

title: “Oefeningen hoofdstuk 3 - analyse op 1 variabele” author: “Tijds Martens” date: “6 maart 2019” output: pdf_document: default html_document: df_print: paged —

oefening 3.1.

(oefening zelf gemaakt, geen oplossing)

opgave

wat is de gemiddelde lengte van de superhelden

lengtes van de helden:

- $x_1 = 141$
- $x_2 = 198$
- $x_3 = 143$
- $x_4 = 201$
- $x_5 = 184$

oplossing

zie cursus blad 1

oplossing is 173.4

oefening 3.2.

opgave

Het gemiddelde van 15 cijfers is 12. Welk nummer moeten we aan de rij van cijfers toevoegen om een gemiddelde van 13 te bekomen?

oplossing

zie cursus blad 1

oplossing is 28

oefening 3.3.

opgave

Met welke voorgaande statistiek komt Q2 overeen

oplossing

Mediaan

Als de mediaan een lijst in 2 delen splitst kan de mediaan van de eerste lijst gezien worden als Q1

Als de mediaan een lijst in 2 delen splitst kan de mediaan van de tweede lijst gezien worden als Q3

oefening 3.4.

(oefening zelf gemaakt, geen oplossing)

opgave

De formules voor gemiddelde μ en variantie σ^2 staan beschreven in secties 3.2 en 3.7, resp. Hoe moeten deze formules aangepast worden om μ en σ^2 te berekenen wanneer we te maken hebben met een frequentietabel? Doe dit voor de data in tabel 3.3.

oplossing

orginele formules:

$$\mu = 1/n * \sum_i^n x_i$$

σ^2 = de som van alle verschillen met het gemiddelde gedeeld door het aantal voorkomens

zie cursus blad 2

oplossing chamilo:

de mogelijke scores

```
x <- 0:10
x
```

```
## [1] 0 1 2 3 4 5 6 7 8 9 10
```

het aantal voorkomens van elke score:

```
f_x <- c(2,1,2,0,2,4,9,11,13,8,8)
f_x
```

```
## [1] 2 1 2 0 2 4 9 11 13 8 8
```

gemiddelde:

```
m <- sum(x*f_x)/sum(f_x)
m
```

```
## [1] 7
```

```
v <- sum(f_x * (x - m) ^ 2) / sum(f_x)
v
```

```
## [1] 5.733333
```

standaardafwijking:

```
stdev <- sqrt(v)
stdev
```

```
## [1] 2.394438
```

oefening 3.5

we gaan na als volgende formules een goed alternatief zijn voor die van de variantie

- $\sum_i^n (x_i - \mu)$
- $\sum_i^n |x_i - \mu|$
- $\sum_i^n (x_i - \mu)^2$

allocatie van de sets

```
x <- c(4,4,-4,-4)
y <- c(7,1,-6,-2)
```

formule 1

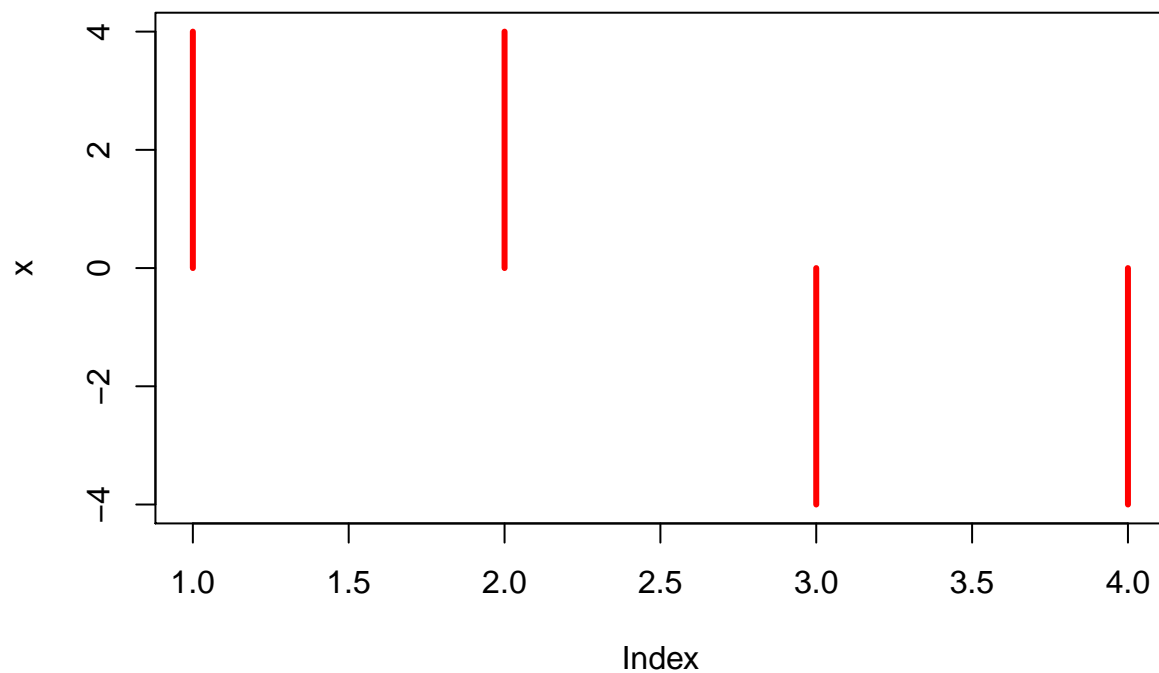
```
meanX <- mean(x)
meanY <- mean(y)
var_1x <- x - meanX
var_1y <- y - meanY
var_1x
```

```
## [1]  4  4 -4 -4
```

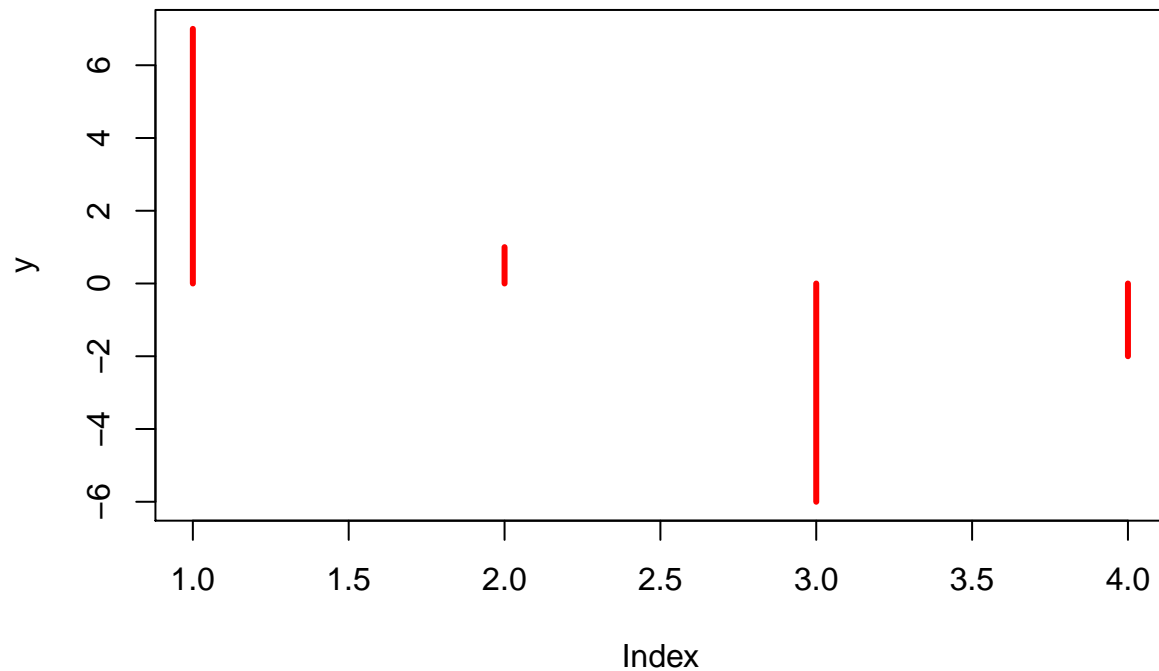
```
var_1y
```

```
## [1]  7  1 -6 -2
```

```
som_1x <- 1/length(x) * sum(var_1x)
som_1y <- 1/length(y) * sum(var_1y)
plot(x,type="h", col='red', lwd=3)
```



```
plot(y,type="h", col='red', lwd=3)
```



