Constructing data frames (Gegevensframes construeren)

Constructing a data frame

In R-programmering is een "gegevensframe" een tweedimensionale structuur die gegevens opslaat in een tabelformaat van cel-rijen en cel-kolommen. In tegenstelling tot matrixstructuren hoeven de waarden die in een gegevensframe zijn opgeslagen niet allemaal van hetzelfde gegevenstype te zijn - ze kunnen waarden van elk gegevenstype bevatten. Dit betekent dat dataframes bijzonder veelzijdig zijn en de meest bruikbare datastructuur in R-programmering zijn.

Een dataframe kan in R Script worden gemaakt door een unieke identificatienaam naar keuze in de Code Editor te schrijven en vervolgens waarden toe te wijzen met behulp van de ingebouwde **data.frame()**-functie. Deze functie vereist vectoren met waarden als argumenten. Elke vector moet dezelfde lengte hebben, anders worden de waarden in extra cellen gerecycleerd om overeen te komen met de lengte van de langste vector.


Vergeet niet om het puntteken op te nemen in de functienamen **dataframe()** en **is.data.frame()**.

De waarden in elke vector verschijnen in afzonderlijke kolommen van het gegevensframe en elke kolom krijgt standaard de naam van de overeenkomstige vector als kolomnaam. Rijnamen worden standaard in oplopende volgorde vanaf één genummerd. Net als bij matrices kan de rangschikking van rijen en kolommen van dataframes worden getransponeerd met behulp van de functie **t()** - om rijen naar kolommen te schakelen.

Als je liever je eigen namen voor dataframe rijen en -kolommen opgeeft, kunnen deze worden toegewezen met behulp van de functies **rownames()** en **colnames()** - net als bij matrices. Je kunt ook een argument **row.range=** opnemen in de aanroep van de functie **data.frame()** om een lijst met namen voor de dataframe rijen op te geven.

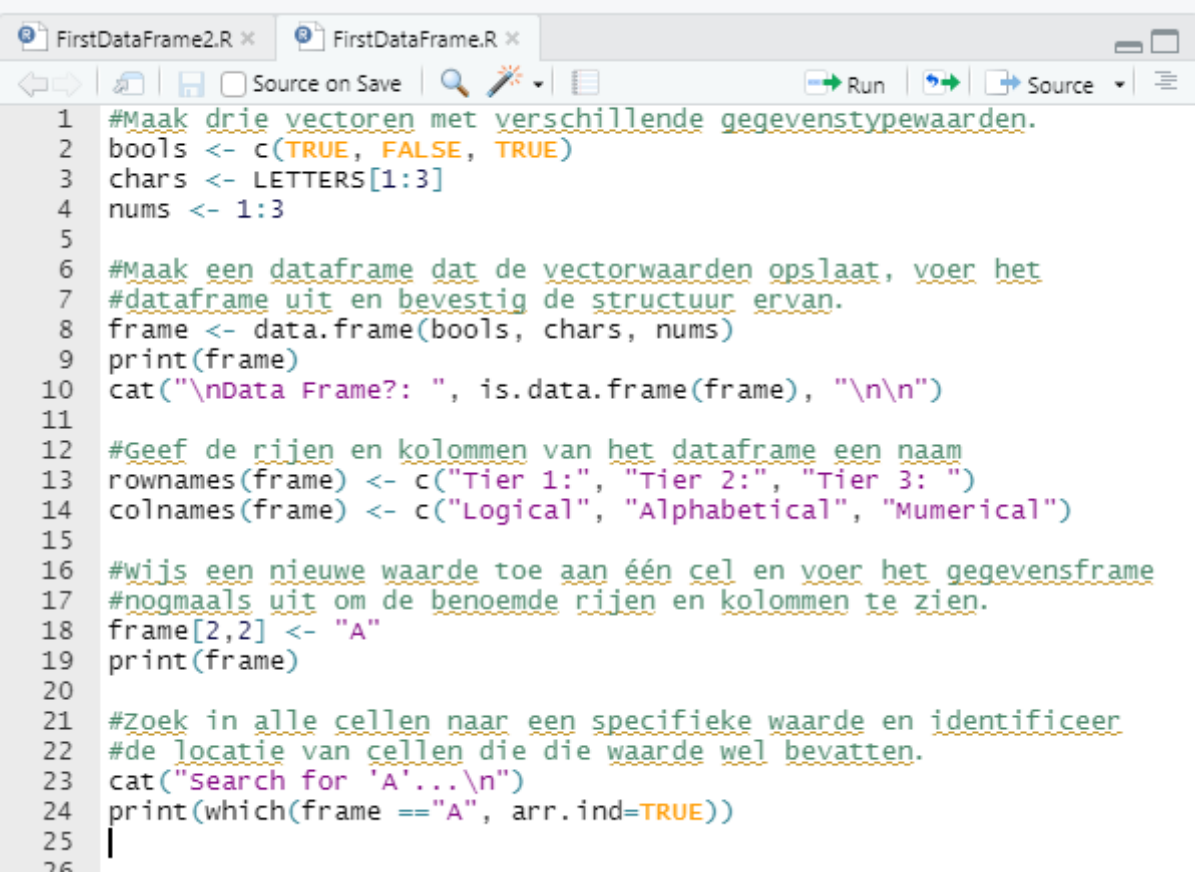
Individuele waarden worden geadresseerd in een dataframe met behulp van de juiste indexnummers van de rij en kolom. Nieuwe waarden kunnen ook aan individuele cellen worden toegewezen met behulp van hun indexnummer van rij en kolom, maar let op het gegevenstype van die cel.

Om te bevestigen dat een structuur inderdaad een dataframe is, biedt R een ingebouwde functie **is.data.frame()** die alleen **TRUE** retourneert als het opgegeven argument een dataframe-object is.

Het is handig om een waarde te zoeken met in een gegevensframe met behulp van de ingebouwde functie **who()** om een logisch argument op te geven en optioneel kunt u een argument **arr.ind=TRUE** opnemen om het rij- en kolom-indexnummer van elke cel te retourneren met daarin de gezochte waarde.

