Linear Regression

2022/2023

Luís Paquete

University of Coimbra

Linear Regression

Contents

Linear Regression

Regression model

- A mathematical model that describes the behavior of a system over a range of input values.
- A regression model allows to predict how the system will perform when given an input value that was not measured.
- A linear regression model assumes a linear relationship between the input variable and the ouput variable.

Linear Regression

Regression model

- A simple **linear regression model** has the form

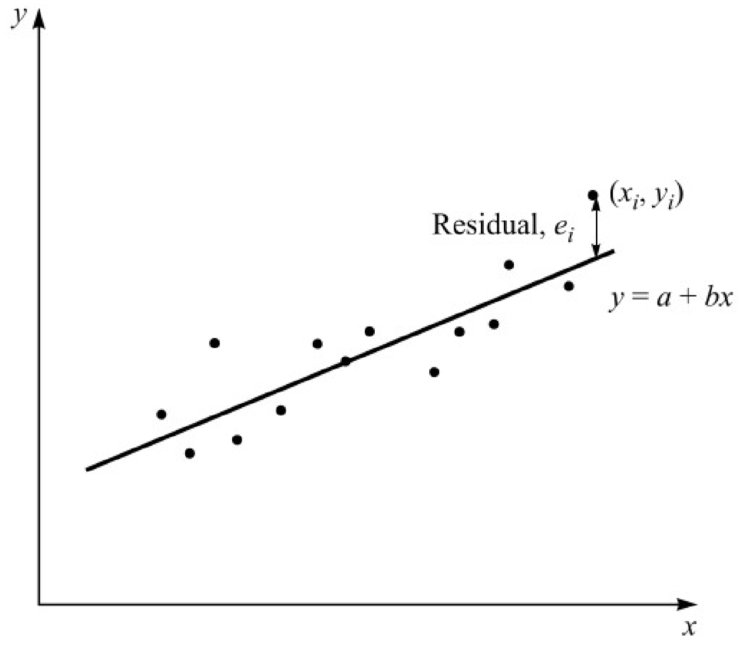
$$y = a + bx$$

where x is the input variable, y is the predicted output variable and a and b are the regression parameters.

- If y_i is the value measured for the input value x_i, then (x_i,y_i) can be written as

$$y_i = a + bx_i + e_i$$

where e_i is the **residual** for the i-th measurement, that is, the difference between the measured value for y_i and that would have been predicted from the model.

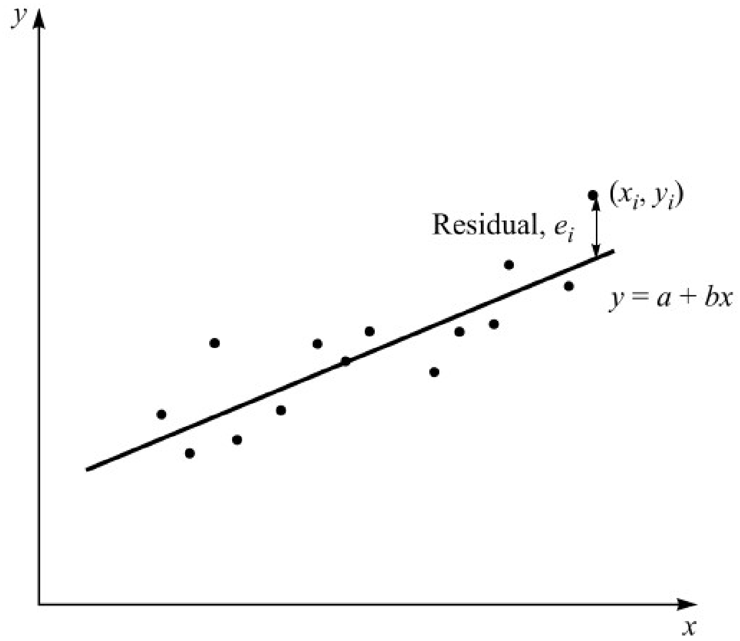


Linear Regression

Regression model

- To find *a* and *b* that will form a line that most closely fits the *n* measured data points, minimize the sum of squares of the residuals, *SSE*:

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$



Linear Regression

A side note: Why the sum of squares?

- Why not the sum of absolute differences? This function is not differentiable at 0. Then, the minimizers of the function cannot be easily found.

- The sum of squares function is differentiable everywhere and it is convex, that is, the local minimum is also global minimum. Moreover, *a* and *b* can be calculated by a closed formula.

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

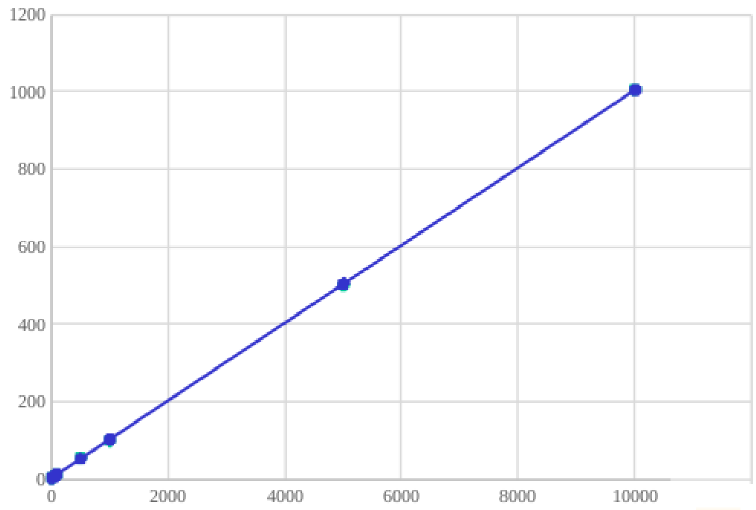
$$a = \bar{y} - b\bar{x}$$

Linear Regression

Example

Develop a regression model to relate the time required to perform a file-read operation to the number of bytes read

File size in bytes	Times in ms
10	3.8
50	8.1
100	11.9
500	55.6
1000	99.6
5000	500.2
10000	1006.1



$$y = 2.24 + 0.1002\, x$$

Linear Regression

Example in R

```
> D = read.table("regr.in",header=TRUE)
> lr.out = lm(D$time~D$size)
> summary(lr.out)

Call:
lm(formula = D$time ~ D$size)

Residuals:
      1       2       3       4       5       6       7 
 0.5584  0.8497 -0.3612  3.2518 -2.8570 -3.1270  1.6854 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.239467   1.163822   1.924    0.112    
D$size      0.100218   0.000274 365.717  2.9e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.55 on 5 degrees of freedom
Multiple R-squared:     1,	Adjusted R-squared:     1 
F-statistic: 1.337e+05 on 1 and 5 DF,  p-value: 2.901e-12
```

Linear Regression

Multiple linear regression

- Multiple linear regression extends linear regression for k > 1 independent input variables

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_k x_k$$

- Each data point (x_{1i}, x_{2i}, ..., x_{ki}, y_i) can be expressed as

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_k x_{ki} + e_i$$

where e_i is the residual

Linear Regression

Multiple linear regression

- The square sum of errors (SSE) is

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i} - \cdots - b_k x_{ki})^2$$

