Laboratorio con R - 1

Metodi e Modelli per l'Inferenza Statistica - Ing. Matematica - a.a. 2023-24

Topics:

- Introduction to linear regression

- Analysis of linear regression components

- Parameters estimation

- Analysis of residuals

0. Required packages

```
library( car )
library( ellipse )
library( faraway )
library( leaps )
library(MASS)
library( GGally)
library(rgl)
# library( qpcR )
```

1. Linear regression and tests for coefficients significance.

1.a Upload `faraway` library and the dataset `savings`, an economic dataset on 50 different countries. These data are averages over 1960-1970 (to remove business cycle or other short-term fluctuations). The recorded variables are:

- **sr** is aggregate personal saving divided by disposable income (risparmio personale diviso per il reddito disponibile).
- **pop15** is the percentage population under 15.
- **pop75** is the percentage population over 75.
- **dpi** is per-capita disposable income in U.S. dollars (reddito pro-capite in dollari, al netto delle tasse).
- **ddpi** is the rate [percentage] of change in per capita disposable income (potere d'acquisto - indice economico aggregato, espresso in %).

Create a summary of the data. How many variables have missing data? Which are quantitative and which are qualitative?

Solution

```
# import data
data(savings)

# Dimensioni
dim(savings)
## [1] 50  5
```

We have 50 observations (50 countries) with 5 attributes each. To visualize the first 5:

```r
# Overview of the first rows
head(savings)
##              sr pop15 pop75     dpi ddpi
## Australia 11.43 29.35  2.87 2329.68 2.87
## Austria   12.07 23.32  4.41 1507.99 3.93
## Belgium   13.17 23.80  4.43 2108.47 3.82
## Bolivia    5.75 41.89  1.67  189.13 0.22
## Brazil    12.88 42.19  0.83  728.47 4.56
## Canada     8.79 31.72  2.85 2982.88 2.43
```

Look at the main statistics for each variable:

```r
# a brief description of each columns
summary(savings)
##        sr             pop15           pop75             dpi
##  Min.   : 0.600   Min.   :21.44   Min.   :0.560   Min.   :  88.94
##  1st Qu.: 6.970   1st Qu.:26.21   1st Qu.:1.125   1st Qu.: 288.21
##  Median :10.510   Median :32.58   Median :2.175   Median : 695.66
##  Mean   : 9.671   Mean   :35.09   Mean   :2.293   Mean   :1106.76
##  3rd Qu.:12.617   3rd Qu.:44.06   3rd Qu.:3.325   3rd Qu.:1795.62
##  Max.   :21.100   Max.   :47.64   Max.   :4.700   Max.   :4001.89
##       ddpi
##  Min.   : 0.220
##  1st Qu.: 2.002
##  Median : 3.000
##  Mean   : 3.758
##  3rd Qu.: 4.478
##  Max.   :16.710
```

If missing values were present, 'summary' function would have informed us. To check it directly:

```r
# observe that there are no missing values
print(sapply(savings,function(x) any(is.na(x))))
##    sr pop15 pop75   dpi  ddpi
## FALSE FALSE FALSE FALSE FALSE
```

Finally we get the data type of each column:

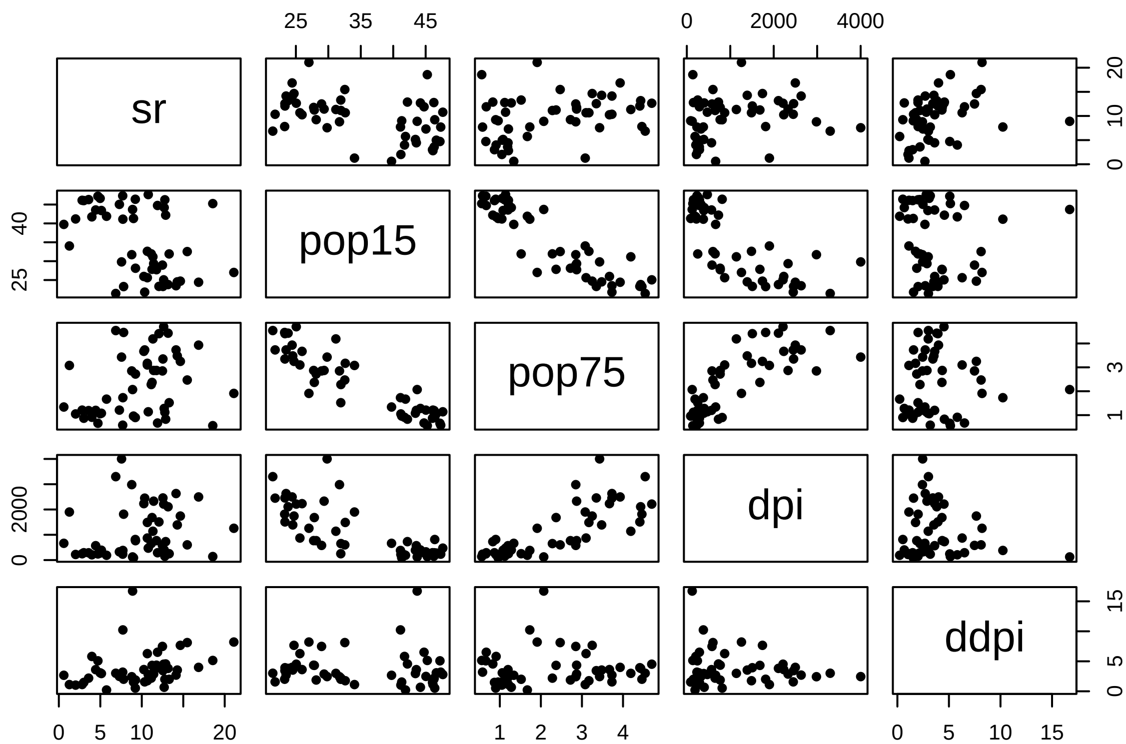
```r
# check the type of each column (integer, double, character, ...)
print(sapply(savings, typeof))
##       sr    pop15    pop75      dpi     ddpi
## "double" "double" "double" "double" "double"

# or
str(savings)
## 'data.frame':    50 obs. of  5 variables:
##  $ sr   : num  11.43 12.07 13.17 5.75 12.88 ...
##  $ pop15: num  29.4 23.3 23.8 41.9 42.2 ...
##  $ pop75: num  2.87 4.41 4.43 1.67 0.83 2.85 1.34 0.67 1.06 1.14 ...
##  $ dpi  : num  2330 1508 2108 189 728 ...
##  $ ddpi : num  2.87 3.93 3.82 0.22 4.56 2.43 2.67 6.51 3.08 2.8 ...
```

1.b Visualize the data and try to fit a complete linear model, in which `sr` is the outcome of interest. Explore the output of the model.

