakykime, norime vizualizuoti visų keturių pagrindinių 2015 m. Vilniaus savivaldos partijų koreliacijas su aktyvumu viename grafike. R itin patogi programa tuo, kad su ja labai paprastai galima sukurti „mega-grafiką“, sudarytą iš kelių panašių grafikų. Tam naudojama, su funkcija „plot()“ suderinama funkcija „par()“. Paprastai kalbant, ji suskaldo dvimatę erdvę, kurioje dėliojame grafikus, pagal mūsų pageidaujamas dimensijas. Pavyzdžiui, jeigu reikia pavaizduoti keturis grafikus vienoje erdvėje, mums reikės 2x2 išmatavimų. Įvedame argumentą „mfrow“, kuris nurodo, kad grafikus vėliau dėliosime eilute, iš kairės į dešinę (vėliau galite pabandyti argumentą „mfcol“). Jam priskiriame reikšmę – elementų, kurie atspindi mūsų norimą dimensijų skaičių, seką. Pavyzdžiui, jeigu norėtume mega-grafiko iš 10 grafikų, galėtume priskirti reikšmes „c(5,2)“, „c(2,5)“.

```
par(mfrow=c(2,2))
```

Paleidus komandą, R atsiranda tuščia pilka erdvė. Užpildykime ją paeiliui, įkeldami keturis grafikus.

```
plot(Aktyvumas_proc, TS_LKD_proc)
plot(Aktyvumas_proc, LRLS_proc)
plot(Aktyvumas_proc, LLRA_proc)
plot(Aktyvumas_proc, LLS_proc)
```



Galite viršuje esančias keturias eilutes paleisti kartu, tačiau tam, kad geriau suprastume, kaip veikia su „par()“ sukurtos dvimatės erdvės užpildymas, geriau tai daryti vieną eilutę po kitos. Dabar viename dideliame grafike galime palyginti skirtingų partijų (vertikalios ašys) balsų koreliacijas su aktyvumu. Kaip ir iš koreliacijų, galima lengvai įžvelgti, kad vienintelė LLRA (apatinis dešinysis grafikas) turi neigiamą ryšį su į rinkimus atėjusių balsuoti rinkėjų procentu.

Apibendrinant šį skyrelį, reikėtų žinoti:

- Kuo kiekybinis ryšys skiriasi nuo nominalaus ir rangų ryšio
- Kas yra Pearsono koreliacijos koeficientas ir kaip jį interpretuoti
- R paketas „Hmisc“
- R funkcijos: „cor()“, „rcorr()“, „plot()“,

3.3 Ryšys tarp nominalių ir rangų kintamųjų

Šiame (jau paskutiniame) metodinės medžiagos poskyryje aptarsime situacijas, kai neturime kiekybinių duomenų, tačiau vis tiek norime statistiškai įvertinti ryšį tarp dviejų kintamųjų. Žinoma, ir išbandysime šį ryšį su *R* programa. Vėl dirbsime su jau pažįstama 2008 m. porinkimine apklausa.

Jeigu nebaigėte darbo sesijos po praėjusio poskyrio, reikėtų „nusegti“ duomenų lentelę su Vilniaus duomenimis. Jei R atsidarėte iš naujo, ignoruokite šią komandą, tačiau nepamirškite nusistatyti darbinės direktorijos.

```
detach(vilnius) # nusegame ankstesnę duomenų lentelę
setwd("C:/R pratimai") # nustatome darbinę direktoriją (galima rankiniu būdu)
```

Iš pradžių padirbėsime su nominaliais kintamaisiais. Tam reikės nusiskaityti apklausos failą taip, kad reikšmės būtų pateiktos tekstiniais pavadinimais („use.value.labels = TRUE“).

```
library(foreign)
apklausa1 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = TRUE, to.data.frame=TRUE)
```

Sakykime, norime sužinoti, ar yra ryšys tarp domėjimosi politika ir kada apsisprendžiama, už ką balsuoti. Galėtume iškelti tyrimo hipotezę, kad tie žmonės, kurie domisi politika, dažniau apsisprendžia dar prieš rinkimų kampaniją. Kintamąjį, kuris matuoja apsisprendimą, jau žinome iš ankstesnių skyrių – K6. Tarp nuskaitytų kintamųjų, pačiame sąrašo gale rasite supaprastintą, dvireikšmį domėjimąsi politika matuojantį kintamąjį „Domisi_politika“ (jo nėra klausimyne, tačiau jis sukurtas iš K1 klausimo). Patikrinkime šių dviejų kintamųjų dažnius.

```
colnames(apklausa1)
table(K6) # K6 dažniai
table(Domisi_politika) # Domėjimosi politika dažniai
```

