Empirical software engineering

Lab 3: Design and simulate an experiment
Group 9

Anton Lutteman Daniel Olsson Gerson Silva Filho Johan Mejborn

10 December 2020

Simulate the data

1) Simulate for a the sample size per group that you defined in the power analysis.

a) How does your fitted model looks like?

- **Test Assumption 1 - Homoscedasticity**

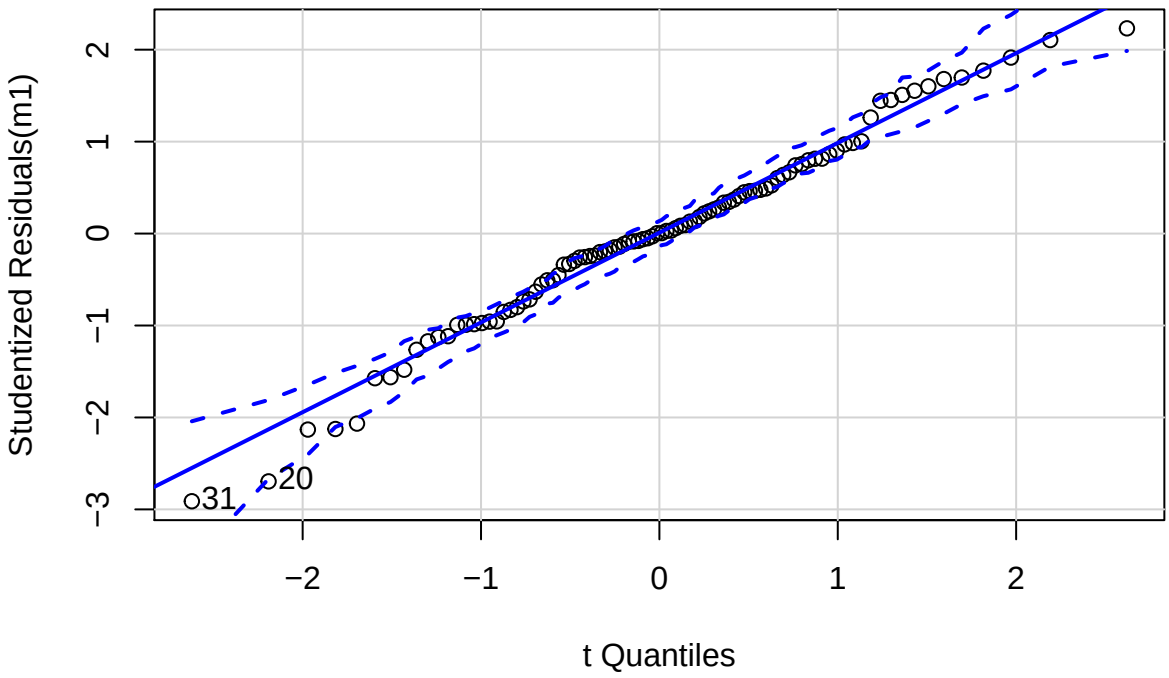
We decided to run a Levene Test to test Homoscedasticity.

```
car::leveneTest(m1)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 11    0.77   0.67
##       84
```

The result show that we cannot reject homoscedasticity. Then let's also verify the QQ-Plot in order to get another point of view.

```
car::qqPlot(m1)
```




```
## [1] 20 31
```

QQplot shows that the data is homoscedastic.

- **Test Assumtion 2 - Normality**

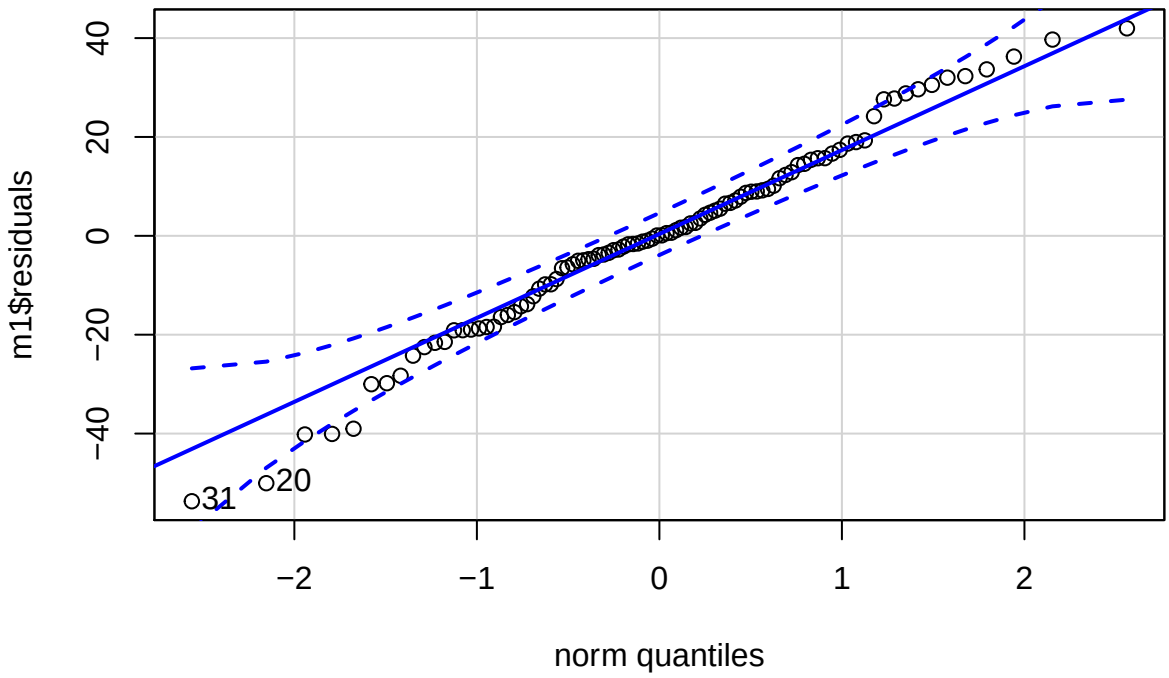
In order to verify normality we run a Shapiro Wilk test.

```
shapiro.test(m1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m1$residuals
## W = 1, p-value = 0.4
```

