Linear Regression

Faïcel Chamroukhi

05/02/2025

Contents

Apparts data Paris

** Some parts of this script may not be optimized as they are done in live tutorial, only **

```
apparts<-read.csv("https://chamroukhi.com/data/apparts_Paris.csv", header=TRUE, sep=",", dec=".")

apparts
```

```
##    surfaces prices
## 1        28    130
## 2        50    280
## 3        55    268
## 4       110    500
## 5        60    320
## 6        48    250
## 7        90    378
## 8        35    250
## 9        86    350
## 10       65    300
## 11       32    155
## 12       52    245
## 13       40    200
## 14       70    325
## 15       28     85
## 16       30     78
## 17      105    375
## 18       52    200
## 19       80    270
## 20       20     85
```

```
surfaces <- apparts$surfaces
prices <- apparts$prices
```

Basic statistics

```
summary(surfaces)
```

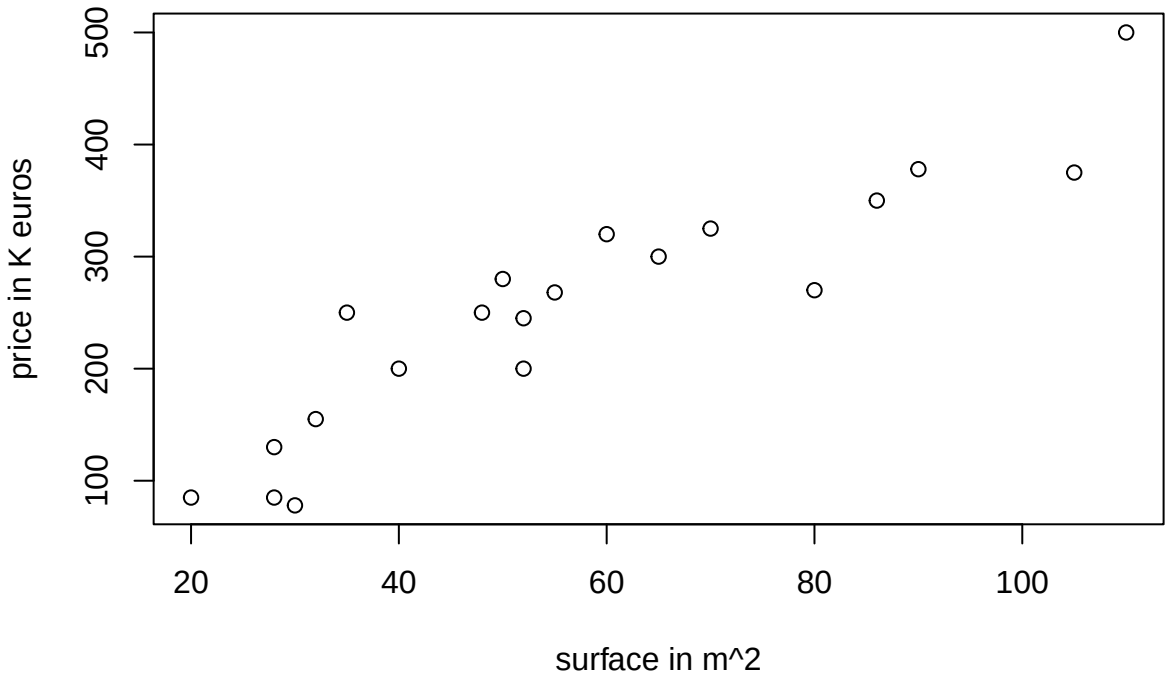
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.00   34.25   52.00   56.80   72.50  110.00
```

```
summary(prices)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    78.0   188.8   259.0   252.2   321.2   500.0
```

Show the sample cloud

```
plot(surfaces, prices, xlab= "surface in m^2",ylab = "price in K euros")
```



cloud-1.pdf

Fit a Linear Regression

```
# Linear Regression

x <- surfaces
y <- prices
lr_fit = lm(y ~ x)
```

Summary of the model

```
summary(lr_fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.469 -27.633   4.748  24.960  81.682
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.6438    24.4450   1.376    0.186
## x             3.8478     0.3922   9.811  1.2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.02 on 18 degrees of freedom
## Multiple R-squared:  0.8425, Adjusted R-squared:  0.8337
## F-statistic: 96.26 on 1 and 18 DF,  p-value: 1.197e-08
```

The fitted parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ and related statistics

```
summary(lr_fit)$coefficients
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 33.64382 24.4449689 1.376308 1.856079e-01
## x            3.84782  0.3921877 9.811170 1.196624e-08
```

Plot the fitted line

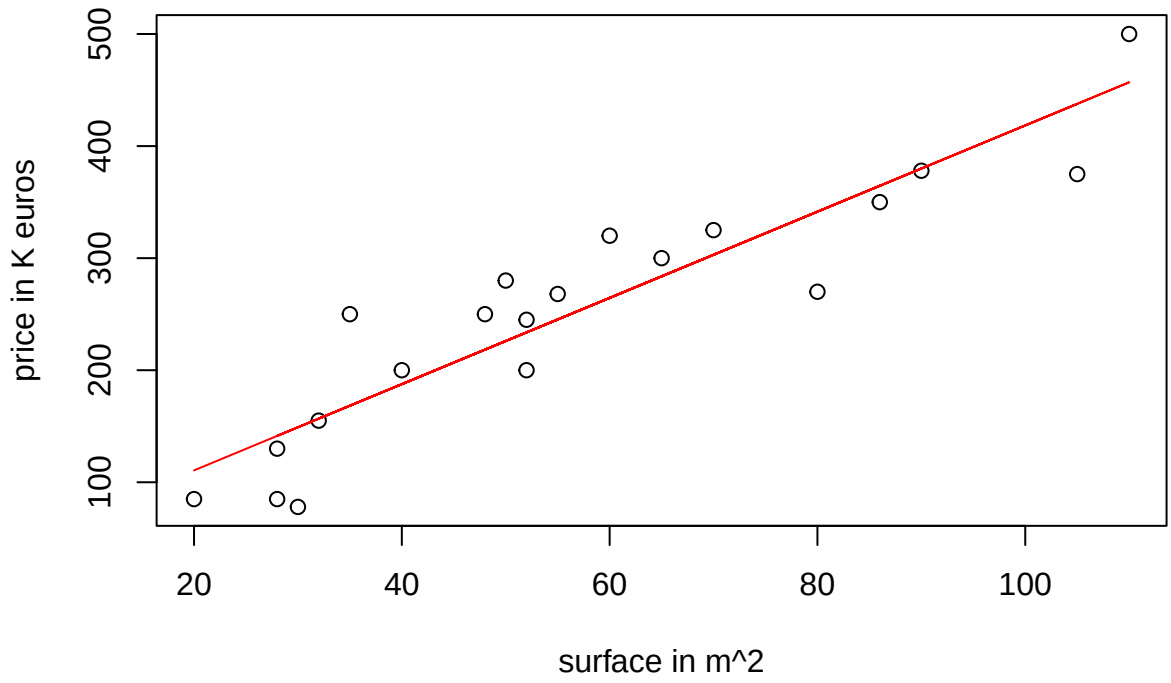
```
y_hat = lr_fit$fitted.values

