# Exercise 2. Obtain an expression to simulate new data

Arnau Abella

21/05/2020

Image that you don't know anything regarding this dataset. You need to explore it because you want to define a model to obtain new data for your DOE (you want to detect the possible relations and the interactions between the factors, or maybe you want to test alternatives or predict future scenarios).

1. Explore the possible relations of all the factors and answer variable, you can use any technique developed during the course.
2. Describe what you find on this analysis and, explain if it is coherent with the knowledge you have from the data.
3. Propose an expression (as an example using a LRM) to understand the relations between the data. What are the factors that affects the answer?

In order to answer these three questions we are going to use the following techniques:

- Multiple Linear Regression Model
- Principal Component Analysis

## Multiple Linear Regression Model

Let's start by applying a *multiple linear regression model* to the generated dataset from the first exercise:

```r
reg_model1<-lm(answer~., data=dataset)
summary(reg_model1)
```

```
##
## Call:
## lm(formula = answer ~ ., data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1928 -0.7032 -0.0417  0.6934  3.2072
##
## Coefficients: (5 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.005562   0.077038   0.072    0.942
## factor1      1.031851   0.038688  26.671   <2e-16 ***
## factor2      0.987810   0.079949  12.355   <2e-16 ***
## factor3     -0.008692   0.011644  -0.746    0.455
## factor4      0.988821   0.037947  26.058   <2e-16 ***
## factor5      4.942126   0.079995  61.780   <2e-16 ***
## factor6            NA         NA      NA       NA
## factor7            NA         NA      NA       NA
## factor8            NA         NA      NA       NA
## factor9            NA         NA      NA       NA
## factor10           NA         NA      NA       NA
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.02 on 1994 degrees of freedom
## Multiple R-squared:  0.7327, Adjusted R-squared:  0.732
## F-statistic:  1093 on 5 and 1994 DF,  p-value: < 2.2e-16
```

The implementation of `lm` is clever enough to detect that the factors f6-f10 are a linear combination of the factors f1-5.

We can analyze them separately. As expected, only factor 6 and factor 9 have a linear relation with the answer.

```r
reg_model2<-lm(answer~factor6+factor7+factor8+factor9+factor10, data=dataset)
summary(reg_model2)
```

```
##
## Call:
## lm(formula = answer ~ factor6 + factor7 + factor8 + factor9 +
##     factor10, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1928 -0.7032 -0.0417  0.6934  3.2072
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.005562   0.077038   0.072    0.942
## factor6      1.024726   0.047754  21.459   <2e-16 ***
## factor7      0.014112   0.038168   0.370    0.712
## factor8      0.029929   0.057247   0.523    0.601
## factor9      0.995808   0.023845  41.762   <2e-16 ***
## factor10    -0.036916   0.075801  -0.487    0.626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.02 on 1994 degrees of freedom
## Multiple R-squared:  0.7327, Adjusted R-squared:  0.732
## F-statistic:  1093 on 5 and 1994 DF,  p-value: < 2.2e-16
```

The expression to compute the **answer** is the following $Ans = +0.005562 + 1.032 * f_1 + 0.988 * f_2 + 0.988 * f_4 + 4.94 * f_5$ which is a very good approximation of the one used to produce this random variables.

