exercises01

Jan Hohenheim

2/24/2022

Problem 1 (Introduction to R/RStudio)

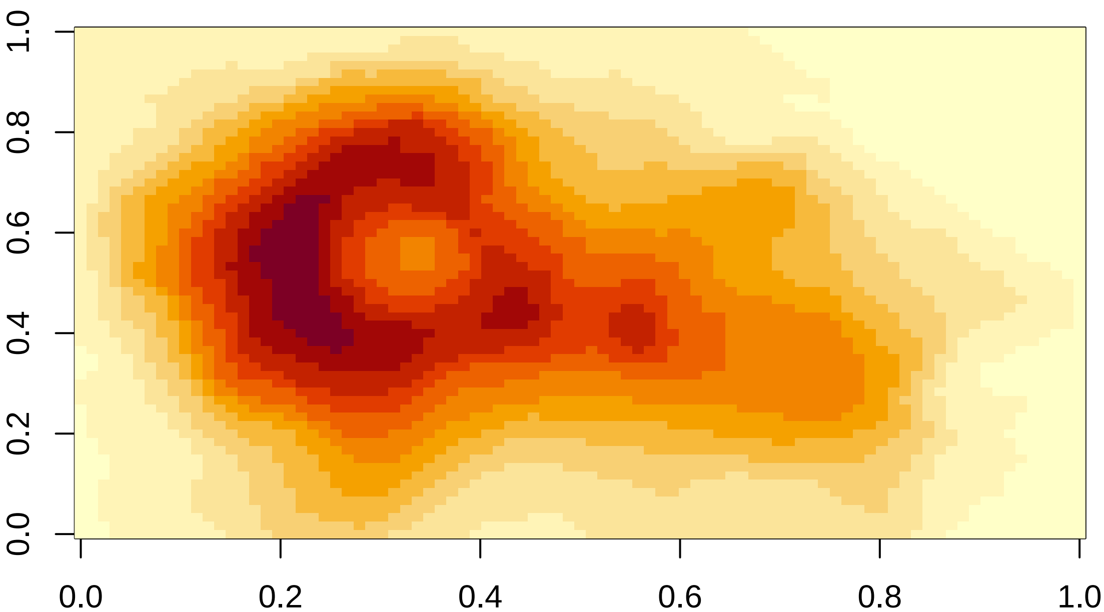
The aim of this exercise is to get some insight on the capabilities of the statistical software environment R and the integrated development environment **RStudio**.

(a) R has many built-in datasets, one example is volcano. Based on the help of the dataset, what is the name of the Volcano? Describe the dataset in a few words.

```
?volcano
```

The name of the volcano is Maunga Whau (Mt Eden). The data set is the height of the terrain in a given square in a grid, where each cell represents a terrain of 10 m by 10 m.

(b) Use the R help function to get information on how to use the *image()* function for plotting matrices. Display the volcano data.



(c) Install the package **fields**. Display the volcano data with the function **image.plot()**. What is the maximum height of the volcano?

```
# install.packages("fields")
require(fields)
```

```
## Loading required package: fields
```

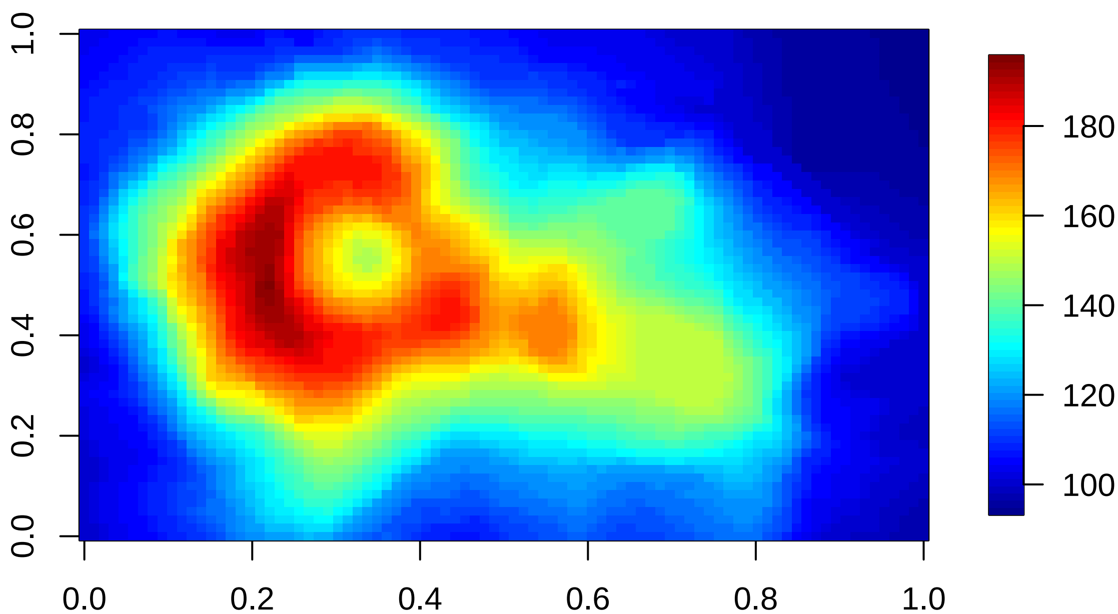
```
## Loading required package: spam
```

```
## Spam version 2.8-0 (2022-01-05) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
max(volcano)
```

```
## [1] 195
```

```
image.plot(volcano)
```

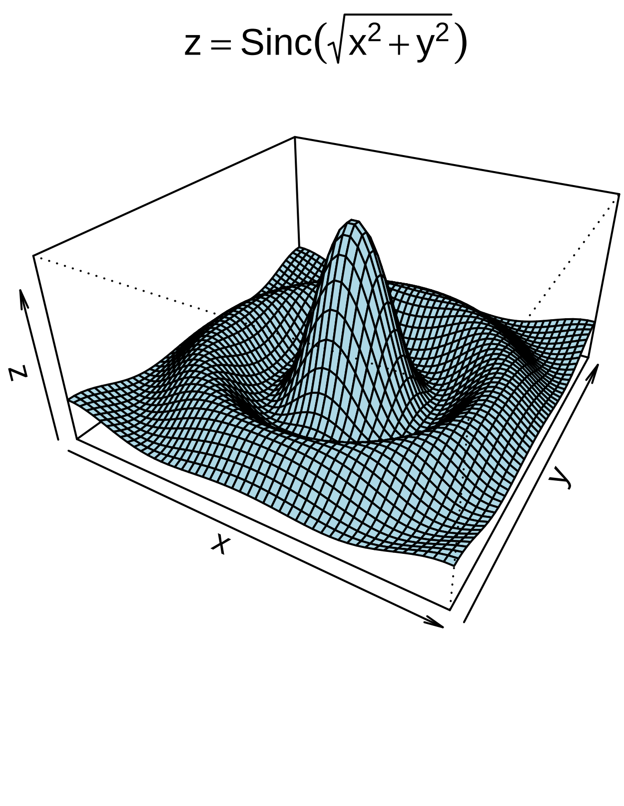


The highest point is 195 m high.