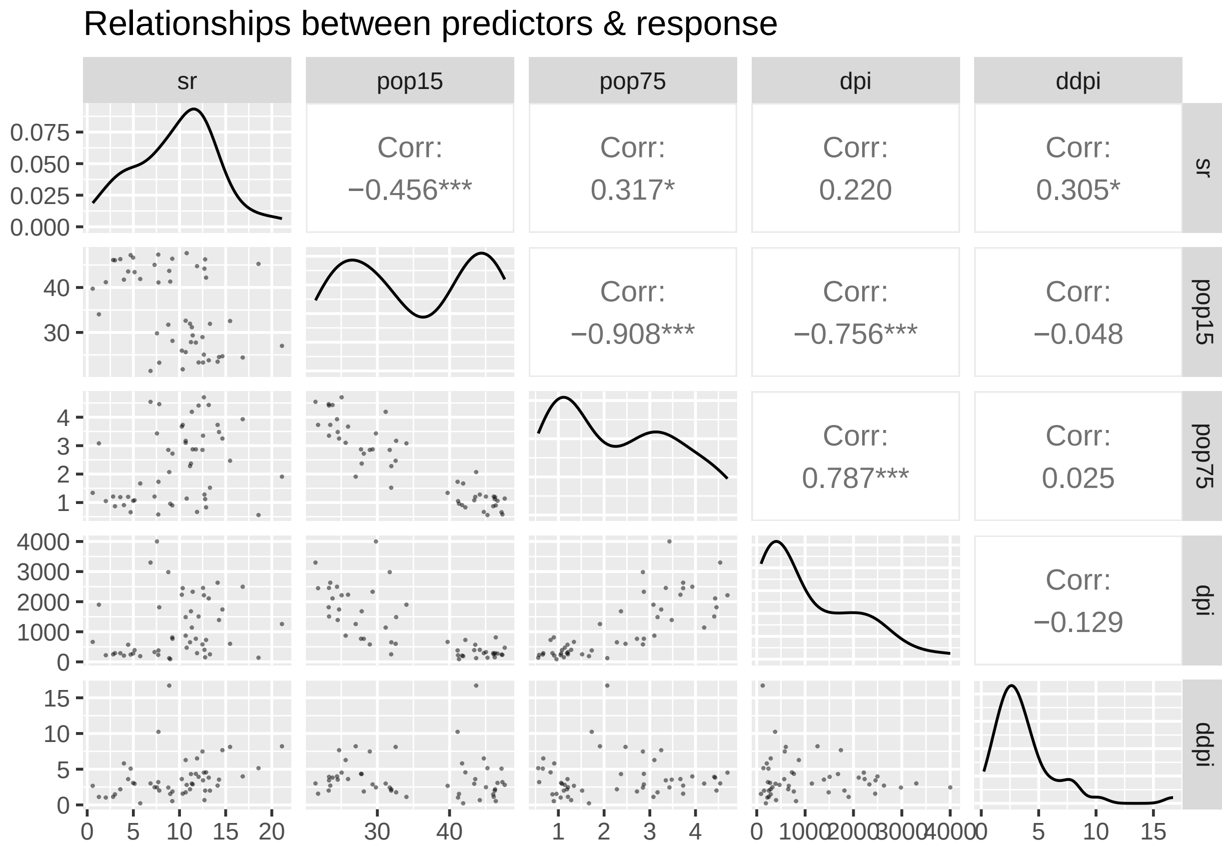
solution For visualizing the data, we can plot the pairs. It is useful also for making an idea about the relationship between the variables.

```r
pairs(savings[ , c('sr', 'pop15', 'pop75', 'dpi', 'ddpi')], pch = 16)
```



For a nicer visualization of these scatterplots, we can use the package 'GGally'. We can easily visualize the relationship of couple of variables, their sample correlation and their approximated density function.

```r
ggpairs(data = savings, title ="Relationships between predictors & response",
        lower = list(continuous=wrap("points", alpha = 0.5, size=0.1)))
```



Secondly, we can fit the complete linear model and look at a summary of the estimated coefficients.

```
g = lm( sr ~ pop15 + pop75 + dpi + ddpi, data = savings )
#g = lm( sr ~ ., savings )
summary( g )
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2422 -2.6857 -0.2488  2.4280  9.7509
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **
## pop75       -1.6914977  1.0835989  -1.561 0.125530
## dpi         -0.0003369  0.0009311  -0.362 0.719173
## ddpi         0.4096949  0.1961971   2.088 0.042471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

```r
gs = summary( g )
```

In order to measure the goodness of fit of the model, we have to look at R^2 and R^2_{adj}. They assume values between 0 and 1 and represent the percentage of explained variability by regressors, thus the more they are near to 1 the more the model explains well the dependent variable. Both of them are low in this case.

Third and last columns of the summary represent univariate statistics and p-values related to each estimated coefficient. They make us know the results of the test of estimated coefficients being equal from 0. In other words, they communicate us if the ICs of $\hat{\beta}_i$ contain or not the 0. Only 'pop15' and 'ddpi' seem to be significant in this model.

Through the F-statistic, we can investigate whether there is at least one covariate's parameter among β_1, β_2, β_3 and β_4 which is different from 0. Since the p-value of F-statistic is so small (0.0007904), the null hypothesis is rejected and there is at least one covariate's parameter that is different from 0.

