

Oefeningen hoofdstuk 5 - Toetsingsprocedures

TijsMartens

3 april 2019

oefening 5.1.

(oefening zelf gemaakt, geen oplossing)

opgave

kunnen we in voorbeeld 5.1. zomaar veronderstellen dat het gemiddelde normaal verdeeld is, waarom

oplossing

Ja, door de centrale limietstelling

oefening 5.2.

(oefening zelf gemaakt, geen oplossing)

opgave

Wat zou je in vergelijking 5.3. moeten veranderen opdat je de correctie kritieke waarde zou berekenen voor een linkszijdige toets

oplossing

originieel (voor rechtse toets): $g = \mu + z * \frac{\sigma}{\sqrt{n}}$

oplossing voor linkse toets: $g = \mu - z * \frac{\sigma}{\sqrt{n}}$

oefening 5.3.

(oplossing vanuit de les)

betrouwbaarheidsintervallen

opgave

1. Wat is de onder- en bovengrens van een betrouwbaarheidsinterval van 99%?
2. Een betrouwbaarheidsinterval van 99% is breder dan een van 95%. Waarom is dit zo?
3. Hoe zou het betrouwbaarheidsinterval voor 100% er uit zien?

oplossing

oplossing:

- 1.

betrouwbaarheidsinterval = 99% => alpha = 1% => er is een tweezijdige toets, de alpha moet in de berekeningen dus gedeeld worden door 2

```
ondergrens <- qnorm(0.01/2)
ondergrens
```

```
## [1] -2.575829
```

```
bovengrens <- -1 * ondergrens
bovengrens
```

```
## [1] 2.575829
```

2.

Als een betrouwbaarheidsinterval 99% is. Dan weten we dat 99% procent van de populatiegemiddelden binnen het betrouwbaarheidsinterval liggen.

Bij het al dan niet verwerpen van een hypothese wordt er gekeken als de gemeten waarde binnen of buiten dit bereik valt. Als het er buiten valt, mag de hypothese verworpen worden.

Het is dus moeilijker om een hypothese te verwerpen waarbij een betrouwbaarheidsinterval gehanteerd wordt van 99% dan bij een betrouwbaarheidsinterval van 95%

Je kan zelf bepalen hoe “streng” je je testen maakt, * kleine alpha = breed betrouwbaarheidsinterval = moeilijk te verwerpen * grote alpha = smaller betrouwbaarheidsinterval = makkelijker om te verwerpen

3.

Je bent dan 100% zeker dat het populatiegemiddelde binnen het betrouwbaarheidsinterval ligt, de volledige x-as is dan je bereik]-oneindig, +oneindig[

elke waarde valt hier binnen, dit is dus helemaal niet nuttig.

oefening 5.4.

(oplossing vanuit de les)

opgave

Er wordt gezegd dat het invoeren van een bindend studieadvies (BSA) een rendementsverhoging tot gevolg heeft in slaagkans. Voor het invoeren van het BSA was in de studentenpopulatie het gemiddelde aantal behaalde studiepunten per jaar per student gelijk aan 44 met een standaardafwijking van 6,2. Na invoering van het BSA wijst een onderzoek uit onder 72 studenten dat deze een gemiddeld aantal studiepunten haalden van 46,2.

hypothesen

- $H_0 : \mu = 44$
- $H_1 : \mu > 44$ (i.e. er is een rendementsverhoging)

gegeven

- $\alpha = 2.5$
- $\sigma = 6.2$
- $n = 72$

oplossing met overschreidingskans

```
alpha <- 0.025
sigma <- 6.2
mean = 44
test = 46.2
n <- 72
p <- 1 - pnorm(test, mean, sd = sigma/sqrt(n))
p
```

```
## [1] 0.001302346
```

```
if (p < alpha) {
  print("we mogen H0 verwerpen")
}else
{
  print("we hebben niet genoeg bewijs om H0 te verwerpen")
}
```

```
## [1] "we mogen H0 verwerpen"
```

