Univariate data analysis
Loading Nations.txt

- We want to load Nations.txt located in ...
- C:/Program Files/R/R-2.13.0/library/Rcmdr/etc/ ...
- And call it mydata
Extracting variables from the data set

- To refer to the variables we type
  `name-dataset$name-variable`
- Put the sign $ between name of the data set and the variable you want to see.

```r
names(mydata)

mydata$GDP
```
names(mydata)

# put the dollar sign between the name of the data and of variable
# to get the values of the variable
mydata$GDP
Graphical displays - barchart
Graphical displays - barchart cont.

- This one is with the default settings
Graphical displays - barchart cont.

- Use the **Script Window** to obtain 'pretty' barchart: **col** to set up the color, and **main**, for the main title, and store frequencies/stats in a variable **b** by writing **b = barplot(...)**

```r
barplot(table(mydata$region), xlab="region", ylab="Frequency", col="blue", main="My Barchart")
```

# For all options of command barplot, type:

```
?barplot
```
b <- barplot(table(mydata$region), xlab = "region", ylab = "Frequency", col = "blue", main = "MY FIRST, REALLY COOL BARCHART")

> b

[,1]
[1,] 0.7
[2,] 1.9
[3,] 3.1
[4,] 4.3
[5,] 5.5
This is the result

![Graphical display - bar chart](image)

**MY FIRST, REALLY COOL BARCHART**

- **Africa**
- **Americas**
- **Asia**
- **Europe**
- **Oceania**

**Y-axis**: Frequency

**X-axis**: Region
A histogram is a graphical display of tabulated frequencies, shown as bars. It shows what proportion of cases fall into each of several categories.

Procedure:

Graph ⇒ Histogram
Select the variable of interest
Select the axis scaling
OK
Graphical displays - histogram

```
R Commander

Data set: mydata

Script Window

h <- hist(mydata$GDP, right=FALSE, col="red")

> h
$breaks
[1] 0 5000 10000 15000 20000 25000 30000 35000 40000 45000

$counts
[1] 138 18 9 12 4 10 2 1 3

$intensities
[6] 1.015228e-05 2.030457e-06 1.015228e-06 3.045685e-06
```
Graphical displays - histogram

- For all options of command `hist`, type:
  `?hist`

- Use the menu or/and modify in the **Script Window** to change color, etc and get stats

- Set `right` to `FALSE` to exclude right-end point of the intervals

  ```r
  hist(mydata$GDP, right=FALSE, col="red")
  ```

- Other nice options, using for example,

  ```r
  xlab="GDP", main="My Histogram"
  ```
Graphical displays - histogram cont.

- This is the result

Histogram of mydata$GDP

mydata$GDP
Frequency
0 10000 20000 30000 40000
0 20 40 60 80 100 120 140
A boxplot graphically visualise data through their five-number summaries: the smallest observation (minimum), lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation (maximum).

A quartile is any of the three values which divide the sorted dataset into four equal parts, so that each part represents one fourth of the sampled population.

Outliers, points which are more than 1.5 the interquartile range (Q3-Q1) away from the interquartile boundaries are marked individually.
Graphical displays - boxplot

- **Select** the variable of interest
- **Plot by groups**: allows you to have boxplots side by side by splitting the variable by a categorical variable.
- **Identify outliers with mouse**: this option allows you to hover over a outlier data point and determine its position in the dataset.
- **OK**
Graphical displays - boxplot
Graphical displays - boxplot

- For all options of command boxplot, type: `?boxplot`
- Use the menu or/and modify in the Script Window to change color, etc and get stats

```r
boxplot(GDP ~ region, ylab="region", data=mydata, col=1:5)
```
Graphical displays - boxplot cont.

- Can be obtained by group if applicable (here by region)
Saving graphs
Numerical summaries

- mean, quasi-standard deviation, min, first quartile, median (second quartile), third quartile, max, sample size, number of missing values

```r
> numSummary(mydata[,c("contraception", "GDP", "infant.mortality", "TFR")],
+   groups=mydata$region, statistics=c("mean", "sd", "quantiles"),
+   quantiles=c(0,.25,.5,.75,1))

Variable: contraception

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
<th>0%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>100%</th>
<th>n</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>23.41463</td>
<td>18.94726</td>
<td>2</td>
<td>8.0</td>
<td>18.0</td>
<td>33.0</td>
<td>75</td>
<td>41</td>
<td>14</td>
</tr>
<tr>
<td>Americas</td>
<td>54.90625</td>
<td>12.99220</td>
<td>18</td>
<td>48.5</td>
<td>54.5</td>
<td>64.0</td>
<td>75</td>
<td>32</td>
<td>9</td>
</tr>
<tr>
<td>Asia</td>
<td>42.61290</td>
<td>23.74963</td>
<td>7</td>
<td>19.5</td>
<td>36.0</td>
<td>62.5</td>
<td>86</td>
<td>31</td>
<td>10</td>
</tr>
<tr>
<td>Europe</td>
<td>60.20000</td>
<td>20.99603</td>
<td>17</td>
<td>50.0</td>
<td>70.0</td>
<td>76.0</td>
<td>82</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>Others</td>
<td>47.42920</td>
<td>12.54340</td>
<td>15</td>
<td>25.5</td>
<td>40.0</td>
<td>61.0</td>
<td>70</td>
<td>15</td>
<td>19</td>
</tr>
</tbody>
</table>
```
Numerical summaries