```r
names(g) # this gives you the attributes of the linear model object
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

We can look through the model's attributes.

```r
g$call # linear model formula
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
g$coefficients #beta_hat
##   (Intercept)         pop15         pop75           dpi          ddpi
## 28.5660865407 -0.4611931471 -1.6914976767 -0.0003369019  0.4096949279
g$fitted.values # estimated 'sr' for each observation
##      Australia        Austria        Belgium        Bolivia         Brazil
##      10.566420      11.453614      10.951042       6.448319       9.327191
##         Canada          Chile          China       Colombia     Costa Rica
##       9.106892       8.842231       9.363964       6.431707       5.654922
##        Denmark        Ecuador        Finland         France        Germany
##      11.449761       5.995631      12.921086      10.164528      12.730699
##         Greece      Guatamala       Honduras        Iceland          India
##      13.786168       6.365284       6.989976       7.480582       8.491326
##        Ireland          Italy          Japan          Korea     Luxembourg
##       7.948869      12.353245      15.818514      10.086981      12.020807
##          Malta         Norway    Netherlands    New Zealand      Nicaragua
##      12.505090      11.121785      14.224454       8.384445       6.653603
##         Panama       Paraguay           Peru    Philippines       Portugal
##       7.734166       8.145759       6.160559       6.104992      13.258445
##   South Africa South Rhodesia          Spain         Sweden    Switzerland
##      10.656834      12.008566      12.441156      11.120283      11.643174
##         Turkey        Tunisia United Kingdom  United States      Venezuela
##       7.795682       5.627920      10.502413       8.671590       5.587482
##         Zambia        Jamaica        Uruguay          Libya       Malaysia
##       8.809086      10.738531      11.503827      11.719526       7.680869
```

We could also compute directly the fitted values of the dependent variable:

```r
X = model.matrix(g)
y_hat_man = X %*% g$coefficients #beta_hat
```

```r
g$residuals # residuals
##      Australia        Austria        Belgium        Bolivia         Brazil
##      0.8635798      0.6163860      2.2189579     -0.6983191      3.5528094
```

```
##          Canada          Chile          China       Colombia     Costa Rica
##      -0.3168924     -8.2422307      2.5360361     -1.4517071      5.1250782
##         Denmark        Ecuador        Finland         France        Germany
##       5.4002388     -2.4056313     -1.6810857      2.4754718     -0.1806993
##          Greece      Guatamala       Honduras        Iceland          India
##      -3.1161685     -3.3552838      0.7100245     -6.2105820      0.5086740
##         Ireland          Italy          Japan          Korea     Luxembourg
##       3.3911306      1.9267549      5.2814855     -6.1069814     -1.6708066
##           Malta         Norway    Netherlands    New Zealand      Nicaragua
##       2.9749098     -0.8717854      0.4255455      2.2855548      0.6463966
##          Panama       Paraguay           Peru    Philippines       Portugal
##      -3.2941656     -6.1257589      6.5394410      6.6750084     -0.7684447
##    South Africa South Rhodesia          Spain         Sweden    Switzerland
##       0.4831656      1.2914342     -0.6711565     -4.2602834      2.4868259
##          Turkey        Tunisia United Kingdom  United States      Venezuela
##      -2.6656824     -2.8179200     -2.6924128     -1.1115901      3.6325177
##          Zambia        Jamaica        Uruguay          Libya       Malaysia
##       9.7509138     -3.0185314     -2.2638273     -2.8295257     -2.9708690

g$rank # the numeric rank of the fitted linear model (number of covariates + 1)
## [1] 5
```

Calculate Variance-Covariance Matrix for a Fitted Model Object

```
#help( vcov )
vcov( g )
##               (Intercept)         pop15         pop75           dpi
## (Intercept) 54.088907156 -1.046928e+00 -6.4480864740 -1.135929e-03
## pop15       -1.046927609  2.092137e-02  0.1199574165  2.422953e-05
## pop75       -6.448086474  1.199574e-01  1.1741866426 -3.703298e-04
## dpi         -0.001135929  2.422953e-05 -0.0003703298  8.669606e-07
## ddpi        -0.271654582  2.907814e-03 -0.0116339234  4.667202e-05
##                      ddpi
## (Intercept) -2.716546e-01
## pop15        2.907814e-03
## pop75       -1.163392e-02
## dpi          4.667202e-05
## ddpi         3.849331e-02
```

1.c Try to compute F-test, manually.

Solution

Recall that the F-test says us if there is at least one estimated coeafficient significantly different from 0. Compute:

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

```
SS_tot = sum( ( savings$sr-mean( savings$sr ) )^2 )
```

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2$$

```
SS_res = sum( g$res^2 )
```

$$F = \frac{(SS_{tot} - SS_{res})/(p-1)}{SS_{res}/(n-p)}$$

```
p = g$rank # p = 5
n = dim(savings)[1] # n = 50

f_test = ( ( SS_tot - SS_res )/(p-1) )/( SS_res/(n-p) )

## p-value (right-hand area of f_test)
1 - pf( f_test, p - 1, n - p )
## [1] 0.0007903779
```

1.d Test the significance of the parameter β_1 (the parameter related to pop_15), manually.

solution

We want to test:

$$H_0 : \beta_1 = 0 \qquad vs \qquad H_1 : \beta_1 \neq 0$$

There are several ways to execute this test:

- **t-test**

We compute the test, whose output is shown in the R summary.

```
X = model.matrix( g )

sigma2 = (summary( g )$sigma)^2
#manually
sigma2 = sum( ( savings$sr - g$fitted.values )^2 ) / ( n -p )

se_beta_1 = summary( g )$coef[ 2, 2 ]
#manually
se_beta_1 = sqrt( sigma2 * diag( solve( t( X ) %*% X ) )[2] )

T.0 = abs( ( g$coefficients[ 2 ] - 0 )/ se_beta_1 )

2*( 1-pt( T.0, n-p ) )
##       pop15
## 0.002603019
```

- **F-test on nested model**

You fit a nested model (the complete model without the covariate in which you are interested) then you compute the residuals of the 2 models and execute the F-test.

REMARK it is NOT the F-test that you find in the summary!

$$F_0 = \frac{\frac{SS_{res}(complete\ model) - SS_{res}(nested\ model)}{df(complete\ model) - df(nested\ model)}}{\frac{SS_{res}(complete\ model)}{df(complete\ model)}} \sim F(df(complete\ model) - df(nested\ model), df(complete\ model))$$

```
g2 = lm( sr ~ pop75 + dpi + ddpi, data = savings )
summary( g2 )
##
## Call:
## lm(formula = sr ~ pop75 + dpi + ddpi, data = savings)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0577 -3.2144  0.1687  2.4260 10.0763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.4874944  1.4276619   3.844  0.00037 ***
## pop75       0.9528574  0.7637455   1.248  0.21849
## dpi         0.0001972  0.0010030   0.197  0.84499
## ddpi        0.4737951  0.2137272   2.217  0.03162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.164 on 46 degrees of freedom
## Multiple R-squared:  0.189,  Adjusted R-squared:  0.1361
## F-statistic: 3.573 on 3 and 46 DF,  p-value: 0.02093
SS_res_2 = sum( g2$residuals^2 )

f_test_2 = ( ( SS_res_2 - SS_res ) / 1 )/( SS_res / (n-p) )

1 - pf( f_test_2, 1, n-p )
## [1] 0.002603019
```

