TEKNILLINEN TIEDEKUNTA

# Kokeellisen datan käsittely ja analysointi R:llä

Teemu Pätsi

Ympäristötekniikka

Kandidaatintyö

Huhtikuu 2018

TEKNILLINEN TIEDEKUNTA

# Kokeellisen datan käsittely ja analysointi R:llä

Teemu Pätsi

Ohjaaja(t): Aki Sorsa

Ympäristötekniikka

Kandidaatintyö

Huhtikuu 2018

# TIIVISTELMÄ
# OPINNÄYTETYÖSTÄ Oulun yliopisto Teknillinen tiedekunta

| Koulutusohjelma (kandidaatintyö, diplomityö)<br>Ympäristötekniikka | | Pääaineopintojen ala (lisensiaatintyö) | |
|---|---|---|---|
| Tekijä<br>Pätsi, T. | | Työn ohjaaja yliopistolla<br>Sorsa, A. Tutkijatohtori | |
| Työn nimi<br>Kokeellisen datan käsittely ja analysointi R:llä | | | |
| Opintosuunta<br>- | Työn laji<br>Kandidaatintyö | Aika<br>Huhtikuu 2018 | Sivumäärä<br>38 |


Tiivistelmä

Työn tavoitteena on tutkia ja käyttää R-ohjelmistoympäristön eri vaihtoehtoja kokeellisen datan käsittelyyn ja analysointiin. Esiteltyjä asioita voidaan käyttää R:n monipuolisimpien menetelmien pohjatietoina sekä R:stä kiinnostuneelle aloittelijalle toimivana oppaana. Työssä esitellään R:n peruskäyttöä, peruskomennot datan tuontiin ja tallentamiseen ja peruskomentoja aineiston muokkaamiseen. Aineistoja voidaan haluttaessa kuvata erilaisilla graafisilla kuvaajilla. Työssä esitellään, miten useita erilaisia kuvaajia voidaan R:llä piirtää. Lisäksi kerrotaan hieman mahdollisista satunnaisaineistojen luomisesta R:llä.

Kokeellisen datan käsittelyssä voidaan analysoida aineistoja erilaisilla tilastollisilla tunnusluvuilla ja käyttää graafisia menetelmiä aineistojen muuttujien tutkimiseen. Regressioanalyysillä voidaan tutkia eri muuttujien riippuvuuksia ja muodostaa malleja, joilla aineiston lähtömuuttujaa voidaan ennustaa. Residuaaleja tutkimalla voidaan löytää puutteita muodostetussa mallissa.


Muita tietoja

# SISÄLLYSLUETTELO

# 1 JOHDANTO

Tämän työn aiheena on kokeellisen datan käsittely ja analysointi R:llä. Tämä aihe valittiin, sillä se on sisällöltään mielenkiintoinen ja uutta asiaa sisältävä, josta voi olla hyötyäkin tulevaisuudessa. Työssä on tavoitteena käydä läpi R:ssä olevia mahdollisuuksia datan käsittelyn ja analysoinnin pohjalta sekä niiden avaaminen selkokielelle. Työssä ei mennä niin pitkälle kuin R:ssä on mahdollista mennä, vaan rajataan alueeksi peruskomennot, graafiset esitykset, tilastolliset tunnusluvut ja muutama analyysimenetelmä. Tämä työ voi toimia hyvänä pohjana R:stä kiinnostuneelle ja sitä opettelevalle henkilölle.

R on S-kielestä kehitetty tilastollinen ja graafinen kieli ja ympäristö. R mahdollistaa erilaisten tilastollisten operaatioiden tekemisen ja niiden esittämisen kuvaajina ja taulukkoina. Lisäksi R on helposti laajennettavissa, sillä se perustuu aitoon tietokonekieleen, joten käyttäjät voivat ohjelmoida siihen uusia funktioita. R kehitettiin Yhdysvalloissa Bell Laboratories -organisaatiossa John Chambersin ja hänen kollegoidensa toimesta käyttäen alkupohjana S-kieltä. Suuri osa S-kielen koodista toimiikin suoraan R:ssä. R:n laajennettavuus perustuu käyttäjien tekemiin paketteihin, jotka lisäävät käytettäväksi lisäfunktioita. (The R Foundation, 2017)

R on jatkuvasti kehittyvä ja uusia päivityksiä julkaistaan usein. Tämän kandidaatintyön tekoaikanakin on julkaistu kaksi uutta versiota, jotka sisältävät bugien korjauksia ja optimointia. Lisäksi R:llä on käytössä monia käyttöä helpottavia käyttöliittymiä, joiden avulla R:n käyttö nopeutuu ja yksinkertaistuu R:n alkuperäiseen versioon verrattuna (RStudio, 2018). Tässä työssä käytetään tosin vain alkuperäistä R-käyttöliittymää.

R-projektilla on vertaisarvioitu ja vapaassa käytössä oleva julkaisu The R Journal. Siinä julkaistaan R:n liittyviä artikkeleita, jotka voivat olla mielenkiintoisia R:n käyttäjille ja kehittäjille, sekä myös R:n päivityksiä, lisäpaketteja ja R:ään liittyviä konferensseja käsitteleviä julkaisuja. (The R Foundation, 2018)

# 2 R:N PERUSKÄYTTÖ

Työssä käytetyt R:n komennot on esitetty ”>”-merkillä ennen komentoa.

## 2.1 R laskukoneena

R:ssä komentojen kanssa käytetään sulkuja, esimerkiksi kirjoittaessa ainoastaan >quit ohjelma ei sulje sovellusta, vaan kertoo, mitä lisätietoja quit-komennon kanssa voi käyttää. Kirjoittaessa >quit() ohjelma kysyy haluatko tallentaa työtilan, jonka jälkeen se sulkeutuu. Ohjelma sulkeutuu komennolla suoraan, kun lisää sulkujen sisään >quit(save=”yes”), jolloin työtila tallentuu ja ohjelma sulkeutuu. Quit-komento voidaan myös lyhentää muotoon “q()”. Kaikista komennoista on saatavilla lisätietoja ”help()”-komennon avulla, tai lisäämällä ”?” komennon eteen. (Oksanen, 2003)

Muuttujiin voi tallentaa lukuja ja laskuja käyttäen sijoitusoperaattoria “=” tai “<-“. R:n laskutoimituksiin kuuluu peruslaskut “+” ja “-“ sekä potenssi-, kerto- ja jakolasku. Potenssilaskua kuvaa “^”, kertolaskua “*” ja jakolaskua “/”. (Oksanen, 2003)

```
> r=2
> r^2
[1] 4
> 3*r
[1] 6
> r/4
[1] 0.5
```

Lisäksi laskuissa voidaan käyttää neliöjuurta “sqrt()”, logaritmeja “log()”, “log10()”, “log2()”, eksponenttifunktiota “exp()”, sekä trigonometrisiä funktioita “sin()”, “cos()” ja “tan()”. Lisäksi voidaan käyttää arkusfunktioita lisäämällä “a” trigonometrisen function eteen, “asin()”. (Oksanen, 2003)

```
> sqrt(3)
[1] 1.732051
```

Komennolla “options()” voidaan muokata näytettävien lukujen määrää.

```
> options(digits=4)
> sqrt(3)
[1] 1.732
```

Lukuja voidaan myös pyöristää komennolla “round()”, jonka lisäparametrina voidaan käyttää “digits=”, jonka avulla haluttua tarkkuutta voidaan muokata.

```
> round(sqrt(3),digits=2)
[1] 1.73
```

## 2.2 Vektoreiden muodostaminen ja niihin viittaaminen

R:ssä voidaan muodostaa vektoreita komennon “c()” avulla.

```
> X=c(3, 1.2, 0.25, 2.28)
> X
[1] 3.00 1.20 0.25 2.28
```

Vektorin tiettyyn alkioon voidaan viitata lisäämällä hakasulkujen sisään alkion numero. Vektorin pituus saadaan funktiolla ”length()”.

```
> X[2]
[1] 1.2
> length(X)
[1] 4
```

Vektoreita voidaan käyttää laskuissa, jolloin tehtävä laskuoperaatio toistetaan joka alkiolle erikseen.

```
> 2*X
[1] 6.00 2.40 0.50 4.56
> 2+X
[1] 5.00 3.20 2.25 4.28
```

Vektoreita muodostaessa voidaan myös käyttää laskuja. Alla olevassa esimerkissä käytetään aiemmin tallennettua r-muuttujaa, jonka arvo on 2.

```
> Y=c(r+2, r-2)
> Y
[1] 4 0
```

Vertailuoperaattoreita R:ssä ovat “>”, “<”, “>=” sekä “<=”. Saatuja arvoja voidaan käyttää muodostamaan looginen vektori, jonka mahdolliset arvot ovat TRUE tai FALSE. Nämä voidaan lyhentää muotoon T ja F. (Oksanen, 2003)

```
> X>1
[1]  TRUE  TRUE FALSE  TRUE
```

Yhdistämällä komentoja voidaan viitata vain alkioihin, joiden arvo saa arvon TRUE. (Oksanen, 2003)

```
> X[X>1]
[1] 3.00 1.20 2.28
```

Vektoreita voidaan manipuloida monella eri tavalla. Esimerkiksi alkiot voidaan järjestää suuruusjärjestykseen komennolla “sort()”. (Oksanen, 2003)

```
> sort(X[X>1])
[1] 1.20 2.28 3.00
```

Aiemmin mainittu “round()” komento toimii myös vektorin alkioiden pyöristykseen. Pyöristyksessä käytettävää tarkkuutta voidaan vaihtaa lisämääreellä ”digits”.

```
> round(X)
[1] 3 1 0 2
> round(X, digits=1)
[1] 3.0 1.2 0.2 2.3
```

Kokonaislukusarjoja voidaan muodostaa komennolla “lähtöluku:loppuluku”. (Oksanen, 2003)

```
> Z=1:10
> Z
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
> Z[4:8]
[1] 4 5 6 7 8
```

Usein tarvitaan myös tietyn pituisia vektoreita, jolloin voidaan käyttää komentoa ”seq()”, jonka parametri ”length” kertoo kuinka monialkioista vektoria muodostetaan. (Oksanen, 2003)

```
> Z=seq(0,5,length=10)
> Z
 [1] 0.0000000 0.5555556 1.1111111 1.6666667 2.2222222 2.7777778 3.3333333
 [8] 3.8888889 4.4444444 5.0000000
```

Käyttämällä lisäparametria ”by” seq-komennon kanssa voidaan muodostaa tasavälinen vektori, jonka alkioiden välinen ero määräytyy ”by”-parametrillä. Lisäksi voidaan muodostaa aiemmin määritellyn vektorin mittainen vektori lisäparametrillä ”along=”. (Oksanen, 2003)

```
> Z=seq(0,5,by=0.5)
> Z
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> Z=seq(along=X)
> Z
[1] 1 2 3 4
```

## 2.3 NaN ja Inf

Mikäli vektorilla tehtävässä laskussa muodostuu määrittelemättömiä arvoja, alkio saa arvon NaN, ”not-a-number”. Nämä pitää poistaa vektorista, mikäli halutaan tehdä laskuja, koska NaN-arvot laskuissa antavat tuloksena NaN. Poistaminen onnistuu parametrillä ”na.rm=T” useimmissa komennoissa. Alla olevassa esimerkissä käytetty ”mean()” laskee keskiarvon vektorille. (Oksanen, 2003)

```
> P=-2:5
> P
[1] -2 -1  0  1  2  3  4  5
```

```
> mean(sqrt(P))
[1] NaN
Warning message:
In sqrt(P) : NaNs produced
> mean(sqrt(P), na.rm=T)
[1] 1.397055
```

