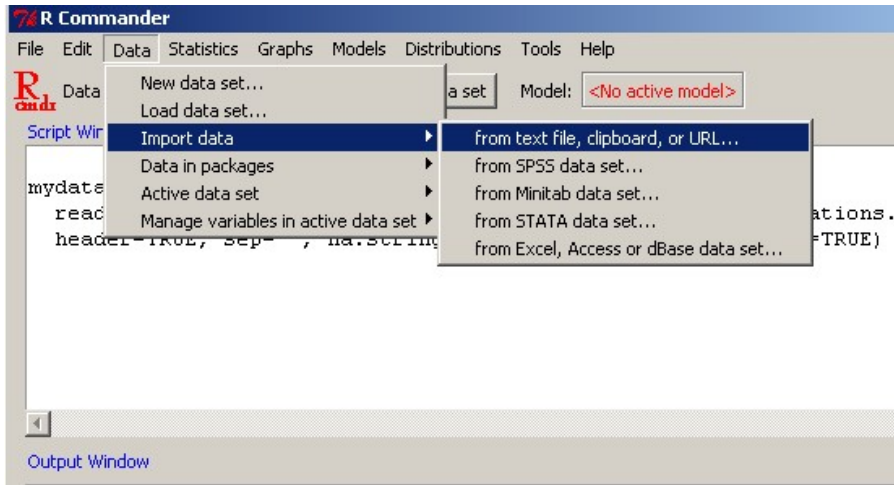


Univariate data analysis

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.

```
names(mydata)
```

```
mydata$GDP
```

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **mydata** Edit data set View data set Model: <No active model>

Script Window

```
names(mydata)

# put the dollar sign between the name of the data and of variable
# to get the values of the variable
mydata$GDP
```

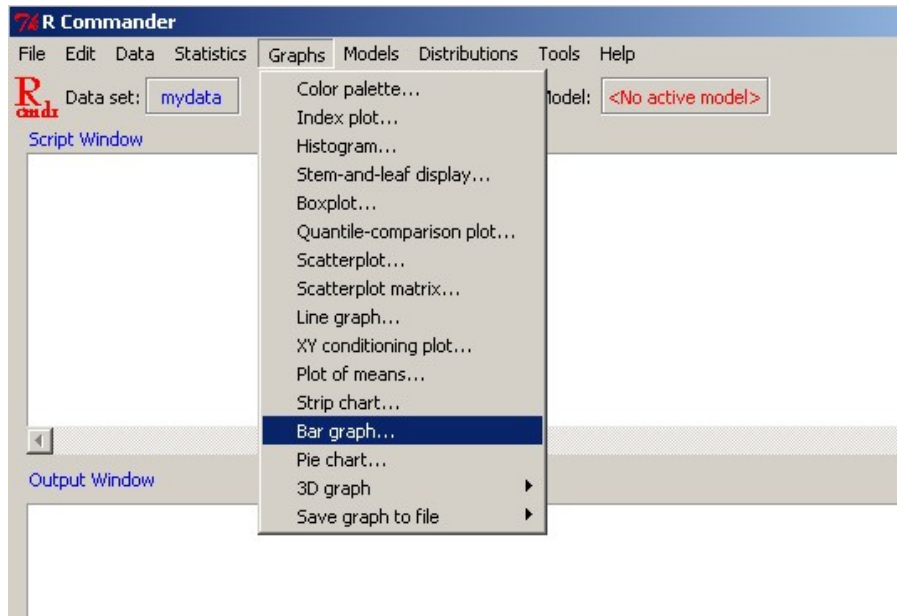
Output Window

```
> names(mydata)
[1] "TFR"                "contraception"      "infant.mortality"  "GDP"
[5] "region"

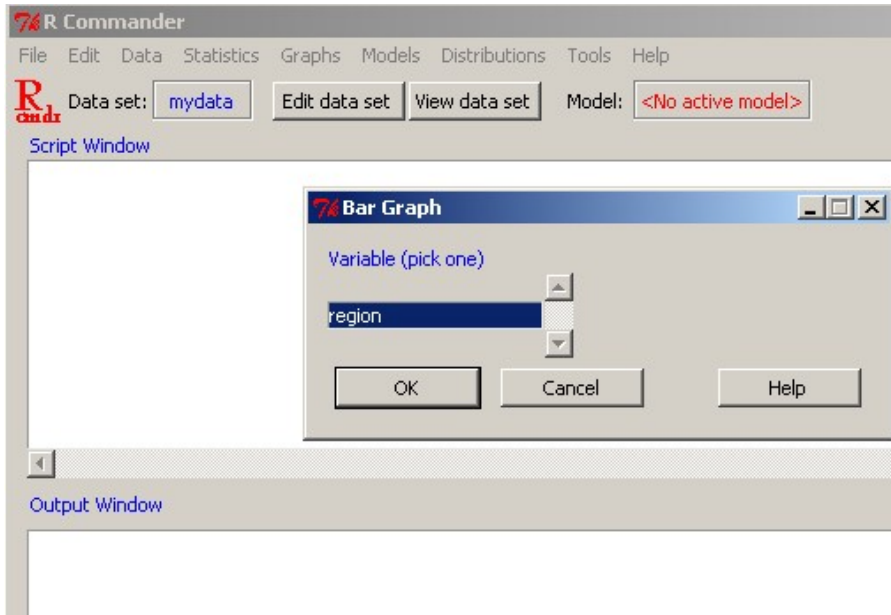
# put the dollar sign between the name of the data and of variable
# to get the values of the variable

> mydata$GDP
  [1] 2848  863 1531  NA   NA   355  6966  8055  354 20046 29006
 [13] 12545 9073 280 7173  994 26582 2569  391  166  909  271
 [25] 4510 16683 1518 165 205 130  627 18943 994 379 187
 [37]  582 2215 367 1008 5432 2696 4014 1983 11459 4450 117
 [49]  893 2831 1508  NA 1565  973 1660 388  96 2433  96
```