Paskaičiuokime porinius dažnius, vadinamuosius “krostabus”. Nepriklausomas kintamasis šiuo atveju yra domėjimasis politika, todėl jį dedam į stulpelius (antras kintamasis skliaustuose) ir dažnius skaičiuojame būtent nuo stulpelių.

```
lentele <- table(K6, Domisi_politika) # sukuriame dažnių lentelę
lentele.dazniai <- prop.table(lentele, 2) # sukuriame dažnių lentelę
lentele.dazniai*100 # paskaičiuojame procentus iš santykinių dažnių
```

Matome, kad skirtumų yra, nors jie ir nėra drastiški: tarp tų, kurie politika domėjosi, prieš rinkimų kampaniją apsisprendė beveik 70 procentų. Tarp tų, kurie politika nesidomėjo, prieš rinkimų kampaniją apsisprendusių buvo beveik 55 procentai (vis tiek gana didelis procentas, kaip politika nesidomintiems!). Rinkimų kampanijos eigoje apsisprendusių buvo daugiau tarp tų, kurie deklaravo, kad politika nesidomi. Atrodytų, tyrimo hipotezę būtų galima patvirtinti. Tačiau turime įsitikinti dėl statistinio reikšmingumo.

Analizuojant nominalius kintamuosius ir „krostabus“, visada rekomenduotina pasitikrinti su jau anksčiau aptartu Chi-kvadratu, ar aptiktas ryšys (procentinis pasiskirstymas) nėra atsitiktinis. Tam R turi paprastą funkciją „chisq.test()“. Skliaustuose įrašome kintamuosius, kurių porinių dažnių atsitiktinumą norime įvertinti.

