R - Datavisualisatie

J.J. van Nijnatten

Contents

contact: J.J.vanNijnatten@uva.nl

broncode: https://github.com/jonasvannijnatten/R_Data_Visualization

Het doel van datavisualisatie

Wat wil je weergeven?

Hoe kies je de juiste manier van weergeven? (exp. design)

Essentiele onderdelen van datavisualisatie

gemiddelde

spreiding

legenda

titel

Introductie GGplot2 package

Installatie

Het package downloaden & installeren:

```r
install.packages(pkgs="ggplot2", repos = "https://www.freestatistics.org/cran/")
install.packages(pkgs="Hmisc",   repos = "https://www.freestatistics.org/cran/")
```

Het package library activeren:

```r
library(package="Hmisc")
library(package="ggplot2")
```

Opbouw van figuren

Voorbeelden

Data-inspectie

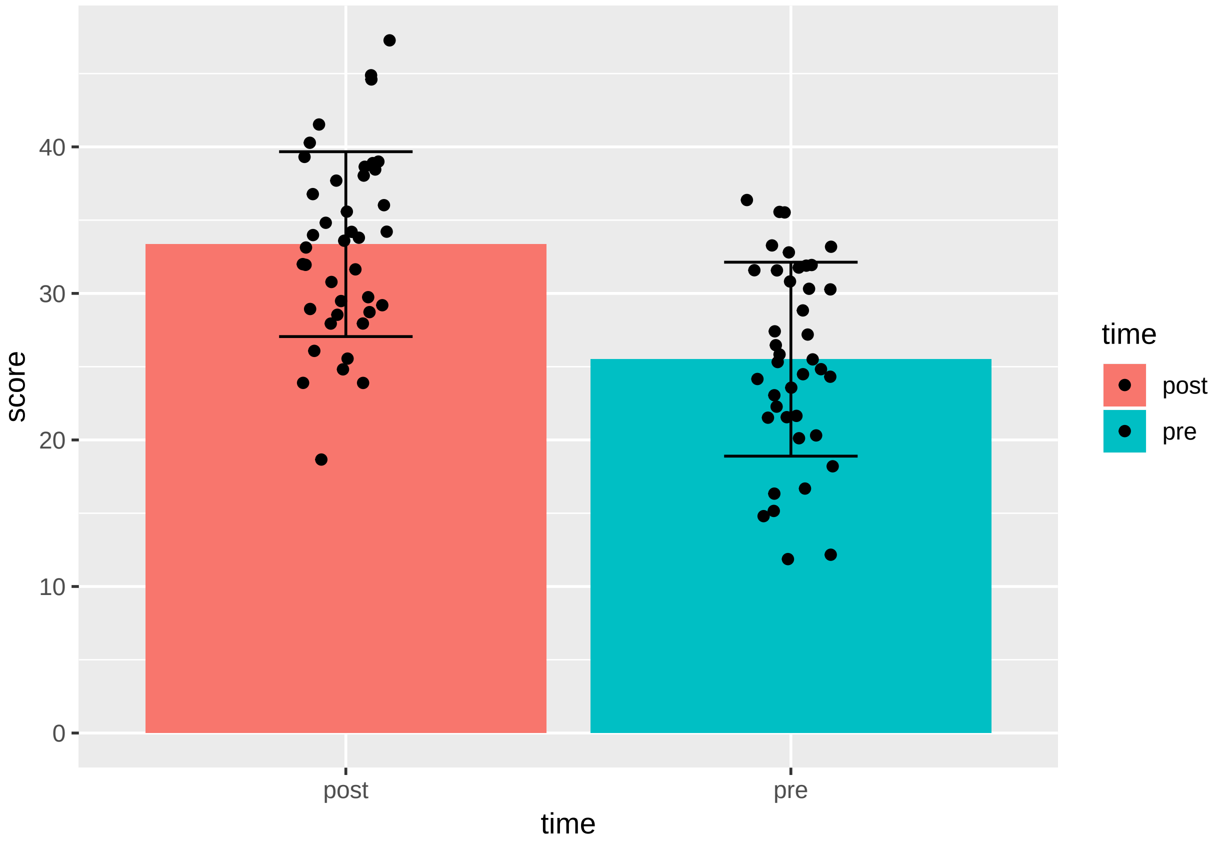
Normaliteit

T-test

```
# generate data
group1 = rnorm(n = 40, mean = 25, sd = 6.5)
group2 = rnorm(n = 40, mean = 35, sd = 6.5)
data.wide = data.frame(group1, group2)
```

```
# reshape data
data.long = reshape(data = data.wide,
                    direction = "long"
                    , varying = c("group1", "group2")
                    , v.names = "score"
                    , times   = c('pre', 'post')
                      )
```

```
# plot means and standard deviations
ggplot(data.long, aes(x=time, y=score, fill=time) ) +
  geom_bar     ( stat = "summary", fun.y    = "mean" ) +
  geom_errorbar( stat = "summary", fun.data = "mean_sdl", fun.args = 1, width = 0.3 ) +
  geom_point   ( position=position_jitter(width=.1) )
```



To plot standard errors instead of standard deviations replace “mean_sdl” with “mean_se”, and it is common use to plot 2 (or 1.96) times the standard error to get an 95% confidence interval, so replace “fun.arg = 1” with “fun.arg = 2”.

```
  geom_errorbar( stat = "summary", fun.data = "mean_se", fun.args = 2, width = 0.3 )
```