- Using matrix notation, we have a multiple linear regression model as follows

$$Y = X\,b + e$$

where $b = (X^T X)^{-1} X^T\, Y$ minimizes SSE

Linear Regression

Example

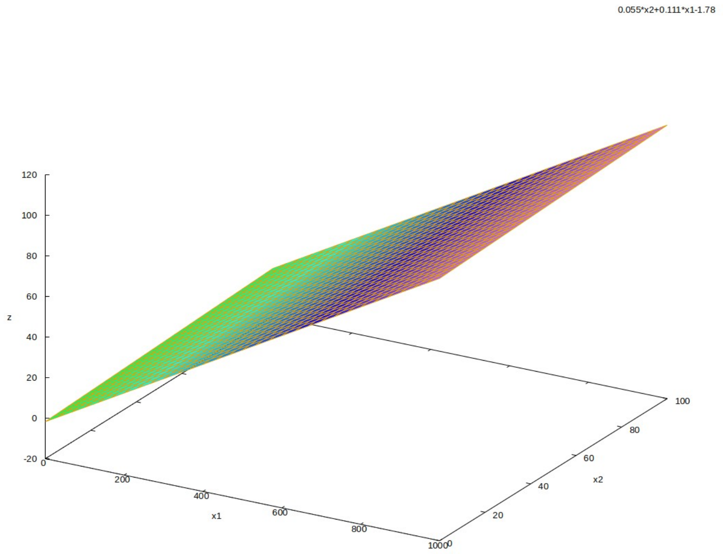
Develop a regression model to relate the time required to perform a certain number of input-output and memory operations

IO operations	Mem. operations	Times in ms
10	10	2.8
10	100	3.1
100	10	10.9
100	100	12.6
1000	10	106.2
1000	100	119.1

Linear Regression

Example in R

```
> D <- read.table("regr5.in",header=TRUE)
> lr.out <- lm(D$time ~ D$IO + D$Mem)
> summary(lr.out)


Call:
lm(formula = R$time ~ R$IO + R$mem)


Residuals:
      1       2       3       4       5       6
 2.9144 -1.7523  0.9941 -2.2725 -3.9086  4.0248

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.779630   2.947538  -0.604    0.589
R$IO         0.111336   0.003698  30.104 8.05e-05 ***
R$Mem        0.055185   0.036737   1.502    0.230
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 4.049 on 3 degrees of freedom
Multiple R-squared:  0.9967, Adjusted R-squared:  0.9945
F-statistic: 454.2 on 2 and 3 DF,  p-value: 0.0001888
```



$$y = -1.780 + 0.111\ x_1 + 0.055\ x_2$$

Linear Regression

Multivariate linear regression

- Multivariate linear regression extends linear regression for m > 1 dependent variables

$$Y = B_0 + B_1 x_1 + B_2 x_2 + \dots + B_k x_k$$

- Each data point ($x_{1i}, x_{2i}, \dots, x_{ki}, y_{ij}$) can be expressed as

$$y_{ij} = b_{0j} + b_{1j} x_{1i} + b_{2j} x_{2i} + \dots + b_{kj} x_{ki} + e_{ij}$$

where e_{ij} is the residual

Linear Regression

Coefficient of determination

- Determine how much of the total variation is "explained" by the linear model.
- ***SST*** is the total variation of the measured system output

$$SST = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

which is partitioned into two components:

SSR: portion of the *SST* that is explained by the regression model

SSE: portion of the *SST* that is due to the measurement error

Linear Regression

Coefficient of determination

- The **coefficient of determination** r^2 is the fraction of *SST* "explained" by the model

$$r^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST}$$

- If $r^2 = 0$, then SSE is as large as SST
- If $r^2 = 1$, then SSE is 0

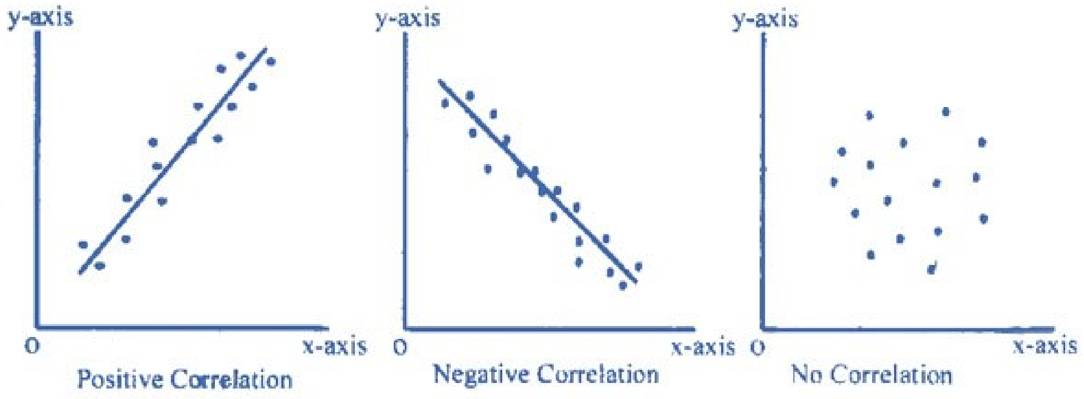
Linear Regression

Coefficient of correlation

- The coeficient of determination is the squared value of the **coefficient of correlation** of *x* and *y***.**

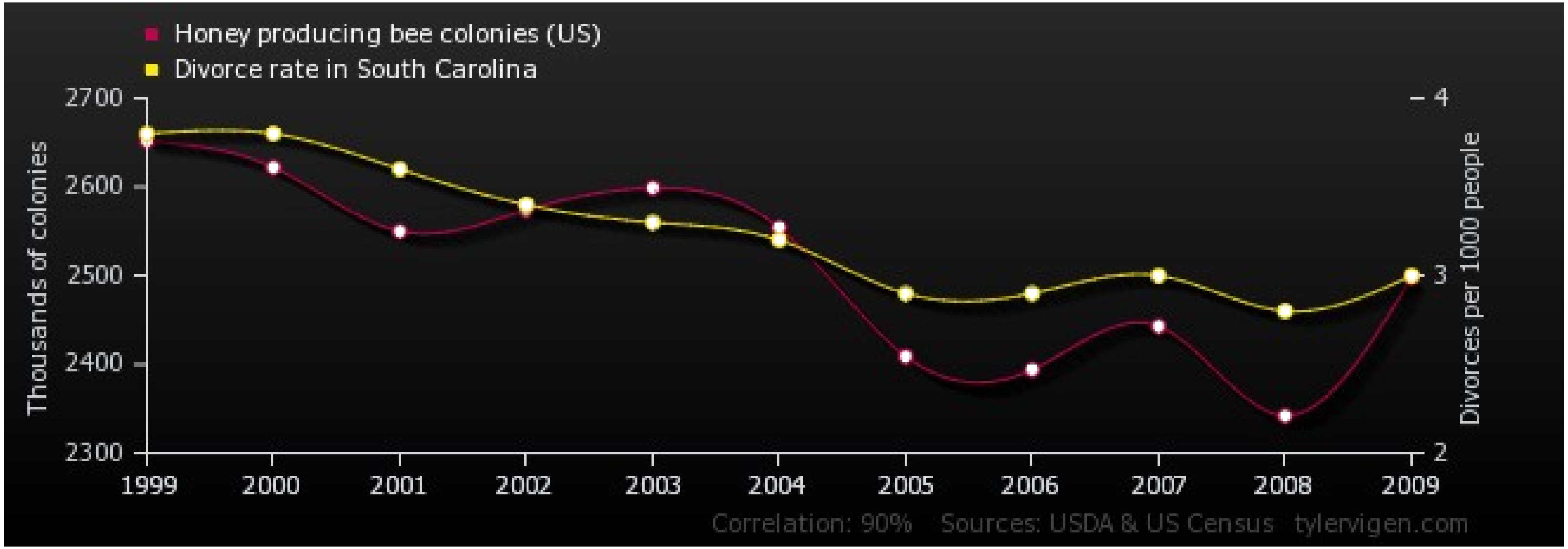
$$r = \pm\sqrt{\frac{SSR}{SST}} = \mathrm{Cor}(x, y)$$

- It allows to investigate whether the correlation between input and output is positive ($0 < r \leq 1$) or negative ($-1 \leq r < 0$). It indicates the strength of the linear relation.