de p-test levere een resultaat op van 1.3% dit is kleiner dan 2.5 procent. het valt dus buiten het betrouwbaarheidsinterval. Dit heeft als gevolg dat we h0 mogen verwerpen

oplossing van kritiek gebied

formule om g te berekenen (grenswaarde)

$$g = \mu + z * \frac{\sigma}{\sqrt{n}}$$

qnorm = onder welk grens zal p% van de waarnemingen liggen

```
g <- qnorm(1 - alpha, mean, sd = sigma/sqrt(n))
g
```

```
## [1] 45.4321
```

```
test
```

```
## [1] 46.2
```

```
if (test > g) {
  print("we mogen H0 verwerpen")
}else
{
  print("we hebben niet genoeg bewijs om H0 te verwerpen")
}
```

```
## [1] "we mogen H0 verwerpen"
```

ons testgeval (waarde: 46.2) is groter dan onze grenswaarde (45.43). Omdat dit het geval is mogen we h0 verwerpen

geef een interpretatie van wat alpha = 2.5% betekend.

De alpha waarde bepaald hoe ‘streng’ we zullen testen. bij een alpha van 2.5% zullen 97.5% van de steekproefgemiddelden zicht in het aanvaardbaar gebied bevinden.

2.5% van de waarden zullen hierbuiten vallen. Deze worden gezien als “extreem” of “significant”. als het steekproefgemiddelde hiertoe behoort, kan de bijhorende nullhypothese verworpen worden.

Deze stellingen zijn enkel correct bij een normaal verdeelde set aan data

oefening 5.4. bis

(oefening zelf gemaakt, geen oplossing)

algemene gegevens

```
mu <- 44
sigma <- 6.2
z <- pnorm(46.2, mu, sigma)
alpha <- 0.025
n <- 72
test <- 46.2
```

1.

```
overschreidingswaardeG <- qnorm(1-alpha, mean, sd = sigma/sqrt(n))
```

```
overschreidingswaardeG
```

```
## [1] 45.4321
```

```
test
```

```
## [1] 46.2
```

de te testen waarde is groter dan de kritieke grenswaarde. Het gemiddelde is dus gestegen

2.

```
p <- 1 - pnorm(test, mean, sd = sigma/sqrt(n))
p
```

```
## [1] 0.001302346
```

```
alpha
```

```
## [1] 0.025
```

de overschreidingskans p (0.0013) is kleiner dan α , we mogen H_0 dus verwerpen

3.

als α 2.5% of 0.025 is kunnen we veronderstellen dat 97.5% van de steekproeven bij een normaal verdeelde dataset binnen het betrouwbaarheidsinterval liggen

Als een waarde buiten de kritieke grenswaarde valt, is er slechts 2.5% kans dat hij uit de populatie getrokken is. Deze kans is dus heel klein. In dit geval met de nulhypothese verworpen worden

oefening 5.5.

(oefening zelf gemaakt, geen oplossing)

opgave

Eén van de motieven voor het kiezen van een garage is de inruilprijs voor de oude auto. De importeur van Ford wil graag dat verschillende dealers een gelijk prijsbeleid voeren. De importeur vindt dat het gemiddelde prijsverschil tussen de dichtstbijzijnde Ford-dealer en de dealer waar men de auto gekocht heeft hoogstens

e300 mag bedragen. De veronderstelling is dat als het verschil groter is, potentiële klanten eerder geneigd zullen zijn om bij hun vorige dealer te blijven. In een steekproef worden volgende verschillen genoteerd:

400 350 400 500 300 350 200 500 200 250 250 500 350 100

Toets of er reden is om aan te nemen dat het gemiddelde prijsverschil in werkelijkheid significant groter is dan ???300. Gebruik een significantieniveau van 5%.