Joskus laskussa muodostuu myös äärettömiä lukuja ”Inf” ja ”-Inf”, jotka tekevät myös laskujen tuloksista äärettömiä. Tällöin voidaan haluta laskea vain vektorin äärellisillä arvoilla. Äärettömät arvot saadaan poistettua komennon ”is.finite()” avulla, joka palauttaa TRUE ja FALSE arvoja riippuen siitä, onko luku äärellinen vai ääretön. (Oksanen, 2003)

Muita mahdollisia is.-alkuisia komentoja ovat esimerkiksi aiemman komennon vastakohta ”is.infinite()”sekä ”is.numeric()”, ”is.logical()”, ”is.matrix()” ja ”is.vector()”, jotka kertovat vektorin tai matriisin ominaisuuksia. ”is.na()” kertoo mitkä alkiot ovat määrittelemättömiä arvoja.

```
> log(P)
[1]      NaN      NaN     -Inf 0.0000000 0.6931472 1.0986123 1.3862944
[8] 1.6094379
Warning message:
In log(P) : NaNs produced
> mean(log(P), na.rm=T)
[1] -Inf
> mean(log(P[is.finite(log(P))]))
[1] 0.9574983
```

## 2.4 Matriisit ja taulukot

R:ssä matriisien muodostus onnistuu komennoilla ”cbind()” ja ”rbind()”, jotka muodostavat vektoreista matriisin sarakkeita tai rivejä. (Oksanen, 2003)

```
> kk=1:12
> mittaus=c(35, 31,28,90,40,60,35,78,96,31,26,14)
```

```
> taulukko= cbind(kk,mittaus)
> taulukko
      kk mittaus
 [1,]  1      35
 [2,]  2      31
 [3,]  3      28
 [4,]  4      90
 [5,]  5      40
 [6,]  6      60
 [7,]  7      35
 [8,]  8      78
 [9,]  9      96
[10,] 10      31
[11,] 11      26
[12,] 12      14
> rbind(kk,mittaus)
         [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
kk          1    2    3    4    5    6    7    8    9   10    11    12
mittaus    35   31   28   90   40   60   35   78   96   31    26    14
```

Toinen tapa muodostaa matriiseja on komennolla ”matrix()”. Tämä muodostaa yhdestä vektorista matriisin. Matriisin täyttöjärjestyksen voi vaihtaa sarakkeista riveihin parametrilla ”byrow=T”. (Maindonald, 2008)

```
> matrix(c(0.25, 1, 0, 0.5),nrow=2, ncol=2)
     [,1] [,2]
[1,] 0.25  0.0
[2,] 1.00  0.5
> M=matrix(c(0.25, 1, 0, 0.5),nrow=2, ncol=2,byrow=T)
     [,1] [,2]
[1,] 0.25  1.0
[2,] 0.00  0.5
```

Matriisin transpoosi saadaan komennolla ”t()”. (R Core Team, 2018)

```
> t(M)
```

```
     [,1] [,2]
[1,] 0.25  0.0
[2,] 1.00  0.5
```

Matriisin tiettyyn alkioon viitattaessa käytetään hakasulkuja. Sarakkeeseen viitattaessa rivi jätetään pois. Komento ”taulukko[,2]” viittaa siis taulukon toiseen sarakkeeseen. Sarakkeen numeron sijasta voidaan käyttää myös sarakkeen muodostaneen vektorin nimeä, tässä tapauksessa ”mittaus”. (Oksanen, 2003)

```
> mean(taulukko[,2])
[1] 47
> mean(taulukko[,"mittaus"])
[1] 47
```

Matriisin ulottuvuudet saadaan tietoon (Oksanen, 2003) ja niitä voidaan myös muokata komennon ”dim()” avulla (R Core Team, 2018).

```
> dim(taulukko)
[1] 12  2
> dim(M)=c(1,4);M
     [,1] [,2] [,3] [,4]
[1,] 0.25    0    1  0.5
```

Matriisiin voidaan lisätä rivejä tai sarakkeita komentojen ”rbind()” ja ”cbind()” avulla tavallaan muodostaen uuden matriisin vanhasta lisäten vektoreita tai matriiseja siihen. (R Core Team, 2018)

```
> dim(M)=c(2,2);M
     [,1] [,2]
[1,] 0.25  1.0
[2,] 0.00  0.5
> M=cbind(M,c(0,0))
> M
     [,1] [,2] [,3]
[1,] 0.25  1.0    0
[2,] 0.00  0.5    0
```

```
> M=rbind(M,c(1,2,3))
> M
     [,1] [,2] [,3]
[1,] 0.25  1.0    0
[2,] 0.00  0.5    0
[3,] 1.00  2.0    3
```

Matriisin käänteismatriisi saadaan komennolla ”solve(matriisi)”. Lisäksi solve-komennolla voidaan ratkaista yhtälöstä Ax=b vektori x. (R Core Team, 2018)

```
> A=matrix(c(0.25, 1, 0, 0.5),nrow=2, ncol=2)
> solve(A)
     [,1] [,2]
[1,]    4    0
[2,]   -8    2
> b=rbind(1,2)
> solve(A,b)
     [,1]
[1,]    4
[2,]   -4
```

## 2.5 Aineiston lataaminen ja tallentaminen

R voi lukea ulkoisia tiedostoja ja mahdollistaa niiden käsittelyn R:n avulla. Komennot ulkoisten tiedostojen lukemiseen ovat esimerkiksi ”read.table()” ja ”read.csv()”, joiden lisäksi on muille aineistotyypeille omat komentonsa. Seuraavassa esimerkissä data.csv-tiedosto sisältää 357 riviä ja kolme saraketta, jotka tallennetaan data-muuttujaan. Lisämääre ”sep” kertoo R:lle, että sarakkeet on erotettu kyseisellä merkillä alkuperäisessä aineistossa. Muita lisämääreitä komennolle on esimerkiksi ”header”, joka kertoo R:lle, että ensimmäisellä rivillä on muuttujien nimet, sekä ”dec”, joka kertoo R:lle aineistossa käytetyn desimaalierottimen. Lisäksi voidaan kertoa luettavien rivien määrä lisämääreellä ”nrows”, jolloin lukeminen katkeaa tietyn rivin kohdalle. Tiedosto luetaan ”list”-tyyppisenä, eikä numeerisena matriisina, jolloin joidenkin toimintojen tekemiseksi täytyy muuttaa aineisto numeeriseksi komennolla ”unlist()”. Tallennettua data-muuttujaa käytetään myöhemmin regressioanalyysissä. Muuttujan data

tulostuksessa rivit on katkaistu työssä viiden rivin jälkeen. (Oksanen, 2003; Maindonald, 2008)

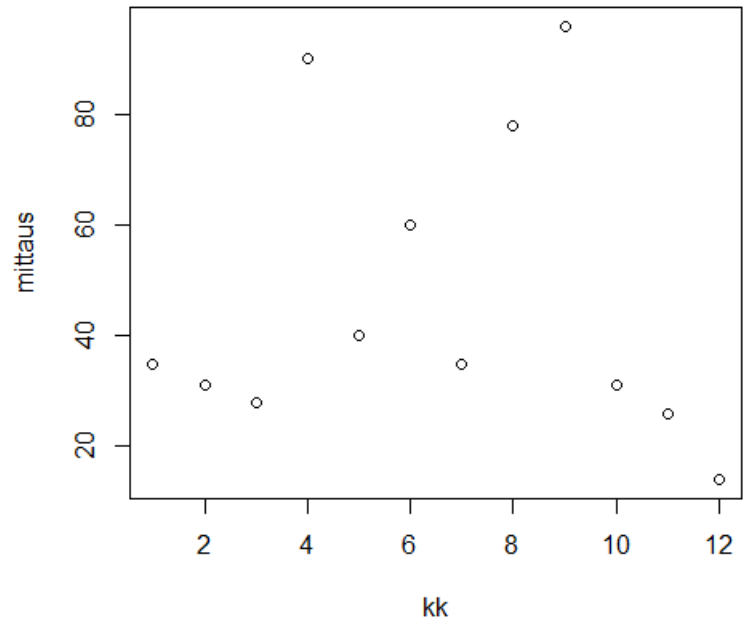
Komennot lukevat tiedostoja R:n aktiivisesta työskentelykansiosta, jota voidaan muuttaa komennolla ”setwd(kansion polku)” ja tarkistaa käytössä oleva kansio komennolla ”getwd()”.

```
> data=read.csv("data.csv", sep=";")
> data
     x1  x2     y
1   -5,0 -10  35,0
2   -5,0  -9  33,0
3   -5,0  -8  33,0
4   -5,0  -7  35,0
5   -5,0  -6  39,0
jne.
> is.matrix(data)
[1] FALSE
```

## 2.6 Kuvien piirtäminen

R:ssä onnistuu monenlaisten kuvaajien piirtäminen. Yksinkertaisimmillaan kuvaaja saadaan piirrettyä komennolla ”plot()”, joka piirtää pisteet kahden vektorin alkioista. Alla olevassa kuvassa 1 nähdään yksinkertaisin kuvaaja. (Oksanen, 2003)

```
> plot(kk,mittaus)
```

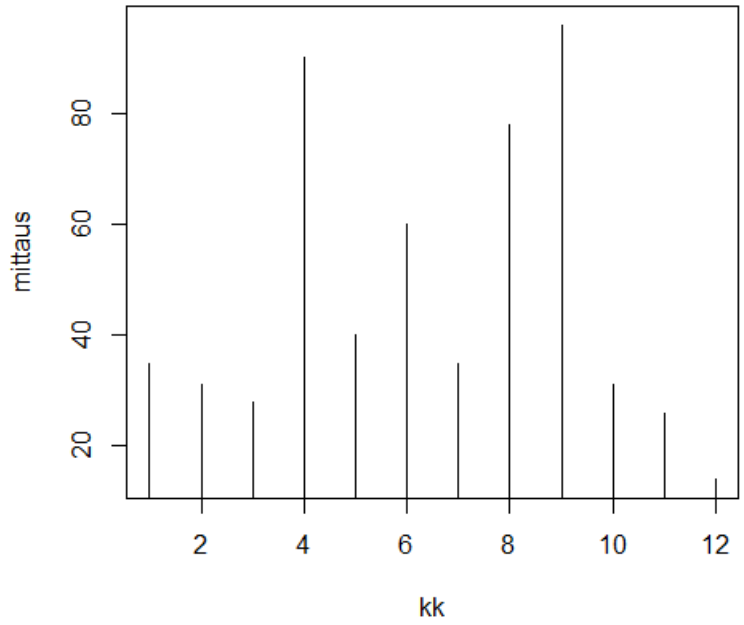



Kuva 1. Plot-komennon tuottama kuvaaja pisteistä

Plot-komentoon voidaan lisätä lisämääreitä, joiden avulla kuvaaja voidaan muuttaa helpommin luettavaksi. Parametrillä ”type” määrätään kuvan tyyppi, esimerkiksi p pelkälle pistekuvaajalle (oletus), l, joka piirtää viivat pisteiden mukaisesti sekä b, joka yhdistää piste- ja viivakuvaajat. Kuvassa 2 on esitetty type=”h”-lisämääreen tuottama histogrammityyppinen kuvaaja. (Oksanen, 2003)

Kuvaajan akseleita voidaan myös muokata parametreilla. ”xlab”- ja ”ylab”-lisämääreitä käytetään akseleiden nimeämiseen ja ”xlim”- ja ”ylim”-lisämääreet muokkaavat akselien pituuksia. Lisäksi kuvaajien värejä voidaan muokata ”bg”, ”fg” ja ”col” parametreilla. (Oksanen, 2003)

