Third Question: Define a linear model for an athlete in the 1500m

Arnau Abella

24/03/2020

Picking the best model

What is the linear expression that better predicts the behaviour of an athlete of 1500m ?

```
# Load the dataset and preprocess it.
library(FactoMineR)
data(decathlon)
head(decathlon)
colnames(decathlon)[c(1,5,6,10)]<-c("x100m","x400m","x110m.hurdle","x1500m")
colnames(decathlon)
```

Let's construct some simple linear regression models and check which better predicts the behaviour:

```
reg1<-lm(x1500m~x100m,data=decathlon)
summary(reg1)
```

```
##
## Call:
## lm(formula = x1500m ~ x100m, data = decathlon)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.005  -9.105  -1.706   5.624  37.604
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  308.578     78.038   3.954 0.000314 ***
## x100m         -2.687      7.094  -0.379 0.706885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.8 on 39 degrees of freedom
## Multiple R-squared:  0.003666,   Adjusted R-squared:  -0.02188
## F-statistic: 0.1435 on 1 and 39 DF,  p-value: 0.7069
```

```
reg2<-lm(x1500m~x110m.hurdle,data=decathlon)
summary(reg2)
```

```
##
## Call:
## lm(formula = x1500m ~ x110m.hurdle, data = decathlon)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.226  -7.804  -0.702   5.653  37.646
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  265.4584    57.8566   4.588 4.55e-05 ***
## x110m.hurdle   0.9288     3.9592   0.235    0.816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.81 on 39 degrees of freedom
## Multiple R-squared:  0.001409,   Adjusted R-squared:  -0.0242
## F-statistic: 0.05504 on 1 and 39 DF,  p-value: 0.8157
```

```
reg3<-lm(x1500m~x400m,data=decathlon)
summary(reg3)
```

```
##
## Call:
## lm(formula = x1500m ~ x400m, data = decathlon)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0877  -6.9098  -0.7062   4.7360  31.5996
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   74.102     73.424   1.009  0.31909
## x400m          4.130      1.479   2.792  0.00808 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.79 on 39 degrees of freedom
## Multiple R-squared:  0.1666, Adjusted R-squared:  0.1452
## F-statistic: 7.793 on 1 and 39 DF,  p-value: 0.008078
```

We are going to use the third model **x1500~x400m** because it has smaller residual standard error, larger R^2 (better fit) and better F-statistic.

Correlation Tests

Only in the third model the coefficient of correlation 0.408 is significant ($p < 0.05$)

```
# In all cases the coefficient of correlation is 0.816 and significant (p=0.002).
cor.test(decathlon$x100m       , decathlon$x1500m)
```

```
##
##  Pearson's product-moment correlation
##
## data:  decathlon$x100m and decathlon$x1500m
## t = -0.37881, df = 39, p-value = 0.7069
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3614639  0.2517942
## sample estimates:
##         cor
## -0.06054645
```

```
cor.test(decathlon$x110m.hurdle, decathlon$x1500m)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  decathlon$x110m.hurdle and decathlon$x1500m
## t = 0.2346, df = 39, p-value = 0.8157
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2732662  0.3412495
## sample estimates:
##        cor
## 0.03754024
```

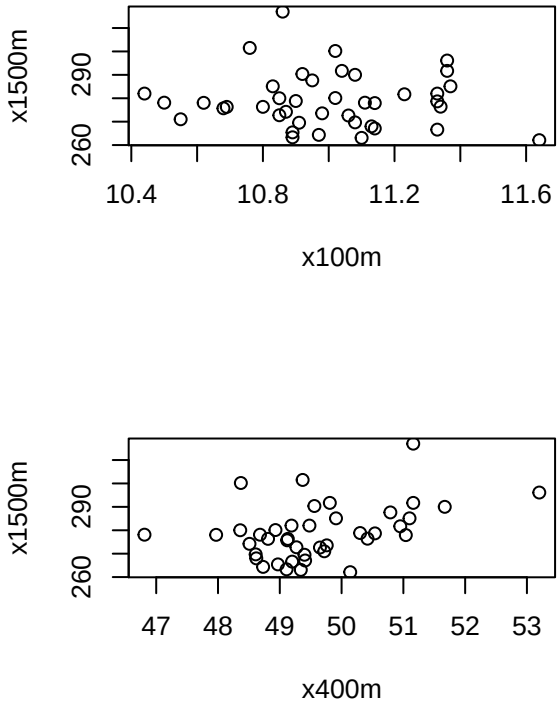
```
cor.test(decathlon$x400m       , decathlon$x1500m)
```