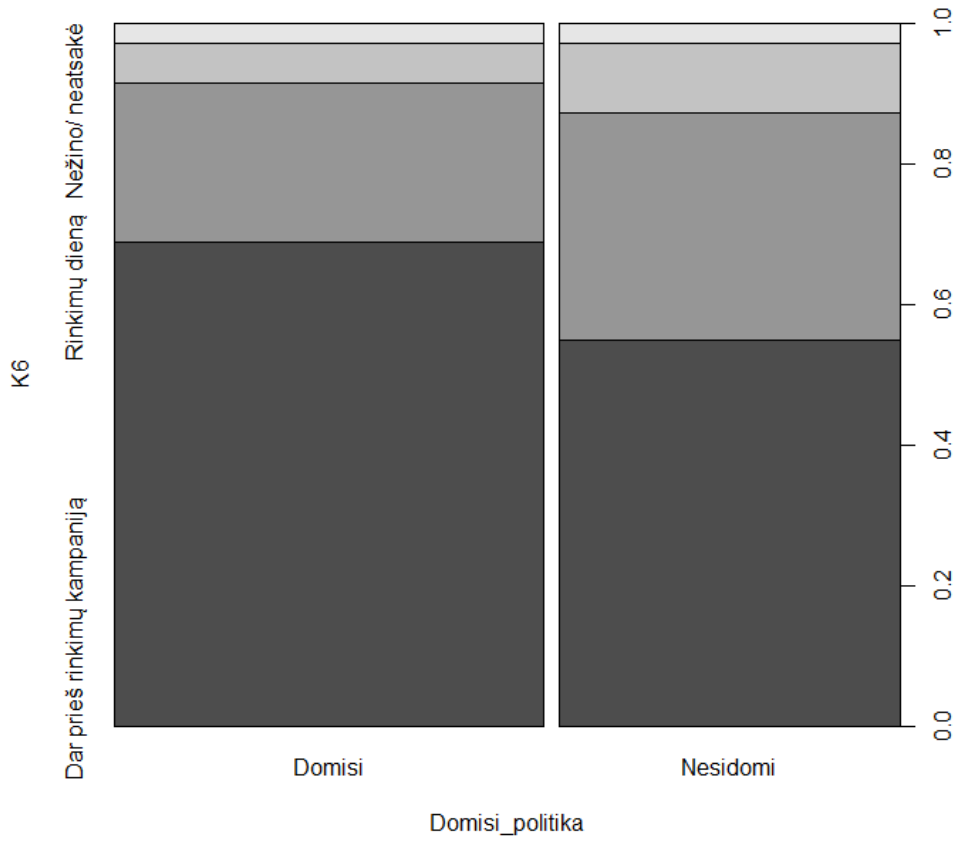
```
chisq.test(Domisi_politika, K6)
```

Reikėtų pastebėti, kad, skirtingai nei Pearsono koreliacijos koeficientas, Chi-kvadratas neturi įspraustų ribų, kuriose svyruoja – jis gali teoriškai būti nuo nulio (jeigu laukiami ir stebimi dažniai idealiai sutampa) iki labai didelių reikšmių (jeigu daug stebėjimo atvejų, daug kintamųjų reikšmių ir daug įvairių dažnių). Bet kokiu atveju, koks yra ryšys, matėme iš „krostabo“. Mums svarbu, ar jis statistiškai reikšmingas. Konsolė rodo „p-value = 0.0006301“. Taigi, $p<0,001$, ryšys yra statistiškai reikšmingas ties šia riba. Galime gana patikimai daryti išvadą, kad daugiau politika besidomintys žmonės vidutiniškai apsisprendžia anksčiau, už ką balsuos.

Ryšį tarp dviejų nominalių kintamųjų galima vizualizuoti. Visi žino stulpelių grafikus, tačiau jie kartais klaidina: viena kategorija gali būti gerokai didesnė (pagal respondentų skaičių) nei kita, o stulpelių plotis bus toks pats. *R* yra funkcija „spineplot()“, kuri iš nominalių kintamųjų „pagamina“ labai naudingą tokioje situacijoje grafiką. Nėra standartinio vertimo į lietuvių kalbą, tačiau galime jį vadinti „mozaikiniu“.

