

Bivariate data analysis

Categorical data - creating data set

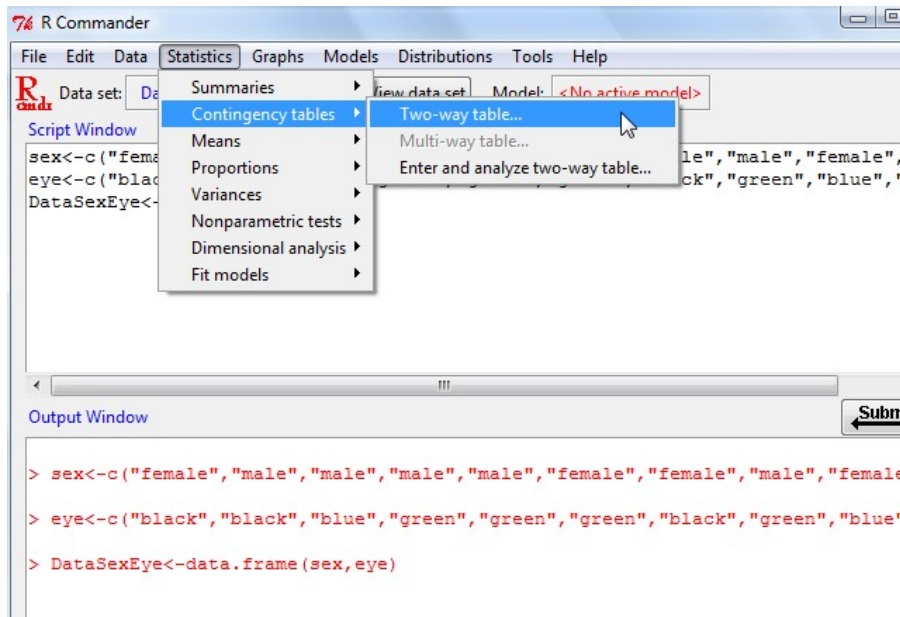
Upload the following data set to R Commander

sex	female	male	male	male	male	female	female	male	female	female
eye	black	black	blue	green	green	green	black	green	blue	blue

- ▶ Method 1: Type the table in the Notepad, save it and import to Rcmdr
- ▶ Method 2: Introduce directly in the [Script Window](#)

```
eye = c("black","black","blue","green","green",  
        "green","black","green","blue","blue")  
  
sex = c("female","male","male","male","male",  
        "female", "female","male","female","female")  
  
DataSexEye = data.frame(sex,eye)
```

Categorical data - contingency table



The screenshot shows the R Commander application window. The 'Statistics' menu is open, and 'Contingency tables' is selected. A sub-menu is visible with the following options:

- Two-way table...
- Multi-way table...
- Enter and analyze two-way table...

The 'Script Window' contains the following R code:

```
sex<-c("female","male","male","male","male","female","female","male","female")
eye<-c("black","black","blue","green","green","green","black","green","blue")
DataSexEye<-data.frame(sex,eye)
```

The 'Output Window' shows the execution of the code:

```
> sex<-c("female","male","male","male","male","female","female","male","female")
> eye<-c("black","black","blue","green","green","green","black","green","blue")
> DataSexEye<-data.frame(sex,eye)
```

Categorical data - contingency table cont.

- ▶ How many of the sampled people are female with black eyes? (2)
- ▶ What % of the sampled people are male with blue eyes? (10%)
- ▶ What % of the sampled people are male? (50%)
- ▶ What % of the sampled people have green eyes? (40%)

The screenshot shows the R Commander interface. The 'Data set' is 'DataSexEye'. The 'Script Window' contains the following code:

```
.Table <- xtabs(~sex+eye, data = DataSexEye)
.Table
totPercents(.Table) # Percentages
.Test <- chisq.test(.Table, conf.level = 0.05)
remove(.Test)
remove(.Table)
```

The 'Output Window' displays the resulting contingency table:

	black	blue	green	Total
female	2	2	1	5
male	1	1	3	5
Total	3	3	4	10

The 'Two-Way Table' dialog box is open, showing 'eye' as the Row variable and 'sex' as the Column variable. Under 'Compute Percentages', 'Percentages of total' is selected (circled in red). Under 'Hypothesis Tests', 'Chi-square test of independence' is checked. The 'Subset expression' is '<all valid cases>'. The 'Submit' button is visible on the right.

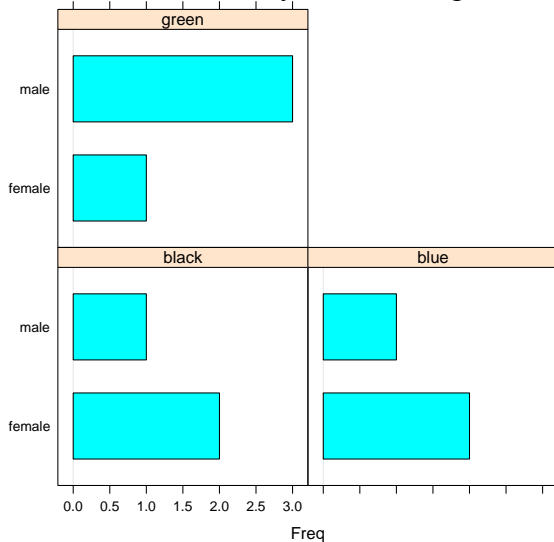
Categorical data - barchart

- ▶ Load the library lattice, then create barchart grouping the data by sex

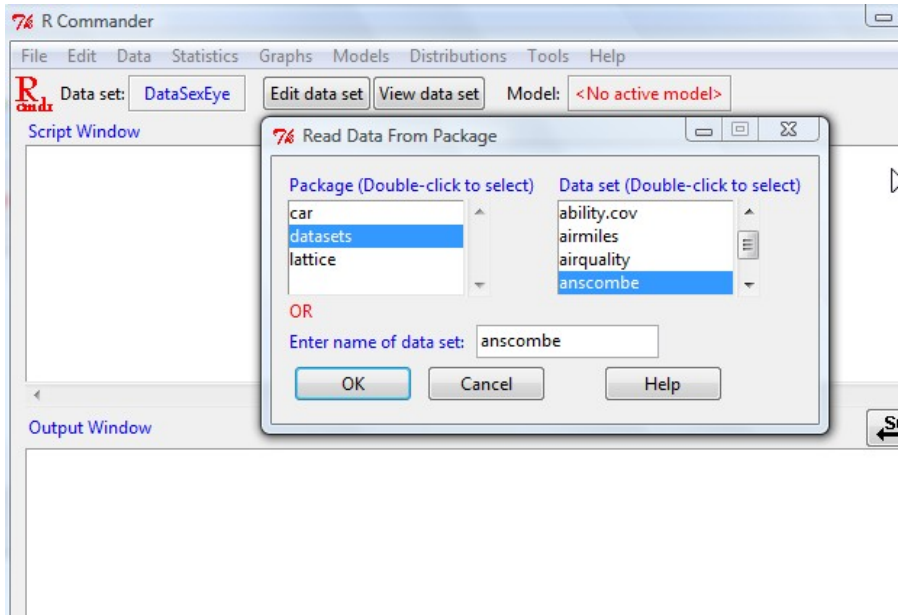
```
library(lattice)  
  
barchart(DataSexEye, groups=DataSexEye$sex)
```

Categorical data - barchart cont.