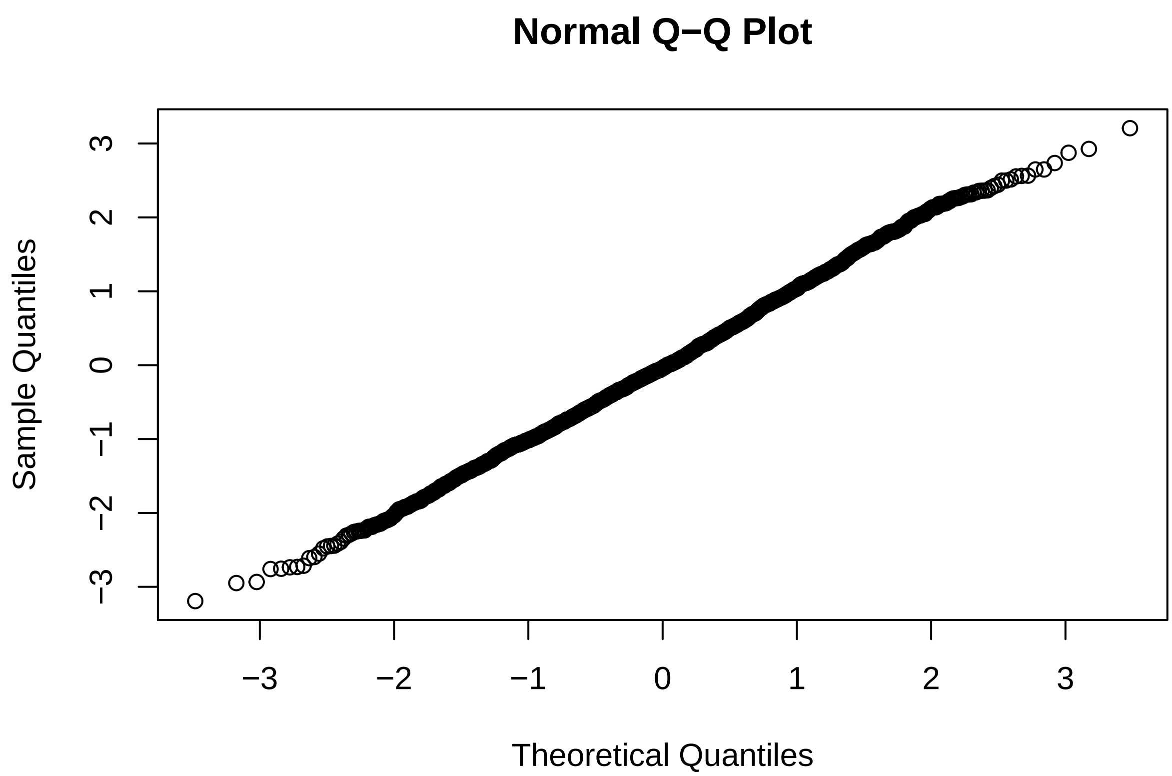
```r
reg_model3<-lm(answer~factor1+factor2+factor3+factor4+factor5, data=dataset)
summary(reg_model3)
```

```
##
## Call:
## lm(formula = answer ~ factor1 + factor2 + factor3 + factor4 +
##     factor5, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1928 -0.7032 -0.0417  0.6934  3.2072
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

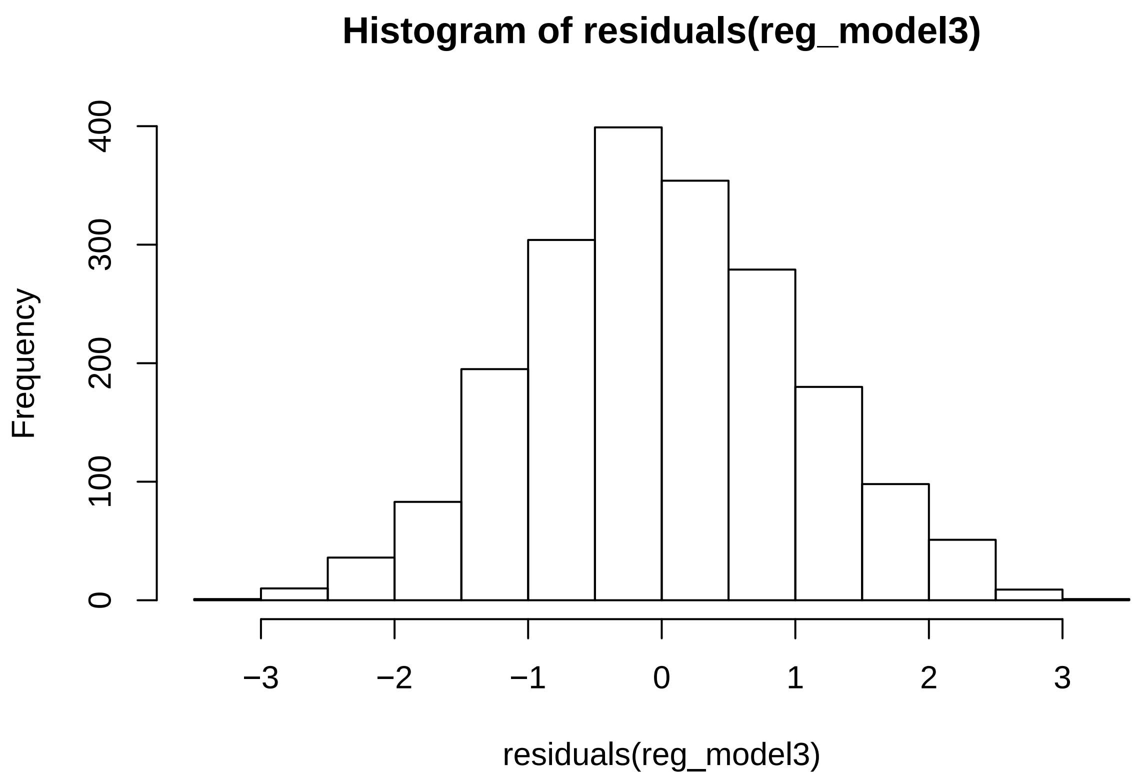
```
## (Intercept)  0.005562   0.077038   0.072    0.942
## factor1      1.031851   0.038688  26.671   <2e-16 ***
## factor2      0.987810   0.079949  12.355   <2e-16 ***
## factor3     -0.008692   0.011644  -0.746    0.455
## factor4      0.988821   0.037947  26.058   <2e-16 ***
## factor5      4.942126   0.079995  61.780   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.02 on 1994 degrees of freedom
## Multiple R-squared:  0.7327, Adjusted R-squared:  0.732
## F-statistic:  1093 on 5 and 1994 DF,  p-value: < 2.2e-16
```

### Testing Regression Assumptions