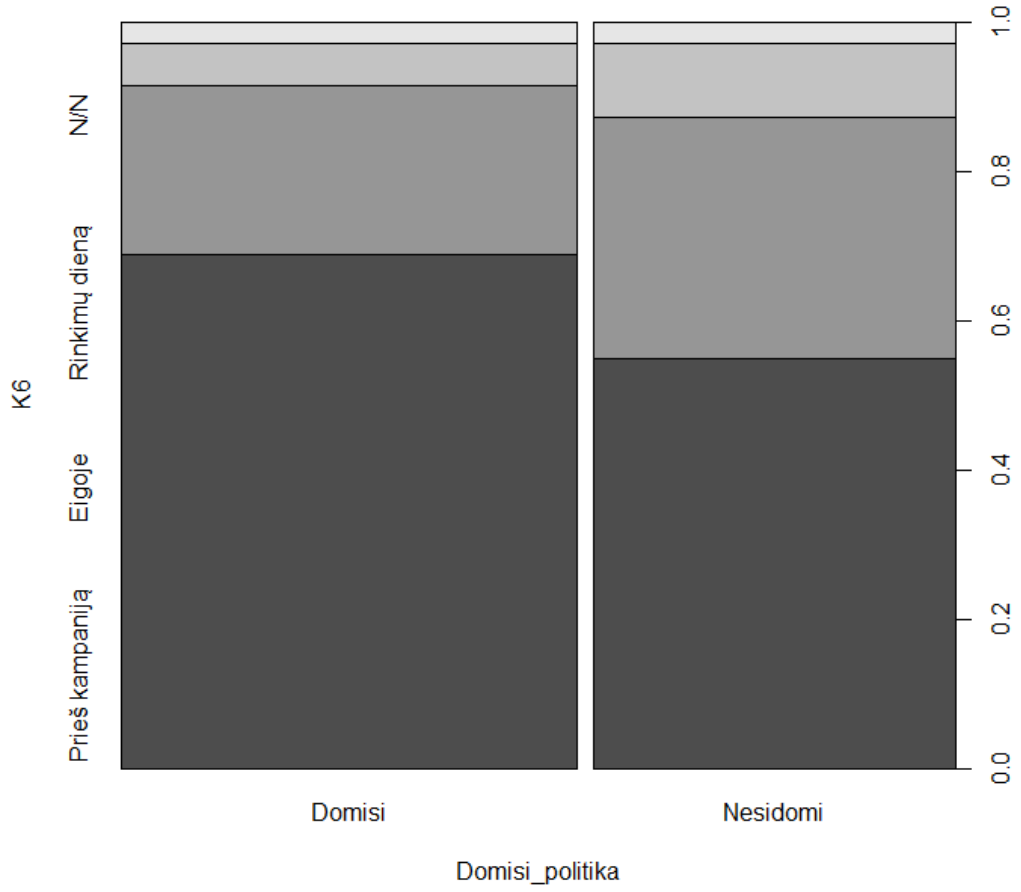
```
spineplot(Domisi_politika, K6)
```

Gauname grafiką, kuriame du stulpeliai atitinka besidominčių ir nesidominčių politika kategorijas. Matome du esminius dalykus. Pirma, kaip jau sužinojome, tarp tų, kurie domisi, yra daugiau apsisprendusių prieš rinkimų kampaniją (juoda stulpelis dalis aukštesnis). Antra, matome, kad nesidominčių dalis tarp balsavusių apskritai yra mažesnė, taigi tų, kurie domisi ir apsisprendžia seniau, santykinis svoris rinkimuose yra didesnis. Mozaikiniame grafike stulpelių ir jų dalių plotas yra proporcingas tos kategorijos dydžiui imtyje. Tai labai naudingas instrumentas, tiriant dažnių pasiskirstymus tarp dviejų kintamųjų.



Beje, galima buvo pastebėti, kad ilgi K6 kintamojo reikšmių pavadinimai iki galo netilpo į mozaikinį grafiką. Tai galime lengvai pakeisti, pakeisdami nominalaus kintamojo reikšmių pavadinimus (R jie vadinami „levels“).

```
levels(K6) <- c("Prieš kampaniją", "Eigoje", "Rinkimų dieną", "N/N")
spineplot(Domisi_politika, K6)
```



Galiausiai, lieka rangų skalė. Jos specifika kiek daugiau panaši į kiekybinę skalę, nei nominalią, kadangi reikšmes galima išdėlioti nuo mažiausios iki didžiausios (arba atvirkščiai). Kita vertus, rangų skalės kintamuosius, jeigu jie įgyja nedaug reikšmių, galima tirti ir su „krostabais“, Chi-kvadratu ir mozaikiniu grafiku, tačiau tai galioja tik tada, kai mums visų pirma įdomus procentinis reikšmių pasiskirstymas. Ranginė koreliacija parodo kitokį dalyką: ar didesnis (mažesnis) vieno kintamojo rangas atitinka didesnį (mažesnį) kito kintamojo rangą.

Dažniausiai naudojami ranginės koreliacijos koeficientai yra Spearmano koeficientas ir Kendall’s Tau-B. Naudosime pirmąjį, kadangi jų praktinis taikymas iš esmės nesiskiria, o Spearmano koeficientą turi jau aptarta funkcija „rcorr()“. Kaip ir Pearsono matas, ranginės koreliacijos koeficientai įgyja reikšmes nuo -1 iki 1. Tačiau jų interpretacija yra kiek kitokia. Pavyzdžiui, yra kintamieji X ir Y, N=4, reikšmių poros tokios: (0, 1), (10, 100), (101, 500), (102, 2000). Spearman’s Rho ir Kendall’s Tau-B bus lygūs 1 ir nurodys tobulą koreliaciją. Pearson bus lygus 0,75. Kodėl? Nors kiekvienu atveju Y padidėjimas nurodo ir X padidėjimą (ranginė koreliacija tobula), pats ryšys toli gražu nėra tobulai tiesinis (padidėjimai nėra tolygūs, sistemingi).

