

R-Workshop

- Inleiding
 - De instructeur
 - Voorbereiding
 - Basis R
 - Naslagwerken
- Sessie 1 - Inlezen / Bekijken
- Sessie 2 - Muteren / Analyseren
- Sessie 3 - Visualiseren / Presenteren

Zie ook online: <https://witusj.github.io/WorkshopSI>

Witek ten Hove - wft1969@gmail.com

05 October, 2017

Inleiding

In deze workshop leer je op een slimme wijze kwantitatieve data verzamelen, deze om te zetten naar bruikbare input voor je analyse en uitkomsten te visualiseren en te rapporteren.

Tijdens de workshop werk je met [R](#). Deze software is speciaal ontwikkeld om makkelijk en snel data in te lezen en te verwerken. Het is op dit moment naast Python de [belangrijkste tool voor data analyse](#) en wordt ingezet door ondernemingen als bijvoorbeeld Facebook, Airbnb en Microsoft om informatie uit hun enorme databestanden te genereren. Je bent dus in uitstekend gezelschap als je klaar bent met het leertraject.

Deze workshop is ontwikkeld voor mensen die nog geen of zeer beperkte ervaring hebben met programmeren. Wellicht heb je al een beetje ervaring met Excel of SPSS. Dit is handig, maar geen vereiste om deel te kunnen nemen. R is een programmeertaal, maar zeer eenvoudig qua opzet en je zult merken dat je al heel snel krachtige stukjes code (scripts) kunt schrijven. Je hoeft geen investeringen te doen, want R is volledig open source en werkt op zowel Mac OS, Windows als Linux.

De instructeur

[Witek ten Hove](#) is hoofddocent aan de Hogeschool van Arnhem en Nijmegen en voorzitter van de R-User Group in Nijmegen. Na zijn studie aan de Universiteit van Maastricht heeft hij bij verschillende multinationals gewerkt als business analist, beurshandelaar en commodity trader. Zijn specialisaties zijn onder andere data science en online / blended learning.

De workshop bestaat uit drie sessies: Inlezen, Analyseren en Presenteren. Tijdens iedere sessie ga je werken aan opdrachten. Je krijgt een dataset (of een verwijzing naar een online databron) en gaat hier informatie uit halen. De instructeur helpt je als je vastloopt of geeft suggesties voor verbeteringen van je code.

Voorbereiding

Om direct aan de gang te kunnen gaan tijdens de workshop sessies is het belangrijk dat je R op je computer installeert. Zie [hier](#) voor instructies (Engels).

Dit is de Nederlandse [downloadsite](#) voor R (je kunt ook iedere andere downloadsite gebruiken).

Als je klaar bent met de installatie kun je in principe al aan de gang gaan met het pakket. R heeft namelijk een eigen interface, waarmee via een command line opdrachten kunnen worden ingevoerd.

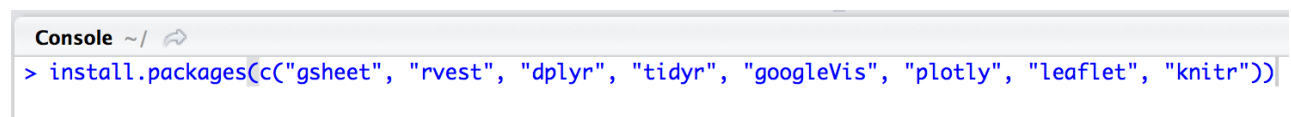
Om echter wat comfortabeler te programmeren zijn er zogenaamde *integrated development environments (IDEs)* ontwikkeld. Wij gaan werken met die van [RStudio](#). Ook deze software is gratis te [downloaden](#).

LET OP: Het is belangrijk dat je eerst R en daarna RStudio installeert.

Als je alles hebt geïnstalleerd, open RStudio, kopieer de volgende code:

```
install.packages(c("gsheet", "rvest", "dplyr", "tidyr", "googleVis",  
"plotly", "leaflet", "knitr"))
```

plak het in de console en druk op *Enter*.

A screenshot of the RStudio console window. The title bar says "Console ~/". The command prompt shows the command: `> install.packages(c("gsheet", "rvest", "dplyr", "tidyr", "googleVis", "plotly", "leaflet", "knitr"))`. The cursor is at the end of the command.

```
Console ~/  
> install.packages(c("gsheet", "rvest", "dplyr", "tidyr", "googleVis", "plotly", "leaflet", "knitr"))
```

Er worden nu een aantal sets van functies geïnstalleerd die we tijdens de sessies gaan gebruiken.

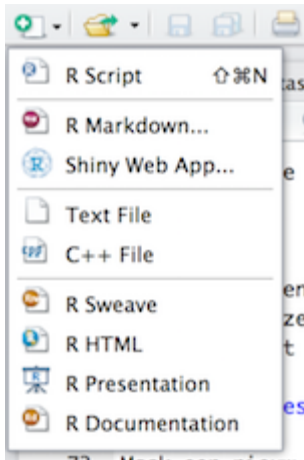
Basis R

In RStudio kun je op twee manieren code schrijven en laten uitvoeren:

- Direct via de console
- Indirect via een script

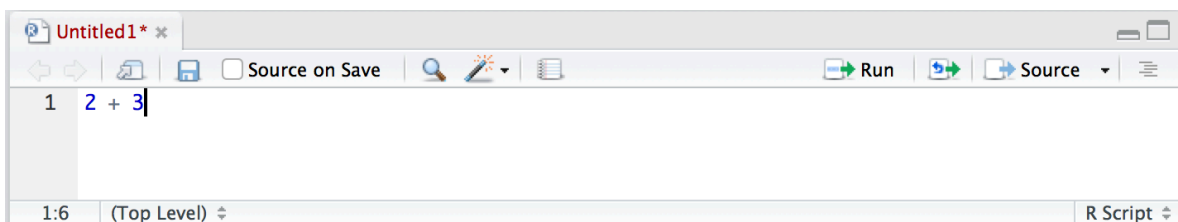
Meestal zul je eerst je code in een script schrijven (of kopiëren uit voorbeelden) en als je helemaal tevreden bent, uitvoeren. Het voordeel is tevens dat je een script kunt bewaren en later dus kunt hergebruiken.

