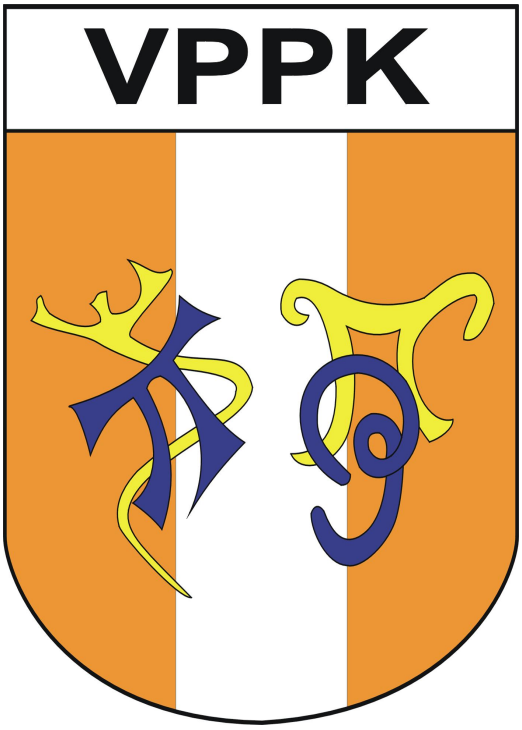
Statistiek II

Sessie 4

Feedback

Deel 4



VPPK

Universiteit Gent

2017-2018

Feedback Oefensessie 4

We hebben besloten de bekomen grafieken in `R` *niet in het document in te voegen, dit omdat het document met de code al vrij lang is, en de grafieken nog veel meer plaats zouden in beginnen nemen dat het onoverzichtelijk wordt, en ook niet meer echt milieuvriendelijk als er mensen dit willen printen. We gaan ervan uit dat jullie met het volgen van de code zelf de grafieken kunnen maken (aangezien ook de code van de grafieken in deze bundel staan).*

1 Statismex en bloeddruk

1. Afhankelijke variabele: Bloeddruk (`bloeddruk$BD`)
   Onafhankelijke variabele: Dosis (`bloeddruk$dosis`)
   $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$
   Nulmodel: bloeddruk $= \beta_0 + \varepsilon$
   Enkelvoudig lineair model: bloeddruk $= \beta_0 + \beta_1 \text{dosis} + \varepsilon$

2. Correctie 38ste observatie:

   ```
   > bloeddruk$slaperigheid2[38] <- 19
   ```

3. De lineariteit is zeer moeilijk te zien op deze plot. We kunnen zeggen dat er misschien een zeer klein stijgend verband is. De homoscedasticiteit is in orde:

   ```
   > plot(bloeddruk$dosis, bloeddruk$BD)
   ```

4. Aanmaken van `myLM`:

   ```
   > myLM <- lm(formula = bloeddruk$BD ~ bloeddruk$dosis)
   ```

5. Doormiddel van de functie `summary()` gaan we de resultaten bekijken:

   ```
   > summary(myLM)

   Call:
   lm(formula = bloeddruk$BD ~ bloeddruk$dosis)

   Residuals:
       Min      1Q  Median      3Q     Max
   -3.4163 -0.7119  0.2881  0.5837  3.2881

   Coefficients:
   Estimate Std. Error t value Pr(>|t|)
   (Intercept)     11.34234    0.32927  34.447   <2e-16 ***
   bloeddruk$dosis  0.07391    0.09735   0.759     0.45
   ---
   Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

   Residual standard error: 1.361 on 98 degrees of freedom
   Multiple R-squared:  0.005848,        Adjusted R-squared:  -0.004296
   F-statistic: 0.5765 on 1 and 98 DF,  p-value: 0.4495
   ```

   Als antwoord op de vorige vraag zien we dat we de nulhypothese aanvaarden omdat de p-waarde groter is dan 0.05, namelijk $p = 0.4495$. De bekomen puntschattingen die we kunnen aflezen:

   - $\hat{\beta}_0 = 11.34234$

- $\hat{\beta}_1 = 0.07391$

- $\hat{\sigma}_\varepsilon = 1.361$ (Dit staat in de output als `Residual standard error`)

6. Als we de gegeven formule omzetten in `R` krijgen we:

```
> sqrt(sum(residuals(myLM)^2)/(length(bloeddruk$BD)-2))
[1] 1.360682
```

   Dit is hetzelfde resultaat als in de vorige deelvraag.

