In order to investigate the dependence of the maximum heart rate of a person from the age, the maximum heart rate and the age of 15 people of different ages are observed. The results are as follows:

```
Age <- c(18, 23, 25, 35, 65, 54, 34, 56, 72, 19, 23, 42, 18, 39, 37)
MaxRate <- c(202, 186, 187, 180, 156, 169, 174, 172, 153, 199, 193, 174, 198, 183, 178)
```

a. Build the simple linear regression model.
b. Estimate the coefficients and plot the regression line on the figure with bivariate distribution of the data.
c. Determine the expected maximum heart rate for any of these persons.
d. Determine the expected maximum heart rate for persons at age 30, 40, 50.
e. Determine the errors(residuals).
f. Determine the mean square error of the model and the residual standard error.
g. Compute the coefficient of deteremination.
h. Check if $\mathbb{E}\varepsilon$
i. Check if the errors are normal.

We can use the following functions: **lm** - linear model **plot** - plot the data **abline** - plot the regression line **simple.lm** - makes everithing required here.

a. The simple linear regression model is $Y = \hat{Y} + \varepsilon = \beta_0 + \beta_1 X + \varepsilon$.
   X is age; Y is maximum heart rate.

b.
```
plot(Age, MaxRate)
abline(lm(MaxRate~Age))


lm(MaxRate~Age)
Call:
lm(formula = MaxRate ~ Age)

Coefficients:
(Intercept)        Age
  210.0485      -0.7977
```

Then, $\beta_0 = 210.0485$, $\beta_1 = -0.7977$. The model is $Y = 210.0485 - 0.7977X + \varepsilon$

```
lmRes=simple.lm(Age, MaxRate)

attributes(lmRes)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
 [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
[11] "terms"         "model"

$class
[1] "lm"

coef(lmRes)
(Intercept)           x
210.0484584  -0.7977266
```

```
b1 = sum((Age - mean(Age)) * (MaxRate - mean(MaxRate))) / sum((Age - mean(Age))^2); b1
[1] -0.7977266
b0 = mean(MaxRate) - b1 * mean(Age); b0
[1] 210.0485
```

```
b1 <- cov(Age, MaxRate) / var(Age); b1
[1] -0.7977266
b0 <- mean(MaxRate) - b1 * mean(Age); b0
[1] 210.0485
```

c. Let us now determine the **expected** maximum heart rate for any of these persons.

```
predict(lmRes)
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15
195.6894 195.6894 194.8917 191.7007 191.7007 190.1053 182.9258 182.1280 180.5326 178.9371 176.5439 166.9712
165.3758 158.1962 152.6121
```

OR

```
yhat=b0+b1*Age; yhat
 [1] 195.6894 191.7007 190.1053 182.1280 158.1962 166.9712 182.9258 165.3758 152.6121
194.8917 191.7007 176.5439 195.6894 178.9371 180.5326
```

d. Determine the expected maximum heart rate for persons at age 30, 40, 50.

```
yhat30 = b0 + b1 * 30; yhat30
[1] 186.1167
yhat40 = b0 + b1 * 40; yhat40
[1] 178.1394
yhat50 = b0 + b1 * 50; yhat50
[1] 170.1621
```

e. We can find the errors (residuals): ε_i

```
resid(lmRes)
         1          2          3          4          5          6          7          8          9         10         11         12         13         14         15
 6.3106197  2.3106197  4.1083463 -5.7007474  1.2992526 -3.1052943 -8.9257552 -2.1280287 -2.5325755  4.0628776
-2.5439427  2.0287761  6.6242292 -2.1962317  0.3878543
```

```
lmRes[["residuals"]]
```

```
lmRes$residuals
```

```
e=MaxRate-yhat
```

```
summary(resid(lmResult))
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-8.9258 -2.5383  0.3879  0.0000  3.1867  6.6242
```

f. It is time to determine the mean square error of the model

```
SSE <- sum(e^2); SSE
[1] 272.4312
n <- length(MaxRate)
MSE <- SSE / (n - 2); MSE     # (mean square error)
[1] 20.95625
```

```
s <- sqrt(MSE); s       # (Residual Standart error)
[1] 4.577799
```

g. Via the function summary we can estimate also the **coefficient of determination**

```
Rsquare <- 1 - MSE/var(MaxRate); Rsquare
[1] 0.9021041
```