Maak een nieuw R Script aan in RStudio via het icoontje met het ‘groene plusje’:



of *File >> New File >> R Script*

Er wordt nu een venster met een veld voor tekstbewerking geopend. Je kunt hier je instructies invoeren, bijvoorbeeld een optelling.



Selecteer alles en klik op *Run*:



of *Ctrl + Enter*

In de console zie je het resultaat.

```
## [1] 5
```

We hebben nu niet meer gedaan dan je ook op een gewone rekenmachine kunt uitvoeren. R is echter geen rekenmachine, maar een programmeertaal. In iedere programmeertaal kun je gegevens in variabelen opslaan en deze variabelen hergebruiken.

We kunnen bijvoorbeeld een variabele *a* en *b* aanmaken en daarin de waarden 2 en 3 bewaren. Het resultaat van de optelling bewaren we in een variabele *c*.

```
a <- 2
b <- 3
c <- a + b
c
## [1] 5
```

Het voordeel hiervan is dat we nu *a* of *b* andere waarden kunnen geven en dat de waarde van *c* automatisch wordt aangepast.

```
a <- 4
b <- 3
c <- a + b
c
## [1] 7
```

Om goed te kunnen begrijpen hoe R werkt, is het belangrijk bekend te raken met een aantal basiselementen, *classes*, *vectors* en *data frames*, en te begrijpen hoe deze elementen zich tot elkaar verhouden.

De belangrijkste datatypes (*classes*) binnen R zijn: *getallen*, *letters* en *factoren*. Een vector is een geordende verzameling van elementen van hetzelfde type data.

Bijvoorbeeld de vector: ("Warszawa", "London", "Paris") is een vector van plaatsen en de elementen zijn van class *letters* (*character*). De elementen uit de vector ("Man", "Man", "Vrouw", "Transgender") kunnen van class *letters* zijn, maar waarschijnlijk is de functie van zo een vector een verzameling van *factoren* met 3 levels: Transgender, Man en Vrouw. Dit kunnen we specifiek aangeven in R.

```
## Vectoren en classes
Plaats <- c("Warszawa", "London", "Paris")
class(Plaats)
## [1] "character"
Geslacht <- c("Man", "Man", "Vrouw", "Transgender")
class(Geslacht)
## [1] "character"
Geslacht <- as.factor(Geslacht)
class(Geslacht)
## [1] "factor"
Geslacht
## [1] Man      Man      Vrouw    Transgender
## Levels: Man Transgender Vrouw
```

Ieder element binnen een vector heeft een indexnummer beginnend bij 1. Als we de vector ("Warszawa", "London", "Paris") de naam `Plaats` geven, dan heeft het element `Plaats[2]` de waarde "London".

```
## Vectorindices
Plaats[2]
## [1] "London"
```

We kunnen met behulp van indexnummers elementen toevoegen of verwijderen.

```
## Vectorelementen toevoegen of verwijderen.
Plaats[4] <- "Amsterdam"
Plaats
## [1] "Warszawa" "London" "Paris" "Amsterdam"
Plaats <- Plaats[-3]
Plaats
## [1] "Warszawa" "London" "Amsterdam"
```

Stel we creëren een tweede vector `Land` met de elementen ("Polska", "UK", "Nederland"). We kunnen nu de vectoren `Plaats` en `Land` samenvoegen in een data frame en via de indexen *[rij,kolom]* de elementen opvragen.

```
## Data frame structuur
Land <- c("Polska", "UK", "Nederland")
Locatie <- data.frame(Plaats, Land)
Locatie
##      Plaats      Land
## 1  Warszawa  Polska
## 2   London    UK
## 3 Amsterdam Nederland
Locatie[3,1]
## [1] Amsterdam
## Levels: Amsterdam London Warszawa
Locatie[3,]
##      Plaats      Land
## 3 Amsterdam Nederland
Locatie[,2]
## [1] Polska    UK      Nederland
## Levels: Nederland Polska UK
```

We kunnen de afzonderlijke variabelen binnen een data frame ook oproepen via het `$` teken.

```
## Data frame structuur
Locatie$Land
## [1] Polska    UK      Nederland
## Levels: Nederland Polska UK
Locatie$Plaats[1]
## [1] Warszawa
## Levels: Amsterdam London Warszawa
```

Vragenset Basis R

De antwoorden kun je eventueel [hier](#) achterlaten en controleren.

1. Zoals je ziet hebben de variabelen `Plaats` en `Land` uit het voorbeeld door de omzetting naar een data frame de `class factor` gekregen. Hoe zet je de `class` van beide variabelen weer terug naar `character`?
2. Hoe vervang je de elementen uit de laatste rij door `("Berlin", "Deutschland")`?
3. Neem volgende code over in een script, voer het uit en analyseer wat er gebeurt.

```
values1 <- c("20", "50")  
sum(values1)
```

```
values2 <- c(20, 50)  
sum(values2)
```

Naslagwerken

Ondanks dat R open source is, wordt er veel controle uitgeoefend op de kwaliteit van het pakket. Dat betekent onder andere dat iedere bijdrage goede documentatie moet bevatten voordat het als onderdeel in het pakket wordt opgenomen. RStudio heeft een helpfunctie, waarmee makkelijk in de documentatie gezocht kan worden. Zie ook de *Cheatsheets* onder het helpmenu.

Er zijn online ontzettend veel gratis handleidingen voor R te vinden. Volgende documenten zijn een willekeurige selectie:

- [An Introduction to R](#)
- [Introduction to R for Excel Users](#)

Al je een indruk wilt krijgen van wat je allemaal met R kunt, is de [R-bloggers](#) website een goed startpunt.

Sessie 1 - Inlezen / Bekijken

In deze sessie gaan we kijken hoe je data uit verschillende bronnen kunt inlezen. We beginnen eenvoudig en lezen een lokaal opgeslagen bestand uit. Daarna gaan we data uitlezen vanuit een online spreadsheet. We eindigen met het scrapen van informatie uit een webpagina.