The result show that we cannot reject normallity. And in order to verify it graphically we used the QQ-plot of the residuals.

```
car::qqPlot(m1$residuals)
```




```
## [1] 31 20
```

The QQplot on the residuals shows normallity.

- **Test Assumtion 3 - Independence**

Based on how we collected the data, we can assume independence.

- **The analysis**

After all the assumptions have been met, we can run an ANOVA test.

```
car::Anova(m1)
```

```
## Anova Table (Type II tests)
##
## Response: y
##                          Sum Sq Df F value  Pr(>F)
## Language                 172356  2  203.60 < 2e-16 ***
## IDE                           9  1    0.02   0.886
## Experience                42384  1  100.14 5.6e-16 ***
## Language:IDE              10323  2   12.19 2.2e-05 ***
## Language:Experience        1497  2    1.77   0.177
## IDE:Experience             2375  1    5.61   0.020 *
```

```
## Language:IDE:Experience   2274  2    2.69   0.074 .
## Residuals                35555 84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test shows that we can reject that Language, Experience, Language:IDE, Language:Experience and IDE:Experience has the same mean for the levels because of the low p values.

Hypothesis 1: IDE alone doesn't seem to have an effect on LOC but we can see that it has a significant effect in combination with Language and Experience.

Hypothesis 2: Experience seem to have a significant effect on LOC based on the result.

Hypothesis 3: Language seem to have a significant effect on LOC bases on the result.

Then to get the highest combination for lines of code we can use the Tukey test:

```
TukeyHSD(aov(m1))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = m1)
##
## $Language
##             diff  lwr upr p adj
## C++-Java      69   56  81     0
## Python-Java  -33  -45 -21     0
## Python-C++  -102 -114 -89     0
##
## $IDE
##                        diff lwr upr p adj
## Visual Studio-Intelij -0.61  -9 7.7  0.89
##
## $Experience
##               diff lwr upr p adj
## Senior-Junior   42  34  50     0
##
## $`Language:IDE`
##                                          diff    lwr     upr p adj
## C++:Intelij-Java:Intelij                 55.5   34.3   76.70  0.00
## Python:Intelij-Java:Intelij             -20.9  -42.1    0.34  0.06
## Java:Visual Studio-Java:Intelij          -1.1  -22.3   20.09  1.00
## C++:Visual Studio-Java:Intelij           80.5   59.3  101.75  0.00
## Python:Visual Studio-Java:Intelij       -46.6  -67.8  -25.41  0.00
## Python:Intelij-C++:Intelij              -76.4  -97.6  -55.15  0.00
## Java:Visual Studio-C++:Intelij          -56.6  -77.8  -35.39  0.00
## C++:Visual Studio-C++:Intelij            25.0    3.8   46.26  0.01
## Python:Visual Studio-C++:Intelij       -102.1 -123.3  -80.89  0.00
## Java:Visual Studio-Python:Intelij        19.8   -1.5   40.97  0.08
## C++:Visual Studio-Python:Intelij        101.4   80.2  122.63  0.00
## Python:Visual Studio-Python:Intelij     -25.7  -47.0   -4.53  0.01
## C++:Visual Studio-Java:Visual Studio     81.7   60.4  102.87  0.00
## Python:Visual Studio-Java:Visual Studio -45.5  -66.7  -24.29  0.00
## Python:Visual Studio-C++:Visual Studio -127.2 -148.4 -105.94  0.00
```