Multiple, R-squared: 0.9091. It does not takes into account that the denominators of the estimators

```
Rsq<-1 - SSE/sum((MaxRate - mean(MaxRate))^2); Rsq
[1] 0.9090967
```

или

```
Rsquare <- cov(Age, MaxRate)^2/(var(Age)*var(MaxRate)); Rsquare
[1] 0.9090967
```

или

```
Rsquare <- cor(Age, MaxRate)^2; Rsquare
[1] 0.9090967
```

h. In order to check $\mathbb{E}\varepsilon = 0$ we use t-test

H0: $\mathbb{E}\varepsilon = 0$

HA: $\mathbb{E}\varepsilon \neq 0$

```
t.test(e, mu = 0)
```

i. The next step is to test the assumptions of the model that the residuals are i.i.d. normally distributed $\varepsilon_i \in \mathcal{N}(0,\sigma_\varepsilon^2)$

```
qqnorm(e)
qqline(e)
```

```
shapiro.test(e)
```

```
qqplot.das(e)
```

```
plot(lmResult)
```

```
simple.lm(Age, MaxRate, show.residuals = TRUE)
```

Example 2.

Compute and plot 90% confidence intervals for $\mathbb{E}(Y|X = X_i)$ in the previous example.

Solution.

The function `predict` computes the estimators for $\mathbb{E}(Y|X = X_i)$ (the fitted values) and the corresponding confidence intervals.

```
> pr = predict(lmResult, interval = "confidence", l
> head(pr)
       fit      lwr      upr
1 195.6894 192.5083 198.8705
2 195.6894 192.5083 198.8705
3 194.8917 191.8028 197.9805
4 191.7007 188.9557 194.4458
5 191.7007 188.9557 194.4458
6 190.1053 187.5137 192.6969
```

level=0.90)

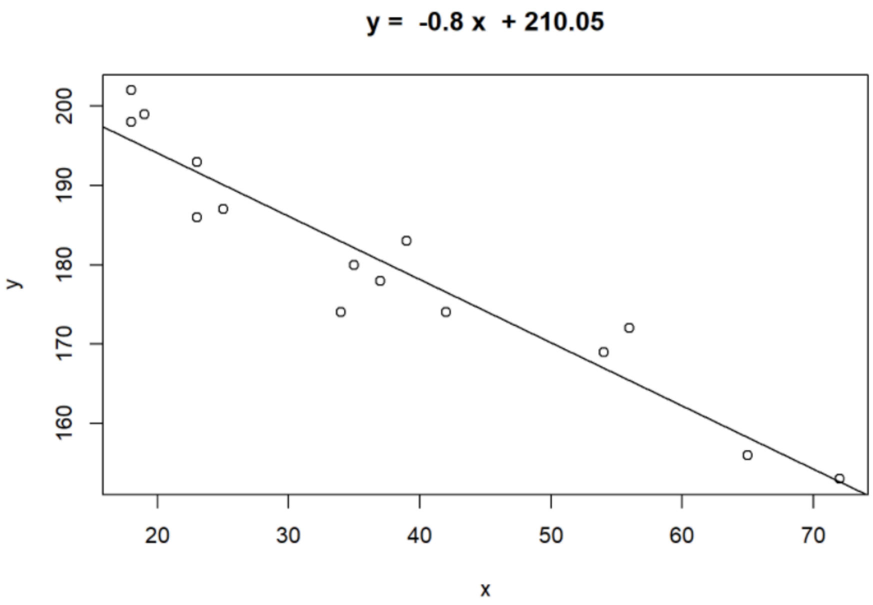
```
simple.lm(Age, MaxRate, show.ci = TRUE, conf.level = 0.90)
```

Example 3.

In the previous example determine 90% confidence intervals for the mean of the maximum heart rate for persons at age 30, 40, 50.