Lokaal bestand

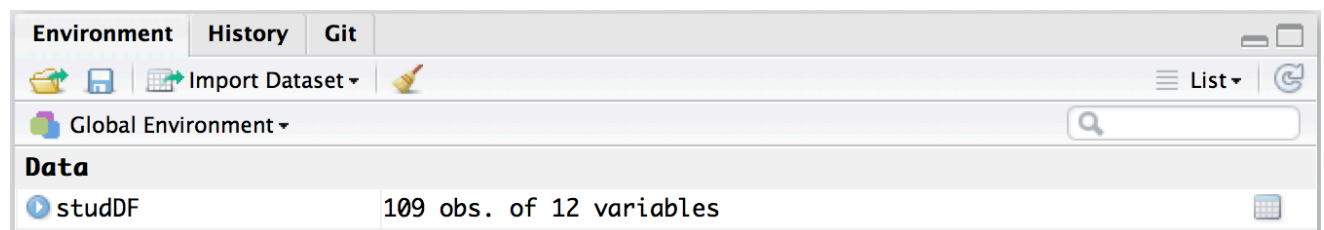
Maak een nieuw R Script aan in RStudio en bewaar het in een voor jou logische folder op je computer (geef het een zinvolle naam, bijvoorbeeld `werkSessie1.R`). Verwijs in RStudio naar deze werkfolder door te klikken op *Session >> Set Working Directory >> To Source File Location*

Download nu het volgende databestand: [Student Performance \(csv\)](#) en sla het op in de werkfolder. Nu ben je klaar om je eerste data in te lezen in R.

In het scriptvenster in RStudio schrijf (of kopieer) je volgende instructie:

```
## Lees data uit lokaal csv-bestand
studDF <- read.csv("stud_perf.csv", stringsAsFactors = FALSE)
```

Selecteer alles en klik op Run. In de console van RStudio zie je dat de instructies worden uitgevoerd, maar je ziet verder geen resultaten. R heeft echter alle data in het geheugen geladen. Je kunt dit zien in het *Environment* veld waar nu een data object *studDF* is toegevoegd:



We kunnen snel controleren of de data goed is overgekomen door naar de eerste zes rijen te kijken:

```
## Bekijk eerste regels van data
head(studDF)
##      Voor      Achter Kennistoetsen Schriftelijk Schriftelijk.Voldoende
## 1  ANTON  Entwistle           0.0           6.0              100
## 2 CHRISTEN Hardaway           4.6           2.2               0
## 3  KYLEE   Mauzey           5.3           8.8             100
## 4   KAI    Tredwell           5.0           4.8               0
## 5  NELL    Darden           6.7           9.2             100
## 6 KIRSTIE Tinklenberg        7.7           7.4             100
##      Schriftelijk.Bonus Schriftelijk.Bonus.Voldoende Clicks Weblectures
## 1           6.0              100          536           5.5
## 2           2.2               0          636           0.8
## 3           8.8             100          666           3.6
## 4           4.8               0          479           1.2
```

```
## 5          10.0          100      298          0.0
## 6           8.6          100      863          0.0
##   BEC.Propedeuse AEC.Propedeuse Vooropleiding
## 1           1           1           HAVO
## 2           0           0           MBO
## 3           1           1           MBO
## 4           1           1           HAVO
## 5           1           1           HAVO
## 6           1           1           MBO
```

Het gaat hier klaarblijkelijk om een cijferlijst van studenten met wat metadata. Alles lijkt correct te zijn ingelezen door R en we kunnen nu wat nauwkeuriger kijken naar de data. Het is een goede gewoonte om eerst de structuur van de data te onderzoeken:

```
## Bekijk de structuur van data
str(studDF)
## 'data.frame':   109 obs. of  12 variables:
##  $ Voor          : chr  "ANTON" "CHRISTEN" "KYLEE" "KAI"
##  ...
##  $ Achter         : chr  "Entwistle" "Hardaway" "Mauzey"
## "Tredwell" ...
##  $ Kennistoetsen  : num  0 4.6 5.3 5 6.7 7.7 3.8 4.8 4.5
## 4.1 ...
##  $ Schriftelijk   : num  6 2.2 8.8 4.8 9.2 7.4 5 5.8 3.2
## 6.6 ...
##  $ Schriftelijk.Voldoende : int  100 0 100 0 100 100 0 100 0 100
##  ...
##  $ Schriftelijk.Bonus : num  6 2.2 8.8 4.8 10 8.6 5 5.8 3.2 6.6
##  ...
##  $ Schriftelijk.Bonus.Voldoende: int  100 0 100 0 100 100 0 100 0 100
##  ...
##  $ Clicks         : int  536 636 666 479 298 863 434 428
## 441 1058 ...
##  $ Weblectures    : num  5.5 0.8 3.6 1.2 0 0 0 0 0 1.2 ...
##  $ BEC.Propedeuse : int  1 0 1 1 1 1 1 0 0 1 ...
##  $ AEC.Propedeuse : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ Vooropleiding  : chr  "HAVO" "MBO" "MBO" "HAVO" ...
```

en we zien dat de data de vorm van een *data frame* (tabel) heeft met 109 rijen (observaties) en 12 kolommen (variabelen). De eerste variabele heet `Voor` en de elementen bestaan uit letters (chr). De derde variabele heet `Kennistoetsen` en de elementen zijn getallen (num). De vijfde variabele heet `Schriftelijk.Voldoende` en bestaat uit gehele getallen (int).

