Multivariate Data Analysis: Workshop1

2024-01-05

Loading a dataset

We will work on the (famous) Titanic dataset. Load the Titanic data set from a csv file (note that dots were used to represent missing data)

```
Titanic <- read.csv("Titanic.csv",na.strings=".")
#
# Define Sex as factor (0 = Female, 1 = Male)
#
Titanic$Sex <- factor(Titanic$Sex,levels=c(0,1),labels=c("Female","Male"))
```

Simple computations

Table

We can represent the empirical joint distribution table (contingency table, cross tab, 2-way table) containing the counts of each categories.

```
table(Titanic$Sex,Titanic$Survived)
```

```
##
##          No Yes
##   Female 154 308
##   Male   708 142
```

We can transform this into proportion table:

```
table(Titanic$Sex,Titanic$Survived)/nrow(Titanic)
```

```
##
##                 No       Yes
##   Female 0.1173780 0.2347561
##   Male   0.5396341 0.1082317
```

The function `ftable` does the same thing but with a “flat” matrix.

Marginal Distributions

The empirical marginal distributions are obtained by considering only the column of interest (i.e. the table contain a sample from (X,Y), so the first column is a sample from X). We can compute, run tests and infer on these samples as usually in the univariate case.

```
table(Titanic$Survived)/nrow(Titanic) #proportion of survivants
```

```
##
##        No       Yes
## 0.6570122 0.3429878
```

It is also possible to add margins to a contingency table:

```r
tableTitanic <- table(Titanic$Sex,Titanic$Survived)/nrow(Titanic)
addmargins(tableTitanic)
```

```
##
##                 No       Yes       Sum
##   Female 0.1173780 0.2347561 0.3521341
##   Male   0.5396341 0.1082317 0.6478659
##   Sum    0.6570122 0.3429878 1.0000000
```

Conditional distributions

Now we want to understand the dependencies between the observations. The conditional density between X and Y writes:

$$f_{X|Y}(x \mid y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

Ex: How do you estimate the probability of a woman to survive?

By using the help, find how to use `prop.table` to do these computations.

Independence tests

By using `summary` on a table, you can perform independence χ^2 tests:

```r
summary(table(Titanic$Sex,Titanic$Survived))
```

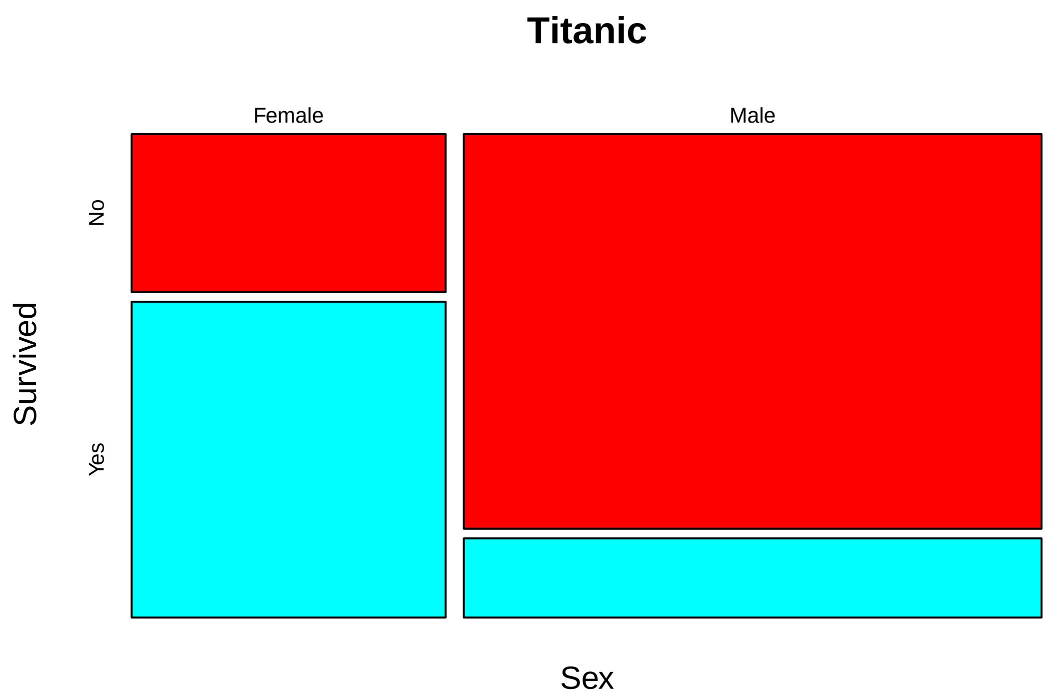
```
## Number of cases in table: 1312
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 331.5, df = 1, p-value = 4.444e-74
```

Remind yourself how to read this result.

