Mitteparameetriline statistika
Testid kasutades R'i

```
# ------------------------------------------------------------------------
#          Märgitest, usaldusintervall mediaanile
# ------------------------------------------------------------------------

valim1=c(10.2, 12.3, 15.1, 9.8, 45.3)

# Kas mediaan võiks olla 30?

# Mitu vaatlust on suuremad kui 30 (mediaan H0 väitel)?

a=sum((valim1-30)>0)

# Märgitest: kui mediaan on 30, on 30 suurema vaatluse saamise tõenäosus 0,5:
binom.test(a, 5, p=0.5)

# Testi illustreeriv graafik
b=dbinom(0:5, 5, 0.5)
names(b)=0:5
b

barplot(b, ylab="Tõenäosus")

# Kontrollime olulisustõenäosuse arvutuse huvi pärast käsitsi üle:
b[1]+b[2]+   b[5]+b[6]


# Muidugi on R'is olemas ka funktsioon (lisamoodulis BSDA) märgitesti tegemiseks:
library(BSDA)
SIGN.test(valim1, md=30)

# Kas erinevuste mediaan on 0?
valim1=c(10.2, 12.3, 15.1, 9.8, 45.3)
valim2=c(14.2, 13.3, 16.2, 9.5, 45.6)

abi=valim2-valim1

a=sum(abi>0)
binom.test(a, length(abi), p=0.5)

# Lisamoodulist BSDA
SIGN.test(valim1, valim2)

set.seed(1)
Isikupara=rnorm(40)
valim1=Isikupara+rnorm(40)
valim2=Isikupara+rnorm(40)+1

SIGN.test(valim1, md=20)
SIGN.test(valim1, valim2)

SIGN.test(valim1-valim2)
```

```
# ------------------------------------------------------------------------
#                    Wilcoxoni astakmärgitest
# ------------------------------------------------------------------------

valim1=c(10.2, 12.3, 15.1, 9.8, 45.3, 200.1)
valim2=c(14.2, 13.3, 16.2, 9.5, 45.6, 200.2)

wilcox.test(valim1, valim2, paired=TRUE)
wilcox.test(valim1, valim2, paired=TRUE, exact=TRUE)
wilcox.test(valim1, valim2, paired=TRUE, exact=FALSE)


# alternatiiv:
a=valim1-valim2
rank(a)
stat=sum(rank(abs(a))[a>0])
stat

psignrank(stat, n=length(a))*2

wilcox.test(valim1, valim2, paired=TRUE, conf.int=T)
wilcox.test(valim1, valim2-0.65, paired=TRUE, conf.int=T)


# ------------------------------------------------------------------------
#                    Wilcoxoni astaksummatest
#          (Wilcoxon rank-sum test, Mann-Whitney test)
# ------------------------------------------------------------------------

set.seed(2)
valim1=rcauchy(100)
valim2=rcauchy(100)+1

wilcox.test(valim1, valim2)
t.test(valim1, valim2)

# Wilcoxoni (Mann-Whitney) statistiku jaotus H0 kehtides
    x <- -1:(4*6 + 1)
    fx <- dwilcox(x, 4, 6)
    Fx <- pwilcox(x, 4, 6)

# Jagame akna kahe graafiku vahel - üks suurem
    layout(rbind(1,2),width=1,heights=c(3,2))
    par(mar=c(2.5, 4, 4, 2))
    plot(x, fx,type='h', col="violet",
        main= "Wilcoxoni statistiku tõenäosusfunktsioon (n=6, m=4)", xlab="")
    plot(x, Fx,type="s", col="blue",
        main= "Wilcoxoni statistiku jaotusfunktsioon (n=6, m=4)", xlab="")
    abline(h=0:1, col="gray20",lty=2)
    layout(1) # Taas üks graafik akna kohta
```

```
# Näide korduvaid väärtuseid sisaldavate andmetega

valim1=c(10, 20, 31, 40, 49)
valim2=c(20, 30, 50, 60, 70, 100, 1100, 100000)

wilcox.test(valim1, valim2)
wilcox.test(valim1, valim2, correct=TRUE)
wilcox.test(valim1, valim2, correct=FALSE)

# Pane tähele: ei suudeta arvutada täpset p-väärtust!
# Korrektsiooni kasutav ja mittekasutav lähend
# annavad väga erinevaid tulemusi!

# Appi tuleb lisamoodul coin:
library(coin)

# Viime andmed sobivamale kujule:

Ytunnus=c(valim1, valim2)
Grupp=rep(1:2, c(length(valim1), length(valim2)))
data.frame(Ytunnus, Grupp)

# lisamoodulis coin pesitsev funktsioon wilcox_test oskab ka
kordustega andmestikus leida täpset
# p-väärtust
wilcox_test(Ytunnus~factor(Grupp), distribution="exact")
wilcox_test(Ytunnus~factor(Grupp), distribution="asymptotic")
wilcox_test(Ytunnus~factor(Grupp),
distribution=approximate(B=100000))

# Võrdluseks t-test:
t.test(valim1, valim2)
```

 # ----------
 # Mitteparameetrilised testid hajuvuste võrdlemiseks
 # ----------

Ansari-Bradley test hajuvuste võrdlemiseks

```
x1=rnorm(10, 0, sd=1)
x2=rexp(50, 0.1)-10
ansari.test(x1, x2)
```

Fligner-Killeeni test (igast grupist lahutatakse maha mediaan, siis jagatakse saadud vahede
absoluutväärtustele astakuid. Skooridena kasutatakse normaaljaotuse kvantiile:
a(i)= qnorm((1 + i/(n+1))/2)

```
Y=c(x1,x2)
Grupp=rep(1:2, c(length(x1), length(x2)))
fligner.test(Y~factor(Grupp))
```

Kolmogorov-Smirnovi test ----------

```
ks.test(x1,x2)
```

```
# ------------------------------------------------------------------
#                          Friedmani test
# ------------------------------------------------------------------

sort1=c(989,860,600,520)
sort2=c(670,123,345,480)
sort3=c(780,870,450,490)

saak=c(sort1, sort2, sort3)
sort=rep(1:3, each=4)
pold=rep(1:4, 3)

data.frame(saak, sort, pold)

friedman.test(saak~sort|pold)

# Lisamooduli coin abiga
friedman_test(saak~factor(sort)|factor(pold))
friedman_test(saak~factor(sort)|factor(pold),
              distribution=approximate(B=1000000))
```

 # Ülesanne

Ülesande legend: Proovisime neljal erineval viisil võita sõpru. Peale iga katset mõõdame, millised on tänaval vastu tulnud inimese ehk katsealuse tunded meie vastu.

Päriselus pole meil aega inimesi tüütama minna, sestap mängime situatsiooni läbi kasutades genereeritud andmeid. Andmeid genereerides kasutame t-jaotusest vigu (mäletate, see normaaljaotuse-moodi kübaraga jaotus?)

```
teguviis=rep(1:4, c(10, 20, 25, 28))
table(teguviis)
tulem=c(0, 1, 2.5, -1)[teguviis]+
            rt(length(teguviis), df=1)
by(tulem, teguviis, median)
```

Millise testi abil saaksime testida, kas mõni teguviis võiks viia teistest parema tulemuseni?

Õige vastus: Antud juhul võiks kasutada Kruskal-Wallise testi!

```
kruskal.test(tulem~factor(teguviis))

# Võrdluseks – sama andmestiku analüüs dispersioon-
#    analüüsi (ANOVA) abil

DispAnalyysiMudel=lm(tulem~factor(teguviis))
anova(DispAnalyysiMudel)
```