beta_0 = lr_fit$coefficients[1]
print(beta_0)
```

```
## (Intercept)
##    33.64382
```

```
beta_1 = lr_fit$coefficients[2]
print(beta_1)
```

```
##       x
## 3.84782
```

```
plot(x, y, xlab= "surface in m^2",ylab = "price in K euros")
lines(x, y_hat, col="red")
```



model-1.pdf

```
#lines(x, beta_0 + beta_1*x , col="blue")
#legend("topright", legend=c("data", "fitted model"),col=c("black", "red"), lty=1:2, cex=0.8)
```

Coefficient of determination R^2

```
summary(lr_fit)$r.squared
```

```
## [1] 0.8424632
```

Confidence Intervals for $\hat{\beta}_0$ and $\hat{\beta}_1$

```
alpha = 0.05
confint(lr_fit, level = 1 - alpha)
```

```
##                   2.5 %    97.5 %
## (Intercept) -17.713158 85.000790
## x             3.023864  4.671776
```

%{r, include=TRUE, echo=TRUE} %n = length(y) %qt(1 - alpha/2, n-2) # quantile of order alpha/2 of student t with n-2 df %

Statistical testing

```
summary(lr_fit)$coefficients
```

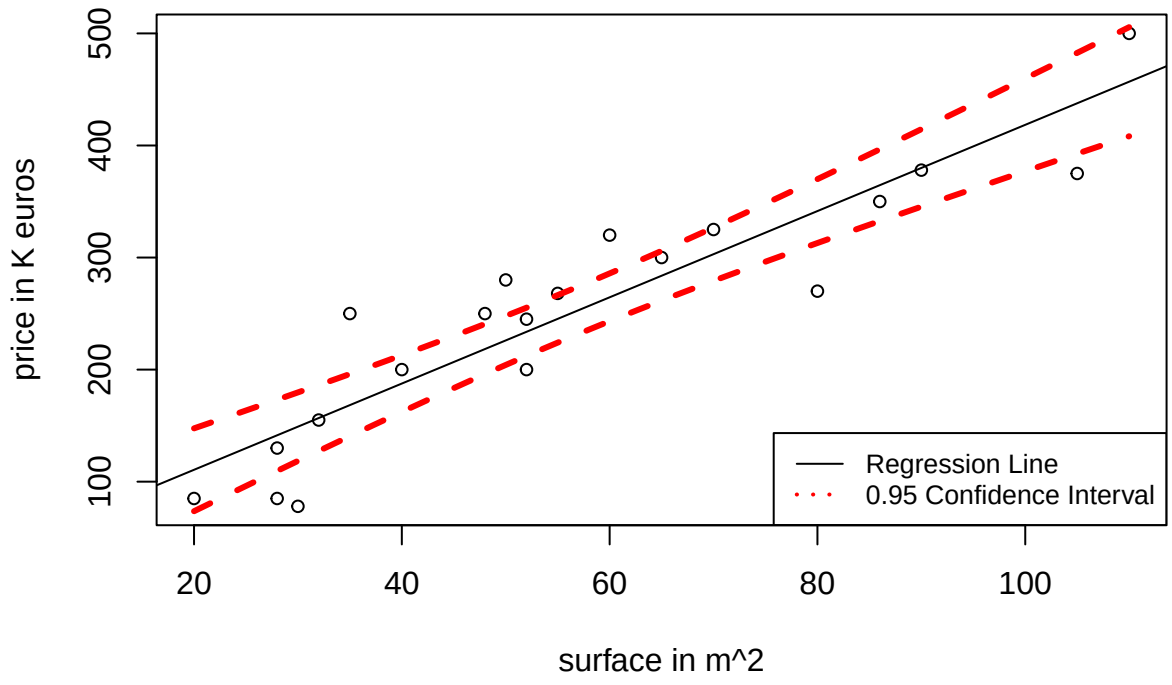
```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 33.64382 24.4449689 1.376308 1.856079e-01
## x            3.84782  0.3921877 9.811170 1.196624e-08
```

Confidence interval for the regression line

```
seqx <- seq(min(x),max(x),length=100)

#apparts <- data.frame(surfaces, prices)
apparts <- data.frame(x, y)


Conf_Int <- predict(lr_fit, data.frame(x = seqx), interval="confidence")[,c("lwr","upr")]
summary(Conf_Int)
```

```
##       lwr              upr
##  Min.   : 73.63   Min.   :147.6
##  1st Qu.:172.97   1st Qu.:221.4
##  Median :261.55   Median :306.0
##  Mean   :253.74   Mean   :313.8
##  3rd Qu.:337.36   3rd Qu.:403.3
##  Max.   :408.23   Max.   :505.6
```

