SMDE First Assignament

Chi-Square, Anova, Linear Models and PCA

Alejandro Montero Rivero
18-10-2016

Index

1. - Chi-Square

The main objective of this section is to prove if two samples follow the same distribution using the Chi-Square test (X^2). Initially a spreadsheet will be used to generate a Normal distribution, in general, the probability density function of this distribution is the following:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{(x-\mu)^2/(2\sigma^2)}$$

To simplify the problem the set of already created functions of excel and R are going to be used to create a sample of 200 observations, then R will be used to test the independence of the observations. The histograms of the initial observations are shown in image 1.

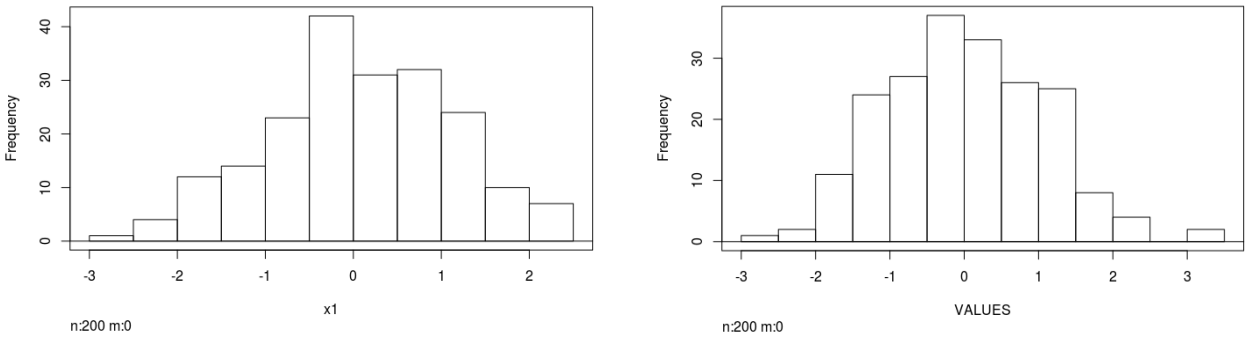


Figure 1: Histograms of Spreadsheet and R samples

As can be seen, the two samples follow a clear normal distribution, though visual observation is not enough to demonstrate it. Then, in R, the observations are divided in at minimum, five intervals, in these case of distance 0.5. To do so we use the R instruction "transform"

```
Tblr_v1_trans=transform(tbl_v1, cat = ifelse (x1 < -1, "-1",
                                              Ifelse(x1 < -0.5,"-0.5")))
```

Each "Ifelse" section corresponds to one of the intervals, in these document we only present a sample of the code to make it more readable. In the real test a total of five different intervals with are used to perform the test.

Next step is to create the contingency table, to do so we use the following R code:

```
Tbl_freq=as.data.frame(with(tbl_v1_trans,table(cat)))
Tbl <- cbind(Tbl_freq_R[,"Freq"],tbl_freq_Excel[,"Freq"])
```

Finally we run the Chi-Square test:

```
Chisq.test(tbl,correct=FALSE)


Pearson's Chi-squared test


data:  tbl
X-squared = 2.863, df = 5, p-value = 0.7211
```

The results of the test shows that X^2 value is 2.863 and the p-value 0.7211. As we are above the value 0.5 for the p-value we cannot reject the H0 that specifies the two samples are from the same distribution. In these case, as expected the test showed that the two observations even if created using two different techniques follow the same distribution.

1.2. - Comparison using five different samples

The next section aims to understand the reasoning behind X^2 test using other samples. To do so we are going to use the normal distribution alongside the exponential distribution. In total five different tests are performed with different construction parameters that may give a better insight on how the test works.

The density function of the exponential distribution follows the following formula:

$$Density = f(x) = \lambda e^{-\lambda z}$$

1.2.1. – Normal vs Normal. Similar Means

The fist tests is to compare two normal distributions with a similar mean but different standard deviation. One of the samples follow a normal distribution with mean=0 and sd=1 while the second one has a mean=0 and sd=2. The histograms of the two distributions are as follows:

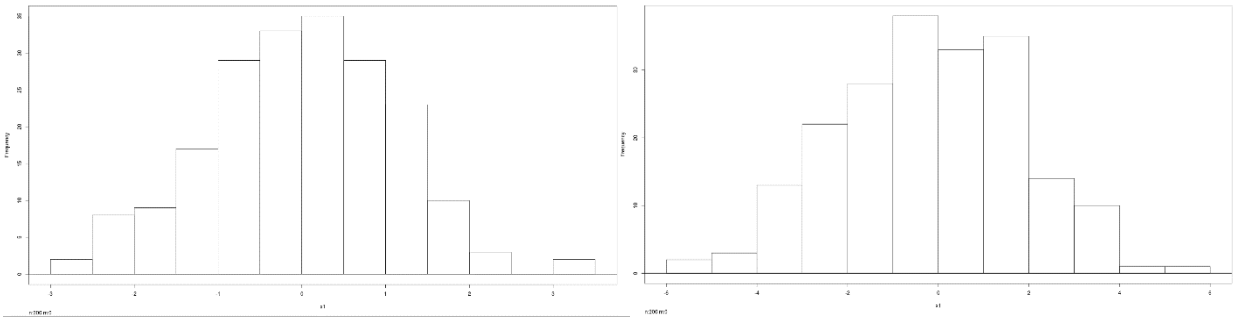


Figure 2. Histograms of two Normal distributions

As expected the two histograms show two samples very close to each other, the initial thought we have is that the two samples follow the very same normal distribution. Let's check it using the X^2 test:

```
      Pearson's Chi-squared test

data:  tbl2
X-squared = 4.918, df = 5, p-value = 0.6344
```

Initially we expected these result, having the same mean but slightly different standard deviation, usually the samples would be very close to each other and in these example the X^2 test demonstrates it.

