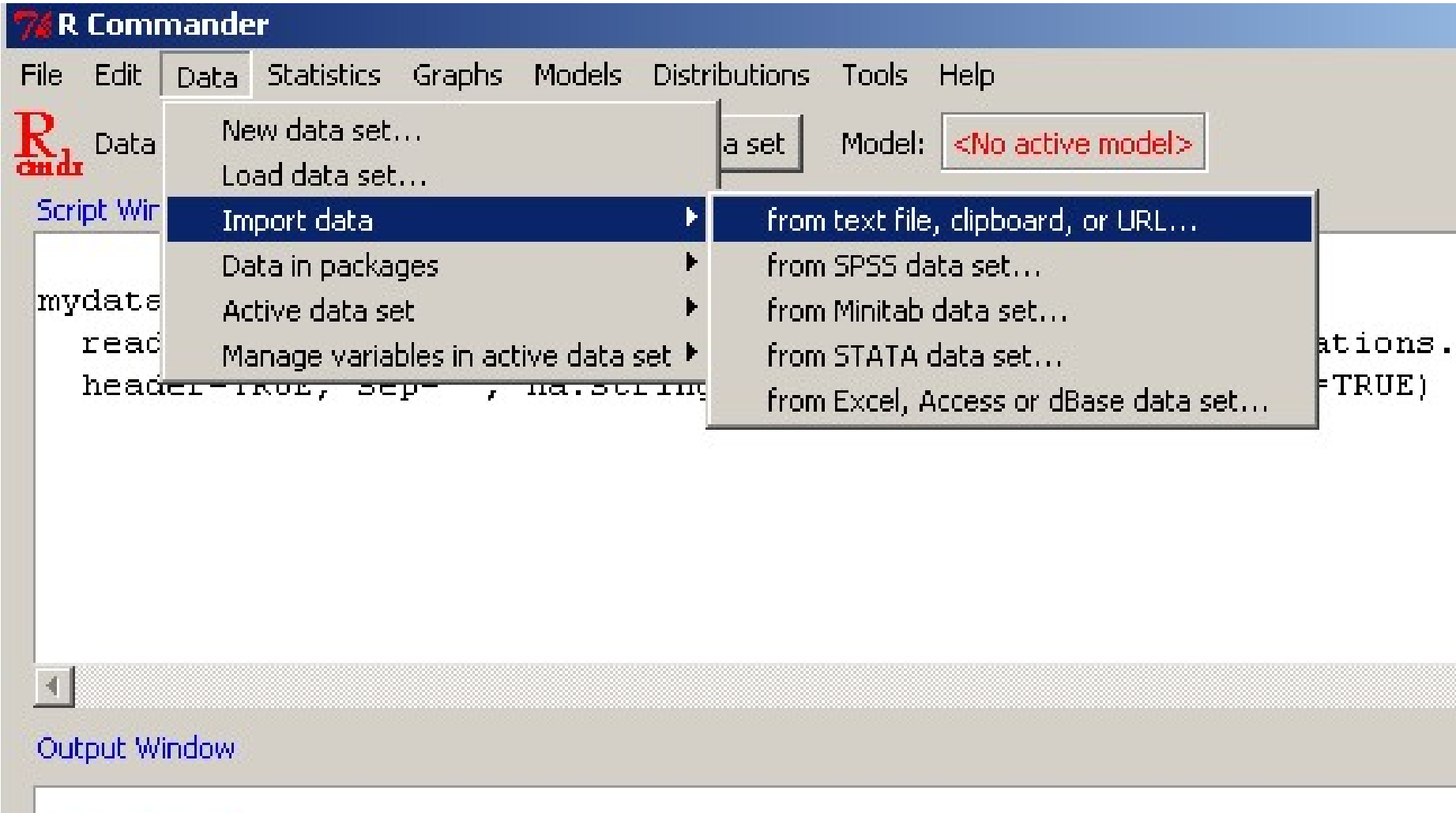
Univariate data analysis

Loading Nations.txt

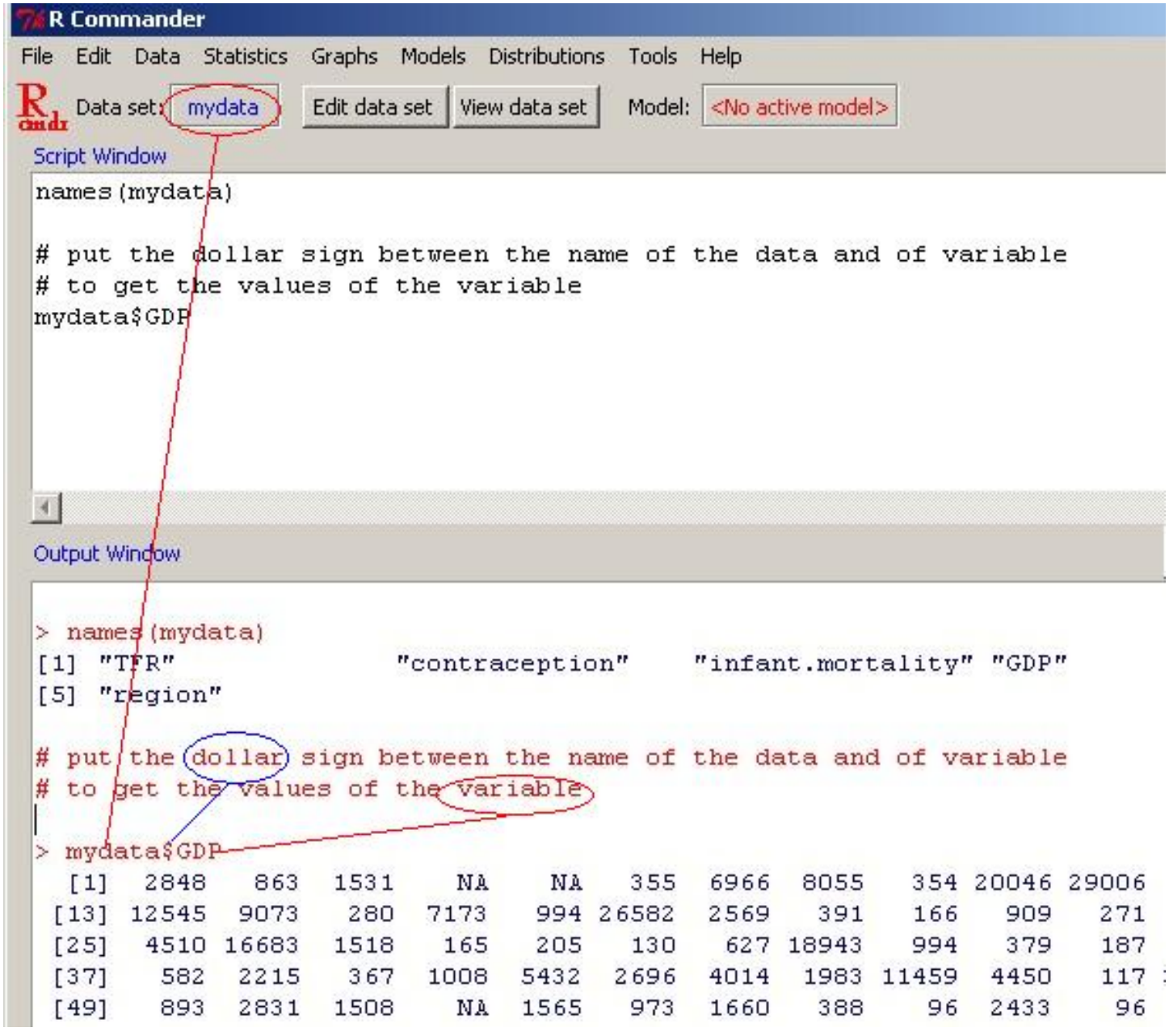
- We want to load Nations.txt located in ...
- C:/Program Files/R/R-2.13.0/library/Rcmdr/etc/ ...
- And call it mydata