Conclusie

we zien dat voor beide datasets een zelfde waarde bepaald wordt, alhowel de grafieken duidelijk een andere spreiding vertonen . Laten we eens onderzoeken of we it kunnen oplossen door de absolute waarde te gaan bepael van de sommatie

formule 2

```
var_2x <- abs(x - meanX)
var_2x
```

```
## [1] 4 4 4 4
```

```
var_2y <- abs(y - meanY)
var_2y
```

```
## [1] 7 1 6 2
```

```
sum_2x <- 1/length(x) * sum(var_2x)
sum_2y <- 1/length(y) * sum(var_2y)
sum_2x
```

```
## [1] 4
```

```
sum_2y
```

```
## [1] 4
```

conclusie

Het probleem hierbij dat we voor beide datasets, hoewel deze duidelijk visueel een verschillende spreiding hebben, dezelfde waarde bepaald wordt. Blijkbaar worden kleine waarden evenwaardig beschouwd als grote waarden (deviaties). Dit wordt natuurlijk opgelost door kwadraat te nemen

formule 3

```
var_3x <- var(x)
var_3y <- var(y)
var_3x
```

```
## [1] 21.33333
```

```
var_3y
```

```
## [1] 30
```

oefening 3.6.

(oefening zelf gemaakt, geen oplossing)

opgave

Zoek eens zelfstandig op wat de variatiecoëfficiënt is, Hoe wordt die gedefinieerd voor een volledige populatie en wat zou je ermee kunnen doen.

oplossing

wat: Het is een relatieve spreidingsmaat. De spreiding wordt gemeten ten opzichte van het gemiddelde.

variatioecoëfficiënt voor de populatie: $C_v = \frac{\sigma}{|\mu|}$

variatioecoëfficiënt voor de populatie $C_v = \frac{\sigma}{a}$

De variatiecoëfficiënt kan gebruikt worden om eenvoudig standaardafwijkingen van datasets met elkaar te vergelijken.

oefening 3.7.

(oefening zelf gemaakt, geen oplossing)

opgave

Beschouw de volgende datasets uit het data frame “ais” (uit de library DAAG)

1. Ontleed de gegevens voor de roeiers
2. ontleed de gegevens voor de roeiers, netballers en de tennissers
3. ontleed de gegevens voor de vrouwelijke basketballers en roeiers

oplossing

0. installeren van de benodigde packages

```
#install.packages("DAAG")
#install.packages("lattice")
```

1. Ontleed de gegevens voor de roeiers.

```
roeiers <- subset(DAAG::ais,sport=="Row")
roeiers
```

```
##      rcc  wcc  hc  hg ferr  bmi  ssf pcBfat  lbm  ht  wt sex sport
## 14  4.26  6.2 41.0 13.9  48 25.44 90.2  17.71 66.24 177.9 80.5  f   Row
## 15  4.63  6.0 43.7 14.7  30 22.63 97.2  18.77 57.92 177.5 71.3  f   Row
## 16  4.36  5.8 40.3 13.3  29 21.86 99.9  19.83 56.52 179.6 70.5  f   Row
## 17  3.91  7.3 37.6 12.9  43 22.27 125.9 25.16 54.78 181.3 73.2  f   Row
## 18  4.51  8.3 43.7 14.7  34 21.27 69.9  18.04 56.31 179.7 68.7  f   Row
## 19  4.37  8.1 41.8 14.3  53 23.47 98.0  21.79 62.96 185.2 80.5  f   Row
## 20  4.90  6.9 44.0 14.5  59 23.19 96.8  22.25 56.68 177.3 72.9  f   Row
## 21  4.46  5.7 39.2 13.0  43 23.17 80.3  16.25 62.39 179.3 74.5  f   Row
## 22  3.95  3.3 36.9 12.5  40 24.54 74.9  16.38 63.05 175.3 75.4  f   Row
## 23  4.46  9.5 41.5 14.5  92 22.96 83.0  19.35 56.05 174.0 69.5  f   Row
## 24  5.02  6.4 44.8 15.2  48 19.76 91.0  19.20 53.65 183.3 66.4  f   Row
## 25  4.26  5.8 41.2 14.1  77 23.36 76.2  17.89 65.45 184.7 79.7  f   Row
## 26  4.46  5.6 41.1 14.3  71 22.67 52.6  12.20 64.62 180.2 73.6  f   Row
## 27  4.16  5.8 39.8 13.3  37 24.24 111.1 23.70 60.05 180.2 78.7  f   Row
## 28  4.49  7.6 41.8 14.4  71 24.21 110.7 24.69 56.48 176.0 75.0  f   Row
## 29  4.21  7.5 38.4 13.2  73 20.46 74.7  16.58 41.54 156.0 49.8  f   Row
## 30  4.57  6.6 42.8 14.5  85 20.81 113.5 21.47 52.78 179.7 67.2  f   Row
## 31  4.87  6.4 44.8 15.0  64 20.17 99.8  20.12 52.72 180.9 66.0  f   Row
## 32  4.44 10.1 42.7 14.0  19 23.06 80.3  17.51 61.29 179.5 74.3  f   Row
## 33  4.45  6.6 42.6 14.1  39 24.40 109.5 23.70 59.59 178.9 78.1  f   Row
## 34  4.41  5.9 41.1 13.5  41 23.97 123.6 22.39 61.70 182.1 79.5  f   Row
## 35  4.87  7.3 44.1 14.8  13 22.62 91.2  20.43 62.46 186.3 78.5  f   Row
## 114 4.87  8.2 43.8 15.0 130 23.57 49.2   9.00 78.00 190.7 85.7  m   Row
## 115 5.04  7.1 44.0 14.8  64 25.84 61.8  12.61 75.00 181.8 85.4  m   Row
## 116 4.40  5.3 42.5 14.5 109 24.06 46.5   9.03 78.00 188.3 85.3  m   Row
## 117 4.95  5.9 45.4 15.5 125 23.85 34.8   6.96 87.00 198.0 93.5  m   Row
## 118 4.78  9.3 43.0 14.7 150 25.09 60.2  10.05 78.00 186.0 86.8  m   Row
## 119 5.21  6.8 44.5 15.4 115 23.84 48.1   9.56 79.00 192.0 87.9  m   Row
## 120 5.22  8.4 47.5 16.2  89 25.31 44.5   9.36 79.00 185.6 87.2  m   Row
## 121 5.18  6.5 45.4 14.9  93 19.69 54.0  10.81 48.00 165.3 53.8  m   Row
## 122 5.40  6.8 49.5 17.3 183 26.07 44.7   8.61 82.00 185.6 89.8  m   Row
## 123 4.92  5.4 46.2 15.8  84 25.50 64.9   9.53 82.00 189.0 91.1  m   Row
## 124 5.24  7.5 46.5 15.5  70 23.69 43.8   7.42 82.00 193.4 88.6  m   Row
## 125 5.09 10.1 44.9 14.8 118 26.79 58.3   9.79 83.00 185.6 92.3  m   Row
## 126 4.83  5.0 43.8 15.1  61 25.61 52.8   8.97 88.00 194.6 97.0  m   Row
## 127 5.22  6.0 46.6 15.7  72 25.06 43.1   7.49 83.00 189.0 89.5  m   Row
## 128 4.71  8.0 45.5 15.6  91 24.93 78.0  11.95 78.00 188.1 88.2  m   Row
```

```
myvars <- c("ht")
newdata <- roeiers[myvars]
summary(newdata)
```

```
##      ht
##  Min.   :156.0
##  1st Qu.:179.3
##  Median :181.8
```

```
## Mean :182.4
## 3rd Qu.:186.3
## Max. :198.0
```