oplossing

```
data <- c(400,350,400,500,300,350,200,500,200,250,250,500,350,100)
mean <- mean(data)
sigma <- sd(data)
alpha <- 0.05
test <- 300
n <- 14
```

```
mean
```

```
## [1] 332.1429
```

```
sigma
```

```
## [1] 123.4241
```

```
g <- qnorm(1 - alpha, mean, sd = sigma/sqrt(n))
g
```

```
## [1] 386.4008
```

we zijn rechts aan het toetsten dus het steekproefgemiddelde ligt niet in het kritieke gebied. het steekproefgemiddelde situeert zich dus wel in het aanvaardbaar gebied

we mogen H_0 niet verwerpen

test = 300 grens = 386.40

```
p <- pnorm(test, mean, sd = sigma/sqrt(n))
p
```

```
## [1] 0.1649228
```

p is groter dan alpha, het ligt dus in het te aanvaarden gebied. we mogen de H_0 niet verwerpen

p = 0.16 alpha = 0.05

oefening 5.6.

opgave

In Oefening 3.9 en volgende hebben we de resultaten van performantiemetingen voor persistentiemogelijkheden in Android geanalyseerd (Akin, 2016). Er werden experimenten uitgevoerd voor verschillende combinaties van hoeveelheid data (klein, gemiddeld, groot) en persistentietype (GreenDAO, Realm, SharedPreferences, SQLite). Voor elke hoeveelheid data hebben we kunnen bepalen welk persistentietype het beste resultaat gaf. Nu gaan we uitzoeken of het op het eerste zicht beste persistentietype ook signi??cant beter is dan de concurrentie. Concreet: ga aan de hand van een toets voor twee steekproeven voor elke datahoeveelheid na of het gemiddelde van het best scorende persistentietype signi??cant lager is dan het gemiddelde van (i) het tweede beste en (ii) het slechtst scorende type . Kunnen we de conclusie aanhouden dat voor een gegeven datahoeveelheid één persistentietype het beste is, d.w.z. signi??cant beter is dan gelijk welk ander persistentietype

oplossing

(overgenomen van chamilo)

inlezen data

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.5.3
```

```
android_cpu <- read.csv("C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderzoekstechnieken/
```

voorbeeld 1: medium reals & dao

1. subset nemen van alle records waarbij de datahoeveelheid medium is

