

Experimenteel Design I: replicatie en power

Lieven Clement

statOmics, Ghent University (<https://statomics.github.io>)

Contents

Concepten	2
Replicatie	2
Voorbeeld	2
Bronnen van variabiliteit	3
Op welk niveau moeten we repliceren?	3
Power, steekproefgrootte en andere design aspecten.	4
Intermezzo lineare regression in matrix vorm	4
kleinste kwadraten techniek/Least Squares (LS)	5
Variantie schatter?	5
Muis voorbeeld	6
Opmerkingen	14
Meer efficiënte code met matrices	15
Meer efficiënte code gebaseerd op de analytische oplossing voor een two group comparison	17

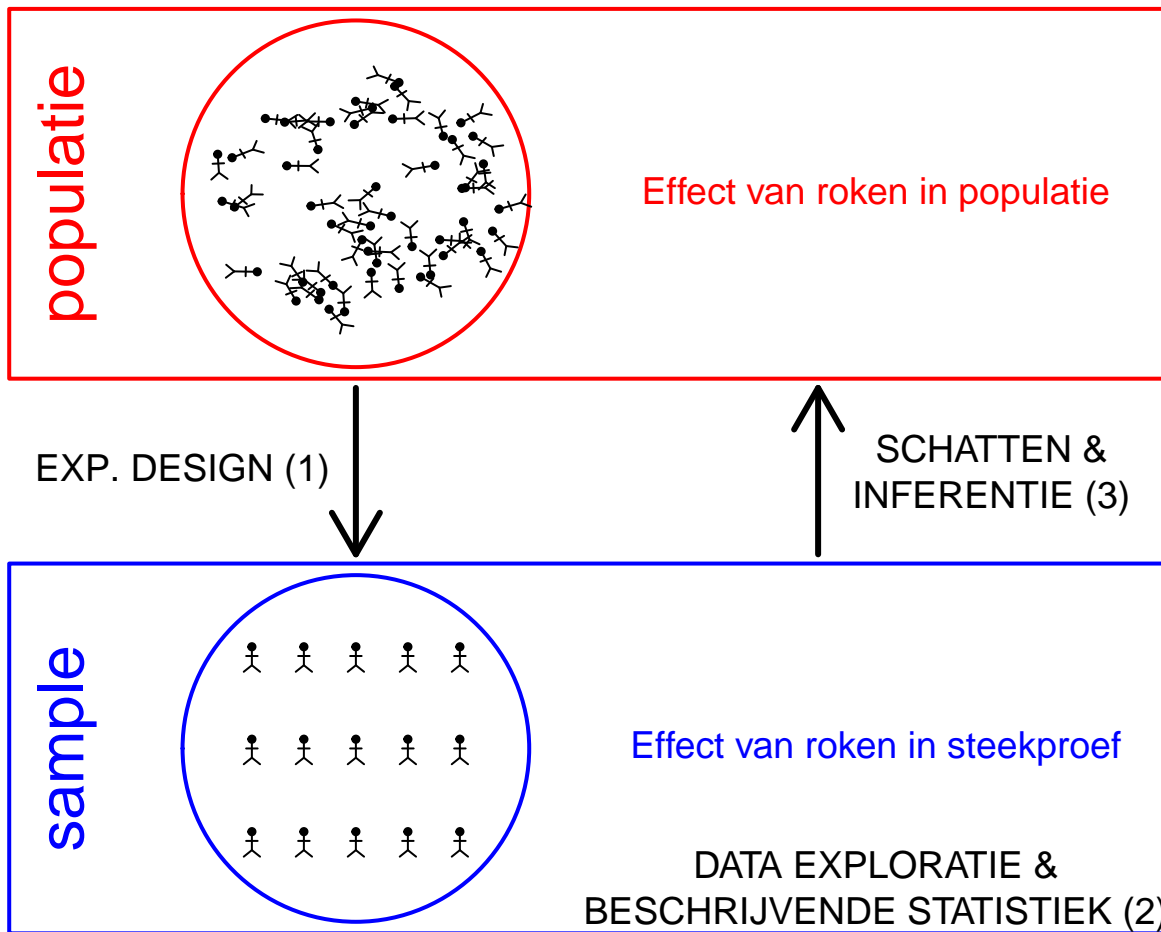
```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5    v purrr  0.3.4
## v tibble  3.1.2    v dplyr  1.0.7
## v tidyr   1.1.3    v stringr 1.4.0
## v readr   1.4.0    v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```



Concepten

- Experimentele eenheden zijn representatief voor populatie: Randomisatie
- Replicatie: technisch vs biologisch, steekproefgrootte - power
- Bronnen van variabiliteit: technisch, biologisch, binnen en tussen subjecten.

Replicatie

Paper over replicatie in Nature Methods (2 pagina's) [PDF]

Voorbeeld

In een RNA-seq experiment willen de onderzoekers het effect nagaan van een medicijn op de gen expressie.
Mogelijke onderzoeksvraag

- Is er een verschil in gemiddelde gen expressie tussen behandelde en niet behandelde samples

Bronnen van variabiliteit

TABLE 1 NATURE METHODS | VOL.11 NO.9 | SEPTEMBER 2014 | 879 - 880

	Replicate type	Replicate category ^a
Animal study subjects	Colonies	B
	Strains	B
	Cohoused groups	B
	Gender	B
	Individuals	B
Sample preparation	Organs from sacrificed animals	B
	Methods for dissociating cells from tissue	T
	Dissociation runs from given tissue sample	T
	Individual cells	B
Sequencing	RNA-seq library construction	T
	Runs from the library of a given cell	T
	Reads from different transcript molecules	V ^b
	Reads with unique molecular identifier (UMI) from a given transcript molecule	T

(a) Replicates are categorized as biological (B), technical (T) or of variable type (V).

(b) Sequence reads serve diverse purposes depending on the application and how reads are used in analysis.

Op welk niveau moeten we repliceren?

- $\text{var}(X) = \sigma_A^2 + \sigma_C^2 + \sigma_M^2 = \sigma_{TOT}^2$
- $\text{var}(\bar{X}) = \frac{\sigma_A^2}{n_A} + \frac{\sigma_C^2}{n_A n_C} + \frac{\sigma_M^2}{n_A n_C n_M}$

figure Figure 1 NATURE METHODS | VOL.11 NO.9 | SEPTEMBER 2014 | 879 - 880

