LAB 11

TOPICS:

- Linear models

```
library(MASS)
library(car)
library(rgl)
```

1.

```
### ------------------------------------------------------------------------
### Linear models
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Example 1: Multiple linear regression
### ------------------------------------------------------------------------
### Dataset cars: distance taken to stop [ft] as a function of velocity [mph]
### for some cars in the 1920s

help(cars)
```

```
## starting httpd help server ... done
```

```
head(cars)
```

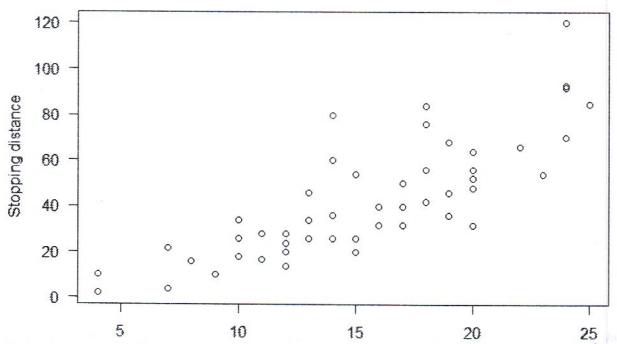
speed → speed of the cars
dist → stopping distance

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
dim(cars)
```

```
## [1] 50  2
```

```
plot(cars, xlab='Speed', ylab='Stopping distance', las=1)
```



response variable (Y) →

Speed → covariate (we want: X_1 = speed, X_2 = speed^2)

(*) we have a very small p-value for the F test (so we have stat. evidence to proceed with a model and to include at least one of the regressors) but if we look at the one-at-time tests on the single β_1 and β_2 we see high p-values. This is due to the fact that we have collinearity between covariates. Each one (β_1 and β_2) is redundant wrt the other.

```
n         <- dim(cars)[[1]]
distance  <- cars$dist
speed1    <- cars$speed
speed2    <- cars$speed^2

### Model:
### distance = beta_0 + beta_1 * speed + beta_2 * speed^2 + Eps
### (linear in the parameters!)

### Assumptions:
## 1) Parameter estimation: E(Eps) = 0  and  Var(Eps) = sigma^2
## 2) Inference:            Eps ~ N(0, sigma^2)
```

→ CI/CR/tests

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

```
### 1) Estimate of the parameters
### Assumptions: E(Eps) = 0  and  Var(Eps) = sigma^2
### ------------------------------------------------------------------------

help(lm)
fm <- lm(distance ~ speed1 + speed2)

summary(fm)
```

(β_0 is included by default)

```
##
## Call:
## lm(formula = distance ~ speed1 + speed2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.720  -9.184  -3.188   4.628  45.152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47014   14.81716   0.167    0.868
## speed1       0.91329    2.03422   0.449    0.656
## speed2       0.09996    0.06597   1.515    0.136
##
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

- symmetry?
- mean?

($\hat{\varepsilon}$)

$H_0: \beta_j = 0$
$H_1: \beta_j \neq 0$

those p-values refers to ONE-AT-THE-TIME TESTS (not simultaneous). If we want simultaneous we have to correct them

estimate of σ

$n - (r+1) = 50 - 3 = \dim(\mathcal{Y}^1(\mathcal{Z}))$ (*)

$H_0: (\beta_1 \wedge \beta_2) = 0$
$H_1: \exists j : \beta_j \neq 0$

this check β_1 and β_2 (if they're both zero or not, simultaneosly 0 (simultaneous check))

$\hat{y}$

```
# Note: collinearity
fitted(fm)      # y hat
```

evaluation of the regression line in correspondance to the training data :

```
##        1        2        3        4        5        6        7        8
##  7.722637  7.722637 13.761157 13.761157 16.173834 18.786430 21.598944 21.598944
##        9       10       11       12       13       14       15       16
## 21.598944 24.611377 24.611377 27.823729 27.823729 27.823729 27.823729 31.235999
##       17       18       19       20       21       22       23       24
## 31.235999 31.235999 31.235999 34.848188 34.848188 34.848188 34.848188 38.660295
##       25       26       27       28       29       30       31       32
## 38.660295 38.660295 42.672321 42.672321 46.884266 46.884266 46.884266 51.296129
##       33       34       35       36       37       38       39       40
## 51.296129 51.296129 51.296129 55.907911 55.907911 55.907911 60.719611 60.719611
##       41       42       43       44       45       46       47       48
## 60.719611 60.719611 60.719611 70.942768 76.354224 81.965599 81.965599 81.965599
##       49       50
## 81.965599 87.776892
```

we have 50 data ⇒ 50 $\hat{y}$

$\hat{\varepsilon}$

```
residuals(fm)     # eps hat
```

```
##           1           2           3           4           5           6
##  -5.7226371   2.2773629  -9.7611569   8.2388431  -0.1738340  -8.7864298
##           7           8           9          10          11          12
##  -3.5989441   4.4010559  12.4010559  -7.6113771   3.3886229 -13.8237287
##          13          14          15          16          17          18
##  -7.8237287  -3.8237287   0.1762713  -5.2359988   2.7640012   2.7640012
##          19          20          21          22          23          24
##  14.7640012  -8.8481876   1.1518124  25.1518124  45.1518124 -18.6602950
##          25          26          27          28          29          30
## -12.6602950  15.3397050 -10.6723209  -2.6723209 -14.8842655  -6.8842655
##          31          32          33          34          35          36
##   3.1157345  -9.2961287   4.7038713  24.7038713  32.7038713 -19.9079105
##          37          38          39          40          41          42
##  -9.9079105  12.0920895 -28.7196109 -12.7196109  -8.7196109  -4.7196109
##          43          44          45          46          47          48
##   3.2803891  -4.9427675 -22.3542237 -11.9655985  10.0344015  11.0344015
##          49          50
##  38.0344015  -2.7768919
```

$\hat{\beta}_j$

```
coefficients(fm)  # beta_i
```

```
## (Intercept)      speed1      speed2
##   2.4701378   0.9132876   0.0999593
```

$cov(\hat{\beta}_j)$

```
vcov(fm)          # cov(beta_i)
```

```
##             (Intercept)      speed1       speed2
## (Intercept) 219.5483705 -28.9523122  0.872858710
## speed1      -28.9523122   4.1380528 -0.131439753
## speed2        0.8728587  -0.1314398  0.004351805
```

```
fm$rank # order of the model [r+1]
```

```
## [1] 3
```

```
fm$df   # degrees of freedom of the residuals [n-(r+1)]
```

```
## [1] 47
```

h_{ii}

```
hatvalues(fm) # h_ii
```

(leverages : we can see the influence of the different data on the estimate)

```
##          1          2          3          4          5          6          7
## 0.28812937 0.28812937 0.09900328 0.09900328 0.06991940 0.05169124 0.04132501
##          8          9         10         11         12         13         14
## 0.04132501 0.04132501 0.03628040 0.03628040 0.03447059 0.03447059 0.03447059
##         15         16         17         18         19         20         21
## 0.03447059 0.03426225 0.03426225 0.03426225 0.03426225 0.03447551 0.03447551
##         22         23         24         25         26         27         28
## 0.03447551 0.03447551 0.03438403 0.03438403 0.03438403 0.03371490 0.03371490
##         29         30         31         32         33         34         35
## 0.03264873 0.03264873 0.03264873 0.03181959 0.03181959 0.03181959 0.03181959
##         36         37         38         39         40         41         42
## 0.03231507 0.03231507 0.03231507 0.03567621 0.03567621 0.03567621 0.03567621
##         43         44         45         46         47         48         49
## 0.03567621 0.05942708 0.08516635 0.12447031 0.12447031 0.12447031 0.12447031
##         50
## 0.18114746
```

```
rstandard(fm) # standardized residuals
```

(studentized residuals)

```
##           1           2           3           4           5           6
## -0.44692674  0.17785758 -0.67761087  0.57193320 -0.01187723 -0.59453611
##           7           8           9          10          11          12
## -0.24220331  0.29618418  0.83457167 -0.51089137  0.22745138 -0.92700723
##          13          14          15          16          17          18
## -0.52465245 -0.25641592  0.01182060 -0.35108365  0.18533152  0.18533152
##          19          20          21          22          23          24
##  0.98995429 -0.59335328  0.07723974  1.68666299  3.02784903 -1.25128708
##          25          26          27          28          29          30
## -0.84895032  1.02862118 -0.71539666 -0.17913343 -0.99718556 -0.46121793
##          31          32          33          34          35          36
##  0.20874160 -0.62253627  0.31500537  1.65435058  2.19008866 -1.33351949
##          37          38          39          40          41          42
## -0.66367546  0.80998139 -1.92711568 -0.85349909 -0.58509494 -0.31669079
##          43          44          45          46          47          48
##  0.22011751 -0.33582624 -1.54003007 -0.84263533  0.70663755  0.77705904
##          49          50
##  2.67843940 -0.20220729
```

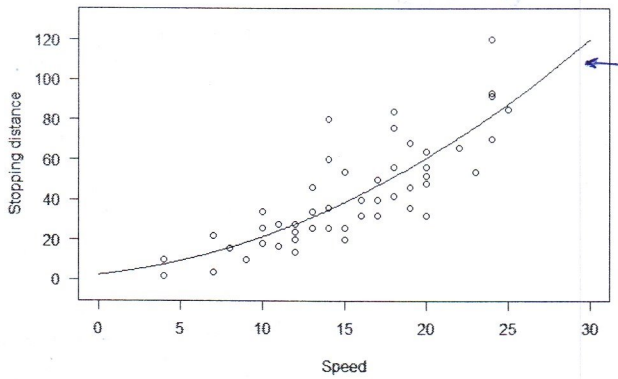
$\hat{\sigma}^2$

```
sum(residuals(fm)^2)/fm$df # estimate of sigma^2
```

```
## [1] 230.3131
```

```
plot(cars, xlab='Speed', ylab='Stopping distance', las=1, xlim=c(0,30), ylim=c(-5,130))
x <- seq(0,30,by=0.1)
b <- coef(fm)
lines(x, b[1]+b[2]*x+b[3]*x^2)
```

Graphical representation of the estimated model:



estimated model

it allows to perform tests on linear comb. of β's.

```
### Inference on the parameters
### Assuption: Eps ~ N(0, sigma^2)
### ------------------------------------------------------------------------
### Test (Fisher):
# H0: (beta1, beta2) == (0, 0) vs H1: (beta1, beta2) != (0, 0)
linearHypothesis(fm, rbind(c(0,1,0), c(0,0,1)), c(0,0))
```

(INFERENCE ON β's)

$$\begin{cases} H_0: (\beta_1, \beta_2) = (0,0) \\ H_1: \exists j : \beta_j \neq 0 \quad j = 1,2 \end{cases}$$

we have to provide the linear model (fm) and a matrix (matrix C of lectures) and the values of the vector "$(\beta_1, \beta_2) = [\cdot, \cdot]$"

```
## Linear hypothesis test
##
## Hypothesis:
## speed1 = 0
## speed2 = 0
##
## Model 1: restricted model
## Model 2: distance ~ speed1 + speed2
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     49 32539
## 2     47 10825  2     21714 47.141 5.852e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

we already know the value of this p-value (F#) since it's the same as the p-value of the F test of the regression model. Anyway, this test is for GENERIC LINEAR COMBINATIONS OF β's.

```
summary(fm)
```

```
##
## Call:
## lm(formula = distance ~ speed1 + speed2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.720  -9.184  -3.188   4.628  45.152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47014   14.81716   0.167    0.868
## speed1       0.91329    2.03422   0.449    0.656
## speed2       0.09996    0.06597   1.515    0.136
##
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

(F#)

```
p <- 2  # number of tested coefficients
r <- 2  # number of regressors

# Confidence region:
# center: point estimate
c(coefficients(fm)[2], coefficients(fm)[3])
```

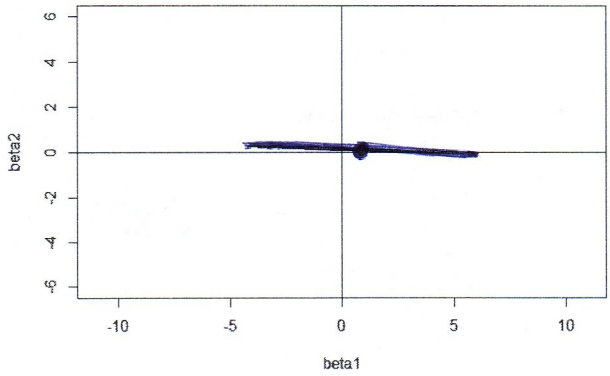
```
##    speed1    speed2
## 0.9132876 0.0999593
```

```
# Direction of the axes?
eigen(vcov(fm)[2:3,2:3])$vectors
```

```
##             [,1]        [,2]
## [1,] -0.99949587 -0.03174901
## [2,]  0.03174901 -0.99949587
```

```
plot(coefficients(fm)[2], coefficients(fm)[3], xlim = c(-6,6), ylim = c(-6,6), asp=1, xlab='beta1', ylab='beta2')
ellipse(coefficients(fm)[2:3], vcov(fm)[2:3,2:3], sqrt(p*qf(1-0.05,p,n-(r+1))))
abline(v=0)
abline(h=0)
```

Graphical representation of the CR:



it should be more an ellipse but we already know that the two covariates are colinear (it's not a line tho (zoom))

Note that $(\beta_1, \beta_2) = (0,0) \notin CR$ (coherent wrt the output of summary(fm))