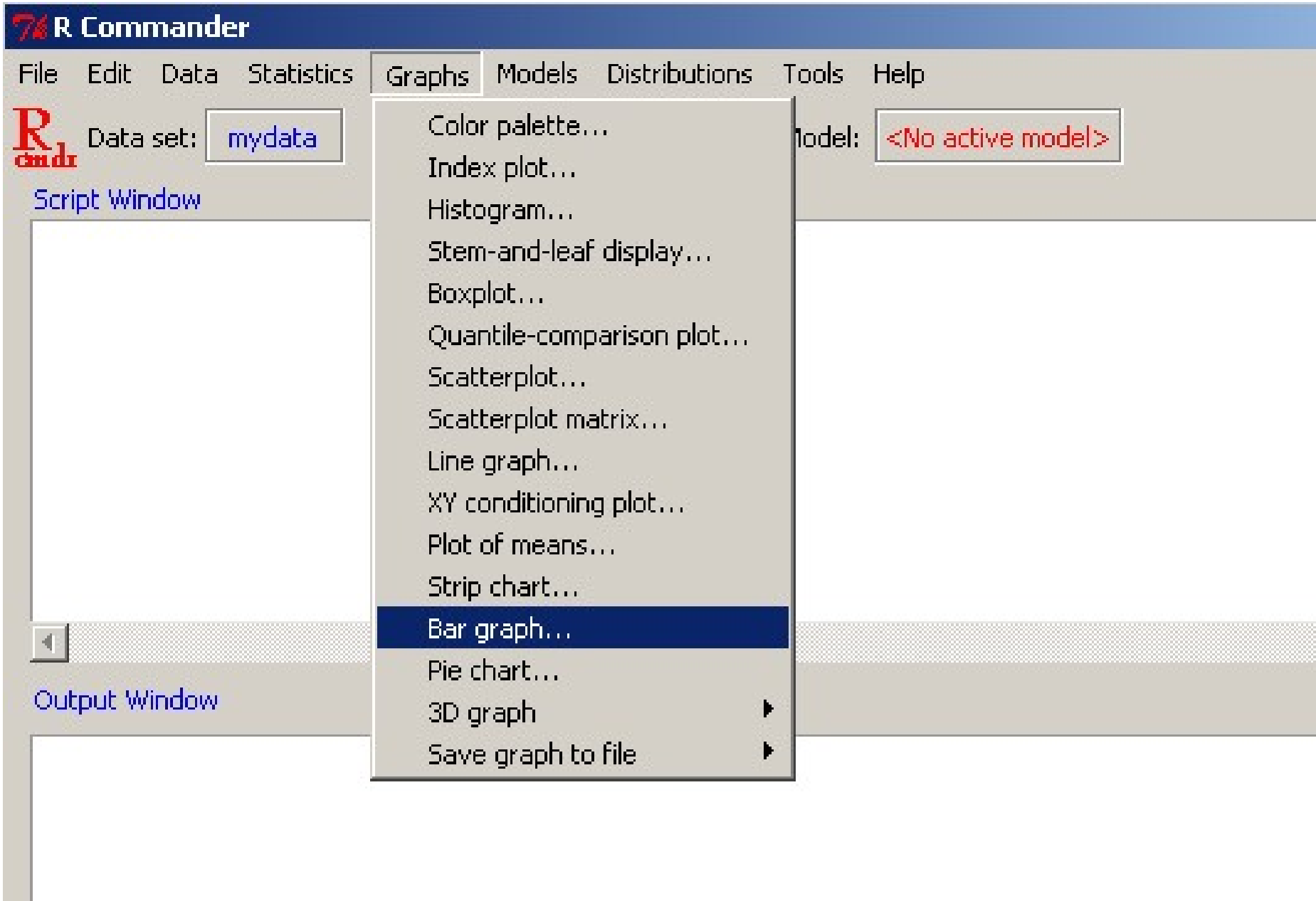
Extracting variables from the data set

- 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.