- **ANOVA between the two nested models** The analysis of variance between two nested models is based on the statistics F_0 that we computed before!

```
anova( g2, g )
## Analysis of Variance Table
##
## Model 1: sr ~ pop75 + dpi + ddpi
## Model 2: sr ~ pop15 + pop75 + dpi + ddpi
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     46 797.72
## 2     45 650.71  1    147.01 10.167 0.002603 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We notice that the result is the same in all the three methods. β_1 is signficant.

Homework

1.e Test the significance of all the regression parameters, separately.

1.f Test the regression parameter β_4 (the one related to 'ddpi') for this test:

$$H_0 : \beta_4 = 0.35 \qquad vs \qquad H_1 : \beta_4 > 0.35$$

2. Confidence Intervals and Regions

Confidence Intervals

2.a Compute the 95% confidence intervals for the regression parameter related to 'pop75'.

solution

The formula for the required confidence interval is:

$$IC_{(1-\alpha)}(\beta_2) = [\hat{\beta}_2 \pm t_{1-\alpha/2}(n-p) \cdot se(\hat{\beta}_2)],$$

where $\alpha = 5\%$ and $df = n - p = 45$.

```
alpha = 0.05
t_alpha2 = qt( 1-alpha/2, n-p )
beta_hat_pop75 = g$coefficients[3]
se_beta_hat_pop75 = summary( g )[[4]][3,2]

IC_pop75 = c( beta_hat_pop75 - t_alpha2 * se_beta_hat_pop75,
              beta_hat_pop75 + t_alpha2 * se_beta_hat_pop75 )
IC_pop75
##      pop75      pop75
## -3.8739780  0.4909826
```

We observe that $IC_{(1-\alpha)}(\beta_2)$ includes 0, so there is no evidence for rejectig $H_0 : \beta_2 = 0$, at the 5% level. Indeed, this parameter was not significant even in the previous section (p-value 12.5%).

```
summary(g)$coef[3,4]
## [1] 0.1255298
```

2.b Compute the 95% confidence intervals for the regression parameter related to 'ddpi'.

solution

```
alpha = 0.05
t_alpha2 = qt( 1-alpha/2, n-p )
beta_hat_ddpi = g$coefficients[5]
se_beta_hat_ddpi = summary( g )[[4]][5,2]

IC_ddpi = c( beta_hat_ddpi - t_alpha2 * se_beta_hat_ddpi,
             beta_hat_ddpi + t_alpha2 * se_beta_hat_ddpi )
IC_ddpi
##       ddpi       ddpi
## 0.01453363 0.80485623
```

In this case, we observe that $IC_{(1-\alpha)}(\beta_4)$ does NOT include 0, so there is evidence for rejecting $H_0 : \beta_4 = 0$, at the 5% level. However, the lower bound of the $IC_{(1-\alpha)}(\beta_4)$ is really close to 0. We can see from the output above that the p-value is 4.2% - lower than 5% - confirming this point.

```
summary(g)$coef[5,4]
## [1] 0.04247114
```

Notice that this confidence interval is pretty wide in the sense that the upper limit is about 80 times larger than the lower limit. This means that we are not really that confident about what the exact effect of growth on savings really is.

REMARK Confidence intervals often have a duality with two-sided hypothesis tests. A 95% confidence interval contains all the null hypotheses that would not be rejected at the 5% level.

Confidence Regions

2.c Build the joint 95% confidence region for parameters 'pop15' e 'pop75'. And add the value of (β_1, β_2) according to the null hypothesis.

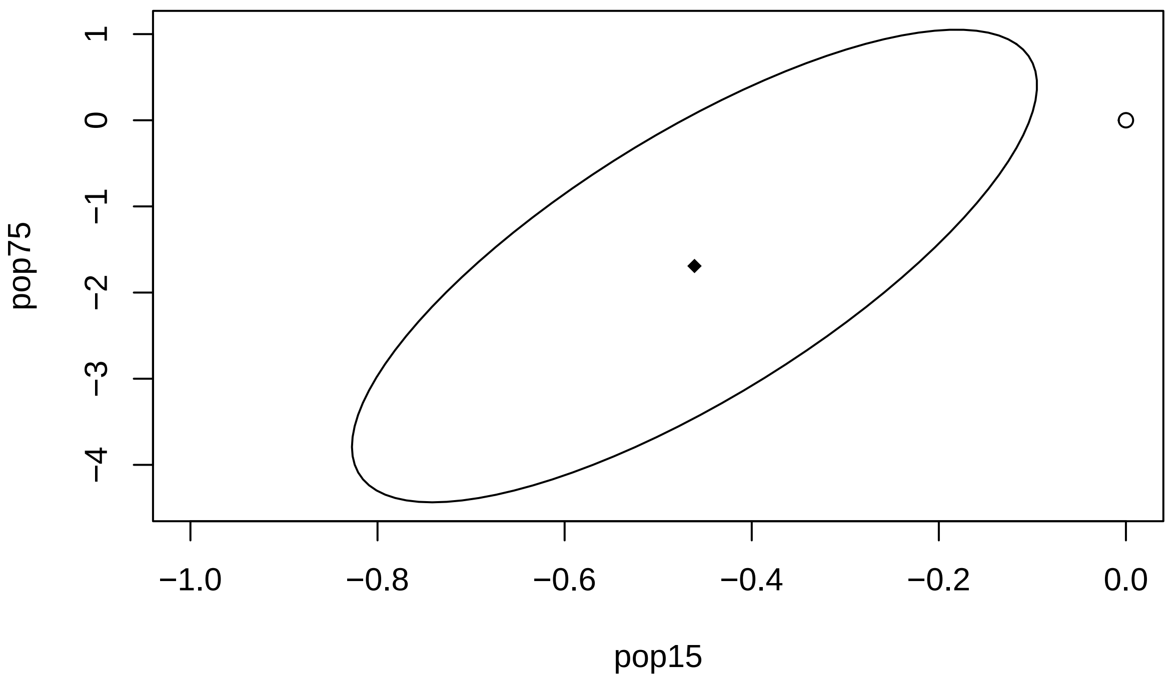
solution

```r
#help( ellipse )

plot( ellipse( g, c( 2, 3 ) ), type = "l", xlim = c( -1, 0 ) )

#add the origin and the point of the estimates:
#vettore che stiamo testando nell'hp nulla
points( 0, 0 )