Bonferroni manually

```
# Note: collinearity!

# Bonferroni intervals (level 95%)
Bf <- rbind(
  beta1=c(coefficients(fm)[2]-sqrt(vcov(fm)[2,2])*qt(1-0.05/(2*p), n-(r+1)),
          coefficients(fm)[2]+sqrt(vcov(fm)[2,2])*qt(1-0.05/(2*p), n-(r+1))),
  beta2=c(coefficients(fm)[3]-sqrt(vcov(fm)[3,3])*qt(1-0.05/(2*p), n-(r+1)),
          coefficients(fm)[3]+sqrt(vcov(fm)[3,3])*qt(1-0.05/(2*p), n-(r+1)))
)
Bf
```

if we compute one-at-the time tests we see that 0 falls inside the confidence intervals (for both)

```
##             speed1   speed1
## beta1 -3.79692941 5.623505
## beta2 -0.05278943 0.252708
```

those are the projections of the ellipse on the axes

Bonferroni "automatic"

```
# or (only for intervals on beta)
confint(fm, level= 1-0.05/p)[2:3,]  # Bonferroni correction!
```

confidence intervals for the beta coefficients (return one at-the-time confidence intervals for β)

```
##              1.25 %  98.75 %
## speed1 -3.79692941 5.623505
## speed2 -0.05278943 0.252708
```

```
# Note: confint() returns the confidence intervals one-at-a-time;
# to have a global level 95% we need to include a correction
```

An other possibility is to use linear hypothesis function to perform different types of tests on different types of linear combinations of the betas

```
### Test:
# H0: (beta0+beta2, beta1) == (0,0) vs H1: (beta0+beta2, beta1) != (0,0)
C <- rbind(c(1,0,1), c(0,1,0))
linearHypothesis(fm, C, c(0,0))
```

$$\begin{cases} H_0: (\beta_0+\beta_2, \beta_1) = (0,0) \\ H_1: (\beta_0+\beta_2, \beta_1) \neq (0,0) \end{cases} \rightarrow \begin{matrix} \beta_0 & \beta_1 & \beta_2 \end{matrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
## Linear hypothesis test
##
## Hypothesis:
## (Intercept)  + speed2 = 0
## speed1 = 0
##
## Model 1: restricted model
## Model 2: distance ~ speed1 + speed2
##
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     49 11973
## 2     47 10825  2    1148.4 2.4932 0.09352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Homework
# Build the associated confidence region
```

(INFERENCE ON Y)

```
### Confidence intervals for the mean
### & prediction (new obs)
### Assumption: Eps ~ N(0, sigma^2)
### ----------------------------------------------------------------------------
# Command predict()

Z0.new <- data.frame(speed1=10, speed2=10^2)
```

→ the new observation must be provided as data.frame containing the values of the new covariates and we have to set the names of the data.frame coherently wrt. the lm function

```
# Conf. int. for the mean
Conf <- predict(fm, Z0.new, interval='confidence', level=1-0.05)
Conf
```

! we can use the function "predict()" both for prediction interval for a new observation and for confidence interval for the mean

```
##        fit      lwr      upr
## 1 21.59894 15.39257 27.80532
```

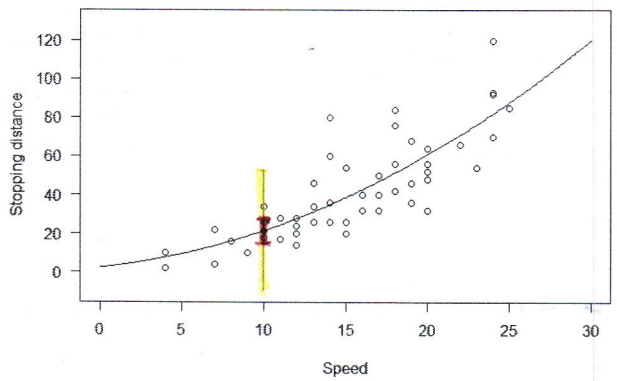
```
# Pred. int. for a new obs
Pred <- predict(fm, Z0.new, interval='prediction', level=1-0.05)
Pred
```

```
##        fit      lwr      upr
## 1 21.59894 -9.555818 52.75371
```

↳ pointwise estimate (evaluation of the regression line estimated)

```
plot(cars, xlab='Speed', ylab='Stopping distance', las=1, xlim=c(0,30), ylim=c(-10,130))
x <- seq(0,30,by=0.1)
b <- coef(fm)
lines(x, b[1]+b[2]*x+b[3]*x^2)
points(10,Conf[1], pch=19)
segments(10,Pred[2],10,Pred[3],col='gold', lwd=2)
segments(10,Conf[2],10,Conf[3],col='red', lwd=2)
points(10,Conf[2], pch='-', col='red', lwd=2)
points(10,Conf[3], pch='-', col='red', lwd=2)
points(10,Pred[2], pch='-', col='gold', lwd=2)
points(10,Pred[3], pch='-', col='gold', lwd=2)
```

Graphical representation of the intervals:



longer since it takes to account the variability around the mean

confidence
prediction

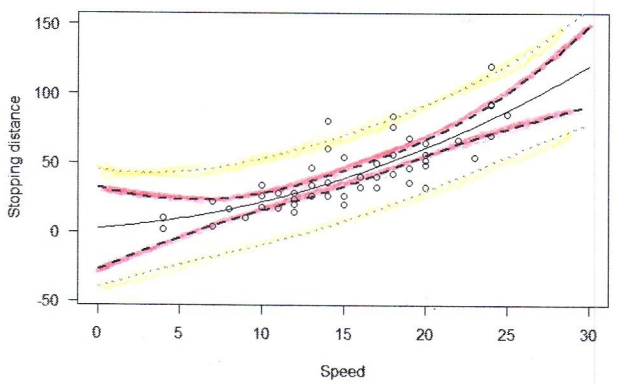
- new observations
- confidence
- prediction

```
# We can repeat these for values of speed between 0 and 30
# (point-wise intervals!)
Z0   <- data.frame(cbind(speed1=seq(0, 30, length=100),
                         speed2=seq(0, 30, length=100)^2))
Conf <- predict(fm, Z0, interval='confidence')
Pred <- predict(fm, Z0, interval='prediction')
```

we repeat the procedure for a finite grid of values (not just one point) in range of the covariates

plot

```
plot(cars, xlab='Speed', ylab='Stopping distance', las=1, xlim=c(0,30), ylim=c(-45,150))
lines(Z0[,1], Conf[,'fit'])
lines(Z0[,1], Conf[,'lwr'], lty=2, col='red', lwd=2)
lines(Z0[,1], Conf[,'upr'], lty=2, col='red', lwd=2)

lines(Z0[,1], Pred[,'lwr'], lty=3, col='gold', lwd=2)
lines(Z0[,1], Pred[,'upr'], lty=3, col='gold', lwd=2)
```



prediction
confidence

Note: they become longer when we move away from the baricenter

ATTENTION: THESE ARE **NOT** CONFIDENCE/PREDICTION BANDS since they're not computed simultaneously, these are just one-at-the-time confidence intervals (the level of confidence is maintained just in correspondence of a single value, we cannot say anything about the global coverage of the band)

```
### Verify assumptions
###

par(mfrow=c(2,2))
plot(fm)
```

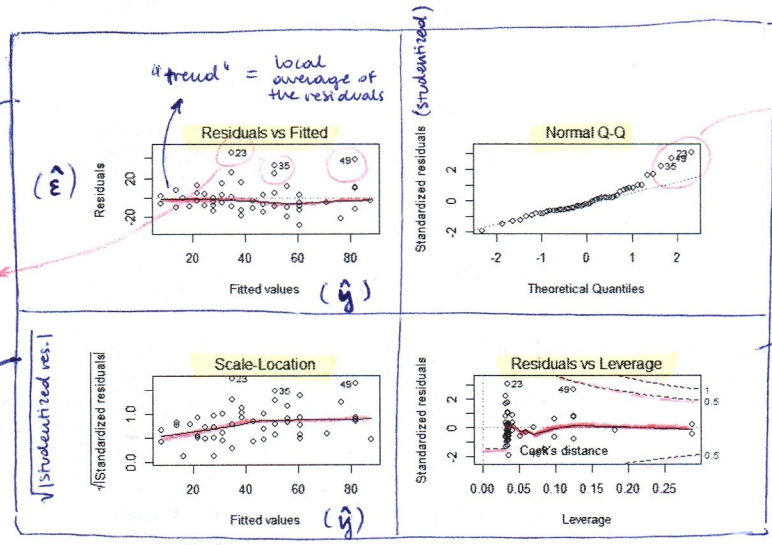
used for inference and estimations of parameters

we have assumed:

- gaussianity
- homoscedasticity (constant variance of the ε's, so σ^2 constant)

we would like to verify these assumptions looking at the residuals of the model (this is automatically done with:

plot (<output_of_lm>))



we want to see a cloud with no patterns around the zero (homogeneous around zero)

we can also see the outliers

"trend" = local average of the residuals

outliers (denoted by numbers (of the row from which they belong))

again, we want to see a homogeneous distribution with no pattern and no trends

we also have the iso-lines of the Cook distance (--): we have for instance a dashed lines that give us all the points with distance (Cook's distance) 0.5 and 1. We can identify the leverage points by looking at those points which lies outside these dashed lines.

```
shapiro.test(residuals(fm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fm)
## W = 0.93419, p-value = 0.007988
```

→ confirms what we saw in the Normal QQ plot (heavy tail of outliers)

```
### What happens if we change unit of measure?
### Exercise: Compare the results obtained from:
distance.m      <- cars$dist*0.3         # feet -> m
speed1.kmh      <- cars$speed*1.6        # miles/hour -> km per hour
speed2.kmh2     <- cars$speed^2 * 1.6^2
### Compare:
### - Coefficient estimates and corresponding covariance matrices
### - confidence intervals on the coefficients
### - fitted values
### - residuals
### - results of the tests
###   H0: (beta1, beta2) == (0,0) vs H1: (beta1, beta2) != (0,0)
###   H0: (beta0+beta2, beta1) == (0,0) vs H1: (beta0+beta2, beta1) != (0,0)
```

2.

```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Example 2: Anscombe
### ------------------------------------------------------------------------

### This example shows some cases for which examining data and
### residuals is crucial (don't look at R2 only!)

anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

```
# Four x-y datasets which have the same traditional statistical
# properties (mean, variance, correlation, regression line, etc.),
# yet are quite different.

attach(anscombe)

# dataset 1
lm1 <- lm(y1~ x1)
summary(lm1)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## 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: 0.00217
```

```
# dataset 2
lm2 <- lm(y2~ x2)
summary(lm2)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x2             0.500      0.118   4.239  0.00218 **
## ---
## 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:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
# dataset 3
lm3 <- lm(y3~ x3)
summary(lm3)
```

```
##
## Call:
## lm(formula = y3 ~ x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

```
# dataset 4
lm4 <- lm(y4~ x4)
summary(lm4)
```

```
##
## Call:
## lm(formula = y4 ~ x4)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x4            0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

for all the 4 dataset we have similar results: estimates of β_0, β_1, p-values, F-statistics, R^2, R^2_{adj}. it seems to be a good model for all the 4 datasets.

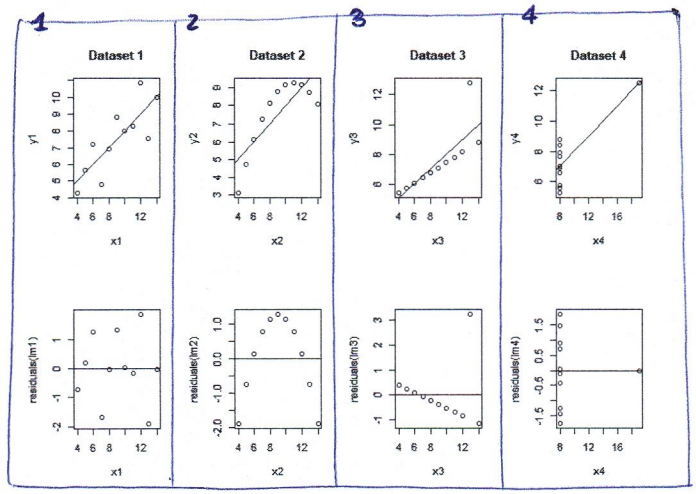
→ it makes sense to perform the regression

```
# same R^2, same coefficient estimate, same residual std error

x11(width=14, height=7)
par(mfcol=c(2,4))
plot(x1,y1, main='Dataset 1')
abline(lm1)

plot(x1,residuals(lm1))
abline(h=0)

plot(x2,y2, main='Dataset 2')
abline(lm2)

plot(x2,residuals(lm2))
abline(h=0)

plot(x3,y3, main='Dataset 3')
abline(lm3)

plot(x3,residuals(lm3))
abline(h=0)

plot(x4,y4, main='Dataset 4')
abline(lm4)

plot(x4,residuals(lm4))
abline(h=0)
```

But when we look at the residuals we see a very different behaviour.