Graphical displays - barchart

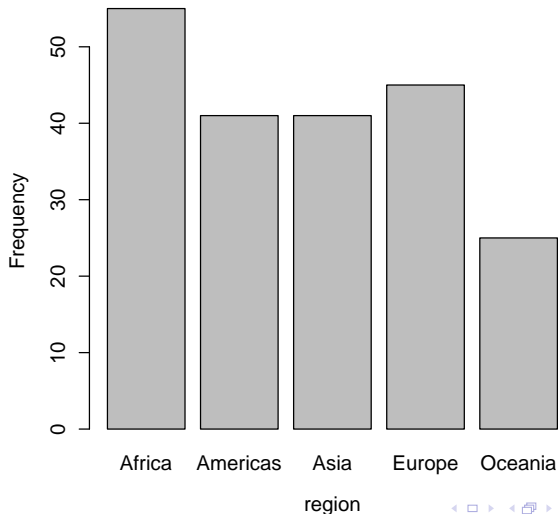


Graphical displays - barchart cont.



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(...)**

```
barplot(table(mydata$region),xlab="region",  
ylab="Frequency",col="blue",main="My Barchart")  
  
# For all options of command barplot, type:  
  
?barplot
```




Data set:

mydata

Edit data set

View data set

Model:

<No active model>

Script Window

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

b

|



Output Window

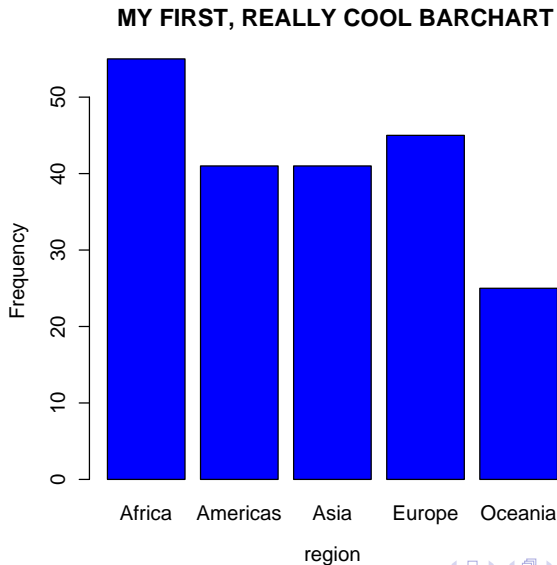
```
> b<-barplot(table(mydata$region), xlab="region", ylab="Frequency",  
+           col="blue", main="MY FIRST, REALLY COOL BARCHART")
```

> b

```
      [,1]  
[1,] 0.7  
[2,] 1.9  
[3,] 3.1  
[4,] 4.3  
[5,] 5.5
```

Graphical displays - barchart cont.