Met `summary()` kunnen we de kwaliteit van de data beoordelen:

```
## Bekijk de kwaliteit van data
summary(studDF)
##      Voor          Achter          Kennistoetsen  Schriftelijk
## Length:109      Length:109      Min.   :0.00      Min.   :0.800
## Class :character Class :character 1st Qu.:3.20      1st Qu.:4.000
## Mode  :character Mode  :character Median :4.80      Median :5.600
##                               Mean  :4.48      Mean   :5.503
##                               3rd Qu.:5.70      3rd Qu.:6.800
##                               Max.   :8.30      Max.   :9.600
##
## Schriftelijk.Voldoende Schriftelijk.Bonus Schriftelijk.Bonus.Voldoende
```



```
## Min. : 0.00 Min. : 0.800 Min. : 0.00
## 1st Qu.: 0.00 1st Qu.: 4.000 1st Qu.: 0.00
## Median :100.00 Median : 5.600 Median :100.00
## Mean : 52.29 Mean : 5.787 Mean : 53.21
## 3rd Qu.:100.00 3rd Qu.: 7.600 3rd Qu.:100.00
## Max. :100.00 Max. :10.000 Max. :100.00
##
## Clicks Weblectures BEC.Propedeuse AEC.Propedeuse
## Min. : 1.0 Min. :0.0000 Min. :0.000 Min. :0.0000
## 1st Qu.: 269.0 1st Qu.:0.0000 1st Qu.:1.000 1st Qu.:1.0000
## Median : 386.0 Median :0.0000 Median :1.000 Median :1.0000
## Mean : 445.7 Mean :0.8505 Mean :0.875 Mean :0.9327
## 3rd Qu.: 541.0 3rd Qu.:1.2000 3rd Qu.:1.000 3rd Qu.:1.0000
## Max. :1058.0 Max. :6.8000 Max. :1.000 Max. :1.0000
## NA's :5 NA's :5
## Vooropleiding
## Length:109
## Class :character
## Mode :character
##
##
##
##
```

We kunnen hieruit onder andere aflezen dat de gemiddelde score van de kennistoetsen 4.48 bedroeg met een maximum van 8.3 en dat bij de variabele `BEC.Propedeuse` van 5 observaties de gegevens ontbreken (NA).

Door in het *Environment* veld op de variabele `studDF` te klikken (of in de *Console* met het commando `View(studDF)`), krijg je een tabel met alle data.

The screenshot shows the RStudio interface. On the left, a data table is displayed with columns: Voor, Achter, Kennistoetsen, Schriftelijk, Schriftelijk.Voldoende, Schriftelijk.Bonus, and Schriftelijk. The table contains 109 rows of data. On the right, the Environment pane shows the variable `studDF` with 109 observations and 12 variables. Below this, the Data pane shows the structure of the data, including factors for 'Voor' and 'Achter', and numeric variables for 'Kennistoetsen', 'Schriftelijk', 'Schriftelijk.Voldoende', and 'Schriftelijk.Bonus'.

	Voor	Achter	Kennistoetsen	Schriftelijk	Schriftelijk.Voldoende	Schriftelijk.Bonus	Schriftelijk
1	ANTON	Entwistle	0.0	6.0	100	6.0	
2	CHRISTEN	Hardaway	4.6	2.2	0	2.2	
3	KYLEE	Mauzey	5.3	8.8	100	8.8	
4	KAI	Tredwell	5.0	4.8	0	4.8	
5	NELL	Darden	6.7	9.2	100	10.0	
6	KIRSTIE	Tinklenberg	7.7	7.4	100	8.6	
7	ELIN	Vollstedt	3.8	5.0	0	5.0	

Vragenset 1A

1. Hoeveel procent van de studenten heeft voor het schriftelijk tentamen (inclusief bonuspunten) een voldoende gehaald?
2. In hoeveel procent van de gevallen hadden studenten de bonuspunten nodig om voor het schriftelijk een voldoende te halen?

Online bestand

We kunnen met R ook heel makkelijk data ophalen die online is opgeslagen. In dit voorbeeld lezen we data uit een Google Spreadsheet uit. Er zijn verschillende packages die voor deze taak geschikt zijn. Wij gaan de package `gsheet` gebruiken.

```
## Laad de benodigde package(s)
library(gsheet)

## Lees online data in
url <- "https://docs.google.com/spreadsheets/d/1j-
bW2MmrbrRYso2IJekNWKsGlWL2SwtJQ_ebSj9hs7kA"
surveyDF <- gsheet2tbl(url)
```

Vragenset 1B

1. Om wat voor een soort data gaat het hier?
2. Welke dimensies heeft de dataset (rijen / kolommen)?
3. Welke classes hebben de variabelen?
4. Wat is het gemiddelde niveau van de ingeschatte vaardigheden?

Web scraping

Een andere wijze om online data te verzamelen is via *web scraping*. Soms zie je op webpagina's tabellen met data staan, zoals bijvoorbeeld [hier](#). Met de `rvest` package kunnen we de data uit de html code filteren.

```
# Laad de benodigde package(s)
library(rvest)

# Lees ruwe html data in.
htmlpage <- read_html("http://www.pginvestor.com/Historic-Prices")

# Extraheer alle tabellen uit de ruwe data
nodes <- html_nodes(htmlpage, "table")
tables <- html_table(nodes, fill = TRUE)

# Kies een tabel en stop de data in een data frame
stockDF <- tables[1][[1]]
```

Vragenset 1C

1. Wat voor een soort data bevat `stockDF`?
 2. Welke classes hebben de variabelen?
 3. Wat was de hoogste waarde voor de variabele `Price`?
-

Sessie 2 - Muteren / Analyseren

In deze sessie gaan we de kwaliteit van de ruwe data bekijken en leren we hoe je met slimme tools data kunt aanpassen, zodat deze gebruikt kan worden voor verdere analyse of visualisatie.

Muteren

Maak een nieuw R Script aan in RStudio en bewaar het in een voor jou logische folder op je computer (geef het een zinvolle naam, bijvoorbeeld `werkSessie2.R`). Verwijs in RStudio naar deze werkfolder door te klikken op *Session >> Set Working Directory >> To Source File Location*

We gaan met functies werken uit de `dplyr` package. Een uitgebreid overzicht van deze functies vind je [hier](#).

```
library(dplyr)
library(tidyr)
```