- **Statistics ⇒ Summaries ⇒ Numerical summary**
- If you have multiple groups (e.g. control versus treatment) click on **summarize by groups** and select the appropriate variable
- **OK**
Numerical summaries

Understanding the output:

<table>
<thead>
<tr>
<th>parameter</th>
<th>What is it?</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>Measure of central tendency</td>
</tr>
<tr>
<td>sd</td>
<td>Standard deviation - a measure of variability in the data</td>
</tr>
<tr>
<td>N</td>
<td>Number of readings</td>
</tr>
<tr>
<td>NA</td>
<td>Number of missing values</td>
</tr>
<tr>
<td>0%</td>
<td>Minimum value</td>
</tr>
<tr>
<td>25%</td>
<td>The value below which 25 percent of the observations may be found.</td>
</tr>
<tr>
<td>50%</td>
<td>The value below which 50 percent of the observations may be found.</td>
</tr>
<tr>
<td>75%</td>
<td>The value below which 75 percent of the observations may be found.</td>
</tr>
<tr>
<td>100%</td>
<td>Maximum value</td>
</tr>
</tbody>
</table>
Numerical summaries

- Can be obtained by group if applicable (here by region)
Numerical summaries

Coefficient of Variation: \[ CV = \frac{S}{\bar{X}} \]

- Coefficient of variation by hand (compute the mean and SD ignoring the missing values coded as NA!)

\[ s = sd(mydata$contraception, na.rm=TRUE) \]
\[ xbar = mean(mydata$contraception, na.rm=TRUE) \]
\[ CV = s/xbar \]
\[ CV \]
Numerical summaries

```r
# coefficient of variation

s <- sd(mydata$contraception, na.rm = TRUE)
xbar <- mean(mydata$contraception, na.rm = TRUE)

CV <- s / xbar

CV

[1] 0.5458538
```
Numerical summaries

Coefficient of kurtosis and skewness:

\[ b_2 = \frac{m_4}{s^4} - 3 \]

\[ b_1 = \frac{m_3}{s^3} \]

▶ You have to load the library `e1071`

```r
library(e1071)
?kurtosis
?skewness

kurtosis(mydata$contraception, na.rm=TRUE)

skewness(mydata$contraception, na.rm=TRUE)
```
Categorical variables are measures on a nominal scale i.e. where you use labels.
The values that can be taken are called levels.
Categorical variables have no numerical meaning, but are often coded for easy of data entry and processing in spreadsheets.
For example, gender is often coded where male=1 and female=2. Data can thus be entered as characters (e.g. 'normal') or numeric (e.g. 0, 1, 2).
Frequency distribution - categorical data

```r
> Table <- table(mydata$region)
> Table # counts for region

Africa Americas Asia Europe Oceania
   55    41   41    45   25

> 100*Table/sum(Table) # percentages for region

Africa Americas Asia Europe Oceania

> remove(Table)
```
Use the **Script Window** to obtain the frequency distribution.

First load the library **agricolae**, then get the stats from the histogram, then use **table.freq**

```r
library(agricolae)

h = hist(mydata$contraception, right=FALSE, plot=FALSE)

table.freq(h)
```
Frequency distribution - numerical data

```r
library(agricolae)
h<-hist(mydata$contraception, right=FALSE)
table.freq(h)
```

```
> library(agricolae)

> h<-hist(mydata$contraception, right=FALSE)

> table.freq(h)

<table>
<thead>
<tr>
<th></th>
<th>Sup</th>
<th>MC</th>
<th>fi</th>
<th>fri</th>
<th>Fi</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>5</td>
<td>14</td>
<td>0.097222222</td>
<td>14 0.09722222</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>15</td>
<td>18</td>
<td>0.125000000</td>
<td>32 0.22222222</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>25</td>
<td>17</td>
<td>0.118055556</td>
<td>49 0.34027778</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>40</td>
<td>35</td>
<td>13</td>
<td>0.090277778</td>
<td>62 0.43055556</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>45</td>
<td>15</td>
<td>0.104166667</td>
<td>77 0.53472222</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>60</td>
<td>55</td>
<td>23</td>
<td>0.159722222</td>
<td>100 0.69444444</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>70</td>
<td>65</td>
<td>18</td>
<td>0.125000000</td>
<td>118 0.81944444</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>75</td>
<td>23</td>
<td>0.159722222</td>
<td>141 0.97916667</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>90</td>
<td>85</td>
<td>3</td>
<td>0.020833333</td>
<td>144 1.00000000</td>
<td></td>
</tr>
</tbody>
</table>
```
Modifying the dataset: Compute a new variable