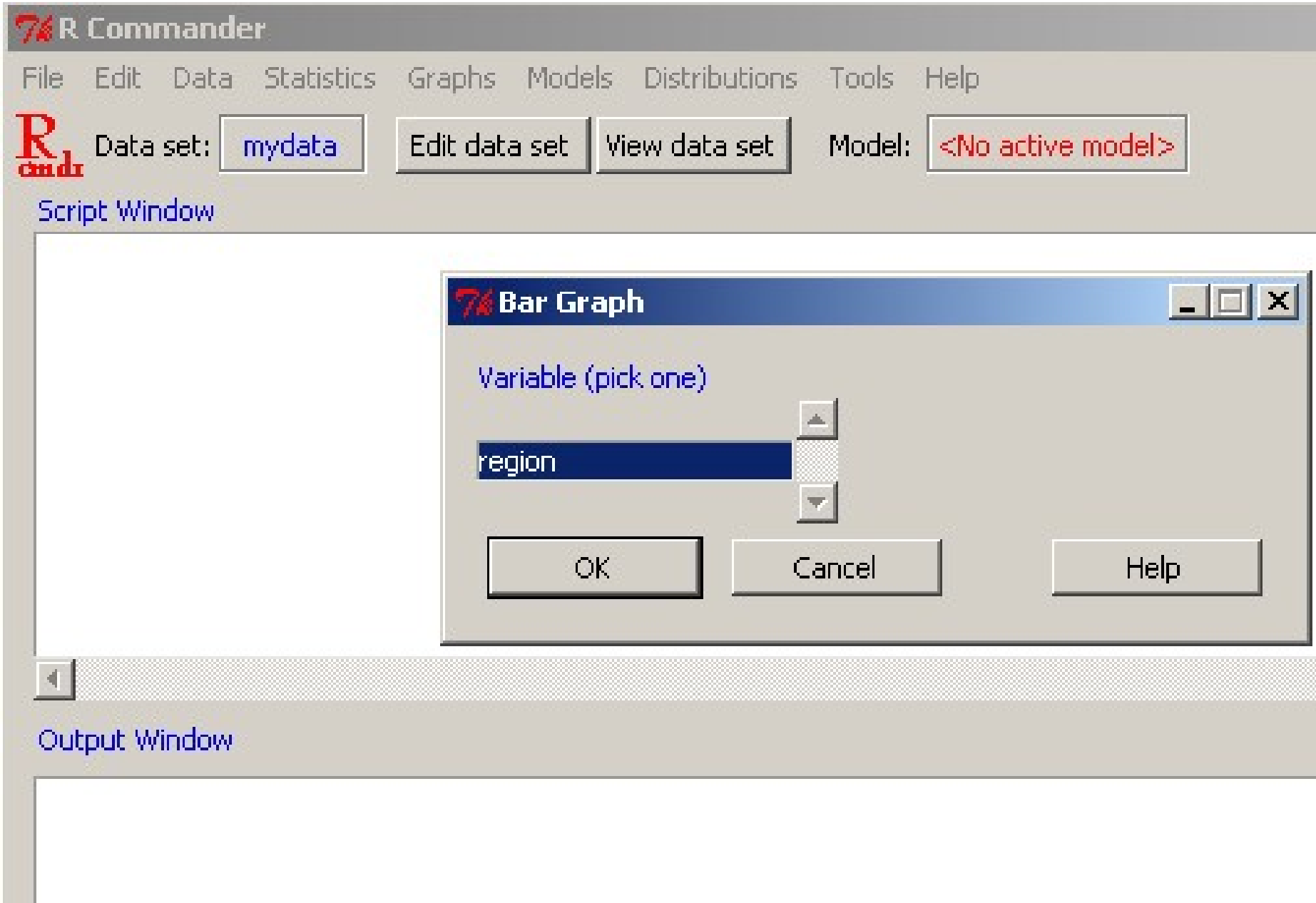
```
names(mydata)

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

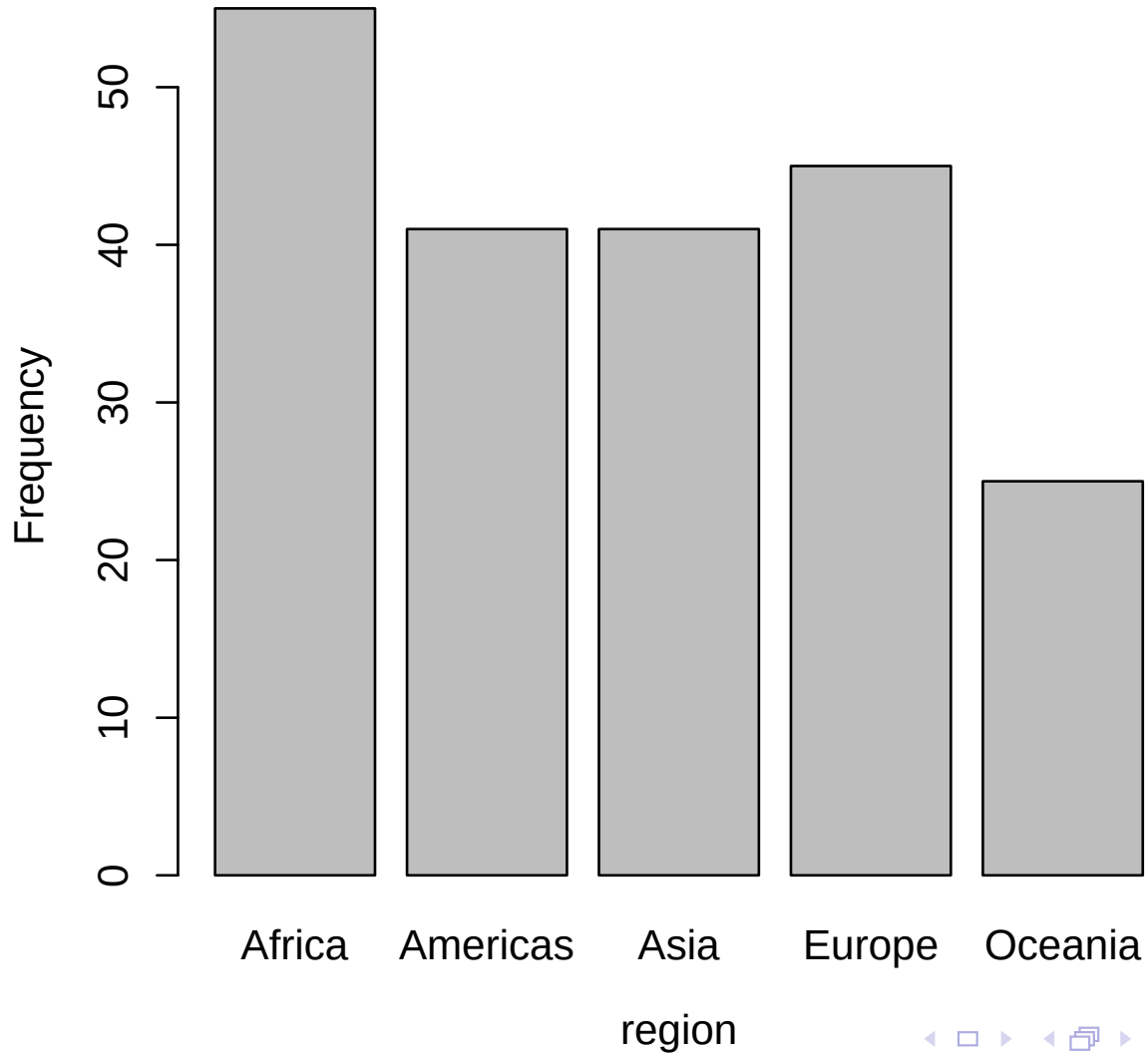


Graphical displays - barchart cont.



Graphical displays - barchart cont.

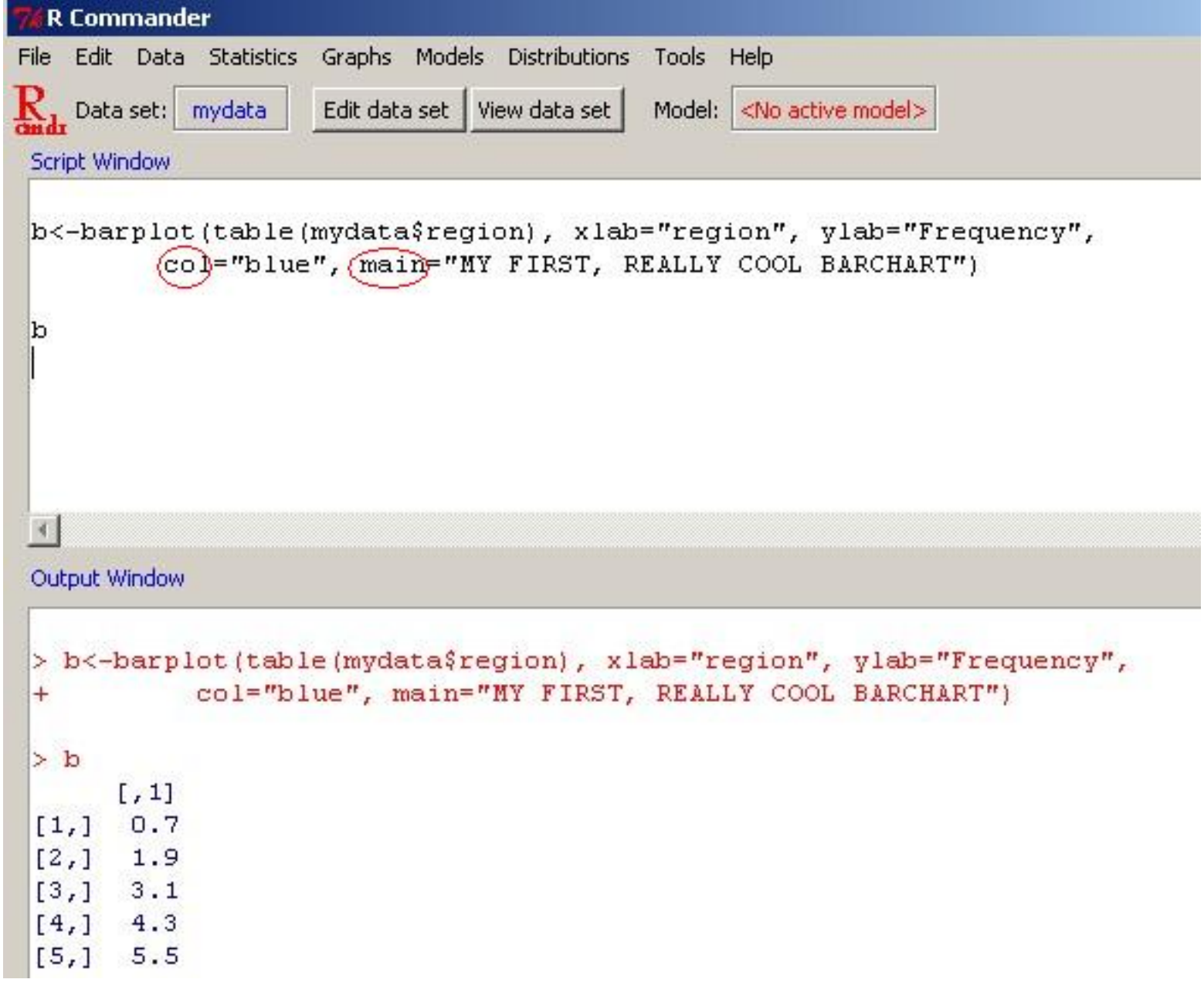
- This one is with the default settings



Graphical displays - barchart cont.

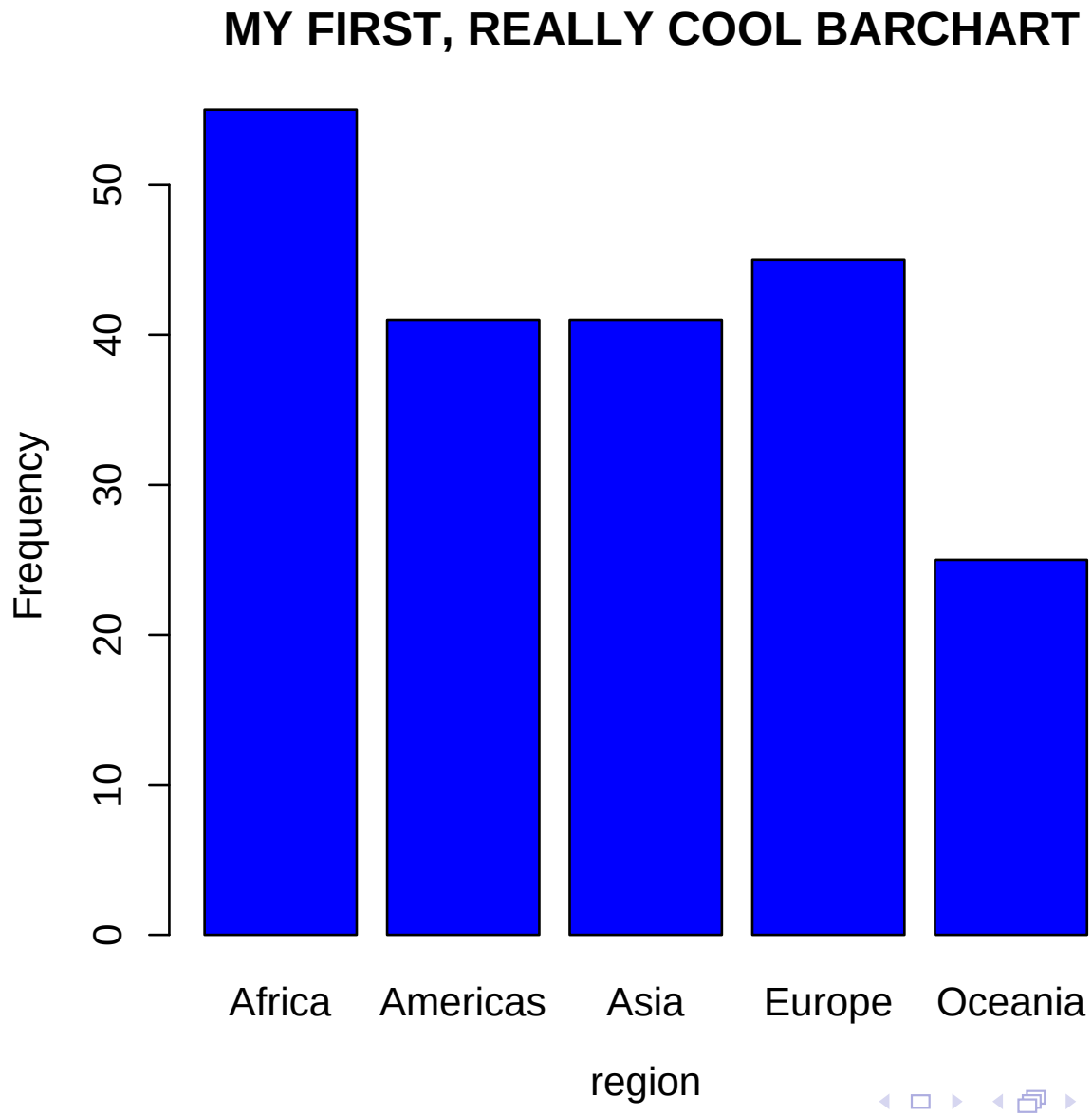
- Use the Script Window to obtain 'pretty' barchart: col to set up the color, and main, for the main title, and store frequencies/stats in a variable b by writing b = barplot(...)

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

# For all options of command barplot, type:

?barplot
```


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
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:

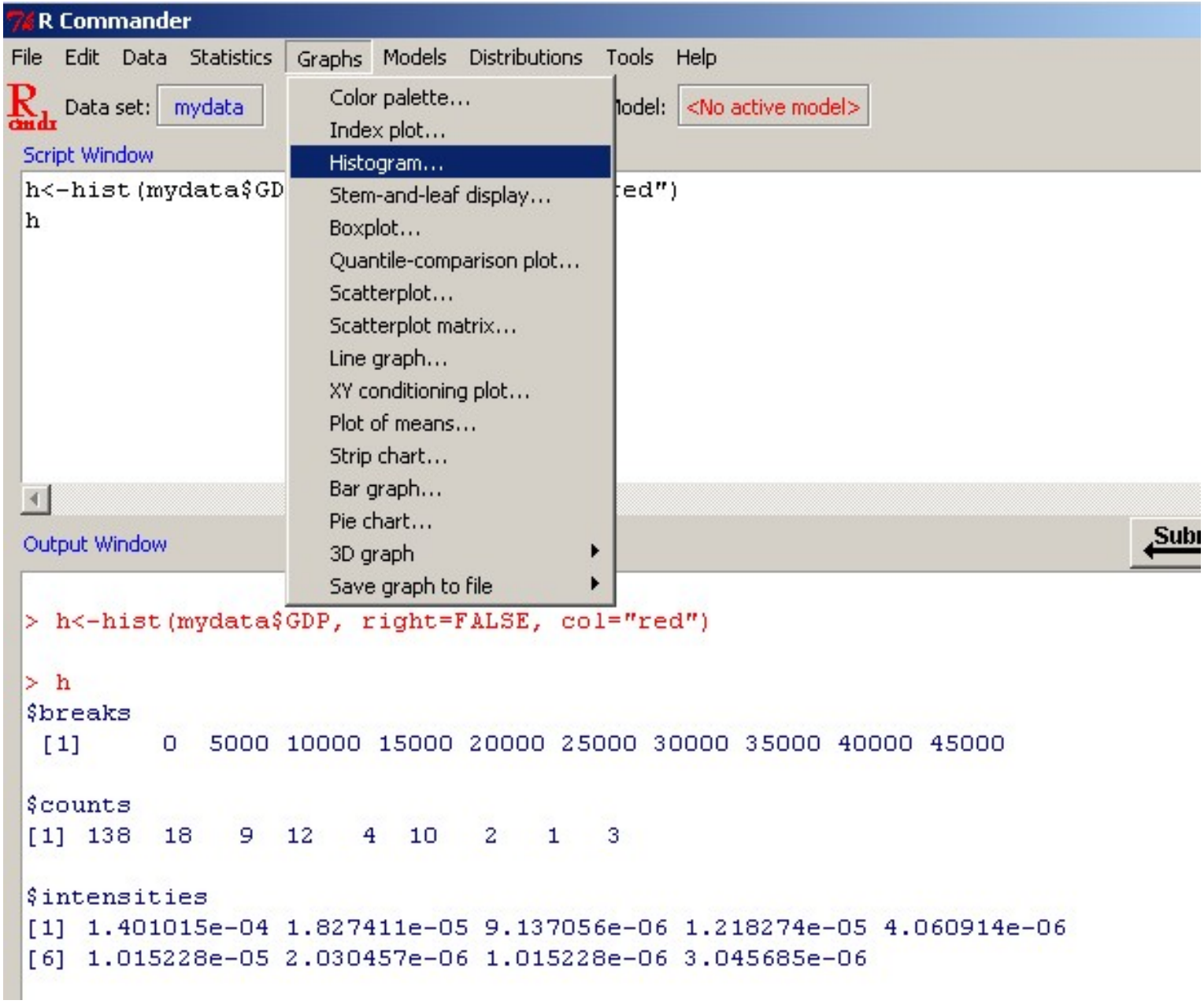
Graph $\Rightarrow$ Histogram

Select the variable of interest

Select the axis scaling

OK

Graphical displays - histogram



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

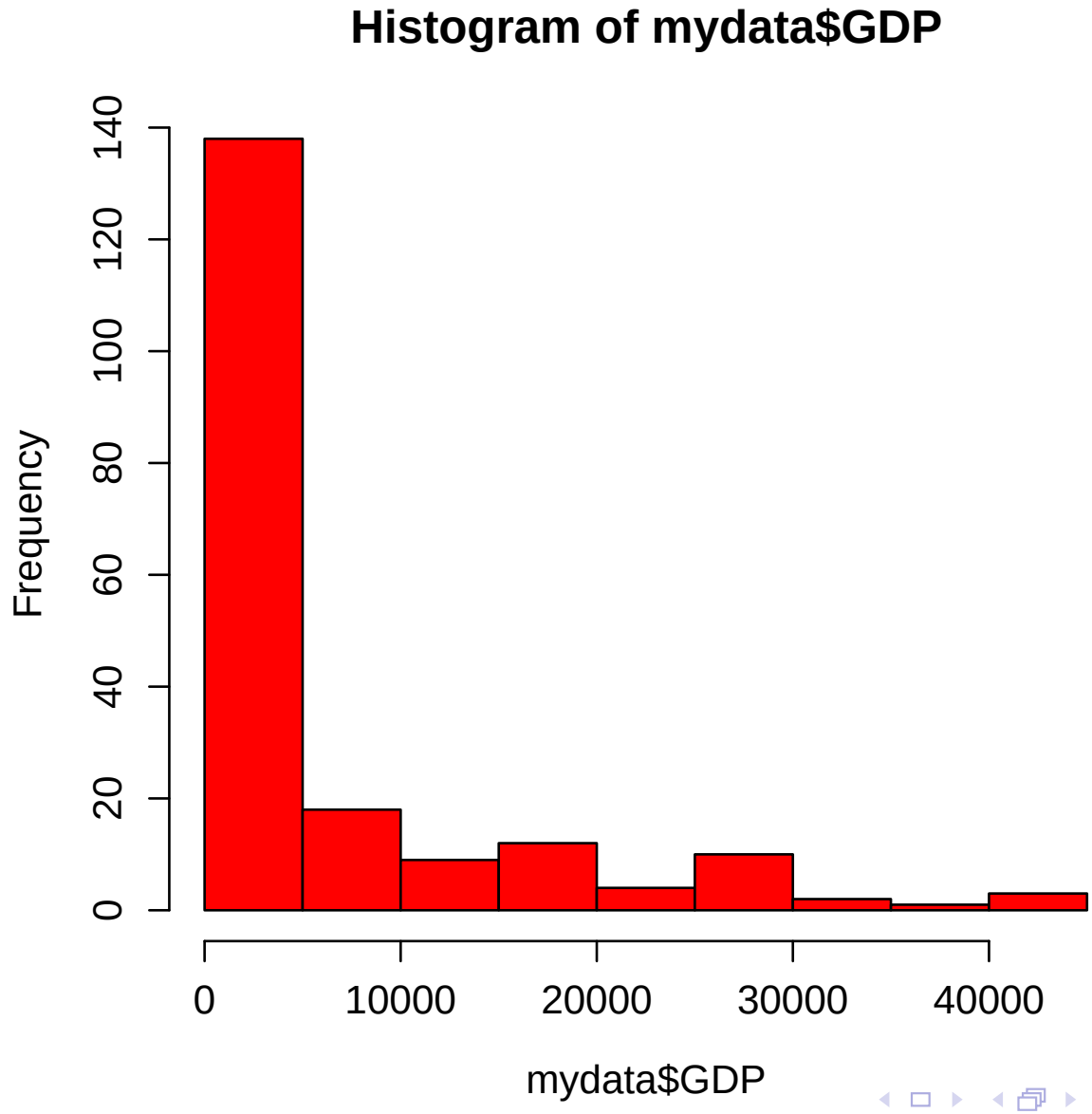
```
hist(mydata$GDP,right=FALSE,col="red")
```

- Other nice options, using for example,

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

Graphical displays - histogram cont.