- This is the result



Graphical displays - histogram

- ▶ 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:

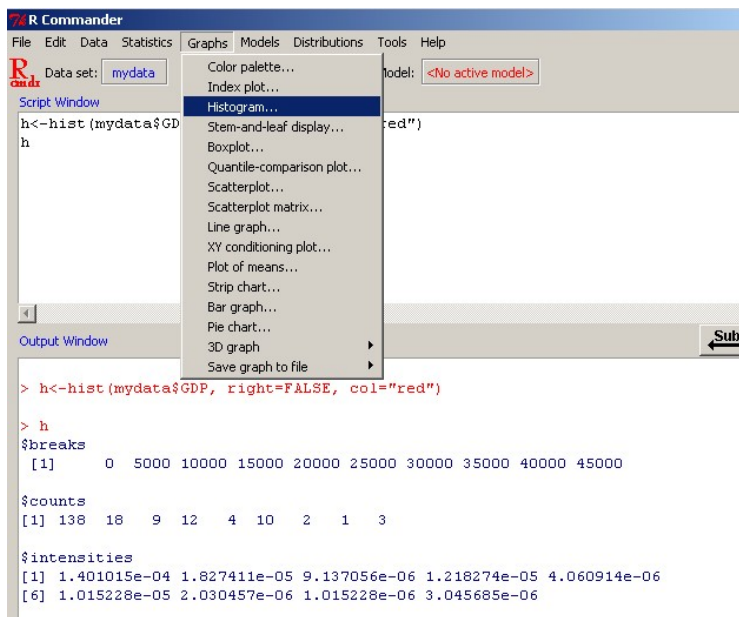
Graph \Rightarrow Histogram

Select the variable of interest

Select the axis scaling

OK

Graphical displays - histogram



The screenshot shows the R Commander interface. The 'Graphs' menu is open, and 'Histogram...' is selected. The 'Data set' is 'mydata'. The 'Script Window' contains the command `h<-hist(mydata$GDP, right=FALSE, col="red")`. The 'Output Window' shows the results of the histogram command.

Script Window

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

Output 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
[1] 1.401015e-04 1.827411e-05 9.137056e-06 1.218274e-05 4.060914e-06
[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

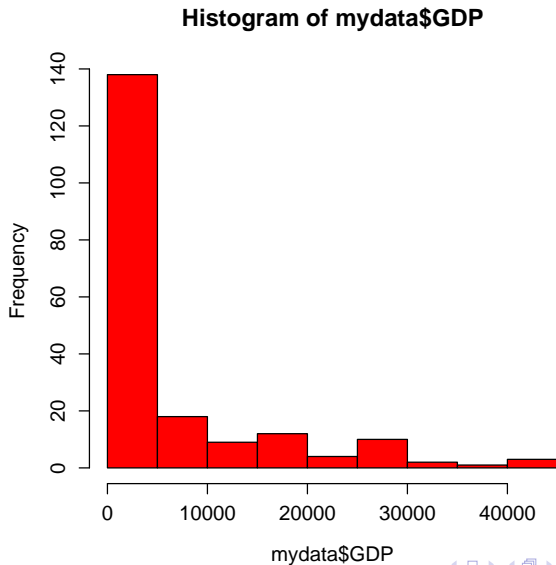
```
hist(mydata$GDP, right=FALSE, col="red")
```

- ▶ Other nice options, using for example,

```
xlab="GDP", main="My Histogram"
```

Graphical displays - histogram cont.

- This is the result



Graphical displays - boxplot

- ▶ 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

The screenshot shows the R Commander interface. The 'Graphs' menu is open, displaying various plotting options. The 'Boxplot...' option is highlighted. In the background, the 'Script Window' contains the command `s<-boxplot(GDP~region, col=1:5)`. The 'Output Window' shows the resulting boxplot object `s` with its structure and data values.