```
subset_medium <- subset(android_cpu, android_cpu$Datahoeveelheid=="Medium")
subset_medium
```

##	Tijd	PersistentieType	Datahoeveelheid
## 121	6.68	GreenDAO	Medium
## 122	5.57	GreenDAO	Medium
## 123	4.18	GreenDAO	Medium
## 124	8.80	GreenDAO	Medium
## 125	9.82	GreenDAO	Medium
## 126	6.31	GreenDAO	Medium
## 127	7.48	GreenDAO	Medium
## 128	6.64	GreenDAO	Medium
## 129	6.84	GreenDAO	Medium
## 130	6.54	GreenDAO	Medium
## 131	5.53	GreenDAO	Medium
## 132	8.46	GreenDAO	Medium
## 133	9.95	GreenDAO	Medium
## 134	8.18	GreenDAO	Medium
## 135	9.98	GreenDAO	Medium
## 136	10.25	GreenDAO	Medium
## 137	5.71	GreenDAO	Medium
## 138	7.64	GreenDAO	Medium
## 139	6.44	GreenDAO	Medium
## 140	7.79	GreenDAO	Medium
## 141	4.33	GreenDAO	Medium
## 142	5.99	GreenDAO	Medium
## 143	11.01	GreenDAO	Medium
## 144	7.59	GreenDAO	Medium
## 145	4.37	GreenDAO	Medium
## 146	5.27	GreenDAO	Medium
## 147	8.02	GreenDAO	Medium
## 148	7.37	GreenDAO	Medium
## 149	8.94	GreenDAO	Medium
## 150	11.94	GreenDAO	Medium
## 151	5.39	SQLite	Medium
## 152	5.20	SQLite	Medium
## 153	7.73	SQLite	Medium
## 154	7.14	SQLite	Medium
## 155	4.04	SQLite	Medium
## 156	5.49	SQLite	Medium
## 157	8.02	SQLite	Medium
## 158	6.68	SQLite	Medium

## 159	7.11	SQLite	Medium
## 160	8.58	SQLite	Medium
## 161	9.98	SQLite	Medium
## 162	7.05	SQLite	Medium
## 163	8.53	SQLite	Medium
## 164	9.26	SQLite	Medium
## 165	8.13	SQLite	Medium
## 166	8.92	SQLite	Medium
## 167	6.49	SQLite	Medium
## 168	8.43	SQLite	Medium
## 169	6.87	SQLite	Medium
## 170	8.93	SQLite	Medium
## 171	9.21	SQLite	Medium
## 172	7.02	SQLite	Medium
## 173	8.11	SQLite	Medium
## 174	7.11	SQLite	Medium
## 175	6.87	SQLite	Medium
## 176	9.13	SQLite	Medium
## 177	10.07	SQLite	Medium
## 178	7.62	SQLite	Medium
## 179	9.45	SQLite	Medium
## 180	11.26	SQLite	Medium
## 181	6.32	Realm	Medium
## 182	3.79	Realm	Medium
## 183	6.53	Realm	Medium
## 184	7.21	Realm	Medium
## 185	10.34	Realm	Medium
## 186	4.01	Realm	Medium
## 187	6.47	Realm	Medium
## 188	6.43	Realm	Medium
## 189	6.30	Realm	Medium
## 190	6.36	Realm	Medium
## 191	4.07	Realm	Medium
## 192	4.43	Realm	Medium
## 193	6.26	Realm	Medium
## 194	5.00	Realm	Medium
## 195	4.80	Realm	Medium
## 196	5.56	Realm	Medium
## 197	5.01	Realm	Medium
## 198	7.42	Realm	Medium
## 199	5.63	Realm	Medium
## 200	4.66	Realm	Medium
## 201	6.67	Realm	Medium
## 202	7.88	Realm	Medium
## 203	4.89	Realm	Medium
## 204	4.87	Realm	Medium
## 205	5.75	Realm	Medium
## 206	5.97	Realm	Medium
## 207	5.38	Realm	Medium
## 208	6.11	Realm	Medium
## 209	5.52	Realm	Medium
## 210	4.90	Realm	Medium

2. filteren van reals en dao

```
times_medium_realm <- subset_medium$Tijd[which(subset_medium$PersistentieType == "Realm")]
times_medium_greendao <- subset_medium$tijd[which(subset_medium$PersistentieType == 'GreenDao')]
```

```
times_medium_realm
```

```
## [1] 6.32 3.79 6.53 7.21 10.34 4.01 6.47 6.43 6.30 6.36 4.07
## [12] 4.43 6.26 5.00 4.80 5.56 5.01 7.42 5.63 4.66 6.67 7.88
## [23] 4.89 4.87 5.75 5.97 5.38 6.11 5.52 4.90
```

```
times_medium_realm
```

```
## [1] 6.32 3.79 6.53 7.21 10.34 4.01 6.47 6.43 6.30 6.36 4.07
## [12] 4.43 6.26 5.00 4.80 5.56 5.01 7.42 5.63 4.66 6.67 7.88
## [23] 4.89 4.87 5.75 5.97 5.38 6.11 5.52 4.90
```

3. uitvoeren van de test

```
t.test(times_medium_realm, times_medium_greendao, alternative = 'less')
```

```
##
## One Sample t-test
##
## data: times_medium_realm
## t = 23.943, df = 29, p-value = 1
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##      -Inf 6.230883
## sample estimates:
## mean of x
##      5.818
```

voorbeeld 2: small voor realm & shared preferences

1. subset nemen van alle records waarbij de datahoeveelheid small is

```
subset_small <- subset(android_cpu, android_cpu$Datahoeveelheid=="Weinig")
subset_small
```