(d) Use the the R help function to find out the purpose of the function demo() and have a look at the list of available demos. The demo of the function persp() utilizes the volcano data to illustrate basic three-dimensional plotting. Call the demo and have a look at the plots.

```
##
##
##  demo(persp)
##  ---- ~~~~~
##
## > ### Demos for  persp()  plots   -- things not in  example(persp)
## > ### -------------------------
```

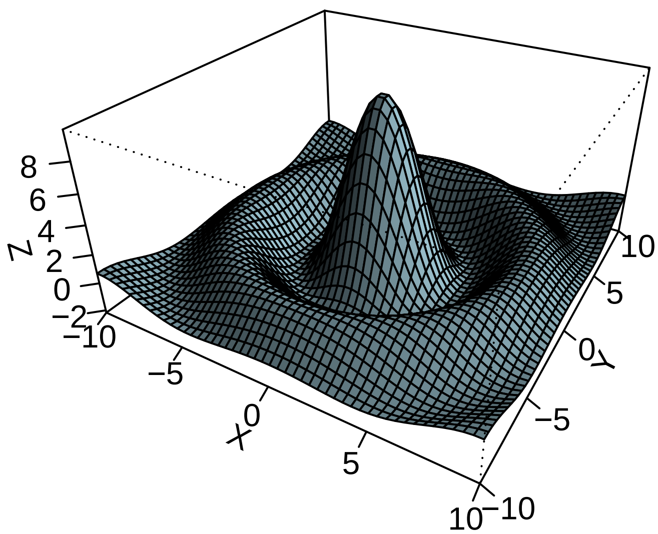
```
## >
## > require(datasets)
##
## > require(grDevices); require(graphics)
##
## > ## (1) The Obligatory Mathematical surface.
## > ##     Rotated sinc function.
## >
## > x <- seq(-10, 10, length.out = 50)
##
## > y <- x
##
## > rotsinc <- function(x,y)
## + {
## +     sinc <- function(x) { y <- sin(x)/x ; y[is.na(y)] <- 1; y }
## +     10 * sinc( sqrt(x^2+y^2) )
## + }
##
## > sinc.exp <- expression(z == Sinc(sqrt(x^2 + y^2)))
##
## > z <- outer(x, y, rotsinc)
##
## > oldpar <- par(bg = "white")
##
## > persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
```




```
##
## > title(sub=".")## work around persp+plotmath bug
```

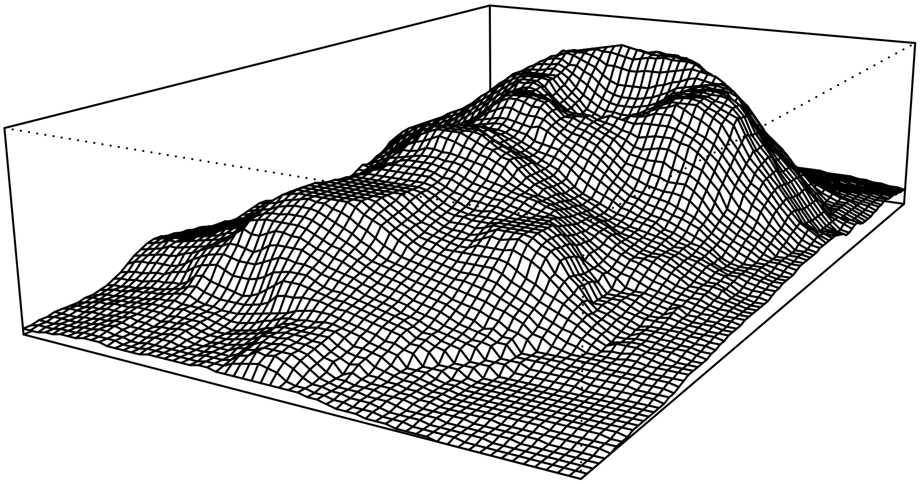
```
##
## > title(main = sinc.exp)
##
## > persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
## +       ltheta = 120, shade = 0.75, ticktype = "detailed",
## +       xlab = "X", ylab = "Y", zlab = "Z")
```