2. ontleed de gegevens van de roeiers, de netballers en de tennisseres

```
roeinetball <- subset(DAAG::ais,sport=="Row" | sport=="Netball" | sport=="Tennis")
roeinetball
```

##	rcc	wcc	hc	hg	ferr	bmi	ssf	pcBfat	lbm	ht	wt	sex
## 14	4.26	6.2	41.0	13.9	48	25.44	90.2	17.71	66.24	177.9	80.5	f
## 15	4.63	6.0	43.7	14.7	30	22.63	97.2	18.77	57.92	177.5	71.3	f
## 16	4.36	5.8	40.3	13.3	29	21.86	99.9	19.83	56.52	179.6	70.5	f
## 17	3.91	7.3	37.6	12.9	43	22.27	125.9	25.16	54.78	181.3	73.2	f
## 18	4.51	8.3	43.7	14.7	34	21.27	69.9	18.04	56.31	179.7	68.7	f
## 19	4.37	8.1	41.8	14.3	53	23.47	98.0	21.79	62.96	185.2	80.5	f
## 20	4.90	6.9	44.0	14.5	59	23.19	96.8	22.25	56.68	177.3	72.9	f
## 21	4.46	5.7	39.2	13.0	43	23.17	80.3	16.25	62.39	179.3	74.5	f
## 22	3.95	3.3	36.9	12.5	40	24.54	74.9	16.38	63.05	175.3	75.4	f
## 23	4.46	9.5	41.5	14.5	92	22.96	83.0	19.35	56.05	174.0	69.5	f
## 24	5.02	6.4	44.8	15.2	48	19.76	91.0	19.20	53.65	183.3	66.4	f
## 25	4.26	5.8	41.2	14.1	77	23.36	76.2	17.89	65.45	184.7	79.7	f
## 26	4.46	5.6	41.1	14.3	71	22.67	52.6	12.20	64.62	180.2	73.6	f
## 27	4.16	5.8	39.8	13.3	37	24.24	111.1	23.70	60.05	180.2	78.7	f
## 28	4.49	7.6	41.8	14.4	71	24.21	110.7	24.69	56.48	176.0	75.0	f
## 29	4.21	7.5	38.4	13.2	73	20.46	74.7	16.58	41.54	156.0	49.8	f
## 30	4.57	6.6	42.8	14.5	85	20.81	113.5	21.47	52.78	179.7	67.2	f
## 31	4.87	6.4	44.8	15.0	64	20.17	99.8	20.12	52.72	180.9	66.0	f
## 32	4.44	10.1	42.7	14.0	19	23.06	80.3	17.51	61.29	179.5	74.3	f
## 33	4.45	6.6	42.6	14.1	39	24.40	109.5	23.70	59.59	178.9	78.1	f
## 34	4.41	5.9	41.1	13.5	41	23.97	123.6	22.39	61.70	182.1	79.5	f
## 35	4.87	7.3	44.1	14.8	13	22.62	91.2	20.43	62.46	186.3	78.5	f
## 36	4.56	13.3	42.2	13.6	20	19.16	49.0	11.29	53.14	176.8	59.9	f
## 37	4.15	6.0	38.0	12.7	59	21.15	110.2	25.26	47.09	172.6	63.0	f
## 38	4.16	7.6	37.5	12.3	22	21.40	89.0	19.39	53.44	176.0	66.3	f
## 39	4.32	6.4	37.7	12.3	30	21.03	98.3	19.63	48.78	169.9	60.7	f
## 40	4.06	5.8	38.7	12.8	78	21.77	122.1	23.11	56.05	183.0	72.9	f
## 41	4.12	6.1	36.6	11.8	21	21.38	90.4	16.86	56.45	178.2	67.9	f
## 42	4.17	5.0	37.4	12.7	109	21.47	106.9	21.32	53.11	177.3	67.5	f
## 43	3.80	6.6	36.5	12.4	102	24.45	156.6	26.57	54.41	174.1	74.1	f
## 44	3.96	5.5	36.3	12.4	71	22.63	101.1	17.93	55.97	173.6	68.2	f
## 45	4.44	9.7	41.4	14.1	64	22.80	126.4	24.97	51.62	173.7	68.8	f
## 46	4.27	10.6	37.7	12.5	68	23.58	114.0	22.62	58.27	178.7	75.3	f
## 47	3.90	6.3	35.9	12.1	78	20.06	70.0	15.01	57.28	183.3	67.4	f
## 48	4.02	9.1	37.7	12.7	107	23.01	77.0	18.14	57.30	174.4	70.0	f
## 49	4.39	9.6	38.3	12.5	39	24.64	148.9	26.78	54.18	173.3	74.0	f
## 50	4.52	5.1	38.8	13.1	58	18.26	80.1	17.22	42.96	168.6	51.9	f
## 51	4.25	10.7	39.5	13.2	127	24.47	156.6	26.50	54.46	174.0	74.1	f
## 52	4.46	10.9	39.7	13.7	102	23.99	115.9	23.01	57.20	176.0	74.3	f
## 53	4.40	9.3	40.4	13.6	86	26.24	181.7	30.10	54.38	172.2	77.8	f
## 54	4.83	8.4	41.8	13.4	40	20.04	71.6	13.93	57.58	182.7	66.9	f
## 55	4.23	6.9	38.3	12.6	50	25.72	143.5	26.65	61.46	180.5	83.8	f
## 56	4.24	8.4	37.6	12.5	58	25.64	200.8	35.52	53.46	179.8	82.9	f
## 57	3.95	6.6	38.4	12.8	33	19.87	68.9	15.59	54.11	179.6	64.1	f
## 58	4.03	8.5	37.7	13.0	51	23.35	103.6	19.61	55.35	171.7	68.8	f