Pritaikykime Spearmano koeficientą ir pažiūrėkime, ar susiję liberalų ir konservatorių vertinimai – tikriausiai tikėtumėmės, kad geriau vienus vertinantys turėtų geriau atsiliepti ir apie kitus. Mums reikės skaičių (rangų koreliacijai reikia įvertinti, kokia reikšmė didesnė, kuri mažesnė), todėl nuskaitome failą iš naujo, nurodydami nenaudoti tekstinių reikšmių pavadinimų („use.value.labels = FALSE“).

```
detach(apklausa1) # nusegame prieš tai naudotą apklausą
apklausa2 <- read.spss(file = "2008 porinkimine.sav", use.value.labels = FALSE, to.data.frame=TRUE) #
```

Prieš skaičiuodami koreliacijas, pasižiūrėkime dažnius. Šalia skalės nuo 1 (labai patinka) iki 5 (labai nepatinka) yra reikšmė 9, kuri (pažiūrėkite į klausimyną) reiškia „nežinau/neatsakė“.

```
table(K10_3) # TS-LKD vertinimas
table(K10_9) # LRLS vertinimas
```

Kaip ir į vidurkio skaičiavimą, taip ir į koreliacijas šių reikšmių įtraukti negalima. Į tai reikia atsižvelgti, skaičiuojant koreliaciją. Prie kiekvieno kintamojo pridedame sąlygą įtraukti tik tas reikšmes, kurios ties kiekvienu iš tiriamų kintamųjų yra mažesnės nei 5.

```
library(Hmisc) # paketas su “rcorr()” funkcija
rcorr(K10_3[K10_3<=5 & K10_9<=5], K10_9[K10_3<=5 & K10_9<=5], type="spearman")
```

```
     x    y
x 1.00 0.37
y 0.37 1.00

n= 882

P
  x  y
x     0
y  0
```

Gauname panašų rezultatą, kaip ir su kiekybiniais kintamaisiais. Koreliacija yra lygi 0,37: taigi, tarp LRLS ir TS-LKD vertinimo egzistuoja tiesioginis ryšys. Jeigu rinkėjas vertina TS-LKD geriau, tikėtina, kad jis geriau vertins ir LRLS. Konsolės apačioje „p“ reikšmė rodo, kad ryšys nėra atsitiktinis, jis statistiškai reikšmingas.

Daug sąlygų laužtiniuose skliaustuose gali labai komplikuoti skaičiavimą, jose lengva pasimesti ir patyrusiam. Todėl galima koreliaciją paskaičiuoti ir be jų, tik tam reikia trupučio pasiruošimo: atskirti norimus kintamuosius, pakeisti reikšmes „9“ į NA (taip vadinamos trūkstamos reikšmės *R* programoje).

```
desine <- data.frame(K10_3, K10_9) # atskiriame kintamuosius į naują lentelę
colnames(desine) <- c("TS_LKD", "LRLS") # priskiriame naujus pavadinimus
desine[desine==9] <- NA # 9 paverčiame trūkstamomis reikšmėmis

detach(apklausa2) # atsegame apklausą
attach(desine) # prisegame naują lentelę
```

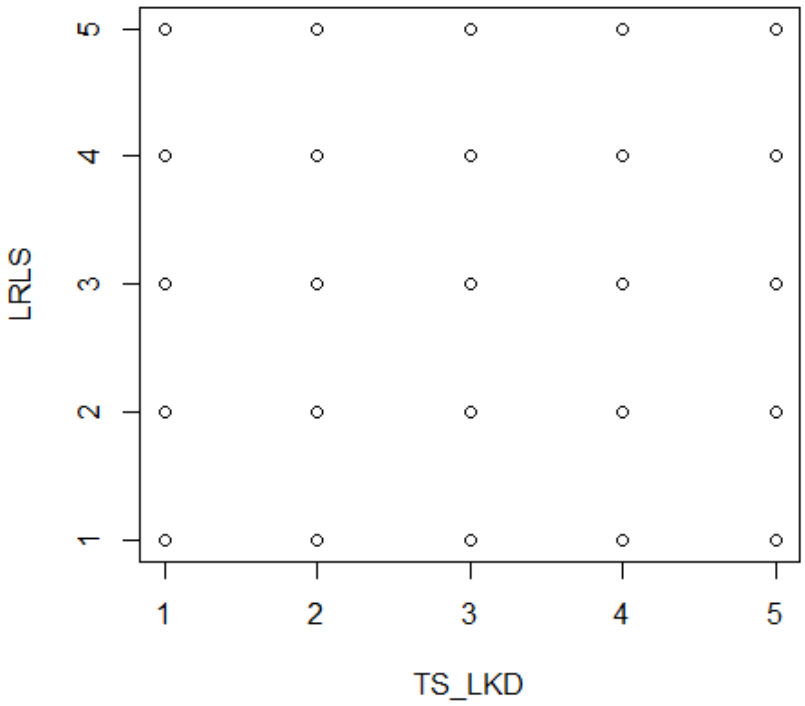
Koreliacija bus lygiai tokia pati, kaip anksčiau. Tik šį kartą skaičiuojant jau nereikia laužtinių skliaustų, nes *R* žino, jog NA į skaičiavimus neįtraukiamos.

```
rcorr(TS_LKD, LRLS, type="spearman") # koreliacija iš naujos lentelės
```