(a) Three levels of replication (two biological, one technical) with animal, cell and measurement replicates normally distributed with a mean across animals of 10 and ratio of variances 1:2:0.5. Solid green (biological) and blue (technical) dots show how a measurement of the expression ($X = 12$) samples from all three sources of variation. Distribution s.d. is shown as horizontal lines.

(b) Expression variance, $\text{Var}(X)$, and variance of expression mean, $\text{Var}(\bar{X})$, computed across 10,000 simulations of $n_A n_C n_M = 48$ measurements for unique combinations of the number of animals ($n_A = 1$ to 48), cells per animal ($n_C = 1$ to 48) and technical replicate measurements per cell ($n_M = 1$ and 3). The ratio of $\text{Var}(X)$ and $\text{Var}(\bar{X})$ is the effective sample size, n , which corresponds to the equivalent number of statistically independent measurements. Horizontal dashed lines correspond to biological and total variation. Error bars on $\text{Var}(X)$ show s.d. from the 10,000 simulated samples ($n_M = 1$).

Waar ligt interesse?

- Karakteriseren van bronnen van variabiliteit in experiment ==> technische en biologische repeats zijn nodig
- Voornamelijke interesse in schatten van effect van behandeling ==> focus op biologische repeats
- Extra biologische repeats lumpen alle bronnen van variabiliteit: biologische + technische variabiliteit.
- Goed experimenteel design vereist na denken over replicatie
 1. Identificeer onderzoeksvragen die men wenst te beantwoorden.
 2. Breng bronnen van variabiliteit in kaart in elke stap van het experiment
 3. Let op voor pseudoreplicatie, beoog zoveel mogelijk onafhankelijke repeats.

Power, steekproefgrootte en andere design aspecten.

Reading materials: Nature Methods (2013), 10(12), 1139–1140

Intermezzo lineare regression in matrix vorm

- Lineaire regressie is een belangrijke methode in statistische data analyse.

Scalaire vorm

- Stel dat men voor elke experimentele eenheid p predictoren meet $\mathbf{x} = (x_1, \dots, x_p)$ en
- continue respons Y
- het linear regressiemodel kan dan worden geschreven als:

$$Y = f(\mathbf{x}) + \epsilon = \beta_0 + \sum_{j=1}^p x_j \beta_j + \epsilon$$

met i.i.d. $\epsilon \sim N(0, \sigma^2)$

Vector/Matrix vorm

- n observatie $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_n, y_n)$ with $\mathbf{x}_1^T = [1x_1 \dots x_p]$
- Regressie in matrix notatie

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\text{met } \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix} \text{ or } \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix} \text{ and } \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