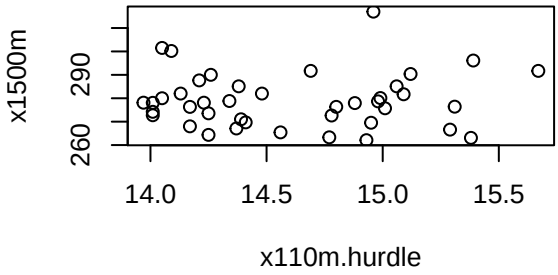
```
##
##  Pearson's product-moment correlation
##
## data:  decathlon$x400m and decathlon$x1500m
## t = 2.7917, df = 39, p-value = 0.008078
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1148796 0.6359151
## sample estimates:
##       cor
## 0.4081064
```

Scatterplots

The first and the second model are totally scattered but on the third model we can appreciate a positive correlation.

```
op<-par(mfrow=c(2,2))
plot(x1500m~x100m       , data=decathlon)
plot(x1500m~x110m.hurdle, data=decathlon)
plot(x1500m~x400m       , data=decathlon)
par(op)
```



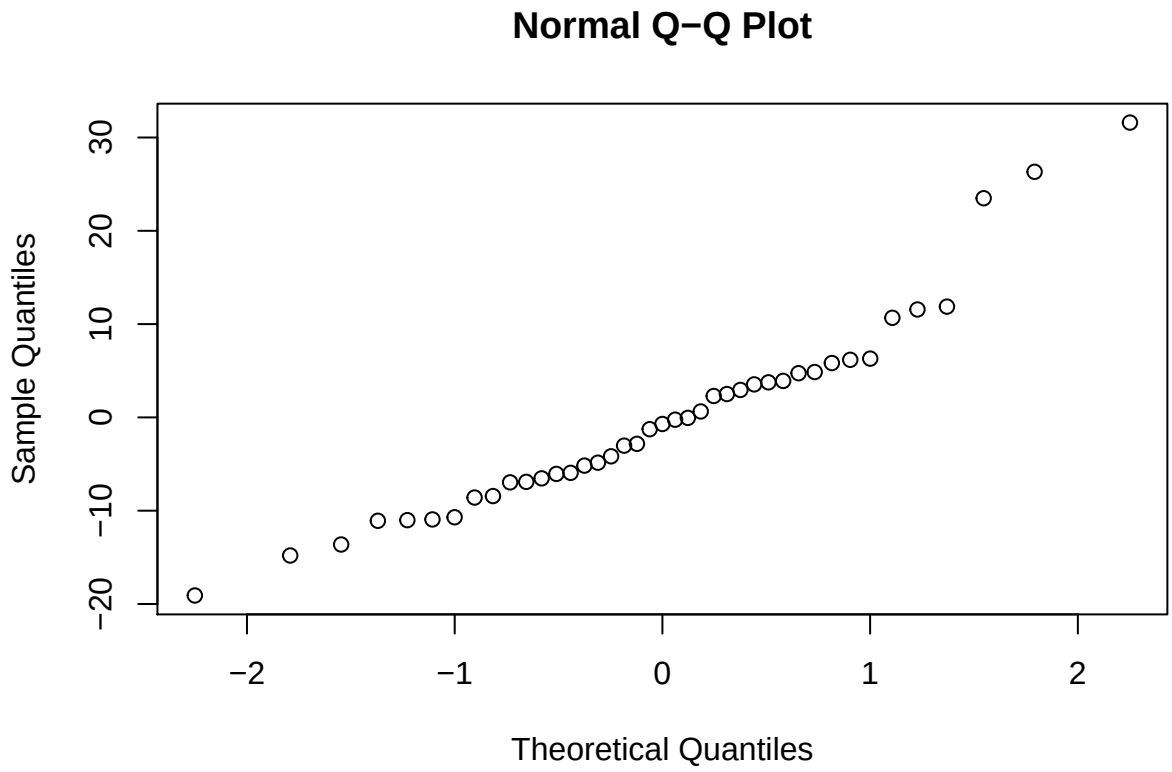


Testing assumptions of the linear model

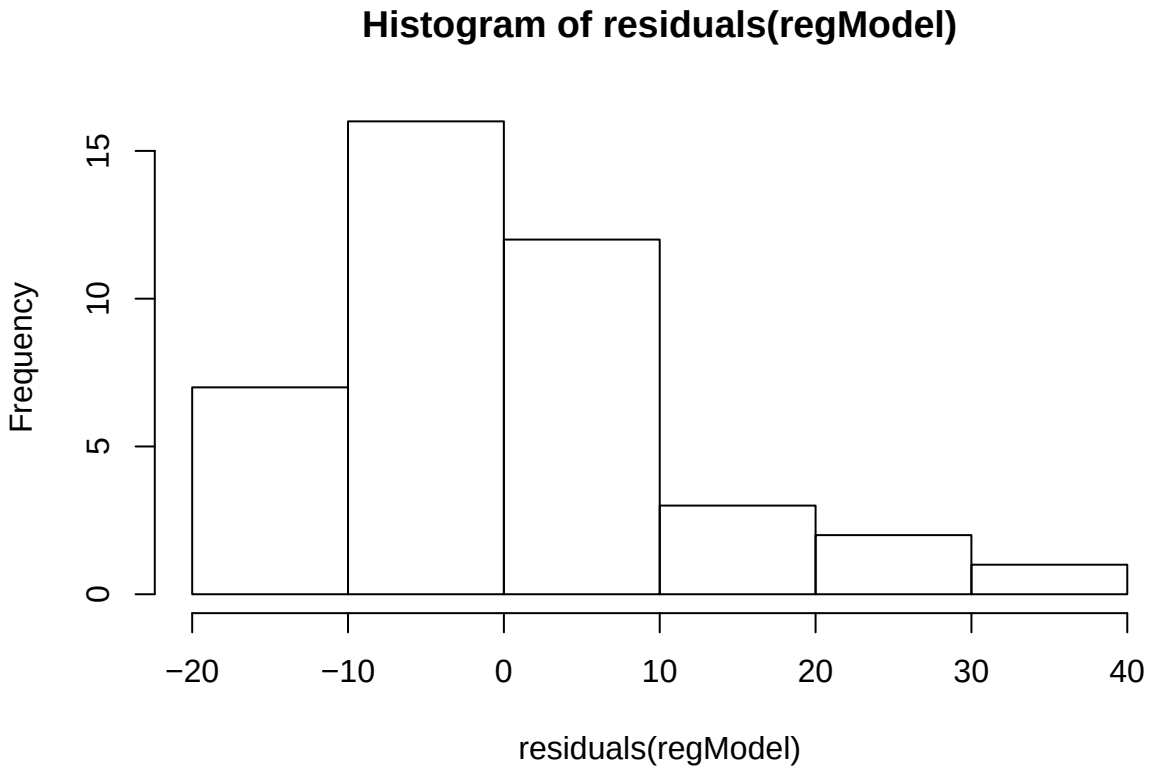
```
regModel <-lm(x1500m~x400m, data=decathlon)
summary(regModel)
```

Normality of the Error Term

```
# QQPlot
qqnorm(residuals(regModel))
```




```
# Since the values are taking part close to the diagonal,
#   the distribution is approximately normal.

# Histogram
hist(residuals(regModel))
```



