Computerzitting 0: Beschrijvende Statistiek in R
2022-2023

1 Probleemomschrijving

We starten met het verkennen van een dataset aan de hand van het statistische computerprogramma **R**. We zullen hierbij de dataset *resultaten* bekijken die de volgende computerzittingen geregeld zal opduiken. Deze dataset bevat resultaten over studenten Scheikunde, Biologie en Geografie van een aantal jaren geleden. Meer specifiek gaat het om de variabelen

lengte	:	lengte van student(e) in cm
geslacht	:	Man (M) of Vrouw (V)
middelbaar	:	studieresultaten in het laatste jaar middelbaar (in procenten)
bachelor	:	studieresultaten in de eerste bachelor (in procenten)
studierichting	:	Scheikunde (S), Biologie (B) of Geografie (G)
kleur	:	kleur van de wagen waarmee de student(e) het meeste rijdt: Licht (L), Donker (D) of Rood (R)

Het onderwijzend personeel stelt zich hierbij enkele vragen die we in de loop van de volgende computerzittingen zullen oplossen.

1. Geven de resultaten van het middelbaar een goede indicatie voor de slaagkansen in de eerste bachelor? Dit zal worden behandeld in Probleem 1.

2. Zijn jongens echt veel groter dan meisjes? Dat onderzoeken we in Probleem 2 en Probleem 3.

3. Zijn de resultaten die in de eerste bachelor werden behaald systematisch lager dan de resultaten die in het laatste jaar middelbaar werden gehaald? Gaat dit resultaat op voor elke groep studenten? Hier gaan we dieper op in in Probleem 4.

4. Kiezen vrouwelijke studenten een andere kleur voor hun wagen dan mannelijke studenten?

5. Verschilt de keuze van de studierichting bij mannen en vrouwen?

2 Downloaden van de R (en RStudio) software

Doorheen de computerzittingen gebruiken we het software programma **R**. Een gratis versie hiervan kan je downloaden via de website *http://cran.r-project.org/bin/windows/base/*.

Zie ook: Help > Manuals (in PDF) > An Introduction to **R**.

Download RStudio via de website *https://rstudio.com/products/rstudio/download/*. RStudio is een populaire en handige interface voor **R**.

3 Openen van data in R

Vooraleer we met enkele basisbegrippen starten, bekijken we eerst hoe we de dataset vanuit verschillende bronnen kunnen importeren in **R** zodat we deze kunnen onderzoeken.

3.1 Verschillende types data

Datasets kunnen in verschillende structuren opgeslagen worden. We bekijken enkel datasets die afkomstig zijn uit Excel (bestandsnaam eindigt op .xls) zoals de dataset *resultaten.xls* en datasets die afkomstig zijn uit een tekstbestand (bestandsnaam eindigt op .txt) zoals *resultaten.txt*. Omdat het openen van datasets uit verschillende programma's iets anders verloopt, bekijken we deze verschillende manieren apart. Bewaar eerst de twee datasets. De datasets zijn terug te vinden op Blackboard onder Opdrachten, in de map Data.

3.2 Werkbladen (Worksheets) uit Excel

Merk op dat je in **R** niet zomaar Excel bestanden kan inlezen. Voor de eenvoud en om problemen te vermijden zullen we daarom .csv bestanden inlezen. Met andere woorden, we moeten eerst ons Excel bestand omzetten naar een .csv bestand. Dit doe je als volgt:

1. Open je Excel bestand (*resultaten.xls*)
2. Ga naar het werkblad waar de data staan (meestal is dit het eerste werkblad).
3. Klik op Bestand > Opslaan als (File > Save as), selecteer onder Opslaan als (Save as) het bestandstype CSV en klik op Opslaan (Save as).
4. Negeer de twee waarschuwingen die je krijgt, m.a.w. klik telkens op Ja (Yes).

Als alles goed verlopen is, is er nu een .csv bestand (*resultaten.csv*) aangemaakt in de directory waar je Excel bestand *resultaten.xls* zich bevindt. Dit .csv bestand bevat enkel de data van het werkblad (uit je Excel bestand) dat je geselecteerd had. Indien er verschillende datasets zijn op verschillende werkbladen, moet je voor elk werkblad een .csv bestand maken.

Tenslotte, vooraleer we de data inlezen in **R**, open je best eens je Excel of .csv bestand zodat je de structuur van de data (i.e. de namen van de variabelen e.d.) kent. Bij de dataset *resultaten.xls* merken we op dat de dataset zich in het eerste werkblad bevindt, dat de namen

van de variabelen in de eerste rij staan (**header = TRUE**) en dat er een ","gebruikt wordt als decimaalteken (**dec = ","**).