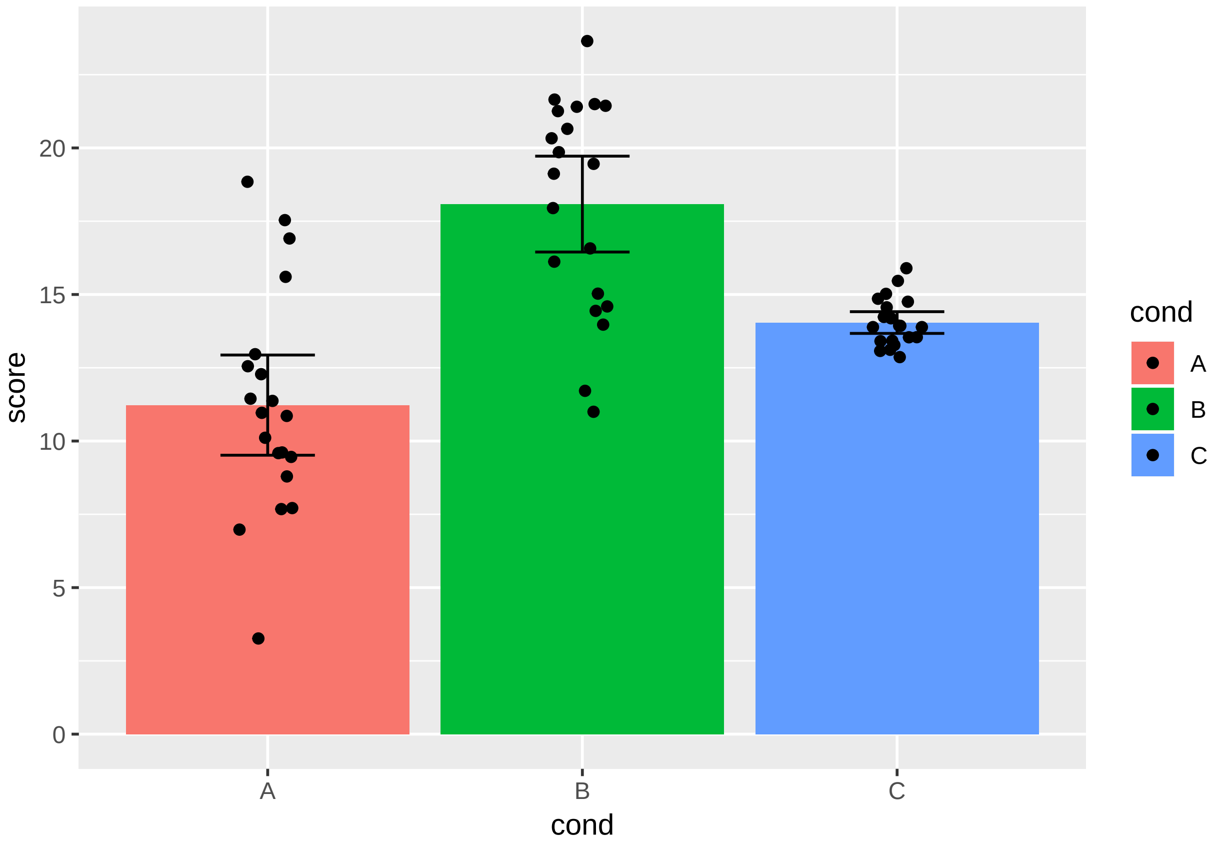
Correlatie

Regressie

One-way independent samples ANOVA

```
set.seed(05)   # set seed
nrofconds = 3  # set number of conditions
nrofsubs  = 20 # set number of subjects
subj = as.factor(1:(nrofsubs*nrofconds))      # create array with subject IDs
cond = as.factor(rep(LETTERS[1:nrofconds],each=nrofsubs))   # create array with condition values
score = as.vector( replicate(
          nrofconds , rnorm(n = nrofsubs, mean = sample(8,1)+10 , sd = sample(5,1) )