Datasets:

Residuals:



1: coherent with what we expect

2: there is a parabolic trend, the assumption of homoschedasticity is not satisfied and the linear model is not appropriate

3: only one point changes the slope, the residuals capture the same behaviour

4: extreme case of the 3rd case, the line (regression) goes through the mean of the first set of point and the other point

⇓

it's not enough to look at R^2, we need to look at the residuals (plot) if we look at this we see that

1. is fine
2. need a quadratic model
3. problem with an outlier
4. problem with a leveraging point

! TAKE HOME MESSAGE:
we always need to criticize the model (diagnostic)

```
dev.off()
```

```
## png
##   2
```

```
detach(anscombe)
```

3.

```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Example 3: brain_weight
### ------------------------------------------------------------------------

data <- read.table('brain_weight.txt', header=T)

head(data)
```

```
##                             body brain
## Lesser short-tailed shrew 0.005  0.14
## Little brown bat          0.010  0.25
## Big brown bat             0.023  0.30
## Mouse                     0.023  0.40
## Musk shrew                0.048  0.33
## Star-nosed mole           0.060  1.00
```

we have two columns (body, brain) and for 62 animals we have the weights of the body and the brain

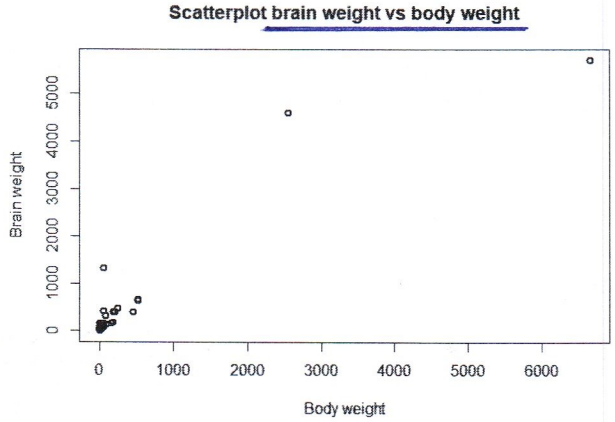
```
dim(data)
```

```
## [1] 62  2
```

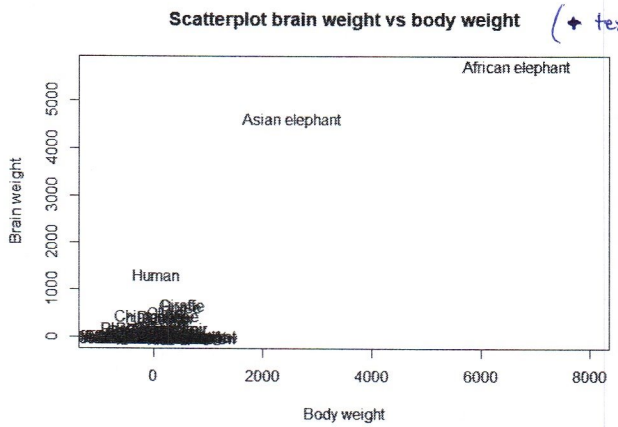
```
dimnames(data)
```

```
## [[1]]
##  [1] "Lesser short-tailed shrew" "Little brown bat"
##  [3] "Big brown bat"             "Mouse"
##  [5] "Musk shrew"                "Star-nosed mole"
##  [7] "E. American mole"          "Ground squirrel"
##  [9] "Tree shrew"                "Golden hamster"
## [11] "Mole"                      "Galago"
## [13] "Rat"                       "Chinchilla"
## [15] "Owl monkey"                "Desert hedgehog"
## [17] "Rock hyrax-a"              "European hedgehog"
## [19] "Tenrec"                    "Artic ground squirrel"
## [21] "African giant pouched rat" "Guinea pig"
## [23] "Mountain beaver"           "Slow loris"
## [25] "Genet"                     "Phalanger"
## [27] "N.A. opossum"              "Tree hyrax"
## [29] "Rabbit"                    "Echidna"
## [31] "Cat"                       "Artic fox"
## [33] "Water opossum"             "Nine-banded armadillo"
## [35] "Rock hyrax-b"              "Yellow-bellied marmot"
## [37] "Verbet"                    "Red fox"
## [39] "Raccoon"                   "Rhesus monkey"
## [41] "Potar monkey"              "Baboon"
## [43] "Roe deer"                  "Goat"
## [45] "Kangaroo"                  "Grey wolf"
## [47] "Chimpanzee"                "Sheep"
## [49] "Giant armadillo"           "Human"
## [51] "Grey seal"                 "Jaguar"
## [53] "Brazilian tapir"           "Donkey"
## [55] "Pig"                       "Gorilla"
## [57] "Okapi"                     "Cow"
## [59] "Horse"                     "Giraffe"
## [61] "Asian elephant"            "African elephant"
##
## [[2]]
## [1] "body"  "brain"
```

```
attach(data)

X <- body
Y <- brain

detach(data)

plot(X, Y, main='Scatterplot brain weight vs body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight')
```




```
plot(X, Y, main='Scatterplot brain weight vs body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight',col='white',xlim=c(-1000,8000))
text(X, Y,dimnames(data)[[1]],cex=1)
```



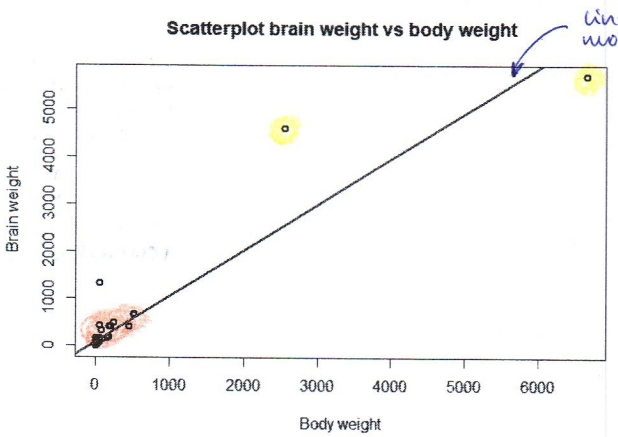
(+ text)

we use the names of the rows of the dataset to see which animal is the outlier

```
result <- lm(Y ~ X)
summary(result)
```

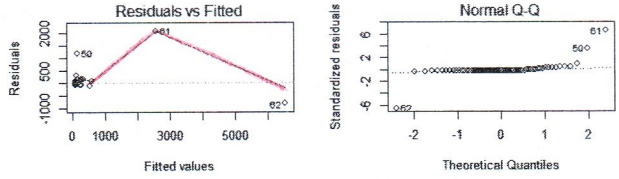
```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -810.07  -88.52  -79.64  -13.02 2050.33
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 91.00440   43.55258    2.09   0.0409 *
## X            0.96650    0.04766   20.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 334.7 on 60 degrees of freedom
## Multiple R-squared:  0.8727, Adjusted R-squared:  0.8705
## F-statistic: 411.2 on 1 and 60 DF,  p-value: < 2.2e-16
```

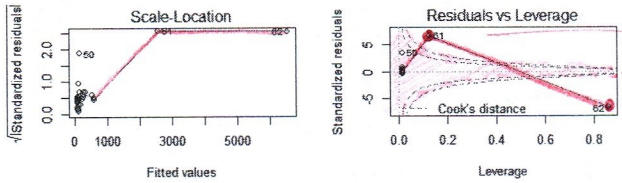
```
coef <- result$coef
plot(X, Y, main='Scatterplot brain weight vs body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight')
abline(coef[1],coef[2], lwd=2,col='red')
```



the points o are driving the estimation of the slope of the model and the points o do not have much part in the estimation of the model → we have 2 very influential points and the other 60 just gives the intercept basically.
Even tho the summary of the results is a very good summary: we can understand what is the problem from the diagnostic of the residuals

```
# diagnostics of the residuals
par(mfrow=c(2,2))
plot(result)
```





there are two points which are influencial points (they're above the level of cook's distance of 1)

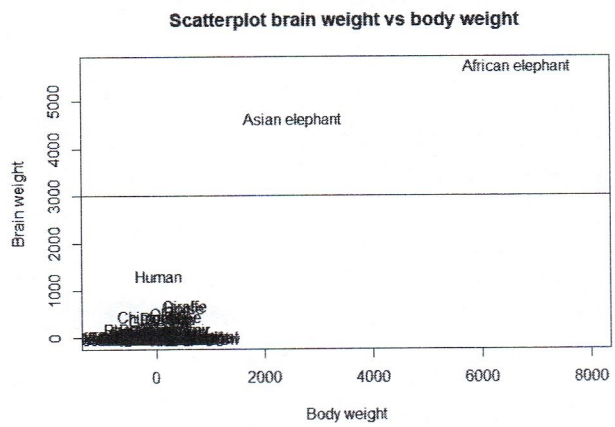
```
shapiro.test(residuals(result))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(result)
## W = 0.41112, p-value = 2.316e-14
```

```
dev.off()
```

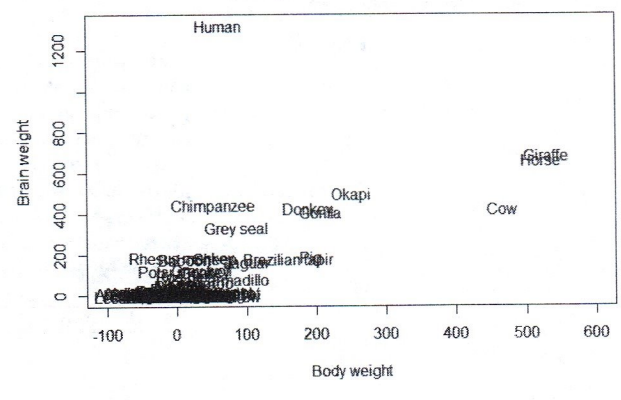
```
## png
##   2
```

```
# exclude outliers?
x11()
plot(X, Y, main='Scatterplot brain weight vs body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight',col='white',xlim=c(-1000,8000))
text(X, Y,dimnames(data)[[1]],cex=1)
abline(h=3000)
```



we cut here the dataset and we obtain a new one

```
plot(X[which(Y<3000)], Y[which(Y<3000)], main='Scatterplot brain weight vs body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight',col='white',xlim=c(-100,600))
text(X[which(Y<3000)], Y[which(Y<3000)],dimnames(data)[[1]],cex=1)
```



it seems like we have the same problem: we have only 60 data now and most of them is again concentrated in one zone

↓

we try to transform the data (LOGARITMIC TRANSF. OF DATA)

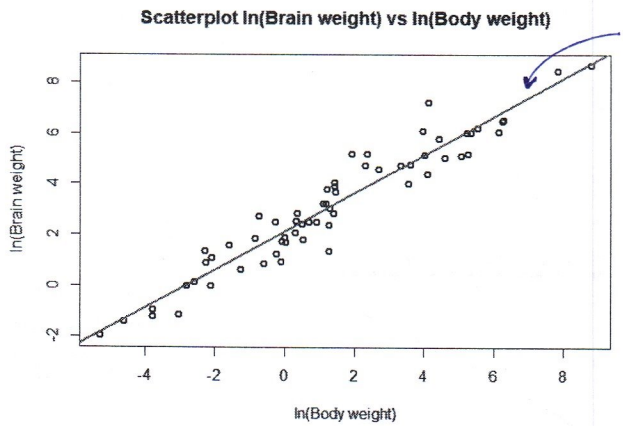
```
dev.off()
```

```
## png
##   2
```

```
# Logarithmic transformation of the data!

log.X <- log(X)
log.Y <- log(Y)

x11()
plot(log.X, log.Y, main='Scatterplot ln(Brain weight) vs ln(Body weight)', lwd=2,
     xlab='ln(Body weight)', ylab='ln(Brain weight)')

result.log <- lm(log.Y ~ log.X)
summary(result.log)
```

```
##
## Call:
## lm(formula = log.Y ~ log.X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71550 -0.49228 -0.06162  0.43597  1.94829
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.13479    0.09604   22.23   <2e-16 ***
## log.X        0.75169    0.02846   26.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6943 on 60 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9195
## F-statistic: 697.4 on 1 and 60 DF,  p-value: < 2.2e-16
```

```
coef.log= result.log$coef
abline(coef.log[1],coef.log[2], lwd=2,col='red')
```



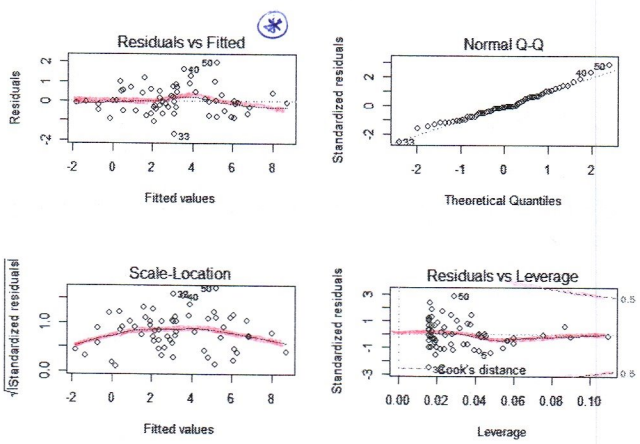
new regression line

Now it's a beauty!

Both the summary (even tho even the other one was ok) and the scatterplot.