It is really important to specify that all these values are randomly generated, these means that if we run the test several times we may encounter that some samples do not follow the same distribution. This behavior exemplifies why we cannot accept and affirmation with a single test, miss adjustments in the tools, wrong analysis techniques or simply, casualty may give us wrong information. In these example most test results in p-values greater than 0.5 though some give values close to 0.1, 0.2 or 0.3.

1.2.2. – Normal vs Normal. Different Means

In the second test we want to compare two normal distributions with a lot of difference between them. One of the samples follow a normal distribution with mean=1 and standard deviation=1. The second one has a mean=10 and standard deviation=15. The histograms of the two distributions are as follows:

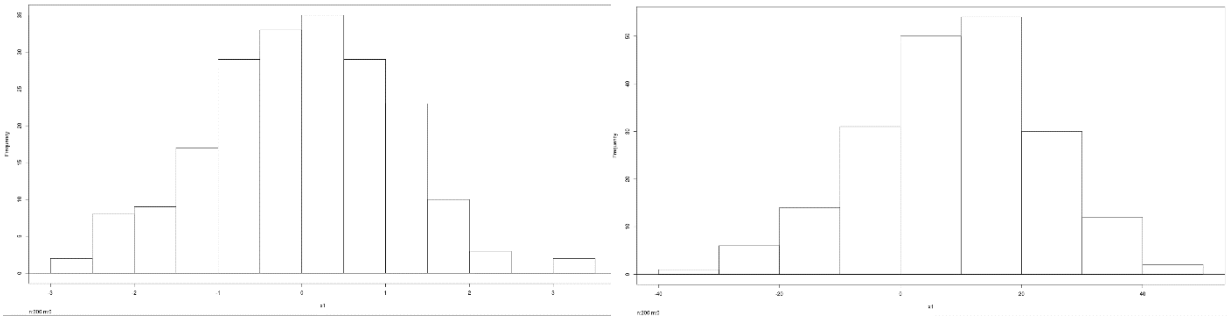


Figure 3. Histograms of two Normal distributions

Initially we see the two samples follow a really close pattern, though the numbers are quite different. In these case we cannot assume neither they are the same distribution or not. After performing the X^2 test we get the following result:

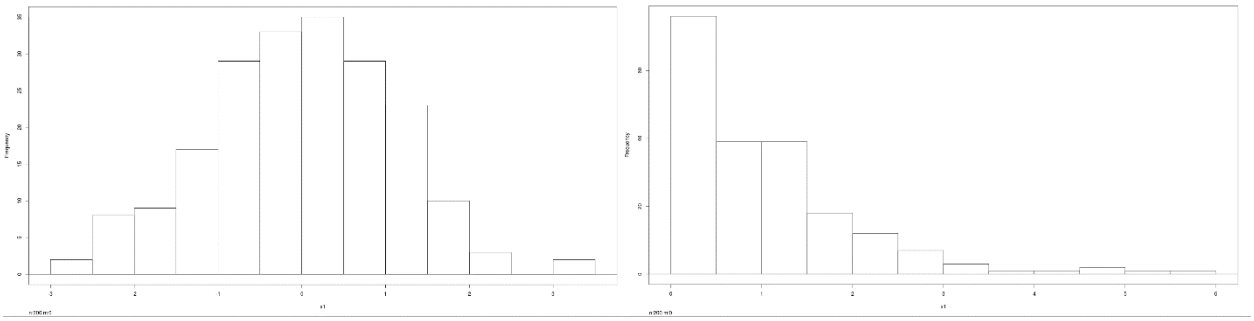
```
      Pearson's Chi-squared test

data:  tbl2
X-squared = 77.986, df = 5, p-value = 2.213e-15
```

Even though the two samples follow a normal distribution, the big differences between them in mean and standard deviation makes them completely different and the test demonstrates it.

In these case even if we perform the test several times we always see that the two samples never follow the same distribution. Taking into account the great variability the second sample has in each iteration due to the standard deviation and the great differences between the two means, it is highly improbable we may find a set of two samples that follow the same distribution. Though this conclusion is logical it is still advisable to perform the test several times to be sure no errors are present in the test.

1.2.3. - Exponential vs Normal

In these test we compare the initial R normal distribution with mean=0 and standard deviation=1 against an exponential distribution with rate=1. The histograms of the two distributions are as follows:



Initially we can observe the two distributions follow a very different shape, our initial thought is that the two samples do not follow the same distribution. After applying the X^2 test we get the following result:

```
      Pearson's Chi-squared test

data:  tbl3
X-squared = 40.425, df = 5, p-value = 1.226e-07
```

As before, the p-value is lower than 0.5 so we can assume the two distributions are not equal.

These test is perhaps one of the most interesting ones, X^2 can be used to check whether two samples follow the same distribution or not, in these case, logically, a normal sample will never follow the same distribution as an exponential sample. These example not only help us understand how the test works but also it gives a correct answer we already knew, demonstrating the test works as intended.