```
plot(y~x, xlab= "surface in m^2",ylab = "price in K euros",  cex=0.8) #,xlim = c(min(x),max(x)), ylim =
abline(lr_fit$coefficients[1],lr_fit$coefficients[2])
#matlines(seqx, cbind(Conf_Int, Pred_Int),lty=c(2,2,3,3), col=c("red","red","blue","blue"),lwd=c(2,2))
matlines(seqx, Conf_Int, lty=c(2,2), lwd=c(3,3), col=c("red","red"))

legend("bottomright",lty=c(1,3),lwd=c(1,2), c("Regression Line", paste(toString(1- alpha),"Confidence I
```



model-1.pdf

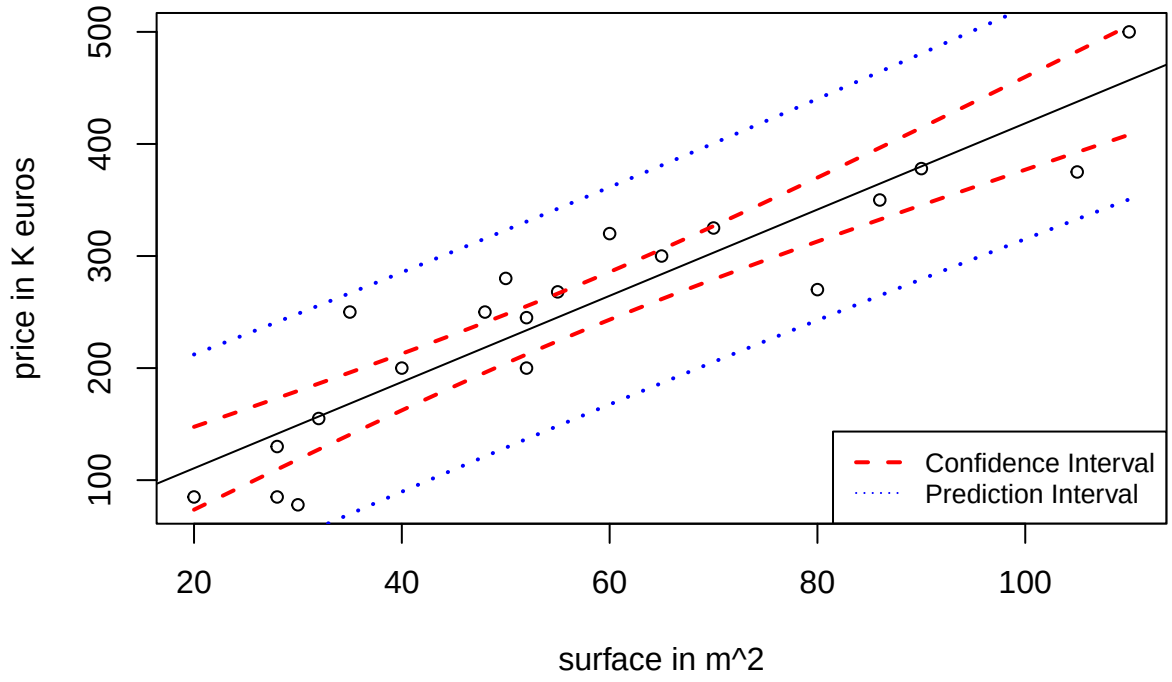
```
#lines(lr_fit$fitted.values)
#matlines(Pred_Int)
```

Prediction interval

```
Pred_Int <- predict(lr_fit, data.frame(x = seqx), interval="prediction")[,c("lwr","upr")]

summary(Conf_Int)
```

```
##       lwr              upr
##  Min.   : 73.63   Min.   :147.6
##  1st Qu.:172.97   1st Qu.:221.4
##  Median :261.55   Median :306.0
##  Mean   :253.74   Mean   :313.8
##  3rd Qu.:337.36   3rd Qu.:403.3
##  Max.   :408.23   Max.   :505.6
```

```
plot(y~x, xlab= "surface in m^2",ylab = "price in K euros",  cex=0.8)

abline(lr_fit$coefficients[1],lr_fit$coefficients[2])

matlines(seqx, cbind(Conf_Int, Pred_Int),lty=c(2,2,3,3), col=c("red","red","blue","blue"),lwd=c(2,2))

legend("bottomright",lty=c(2,3),lwd=c(2,1), c("Confidence Interval","Prediction Interval"),col=c("red",
```



model-1.pdf

```
#lines(lr_fit$fitted.values)
#matlines(Pred_Int)
```

Example, for an appart of 30m^2, would 120K be a good deal ?

```
x0 <- 30
predict(lr_fit, data.frame(x = x0), interval="confidence")
```

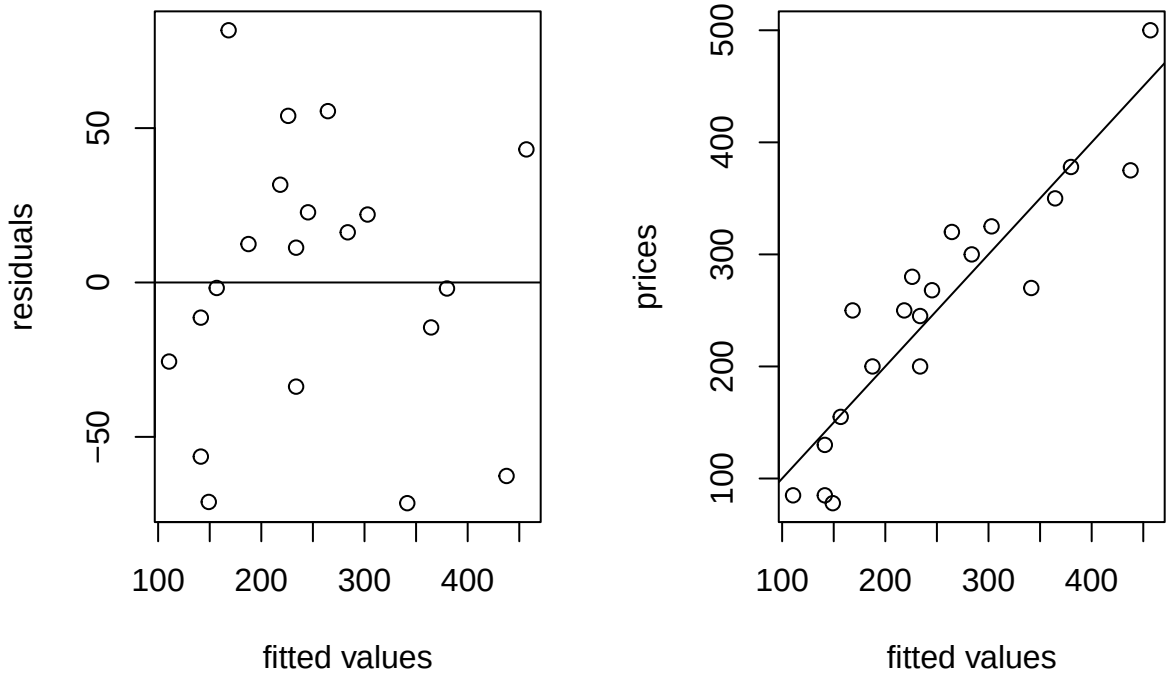
```
##        fit      lwr      upr
## 1 149.0784 118.5031 179.6537
```