1 Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

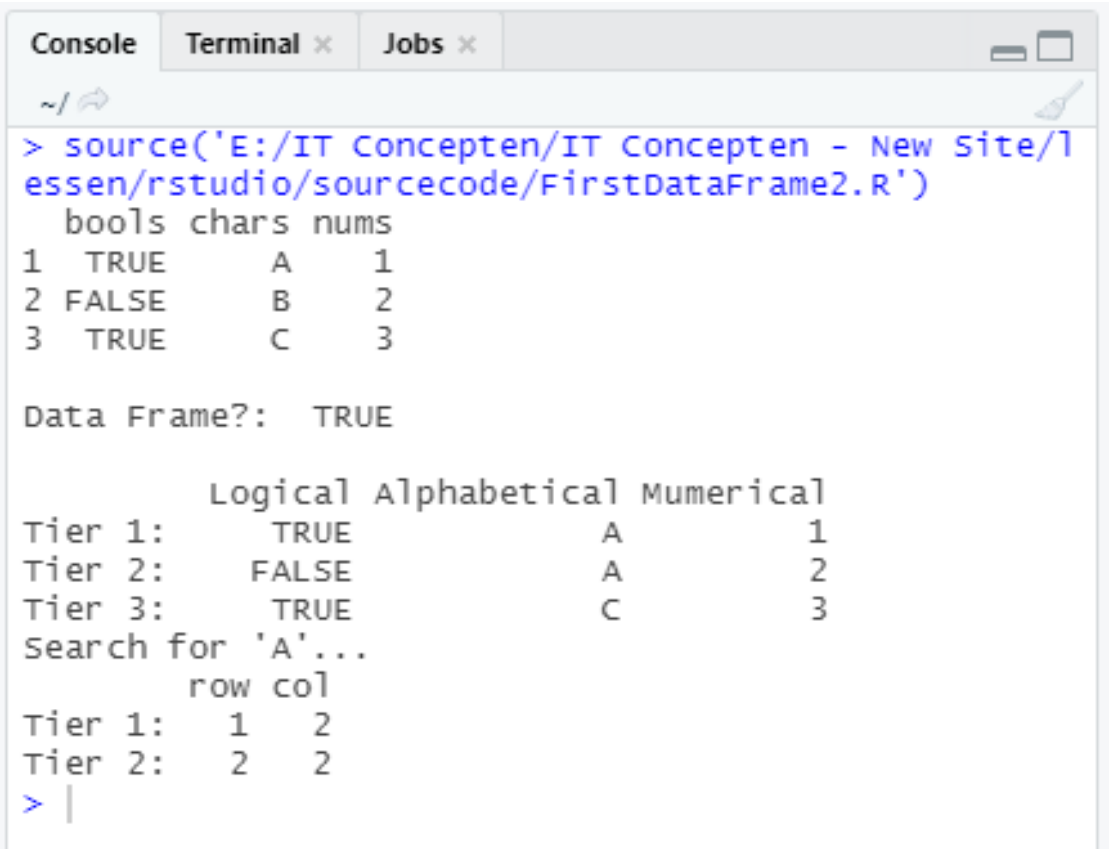
2 Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.




```
#Maak drie vectoren met verschillende gegevenstypewaarden.
bools <- c(TRUE, FALSE, TRUE)
chars <- LETTERS[1:3]
nums <- 1:3

#Maak een dataframe dat de vectorwaarden opslaat, voer het
#dataframe uit en bevestig de structuur ervan.
frame <- data.frame(bools, chars, nums)
print(frame)
cat("\nData Frame?: ", is.data.frame(frame), "\n\n")

#Geef de rijen en kolommen van het dataframe een naam
rownames(frame) <- c("Tier 1:", "Tier 2:", "Tier 3: ")
colnames(frame) <- c("Logical", "Alphabetical", "Mumerical")

#Wijs een nieuwe waarde toe aan één cel en voer het gegevensframe
#nogmaals uit om de benoemde rijen en kolommen te zien.
frame[2,2] <- "A"
print(frame)

#Zoek in alle cellen naar een specifieke waarde en identificeer
#de locatie van cellen die die waarde wel bevatten.
cat("Search for 'A'...\n")
print(which(frame =="A", arr.ind=TRUE))
```

Als alternatief kunnen de rij-namen worden toegewezen aan een argument **row.names=** in de aanroep van de functie **data.frames()**.

3 Sla het **R Script** op met de naam **FirstDataframe.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.



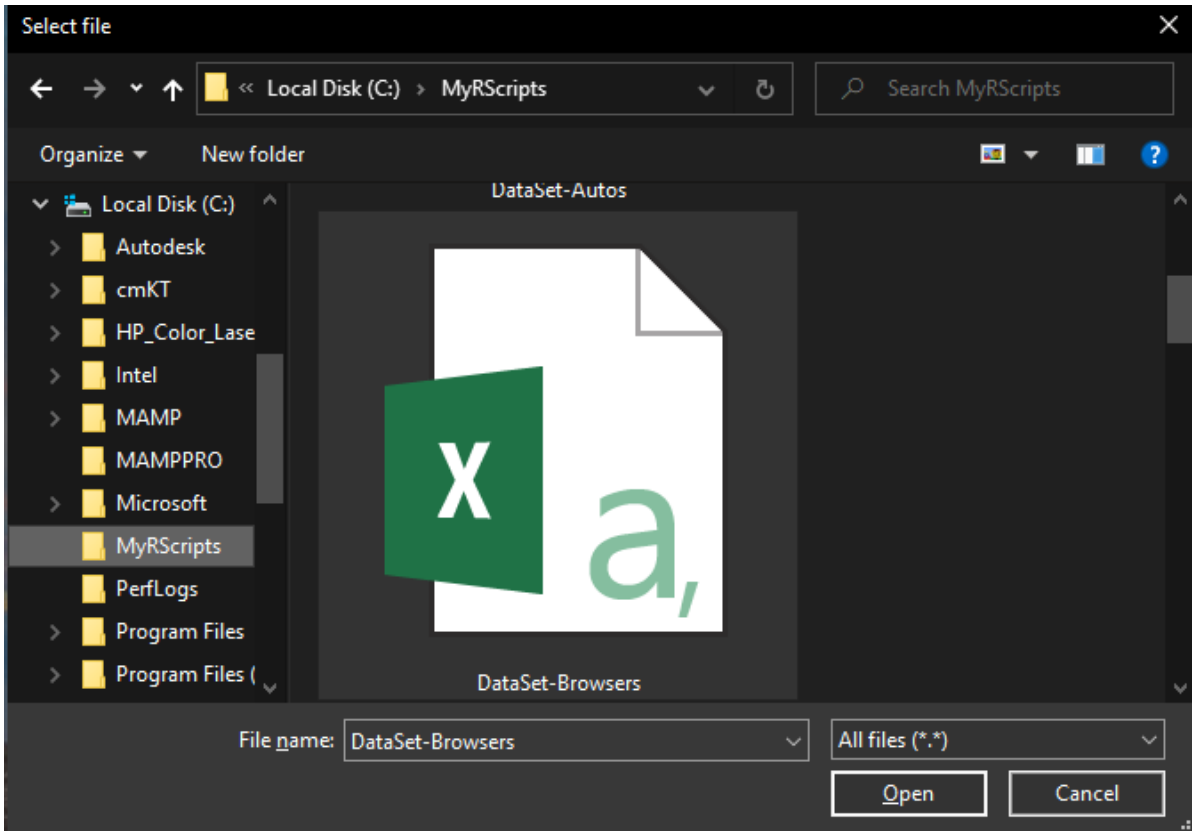
Importing data sets

Verzamelingen van gegevens in tabelvorm worden vaak opgeslagen als een "***data set***" (gegevensset) in een bestand met door komma's gescheiden waarden (CSV = Comma-Separated Values). Deze datasets kunnen eenvoudig worden geïmporteerd in **RStudio** en hun gegevens kunnen worden gekopieerd naar een dataframe met behulp van de ingebouwde functie **read.csv( )**. Deze functie vereist eenvoudigweg het CSV-bestands-pad als argument.

Het pad naar een CSV-bestand kan worden opgegeven voor de functie **read.csv()** door de functie **file.choose()** als argument op te geven. Hierdoor wordt een dialoogvenster "Bestand selecteren" geopend waarmee u naar de locatie van het CSV-bestand kunt bladeren. Eenmaal geselecteerd, wordt het pad geleverd aan de functie **read.csv()** zodat de gegevens naar een gegevensframe kunnen worden gekopieerd. Het instructie **frame <- read.csv(file.choose())** biedt bijvoorbeeld een dialoogvenster om een CSV-bestand te selecteren waarvan de gegevens vervolgens worden gekopieerd naar een gegevensframe met de naam "*frame*".

Als alternatief kan het volledige pad van een CSV-bestand worden opgegeven voor de functie **read.csv()** als argument. Deze instructie

geeft bijvoorbeeld het volledige pad naar het geselecteerde CSV-bestand dat hieronder wordt geïllustreerd:



frame <- read.csv("C:/MyRScripts/DataSet-Browsers.csv")

Volledige pad-namen kunnen lang zijn, dus het is handig om de locatie van het CSV-bestand in te stellen als de werkdirectory van **RStudio** door de directory op te geven als argument voor de functie **setwd()**. De functie **read.csv()** vereist dan alleen de CSV-bestandsnaam als argument. U kunt de huidige werkdirectory op elk moment vinden door de ingebouwde **getwd()**-functie aan te roepen.