Some graphical representations

A simple way to graphically represent a table is to produce a *mosaic*:

```r
plot(tableTitanic,main="Titanic",ylab="Survived",xlab="Sex",col=rainbow(2))
```

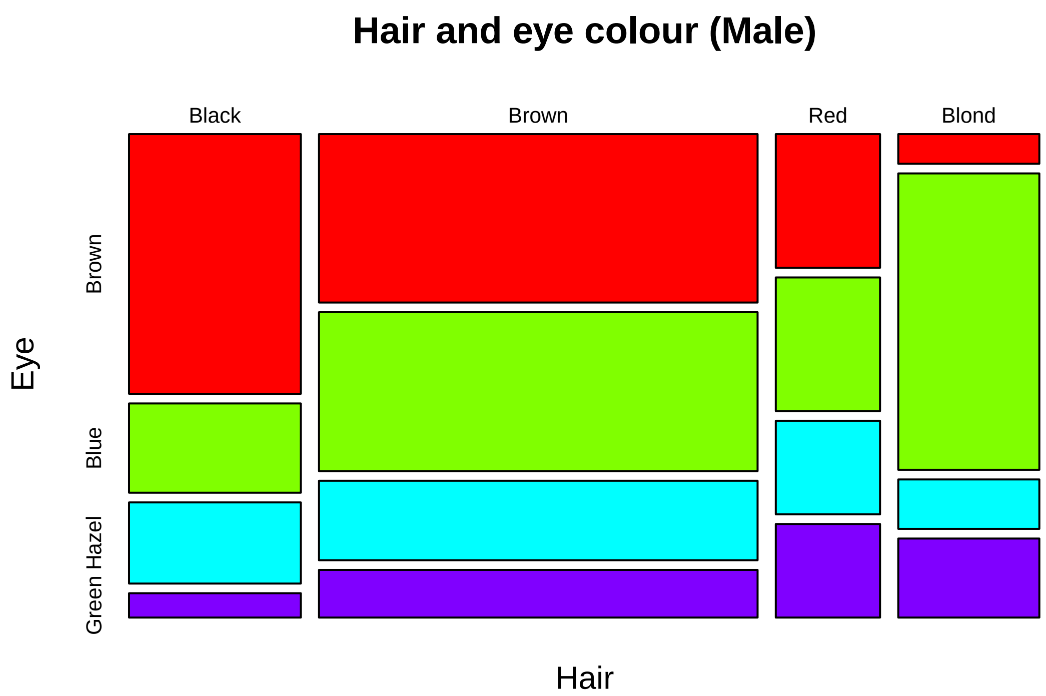


With a larger dataset, you can envision a larger spectrum of colors, for example using the `HairEyeColor` dataset.

```
a <- as.table(HairEyeColor[,,"Male"]) # Extract table for Male category
a
```

```
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    32   11    10     3
##   Brown    53   50    25    15
##   Red      10   10     7     7
##   Blond     3   30     5     8
```

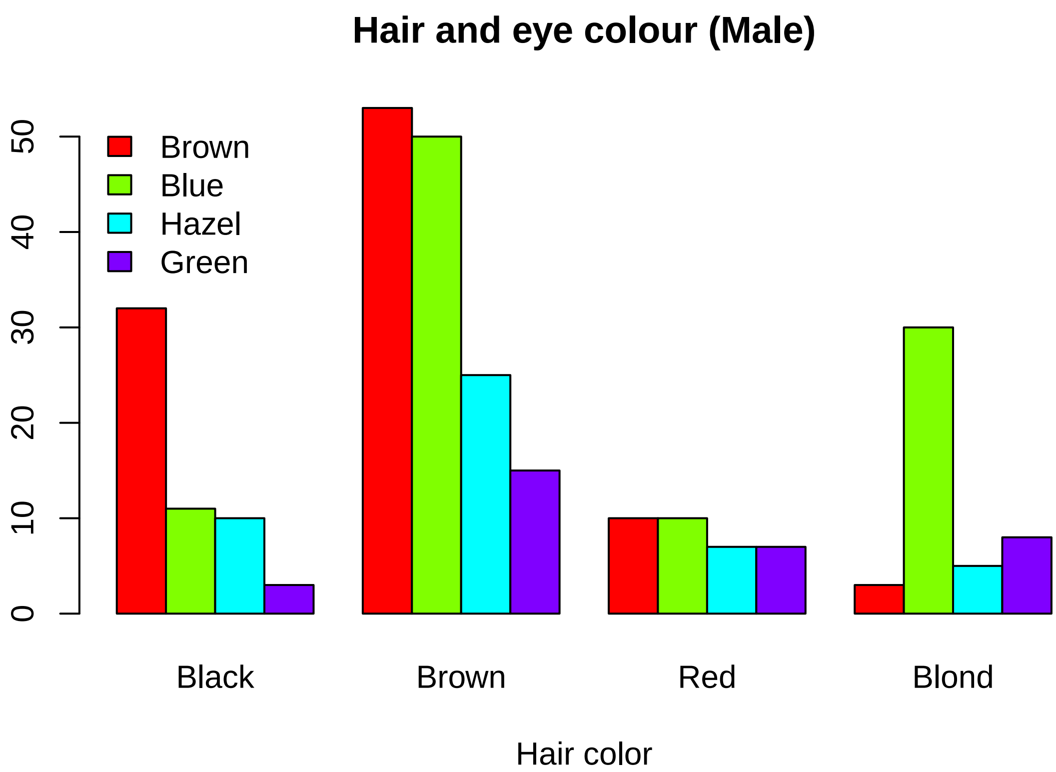
```
plot(a,main="Hair and eye colour (Male)",ylab="Eye",xlab="Hair",col=rainbow(4))
```