```r
predict(lr_fit, data.frame(x = x0), interval="prediction")
```

```
##        fit      lwr      upr
## 1 149.0784 49.68257 248.4743
```

Residuals

```r
y_hat = lr_fit$fitted.values

residuals <-  y - y_hat

#plot(x, residuals )
par(mfrow = c(1, 2))
plot(lr_fit$fitted.values, residuals, xlab = "fitted values", ylab = "residuals")
abline(0,0, col="black")
plot(lr_fit$fitted.values, y, xlab = "fitted values", ylab = "prices")
abline(0,1, col="black")
```



Apparts data Marseille

```r
apparts<-read.csv("https://chamroukhi.com/data/apparts_Marseille.csv", header=TRUE, sep=",", dec=".")

apparts
```

```
##    surfaces prices
## 1        62  190.8
## 2        68  245.0
## 3        67  200.0
## 4        76  350.0
```

```
## 5        90  274.0
## 6        67  242.0
## 7        90  420.0
## 8        60  195.0
## 9        81  220.0
## 10       90  265.0
## 11       70  182.0
## 12       63  208.0
## 13       80  223.0
## 14       84  252.0
## 15       73  343.0
## 16       70  199.0
## 17       80  230.0
## 18       78  245.0
## 19       63  222.0
## 20       66  185.0
```

```
surfaces <- apparts$surfaces
prices <- apparts$prices
```

Basis statistics

```
summary(surfaces)
```

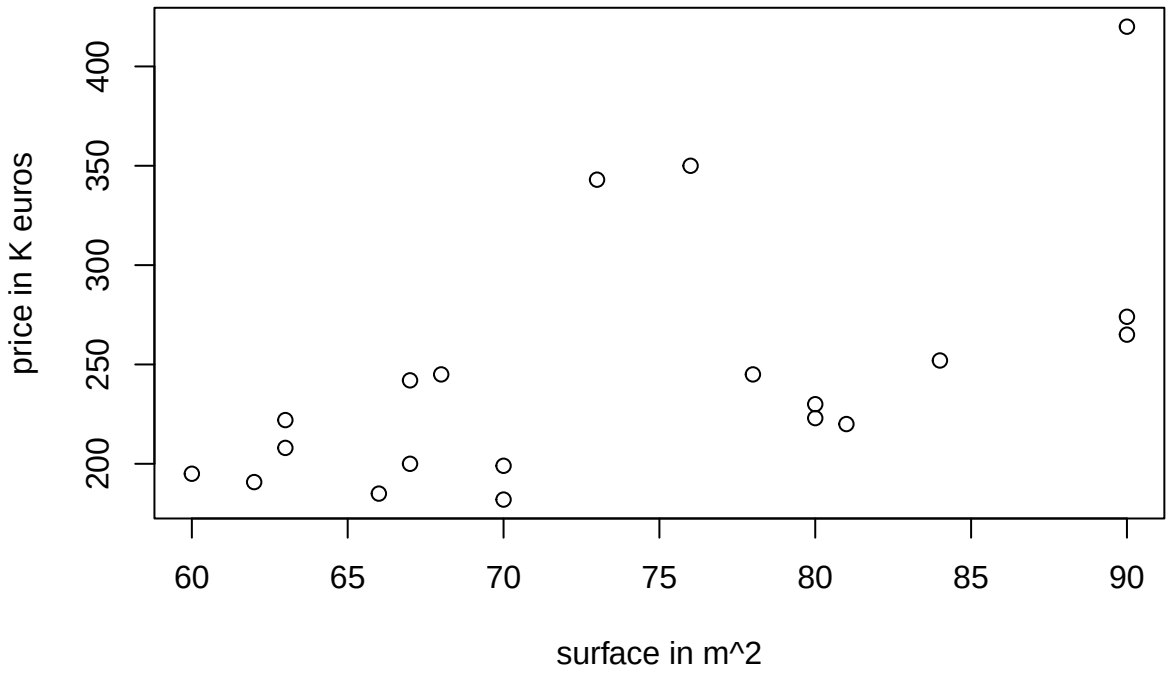
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   66.75   71.50   73.90   80.25   90.00
```

```
summary(prices)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   182.0   199.8   226.5   244.5   255.2   420.0
```

Show the sample cloud

```
plot(surfaces, prices, xlab= "surface in m^2",ylab = "price in K euros")
```



Fit a Linear Regression

```
# Linear Regression

x <- surfaces
y <- prices
lr_fit = lm(y ~ x)
```

Summary of the model

```
summary(lr_fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -50.38 -32.70 -16.95  18.33 116.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -24.42      90.90  -0.269  0.79124
## x               3.64       1.22   2.984  0.00796 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.07 on 18 degrees of freedom
## Multiple R-squared:  0.3309, Adjusted R-squared:  0.2937
```

```
## F-statistic: 8.902 on 1 and 18 DF,  p-value: 0.007965
```

The fitted parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ and related statistics

```
summary(lr_fit)$coefficients
```

```
##               Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) -24.420628  90.896005 -0.2686656 0.79124212
## x             3.639521   1.219855  2.9835680 0.00796485
```