        ) )                                     # create array with measurement values
data.long = data.frame(subj, cond, score);      # combine arrays into a data.frame
rm(list=setdiff(ls(), c("data.long", "nrofsubs","nrofconds"))) # delete arrays
```

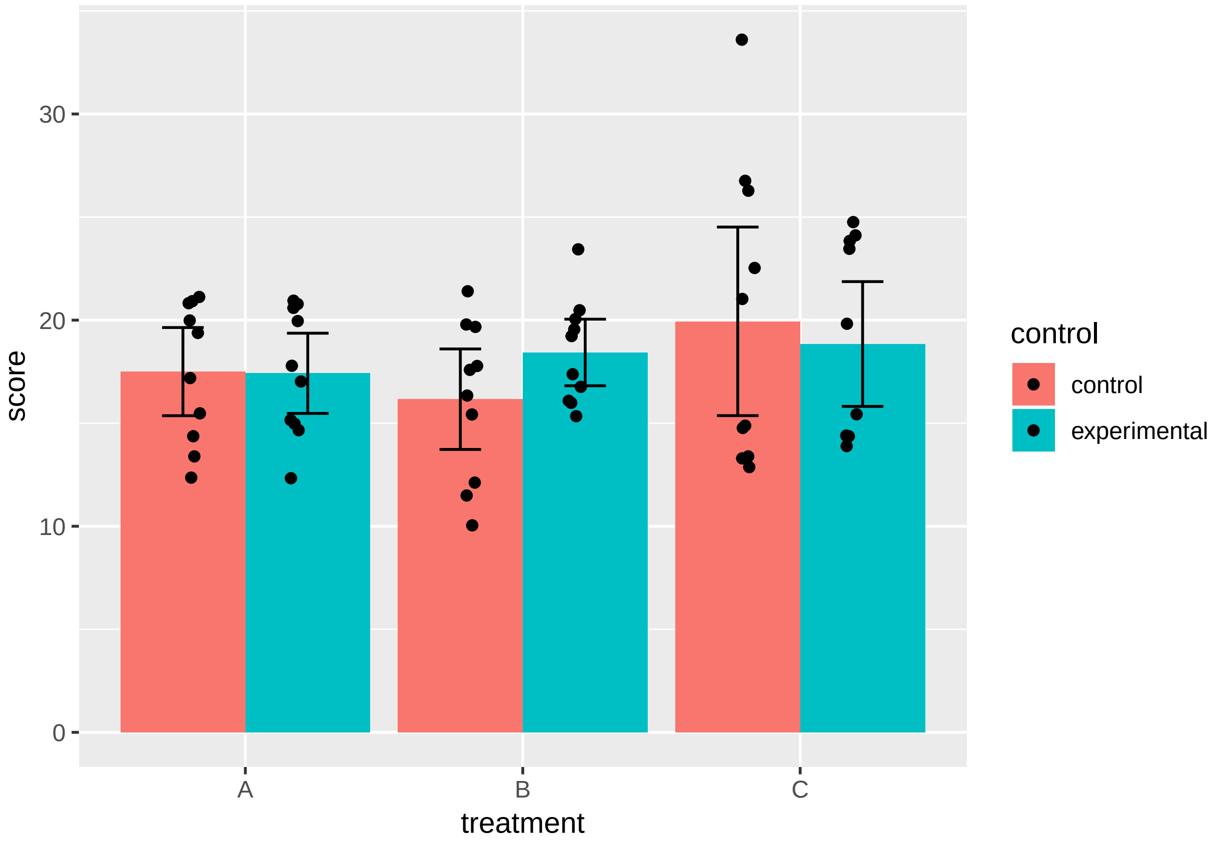
```
ggplot(data.long, aes(x=cond, y=score, fill=cond) ) +
  geom_bar     ( stat = "summary", fun.y    = "mean" ) +
  geom_errorbar( stat = "summary", fun.data = "mean_se", fun.args = 2, width = 0.3 ) +
  geom_point   ( position=position_jitter(width = .1) )
```