$$z = \mathrm{Sinc}\left(\sqrt{x^2 + y^2}\right)$$

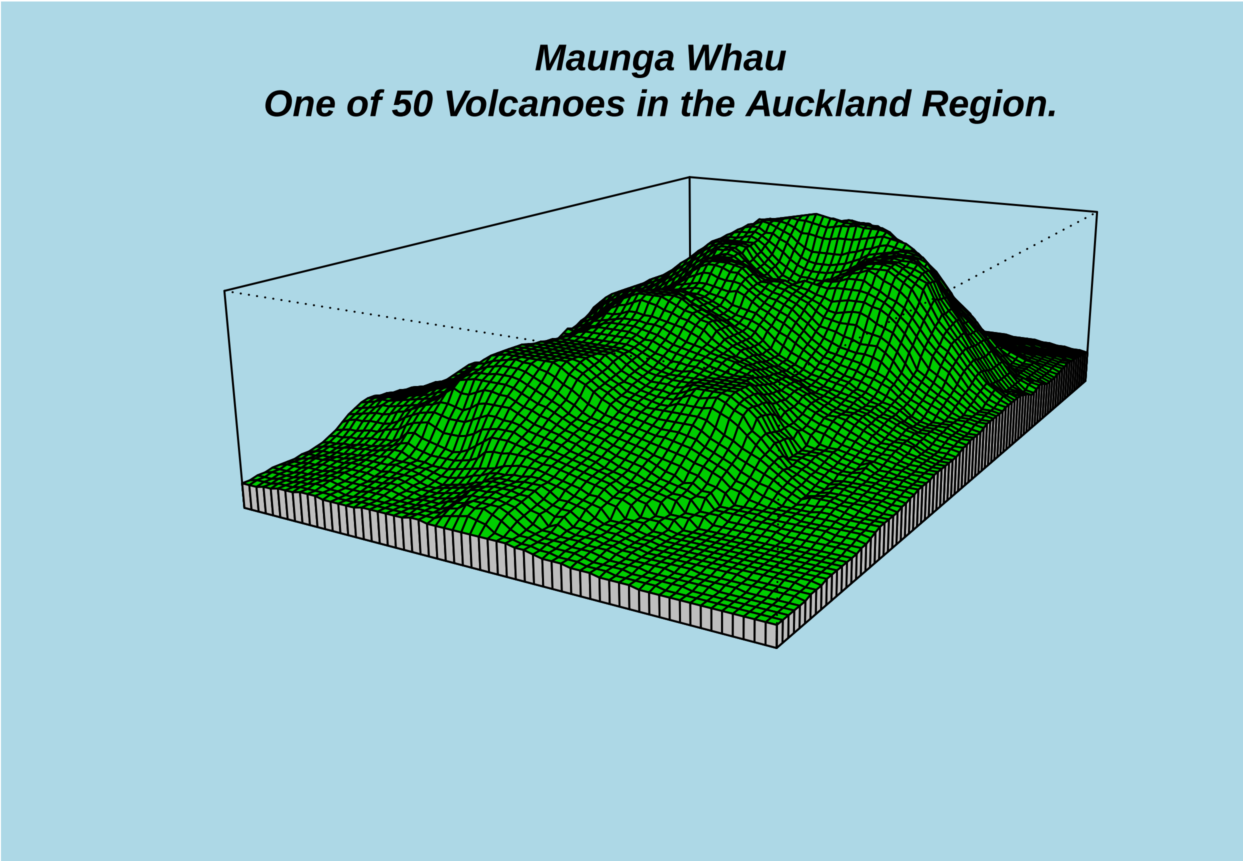


.

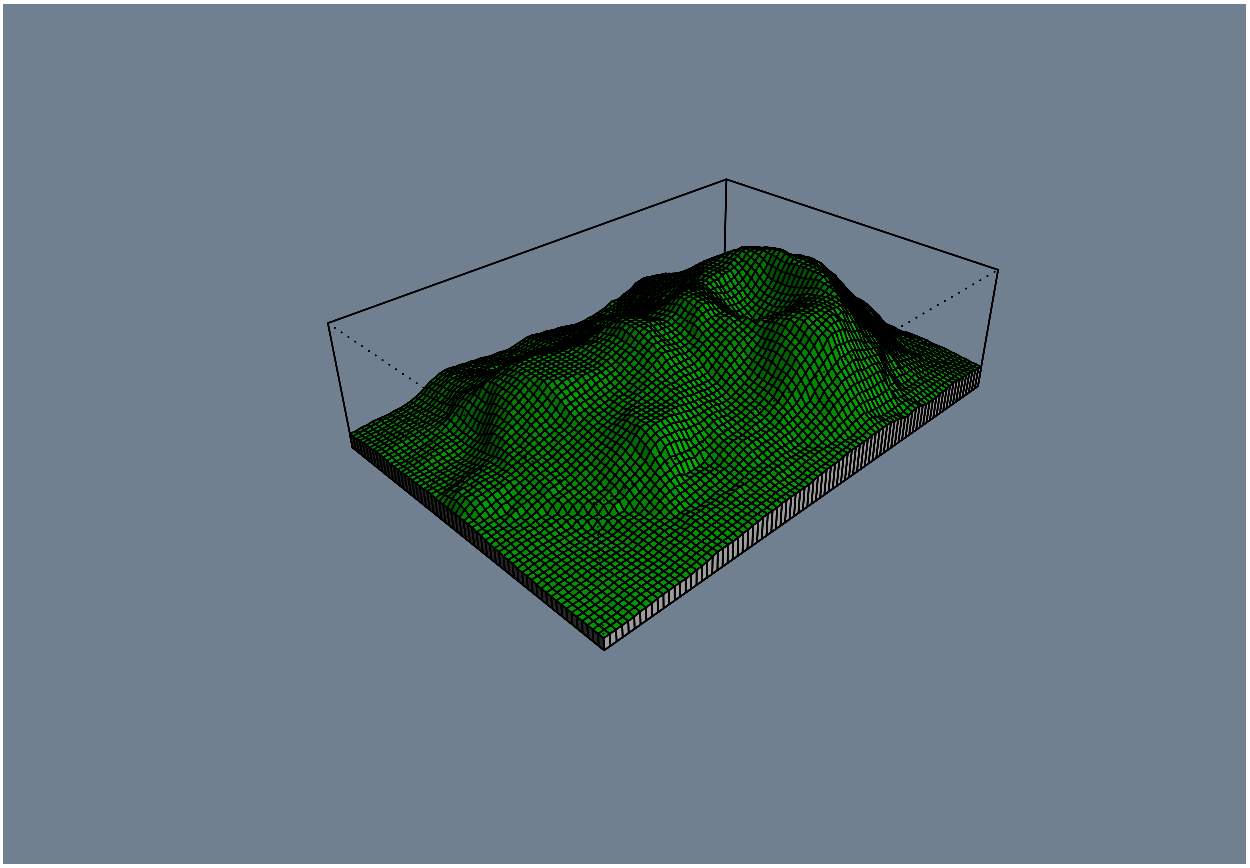
```
##
## > title(sub=".")## work around persp+plotmath bug
##
## > title(main = sinc.exp)
##
## > ## (2) Visualizing a simple DEM model
## >
## > z <- 2 * volcano        # Exaggerate the relief
##
## > x <- 10 * (1:nrow(z))   # 10 meter spacing (S to N)
##
## > y <- 10 * (1:ncol(z))   # 10 meter spacing (E to W)
##
## > persp(x, y, z, theta = 120, phi = 15, scale = FALSE, axes = FALSE)
```