Plot the fitted line

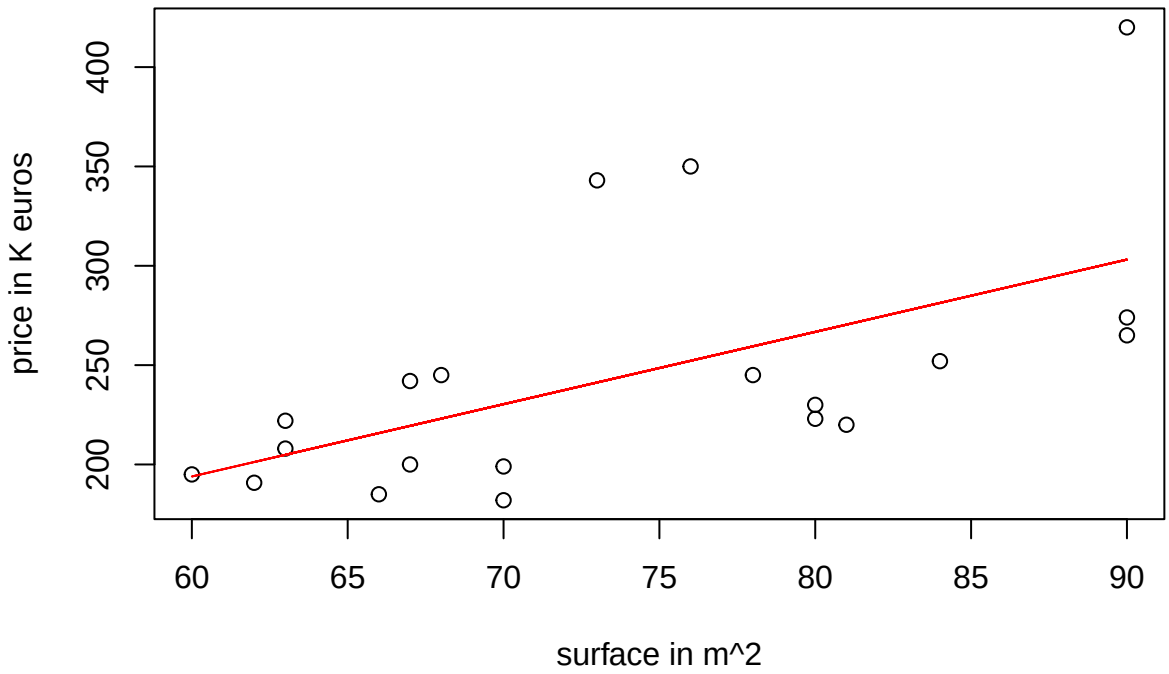
```
y_hat = lr_fit$fitted.values

beta_0 = lr_fit$coefficients[1]
print(beta_0)
```

```
## (Intercept)
##   -24.42063
```

```
beta_1 = lr_fit$coefficients[2]
print(beta_1)
```

```
##        x
## 3.639521
```

```
plot(x, y, xlab= "surface in m^2",ylab = "price in K euros")
lines(x, y_hat, col="red")
```




```
#lines(x, beta_0 + beta_1*x , col="blue")
#legend("topright", legend=c("data", "fitted model"),col=c("black", "red"), lty=1:2, cex=0.8)
```

Coefficient of determination R^2

```
summary(lr_fit)$r.squared
```

```
## [1] 0.3308968
```

Confidence Intervals for $\hat{\beta}_0$ and $\hat{\beta}_1$

```
alpha = 0.05
confint(lr_fit, level = 1 - alpha)
```

```
##                 2.5 %     97.5 %
## (Intercept) -215.3860 166.544792
## x              1.0767   6.202342
```

Statistical testing

```
summary(lr_fit)$coefficients
```

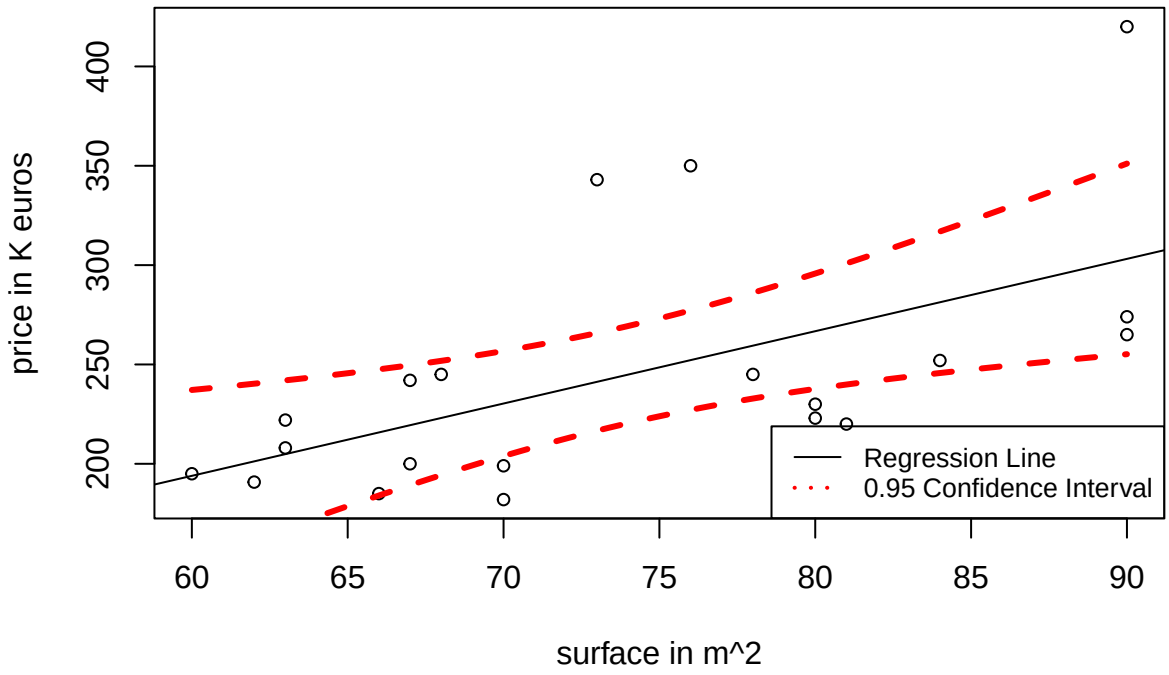
```
##               Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) -24.420628  90.896005 -0.2686656 0.79124212
## x             3.639521   1.219855  2.9835680 0.00796485
```

Confidence interval for the regression line

```
seqx <- seq(min(x),max(x),length=100)

#apparts <- data.frame(surfaces, prices)
apparts <- data.frame(x, y)


Conf_Int <- predict(lr_fit, data.frame(x = seqx), interval="confidence")[,c("lwr","upr")]
summary(Conf_Int)
```

```
##       lwr             upr
##  Min.   :150.7   Min.   :237.2
##  1st Qu.:191.8   1st Qu.:250.7
##  Median :223.9   Median :273.2
##  Mean   :215.9   Mean   :281.1
##  3rd Qu.:242.9   3rd Qu.:308.8
##  Max.   :255.2   Max.   :351.1
```

```
plot(y~x, xlab= "surface in m^2",ylab = "price in K euros",  cex=0.8) #,xlim = c(min(x),max(x)), ylim =
abline(lr_fit$coefficients[1],lr_fit$coefficients[2])
#matlines(seqx, cbind(Conf_Int, Pred_Int),lty=c(2,2,3,3), col=c("red","red","blue","blue"),lwd=c(2,2))
matlines(seqx, Conf_Int, lty=c(2,2), lwd=c(3,3), col=c("red","red"))

legend("bottomright",lty=c(1,3),lwd=c(1,2), c("Regression Line", paste(toString(1- alpha),"Confidence I
```