```
##
## $`Language:Experience`
##                                diff  lwr   upr p adj
## C++:Junior-Java:Junior         61.8   41  83.1  0.00
## Python:Junior-Java:Junior     -30.5  -52  -9.3  0.00
## Java:Senior-Java:Junior        39.3   18  60.5  0.00
## C++:Senior-Java:Junior        114.6   93 135.8  0.00
## Python:Senior-Java:Junior       3.5  -18  24.7  1.00
## Python:Junior-C++:Junior      -92.4 -114 -71.2  0.00
## Java:Senior-C++:Junior        -22.5  -44  -1.3  0.03
## C++:Senior-C++:Junior          52.8   32  74.0  0.00
## Python:Senior-C++:Junior      -58.4  -80 -37.2  0.00
## Java:Senior-Python:Junior      69.8   49  91.1  0.00
## C++:Senior-Python:Junior      145.1  124 166.4  0.00
## Python:Senior-Python:Junior    34.0   13  55.2  0.00
## C++:Senior-Java:Senior         75.3   54  96.5  0.00
## Python:Senior-Java:Senior     -35.8  -57 -14.6  0.00
## Python:Senior-C++:Senior     -111.1 -132 -89.9  0.00
##
## $`IDE:Experience`
##                                           diff   lwr upr p adj
## Visual Studio:Junior-Intelij:Junior      -10.6 -26.1   5  0.29
## Intelij:Senior-Intelij:Junior             32.1  16.5  48  0.00
## Visual Studio:Senior-Intelij:Junior       41.4  25.9  57  0.00
## Intelij:Senior-Visual Studio:Junior       42.6  27.1  58  0.00
## Visual Studio:Senior-Visual Studio:Junior 52.0  36.4  68  0.00
## Visual Studio:Senior-Intelij:Senior        9.3  -6.2  25  0.40
##
## $`Language:IDE:Experience`
##                                                    diff    lwr    upr p adj
## C++:Intelij:Junior-Java:Intelij:Junior            44.81   10.2   79.4  0.00
## Python:Intelij:Junior-Java:Intelij:Junior        -29.94  -64.5    4.6  0.16
## Java:Visual Studio:Junior-Java:Intelij:Junior    -21.51  -56.1   13.1  0.63
## C++:Visual Studio:Junior-Java:Intelij:Junior      57.36   22.8   91.9  0.00
## Python:Visual Studio:Junior-Java:Intelij:Junior  -52.64  -87.2  -18.1  0.00
## Java:Intelij:Senior-Java:Intelij:Junior           18.92  -15.7   53.5  0.79
## C++:Intelij:Senior-Java:Intelij:Junior            85.08   50.5  119.7  0.00
## Python:Intelij:Senior-Java:Intelij:Junior          7.10  -27.5   41.7  1.00
## Java:Visual Studio:Senior-Java:Intelij:Junior     38.18    3.6   72.8  0.02
## C++:Visual Studio:Senior-Java:Intelij:Junior      122.62   88.0  157.2  0.00
## Python:Visual Studio:Senior-Java:Intelij:Junior  -21.68  -56.3   12.9  0.62
## Python:Intelij:Junior-C++:Intelij:Junior         -74.75 -109.3  -40.2  0.00
## Java:Visual Studio:Junior-C++:Intelij:Junior     -66.32 -100.9  -31.7  0.00
## C++:Visual Studio:Junior-C++:Intelij:Junior       12.55  -22.0   47.1  0.99
## Python:Visual Studio:Junior-C++:Intelij:Junior   -97.45 -132.0  -62.9  0.00
## Java:Intelij:Senior-C++:Intelij:Junior           -25.89  -60.5    8.7  0.34
## C++:Intelij:Senior-C++:Intelij:Junior             40.27    5.7   74.9  0.01
## Python:Intelij:Senior-C++:Intelij:Junior         -37.71  -72.3   -3.1  0.02
## Java:Visual Studio:Senior-C++:Intelij:Junior      -6.63  -41.2   28.0  1.00
## C++:Visual Studio:Senior-C++:Intelij:Junior       77.81   43.2  112.4  0.00
## Python:Visual Studio:Senior-C++:Intelij:Junior   -66.49 -101.1  -31.9  0.00
## Java:Visual Studio:Junior-Python:Intelij:Junior    8.43  -26.2   43.0  1.00
## C++:Visual Studio:Junior-Python:Intelij:Junior    87.30   52.7  121.9  0.00
## Python:Visual Studio:Junior-Python:Intelij:Junior -22.70  -57.3   11.9  0.55
```