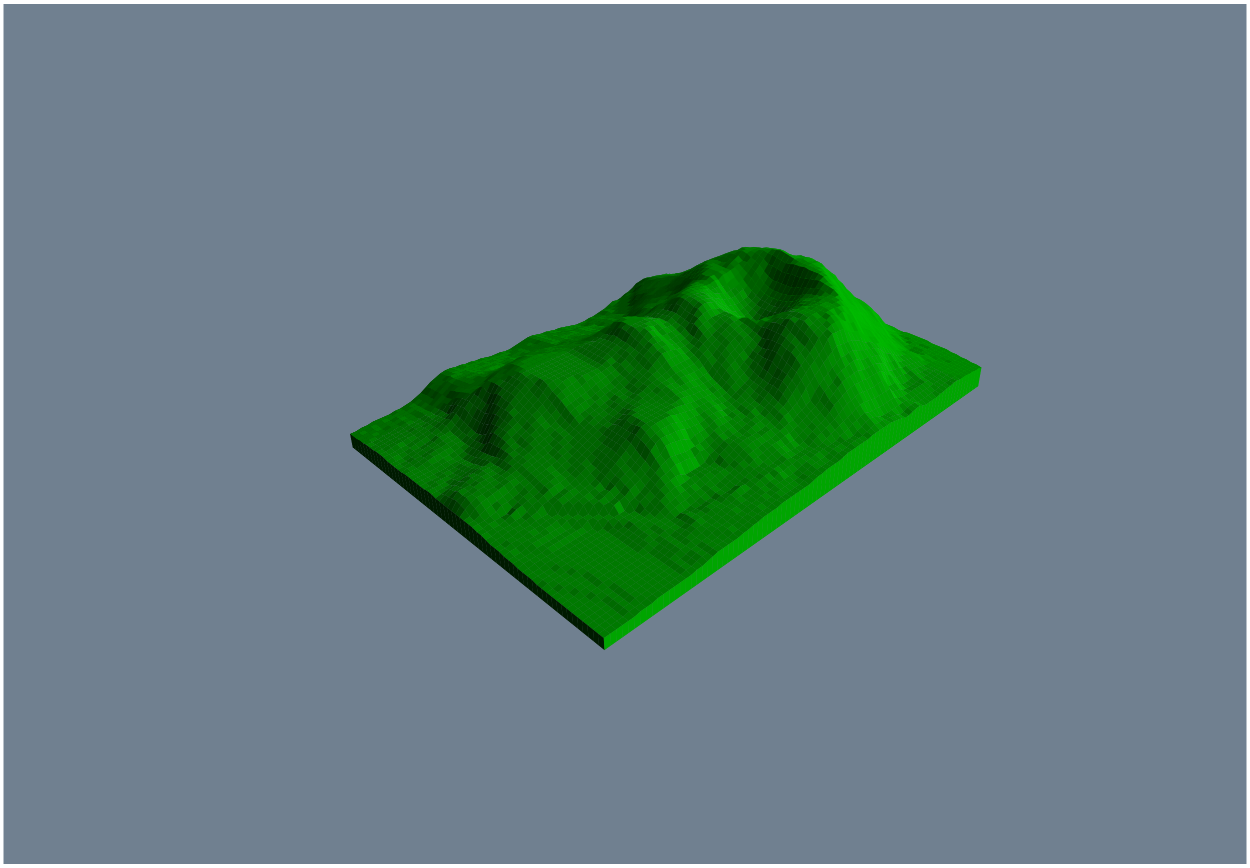
```
##
## > ## (3) Now something more complex
## > ##     We border the surface, to make it more "slice like"
## > ##     and color the top and sides of the surface differently.
## >
## > z0 <- min(z) - 20
##
## > z <- rbind(z0, cbind(z0, z, z0), z0)
##
## > x <- c(min(x) - 1e-10, x, max(x) + 1e-10)
##
## > y <- c(min(y) - 1e-10, y, max(y) + 1e-10)
##
## > fill <- matrix("green3", nrow = nrow(z)-1, ncol = ncol(z)-1)
##
## > fill[ , i2 <- c(1,ncol(fill))] <- "gray"
##
## > fill[i1 <- c(1,nrow(fill)) , ] <- "gray"
##
## > par(bg = "lightblue")
##
## > persp(x, y, z, theta = 120, phi = 15, col = fill, scale = FALSE, axes = FALSE)
```



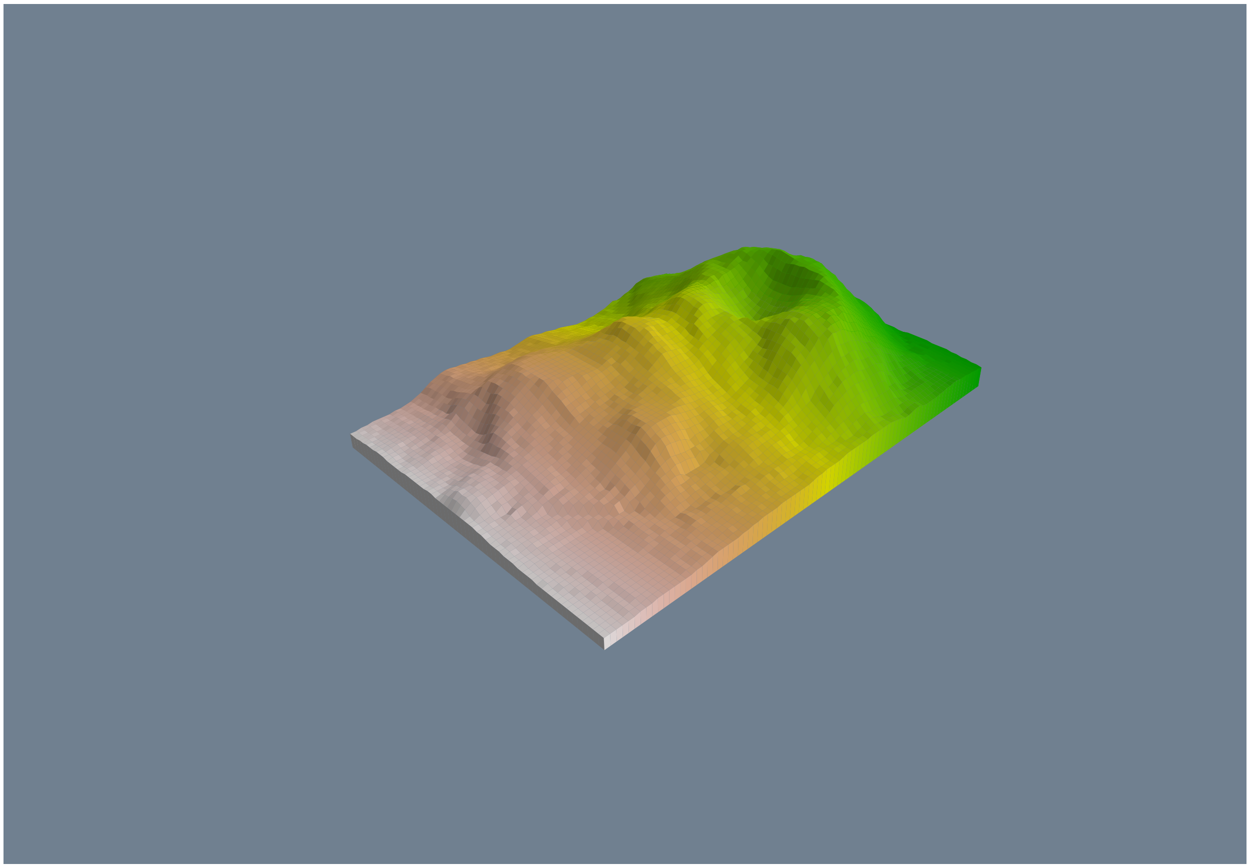

```
##
## > title(main = "Maunga Whau\nOne of 50 Volcanoes in the Auckland Region.",
## +       font.main = 4)
##
## > par(bg = "slategray")
##
## > persp(x, y, z, theta = 135, phi = 30, col = fill, scale = FALSE,
## +       ltheta = -120, lphi = 15, shade = 0.65, axes = FALSE)
```