JE kunt de standaard werkmap in **RStudio** instellen door op **Tools**, **Global Options** en de optie **General** te klikken.

Paden kunnen de voorwaartse slash **/** teken of escaped backslash gebruiken, zoals **C:\\MyRScripts\\DataSets-Browser.csv**

1 Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2 Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

3 Sla het **R Script** op met de naam **ImportData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.

```
Console  Terminal  Jobs
C:/MyRScripts/
> source('E:/IT Concepten/IT Concepten - New Site/lessen/rstudio/sourceco
de/ImportData2.R')
Working Directory: C:/MyRScripts

                Web.Browser.Version PerCentage.Market.Share
1                       Chrome 61.0                   35.16
2  Microsoft Internet Explorer 11.0                   12.52
3                       Chrome 55.0                    7.40
4                        Firefox 56                    6.53
5                  Microsoft Edge 14                    3.60
6                       Chrome 49.0                    2.26
7                       Chrome 60.0                    2.19
8                       Chrome 45.0                    1.86
9                       Chrome 62.0                    1.76
10                      Safari 11.0                    1.70
11                      Chrome 50.0                    1.59
12                 Microsoft Edge 15                   1.43
13                      Chrome 58.0                    1.25
14                       Firefox 55                    1.24
15  Microsoft Internet Explorer 8.0                    1.12
16                       Firefox 52                    1.06
17                      Chrome 53.0                    0.93
18                      Safari 10.0                    0.85
19                         Opera 48                    0.85
20                       Firefox 53                    0.83
21  Microsoft Internet Explorer 9.0                    0.81
22                       Firefox 40                    0.76
23                      Safari 10.1                    0.66
24                 Microsoft Edge 13                   0.65
25                      Chrome 47.0                    0.64
26                      Chrome 59.0                    0.56
27 Microsoft Internet Explorer 10.0                    0.54
28                       Safari 8.0                    0.52
29                      Chrome 56.0                    0.49
30                       Firefox 57                    0.47
31                      Chrome 57.0                    0.47
32                      Chrome 43.0                    0.41
33                 Microsoft Edge 12                   0.33
34                      Chrome 60.4                    0.28
35                       Safari 9.1                    0.26
36                       Firefox 50                    0.21
37                       Firefox 54                    0.20
38                      Firefox 8.0                    0.19
39                       Firefox 47                    0.19
40                      Chrome 54.0                    0.17
>
```

Hier worden de kolom-namen geleverd door de eerste regel van het CSV-bestand, maar de rijen worden automatisch genummerd door de R-interpreter. Merk op dat er drie gevallen van identiek marktaandeel zijn.

Examining data frames

De programmeertaal **R** biedt een aantal functies voor het onderzoeken van data frame structuren:

- **nrow()**
  - retourneert een geheel getal dat het totale aantal rijen is binnen het gegevensframe dat is opgegeven als argument.
- **ncol()**
  - retourneert een geheel getal dat het totale aantal kolommen is binnen het gegevensframe dat is opgegeven als argument.
- **head()**
  - retourneert standaard de bovenste zes rijen van het dataframe dat is opgegeven als zijn argument, plus de kolom- en rijnamen. Optioneel kan een argument **n=** worden toegevoegd om aan te geven hoeveel rijen moeten worden geretourneerd. **n=3** retourneert bijvoorbeeld de bovenste drie rijen.
- **tail()**
  - retourneert standaard de onderste zes rijen van het dataframe dat is opgegeven als argument, plus kolom- en rijnamen. Een argument **n=** kan worden toegevoegd om het aantal te retourneren rijen op te geven. **n=3** retourneert bijvoorbeeld de onderste drie rijen.
- **str()**
  - geeft een overzicht weer van de structuur van het dataframe dat als argument is opgegeven - met een lijst van het totale aantal objecten en variabelen dat het bevat, samen met het aantal unieke factoren *(**factors**, zijn unieke waarden)* in elke kolom. Intern worden de factoren numeriek gerangschikt in aflopende volgorde, waarbij de hoogste rangorde niveau 1 is.
- **summry()**
  - geeft een samenvatting weer van de inhoud van het dataframe dat is opgegeven als argument. Standaard worden zes niveaus van factoren weergegeven, maar een optioneel **maxsum=**-argument kan worden toegevoegd om te specificeren hoeveel niveaus moeten worden weergegeven. Voor kolommen die numerieke gegevens bevatten, biedt de samenvatting statistieken:
    - **Minimum** - het laagste getal in de kolom.
    - **1e kwartiel** (1st Quartile) - de middelpuntwaarde tussen het minimumaantal en de mediaanwaarde.
    - **Mediaan** (Median) - de middelste waarde van de kolom.
    - **Gemiddelde** (Mean) - de gemiddelde waarde van de getallen in de kolom.

- **3e kwartiel** (3rd Quartile) - de middelpuntwaarde tussen de mediaanwaarde en het maximale aantal.
- **Maximum** - het hoogste getal in de kolom.

1 Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2 Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

ExamineData.R* ×

Source on Save | Run | Source

```
1  #Stel de werkmap in
2  setwd("C:/MyRScripts")
3
4  #Maak een gegevensframe dat gegevens importeert uit een
5  #CSV-bestand dat zich in de werkmap bevindt.
6  frame <- read.csv("DataSet-Browsers.csv")
7
8  #Geef het totale aantal rijen en kolommen weer.
9  cat("Rows: ", nrow(frame), "\tColumns: ", ncol(frame))
10
11 #Voer de eerste en laatste drie rijen en kolommen uit
12 cat("\nHead...\n")
13 print(head(frame, n=3))
14 cat("\nTail...\n")
15 print(tail(frame, n=3))
16
17 #Toon de structuur en samenvatting
18 cat("\nStructure...\n")
19 print(str(frame))
20 cat("\nSummary...\n")
21 print(summary(frame))
22
```

3 Sla het **R Script** op met de naam **ExaminData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.

Het is handig om te controleren of het totale aantal geïmporteerde rijen en kolommen overeenkomt met wat wordt verwacht van de bron-gegevens-set.

In R-programmering verwijst de functie str() naar "structuur", in tegenstelling tot andere programmeertalen waarin het verwijst naar "string".

```
Console   Terminal ×   Jobs ×
C:/MyRScripts/
> source('E:/IT Concepten/IT Concepten - New Site/lessen/rstudio/
sourcecode/ExamineData2.R')
Rows:  40       Columns:  2
Head...
                Web.Browser.Version PerCentage.Market.Share
1                    Chrome 61.0                    35.16
2 Microsoft Internet Explorer 11.0                  12.52
3                    Chrome 55.0                     7.40

Tail...
   Web.Browser.Version PerCentage.Market.Share
38         Firefox 8.0                    0.19
39          Firefox 47                    0.19
40         Chrome 54.0                    0.17

Structure...
'data.frame':   40 obs. of  2 variables:
 $ Web.Browser.Version    : chr  "Chrome 61.0" "Microsoft Interne
t Explorer 11.0" "Chrome 55.0" "Firefox 56" ...
 $ PerCentage.Market.Share: num  35.16 12.52 7.4 6.53 3.6 ...
NULL

Summary...
 Web.Browser.Version PerCentage.Market.Share
 Length:40           Min.    : 0.170
 Class :character    1st Qu.: 0.470
 Mode  :character    Median : 0.820
                     Mean    : 2.373
                     3rd Qu.: 1.617
                     Max.    :35.160
> |
```

Hier bevat de eerste kolom gegevens van het gegevenstype karakter, terwijl de tweede kolom gegevens van het gegevenstype numeriek dubbel bevat - dus statistieken worden voor die kolom in de samenvatting gegeven.

