R_21_5.R

r1467469

2022-05-22

```r
# Installing packages

install.packages("readxl")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("lmtest")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("stargazer")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("car")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("tseries")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("sandwich")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("olsrr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("lmtest")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("gvlma")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("boot")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("skedastic")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("rcompanion")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
install.packages("ggfortify")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
# Loading packages

library('readxl')
library('lmtest')
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
library('stargazer')
```

```
##
## Please cite as:

##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
library('car')
```

```
## Loading required package: carData
```

```r
library('tseries')
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library('ggplot2')
library('sandwich')
library('dplyr')
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library('olsrr')
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
library('lmtest')
library('gvlma')
library('boot')
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit
```

```r
library('skedastic')
library('rcompanion')
library('ggfortify')

# Importing Datasets

Euronext = read_excel("10.5.22 - filtrert.xlsx", sheet = "Data")
Amsterdam = read_excel("10.5.22 - filtrert.xlsx", sheet = "Amsterdam")
Brussels = read_excel("10.5.22 - filtrert.xlsx", sheet = "Brussels")
Paris = read_excel("10.5.22 - filtrert.xlsx", sheet = "Paris")


## underpricing significance test

# Euronext
mean(Euronext$MAR)         # mean = 0.03017339
```

```
## [1] 0.03017339
```

```r
median(Euronext$MAR)       # median = 0.01212248
```

```
## [1] 0.01212248
```

```r
t.test(Euronext$MAR)       # P-value = 4.751e-05, t-stat = 4.1724
```

```
##
##  One Sample t-test
##
## data:  Euronext$MAR
## t = 4.1724, df = 174, p-value = 4.751e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.01590033 0.04444646
## sample estimates:
##  mean of x
## 0.03017339
```

```r
wilcox.test(Euronext$MAR)  # p-value = 0.000203, V = 10194
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  Euronext$MAR
## V = 10194, p-value = 0.000203
## alternative hypothesis: true location is not equal to 0
```

```r
# Amsterdam
mean(Amsterdam$MAR)        # mean = 0.05139293
```

```
## [1] 0.05139293
```

```r
median(Amsterdam$MAR)      # median = 0.025018
```

```
## [1] 0.025018
```

```r
t.test(Amsterdam$MAR)      # p-value = 0.0055336, t-stat = 2.983
```

```
##
##  One Sample t-test
##
## data:  Amsterdam$MAR
## t = 2.983, df = 33, p-value = 0.005336
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.01634056 0.08644531
## sample estimates:
##  mean of x
## 0.05139293
```

```r
wilcox.test(Amsterdam$MAR) # p-value = 0.0007819, V = 487
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  Amsterdam$MAR
## V = 487, p-value = 0.0007819
## alternative hypothesis: true location is not equal to 0
```

```r
# Brussels
mean(Brussels$MAR)          # mean = 0.02295195
```

```
## [1] 0.02295195
```

```r