R Commander

File Edit Data Statistics **Graphs** Models Distributions Tools Help

Data set: `mydata` Model: `<No active model>`

Script Window

```
s<-boxplot(GDP~region, col=1:5)
```

Output Window

```
> s<-boxplot(GDP~region, ylab="GDP", xlab="region", data=mydata, col=1:5)
+
> s
$stats
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,]  36.0    386.0    122.0    271.0    654.0
[2,] 207.0   1719.5    345.0   1627.5   1099.5
[3,] 389.5   2765.5   1079.0   9222.5   2348.5
[4,] 1008.0  7069.5   6407.5  26235.0  18158.0
[5,] 2059.0 12717.0  14013.0 42416.0  41718.0
attr(,"class")
      "boxplot"
      "integer"

$n
[1] 54 40 39 44 20
```

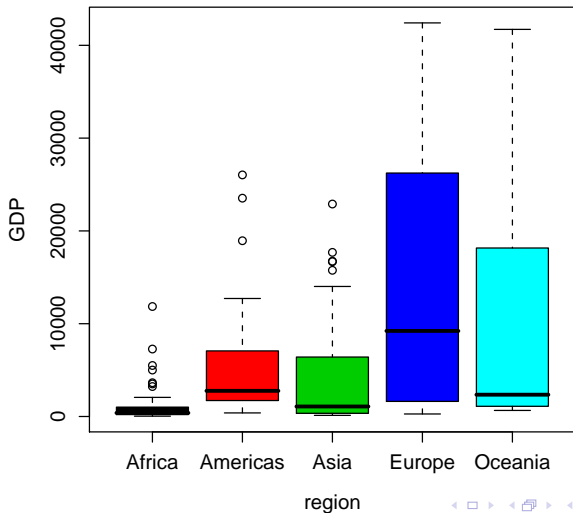
Graphical displays - boxplot

- ▶ 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

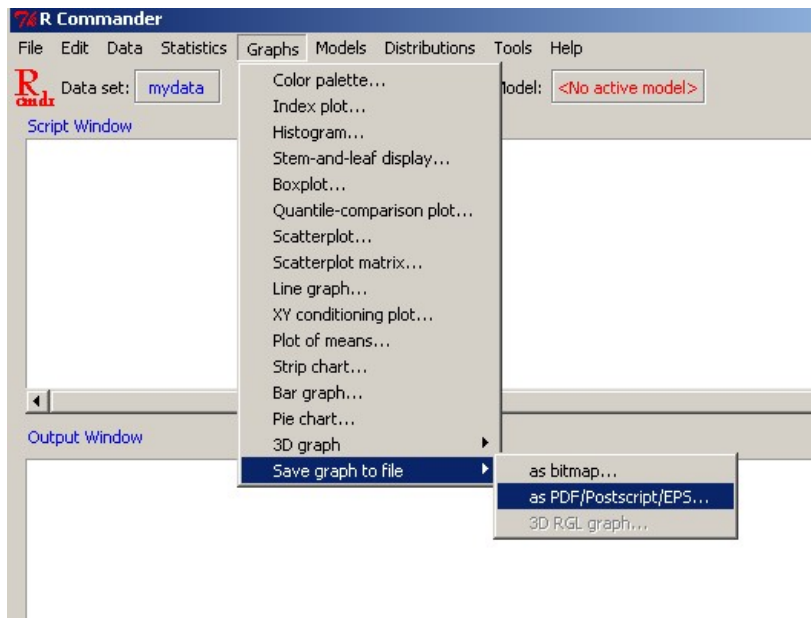
```
boxplot(GDP ~ region, ylab="region",  
data=mydata, col=1:5)
```

Graphical displays - boxplot cont.

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

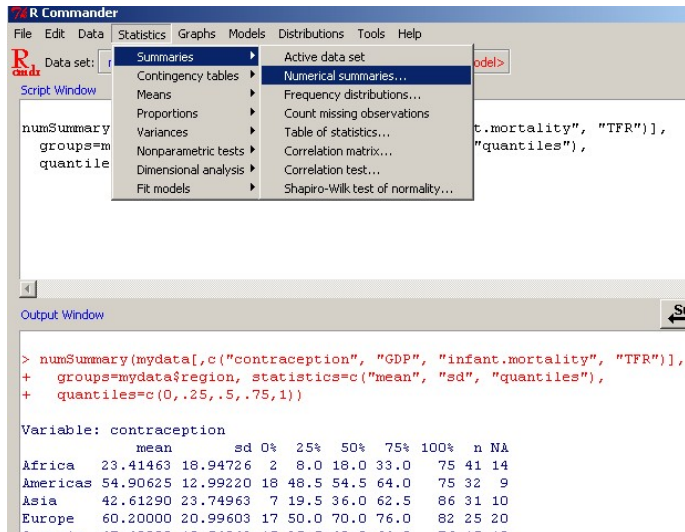


Saving graphs



Numerical summaries

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



R Commander

File Edit Data **Statistics** Graphs Models Distributions Tools Help

Data set: []