Coefficient of correlation

- **A side note**: *correlation does not imply causation*

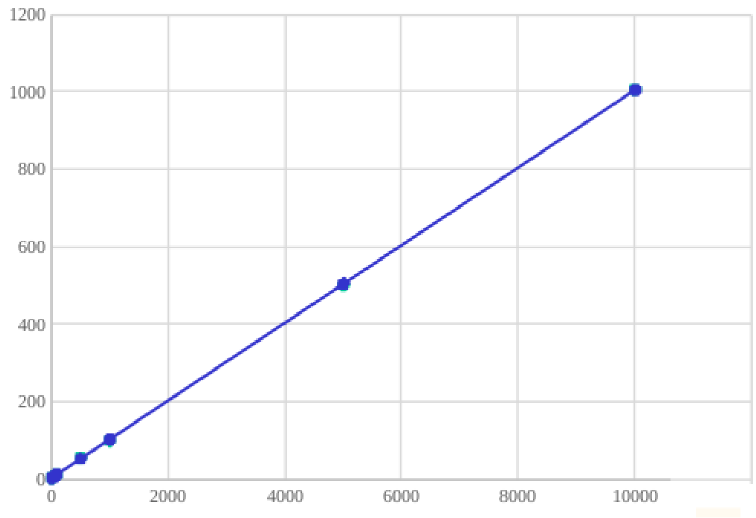


Linear Regression

Example

Develop a regression model to relate the time required to perform a file-read operation to the number of bytes read

File size in bytes	Times in ms
10	3.8
50	8.1
100	11.9
500	55.6
1000	99.6
5000	500.2
10000	1006.1



$y = 2.24 + 0.1002\, x$ $\qquad r^2 = 0.9996$

Linear Regression

Example in R

```
> D = read.table("regr.in",header=TRUE)
> lr.out = lm(D$time~D$size)
> summary(lr.out)

Call:
lm(formula = D$time ~ D$size)

Residuals:
      1       2       3       4       5       6       7
 0.5584  0.8497 -0.3612  3.2518 -2.8570 -3.1270  1.6854

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.239467   1.163822   1.924    0.112
D$size      0.100218   0.000274 365.717  2.9e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.55 on 5 degrees of freedom
Multiple R-squared:     1,     Adjusted R-squared:     1
F-statistic: 1.337e+05 on 1 and 5 DF,  p-value: 2.901e-12
```

Linear Regression

Assumptions of linear regression

- A more complete examination of the underying **assumptions** of linear regression may indicate whether the model can be used for **prediction** (inference).
- In R, the linear regression assumptions can be verified by doing

```
plot(<linear model>)
```

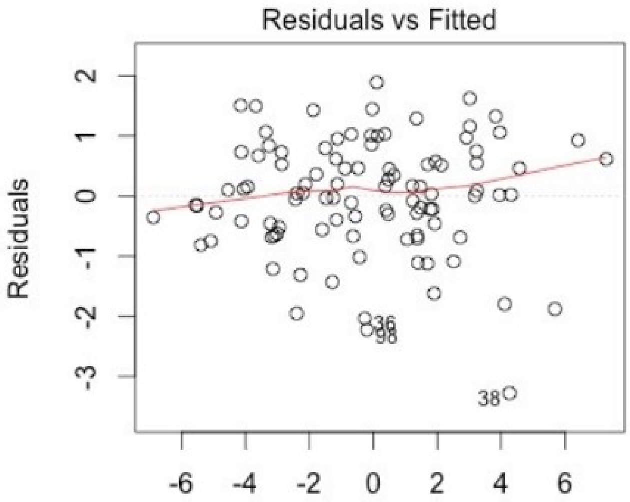
Linear Regression

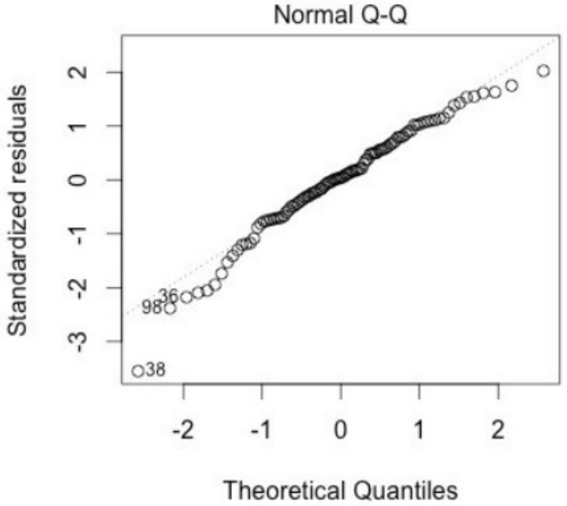
Assumptions of linear regression

Residuals-vs-fitted plot allows to verify:

- **Linearity**: the mean residual value for every fitted value region (red line) should be close to 0.
- **Homoskedasticity** (constante variance): The spread of residuals should be approximately the same across the x-axis.
- **Outliers:** identify extreme residuals