Met de volgende code kun je een online csv bestand direct inlezen in R en met de `dplyr::glimpse()` functie de data bekijken. De oorspronkelijke data komt van de [Gemeente Alphen a/d Rijn](#).

```
## Lees data uit online csv-bestand
openDF <- read.csv(file = "https://raw.githubusercontent.com/witusj/R-workshop/gh-pages/Datasets/sessie%202/open_data.csv", stringsAsFactors = TRUE)
glimpse(openDF)
## Observations: 45,669
## Variables: 15
## $ X                (int) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1...
## $ id                (int) 2934, 1561, 1563, 1564, 1565, 1567, 1569...
## $ kl_imagcode       (fctr) PLACERIF, MEGLYPTO, MEGLYPTO, MEGLYPTO,...
## $ nedboomsoort      (fctr) Gewone plataan, Watercipres, Watercipre...
## $ latboomsoort      (fctr) Platanus x hispanica, Metasequoia glypt...
## $ aanlegjaar        (int) 1985, 1970, 1970, 1970, 1970, 1970, 1970...
## $ inspectiedatum    (fctr) 29-07-2014, 04-08-2014, 04-08-2014, 04-...
## $ kl_veilig_maatregelen (fctr) Geen veiligheidsmaatregel;, Geen veilig...
## $ structuurelement (fctr) Woongebied, Woongebied, Woongebied, Woo...
## $ woonplaats        (fctr) Boskoop, Boskoop, Boskoop, Boskoop, Bos...
## $ wijk              (fctr) Boskoop, Boskoop, Boskoop, Boskoop, Bos...
## $ buurt             (fctr) Snijdelwijk, Boskoop, Boskoop, Boskoop,...
```

```
## $ openbare_ruimte      (fctr) Bosweg, A.p. van neslaan, A.p. van
nesl...
## $ x                    (fctr) 104066,236, 105447,454, 105455,527,
105...
## $ y                    (fctr) 453323,99, 454397,319, 454442,028,
4544...
```

Vragenset 2A

1. Wat voor een soort data bevat `openDF`?
2. Wat is de latijnse naam voor ‘Moeraseik’?
3. Wat is het vroegste aanlegjaar in de dataset?
4. Van hoeveel boomsoorten binnen de Gemeente Alphen aan de Rijn is de latijnse naam vastgesteld? (HINT: gebruik de functie `levels()` en kijk goed bij de namen die met een ‘N’ beginnen)

Het kan wellicht handig zijn om de inspectiedatum op te splitsen in *dag*, *maand* en *jaar*.

```
## Splits de inspectiedatum
openDF <- separate(openDF, inspectiedatum, into = c("inspectiedag",
"inspectiemaand", "inspectiejaar"), sep="-", remove = FALSE, extra =
"drop", fill = "right")
```

Nu kunnen we makkelijk de inspecties van een bepaalde periode uit de data filteren (let op het dubbele = teken) en kijken welke bomen zijn aangemerkt om gerooid te worden.

```
## Filter op jaar en maand
openDF2014 <- filter(openDF, inspectiejaar == "2014", kl_veilig_maatregelen
== "Rooien;")
head(openDF2014)
##      X    id kl_imagcode      nedboomsoort      latboomsoort aanlegjaar
## 1  63 2857      SAALBA      Schietwilg      Salix alba      1990
## 2 356 2880      SAALBA      Schietwilg      Salix alba      2000
## 3 451 3349      SAALBA      Schietwilg      Salix alba      1970
## 4 455 1952      BEPENDUL      Ruwe berk      Betula pendula      1980
## 5 749 3445      TICORDAT Kleinbladige linde      Tilia cordata      1985
## 6 847 1840      LISTYRAC      Sassafras Sassafras albidum      2012
##      inspectiedatum inspectiedag inspectiemaand inspectiejaar
## 1      30-07-2014           30           07           2014
## 2      31-07-2014           31           07           2014
## 3      06-08-2014           06           08           2014
## 4      07-08-2014           07           08           2014
## 5      05-08-2014           05           08           2014
## 6      08-08-2014           08           08           2014
##      kl_veilig_maatregelen structuurelement woonplaats      wijk      buurt
## 1      Rooien;      Woongebied      Boskoop Boskoop Snijdelwijk
## 2      Rooien;      Woongebied      Boskoop Boskoop Snijdelwijk
## 3      Rooien;      Woongebied      Boskoop Boskoop      Boskoop
## 4      Rooien;      Woongebied      Boskoop Boskoop Snijdelwijk
## 5      Rooien;      Woongebied      Boskoop Boskoop      Boskoop
## 6      Rooien;      Woongebied      Boskoop Boskoop Snijdelwijk
##      openbare_ruimte      x      y
## 1      Lage weide 103890,645 453625,156
## 2      Emmakade 104931,9 453665,223
## 3      Weteringpad 103938,935 454361,435
## 4      Klaverblad 103958,989 453882,047
## 5      Reyerskoop 105583,114 454224,553
```

6 Wilgehout 103800,245 453831,052

Vragenset 2B

1. Maak een subset van de data waarin alleen de bomen staan die in 2014 in de Horstenbuurt aangemerkt werden om gerooid te worden.
2. Van welke boomsoort zullen de meeste bomen worden gerooid in deze buurt?
3. Maak een subset van alle bomen die **sinds** 2010 geïnspecteerd zijn.

Analyseren

We gaan data aggregeren om een beter beeld te krijgen van het inspectieproces bij de Gemeente Alphen a/d Rijn. Met de `table()` functie kunnen we een matrix bouwen. Horizontaal (*rij*) plaatsen we de afzonderlijke *woonplaatsen* en verticaal (*kolom*) de *inspectiejaren*. In de cellen van de matrix staat het aantal keren dat de gegeven combinatie (*woonplaats, jaar*) in de dataset voorkomt.

```
## Aantal inspecties per woonplaats per jaar
inspectTab <- table(openDF$woonplaats, openDF$inspectiejaar)
inspectTab
##
##           2003  2005  2006  2007  2008  2009  2010  2011
##           0     0     0     0     0     0     0     0
## Aarlanderveen  0     0     0    56     2    207     2     0
## Alphen aan den Rijn  0 2253 1643 1043    243    980    374  4510
## Benthuizen     0     0     0     0     0     0     8     0
## Boskoop        0     0     0     0     0    132     0     0
## Hazerswoude-Dorp  0     0     0     0     0     4    288     0
## Hazerswoude-Rijndijk  0     0     1     0     0     0    14     0
## Koudekerk aan den Rijn  0     0     0     0     0     0     0     0
## Zwammerdam     50    30     0     1     0    142     4     0
##
##           2012  2013  2014  2015
##           0     0     0     1
## Aarlanderveen  767     4    11     0
## Alphen aan den Rijn 11983 1510 6457     0
## Benthuizen     0     0  1617     0
## Boskoop        0     0  5235     0
## Hazerswoude-Dorp  0     0  1691     0
## Hazerswoude-Rijndijk  0     0  1907     0
## Koudekerk aan den Rijn  0     0  1444     0
## Zwammerdam     589    27    90     0
str(inspectTab)
## 'table' int [1:9, 1:12] 0 0 0 0 0 0 0 0 50 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:9] "" "Aarlanderveen" "Alphen aan den Rijn" "Benthuizen"
## ..$ : chr [1:12] "2003" "2005" "2006" "2007" ...
```