Let's check the diagnostic for the residuals!

```
# diagnostics of the residuals
par(mfrow=c(2,2))
plot(result.log)
```



(*) there is a little problem with the homoschedasticity since there is a larger variability in the central part but it's so much better wrt the last model.

```
shapiro.test(residuals(result.log))
```

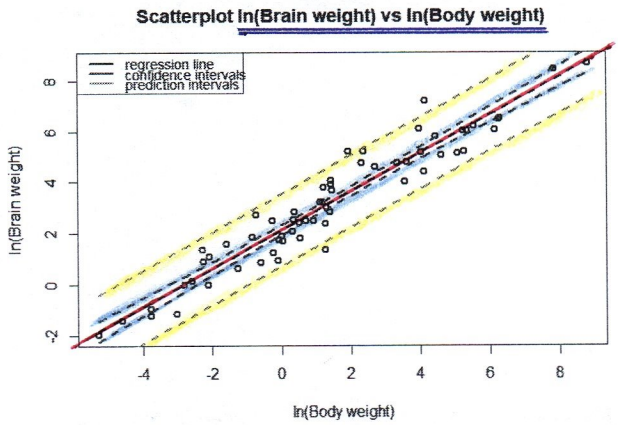
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(result.log)
## W = 0.98268, p-value = 0.5293
```

```
dev.off()
```

```
## png
##   2
```

```
# confidence intervals and prediction intervals
x11()
plot(log.X, log.Y, main='Scatterplot ln(Brain weight) vs ln(Body weight)', lwd=2,
     xlab='ln(Body weight)', ylab='ln(Brain weight)')

X.new.log <- data.frame(log.X = seq(min(log.X), max(log.X), len=100))

IC.log <-predict(result.log ,X.new.log,interval="confidence",level=0.95)
matplot(X.new.log,IC.log,add=T,type='l',col=c('black','blue','blue'),lwd=2,lty=2)

IP.log <-predict(result.log ,X.new.log,interval="prediction",level=0.95)
matplot(X.new.log,IP.log,add=T,type='l',col=c('black','green','green'),lwd=2,lty=2)

legend('topleft', legend=c('regression line','confidence intervals','prediction intervals'),
       col=c('black','blue','green'), lwd=2, cex=0.85)
```

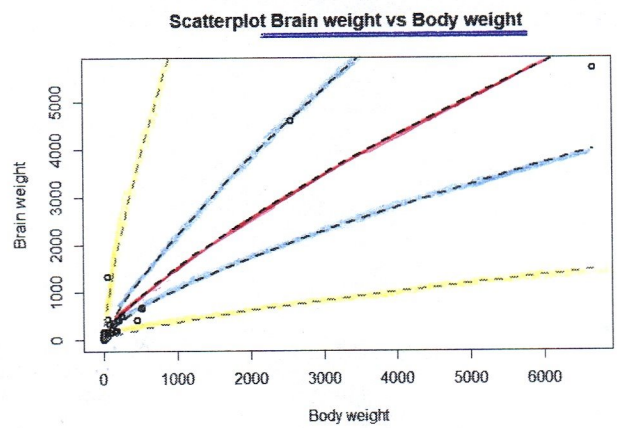
grid
CI
prediction



regression line
confidence intervals
prediction intervals

AGAIN: they're POINTWISE (one-at-the-time)

```
# plot on the original data
plot(X, Y, main='Scatterplot Brain weight vs Body weight', lwd=2,
     xlab='Body weight', ylab='Brain weight')
IC <- exp(IC.log)
IP <- exp(IP.log)
X.new <- exp(X.new.log)
matplot(X.new,IC,add=T,type='l',col=c('black','blue','blue'),lwd=2,lty=2)
matplot(X.new,IP,add=T,type='l',col=c('black','green','green'),lwd=2,lty=2)
```



regression line
confidence intervals
prediction intervals

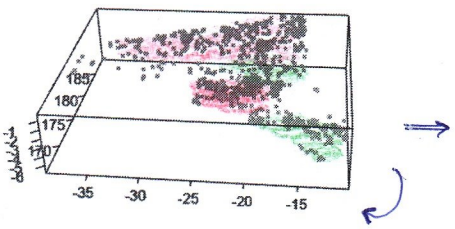
4.

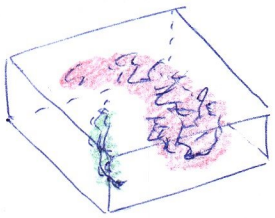
```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Example 4: Earthquakes
### ------------------------------------------------------------------------
Q <- cbind(quakes[,1:2], depth=-quakes[,3]/100)

d <- dist(Q)
clusterw <- cutree(hclust(d, method='ward.D2'), 2)

open3d()
```

→ some data that we used for clustering

```
## wgl
##   1
```

```
par3d(windowRect=c(680,40,1350,720))
points3d(x=Q$lat, y=Q$long, z=Q$depth, size=4, col=clusterw+1, aspect = T)
box3d()
axes3d()

a = scene3d()
rgl.close()
x11()
rglwidget(a)
```





We first perform the hierarchical clustering and we consider the labels obtained with the clustering as features that can be used in the linear model. We consider a dataset composed by:

- latitude
- longitude
- depth (Y)

and an index that says if o ↔ o (so we got also a CATEGORICAL VARIABLE)

only latitude and longitude

```
### Model 1 (all together)
### ------------------------------------------------------------------------
# Model:
# depth = beta0 + beta1*lat + beta2*long + eps

fit  <- lm(depth ~ lat + long, data=Q)
summary(fit)
```

```
##
## Call:
## lm(formula = depth ~ lat + long, data = Q)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8314 -2.2432  0.5165  1.9615  3.4606
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.47977    2.04760   3.653 0.000273 ***
## lat         -0.04136    0.01436  -2.880 0.004068 **
## long        -0.06379    0.01190  -5.360 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.126 on 997 degrees of freedom
## Multiple R-squared:  0.02894,    Adjusted R-squared:  0.02699
## F-statistic: 14.86 on 2 and 997 DF,  p-value: 4.387e-07
```

the blue points are ok but the red one no (very small value of R^2)

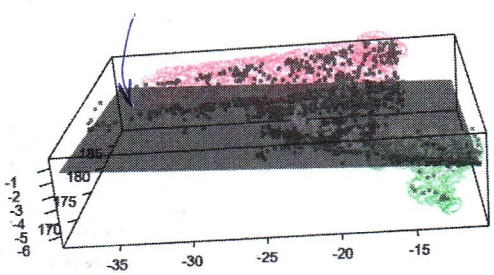
→ why we plot to discover:

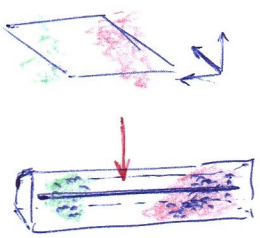
```
open3d()
```

```
## wgl
##   4
```

```
par3d(windowRect=c(680,40,1350,720))
points3d(x=Q$lat, y=Q$long, z=Q$depth, size=4, col=clusterw+1, aspect = T)
box3d()
axes3d()
points3d(x=Q$lat, y=Q$long, z=fitted(fit), size=4, col = 'blue')
surface3d(range(Q$lat), range(Q$long),
          matrix(predict(fit, expand.grid(lat=range(Q$lat), long=range(Q$long))),2,2),
          alpha = 0.5)

a = scene3d()
rgl.close()
x11()
rglwidget(a)
```

we're fitting a plane in the space



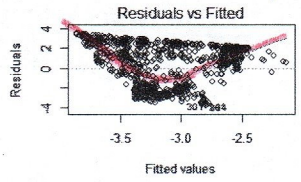


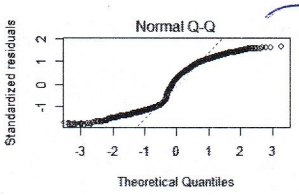
this is the best that a plane can do

diagnostic of the residuals

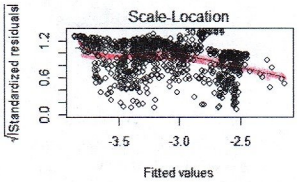
```
par(mfrow=c(2,2))
plot(fit)
```

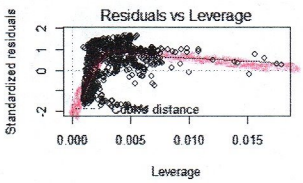
terrible!





→ we're not dealing with a normal distribution (confirmed by shapiro)





⟹ not a good model to describe the data

```
shapiro.test(rstandard(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(fit)
## W = 0.90507, p-value < 2.2e-16
```

```
### Model 2 (dummy variable)
### ------------------------
dummy <- clusterw - 1   # 0 = red
                        # 1 = green

Qd <- cbind(Q, dummy)
head(Qd)
```

: we add the lable (red/green)

```
##      lat   long depth dummy
## 1 -20.42 181.62 -5.62     0
## 2 -20.62 181.03 -6.50     0
## 3 -26.00 184.10 -0.42     0
## 4 -17.97 181.66 -6.26     0
## 5 -20.42 181.96 -6.49     0
## 6 -19.68 184.31 -1.95     0
```

```
# Model:
# depth = beta0       + beta1*lat       + beta2*long       +
#       + beta3*dummy + beta4*dummy*lat + beta5*dummy*long  + eps
# i.e.,
# depth = B0[g] + B1[g]*lat + B2[g]*long + eps
# with B0[g]=beta0       if the unit is in group s.t. dummy=0 (red)
#      B0[g]=beta0+beta3 if the unit is in group s.t. dummy=1 (green)
#      B1[g]=beta1       if the unit is in group s.t. dummy=0 (red)
#      B1[g]=beta1+beta4 if the unit is in group s.t. dummy=1 (green)
#      B2[g]=beta2       if the unit is in group s.t. dummy=0 (red)
#      B2[g]=beta2+beta5 if the unit is in group s.t. dummy=1 (green)

fitd <- lm(depth ~ lat + long + dummy + lat:dummy + long:dummy, data=Qd)
summary(fitd)
```

$$Y = \beta_0 + \beta_1\,lat + \beta_2\,long + \beta_3\,dummy + \beta_4\,dummy*lat + \beta_5\,dummy*long + \varepsilon$$

interaction of the tables (0/1) with each of the one continue variable

equivalent to:

$$Y = \beta_0(g) + \beta_1(g)\,lat + \beta_2(g)\,long + \varepsilon$$

$$\beta_1(g) = \begin{cases} \beta_1 & dummy = 0 \\ \beta_1 + \beta_4 & dummy = 1 \end{cases}$$

$$\beta_2(g) = \begin{cases} \beta_2 & dummy = 0 \\ \beta_2 + \beta_5 & dummy = 1 \end{cases}$$

$$\beta_0(g) = \begin{cases} \beta_0 & dummy = 0 \\ \beta_0 + \beta_3 & dummy = 1 \end{cases}$$

```
##
## Call:
## lm(formula = depth ~ lat + long + dummy + lat:dummy + long:dummy,
##     data = Qd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0644 -0.5425 -0.0247  0.4532  8.3771
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.896e+02  3.351e+00  -56.59   <2e-16 ***
## lat         -3.304e-01  8.412e-03  -39.28   <2e-16 ***
## long         9.813e-01  1.788e-02   54.88   <2e-16 ***
## dummy        3.061e+02  9.119e+00   33.57   <2e-16 ***
## lat:dummy   -3.924e-02  2.929e-02   -1.34    0.181
## long:dummy  -1.718e+00  5.532e-02  -31.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.961 on 994 degrees of freedom
## Multiple R-squared:  0.8022, Adjusted R-squared:  0.8012
## F-statistic: 806.2 on 5 and 994 DF,  p-value: < 2.2e-16
```

much better!

practically we're estimating 2 planes: one for the red group, one for the green group

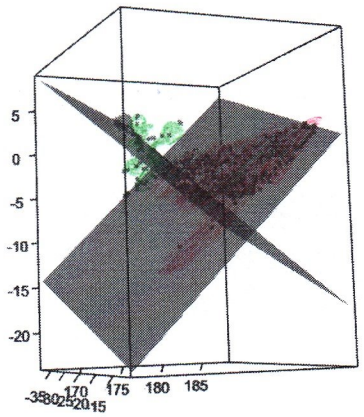
Note that we can split the dataset and do 2 separate regressions but keeping the dataset all together we have the opportunity to check if it is reasonable to split or not (by tests on β_3, β_4, β_5)

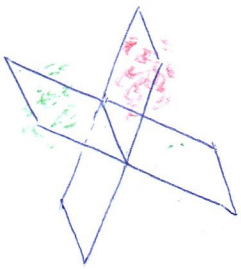
```
# Fitted model:
open3d()
```

```
## wgl
##   7
```

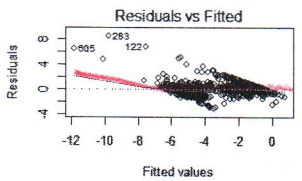
we're mixing ANOVA and the modelling part. This is usually called ANCOVA (analysis of covariance)

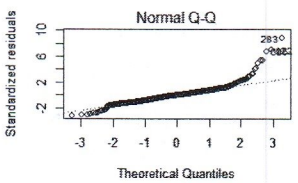
```
par3d(windowRect=c(680,40,1350,720))
points3d(x=Q$lat, y=Q$long, z=Q$depth, size=4, col=clusterw+1, aspect = T)
points3d(x=Qd$lat, y=Qd$long, z=fitted(fitd), size=4, col = 'blue')

surface3d(range(Q$lat), range(Q$long),
          matrix(predict(fitd, expand.grid(lat=range(Q$lat), long=range(Q$long), dummy=c(1,1))),2,2),
          alpha = 0.5, col='green')
surface3d(range(Q$lat), range(Q$long),
          matrix(predict(fitd, expand.grid(lat=range(Q$lat), long=range(Q$long), dummy=c(0,0))),2,2),
          alpha = 0.5, col='red')
box3d()
axes3d()

a = scene3d()
rgl.close()
x11()
rglwidget(a)
```



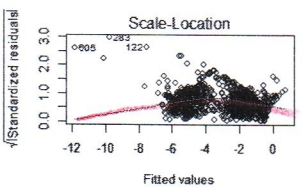


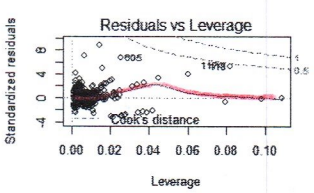
```
# Residuals:
par(mfrow=c(2,2))
plot(fitd)
```





generally the situation improved. There are some outliers (we can see them because they have the number of the row on which they lay).