median(Brussels$MAR)        # median = 0.02717771
```

```
## [1] 0.02717771
```

```r
t.test(Brussels$MAR)        # p-value = 0.1395, t-stat = 1.5333
```

```
##
##  One Sample t-test
##
## data:  Brussels$MAR
## t = 1.5333, df = 22, p-value = 0.1395
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.008091317  0.053995218
## sample estimates:
##  mean of x
## 0.02295195
```

```r
wilcox.test(Brussels$MAR)  # p-value = 0.2226, V = 179
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  Brussels$MAR
## V = 179, p-value = 0.2226
## alternative hypothesis: true location is not equal to 0
```

```r
# Paris
mean(Paris$MAR)             # mean = 0.02546686
```

```
## [1] 0.02546686
```

```r
median(Paris$MAR)           # median = 0.007833142
```

```
## [1] 0.007833142
```

```r
t.test(Paris$MAR)           # p-value = 0.005662, t-stat = 2.8188
```

```
##
##  One Sample t-test
##
## data:  Paris$MAR
## t = 2.8188, df = 117, p-value = 0.005662
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.007574354 0.043359366
## sample estimates:
##  mean of x
## 0.02546686
```

```r
wilcox.test(Paris$MAR)     # p-value = 0.03022, V = 4318
```

```
##
##  Wilcoxon signed rank test with continuity correction
```

```
##
## data:  Paris$MAR
## V = 4318, p-value = 0.03022
## alternative hypothesis: true location is not equal to 0
```

```
## Normality tests --> SW + JB: normally distributed if p-value > 0.05

# Euronext
shapiro.test(Euronext$MAR)      # p-value = 1.1901e-12
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Euronext$MAR
## W = 0.84285, p-value = 1.901e-12
```

```
jarque.bera.test(Euronext$MAR)  # p-value < 2.2e-16
```

```
##
##  Jarque Bera Test
##
## data:  Euronext$MAR
## X-squared = 332.79, df = 2, p-value < 2.2e-16
```