Als we de structuur van `inspectTab` bekijken zien we dat het class `table` heeft en bestaat uit de waarden (`int`) en namen van de *rijen* en *kolommen* (`attr`). Om dit om te zetten naar een data frame kunnen we de functie `as.data.frame.matrix()` gebruiken. Wat we zo direct nodig hebben, zijn de de *namen* van de afzonderlijke *rijen*. Deze kunnen we via de functie `rownames()` opslaan in een vector.

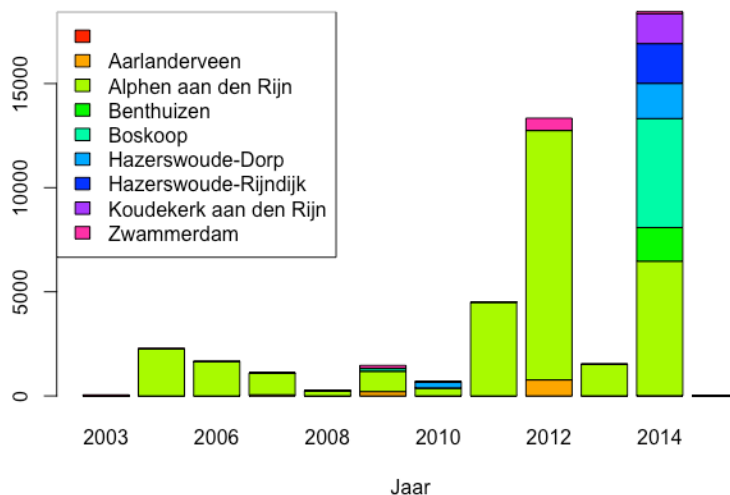
```
## Conversie tabel naar data frame
inspectDF <- as.data.frame.matrix(inspectTab)
head(inspectDF)
##           2003  2005  2006  2007  2008  2009  2010  2011  2012  2013
##           0     0     0     0     0     0     0     0     0     0
## Aarlanderveen  0     0     0    56     2    207     2     0    767     4
## Alphen aan den Rijn  0 2253 1643 1043    243    980    374  4510 11983 1510
## Benthuizen     0     0     0     0     0     0     8     0     0     0
## Boskoop        0     0     0     0     0    132     0     0     0     0
## Hazerswoude-Dorp  0     0     0     0     0     4    288     0     0     0
##           2014  2015
```

```
##              0      1
## Aarlanderveen      11      0
## Alphen aan den Rijn 6457      0
## Benthuisen        1617      0
## Boskoop           5235      0
## Hazerswoude-Dorp   1691      0
## Bewaar rijnamen
plaatsenVec <- rownames(inspecDF)
cat(plaatsenVec, sep = ", ")
## , Aarlanderveen, Alphen aan den Rijn, Benthuisen, Boskoop, Hazerswoude-
Dorp, Hazerswoude-Rijndijk, Koudekerk aan den Rijn, Zwammerdam
```

We kunnen nu de data in een staafdiagram weergeven m.b.v. de functie `barplot()`. De functie `barplot` accepteert als input alleen een matrix of een vector. Daarom gebruiken we de tabel in plaats van de data frame.

```
## Barplot van aantal inspecties per woonplaats per jaar
barplot(inspecTab,
       xlab = "Jaar",
       col=rainbow(9)
)

legend("topleft",
      legend = plaatsenVec,
      fill=rainbow(9)
)
```



Vragenset 2C

1. Wat valt op als je de `inspecTab` dataset bekijkt?
2. Maak een tabel waarin je het aantal afzonderlijke categorieën veiligheidsmaatregelen per jaar telt.
3. Voor hoeveel bomen werd in 2014 een jaarlijkse inspectie geadviseerd?
4. Analyseer deze [data](#)

Sessie 3 - Visualiseren / Presenteren

In deze sessie kijken we hoe je data aantrekkelijk kunt visualiseren en op een makkelijke manier toegankelijk maakt voor de gebruikers.

Visualiseren

Er zijn talrijke packages voor R gebouwd waarmee je data op veelzijdige manieren grafisch kunt weergeven. In deze workshop gaan we werken met de R interface voor de Google Charts API (`googleVis`). Op deze [pagina](#) vind je voorbeelden van de mogelijkheden die de interface biedt.

```
## Laad de benodigde packages
library(googleVis)
library(gsheet)
library(dplyr)
```

De data die we gebruiken, komt uit een Google Spreadsheet.

```
## Lees data uit Google Spreadsheet
url <- 'https://docs.google.com/spreadsheets/d/1f0qX_BBu-
4qhKHa6mBiwbUHmUahrXcHeWieBDD9fiXg'
surveyDF <- gsheets2tbl(url)
surveyDF <- rename(surveyDF, Soort.Risico = `Soort Risico`)
```