Start nu **R** op en ga als volgt te werk (druk na elk commando op < ENTER >):

1. Lees het .csv bestand in via het commando

   ```
   gegevens=read.csv(file=file.choose(),header=TRUE,dec=",",sep=";")
   ```

   Dit zal een venster openen waarin je naar je .csv bestand kan browsen. Selecteer de gewenste file en klik op Openen.

2. Typ **help(read.csv)** voor meer informatie omtrent de functie read.csv.

3. Wanneer je code uit deze werkblaadjes gaat copy-pasten in het codescherm van **R** zal het programma een foutmelding geven. Typ daarom voor de veiligheid voorbeeldcode uit de blaadjes altijd over in het codescherm.

Je kan de inhoud van een variabele altijd weergeven door de naam van de variabele te typen. Indien je bv. wil weten wat de inhoud van de variabele *gegevens* is, dan typ je **gegevens** (na het > teken uiteraard).

3.3 Tekstbestanden

Vervolgens openen we het tekstbestand *resultaten.txt* in **R**. Dit doe je als volgt:

1. Open je .txt file en ga na

   (a) of de eerste lijn bestaat uit namen van de variabelen (kolommen) (**header = TRUE**); of niet (meteen gegevens) (**header = FALSE**);

   (b) hoe de verschillende kolommen gescheiden zijn (i) witruimte - spaties, tabs (**sep = " "**), (ii) een komma (**sep = ","**), (iii) een puntkomma (**sep = ";"**), ...

2. Keer terug naar **R**.

3. Typ

   ```
   gegevens = read.table(file=file.choose(),header=TRUE)
   ```

   Dit zal een venster openen waarin je naar je .txt file kan browsen. Selecteer de gewenste file en klik op Open.

4. Typ **help(read.table)** voor meer informatie omtrent de functie read.table.

3.4 Eigen dataset intoetsen

Tenslotte moet je ook in staat zijn om je eigen dataset in te geven. Dit gebeurt door het uitvoeren van verschillende commando's. Die code kan je typen in een codescherm (command window) of in een script.

In het laatste geval open je eerste een script via File > New script en typ je vervolgens je code (commando's) onder elkaar in dit script. Je kan een specifiek commando (lijn) of een blok code uitvoeren door het te selecteren en vervolgens op je rechtermuisknop te klikken en Run line or selection te kiezen. Het voordeel van een script is dat je het kan opslaan, zodat je later niet al je code opnieuw hoeft te typen.

Oefening: Maak volgende dataset :

Var1 = kleur	Var2 = geslacht	Var3 = aantal
D	M	9
D	V	14
L	M	12
L	V	11
R	M	3
R	V	7

Dit doe je door het volgende in te typen :

```
kleur = c('D','D','L','L','R','R')
geslacht = c('M','V','M','V','M','V')
aantal = c(9,14,12,11,3,7)
gegevens = data.frame(kleur,geslacht,aantal)
gegevens
```

kleur, geslacht, aantal en gegevens zijn de namen van de variabelen waarin de gegevens bewaard worden. Je kan deze gerust anders noemen, bv.

```
x = c('D','D','L','L','R','R')
y = c('M','V','M','V','M','V')
z = c(9,14,12,11,3,7)
d = data.frame(x,y,z)
d
```

Let op: gebruik nooit de letters c en t voor namen van variabelen, vermits dit tevens namen van functies zijn en er hierdoor conflicten kunnen ontstaan.

3.5 Bewaren van workspaces en variabelen

Om te vermijden dat je datasets of hele stukken code steeds weer moet ingeven, kan je deze opslaan en later terug oproepen. Je kan een of meerdere specifieke variabelen (bv. de variabele kleur en de dataset) opslaan via het commando

```
save(kleur,gegevens,file="naam.Rdata")
```

Wanneer je alle variabelen, datasets, ... die je hebt aangemaakt, wil opslaan, dan typ je **save.image()**, of klik je File > Save Workspace en browse naar je favoriete locatie. Analoog kan je opgeslapen workspaces terugladen via File > Open Workspace.

Wanneer je de commando's **save(x,gegevens,file="naam.Rdata")** en **save.image()** gebruikt, zullen de bestanden worden opgeslagen in de directory waarin je op dat moment aan het werken bent. Om zeker te zijn dat je achteraf je bestanden nog terugvindt, is het daarom goed om de actieve directory te veranderen naar je favoriete locatie. Dit kan je doen via File > Change dir. Zie **help(save)** voor meer informatie.