7. We zien geen grote afwijkingen van de normaliteit. Een diagonaal past vrij goed in de plot:

```
> qqnorm(residuals(myLM))
```

8. We gebruiken de gekende formule voor SS_X en krijgen dan volgende resultaat:

```
> SSx <- sum((bloeddruk$dosis - mean(bloeddruk$dosis))^2)
> SSx
[1] 195.36
```

9. We zien dat 0 in het betrouwbaarheidsinterval ligt, dit wil dus zeggen dat we de nulhypothese aanvaarden. Met andere woorden is er geen effect van dosis op bloeddruk:

```
> confint(myLM, level=0.95)
                     2.5 %    97.5 %
(Intercept)     10.6889172 11.995767
bloeddruk$dosis -0.1192744  0.267104
```

10. Als we de formules voor het betrouwbaarheidsinterval gebruiken krijgen we:

```
> n <- length(bloeddruk$BD)
> b1 <- myLM$coefficients[2]
> KW <- qt(p=0.975, df=n-2)
> grens <- KW*(sqrt(1.360682^2/SSx))
> b1 - grens
bloeddruk$dosis
         -0.1192744
> b1 + grens
bloeddruk$dosis
           0.267104
```

    In de gegeven code staat `KW` voor de kritieke waarde in de nodige t-verdeling. We komen dezelfde resultaten uit als bij de vorige deelvraag.

11. We berekenen eerst de toetsingsgrootheid:

```
> SSres <- sum((bloeddruk$BD-fitted(myLM))^2)
> SSres <- 1.360682^2*(n-2)
> g <- b1/sqrt(SSres/((n-2)*SSx))
> g
bloeddruk$dosis
      0.7592639
```

    Deze waarde vinden we ook in de output van deelvraag 5, met name de derde waarde in de rij `bloeddruk$dosis`. Nu gaan we de toetsingsgrootheid gebruiken om de p-waarde te berekenen. Let op, we testen tweezijdig. Wanneer we een t-test uitvoeren in deze context is deze altijd tweezijdig:

```
> (1-pt(g, df=n-2))*2
bloeddruk$dosis
      0.4495172
```

Deze waarde kunnen we op 2 plaatsen vinden in de output. Met name de laatste waarde in de rij van dosis, en helemaal op het einde van de output. Wanneer we 1 predictor hebben komen de p-waarde van de t-test en de F-test overeen.

12. We weten van de vorige sessie dat $SS_{\text{Res0}} = SS_Y$. Dus we gebruiken de formule voor SS_Y:

```
> SSres0 <- sum((bloeddruk$BD-mean(bloeddruk$BD))^2)
> SSres0
[1] 182.51
```

13. We berekenen de F-verhouding zoals gegeven in de cursus:

```
> f <- (SSres0 - SSres)/(SSres/(n-2))
> f
[1] 0.5764981
```

De waarde vinden we op de laatste lijn bij `F-statistic`. We kunnen dan ook de bijhorende p-waarde gaan berekenen, met vrijheidsgraden 1 en $n - 2 = 98$ (deze staan ook in de output op dezelfde regel):

```
> 1 - pf(f, df1=1, df2=n-2)
[1] 0.4495107
```

We bekomen dezelfde p-waarde als in de output en als bij de t-toets.