```
## Java:Intelij:Senior-Python:Intelij:Junior                 48.86   14.3   83.4  0.00
## C++:Intelij:Senior-Python:Intelij:Junior                 115.02   80.4  149.6  0.00
## Python:Intelij:Senior-Python:Intelij:Junior               37.05    2.5   71.6  0.03
## Java:Visual Studio:Senior-Python:Intelij:Junior           68.12   33.5  102.7  0.00
## C++:Visual Studio:Senior-Python:Intelij:Junior           152.56  118.0  187.1  0.00
## Python:Visual Studio:Senior-Python:Intelij:Junior          8.26  -26.3   42.8  1.00
## C++:Visual Studio:Junior-Java:Visual Studio:Junior        78.87   44.3  113.5  0.00
## Python:Visual Studio:Junior-Java:Visual Studio:Junior    -31.13  -65.7    3.5  0.12
## Java:Intelij:Senior-Java:Visual Studio:Junior             40.43    5.8   75.0  0.01
## C++:Intelij:Senior-Java:Visual Studio:Junior             106.59   72.0  141.2  0.00
## Python:Intelij:Senior-Java:Visual Studio:Junior           28.61   -6.0   63.2  0.21
## Java:Visual Studio:Senior-Java:Visual Studio:Junior       59.69   25.1   94.3  0.00
## C++:Visual Studio:Senior-Java:Visual Studio:Junior       144.13  109.6  178.7  0.00
## Python:Visual Studio:Senior-Java:Visual Studio:Junior     -0.17  -34.8   34.4  1.00
## Python:Visual Studio:Junior-C++:Visual Studio:Junior    -110.00 -144.6  -75.4  0.00
## Java:Intelij:Senior-C++:Visual Studio:Junior            -38.44  -73.0   -3.9  0.02
## C++:Intelij:Senior-C++:Visual Studio:Junior              27.72   -6.9   62.3  0.25
## Python:Intelij:Senior-C++:Visual Studio:Junior          -50.26  -84.8  -15.7  0.00
## Java:Visual Studio:Senior-C++:Visual Studio:Junior      -19.18  -53.8   15.4  0.78
## C++:Visual Studio:Senior-C++:Visual Studio:Junior        65.26   30.7   99.8  0.00
## Python:Visual Studio:Senior-C++:Visual Studio:Junior    -79.05 -113.6  -44.5  0.00
## Java:Intelij:Senior-Python:Visual Studio:Junior          71.56   37.0  106.1  0.00
## C++:Intelij:Senior-Python:Visual Studio:Junior          137.72  103.1  172.3  0.00
## Python:Intelij:Senior-Python:Visual Studio:Junior        59.75   25.2   94.3  0.00
## Java:Visual Studio:Senior-Python:Visual Studio:Junior    90.83   56.2  125.4  0.00
## C++:Visual Studio:Senior-Python:Visual Studio:Junior    175.27  140.7  209.8  0.00
## Python:Visual Studio:Senior-Python:Visual Studio:Junior  30.96   -3.6   65.5  0.12
## C++:Intelij:Senior-Java:Intelij:Senior                   66.16   31.6  100.7  0.00
## Python:Intelij:Senior-Java:Intelij:Senior               -11.81  -46.4   22.8  0.99
## Java:Visual Studio:Senior-Java:Intelij:Senior            19.27  -15.3   53.8  0.77
## C++:Visual Studio:Senior-Java:Intelij:Senior            103.71   69.1  138.3  0.00
## Python:Visual Studio:Senior-Java:Intelij:Senior         -40.60  -75.2   -6.0  0.01
## Python:Intelij:Senior-C++:Intelij:Senior                -77.97 -112.6  -43.4  0.00
## Java:Visual Studio:Senior-C++:Intelij:Senior            -46.89  -81.5  -12.3  0.00
## C++:Visual Studio:Senior-C++:Intelij:Senior              37.55    3.0   72.1  0.02
## Python:Visual Studio:Senior-C++:Intelij:Senior         -106.76 -141.3  -72.2  0.00
## Java:Visual Studio:Senior-Python:Intelij:Senior          31.08   -3.5   65.7  0.12
## C++:Visual Studio:Senior-Python:Intelij:Senior          115.52   80.9  150.1  0.00
## Python:Visual Studio:Senior-Python:Intelij:Senior       -28.79  -63.4    5.8  0.20
## C++:Visual Studio:Senior-Java:Visual Studio:Senior       84.44   49.9  119.0  0.00
## Python:Visual Studio:Senior-Java:Visual Studio:Senior   -59.87  -94.5  -25.3  0.00
## Python:Visual Studio:Senior-C++:Visual Studio:Senior    -144.31 -178.9 -109.7  0.00
```

Without drawing conclusions for all combinations presented by the Tukey test we can say that the highest combination for lines of code is **Language = C++, IDE = Visual Studio and Experience = Senior**.

b) How do it compare with your 'true' model you defined in the simulation.

When we compare the created model defined in our main function with the defined coefficients, with the data generated by the model, we can see that they are similar. Also we can notice some variations that can be explained either by the defined standard deviation of the model, or that the model interpreted the data in another way than what we set up. The difference in the first order is very small compared to our true model, however the third order interactions are deviating quite a bit compared to our true model.

We can observe the differences looking in the following table and the defined factors:

	Defined Coeficients	Model Coeficients
(Intercept)	100	100.7
LanguageC++	50	44.8
LanguagePython	-30	-29.9
IDEVisual Studio	-10	-21.5
ExperienceSenior	30	18.9
LanguageC++:IDEVisual Studio	20	34.1
LanguagePython:IDEVisual Studio	-20	-1.2
LanguageC++:ExperienceSenior	10	21.4
LanguagePython:ExperienceSenior	10	18.1
IDEVisual Studio:ExperienceSenior	20	40.8
LanguageC++:IDEVisual Studio:ExperienceSenior	10	-15.8
LanguagePython:IDEVisual Studio:ExperienceSenior	-15	-46.9

2) Simulate for an underpowered study with half the sample size you calculated in the power analysis

a) How does your fitted model looks like?

- **Test Assumption 1 - Homoscedasticity**

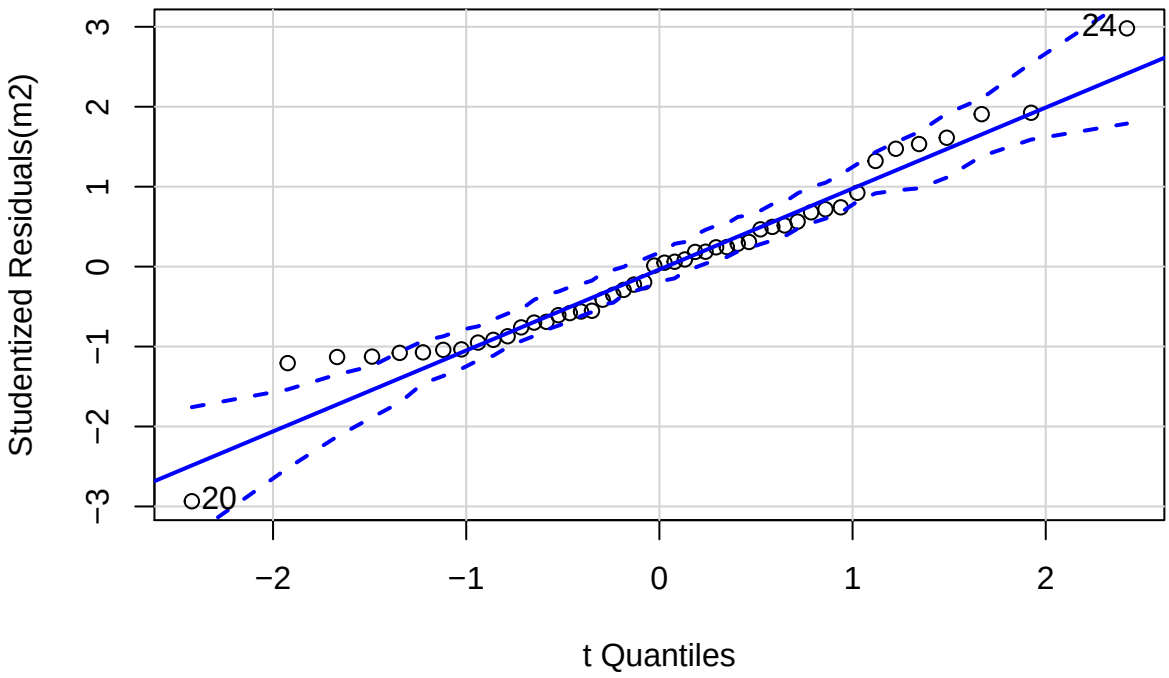
We decided to run a Levene Test to test Homoscedasticity.