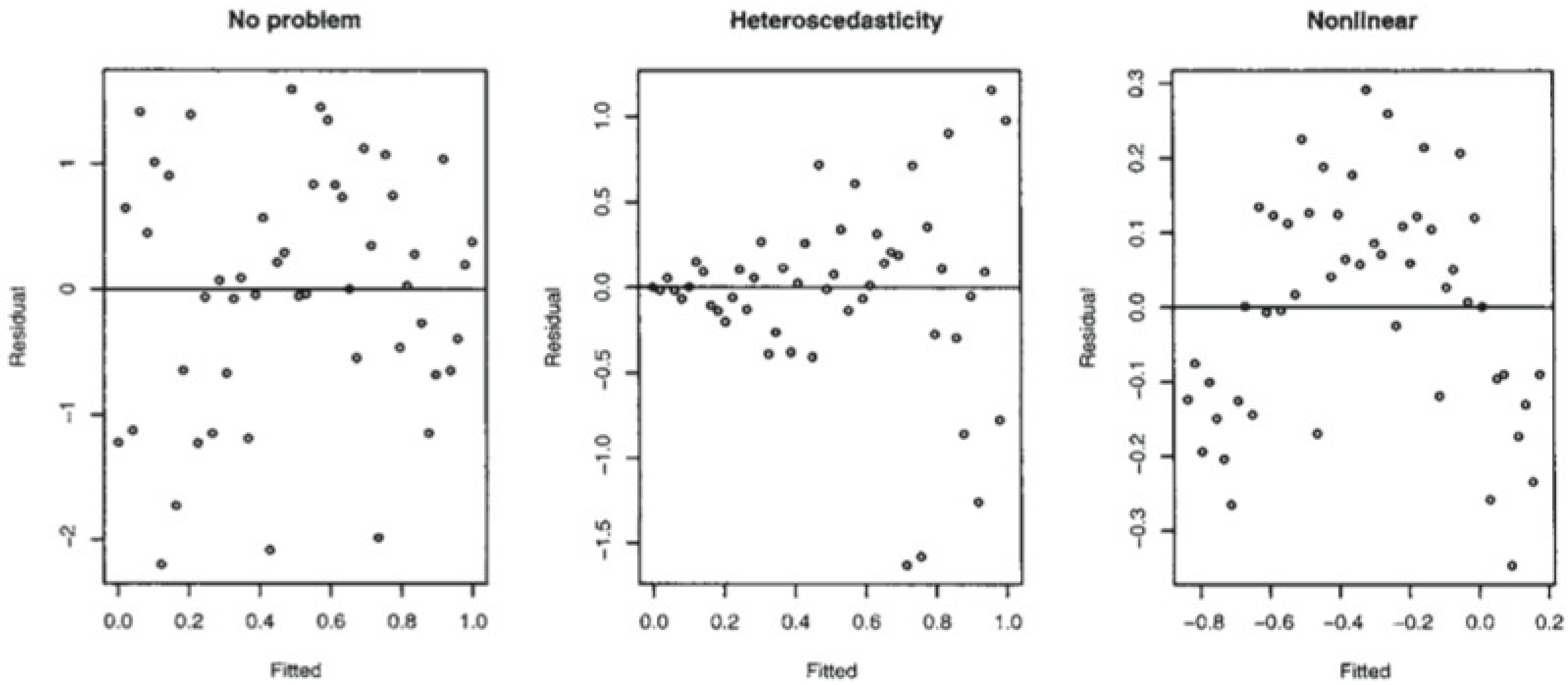
Normal Q-Q plot to verify the **normality of residuals**





Linear Regression

Example:



Linear Regression

Transformations

- A way of overcoming the problem with assumptions is to transform the data

  Rule of Thumb 1: Transforming y may correct problems with the error terms.

  Rule of Thumb 2: Transforming x may correct the non-linearity.

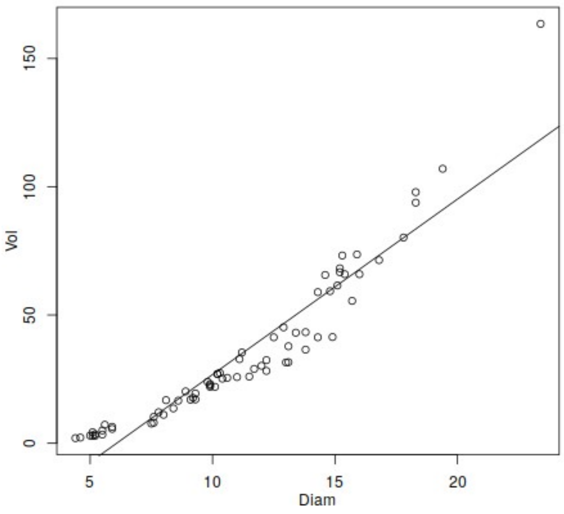
- However, a transformed model may be harder to interpret

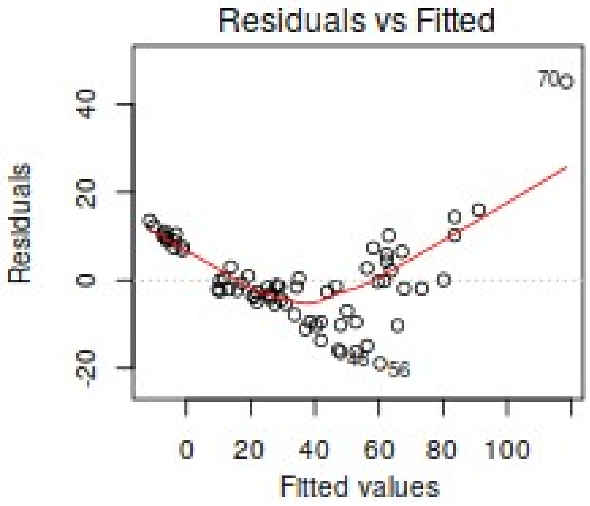
Linear Regression

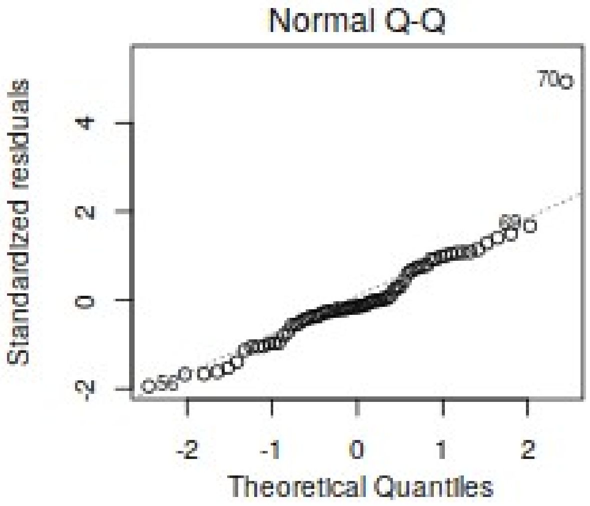
Transformations

Example: (D. Bruce and F. X. Schumacher, 1935)

- Predict the volume of a tree (y) from its diameter (x)







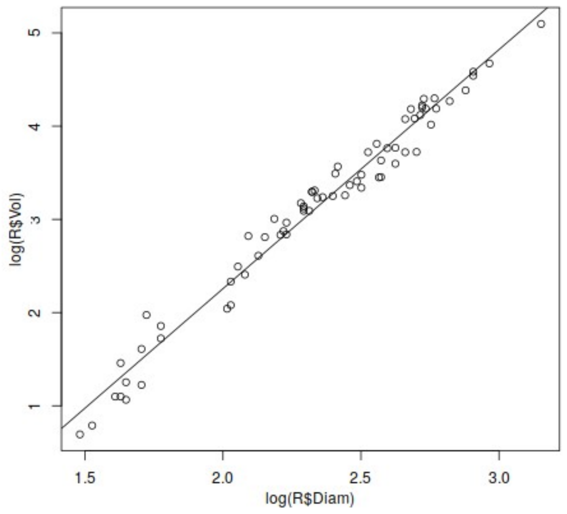
$y = -41.57 + 6.93\ x \qquad r^2 = 0.89$