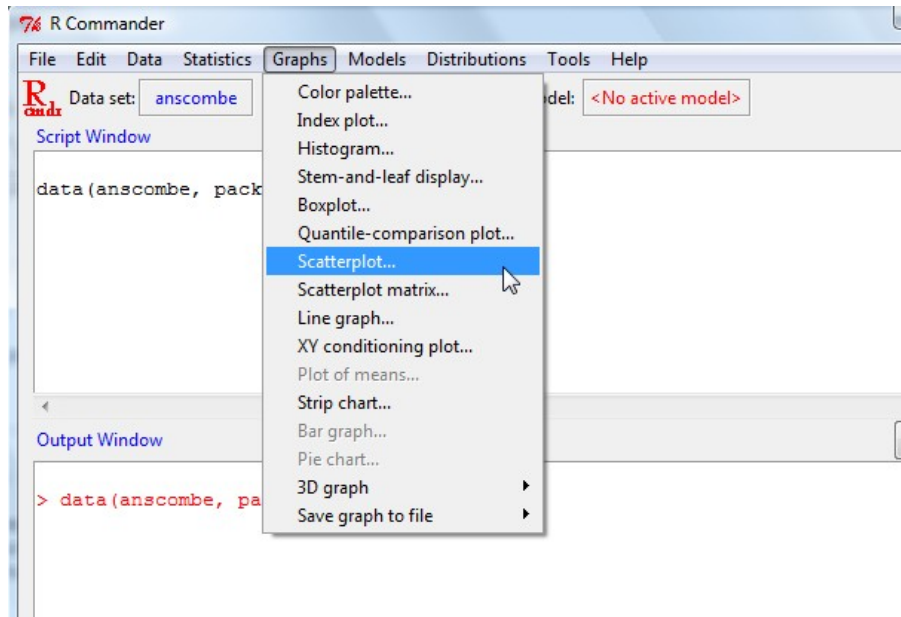
- Are there more females or males with blue eyes? (females)
- What is the most common eye color among males? (green)



Numerical data - load anscombe data set from R library

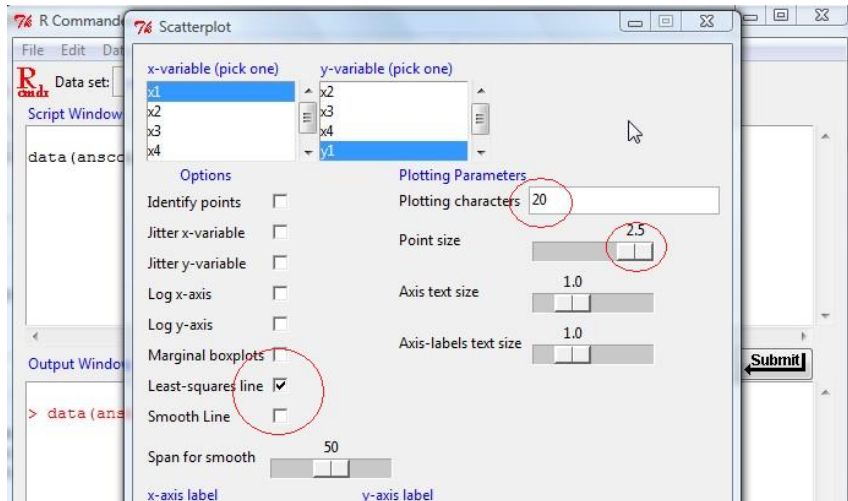


Numerical data - scatterplot of y_1 versus x_1

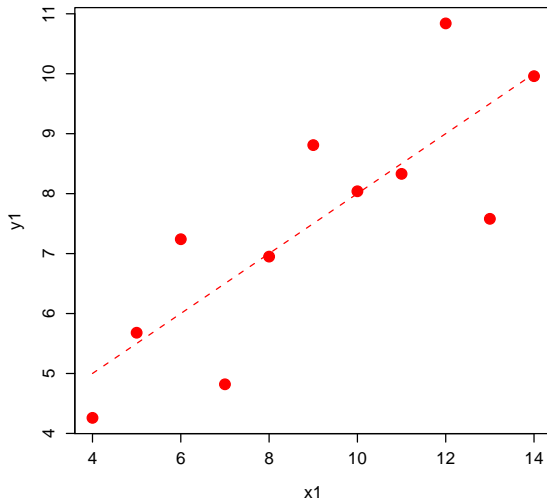


Numerical data - scatterplot of y1 versus x1 cont.

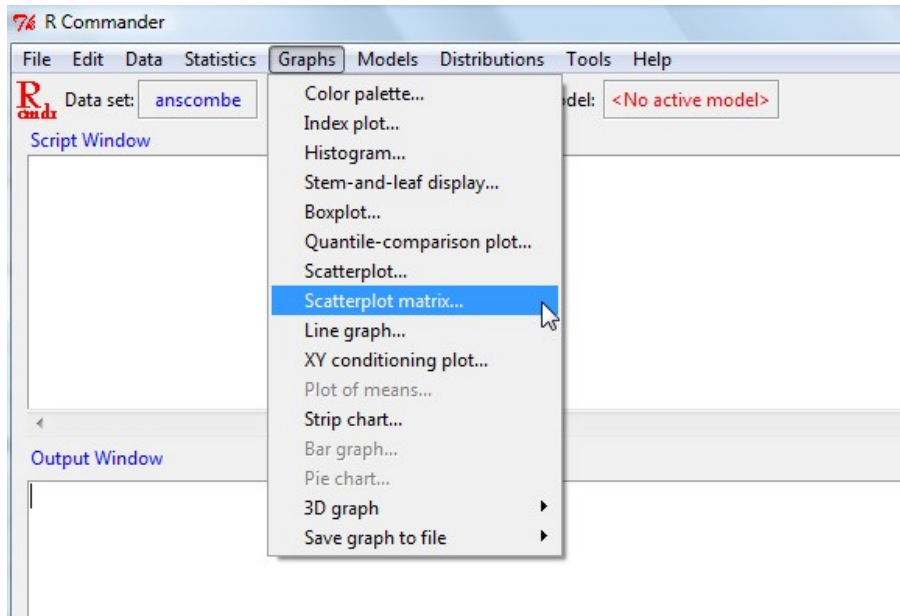
- ▶ Uncheck all but the **Least-squares line**
- ▶ **Plotting characters** 20 corresponds to bullets
- ▶ Increase the **Point size** to 2.5



Numerical data - scatterplot of y_1 versus x_1 cont.