##	90	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86	56.58	167.9	71.5	f
##	91	4.40	4.0	40.8	13.9	73	22.12	98.1	19.64	56.01	177.5	69.7	f
##	92	4.38	7.9	39.8	13.5	88	21.25	80.6	17.07	46.52	162.5	56.1	f
##	93	4.08	6.6	37.8	12.1	182	20.53	68.3	15.31	51.75	172.5	61.1	f
##	94	4.98	6.4	44.8	14.8	80	17.06	47.6	11.07	42.15	166.7	47.4	f
##	95	5.16	7.2	44.3	14.5	88	18.29	61.9	12.92	48.76	175.0	56.0	f
##	96	4.66	6.4	40.9	13.9	109	18.37	38.2	8.45	41.93	157.9	45.8	f
##	114	4.87	8.2	43.8	15.0	130	23.57	49.2	9.00	78.00	190.7	85.7	m
##	115	5.04	7.1	44.0	14.8	64	25.84	61.8	12.61	75.00	181.8	85.4	m
##	116	4.40	5.3	42.5	14.5	109	24.06	46.5	9.03	78.00	188.3	85.3	m
##	117	4.95	5.9	45.4	15.5	125	23.85	34.8	6.96	87.00	198.0	93.5	m
##	118	4.78	9.3	43.0	14.7	150	25.09	60.2	10.05	78.00	186.0	86.8	m
##	119	5.21	6.8	44.5	15.4	115	23.84	48.1	9.56	79.00	192.0	87.9	m
##	120	5.22	8.4	47.5	16.2	89	25.31	44.5	9.36	79.00	185.6	87.2	m
##	121	5.18	6.5	45.4	14.9	93	19.69	54.0	10.81	48.00	165.3	53.8	m
##	122	5.40	6.8	49.5	17.3	183	26.07	44.7	8.61	82.00	185.6	89.8	m
##	123	4.92	5.4	46.2	15.8	84	25.50	64.9	9.53	82.00	189.0	91.1	m
##	124	5.24	7.5	46.5	15.5	70	23.69	43.8	7.42	82.00	193.4	88.6	m
##	125	5.09	10.1	44.9	14.8	118	26.79	58.3	9.79	83.00	185.6	92.3	m
##	126	4.83	5.0	43.8	15.1	61	25.61	52.8	8.97	88.00	194.6	97.0	m
##	127	5.22	6.0	46.6	15.7	72	25.06	43.1	7.49	83.00	189.0	89.5	m
##	128	4.71	8.0	45.5	15.6	91	24.93	78.0	11.95	78.00	188.1	88.2	m
##	199	5.66	8.3	50.2	17.7	38	23.76	56.5	10.05	72.00	183.5	80.0	m
##	200	5.03	6.4	42.7	14.3	122	22.01	47.6	8.51	68.00	183.1	73.8	m
##	201	4.97	8.8	43.0	14.9	233	22.34	60.4	11.50	63.00	178.4	71.1	m
##	202	5.38	6.3	46.0	15.7	32	21.07	34.9	6.26	72.00	190.8	76.7	m
##		sport											
##	14	Row											
##	15	Row											
##	16	Row											
##	17	Row											
##	18	Row											
##	19	Row											
##	20	Row											
##	21	Row											
##	22	Row											
##	23	Row											
##	24	Row											
##	25	Row											
##	26	Row											
##	27	Row											
##	28	Row											
##	29	Row											
##	30	Row											
##	31	Row											
##	32	Row											
##	33	Row											
##	34	Row											
##	35	Row											
##	36	Netball											
##	37	Netball											
##	38	Netball											
##	39	Netball											
##	40	Netball											


```

## 41 Netball
## 42 Netball
## 43 Netball
## 44 Netball
## 45 Netball
## 46 Netball
## 47 Netball
## 48 Netball
## 49 Netball
## 50 Netball
## 51 Netball
## 52 Netball
## 53 Netball
## 54 Netball
## 55 Netball
## 56 Netball
## 57 Netball
## 58 Netball
## 90 Tennis
## 91 Tennis
## 92 Tennis
## 93 Tennis
## 94 Tennis
## 95 Tennis
## 96 Tennis
## 114 Row
## 115 Row
## 116 Row
## 117 Row
## 118 Row
## 119 Row
## 120 Row
## 121 Row
## 122 Row
## 123 Row
## 124 Row
## 125 Row
## 126 Row
## 127 Row
## 128 Row
## 199 Tennis
## 200 Tennis
## 201 Tennis
## 202 Tennis

myvars <- c("ht")
newdata <- roeinetball[myvars]
summary(newdata)

##      ht
## Min.   :156.0
## 1st Qu.:174.2
## Median :179.5
## Mean    :179.1
## 3rd Qu.:183.4

```

```
## Max. :198.0
```

3. ontleed de gegevens voor de vrouwelijke basketballers en roeiers

```
vrouwen <- DAAG::ais[DAAG::ais$sex == 'f' & (DAAG::ais$sport == 'B_Ball' | DAAG::ais$sport == 'Row'),]  
vrouwen$sex <- factor(vrouwen$sex)  
vrouwen$sport <- factor(vrouwen$sport)  
vrouwen
```

```
##      rcc  wcc  hc  hg ferr  bmi  ssf pcBfat  lbm  ht  wt sex  sport  
## 1  3.96  7.5 37.5 12.3  60 20.56 109.1  19.75 63.32 195.9 78.9  f B_Ball  
## 2  4.41  8.3 38.2 12.7  68 20.67 102.8  21.30 58.55 189.7 74.4  f B_Ball  
## 3  4.14  5.0 36.4 11.6  21 21.86 104.6  19.88 55.36 177.8 69.1  f B_Ball  
## 4  4.11  5.3 37.3 12.6  69 21.88 126.4  23.66 57.18 185.0 74.9  f B_Ball  
## 5  4.45  6.8 41.5 14.0  29 18.96  80.3  17.64 53.20 184.6 64.6  f B_Ball  
## 6  4.10  4.4 37.4 12.5  42 21.04  75.2  15.58 53.77 174.0 63.7  f B_Ball  
## 7  4.31  5.3 39.6 12.8  73 21.69  87.2  19.99 60.17 186.2 75.2  f B_Ball  
## 8  4.42  5.7 39.9 13.2  44 20.62  97.9  22.43 48.33 173.8 62.3  f B_Ball  
## 9  4.30  8.9 41.1 13.5  41 22.64  75.1  17.95 54.57 171.4 66.5  f B_Ball  
## 10 4.51  4.4 41.6 12.7  44 19.44  65.1  15.07 53.42 179.9 62.9  f B_Ball  
## 11 4.71  5.3 41.4 14.0  38 25.75 171.1  28.83 68.53 193.4 96.3  f B_Ball  
## 12 4.62  7.3 43.8 14.7  26 21.20  76.8  18.08 61.85 188.7 75.5  f B_Ball  
## 13 4.35  7.8 41.4 14.1  30 22.03 117.8  23.30 48.32 169.1 63.0  f B_Ball  
## 14 4.26  6.2 41.0 13.9  48 25.44  90.2  17.71 66.24 177.9 80.5  f   Row  
## 15 4.63  6.0 43.7 14.7  30 22.63  97.2  18.77 57.92 177.5 71.3  f   Row  
## 16 4.36  5.8 40.3 13.3  29 21.86  99.9  19.83 56.52 179.6 70.5  f   Row  
## 17 3.91  7.3 37.6 12.9  43 22.27 125.9  25.16 54.78 181.3 73.2  f   Row  
## 18 4.51  8.3 43.7 14.7  34 21.27  69.9  18.04 56.31 179.7 68.7  f   Row  
## 19 4.37  8.1 41.8 14.3  53 23.47  98.0  21.79 62.96 185.2 80.5  f   Row  
## 20 4.90  6.9 44.0 14.5  59 23.19  96.8  22.25 56.68 177.3 72.9  f   Row  
## 21 4.46  5.7 39.2 13.0  43 23.17  80.3  16.25 62.39 179.3 74.5  f   Row  
## 22 3.95  3.3 36.9 12.5  40 24.54  74.9  16.38 63.05 175.3 75.4  f   Row  
## 23 4.46  9.5 41.5 14.5  92 22.96  83.0  19.35 56.05 174.0 69.5  f   Row  
## 24 5.02  6.4 44.8 15.2  48 19.76  91.0  19.20 53.65 183.3 66.4  f   Row  
## 25 4.26  5.8 41.2 14.1  77 23.36  76.2  17.89 65.45 184.7 79.7  f   Row  
## 26 4.46  5.6 41.1 14.3  71 22.67  52.6  12.20 64.62 180.2 73.6  f   Row  
## 27 4.16  5.8 39.8 13.3  37 24.24 111.1  23.70 60.05 180.2 78.7  f   Row  
## 28 4.49  7.6 41.8 14.4  71 24.21 110.7  24.69 56.48 176.0 75.0  f   Row  
## 29 4.21  7.5 38.4 13.2  73 20.46  74.7  16.58 41.54 156.0 49.8  f   Row  
## 30 4.57  6.6 42.8 14.5  85 20.81 113.5  21.47 52.78 179.7 67.2  f   Row  
## 31 4.87  6.4 44.8 15.0  64 20.17  99.8  20.12 52.72 180.9 66.0  f   Row  
## 32 4.44 10.1 42.7 14.0  19 23.06  80.3  17.51 61.29 179.5 74.3  f   Row  
## 33 4.45  6.6 42.6 14.1  39 24.40 109.5  23.70 59.59 178.9 78.1  f   Row  
## 34 4.41  5.9 41.1 13.5  41 23.97 123.6  22.39 61.70 182.1 79.5  f   Row  
## 35 4.87  7.3 44.1 14.8  13 22.62  91.2  20.43 62.46 186.3 78.5  f   Row
```