```
# Amsterdam
shapiro.test(Amsterdam$MAR)     # p-value = 2.343e-06
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Amsterdam$MAR
## W = 0.74225, p-value = 2.343e-06
```

```
jarque.bera.test(Amsterdam$MAR) # p-value < 2.2e-16
```

```
##
##  Jarque Bera Test
##
## data:  Amsterdam$MAR
## X-squared = 105.51, df = 2, p-value < 2.2e-16
```

```
# Brussels
shapiro.test(Brussels$MAR)      # p-value = 0.6072
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Brussels$MAR
## W = 0.96655, p-value = 0.6072
```

```
jarque.bera.test(Brussels$MAR)  # p-value = 0.6339
```

```
##
##  Jarque Bera Test
##
## data:  Brussels$MAR
## X-squared = 0.91165, df = 2, p-value = 0.6339
```

```r
# Paris
shapiro.test(Paris$MAR)         # p-value = 3.392e-10
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Paris$MAR
## W = 0.83397, p-value = 3.392e-10
```

```r
jarque.bera.test(Paris$MAR)     # p-value < 2.2e-16
```

```
##
##  Jarque Bera Test
##
## data:  Paris$MAR
## X-squared = 188.14, df = 2, p-value < 2.2e-16
```

```r
## Regressions for entire sample

# Model 1

Model1_Euronext = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                     + Sentiment + HC + VC, data = Euronext)

Model2_Euronext = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                     + Sentiment + HC + VC + Marketreturn + Marketvolatility, data = Euronext)

Model3_Euronext = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                     + Sentiment + HC + VC + Marketreturn + Marketvolatility
                     + Dummy2014 + Dummy2015 + Dummy2016 + Dummy2017 + Dummy2018 + Dummy2019 + Dummy202


# Heteroscedasticity test --> if p-value > 0.05 == homosked

bptest(Model1_Euronext)     # p-value = 0.1154