4 Enkele resultaten

We zijn nu in staat om al enkele resultaten te bekomen die ons kunnen helpen bij het oplossen van de vragen gesteld aan het begin van deze computerzitting. We kunnen deze vragen op dit moment echter enkel intuïtief oplossen. Formele tests zullen gezien worden in de volgende oefenzittingen en computerzittingen.

4.1 Probleem 1: Verband tussen twee continue variabelen nagaan

Om het verband tussen twee **continue** variabelen te bestuderen op een grafische manier, kunnen we een *scatterplot* opstellen.

We willen onderzoeken hoe de variabele *middelbaar* en de variabele *bachelor* elkaar beïnvloeden. Aangezien we voor 56 studenten zowel het resultaat uit het middelbaar ($= x$), als het resultaat uit de bachelors ($= y$) kennen, kunnen we per i-de student het punt (x_i, y_i) uitzetten in een grafiek. Dit noemen we een *scatterplot*.

Ga na hoe de variabelen *middelbaar* en *bachelor* elkaar beïnvloeden op een grafische manier. (Later zien we een meer formele test (zie correlatie en regressie)!)

Oplossing:

- Maak een scatterplot op de volgende manier:

  1. Laad de file resultaten.txt via

     ```
     gegevens = read.table(file=file.choose(),header=TRUE)
     ```

  2. Zet de score in de 1ste bachelor (Y-as) uit als functie van de score in het middelbaar (X-as) met de code

     ```
     plot(gegevens$middelbaar,gegevens$bachelor,type="p",
     xlab="Score middelbaar",ylab="Score 1e bachelor",
     main="Score 1e bachelor = f(Score middelbaar)")
     ```

Druk < ENTER >. Het commando gegevens$ <NAAM> maakt **R** duidelijk welke kolom uit de dataset gegevens hij moet gebruiken, nl. de kolom met als naam < NAAM >.
De eerste parameter van het commando plot (gegevens$middelbaar) wordt uitgezet op de X-as en de tweede (gegevens$bachelor) op de Y-as. De volgorde van de overige parameters is onbelangrijk. **type** specifieert het soort plot (punten, lijnen, ...), **xlab** geeft een naam aan de X-as, **ylab** geeft een naam aan de Y-as en **main** geeft een titel aan de grafiek. Zie **help(plot)** voor meer informatie over het commando plot.

- Wat besluit je nu? Hoe beïnvloedt de variabele *middelbaar* de variabele *bachelor*?

.....

4.2 Probleem 2: Centrumkenmerken

Ga na of jongens groter zijn dan meisjes door de mediaan en het gemiddelde te berekenen van de variabele *lengte*. Merk op dat je de gemiddelde lengte van de jongens en meisjes apart moet berekenen. (Later zien we een formele test (hypothesetoetsen)!)

Rangschik mediaan en gemiddelde van klein naar groot. Is deze volgorde steeds dezelfde? Verklaar.

Oplossing:

- We beginnen met 2 nieuwe variabelen (lengteM en lengteV) te maken, die respectievelijk de lengtes van alle jongens en de lengtes van alle meisjes uit de dataset bevatten. (Er wordt verondersteld dat de dataset reeds is ingelezen.)

```
lengteM = gegevens$lengte[gegevens$geslacht=="M"]
lengteV = gegevens$lengte[gegevens$geslacht=="V"]
```

- Vervolgens berekenen we van elke variabele het gemiddelde en de mediaan:

```
mu_M = mean(lengteM)
mu_M
med_M = median(lengteM)
med_M


mu_V = mean(lengteV)
mu_V
med_V = median(lengteV)
med_V
```

- Vul aan voor vrouwen:

gemiddelde	=	.....
mediaan	=	.....

- Vul aan voor mannen:

gemiddelde	=	.....
mediaan	=	.....

