Viikko 6

Viimeistä viedään! Kuudennen viikon teemana ovat lineaarinen malli ja Bayes-päättely.

1 Yhden selittäjän lineaarinen regressio

Esimerkki 1. Tutkitaan fiktiivistä aineistoa, jossa x on ajoneuvon tankkiin laitetun polttoaineen määrä litroissa ja y kertoo montako kilometria ajoneuvolla päästiin ennen polttoaineen loppumista. Havaitaan seuraava aineisto:

Ajetut kilometrit (y)	Polttoaineen määrä (x)
76	8
72	9
89	11
144	18
158	19
92	10
156	20
109	14
138	17
100	12
51	6
107	13
129	16
65	7
121	15

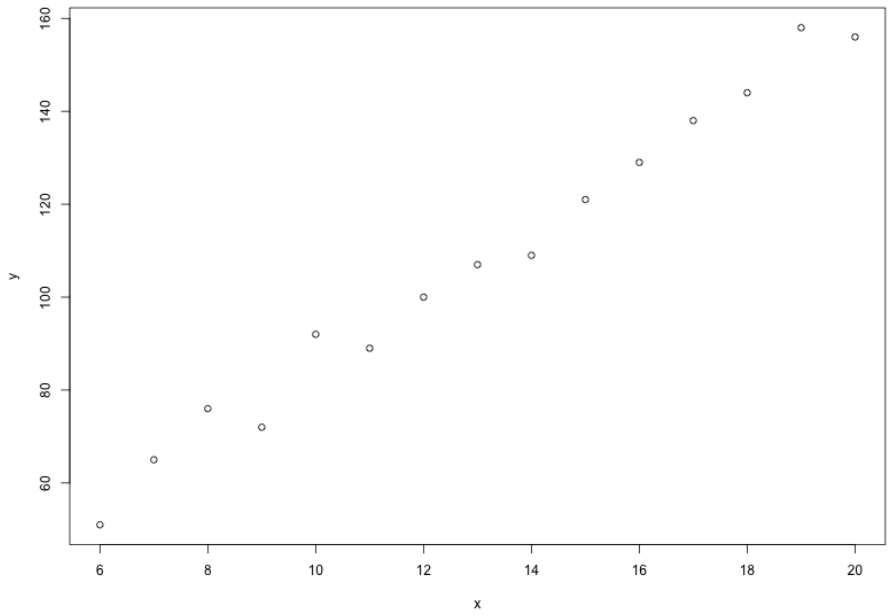
Nyt tuntuu hyvinkin luonnolliselta ajatella, että ajettujen kilometrien määrä riippuisi tankatun polttoaineen määrästä kutakuinkin lineaarisesti. Kuvasta 1 voidaan saada myös vahvistusta tälle intuitiolle. Kuvataan suhdetta kaavalla

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

missä $\beta_0, \beta_1 \in \mathbb{R}$ ja $\varepsilon \sim N(0, \sigma)$ on virhetermi. Estimoidaan nyt regressiosuoran vakio β_0 ja kulmakerroin β_1 R:llä.

```
> fit <- lm(y~x)
> fit

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     10.515        7.432
```

Esimerkki 2. Tarkastellaan edellisen esimerkin estimointitulosta. Saadut suurimman uskottavuuden estimaatit ovat $\hat{\beta}_0 = 10.515$ ja $\hat{\beta}_1 = 7.432$. Lisätään edellisessä esimerkissä piirrettyyn kuvaan punaisella regressiosuora käyttämällä kaavaa