```
shapiro.test(rstandard(fitd))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(fitd)
## W = 0.8803, p-value < 2.2e-16
```

even tho is not satisfied we can rely on **CLT** because we have large number of data and the tools that we used are based on the gaussianity of the sample means

```
# test: are the two planes needed?
A <- rbind(c(0,0,0,1,0,0), c(0,0,0,0,1,0), c(0,0,0,0,0,1))
b <- c(0,0,0)
linearHypothesis(fitd, A, b)
```

$$A = \begin{matrix} \beta_0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

: we want to see if $\beta_3, \beta_4, \beta_5$ are necessary (so if the division green/red is necessary)

```
## Linear hypothesis test
##
## Hypothesis:
## dummy = 0
## lat:dummy = 0
## long:dummy = 0
##
## Model 1: restricted model
## Model 2: depth ~ lat + long + dummy + lat:dummy + long:dummy
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    997 4506.6
## 2    994  918.1  3    3588.5 1295.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

⟹ there is statistical evidence to take in consideration the division green/red
AT LEAST one dummy - something must be taken into consideration, which one?

```
# Reduce the model:
summary(fitd)
```

```
##
## Call:
## lm(formula = depth ~ lat + long + dummy + lat:dummy + long:dummy,
##     data = Qd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0644 -0.5425 -0.0247  0.4532  8.3771
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.896e+02  3.351e+00  -56.59   <2e-16 ***
## lat         -3.304e-01  8.412e-03  -39.28   <2e-16 ***
## long         9.813e-01  1.788e-02   54.88   <2e-16 ***
## dummy        3.061e+02  9.119e+00   33.57   <2e-16 ***
## lat:dummy   -3.924e-02  2.929e-02   -1.34    0.181
## long:dummy  -1.718e+00  5.532e-02  -31.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.961 on 994 degrees of freedom
## Multiple R-squared:  0.8022, Adjusted R-squared:  0.8012
## F-statistic: 806.2 on 5 and 994 DF,  p-value: < 2.2e-16
```

→ we don't have evidence to consider this among the dummy-something group

```
### Model 3 (reduced model)
### ------------------------------------------------------------------
# Model:
# depth = beta0 +beta1*lat +beta2*long +beta3*dummy +beta4*dummy*long +eps
# i.e.,
# depth = B0[g] + B1*lat + B2[g]*long
# with B0[g]=beta0       if the unit is in group s.t. dummy=0 (red)
#      B0[g]=beta0+beta3 if the unit is in group s.t. dummy=1 (green)
#      B1=beta1
#      B2[g]=beta2       if the unit is in group s.t. dummy=0 (red)
#      B2[g]=beta2+beta5 if the unit is in group s.t. dummy=1 (green)

fitD <- lm(depth ~ lat + long + dummy + long:dummy, data=Qd)
summary(fitD)
```

: we don't consider the interaction: lat : dummy $\Rightarrow \beta_1(g) \rightarrow \beta_1$

```
##
## Call:
## lm(formula = depth ~ lat + long + dummy + long:dummy, data = Qd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2167 -0.5394 -0.0330  0.4690  8.4094
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.903e+02  3.316e+00  -57.39   <2e-16 ***
## lat         -3.336e-01  8.061e-03  -41.39   <2e-16 ***
## long         9.845e-01  1.772e-02   55.55   <2e-16 ***
## dummy        2.989e+02  7.378e+00   40.52   <2e-16 ***
## long:dummy  -1.672e+00  4.291e-02  -38.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9614 on 995 degrees of freedom
## Multiple R-squared:  0.8018, Adjusted R-squared:  0.801
## F-statistic:  1006 on 4 and 995 DF,  p-value: < 2.2e-16
```

all good

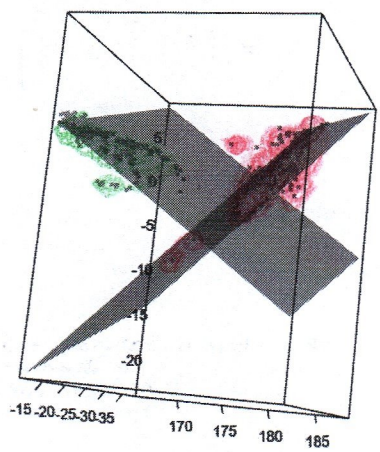
all significant

→ it almost doesn't change (we can reduce the model without losing significant percentage of variability)

```
# Fitted model
open3d()
```

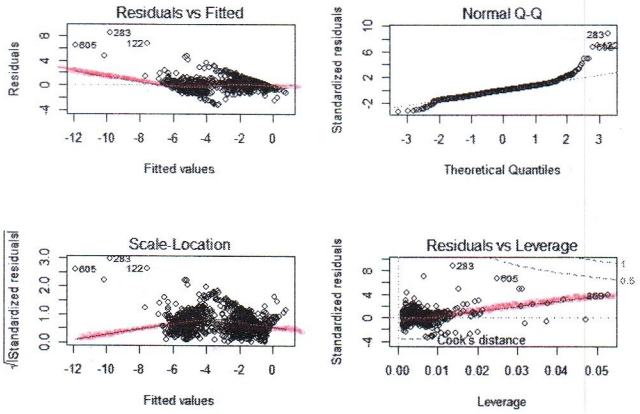
```
## wgl
##  10
```

```
par3d(windowRect=c(680,40,1350,720))
points3d(x=Q$lat, y=Q$long, z=Q$depth, size=4, col=clusterw+1, aspect = T)
axes3d()
points3d(x=Qd$lat, y=Qd$long, z=fitted(fitD), size=4, col = 'blue')
surface3d(range(Q$lat), range(Q$long),
          matrix(predict(fitD, expand.grid(lat = range(Q$lat), long = range(Q$long), dummy=c(1,1))),2,2),
          alpha = 0.5, col='green')
surface3d(range(Q$lat), range(Q$long),
          matrix(predict(fitD, expand.grid(lat = range(Q$lat), long = range(Q$long), dummy=c(0,0))),2,2),
          alpha = 0.5, col='red')

a = scene3d()
rgl.close()
x11()
rglwidget(a)
```



← we obtain something very similar to the previous one (we see that we didn't lose any significant part of the model)

```
# Residuals:
par(mfrow=c(2,2))
plot(fitD)
```



there is still something that can suggest a change in the model (outliers, influencial points, ..). One possibility can be to introduce quadratic regressors in the model. (Homework)

```
shapiro.test(rstandard(fitD))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(fitD)
## W = 0.88297, p-value < 2.2e-16
```

```
dev.off()
```

```
## png
##   2
```

```
### Homework: Fit a linear model with quadratic regressors
### (lat, Long, lat^2, Long^2, lat:Long)
### Perform appropriate statistical tests to answer the following questions:
### Q1: are the quadratic terms needed?
### Q2: is the Latitude needed?
### Q3: is the Longitude needed?
```

5.

```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Example 5: simulated data (Bias-variance trade-off)
### ------------------------------------------------------------------------

# generation of training set and test set
set.seed(1)

# true model: regressors x, x^2, x^3; coefficients (1,1,1,-1)
f <- function(x){1+x+x^2-x^3}
sigma <- 0.25
x <- seq(-1, 1.5, length = 21)
y <- f(x) + rnorm(21, sd = sigma)
y.new <- f(x) + rnorm(21, sd = sigma)

# build design matrix
data <- NULL
for(p in 0:20)
  data <- cbind(data, x^p)
colnames(data) <- c(paste('x', 0:20, sep=''))
data <- data.frame(data)
head(data)
```

we generate data from: $f(x) = x + x^2 + x^3$ adding some noise (gaussian noise)

→ same $f(x)$, different realizations of the noise

to perform the estimation of the model

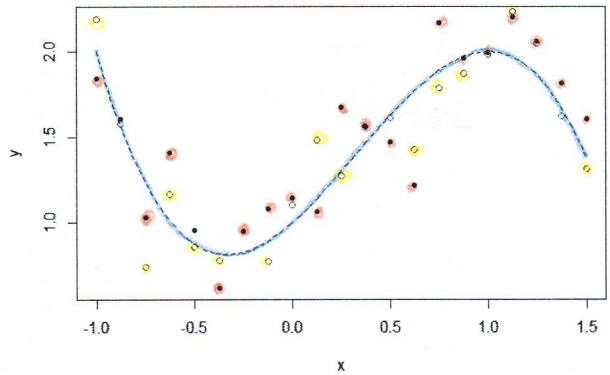
as covariates we use $1, x, x^2, \dots, x^{20}$

```
##   x0      x1       x2          x3         x4           x5          x6
## 1  1 -1.000 1.000000 -1.00000000 1.00000000 -1.000000000 1.000000000
## 2  1 -0.875 0.765625 -0.66992188 0.58618164 -0.512908936 0.448795319
## 3  1 -0.750 0.562500 -0.42187500 0.31640625 -0.237304688 0.177978516
## 4  1 -0.625 0.390625 -0.24414062 0.15258789 -0.095367432 0.059604645
## 5  1 -0.500 0.250000 -0.12500000 0.06250000 -0.031250000 0.015625000
## 6  1 -0.375 0.140625 -0.05273438 0.01977539 -0.007415771 0.002780914
##             x7           x8           x9          x10           x11
## 1 -1.000000000 1.000000000 -1.0000000000 1.000000e+00 -1.000000e+00
## 2 -0.392695904 0.3436089158 -0.3006578013 2.630756e-01 -2.301911e-01
## 3 -0.133483887 0.1001129150 -0.0750846863 5.631351e-02 -4.223514e-02
## 4 -0.037252903 0.0232830644 -0.0145519152 9.094947e-03 -5.684342e-03
## 5 -0.007812500 0.0039062500 -0.0019531250 9.765625e-04 -4.882812e-04
## 6 -0.001042843 0.0003910661 -0.0001466498 5.499367e-05 -2.062263e-05
##            x12           x13          x14           x15           x16
## 1 1.000000e+00 -1.000000e+00 1.000000e+00 -1.000000e+00 1.000000e+00
## 2 2.014172e-01 -1.762401e-01 1.542101e-01 -1.349338e-01 1.180671e-01
## 3 3.167635e-02 -2.375726e-02 1.781795e-02 -1.336346e-02 1.002260e-02
## 4 3.552714e-03 -2.220446e-03 1.387779e-03 -8.673617e-04 5.421011e-04
## 5 2.441406e-04 -1.220703e-04 6.103516e-05 -3.051758e-05 1.525879e-05
## 6 7.733484e-06 -2.900057e-06 1.087521e-06 -4.078205e-07 1.529327e-07
##             x17           x18           x19           x20
## 1 -1.000000e+00 1.000000e+00 -1.000000e+00 1.000000e+00
## 2 -1.033087e-01 9.039511e-02 -7.909572e-02 6.920876e-02
## 3 -7.516947e-03 5.637710e-03 -4.228283e-03 3.171212e-03
## 4 -3.388132e-04 2.117582e-04 -1.323489e-04 8.271806e-05
## 5 -7.629395e-06 3.814697e-06 -1.907349e-06 9.536743e-07
## 6 -5.734975e-08 2.150616e-08 -8.064809e-09 3.024303e-09
```

```
dim(data)
```

```
## [1] 21 21
```

```
# grid to plot
data.plot <- NULL
x.plot <- seq(-1, 1.5, length = 210)
for(p in 0:20)
  data.plot <- cbind(data.plot, x.plot^p)
colnames(data.plot) <- c(paste('x', 0:20, sep=''))
data.frame(data.plot)

# plot of the training set, test set and "true" mean curve
x11()
plot(x, y, pch=20)                          # training set
points(x, y.new, col='red')                 # test set
lines(x.plot, f(x.plot), col='blue', lty=2)  # true mean
```



true model (which we know because we simulated it)

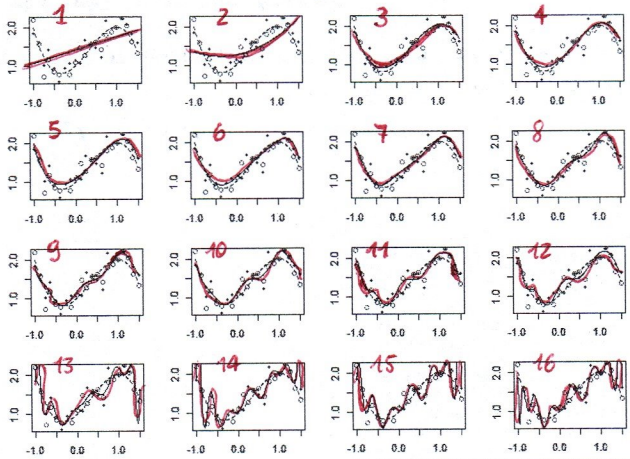
training data

test data

```
# regression with polynomials of increasing order
SSres <- SSres.new <- s2 <- b <- R2 <- R2.adj <- NULL
n <- 21

x11(width=14, height=7)
par(mfrow=c(4,4), mar=rep(2,4))
for(p in 1:16)
{
  fit <- lm(y ~ 0 + . , data[,1:(p+1)])
  plot(x, y, pch=20)
  points(x, y.new, col='red')
  lines(x.plot, predict(fit, data.plot))
  lines(x.plot, f(x.plot), col='blue', lty=2)

  SSres <- c(SSres, sum((y - fitted(fit))^2))
  SSres.new <- c(SSres.new, sum((y.new - fitted(fit))^2))
  s2 <- c(s2, sum((y - fitted(fit))^2)/(n - (p+1)))
  R2 <- c(R2, summary(fit)$r.squared)
  R2.adj <- c(R2.adj, summary(fit)$adj.r.squared)
  bp <- rep(0,17)
  bp[1:(p+1)] <- coefficients(fit)
  b <- cbind(b, bp)
}
```

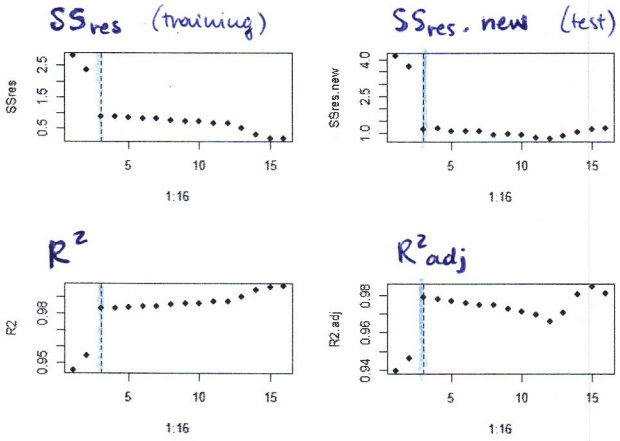
we start from x, then $x+x^2$, then $x+x^2+x^3$, .. until $x+x^2+x^3+..+x^{16}$.

Along the estimation we compute some descriptive quantities (R^2, SS_{res}, ..) and produce some plots.



we can see the bias-variance trade off:
with a very simple model we have small variance (if we change a point the line will be ± the same) but there is a large bias (the estimation is far from the true curve).
Increasing the polynomial we see that we're getting closer to the true polynomial (and the variance is getting higher).
If we increase too much (16 ex.) we go almost through all the points (small bias, high variance).