Script Window

```
numSummary  
groups=m  
quantile
```

Statistics menu:

- Summaries
 - Active data set
 - Numerical summaries...**
 - Frequency distributions...
 - Count missing observations
 - Table of statistics...
 - Correlation matrix...
 - Correlation test...
 - Shapiro-Wilk test of normality...
- Contingency tables
- Means
- Proportions
- Variances
- Nonparametric tests
- Dimensional analysis
- Fit models

Output Window

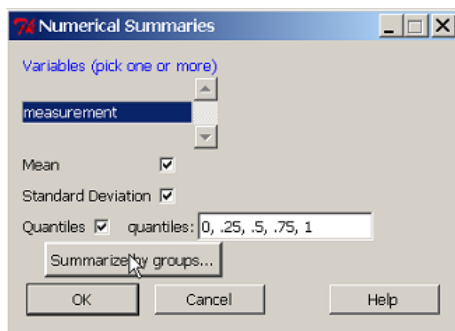
```
> numSummary(mydata[,c("contraception", "GDP", "infant.mortality", "TFR")],  
+ groups=mydata$region, statistics=c("mean", "sd", "quantiles"),  
+ quantiles=c(0,.25,.5,.75,1))
```

Variable: contraception

	mean	sd	0%	25%	50%	75%	100%	n	NA
Africa	23.41463	18.94726	2	8.0	18.0	33.0	75	41	14
Americas	54.90625	12.99220	18	48.5	54.5	64.0	75	32	9
Asia	42.61290	23.74963	7	19.5	36.0	62.5	86	31	10
Europe	60.20000	20.99603	17	50.0	70.0	76.0	82	25	20

Numerical summaries

- ▶ **Statistics** \Rightarrow **Summaries** \Rightarrow **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:

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

Numerical summaries

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

The screenshot shows the R Commander interface with the 'Numerical Summaries' dialog box open. The 'Variables' list includes 'contraception', 'GDP', 'infant.mortality', and 'TFR'. The 'Quantiles' section is set to '0, .25, .5, .75, 1'. The 'Summarize by groups...' button is highlighted with a red circle. Below this, the 'Groups' dialog box is open, with 'region' selected as the 'Groups variable (pick one)'. The 'Output Window' shows the results of the command `numSummary(mydata, groups=mydata$region, quantiles=c(0, .25, .5, .75, 1))`. The output is a table with columns for the variable, mean, sd, and quantiles (0%, 25%, 50%, 75%, 100%), and rows for each region: Africa, Americas, Asia, Europe, and Oceania. The 'region' variable is also highlighted with a red circle in the output table.

R Commander
File Edit Data Statistics Graphs Models Distributions Tools Help
Data set: mydata
Script Window
`numSummary(mydata, groups=mydata$region, quantiles=c(0, .25, .5, .75, 1))`
Output Window
`> numSummary(mydata, groups=mydata$region, quantiles=c(0, .25, .5, .75, 1))`
Variable: contraception
mean sd 0% 25% 50% 75% 100% n NA
Africa 23.41463 12.99228 16 40.3 34.3 64.0 75 41 14
Americas 54.90625 12.99228 16 40.3 34.3 64.0 75 32 9
Asia 42.61290 23.74963 7 19.5 36.0 62.5 86 31 10
Europe 60.20000 20.99603 17 50.0 70.0 76.0 82 25 20
Oceania 47.40000 19.54043 15 35.5 40.0 64.0 76 15 10
Variable: GDP
mean sd 0% 25% 50% 75% 100% n N
Africa 1196.000 2089.614 36 209.00 389.5 1004.50 11854 54