oefening 3.8.

(oefening zelf gemaakt, geen oplossing)

opgave

Gebruik de functies mean en range om het gemiddelde en bereik van:

1. de cijfers 1, 2, . . . , 21

2. 50 willekeurige normale waarden, die worden gegenereerd vanuit een normale distributie met gemiddelde 0 en variantie 1 (functie rnorm)
3. de kolommen height en weight in de data frame women (standaard in R).

oplossing

deel 1

```
lijst <- c(1:21)
lijst

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
mean(lijst)

## [1] 11
range(lijst)

## [1] 1 21
```

deel 2

```
randomLijst <- rnorm(n = 50, mean = 0, sd = sqrt(1))
randomLijst

## [1] 0.10256756 -0.86030465 -1.15839200 -0.32477092 1.35078634
## [6] -0.34146066 1.78227786 1.24254431 -0.87024797 -0.33808449
## [11] 0.16023684 0.79721602 2.17429302 0.88263311 -1.78568963
## [16] -0.11257835 0.34098060 0.60292464 -0.04283279 -0.93318491
## [21] -0.77911888 -0.29137997 1.89584642 0.57577512 0.50468327
## [26] 0.34747459 1.43317122 0.46172495 0.32857240 0.07620997
## [31] 0.06521833 0.55955247 -0.42509439 0.78763115 0.11044298
## [36] -0.09100233 -0.83664997 1.70850465 0.36248670 0.28247190
## [41] 0.87487694 1.18662020 -0.45025677 0.05861508 1.22521739
## [46] -0.83448466 -0.56911824 1.10933372 -0.48828176 -0.57042273
mean(randomLijst)

## [1] 0.2257507
sd(randomLijst)

## [1] 0.8683005
```

deel 3

```
women # is een tabel die standaard in R zit.

##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
## 7     64    132
## 8     65    135
```

```
## 9      66      139
## 10     67      142
## 11     68      146
## 12     69      150
## 13     70      154
## 14     71      159
## 15     72      164

mean(women$height)

## [1] 65

median(women$height)

## [1] 65
## r heeft geen functie voor modus
range(women$height)

## [1] 58 72

quantile(women$height)

## 0% 25% 50% 75% 100%
## 58.0 61.5 65.0 68.5 72.0

sd(women$height)

## [1] 4.472136

mean(women$weight)

## [1] 136.7333

median(women$weight)

## [1] 135
## r heeft geen functie voor modus
range(women$weight)

## [1] 115 164

quantile(women$weight )

## 0% 25% 50% 75% 100%
## 115.0 124.5 135.0 148.0 164.0

sd(women$weight)

## [1] 15.49869
```

oefening 3.9

(oefening zelf gemaakt, geen oplossing)

opgave

Open de file met excel en bekijk de structuur van het document. Hoe ziet die er uit? Kan je de variabelen identificeren en hun type benoemen

oplossing

```
android_cpu <- read.csv("C:/Users/tijism/Google Drive/HoGent 2018-2019/2e semester/Onderzoekstechnieken/1  
attach(android_cpu)  
#android_cpu  
typeof(Tijd)
```

```
## [1] "double"
```

```
typeof(PersistentieType)
```

```
## [1] "integer"
```

```
typeof(Datahoeveelheid)
```

```
## [1] "integer"
```

variabelen zijn

- tijd
- persistentietype
- datahoeveelheids

```
summary(android_cpu)
```

```
##      Tijd      PersistentieType Datahoeveelheid  
## Min.   : 1.090   GreenDAO      :90      Medium: 90  
## 1st Qu.: 1.790   Realm        :90      Veel  : 90  
## Median : 6.185   Sharedpreferences:30      Weinig:120  
## Mean   : 6.231   SQLite       :90  
## 3rd Qu.:10.662  
## Max.   :13.560
```

```
sd(android_cpu$Tijd)
```

```
## [1] 4.229599
```

```
mean(android_cpu$Tijd)
```

```
## [1] 6.230833
```

Oefening 3.10.

(oefening zelf gemaakt, geen oplossing)

opgave

Als je de vorige metrieken berekend hebt, wat kan je daar dan over zeggen. Kan je zinnige conclusies trekken uit de vorige resultaten. Zo ja vermeld ze, zo nee beschrijf waarom je dat denkt.

oplossing

Ik denk niet dat je uit vorige data conclusies kan trekken omdat je niet weet over welk PU ze praten. je weet bv. niet bij welke PU het minimum van 1.090 werd bereikt.

Je kan wel afleiden dat er wel degelijk veel tijdsverschil was tussen de de PU's maar dit kan natuurlijk ook een gevolg zijn van de data hoeveelheid. Maar deze relaties kan je dus niet afleiden uit deze data.

Oplossing Chamillo:

Enkel gegevens over de tijd zijn zichtbaar maar niet per categorie. Zinnige conclusies trekken is dus niet evident.