```
##      Tijd PersistentieType Datahoeveelheid
## 1  1.81 Sharedpreferences      Weinig
## 2  1.35 Sharedpreferences      Weinig
## 3  1.84 Sharedpreferences      Weinig
## 4  1.54 Sharedpreferences      Weinig
## 5  1.81 Sharedpreferences      Weinig
## 6  1.82 Sharedpreferences      Weinig
## 7  1.79 Sharedpreferences      Weinig
## 8  1.57 Sharedpreferences      Weinig
## 9  1.78 Sharedpreferences      Weinig
## 10 1.79 Sharedpreferences      Weinig
## 11 1.97 Sharedpreferences      Weinig
## 12 1.60 Sharedpreferences      Weinig
## 13 1.77 Sharedpreferences      Weinig
## 14 2.23 Sharedpreferences      Weinig
## 15 1.79 Sharedpreferences      Weinig
## 16 1.77 Sharedpreferences      Weinig
## 17 1.32 Sharedpreferences      Weinig
## 18 1.76 Sharedpreferences      Weinig
## 19 2.00 Sharedpreferences      Weinig
```


## 20	1.30	Sharedpreferences	Weinig
## 21	1.77	Sharedpreferences	Weinig
## 22	1.35	Sharedpreferences	Weinig
## 23	1.56	Sharedpreferences	Weinig
## 24	1.79	Sharedpreferences	Weinig
## 25	1.31	Sharedpreferences	Weinig
## 26	2.23	Sharedpreferences	Weinig
## 27	1.83	Sharedpreferences	Weinig
## 28	1.09	Sharedpreferences	Weinig
## 29	1.57	Sharedpreferences	Weinig
## 30	1.10	Sharedpreferences	Weinig
## 31	1.61	GreenDAO	Weinig
## 32	1.78	GreenDAO	Weinig
## 33	2.00	GreenDAO	Weinig
## 34	2.47	GreenDAO	Weinig
## 35	2.44	GreenDAO	Weinig
## 36	2.06	GreenDAO	Weinig
## 37	2.21	GreenDAO	Weinig
## 38	1.75	GreenDAO	Weinig
## 39	1.58	GreenDAO	Weinig
## 40	1.60	GreenDAO	Weinig
## 41	1.77	GreenDAO	Weinig
## 42	2.00	GreenDAO	Weinig
## 43	1.86	GreenDAO	Weinig
## 44	1.77	GreenDAO	Weinig
## 45	2.20	GreenDAO	Weinig
## 46	1.75	GreenDAO	Weinig
## 47	1.58	GreenDAO	Weinig
## 48	1.53	GreenDAO	Weinig
## 49	2.25	GreenDAO	Weinig
## 50	2.06	GreenDAO	Weinig
## 51	1.37	GreenDAO	Weinig
## 52	2.46	GreenDAO	Weinig
## 53	1.80	GreenDAO	Weinig
## 54	1.83	GreenDAO	Weinig
## 55	2.67	GreenDAO	Weinig
## 56	2.02	GreenDAO	Weinig
## 57	1.98	GreenDAO	Weinig
## 58	1.34	GreenDAO	Weinig
## 59	1.76	GreenDAO	Weinig
## 60	1.31	GreenDAO	Weinig
## 61	1.57	SQLLite	Weinig
## 62	1.36	SQLLite	Weinig
## 63	1.31	SQLLite	Weinig
## 64	2.23	SQLLite	Weinig
## 65	2.43	SQLLite	Weinig
## 66	1.10	SQLLite	Weinig
## 67	1.78	SQLLite	Weinig
## 68	2.04	SQLLite	Weinig
## 69	1.81	SQLLite	Weinig
## 70	1.34	SQLLite	Weinig
## 71	2.06	SQLLite	Weinig
## 72	2.68	SQLLite	Weinig
## 73	1.58	SQLLite	Weinig

## 74	1.56	SQLite	Weinig
## 75	2.22	SQLite	Weinig
## 76	1.58	SQLite	Weinig
## 77	2.35	SQLite	Weinig
## 78	1.79	SQLite	Weinig
## 79	1.75	SQLite	Weinig
## 80	1.55	SQLite	Weinig
## 81	1.78	SQLite	Weinig
## 82	2.49	SQLite	Weinig
## 83	1.64	SQLite	Weinig
## 84	1.77	SQLite	Weinig
## 85	1.74	SQLite	Weinig
## 86	2.01	SQLite	Weinig
## 87	2.04	SQLite	Weinig
## 88	2.16	SQLite	Weinig
## 89	1.12	SQLite	Weinig
## 90	1.13	SQLite	Weinig
## 91	1.57	Realm	Weinig
## 92	1.77	Realm	Weinig
## 93	1.75	Realm	Weinig
## 94	1.78	Realm	Weinig
## 95	1.36	Realm	Weinig
## 96	2.17	Realm	Weinig
## 97	1.58	Realm	Weinig
## 98	1.59	Realm	Weinig
## 99	1.36	Realm	Weinig
## 100	1.55	Realm	Weinig
## 101	1.33	Realm	Weinig
## 102	1.96	Realm	Weinig
## 103	1.58	Realm	Weinig
## 104	1.59	Realm	Weinig
## 105	1.78	Realm	Weinig
## 106	1.31	Realm	Weinig
## 107	1.59	Realm	Weinig
## 108	2.25	Realm	Weinig
## 109	1.34	Realm	Weinig
## 110	2.27	Realm	Weinig
## 111	1.74	Realm	Weinig
## 112	1.31	Realm	Weinig
## 113	1.81	Realm	Weinig
## 114	1.13	Realm	Weinig
## 115	1.58	Realm	Weinig
## 116	1.14	Realm	Weinig
## 117	1.98	Realm	Weinig
## 118	1.33	Realm	Weinig
## 119	1.12	Realm	Weinig
## 120	1.35	Realm	Weinig

2. realm en greendao filteren