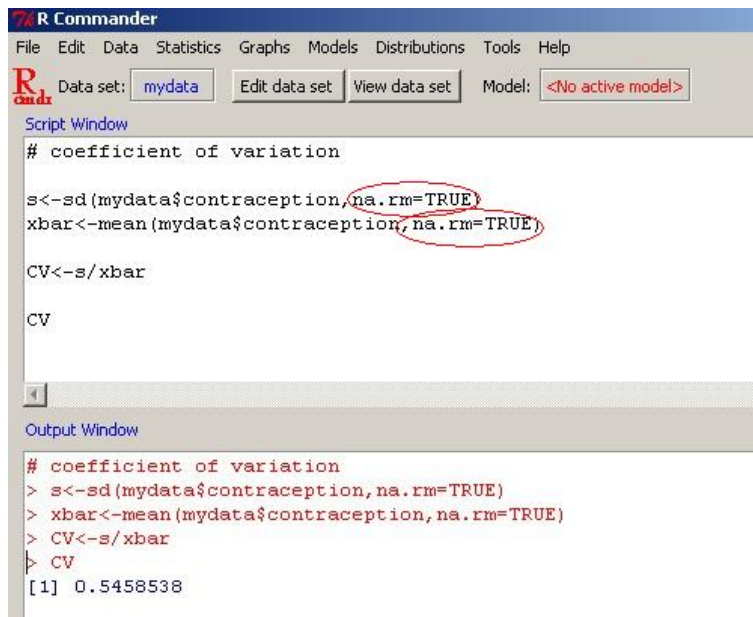
Numerical summaries

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



The screenshot shows the R Commander interface. The menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. The Data set is 'mydata'. The Model is '<No active model>'. The Script Window contains the following R code:

```
# coefficient of variation

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

CV<-s/xbar

CV
```

The Output Window shows the execution results:

```
# 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**

```
library(e1071)
?kurtosis
?skewness

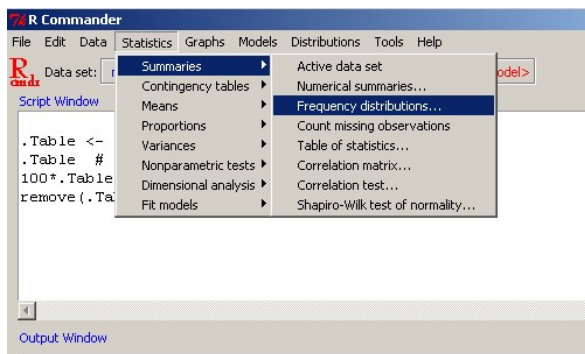
kurtosis(mydata$contraception, na.rm=TRUE)

skewness(mydata$contraception, na.rm=TRUE)
```

Frequency distribution - categorical data

- ▶ **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



The screenshot shows the R Commander application window. The 'Statistics' menu is open, and 'Frequency distributions...' is selected. The 'Script Window' contains the following R code:

```
.Table <-  
.Table #  
100*.Table  
remove(.Table)
```

The 'Output Window' shows the execution results:

```
> .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  
26.57005 19.80676 19.80676 21.73913 12.07729  
  
> remove(.Table)
```

At the bottom right of the R Commander window, there are navigation icons: a set of arrows for navigating between scripts, a magnifying glass for search, and a refresh icon.

Frequency distribution - numerical data

- ▶ 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**

```
library(agricolae)

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

table.freq(h)
```

Frequency distribution - numerical data

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **mydata** Edit data set View data set Model: **<No active model>**

Script Window

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

Output Window

```
> library(agricolae)

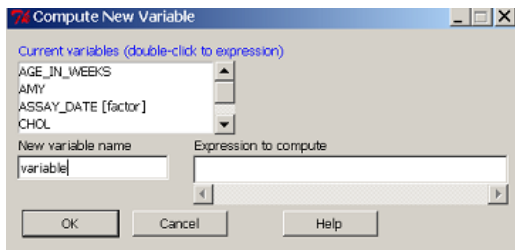
> h<-hist(mydata$contraception, right=FALSE)