kleinste kwadraten techniek/Least Squares (LS)

- Minimaliseer de residuele kwadratensom

$$\begin{aligned}RSS(\beta) &= \sum_{i=1}^n e_i^2 \\ &= \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2\end{aligned}$$

- of in matrix notatie

$$RSS(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)$$

$$\rightarrow \hat{\beta} = \operatorname{argmin}_{\beta} RSS(\beta)$$

Minimaliseer RSS

$$\begin{aligned}\frac{\partial RSS}{\partial \beta} &= \mathbf{0} \\ \frac{(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)}{\partial \beta} &= \mathbf{0} \\ -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta) &= \mathbf{0} \\ \mathbf{X}^T\mathbf{X}\beta &= \mathbf{X}^T\mathbf{Y} \\ \hat{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\end{aligned}$$

Variantie schatter?

$$\begin{aligned}\hat{\Sigma}_{\beta} &= \operatorname{var} [(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}] \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \operatorname{var} [\mathbf{Y}] \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I}\sigma^2)\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{I} \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2\end{aligned}$$

De onzekerheid op model parameters hangt dus af van de residuele variabiliteit en het de proefopzet!

- Hoe groter $\mathbf{X}^T\mathbf{X}$ hoe meer informatie het experiment zal hebben over de model parameters en hoe kleiner hun variantie en standaard errors!

- Factoriële designs?
- Designs with continue predictoren?

De effectgrootte van interesse is typisch een lineaire combinatie van de model parameters,

$$l_0 \times \beta_0 + l_1 \times \beta_1 + \dots + l_{p-1} \times \beta_{p-1} = \mathbf{L}^T \boldsymbol{\beta}$$

De nulhypothese van onze test kan dan worden geschreven als

$$H_0 : \mathbf{L}^T \boldsymbol{\beta} = 0$$

vs de alternatieve hypothese

$$H_0 : \mathbf{L}^T \boldsymbol{\beta} \neq 0$$

En het bewijs in het experiment tegen H_0 kan worden gekwantificeerd met de t-test statistiek:

$$t = \frac{\mathbf{L}^T \hat{\boldsymbol{\beta}} - 0}{\text{se}_{\mathbf{L}^T \hat{\boldsymbol{\beta}}}}$$

die een t-distributie volgt met n-p vrijheidsgraden onder de nul hypothese als alle aannames van het model geldig zijn.

De kracht van de toets is dan

$$P(p < 0.05 | H_1)$$

en hangt af van

- de werkelijke effectgrootte in de populatie $\mathbf{L}^T \boldsymbol{\beta}$.
- Het aantal observaties: SE en df van t-test.
- Keuze van de designpunten
- Keuze van significantie-niveau α .

We kunnen dit evalueren aan de hand van simulaties.