Ex. Do the same with `b` which will contain the same results for the Female category.

You can represent all the results of this bivariate categorical distribution on a grouped barplot:

```
barplot(t(a),main="Hair and eye colour (Male)",col=rainbow(4),beside=T,xlab="Hair color",
legend=T,args.legend=list(x="topleft",bty="n"))
```



More than two variates

Use the `ftable` function on the `HairEyeColor` dataset, and understand it.

Load the `Personal.csv` dataset it contains 4 variables, check how to use the `row.vars` and `col.vars` arguments to change the representation of the table.

Numerical variables

In the case of continuous variables, covariance and correlation are better way to understand relationships between variables:

$$S_{x,y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{n}, \quad r_{x,y} = \frac{S_{x,y}}{S_x S_y}.$$

Use the `airquality` dataset, considering the first 4 variates to be continuous. Check the help on this dataset.

```
air=airquality[,1:4]
```

There is missing data on this dataset. Check how the different functions react to the way to treat the missing data. For example:

```
colMeans(air)
```

```
##     Ozone   Solar.R      Wind      Temp
##        NA        NA  9.957516 77.882353
```

```
colMeans(air,na.rm=T)
```

```
##      Ozone    Solar.R       Wind       Temp
##  42.129310 185.931507   9.957516  77.882353
```

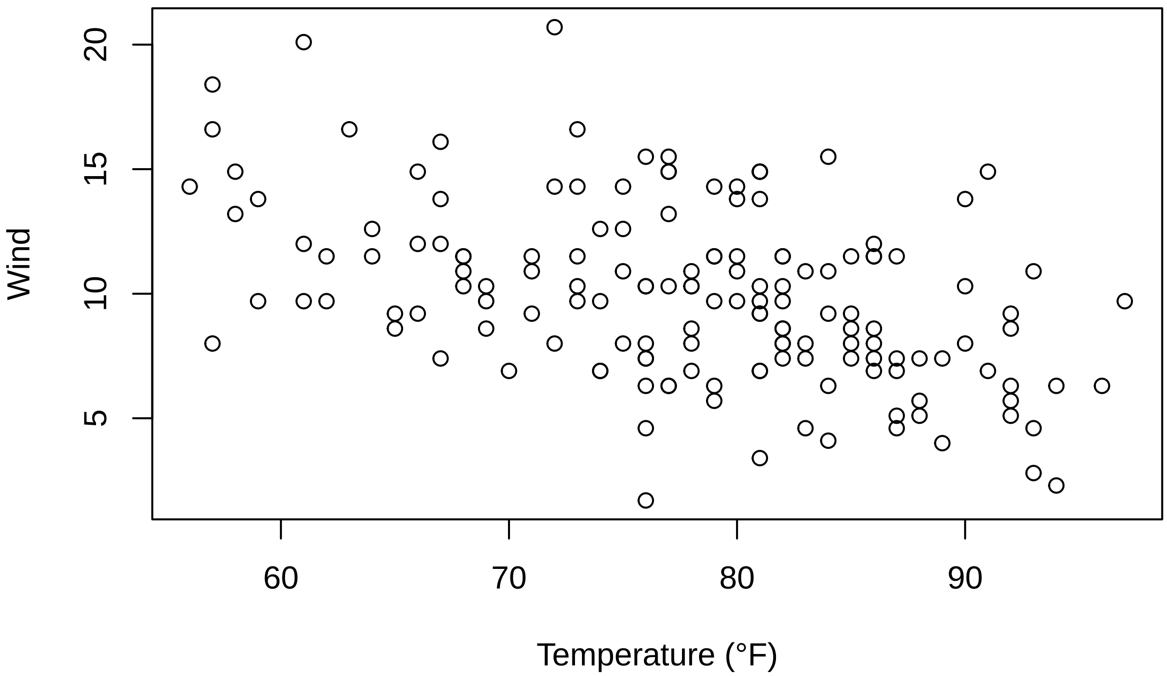
Use the `apply` function to compute empirical variances of the variables.