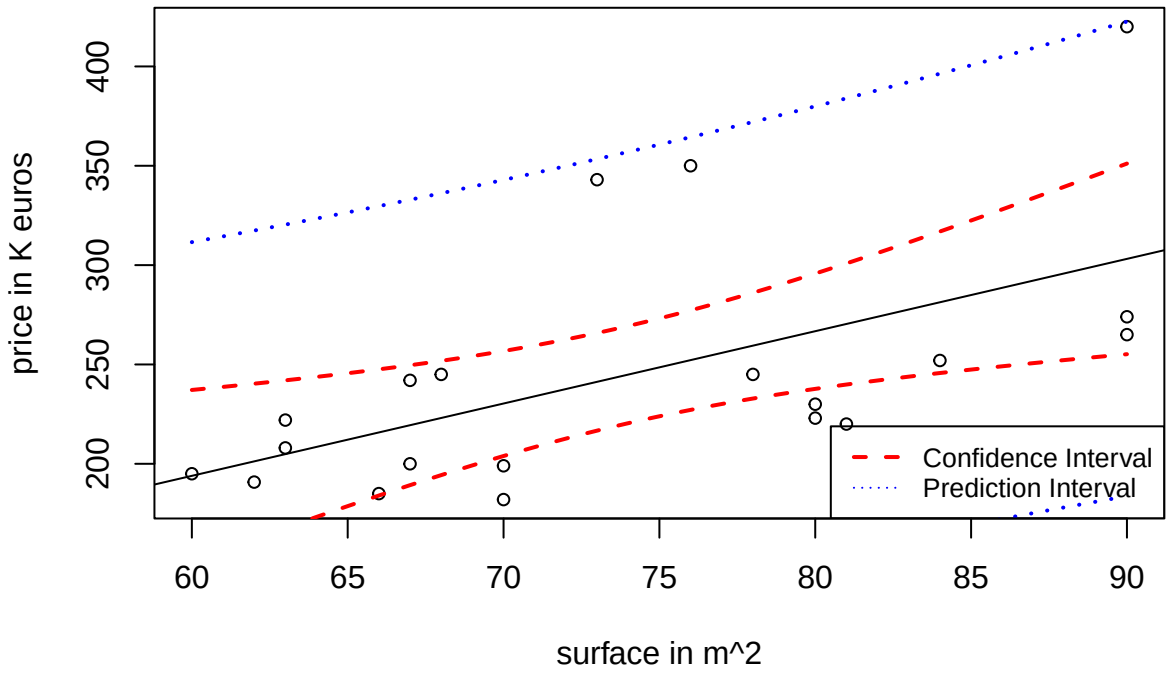

```
#lines(lr_fit$fitted.values)
#matlines(Pred_Int)
```

Prediction interval

```
Pred_Int <- predict(lr_fit, data.frame(x = seqx), interval="prediction")[,c("lwr","upr")]

summary(Conf_Int)
```

```
##       lwr             upr
##  Min.   :150.7   Min.   :237.2
##  1st Qu.:191.8   1st Qu.:250.7
##  Median :223.9   Median :273.2
##  Mean   :215.9   Mean   :281.1
##  3rd Qu.:242.9   3rd Qu.:308.8
##  Max.   :255.2   Max.   :351.1
```

```
plot(y~x, xlab= "surface in m^2",ylab = "price in K euros",  cex=0.8)

abline(lr_fit$coefficients[1],lr_fit$coefficients[2])

matlines(seqx, cbind(Conf_Int, Pred_Int),lty=c(2,2,3,3), col=c("red","red","blue","blue"),lwd=c(2,2))

legend("bottomright",lty=c(2,3),lwd=c(2,1), c("Confidence Interval","Prediction Interval"),col=c("red",
```



Example, for an appart of $50m^2$, would 120K be a good deal ?

```
x0 <- 50
predict(lr_fit, data.frame(x = x0), interval="confidence")
```

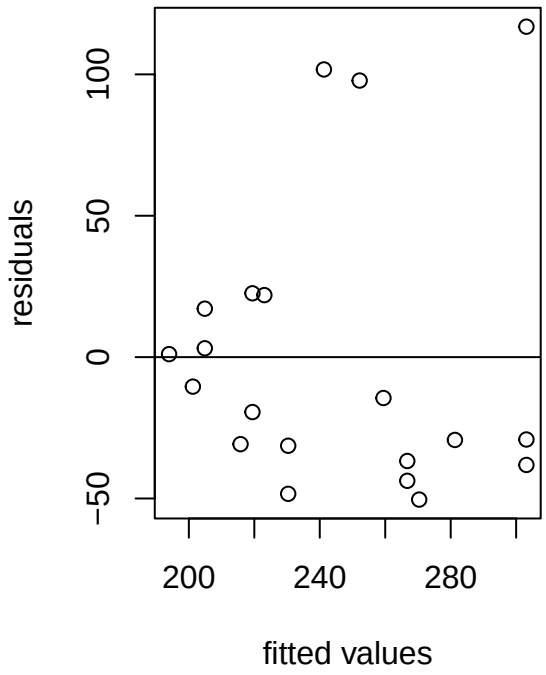
```
##        fit      lwr      upr
## 1 157.5554 91.60076 223.5101
```