Muis voorbeeld

- In 2021 publiceerden Choa et al. dat cytokine Thymic stromal lymphopoietin (TSLP) vetverlies induceert bij muizen die een vet dieet krijgen door de secretie van talg. [html] [PDF]
- Stel dat je een gelijkaardig experiment op wil zetten om na te gaan of cytokine interleukin 25 (IL) ook een effect heeft op het gewicht van muizen.
- Je plant een studie met een controle groep muizen die een hoog vet dieet krijgen (HFD) en een behandelingsgroep die een HFD dieet met IL krijgt.
- Wat is de vereiste steekproefgrootte om een bepaald effect op te pikken van de behandeling?

Hoe zal je de data van het experiment analyseren?

- H_0 : Er is geen verschil in het gewicht van muizen die een dieet met IL krijgen en muizen die het controle dieet krijgen.
- H_1 : Het gewicht is gemiddelde verschillend tussen muizen die het IL dieet krijgen en muizen die het controle dieet krijgen.
- Two sample t-test of een t-test op de helling van het lineaire model met 1 dummy variabele

$$Y_i = \beta_0 + \beta_1 X_{iIL} + \epsilon_i$$

$$\text{met } X_{iIL} = \begin{cases} X_{iIL} = 0 & \text{HFD} \\ X_{iIL} = 1 & \text{HFD} + \text{IL} \end{cases}$$

- Geschatte effectgrootte?

$$\hat{\delta} = \bar{X}_{IL} - \bar{X}_c = \hat{\beta}_1$$

- Test statistiek

$$T = \frac{\bar{X}_{IL} - \bar{X}_c}{SD_{\text{pooled}} \times \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = \frac{\hat{\beta}_1}{SE_{\hat{\beta}_1}}$$

- $\hat{\beta}_1$ is een onvertekende schatter van het werkelijk gewichtsverschil ($\delta = \mu_{IL} - \mu_c = \beta_1$) dat zou voorkomen in de populatie van muizen gevoed met HFD vs muizen gevoed met HFD + IL.

Power?

$$P[p < \alpha | \beta_1 \neq 0]$$

De kracht (power) zal dus afhangen van

- Het werkelijke gemiddelde gewichtsverschil β_1 in de populatie.
- De variabiliteit van de gewichtsmetingen
- het significantie-niveau α
- steekproefgrootte n_{IL} en n_c in beide groepen

We kunnen de power schatten als

1. voldaan is aan de aannemens van het model: gewicht in beide groepen is normaal verdeeld met gelijke variantie

en als we de

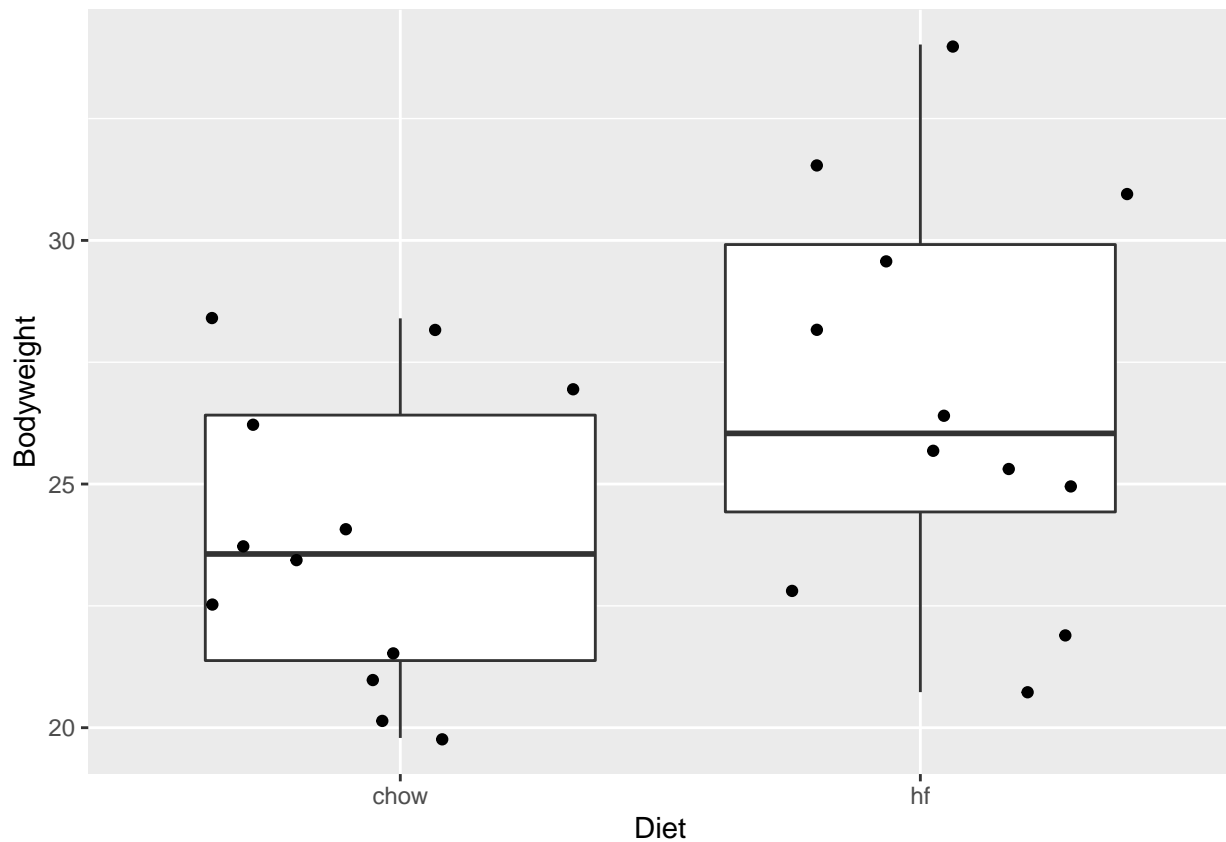
2. standaard deviatie kennen van gewichtsmetingen rond hun groepsgemiddelde voor muizen die een HFD dieet krijgen
3. de echte effectgrootte in de populatie
4. n_1 en n_2

Gebruik data uit vorig experiment om inzicht te krijgen in de gewichtsdata van muizen

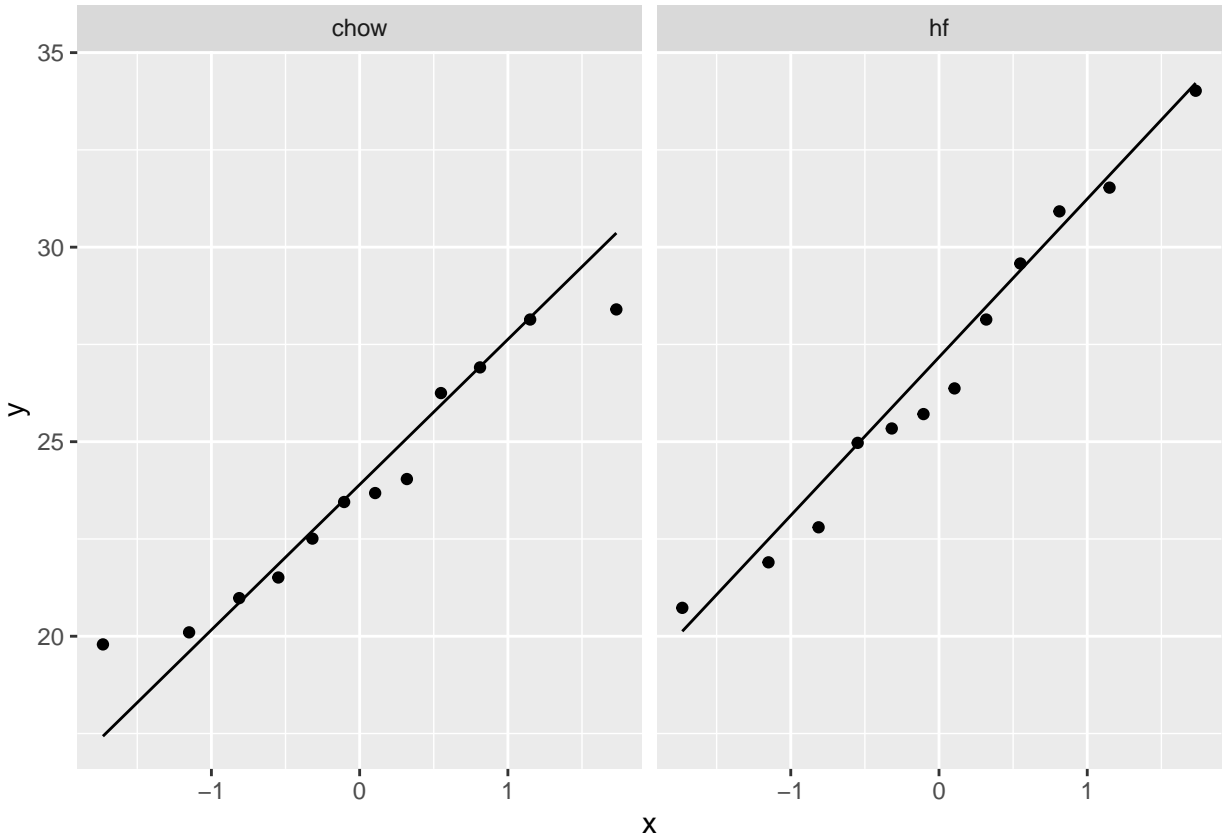
Stel dat we toegang hebben tot data van een vorig experiment (hier dat van Karen Svenson en Dan Gatti, gefinancierd door P50 GM070683 en beschikbaar op PH525x)