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model1_Euronext
## BP = 12.898, df = 8, p-value = 0.1154
```

```r
white_lm(Model1_Euronext)   # p-value = 0.227
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      19.9   0.227        16 White's Test greater
```

```r
bptest(Model2_Euronext)     # p-value = 0.2183
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model2_Euronext
## BP = 13.097, df = 10, p-value = 0.2183
```

```r
white_lm(Model2_Euronext)    # p-value = 0.382
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      21.3   0.382        20 White's Test greater
```

```r
bptest(Model3_Euronext)      # p-value = 0.2882
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model3_Euronext
## BP = 19.731, df = 17, p-value = 0.2882
```

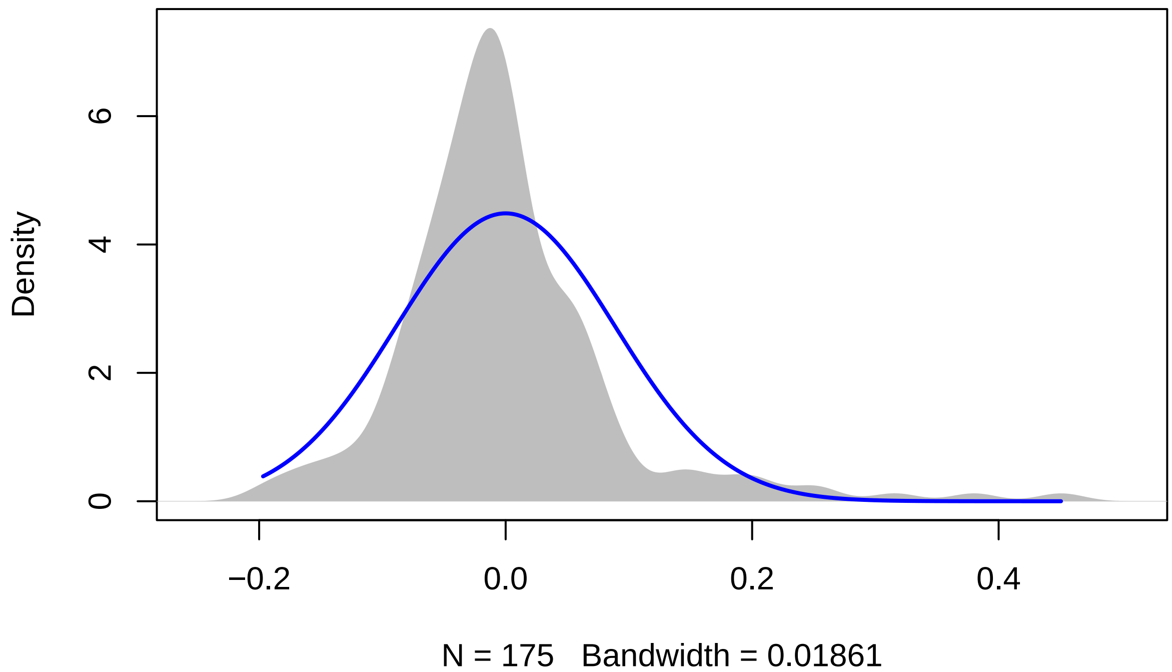
```r
white_lm(Model3_Euronext)    # p-value = 0.906
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      23.7   0.906        34 White's Test greater
```

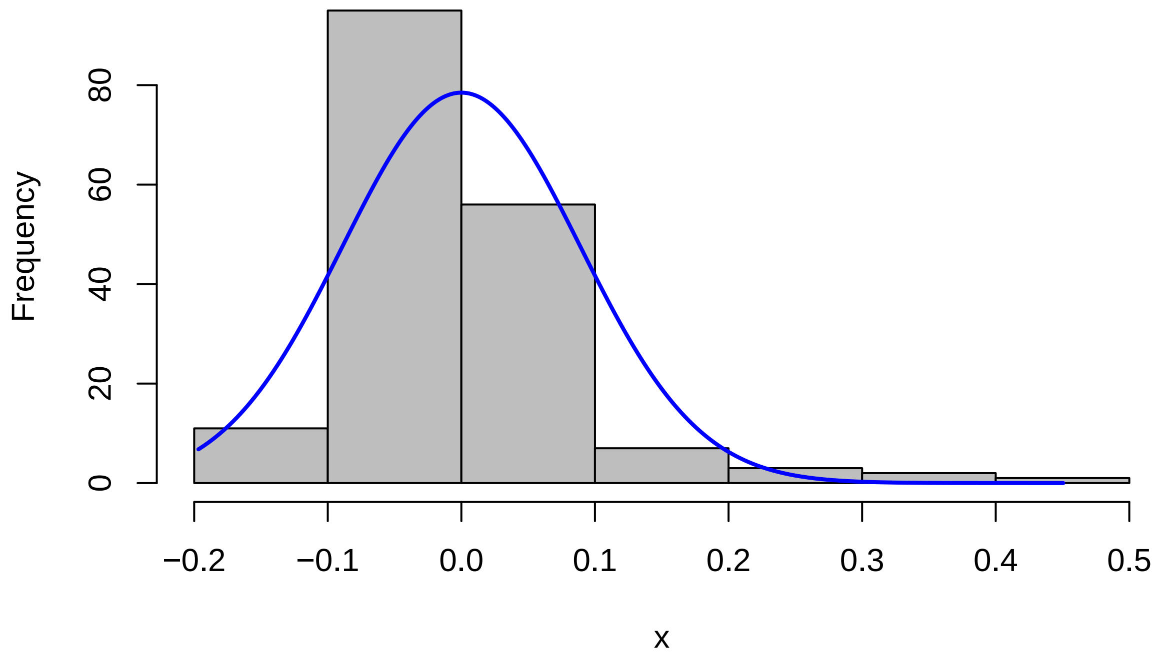
```r
# Normality of Residuals

jarque.bera.test(Model1_Euronext$residuals) # P-value < 2.2e-16 --> Non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model1_Euronext$residuals
## X-squared = 330.01, df = 2, p-value < 2.2e-16
```

```r
plotNormalDensity(Model1_Euronext$residuals)
```



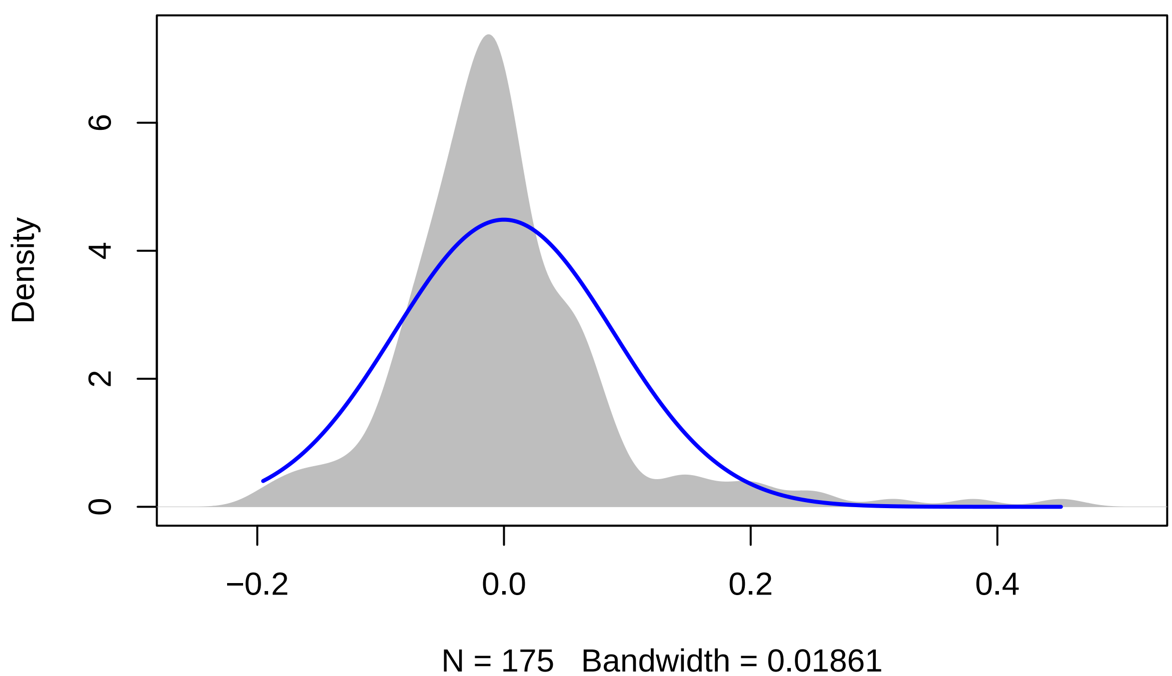
```
plotNormalHistogram(Model1_Euronext$residuals)
```



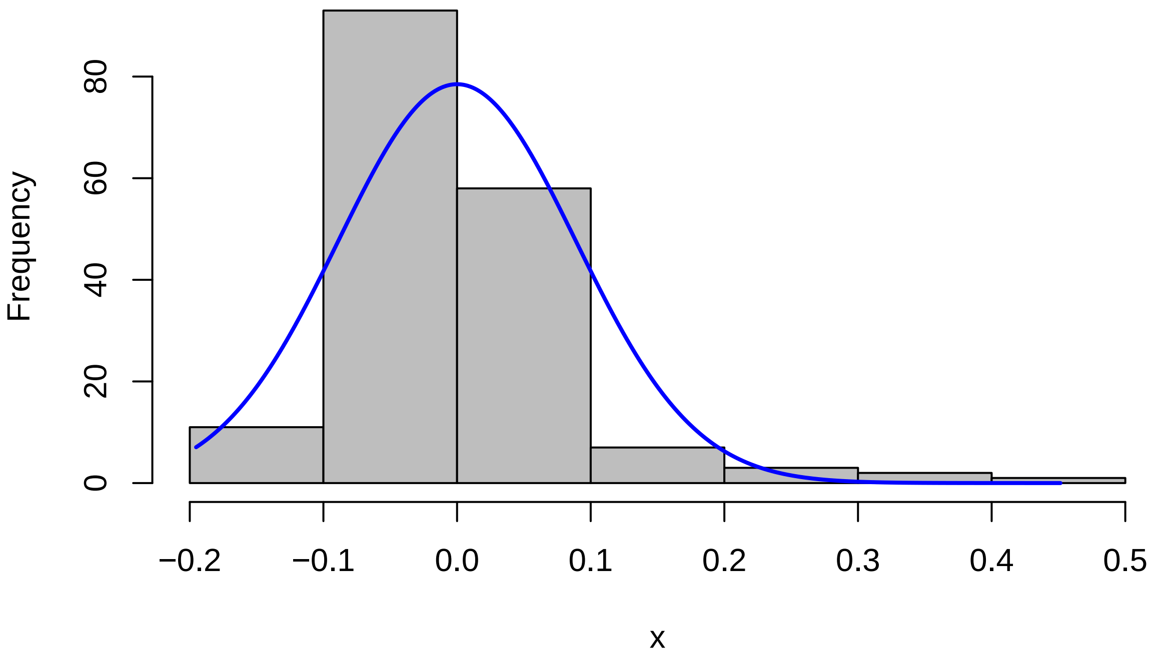

```
jarque.bera.test(Model2_Euronext$residuals) # P-value < 2.2e-16 --> Non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model2_Euronext$residuals
## X-squared = 333.56, df = 2, p-value < 2.2e-16
```

```
plotNormalDensity(Model2_Euronext$residuals)
```



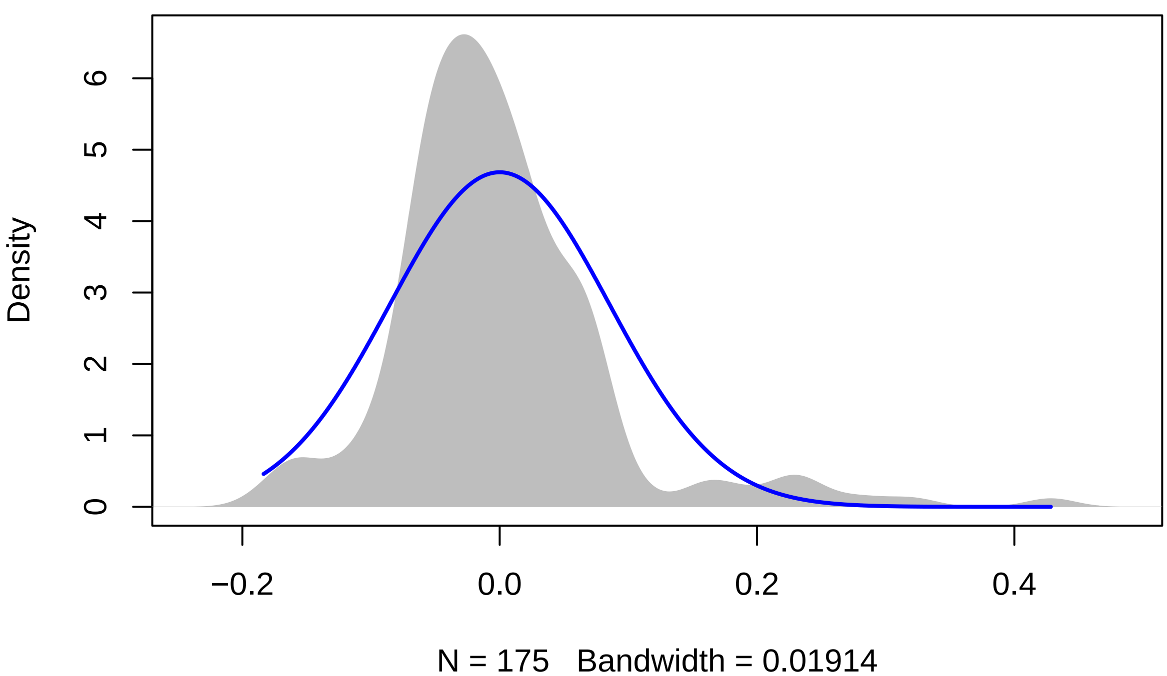
```r