You can use the `cor` function to plot correlation matrix, the `cov` function to plot the covariance matrix and the `cov2cor` function to convert a covariance matrix into a correlation matrix.

Visualisation of the relationships

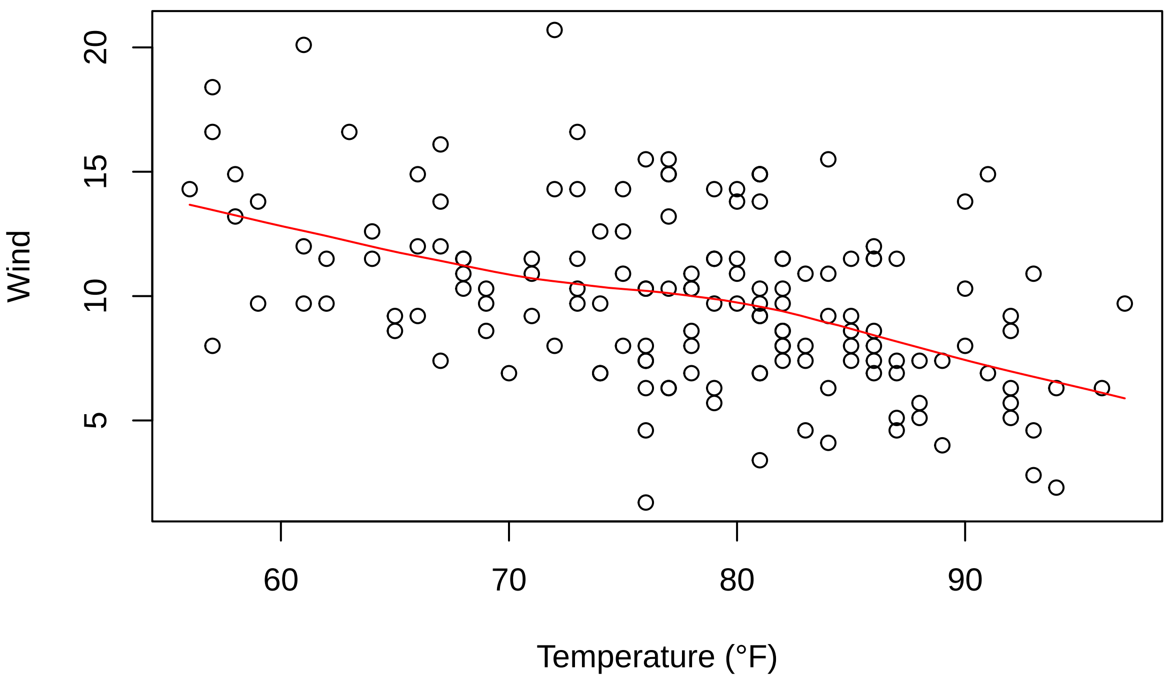
First, you can produce simple scatterplot of two variables.

```
plot(air$Temp,air$Wind,ylab="Wind",xlab="Temperature (°F)")
```