```
mice <- read.csv("https://raw.githubusercontent.com/genomicsclass/dagdata/master/inst/extdata/femaleMiceWeights.csv")

mice %>%
  ggplot(aes(x=Diet,y=Bodyweight)) +
  geom_boxplot(outlier.shape=FALSE) +
  geom_jitter()
```



```
mice %>%
  ggplot(aes(sample=Bodyweight)) +
  geom_qq() +
  geom_qq_line() +
  facet_wrap(~Diet)
```

```
mice <- mice %>% mutate(Diet = as.factor(Diet))
miceSum <- mice %>%
  group_by(Diet) %>%
  summarize_at("Bodyweight", list(mean=~mean(.,na.rm=TRUE),
                                   sd=~sd(.,na.rm=TRUE),
                                   n=function(x) x%>%is.na%>%`!`%>%sum)) %>%
  mutate(se=sd/sqrt(n))
miceSum
```

```
## # A tibble: 2 x 5
##   Diet mean sd n se
##   <fct> <dbl> <dbl> <int> <dbl>
## 1 chow 23.8 3.02 12 0.873
## 2 hf 26.8 4.10 12 1.18
```

In het experiment werden twee diëten gebruikt:

- gewoon diet o.b.v. granen (Chow)
- Heel vet dieet (high fat: hf)

We kunnen de data van de hf muizen gebruiken als input voor onze power analyse.

- De data van hf muizen lijken normaal verdeeld te zijn.

- Het gemiddelde gewicht van een hf muis is 26.8g
- De standaard deviatie (SD) van het gewicht van hf muizen is 4.1g

Effectgrootte?

- De alternative hypothese is complex.
- Omvat alle mogelijke effecten!
- Om een power analyse te doen kiezen we daarom een minimale effectgrootte die we wensen op te pikken in het nieuwe experiment.
- Veronderstel dat we een gewichtsverschil willen oppikken van minimaal 10%.

```
delta <- - round(miceSum$mean[2] *.1,1)
delta
```

```
## [1] -2.7
```

Merk op dat het gemiddeld gewicht dan dicht komt bij dat van muizen in het pilootexperiment die werden gevoed met het normale dieet.

We kunnen nu een simulatie studie opzetten om de power te berekenen voor een experiment met 8 muizen in elke groep.

```
set.seed(1423)
n1 <- n2 <- 8
b0 <- round(miceSum$mean[2],1)
b1 <- - delta
sd <- round(miceSum$sd[2],1)
alpha <- 0.05

x <- rep(0:1,c(n1,n2))
y <- b0 + b1 * x + rnorm(n1+n2,0, sd = sd)
fit <- lm(y~x)
bhat <- fit$coef
stat <- summary(fit)
summary(fit)$coef[2,]
```

Eén simulatie

```
## Estimate Std. Error t value Pr(>|t|)
## 2.5972480 1.7919658 1.4493848 0.1692595
```

Voor het gesimuleerde experiment konden we het effect van de behandeling niet oppikken bij het $\alpha=0.05$ ($p = 0.17$) significantie-niveau!

- We moeten het experiment herhalen om de power te schatten.
- We zullen de simulatie daarom coderen in een functie die we kunnen hergebruiken.

```
library(multcomp)
```

Simulatie van herhaalde experimenten (repeated experiments)

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##   select
```

```
##
```

```
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##   geyser
```

```
n1 <- n2 <- 8
```

```
b0 <- round(miceSum$mean[2],1)
```

```
b1 <- - delta
```

```
sd <- round(miceSum$sd[2],1)
```

```
predictorData <- data.frame(Diet = rep(c("c","hf"),c(n1,n2)) %>% as.factor)
```

```
alpha <- 0.05
```

```
simLm <- function(form, data, betas, sd, contrasts, simIndex = NA)
```

```
{
```

```
  dataSim <- data
```

```
  X <- model.matrix(form,dataSim)
```

```
  dataSim$ySim <- X%*%betas + rnorm(nrow(dataSim),0,sd)
```

```
  form <- formula(paste("ySim ~",form[[2]]))
```

```
  fitSim <- lm(form,dataSim)
```

```
  mcp <- glht(fitSim,linfct = contrasts)
```

```
  return(summary(mcp)$test[c("coefficients","sigma","tstat","pvalues")]%>% unlist)
```

```
}
```

```
simLm(
```

```
  form = ~Diet,
```

```

data = predictorData,
betas = c(b0,b1),
sd = sd,
contrasts = "Diethf = 0")

```

```

## coefficients.Diethf      sigma.Diethf      tstat.Diethf      pvalues
##           1.9444623           2.0931302           0.9289734           0.3686439

```

We hebben nu een generiek functie waarmee we data kunnen simuleren die normaal verdeeld zijn voor elk mogelijk design die we met het lineair model kunnen analyseren.

De functie heeft volgende argumenten:

- **form**: One sided formula including the structure of the predictors in the model
- **data**: Data frame with the predictor values for the design
- **betas**: A vector with values for all mean model parameters
- **sd**: The standard deviation of the error
- **contrasts**: a scalar or a vector with the null hypotheses that we would like to assess.
- **simIndex**: an arbitrary argument that is not used by the function but that will allow it to be used in an sapply loop that runs from 1 up to the number of simulations.

Simuleer nSim = 1000 herhaalde experimenten:

```

set.seed(1425)
nSim <- 1000

simResults <- t(sapply(1:nSim,simLm,form = ~Diet,data = predictorData,betas = c(b0,b1),sd = sd,contrasts = "Diethf = 0"))
power <- mean(simResults[,4] < alpha)
power

```

```
## [1] 0.221
```

We hebben een power van 22.1% om het behandelingseffect op te pikken met 8 biologische herhalingen in elke behandelingsgroep.