14. We bereken R^2 op basis van de formule, dit geeft:

```
> rsquared <- (SSres0 - SSres)/SSres0
> rsquared
[1] 0.00584823
```

We verkrijgen inderdaad dezelfde waarde. Dit wil zeggen dat 0.58% van het model verklaart wordt door onze predictor. Wat zeer weinig is. We bereken ook $\bar{R}^2$:

```
> rsquared_bar <- 1 - (1-rsquared)*((n-1)/(n-2))
> rsquared_bar
[1] -0.004296176
```

Aangezien een proportie nooit negatief kan zijn (en onze R^2 staat voor de proportie verklaarde variantie) stellen we een negatief getal gelijk aan 0. Dus op basis van deze waarde concluderen we dat er niets verklaard wordt in het model door dosis.

15. Voor de power te berekenen gaan we eerst onze package laden:

```
> library(pwr)
```

We gaan dan onze ρ berekenen met de gevraagde $\beta_1 = 1/5$:

```
> rho <- (1/5)*(sd(bloeddruk$dosis)/sd(bloeddruk$BD))
```

De kritische student zal al gemerkt hebben dat hier een fout is gemaakt door prof. Marchant. Als we een bloeddrukstijging van 1 eenheid willen kunnen detecteren wanneer de dosis van 1 tot 5 mg stijgt, dan stijgen we niet 5 mg, maar 4 mg, en willen we dus een $\beta_1 \geq 1/4$ kunnen detecteren.
De power van onze huidige steekproef:

```
> pwr.r.test(n=n, r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 100
              r = 0.206921
      sig.level = 0.05
          power = 0.5465004
    alternative = two.sided
```

We zien dus een power van 0.55 wat gemiddeld is, maar we willen over het algemeen een grotere power. Als we een power van 0.9 willen garanderen hebben we een grotere steekproef nodig, met name 241:

```
> pwr.r.test(power=0.90, r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 240.4769
              r = 0.206921
      sig.level = 0.05
          power = 0.9
    alternative = two.sided
$
```

2 Statismex: slaperigheid2 en gewicht

1. Voor het spreidingsdiagram gebruiken we:

   ```
   > plot(bloeddruk$gewicht, bloeddruk$slaperigheid2)
   ```

   Onze richtingscoëfficiënt kunnen we benaderen door te zien hoeveel onze slaperigheid stijgt als we 10 eenheden op gewicht omhoog gaan. Dit is ongeveer 5. Dus als we maar 1 eenheid omhoog gaan op gewicht zullen we $5/10 = 0.5$ omhoog gaan op slaperigheid. Om ons intercept te schatten moeten we zien welke waarde slaperigheid heeft wanneer gewicht gelijk is aan 0. Dus we moeten de grafiek mentaal gaan vergroten, dat we meer waarden voor gewicht zouden krijgen. We zien dat wanneer gewicht 60 is, dat dan slaperigheid 0 is. Dus als we dan 60 eenheden dalen voor gewicht dan zullen we 60×0.5 dalen voor slaperigheid. Dus we komen op een intercept van ongeveer 30.
   Als we gewicht gaan herschalen van kg naar g, dan zal ons intercept hetzelfde blijven. Want 0 kg is hetzelfde als 0 g. Maar onze richtingscoëfficiënt zal veranderen. Als we 0.5 stijgen op slaperigheid wanneer we 1 kg stijgen, dan kunnen we dit gelijk stellen aan 0.5 stijgen op slaperigheid als we 1000 g stijgen. Dus voor 1 g stijging zullen we 0.5/1000 op slaperigheid stijgen. Onze richtingscoëfficiënt zal dus 1000 keer kleiner zijn.

2. $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$ (dit is altijd hetzelfde voor een enkelvoudige lineaire regressie)

   ```
   > myLM <- lm(bloeddruk$slaperigheid2 ~ bloeddruk$gewicht)
   > qqnorm(residuals(lmgewicht))
   > plot(bloeddruk$gewicht, residuals(lmgewicht))
   ```

   In de QQ-plot zien we dat de residuals vrij normaal verdeeld zijn, en in het spreidingsdiagram zien we dat de data homoscedastisch is.