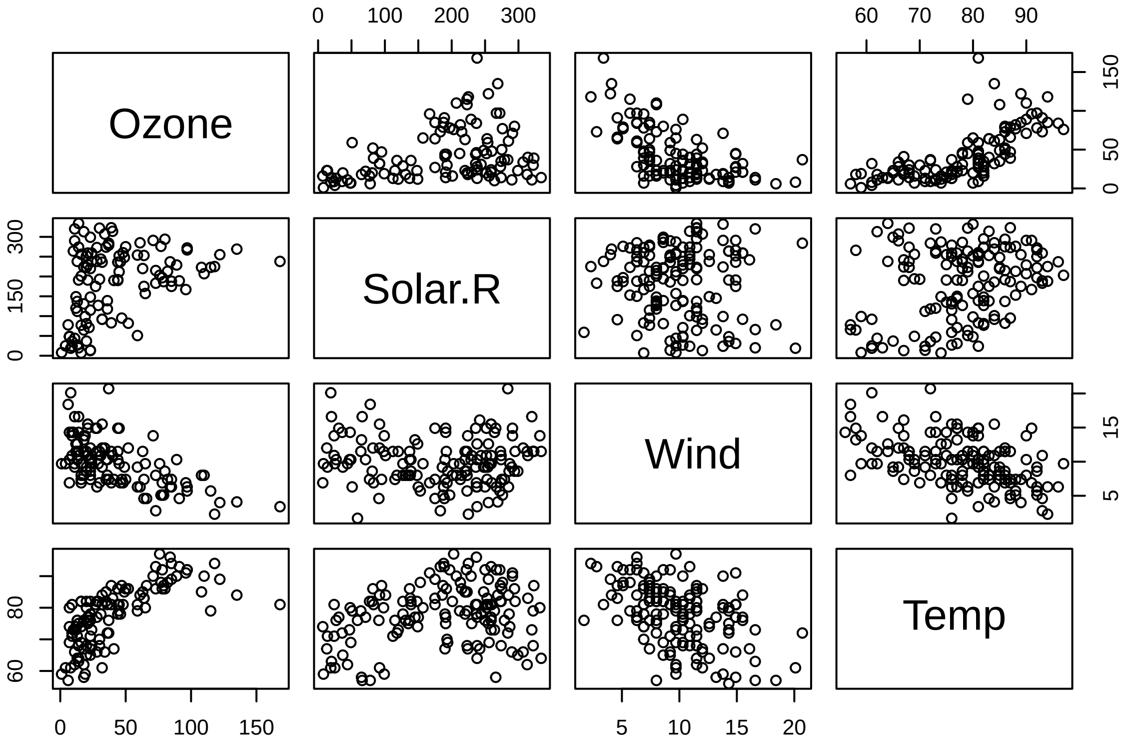
You can then fitt a Loess curve to this plot:

```
scatter.smooth(air$Temp,air$Wind,ylab="Wind",xlab="Temperature (°F)",lpars=list(col="red"))
```



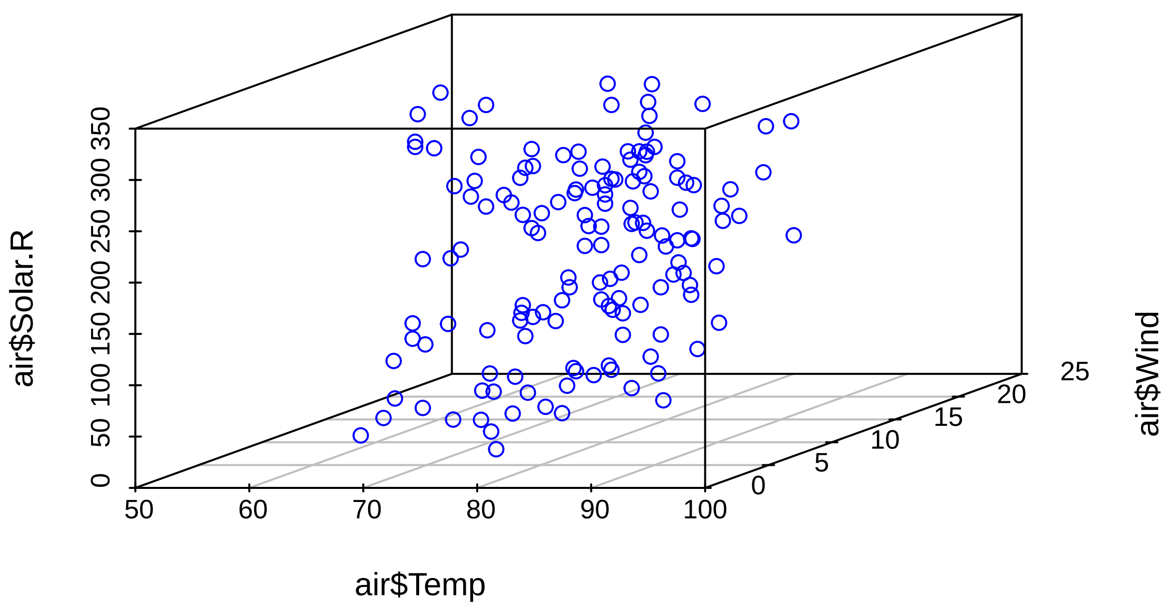
You can visualise all the pairs of variables at once using:

```
plot(air) #same as pairs()
```



For three variables, you can use the less readable 3D scatterplots:

```r
#install.packages("scatterplot3d")
library(scatterplot3d)
scatterplot3d(air$Temp,air$Wind,air$Solar.R,color="blue")
```



Ex. Do the same with the `iris` dataset.