1.2.4. –Exponential vs Exponential. Similar Rates

This last tests aims to see if two exponential samples with small changes follow the same exponential distribution. We have seen before that even two samples of normal dist. with enough differences may not follow the same distribution, so let's check how exponential performs. The first sample has a rate=1 and the second one a rate=1.5. The histograms of the two distributions are as follow:

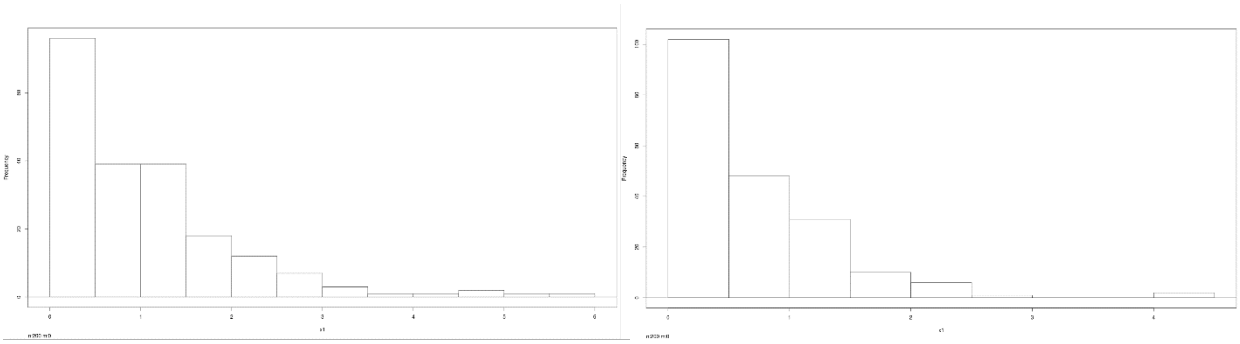


Figure 4. Histograms of two similar Exponential distributions

```
      Pearson's Chi-squared test

data:  tbl2
X-squared = 77.986, df = 5, p-value = 0.5535
```

The histograms show two samples that follows a clear exponential distribution, let's see what results in performing the X^2 test:

In these case and as expected, the p-value is greater than 0.5 meaning the two samples follow the same distribution.

As with the test comparing two normal samples we have to accept these result with a pinch of salt due to, in some other iterations and with the same constructor values we may find two samples that do not follow the same distribution. As before this is an occurrent behavior of randomized samples. Mostly, all iterations seem to be the same distribution though.

1.2.5. – Exponential vs Exponential. Different rates

In contrast with the previous test we now aim to see how the X^2 responds to a two samples of the exponential distribution but with noticeable rate differences. The fist sample is our well known exponential sample of rate=1 while the second one has a rate of 15. The following histograms shows the shape of the exponential samples:

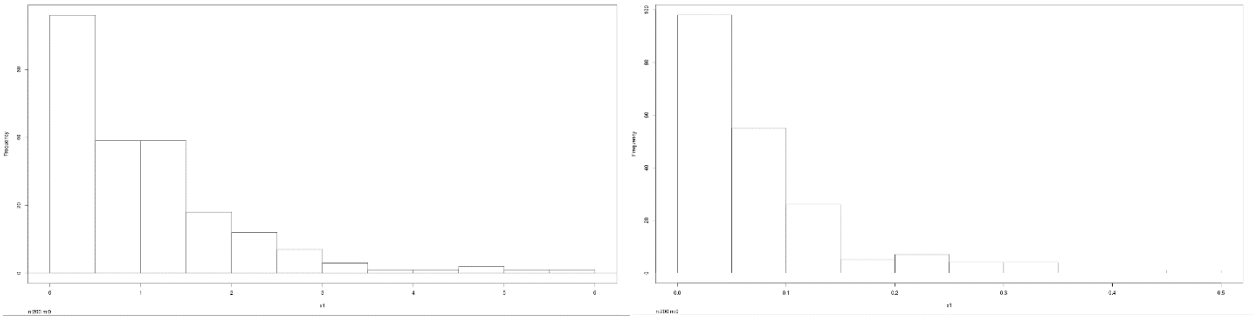


Figure 5. Histograms of two Exponential distributions with different rates

Logically though the shapes are mostly the same, we can see the maximum values are quite larger in the second sample, as a result, we would initially think the two samples are two different distributions. Let's check it with X^2 test:

```
      Pearson's Chi-squared test

data:  tbl6
X-squared = 93.739, df = 5, p-value < 2.2e-16
```

As expected the result of the test confirm the two samples are not the same distribution as we previously thought. As usual, to avoid possible misinterpretations, tools failure or casualty we would perform the test a few times. In these case though, we always see the two samples are different.

1.3. – Conclusions

Our first conclusion of this section are quite as expected, initially there was no reason to think a normal sample created in a spreadsheet would not be the same distribution as a sample created in R and the X^2 proved it.

The five test performed with five different samples gives a few interesting conclusions. First, if the two samples are created following a density function with close enough constructor values (mean and standard deviation for Normal distribution or rate for Exponential), X^2 will prove the two samples follow the same distribution. Though this is quite logical it's important to notice that the samples are created using randomization, therefore, running the test lots of times will eventually create two samples that do not follow the same distribution. This behavior results in two main conclusions:

- It's mandatory to perform an experiment several times. Human errors, hardware or software issues or causality may induce in an incorrect conclusion.
- Performing the same test several times with different samples and same constructor values may inquire how good the quality of the sample constructor is.

Third conclusion is that X^2 test is a really good tool to test whether our unknown samples follow the same distribution or not. Comparing a normal sample versus an exponential sample will always result in a failed X^2 test.

