Practical training Rmarkdown

Willem Stolte

February 25, 2016

Start a new Rmarkdown document

- In Rstudio, choose from file menu: New File / R Markdown…
- Determine what kind of document you want it to be
    - for this training, choose "html" Note that you can also choose a presentation, which changes the choices of the formats that it can handle
- The document opens with some standard, functional, content to get going
- Note that the document contains both "normal" text (with Markdown formatting) combined with 'chunks' of R code (the grey pieces)
- Try and run the document by pressing the "Knit html" button just above the document in Rstudio
    - an html document appears in which the Rmarkdown + R code is combined to a single document.

replace the standard code and text with new code and text We are going to use data from an online source. We will use data from the Marine Projects database.

Code can be inserted by "insert a new code chunk" button (green c) or by ctrl-alt-i

Make a document that reads data, and uses the data for a time plot. It is a good habit to name the code chunks. For example "'{r ReadData}

```
## first look which layers there are in the wfs
## (can also interactively be done in QGIS)
library(gdalUtils)
library(rgdal)
dsn <- "http://marineprojects.openearth.nl/geoserver/mep-nsw/ows?service=WFS&request=GetCapabilities"
dsn <- "http://marineprojects.openearth.nl/geoserver/ows?service=WFS&request=GetCapabilities"
ogrinfo(dsn, so=TRUE)
```

```
##  [1] "INFO: Open of `http://marineprojects.openearth.nl/geoserver/ows?service=WFS&request=GetCapabil
##  [2] "      using driver `WFS' successful."
##  [3] "1: inspire_marine:Inspire_Marine"
##  [4] "2: shwoz:shwoz"
##  [5] "3: RijMaMo:Zoutgehalte bij Hoek van Holland"
##  [6] "4: shwoz:kleinemantelmeeuw (Point)"
##  [7] "5: rwsbenthos:meetobject"
##  [8] "6: mep-nsw:mep-nsw_bentcom"
##  [9] "7: ihm_viewer:mep-nsw_benthos"
## [10] "8: mep-nsw:mep-nsw_bivalves"
## [11] "9: mep-nsw:mep-nsw_didson"
## [12] "10: mep-nsw:mep-nsw_dvis"
## [13] "11: mep-nsw:mep-nsw_gillnet"
## [14] "12: mep-nsw:mep-nsw_macrobenthos"
## [15] "13: mep-nsw:mep-nsw_pvis"
## [16] "14: ihm_viewer:mep-nsw_vis"
## [17] "15: rwsbenthos:meting"
## [18] "16: rwsbenthos:meting_1992"
## [19] "17: rwsbenthos:meting_1993"
## [20] "18: rwsbenthos:monster"
## [21] "19: ihm_viewer:shwoz_vliegroute_mantelmeeuw"
## [22] "20: ihm_viewer:shwoz_vogels"
## [23] "21: ihm_viewer:shwoz_zeezoogdieren"
```

```
## define url of WFS service, where data are served as csv file:
url <- "http://marineprojects.openearth.nl/geoserver/mep-nsw/ows?service=WFS&version=1.0.0&request=GetF
## read data as csv (strings are read as character type, not as factors)
bivalves <- read.csv(url, stringsAsFactors = F)
```

Observe that these are only the first 50 records. This is perfect for testing.