Addressing frame data

Er zijn een aantal manieren om gegevens in een dataframe aan te pakken. Net als bij matrices kunnen cellen worden geadresseerd door hun rij- en kolom index nummer tussen [] vierkante haken te vermelden. Als alternatief kunnen cellen worden geadresseerd door hun rij- en kolomnaam tussen [] vierkante haken te vermelden. Bovendien kunnen dataframes **de $ dollar-operator** gebruiken, zodat een kolom kan worden geadresseerd met deze handige syntaxis:

data.frame.name$column.name

Evenzo kunnen cellen worden geadresseerd door [] vierkante haken in de bovenstaande syntaxis waarin rij-indexnummers moeten worden opgegeven. De factorniveaus van een kolom kunnen worden

geadresseerd door de kolom op te geven als argument voor de ingebouwde functie **levels()** met dezelfde syntaxis:

1. Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2. Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

AddressData.R

Source on Save | Run | Source

```
#Stel de werkmap in
setwd("C:/MyRScripts")

#Maak een gegevensframe dat gegevens importeert uit een
#CSV-bestand dat zich in de werkmap bevindt.
frame <- read.csv("DataSet-Browsers.csv")

#Voer de eerste drie rijen en kolommen uit.
cat("\nHead...\n")
print(head(frame, n=3))

#Adresseer een enkele cel per rij- en kolomindexnummer.
data <- frame[ 1, 2]
cat("\nRow #1, Column #2:", data,"\n")

#Adresseer een enkele cel op rij-indexnummer en kolomnaam.
data <- frame[ 2,"Percentage.Market.Share"]
cat("\nRow #2, Column #2:", data,"\n")

#Adresseer een enkele cel op kolomnaam en rijindex.
data <- frame$Percentage.Market.Share[3]
cat("\nRow #3, Column #2:", data,"\n")

#Adresseer een hele kolom op naam om alle niveaus uit
#te voeren.
print(levels(factor(frame$Web.Browser.Version)))
```

3. Sla het **R Script** op met de naam **AddressData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.

```
Console   Terminal ×   Jobs ×
C:/MyRScripts/
> source('E:/IT Concepten/IT Concepten - New Site/lessen/rstudi
o/sourcecode/AddressData3.R')

Head...
                Web.Browser.Version PerCentage.Market.Share
1                       Chrome 61.0                   35.16
2 Microsoft Internet Explorer 11.0                   12.52
3                       Chrome 55.0                    7.40

Row #1, Column #2: 35.16

Row #2, Column #2: 12.52

Row #3, Column #2: 7.4
 [1] "Chrome 43.0"
 [2] "Chrome 45.0"
 [3] "Chrome 47.0"
 [4] "Chrome 49.0"
 [5] "Chrome 50.0"
 [6] "Chrome 53.0"
 [7] "Chrome 54.0"
 [8] "Chrome 55.0"
 [9] "Chrome 56.0"
[10] "Chrome 57.0"
[11] "Chrome 58.0"
[12] "Chrome 59.0"
[13] "Chrome 60.0"
[14] "Chrome 60.4"
[15] "Chrome 61.0"
[16] "Chrome 62.0"
[17] "Firefox 40"
[18] "Firefox 47"
[19] "Firefox 50"
[20] "Firefox 52"
[21] "Firefox 53"
[22] "Firefox 54"
[23] "Firefox 55"
[24] "Firefox 56"
[25] "Firefox 57"
[26] "Firefox 8.0"
[27] "Microsoft Edge 12"
[28] "Microsoft Edge 13"
[29] "Microsoft Edge 14"
[30] "Microsoft Edge 15"
[31] "Microsoft Internet Explorer 10.0"
[32] "Microsoft Internet Explorer 11.0"
[33] "Microsoft Internet Explorer 8.0"
[34] "Microsoft Internet Explorer 9.0"
[35] "Opera 48"
[36] "Safari 10.0"
[37] "Safari 10.1"
[38] "Safari 11.0"
[39] "Safari 8.0"
[40] "Safari 9.1"
>
```

Merk op hoe de niveaus automatisch worden aangemaakt in aflopende alfabetische en numerieke volgorde.

Extracting frame subsets

Subsets kunnen worden geëxtraheerd uit dataframes door hun rij- en kolomindex nummers tussen [] vierkante haken te vermelden - op dezelfde manier als subsets worden geëxtraheerd uit matrices. Gebruik bijvoorbeeld **frame[1:3, 2]** om een subset te extraheren die alleen de tweede kolom van de eerste drie rijen gegevensframe bevat, of **frame[1:3,]** om een subset te extraheren die alle kolommen van de eerste drie rijen bevat.

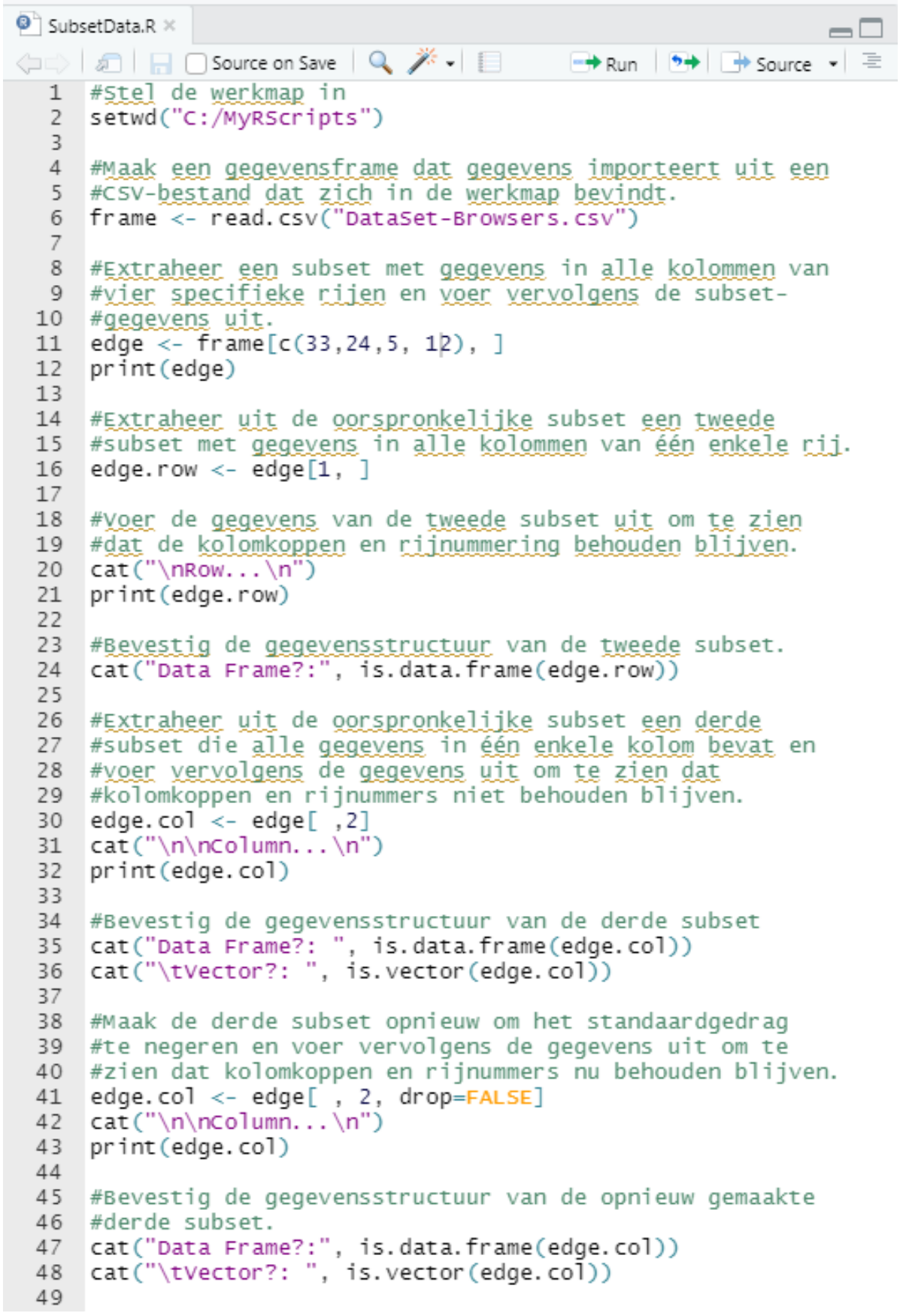
Als alternatief kan een subset van specifieke cellen worden geëxtraheerd met behulp van de bekende **c()**-functie. Als u bijvoorbeeld **frame[c(1, 3, 5), 2]** gebruikt om een subset te extraheren die alleen de tweede kolom van de eerste, derde en vijfde rij van een gegevensframe bevat, of **frame[c(1, 3, 5),]** om een subset te extraheren die alle kolommen van die specifieke rijen bevat.