Kuva 1: Ajettujen kilometrien ja tankatun polttoaineen yhteys esimerkissä

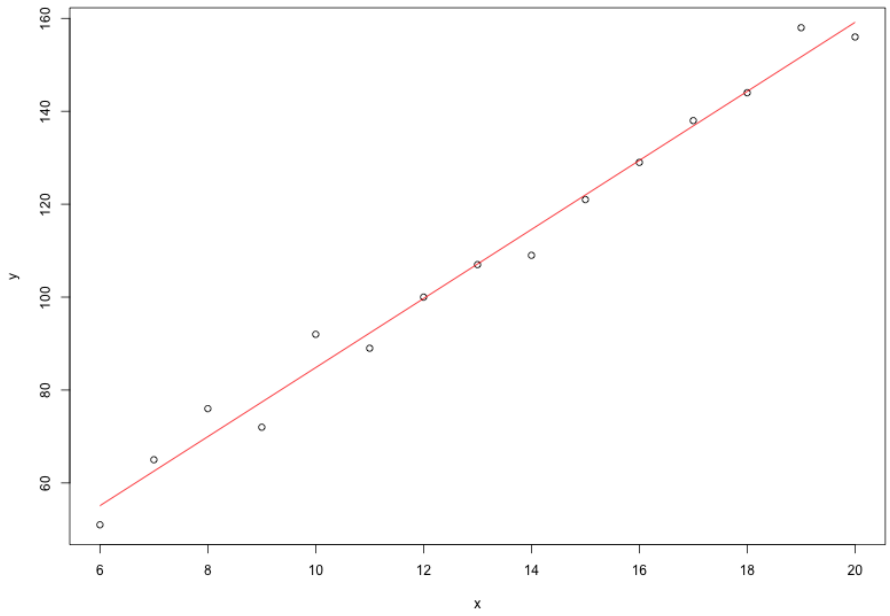
$$y = 10.515 + 7.432x.$$

Tämän voi tehdä monella tavalla, mutta katsotaan nyt toteutus `lines()`-ja `curve()`-funktioilla:

```
# Viiva lines()-funktiolla
x <- c(0,50)
y <- 10.515 + 7.432*x
lines(x,y, col="red")

# Viiva curve() -funktiolla. Argumentti add=T lisää viivan
# olemassaolevaan kuvaan, eikä luo uutta ikkunaa.
curve(10.515 + 7.432*x, from=0, to=50, add=T, col="red")
```

Esimerkki 3. Lisätään edellisissä esimerkeissä kasattuun kuvaan vielä ennustettujen arvojen luottamusvälit. Muodostetaan ensin muuttuja `x_values`, joka sisältää kaikki ne pisteet, joissa ennustetun arvon luottamusväli halutaan laskea. Tämän jälkeen 95% luottamusväli saadaan laskettua ja lisättyä kuvaan seuraavasti

```
> x_values <- seq(6,20, length.out=100)
> pred <- predict(fit, interval = "conf",
+ level = 0.95, newdata = list(x=x_values))
> lines(x=x_values, y=pred[,2], col="green")
> lines(x=x_values, y=pred[,3], col="green")
```

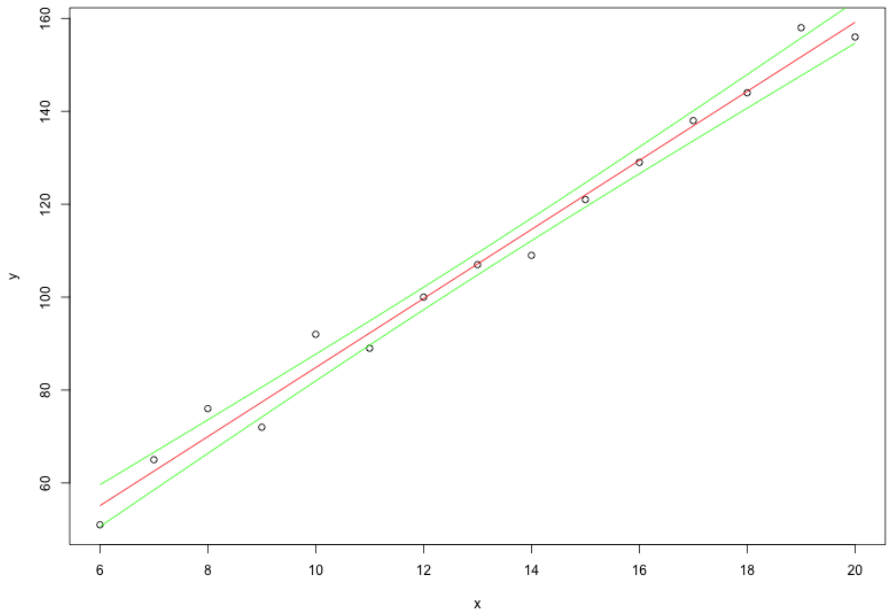


Kuva 2: Ajettujen kilometrien ja tankatun polttoaineen aineisto ja siihen sovitettu regressiosuora