2. – ANOVA

In these section three samples of the normal distribution are created with different constructor values in order to check whether or not those samples belong to the same population. To do that analysis we are going to use ANOVA.

```
nexp1=data.frame(x1=rexp(200, rate=10),x2="exp1")
nexp2=data.frame(x1=rexp(200, rate=20),x2="exp2")
nexp3=data.frame(x1=rexp(200, rate=30),x2="exp3")
```

The first step is to generate three samples of the exponential distribution with different rates: $\lambda = 10, \lambda = 20, \lambda = 30$ and 200 observations each. The samples are transformed into data frames to ease the next steps, a second column indicates which distribution

```
tbl=mergeRows(nexp1,nexp2,common.only = FALSE)
tbl=mergeRows(as.data.frame(tbl),nexp3,common.only = FALSE)
```

Second step is to merge al the samples in a single table:

Now the final step is to create the ANOVA model:

```
AnovaModel.1 <- aov(x1~x2, data=tbl)
```

2.1. – Initial analysis

Before continuing with the ANOVA test it's a very good idea to take a look at the samples, a good starting point is checking the summary of the combined tables:

```
       x1                x2
 Min.   :0.0000259   exp1:200
 1st Qu.:0.0123240   exp2:200
 Median :0.0364971   exp3:200
 Mean   :0.0589557
 3rd Qu.:0.0769263
 Max.   :0.5566571
```

We can also check the histograms of the three samples:

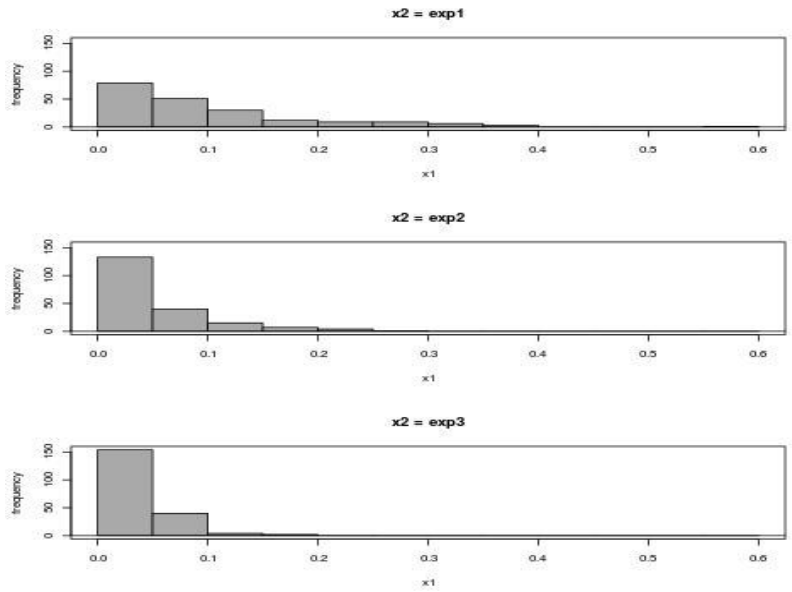


Figure 6. Histograms of the three samples

And the boxplots:

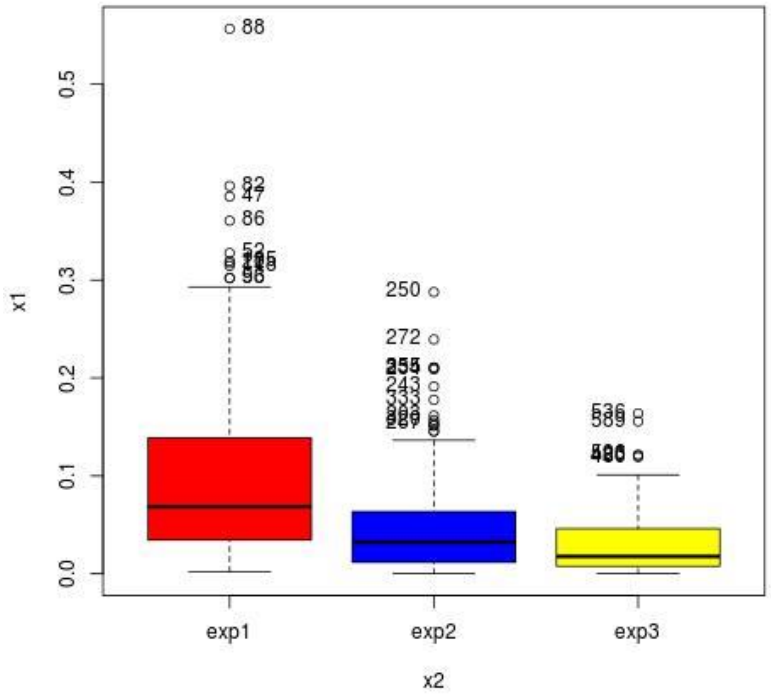


Figure 7. Boxplots of the three samples

As we can clearly see, the three samples follow an exponential behavior being this fact more pronounced in each next sample. Our main objective is to test with ANOVA if our initial hypothesis $H_0: \mu_0 = \mu_1 = \mu_2$ is true, and as we can see on both the boxplot and the summary the means of each sample are pretty close to each other, at a value around 0.3 ~ 0.9, this assumption may give us the idea the ANOVA test will succeed.

2.2. – ANOVA's assumptions

Before analyzing the model we have to test whether or not all the ANOVA assumptions are satisfied:

1. The population from the samples must be normal
2. The population from the samples must have equal variances.
3. All observations within each sample must be independent.