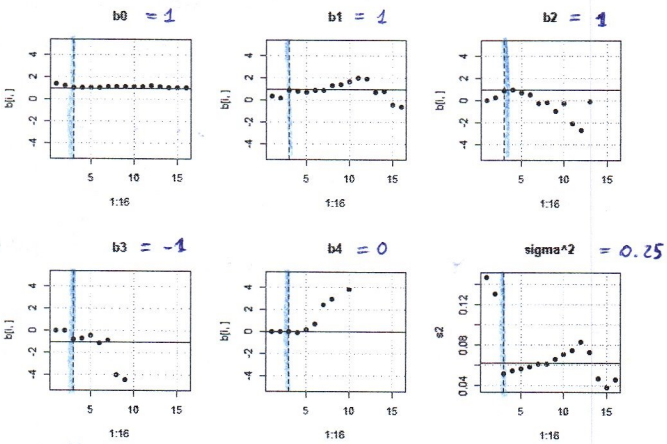
```
# compare some indices

x11()
par(mfrow=c(2,2))
plot(1:16, SSres, pch=16)
abline(v=3, col='blue', lty=2)
plot(1:16, SSres.new, pch=16)
abline(v=3, col='blue', lty=2)
plot(1:16, R2, pch=16)
abline(v=3, col='blue', lty=2)
plot(1:16, R2.adj, pch=16)
abline(v=3, col='blue', lty=2)
```



— true answer (which we know since we generated the data) (we want the degree to be 3)

all the plots give evidence to conclude that the best degree (to stop) is 3

```
# compare parameter estimates
# true model: regressors x, x^2, x^3; coefficients (1,1,1,-1)

b.true <- c(1,1,1,-1, rep(0,8))

x11()
par(mfrow=c(2,3))
for(i in 1:5)
{
  plot(1:16, b[i,], ylim=c(-5, 5), main=paste('b',i-1,sep=''), pch=16)
  grid()
  abline(v=3, col='blue', lty=2)
  abline(h=b.true[i], col='blue')
}
plot(1:16, s2, main='sigma^2', pch=16)
grid()
abline(v=3, col='blue', lty=2)
abline(h=sigma^2, col='blue')
```



— true value of the coefficient

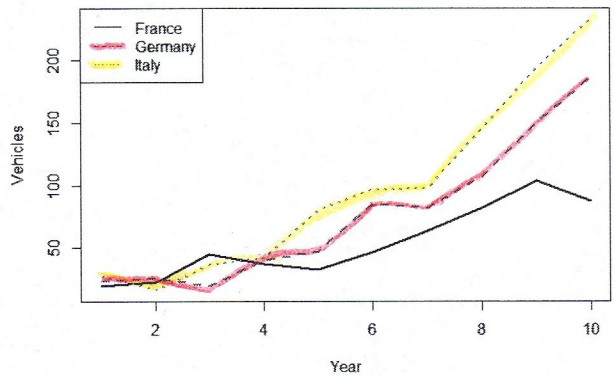
```
graphics.off()
```

```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Exercises on Linear models
### ------------------------------------------------------------------------
###
### Problem 4 of 6/2/2007
### ------------------------------------------------------------------------
# The file Pb4.txt reports the number Y (expressed in thousands of units)
# of vehicles registered annually in three countries of the European Union
# (France, Germany and Italy) during a reference period of 10 years.
# Recent economic models describe the behavior of this variable according
# to the model:
# Y | (X = x, G = g) = beta0.g + beta1.g * x^2 + eps
# with eps ~ N (0, sigma^2), x = 1, 2, ... , 10 (year) and
# g = France, Germany, Italy (EU country).
# (a) With the Least squares method, estimate the 7 parameters of the model.
# (b) Using appropriate statistical tests, state if you deem necessary to
#     include into the model:
#     1. the variable x^2;
#     2. the variable G;
#     3. the effect of the variable G on the coefficient that multiplies
#        the regressor x^2;
#     4. the effect of the variable G on the intercept.
# (c) Once identified the "best model", build three prediction intervals
#     for the number of vehicles registered in the three countries
#     during the eleventh year, so that the three new observations
#     will fall simultaneously within the respective ranges with 95%
#     of probability.

pb4  <- read.table('Pb4.txt')
pb4
```

$$Y \mid X = x,\ G = g \;=\; \underbrace{\beta_0(g)}_{3} + \underbrace{\beta_1(g)\, x^2}_{3} + \underbrace{\varepsilon}_{1}$$

```
##    Francia Germania Italia
## 1    18.99    23.40  26.68
## 2    22.75    25.17  15.97
## 3    45.00    19.46  36.31
## 4    36.93    39.86  41.80
## 5    32.61    46.86  80.48
## 6    46.51    85.22  96.91
## 7    63.71    82.27  97.29
## 8    81.78   106.83 144.87
## 9   103.57   149.71 194.06
## 10   87.08   186.36 232.97
```

```
matplot(pb4, type='l',lwd=2, xlab='Year', ylab='Vehicles')
legend("topleft",c("France", "Germany", "Italy"),lty=1:3,col=1:3)
```



! we want Y= ... but that Y is in 3 different columns of the original data

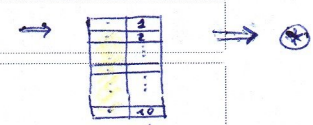
```
### question (a)

# We first build the design matrix and the vector of the responses
Year <- rep(1:10,3)
Year
```

One trick: we change the form of the dataset from matrix to a vector

Francia	Germ	Italia
:	:	:

```
##  [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
## [26]  6  7  8  9 10
```

```
Reg <- c(pb4[,1], pb4[,2], pb4[,3])
Reg
```



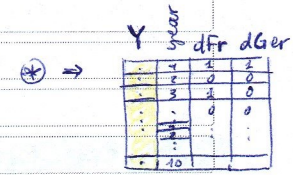
```
##  [1]  18.99  22.75  45.00  36.93  32.61  46.51  63.71  81.78 103.57  87.08
## [11]  23.40  25.17  19.46  39.86  46.86  85.22  82.27 106.83 149.71 186.36
## [21]  26.68  15.97  36.31  41.80  80.48  96.91  97.29 144.87 194.06 232.97
```

Year

$Y = \beta_0(g) + \beta_1(g) \cdot Year^2 + \varepsilon$

```
# Model: Reg = beta0.g + beta1.g*Year^2 + eps
# with g=France,Germany,Italy; E[eps]=0, Var(eps)=sigma^2

# We need to build appropriate dummy variables to account for the
# levels of the categorical variable g=France,Germany,Italy (3 levels)
# g groups => g-1 dummy variables (3 groups => 2 dummy variables)

dFr <- rep(c(1,0), c(10,20))      # dFr = 1 if France,  0 otherwise
dGer<- rep(c(0,1,0), c(10,10,10)) # dGer= 1 if Germany, 0 otherwise
dFr
```

We need more than 1 dummy variable since we have 3 categories (Italia, Germania, Francia)



```
##  [1] 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
dGer
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
```

$Y = \beta_0 + \beta_1 dFr + \beta_2 dGer + \beta_3 Year^2 + \beta_4 dFr \cdot Year^2 + \beta_5 dGer \cdot Year^2 + \varepsilon$

```
# Equivalent Model:
# Reg = b.0+b.1*dFr+b.2*dGer +b.3*Year^2+b.4*dFr*Year^2+b.5*dGer*Year^2 + eps

# Indeed:
# beta0.It=b.0;      beta1.It=b.3;
# beta0.Fr=b.0+b.1;  beta1.Fr=b.3+b.4;
# beta0.Ger=b.0+b.2; beta1.Ger=b.3+b.5

dati <- data.frame(Reg   = Reg,
                   Year2 = Year^2,
                   dFr   = rep(c(1,0), c(10,20)),      # dummy for France
                   dGer  = rep(c(0,1,0), c(10,10,10))) # dummy for Germany
dati
```

```
##       Reg Year2 dFr dGer
## 1   18.99     1   1    0
## 2   22.75     4   1    0
## 3   45.00     9   1    0
## 4   36.93    16   1    0
## 5   32.61    25   1    0
## 6   46.51    36   1    0
## 7   63.71    49   1    0
## 8   81.78    64   1    0
## 9  103.57    81   1    0
## 10  87.08   100   1    0
## 11  23.40     1   0    1
## 12  25.17     4   0    1
## 13  19.46     9   0    1
## 14  39.86    16   0    1
## 15  46.86    25   0    1
## 16  85.22    36   0    1
## 17  82.27    49   0    1
## 18 106.83    64   0    1
## 19 149.71    81   0    1
## 20 186.36   100   0    1
## 21  26.68     1   0    0
## 22  15.97     4   0    0
## 23  36.31     9   0    0
## 24  41.80    16   0    0
## 25  80.48    25   0    0
## 26  96.91    36   0    0
## 27  97.29    49   0    0
## 28 144.87    64   0    0
## 29 194.06    81   0    0
## 30 232.97   100   0    0
```

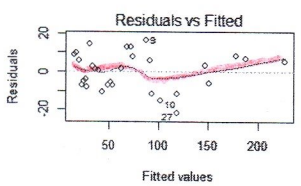
```
fit <- lm(Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer, data=dati)

# Equivalent syntax:
# fit <- lm(Reg ~ dFr + dGer + I(Year^2) + I(Year^2*dFr) + I(Year^2*dGer),
#           data=data.frame(Reg=Reg, Year=Year, dFr=rep(c(1,0), c(10,20)),
#                           dGer = rep(c(0,1,0), c(10,10,10))))

summary(fit)
```

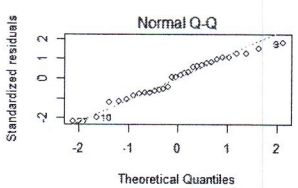
```
##
## Call:
## lm(formula = Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer,
##     data = dati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.821  -7.176   1.165   7.502  16.117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.6861     5.1874   2.831  0.00924 **
## dFr           8.8056     7.3360   1.200  0.24173
## dGer         -1.7229     7.3360  -0.235  0.81631
## Year2         2.1311     0.1031  20.678  < 2e-16 ***
## dFr:Year2    -1.3415     0.1458  -9.204 2.42e-09 ***
## dGer:Year2   -0.4804     0.1458  -3.296  0.00304 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.57 on 24 degrees of freedom
## Multiple R-squared:  0.9717, Adjusted R-squared:  0.9658
## F-statistic:   165 on 5 and 24 DF,  p-value: < 2.2e-16
```

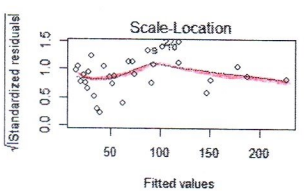
: answers to (a)

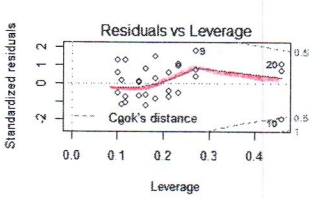
```
### question (b)
par(mfrow=c(2,2))
plot(fit)
```

: before we start with the tests we check if the assumptions for tests are satisfied









good results

```
shapiro.test(residuals(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.9713, p-value = 0.5753
```

good

```
dev.off()
```

```
## null device
##           1
```

```
# 1. the variable x^2;
linearHypothesis(fit,
                 rbind(c(0,0,0,1,0,0),
                       c(0,0,0,0,1,0),
                       c(0,0,0,0,0,1)),
                 c(0,0,0))
```

```
## Linear hypothesis test
##
## Hypothesis:
## Year2 = 0
## dFr:Year2 = 0
## dGer:Year2 = 0
##
## Model 1: restricted model
## Model 2: Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer
##
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1     27 85606
## 2     24  2679  3     82927 247.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 2. the variable G;
linearHypothesis(fit,
                 rbind(c(0,1,0,0,0,0),
                       c(0,0,1,0,0,0),
                       c(0,0,0,0,1,0),
                       c(0,0,0,0,0,1)),
                 c(0,0,0,0))
```

```
## Linear hypothesis test
##
## Hypothesis:
## dFr = 0
## dGer = 0
## dFr:Year2 = 0
## dGer:Year2 = 0
##
## Model 1: restricted model
## Model 2: Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer
##
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     28 21576.5
## 2     24  2679.4  4     18897 42.316 1.548e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#     3. the effect of the variable G onto the coefficient that multiplies
#        the regressor x^2;
linearHypothesis(fit,
                 rbind(c(0,0,0,0,1,0),
                       c(0,0,0,0,0,1)),
                 c(0,0))
```

```
## Linear hypothesis test
##
## Hypothesis:
## dFr:Year2 = 0
## dGer:Year2 = 0
##
## Model 1: restricted model
## Model 2: Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer
##
##   Res.Df     RSS Df Sum of Sq     F    Pr(>F)
## 1     26 12390.2
## 2     24  2679.4  2    9710.8 43.49 1.046e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#     4. the effect of the variable G on the intercept.
linearHypothesis(fit,
                 rbind(c(0,1,0,0,0,0),
                       c(0,0,1,0,0,0)),
                 c(0,0))
```

```
## Linear hypothesis test
##
## Hypothesis:
## dFr = 0
## dGer = 0
##
## Model 1: restricted model
## Model 2: Reg ~ dFr + dGer + Year2 + Year2:dFr + Year2:dGer
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     26 2944.1
## 2     24 2679.4  2    264.64 1.1852 0.3229
```

```
### question (c)
fit2 <- lm(Reg ~ Year2 + Year2:dFr + Year2:dGer, data=dati)
summary(fit2)
```