```
car::leveneTest(m2)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 11     0.4   0.95
##       36
```

The result show that we cannot reject homoscedasticity. Then let's also verify the QQ-Plot in order to get another point of view.

```
car::qqPlot(m2)
```




```
## [1] 20 24
```

QQplot shows that the data is homoscedastic.

- **Test Assumtion 2 - Normality**

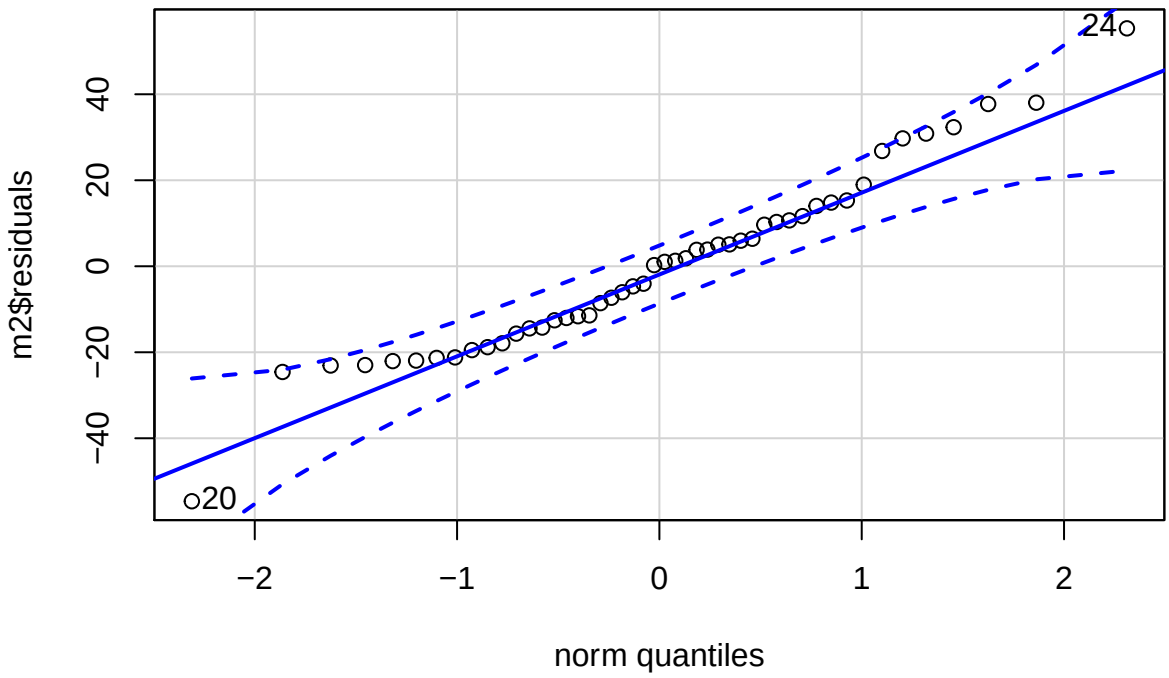
In order to verify normality we run a Shapiro Wilk test.