   ```
   > summary(lmgewicht)
   ```

```
Call:
lm(formula = bloeddruk$slaperigheid2 ~ bloeddruk$gewicht)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0192  -2.9653   0.1945   2.7111  11.1260

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)       -36.21723    3.39400  -10.67   <2e-16 ***
bloeddruk$gewicht   0.64315    0.04778   13.46   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 4.412 on 98 degrees of freedom
Multiple R-squared:  0.649,        Adjusted R-squared:  0.6454
F-statistic: 181.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

We hebben een zeer kleine p-waarde, veel kleiner dan 0.05 dus we verwerpen de nulhypothese. Er is een lineair verband tussen gewicht en slaperigheid.

3. De proportie verklaarde variantie kunnen we aflezen in de output bij de `Multiple R-squared`. Dus $R^2 = 0.649$. SS_Y bereken we via de gekende formule:

```
> SSy <- sum((bloeddruk$slaperigheid2-mean(bloeddruk$slaperigheid2))^2)
> SSy
[1] 5433.36
```

En uit de formules weten we dat $R^2 = SS_{\text{Mod}}/SS_Y$, als we dit hervormen krijgen we $SS_{\text{Mod}} = R^2 * SS_Y$:

```
> SSMod <- 0.649 * SSy
> SSMod
[1] 3526.251
```

4. Eerst berekenen we terug ρ zoals reeds gezien, om deze dan in de power-functie te steken:

```
> rho <- (1/4)*(sd(bloeddruk$gewicht)/sd(bloeddruk$slaperigheid2))
> pwr.r.test(n=length(bloeddruk$slaperigheid2), r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 100
              r = 0.3131414
      sig.level = 0.05
          power = 0.8935299
    alternative = two.sided
$
```

We observeren een power van 0.89 welke vrij goed is.

5. Als we de gegeven functies gebruiken in `R` dan krijgen we punten die allemaal mooi op een diagonaal liggen. Elk punt representeert de verwachte waarde van slaperigheid die bij een bepaald gewicht hoort. Dus de rechte die we door de punten kunnen trekken is onze regressierechte.

6. De plot die we bekomen op basis van het gegeven commando is exact dezelfde als die van het begin. Want we berekenen eerst de verwachte waarde van elke observatie, en dan gaan we de residuen (of anders

gezegd de afwijking van de verwachte waarde) er terug bijvoegen. Dus krijgen we terug de oorspronkelijke geobserveerde waarde.

3 Statismex: slaperigheid2 en geslacht

1. $H_0 : \mu_{\male} = \mu_{\female}$ vs. $H_0 : \mu_{\male} \neq \mu_{\female}$
   Aangezien $n > 30$ mogen we verder met de Welch-test.

```
> man <- bloeddruk$slaperigheid2[bloeddruk$geslacht=="m"]
> vrouw <- bloeddruk$slaperigheid2[bloeddruk$geslacht=="v"]
> nm <- length(man)
> nv <- length(vrouw)
> t.test(man, vrouw)

Welch Two Sample t-test

data:  man and vrouw
t = -0.086205, df = 89.609, p-value = 0.9315
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.127795  2.867659
sample estimates:
mean of x mean of y
9.018868  9.148936
```

   We zien dat de schattingen voor de verwachtingen zijn: voor mannen 9.018868 en voor vrouwen 9.148936. We krijgen een p-waarde van 0.9315 en dus aanvaarden we de nulhypothese.

2. De power-berekening bij een t-test is in een vorige bundel al uitgebreid besproken, dus gaan we er hier niet gedetailleerd op in. We observeren een power van 0.51:

```
> sp <- sqrt((52*var(man) + 46* var(vrouw))/(nm+nv-2))
> d <- 3/sp
> pwr.t2n.test(n1=nm, n2=nv, d=d, sig.level=0.05, alternative="two.sided")

     t test power calculation

             n1 = 53
             n2 = 47
              d = 0.4029182
      sig.level = 0.05
          power = 0.5125123
    alternative = two.sided
```

3. De predictor is geslacht en de afhankelijke variabele slaperigheid2. Het meetniveau van de predictor moet ofwel minstens intervalniveau zijn ofwel dichotoom. Het meetniveau van de afhankelijke variabele moet minstens van intervalniveau zijn:

```
> myLM <- lm(formula = bloeddruk$slaperigheid2 ~ bloeddruk$geslacht)
> qqnorm(residuals(myLM))
> plot(bloeddruk$geslacht, residuals(myLM))
> summary(lmgeslacht)
```

```
Call:
lm(formula = bloeddruk$slaperigheid2 ~ bloeddruk$geslacht)

Residuals:
   Min     1Q Median     3Q    Max
-9.149 -7.019 -1.084  6.199 15.981

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)           9.0189     1.0227   8.818 4.42e-14 ***
bloeddruk$geslachtv   0.1301     1.4918   0.087    0.931
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 7.446 on 98 degrees of freedom
Multiple R-squared:  7.756e-05,        Adjusted R-squared:  -0.01013
F-statistic: 0.007602 on 1 and 98 DF,  p-value: 0.9307
```

We krijgen dezelfde p-waarde als bij de Welch-toets. Herinner van een vorige bundel dat een F-statistiek bij een enkelvoudige lineaire regressie hetzelfde is als het kwadraat van de t-statistiek. Dus we komen ook op dezelfde p-waarde en dezelfde conclusie.

4 Marketingbudget en verkoopcijfers

1. $H_0 : \beta_1 = 0$ vs. $H_a \neq 0$

   ```
   > plot(lancering$budget, lancering$verkoop)
   ```

2. We gaan terug een enkelvoudige lineaire regressie gebruiken omdat beide variabelen van minstens intervalniveau zijn. We zien dat de assumpties toch wel voldaan zijn, er zijn geen grote afwijkingen.

   ```
   > myLM <- lm(formula = lancering$verkoop ~ lancering$budget)
   > qqnorm(residuals(myLM))
   > plot(fitted(myLM), residuals(myLM))
   ```

3. We vinden een p-waarde van 0.0175 en verwerpen dus de nulhypothese. De verkoop kan verklaard worden door het budget.

   ```
   > summary(lmbudget)

   Call:
   lm(formula = lancering$verkoop ~ lancering$budget)

   Residuals:
       Min      1Q  Median      3Q     Max
   -815477 -170746  -24199  195515  783864

   Coefficients:
   Estimate Std. Error t value Pr(>|t|)
   (Intercept)      5.526e+05  9.914e+04   5.575 8.05e-08 ***
   lancering$budget 2.375e+00  9.908e-01   2.397   0.0175 *
   ---
   Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
   ```

```
Residual standard error: 291000 on 198 degrees of freedom
Multiple R-squared:  0.0282,        Adjusted R-squared:  0.02329
F-statistic: 5.746 on 1 and 198 DF,  p-value: 0.01746
```

4. We gaan terug eerst ρ berekenen om de power te berekenen.

```
> rho <- 2*(sd(lancering$budget)/sd(lancering$verkoop))
> pwr.r.test(n=length(lancering$budget), r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 200
              r = 0.1414208
      sig.level = 0.05
          power = 0.5170537
    alternative = two.sided
```

   We observeren dus een power van 0.52 welke gemiddeld is, maar we willen liever een grotere power.

5. Om een power van 90% te bekomen hebben we minstens $n = 521$ nodig:

```
> pwr.r.test(power=0.90, r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 520.4506
              r = 0.1414208
      sig.level = 0.05
          power = 0.9
    alternative = two.sided
```

5 Coma en IQ

1. $\texttt{piq} = \beta_0 + \beta_1 \texttt{days} + \beta_2 \texttt{duration} + \beta_3 \texttt{age} + \varepsilon$

2. We kunnen deze vraag benaderen door ofwel plots op te stellen, ofwel de correlaties te berekenen:

```
> plot(coma$piq, coma$days)
> plot(coma$piq, coma$duration)
> plot(coma$piq, coma$age)
> cor(coma$piq, coma$days)
[1] 0.02255264
> cor(coma$piq, coma$duration)
[1] -0.2869127
> cor(coma$piq, coma$age)
[1] 0.1120887
```

   We verwachten een positief verband met de leeftijd, een negatief verband met de duur en over de dagen kunnen we niet echt iets zeggen omdat het een zeer kleine correlatie is.

3. We krijgen dezelfde indrukken als bij de vorige deelvraag. We mogen een meervoudige lineaire regressie gebruiken omdat alle predictoren van minstens intervalniveu zijn.

```
> pairs(coma[c(2,3,5,6)], lower.panel=NULL)
```

4. We zien dat de verwachtingen uitkomen.

```
> myLM <- lm(formula = coma$piq ~ coma$days + coma$duration + coma$age)
> summary(myLM)

Call:
lm(formula = coma$piq ~ coma$days + coma$duration + coma$age)

Residuals:
    Min      1Q  Median      3Q     Max
-34.787 -10.180  -0.182   8.417  41.834

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   84.1125810  2.7212421  30.910  < 2e-16 ***
coma$days      0.0009488  0.0007460   1.272    0.205
coma$duration -0.2643666  0.0650955  -4.061 7.09e-05 ***
coma$age       0.0599741  0.0693659   0.865    0.388
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 13.45 on 193 degrees of freedom
Multiple R-squared:  0.09212,        Adjusted R-squared:  0.078
F-statistic: 6.527 on 3 and 193 DF,  p-value: 0.0003155
```

5. De assumpties blijken stand te houden als we naar de plots kijken

6. Nulmodel: $\texttt{piq} = \beta_0 + \varepsilon$
   $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ vs H_a : minstens 1 β verschilt van 0.

7. In deelvraag 4 hebben we de output van deze functie al. We hebben $p = 0.0003155$ en dus verwerpen we de nulhypothese.

8. Met de `anova()` functie komen we op dezelfde p-waarde uit:

```
> lmcoma0 <- lm(formula = coma$piq ~ NULL)
> anova(lmcoma0, myLM)
Analysis of Variance Table

Model 1: coma$piq ~ NULL
Model 2: coma$piq ~ coma$days + coma$duration + coma$age
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    196 38438
2    193 34897  3    3540.8 6.5275 0.0003155 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

9. De proportie verklaarde variantie kunnen we aflezen uit de output van deelvraag 4: $R^2 = 0.09212$. Dus ongeveer 9% van `piq` wordt verklaard door onze 3 predictoren. Dit is vrij weinig.

10. Mochten we IQ voor de coma mee opnemen gaat onze proportie verklaarde variantie zwaar de hoogte in. Maar dan is er veel kans dat we de andere invloeden niet meer gaan kunnen detecteren. Het is vaak interessanter om evidente variabelen weg te laten omdat deze te veel van de variantie naar zich gaan trekken, om zo dan de invloed van andere variabelen beter te bestuderen.