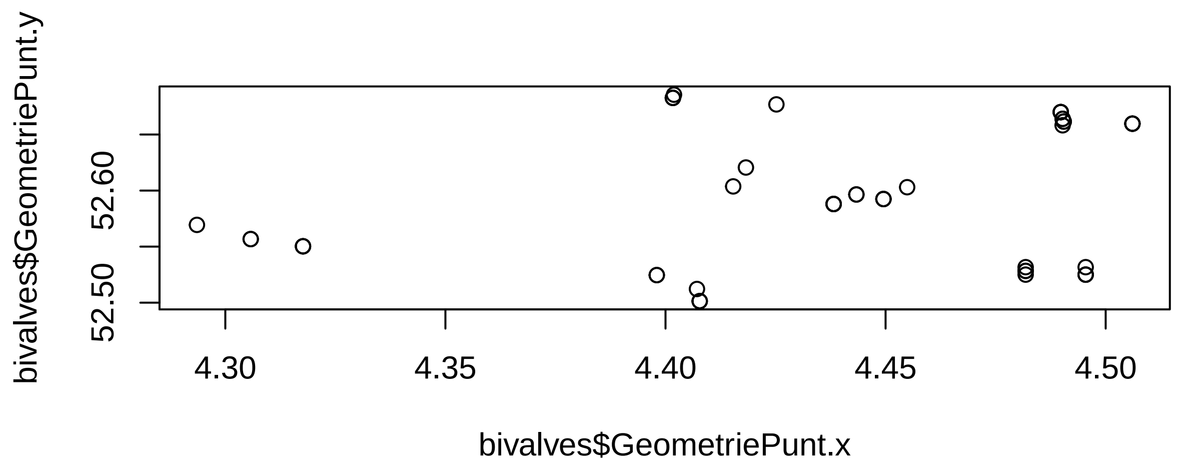
```
##Look at the structure of the data set in the environment pane, or by running:
str(bivalves)
```

```
## 'data.frame':    50 obs. of  30 variables:
##  $ FID                           : chr  "mep-nsw_bivalves.fid-336b7e76_1533271048f_-32f8" "mep-nsw_biva
##  $ Meetpunt.identificatie        : chr  "Windmolenpark Egmond aan Zee" "Windmolenpark Egmond aan Zee"
##  $ Metingomschrijving            : chr  "R4-84" "R3-60" "R3-55" "R3-51" ...
##  $ GeometriePunt.x               : num  4.31 4.51 4.49 4.49 4.4 ...
##  $ GeometriePunt.y               : num  52.6 52.7 52.7 52.7 52.7 ...
##  $ Geometrie                     : chr  "POINT (4.30577 52.55672)" "POINT (4.50608 52.65967)" "POINT (4
##  $ Referentiehorizontaal.code    : chr  "EPSG:4258" "EPSG:4258" "EPSG:4258" "EPSG:4258" ...
##  $ Monster.identificatie         : int  319 314 310 298 291 274 258 328 324 301 ...
##  $ compartiment.code             : chr  "OE" "OE" "OE" "OE" ...
##  $ orgaan.code                   : logi  NA NA NA NA NA NA ...
##  $ organisme.naam                : chr  "Abra alba" "Abra alba" "Abra alba" "Abra alba" ...
##  $ begindiepte_m                 : num  18.6 22 20.6 20.7 22.6 ...
##  $ einddiepte_m                  : num  18.6 22 20.6 20.7 22.6 ...
##  $ bemonsteringsmethode.code     : logi  NA NA NA NA NA NA ...
##  $ Veldapparaat.omschrijving     : chr  "Niet van Toepassing" "Niet van Toepassing" "Niet van Toepassi
##  $ monsternemingsdatum           : chr  "2007-11-10" "2007-08-10" "2007-08-10" "2007-08-10" ...
##  $ monsternemingstijd            : chr  "09:29" "20:24" "19:12" "18:17" ...
##  $ Parameter.omschrijving        : chr  "Aantal per oppervlakte" "Aantal per oppervlakte" "Aantal per o
##  $ Eenheid.code                  : chr  "n/m2" "n/m2" "n/m2" "n/m2" ...
##  $ Hoedanigheid.code             : chr  "NVT" "NVT" "NVT" "NVT" ...
##  $ Waardebewerkingsmethode.code: chr  "NVT" "NVT" "NVT" "NVT" ...
##  $ Waardebepalingsmethode.code : logi  NA NA NA NA NA NA ...
##  $ Begindatum                    : chr  "2007-11-10" "2007-08-10" "2007-08-10" "2007-08-10" ...
##  $ Begintijd                     : chr  "09:29" "20:24" "19:12" "18:17" ...
##  $ tijd_UTCoffset                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resultaatdatum                : chr  "2007-11-10" "2007-08-10" "2007-08-10" "2007-08-10" ...
##  $ Limietsymbool                 : chr  "=" "=" "=" "=" ...
##  $ Numeriekewaarde               : num  41.7 41.7 83.3 166.7 416.7 ...
##  $ Alfanumeriekewaarde           : logi  NA NA NA NA NA NA ...
##  $ Kwaliteitsoordeel.code        : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
## Date and time are characters at this moment. They can be converted to date and time object by combin
bivalves$datetime <- paste(bivalves$Begindatum, bivalves$Begintijd)
## and convert to POSIXct format
bivalves$datetime <- as.POSIXct(bivalves$datetime, format = "%Y-%m-%d %H:%M")
```

First, we plot locations, longitude and latitude fields are used. Using Base functionality

```
plot(bivalves$GeometriePunt.x, bivalves$GeometriePunt.y)
```