Tarkempaa tietoa mallista ja sen onnistumisesta saadaan käyttäen funktiota `summary()`:

```
> summary(fit)

Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q Median     3Q    Max
-5.566 -3.214 -0.294  1.799  7.163

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.5155     3.4692   3.031  0.00965 **
x             7.4321     0.2532  29.348 2.88e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.238 on 13 degrees of freedom
Multiple R-squared:  0.9851,Adjusted R-squared:  0.984
F-statistic: 861.3 on 1 and 13 DF,  p-value: 2.88e-13
```

Esimerkki 4. Lasketaan vielä esimerkkitapauksen parametrien luottamusvä-



Kuva 3: Ajettujen kilometrien ja tankatun polttoaineen aineisto ja siihen sovitettu regressiosuora luottamusväleineen

lit. Funktio `summary()` ei näitä suoraan palauta, vaan ne saadaan laskettua funktiolla `confint()`:

```
> confint(fit)
                 2.5 %     97.5 %
(Intercept) -2.2924922 -0.1085592
y            0.1227926  0.1423074
```

Residuaaleja voidaan tarkastella esimerkiksi seuraavasti komennolla `plot(predict(fit),residuals(fit))`, joka piirtää mallin ennustamat selitettävän muuttujan arvot ja residuaalit vastakkain tavalliseen plot()-kuvaajaan. Mallioletuksen pätiessä residuaalien pitäisi olla likimain normaalijakautuneita, eikä residuaalikuviossa pitäisi olla nähtävissä mitään systemaattista vaihtelua.

2 Useamman selittäjän lineaarinen regressio

Esimerkki 5. Tutkitaan edellisen luvun esimerkkiä polttoaineen määrän ja ajettujen kilometrien välillä, mutta lisätään aineistoon kulloisenkin ajokerran aikana mitattu ulkolämpötila.

Ajetut kilometrit (y)	Polttoaineen määrä (x)	Ulkolämpötila (z)
76	8	21.6
72	9	24.2
89	11	23.1
144	18	23.6
158	19	27.1
92	10	22.2
156	20	26.0
109	14	23.3
138	17	25.8
100	12	24.4
51	6	21.8
107	13	23.8
129	16	22.9
65	7	21.8
121	15	25.2

Muodostetaan malli nyt käyttäen funktiota `lm()`, aivan kuten yhdenkin selittäjän tapauksessa.

```
> fit <- lm(y~x+z)
> fit

Call:
lm(formula = y ~ x + z)

Coefficients:
(Intercept)            x            z
    14.9273       7.4976      -0.2212

> summary(fit)

Call:
lm(formula = y ~ x + z)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7386 -3.2084 -0.6624  1.8672  7.0083

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.9273    23.1786   0.644    0.532
x             7.4976     0.4296  17.453  6.8e-10 ***
z            -0.2212     1.1481  -0.193    0.850
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.404 on 12 degrees of freedom
Multiple R-squared:  0.9852,Adjusted R-squared:  0.9827
F-statistic: 398.8 on 2 and 12 DF,  p-value: 1.061e-11
```

Voidaan havaita, että muuttujaa z vastaava kerroinparametri β_2 ei ole merkitsevästi poikkeava nollasta (p-arvo 0.850).

3 If else - rakenne

Ehtolauseita voi ohjelmoida R:ssä `if else`-rakenteella. Tulostetaan esimerkiksi näytölle, kumpi muuttujien `a` ja `b` arvoista on suurempi. Samalla esitellään myös funktio `cat`, joka tulostaa näytölle kaikki sille argumentteina annetut muuttujat, tässä tapauksessa muuttujan `a` arvon, sen jälkeen merkkijonon `on pienempi tai yhtä suuri kuin`, ja sen jälkeen muuttujan `b` arvon.