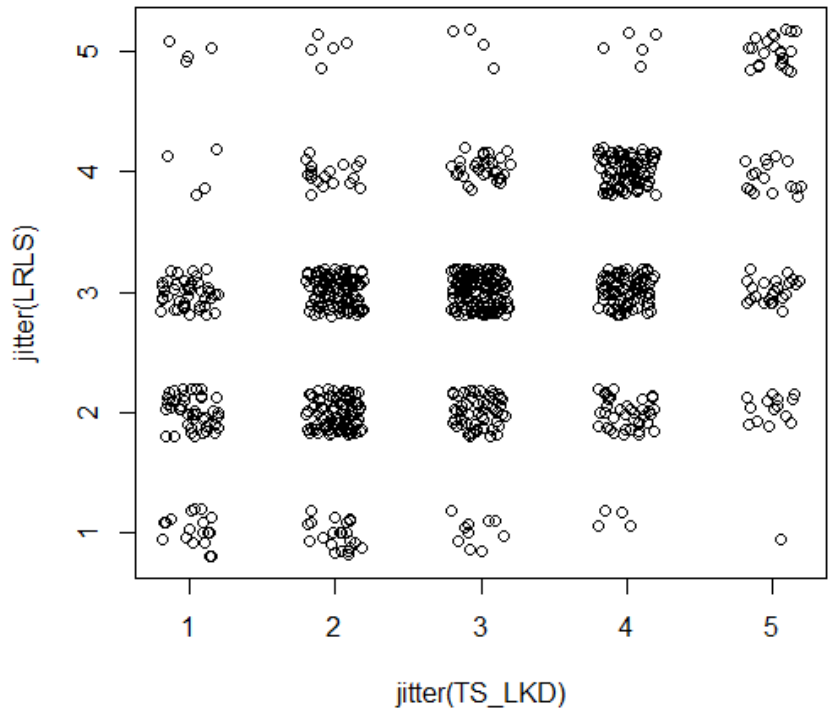
Kaip šį ryšį vizualizuoti? Pabandykime „plot()“ funkciją.

```
plot(TS_LKD, LRLS)
```

Kaip buvo demonstruota skyriaus pradžioje, gauname nieko nepasakantį „kiaurasamtį“. Ką daryti?