# add also the cenre of the ellipse, that is, the estimated couple of coefficients
points( g$coef[ 2 ] , g$coef[ 3 ] , pch = 18 )
```



The filled dot is the center of the ellipse and represents the estimates of the 2 parameters $(\hat{\beta}_1, \hat{\beta}_2)$. Now, we are interested in this test:

$$H_0 : (\beta_1, \beta_2) = (0, 0) \qquad vs \qquad H_1 : (\beta_1, \beta_2) \neq (0, 0)$$

We observe that the empty dot (0,0) is not included in the Confidence Region (which is now an ellipse), so we reject H_0 at 5% level. In other words, we are saying that there is at least one parameter between β_1 and β_2 which is not equal to 0.

REMARK It is important to stress that this Confidence Region is different from the one obtained by the cartesian product of the two Confidence Intervals, $IC_{(1-\alpha)}(\beta_1)$ X $IC_{(1-\alpha)}(\beta_2)$. The cartesian product of the two Confidence Intervals is represented by the four dashed lines.

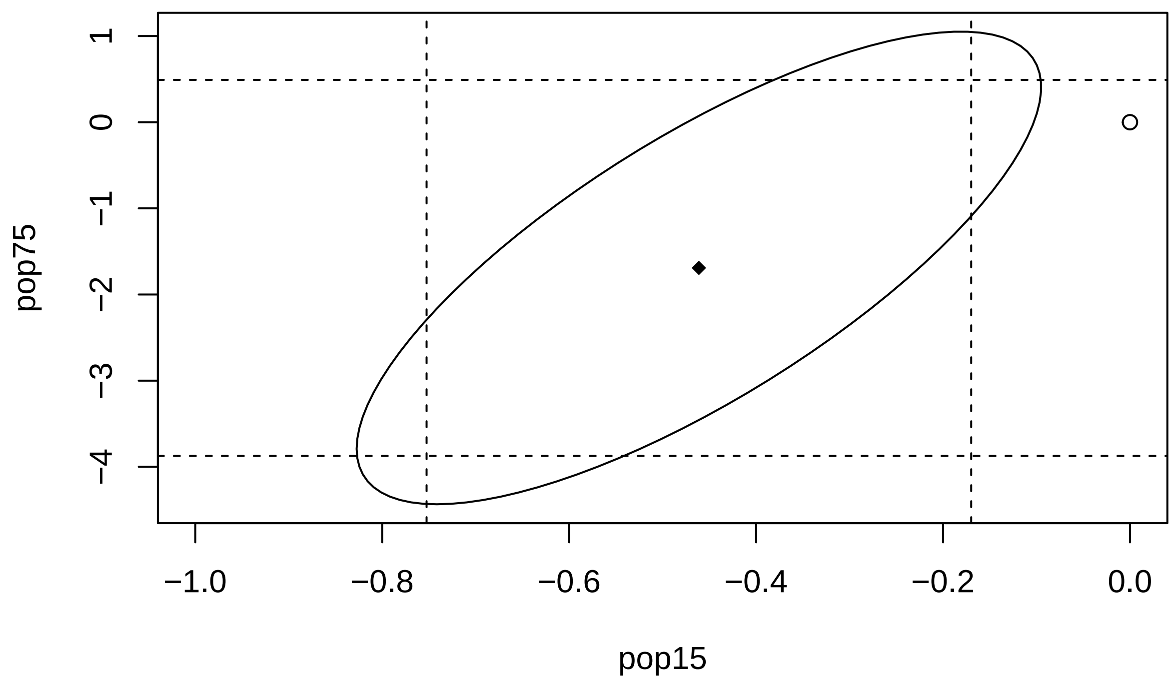
```r
beta_hat_pop15 = g$coefficients[2]
se_beta_hat_pop15 = summary( g )[[4]][2,2]
```

```
IC_pop15 = c( beta_hat_pop15 - t_alpha2 * se_beta_hat_pop15,
              beta_hat_pop15 + t_alpha2 * se_beta_hat_pop15 )
IC_pop15
##      pop15      pop15
## -0.7525175 -0.1698688

plot( ellipse( g, c( 2, 3 ) ), type = "l", xlim = c( -1, 0 ) )

points( 0, 0 )
points( g$coef[ 2 ] , g$coef[ 3 ] , pch = 18 )

#new part
abline( v = c( IC_pop15[1], IC_pop15[2] ), lty = 2 )
abline( h = c( IC_pop75[1], IC_pop75[2] ), lty = 2 )
```



REMARK The origin $(0,0)$ is included in the $IC_{(1-\alpha)}(\beta_2)$ and is NOT included in the $IC_{(1-\alpha)}(\beta_1)$, as expected from the previous point (we are expecting that β_1 is significantly different from 0, while β_2 is not.)

REMARK It can happen that you could reject according to one Confidence Region and accept according to the other Confidence Region. So which region should we choose?

- blue point: inside the the cartesian product of marginal ICs, outside the joint Confidence Region
- red point: outside the the cartesian product of marginal ICs, inside the joint Confidence Region

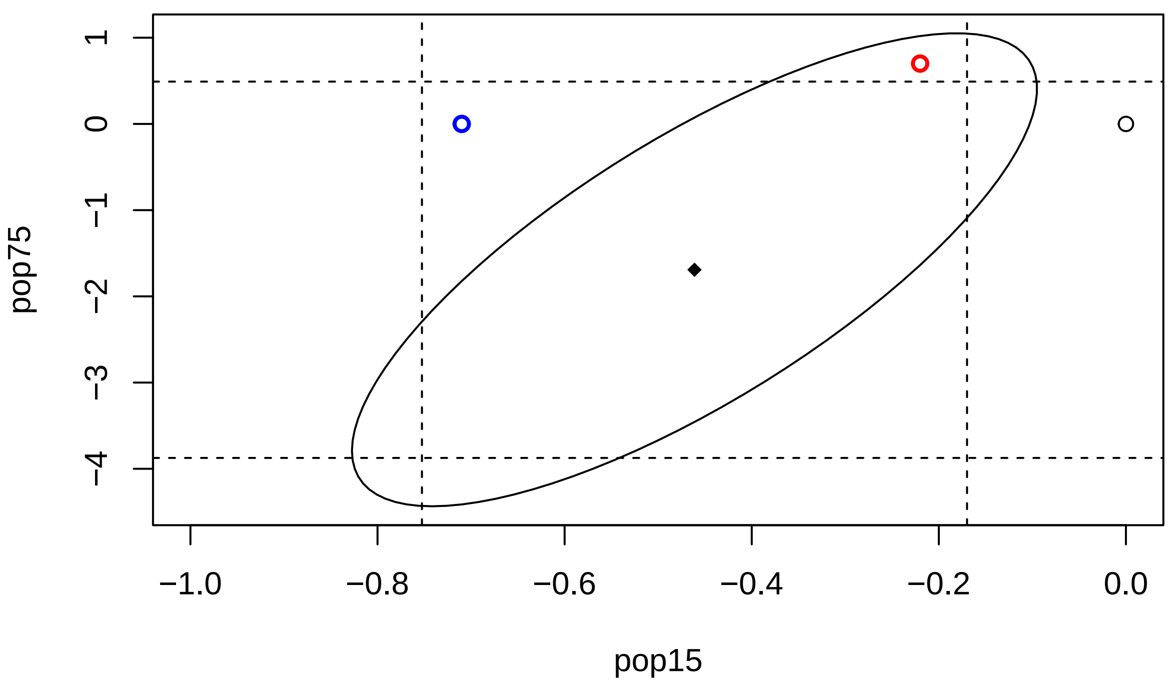
```
plot( ellipse( g, c( 2, 3 ) ), type = "l", xlim = c( -1, 0 ) )