11. We moeten naar de grafieken kijken waar de predictoren met elkaar vergeleken worden. We zien hier niet echt tendensen in voor collineariteit. Dit wordt bevestigd door de functie `vif()`, want ze zitten allemaal vrij dicht bij 1:

```
> library(car)
> vif(myLM)
    coma$days coma$duration      coma$age
     1.062245      1.083266      1.081130
```

6 Statismex: gewicht en geslacht

1. $H_0 : \mu_{\male} = \mu_{\female}$ vs $H_a : \mu_{\male} \neq \mu_{\female}$
   Omdat dit zeer gelijkaardig is met vraag 3 gaan we niet overal gedetailleerd op in.

```
> man <- bloeddruk$gewicht[bloeddruk$geslacht=="m"]
> vrouw <- bloeddruk$gewicht[bloeddruk$geslacht=="v"]
> nm <- length(man)
> nv <- length(vrouw)
> t.test(man, vrouw)

Welch Two Sample t-test

data:  man and vrouw
t = -0.88241, df = 97.503, p-value = 0.3797
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-5.320323  2.045333
sample estimates:
mean of x mean of y
69.66038  71.29787
```

   De assumptie van $n > 30$ is voldaan. De verwachting voor gewicht bij mannen is 69.66 en bij vrouwen 71.30. We hebben $p = 0.3797$ en dus aanvaarden we de nulhypothese.

2. De geobserveerde power is 0.19 en dus vrij klein:

```
> sp <- sqrt((52*var(man) + 46* var(vrouw))/(nm+nv-2))
> d <- 2/sp
> pwr.t2n.test(n1=nm, n2=nv, d=d, sig.level=0.05, alternative="two.sided")

     t test power calculation

             n1 = 53
             n2 = 47
              d = 0.2152863
      sig.level = 0.05
          power = 0.1863792
    alternative = two.sided
```

3. De predictor is geslacht en de afhankelijke variabele gewicht. Het meetniveau van de predictor moet ofwel minstens intervalniveau zijn ofwel dichotoom. Het meetniveau van de afhankelijke variabele moet minstens van intervalniveau zijn:

```
> myLM <- lm(formula = bloeddruk$gewicht ~ bloeddruk$geslacht)
> plot(bloeddruk$geslacht, residuals(myLM))
> qqnorm(residuals(myLM))
> summary(myLM)

Call:
lm(formula = bloeddruk$gewicht ~ bloeddruk$geslacht)

Residuals:
     Min       1Q   Median       3Q      Max
-17.6604  -7.3885  -0.6604   6.3396  21.3396

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)           69.660      1.276   54.59   <2e-16 ***
bloeddruk$geslachtv    1.637      1.861    0.88    0.380
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 9.29 on 98 degrees of freedom
Multiple R-squared:  0.007835,        Adjusted R-squared:  -0.002289
F-statistic: 0.7739 on 1 and 98 DF,  p-value: 0.3797
```

De assumpties zijn voldaan. De p-waarde is dezelfde als in deelvraag 1.
We kunnen de verwachtingen opvragen via de functie `fitted()`. We krijgen dan dezelfde verwachtingen als in deelvraag 1:

```
> fitted(lmgeslacht2)
1        2        3        4        5        6        7        8        9       10
69.66038 71.29787 69.66038 69.66038 71.29787 69.66038 69.66038 69.66038 71.29787 71.
...
```

4. Ja, de residuen waren normaalverdeeld.

5. Ja, de boxplots toonden homoscedasticiteit. De schattingen van de varianties liggen ook zeer dicht bij elkaar:

```
> var(man)
[1] 90.34398
> var(vrouw)
[1] 81.73543
```

Wanneer we beide plots maken op basis van de gegeven codes dan is de boxplot veel handiger om de homoscedasticiteit te testen.

6. De power gaan we terug zoals eerder berekenen, door eerst ρ te berekenen.

```
> rho <- 2*(sd(as.numeric(bloeddruk$geslacht))/sd(bloeddruk$gewicht))
> pwr.r.test(n=length(bloeddruk$gewicht), r=rho, sig.level=0.05)

     approximate correlation power calculation (arctangh transformation)

              n = 100
              r = 0.1081141
```

```
  sig.level = 0.05
      power = 0.1887471
alternative = two.sided
```

We zien dat de power zeer dicht aanleunt bij de power van deelvraag 2.