2.2.1. – Normality test

The first assumption is that the population of the samples must be all normal. We have to test the normality of all three samples, which are created by ourselves using the exponential distribution density function.

```
	Shapiro-Wilk normality test
data:  tbl1$x1
W = 0.80733, p-value = 5.494e-15
```

Inevitably the test fails as the sample is completely exponential. The same happens with the two remaining samples:

It is really important to note that the samples have a really small number of observations, though in reality a big enough sample would tend to the normality regardless of the distribution. In other words, in real world scenarios with lots of observations the normality test for ANOVA would not fail most of the times (central limit theorem).

```
	Shapiro-Wilk normality test
data:  tbl2$x1
W = 0.7252, p-value < 2.2e-16
```

```
	Shapiro-Wilk normality test
data:  tbl3$x1
W = 0.80463, p-value = 4.273e-15
```

2.2.2. – Homogeneity of variances

The result of these test easily predictable, as explained as the variance of the exponential distribution is : $V(x) = \frac{1}{\lambda^2}$ and as explained at beginning of this section all samples are created using a different λ. As a result, the homogeneity test, performed with bptest will fail.

```
        studentized Breusch-Pagan test

data:  AnovaModel.1

BP = 45.233, df = 2, p-value = 1.505e-10
```

2.2.3. – Independency of the observations

Finally the last assumption is whether or not the observations are correlated or not. In these case our H_0 is that the residuals of our model are uncorrelated.

```
        Durbin-Watson test

data:  AnovaModel.1

DW = 2.1088, p-value = 0.8948

alternative hypothesis: true autocorrelation is greater than 0
```

Having a p-value of 0.89 demonstrate us that effectively our observations are uncorrelated to one another. Though this conclusion was the obvious due to all observations where created using a randomized fashion, it is possible that certain random number generators have specific rules resulting in samples that have correlation between the observations. In our case, luckily it seem the random generator works as advertised creating samples with non-correlated observations.

2.3. – ANOVA test

In our example we were not lucky enough to pass all ANOVA's assumptions, this means the results we could get from the analysis cannot be formally accepted. Bottom line is, in our case ANOVA cannot conclude anything. However we can still analyze the data the model gives us.

Initially we specified in our initial Hypothesis H_0 that the three samples would have same mean, and in consequence the variance of the samples would be small. As seen in the previous three sample boxplot (Figure 7) the two lasts samples have a close meanwhile the first sample differs a little. After the variance test we can now see in the boxplot that the variances are clearly different in all three samples.

Though we cannot conclude anything, let's take a look into the ANOVA model.

```
             Df Sum Sq Mean Sq F value Pr(>F)
x2            2 0.3801 0.19005   51.25 <2e-16 ***
Residuals   597 2.2138 0.00371
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this summary we can extract interesting information:

- **SSB:** 0.3801
- **MSE**: 0.00371
- **F-statistic**: 51.25
- **P-value**: 0

Once again, even though we should not officially accept the ANOVA results we can see a very low p-value, as it is lower than 0.05 we would reject our hypothesis, this means our three samples do not share the same mean.

2.4. Conclusions

In these section we wanted to test if three different samples had the same population mean, so ANOVA was the best choice to test our hypothesis.

In order to accept the results of the diverse algorithms that forms the analysis a set of preconditions must be accomplished by the population: Independence of the observations, normality of the sample and homogeneity of variation. Though the observations are independent from one another, the low number of entries in Exponential sample where not enough to show a normal behavior, moreover, the variances in each sample where completely different so ANOVA cannot be officially accepted.

That being said ANOVA it's still a good tool to extract useful information that can give insight of the behavior of the samples and can be a good complementary information for other analysis.

3– Linear models

Regression analysis is a statistical process for estimating the relationship among variables. In other words, it helps understand how the value of a dependent variable changes when one of the independent variables varies while the others remain fixed. In this section we want to create a linear regression model that can predict how well an athlete will perform when running 1500m.

The data we have used is the Decathlon dataset, available in the FactoMineR package in R. As in the section 5 we want to validate the model, we will use only half of the dataset while the remaining half will be used for validations.

3.1 Initial model

The first step to find a suitable model for an athlete is to create one with the whole set of independent variables given the dependent variable. Obviously this will give us insight of which variables are really useful or not.

As we can see in the snippet below we have one variable (Rank) which does not seem to influence on the performance of an athlete when running 1500m, this can be seen with the number of "*" next to the variable, as more it has (up to three) the better it predicts the dependent variable.

```
Call:
lm(formula = X1500m ~ Discus + High.jump + Javeline + Long.jump +
    Points + Pole.vault + Rank + Shot.put + X100m + X110m.hurdle +
    X400m, data = decHalf)
Residuals:
     Min       1Q   Median       3Q      Max
-0.55765 -0.19509 -0.01648  0.11237  0.75949
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.375e+03  2.622e+01  52.437 1.94e-11 ***
Discus        3.537e+00  8.870e-02  39.878 1.72e-10 ***
High.jump     1.481e+02  3.800e+00  38.971 2.07e-10 ***
Javeline      2.390e+00  6.227e-02  38.381 2.33e-10 ***
Long.jump     3.974e+01  8.817e-01  45.075 6.48e-11 ***
Points       -1.604e-01  2.545e-03 -63.019 4.47e-12 ***
```

```
Pole.vault    4.986e+01  1.190e+00  41.900 1.16e-10 ***
Rank          1.413e-01  8.711e-02   1.622    0.143
Shot.put      9.380e+00  3.772e-01  24.868 7.31e-09 ***
X100m        -3.699e+01  7.939e-01 -46.593 4.98e-11 ***
X110m.hurdle -1.861e+01  5.448e-01 -34.162 5.89e-10 ***
X400m        -7.762e+00  3.264e-01 -23.779 1.04e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5088 on 8 degrees of freedom
Multiple R-squared:  0.9991,  Adjusted R-squared:  0.998
F-statistic: 852.3 on 11 and 8 DF,  p-value: 5.002e-11
```

3.2. Final model

The next step is to create the model, this time without the useless variables. The snippet below shows the resulting model.

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.48275 -0.29738  0.00307  0.16418  0.98081

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.376e+03  2.848e+01   48.32 3.49e-12 ***
Discus        3.453e+00  7.834e-02   44.08 7.95e-12 ***
High.jump     1.445e+02  3.343e+00   43.21 9.50e-12 ***
Javeline      2.356e+00  6.373e-02   36.97 3.84e-11 ***
Long.jump     3.916e+01  8.737e-01   44.82 6.85e-12 ***
Points       -1.592e-01  2.641e-03  -60.26 4.81e-13 ***
Pole.vault    4.851e+01  9.235e-01   52.53 1.65e-12 ***
```

```
Shot.put      9.383e+00  4.100e-01   22.89 2.76e-09 ***
X100m        -3.682e+01  8.556e-01  -43.04 9.85e-12 ***
X110m.hurdle -1.837e+01  5.693e-01  -32.27 1.30e-10 ***
X400m        -7.600e+00  3.377e-01  -22.50 3.20e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.553 on 9 degrees of freedom
Multiple R-squared:  0.9989,  Adjusted R-squared:  0.9976
F-statistic: 793.5 on 10 and 9 DF,  p-value: 5.247e-12
```