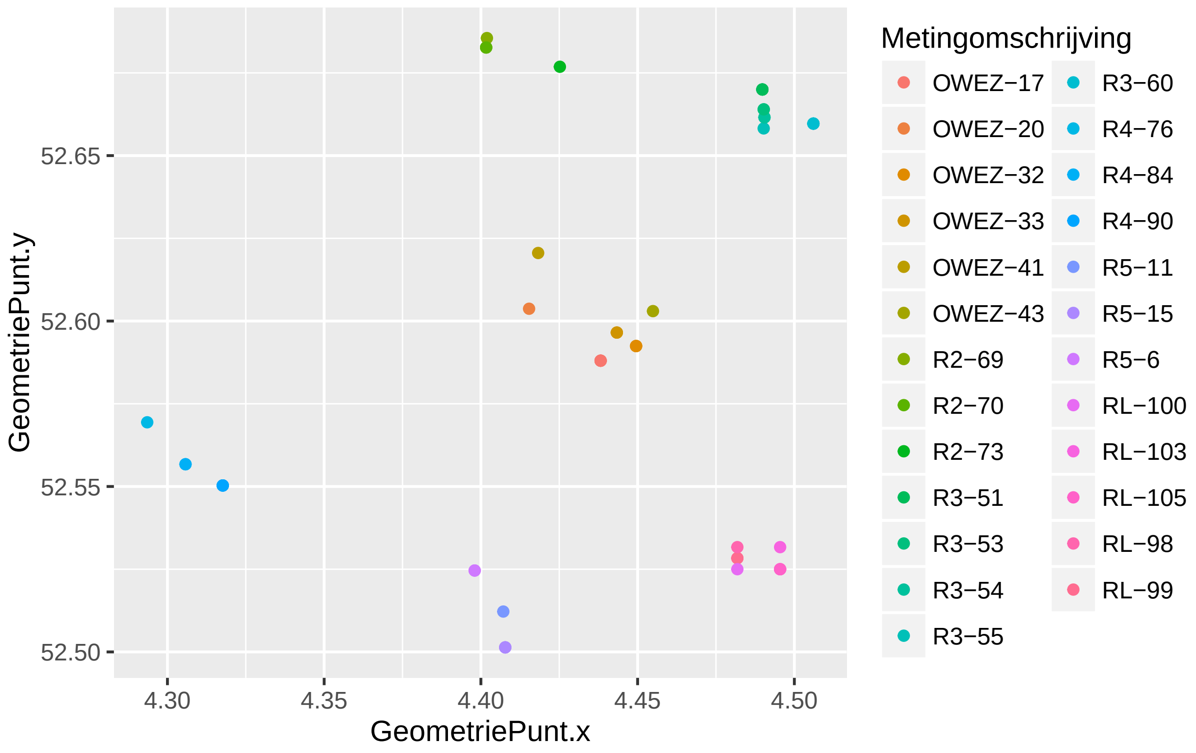
Or using the ggplot package

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```
p <- ggplot(bivalves, aes(GeometriePunt.x, GeometriePunt.y))
p + geom_point(aes(color = Metingomschrijving))
```



Or using an interactive map

```r
require(leaflet)

leaflet() %>%
  addTiles(group = "OSM (default)") %>%
  addCircleMarkers(data=bivalves,
                   lng = bivalves$GeometriePunt.x,
                   lat = bivalves$GeometriePunt.y,
                   popup = sprintf(as.character(bivalves$Metingomschrijving))
                   # clusterOptions = markerClusterOptions()
                   )
```

(leaflet maps doe not work with pdf or word documents)

Neat Reporting

creating nice looking reports

For more serious reporting, some more styling can be done. In the first place, often you don't want the code to be reproduced in the report. This is done by the "echo = F" command in the code specification. Explore the other options for th code specification by pressing "tab" inside the specification {r, }

```
\{r, echo = F}   url <- "http://marineprojects.openearth.nl/geoserver/mep-nsw/ows?service=WFS&version=1.
## read data as csv (strings are read as character type, not as factors)     bivalves <-
read.csv(url, stringsAsFactors = F)     # plot the results    require(ggplot2)   p <-
ggplot(bivalves, aes(x = Metingomschrijving, y = Numeriekewaarde))    p + geom_bar(stat =
"identity") +      theme(axis.text.x = element_text(angle = 90, hjust = 1)) +     ylab(unique(bivalves$
+     facet_grid(organisme.naam ~ .)   \\
```

Creating tables

Lets now present the data in a cross-table. The long table “bivalves” will be turned into a cross-table using the cast function from the reshape2 package. dcast is used, as the input is a dataframe.

```
# first make a cross table of species and locations
require(reshape2)
bivalcross <- dcast(data = bivalves, formula = Metingomschrijving ~ organisme.naam, value.var = "Numeri
knitr::kable(head(bivalcross, 10), align = "l")
```

Metingomschrijving	Abra alba	Chamelea gallina	Ensis	Montacuta ferruginosa	Mysella bidentata	Spisula	Tellin
OWEZ-17	41.7	NA	NA	NA	1125.0	41.7	41.7
OWEZ-20	NA	41.7	NA	NA	NA	NA	NA
OWEZ-32	NA	NA	NA	NA	166.7	41.7	458.3
OWEZ-33	41.7	NA	NA	41.7	NA	NA	NA
OWEZ-41	NA	NA	NA	41.7	NA	NA	NA
OWEZ-43	NA	NA	NA	NA	NA	NA	41.7
R2-69	NA	NA	NA	208.3	NA	NA	NA
R2-70	416.7	NA	NA	458.3	NA	NA	666.7
R2-73	NA	NA	NA	41.7	NA	NA	NA
R3-51	166.7	NA	NA	125.0	41.7	NA	208.3

some tips

- Do not use the “reindent lines” command on the whole document, spaces are important in Rmarkdown.
- Use the help (F1, tab). and cheatsheet and format specific help (/formats)
- tables can be made in a number of ways.