> table.freq(h)
```

	Inf	Sup	MC	fi	fri	Fi	Fri
	0	10	5	14	0.09722222	14	0.09722222
	10	20	15	18	0.12500000	32	0.22222222
	20	30	25	17	0.11805556	49	0.34027778
	30	40	35	13	0.09027778	62	0.43055556
	40	50	45	15	0.10416667	77	0.53472222
	50	60	55	23	0.15972222	100	0.69444444
	60	70	65	18	0.12500000	118	0.81944444
	70	80	75	23	0.15972222	141	0.97916667
	80	90	85	3	0.02083333	144	1.00000000

Modifying the dataset: Compute a new variable

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



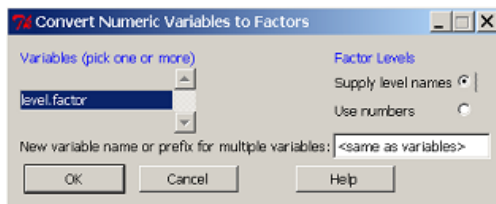
Modifying the dataset: Compute a new variable

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.

Operators	Function	Example 1	Example 2
$x + y$	Addition	Variable 1 + Variable 2	Variable 1 + 25
$x - y$	Subtraction	Variable 1 - Variable 2	35 - Variable 1
$x * y$	Multiple	Variable 1*Variable 2	100*Variable 1
x / y	Division	Variable 1/Variable 2	Variable 1 / 63
$x ^ y$	X to the power of Y	Variable 1 ^ Variable2	Variable1^10
$\log_{10}(x)$	Log10 transformation	$\log_{10}(\text{Variable 1})$	
$\log(x, \text{base})$	Log transformation to a specified base	$\log(\text{Variable 1}, 2)$	

Converting numeric variables to factors

- ▶ **Data** \Rightarrow **Manage variables in active dataset** \Rightarrow **Convert numeric variables to factors**
- ▶ Select the variables.



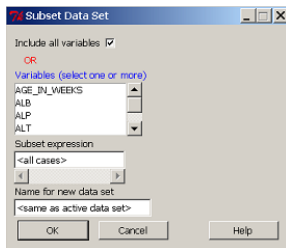
Converting numeric variables to factors

- ▶ 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



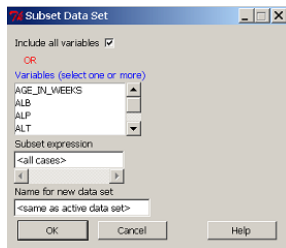
Sub-dividing data by columns (variables)

- ▶ **Data** ⇒ **active dataset** ⇒ **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** \Rightarrow **active dataset** \Rightarrow **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`

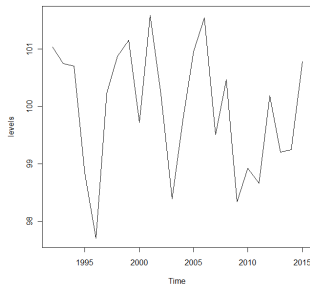
Symbol/code	Name	Use
<code>==</code>	equality	used to indicate the variable should equal
<code>!=</code>	Inequality	used to indicate the variable should not equal
<code>&</code>	And	used to combine multiple expressions
<code> </code>	Or	used to combine multiple expressions
<code>is.na(varname)</code>		Include the missing values of a variable
<code>!is.na(varname)</code>		Exclude the missing values of a variable
<code>></code>	Greater than	
<code><</code>	Less than	
<code>>=</code>		More than or equal to
<code><=</code>		Less than or equal to

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.

```
x = rnorm(24) + 100  
  
plot(ts(x, start=1992), ylab="levels")
```



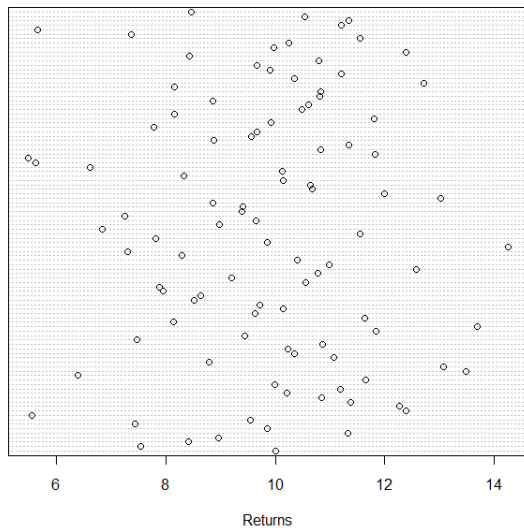
DotPlots I

Example: Simulate 100 observations from a time series given two years.

Note: Better use the library `lattice`

```
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