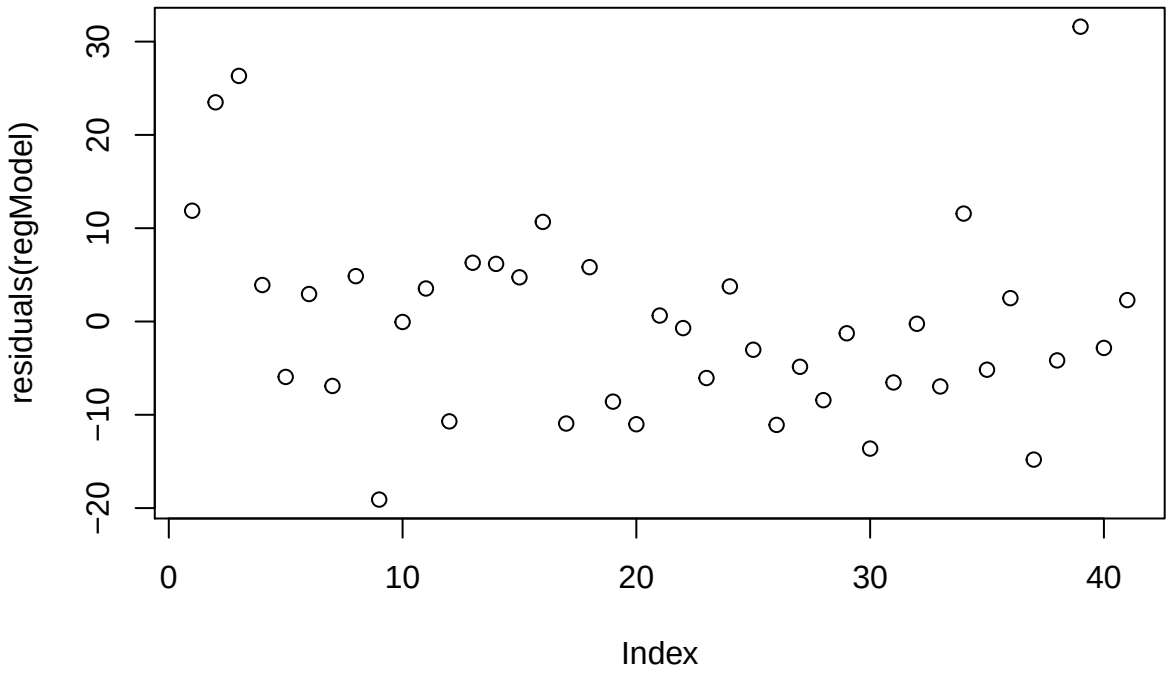

```
# It is approximately normal (skew to the left).

# Shapiro Wilks Test
shapiro.test(residuals(regModel))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(regModel)
## W = 0.93244, p-value = 0.01742
```

```
# The error term doesn't follows a Normal distribution. (p<0.05)
# This should be taken into consideration.
```

Homogenity of Variance

```
# Residual Analysis #
plot(residuals(regModel))
```




```
# Residuals have a rectangular pattern around the zero mean.
# There is no violation of this assumption.

##Breusch Pagan Test
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(regModel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  regModel
## BP = 0.0010727, df = 1, p-value = 0.9739
```

```
# H0 is accepted (p>0.05). Hence, the homogenity of variances is provided.
```

The independence of errors

```
# Durbin-Watson Test
dwtest(regModel, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  regModel
## DW = 1.7274, p-value = 0.3458
```

```
## alternative hypothesis: true autocorrelation is not 0
```

```
# There is not an autocorrelaiton in the data set (p>0.05).
# The errors/observations are independent.
```

Predicting new values

Is the model accurate ? What do you expect ?

The F test shows that the model is significant ($p<0.05$).

```
summary(regModel)
```

```
##
## Call:
## lm(formula = x1500m ~ x400m, data = decathlon)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0877  -6.9098  -0.7062   4.7360  31.5996
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   74.102     73.424   1.009  0.31909
## x400m          4.130      1.479   2.792  0.00808 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.79 on 39 degrees of freedom
## Multiple R-squared:  0.1666, Adjusted R-squared:  0.1452
## F-statistic: 7.793 on 1 and 39 DF,  p-value: 0.008078
```

```
confint(regModel)
```

```
##                   2.5 %     97.5 %
## (Intercept) -74.412562 222.616246
## x400m         1.137685   7.122619
```

```
# The null hypothesis is H0:B1=0.
# If the confidence interval includes 0 => we accept the null hypothesis.
#
# (1.137685, 7.122619) the confidence intervals of the parameters does not include 0.
#      => The null hypothesis B0=0 and B1=0 are rejected.
#
# Therefore the coefficients are significant.
```

Let's predict the behaviour of an athlete in the 1500m that runned the 400m in 55.5 seconds.

```
new=data.frame(x400m=55.5)
predict.lm(regModel, newdata=new, interval="prediction")
```

```
##        fit      lwr      upr
## 1 303.3253 275.0733 331.5773
```

The model predicted that the athlete would run the 1500m in between (303.32, 331.57) seconds with a high probability.