```
times_small_realm <- subset_small$Tijd[which(subset_medium$PersistentieType == "Realm")]
times_small_greendao <- subset_small$Tijd[which(subset_medium$PersistentieType == 'GreenDao')]

times_medium_realm
```

```
## [1] 6.32 3.79 6.53 7.21 10.34 4.01 6.47 6.43 6.30 6.36 4.07
## [12] 4.43 6.26 5.00 4.80 5.56 5.01 7.42 5.63 4.66 6.67 7.88
## [23] 4.89 4.87 5.75 5.97 5.38 6.11 5.52 4.90
```

```
times_medium_realm
```

```
## [1] 6.32 3.79 6.53 7.21 10.34 4.01 6.47 6.43 6.30 6.36 4.07
## [12] 4.43 6.26 5.00 4.80 5.56 5.01 7.42 5.63 4.66 6.67 7.88
## [23] 4.89 4.87 5.75 5.97 5.38 6.11 5.52 4.90
```

3. uitvoeren van de t test

```
t.test(times_small_realm, times_small_greendao, alternative = 'less')
```

```
##
## One Sample t-test
##
## data: times_small_realm
## t = 23.689, df = 29, p-value = 1
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##      -Inf 1.928038
## sample estimates:
## mean of x
##      1.799
```

oefening 5.7.

(oplossing vanuit de les)

dataset inladen:

```
library(readr)
```

```
puntenlijst <- read_csv("C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderzoekstechnieken/
```

```
## Warning: 28 parsing failures.
## row   col           expected actual
## 69 Groep value in level set      D 'C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderz
## 70 Groep value in level set      D 'C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderz
## 71 Groep value in level set      D 'C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderz
## 72 Groep value in level set      D 'C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderz
## 73 Groep value in level set      D 'C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderz
## ... ..
## See problems(...) for more details.
```

```
puntenlijst
```

```
## # A tibble: 203 x 2
##   Groep Score
##   <fct> <dbl>
## 1 A      12.5
## 2 A       7.5
## 3 A      NA
## 4 A      21
## 5 A      10
## 6 A     20.5
```

```
## 7 A      14
## 8 A      4
## 9 A      7
## 10 A     NA
## # ... with 193 more rows
```

we bekijken eerst de centrummaten en spreidingsmaten voor de ganse dataset

```
summarySet <- summary(puntenlijst)
summarySet
```

```
##      Groep      Score
## G      :29  Min.   : 0.00
## F      :28  1st Qu.:13.25
## H      :28  Median :19.00
## C      :26  Mean    :18.79
## B      :22  3rd Qu.:24.50
## (Other):42  Max.    :34.50
## NA's   :28  NA's    :8
```