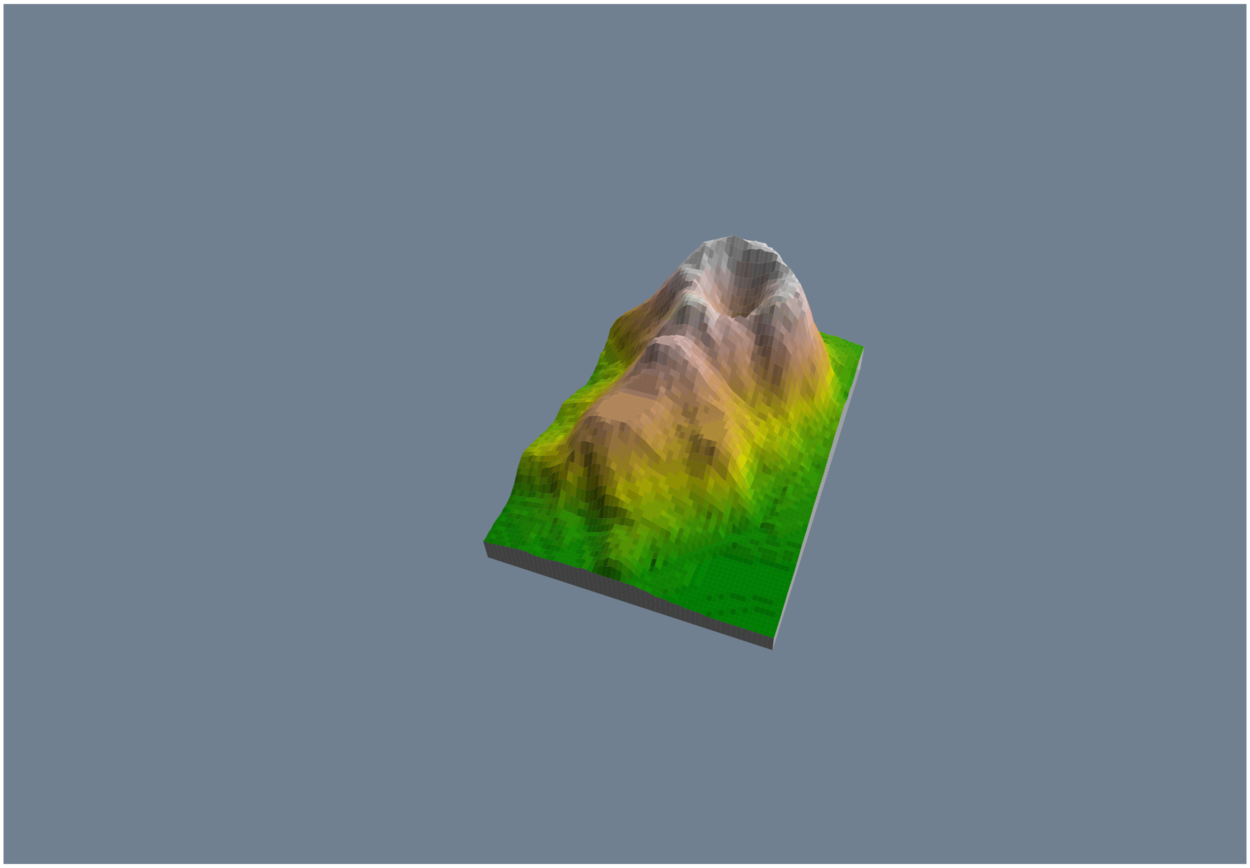
```
##
## > ## Don't draw the grid lines :  border = NA
## > persp(x, y, z, theta = 135, phi = 30, col = "green3", scale = FALSE,
## +       ltheta = -120, shade = 0.75, border = NA, box = FALSE)
```



```
##
## > ## `color gradient in the soil' :
## > fcol <- fill ; fcol[] <- terrain.colors(nrow(fcol))
##
## > persp(x, y, z, theta = 135, phi = 30, col = fcol, scale = FALSE,
## +       ltheta = -120, shade = 0.3, border = NA, box = FALSE)
```



```
##
## > ## `image like' colors on top :
## > fcol <- fill
##
## > zi <- volcano[ -1,-1] + volcano[ -1,-61] +
## +            volcano[-87,-1] + volcano[-87,-61]  ## / 4
##
## > fcol[-i1,-i2] <-
## +     terrain.colors(20)[cut(zi,
## +                            stats::quantile(zi, seq(0,1, length.out = 21)),
## +                            include.lowest = TRUE)]
##
## > persp(x, y, 2*z, theta = 110, phi = 40, col = fcol, scale = FALSE,
## +       ltheta = -120, shade = 0.4, border = NA, box = FALSE)
```



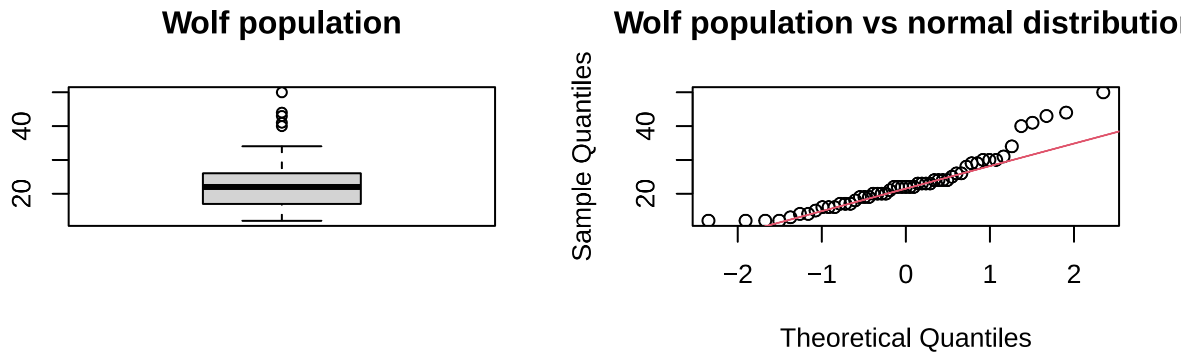
```
##
## > ## reset par():
## > par(oldpar)
```

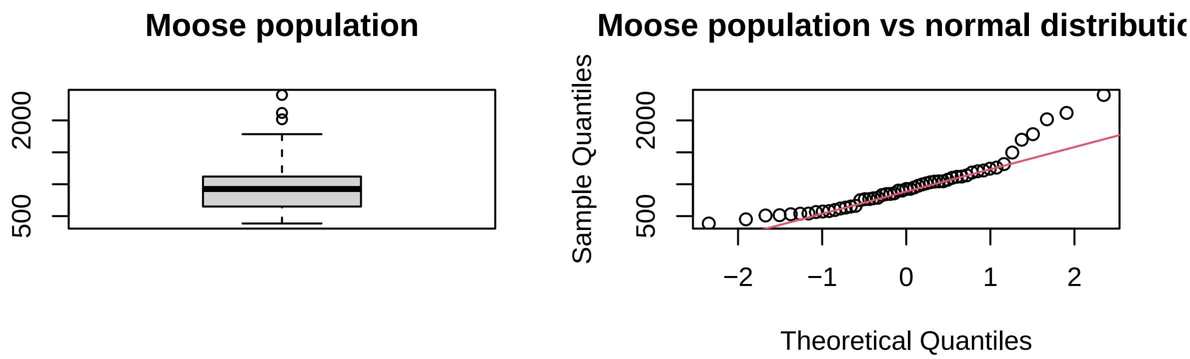
Problem 2 (EDA of bivariate data)

On www.isleroyalewolf.org/data/data/home.html the file isleroyale_graph_data_28Dec2011.xlsx contains popula- tion data from wolves and moose. The information in this file is extracted and saved in the file **01wolvesmoose.csv**. Download and read the file **01wolvesmoose.csv** from the STA120 course page.