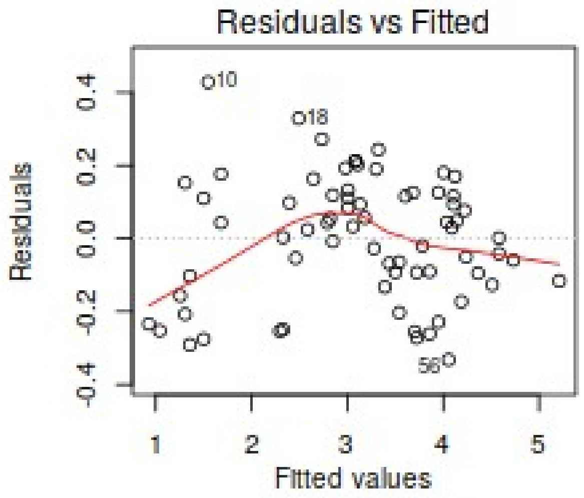
Linear Regression

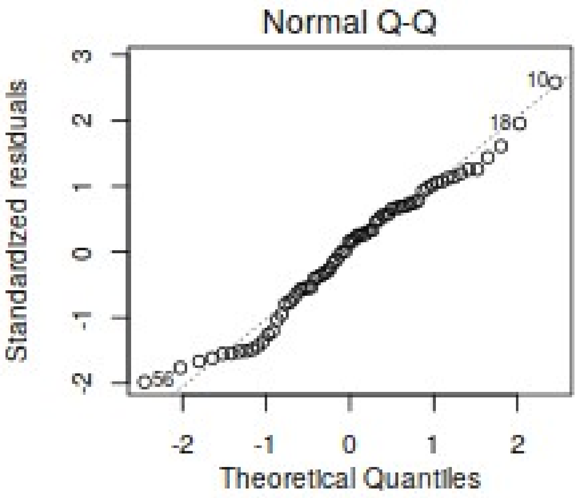
Transformations

Example: (D. Bruce and F. X. Schumacher, 1935)

- Predict the log of the volume of a tree (ln y) from the log of its diameter (ln x)







$\ln y = -2.87 + 2.56 \ln x \qquad r^2 = 0.97$

Linear Regression

Transformations

- It is also possible to deduce a possible transformation by plotting the data or having some assumption about the process of generating y values
- For instance, if an exponential behavior is expected, such as

$$y = ab^x$$

by taking the logarithm of both sides

$$\ln y = \ln a + (\ln b)x$$

the expression has a linear form:

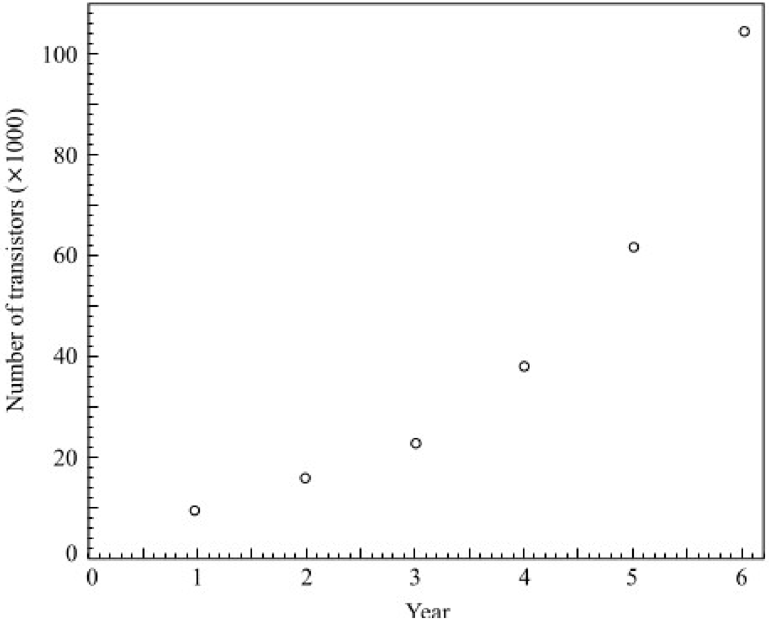
$$y' = a' + b'x$$

Linear Regression

Example

Develop a regression model for the number of transistors in the following years

Year	Transistors
1	9500
2	16000
3	23000
4	38000
5	62000
6	105000



Linear Regression

Example in R

```
> D = read.table("regr1.in",header=TRUE)
> lr.out = lm(D$number~D$year)
> summary(lr.out)

Call:
lm(formula = D$number ~ D$year)

Residuals:
       1        2         3         4        5        6
 12285.7    771.4 -10242.9 -13257.1  -7271.4  17714.3

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -20800      13156  -1.581  0.18904
D$year         18014       3378   5.332  0.00596 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14130 on 4 degrees of freedom
Multiple R-squared: 0.8767,   Adjusted R-squared: 0.8458
F-statistic: 28.44 on 1 and 4 DF,  p-value: 0.005955
```

Linear Regression

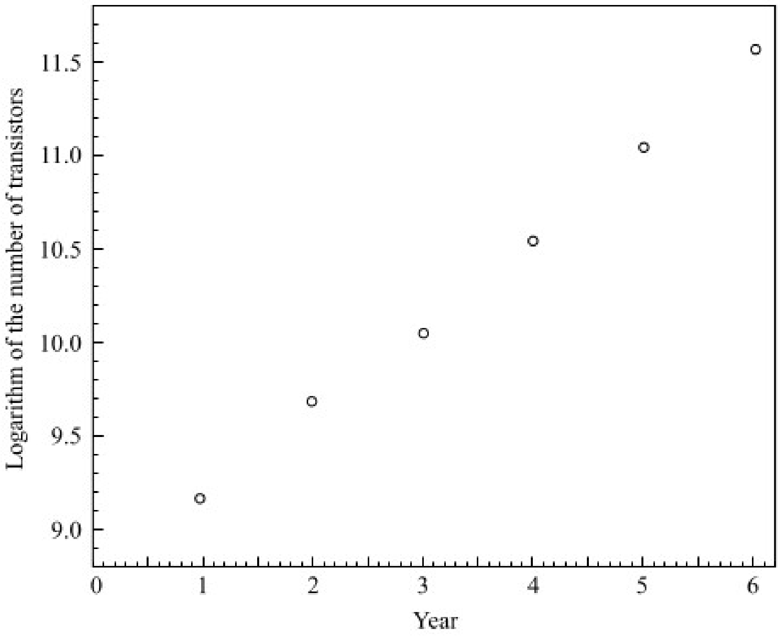
Example

Develop a regression model for the number of transistors in the following years

Year	ln(Transistors)
1	9.1590
2	9.6803
3	10.0432
4	10.5453
5	11.0349
6	11.5617

$b' = 0.474$

$a' = 8.679$



$y' = 8.679 + 0.474x$

Linear Regression

Example in R

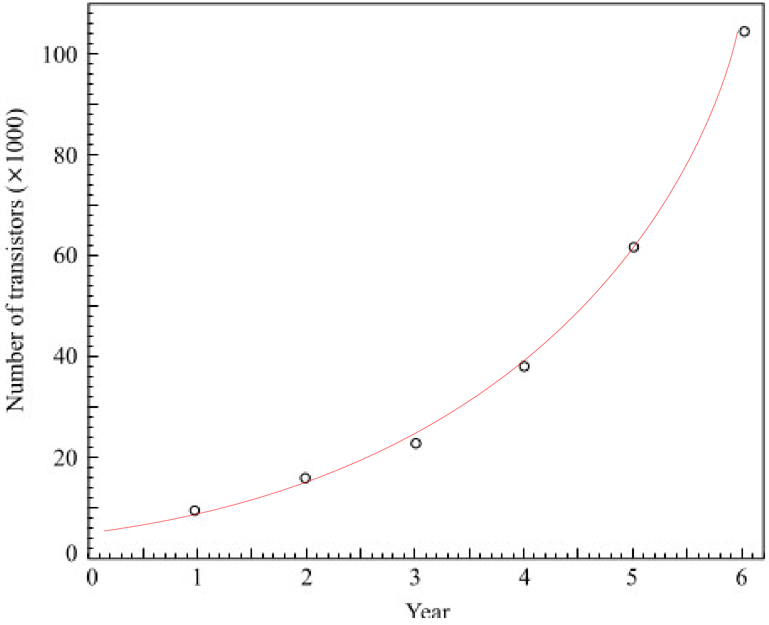
```
> D = read.table("regr1.in",header=TRUE)
> lr.out = lm(log(D$number)~D$year)
> summary(lr.out)

Call:
lm(formula = log(D$number) ~ D$year)

Residuals:
        1         2         3         4         5         6 
 0.005835  0.053444 -0.057338 -0.028934 -0.013073  0.040065 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.67952    0.04364  198.87 3.84e-09 ***
D$year       0.47369    0.01121   42.27 1.87e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04688 on 4 degrees of freedom
Multiple R-squared: 0.9978,   Adjusted R-squared: 0.9972
F-statistic:  1787 on 1 and 4 DF,  p-value: 1.873e-06
```