Solution.

```
> library(UsingR)
> lmResult <- simple.lm(Age, MaxRate)
```

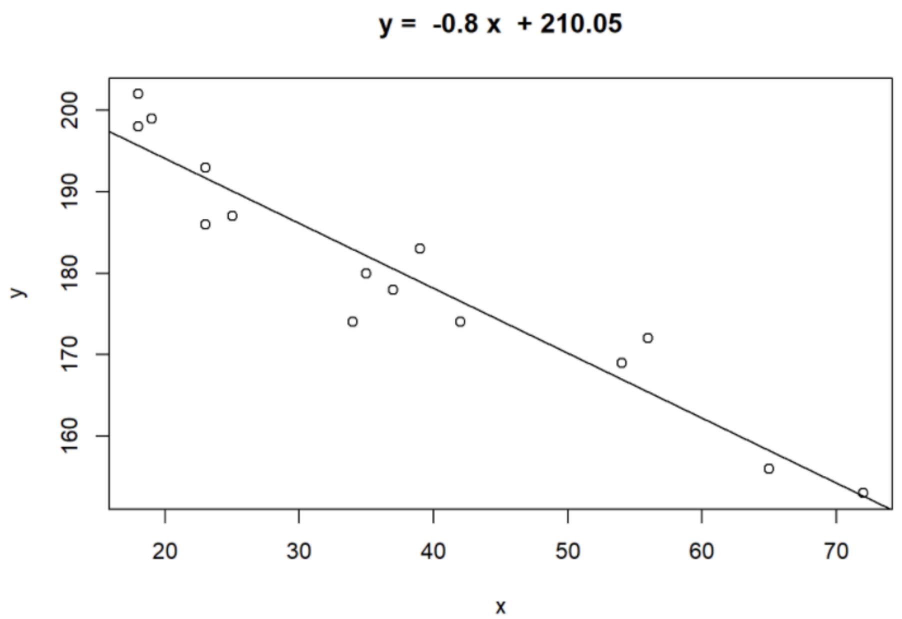



```
e<-resid(lmResult)
SSE <- sum(e^2); SSE
[1] 272.4312
n <- length(MaxRate)
MSE <- SSE / (n - 2); MSE
[1] 20.95625
Seps<-sqrt(MSE)
ci30<-yhat30 + c(-1,1)*Seps*sqrt(1/n+(30-mean(Age))/sum((Age-mean(Age))^2)); ci30
[1] 184.9500 187.2834
ci40<-yhat40 + c(-1,1)*Seps*sqrt(1/n+(40-mean(Age))/sum((Age-mean(Age))^2)); ci40
[1] 176.9519 179.3269
ci50<-yhat50 + c(-1,1)*Seps*sqrt(1/n+(50-mean(Age))/sum((Age-mean(Age))^2)); ci50
[1] 168.9542 171.3701
```

Example 4.

In the previous example determine 90% confidence intervals for the next observed maximum heart rate for persons at age 30, 40, 50.

```
> library(UsingR)
> lmResult <- simple.lm(Age, MaxRate)
```




```
e<-resid(lmResult)
SSE <- sum(e^2); SSE
[1] 272.4312
n <- length(MaxRate)
MSE <- SSE / (n - 2); MSE
[1] 20.95625
Seps<-sqrt(MSE)
ci30<-yhat30 + c(-1,1)*Seps*sqrt(1/n+1); ci30
[1] 181.3887 190.8446
ci40<-yhat40 + c(-1,1)*Seps*sqrt(1/n+1); ci40
[1] 173.4115 182.8673
ci50<-yhat50 + c(-1,1)*Seps*sqrt(1/n+1); ci50
[1] 165.4342 174.8901
```

When compare the results from this and the previous task we see that the confidence interval for unique values are wider than those for the corresponding means.

Statistical inference related with simple linear regression models

Confidence intervals for $\mathbb{E}\beta_1$ and hypothesis testing related with the slope β_1 of the regression line

Example 5

In the previous example

a. construct confidence interval for the parameter β_1.

b. Test the hypothesis that it is equal to -1.

c. Test the hypothesis that it is equal to 0.

a. We compute the required confidence interval via the following function which computes confidence intervals given the corresponding statistics $bhat$ computed from the data, the corresponding quantile t and the corresponding SE

```
> myCI = function(bhat, SE, t) {
+ bhat + c(-1,1)*SE*t
+ }
```

In this case first we have to compute

```
> library(UsingR)
> lmResult <- simple.lm(Age, MaxRate)
```

```
> e <- resid(lmResult)
> n<-length(e)
> beta1hat <- (coef(lmResult))[['x']]; beta1hat
[1] -0.7977266
> Seps <- sqrt(sum(e^2)/(n-2))
> SEbeta1 <- Seps / sqrt(sum((Age - mean(Age))^2));
[1] 0.06996281
> alpha<-0.05
> t <-qt(1-alpha/2, n - 2, lower.tail = TRUE)
> myCI(beta1hat, SEbeta1,t)
[1] -0.9488720 -0.6465811
```

As far as -1 is not in this confidence interval we can guess that the following H_0 will be rejected, however let us see.

b. We test

$H_0 : \beta_1 = -1$

$H_A : \beta_1 \neq -1$

```
> const <- -1
> temp <- abs(beta1hat-const)/SEbeta1; temp
[1] 2.891157
> pvalue<-2*pt(temp, n - 2, lower.tail = FALSE); pv
[1] 0.01262031
```

Confidence intervals for $\mathbb{E}\beta_0$ and hypothesis testing related with the intercept β_0 of the regression line on Oy.