(a) Construct a boxplot and a QQ-plot of the moose and wolf data. Give a short interpretation.

```
par( mfcol=c(2, 2))
data <- read.csv("01_wolvesmoose.csv")
?boxplot
boxplot(data$Wolf, main="Wolf population")
boxplot(data$Moose, main="Moose population")

qqnorm(data$Wolf, main="Wolf population vs normal distribution")
qqline(data$Wolf, col=2, main='')  # add read line
qqnorm(data$Moose, main="Moose population vs normal distribution")
qqline(data$Moose, col=2, main='')  # add read line
```

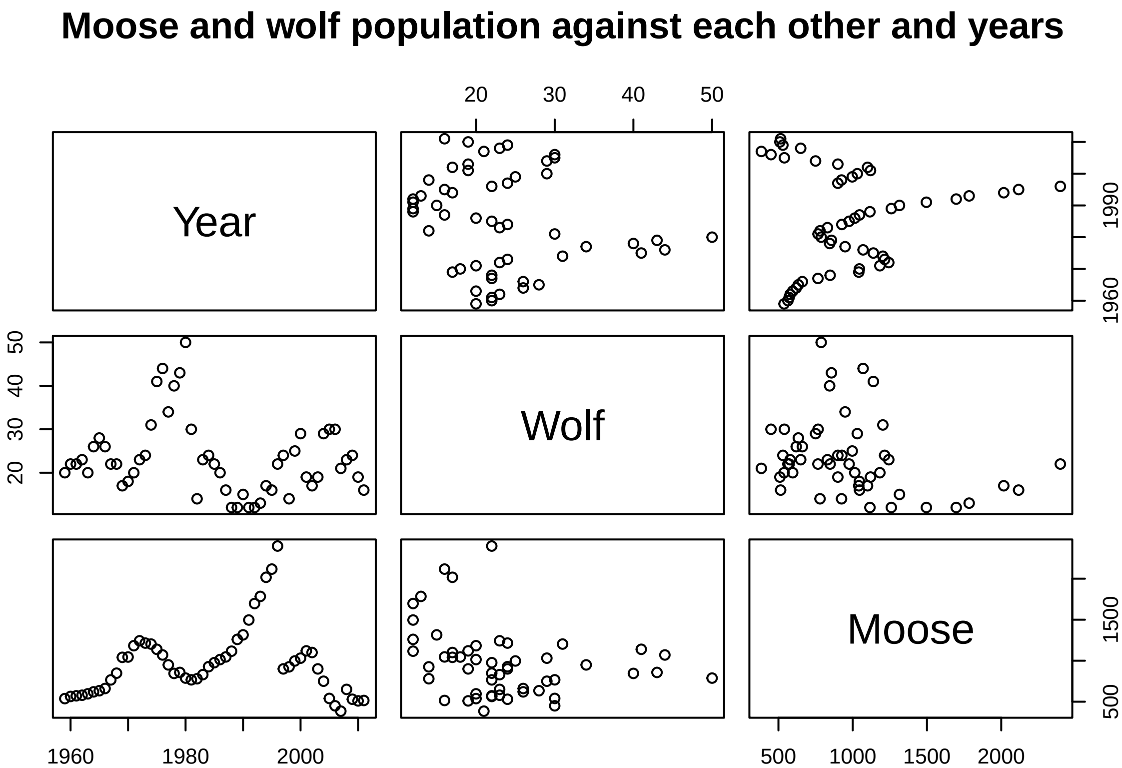




The wolf and moose populations seem to follow a very similar distribution, with the moose population being skewed to the right. The normal distribution fits for all but the outer of the 5 quantiles.

(b) Jointly visualize the wolves and moose data, as well as their abundances over the years. Give a short interpretation of what you see in the figures. (Of course you may compare the result with what is given on the aforementioned web page).

```
pairs(x=data, main="Moose and wolf population against each other and years")
```



The populations seem to follow the Lotka-Volterra model: The wolf population lags behind the moose population, they both oscillate. Plotting the populations against each other reveals that a large amount of either population is not compatible with a large amount of the other.

Problem 3 (EDA of multivariate data)

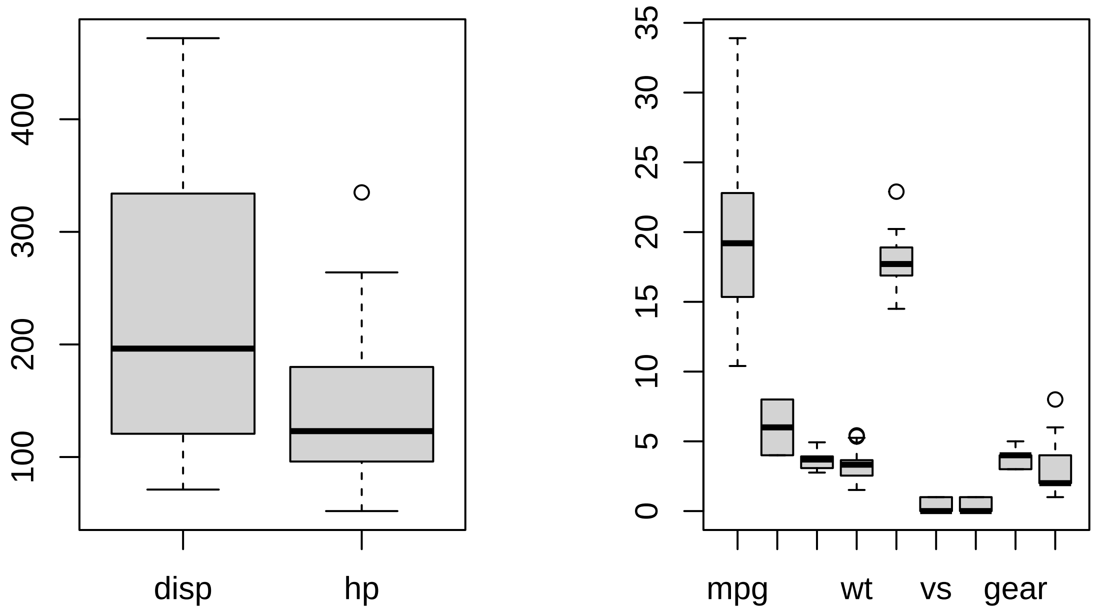
In this problem we want to explore the classical **mtcars** dataset (directly available through the package **datasets**). Perform an EDA thereof and provide at least three meaningful plots (as part of the EDA) and a short description of what they display. In what measurement scale are the variables stored and what would be the natural or original measurement scale?