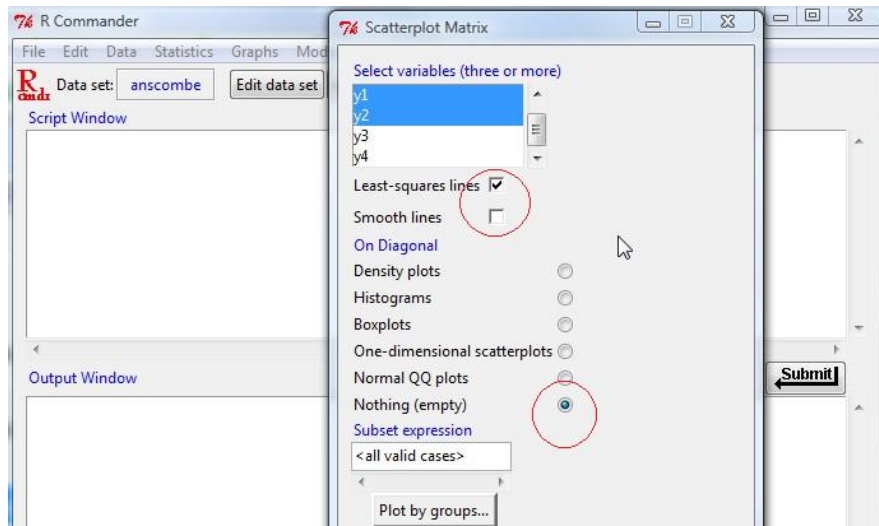


Numerical data - scatterplot matrix (only x_1, x_2, y_1, y_2)

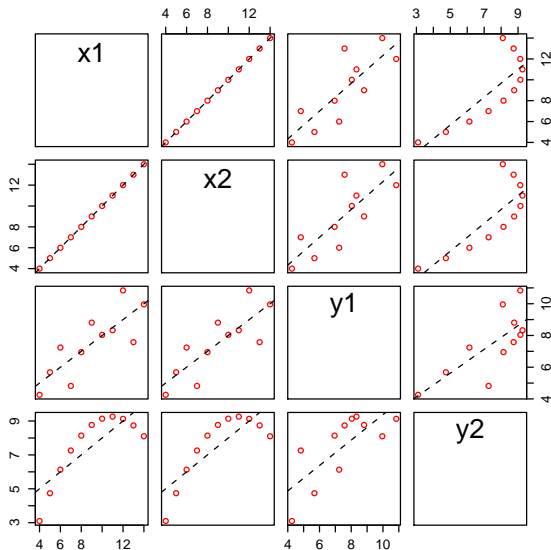


Numerical data - scatterplot matrix (only x_1, x_2, y_1, y_2) cont.

- Check Least-squares line



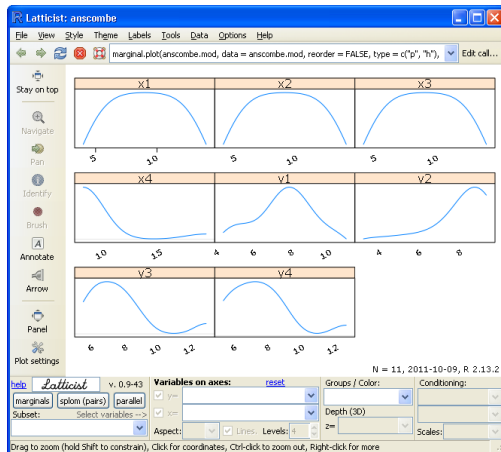
Numerical data - scatterplot matrix (only x_1, x_2, y_1, y_2) cont.



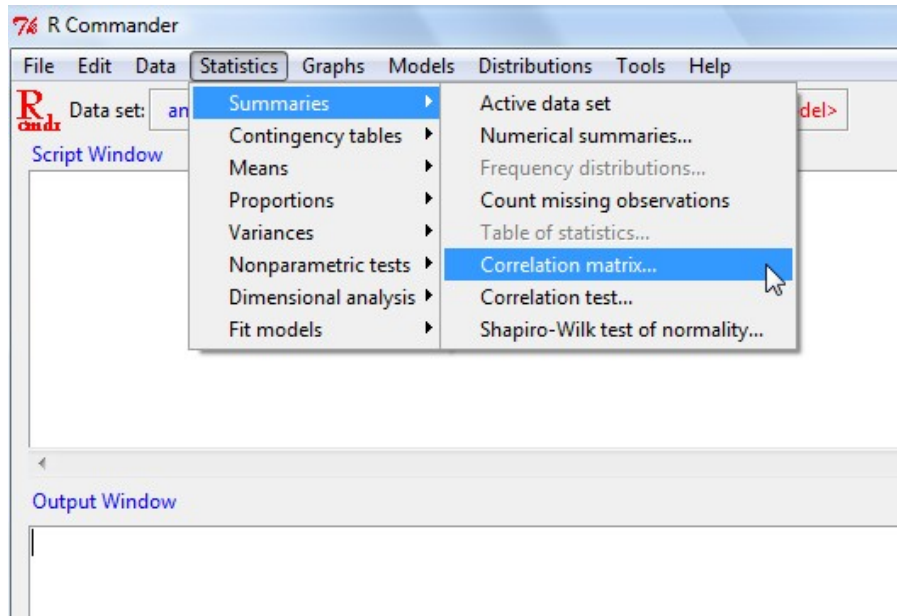
lattice environment

You can create interactive graphics:

```
data(anscombe, package="datasets")  
library(latticeist)  
latticeist(anscombe)
```



Numerical data - correlation matrix



Numerical data - correlation matrix (only x_1, x_2, y_1, y_2) cont.

- ▶ Matrix is symmetrical with values on the diagonal = 1
- ▶ $cor(x_1, y_1) = cor(y_1, x_1) = 0.8164205$

The screenshot shows the R Commander interface. The 'Data set' is 'anscombe'. The 'Script Window' contains the command `cor(anscombe[,c("x1", "x2", "y1", "y2")])`. The 'Output Window' displays the resulting correlation matrix. A dialog box titled 'Correlation Matrix' is open, showing the 'Variables (pick two or more)' list with 'x1', 'x2', 'y1', and 'y2'. The 'Type of Correlations' section has 'Pearson product-moment' selected. The 'OK' button is highlighted.