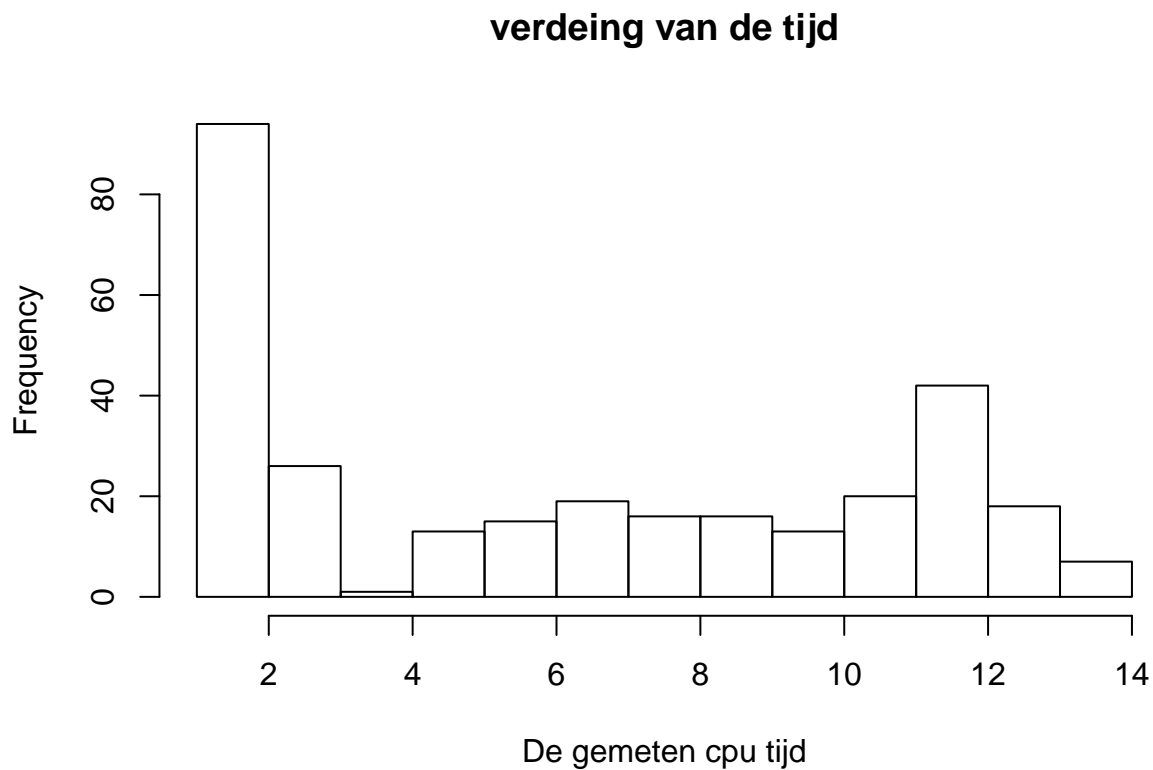
oefening 3.11.

(oefening zelf gemaakt, geen oplossing)

opgave

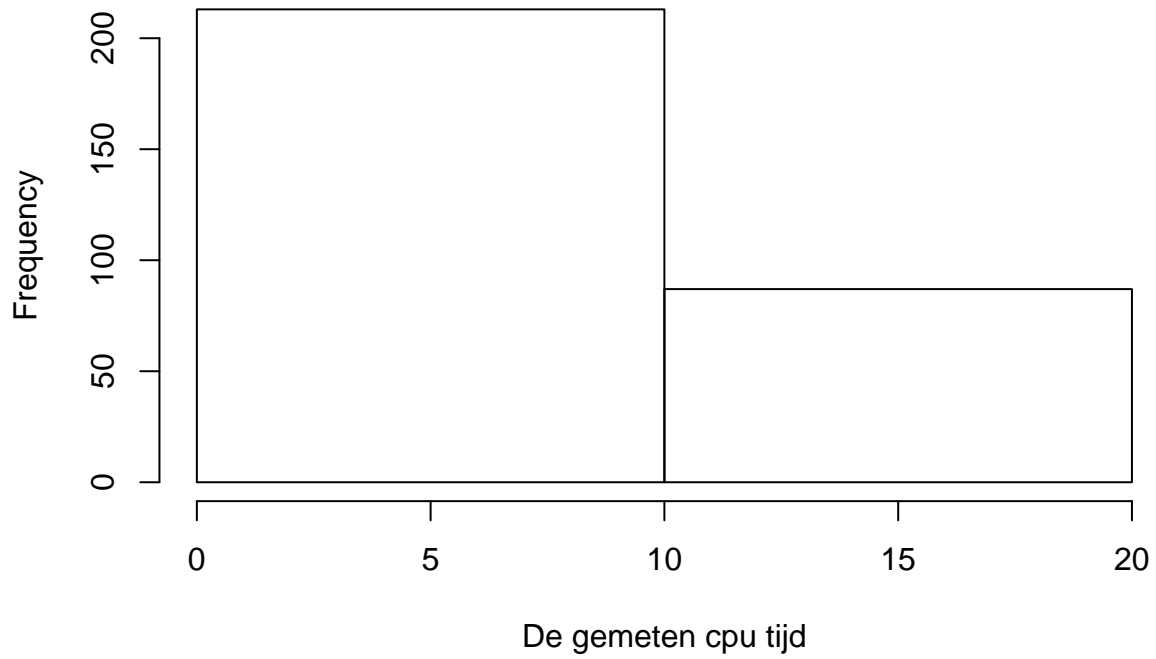
Een histogram is een eenvoudige plot. het toont de frequenties van de data die in een bepaald bereik voorkomen.

```
hist(android_cpu$Tijd, main = "verdeling van de tijd", xlab = "De gemeten cpu tijd");
```



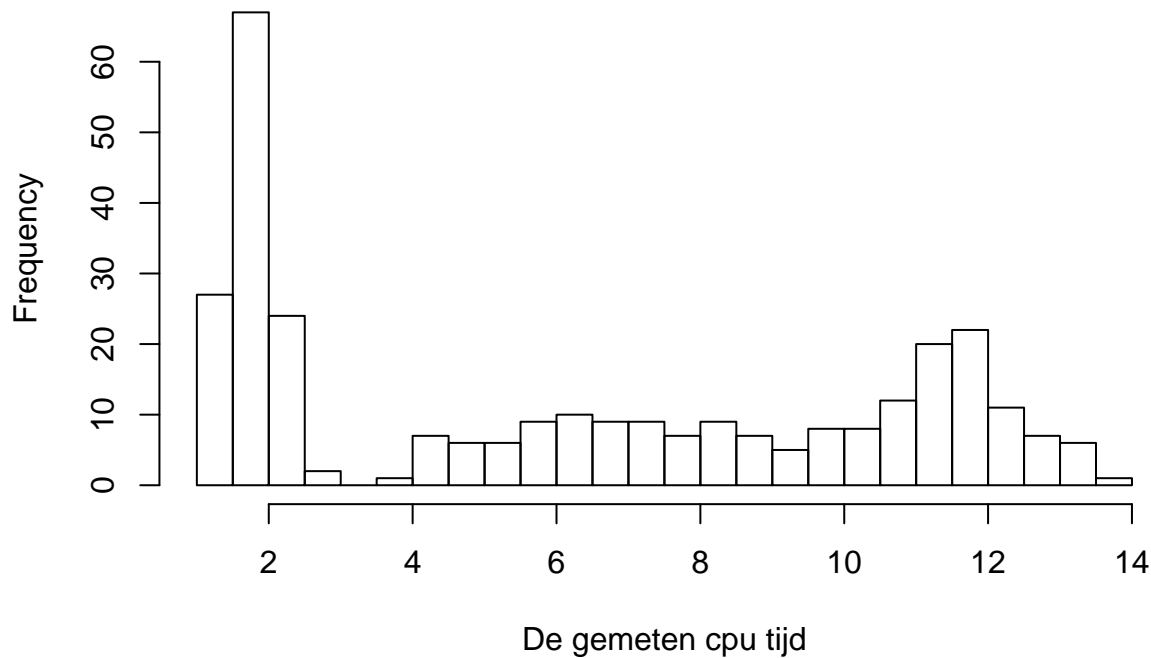
```
#default "break" = 10  
hist(android_cpu$Tijd, main = "verdeling van de tijd", xlab = "De gemeten cpu tijd", breaks = 1);
```

verdeling van de tijd



```
hist(android_cpu$Tijd, main = "verdeling van de tijd", xlab = "De gemeten cpu tijd", breaks = 20);
```

verdeing van de tijd



Wat concludeer je als je bovenstaande grafieka genereert? Is dit een zinnig resultaat? Wat gebeurt er als je de variabele breaks verhoogt?

oplossing

nuttig?