Wanneer een subset een of meer rijen of meer dan één kolom extraheert uit een dataframe, worden de gegevens geretourneerd in een dataframe-object. Omgekeerd, wanneer een subset slechts één kolom extraheert, worden de gegevens standaard geretourneerd in een vectorobject. Het standaardgedrag kan worden omzeild door een laatste argument **drop=FALSE** op te nemen tussen de [] vierkante haken die de kolom specificeren die moet worden geëxtraheerd - zodat de gegevens nu worden geretourneerd in een vectorstructuur:

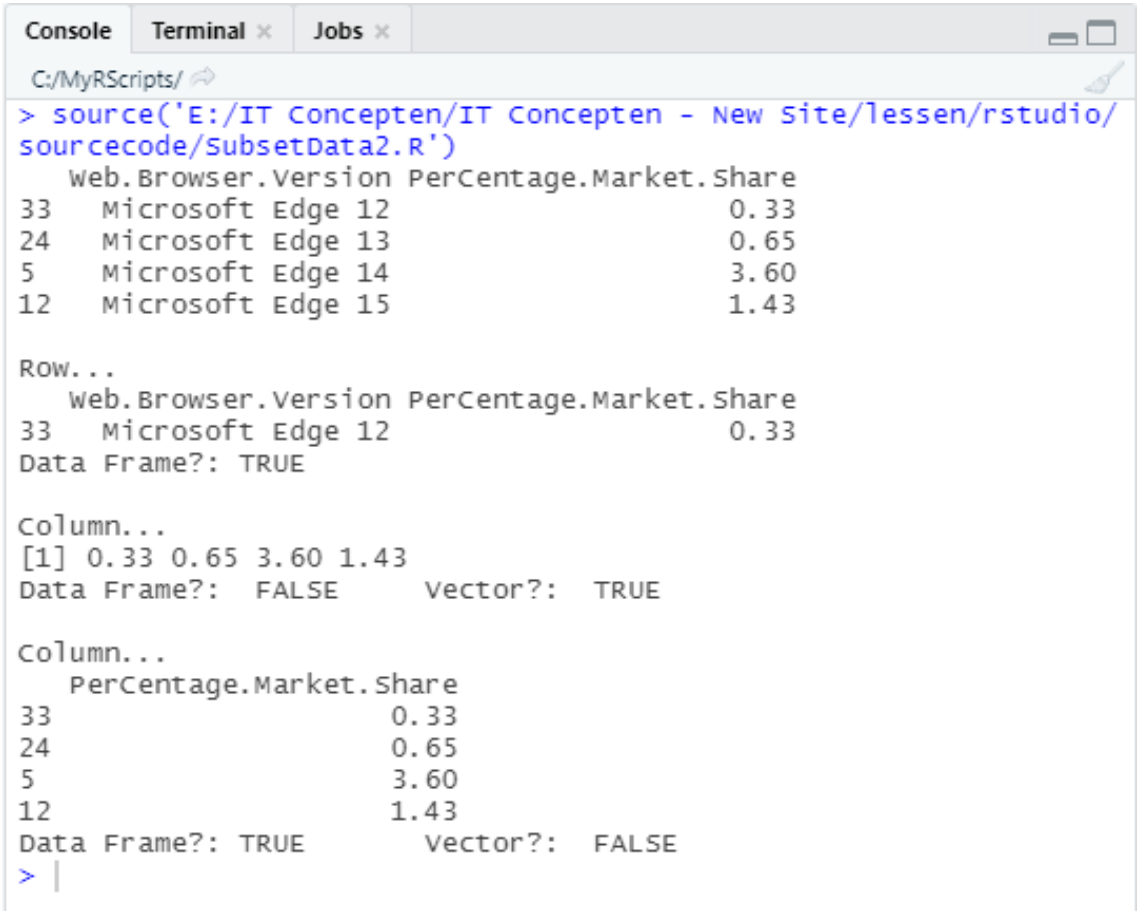
1. Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2. Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

Vergeet de extra spatie en komma binnen de vierkante haken niet om een hele rij of een hele kolom aan te duiden. Regels 16, 30 en 41.


SubsetData.R
Source on Save
Run
Source
#Stel de werkmap in
setwd("C:/MyRScripts")
#Maak een gegevensframe dat gegevens importeert uit een
#CSV-bestand dat zich in de werkmap bevindt.
frame <- read.csv("DataSet-Browsers.csv")
#Extraheer een subset met gegevens in alle kolommen van
#vier specifieke rijen en voer vervolgens de subset-
#gegevens uit.
edge <- frame[c(33,24,5, 12),]
print(edge)
#Extraheer uit de oorspronkelijke subset een tweede
#subset met gegevens in alle kolommen van één enkele rij.
edge.row <- edge[1,]
#Voer de gegevens van de tweede subset uit om te zien
#dat de kolomkoppen en rijnummering behouden blijven.
cat("\nRow...\n")
print(edge.row)
#Bevestig de gegevensstructuur van de tweede subset.
cat("Data Frame?:", is.data.frame(edge.row))
#Extraheer uit de oorspronkelijke subset een derde
#subset die alle gegevens in één enkele kolom bevat en
#voer vervolgens de gegevens uit om te zien dat
#kolomkoppen en rijnummers niet behouden blijven.
edge.col <- edge[,2]
cat("\n\nColumn...\n")
print(edge.col)
#Bevestig de gegevensstructuur van de derde subset
cat("Data Frame?: ", is.data.frame(edge.col))
cat("\tVector?: ", is.vector(edge.col))
#Maak de derde subset opnieuw om het standaardgedrag
#te negeren en voer vervolgens de gegevens uit om te
#zien dat kolomkoppen en rijnummers nu behouden blijven.
edge.col <- edge[, 2, drop=FALSE]
cat("\n\nColumn...\n")
print(edge.col)
#Bevestig de gegevensstructuur van de opnieuw gemaakte
#derde subset.
cat("Data Frame?:", is.data.frame(edge.col))
cat("\tVector?: ", is.vector(edge.col))

3 Sla het **R Script** op met de naam **SubsetData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.



Changing frame columns

De kolomnamen van gegevensframes die zijn gemaakt op basis van gegevens die zijn geïmporteerd uit een CSV-bestand, worden meestal overgenomen van de koptekstnamen die zijn opgegeven op de eerste regel van het CSV-bestand. Als u de headernamen liever negeert, kunt u een **header=FALSE**-argument toevoegen in de aanroep van de **read.csv()**-functie.

Kolomnamen kunnen worden toegevoegd aan naamloze kolommen, of bestaande kolomnamen kunnen worden gewijzigd door kolomnamen toe te wijzen aan de functie **colnames()**, zoals bij matrices.