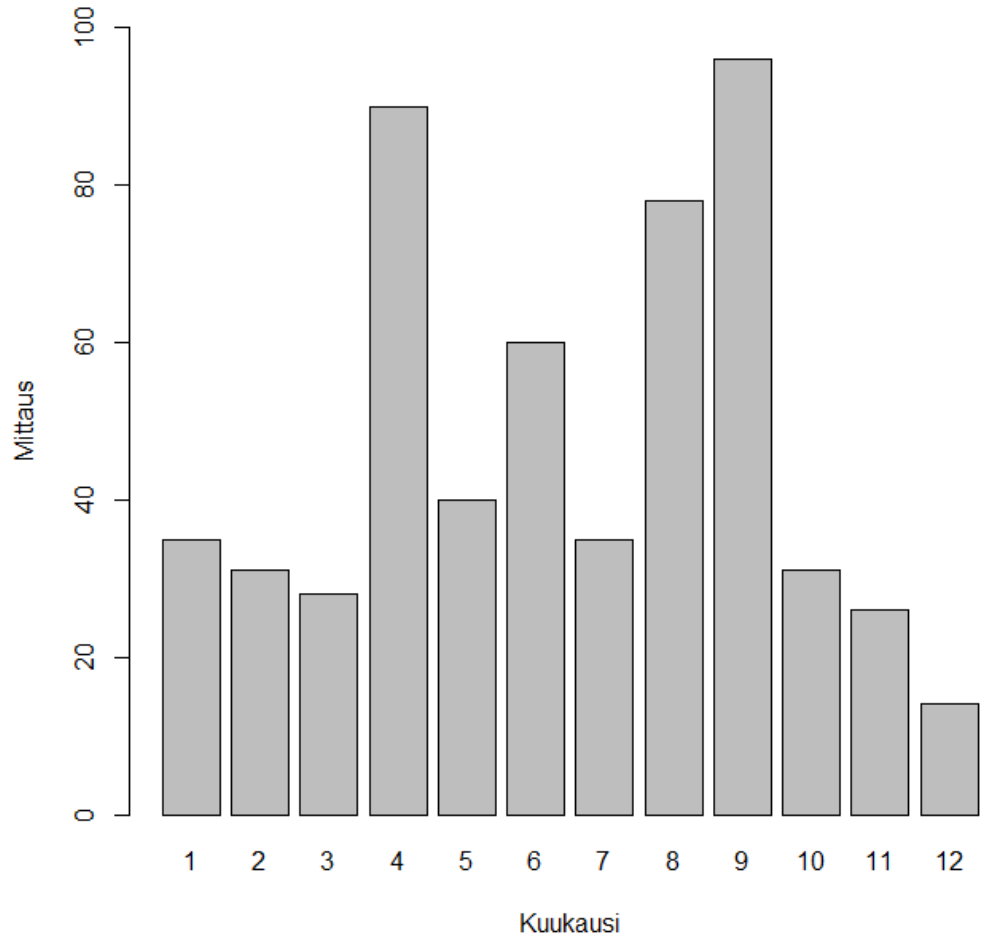
```
> plot(kk,mittaus,type="h")
```




Kuva 2. Lisämääreellä muodostettu histogrammityyppinen kuvaaja aineistosta

Aineistosta voidaan piirtää pylväsdiagrammi komennolla ”barplot()”, joka piirtää pylväitä aineiston mukaan (kuva 3). Tässä kuvaajassa ei automaattisesti nimetä akseleita, joten ne pitää määrittää aiemmilla lisämääreillä. Lisäksi nimetään pylväät vektorilla 1-12 lisämääreen ”names.arg” avulla. (Oksanen, 2003)

```
> barplot(mittaus,xlab="Kuukausi",ylab="Mittaus",ylim=c(0,100),names.arg=1:12)
```

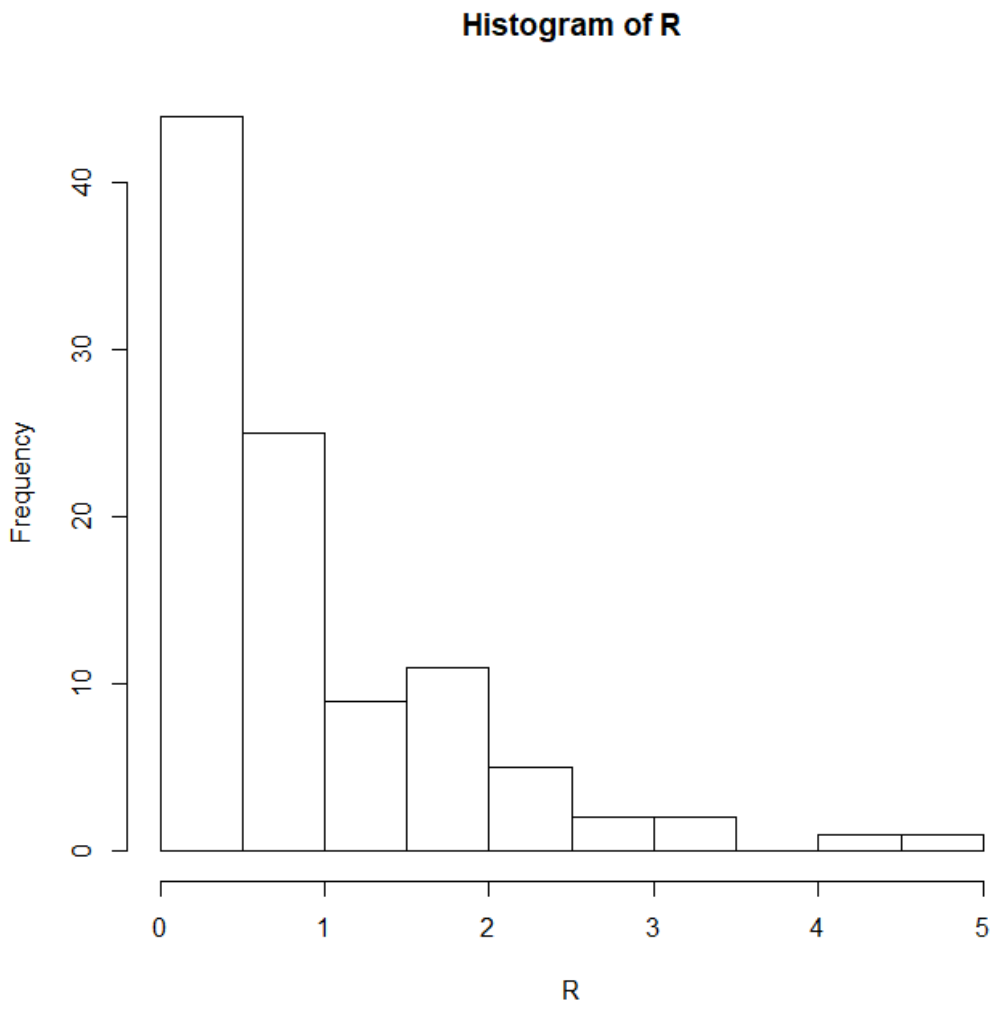



Kuva 3. Barplot-komennon muodostama pylväsdiagrammi aineistolle

Histogrammien piirtämiseen käytetään komentoa ”hist()”. Histogrammi on samankaltainen pylväsdiagrammin kanssa, mutta histogrammissa pylväiden korkeus edustaa muuttujien arvojen osuuksia aineistosta. (Oksanen, 2003)

Luodaan aineisto histogrammille komennolla ”rexp()”, joka luo satunnaislukuja siten, että alkiot noudattavat eksponenttijakaumaa. Alla olevassa kuvassa 4 on eksponenttijakautuneen aineiston histogrammi.

```
> R=rexp(100)
> hist(R)
```




Kuva 4. Histogrammi eksponenttijakaumaa seuraavalle aineistolle

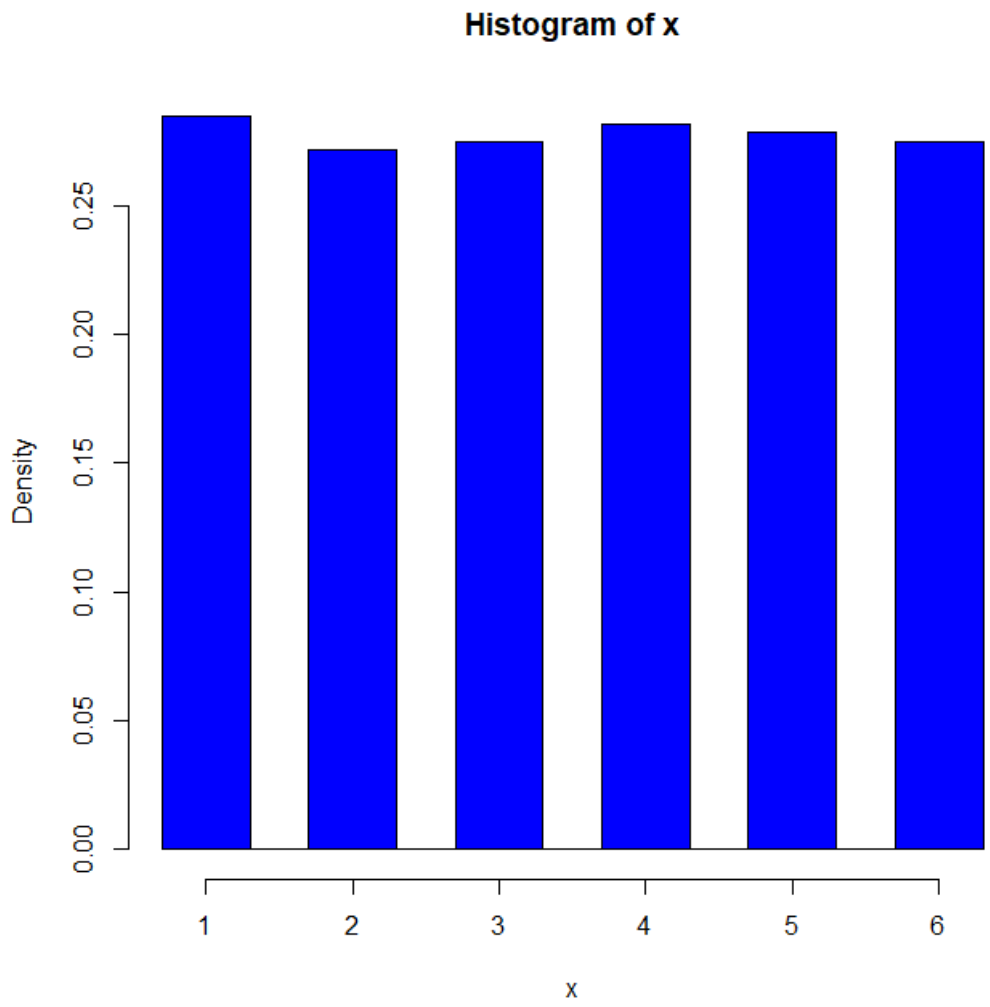
## 2.7 Satunnaisluvut

Yksinkertaisin satunnaislukuvalitsin saadaan komennolla ”sample()”, joka valitsee vektorilta satunnaisen alkion. Lisämääreellä ”size” voidaan valita, kuinka monta alkiota komento valitsee. Oletusasetuksena komento valitsee kaikki vektorin alkiot kerran, luoden saman vektorin eri järjestyksessä. Asetuksella ”replace=T” komento voi valita saman alkion useamman kerran, jolloin myös useamman kuin yhden alkion satunnainen valinta onnistuu samalla todennäköisyydellä joka kerta. Seuraava esimerkki heittää noppaa 1000 kertaa. (Maindonald, 2008)

```
> x=sample(1:6,size=1000,replace=T)
```

Aineistosta tehdään histogrammi kuvaan 5, joka muokataan helpommin luettavaksi lisämääreen ”breaks” avulla.