Factorial independent samples ANOVA

```
set.seed(01)   # set seed
nrofcondsf1 = 3  # set number of conditions for factor 1
nrofcondsf2 = 2  # set number of conditions for factor 2
nrofsubs    = nrofcondsf1*nrofcondsf2*10 # set number of subjects per condition
subj = as.factor(1:(nrofsubs))      # create array with subject IDs
treatment = as.factor(rep(LETTERS[1:nrofcondsf1],each=nrofsubs/nrofcondsf1))   # create array witht tre
control   = as.factor(rep(c("control","experimental"),times=nrofsubs/nrofcondsf2))   # create array wit
score = as.vector( replicate(nrofcondsf1, replicate (
          nrofcondsf2 , rnorm(n = (nrofsubs/(nrofcondsf1*nrofcondsf2)), mean = sample(14,1)+10 , sd = sa
        ) ) )                                    # create array with measurement values
data.long = data.frame(subj, score, treatment, control);      # combine arrays into a data.frame
rm(list=setdiff(ls(), c("data.long", "nrofsubs","nrofconds"))) # delete arrays
```

```
ggplot(data.long, aes(x=treatment, y=score, fill=control) ) +
  geom_bar     ( stat = "summary", fun.y    = "mean" , position = "dodge") +
  geom_errorbar( stat = "summary", fun.data = "mean_se", fun.args = 2, width = 0.3,
                 position = position_dodge(width=.9) ) +
  geom_point  ( position = position_jitterdodge(jitter.width = .1) )
```