Met de functie `gvisBubbleChart()` wordt alle code gegenereerd die nodig is om de grafiek in een html pagina weer te geven. Met de `plot()` functie wordt de grafiek in een webbrowser getoond.

```
## Bouw Google Bubble Chart
survChart <- gvisBubbleChart(surveyDF,
                             idvar = "Naam",
                             xvar = "Waarschijnlijkheid",
                             yvar = "Gevolg",
                             colorvar = "Soort.Risico",
                             options = list(hAxis = '{minValue:1,
maxValue:5, title:"Gevolg"}',
                                           ,vAxis = '{minValue:1,
maxValue:5, title:"Waarschijnlijkheid"}',
                                           width = 700,
                                           height = 700
                                           )
plot(survChart)
KansBedreigingELINELINNIKIANIKIAMARIOMARIOROMAINEROMAINEAALLYSONALLYSONVADAVADAMYR
LEMYRLEVERTIEVERTIEDAISYDAISYKATHERINEKATHERINESIDSIDLERALERAMARLYSMARLYSANTON
ANTONCHRISTENCHRISTENKYLEEKYLEEKAIKAINELLNELLKIRSTIEKIRSTIETHERESIATHERESIAMOSESM
OSESBYRYANTBRYANTLEZLIELEZLIESANDISANDIARNETTAARNETTALAVINALAVINAIGNACIAIGNACIAA
RLETHAARLETHAWILTONWILTONALTHAALTHAGEORGIANNGEORGIANNEMMETTEMMETTMAMIEMIEMAMI
EROXANAROXANAJERRYJERRYMARINEMARINEHONGHONGRHETTRHETTTDEDRADEDDRAMALIKMALIKKE
LSEYKELSEYSHANDISHANDIHERMINAHERMINACATHARINECATHARINESHERRILSHERRILSHAWANASHA
WANAALICAALICASHARENSHARENCAROLEECAROLEEYASUKOYASUKOFUMIKOFUMIKOSHONSHONMO
NNIEMONNIEBETTYEBETTYESUESUEJAREDJAREDLAURILAUWFAWNFAWN0,01,53,04,56,00,01,53,04,56,0Gev
olgWaarschiijnlijkheid
```


Naam	Waarschijnlijkheid	Gevolg	Soort.Risico
ELIN	2	4	Kans
NIKIA	4	5	Kans
MARIO	3	5	Kans
ROMAINE	3	3	Kans
ALLYSON	4	2	Kans
VADA	2	4	Kans
MYRLE	1	4	Kans
VERTIE	3	3	Kans
DAISY	3	3	Kans
KATHERINE	4	4	Kans
SID	4	3	Kans
LERA	4	4	Kans
MARLYS	3	4	Kans
ANTON	5	3	Bedreiging
CHRISTEN	3	4	Bedreiging
KYLEE	4	4	Bedreiging
KAI	2	4	Bedreiging
NELL	3	4	Bedreiging
KIRSTIE	4	4	Bedreiging
THERESIA	4	4	Bedreiging
MOSES	3	5	Bedreiging
BRYANT	4	4	Bedreiging
LEZLIE	4	4	Bedreiging
SANDI	4	4	Bedreiging
ARNETTA	4	4	Bedreiging
LAVINA	2	4	Bedreiging
IGNACIA	4	4	Bedreiging
ARLETHA	4	3	Bedreiging
WILTON	4	4	Bedreiging
ALTHA	4	4	Bedreiging
GEORGIANN	5	5	Bedreiging
EMMETT	2	5	Bedreiging
MAMIE	4	4	Bedreiging
ROXANA	4	4	Bedreiging
JERRY	4	4	Bedreiging
MARINE	2	5	Bedreiging
HONG	3	4	Bedreiging
RHETT	2	2	Bedreiging
DEDRA	5	4	Bedreiging
MALIK	5	4	Bedreiging
KELSEY	3	4	Bedreiging
SHANDI	3	4	Bedreiging
HERMINA	4	4	Bedreiging
CATHARINE	3	4	Bedreiging
SHERRIL	2	5	Bedreiging
SHAWANA	4	4	Bedreiging
ALICA	3	3	Bedreiging
SHAREN	4	3	Bedreiging
CAROLEE	4	3	Bedreiging
YASUKO	4	3	Bedreiging
FUMIKO	4	4	Bedreiging

Naam	Waarschijnlijkheid	Gevolg	Soort.Risico
SHON	5	5	Bedreiging
MONNIE	4	5	Bedreiging
BETTYE	5	5	Bedreiging
SUE	4	4	Bedreiging
JARED	2	2	Bedreiging
LAURI	5	4	Bedreiging
FAWN	4	3	Bedreiging

Vragenset 3A

1. Wat voor een structuur heeft `survDF`?
2. Om wat voor een soort data gaat het?
3. Maak een nieuwe bubble chart met de variabelen die op “.1” eindigen.
4. Wat valt je op als je de twee bubble charts met elkaar vergelijkt?

Zoals je wellicht al is opgevallen, bevat `survDF` 7 sets van variabelen die een gelijke opzet hebben. Het gaat hier om een survey, waarbij deelnemers moesten reageren op stellingen. In de vorige exercitie hebben we de afzonderlijke reacties grafisch weergegeven. We gaan nu kijken naar de centrum- en spreidingsmaten. Voor het gemak zijn deze waarden al berekend en in dit [bestand](#) bewaard.

We gebruiken nu de package `plotly`. Voor het gebruik en voorbeelden zie [hier](#).

```
## Laad packages
library(plotly)

p <- plot_ly(meanvarDF,
  x = ~Waarschijnlijkheid,
  y = ~Gevolgen, text = ~Topic,
  type = 'scatter', mode = 'text+markers',
  color = ~Perc.Kans,
  colors = c("red", "orange", "white", "light blue" , "dodgerblue"),
  marker = list(size = ~Spreiding*50))

p
W&RICTHRTtoetredersRenteVerdienmodelVermogenverschaffers33.544.53.43.53.63.73.83.
944.1
WaarschijnlijkheidGevolgen4060Perc.Kans
```

Vragenset 3B

1. Wat voor een structuur heeft `meanvarDF`?
2. Analyseer de chart en beschrijf wat je opvalt.
3. Verander de chart, zodat in plaats van het thema het percentage van deelnemers wordt getoond dat aangaf het thema als een kans te zien.