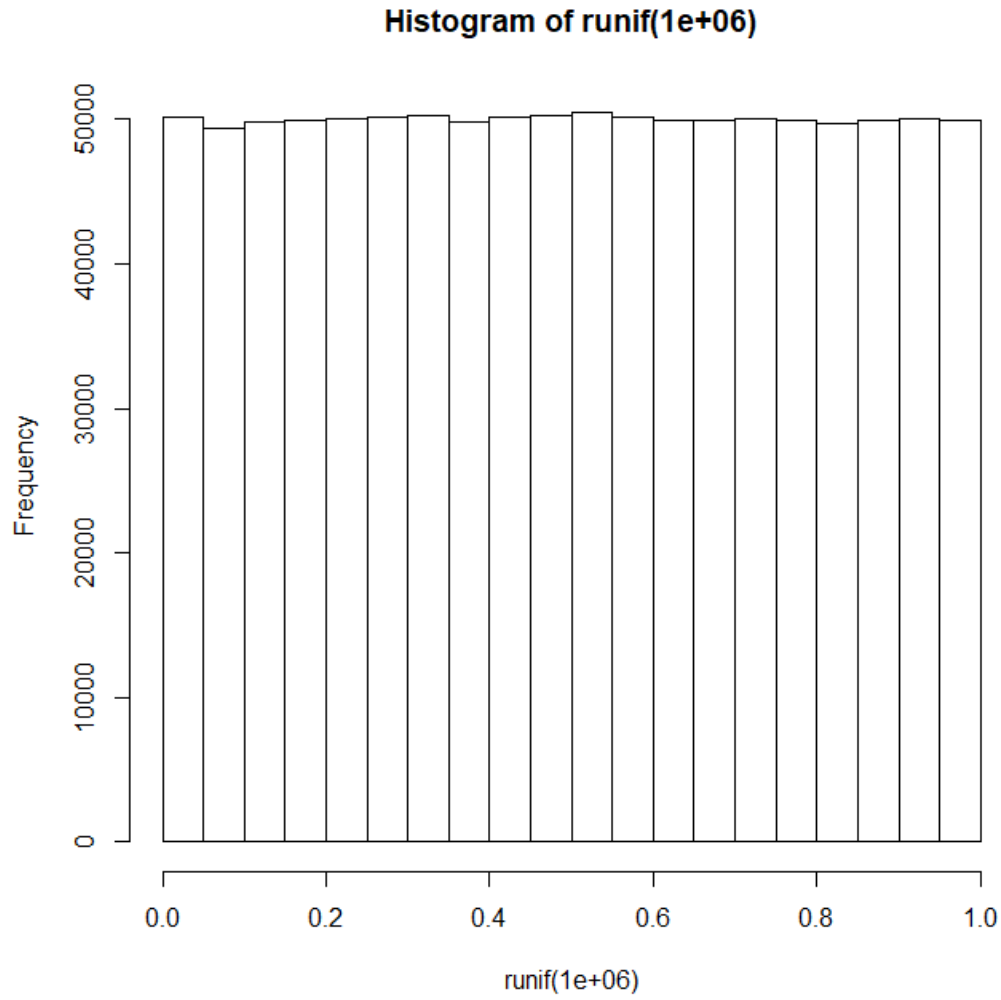
> hist(x,col=4,breaks=rep(1:6,each=2)+c(-.3,.3))




Kuva 5. Histogrammi nopanheiton tuloksista

Tasaisen jakauman saa aikaan komennolla ”runif()” (Oksanen, 2003). Tuotettavat alkiot ovat väliltä nollasta yhteen, joten keskiarvoksi tulee 0,5. Kuvassa 6 on tasajakautuneelle aineistolle piirretty histogrammi.

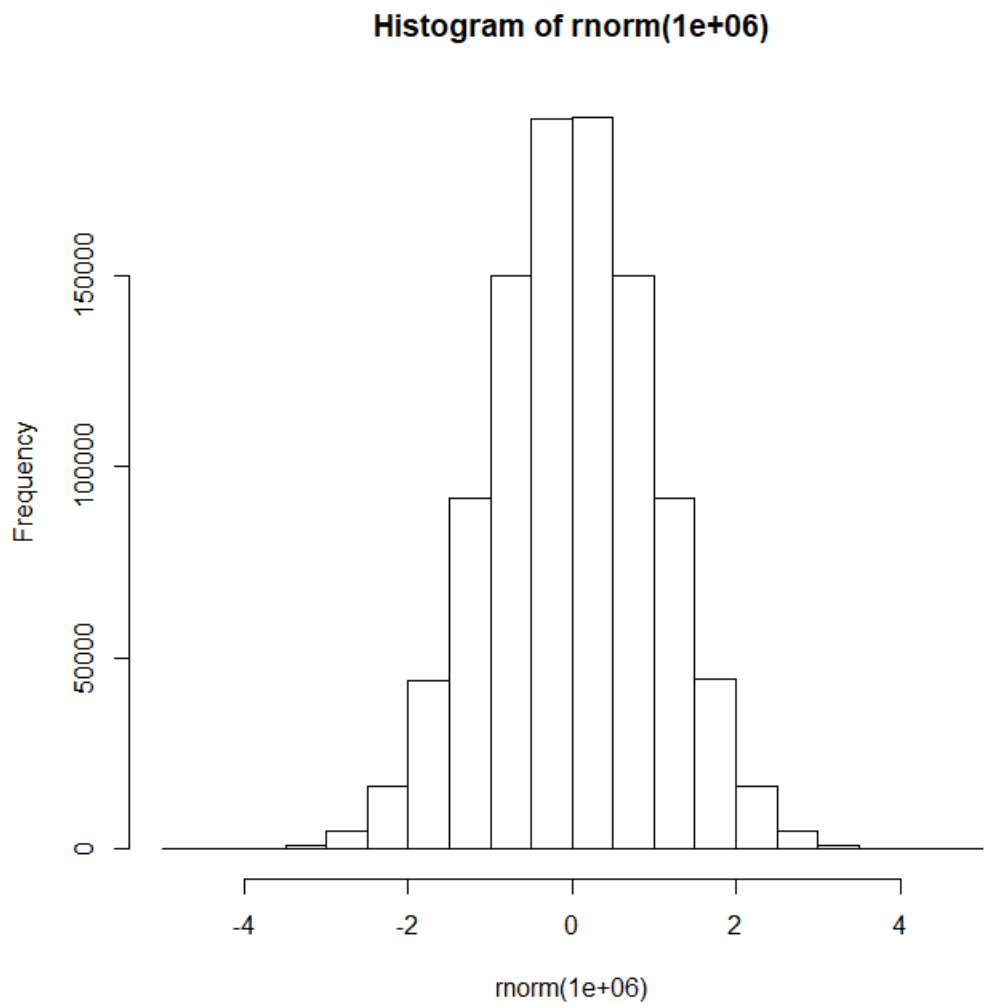
> hist(runif(1000000))




Kuva 6. Histogrammi tasajakautuneelle aineistolle

Normaalijakauma saadaan komennolla ”rnorm()” (Oksanen, 2003). Alkioiden keskiarvo on 0 ja keskijakauma on 1. Kuvassa 7 nähdään normaalijakautuneen aineiston histogrammi.

> hist(rnorm(1000000))



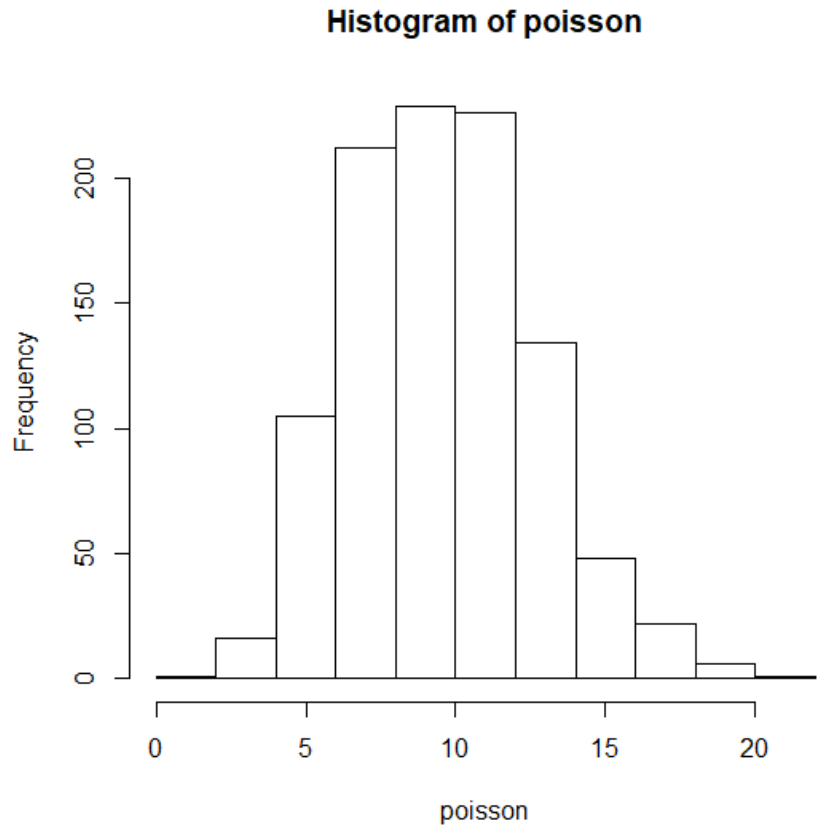

Kuva 7. Normaalijakautuneen aineiston histogrammi

Aiemmin käytetty ”rexp()”-komento luo eksponenttijakauman mukaisia alkioita (Oksanen, 2003). Lisämääre ”rate” muuttaa keskiarvon suuretta, 1/(rate). Esimerkiksi ”rate=10” antaa keskiarvoksi 0.1 = 1/10.

```
> mean(rexp(100000, rate=10))
[1] 0.09998889
```

Näiden lisäksi on myös Poissonin jakauma, ”rpois(määrä,lambda)”, jonka histogrammi on piirretty alle kuvaan 8. (Oksanen, 2003)

```
> poisson=rpois(1000,10)
> hist(poisson)
```




Kuva 8. Poisson-jakautunut aineisto

## 2.8 If-lause

R käyttää yleisenä ehtolauseena ”if” komentoa, jonka yleinen muoto on ”if (ehto) (komento1) else (komento2)”. If-lauseita voidaan yhdistää peräkkäin, jolloin voidaan toteuttaa komentoja monella eri ehdolla. Sopiva ehto if-lauseeseen saadaan loogisen muuttujan, TRUE tai FALSE, palauttavasta komennosta. (R Core Team, 2018)

Seuraavassa esimerkissä tehdään kahden nopanheiton vertailu if-lauseen avulla.

```
> heitto1=sample(1:6,size=1)
> heitto2=sample(1:6,size=1)
> if (heitto1<heitto2){
+ print("Toinen heitto suurempi")
+ } else if (heitto1>heitto2) {
+ print("Ensimmäinen heitto suurempi")
+ } else {
```

```
+  print("Sama molemmilla heitoilla") }
```