- This is the result



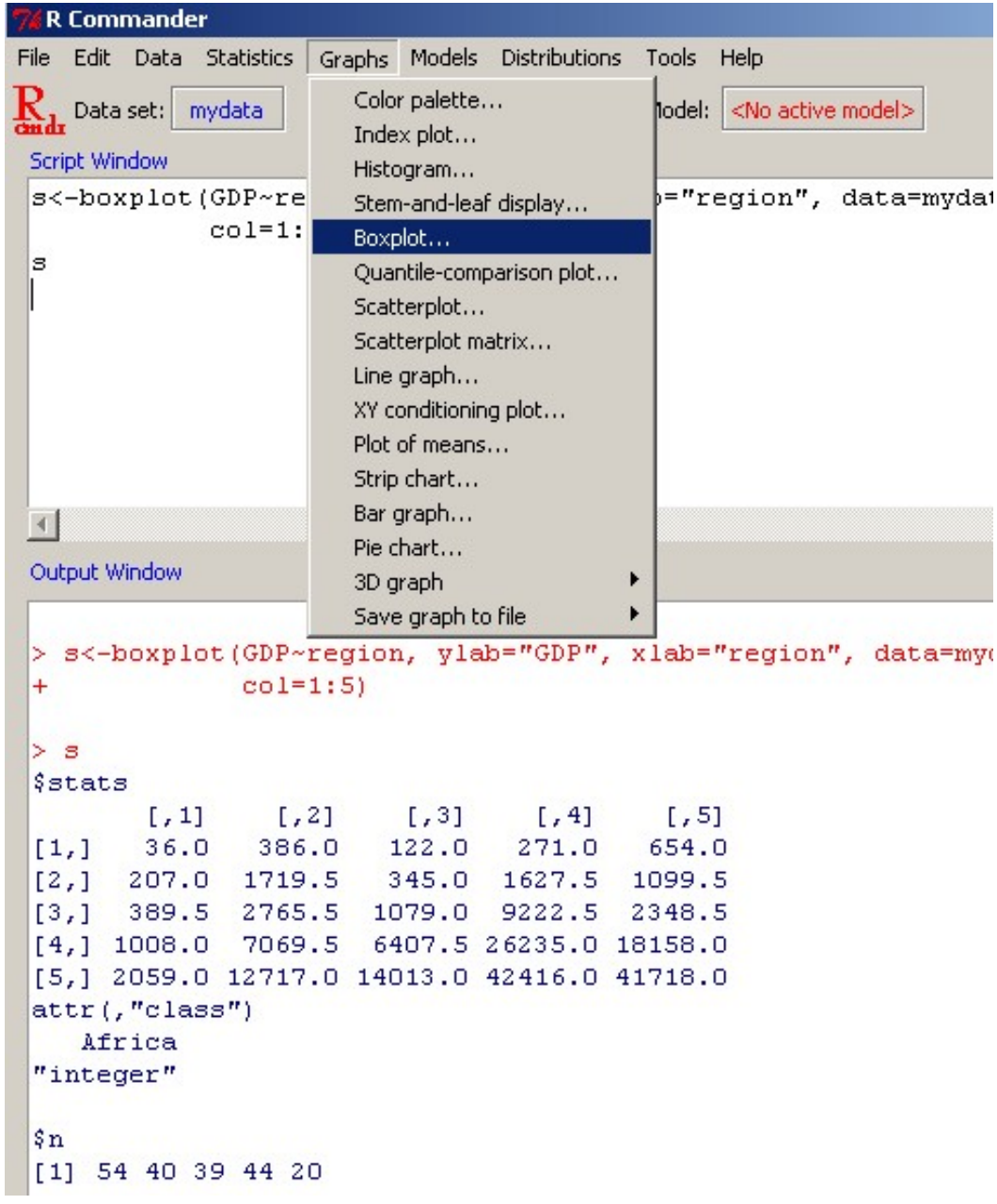
Graphical displays - boxplot

- 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



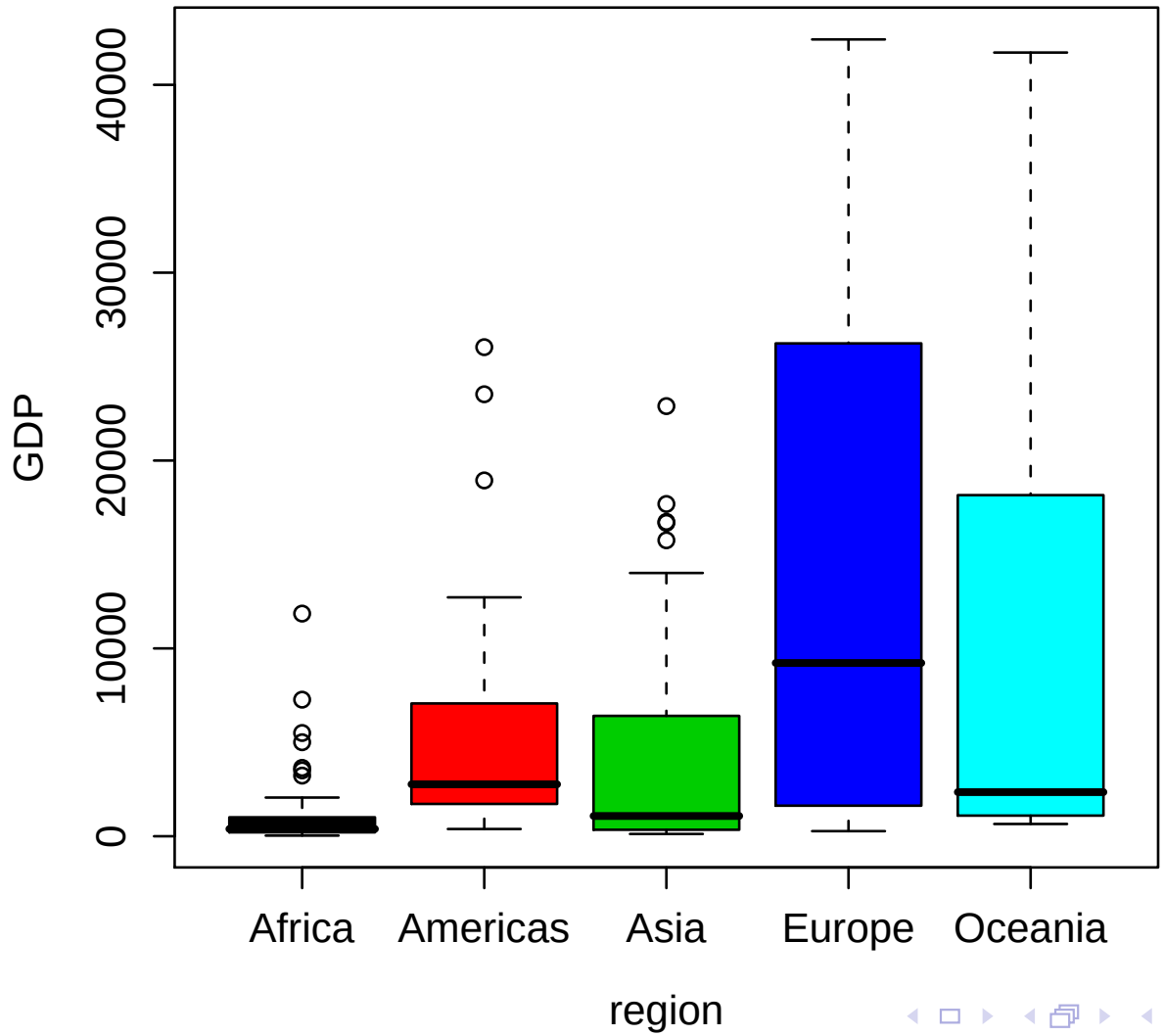
Graphical displays - boxplot

- 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

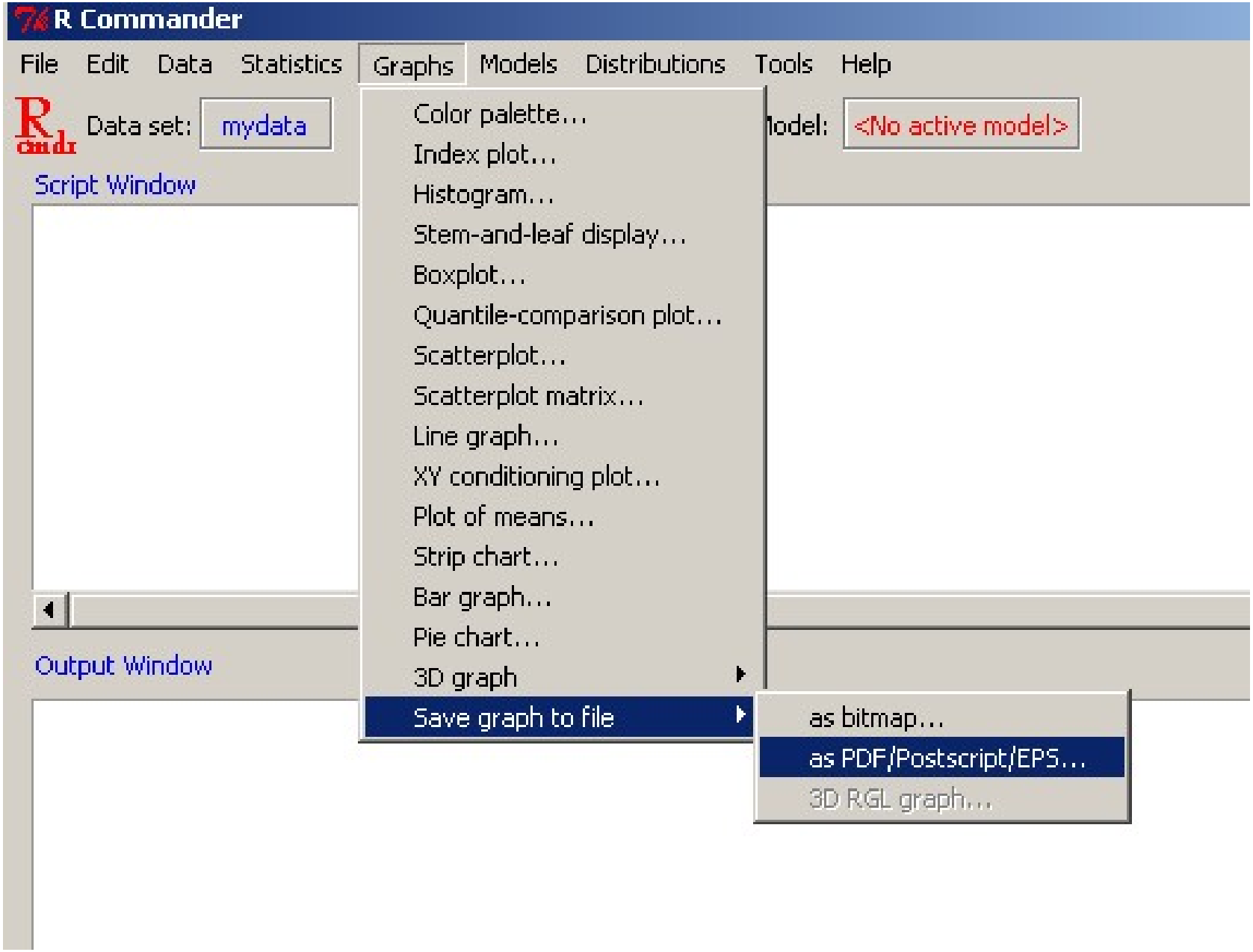
```
boxplot(GDP ~ region, ylab="region",
data=mydata, col=1:5)
```

Graphical displays - boxplot cont.