Esimerkki 6.

```
> a <- 5
> b <- 10
> if(a > b) {
    cat(a,"on suurempi kuin",b)
  } else {
    cat(a,"on pienempi tai yhtä suuri kuin", b)
  }
5 on pienempi tai yhtä suuri kuin 10
```

Jos `if`-lauseen ehto, tässä tapauksessa `a > b`, on totta, suoritetaan `if`-osan jälkeen aaltosuluissa oleva osa. Jos taas se on epätosi, suoritetaan `else`-osan jälkeen aaltosuluissa oleva osa, eli tulostetaan että `a`:n arvo on pienempi tai yhtäs suuri kuin `b`:n arvo.

`Else`-osa ei ole pakollinen, vaan voidaan käyttää pelkästään `if`-osaa. Tällöin jos ehto on epätosi, mitään ei tapahdu; esimerkiksi seuraava komento ei tulosta mitään.

Esimerkki 7.

```
> a <- 5
> b <- 10
> if(a > b) {
    cat(a,"on suurempi kuin",b)
  }
```

Jos halutaan testata useampaa ehtoa peräkkäin, voidaan lisätä `if`-lauseita. Esimerkiksi seuraavassa testataan ensin onko `a` suurempaa kuin `b`, ja jos ei ole, testataan onko se suurempaa kuin `b`. Jos tämäkään ei pidä paikkaansa, toteutetaan lopulta `else`-osa, eli tulostetaan että luvut ovat yhtä suuret.

Esimerkki 8.

```
> a <- 10
> b <- 10
> if(a > b) {
    cat(a,"on suurempi kuin",b)
  } else if(a < b){
    cat(a,"on pienempi kuin", b)
  } else {
    cat(a,"on yhtä suuri kuin", b)
```

```
  }
10 on yhtä suuri kuin 10
```

Kuten funktioiden ja `for`-silmukoiden tapauksessa, aaltosuluissa oleva osa on yleensä tapana sisentää, kuten yllä olevissa esimerkeissä. Sen sijaan aaltosulkujen poisjättäminen ei onnistu samalla tavalla: jos `if`-osan aaltosulut jättää kirjoittamatta, niin R ei osaa arvata, että tulossa on vielä `else`-osa, ja antaa virheilmoituksen. Esimerkiksi seuraava koodi ei ajettuna toimi, vaan antaa virheilmoituksen `Error: unexpected 'else' in "else"`.

Esimerkki 9.

```
# Huom. ei toimi!
if(a > b)
  cat(a,"on suurempi kuin",b)
else
  cat(a,"on pienempi tai yhtä suuri kuin", b)
```

Jos aaltosulut haluaa jättää pois, koko `if else`-rakenne on kirjoitettava yhdelle riville seuraavaan tapaan. Koska välissä ei ole rivinvaihtoa, R ei katkaise rakennetta ennen `else`:ä.

Esimerkki 10.

```
> if(a > b) cat(a,">",b) else cat(a,"<=", b)
10 > 5
```

R:ssä myös `if else`-rakenne on funktio, joten se palauttaa arvon. Tämä arvo on sen aaltosuluissa olevan (tai sen osan, joka kirjoitettaisiin aaltosulkuihin, jos ne kirjoitettaisiin näkyviin) osan, joka toteteutaan, viimeinen käsittelemä arvo. Tätä voidaan hyödyntää esimerksi valitsemalla suurempi luvuista `a` ja `b` ja sijoittamalla se muuttujaan `suurempi`.

Esimerkki 11.

```
> a <- 10
> b <- 5
> suurempi <- if(a > b) a else b
> suurempi
[1] 10
```

Monet aloittelevat R-ohjelmoijat, joilla on taustaa muista kielistä, käyttävät usein turhan paljon `for`-silmukoita ja `if else`-rakenteita, kun usein samat operaatiot ovat totetettavissa helpommin ja nopeammin R:n omien vektorisoitujen operaatioiden tai `apply`-perheen funktioiden avulla.