```r
### 1. Normality of the Error Term
# Using QQ plot
qqnorm(residuals(reg_model3))
```




```r
# Using Histogram
hist(residuals(reg_model3))
```

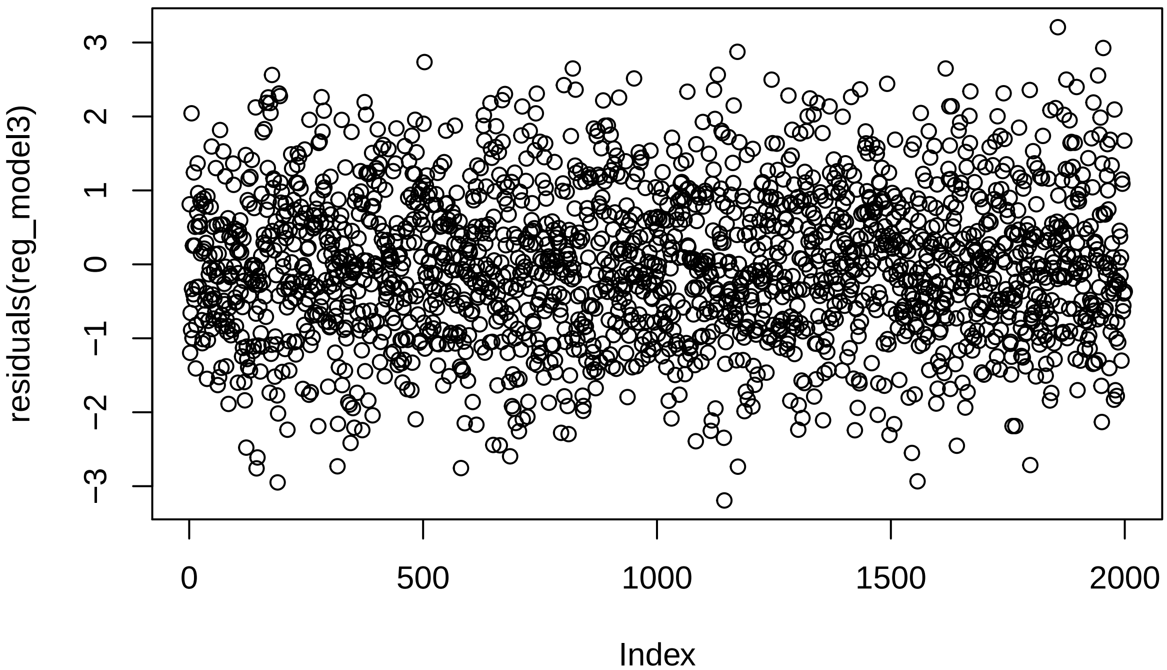