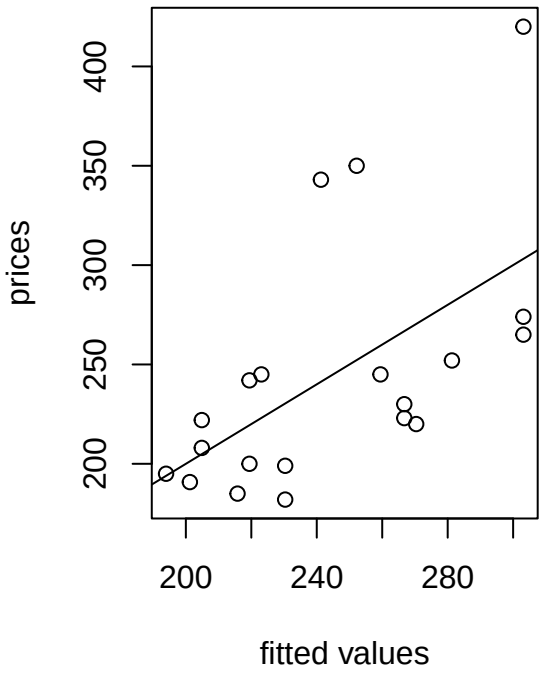
```
predict(lr_fit, data.frame(x = x0), interval="prediction")
```

```
##        fit      lwr      upr
## 1 157.5554 29.82253 285.2884
```

Residuals

```
y_hat = lr_fit$fitted.values

residuals <-  y - y_hat

par(mfrow = c(1, 2))
plot(lr_fit$fitted.values, residuals, xlab = "fitted values", ylab = "residuals")
abline(0,0, col="black")
plot(lr_fit$fitted.values, y, xlab = "fitted values", ylab = "prices")
abline(0,1, col="black")
```





Apparts data Marseille without "luxury" apparts

```
y_new = y[y<=300]
x_new = x[y<=300]

lr_fit_new = lm(y_new ~ x_new)
y_hat_new = lr_fit_new$fitted.values

beta_0 = lr_fit_new$coefficients[1]
beta_1 = lr_fit_new$coefficients[2]

summary(lr_fit_new)$coefficients
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 66.655234 36.9729083 1.802813 0.091541417
## x_new        2.134513  0.5030252 4.243351 0.000708159
```

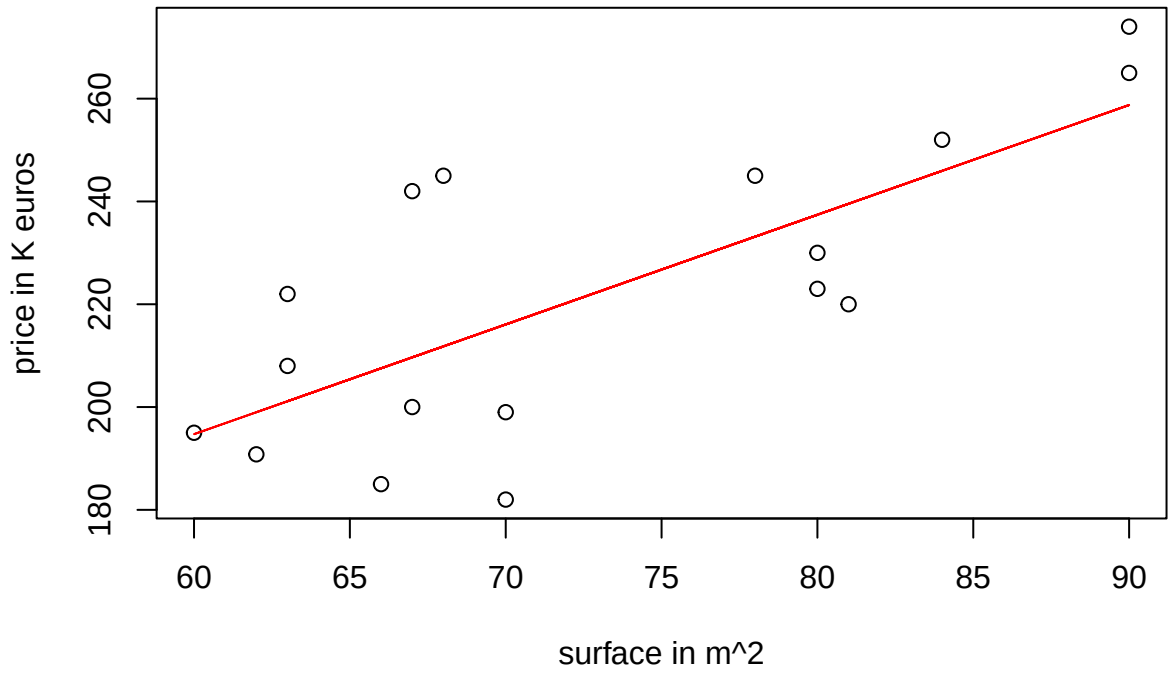
```
summary(lr_fit_new)$r.squared
```

```
## [1] 0.5455375
```

```
confint(lr_fit_new)
```

```
##                 2.5 %     97.5 %
## (Intercept) -12.15065 145.461122
## x_new         1.06234   3.206685
```

```
plot(x_new, y_new, xlab= "surface in m^2",ylab = "price in K euros")
lines(x_new, y_hat_new, col="red")
```



model-1.pdf

```
x0 <- 50
predict(lr_fit_new, data.frame(x_new = x0), interval="confidence")
```

```
##        fit      lwr      upr
## 1 173.3809 146.8096 199.9521
```

```
predict(lr_fit_new, data.frame(x_new = x0), interval="prediction")
```