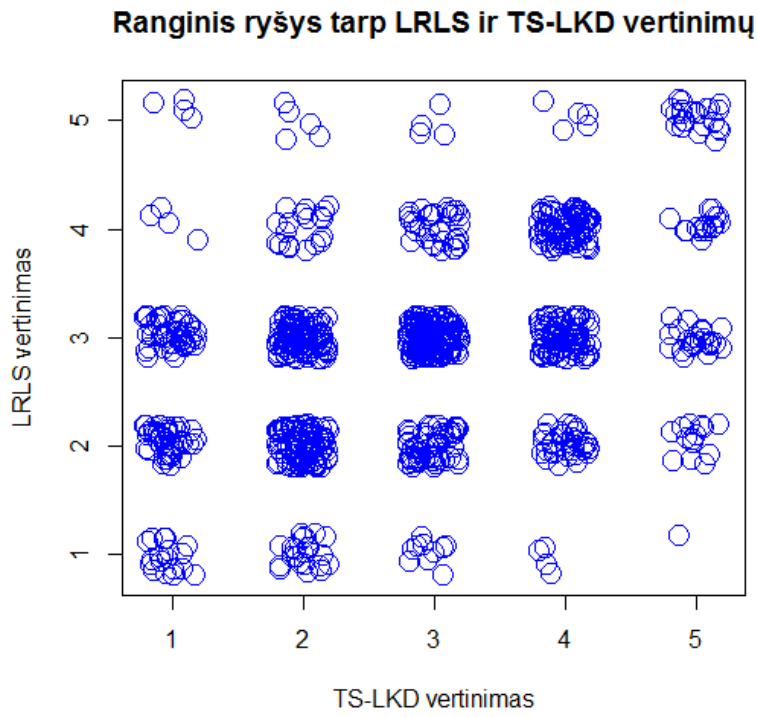


Egzistuoja viena paprasta, papildoma funkcija „jitter()“, kurią pritaikę abiems kintamiesiems funkcijos „plot()“ viduje, gausime „pabarstytą grafiką“. Dabar matome respondentų tankumą konkrečiose vertinimų kombinacijose. Pavyzdžiui, labai daug žmonių abi partijas vertino neutraliai (3 ir 3). Nemažai buvo tokių, kuriems abi nepatiko (4 ir 4), arba, atvirkščiai, patiko (2 ir 2). Matomas teigiamas ryšys, kurį anksčiau perteikėme koreliacijos koeficientu.



Jeigu norime, grafiką galime pagražinti: pridėti ašių ir pagrindinį pavadinimus, padidinti apskritimų dydį, juos nuspalvinti. *R* programa yra gana patogu įvairiausiems grafikų parametrų keitimams: ir, reikėtų akcentuoti, „plot()“ funkcija yra tik pati pradžia. Paketai ggplot2, lattice suteikia itin plačias pačių įvairiausių grafikų galimybes. Tačiau tai jau kitos metodinės medžiagos tema.

```
plot(jitter(TS_LKD), jitter(LRLS), xlab="TS-LKD vertinimas",
ylab="LRLS vertinimas",
main="Ranginis ryšys tarp LRLS ir TS-LKD vertinimų",
cex=2, col="blue")
```



Apibendrinant šį skyrių, reikėtų žinoti:

- Kas būdinga ryšiui tarp nominalių kintamųjų
- Kuo rangų ryšys skiriasi nuo kiekybinės koreliacijos
- Kaip interpretuoti Chi-kvadrato reikšmingumą ir Spearmano koeficientą
- R funkcijos: „chisq.test()“, „spineplot()“, „jitter()“

Vietoje pabaigos: rekomenduojama literatūra tolesnėms studijoms

Norint gilintis į statistiką:

- Vydas Čekanavičius ir Gediminas Murauskas, *Statistika I ir jos taikymai*, Vilnius: TEV, 2000.
- Vydas Čekanavičius ir Gediminas Murauskas, *Statistika II ir jos taikymai*, Vilnius: TEV, 2002.

Norint išmokti regresinių statistinės analizės metodų:

- Vydas Čekanavičius ir Gediminas Murauskas, *Taikomoji regresinė analizė socialiniuose tyrimuose*. Vilnius: Vilniaus universiteto leidykla, 2014. Interneto prieiga: http://www.lidata.eu/index.php?file=files/mokymai/trast/trast.html&course_file=trast_turinys.html.

Norint gilinti kiekybinių metodų žinias ir išmokti dirbti su SPSS

- Andy Field, *Discovering Statistics Using SPSS*, London: Sage Publications, 2005.
- Doc. Vytautas Janilionis, Dr. Vaidas Morkevičius, Dr. Rimantas Rauleckas, *Statistinė kiekybinių duomenų analizė su SPSS ir STATA*. Kaunas, 2008. Interneto prieiga: http://www.lidata.eu/index.php?file=files/mokymai/stat/stat.html&course_file=stat_turinys.html.

Norint gilinti kiekybinių metodų žinias ir toliau mokintis dirbti su R:

- Robert I. Kabacoff, *R in Action: Data Analysis and Graphics with R*, New York: Manning Publications, 2011. Autoriaus puslapis „Quick-R“ su daug praktiškų patarimų: http://www.statmethods.net/index.html.
- Andy Field, Jeremy Miles ir Zoë Field, *Discovering Statistics Using R*, London: Sage Publications, 2012.

Norint rimčiau gilintis į R ir gal net rašyti savo kodą:

- Joseph Adler, *R in a Nutshell: A Desktop Quick Reference*, Sebastopol: O'Reilly Media, 2010.