Een nieuwe kolom kan aan een gegevensframe worden toegevoegd door cel-waarden als nieuwe kolomnaam toe te wijzen, zoals deze:

frame[“New Column”] <- 1:10

Als alternatief kan de **$** dollar-operator worden gebruikt om een nieuwe kolom toe te voegen, maar een kolomnaam die spaties bevat, moet worden ingesloten tussen ` **backtick**-tekens, zoals deze:


De ` Backtick-toets bevindt zich meestal linksboven op het toetsenbord, naast de cijfertoets 1.

frame$`New Column`<- 1:10

Een bestaande kolom kan eenvoudig uit een gegevensframe worden verwijderd door er een NULL-waarde aan toe te kennen:

frame$Existing.Column <- NULL

Waarden die aan een nieuwe kolom zijn toegewezen, kunnen worden gekopieerd uit een bestaande kolom en worden gemanipuleerd om nieuwe gegevenswaarden te verschaffen. Om bijvoorbeeld twee kolommen in een derde kolom te tellen:

frame$Total <- frame$Price + frame$Tax

Numerieke gegevens kunnen worden geconverteerd naar het tekengegevenstype voor tekenreeksmanipulatie door de numerieke gegevenswaarde op te geven als argument voor de ingebouwde functie **as.character()**. Het kan vervolgens worden samengevoegd met andere tekengegevens met behulp van de functie **paste()**. Standaard voegt de functie paste() automatisch spatietekens toe als scheidingstekens tussen aaneengeschakelde tekenreeksen. Dit gedrag kan echter worden overschreven door een alternatief scheidingsteken op te geven voor een optioneel **sep=**-argument voor de functie **paste()**. Als je liever geen scheidingsteken hebt tussen aaneengeschakelde tekenreeksen, kan dit argument helemaal geen tekens bevatten.


Rekenkundige en wiskundige bewerkingen kunnen alleen worden uitgevoerd op numerieke gegevenswaarden, zoals dubbele pr integer gegevenstypewaarden.

1. Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2. Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

ColumnData.R*

Source on Save | Run | Source

```
#Stel de werkmap in
setwd("C:/MyRScripts")

#Maak een gegevensframe dat gegevens importeert uit een
#CSV-bestand dat zich in de werkmap bevindt.
frame <- read.csv("DataSet-Browsers.csv")

#Maak een functie om een titel en de eerste twee regels
#van een dataframe uit te voeren - inclusief rij- en
#kolomnamen.
display <- function(title){
  cat("\n", title, "...\n")
  print(head(frame, n=2))
}

#Voer twee gegevensframerijen en -kolommen uit, hernoem
#vervolgens de bestaande kolommen en voer de wijzigingen
#uit.
display("Original Columns")
colnames(frame) <- c("Web.Browser", "PerCentage")
display("Renamed Columns")

#Kopieer bestaande numerieke kolomgegevens naar een nieuwe
#kolom - geconverteerd naar tekengegevens en aaneengeschakeld.
frame$Market.Share <-
  paste(as.character(frame$PerCentage), "%", sep="")

#Verwijder de kolom met numerieke gegevens en voer de
display(title) gen nogmaals uit.
frame$PerCentage <- NULL
display("Switch Columns")
```

3 Sla het **R Script** op met de naam **ColumnData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.

Het verwijderen van kolommen heeft invloed op hun indexnummer. Als u bijvoorbeeld de eerste kolom verwijdert, betekent dit dat de tweede kolom index nummer één aanneemt.

```
Console   Terminal ×   Jobs ×
C:/MyRScripts/
> source('E:/IT Concepten/IT Concepten - New Site/lessen/rstud
io/sourcecode/ColumnData2.R')

 Original Columns ...
              Web.Browser.Version PerCentage.Market.Share
1                  Chrome 61.0                    35.16
2 Microsoft Internet Explorer 11.0                    12.52

 Renamed Columns ...
                      Web.Browser PerCentage
1                  Chrome 61.0      35.16
2 Microsoft Internet Explorer 11.0      12.52

 Switch Columns ...
                      Web.Browser Market.Share
1                  Chrome 61.0       35.16%
2 Microsoft Internet Explorer 11.0       12.52%
>
```

Filtering data frames

De gegevens in de cellen van een gegevensframe kunnen worden gefilterd door een voorwaardelijke test uit te voeren op de waarde in elke cel, om slechts één gegevens te selecteren uit cellen waarvan de voorwaardelijke test **TRUE** retourneert.

Numerieke gegevens kunnen worden gefilterd door de waarde in elke cel te vergelijken met een waarde die is opgegeven in een voorwaardelijke test. Er kan bijvoorbeeld een filter worden gemaakt voor een kolom die numerieke gegevens bevat, om de pariteit (gelijkheid/overeenkomst) van gegevens in elke cel te onderzoeken en alleen even waarden te selecteren met dit instructie **filter <- frame[,1] %% 2== 0**. Het filter is gemaakt als een vector van Booleaanse waarden waarin even waarden worden weergegeven door **TRUE** en oneven waarden door **FALSE**. Als u het filter opneemt tussen de **[]** vierkante haken die een gebied van het gegevensframe specificeren, worden alleen cellen geselecteerd waarvan de gegevens de voorwaardelijke test doorstaan:

```
Console   Terminal ×   Jobs ×
~/
> frame <- data.frame(A=1:5,B=NA)
> filter <- frame[ , 1] %% 2 == 0
> print(t(frame))
  [,1] [,2] [,3] [,4] [,5]
A    1    2    3    4    5
B   NA   NA   NA   NA   NA
> print(filter)
[1] FALSE  TRUE FALSE  TRUE FALSE
> print(frame$A[filter])
[1] 2 4
>
```

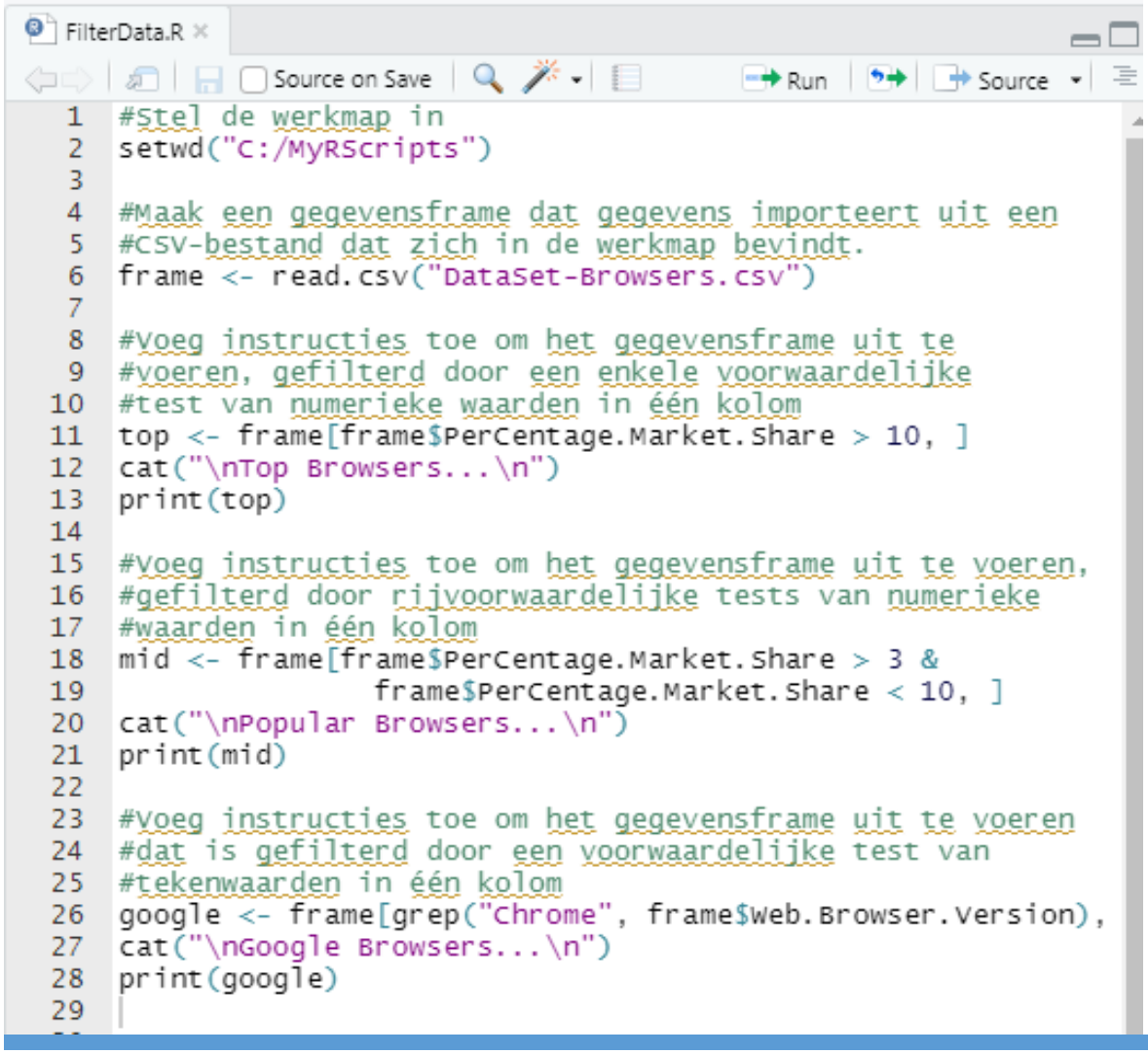
De functie **t()** wordt gebruikt om het gegevensframe te transponeren, zodat de kolommen ervan als rijen worden weergegeven.