Aina kuitenkaan tämä ei ole mahdollista, esimerkiksi monimutkaisempia simulaatioita voi olla hankala vektorisoida, ja ne voi olla helpompaa toteuttaa silmukoilla ja `if else`-valintarakenteilla.

4 Bayes-päättelyä

4.1 Doping-testi-esimerkki

(Bayes-päättelyn kurssin 2015 viikon 1 tehtävä 3) Merkitään satunnaismuuttujalla D sitä, käyttääkö urheilija ainetta, eli $D = 1$, jos urheilija on doupattu ja

$D = 0$, jos urheilija on puhdas. Merkitään testin tulosta satunnaismuuttujalla T, eli $T = 1$, jos testitulos on positiivinen, ja $T = 0$, jos testitulos on negatiivinen. Testin sensitiivisyys, eli todennäköisyys että testitulos on positiivinen jos urheilija käyttää ainetta, eli $P(T = 1|D = 1) = 0.98$. Testin spesifisyys, eli todennäköisyys että testitulos on negatiivinen jos urheilija on puhdas, eli $P(T = 0|D = 0) = 0.95$.

Oletetaan että testattavista urheilijoista 1 % käyttää ainetta, eli $P(D = 1) = 0.01$. Havaitaan positiivinen testitulos. Mikä nyt on todennäköisyys, että kyseinen positiivisen testituloksen saanut urheilija käyttää ainetta?

Tämä on helppo ratkaista kynällä ja paperilla (ks. JTP:n luku 10), mutta lasketaan harjoituksen vuoksi approksimaatio kyseiselle todennäköisyydelle simuloimalla. Käytetään simulaation otoskokona $n = 1000000$, eli simuloidaan miljoonan urheilijan otos, ja katsotaan mikä on käyttäjien osuus niistä urheilijoista jotka saavat positiivisen testituloksen.

Esimerkki 12.

```
> n <- 1000000
> doupatut <- rbinom(n=n, size=1, prob=0.01)
> positiiviset <- numeric(n)
> for(i in 1:n) {
    if(doupatut[i] == 1) {
      positiiviset[i] <- rbinom(n=1, size=1, prob=0.98)
    } else {
      positiiviset[i] <- rbinom(n=1, size=1, prob=1-0.95)
    }
  }
> sum(doupatut * positiiviset) / sum(positiiviset)
[1] 0.1634877
```

Arvotaan ensin, onko urheilija doupattu vai ei, ja tallennetaan kunkin urheilijan doping-status vektoriin `doupatut` (1=käyttää, 0=ei käytä). Sen jälkeen arvotaan kunkin urheilijan testitulos ja tallennetaan tulos vektoriin `positiiviset` (1=positiivinen, 0=negatiivinen): jos urheilija käyttää ainetta, positiivisen testituloksen todennäköisyys on 0.98, ja jos ei käytä, positiivisen testituloksen todennäköisyys on $1 - 0.95 = 0.05$.

Lopuksi vain lasketaan niiden urheilijoiden määrä, jotka sekä käyttävät ainetta että saivat positiivisen testituloksen (`doupatut * positiiviset` saa arvon 1 vain niille urheilijoille, joille sekä vektorin `doupatut` että `positiiviset` arvo on yksi, eli jotka sekä käyttävät ainetta että saavat positiivisen testituloksen) ja jaetaan se kaikkien positiivisen testituloksen saaneiden urheilijoiden määrällä.

Näin saadaan laskettua miljoonan kokoisesta simuloidusta otoksestamme käyttäjien osuus positiivisen testituloksen saaneista. Se on noin 0.163, eli 16.3% (voit tarkastaa miten lähellä tulos on tarkkaa arvoa ratkaisemalla laskun Bayesin kaavan avulla), mikä on yllättävän pieni ottaen huomioon testin hyvän tarkkuuden (sekä sensitiivisyys että spesifisyys ovat vähintään 0.95).