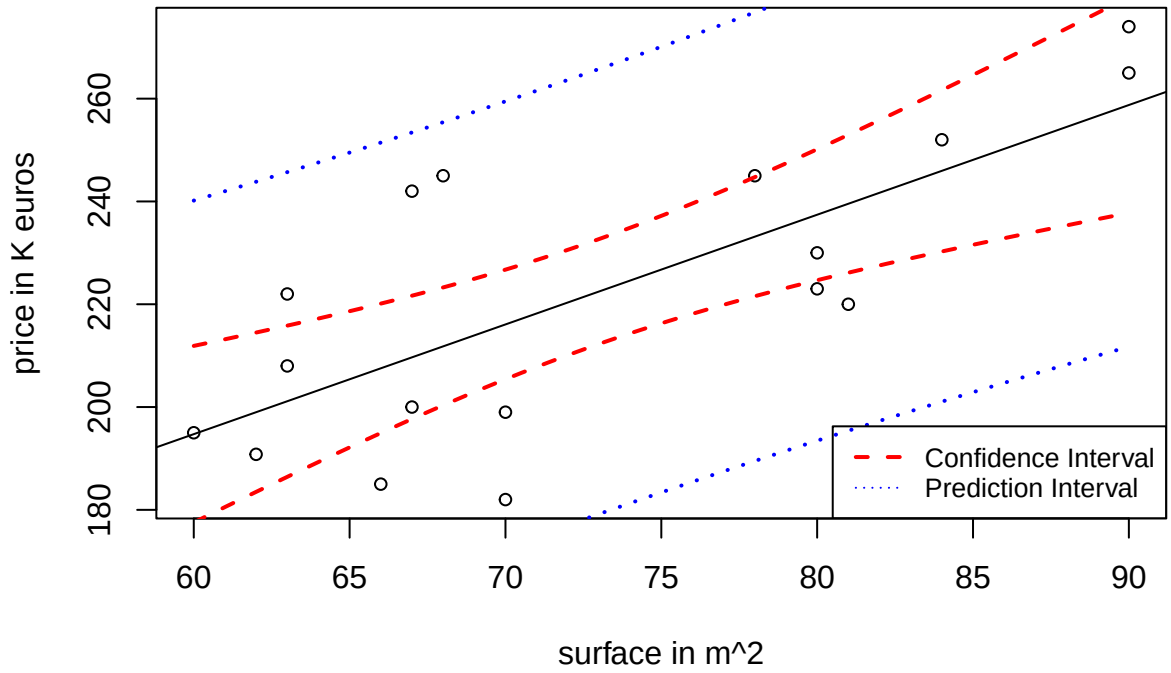
```
##        fit      lwr     upr
## 1 173.3809 123.6207 223.141
```

Confidence Interval

```
seqx <- seq(min(x_new),max(x_new),length=1000)

#apparts <- data.frame(surfaces, prices)
apparts <- data.frame(x_new, y_new)


Pred_Int <- predict(lr_fit_new, data.frame(x_new = seqx), interval="prediction")[,c("lwr","upr")]
Conf_Int <- predict(lr_fit_new, data.frame(x_new = seqx), interval="confidence")[,c("lwr","upr")]
summary(Conf_Int)
```

```
##       lwr             upr
##  Min.   :177.6   Min.   :211.9
##  1st Qu.:199.0   1st Qu.:222.5
##  Median :216.3   Median :237.2
##  Mean   :213.1   Mean   :240.4
##  3rd Qu.:228.2   3rd Qu.:257.3
##  Max.   :237.8   Max.   :279.8
```

```
plot(y_new~x_new, xlab= "surface in m^2",ylab = "price in K euros",  cex=0.8) #,xlim = c(min(x_new),max
abline(lr_fit_new$coefficients[1],lr_fit_new$coefficients[2])
matlines(seqx, cbind(Conf_Int, Pred_Int),lty=c(2,2,3,3), col=c("red","red","blue","blue"),lwd=c(2,2))
legend("bottomright",lty=c(2,3),lwd=c(2,1), c("Confidence Interval","Prediction Interval"),col=c("red",
```

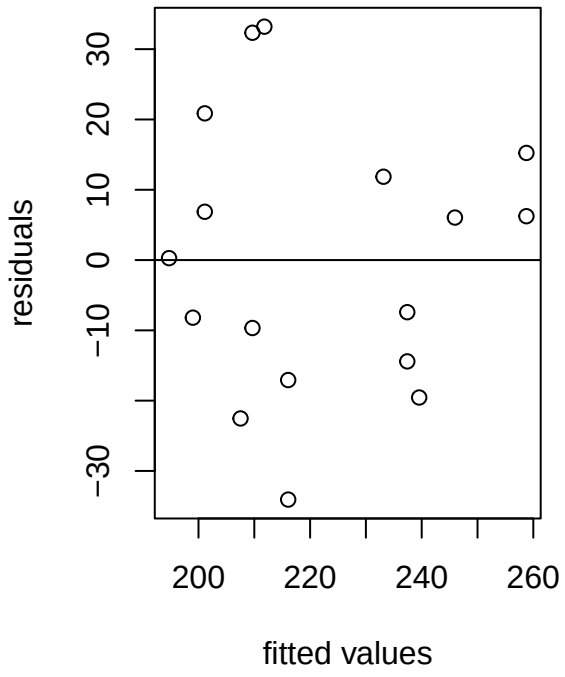


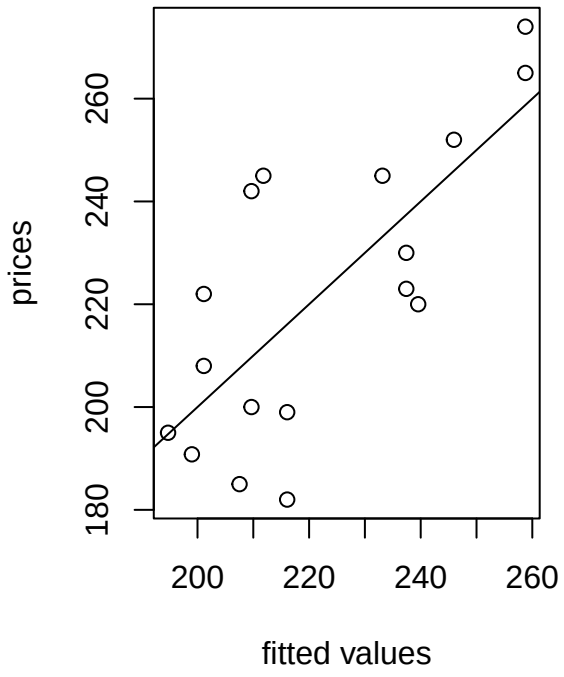
new model-1.pdf

```
#lines(lr_fit_new$fitted.values)
#matlines(Pred_Int)
```

Residuals

```
y_hat = lr_fit_new$fitted.values

residuals <-  y_new - y_hat

par(mfrow = c(1, 2))
plot(lr_fit_new$fitted.values, residuals, xlab = "fitted values", ylab = "residuals")
abline(0,0, col="black")
plot(lr_fit_new$fitted.values, y_new, xlab = "fitted values", ylab = "prices")
abline(0,1, col="black")
```





heterogeneity in the data ?

```
X = matrix(c(surfaces, prices),ncol = 2)
K=2
sol = kmeans(X, K)
plot(X[,1],X[,2], col = factor(sol$cluster), xlab= "surface in m^2",ylab = "price in K euros", main = "
```