```
#Shapiro Wilks Test
shapiro.test(residuals(reg_model3))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(reg_model3)
## W = 0.99851, p-value = 0.0734
```

```
# H_0 is accepted: the error term does follows a Normal distribution (p > 0.05)

### 2. Homogenity of Variance ###
# Residual Analysis #
plot(residuals(reg_model3))
```




```r
##Breusch Pagan Test
bptest(reg_model3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg_model3
## BP = 2.5905, df = 5, p-value = 0.7628
```

```r
# H_0 is accepted (p>0.05): the homogenity of variances is provided.

### 3. The independence of errors ###
dwtest(reg_model3, alternative = "two.sided")
```

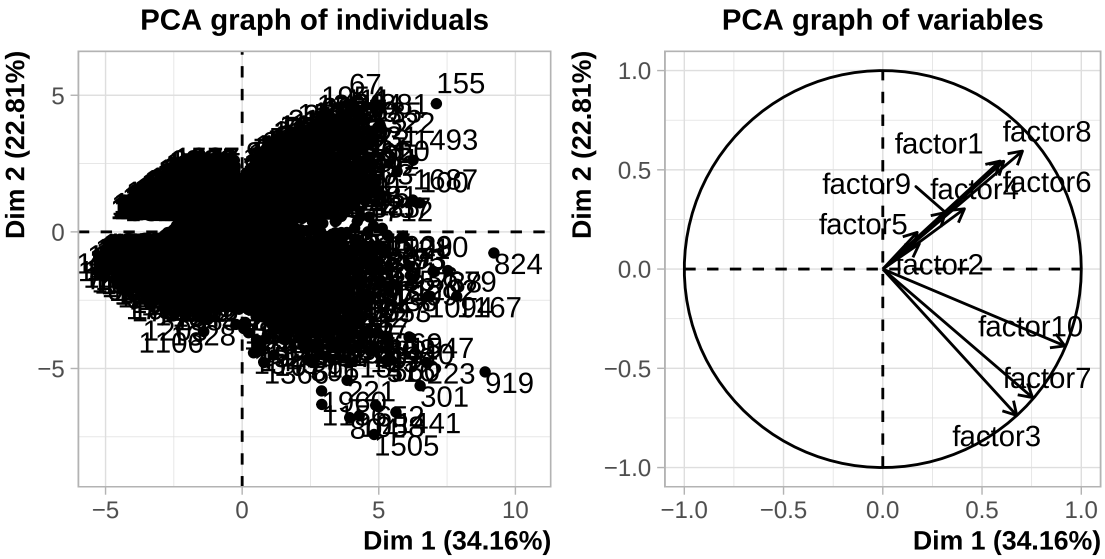
```
##
##  Durbin-Watson test
##
## data:  reg_model3
## DW = 1.9725, p-value = 0.5378
## alternative hypothesis: true autocorrelation is not 0
```

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

## Principal Component Analysis

Let's use **Principal Component Analysis** technique to analyze the dataset and its factors and extract an expression to predict an answer:

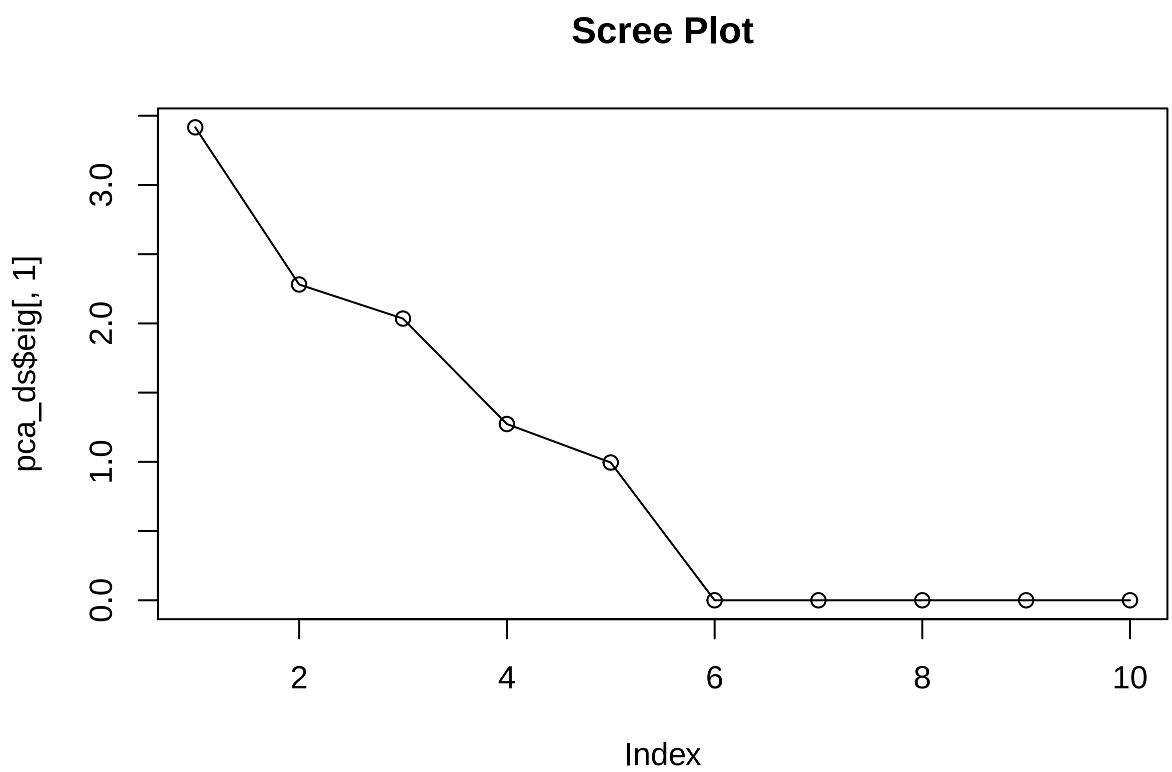
```
pca_ds<-PCA(dataset[,-answer]) # Remove the dependent variable score.
```