Het is niet zo nuttig om te weten hoe vaak een bepaalde uitkomst voorkomt. Het lijkt mij interessanter om te zien wat de gemiddelde cpu tijd per persistence unit is.

breaks

de “breaks” variabele stelt in hoeveel categorieën de dataset gesplitst wordt. default is deze waarde 10.

oplossing chamilo: De voorkomens van per cpu tijd zijn zichtbaar. Je kan afleiden dat er 1 groot interval is en deze dus de mean kan beïnvloeden. De categorieën zijn echter niet zichtbaar dus er is niet veel nut aan de grafiek. Op de x as zijn de intervallen vergroot.

Oefening 3.12.

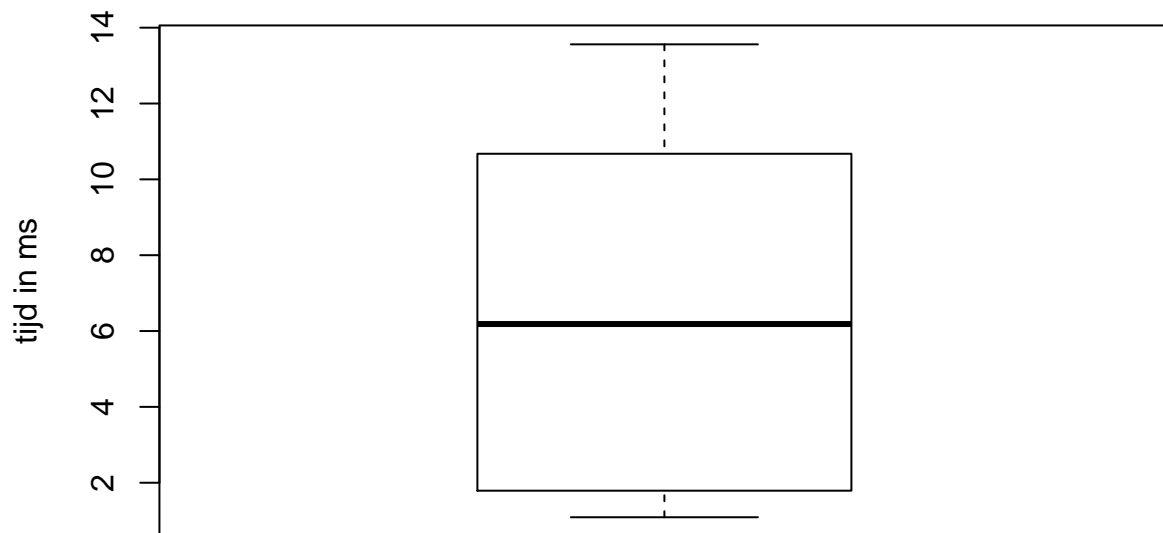
(oefening zelf gemaakt, geen oplossing)

Opgave

De boxplot wordt standaard verticaal getekend

```
boxplot(android_cpu$Tijd, main="spreiding van de CPU tijd", ylab="tijd in ms")
```


spreiding van de CPU tijd



Oplossing

oplossing chamilo:

Oefening 3.13.

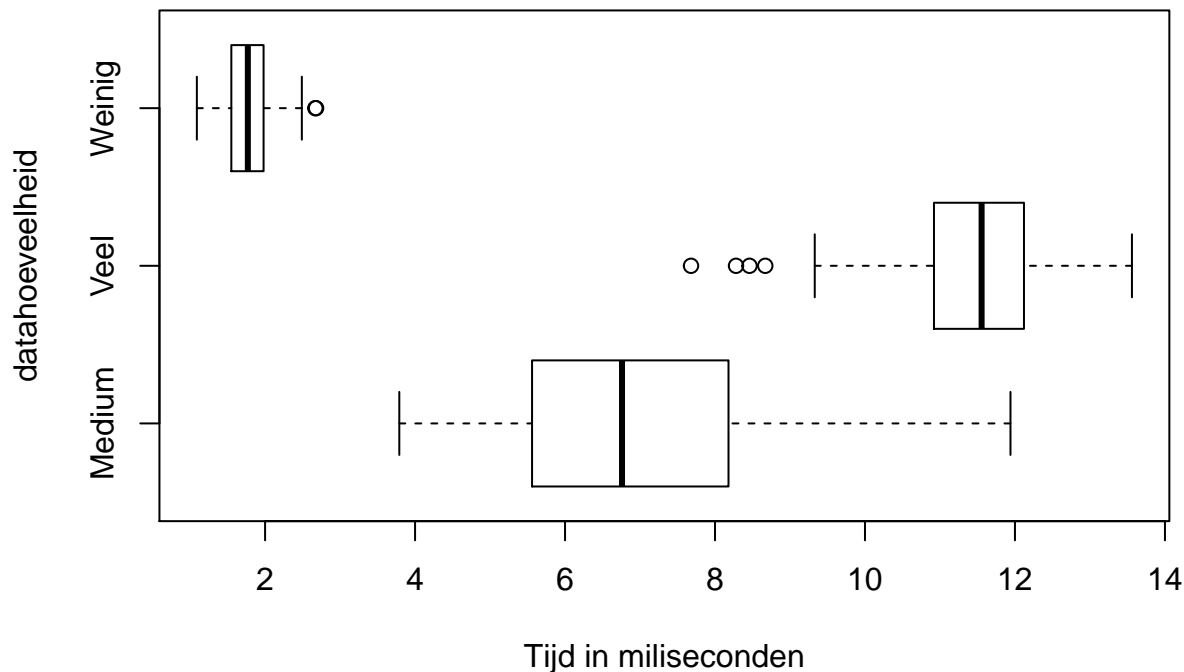
(oefening zelf gemaakt, geen oplossing)

Opgave

Als je goed geantwoord hebt op de volgende vragen merk je natuurlijk dat het weinig zin heeft de volledige dataset te analyseren, aangezien de dataset verdeeld is over verschillende categorieën. We willen dus wel deze statistieken weten, maar per categorie. We kunnen dus een boxplot maken voor elke categorie

```
boxplot(android_cpu$Tijd~android_cpu$Datahoeveelheid, main = "spreiding van de cpu tijd t.o.v. datahoeveelheid")
```

spreiding van de cpu tijd t.o.v. datahoeveelheid



Interpreteer de resultaten die je behaalt uit deze grafiek. Zijn deze al wat zinniger?

Oplossing

Nee niet echt, Het is logisch dat bij een grote datahoeveelheid de tijd in miliseconden groter zal zijn

Het is wel opvallend dat de spreiding bij medium grote datahoeveelheden heel groot is. We weten natuurlijk niet wat de intervallen zijn die “medium” definiëren. Als dit interval breder is dan bij “klein” en “groot” is deze observatie logisch

oplossing chamilo:

Er is een duidelijker overzicht met uitschieters van de datahoeveelheid. De volledige data is echter nog niet weergegeven.