Tekengegevens kunnen worden gefilterd door de waarde in elke cel te vergelijken met een waarde die is opgegeven als het eerste argument voor de ingebouwde **grep()**-functie. Deze functie vereist ook een tweede argument om het gebied van het gegevensframe op te geven waarvan de cel waarden moeten worden vergeleken.

Het voorwaardelijke testresultaat van een filter kan worden toegewezen aan een variabele, zoals in de bovenstaande schermafbeelding, of de voorwaardelijke test kan worden gemaakt tussen de [] vierkante haken die een gebied van het gegevensframe specificeren:

1. Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2. Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.



De functie **grep()** kan optioneel een argument **ignore.case=TRUE** bevatten als hoofdlettergevoeligheid niet vereist is.

3 Sla het **R Script** op met de naam **ColumnData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.

```
Console  Terminal  Jobs
C:/MyRScripts/
> source('E:/IT Concepten/IT Concepten - New Site/lessen/rstudio/s
ourcecode/FilterData2.R')

Top Browsers...
              Web.Browser.Version PerCentage.Market.Share
1                  Chrome 61.0                   35.16
2 Microsoft Internet Explorer 11.0                   12.52

Popular Browsers...
  Web.Browser.Version PerCentage.Market.Share
3         Chrome 55.0                   7.40
4          Firefox 56                   6.53
5   Microsoft Edge 14                   3.60

Google Browsers...
   Web.Browser.Version PerCentage.Market.Share
1          Chrome 61.0                   35.16
3          Chrome 55.0                    7.40
6          Chrome 49.0                    2.26
7          Chrome 60.0                    2.19
8          Chrome 45.0                    1.86
9          Chrome 62.0                    1.76
11         Chrome 50.0                    1.59
13         Chrome 58.0                    1.25
17         Chrome 53.0                    0.93
25         Chrome 47.0                    0.64
26         Chrome 59.0                    0.56
29         Chrome 56.0                    0.49
31         Chrome 57.0                    0.47
32         Chrome 43.0                    0.41
34         Chrome 60.4                    0.28
40         Chrome 54.0                    0.17
>
```

De functie **grep()** wordt hier gebruikt om alle rijen te selecteren waarvan de eerste kolom cel de tekenreeks "Chrome" bevat.

Merging data frames

Gegevens uit twee gegevensframes kunnen worden gecombineerd in een enkel gegevensframe door een gemeenschappelijke kolomnaam of een gemeenschappelijke rij-naam met behulp van de ingebouwde functie **merge()**. Deze functie vereist de namen van de twee bron data frames als de eerste twee argumenten. Het vereist ook dat de namen van het kolom- of rijveld met gemeenschappelijke waarden worden opgegeven voor de argumenten **by.x=** en **by.y=**.


Als twee dataframes worden samengevoegd door gemeenschappelijke velden die identieke namen hebben, kunnen de argumenten **by.x=** en **by.y=** worden vervangen door één enkel argument **by=** dat de naam van het gemeenschappelijke veld vermeldt. Bijvoorbeeld **by="State"**.

Als je er de voorkeur aan geeft alle velden te behouden, ongeacht of ze precies overeenkomen, kunt je een **all=TRUE** argument opnemen in de aanroep van de **merge()**-functie. In dit geval worden cellen die niet direct overeenkomen in elk bron data frame, gevuld met de speciale **NA** *(Not Available = niet beschikbaar)* constante in het samengevoegde data frame.

De twee algemene velden van de bron data frames verschijnen als een enkel veld in de samengevoegde data frame, maar andere velden die gegevens bevatten die gemeenschappelijk zijn voor beide bron data frames, verschijnen in afzonderlijke velden. Dit betekent dat het samengevoegde data frame rijen of kolommen met gedupliceerde gegevens kan bevatten. Door een **NULL**-waarde toe te wijzen aan onnodige velden, worden ze verwijderd, zodat alleen unieke gegevens in het samengevoegde data frame overblijven.

Extra kolommen kunnen zoals gebruikelijk aan een samengevoegd data frame worden toegevoegd en kunnen gegevens bevatten die zijn berekend op basis van de gegevens die uit elk bron data frame zijn geleverd:

1. Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

2. Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

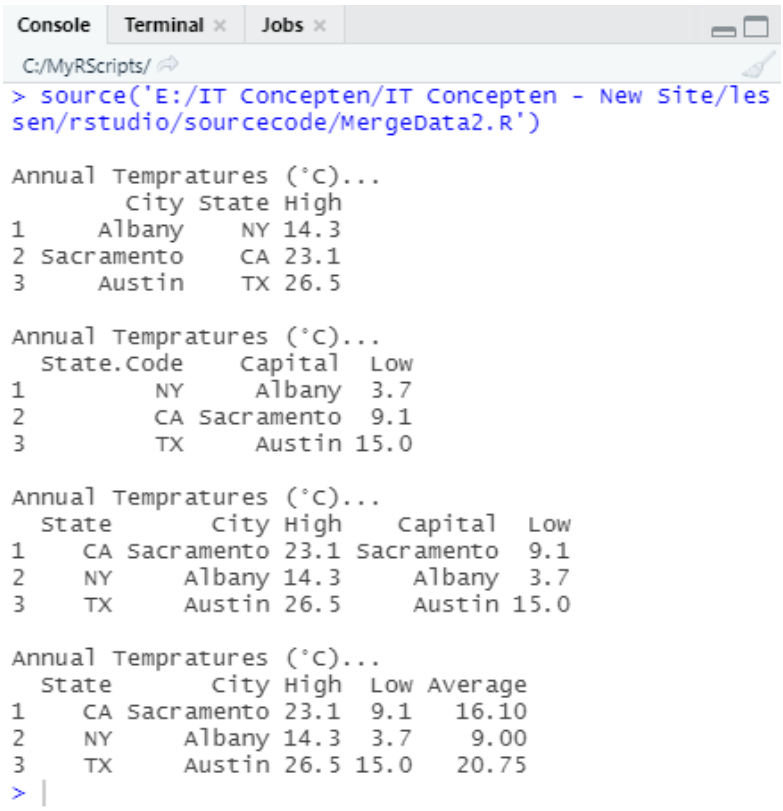

In de onderstaande broncode hebben de kolomvelden met staats-afkortingen verschillende namen en verschijnen op verschillende posities binnen de bron data frames.