```
shapiro.test(m2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m2$residuals
## W = 1, p-value = 0.2
```

The result show that we cannot reject normallity. And in order to verify it graphically we used the QQ-plot of the residuals.

```
car::qqPlot(m2$residuals)
```




```
## [1] 24 20
```

The QQplot on the residuals shows normallity.

- **Test Assumtion 3 - Independence**

Based on how we collected the data, we can assume independence.

- **The analysis**

After all the assumptions have been met, we can run an ANOVA test.

```
car::Anova(m2)
```

```
## Anova Table (Type II tests)
##
## Response: y
##                          Sum Sq Df F value  Pr(>F)
## Language                  73167  2   65.33 1.0e-12 ***
## IDE                         208  1    0.37   0.546
## Experience                25459  1   45.46 7.2e-08 ***
## Language:IDE               2203  2    1.97   0.155
## Language:Experience         536  2    0.48   0.624
## IDE:Experience               53  1    0.09   0.760
```

```
## Language:IDE:Experience   2794  2    2.49   0.097 .
## Residuals                20161 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test shows that we can reject that Language, Experience, Language:IDE, Language:Experience and IDE:Experience has the same mean for the levels because of the low p values.

Hypothesis 1: IDE doesn't seem to have an effect on LOC neither alone nor as an interaction.

Hypothesis 2: Experience seem to have a significant effect on LOC based on the result.

Hypothesis 3: Language seem to have a significant effect on LOC bases on the result.