← reduced model

```
##
## Call:
## lm(formula = Reg ~ Year2 + Year2:dFr + Year2:dGer, data = dati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.424  -8.235   1.961   7.074  19.965
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.04701    3.01618   5.652 6.07e-06 ***
## Year2        2.09523    0.08106  25.847  < 2e-16 ***
## Year2:dFr   -1.20765    0.09455 -12.773 1.04e-12 ***
## Year2:dGer  -0.50663    0.09455  -5.358 1.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.64 on 26 degrees of freedom
## Multiple R-squared:  0.9689, Adjusted R-squared:  0.9654
## F-statistic: 270.4 on 3 and 26 DF,  p-value: < 2.2e-16
```

```
new_data <- data.frame(Year2 = c(11,11,11)^2, dFr=c(1,0,0), dGer=c(0,1,0))
IP <- predict(fit2, newdata=new_data, interval='prediction', level=1-0.05/3)
rownames(IP) <- c('Fr','Ger','It')
IP
```

```
##          fit       lwr      upr
## Fr  124.4451  89.63246 159.2578
## Ger 209.2684 174.45569 244.0810
## It  270.5703 235.75764 305.3830
```

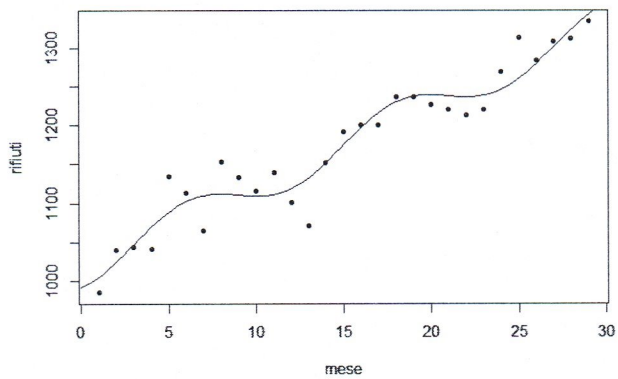
```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Problem 4 of 29/6/2011
### ------------------------------------------------------------------------
# The file people.txt records the tons of waste collected monthly
# in the city of Santander since January 2009 (t = 1) until May 2011
# (t = 29). Assuming a model of the type:
#   Waste = A + B * t  + C * (1-cos(2pi / 12 * t)) + eps
# with eps ~ N(0, sigma^2) and identifying the contribution of the residents
# with the first two factors, and that of the tourists with the third
# addendum, answer the following questions.
# a) Estimate the parameters of the model.
# b) On the basis of the model (a), is there statistical evidence of an
#    increase attributable to residents?
# c) On the basis of the model (a), is there statistical evidence of a
#    significant contribution by tourists?
# d) The University of Cantabria considered that the GROWTH attributable to
#    residents is quantifiable in an increase of 10 tons per month.
#    Can you deny this statement?
# e) Based on the test (b), (c) and (d) propose a possible reduced and/or
#    constrained model and estimate its parameters.
# f) On the basis of model (e), provide three pointwise forecasts for the
#    waste that will be collected in June 2011, for the waste that will be
#    collected in June 2011 due to residents and that which will be collected
#    in June 2011 due to the tourists.

people <- read.table('people.txt', header=T)
people
```

```
##    mese rifiuti
## 1     1  984.48
## 2     2 1039.32
## 3     3 1043.92
## 4     4 1041.11
## 5     5 1133.93
## 6     6 1113.26
## 7     7 1064.64
## 8     8 1152.77
## 9     9 1132.39
## 10   10 1115.10
## 11   11 1139.01
## 12   12 1101.04
## 13   13 1071.19
## 14   14 1151.87
## 15   15 1191.16
## 16   16 1200.03
## 17   17 1199.90
## 18   18 1235.97
## 19   19 1235.77
## 20   20 1225.97
## 21   21 1220.18
## 22   22 1213.21
## 23   23 1220.21
## 24   24 1268.82
## 25   25 1313.18
## 26   26 1283.94
## 27   27 1308.87
## 28   28 1312.18
## 29   29 1333.98
```

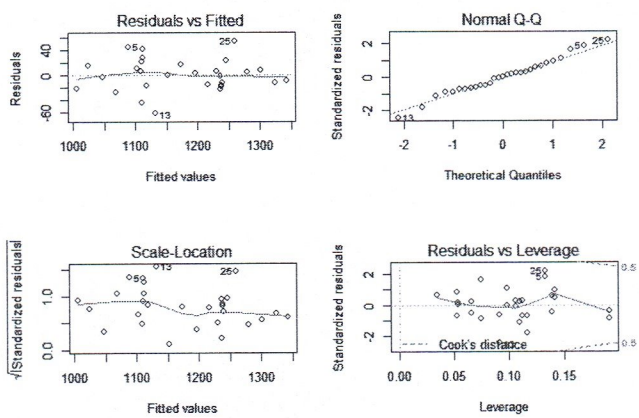
```
x11()
plot(people,pch=20)

attach(people)

### question a)
fit <- lm(rifiuti ~ mese + I(1 - cos(2*pi/12*mese)))
summary(fit)
```

```
##
## Call:
## lm(formula = rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.162 -17.438   0.256  14.846  53.688
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 991.445     12.600  78.689  < 2e-16 ***
## mese                         10.595      0.590  17.957 3.57e-16 ***
## I(1 - cos(2 * pi/12 * mese))   23.678      7.105   3.333  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.56 on 26 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.9211
## F-statistic: 164.4 on 2 and 26 DF,  p-value: 1.753e-15
```

```
t <- seq(from=0,to=30,length=100)
points(t,fit$coeff[1]+fit$coeff[2]*t+fit$coeff[3]*(1-cos(2*pi/12*t)),type='l')
```




```
### question b)
par(mfrow=c(2,2))
plot(fit)
```




```
shapiro.test(residuals(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.98413, p-value = 0.9285
```

```
dev.off()
```

```
## png
##   2
```

```
# Test: H0: beta_1==0 vs beta_1!=0
summary(fit)
```

```
##
## Call:
## lm(formula = rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.162 -17.438   0.256  14.846  53.688
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  991.445     12.600  78.689  < 2e-16 ***
## mese                          10.595      0.590  17.957 3.57e-16 ***
## I(1 - cos(2 * pi/12 * mese))  23.678      7.105   3.333  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.56 on 26 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.9211
## F-statistic: 164.4 on 2 and 26 DF,  p-value: 1.753e-15
```

```
## or
linearHypothesis(fit,rbind(c(0,1,0)),0)
```

```
## Linear hypothesis test
##
## Hypothesis:
## mese = 0
##
## Model 1: restricted model
## Model 2: rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese))
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     27 245757
## 2     26  18337  1    227420 322.45 3.57e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### question c)
# Test: H0: beta_2==0 vs beta_2!=0
summary(fit)
```

```
##
## Call:
## lm(formula = rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.162 -17.438   0.256  14.846  53.688
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  991.445     12.600  78.689  < 2e-16 ***
## mese                          10.595      0.590  17.957 3.57e-16 ***
## I(1 - cos(2 * pi/12 * mese))  23.678      7.105   3.333  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.56 on 26 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.9211
## F-statistic: 164.4 on 2 and 26 DF,  p-value: 1.753e-15
```

```
## or
linearHypothesis(fit,rbind(c(0,0,1)),0)
```

```
## Linear hypothesis test
##
## Hypothesis:
## I(1 - cos(2 * pi/12 * mese)) = 0
##
## Model 1: restricted model
## Model 2: rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese))
##
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     27 26171
## 2     26 18337  1    7833.7 11.107 0.002587 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### question d)
linearHypothesis(fit,rbind(c(0,1,0)),10)
```

```
## Linear hypothesis test
##
## Hypothesis:
## mese = 10
##
## Model 1: restricted model
## Model 2: rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese))
##
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     27 19055
## 2     26 18337  1    717.21 1.0169 0.3225
```

```
# or (from the summary)
summary(fit)
```

```
##
## Call:
## lm(formula = rifiuti ~ mese + I(1 - cos(2 * pi/12 * mese)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.162 -17.438   0.256  14.846  53.688
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 991.445     12.600  78.689  < 2e-16 ***
## mese                         10.595      0.590  17.957 3.57e-16 ***
## I(1 - cos(2 * pi/12 * mese))   23.678      7.105   3.333  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.56 on 26 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.9211
## F-statistic: 164.4 on 2 and 26 DF,  p-value: 1.753e-15
```

```
t <- (coef(fit)[2]-10)/sqrt(diag(vcov(fit))[2])
t
```

```
##    mese
## 1.00842
```

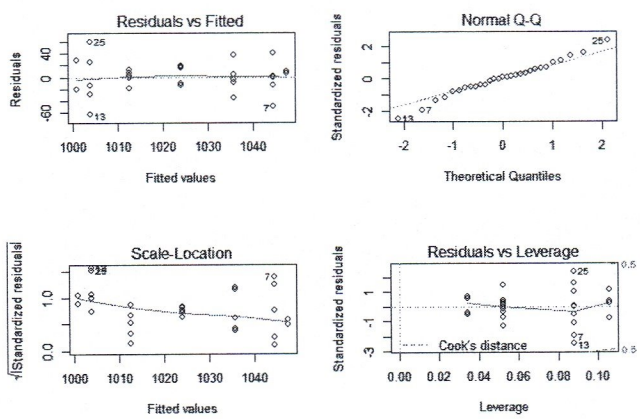
```
pval <- 2*(1-pt(t,29-(2+1)))
pval
```

```
##      mese
## 0.3225463
```

```
### question e)
rifiuti.vinc <- rifiuti - 10*mese

fit2 <- lm(rifiuti.vinc ~ I(1 - cos(2*pi/12*mese)))
summary(fit2)
```

```
##
## Call:
## lm(formula = rifiuti.vinc ~ I(1 - cos(2 * pi/12 * mese)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.629 -13.868   1.493  14.822  59.361
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1000.690      8.645  115.75  < 2e-16 ***
## I(1 - cos(2 * pi/12 * mese))   23.358      7.100    3.29  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.57 on 27 degrees of freedom
## Multiple R-squared:  0.2862, Adjusted R-squared:  0.2597
## F-statistic: 10.82 on 1 and 27 DF,  p-value: 0.002791
```

```
### question f)
# f) On the basis of model (e), provide three pointwise forecasts for the
#    waste that will be collected in June 2011, for the waste that will be
#    collected in June 2011 due to residents and that which will be collected
#    in June 2011 due to the tourists.
x11()
par(mfrow=c(2,2))
plot(fit2)
```




```
shapiro.test(residuals(fit2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit2)
## W = 0.98643, p-value = 0.9631
```

```
dev.off()
```

```
## png
##   2
```

```
coefficients(fit2)
```

```
##                 (Intercept) I(1 - cos(2 * pi/12 * mese))
##                  1000.69006                    23.35821
```

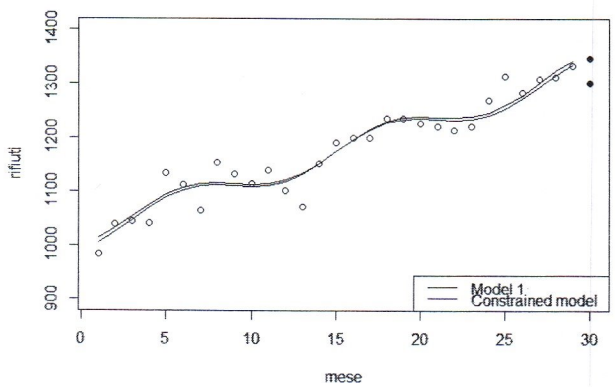
```
C <- rbind(c(1,(1 - cos(2*pi/12*30))),   # total waste in June 2011 [mese=30]
           c(1,0),                       # waste due to residents in June 2011
           c(0,(1 - cos(2*pi/12*30))))   # waste due to tourists in June 2011
C
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    0
## [3,]    0    2
```

```
pred <- C %*% coefficients(fit2) + c(10*30, 10*30, 0)
# pred=C%*%beta.hat[fit.mod.constrained] + 10*mese[constrained part]
pred
```

```
##           [,1]
## [1,] 1347.40649
## [2,] 1300.69006
## [3,]   46.71643
```

```
x11()
plot(people, xlim=c(1,30), ylim=c(900,1400))
lines(mese, fitted(fit))
lines(mese, fitted(fit2) + 10*mese, col='blue')
points(c(30,30,30), pred, pch=16)
legend('bottomright',c('Model 1', 'Constrained model'), lty=1, col=c('black','blue'))
```




```
dev.off()
```

```
## png
##   2
```

```
### ------------------------------------------------------------------------
### ------------------------------------------------------------------------
### Multiple linear regression with qualitative predictors
### ------------------------------------------------------------------------

data <- read.table('work.txt', header=T)
head(data)
```

```
##   Employee Average_Score Years_Service    Sex      Race
## 1        1           7.6             5 Female Nonwhite
## 2        2           9.0            30 Female Nonwhite
## 3        3           8.0            12 Female Nonwhite
## 4        4           6.8             7 Female Nonwhite
## 5        5           7.4             7 Female Nonwhite
## 6        6           9.8            27 Female    White
```

```
dim(data)
```

```
## [1] 20  5
```

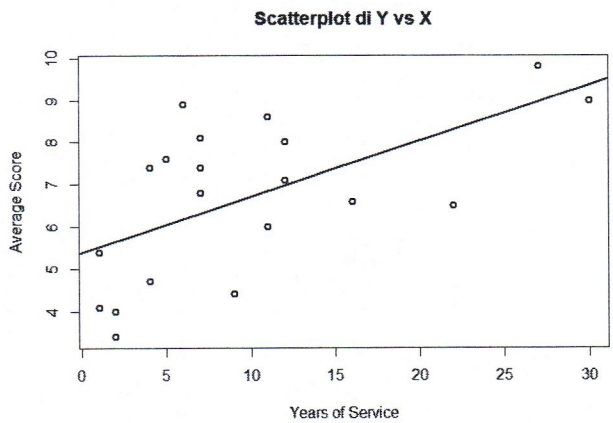
```
n <- dim(data)[1]

names(data)
```

```
## [1] "Employee"      "Average_Score" "Years_Service" "Sex"
## [5] "Race"
```