points( 0, 0 )
```

```r
points( g$coef[ 2 ] , g$coef[ 3 ] , pch = 18 )

abline( v = c( IC_pop15[1], IC_pop15[2] ), lty = 2 )
abline( h = c( IC_pop75[1], IC_pop75[2] ), lty = 2 )

#new part
points( -0.22, 0.7, col = "red", lwd = 2 )
points( -0.71, 0, col = "blue", lwd = 2 )
```



We should alaways refer the joint Confidence Region (the elliptic one), because it is taking into account the correlation between the parameters. So we will accept the hypothesis represented by the red point and reject the hypothesis represented by the blue one.

In this case, correlation is near to -1. This means that the variables share a lot of variability and, consequently, of information

```r
cor( savings$pop15, savings$pop75 )
## [1] -0.9084787
```

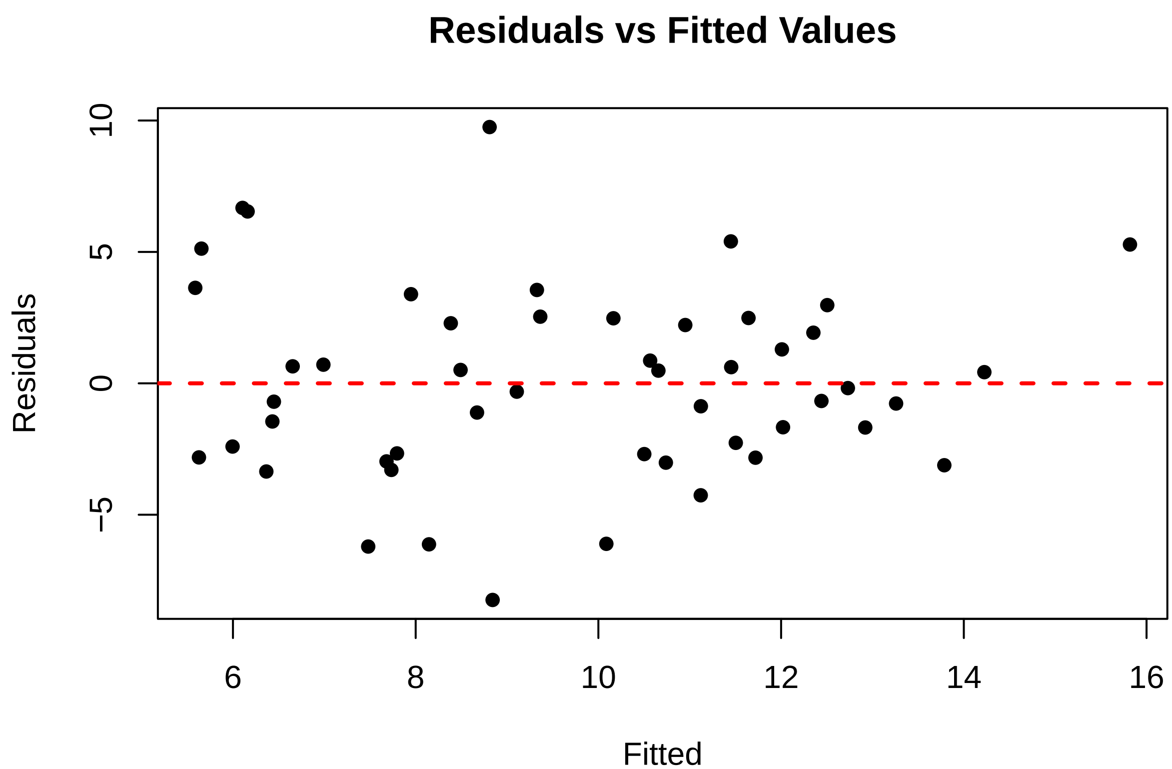
3. Hypotheses of the model

Homoscedasticity

3.a Plot residuals ($\hat{\varepsilon}$) vs fitted values ($\hat{y}$).

A sequence of random variables is homoscedastic if all its random variables have the same finite variance. Homoscedasticity is a fundamental assumptions in OLS.

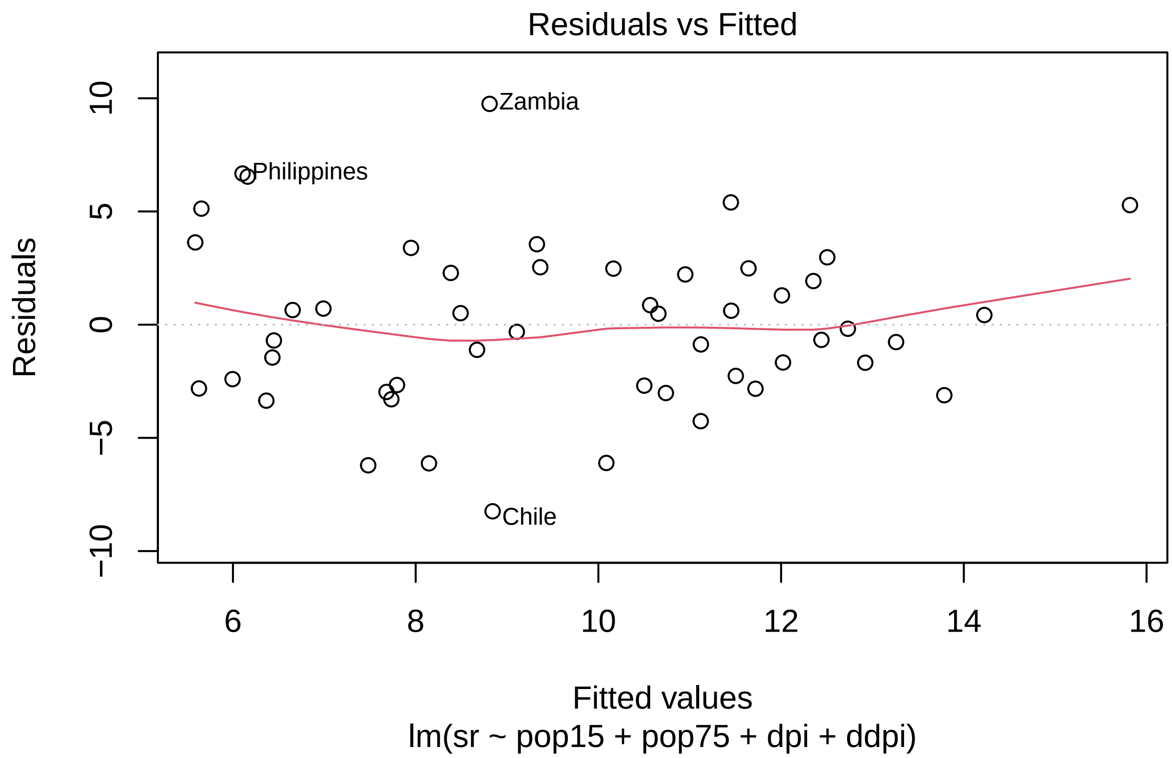
```r
plot( g$fit, g$res, xlab = "Fitted", ylab = "Residuals",
      main = "Residuals vs Fitted Values", pch = 16 )
abline( h = 0, lwd = 2, lty = 2, col = 'red' )  # variance seems uniform across the fitted values
```



The residual vs fitted value plot does not show any particular features that challenge the normality and constant variance assumptions of the residuals.

Alternatively, we can plot