Merk op dat de power van een experiment ook het hoogste is als het aantal observaties gebalanceerd is, m.a.w. $n_1 = n_2$.

Power voor verschillende steekproefgroottes We kunnen de power nu berekenen voor meerdere steekproefgroottes.

```

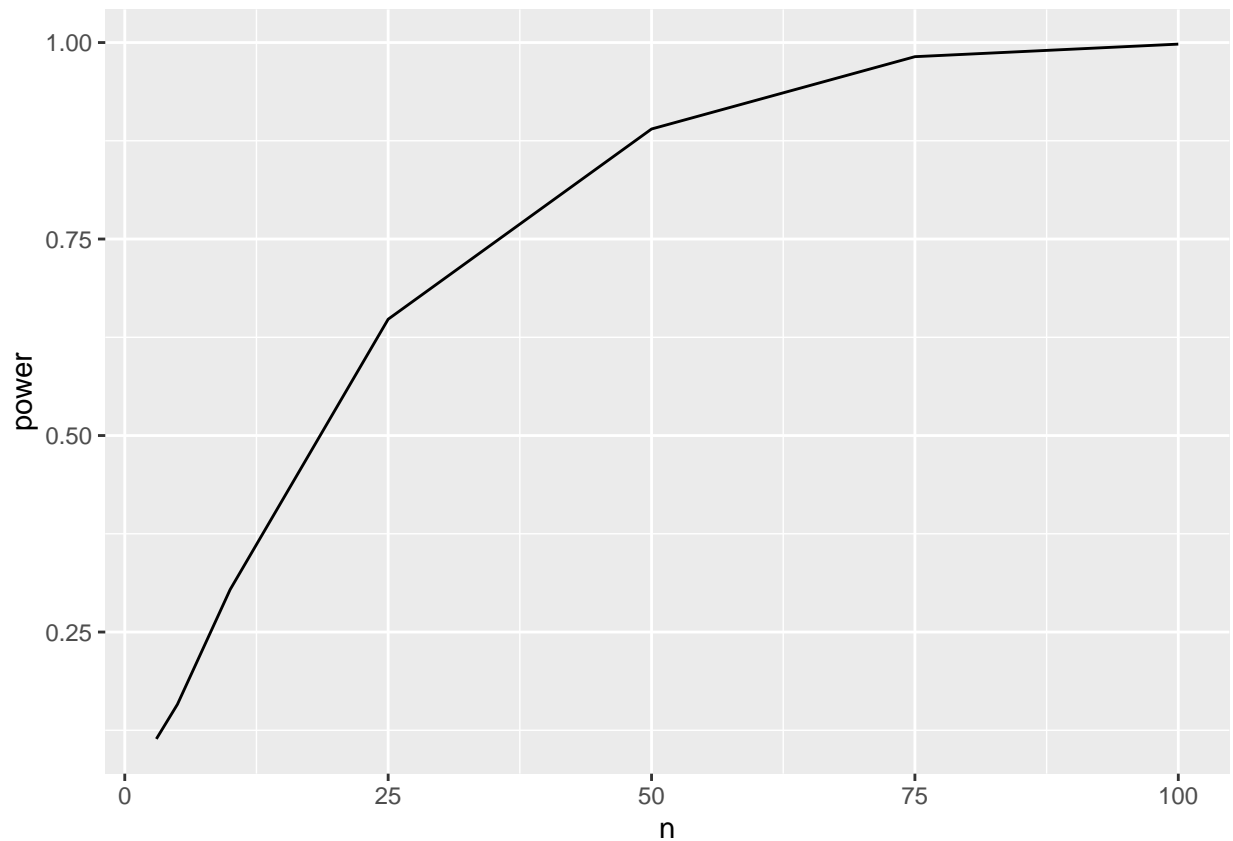
power <- data.frame(n=c(3,5,10,25,50,75,100),power=NA)

for (i in 1:nrow(power))
{
  n1 <- n2 <- power$n[i]
  predictorData <- data.frame(Diet = rep(c("c","hf"),c(n1,n2)) %>% as.factor)
  simResults <- t(sapply(1:nSim,simLm,form = ~Diet,data = predictorData,betas = c(b0,b1),sd = sd,contrasts = "Diethf = 0"))
  power$power[i] <- mean(simResults[, "pvalues"] < alpha)
}
power

```

```
##      n power
## 1     3 0.114
## 2     5 0.158
## 3    10 0.304
## 4    25 0.648
## 5    50 0.890
## 6    75 0.982
## 7   100 0.998
```

```
power %>%
  ggplot(aes(x=n,y=power)) +
  geom_line()
```