```
stdevSummary <- sd(puntenlijst$Score, na.rm = TRUE)
stdevSummary
```

```
## [1] 7.014017
```

Nu bekijken we de centrummaten en spreidingsmaten voor de verschillende klassen

```
subsetA <- puntenlijst[which(puntenlijst$Groep == "A"),]
summaryA <- summary(subsetA)
summaryA
```

```
##      Groep      Score
## A      :20  Min.   : 4.00
## B      : 0  1st Qu.: 8.00
## C      : 0  Median :12.50
## d      : 0  Mean    :13.12
## E      : 0  3rd Qu.:17.00
## F      : 0  Max.    :26.50
## (Other): 0  NA's    :3
```

```
subsetB <- puntenlijst[which(puntenlijst$Groep == "B"),]
summaryB <- summary(subsetB)
summaryB
```

```
##      Groep      Score
## B      :22  Min.   : 1.00
## A      : 0  1st Qu.:10.00
## C      : 0  Median :14.50
## d      : 0  Mean    :17.21
## E      : 0  3rd Qu.:26.50
## F      : 0  Max.    :31.50
## (Other): 0  NA's    :1
```

```
subsetC <- puntenlijst[which(puntenlijst$Groep == "C"),]
summaryC <- summary(subsetC)
summaryC
```

```
##      Groep      Score
## C      :26  Min.    :10.0
```

```
## A      : 0  1st Qu.:13.0
## B      : 0  Median :18.5
## d      : 0  Mean   :18.8
## E      : 0  3rd Qu.:22.0
## F      : 0  Max.   :34.5
## (Other): 0  NA's   :1
```

```
subsetD <- puntenlijst[which(puntenlijst$Groep == "D"),]
summaryD <- summary(subsetD)
summaryD
```

```
##      Groep      Score
## A      :0  Min.    : NA
## B      :0  1st Qu.: NA
## C      :0  Median : NA
## d      :0  Mean    :NaN
## E      :0  3rd Qu.: NA
## F      :0  Max.    : NA
## (Other):0
```

```
subsetE <- puntenlijst[which(puntenlijst$Groep == "E"),]
summaryE <- summary(subsetE)
summaryE
```

```
##      Groep      Score
## E      :22  Min.    : 0.00
## A      : 0  1st Qu.:16.38
## B      : 0  Median :19.25
## C      : 0  Mean    :18.85
## d      : 0  3rd Qu.:21.38
## F      : 0  Max.    :30.75
## (Other): 0
```

```
subsetF <- puntenlijst[which(puntenlijst$Groep == "F"),]
summaryF <- summary(subsetF)
summaryF
```

```
##      Groep      Score
## F      :28  Min.    : 6.50
## A      : 0  1st Qu.:12.88
## B      : 0  Median :17.50
## C      : 0  Mean    :17.81
## d      : 0  3rd Qu.:22.00
## E      : 0  Max.    :32.00
## (Other): 0  NA's    :1
```

```
subsetG <- puntenlijst[which(puntenlijst$Groep == "G"),]
summaryG <- summary(subsetG)
summaryG
```

```
##      Groep      Score
## G      :29  Min.    :10.00
## A      : 0  1st Qu.:16.50
## B      : 0  Median :18.75
## C      : 0  Mean    :18.74
## d      : 0  3rd Qu.:23.00
## E      : 0  Max.    :27.75
```

```
## (Other): 0
```

```
subsetH <- puntenlijst[which(puntenlijst$Groep == "H"),]  
summaryH <- summary(subsetH)  
summaryH
```

```
##      Groep      Score  
## H      :28   Min.    : 3.00  
## A      : 0   1st Qu.:15.75  
## B      : 0   Median :22.00  
## C      : 0   Mean    :20.95  
## d      : 0   3rd Qu.:26.62  
## E      : 0   Max.    :34.50  
## (Other): 0
```