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

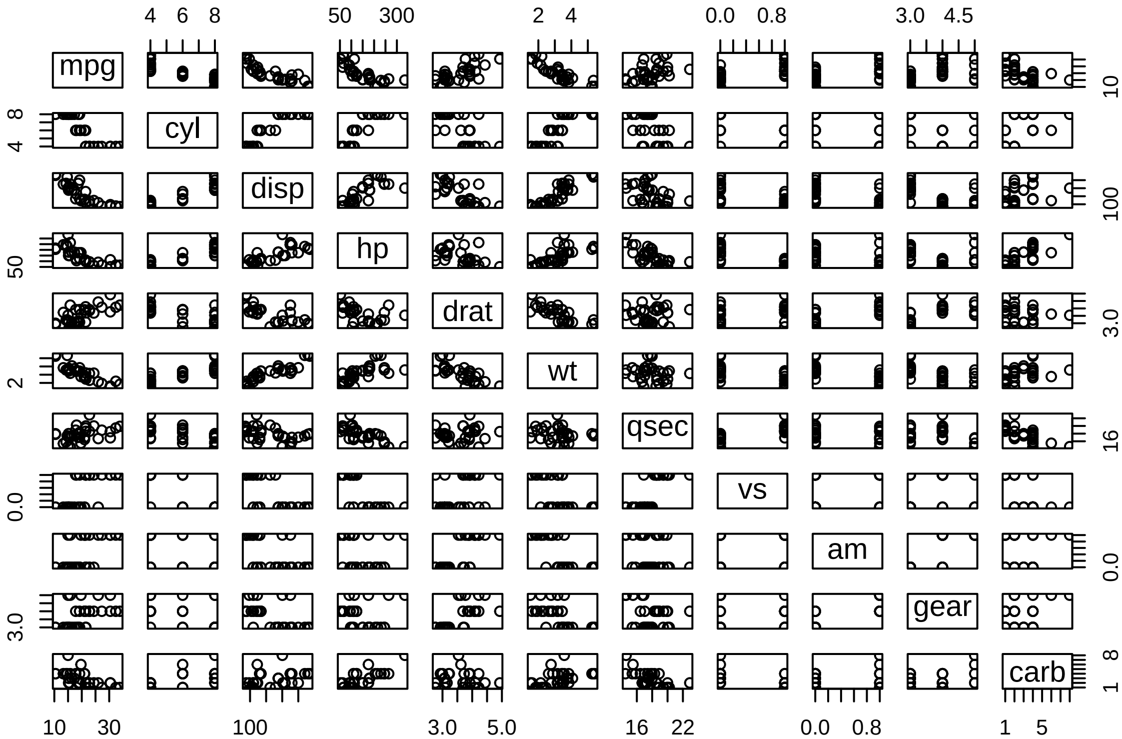
```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
```

```
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
## integer(0)
```