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

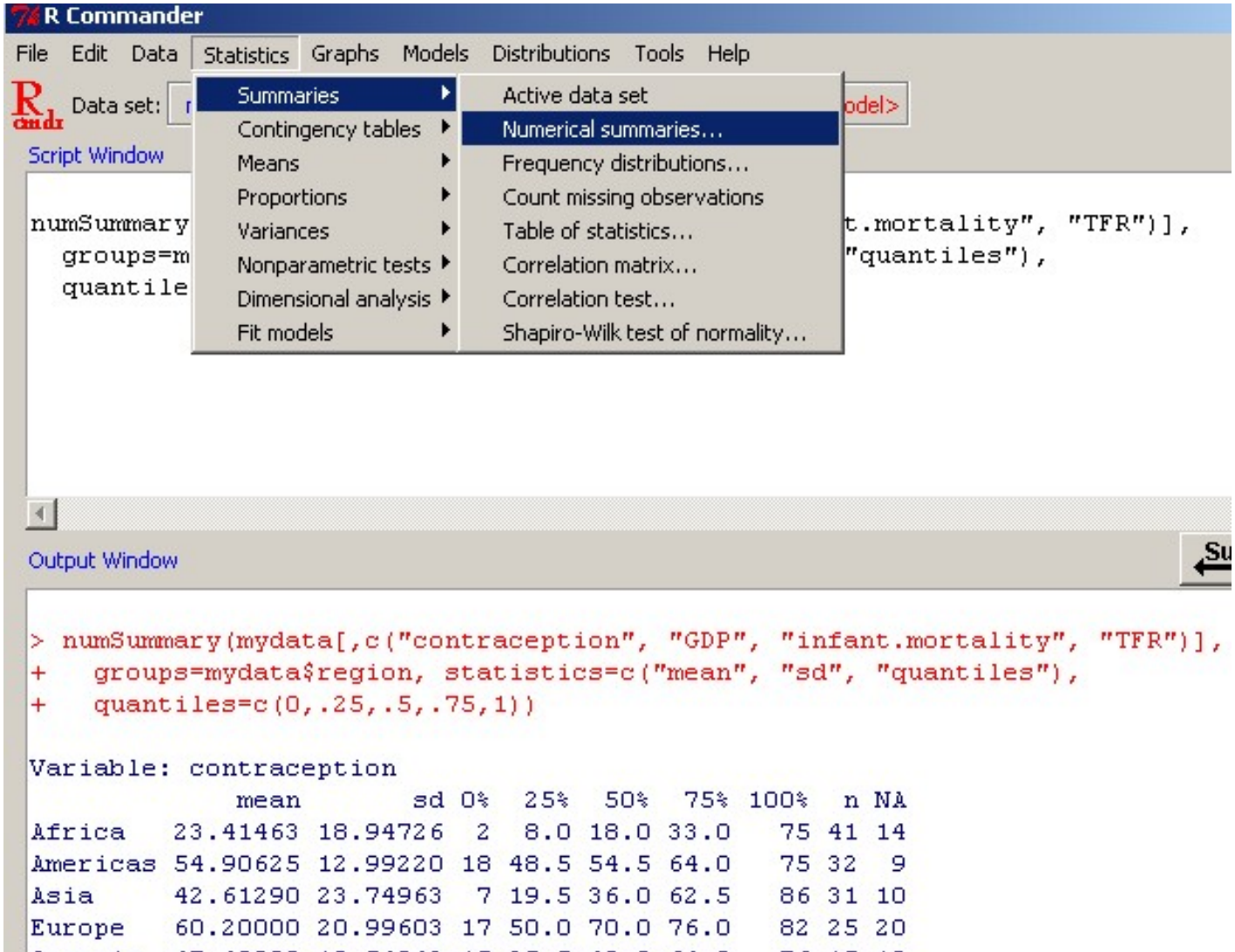


Saving graphs



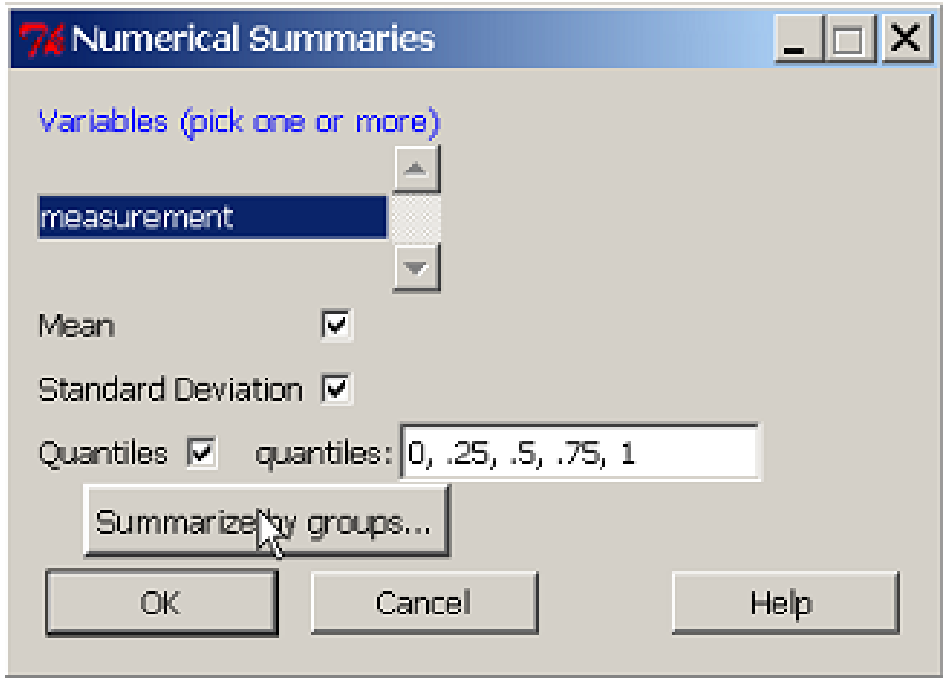
Numerical summaries

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



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**



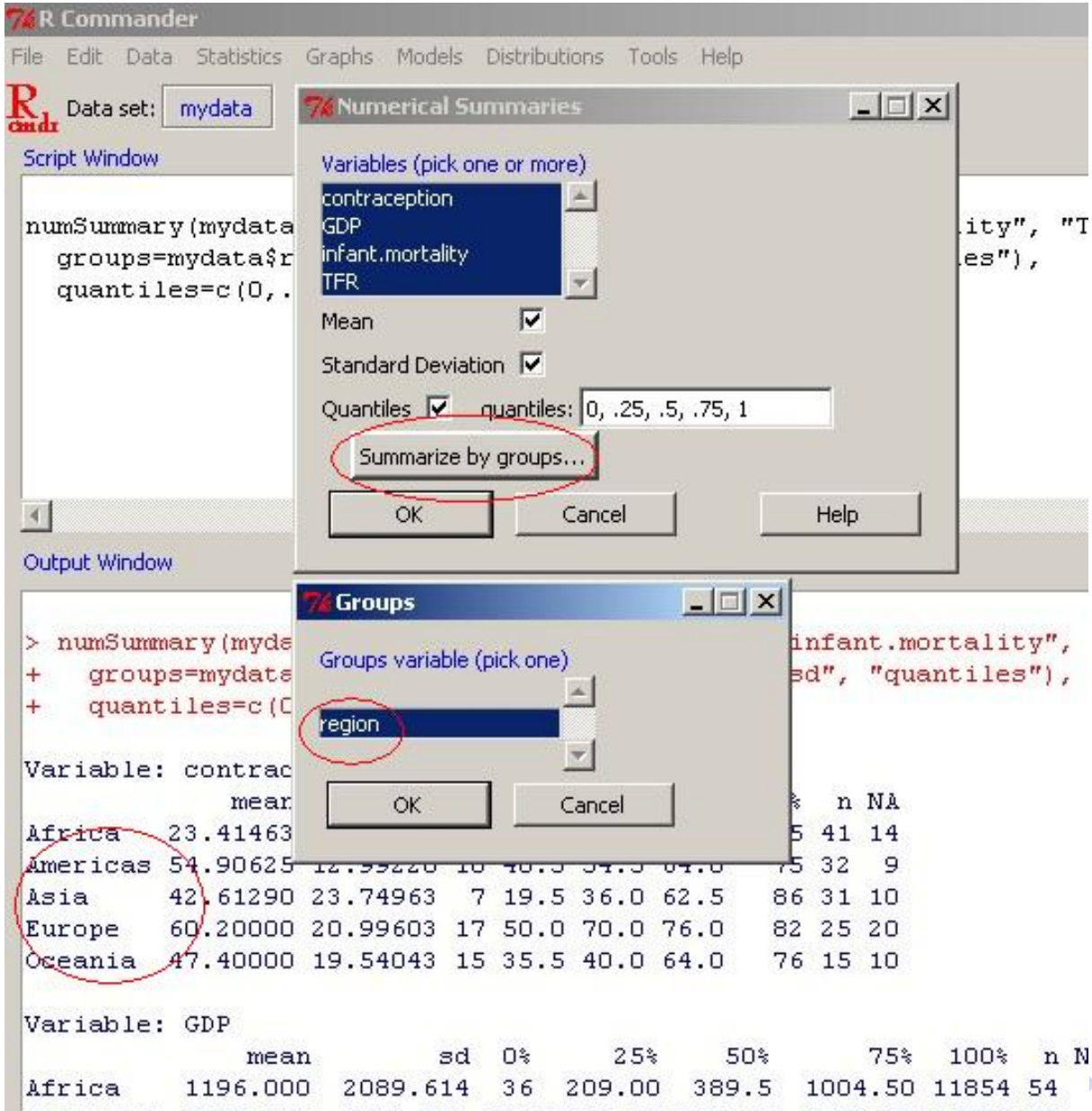
Numerical summaries

Understanding the output:

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)