- **Data** ⇒ **Manage variables in active dataset** ⇒ **compute new variables**
- Enter new variable name
- An expression (equation) is written to reflect the calculation required.

![Image of Compute New Variable window]
Modifying the dataset: Compute a new variable

The table below indicates the operators available and examples of how it could be used. Double clicking on a variable in the current variables box will send the variable to the expression.

<table>
<thead>
<tr>
<th>Operators</th>
<th>Function</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x + y</td>
<td>Addition</td>
<td>Variable 1 + Variable 2</td>
<td>Variable 1 + 25</td>
</tr>
<tr>
<td>x - y</td>
<td>Subtraction</td>
<td>Variable 1 – Variable 2</td>
<td>35 - Variable 1</td>
</tr>
<tr>
<td>x * y</td>
<td>Multiple</td>
<td>Variable 1*Variable 2</td>
<td>100*Variable 1</td>
</tr>
<tr>
<td>x / y</td>
<td>Division</td>
<td>Variable 1/Variable 2</td>
<td>Variable 1 / 63</td>
</tr>
<tr>
<td>x ^ y</td>
<td>X to the power of Y</td>
<td>Variable 1 ^ Variable2</td>
<td>Variable1^10</td>
</tr>
<tr>
<td>log10(x)</td>
<td>Log10 transformation</td>
<td>Log10(Variable 1)</td>
<td></td>
</tr>
<tr>
<td>log(x, base)</td>
<td>Log transformation to a specified base</td>
<td>Log(Variable 1, 2)</td>
<td></td>
</tr>
</tbody>
</table>
Converting numeric variables to factors

- Data ⇒ Manage variables in active dataset ⇒ Convert numeric variables to factors
- Select the variables.
Converting numeric variables to factors

- You can generate a new variable by entering a name in box <new variable name> or over-write the original name.
  1. The levels can be formatted as Levels by selecting <use numbers>
  2. Recoded to a name by selecting <supply level names>
- OK
Sub-dividing data by columns (variables)

- Data $\Rightarrow$ active dataset $\Rightarrow$ subset active dataset
- Hold the CTRL key to select the variables you wish to keep
- Give the new dataset a name
Sub-dividing data by rows (and variables if you wish)

- **Data** ⇒ **active dataset** ⇒ **subset active dataset**
- Select the variables you wish to include in the new dataset
- Write a **subset expression** which is a rule to drive the selection of rows
Sub-dividing data by rows (and variables if you wish)

**Note**: If you use a name in an expression you need to surround the name with double quotes e.g. ”name”

**Example**: `GENDER == "Female" & AGE ≤ 25`

<table>
<thead>
<tr>
<th>Symbol/code</th>
<th>Name</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equality</td>
<td>used to indicate the variable should equal</td>
</tr>
<tr>
<td>!=</td>
<td>Inequality</td>
<td>used to indicate the variable should not equal</td>
</tr>
<tr>
<td>&amp;</td>
<td>And</td>
<td>used to combine multiple expressions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Or</td>
</tr>
<tr>
<td>is.na(varname)</td>
<td></td>
<td>Include the missing values of a variable</td>
</tr>
<tr>
<td>!is.na(varname)</td>
<td></td>
<td>Exclude the missing values of a variable</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>More than or equal to</td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>Less than or equal to</td>
<td></td>
</tr>
</tbody>
</table>
Plot time series

**Note:** Time series are plotted with a different method with respect to usual variables.

**Example:** Simulate 24 observations from a given time series. Plot observations.

```r
x = rnorm(24) + 100
plot(ts(x, start=1992), ylab="levels")
```
Example: Simulate 100 observations from a time series given two years.

Note: Better use the library lattice

```r
thing = data.frame(rnorm(100,10,2),
c(rep("A",50), rep("B",50)))

colnames(thing) <- c("Returns","Year")
X11()
dotchart(thing$Returns, xlab="Returns")

X11()
dotplot(thing$Returns ~ thing$Year, ylab="Returns", xlab="years")
```
DotPlots II
DotPlots III

![Graph showing dot plots for returns over years A and B.](image)