In de volgende oefening gaan we data in een geografische kaart plaatsen. De data is via een mobiele telefoon verzameld en kan [hier](#) opgehaald worden. We gebruiken de [Leaflet](#) package.

```
## Laad packages
library(leaflet)
```

```
## Bepaal middelpunt van de kaart
latCtr <- mean(gpsDF$lat)
lonCtr <- mean(gpsDF$lon)

## Bouw de kaart en geef weer
m1 <- leaflet() %>%
  addProviderTiles(providers$OpenStreetMap) %>%
  addTiles() %>%
  setView(lonCtr, latCtr, zoom = 12) %>% # map location
  # add som circles:
  addCircles(color = "Red", lng=gpsDF$lon, lat=gpsDF$lat, 20)
m1
```

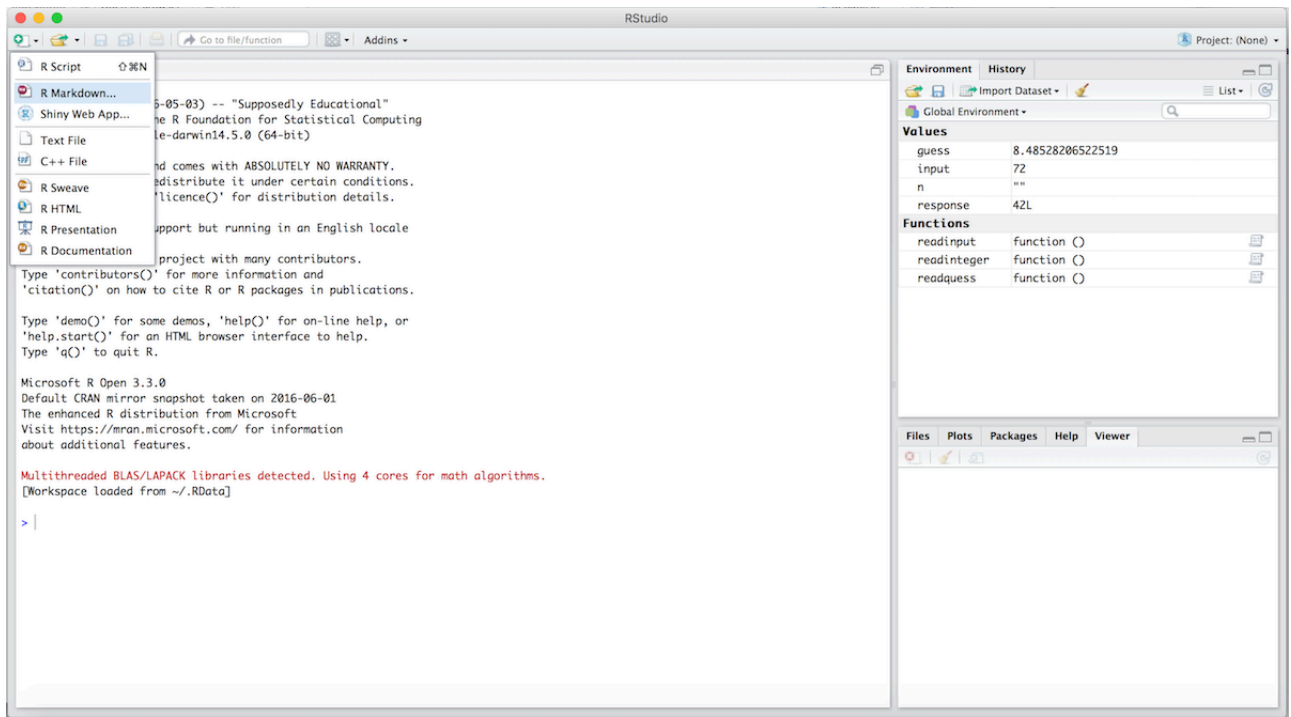
[+-](#)
[Leaflet](#) | © [OpenStreetMap](#) contributors, [CC-BY-SA](#)

Vragenset 3C

1. Welke data is door de mobiele telefoon verzameld en op welke momenten?
2. Waarom zijn sommige punten verder van elkaar verwijderd dan andere?
3. Gebruik deze [data](#) om een nieuwe geografische kaart te bouwen en analyseer de informatie.

Presenteren

Natuurlijk wil je uiteindelijk de uitkomsten van je werk met anderen delen. In R kun je heel gemakkelijk rapportages in verschillende formats (HTML, pdf of Word) creëren, waarin je tekst, code, tabellen en grafieken integreerd. Hiervoor open je een RMarkdown script.



Geef je document een titel en kies wat voor een soort document je wilt maken. In dit voorbeeld gaan we een webpagina bouwen en kiezen dus voor HTML.

R bouwt alvast een template met wat voorbeelden, zodat je direct aan de slag kunt. Als je wilt testen hoe het document er uit gaat zien, klik je op *Knit HTML*.



In het bovenste gedeelte van het *RMarkdown script* zie je een zogenaamd YAML veld waarin meta data staat. Je kunt deze veranderen en extra variabelen toevoegen (bijvoorbeeld een inhoudsopgave). Online is veel informatie te vinden over het gebruik van YAML en (R) Markdown. Voor RMarkdown vind je onder het helpmenu van RStudio ook een *Cheatsheet* en een handleiding.

```
---
title: "Rapportage Q1 2016"
author: "Nomen Nescio"
date: "17-08-1016"
output:
  html_document:
    toc: yes
---
```

```

Deze rapportage gaat over de subsidie-uitkeringen van de Gemeente
Purmerend.
```{r, message=FALSE, warning=FALSE}

library(knitr)
library(dplyr)

subsDF <- read.csv(file = "https://raw.githubusercontent.com/witusj/R-
workshop/gh-pages/Datasets/sessie%202/subs_data.csv", stringsAsFactors =
TRUE)
glimpse(subsDF)

Tabel waarde subsidies per jaar per categorie
subsTable <- xtabs(Bedrag ~ Categorie + Jaar, subsDF)
subsDF <- as.data.frame.matrix(subsTable)
subsDF <- round(subsDF, 0)
kable(subsDF, caption = "Tabel 1. Waarde subsidies per jaar per categorie")

Staafdiagram
barplot(subsTable,
 main="Subsidiebedragen per categorie",
 xlab="Jaar",
 ylab="Bedrag",
 col=rainbow(5)
)

legend("topright",
 legend = dimnames(subsTable)$Categorie,
 fill=rainbow(5)
)
```

```

Als het document volledig naar wens is, kun je het snel publiceren op internet via [Rpubs](#). Klik hiervoor op het ‘blauwe oogje’.



en volg de instructies. De [link](#) naar je document op Rpubs kun je delen met betrokkenen en bijvoorbeeld integreren in het intranet van je organisatie of toevoegen aan een mail. LET OP: de informatie op Rpubs is niet afgeschermd en zichtbaar voor iedereen die het webadres heeft.