## 2.9 While-silmukka

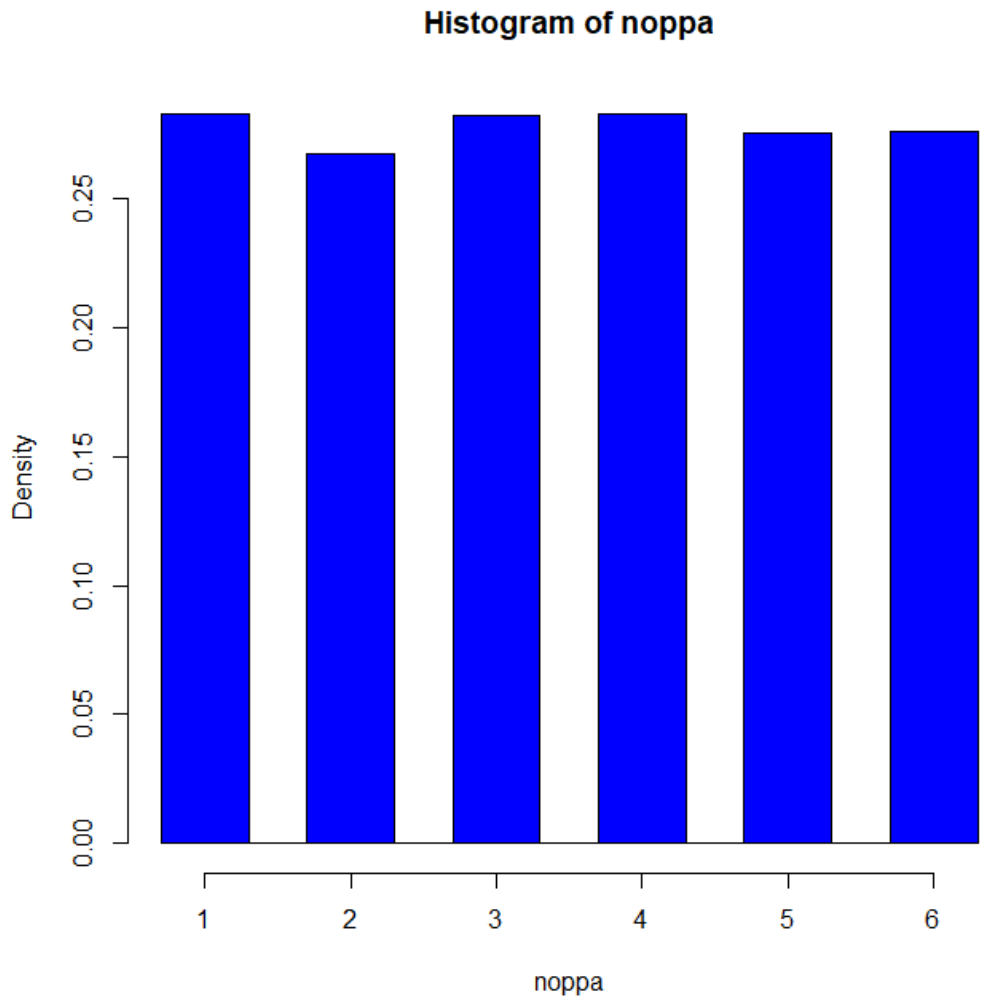
Komennon toistaminen niin kauan kuin jokin ehto on voimassa onnistuu komennolla ”while()”. Seuraavassa esimerkissä heitetään noppaa 10 kertaa ja tallennetaan heitot vektoriin. Yleinen muoto while-komennosta on ”while (ehto) (komento1 komento2)”. While-komennon toistaminen saattaa helposti jatkua äärettömästi, mikäli ehto ei ole hyvin määritelty eikä while-silmukan sisällä ole ehtoa lähestyvää komentoa, yleisimmin johonkin laskuriin tapahtuva yhteenlaskukomento. (R Core Team, 2018)

```
>laskuri=0
>i=1:10
>while (laskuri<10) {
+ i[laskuri+1]=sample(1:6,size=1)
+ laskuri=laskuri+1}
> i
 [1] 1 2 1 1 6 2 1 5 3 1
```

## 2.10 For-silmukka

Komennon toistaminen onnistuu R:ssä komennolla ”for()”. Yleinen muoto for-lauseelle on ”for([muuttuja] in 1:[toistojen määrä]) { (toistettavat komennot) }”. Alla on esimerkki nopanheitosta for-lauseella samalla tallentaen tulokset vektoriin, joka piirtää kuvan 9 histogrammin. (Maindonald, 2008; R Core Team, 2018)

```
>noppa=1:10000
> for(i in 1:10000)
+ noppa[i]=sample(1:6,size=1)
> hist(noppa,col=4,breaks=rep(1:6,each=2)+c(-.3,.3))
```




Kuva 9. For-silmukalla muodostettujen nopanheittojen histogrammi

# 3 KOKEELLISEN DATAN KÄSITTELY

## 3.1 Tilastolliset tunnusluvut

Usein aineistojen arviointiin tarvitaan lisätietoa mittauksista ja vektoreista tilastollisten tunnuslukujen muodossa. Yleisimpiin komentoihin kuuluu aiemmin mainittu keskiarvo, ”mean()”, varianssi ”var()”, keskihajonta ”sd()” ja mediaani ”median()”. Lisäksi on myös kvantiilit ”quantile()” eli minimi, alakvartaali eli 25 % aineistosta, mediaani eli puolet aineistosta, yläkvartaali eli 75 % aineistosta ja maksimi, sekä ”fivenum()”, joka eroaa kvantiileista siten, että toinen ja neljäs arvo ovat mediaanit vastaavista puolikkaista aineistoista. (Oksanen, 2003)

```
> var(mittaus)
[1] 738.1818

> sd(mittaus)
[1] 27.1695

> median(mittaus)
[1] 35

> quantile(mittaus)
  0%   25%   50%   75%  100%
14.00 30.25 35.00 64.50 96.00

> fivenum(mittaus)
[1] 14.0 29.5 35.0 69.0 96.0
```

Komento ”summary()” kertoo kvantiilit ja keskiarvon aineistolle.

```
> mittaus
 [1] 35 31 28 90 40 60 35 78 96 31 26 14

> summary(mittaus)
   Min.   1st Qu.  Median   Mean   3rd Qu.    Max.
  14.00   30.25     35.00     47.00    64.50     96.00
```

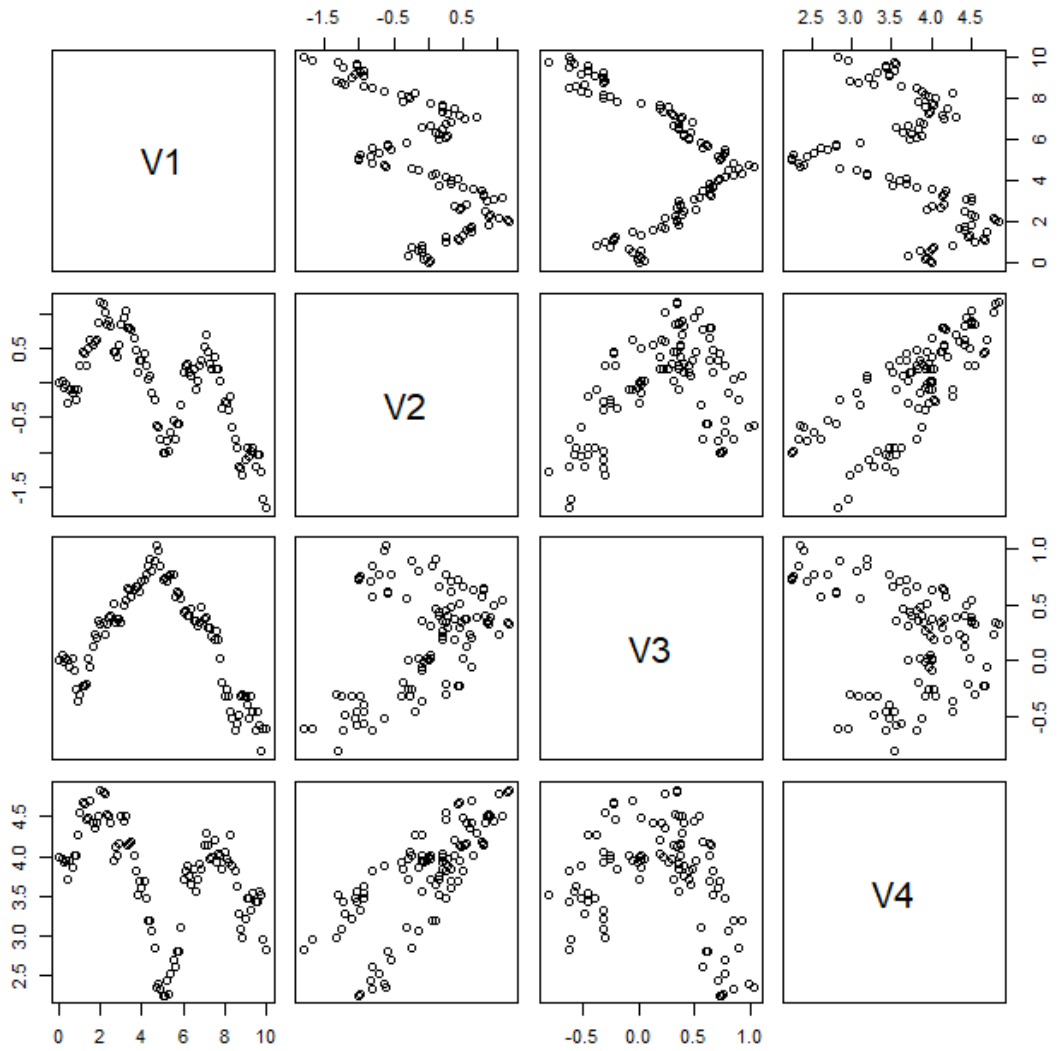
Näiden lisäksi voidaan laskea tunnusluvut vinous (skewness) ja huipukkuus (kurtosis), jotka eivät ole R:ssä suoraan, mutta lisäpaketeilla voi halutessaan lisätä funktion R:ään.

## 3.2 Graafiset menetelmät

Graafisiin menetelmiin kuuluu alla olevien menetelmien lisäksi myös histogrammi, joka on jo esitetty kappaleessa 2.6.

XY-kuvaaja ”scatterplot” saadaan ”plot()” komennolla suoraan. Komennolla ”abline()” voidaan sovittaa suora aineistoon. XY-kuvaajista on useammalle muuttujalle yhdistetty komento ”pairs()”, joka luo jokaiselle muuttujaparille oman kuvaajansa. Kuvassa 10 on esitetty ”pairsdata.txt”-tiedostosta tuodun aineiston XY-kuvaajat. (Oksanen, 2003)