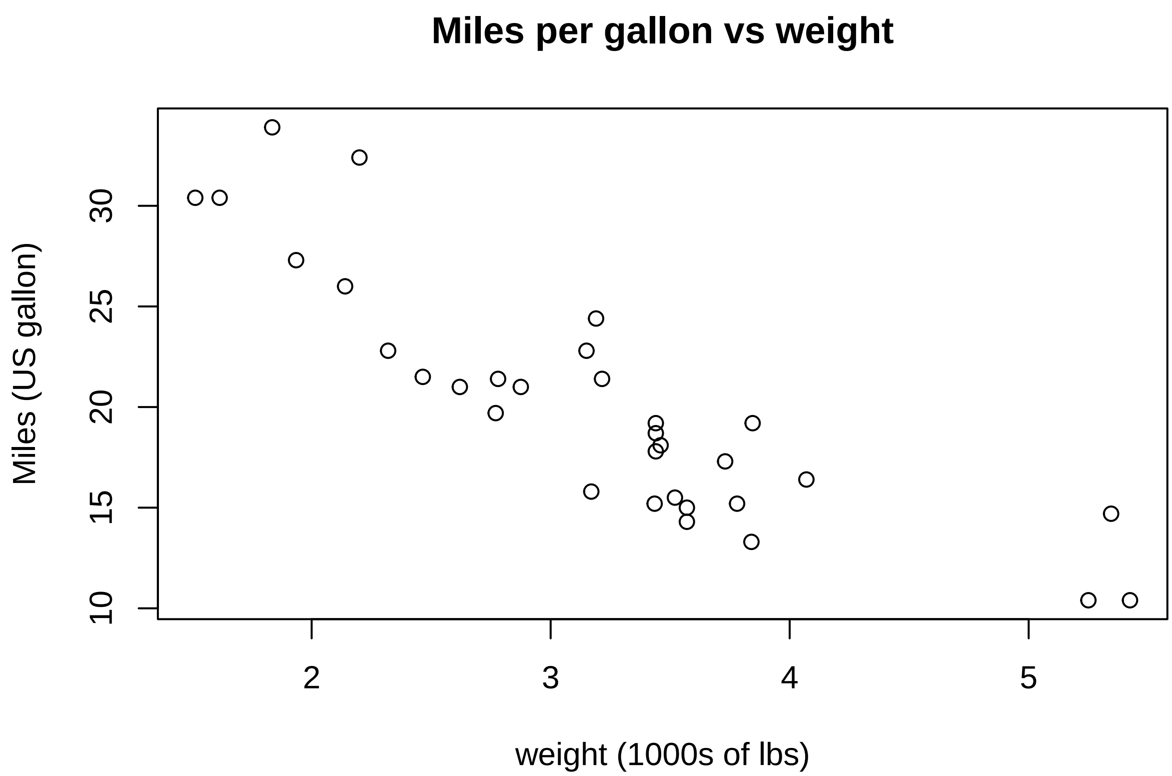



```
pairs(mtcars)
```

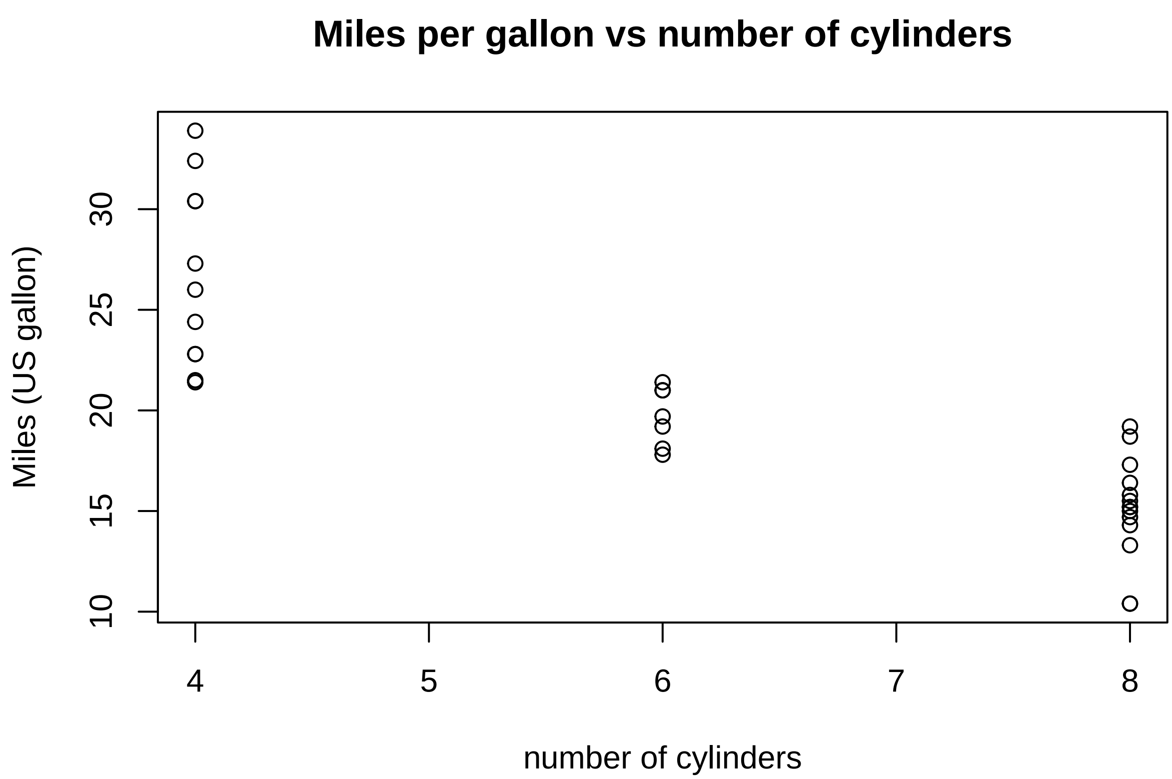


Data collection: Extracter from 1974 *Motor Trend* US magazine. No missing values. Types of data:
- **Discrete**: cyl (Number of cylinders), vs (Engine, 0 = V-shaped, 1 = straight), gear (Number of forward gears), carb (Number of carburetors), am (Transmission, 0 = automatic, 1 = manual) - **Continuous**: mpg (Miles/(US) gallon), disp (Displacement in cubic inches), hp (Gross horsepower), drat (Rear axle ratio), wt (Weight in 1000s of lbs), qsec (1/4 mile speed)

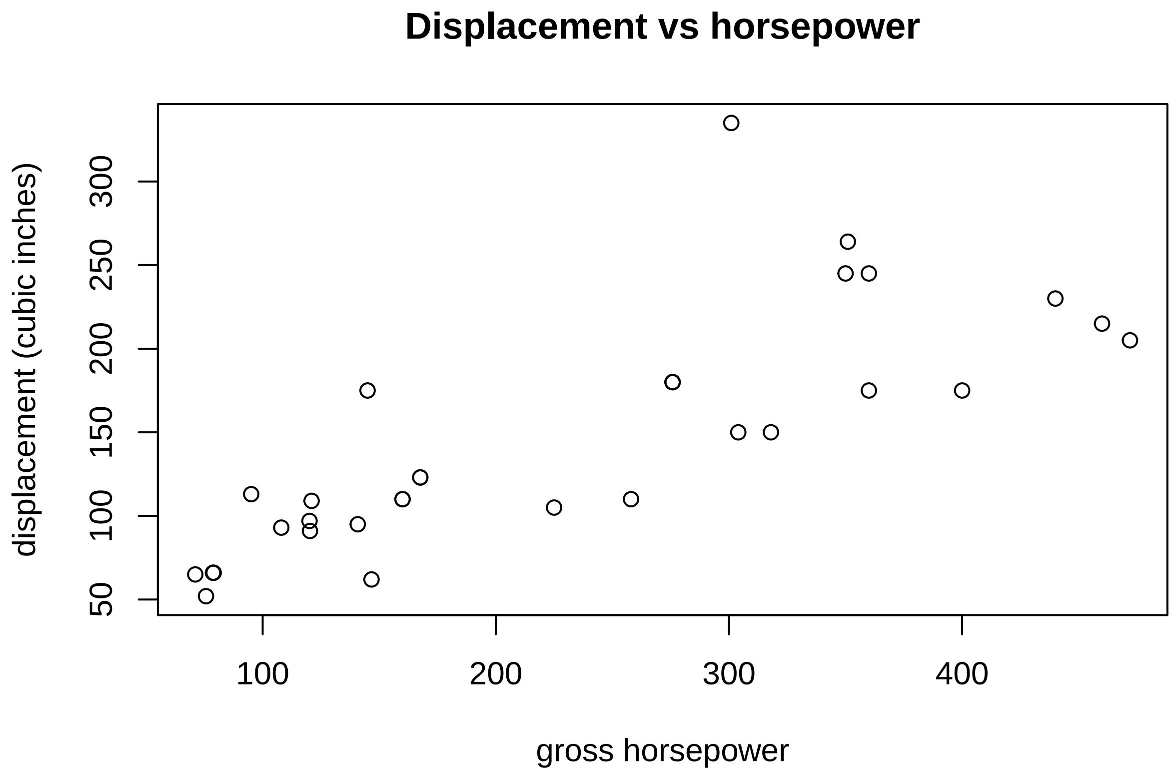
```
plot(mtcars$wt,
     mtcars$mpg,
     main="Miles per gallon vs weight",
     xlab="weight (1000s of lbs)",
     ylab="Miles (US gallon)")
```




```
plot(mtcars$cyl,
     mtcars$mpg,
     main="Miles per gallon vs number of cylinders",
     xlab="number of cylinders",
     ylab="Miles (US gallon)")
```



```r
plot(mtcars$disp,
     mtcars$hp,
     main="Displacement vs horsepower",
     xlab="gross horsepower",
     ylab="displacement (cubic inches)")
```



It seems that the miles per gallon correlates negatively with the weight and the number of cylinders. It also seems that the displacement correlates with the horsepower.