```r
plot(g, which=1 )
```

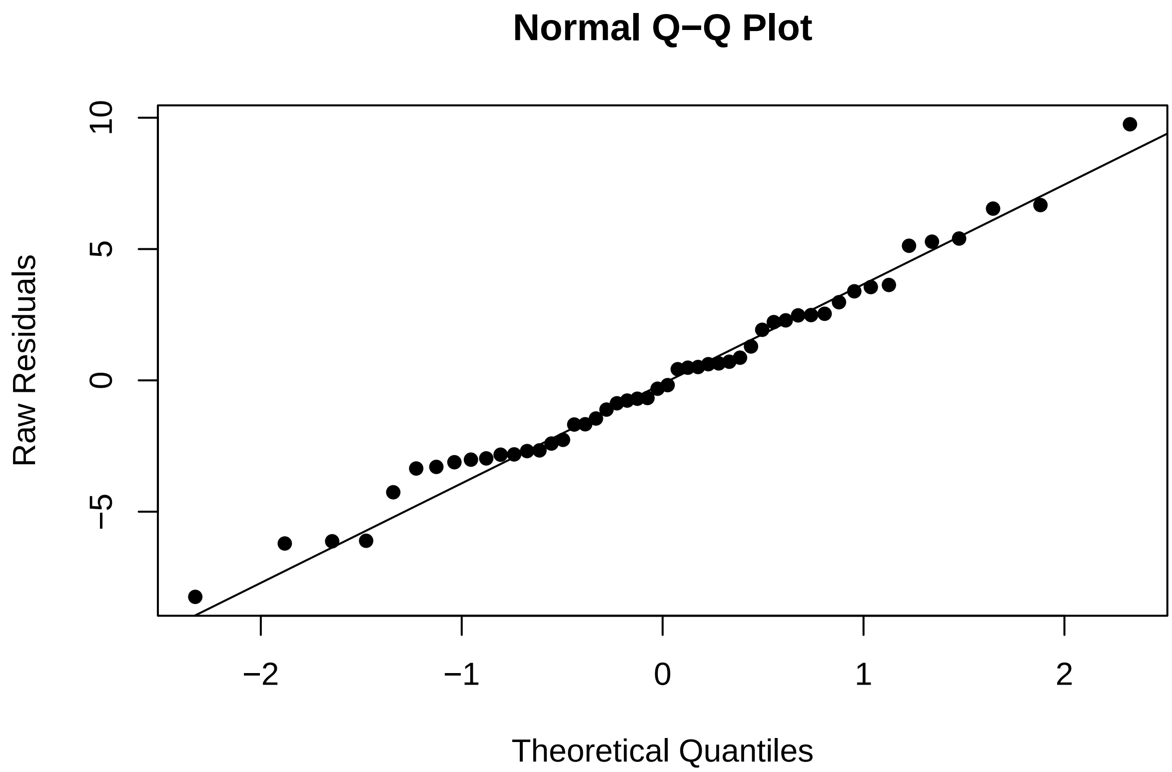


Normality

3.b Plot QQ plot and test the normality of the residuals with Shapiro-Wilks test.

Normality of data is another strong assumption of the OLS model.

```
# QQ plot
qqnorm( g$res, ylab = "Raw Residuals", pch = 16 )
qqline( g$res )
```



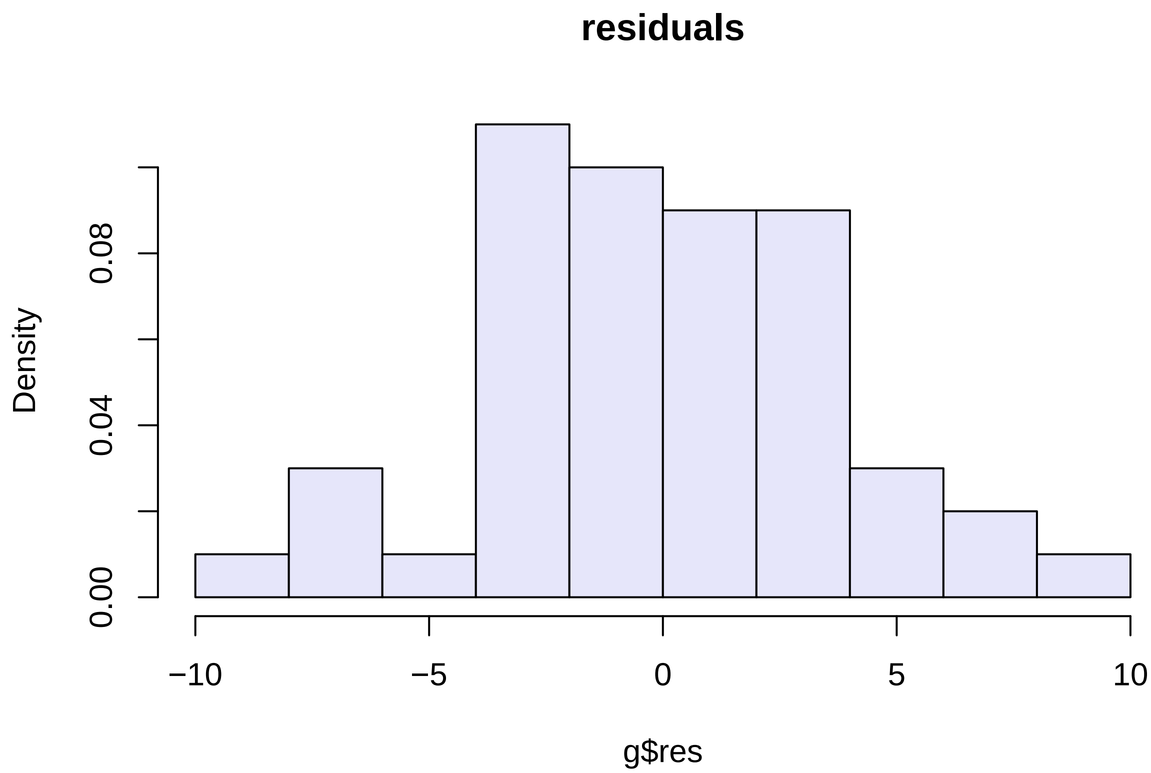