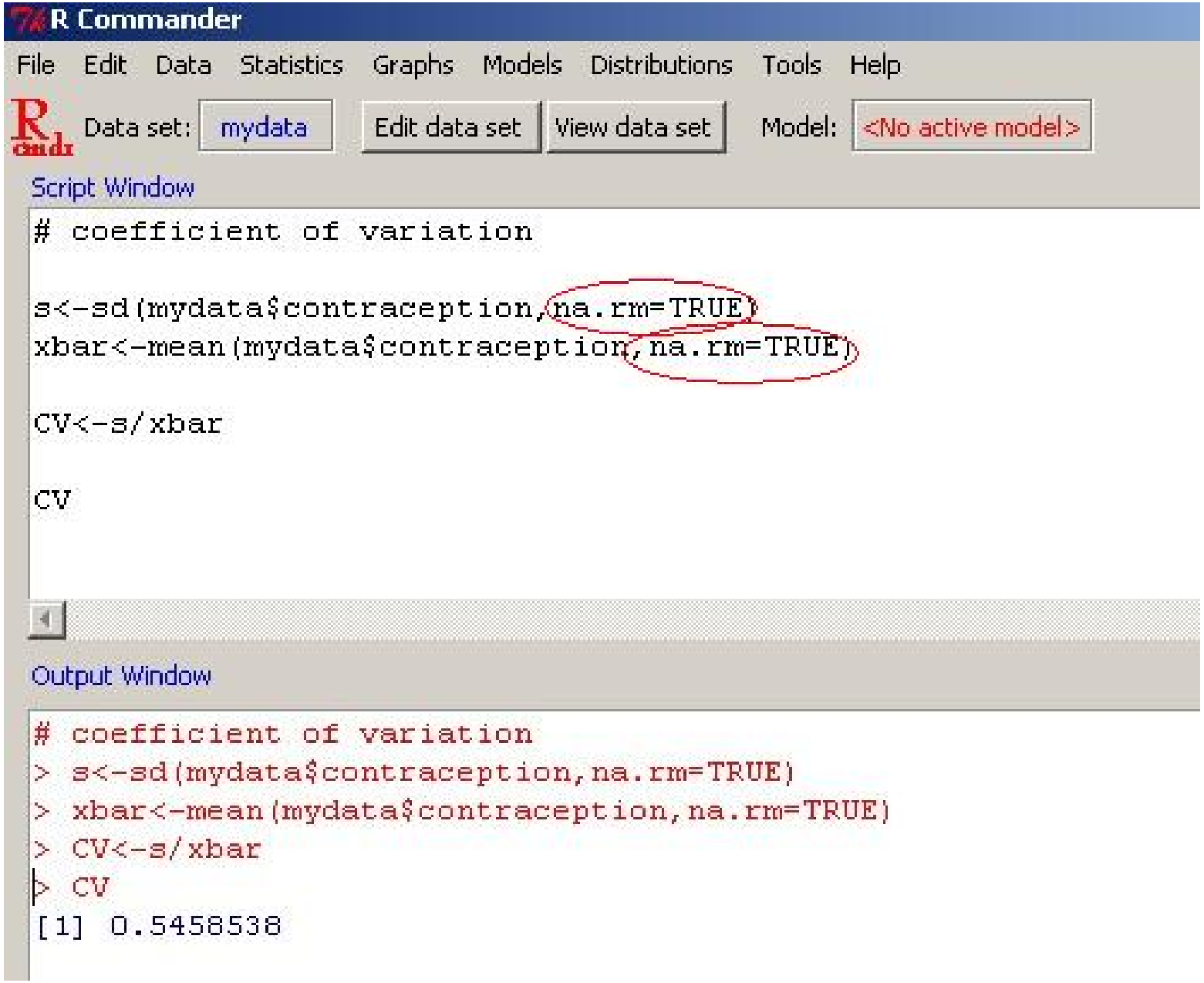
Numerical summaries

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



Numerical summaries

Coefficient of kurtosis and skewness:

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

$$b_1 = \frac{m_3}{s^3}$$

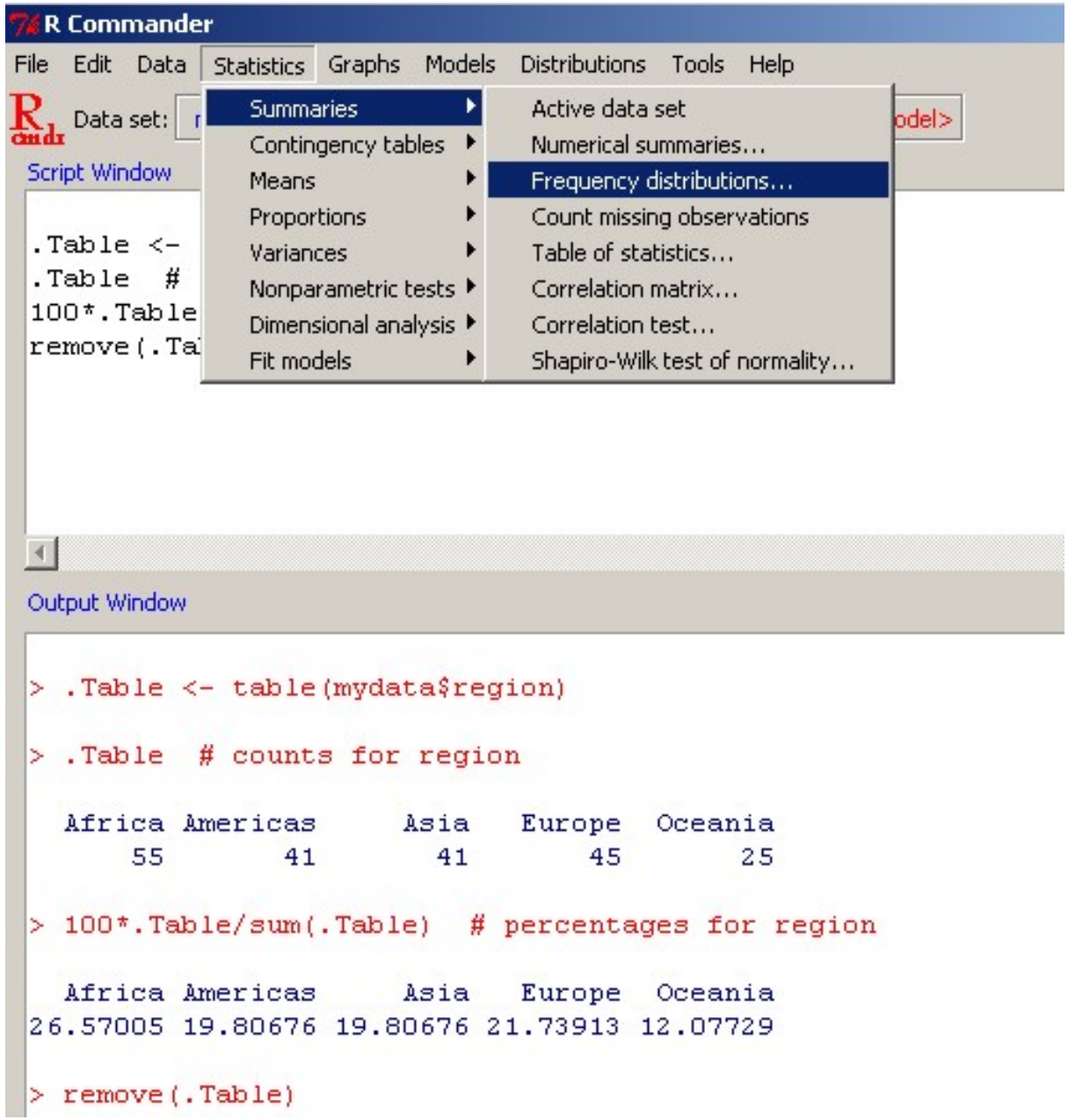
- You have to load the library e1071

```
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



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

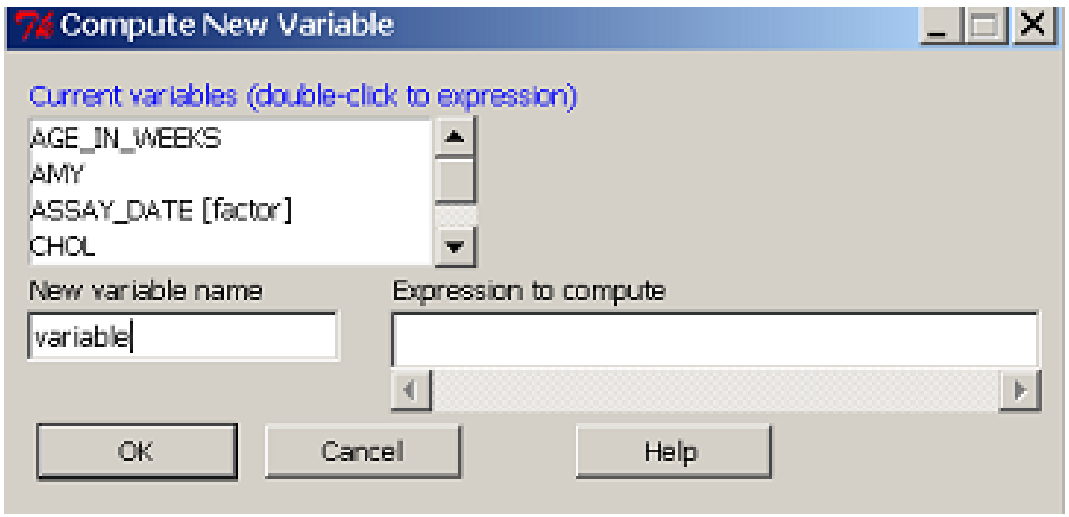
```
library(agricolae)
h<-hist(mydata$contraception, right=FALSE)
table.freq(h)
```

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



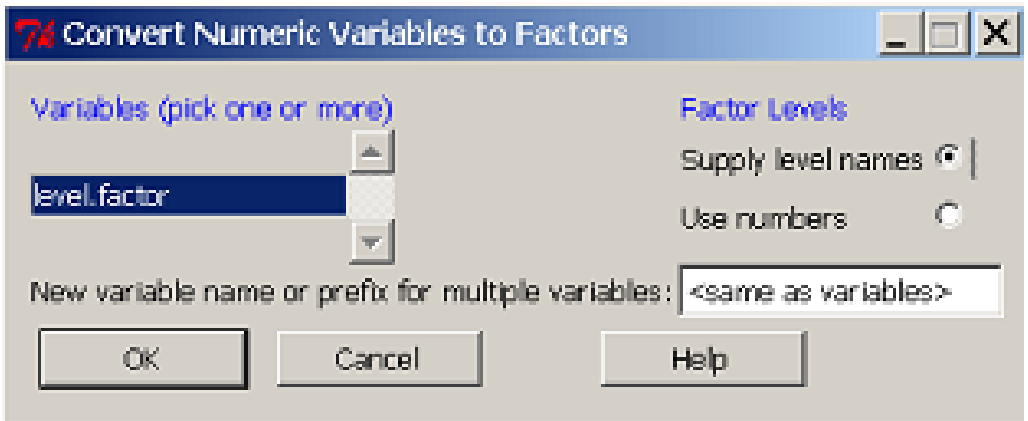
Modifying the dataset: Compute a new variable

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
log10(x)	Log10 transformation	Log10(Variable 1)	
log(x, base)	Log transformation to a specified base	Log(Variable 1, 2)	

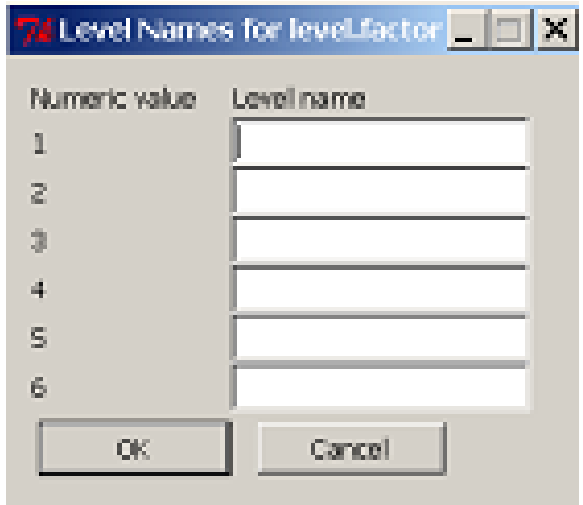
Converting numeric variables to factors

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



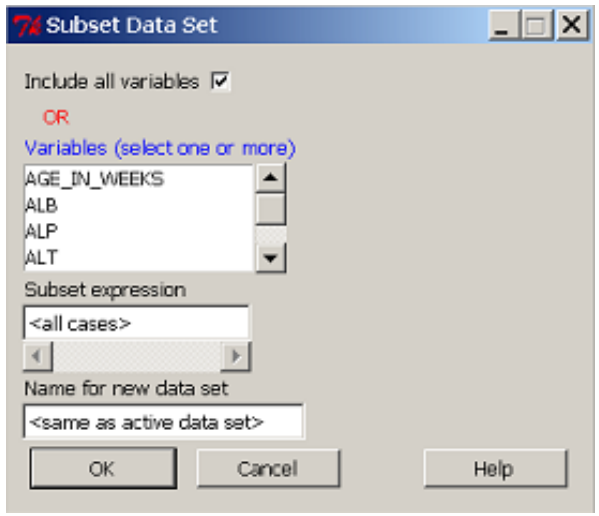
Converting numeric variables to factors

- 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



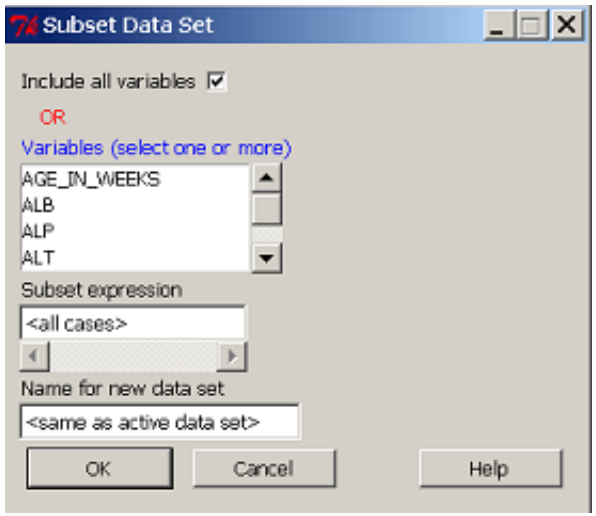
Sub-dividing data by columns (variables)

- **Data ⇒ 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`

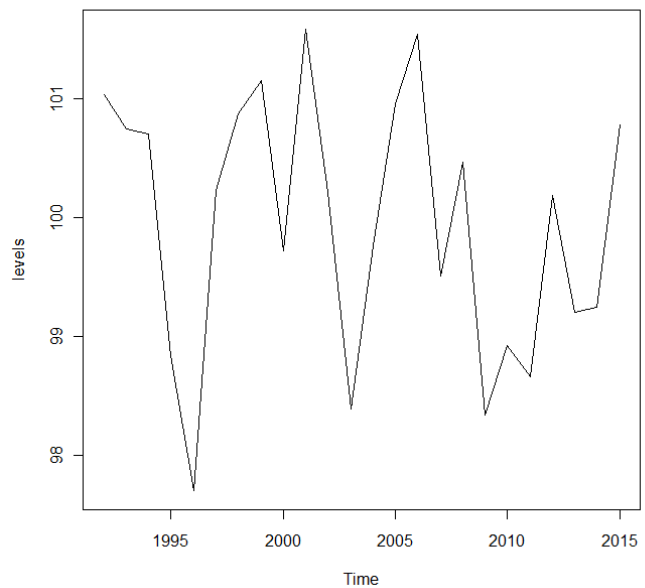
Symbol/code	Name	Use
==	equality	used to indicate the variable should equal
!=	Inequality	used to indicate the variable should not equal
&	And	used to combine multiple expressions
|	Or	used to combine multiple expressions
is.na(varname)		Include the missing values of a variable
!is.na(varname)		Exclude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



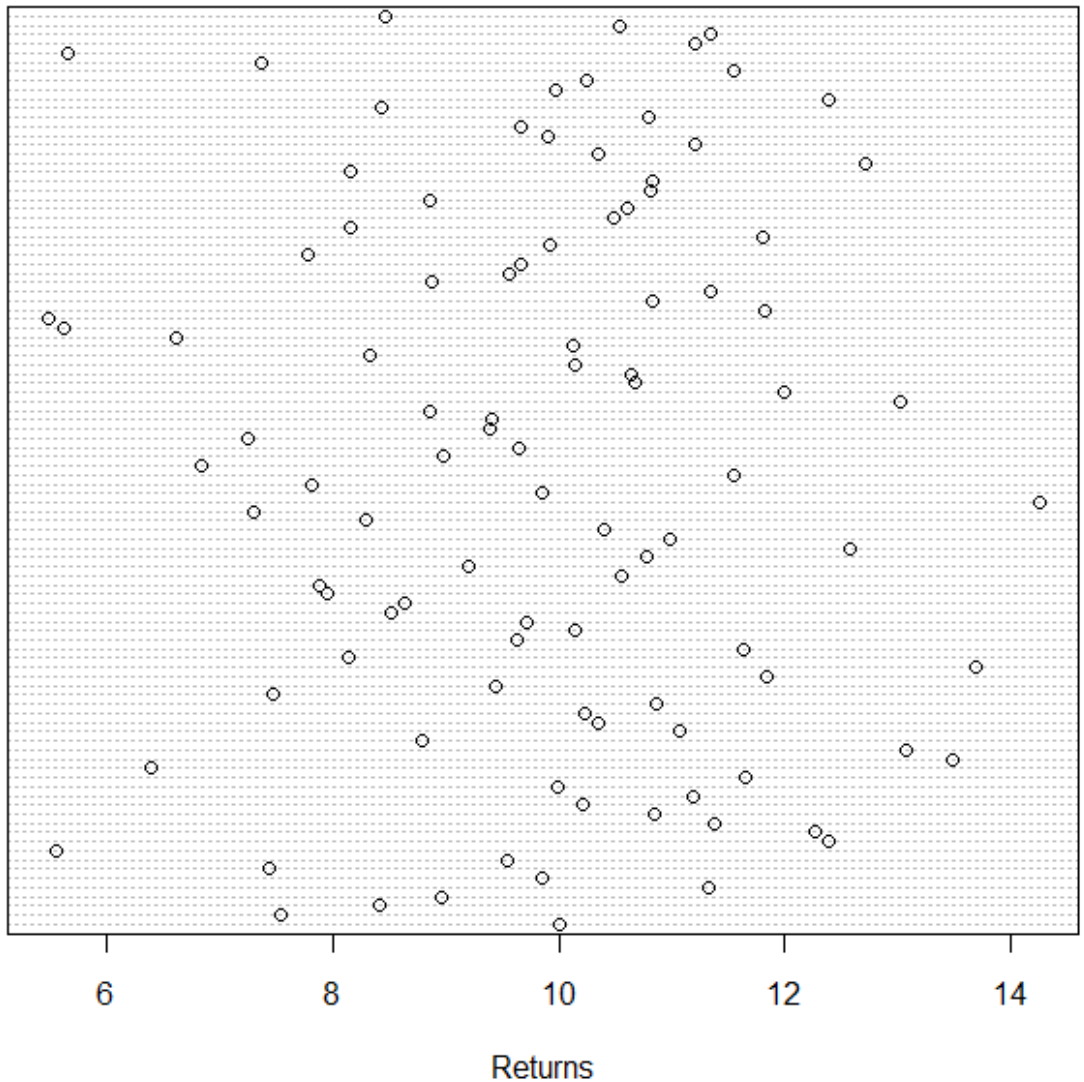
DotPlots I

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