plotNormalHistogram(Model2_Euronext$residuals)
```



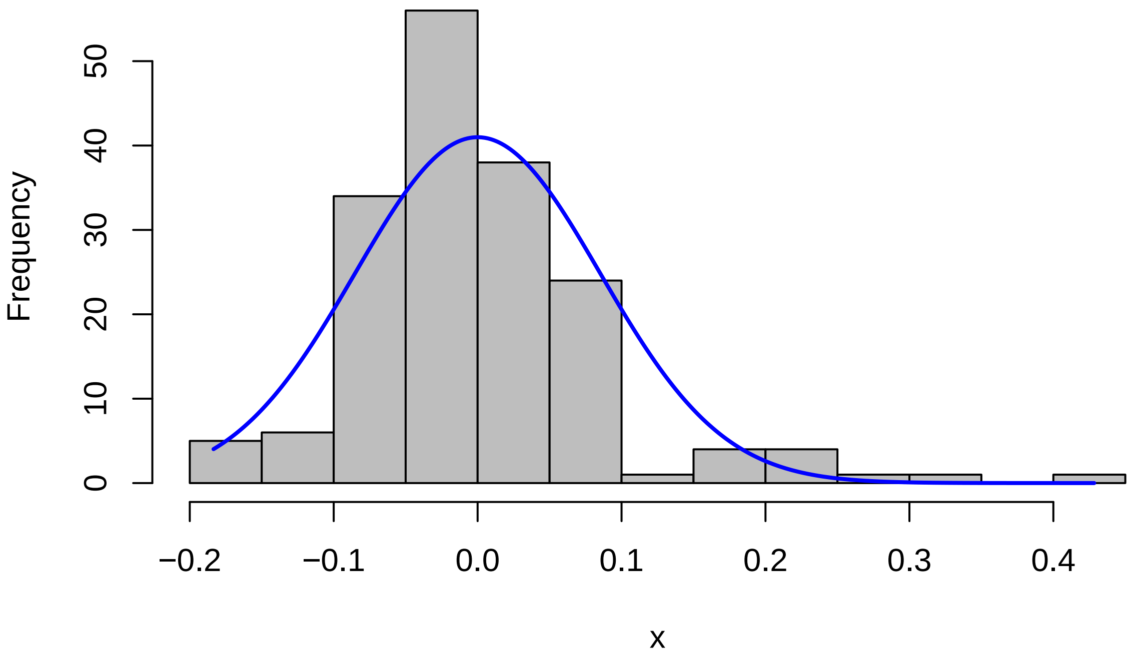

```r
jarque.bera.test(Model3_Euronext$residuals) # P-value < 2.2e-16 --> Non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model3_Euronext$residuals
## X-squared = 254.65, df = 2, p-value < 2.2e-16
```

```r
plotNormalDensity(Model3_Euronext$residuals)
```



```
plotNormalHistogram(Model3_Euronext$residuals)
```




```
# multicollinearity

ols_vif_tol(Model1_Euronext) # all good
```

```
##      Variables Tolerance      VIF
## 1       LN_age 0.8261668 1.210409
## 2 LN_OfferSize 0.5949145 1.680914
## 3         Tech 0.9729079 1.027847
## 4         Rank 0.6601647 1.514773
## 5    Technique 0.9067247 1.102871
## 6    Sentiment 0.9771633 1.023370
## 7           HC 0.8791107 1.137513
## 8           VC 0.9073228 1.102144
```

```
ols_vif_tol(Model2_Euronext) # all good
```

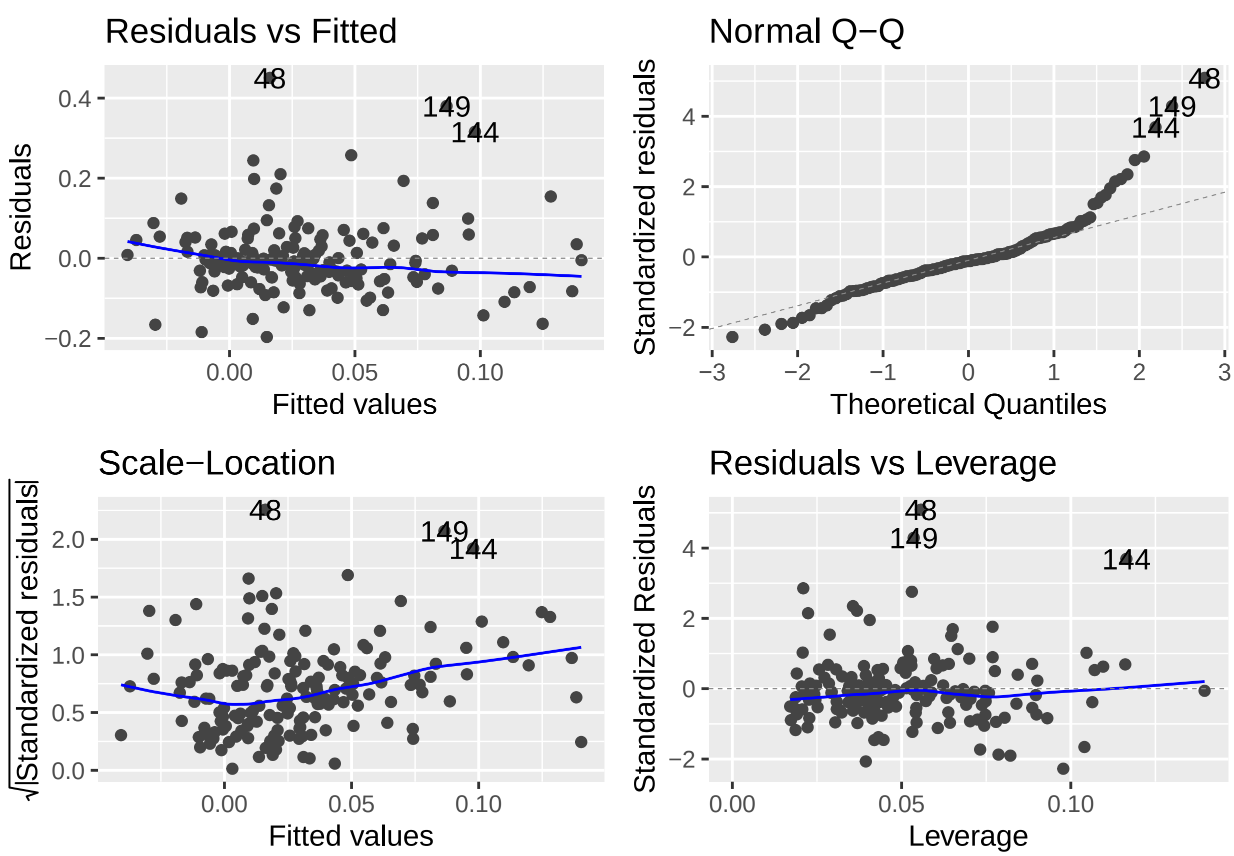
```
##           Variables Tolerance      VIF
## 1            LN_age 0.8252731 1.211720
## 2      LN_OfferSize 0.5910844 1.691806
## 3              Tech 0.9550713 1.047042
## 4              Rank 0.6586139 1.518340
## 5         Technique 0.9026506 1.107848
## 6         Sentiment 0.8526486 1.172816
## 7                HC 0.8668501 1.153602
## 8                VC 0.9010899 1.109767
## 9       Marketreturn 0.9471755 1.055771
## 10 Marketvolatility 0.8247133 1.212543
```

```
ols_vif_tol(Model3_Euronext) # all good
```

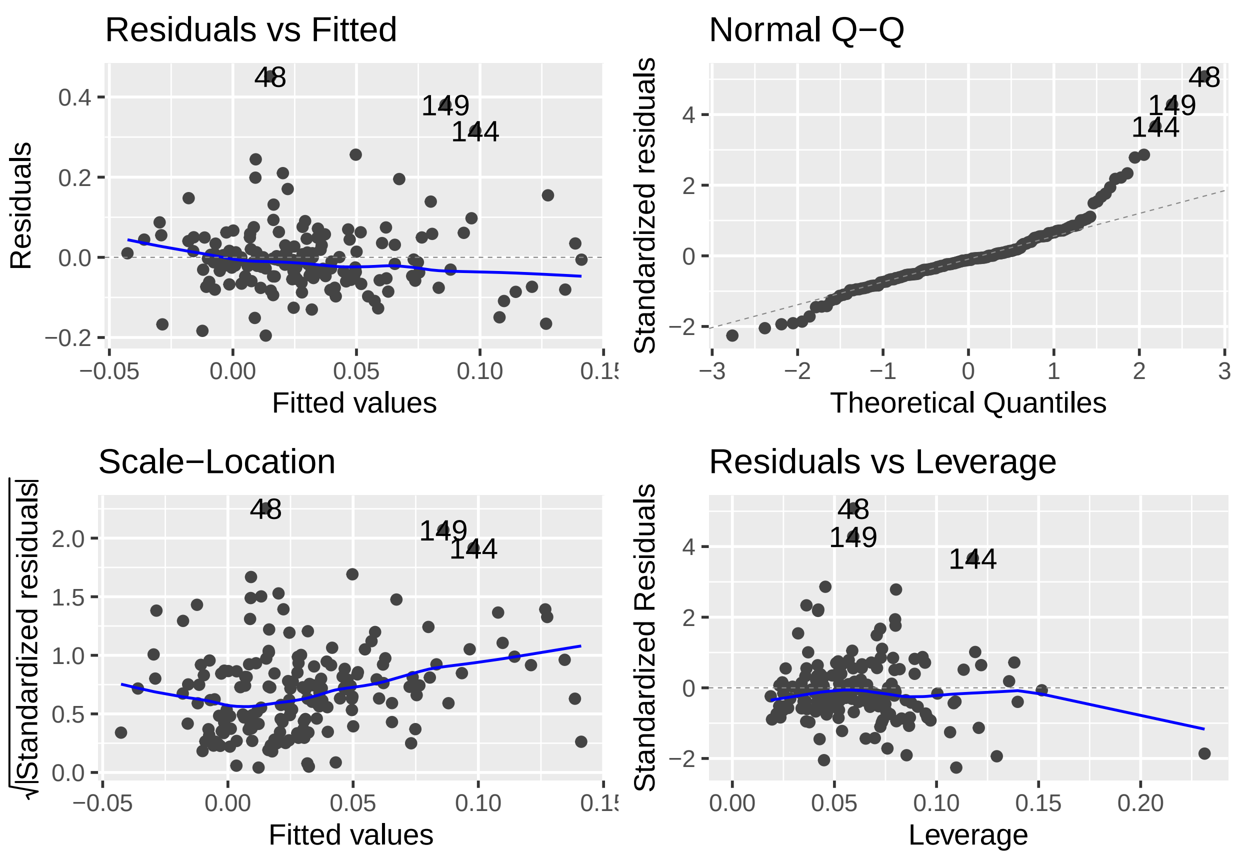
```
##           Variables Tolerance      VIF
## 1            LN_age 0.8091848 1.235812
## 2      LN_OfferSize 0.5804808 1.722710
```

```