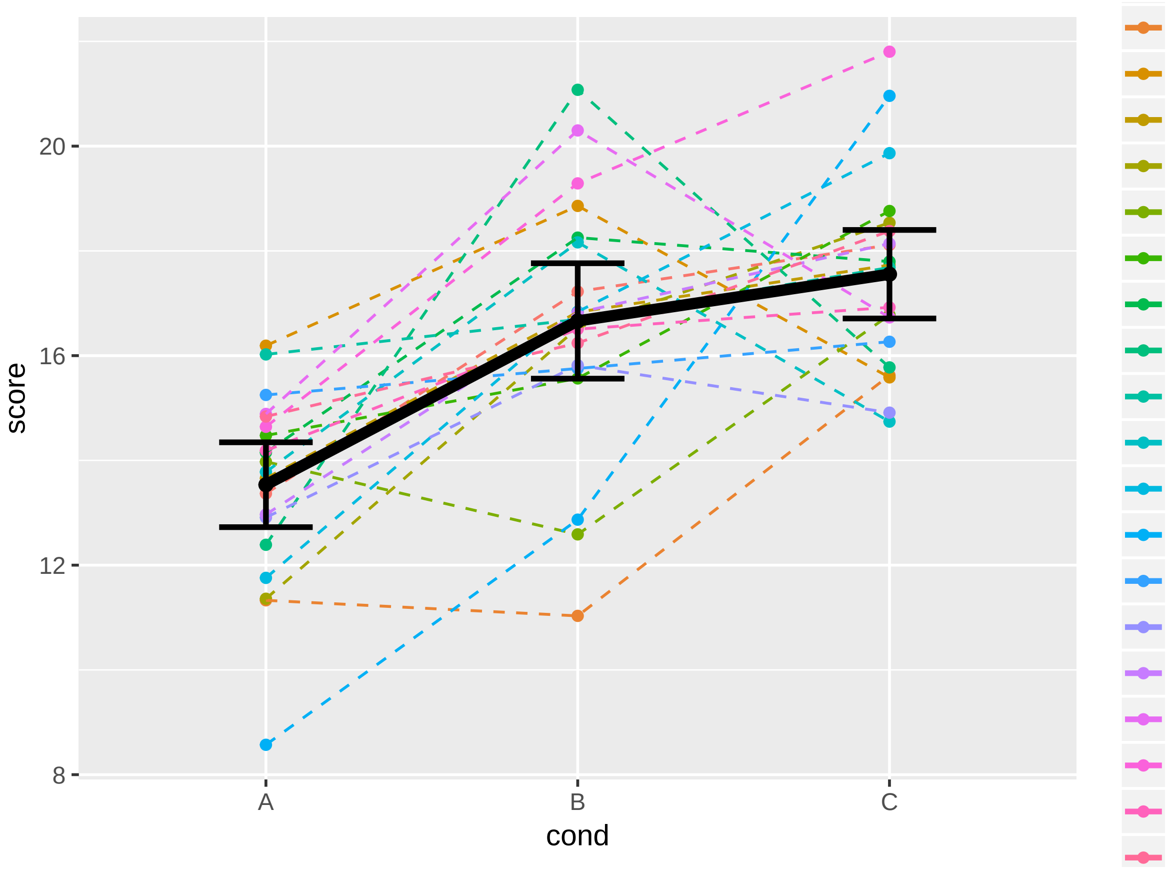


One-way repeated measures ANOVA

Generate dataset

```
set.seed(01)   # set seed
nrofsubs  = 20 # set number of subjects
nrofconds = 3  # set number of conditions
subj = as.factor(rep(1:nrofsubs,nrofconds))      # create array with subject IDs
cond = as.factor(rep(LETTERS[1:nrofconds],each=nrofsubs))   # create array with condition values
score = as.vector( replicate(
          nrofconds , rnorm(n = nrofsubs, mean = sample(8,1)+10 , sd = sample(5,1) )
        ) )                                      # create array with measurement values
data.long = data.frame(subj, cond, score);       # combine arrays into a data.frame
rm(list=setdiff(ls(), c("data.long", "nrofsubs","nrofconds"))) # delete arrays
```