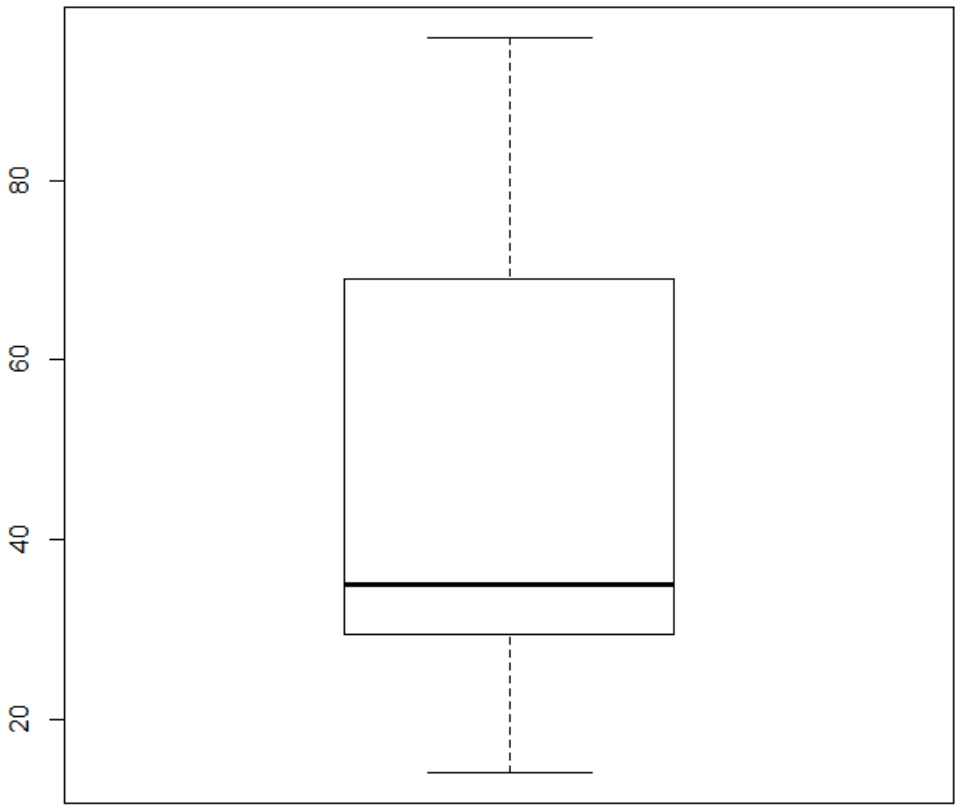
```
> pairsdata=read.table("pairsdata.txt",sep=",")
> pairs(pairsdata)
```




Kuva 10. Pairs-komennon luoma kuva muuttujaparien XY-kuvaajista

Tilastollisiin tunnuslukuihin perustuva kuvanesitys onnistuu ”boxplot()”-komennolla. Boxplot käyttää kuvaajan piirtämisessä fivenum-komennolla saatuja arvoja, eli minimiä, alapuolikkaan mediaania, koko aineiston mediaania, yläpuolikkaan mediaania ja maksimia. Mikäli aineistossa on muusta aineistosta reilusti eroava piste, se piirretään yksittäisenä pisteenä eroon boxplotista. Raja tälläisen ”outlier”-pisteen piirtämiseen erilleen on 1,5 kertaa sisälaatikon vaihteluväli ja sitä voi muuttaa lisämääreellä ”range”. Kun ”range” asetetaan nollaksi, ottaa R kaikki aineiston pisteet huomioon, mukaan lukien outlierit. Alla olevassa kuvassa 11 on piirretty boxplot aiemmin määritellylle ”mittaus”-aineistolle. (Oksanen, 2003)

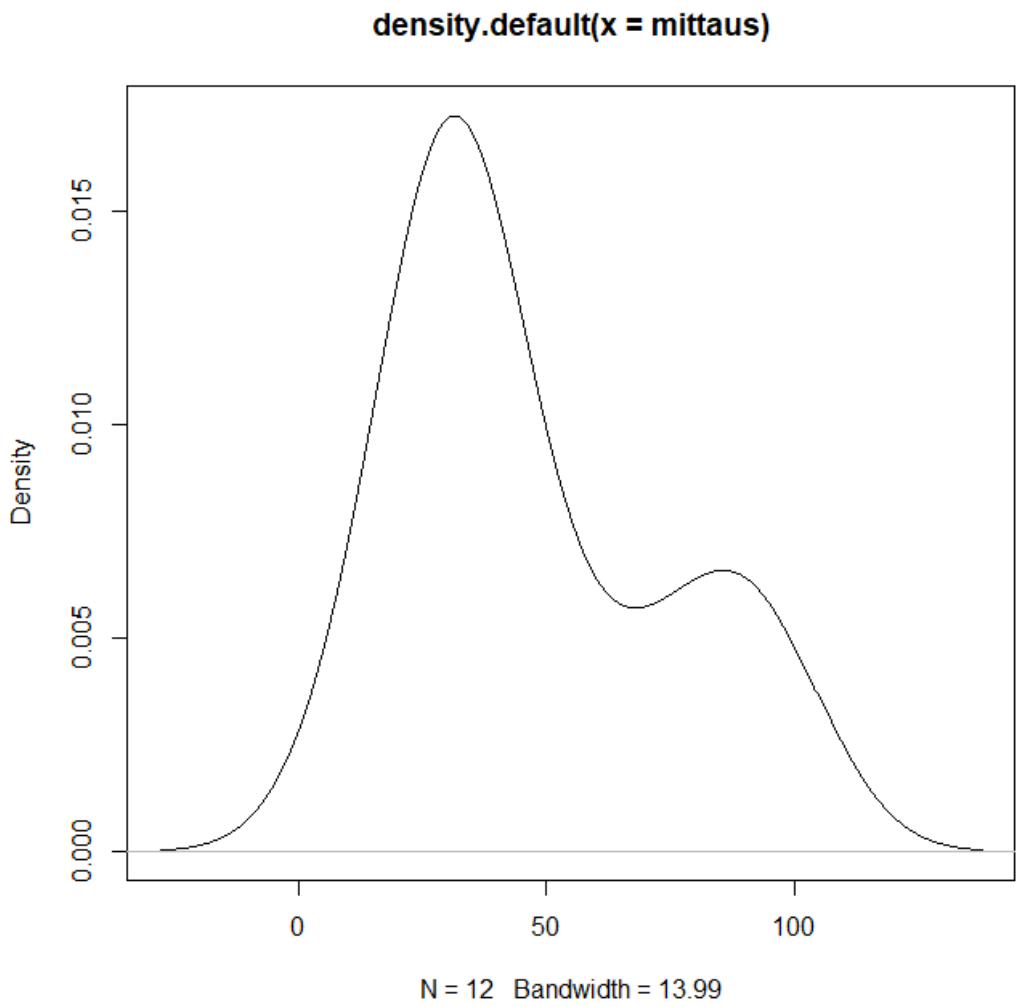
> boxplot(mittaus)




Kuva 11. Mittaus-aineiston boxplot. Tässä aineistossa ei ole outlier-pisteitä.

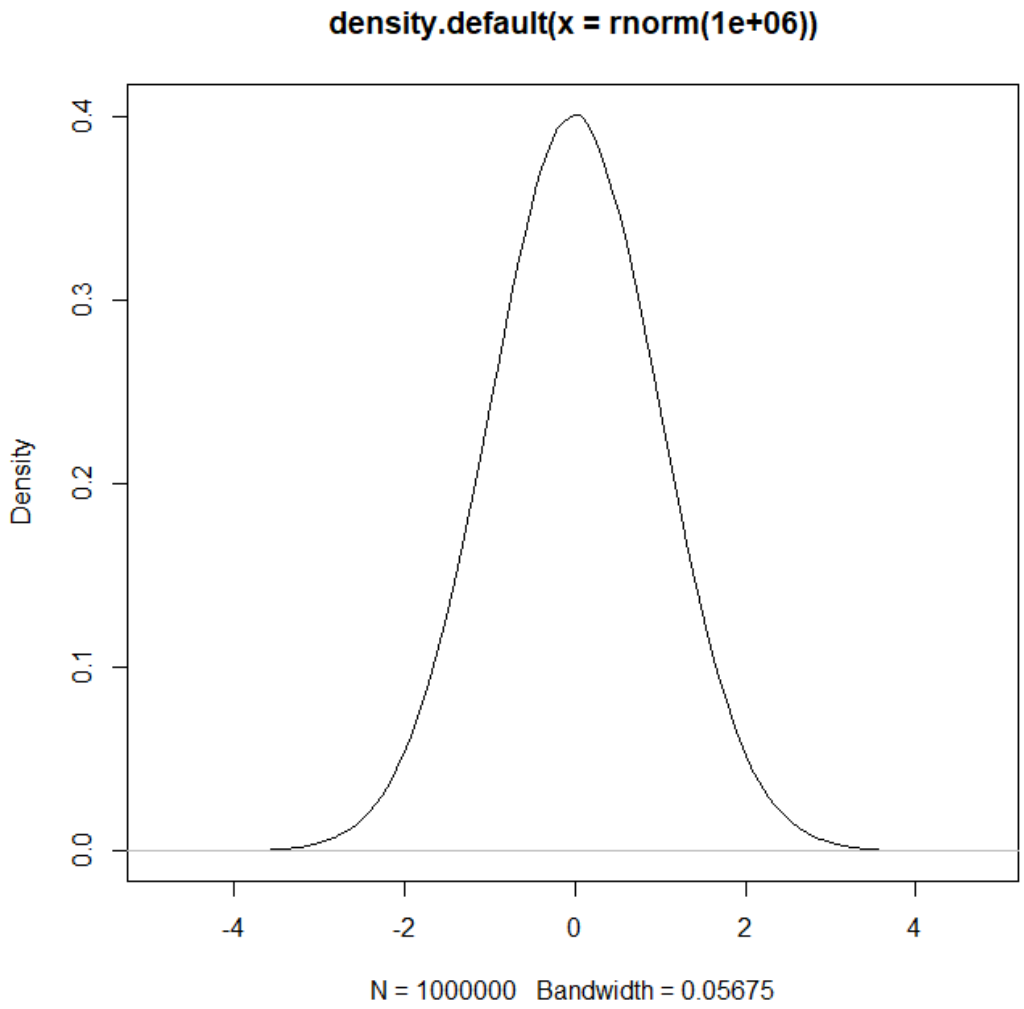
Tiheyskuvaajaa varten datapisteet lasketaan ”density()”-komennolla. Itse kuvaaja piirretään plot-komennon avulla. Komento soveltuu hyvin jatkuvan tiheyskuvaajan piirtämiseen histogrammin sijaan, sillä histogrammi jakaa aineiston palasiin ja kuvaajasta voidaan saada erilainen kuva. Kuvassa 12 on esitetty aiemmin määritellyn ”mittaus”-muuttujan tiheysfunktio ja kuvassa 13 normaalijakautuneen aineiston tiheysfunktio. Kuvasta 12 nähdään, että x-akselilla negatiivisella puolella tiheyskuvaaja jatkuu vielä, vaikka käytetyssä aineistossa ei ole lainkaan negatiivisia arvoja. Tiheyskuvaaja siis vääristää ääripäitä hieman. (Oksanen, 2003)

```
> plot(density(mittaus))
```




Kuva 12. Tiheysfunktion kuvaaja piirrettynä plot-komennon avulla

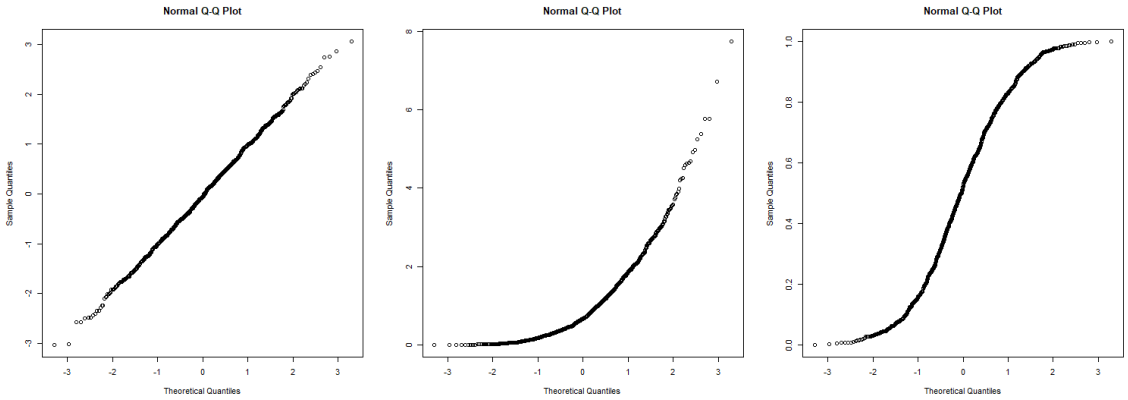
```
> plot(density(rnorm(1000000)))
```




Kuva 13. Normaalijakauman tiheyskuvaaja

Komennolla ”qqnorm()” muodostetaan aineiston normaalisuuskuvaaja (normal probability plot). Kun normaalisuuskuvaajan pisteet muodostavat suoran, aineisto on normaalijakautunut. Kuvassa 14 on esitetty normaalijakautuneen, eksponenttijakautuneen ja tasajakautuneen aineiston normaalisuuskuvaajat. (Oksanen, 2003)

```
> par(mfrow=c(1,3))
> qqnorm(rnorm(1000))
> qqnorm(rexp(1000))
> qqnorm(runif(1000))
```




Kuva 14. Normaalisuuskuvaajat a) normaalijakauneelle, b) eksponenttijakautuneelle ja c) tasajakautuneelle aineistolle

## 3.3 Lineaariset regressiomallit

Regressioanalyysissä aineistolle voidaan muodostaa erilaisia malleja. Tässä työssä rajoitutaan kuitenkin vain lineaarisiin malleihin.