4.2 Doping-testi-esimerkki vektorisoituna

Edellä todettiin, että R:ssä `for`-silmukat ja `if else`-rakenteet voidaan monesti korvata R:n omilla vektorisoiduilla operaatioilla. Niin myös tässä tapauksessa. Seuraavassa täsmälleen sama simulaatio on totetettu hieman erilaisella ja ehkä R:lle ominaisemmalla tavalla.

Esimerkki 13.

```
> n <- 1000000
> doupatut <- rbinom(n=n, size=1, prob=0.01)
> positiiviset <- numeric(n)
> n_doupatut <- sum(doupatut)
> positiiviset[which(doupatut == 1)] <- rbinom(n=n_doupatut, size=1, prob=0.98)
> positiiviset[which(doupatut == 0)] <- rbinom(n=n-n_doupatut, size=1, prob=1-0.95)
> sum(doupatut * positiiviset) / sum(positiiviset)
[1] 0.1668753
```

Koodin alku ja loppu ovat täsmälleen samoja kuin aiemmin, mutta `for`-silmukka on korvattu suorilla sijoituksilla vektoriin. Huomaa, että sekä vektori johon sijoitetaan ja sijoitettava vektori ovat samanpituisia, minkä takia sijoitus toimii:

Esimerkki 14.

```
> length(positiiviset[which(doupatut == 1)])
[1] 10064
> length(rbinom(n=n_doupatut, size=1, prob=0.98))
[1] 10064
```

5 Tehtäviä

1. Tutkitaan Pyöräilybarometria lineaarisen regression avulla

   a) Muuta sarakkeiden Aq6 - Aq15 ”En osaa sanoa”-vastaukset puuttuviksi. Luo sen jälkeen summamuuttuja (siis vastausten riveittäiset keskiarvot) sarakkeista Aq6 - Aq15. Muuta summamuuttujan arvot puuttuviksi niillä riveillä, joilla on 5 tai useampi puuttuvaa vastausta. (Tämä on siis täsmälleen sama kuin viikon 5 tehtävä 1. Jos olet tehnyt sen, niin voit käyttää samaa koodia.)

   b) Muodosta lineaarinen malli funktiolla `lm()`, missä selittäjänä on vastaajan ikä ja selitettävänä edellä luotu summamuuttuja. Tallenna malli muuttujaan `fit`.

   c) Piirrä aineistosta kuva, jossa x-akselilla on selittäjä ja y-akselilla selitettävä muuttuja

2. Jatkoa edelliseen tehtävään

   a) Lisää edellisen tehtävän kuvaan regressiosuora valitsemallasi värillä

   b) Tutki komennon `summary(fit)` tulostetta. Tulkitse parametrien estimointituloksia sanallisesti.

c) Piirrä mallin residuaaleista kuvaaja ja histogrammi. Tulkitse tulosta sanallisesti.

3. Laske edellisen tehtävän lineaarisen mallin parametrien 95% luottamusvälit. Voidaanko hylätä nollahypoteesi H_0: $\beta_0 = 2.0$ (ts. vakio = 2.0) luottamustasolla $1 - \alpha = 0.95$? Vihje: Esimerkki 4.

4. Tutustu esimerkkiin 5. Ota nyt selittäjä `x` pois mallista ja piirrä kuva. Vertaa saamaasi mallia esimerkin 1 tulokseen ja pohdi kumpi on parempi selittäjä havainnoille y. Tässä voit myös tutkia funktion `summary()` tulostetta.

5. JTP:n esimerkki 10.1. Tiedetään, että kulhossa 5 palloa. Näistä θ palloa on valkoista ja $5-\theta$ mustaa. Oletetaan, että kaikki valkoisten pallojen määrät $\theta = 0, \ldots, 5$ ovat yhtä todennäköisiä. Nostetaan nyt satunnaisesti 7 palloa (nostettu pallo palautetaan takaisin koriin noston jälkeen, eli kyseessä on otanta takaisinpanolla). Nostetuista palloista 2 on valkoista. Mitkä ovat nyt valkoisten pallojen eri määrien θ todennäköisyydet?