Then to get the highest combination for lines of code we can use the Tukey test:

```
TukeyHSD(aov(m2))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = m2)
##
## $Language
##             diff  lwr upr p adj
## C++-Java      60   40  81     0
## Python-Java  -34  -55 -14     0
## Python-C++   -94 -115 -74     0
##
## $IDE
##                       diff lwr upr p adj
## Visual Studio-Intelij -4.2 -18 9.7  0.55
##
## $Experience
##               diff lwr upr p adj
## Senior-Junior   46  32  60     0
##
## $`Language:IDE`
##                                          diff  lwr   upr p adj
## C++:Intelij-Java:Intelij                 56.2   21  91.8  0.00
## Python:Intelij-Java:Intelij             -22.3  -58  13.3  0.43
## Java:Visual Studio-Java:Intelij           1.1  -34  36.7  1.00
## C++:Visual Studio-Java:Intelij           65.4   30 100.9  0.00
## Python:Visual Studio-Java:Intelij       -45.1  -81  -9.5  0.01
## Python:Intelij-C++:Intelij              -78.5 -114 -42.9  0.00
## Java:Visual Studio-C++:Intelij          -55.1  -91 -19.5  0.00
## C++:Visual Studio-C++:Intelij             9.2  -26  44.8  0.97
## Python:Visual Studio-C++:Intelij       -101.3 -137 -65.7  0.00
## Java:Visual Studio-Python:Intelij         23.4  -12  59.0  0.37
## C++:Visual Studio-Python:Intelij          87.6   52 123.2  0.00
## Python:Visual Studio-Python:Intelij      -22.8  -58  12.8  0.41
## C++:Visual Studio-Java:Visual Studio      64.2   29  99.8  0.00
## Python:Visual Studio-Java:Visual Studio  -46.2  -82 -10.6  0.00
## Python:Visual Studio-C++:Visual Studio  -110.4 -146 -74.8  0.00
##
```

```
## $`Language:Experience`
##                                diff    lwr   upr p adj
## C++:Junior-Java:Junior         55.8   20.2  91.4  0.00
## Python:Junior-Java:Junior     -30.5  -66.1   5.1  0.13
## Java:Senior-Java:Junior        45.6   10.0  81.2  0.01
## C++:Senior-Java:Junior        110.2   74.6 145.8  0.00
## Python:Senior-Java:Junior       7.6  -28.0  43.2  0.99
## Python:Junior-C++:Junior      -86.3 -121.9 -50.7  0.00
## Java:Senior-C++:Junior        -10.2  -45.8  25.4  0.95
## C++:Senior-C++:Junior          54.5   18.9  90.1  0.00
## Python:Senior-C++:Junior      -48.1  -83.7 -12.5  0.00
## Java:Senior-Python:Junior      76.1   40.5 111.7  0.00
## C++:Senior-Python:Junior      140.8  105.2 176.4  0.00
## Python:Senior-Python:Junior    38.1    2.5  73.7  0.03
## C++:Senior-Java:Senior         64.7   29.1 100.3  0.00
## Python:Senior-Java:Senior     -37.9  -73.5  -2.3  0.03
## Python:Senior-C++:Senior     -102.6 -138.2 -67.0  0.00
##
## $`IDE:Experience`
##                                          diff lwr upr p adj
## Visual Studio:Junior-Intelij:Junior      -6.3 -32  20  0.92
## Intelij:Senior-Intelij:Junior            44.0  18  70  0.00
## Visual Studio:Senior-Intelij:Junior      41.9  16  68  0.00
## Intelij:Senior-Visual Studio:Junior      50.2  24  76  0.00
## Visual Studio:Senior-Visual Studio:Junior 48.2  22  74  0.00
## Visual Studio:Senior-Intelij:Senior      -2.1 -28  24  1.00
##
## $`Language:IDE:Experience`
##                                                      diff     lwr   upr p adj
## C++:Intelij:Junior-Java:Intelij:Junior                58.8    0.36 117.2  0.05
## Python:Intelij:Junior-Java:Intelij:Junior            -30.1  -88.47  28.3  0.81
## Java:Visual Studio:Junior-Java:Intelij:Junior         -4.0  -62.36  54.4  1.00
## C++:Visual Studio:Junior-Java:Intelij:Junior          48.8   -9.62 107.2  0.18
## Python:Visual Studio:Junior-Java:Intelij:Junior      -34.9  -93.34  23.5  0.63
## Java:Intelij:Senior-Java:Intelij:Junior               40.5  -17.91  98.9  0.42
## C++:Intelij:Senior-Java:Intelij:Junior                94.1   35.71 152.5  0.00
## Python:Intelij:Senior-Java:Intelij:Junior             26.0  -32.44  84.4  0.91
## Java:Visual Studio:Senior-Java:Intelij:Junior         46.7  -11.74 105.1  0.23
## C++:Visual Studio:Senior-Java:Intelij:Junior         122.4   64.00 180.8  0.00
## Python:Visual Studio:Senior-Java:Intelij:Junior      -14.7  -73.09  43.7  1.00
## Python:Intelij:Junior-C++:Intelij:Junior             -88.8 -147.24 -30.4  0.00
## Java:Visual Studio:Junior-C++:Intelij:Junior         -62.7 -121.13  -4.3  0.03
## C++:Visual Studio:Junior-C++:Intelij:Junior          -10.0  -68.39  48.4  1.00
## Python:Visual Studio:Junior-C++:Intelij:Junior       -93.7 -152.11 -35.3  0.00
## Java:Intelij:Senior-C++:Intelij:Junior               -18.3  -76.68  40.1  0.99
## C++:Intelij:Senior-C++:Intelij:Junior                 35.4  -23.05  93.8  0.62
## Python:Intelij:Senior-C++:Intelij:Junior             -32.8  -91.21  25.6  0.72
## Java:Visual Studio:Senior-C++:Intelij:Junior         -12.1  -70.51  46.3  1.00
## C++:Visual Studio:Senior-C++:Intelij:Junior           63.6    5.23 122.0  0.02
## Python:Visual Studio:Senior-C++:Intelij:Junior       -73.4 -131.85 -15.0  0.00
## Java:Visual Studio:Junior-Python:Intelij:Junior       26.1  -32.29  84.5  0.91
## C++:Visual Studio:Junior-Python:Intelij:Junior        78.9   20.45 137.3  0.00
## Python:Visual Studio:Junior-Python:Intelij:Junior     -4.9  -63.27  53.5  1.00
## Java:Intelij:Senior-Python:Intelij:Junior             70.6   12.15 129.0  0.01
```

```
## C++:Intelij:Senior-Python:Intelij:Junior                 124.2   65.78 182.6  0.00
## Python:Intelij:Senior-Python:Intelij:Junior               56.0   -2.37 114.4  0.07
## Java:Visual Studio:Senior-Python:Intelij:Junior           76.7   18.33 135.1  0.00
## C++:Visual Studio:Senior-Python:Intelij:Junior           152.5   94.07 210.9  0.00
## Python:Visual Studio:Senior-Python:Intelij:Junior         15.4  -43.02  73.8  1.00
## C++:Visual Studio:Junior-Java:Visual Studio:Junior        52.7   -5.66 111.2  0.11
## Python:Visual Studio:Junior-Java:Visual Studio:Junior    -31.0  -89.38  27.4  0.78
## Java:Intelij:Senior-Java:Visual Studio:Junior             44.4  -13.96 102.9  0.29
## C++:Intelij:Senior-Java:Visual Studio:Junior              98.1   39.67 156.5  0.00
## Python:Intelij:Senior-Java:Visual Studio:Junior           29.9  -28.48  88.3  0.81
## Java:Visual Studio:Senior-Java:Visual Studio:Junior       50.6   -7.79 109.0  0.14
## C++:Visual Studio:Senior-Java:Visual Studio:Junior       126.4   67.96 184.8  0.00
## Python:Visual Studio:Senior-Java:Visual Studio:Junior    -10.7  -69.13  47.7  1.00
## Python:Visual Studio:Junior-C++:Visual Studio:Junior     -83.7 -142.13 -25.3  0.00
## Java:Intelij:Senior-C++:Visual Studio:Junior              -8.3  -66.70  50.1  1.00
## C++:Intelij:Senior-C++:Visual Studio:Junior               45.3  -13.07 103.7  0.26
## Python:Intelij:Senior-C++:Visual Studio:Junior           -22.8  -81.23  35.6  0.96
## Java:Visual Studio:Senior-C++:Visual Studio:Junior        -2.1  -60.53  56.3  1.00
## C++:Visual Studio:Senior-C++:Visual Studio:Junior         73.6   15.21 132.0  0.00
## Python:Visual Studio:Senior-C++:Visual Studio:Junior     -63.5 -121.87  -5.1  0.02
## Java:Intelij:Senior-Python:Visual Studio:Junior           75.4   17.02 133.8  0.00
## C++:Intelij:Senior-Python:Visual Studio:Junior           129.1   70.65 187.5  0.00
## Python:Intelij:Senior-Python:Visual Studio:Junior         60.9    2.49 119.3  0.03
## Java:Visual Studio:Senior-Python:Visual Studio:Junior     81.6   23.19 140.0  0.00
## C++:Visual Studio:Senior-Python:Visual Studio:Junior     157.3   98.94 215.7  0.00
## Python:Visual Studio:Senior-Python:Visual Studio:Junior   20.3  -38.15  78.7  0.98
## C++:Intelij:Senior-Java:Intelij:Senior                    53.6   -4.78 112.0  0.10
## Python:Intelij:Senior-Java:Intelij:Senior                -14.5  -72.93  43.9  1.00
## Java:Visual Studio:Senior-Java:Intelij:Senior              6.2  -52.23  64.6  1.00
## C++:Visual Studio:Senior-Java:Intelij:Senior              81.9   23.51 140.3  0.00
## Python:Visual Studio:Senior-Java:Intelij:Senior          -55.2 -113.58   3.2  0.08
## Python:Intelij:Senior-C++:Intelij:Senior                 -68.2 -126.56  -9.7  0.01
## Java:Visual Studio:Senior-C++:Intelij:Senior             -47.5 -105.86  10.9  0.21
## C++:Visual Studio:Senior-C++:Intelij:Senior               28.3  -30.12  86.7  0.86
## Python:Visual Studio:Senior-C++:Intelij:Senior         -108.8 -167.20 -50.4  0.00
## Java:Visual Studio:Senior-Python:Intelij:Senior           20.7  -37.71  79.1  0.98
## C++:Visual Studio:Senior-Python:Intelij:Senior            96.4   38.04 154.8  0.00
## Python:Visual Studio:Senior-Python:Intelij:Senior        -40.6  -99.05  17.8  0.41
## C++:Visual Studio:Senior-Java:Visual Studio:Senior        75.7   17.34 134.1  0.00
## Python:Visual Studio:Senior-Java:Visual Studio:Senior    -61.3 -119.75  -2.9  0.03
## Python:Visual Studio:Senior-C++:Visual Studio:Senior     -137.1 -195.49 -78.7  0.00
```