### 3.3.1 Yhden muuttujan malli

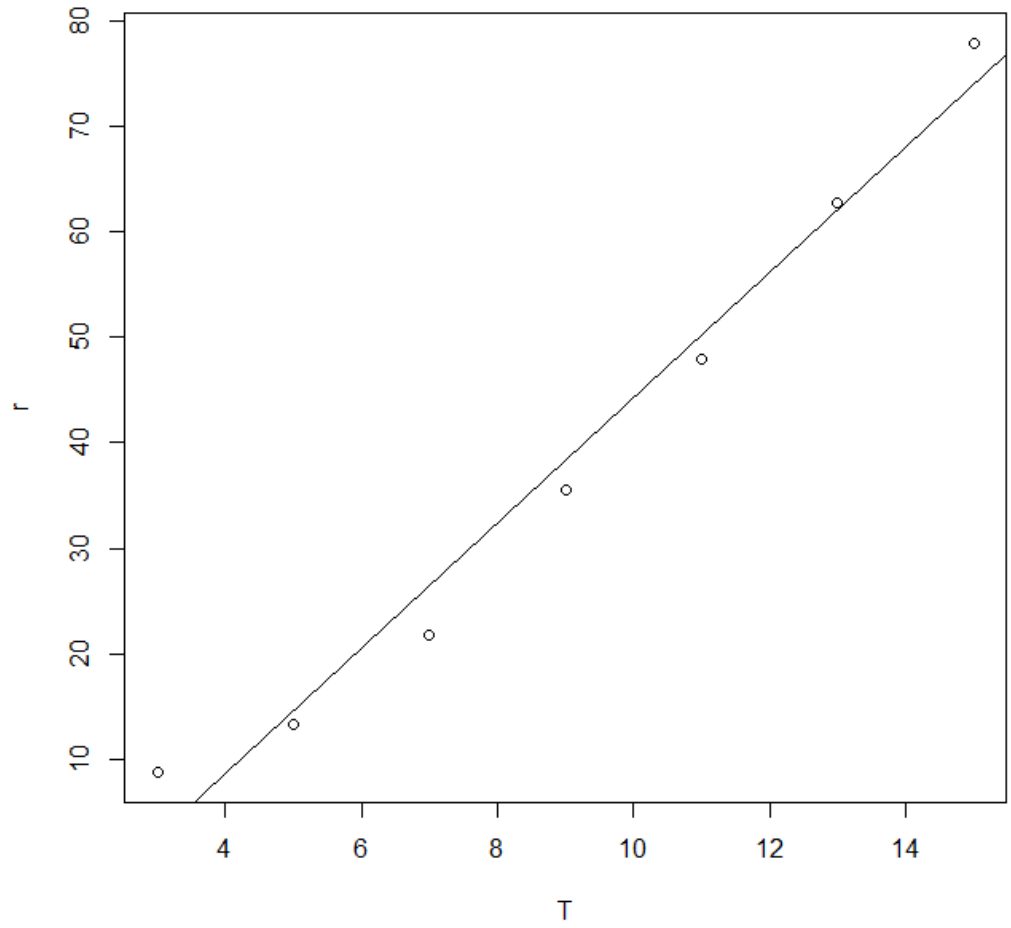
Yhden muuttujan regressiomallin muodostaminen onnistuu komennolla ”lm(y ~ x)”. Komento muodostaa lineaarisen mallin, jossa tutkitaan y:n riippuvuutta x:stä. Mallin ominaisuudet saadaan esiin komennolla “summary()”. (Oksanen, 2003)

```
> r=c(8.76,13.26,21.84,35.57,47.99,62.66,77.91)
> T=c(3,5,7,9,11,13,15)
> reg=lm(r~T)
> summary(reg)

Call:
lm(formula = r ~ T)
Residuals:
      1       2       3       4       5       6       7
 6.0900 -1.2814 -4.5729 -2.7143 -2.1657  0.6329  4.0114
```

```
Coefficients:
             Estimate   Std. Error      t value      Pr(>|t|)
(Intercept)  -15.1371    3.9084        -3.873       0.0117 *
T              5.9357    0.3968        14.957       2.42e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.2 on 5 degrees of freedom
Multiple R-squared:  0.9781,    Adjusted R-squared:  0.9738
F-statistic: 223.7 on 1 and 5 DF,  p-value: 2.418e-05
```

Komento antaa pitkän luettelon analyysin tuloksista. Ensimmäisenä kerrotaan residuaalit aineistolle sovitetun suoran ja pisteiden välille. Seuraavaksi kerrotaan regressiokertoimet ja niihin liittyviä merkitsevyyden arvioita. “Estimate” kertoo aineistolle sovitetun suoran kulmakertoimen, tässä tapauksessa r:n arvo nousee noin 5.5, kun T kasvaa yhden verran. Seuraavilla sarakkeilla on muuttujien standardivirhe “Std. Error” ja t-arvo, joka saadaan, kun jaetaan “Estimate” standardivirheellä. Tällä voidaan testata, eroaako laskettu kulmakerroin merkittävästi nollasta. Viimeisen sarakkeen arvo kertoo kertoimien p-arvon, joka kertoo todennäköisyyden sille, että tehdään väärä johtopäätös, jos muuttuja todetaan merkittäväksi. Seuraavaksi R tulostaa “R-squared”-arvot, jotka kertovat mallin selitysastetta, eli sitä, kuinka hyvin malli pystyy selittämään muuttujan varianssia. “Adjusted R-squared” ottaa lisäksi huomioon datapisteiden lukumäärän muuttujien lukumäärään nähden. Selitysasteen arvot vaihtelevat nollan ja yhden välillä. Viimeisellä rivillä saadaan F-testin tulos, joka arvioi mallin merkittävyyttä varianssianalyysillä arvioituna. Tulokselle voidaan piirtää pisteet ja sovittaa saatuihin pisteisiin suora komennoilla “plot()” ja “abline()”. Sovitetun suoran ja pisteiden kuvaaja on esitetty alla kuvassa 15. (Oksanen, 2003)

```
> plot(T,r)
> abline(reg)
```



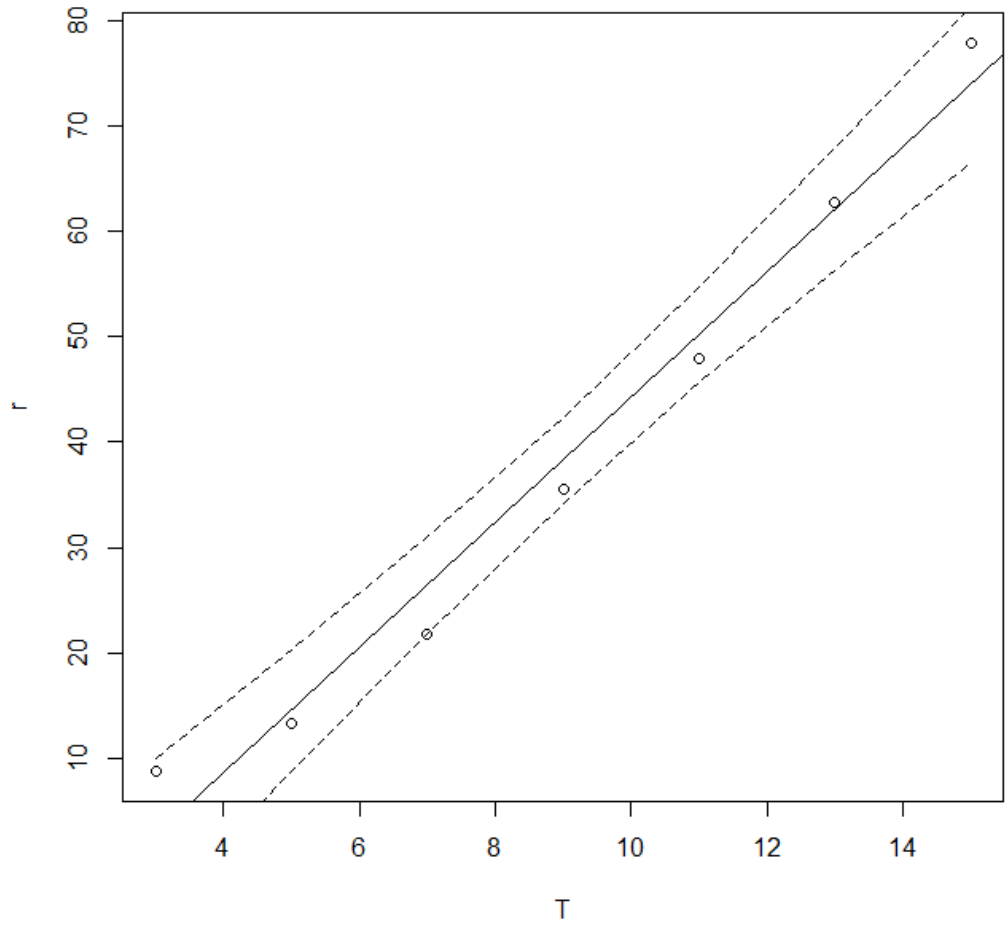

Kuva 15. Aineiston pisteet ja niihin sovitettu suora

Kuvaan voidaan lisätä ennustusteelle 95 % luottamusvälit käyttämällä komentoa “lines()”. Luottamusvälin ylä- ja alarajat saadaan komennolla “predict()” muodostetun matriisin sarakkeista. Luottamusvälin kuvaaja on esitetty katkoviivana lisämääreen ”lty=2” avulla kuvassa 16. (Oksanen, 2003)

```
> predict(reg,interval="confidence")

       fit       lwr       upr
1  2.67000 -4.686157 10.02616
2 14.54143  8.770784 20.31207
3 26.41286 21.850762 30.97495
4 38.28429 34.203824 42.36475
5 50.15571 45.593619 54.71781
6 62.02714 56.256498 67.79779
```