Bestudeer power in functie van steekproefgrootte en effectgrootte Merk op dat

- het teken van delta arbitrair is omdat we tweezijdig testen.
- het intercept arbitrair is omdat we alleen testen op β_1
- We nemen daarom typisch $\beta_0 = 0$

```
nSim <- 1000
b0 <- 0
deltas <- c(1,2,3,5,10)
sd <- round(miceSum$sd[2],1)
ns <- c(3,5,10,20,25,50,75,100)
```

```

power <- data.frame(b1=rep(deltas,each=length(ns)),
                   n=rep(ns,length(deltas)),
                   power=NA)

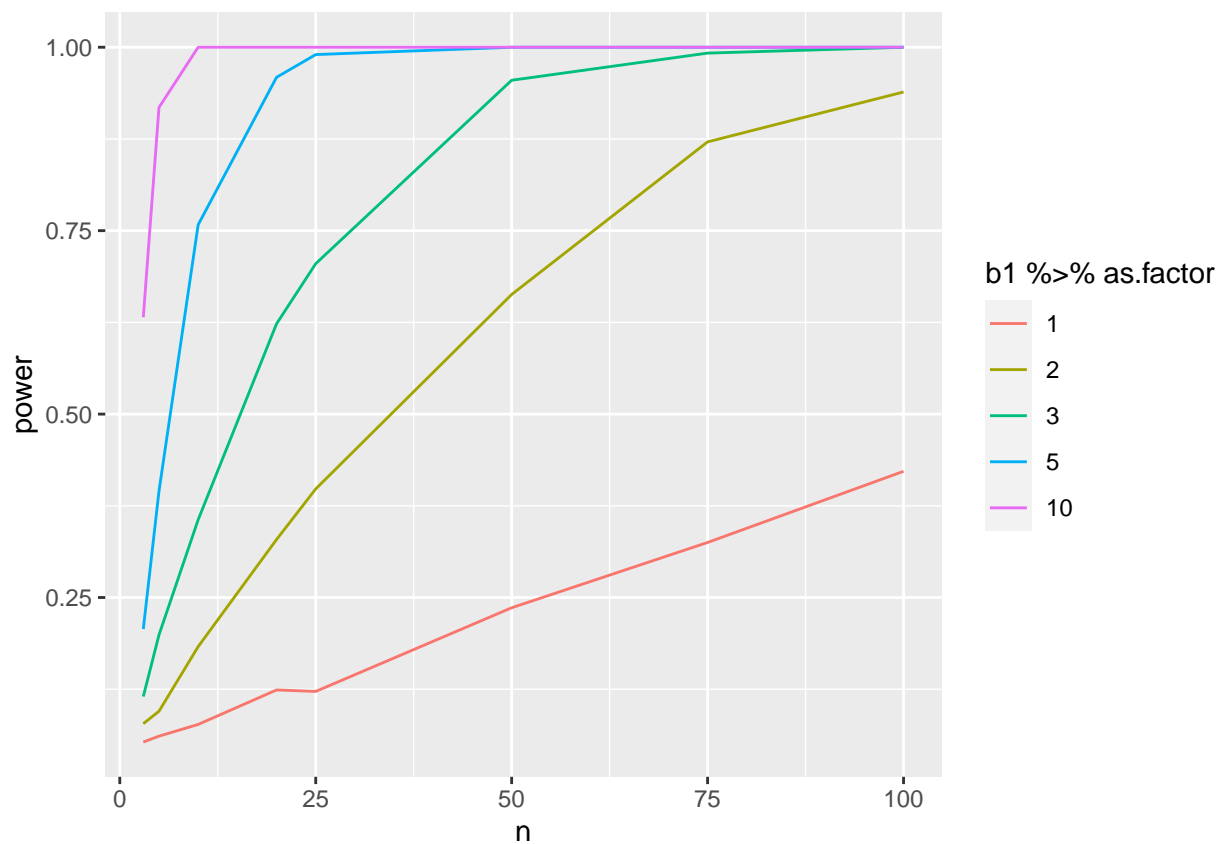
for (i in 1:nrow(power))
{
  b1 <- power$b1[i]
  n1 <- n2 <- power$n[i]
  predictorData <- data.frame(Diet = rep(c("c","hf"),c(n1,n2)) %>% as.factor)
  simResults <- t(sapply(1:nSim,simLm,form = ~Diet,data = predictorData,betas = c(b0,b1),sd = sd,contr=
  power$power[i] <- mean(simResults[, "pvalues"] < alpha)
}

```

```

power %>%
  ggplot(aes(x=n,y=power,col=b1%>%as.factor)) +
  geom_line()

```



Merk op dat de power curves wat schokkerig zijn omdat we maar een beperkt aantal steekproefgroottes en een beperkt aantal simulaties uitvoerden. Voor de lage powers is het aantal simulaties zeker onvoldoende om heel betrouwbare schattingen van de power te bekomen.

Opmerkingen

- De code kan eenvoudig aangepast worden naar andere designs (zie oefeningen)

- predictor data
 - formula
 - contrast
- De simulaties kunnen lang duren wanneer je veel scenario's evalueert.
 - Meer efficiënte code met matrices
 - Voor two group comparison bestaat een analytische oplossing.