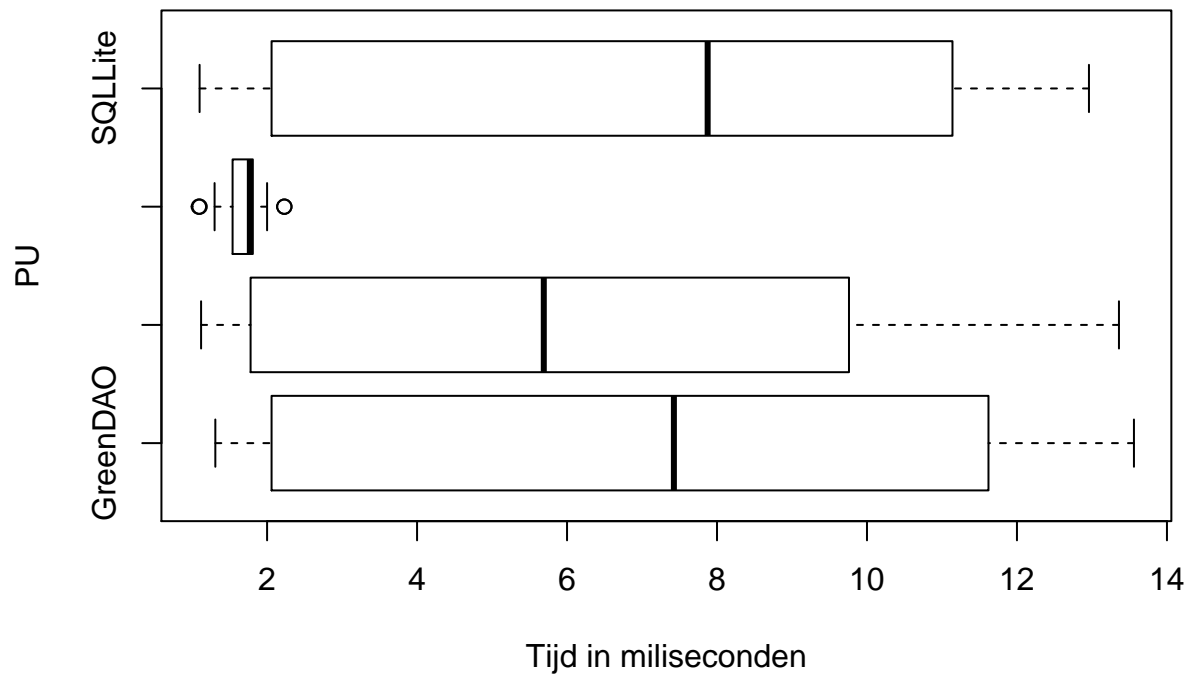
oefening 3.14

(oefening zelf gemaakt, geen oplossing)

opgave

```
boxplot(android_cpu$Tijd~android_cpu$PersistentieType, main = "spreiding van de cpu tijd t.o.v. persiste
```

spreiding van de cpu tijd t.o.v. persistentieunit



Zijn deze resultaten zinniger?

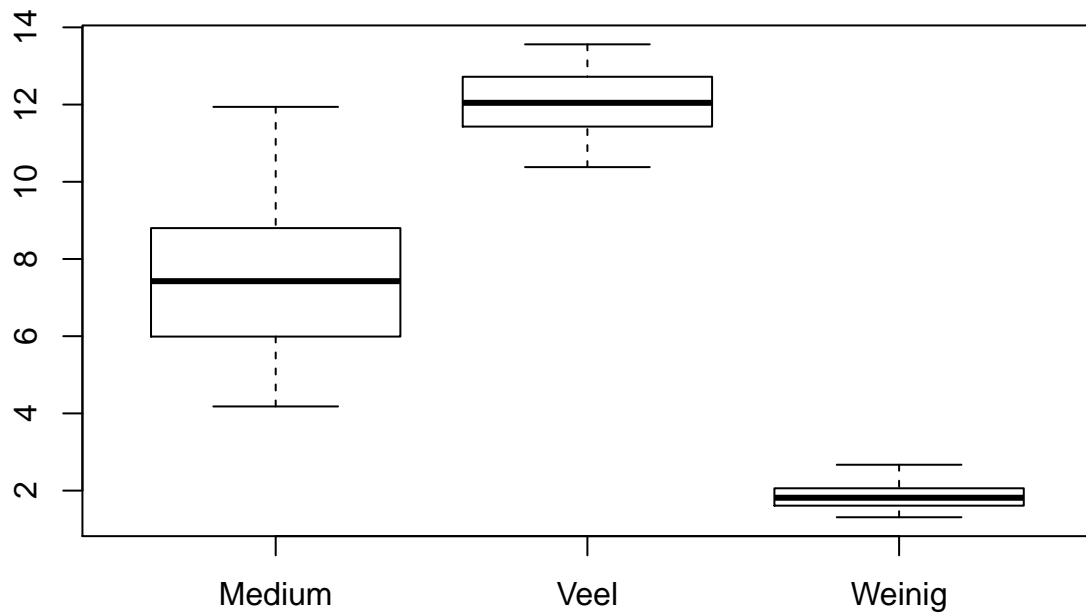
oplossing

Nee niet echt

oefening 15

We moeten de data dus onderverdelen in categorieën namelijk onder PersistentieType en Datahoeveelheid. We gaan hiervoor de functie `which1` gebruiken en kijken hoe de verschillende datahoeveelheden verschillen per datahoeveelheidscategorie.

```
greenDOA <- android_cpu[which(android_cpu$PersistentieType=='GreenDAO'),];  
boxplot(greenDOA$Tijd~greenDOA$Datahoeveelheid)
```



wat concludeer je

oplossing

dat deze persistence unit recht evenredig werkt in functie van de datahoeveelheid. (zoals verwacht)

Bij medium is de spreiding wel opvallend hoog.

voorbeelden

blz 38

bewerkingen op een vector

```
a <- c(1,2,3,4)
a
```

```
## [1] 1 2 3 4
```

```
a <- a + 5
a
```

```
## [1] 6 7 8 9
```

```
a <- a * 3
a
```

```
## [1] 18 21 24 27
```

```
b <- a -15
b
```

```
## [1] 3 6 9 12
```

```
# wortel
sqrt(a)
```

```
## [1] 4.242641 4.582576 4.898979 5.196152
```

analyses op 1 variabele

```
#inlezen csv
computers <- read.csv("C:/Users/tijsm/Google Drive/HoGent 2018-2019/2e semester/Onderzoekstechnieken/fi
# vermijden dat er steeds computers$ moet getypt worden
attach(computers)
```

```
#gemiddelde
mean(price)
```

```
## [1] 2219.577
```

```
#mediaan
median(price)
```

```
## [1] 2144
```

```
#kwartielen
quantile(price)
```

```
## 0% 25% 50% 75% 100%
## 949 1794 2144 2595 5399
```

```
#minimum
min(price)
```

```
## [1] 949
```

```
#maximum
max(price)
```

```
## [1] 5399
```

```
#variantie
var(price)
```

```
## [1] 337333.2
```

```
#standaardafwijking
sd(price)
```

```
## [1] 580.804
```

```
## alles
summary(computers)
```

```
##      price      speed      hd      ram
## Min.   : 949   Min.   : 25.00   Min.   : 80.0   Min.   : 2.000
## 1st Qu.:1794   1st Qu.: 33.00   1st Qu.: 214.0   1st Qu.: 4.000
## Median :2144   Median : 50.00   Median : 340.0   Median : 8.000
## Mean   :2220   Mean    : 52.01   Mean    : 416.6   Mean    : 8.287
```

##	3rd Qu.:2595	3rd Qu.: 66.00	3rd Qu.: 528.0	3rd Qu.: 8.000
##	Max. :5399	Max. :100.00	Max. :2100.0	Max. :32.000
##	screen	cd	multi	premium
##	Min. :14.00	no :3351	no :5386	no : 612
##	1st Qu.:14.00	yes:2908	yes: 873	yes:5647
##	Median :14.00			
##	Mean :14.61			
##	3rd Qu.:15.00			
##	Max. :17.00			
##	trend			
##	Min. : 1.00			
##	1st Qu.:10.00			
##	Median :16.00			
##	Mean :15.93			
##	3rd Qu.:21.50			
##	Max. :35.00			