Linear Regression

Example

Develop a regression model for the number of transistors in the following years

Year	Transistors
1	9500
2	16000
3	23000
4	38000
5	62000
6	105000



$b' = 0.474$ $\qquad b = e^{b'} = 1.61$

$a' = 8.679$ $\qquad a = e^{a'} = 5878$

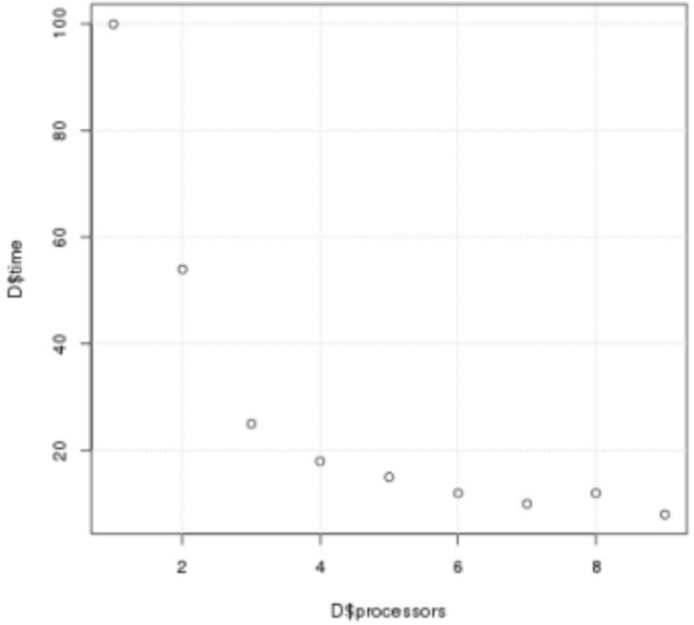
$y = (5878)1.61^x$

Linear Regression

Example

Develop a regression model for the relation between CPU-time and number of processors

Processors	CPU-time
1	100
2	54
3	25
4	18
5	15
6	12
7	10
8	12
9	8



Linear Regression

Example in R

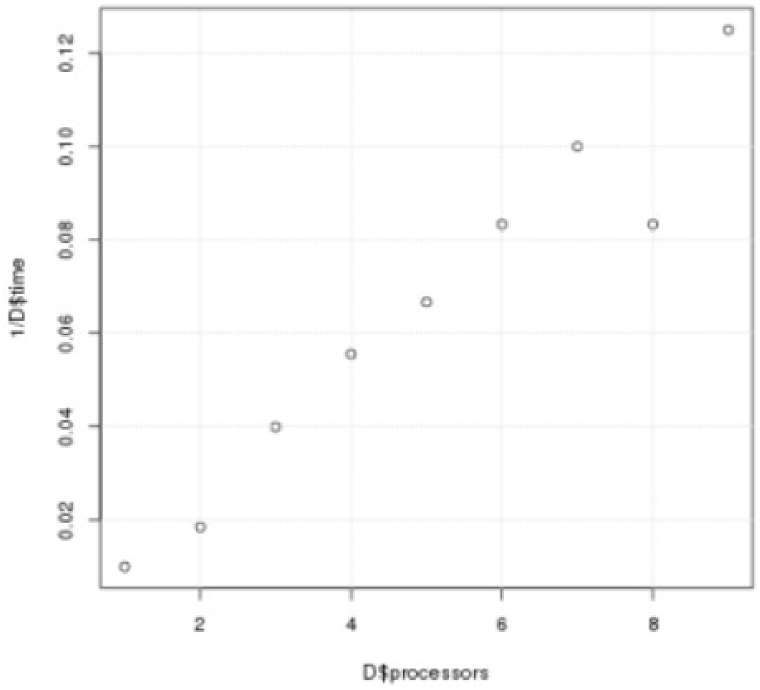
```
> D = read.table("regr3.in",header=TRUE
> lr.out = lm(D$time~D$processors)
> summary(lr.out)

Call:
lm(formula = D$time ~ D$processors)

Residuals:
    Min      1Q  Median      3Q     Max
-20.889 -13.222  -0.722  10.278  36.444

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    72.389     14.246   5.081  0.00143 **
D$processors   -8.833      2.532  -3.489  0.01014 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.61 on 7 degrees of freedom
Multiple R-squared: 0.6349,   Adjusted R-squared: 0.5828
F-statistic: 12.17 on 1 and 7 DF,  p-value: 0.01014
```

Linear Regression

Example

Reciprocal transformation: $\frac{1}{y} = a + bx$

Processors	CPU-time^{-1}
1	0.01
2	0.02
3	0.04
4	0.06
5	0.07
6	0.08
7	0.10
8	0.08
9	0.13



Linear Regression

Example in R

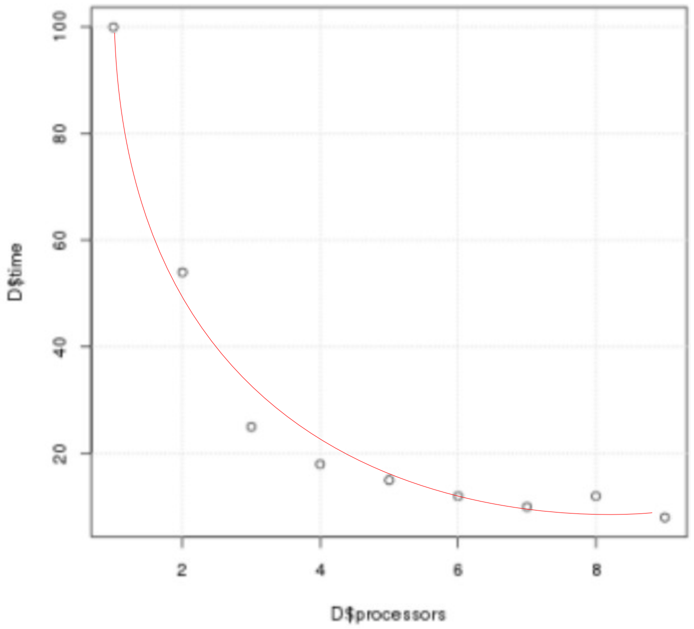
```
> D = read.table("regr3.in",header=TRUE)
> lr.out = lm(1/D$time~D$processors)
> summary(lr.out)

Call:
lm(formula = 1/D$time ~ D$processors)

Residuals:
      Min        1Q    Median        3Q       Max
-0.021490 -0.001231  0.002029  0.005251  0.008547

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.002140   0.007123   -0.30    0.773
D$processors  0.013370   0.001266   10.56 1.49e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.009805 on 7 degrees of freedom
Multiple R-squared: 0.941,    Adjusted R-squared: 0.9325
F-statistic: 111.6 on 1 and 7 DF,  p-value: 1.49e-05
```