Meer efficiënte code met matrices

Code is veel sneller. We simuleren nu 20 keer meer experimenten in een veel kortere tijd!

```
nSim <- 20000
b0 <- 0
sd <- round(miceSum$sd[2],1)
ns <- c(3,5,10,20,25,50,75,100)
deltas <- c(1,2,3,5,10)

powerFast <- matrix(NA,nrow=length(ns)*length(deltas),ncol=3) %>% as.data.frame
names(powerFast) <- c("b1", "n", "power")

i <- 0

for (n in ns)
{
  n1 <- n2 <- n

  ### Simulation
  predictorData <- data.frame(Diet = rep(c("c", "hf"),c(n1,n2)) %>% as.factor)
  design <- model.matrix(~Diet,predictorData)
  L <- limma::makeContrasts("Diethf",levels = colnames(design))

  for (b1 in deltas)
  {
    ySim <- rnorm(nrow(predictorData)*nSim,sd=sd)
    dim(ySim) <-c(nrow(predictorData),nSim)
    ySim <- ySim + c(design %*%c(b0,b1))
    ySim <- t(ySim)

    ### Fitting
    fitAll <- limma::lmFit(ySim,design)

    ### Inference
    varUnscaled <- c(t(L)%*%fitAll$cov.coefficients%*%L)
    contrasts <- fitAll$coefficients %*%L
    seContrasts <- varUnscaled^.5*fitAll$sigma
    tstats <- contrasts/seContrasts
    pvals <- pt(abs(tstats),fitAll$df.residual,lower.tail = FALSE)*2

    i <- i+1
  }
}
```

```
    powerFast[i,] <- c(b1,n,mean(pvals < alpha))
  }
}
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

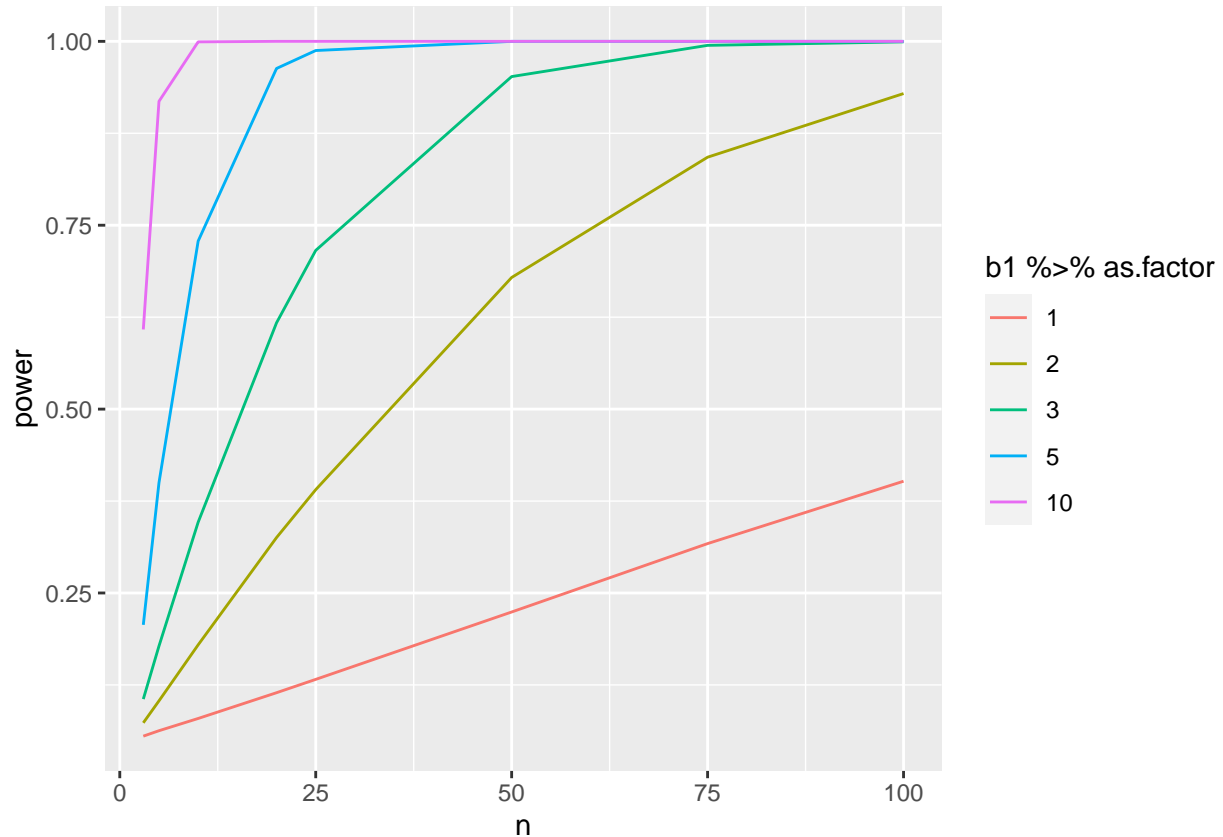
```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
## Warning in limma::makeContrasts("Diethf", levels = colnames(design)): Renaming
## (Intercept) to Intercept
```

```
powerFast %>%
  ggplot(aes(x=n,y=power,col=b1%>%as.factor)) +
  geom_line()
```

Meer efficiënte code gebaseerd op de analytische oplossing voor een two group comparison

- Voor de two sample t-test bestaat er een analytische oplossing om de power te berekenen.
- In deze context maakt men typisch gebruik van de “Cohen’s effect size”:

$$D = \frac{\delta}{sD}$$

```
power.t.test(n=8, delta = delta, sd = sd, type='two.sample')
```

```
##
## Two-sample t test power calculation
##
##      n = 8
##      delta = 2.7
##      sd = 4.1
##      sig.level = 0.05
##      power = 0.2320512
##      alternative = two.sided
##
## NOTE: n is number in *each* group
```

Merk op dat de power overeenkomt met degene die we via simulaties hebben geschat!

```

b0 <- 0
sd <- round(miceSum$sd[2],1)
ns <- c(3,5,10,20,25,50,75,100)
deltas <- c(1,2,3,5,10)

powerTheo <- data.frame(deltas=rep(deltas,each=length(ns)),
                        n=rep(ns,length(deltas)),
                        power=NA)

powerTheo$power <- apply(powerTheo[,1:2],1,function(x) power.t.test(delta=x[1],n=x[2],sd=sd,type="two.s

powerTheo %>%
  ggplot(aes(x=n,y=power,col=deltas%>%as.factor)) +
  geom_line()

```