Correlation Matrix

Variables (pick two or more)

x1
x2
y1
y2

Type of Correlations

Pearson product-moment ☒
Spearman rank-order ☐
Partial ☐

OK Cancel Help

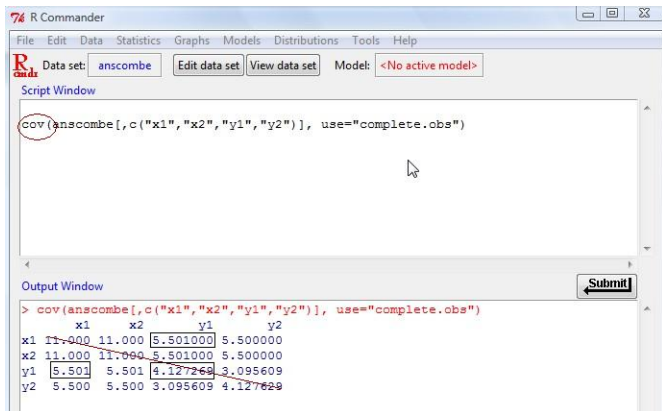
Output Window

```
> cor(anscombe[,c("x1", "x2", "y1", "y2")], use="complete.obs")
```

	x1	x2	y1	y2
x1	1.0000000	1.0000000	0.8164205	0.8162365
x2	1.0000000	1.0000000	0.8164205	0.8162365
y1	0.8164205	0.8164205	1.0000000	0.7500054
y2	0.8162365	0.8162365	0.7500054	1.0000000

Numerical data - covariance matrix (only x_1, x_2, y_1, y_2)

- ▶ Replace `cor` by `cov` in the last command in the Script Window
- ▶ $\text{cov}(x_1, y_1) = 5.501$
- ▶ Matrix is symmetrical with values on the diagonal = variances, eg, $\text{cov}(y_1, y_1) = \text{var}(y_1) = 4.127269$

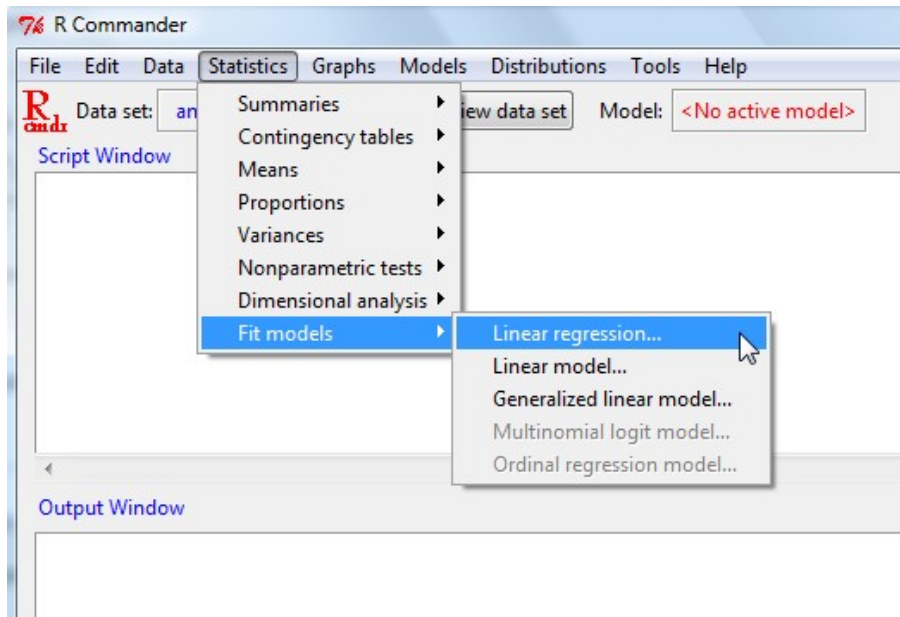


The screenshot shows the R Commander interface. In the Script Window, the command `cov(anscombe[,c("x1", "x2", "y1", "y2")], use="complete.obs")` is entered. The Output Window displays the resulting covariance matrix, which is a 4x4 symmetric matrix. The diagonal elements represent the variances of x_1 , x_2 , y_1 , and y_2 . The off-diagonal elements represent the covariances between these variables. Red boxes highlight the values 5.501000 and 4.127269, which correspond to the covariances between x_1 and y_1 , and y_1 and y_1 (variance of y_1), respectively.

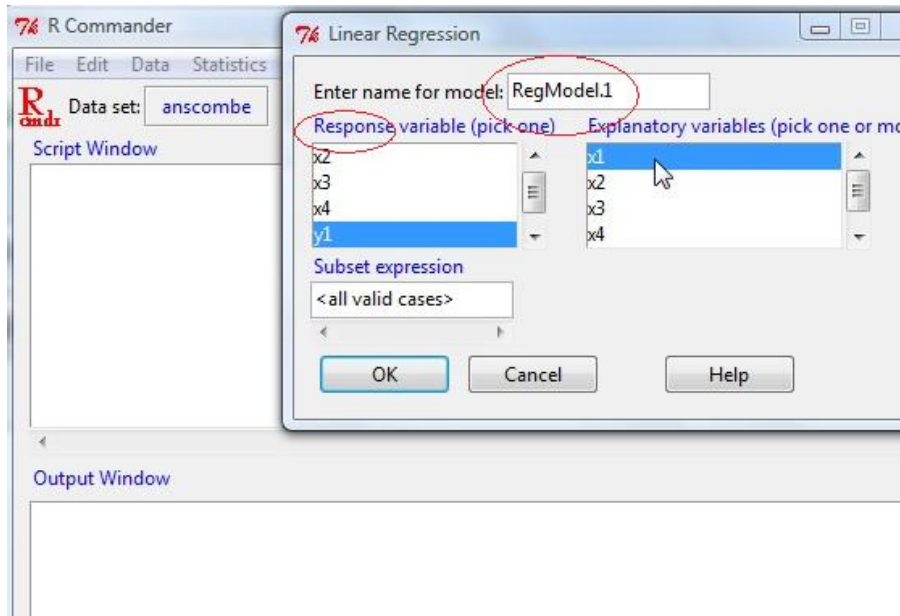
```
> cov(anscombe[,c("x1", "x2", "y1", "y2")], use="complete.obs")
```

	x1	x2	y1	y2
x1	11.000	11.000	5.501000	5.500000
x2	11.000	11.000	5.501000	5.500000
y1	5.501	5.501	4.127269	3.095609
y2	5.500	5.500	3.095609	4.127269

Simple linear regression - y_1 versus x_1

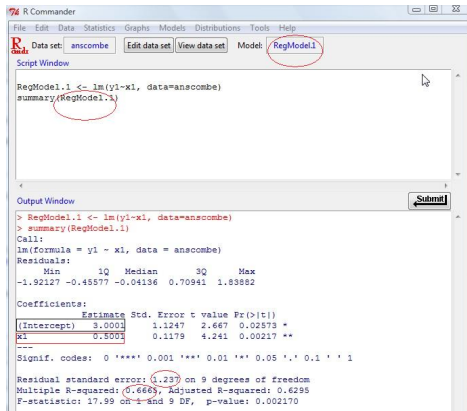


Simple linear regression - y_1 versus x_1 cont.