```
# linear trend, the hypothesis is verified

# Shapiro-Wilk normality test
shapiro.test( g$res )
##
##  Shapiro-Wilk normality test
##
## data:  g$res
## W = 0.98698, p-value = 0.8524
```

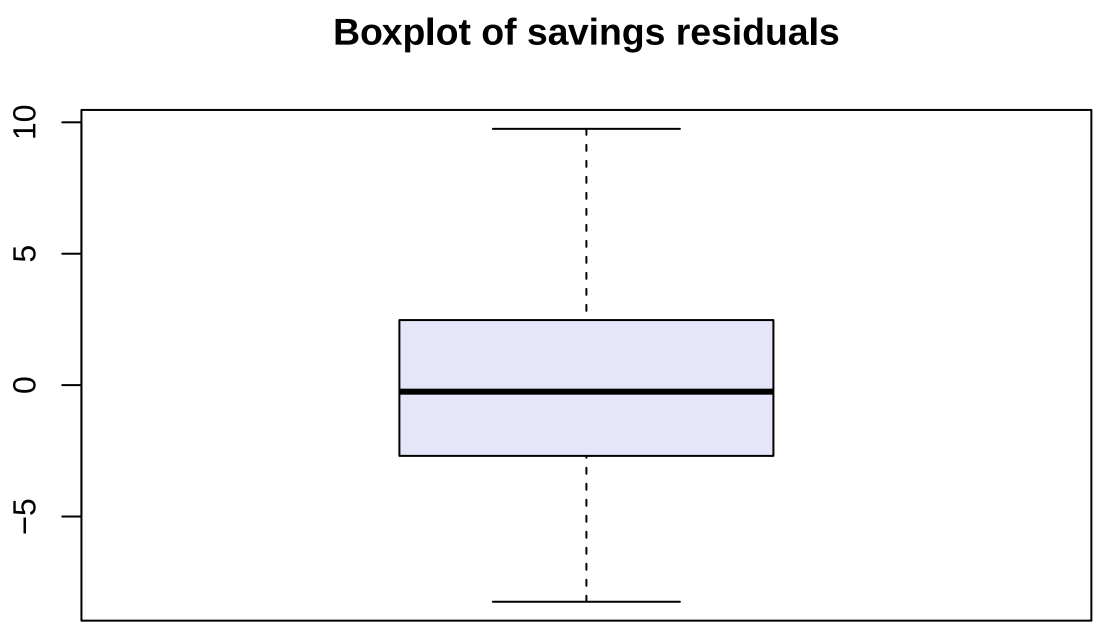
Since p-value is very high, I have no evidence to reject H_0, which is the gaussianity of the data.

Histogram of residuals and boxplot are useful tools to look at the shape of the residual distribution and to see if the tails of the distributions are souspiciously heavy (i.e., there are many outliers) respectively.

```
# other useful tools...
hist( g$res, 10, probability = TRUE, col = 'lavender', main = 'residuals'  )
```




```r
boxplot( g$res, main = "Boxplot of savings residuals", pch = 16, col = 'lavender' )
```



Treatment of not informative covariates

Covariates that have null estimated coefficients are likely to be removed from the model. We never remove more than one covariate at the same time. We start by removing the less significant covariate (the one that has the highest pvalue).

```
summary(g)
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2422 -2.6857 -0.2488  2.4280  9.7509
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **
## pop75       -1.6914977  1.0835989  -1.561 0.125530
## dpi         -0.0003369  0.0009311  -0.362 0.719173
## ddpi         0.4096949  0.1961971   2.088 0.042471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
```

```
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904

g2 = lm(sr ~ pop15 + pop75 + ddpi, data = savings)
summary(g2)
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2539 -2.6159 -0.3913  2.3344  9.7070
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.1247     7.1838   3.915 0.000297 ***
## pop15        -0.4518     0.1409  -3.206 0.002452 **
## pop75        -1.8354     0.9984  -1.838 0.072473 .
## ddpi          0.4278     0.1879   2.277 0.027478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.767 on 46 degrees of freedom
## Multiple R-squared:  0.3365, Adjusted R-squared:  0.2933
## F-statistic: 7.778 on 3 and 46 DF,  p-value: 0.0002646

g3 = lm(sr ~ pop15 +  ddpi, data = savings)
summary(g3)
##
## Call:
## lm(formula = sr ~ pop15 + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5831 -2.8632  0.0453  2.2273 10.4753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.59958    2.33439   6.682 2.48e-08 ***
## pop15       -0.21638    0.06033  -3.586 0.000796 ***
## ddpi         0.44283    0.19240   2.302 0.025837 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.861 on 47 degrees of freedom
## Multiple R-squared:  0.2878, Adjusted R-squared:  0.2575
## F-statistic: 9.496 on 2 and 47 DF,  p-value: 0.0003438


plot3d(savings$pop15, savings$ddpi, savings$sr, xlab='pop15', ylab='ddpi', zlab='sr')
planes3d( -0.21638, 0.44283,-1, 15.59958, col = 'blue', alpha = 0.6) # ax + by + cz +d
```