The resulting model has almost all independent variables, this can conclude that any given athlete can perform fairly well if he does well in any other discipline, also, a good mental preparation with a proper speech can improve an athlete performance. Though this conclusion seems reasonable it is true that some disciplines may influence the dependent variable better than others, we would see that when we analysis this dataset with PCA (section 5).

Finally and most important, if we take a look at the R-squared value we see it is almost 1, this means the model we just created is really good for the given data. This is usually true for the training dataset but in reality it may not adjust as well in the real world, which is why it is mandatory to test it against well-known data that should give expected predictions (validation dataset) before using the model in production. In section 4 we are going to validate the model we just created using the the second half of the decathlon dataset as validation data.

3.3. Linear regression model assumptions

Before we can even test the model we have to make sure the model passes all assumptions. Similar to ANOVA the assumptions are as follows:

- Observations must be independent from each other.
- Population must be normal in each sample
- Population must have equal variances.

3.3.1. Normality test

Let's start with the normality test using Saphiro:

```
shapiro.test(residuals(RegModel2.2))

      Shapiro-Wilk normality test

data:  residuals(RegModel2.2)
W = 0.91695, p-value = 0.08655
```

As expected the p-value > 0.05 meaning the population follows a normal behavior. We passed the first assumption.

3.3.2. Auto-correlation test

Continuing with test the observation independency using Durbin-Watson:

```
dwtest(RegModel2.2)

      Durbin-Watson test

data:  RegModel2.2
DW = 0.98232, p-value = 0.005869
```

3.3.3. Variance Homogeneity test

Finally we test the homogeneity of variances using Bresuch-Pagant test:

```
bptest(RegModel2.2)

      studentized Breusch-Pagan test

data:  RegModel2.2
BP = 9.5217, df = 10, p-value = 0.4834
```

As the p-value we get from the test is greater than 0.05 we can affirm the data passes the test.

3.4. Model evaluation

To check whether the model is rich we can check two main concepts: Standard Error and the Slope.

3.4.1. Standard Error

We can compare the Residual error of the model against the sample mean of the dependent variable. In our case we have a Standard Residual Error (S_ε) of 0.553 while the mean is 279.304. We can confidently say the error extremely low and, as far as we know, the model seems rich.

3.4.2. Slope

When creating a model, the regression line should not be close to 0 if there is some sort of linear relationship between variables. More formally, we have to see if the slope in the linear function $Y = B_0 + B_1 * X$, in this case B_1, is far from 0. As we have seen previously, for all independent variables the t-value is never close to 0, with values ranging from ~-60 to ~+50. We can assume that a linear relation between variables those indeed exist.

3.4. Conclusions

The model we created in these section seems to fit really well the test data though it seems possible than the model overestimates which variables are capable of predict the performance of an athlete when running 1500m. This is even clearer when testing the independence test, as it seems many observations are related to each other.

In a real world scenario though, the model may not feet as well as it seems, we have to take into account that we are creating a model relatively to a specific dataset and obviously that model will tend to fit that very same data. Next step is to check the validness of the model making predictions with a second dataset from which we already know everything.

4. Model predictions

In the last section we created a model that aims to predict the behavior of an athlete when running 1500m, using a set on independent variables. In this section we want to realize predictions for the second half of the athletes from the decathlon dataset. We already have the real observed values for this competition so we want to check whether or not our predictions fits the real world observations and in therefore, if our model was well created.

4.2. Model prediction

The method "predict" in R helps to create a prediction based on a regression model and a specified data set. After loading the second half of the decathlon we can see the results of the prediction are as follows:

```
predict(RegModel2.2,newdata = decHalf2,interval="prediction")

             fit      lwr      upr
Bernard     275.9921 274.3881 277.5961
Smirnov     263.8927 262.2669 265.5186
Schwarzl    274.0269 272.3171 275.7367
Schoenbeck  278.9714 277.1903 280.7525
Casarsa     293.8514 291.5980 296.1048
Barras      266.4529 264.3821 268.5237
Nool        276.8903 275.2076 278.5729
Smith       271.4519 268.5816 274.3223
Averyanov   271.0242 268.7873 273.2611
Ojaniemi    276.1471 273.9895 278.3048
Qi          273.8184 272.3247 275.3121
Drews       274.0573 272.3799 275.7347
Parkhomenko 276.4315 273.9749 278.8881
Terek       291.4907 289.5526 293.4288
Gomez       268.3924 265.7147 271.0701
Turi        288.5770 286.3153 290.8387
Pogorelov   287.2996 285.2193 289.3799
```

```
Hernu       264.7436 263.2907 266.1966
Lorenzo     262.8656 260.9177 264.8136
Karlivans   279.1064 277.6276 280.5853
Korkizoglou 315.4889 313.3798 317.5980
Uldal       281.4501 279.6086 283.2917
```

If we take a look at the actual real observations we can see most of the predictions fit really well, this gives us the idea that the model is capable of predicting correctly real world scenarios and therefore that we could put the model in production.