Simple linear regression - y1 versus x1 cont.

- ▶ Intercept estimate: $a = 3.0001$
- ▶ Slope estimate: $b = 0.5001$
- ▶ Residual standard deviation: $s_R = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}} = 1.237$
- ▶ R-squared: $R^2 = 0.6665 \Rightarrow \text{cor}(x, y) = \sqrt{0.6665}$



The screenshot shows the R Commander window. The 'Data set' is 'anscombe' and the 'Model' is 'RegModel.1'. The 'Script Window' contains the following code:

```
RegModel.1 <- lm(y1~x1, data=anscombe)
summary(RegModel.1)
```

The 'Output Window' displays the results of the linear regression:

```
> RegModel.1 <- lm(y1~x1, data=anscombe)
> summary(RegModel.1)
Call:
lm(formula = y1 ~ x1, data = anscombe)
Residuals:
    Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.0001     1.1247    2.667  0.02573 *
x1           0.5001     0.1179    4.241  0.00217 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared: 0.6665, Adjusted R-squared: 0.6295
F-statistic: 17.99 on 1 and 9 DF, p-value: 0.002170
```

Red circles highlight the 'RegModel.1' model name, the 'summary(RegModel.1)' function call, the '1.237' value in the residual standard error, and the '0.6665' value in the multiple R-squared.

Regression Diagnostics: Tools for Checking the Validity of a Model (I)

- ▶ Determine whether the proposed regression model provides an adequate fit to the data: **plots of standardized residuals**.
- ▶ The plots assess visually whether the assumptions are being violated.
- ▶ Determine which (if any) of the data points have x values that have an unusually large effect on the estimated regression model (**leverage points**).
- ▶ Determine which (if any) of the data points are *outliers*: points which do not follow the pattern set by the bulk of the data.

Regression Diagnostics: Tools for Checking the Validity of a Model (II)

- ▶ If leverage points exist, determine whether each is a *bad leverage point*. If a bad leverage point exists we shall assess its influence on the fitted model.
- ▶ Examine whether the assumption of constant variance of the errors is reasonable. If not, we shall look at how to overcome this problem.
- ▶ If the data are collected over time, examine whether the data are correlated over time.
- ▶ If the sample size is small or prediction intervals are of interest, examine whether the assumption that the errors are normally distributed is reasonable.

Data

Sources:

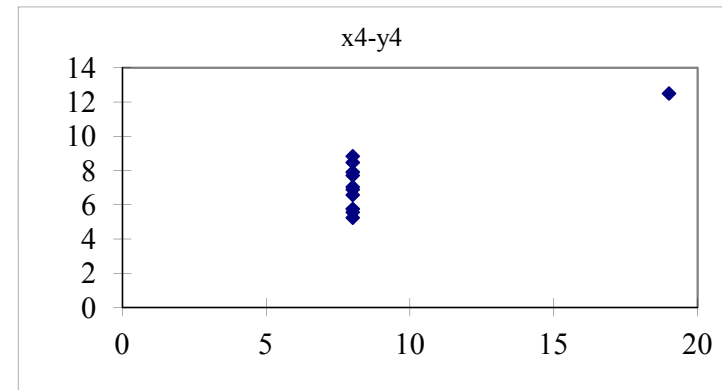
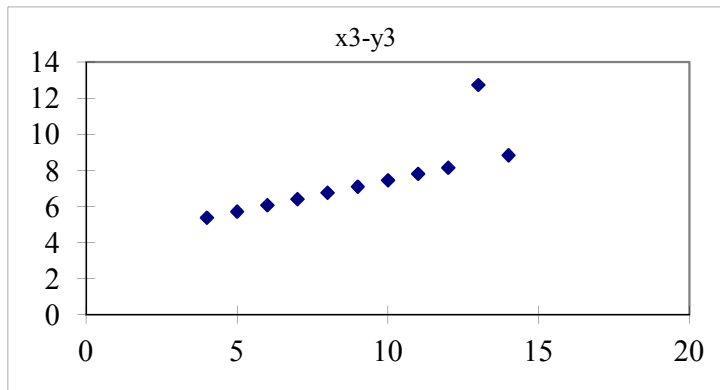
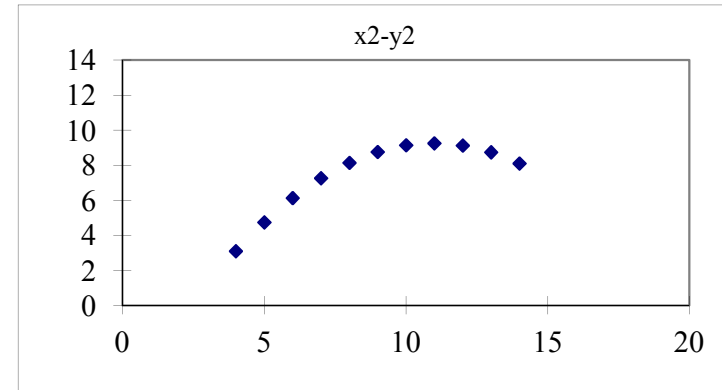
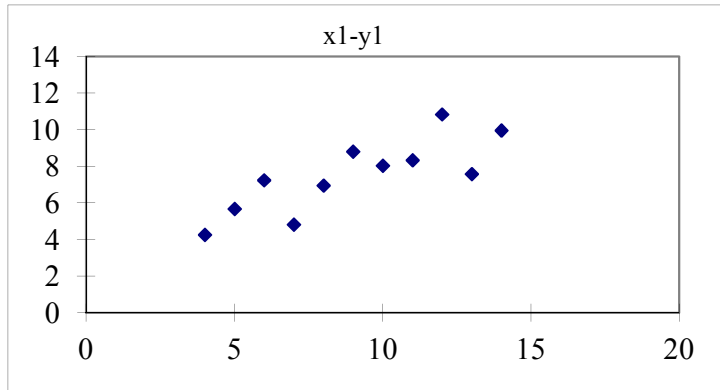
Edward R. Tufte, *The Visual Display of Quantitative Information* (Cheshire, Connecticut: Graphics Press, 1983), pp. 14-15.

F.J. Anscombe, "Graphs in Statistical Analysis," *American Statistician*, vol. 27 (Feb 1973), pp. 17-21.