MergeData.R

Source on Save | Run | Source

```r
#Stel de werkmap in
setwd("C:/MyRScripts")

#Maak twee data frames van gegevensimport uit CSV-bestanden
#in de werkmap
high.temp <- read.csv("DataSet-HighTemps.csv")
low.temp <- read.csv("DataSet-LowTemps.csv")

#Maak een functie om een titel en een dataframe uit te voeren
#wanneer aangeroepen
display <- function(frame){
  cat("\nAnnual Tempratures (°C)...\n")
  print(frame)
}

#Voeg een instructie toe om elk brongegevensframe uit te voeren
display(high.temp)
display(low.temp)

#Voeg een instructie toe om de twee gegevensframes samen te
#voegen op basis van gemeenschappelijke kolomwaarden
avg.temp <- merge(high.temp, low.temp,
                  by.x="State", by.y="State.Code")
display(avg.temp)

#Voeg een instructie toe om een kolom te verwijderen die dubbele
#gegevens bevat
avg.temp$Capital <- NULL

#Voeg een nieuwe kolom toe met gegevens die zijn berekend op
#basis van de gegevens die zijn geleverd door de twee bron
#data frames en voer de gecombineerde gegevens nogmaals uit
avg.temp$Average <- (avg.temp$High + avg.temp$Low)/2
display(avg.temp)
```

3 Sla het **R Script** op met de naam **MergeData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.



De stadsnamen worden gedupliceerd in het samengevoegde data frame, zodat één kolom niet nodig is en kan worden verwijderd.

Adjusting factors

De structuur van gegevens die vanuit een data frame naar een vectorvariabele zijn gekopieerd, blijft behouden, zodat de **factors** en **levels** behouden blijven. Vectoren bieden niet automatisch factorcategorieën en gerangschikte levels, maar de naam van een vectorvariabele kan worden opgegeven als argument voor de ingebouwde **factor()**-functie om **factors** en **levels** te creëren.

De functie **factor()** rangschikt standaard de factor levels (niveaus) in alfabetische en numerieke volgorde. Als de gegevenswaarden van bijvoorbeeld "B", "C2", "A" en "C1" worden gegeven, worden de factorniveaus gerangschikt als "A" (1), "B" (2), "C1" (3), en "C2" (4).

U kunt optioneel een rangordevoorkeur opgeven door een argument **levels=** op te nemen in de aanroep van de functie **factor()**. Aan dit argument moet een vector van de data values (gegevenswaarden) worden toegewezen in de aflopende volgorde van het rangordeniveau van uw voorkeur, **levels=c("C1", "C2", "A", "B")** zorgt er bijvoorbeeld voor dat de factor levels worden gerangschikt als "C1" (1), "C2"(2), "A" (3) en "B" (4).

Als u de rangorde van factor levels wilt omkeren, kan de vector van data values (gegevenswaarden) worden opgegeven als een argument voor de ingebouwde **rev()**-functie in de toewijzing aan het argument **levels=**:

1 Open RStudio en klik vervolgens op **File**, **New File**, **R Script** of druk op de toetsen **Ctrl + Shift + N** om een nieuw Code Editor venster te openen.

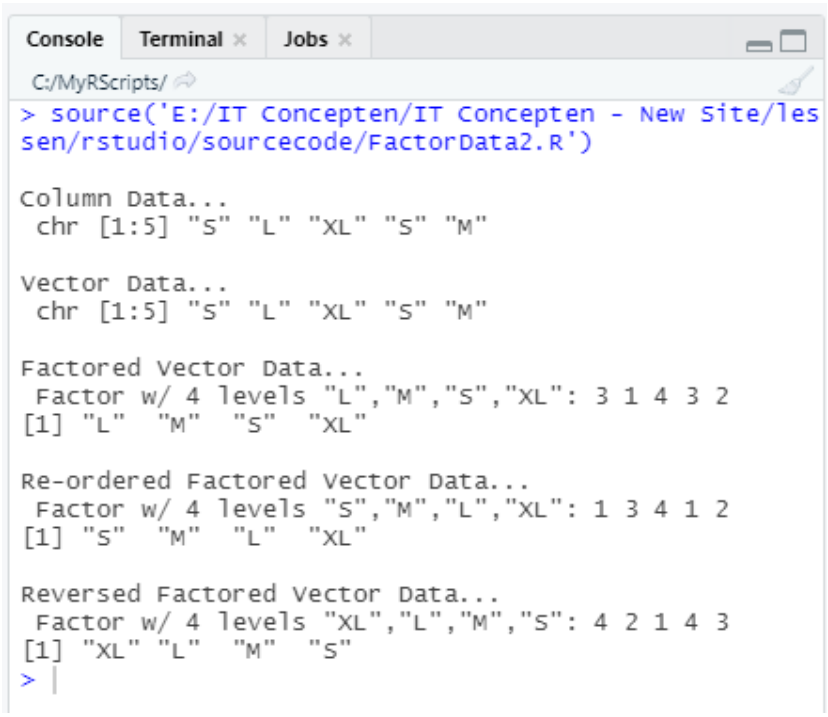
2 Voer de onderstaande broncode in de Code Editor in. Laat de groene commentaarregels weg.

FactorData.R*

```
#Maak een data frame genaamed frame
frame <- data.frame( 1:5, sizes=c( "S", "L", "XL", "S", "M") )

#Voer de structuur van één kolom van het dataframe uit om de
#factoren en hun rangschikkingsniveaus te zien
cat( "\nColumn Data...\n" )
str( frame$sizes )

#Maak een vector met dezelfde data values als de
#kolomgegevens, en in dezelfde volgorde
var.sizes <- c( "S", "L", "XL", "S", "M" )

#Voer de structuur van de vector uit om te zien dat deze alleen
#karaktergegevens bevat zonder factoren of levels
cat( "\nVector Data...\n" )
str( var.sizes )

#Reproduceer de vector om factoren en rangschikkings-levels
#te creëren
var.sizes <- factor( var.sizes )

#Voer de structuur van de vector nogmaals uit om te zien dat
#deze nu factoren en levels heeft
cat( "\nFactored Vector Data...\n" )
str( var.sizes )
print( levels( var.sizes ) )

#Reproduceer de vector om factoren en niveaus te creëren in een
#gewenste rangorde
var.sizes <- factor( var.sizes,
                     levels = c( "S","M","L","XL" ) )

#Voer de herziene structuur van de vector levels uit
cat( "\nRe-ordered Factored Vector Data...\n" )
str( var.sizes )
print( levels( var.sizes ) )

#Reproduceer de vector om factoren en niveaus in een omgekeerde
#rangorde te maken en voer vervolgens de omgekeerde structuur uit
var.sizes <- factor( var.sizes,
                     levels=rev( levels( var.sizes ) ) )
cat( "\nReversed Factored Vector Data...\n" )
str( var.sizes )
print( levels( var.sizes ) )
```

De functie **levels()** biedt toegang tot het attribuut levels van de variabele die als argument is opgegeven.

3 Sla het **R Script** op met de naam **FactorData.R** en klik vervolgens op de knop **Source** of druk op **Ctrl + Shift + S** om de rekenkundige uitvoer te zien.



Factoren zijn unieke waarden. Dit voorbeeld heeft vijf gegevenswaarden, maar slechts vier unieke waarden - er