7 73.89857 66.542414 81.25473

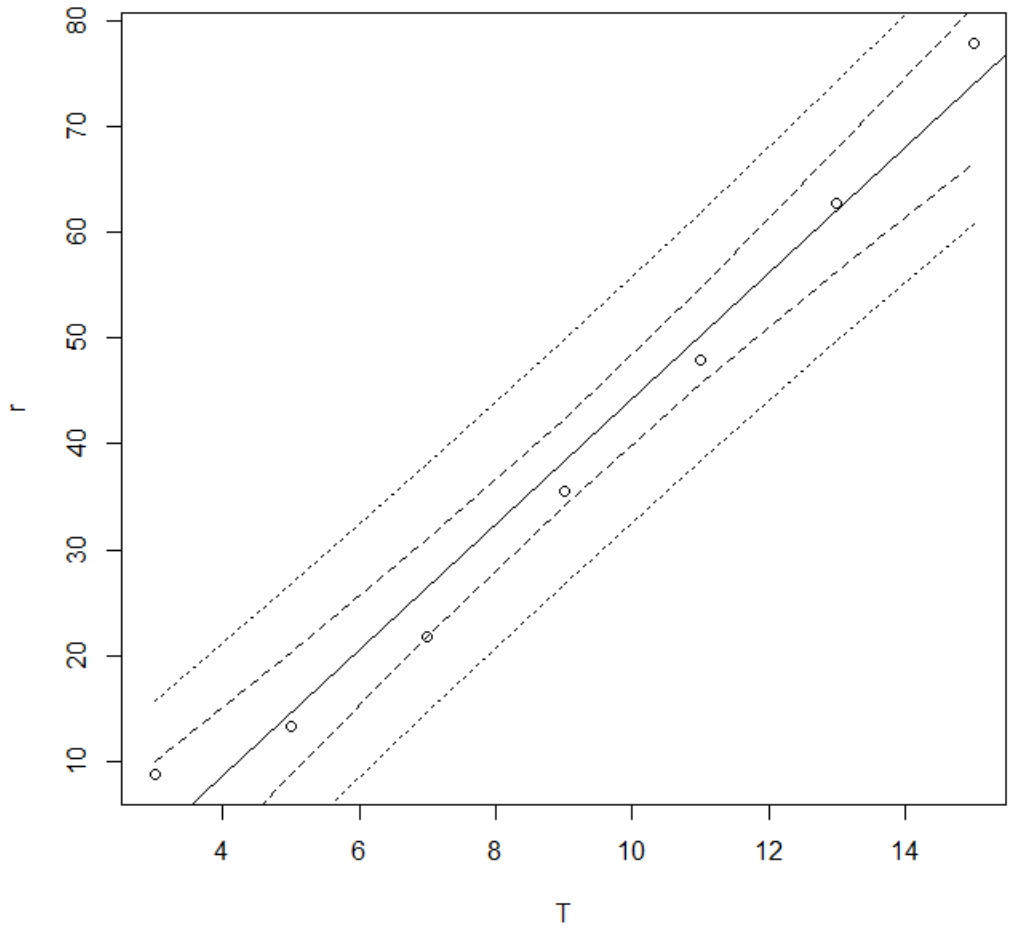
```
> lines(T,predict(reg,interval="confidence")[,"upr"],lty=2)
> lines(T,predict(reg,interval="confidence")[,"lwr"],lty=2)
```




Kuva 16. Sovitetun suoran 95 % luottamusvälit

Komennossa ”predict()” voi käyttää lisämäärettä interval=”prediction”, jolloin mallia käytetään ennustuksessa. Lisämääre antaa ennustuksen 95 % luottamusvälit, joista voidaan piirtää kuvan 17 mukainen kuvaaja. Kuvasta nähdään, että T:n ollessa 10 r on arvojen 30 ja 55 välissä 95 % todennäköisyydellä. (Oksanen, 2003)

```
> matlines(T,predict(reg,interval="prediction")[,c("lwr","upr")],lty=3,col="black")
```




Kuva 17. Mallilla tehdyn ennusteen 95%-luottamusvälit

### 3.3.2 Usean muuttujan regressio

Regressioanalyysissä voidaan tutkia myös monen muuttujan malleja lisäämällä ”lm()”-komentoon useampia muuttujia. Seuraavassa esimerkissä analysoidaan kappaleessa 2.5 R:ään aiemmin ladattua aineistoa.

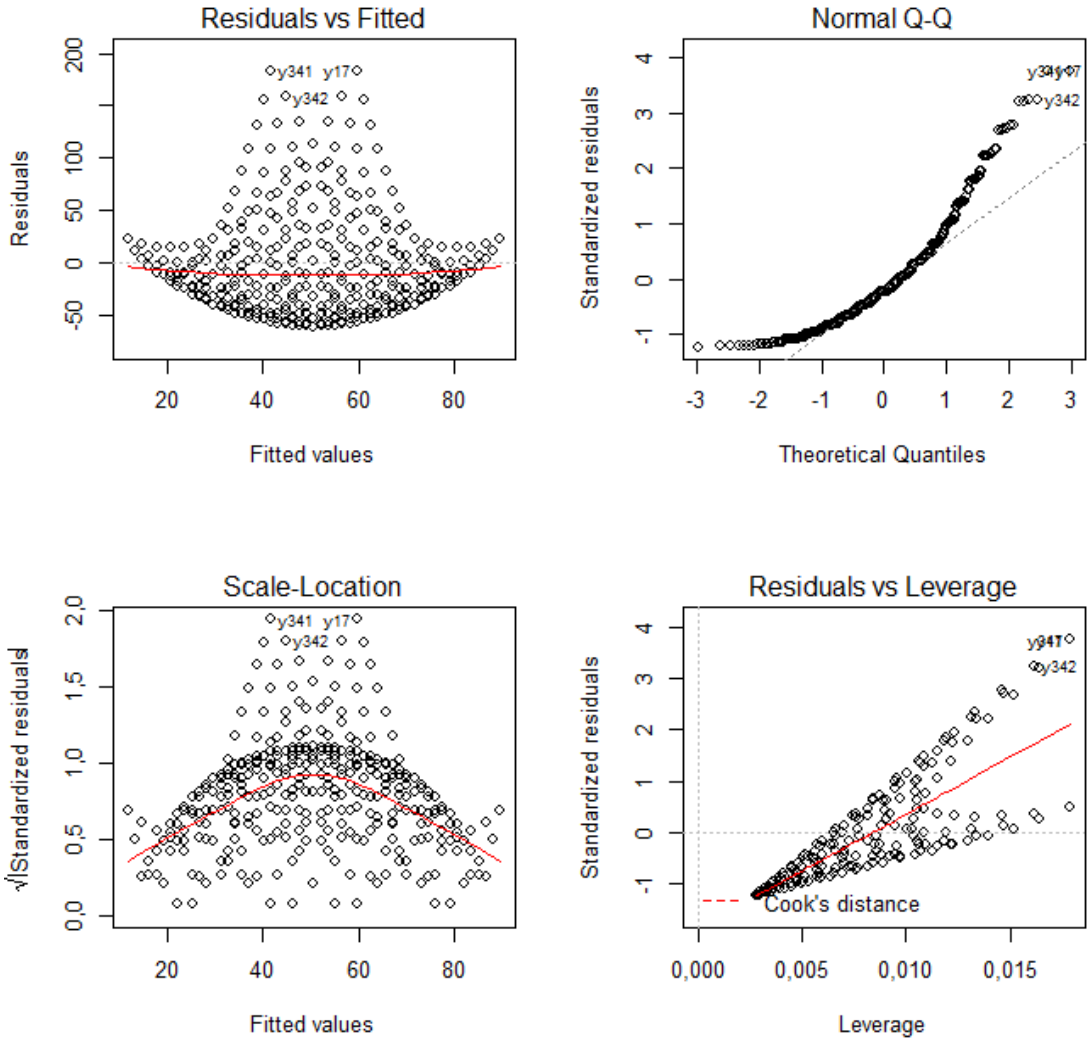
```
> data1=unlist(data[1])
> data2=unlist(data[2])
> data3=unlist(data[3])
> mulreg=lm(data3 ~ data1 + data2)
> summary(mulreg)

Call:
lm(formula = data3 ~ data1 + data2)
Residuals:
   Min      1Q  Median      3Q     Max
```

```
-60,67  -35,67   -11,67     18,33   183,33

Coefficients:
             Estimate     Std. Error    t value      Pr(>|t|)
(Intercept)  56,6667      2,8149        20,131       < 2e-16 ***
data1        3,0000        0,8608        3,485        0,000553 ***
data2        3,0000        0,5320        5,640        3,49e-08 ***
---
Signif. codes:  0 ‘***’ 0,001 ‘**’ 0,01 ‘*’ 0,05 ‘.’ 0,1 ‘ ’ 1

Residual standard error: 49,24 on 354 degrees of freedom
Multiple R-squared:  0,1104,    Adjusted R-squared:  0,1054
F-statistic: 21,98 on 2 and 354 DF,  p-value: 1,008e-09
```

Komennosta saadut tulokset ovat lähes samanlaiset kuin yksittäisen muuttujan analyysissä. Jokaisen muuttujan tulokset tulevat omalle rivilleen.

Regressioanalyysin tulokset voidaan piirtää kuvaajiin seuraavilla komennoilla. Tässä tapauksessa komento ”par()” kertoo, että piirretään 4 kuvaa neliöön.

```
> par(mfrow=c(2,2))
> plot(mulreg)
```




Kuva 18. Monimuuttujaregression tuloksen ominaisuuksien kuvaajat

Komento muodostaa neljä kuvaajaa analyysimallille, kuten kuvassa 18 on esitetty. Ensimmäisestä kuvaajasta nähdään, että residuaalit eivät täysin vastaa yksinkertaista lineaarista mallia. Yläoikealla olevassa kuvaajassa olisi suora, mikäli residuaalit olisivat normaalisti jakautuneita. Alavasemmalla residuaalit on standardoitu ja nähdään, että muodostettu lineaarinen malli ei ole sopiva. Viimeisestä kuvaajasta nähdään, onko aineistossa yksittäisiä pisteitä, joilla on suuri merkitys suoraan. (Oksanen, 2003)

# 4 JOHTOPÄÄTÖKSET JA YHTEENVETO

Työssä käsiteltiin ensimmäisenä R:ssä käytettävät peruskomennot, jotka antavat pohjan R:n jatkokäytölle. Tämän jälkeen nostettiin esille R:n graafiset ominaisuudet sekä tilastollisten tunnuslukujen käyttö, jotka mahdollistavat datan käsittelyn, analysoinnin ja esityksen. Myös monimutkaisempia analyysimalleja on käytettävissä R:n vakiomuodossa ja paketeilla lisäten, mutta työ rajoitettiin hieman yksinkertaisemmaksi. Työssä käydyt menetelmät antavatkin hyvän pohjan R:n käyttöön datan kanssa työskentelyssä.

Työ toimii aloittelevan R:n käyttäjän oppaana R:n käyttömahdollisuuksiin esittelemällä peruskäyttöön liittyviä komentoja kappaleessa R:n peruskäyttö. R:n käyttömahdollisuuksina esiteltiin data-analyysin työkaluja kappaleessa Kokeellisen datan käsittely. Työssä esitellyt asiat ovat vain pieni pintakosketus R:n mahdollistamiin menetelmiin.

# LÄHDELUETTELO

Maindonald, J. H., 2008. Using R for Data Analysis and Graphics: Introduction, Code and Commentary. [Online] 2008. [Viitattu: 24. 4 2018.] https://cran.r-project.org/doc/contrib/usingR.pdf.

Oksanen, Jari, 2003. R: Opas ekologeille. [Online] 2003. [Viitattu: 24. 4 2018.] http://cc.oulu.fi/~jarioksa/opetus/rekola/Rekola.pdf.

R Core Team, 2018. An Introduction to R. [Online] 2018. [Viitattu: 24. 4 2018.] https://cran.r-project.org/doc/manuals/r-release/R-intro.html.

RStudio, 2018. Why RStudio? [Online] 2018. [Viitattu: 24. 4 2018.] https://www.rstudio.com/about/.

The R Foundation, 2018. The R Journal. [Online] 2018. [Viitattu: 24. 4 2018.] https://journal.r-project.org/.

The R Foundation, 2017. What is R? [Online] 2017. [Viitattu: 24. 4 2018.] https://www.r-project.org/about.html.