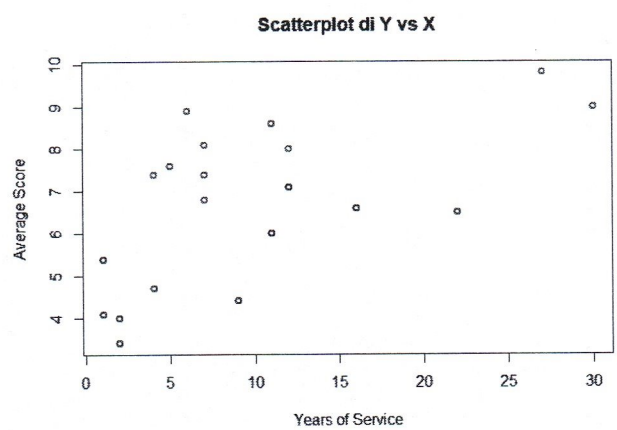
```r
attach(data)

Y <- Average_Score
X <- Years_Service
C1 <- Sex
C2 <- Race

detach(data)

x11()
plot(X, Y, main='Scatterplot di Y vs X', lwd=2,
     xlab='Years of Service', ylab='Average Score')

result <- lm(Y ~ X)
summary(result)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25596 -1.27167 -0.00252  1.18062  2.71376
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.39082    0.53911   9.999 8.94e-09 ***
## X            0.13257    0.04242   3.125  0.00585 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.535 on 18 degrees of freedom
## Multiple R-squared:  0.3517, Adjusted R-squared:  0.3157
## F-statistic: 9.766 on 1 and 18 DF,  p-value: 0.00585
```

```r
coef <- result$coef
abline(coef[1],coef[2],lwd=2)
```




```r
# differences between males and females:
col <- rep('blue',n)
females <- which(C1=='Female')
males <- which(C1=='Male')
col[females] <- 'red'

plot(X, Y, main='Scatterplot di Y vs X', lwd=2,
     xlab='Years of Service', ylab='Average Score', col = col)
```



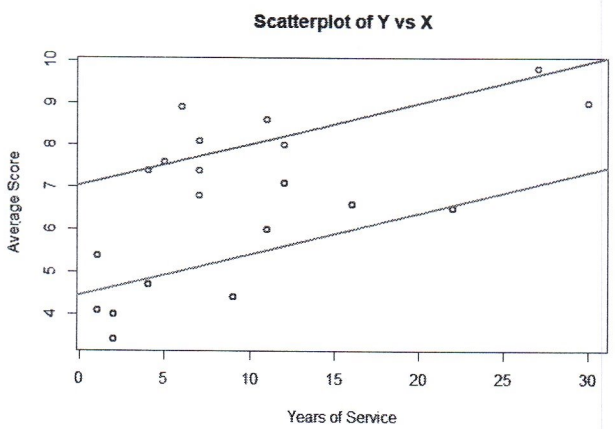

```r
### Multiple linear regression with one qualitative predictor

C1.new <- rep(0,n)
C1.new[males] <- 1

result1 <- lm(Y ~  X + C1.new + X:C1.new)
summary(result1)
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + X:C1.new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0280 -0.4430 -0.1206  0.3965  1.3300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.32289    0.39824  18.388 3.48e-12 ***
## X            0.07216    0.02736   2.637   0.0179 *
## C1.new      -3.20289    0.54300  -5.898 2.25e-05 ***
## X:C1.new     0.06534    0.04443   1.470   0.1608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7605 on 16 degrees of freedom
## Multiple R-squared:  0.8585, Adjusted R-squared:  0.832
## F-statistic: 32.37 on 3 and 16 DF,  p-value: 5.004e-07
```

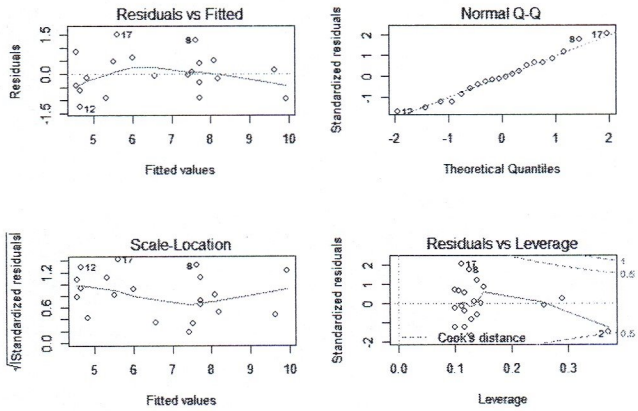
```
result2 <- lm(Y ~ C1.new + X)
summary(result2)
```

```
##
## Call:
## lm(formula = Y ~ C1.new + X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23832 -0.49061 -0.05023  0.49141  1.49221
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.03542    0.35862  19.618 4.10e-13 ***
## C1.new      -2.59099    0.36058  -7.186 1.52e-06 ***
## X            0.09695    0.02228   4.351 0.000435 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7861 on 17 degrees of freedom
## Multiple R-squared:  0.8394, Adjusted R-squared:  0.8205
## F-statistic: 44.44 on 2 and 17 DF,  p-value: 1.771e-07
```

```
# interpretation of the model:
# modell for females: Y = 7.035 + 0.097 X
# modell males:       Y = 7.035 - 2.59 + 0.097 X = 4.44 + 0.097 X

plot(X, Y, main='Scatterplot of Y vs X', lwd=2,
     xlab='Years of Service', ylab='Average Score', col = col)

coef <- result2$coef
abline(coef[1],coef[3],lwd=2,col='indianred')
abline(coef[1]+coef[2],coef[3],lwd=2,col='cornflowerblue')
```




```
# diagnostics of the residuals
par(mfrow=c(2,2))
plot(result2)
```




```
shapiro.test(residuals(result2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(result2)
## W = 0.9752, p-value = 0.8584
```

```
dev.off()
```

```
## png
##   2
```

```
### Multiple linear regression with two qualitative predictors

# qualitative predictors:
C1
```

```
##  [1] "Female" "Female" "Female" "Female" "Female" "Female" "Female" "Female"
##  [9] "Female" "Female" "Male"   "Male"   "Male"   "Male"   "Male"   "Male"
## [17] "Male"   "Male"   "Male"   "Male"
```

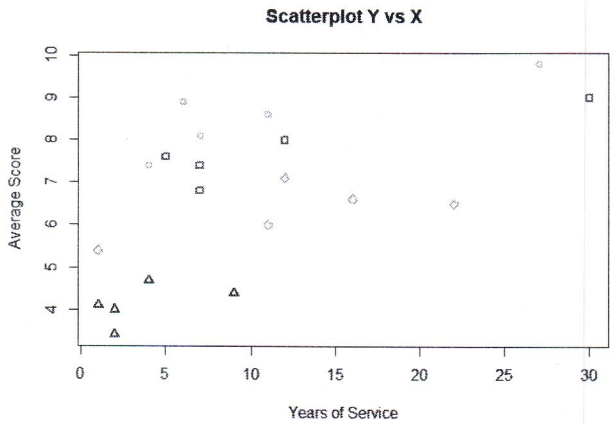
```
C2
```

```
##  [1] "Nonwhite" "Nonwhite" "Nonwhite" "Nonwhite" "Nonwhite" "White"
##  [7] "White"    "White"    "White"    "White"    "Nonwhite" "Nonwhite"
## [13] "Nonwhite" "Nonwhite" "Nonwhite" "White"    "White"    "White"
## [19] "White"    "White"
```

```
C1.new
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
```

```
notwhite <- which(C2=='Nonwhite')
white <- which(C2=='White')
C2.new <- rep(0,n)
C2.new[notwhite] <- 0
C2.new[white] <- 1

# 4 cases:
# females white
FB <- which(C1.new==0 & C2.new==1)
# females not white
FNB <- which(C1.new==0 & C2.new==0)
# males white
MB <- which(C1.new==1 & C2.new==1)
# males not white
MNB <- which(C1.new==1 & C2.new==0)


# colors for the plot
col <- rep(NA,n)
col[FB] <- 'pink'
col[FNB] <- 'red'
col[MB] <- 'light blue'
col[MNB] <- 'blue'
# shape of the dots for the plot
shape <- rep(0,n)
shape[FB] <- 21
shape[FNB] <- 22
shape[MB] <- 23
shape[MNB] <- 24

x11()
plot(X, Y, main='Scatterplot Y vs X', lwd=2,
     xlab='Years of Service', ylab='Average Score', col = col, pch = shape)
```




```
result3 <- lm(Y ~ X + C1.new + C2.new + X:C1.new + X:C2.new)
result3
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new + X:C1.new + X:C2.new)
##
## Coefficients:
## (Intercept)            X       C1.new       C2.new     X:C1.new     X:C2.new
##    6.621494     0.082729    -2.700704     1.306699     0.008501    -0.013545
```

```
summary(result3)
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new + X:C1.new + X:C2.new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80493 -0.31983 -0.03909  0.24599  0.94029
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.621494   0.328461  20.159 9.66e-12 ***
## X            0.082729   0.022993   3.598  0.00291 **
## C1.new      -2.700704   0.380441  -7.099 5.35e-06 ***
## C2.new       1.306699   0.382630   3.415  0.00419 **
## X:C1.new     0.008501   0.033229   0.256  0.80180
## X:C2.new    -0.013545   0.031301  -0.433  0.67180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5107 on 14 degrees of freedom
## Multiple R-squared:  0.9442, Adjusted R-squared:  0.9243
## F-statistic: 47.37 on 5 and 14 DF,  p-value: 2.785e-08
```

```
result4 <- lm(Y ~ X + C1.new + C2.new + X:C2.new)
result4
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new + X:C2.new)
##
## Coefficients:
## (Intercept)            X       C1.new       C2.new     X:C2.new
##     6.58419      0.08471     -2.62673      1.31742     -0.01191
```

```
summary(result4)
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new + X:C2.new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79280 -0.31336 -0.07144  0.26699  0.95152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.58419    0.28500  23.102 3.87e-13 ***
## X            0.08471    0.02097   4.039  0.00107 **
## C1.new      -2.62673    0.23940 -10.972 1.45e-08 ***
## C2.new       1.31742    0.36829   3.577  0.00275 **
## X:C2.new    -0.01191    0.02967  -0.401  0.69386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4945 on 15 degrees of freedom
## Multiple R-squared:  0.9439, Adjusted R-squared:  0.929
## F-statistic: 63.13 on 4 and 15 DF,  p-value: 3.333e-09
```

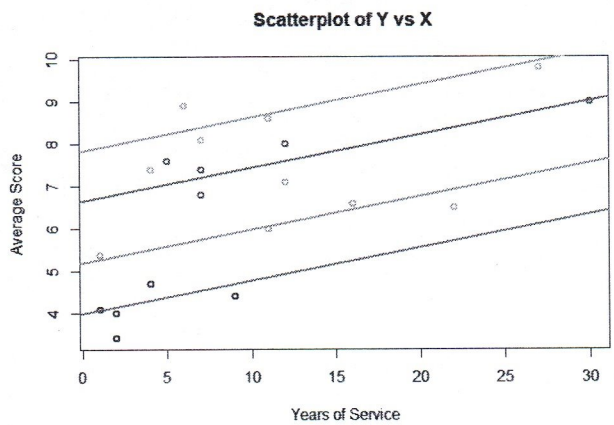
```
result5 <- lm(Y ~ X + C1.new + C2.new)
result5
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new)
##
## Coefficients:
## (Intercept)            X       C1.new       C2.new
##      6.6475       0.0786      -2.6570       1.2013
```

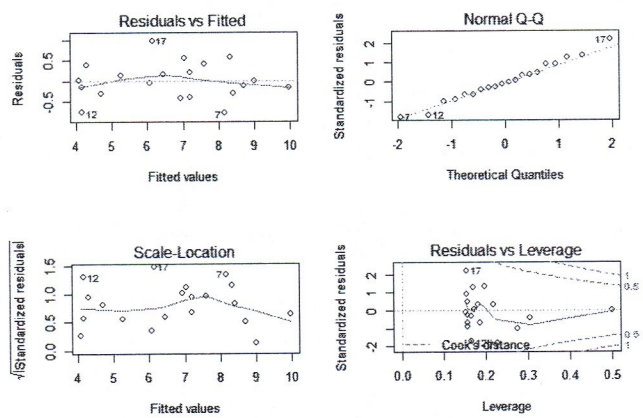
```
summary(result5)
```

```
##
## Call:
## lm(formula = Y ~ X + C1.new + C2.new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76326 -0.29823 -0.03107  0.25044  0.96493
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.64754    0.23099  28.779 3.30e-15 ***
## X            0.07860    0.01406   5.591 4.06e-05 ***
## C1.new      -2.65702    0.22114 -12.015 2.02e-09 ***
## C2.new       1.20130    0.22180   5.416 5.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 16 degrees of freedom
## Multiple R-squared:  0.9433, Adjusted R-squared:  0.9327
## F-statistic: 88.77 on 3 and 16 DF,  p-value: 3.462e-10
```

```
plot(X, Y, main='Scatterplot of Y vs X', lwd=2,
     xlab='Years of Service', ylab='Average Score', col = col)

coef <- result5$coef
abline(coef[1],coef[2],lwd=2,col='indianred') # female, nonwhite
abline(coef[1]+coef[3],coef[2],lwd=2,col='cornflowerblue') # male, nonwhite
abline(coef[1]+coef[4],coef[2],lwd=2,col='pink2') # female, white
abline(coef[1]+coef[3]+coef[4],coef[2],lwd=2,col='lightblue3') # male, white
```




```
# diagnostics of the residuals
par(mfrow=c(2,2))
plot(result5)
```




```
shapiro.test(residuals(result5))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(result5)
## W = 0.98092, p-value = 0.9454
```

```
dev.off()
```

```
## png
##   2
```