To give even more insight we can make a plot to check even further if our predictions are accurate. We selected a confidence level of 95% to represent the vertical lines in each observation as well as the predicted value and the real observation value (green and red dotes respectively).

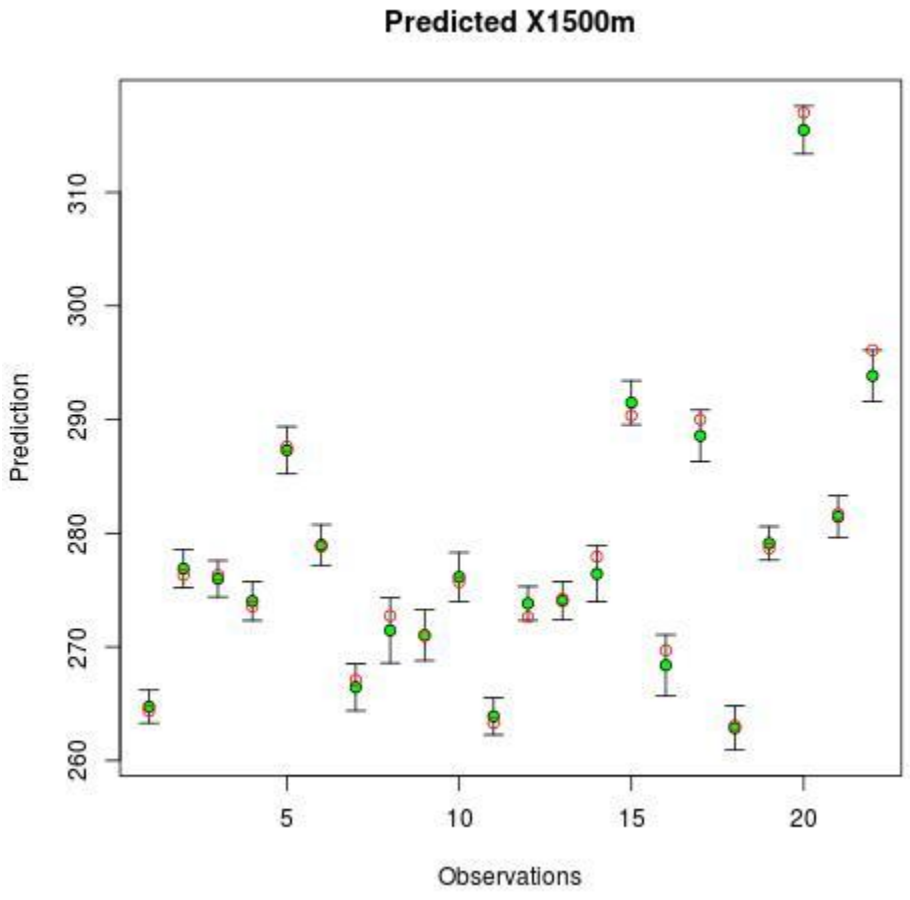


Figure 8. Predicted values

4.3. Conclusions

As we can clearly see the predicted values are all inside the 95% of confidence. Specifically this test has been done with the remaining half of a dataset that probably is correlated somehow to the first part resulting in a test that maybe does not fit the real world perfectly, thus, in a real model we should test the model lot further and check with other datasets with no kind of relation with the original dataset.

Moreover we could see the test does not pass all three Assumptions making these very same model inviable in a production scenario, even if it predicts correctly and surpasses all validation datasets.

5. PCA

Principal Component Analysis (PCA) is a statistical set of algorithms that can be used to reduce the dimensionality of a dataset by reducing the number of random variables under consideration via obtaining a set of principal variables. PCA can also provide the user of an overview using a simple graphical visualization of the information contained in the dataset. That graph and a simple results analysis is what we aim to obtain in these section.

In a real world scenario we may have to consider hundreds of variables alongside a great number of samples resulting in an extremely hard analysis, using PCA we can find which variables are really explanatory, focusing on those and creating a smaller size problem which can ease the analysis.

5.1 Variables description

As explained in the introduction of this section PCA is a useful tool for reduce the dimensionality of a very huge dataset, in our case though we are using a really simple one with just a few observations, the decathlon dataset. Even with a small amount of observations this dataset can give a good demonstration of what PCA is capable of.

In order to reduce the number of variables in the dataset we want to just use those that explains the behavior of an athlete in the most efficient manner. To do so, we want to keep those variables that accomplish the following conditions:

- Independent variables. Dependent variables are those that, as the name suggest, depend on the values of others, and in result that explains the same things, adding extra complexity that is not needed.
- Variables with big variance. We want to keep just those variables with value changes, static variables or constants tend to explain very few information, we want to discard those.

The second condition can easily checked using a plot called “eigenvalues”, R can plot:

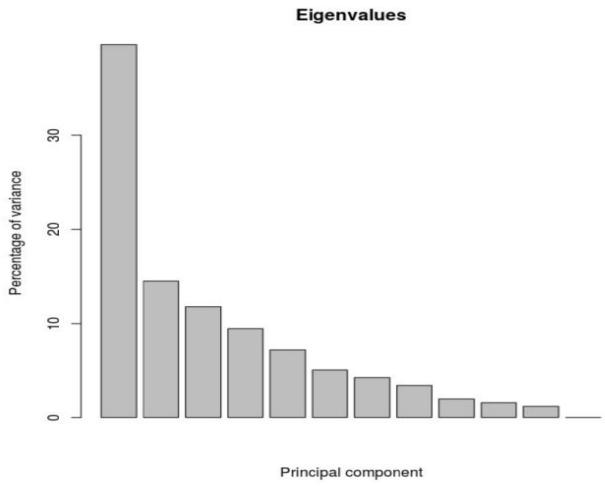


Figure 9. Decathlon Eigenvalues

The plot shows that more than 60% of the variance is explained by jus the first two principal components.