```
pca_ds$eig # cumulative percentage of variance > 75%
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1  3.416127e+00           3.416127e+01                          34.16127
## comp 2  2.280985e+00           2.280985e+01                          56.97113
## comp 3  2.034760e+00           2.034760e+01                          77.31873
## comp 4  1.273078e+00           1.273078e+01                          90.04950
## comp 5  9.950498e-01           9.950498e+00                         100.00000
## comp 6  4.878116e-30           4.878116e-29                         100.00000
## comp 7  2.267126e-31           2.267126e-30                         100.00000
## comp 8  5.748919e-32           5.748919e-31                         100.00000
## comp 9  3.161321e-32           3.161321e-31                         100.00000
## comp 10 1.613421e-32           1.613421e-31                         100.00000
```

```
plot(pca_ds$eig[,1], type="o", main="Scree Plot")
```




As you may see, the factors that are involved in the answer have the same direction in the plane. On the other hand, the ones that are not related with the answer, have another direction.

In order to extract an expression to predict the answer variable we are going to use a **principal component regression**:

```
### Principal Component Regression
dataset$PC1<-pca_ds$ind$coord[,1]
dataset$PC2<-pca_ds$ind$coord[,2]
dataset$PC3<-pca_ds$ind$coord[,3]
reg_pc<-lm(answer~PC1 + PC2 + PC3, data=dataset)
summary(reg_pc)
```