Linear Regression

Example

Develop a regression model for the relation between CPU-time and number of processors

Processors	CPU-time
1	100
2	54
3	25
4	18
5	15
6	12
7	10
8	12
9	8



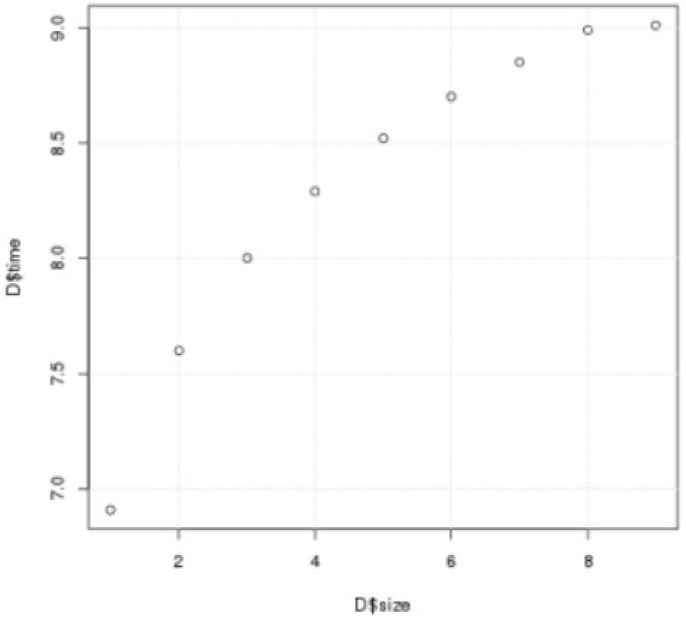
$$y = (-0.002+0.013\ x)^{-1}$$

Linear Regression

Example

Develop a regression model for the CPU-time of binary search given a list size

Size	CPU-time
1	6.91
2	7.60
3	8.00
4	8.29
5	8.52
6	8.70
7	8.85
8	8.99
9	9.01



Linear Regression

Example in R

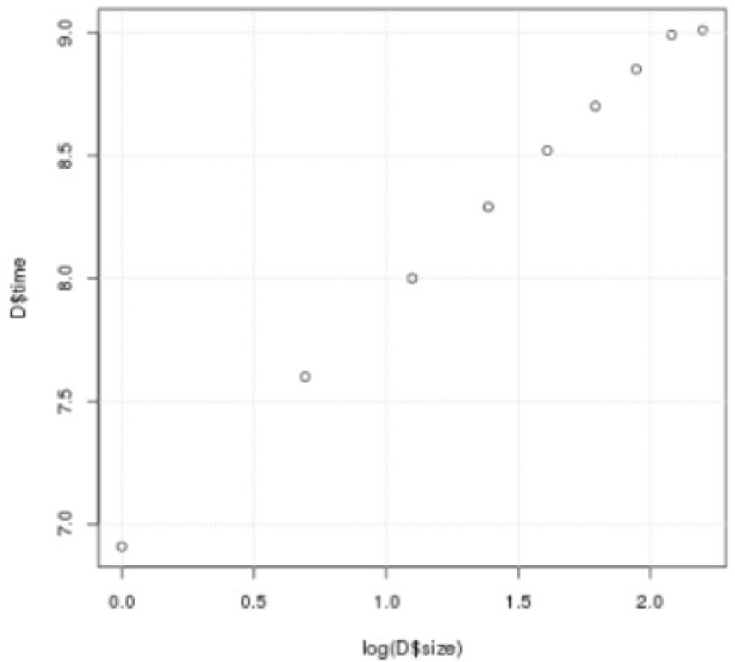
```
> D = read.table("regr4.in",header=TRUE
> lr.out = lm(D$time~D$size)
> summary(lr.out)

Call:
lm(formula = D$time ~ D$size)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43022 -0.06289  0.04178  0.17044  0.21578

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.09556    0.17547  40.437 1.47e-09 ***
D$size       0.24467    0.03118   7.846 0.000103 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2415 on 7 degrees of freedom
Multiple R-squared: 0.8979,   Adjusted R-squared: 0.8833
F-statistic: 61.56 on 1 and 7 DF,  p-value: 0.0001031
```

Linear Regression

Example

Logarithmic transformation: $y = a + b \log x$

log Size	CPU-time
0.00	6.91
0.69	7.60
1.10	8.00
1.39	8.29
1.61	8.52
1.79	8.70
1.95	8.85
2.08	8.99
2.20	9.01



Linear Regression

Example in R

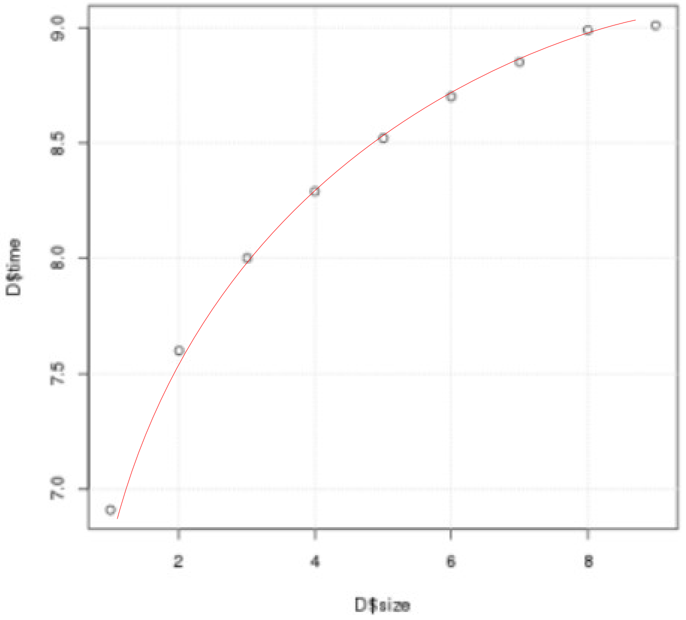
```
> D = read.table("regr4.in",header=TRUE)
> lr.out = lm(D$time~log(D$size))
> summary(lr.out)

Call:
lm(formula = D$time ~ log(D$size))

Residuals:
      Min         1Q    Median        3Q       Max
-0.069968 -0.002525  0.006602  0.017410  0.025730

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.92165    0.02391  289.53 1.55e-15 ***
log(D$size)  0.98229    0.01517   64.75 5.51e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03086 on 7 degrees of freedom
Multiple R-squared: 0.9983,   Adjusted R-squared: 0.9981
F-statistic:  4192 on 1 and 7 DF,  p-value: 5.508e-11
```