Anscombe's Data											
Observation	x1	y1		x2	y2		x3	y3		x4	y4
1	10	8,04		10	9,14		10	7,46		8	6,58
2	8	6,95		8	8,14		8	6,77		8	5,76
3	13	7,58		13	8,74		13	12,74		8	7,71
4	9	8,81		9	8,77		9	7,11		8	8,84
5	11	8,33		11	9,26		11	7,81		8	8,47
6	14	9,96		14	8,1		14	8,84		8	7,04
7	6	7,24		6	6,13		6	6,08		8	5,25
8	4	4,26		4	3,1		4	5,39		19	12,5
9	12	10,84		12	9,13		12	8,15		8	5,56
10	7	4,82		7	7,26		7	6,42		8	7,91
11	5	5,68		5	4,74		5	5,73		8	6,89
Summary Statistics											
N	11	11		11	11		11	11		11	11
mean	9,00	7,50		9,00	7,50091		9,00	7,50		9,00	7,50
SD	3,16	1,94		3,16	1,94		3,16	1,94		3,16	1,94
r	0,82			0,82			0,82			0,82	

Use the charts below to get the regression lines via Excel's Trendline feature.

Data

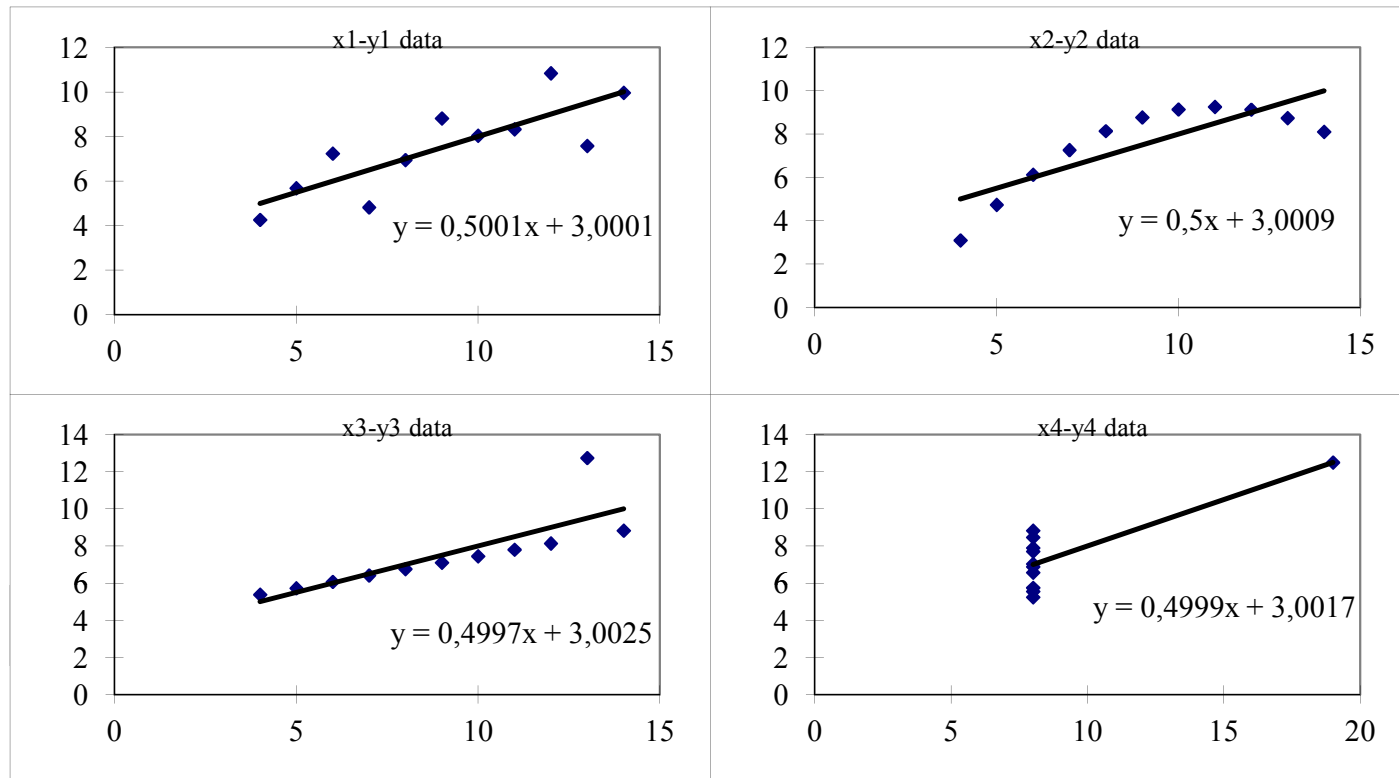


Data

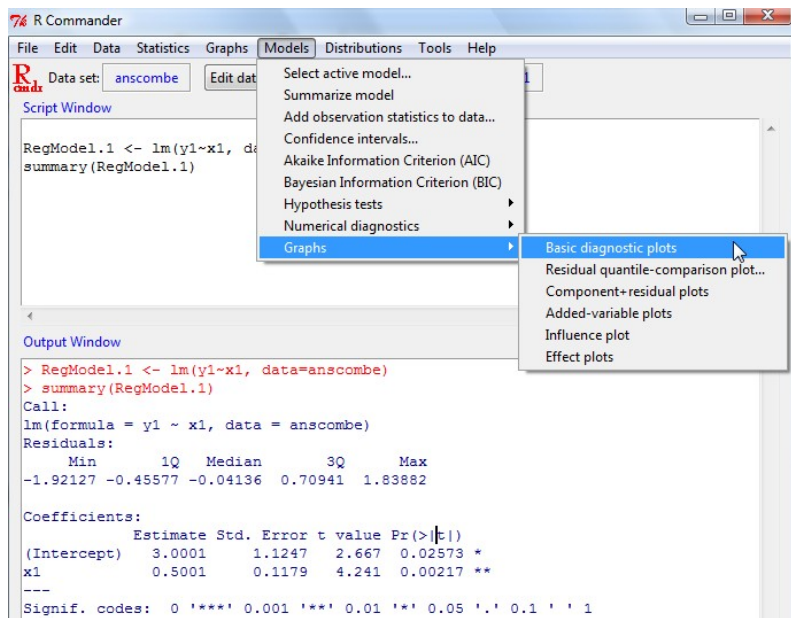
Regression Results

LINEST OUTPUT		<u>x1-y1</u>		<u>x2-y2</u>		<u>x3-y3</u>		<u>x4-y4</u>	
slope	intercept	0,50	3	0,50	3	0,50	3	0,50	3
SE	SE	0,12	1,12	0,12	1,13	0,12	1,12	0,12	1,12
R ²	RMSE	0,67	1,24	0,67	1,24	0,67	1,24	0,67	1,24
F	df	17,99	9	17,97	9	17,97	9	18,00	9
Reg SS	SSR	27,51	13,76	27,50	13,78	27,47	13,76	27,49	13,74