5.2. Factor map

One of the most interesting applications on PCA is to give the user a friendly and simple overview of the variable relationship using a two dimensional graph which can be seen in the following illustration:

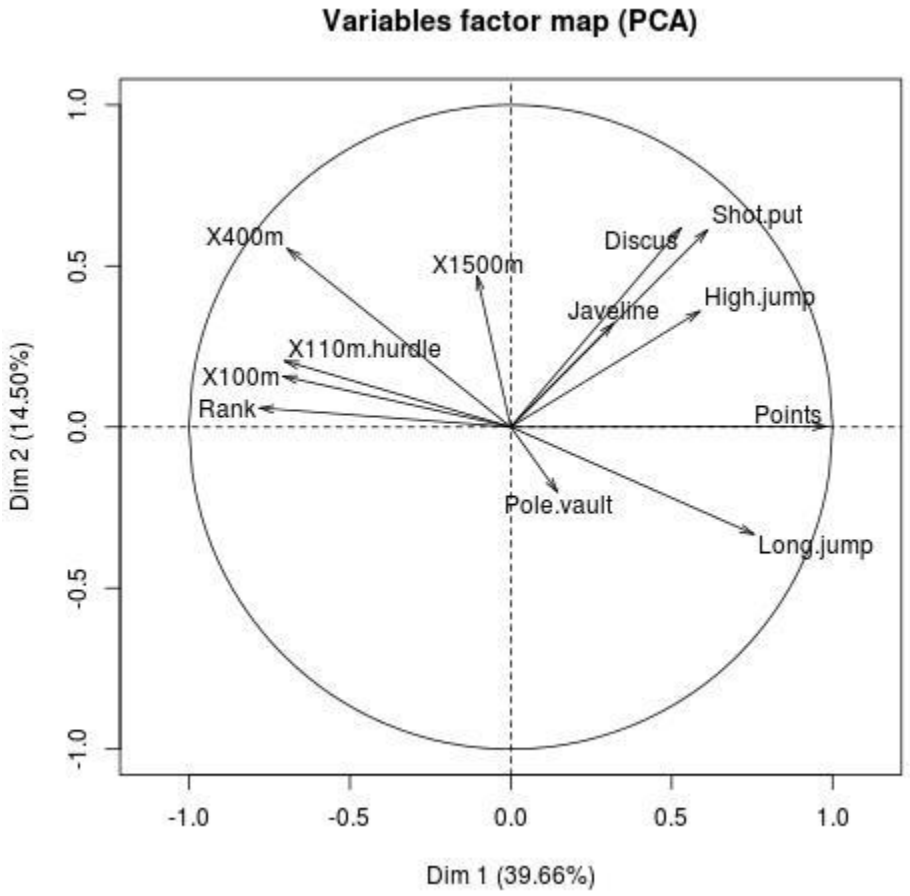


Figure 10. Factor Map

From these graph we can extract useful information:

- Point is the main explanatory variable in the whole dataset.
- Some variables as Javeline, Shotput, HighJump or Discuss are correlated to each other, this behavior can also be seen in variables such as Rank, X110m, X100m and X400m.
- Variables Points and Rank has negative correlation. A possible explanation is that as the lowest the rank of an athlete is, the higher the score is.

5.3. Dimensionality reduction

In this last section we want to use PCA to reduce the dimensionality of the dataset, to do so we can take a look at the contribution of each variable to the first two principal components of the dataset. To do so, we can sort the variables by contribution to the first and second dimensions:

```
res$var$contrib[,1:2]
                  Dim.1        Dim.2
X100m         10.5382802 1.427857e+00
Long.jump     12.0072309 6.369260e+00
Shot.put       7.8615458 2.154206e+01
High.jump      7.2626467 7.436236e+00
X400m         10.1582977 1.767542e+01
X110m.hurdle  10.3979678 2.415117e+00
Discus         5.8889946 2.199468e+01
Pole.vault     0.4408398 2.336074e+00
Javeline       2.1378946 5.955215e+00
X1500m         0.2312995 1.264716e+01
Rank          12.8233634 2.007797e-01
Points        20.2516390 1.397489e-04
```

Only those variables that have a contribution higher than 10 are considered worth of being main variables. Also, we want to take out those variables that are linearly dependent between them in order to avoid adding redundant information.

All in all we can consider the following variables as main variables:

- Points.
- Long.jump.
- X100m.

5.4 Conclusions

PCA is a really interesting tool to analyze really big datasets and avoid using useless information if possible models. As we could see in our study case, most of our initial variables have been eliminated in favor on just 3 out of 13 variables in contrast with the model created in section 4 that used 12 variables, that is to say, most of the dataset information.

It's really important to validate the PCA model as well, but as this is a non-parametric method the assumptions are really easy to analyze. We only have to take into account that the

number of observations must explain all the existing variables, ergo, the number of observations must be bigger than the number of observations. In the decathlon study we can safely say that the number of observations (42) is quite higher than the number of variables (13), so our model can be officially accepted, alongside our conclusions.

Finally, we could see that the main variable we were studying, X1500m, has very low importance in the overall dataset as it has very little variance. That variable neither has very strong correlation with any other variable.