- Vervolgens bekijken we de volgorde van mediaan en gemiddelde. Vul aan (kies uit mediaan, gemiddelde):

  Bij de vrouwelijke studenten geldt:

  $$\ldots\ldots\ldots\ldots \leq \ldots\ldots\ldots\ldots\ldots\ldots$$

  Bij de mannelijke studenten geldt:

  $$\ldots\ldots\ldots\ldots \leq \ldots\ldots\ldots\ldots\ldots\ldots$$

  Om deze volgorde te verduidelijken, bekijken we het histogram van *lengte* voor de mannelijke en vrouwelijke studenten apart. Voor een histogram gebruik je de volgende code

```
par(mfcol=c(1,2))
h_M = hist(lengteM)
h_V = hist(lengteV)
```

  De eerste lijn maakt **R** duidelijk dat de figuur moet worden opgedeeld in een matrix met 1 rij en 2 kolommen, die daarna worden opgevuld met de 2 histogrammen (naast elkaar). Wat zou er gebeuren indien je par(mfcol=c(2,1)) i.p.v. par(mfcol=c(1,2)) zou gebruiken?

4.3 Probleem 3: Spreidingskenmerken

Bereken de variantie, het maximum en het minimum, de range (maximum - minimum), IQR (Inter Quartile Range = verschil tussen 75% en 25% kwantiel) en de standaarddeviatie voor de variabele lengte in het algemeen.

Zijn er eventuele foutieve metingen gebeurd, of zijn er studenten die extreem groot of klein zijn in de vergelijking met de rest?

Oplossing:

- Maak eerst een variabele (lengte) die de lengtes van alle studenten (jongens en meisjes) bevat:

```
lengte = gegevens$lengte
lengte
```

- Bereken de gevraagde spreidingskenmerken:

```
var_l = var(lengte)
var_l
min_max_l=range(lengte)
min_max_l
range_l=min_max_l [2] - min_max_l [1]
range_l
iqr_l = IQR(lengte)
iqr_l
sd_l=sd(lengte)
sd_l
```

lengte		
variantie	=	.....
maximum	=	.....
minimum	=	.....
range	=	.....
IQR	=	.....
standaarddeviatie	=	.....

- Je kan de berekende numerieke waarden ook samenvatten in een **boxplot**. Hiervoor kan je de volgende codes gebruiken.

  1. Een boxplot voor de variabele lengte, waarbij je geen onderscheid maakt tussen meisjes en jongens:

```
boxplot(lengte,ylab="lengte (in cm)",
main="Boxplot van de variabele lengte")
```

  2. Een boxplot voor de lengtes van de jongens en de meisjes apart (we veronderstellen dat de variabelen lengteM en lengteV reeds gecreëerd zijn):

```
boxplot(list(lengteM, lengteV),ylab="lengte (in cm)",
names = c("jongens", "meisjes"),
main="Boxplot van de lengte van de jongens en de meisjes apart")
```

- Zijn er vreemde metingen (ook wel uitschieters genoemd)? Deze kan je ook opsporen a.h.v. een boxplot.

  .....

  .....

4.4 Probleem 4: Vergelijken van twee of meerdere groepen

1. Ga intuïtief na of de punten die behaald zijn in de bachelors lager liggen dan de punten behaald in het middelbaar.
2. Vergelijk ook de punten behaald in de bachelors van de verschillende studierichtingen.

   (Later zien we een formele test (hypothesetoetsen)!)

Oplossing:

1. Vergelijken van 'middelbaar' met 'bachelor': gepaarde gegevens

   - Vooraleer we dit probleem oplossen, moeten we eerst een belangrijke eigenschap opmerken met betrekking tot de variabelen. We beschikken hier namelijk voor elk student over 2 gegevens ('middelbaar' en 'bachelor') die we met elkaar gaan vergelijken. In zo een situatie spreken we over **gepaarde** gegevens. In plaats van naar de gegevens zelf te gaan kijken kan je dan even goed naar het verschil tussen beide gegevens gaan kijken.
   - Creëer een nieuwe variabele 'verschil', die het verschil geeft tussen de score behaald in het middelbaar en de score behaald in de eerste bachelor. Dit kan met behulp van de volgende code:

     ```
     verschil=gegevens$middelbaar-gegevens$bachelor;
     ```

   - Bereken gemiddelde en mediaan van deze nieuwe variabele. Wat besluit je?

     .....

     .....

   - Bekijk de boxplot van de variabele 'verschil'. Merk op dat deze heel symmetrisch is rond de mediaan.

2. Vergelijken van 'bachelor' per studierichting: ongepaarde gegevens

   - In tegenstelling tot het vorige geval beschikken we nu niet meer over 2 waarnemingen per object (student). De gegevens die we met elkaar moeten vergelijken noemen we **ongepaard**.

- Bereken het gemiddelde en de mediaan van de variabele ‘bachelor’.
- Maak ook een boxplot van deze variabele per studierichting.
- Bespreek kort de resultaten.

.....

.....

.....

.....