Data



Simple linear regression - residual plot (method 1)



The screenshot shows the R Commander interface. The 'Data set' is 'anscombe'. The 'Script Window' contains the following R code:

```
RegModel.1 <- lm(y1~x1, data=anscombe)
summary(RegModel.1)
```

The 'Output Window' displays the summary of the linear model:

```
> RegModel.1 <- lm(y1~x1, data=anscombe)
> summary(RegModel.1)
Call:
lm(formula = y1 ~ x1, data = anscombe)
Residuals:
    Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x1             0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

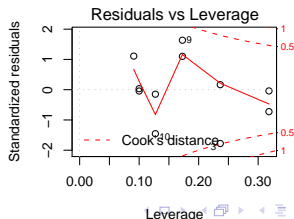
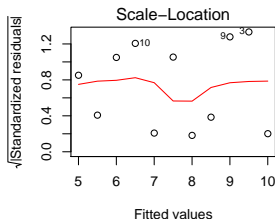
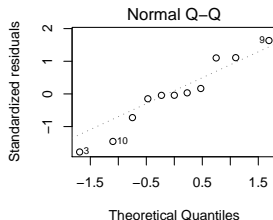
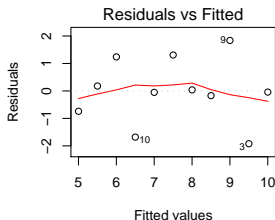
The 'Models' menu is open, and the 'Basic diagnostic plots' option is selected, which has opened a sub-menu. The sub-menu options are:

- Basic diagnostic plots
- Residual quantile-comparison plot...
- Component+residual plots
- Added-variable plots
- Influence plot
- Effect plots

Simple linear regression - residual plot (method 1) cont.

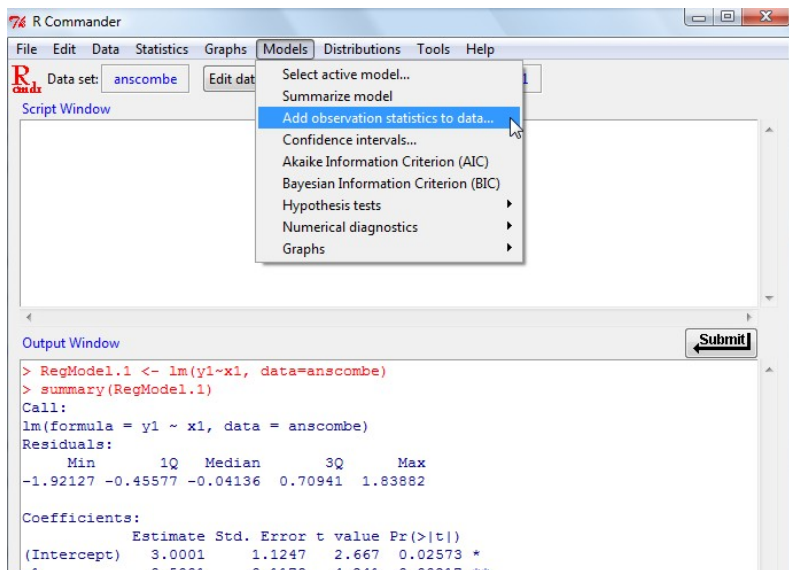
- Residuals versus fitted (top left plot)

$\text{lm}(y1 \sim x1)$



Simple linear regression - residual plot (method 2)

- Append the fitted values, residuals, standardized residuals etc to the existing data set



The screenshot shows the R Commander application window. The 'Models' menu is open, and the option 'Add observation statistics to data...' is highlighted. The 'Data set' is 'anscombe'. The 'Output Window' at the bottom displays the results of a linear regression model.

Output Window