Without drawing conclusions for all combinations presented by the Tukey test we can say that the highest combination for lines of code is **Language = C++ and Experience = Senior**.

b) How do it compare with your 'true' model you defined in the simulation

When we compare the created model defined in our main function with the defined coefficients, with the data generated by the model, we can see that they are similar. Also we can notice some variations that can be explained either by the defined standard deviation of the model, or that the model interpreted the data in another way than what we set up. The difference in the first order is very small compared to our true model, however the third order interactions are deviating quite a bit compared to our true model.

We can observe the differences looking in the following table and the defined factors:

	Defined Coeficients	Model Coeficients
(Intercept)	100	97.97
LanguageC++	50	58.77
LanguagePython	-30	-30.07
IDEVisual Studio	-10	-3.96
ExperienceSenior	30	40.49
LanguageC++:IDEVisual Studio	20	-6.02
LanguagePython:IDEVisual Studio	-20	-0.91
LanguageC++:ExperienceSenior	10	-5.14
LanguagePython:ExperienceSenior	10	15.54
IDEVisual Studio:ExperienceSenior	20	10.13
LanguageC++:IDEVisual Studio:ExperienceSenior	10	28.14
LanguagePython:IDEVisual Studio:ExperienceSenior	-15	-45.91

c) How do the results change compared to the correctly powered experiment.

When we compare the two models we see some differences between them. These could be explained by the standard deviation and the small sample size. We also get more coefficients with small p values that are to be considered statistically significant for m1.

Comparing the analysis on the two models show similar result for Hypothesis 2 and 3 but for model two we do not see an interaction effect for IDE. This means we can't reject the null hypothesis for hypothesis 1 that the means are equal for the different IDEs.