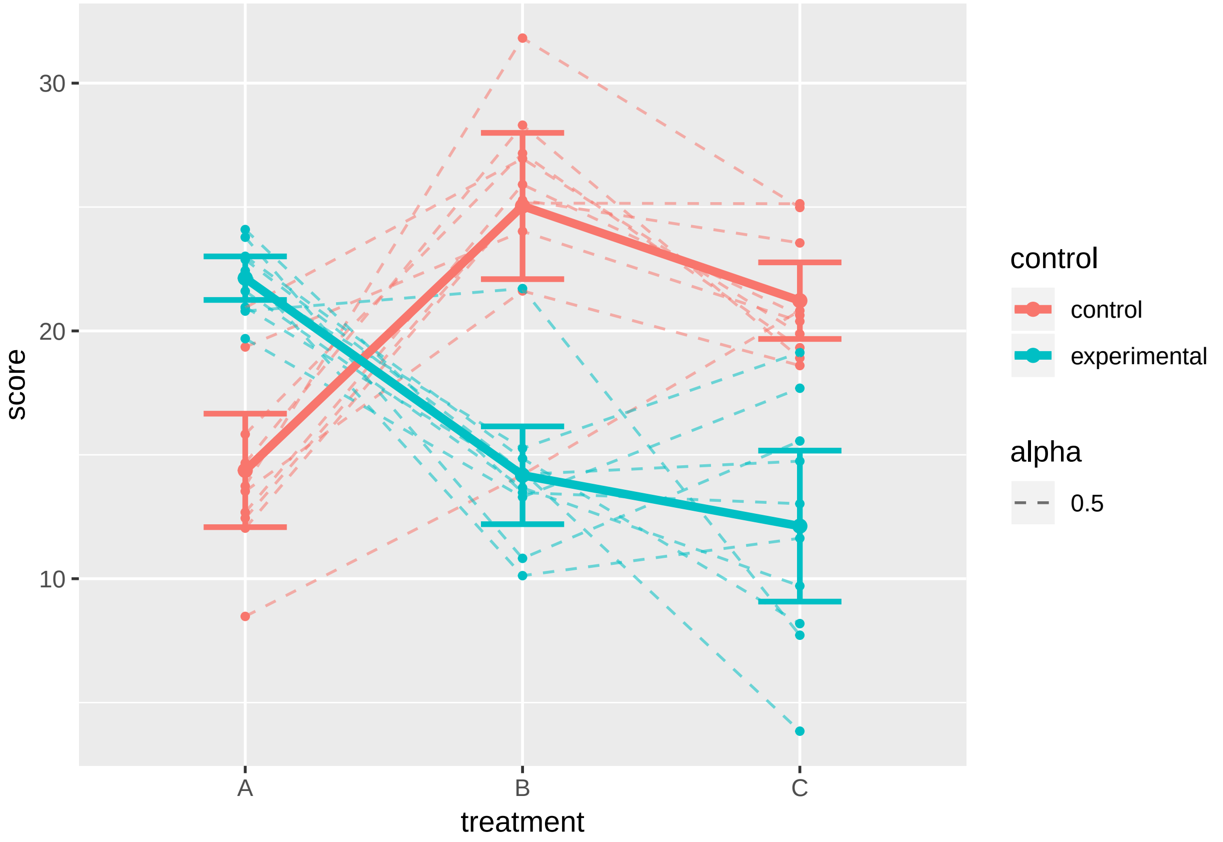
```
ggplot(data.long, aes(x=cond, y=score, group=1, colour=subj)) +
  geom_point    () +
  geom_line     ( linetype= "dashed", aes(group=subj) ) +
  geom_line     ( stat = "summary", fun.y = "mean",       size=2, colour = "black", linetype= "solid" ) +
  geom_point    ( stat = "summary", fun.y = "mean",       size=2, colour = "black" ) +
  geom_errorbar( stat = "summary", fun.data = "mean_se", size=1, fun.args = 2, width = 0.3 )
```



Factorial repeated measures ANOVA

```r
set.seed(02)  # set seed
nrofcondsf1 = 3  # set number of conditions for factor 1
nrofcondsf2 = 2  # set number of conditions for factor 2
nrofsubs    = 10 # set number of subjects
subj = as.factor(rep(1:(nrofsubs),times=nrofcondsf1*nrofcondsf2))    # create array with subject IDs
treatment = as.factor(rep(LETTERS[1:nrofcondsf1],each=nrofsubs*nrofcondsf2))   # create array witht tre
control   = as.factor(rep(rep(c("control","experimental"),each=nrofsubs),times=nrofcondsf1))   # create
score = as.vector( replicate(nrofcondsf1, replicate(nrofcondsf2,    # create array with measurement val
                             rnorm(n = (nrofsubs), mean = sample(14,1)+10 , sd = sample(5,1)
                                   ) ) ) )
data.long = data.frame(subj, score, treatment, control);      # combine arrays into a data.frame
rm(list=setdiff(ls(), c("data.long", "nrofsubs","nrofconds"))) # delete arrays
```

```r
ggplot(data.long, aes(x=treatment, y=score, group=control, colour=control)) +
  geom_point   (size=1)  +
  geom_line    (linetype="dashed" ,aes(group=interaction(subj,control), alpha=.5)) +
  geom_line    ( stat = "summary", fun.y    = "mean",    size=1.5 )  +
  geom_point   ( stat = "summary", fun.y    = "mean",    size=2 ) +
  geom_errorbar( stat = "summary", fun.data = "mean_se", size=1, fun.args = 2, width = 0.3)
```


score
A
B
C
treatment
control
control
experimental
alpha