```
> RegModel.1 <- lm(y1~x1, data=anscombe)
> summary(RegModel.1)
```

Call:
lm(formula = y1 ~ x1, data = anscombe)

Residuals:

	Min	1Q	Median	3Q	Max
	-1.92127	-0.45577	-0.04136	0.70941	1.83882

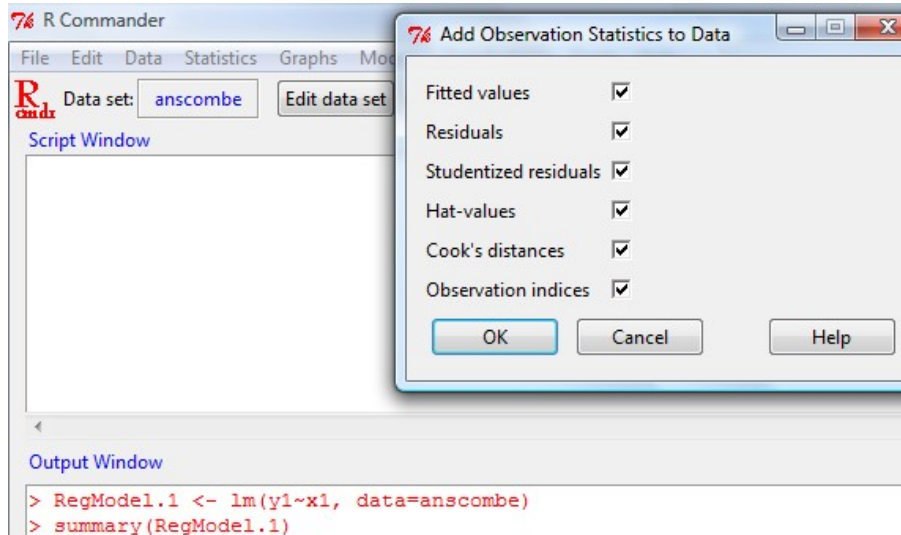
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.0001	1.1247	2.667	0.02573 *

Submit

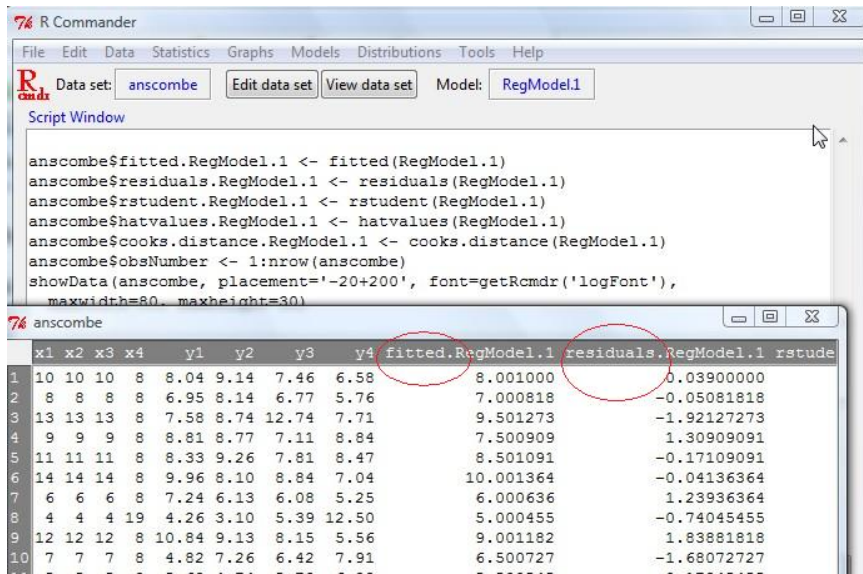
Simple linear regression - residual plot (method 2 cont.)

- Append the fitted values, residuals, studentized residuals etc to the existing data set



Simple linear regression - residual plot (method 2 cont.)

- Now the data set has new columns on the right with \hat{y} , r , etc



The screenshot shows the R Commander interface. The top menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. Below the menu bar, the 'Data set' is 'anscombe', and the 'Model' is 'RegModel.1'. The 'Script Window' contains the following R code:

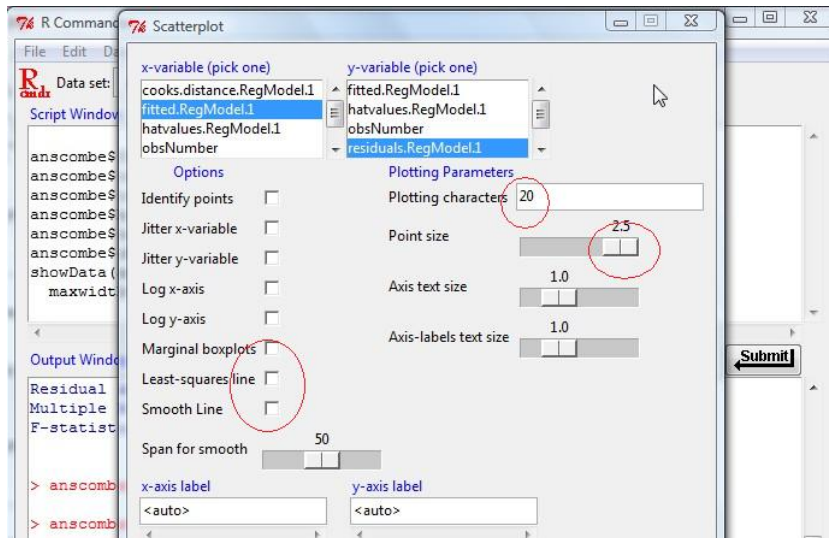
```
anscombe$fitted.RegModel.1 <- fitted(RegModel.1)
anscombe$residuals.RegModel.1 <- residuals(RegModel.1)
anscombe$rstudent.RegModel.1 <- rstudent(RegModel.1)
anscombe$hatvalues.RegModel.1 <- hatvalues(RegModel.1)
anscombe$cooks.distance.RegModel.1 <- cooks.distance(RegModel.1)
anscombe$obsNumber <- 1:nrow(anscombe)
showData(anscombe, placement='-20+200', font=getRcmdr('logFont'),
maxwidth=80, maxheight=30)
```

The 'Data Window' displays a table with 10 rows and 13 columns. The columns are labeled x1, x2, x3, x4, y1, y2, y3, y4, fitted.RegModel.1, residuals.RegModel.1, and rstudent. The first four columns (x1-x4) contain the same values for each row. The fifth column (y1) contains values ranging from 6.58 to 10.84. The sixth column (y2) contains values ranging from 3.10 to 9.14. The seventh column (y3) contains values ranging from 5.39 to 8.15. The eighth column (y4) contains values ranging from 5.56 to 7.91. The ninth column (fitted.RegModel.1) contains values ranging from 5.000455 to 10.001364. The tenth column (residuals.RegModel.1) contains values ranging from -1.68072727 to 1.30909091. The eleventh column (rstudent) contains values ranging from -0.17109091 to 1.83881818. Red circles highlight the 'fitted.RegModel.1' and 'residuals.RegModel.1' columns in the table.

	x1	x2	x3	x4	y1	y2	y3	y4	fitted.RegModel.1	residuals.RegModel.1	rstudent
1	10	10	10	8	8.04	9.14	7.46	6.58	8.001000	0.03900000	
2	8	8	8	8	6.95	8.14	6.77	5.76	7.000818	-0.05081818	
3	13	13	13	8	7.58	8.74	12.74	7.71	9.501273	-1.92127273	
4	9	9	9	8	8.81	8.77	7.11	8.84	7.500909	1.30909091	
5	11	11	11	8	8.33	9.26	7.81	8.47	8.501091	-0.17109091	
6	14	14	14	8	9.96	8.10	8.84	7.04	10.001364	-0.04136364	
7	6	6	6	8	7.24	6.13	6.08	5.25	6.000636	1.23936364	
8	4	4	4	19	4.26	3.10	5.39	12.50	5.000455	-0.74045455	
9	12	12	12	8	10.84	9.13	8.15	5.56	9.001182	1.83881818	
10	7	7	7	8	4.82	7.26	6.42	7.91	6.500727	-1.68072727	

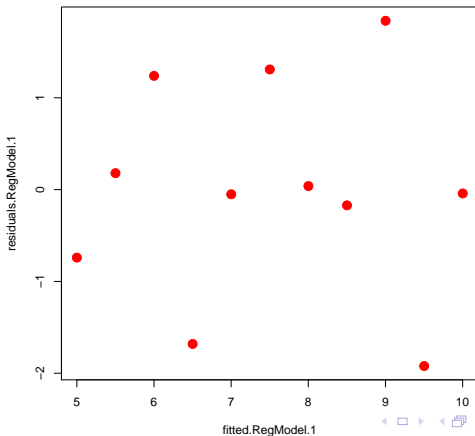
Simple linear regression - residual plot (method 2 cont.)

- Use the **scatterplot** option in the **Graphs** menu to plot residuals versus fitted



Simple linear regression - residual plot (method 2 cont.)

- ▶ Residuals versus fitted (cloud of points oscillates around the horizontal axis $y = 0$)
- ▶ There is no pattern, no heteroscedasticity \Rightarrow regression model is appropriate



Simple linear regression - residual plot (method 2 cont.)

- ▶ Studentized Residuals ($\frac{r_i}{s_R}$) versus x_1 (cloud of points oscillates around the horizontal axis $y = 0$)
- ▶ There is no pattern, no heteroscedasticity \Rightarrow regression model is appropriate