Linear Regression

Example

Develop a regression model for the CPU-time of binary search given a list size

Size	CPU-time
1	6.91
2	7.60
3	8.00
4	8.29
5	8.52
6	8.70
7	8.85
8	8.99
9	9.01



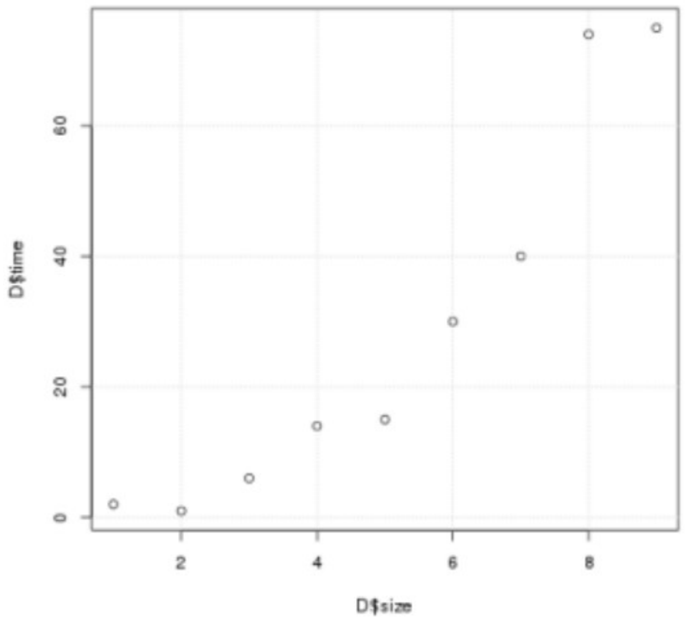
$y = 6.92 + 0.98 \log x$

Linear Regression

Example

Develop a regression model for the CPU-time of insertion sort

Size	CPU-time
1	2
2	1
3	6
4	14
5	15
6	30
7	40
8	74
9	75



Linear Regression

Example in R

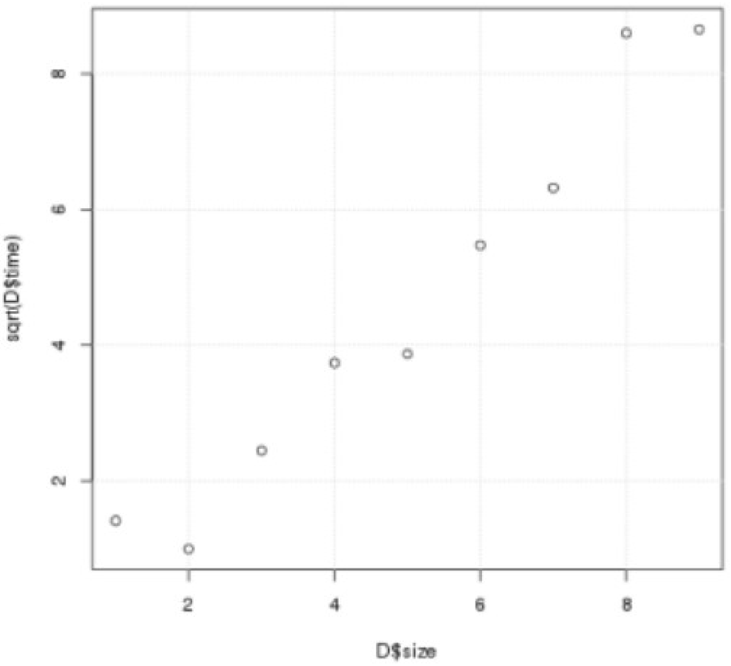
```
> D = read.table("regr2.in",header=TRUE)
> lr.out = lm(D$time~D$size)
> summary(lr.out)

Call:
lm(formula = D$time ~ D$size)

Residuals:
    Min       1Q   Median       3Q      Max
-13.556  -8.389  -2.722   6.778  15.694

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -21.028      7.881  -2.668 0.032087 *
D$size         9.917      1.401   7.081 0.000197 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.85 on 7 degrees of freedom
Multiple R-squared: 0.8775,   Adjusted R-squared:  0.86
F-statistic: 50.14 on 1 and 7 DF,  p-value: 0.000197
```

Linear Regression

Example

Square root transformation: $y^{1/2} = a + b\,x$

Size	CPU-time
1	1.00
2	1.41
3	2.45
4	3.74
5	3.87
6	5.48
7	6.32
8	8.60
9	8.66



Linear Regression

Example in R

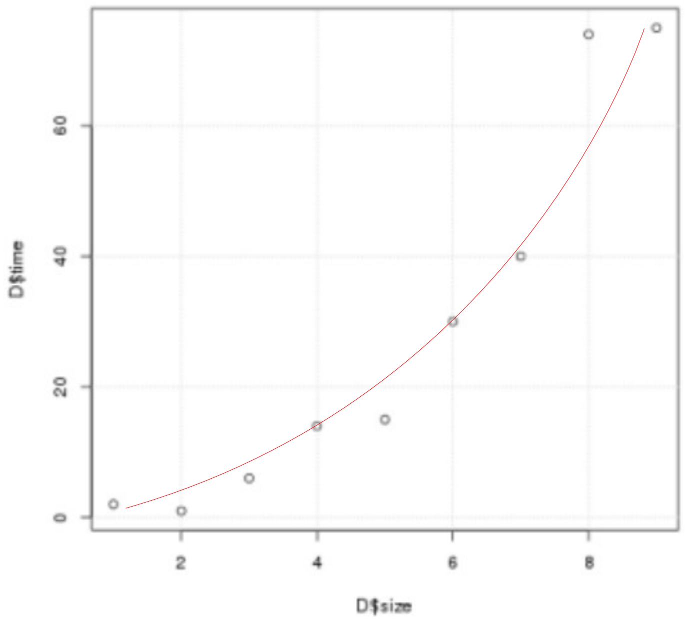
```
> D = read.table("regr2.in",header=TRUE)
> lr.out = lm(sqrt(D$time)~D$size)
> summary(lr.out)

Call:
lm(formula = sqrt(D$time) ~ D$size)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7429 -0.3339 -0.1238  0.1471  0.9226

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.49055    0.44817  -1.095     0.31
D$size       1.02128    0.07964  12.823 4.07e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6169 on 7 degrees of freedom
Multiple R-squared: 0.9592,   Adjusted R-squared: 0.9533
F-statistic: 164.4 on 1 and 7 DF,  p-value: 4.068e-06
```

Linear Regression

Example

Develop a regression model for the CPU-time of insertion sort

Size	CPU-time
1	2
2	1
3	6
4	14
5	15
6	30
7	40
8	74
9	75



$$y = (-0.49 + 1.02\,x)^2$$

Linear Regression

Recap:

- Linear regression model assumes a linear relationship between the input variable and the output variable.
- Multiple linear regression model deals with more than one input variable
- Coefficient of determination is the fraction of total variation that is provided by the linear model
- The assumptions of linear regression need to be met in order to ensure that the model can be used for inference (e.g prediction).
- Transformations can be applied in order to model polynomial, exponential or inverse relationships, but some care must be taken in the interpretation of the resulting model.

Linear Regression

References:

- D.J.Lilja, *Measuring computer performance*, Cambridge University Press, 2002 (see chapter 8)
- C.C. McGeoch, *A Guide to Experimental Algorithmics*, Cambrigde University Press, 2012 (see chapter 7)
- J. Faraway, Practical regression and ANOVA in R, chapter 8.
- D. Bruce , F. X. Schumacher, *Forest Mensuration*, Botanical Gazette, 1935
- J.W. Tukey, *Exploratory Data Anaysis*, Addison Wesley, 1977
- F. Mosteller and J.W. Tukey, Data Anaysis and Regression: A Second Course in Statistics. Addison Wesley, 1977.