Example 6

In the previous example

a. construct confidence interval for the parameter β_0.

b. Test the hypothesis that the regression line goes trough the coordinate origin.

c. Test the hypothesis that it is equal to 220.

a. In order to compute the required confidence interval we are going to use again our function $myCI$ In this case

```
> library(UsingR)
> lmResult <- simple.lm(Age, MaxRate)
```

As far as 0 is not in this confidence interval we can guess that the following H_0 will be rejected, however let us see.

b. We test

$H_0 : \beta_0 = 0$ which means that there is no intercept of Oy in the regression line.

$H_A : \beta_0 \neq 0$

```
> const <- 0
> temp <- abs(beta0hat-const)/SEbeta0; temp
[1] 73.26576
> pvalue<-2*pt(temp, n - 2, lower.tail = FALSE); pv
[1] 2.124074e-18
```

c. As far as 220 is outside the built confidence interval we can guess that we will reject the next H_0. Now let us automatically test for

$H_0 : \beta_0 = 220$, which means that there is no statistically significant difference between the intercept and 220.

$H_A : \beta_0 \neq 220$

```
> SEbeta0 <- Seps * sqrt(sum(Age^2) / (n * sum((Age
[1] 2.866939
> temp <- abs(beta0hat - 220) / SEbeta0; temp
[1] 3.471138
> pvalue<-2*pt(temp, n - 2, lower.tail = FALSE); pv
[1] 0.004136843
```

The $p - value = 0.004136843 < 0.05 = \alpha$, so we reject the value H_0. The difference between β_1 and 220 is statistically significant.

```
SEbeta0 <- Seps * sqrt(sum(Age^2) / (n * sum((Age - mean(Age))^2))); SEbeta0
[1] 2.866939
> temp <- abs(beta0hat - 220) / SEbeta0; temp
[1] 3.471138
> pvalue<-2*pt(temp, n - 2, lower.tail = FALSE); pvalue
[1] 0.004136843
```

Tests for adequacy

Tests for adequacy check if the independent variable X has no statistically significant influence on Y.

H_0 : The model is not adequate. The linear dependence between X and Y is not statistically significant. I.e. the slope $\beta_1 = 0$.

H_A : The model is adequate. The linear dependence between X and Y is statistically significant. I.e. the slope $\beta_1 \neq 0$.

As you can see for this model the test for adequacy is equivalent to the one for $H_0 : \beta_1 = 0$.

Example 7

In the previous example test the simple linear regression model for adequacy.

```
> library(UsingR)
> lmResult <- simple.lm(Age, MaxRate)
```

```
> summary(lmResult)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.9258 -2.5383  0.3879  3.1867  6.6242 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 210.04846    2.86694   73.27  < 2e-16 *
x            -0.79773    0.06996  -11.40 3.85e-08 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.

Residual standard error: 4.578 on 13 degrees of fre
Multiple R-squared:  0.9091,	Adjusted R-squared:
F-statistic:   130 on 1 and 13 DF,  p-value: 3.848e
```

Here $F - statistic: 130$ is the empirical value of $\frac{\frac{SS(\hat{Y})}{r}}{\frac{SSE}{n-r-1}}$. We use

Third way to make the same.

It is faster to use the function `anova`. Its names comes from **Analysis of Variances /Дисперсионния анализ/**

```
> anova(lmResult)
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          1 2724.50 2724.50  130.01 3.848e-08 ***
Residuals 13  272.43   20.96
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.
```

Here $F - statistic = 130.01$ is the empirical value of $\frac{\frac{SS(\hat{Y})}{r}}{\frac{SSE}{n-r-1}}$. We use the p-value of the F-statistics $p - value = 3.848 * 10^{-08} < 0.05 = \alpha$, therefore, we reject H_0. The model is adequate. The linear dependence between X and Y is statistically significant.