   Tarkastetaan esimerkissä analyyttisesti laskettu todennäköisyysjakauma valkoisten pallojen määrälle simuloimalla. Valkoisten pallojen määrä korissa, eli θ, noudattaa siis diskreettiä tasajakaumaa[1] joukossa $\{0, 1, \ldots, 5\}$, eli $\theta \sim \text{Tas}\{0, 1, \ldots, 5\}$. Jos oletetaan, että korissa on θ valkoista palloa, ja jos merkitään nostettujen valkoisten pallojen määrää satunnaismuuttujalla Y, se noudattaa binomijakaumaa otoskoolla 7 ja onnistumistodennäköisyydellä $\frac{\theta}{5}$, eli $Y|\theta \sim \text{Bin}\left(7, \frac{\theta}{5}\right)$.

   a) Simuloidaan tilannetta 1000000 kertaa. Arvo ensin korissa olevien valkoisten pallojen määrät vektoriin `valkoisia_korissa`. Vihje. `sample`-funktio.

   b) Arvo sitten nostettujen valkoisten pallojen määrät 1000000:n pituiseen vektoriin `valkoisia_nostettu` olettaen että jokaisella kerralla korissa on a-kohdassa laskettu määrä valkoisia palloja (esim. jos vektorin `valkoisia_korissa` ensimmäinen arvo on kolme, arvot vektorin `valkoisia_nostettu` ensimmäisen arvon ehdolla $\theta = 3$ ja niin edelleen). Vihje. `rbinom`-funktiolle voi antaa `prob`-argumentiksi vektorin: tällöin R käyttää ensimmäisessä otoksessa todennäköisyytenä vektorin ensimmäistä alkoita, toisessa toista ja niin edelleen.

   c) Valitse lopuksi vektoriin `otokset` ne otokset (eli oikeastaan vektorin `valkoisia_korissa` arvot), joilla on nostettu täsmälleen 2 valkoista palloa (eli vektorin `valkoisia_nostettu` arvo on 2). Laske vektorin `otokset` frekvenssitaulu ja normalisoi se jakamalla se niiden otosten, joissa on nostettu 2 valkoista palloa määrällä (eli vektorin `otokset` pituudella). Vertaa simulaatiosi tuottamaa todennäköisyysjakaumaa valkoisten pallojen määrille esimerkissä 10.1 laskettu eksaktiin jakaumaan.

6. JTP:n viikon 5 tehtävä 5: Vuonna 1975 uutisoitiin tutkimuksesta, jonka mukaan 50 % kanadalaisista miehistä käytti värillisiä (= muita kuin valkoisia) alushousuja kun taas amerikkalaisista miehistä sellaisia käytti vain

[1] Huom. kyseessä on siis diskreetti eikä jatkuva tasajakauma, joten siitä simuloidaan R:ssä käyttäen funktiota `sample` eikä funktiota `rbinom`.

20 %. Bermudalaisen hotellin asiakaskunta koostui yksinomaan amerikkalaisista ja kanadalaisista siten, että miesasiakkaista 80 % oli amerikkalaisia ja 20 % kanadalaisia. Todennäköisyyslaskentaa opiskellut siivooja huomasi miesasiakkaan huoneessa punaiset alushousut. Millä todennäköisyydellä hän päätteli asiakkaan olevan kanadalaisen?

Tarkistetaan kynällä ja paperilla laskettu vastaus simuloimalla, eli luomalla otos hotellin asiakkaista ja heidän alushousujen väreistään käyttämällä yllämainittuja osuuksia todennäköisyyksinä.

a) R:ssä ei ole vakiona funktiota Bernoullin jakauman simuloimiseksi. Bernoullin jakauma on sama asia kuin binomijakauma, jonka otoskoko (huom. R:ssä argumentti `size`) on 1. Luo apufunktio `rbern`, jonka parametrit ovat `n`, eli haluttu otoskoko ja `prob`, eli yksittäisen satunnaiskokeen onnistumisen todennäköisyys. Funktion tulee toimia siis seuraavasti (tietenkin otos on joka kerta eri):

```
> rbern(n=10, prob=0.5)
 [1] 0 1 0 0 0 0 0 0 1 1
```

Vihje: kannattaa hyödyntää valmista `rbinom`-funktiota.

b) Kirjoita funktio `hotellisimulaatio`, jonka argumenttina on hotellin miespuolisten asukkaiden määrä `n`, ja simuloi edellisen tehtävän tilannetta (käytä yllämainittuja osuuksia todennäköisyyksinä, esim. jos kanadalaisten osuus on 20 %, todennäköisyys että asukas on kanadalainen on 0.2). Funktion tulee palauttaa kanadalaisten osuus niistä asukkaista joilla on värilliset alushousut.