```
##
## Call:
## lm(formula = answer ~ PC1 + PC2 + PC3, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0129 -0.7970  0.0115  0.7871  3.6616
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.56999    0.02549  179.29   <2e-16 ***
## PC1          0.48107    0.01379   34.88   <2e-16 ***
## PC2          0.54335    0.01688   32.19   <2e-16 ***
## PC3          0.74138    0.01787   41.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.14 on 1996 degrees of freedom
```

```
## Multiple R-squared:  0.6657, Adjusted R-squared:  0.6652
## F-statistic:  1325 on 3 and 1996 DF,  p-value: < 2.2e-16
```

This expression can be used to predict the answer variable.

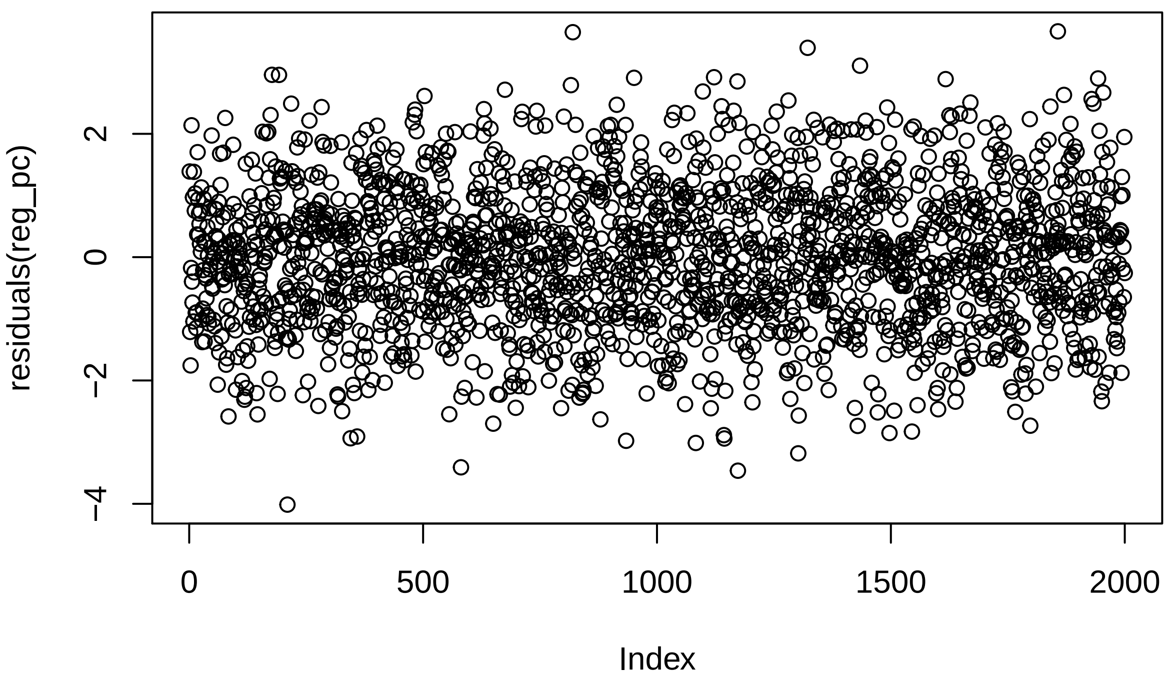
### Testing PCA Assumptions

```
#1. Normality
#Shapiro Wilks Test
shapiro.test(residuals(reg_pc))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(reg_pc)
## W = 0.99914, p-value = 0.4797
```

```
# The error term does follow a Normal distribution. (p>0.05)

### 2. Homogenity of Variance ###
# Residual Analysis #
plot(residuals(reg_pc))
```




```
##Breusch Pagan Test
bptest(reg_pc)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg_pc
## BP = 1.6999, df = 3, p-value = 0.637
```

```
# H0 is accepted (p>0.05).
# The homogenity of variances is provided.
```

```r
### 3. The independence of errors ###
dwtest(reg_pc, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  reg_pc
## DW = 1.9668, p-value = 0.4573
## alternative hypothesis: true autocorrelation is not 0
```

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