Arvo ensin kanadalaisten määrä, ja sen jälkeen arvo `for`-silmukan ja `if else` - rakenteen avulla niiden asukkaiden, joilla on värilliset alushousut, määrä. Vaikka osaisit tehdä tämän käyttämättä `for`-silmukkaa, käytä sitä kuitenkin tällä kertaa: sama simulaatio tehdään seuraavassa tehtävässä uudelleen hieman fiksummin vektorisoitujen operaatioiden avulla.

Lopuksi laske niiden asukkaiden, jotka ovat kanadalaisia JA joilla on värilliset alushousut, määrä, ja jaa se niiden asukkaiden, joilla ylipäänsä on värilliset alushousut, määrällä.

c) Testaa funktiotasi otoskoolla 100, 10000 ja 1000000. Mikä on simuloinnin perusteella approksimaatio todennäköisyydelle, että punaalushousuinen asiakas on kanadalainen? Jos olet laskenut JTP:n tehtävän käsin, voit verrata simuloimalla laskettua arvoa tarkkaan arvoon.

7. Tehdään edellisen tehtävän funktio hieman ”fiksummin” vektorisoitujen operaatioiden avulla.

a) Kirjoita funktio `hotellisimulaatio2`, jonka argumentti ja palautusarvo ovat samat kuin edellisessä tehtävässä. Tällä kertaa kuitenkin kanadalaisten määrän arpomisen jälkeen arvo niiden asukkaiden, joilla on värilliset alushousut, määrä käyttämättä `for`-silmukkaa.

b) Testaa otoskoolla 1000000, että funktiosi antaa samansuuntaisen tuloksen kuin edellisessä tehtävässä.

c) Vertaa funktioiden `hotellisimulaatio` ja `hotellisimulaatio2` nopeutta otoskoolla 1000000 käyttämällä funktiota `system.time`. Kumpi oli nopeampi, ja oliko ero selkeä (ajat vaihtelevat testikerrasta toiseen, mutta suuruusluokan pitäisi pysyä samana)?

8. Toteutetaan simulaatio vielä kolmannella eri tavalla. Jokaista satunnaiskoetta ei tarvitse arpoa erikseen vektoriin, vaan voidaan arpoa pelkästään kanadalaisten ja värillisalushousuisten määrät suoraan binomijakaumasta.

   a) Toteuta funktio `hotellisimulaatio3`, jonka argumentti ja palautusarvot ovat samat kuin edellisessä tehtävässä. Tällä kertaa kuitenkin arvo ensin kanadalaisten määrä käyttäen suoraan funktiota `rbinom` käyttäen otoskokona (`size`-argumentti) hotellin asukkaiden määrää. Laske tämän jälkeen niiden kanadalaisten määrä, joiden alushousut ovat värilliset, jälleen arpomalla binomijakaumasta, tällä kertaa otoskokona tietenkin kanadalaisten määrä. Sen jälkeen arvo vastaavasti kuinka monella amerikkalaisista on värilliset alushousut, ja laske kanadalaisten värillisalushousuisten osuus kaikista värillisalushousuisista.

   b) Testaa otoskooolla 1000000, että funktiosi antaa samansuuntaisen tuloksen kuin edellisissä tehtävissä.

   c) Vertaa funktioiden `hotellisimulaatio2` ja `hotellisimulaatio3` nopeutta otoskoolla 1000000 käyttämällä funktiota `system.time`. Kumpi oli nopeampi, ja oliko ero selkeä? Jos nopeuseroa on, mistä arvelet sen johtuvan?