## 3              Tech 0.9173188 1.090134
## 4              Rank 0.6399583 1.562602
## 5         Technique 0.8555027 1.168903
## 6         Sentiment 0.3350532 2.984600
## 7                HC 0.6334298 1.578707
## 8                VC 0.8026964 1.245801
## 9      Marketreturn 0.8129340 1.230112
## 10 Marketvolatility 0.4733008 2.112821
## 11        Dummy2014 0.3757544 2.661313
## 12        Dummy2015 0.5385819 1.856728
## 13        Dummy2016 0.6289370 1.589984
## 14        Dummy2017 0.5903334 1.693958
## 15        Dummy2018 0.5746005 1.740340
## 16        Dummy2019 0.8025739 1.245991
## 17        Dummy2020 0.4638905 2.155681
```

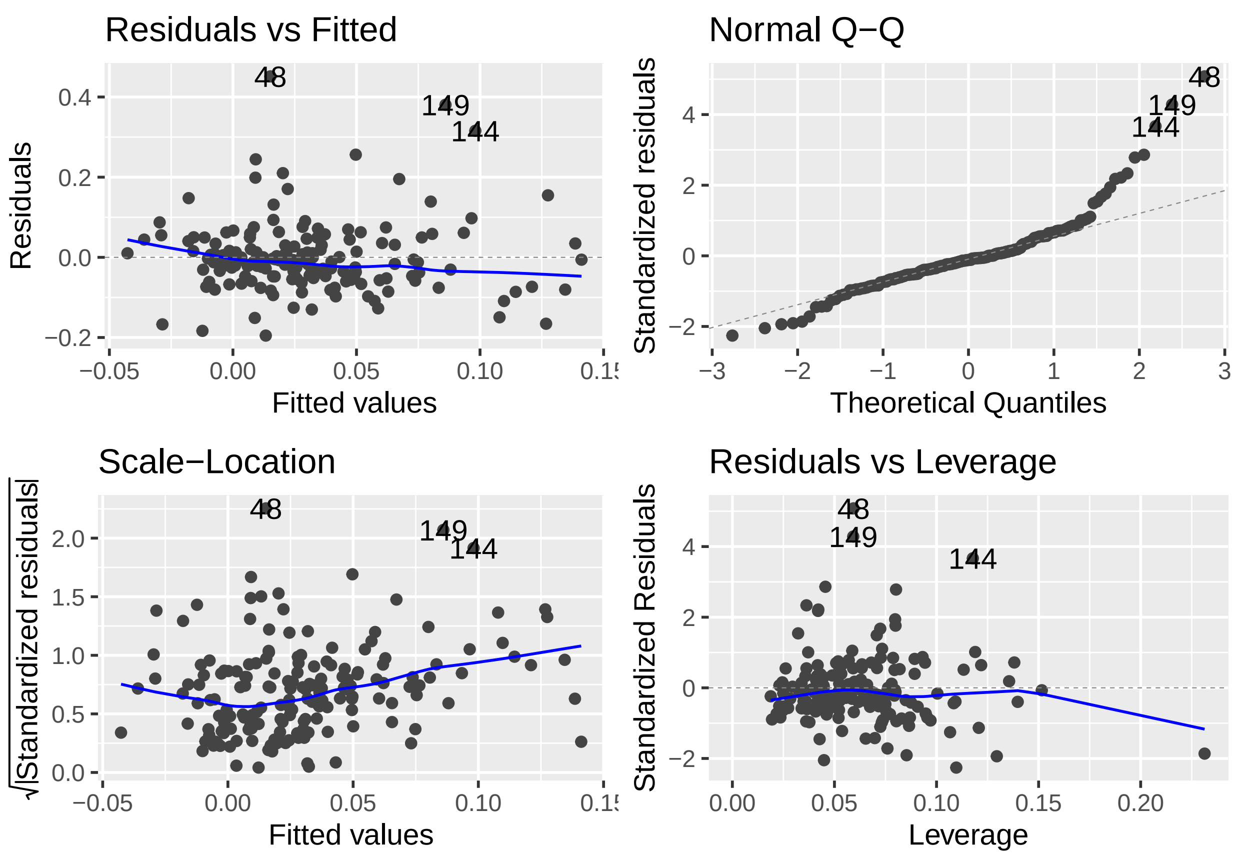
```
# Regression Model Diagnostics
autoplot(Model1_Euronext)
```




```
autoplot(Model2_Euronext)
```




```
autoplot(Model2_Euronext)
```




```r
# List of models

stargazer(Model1_Euronext, Model2_Euronext, Model3_Euronext, type = "text")
```

```
##
## ==============================================================================================
##                                          Dependent variable:
##                        -----------------------------------------------------------------------
##                                                  MAR
##                                  (1)                    (2)                    (3)
## ----------------------------------------------------------------------------------------------
## LN_age                         -0.008                 -0.008                 -0.006
##                                (0.008)                (0.008)                (0.008)
##
## LN_OfferSize                   0.012**                0.012**                0.011**
##                                (0.005)                (0.005)                (0.005)
##
## Tech                           -0.005                 -0.005                 -0.006
##                                (0.021)                (0.022)                (0.022)
##
## Rank                           -0.029*                -0.029*                -0.030*
##                                (0.017)                (0.018)                (0.017)
##
## Technique                       0.011                  0.011                  0.006
##                                (0.018)                (0.018)                (0.019)
##
## Sentiment                     -0.009***              -0.008***              -0.018***
```

```
##                          (0.002)              (0.002)               (0.004)
##
## HC                       -0.028*              -0.028*               -0.011
##                          (0.016)              (0.016)               (0.018)
##
## VC                        -0.018              -0.018                -0.010
##                          (0.018)              (0.018)               (0.019)
##
## Marketreturn                                   0.039                -0.085
##                                               (0.195)               (0.206)
##
## Marketvolatility                               0.138                -1.197
##                                               (2.116)               (2.734)
##
## Dummy2014                                                          -0.093***
##                                                                     (0.031)
##
## Dummy2015                                                           -0.019
##                                                                     (0.023)
##
## Dummy2016                                                          -0.059**
##                                                                     (0.028)
##
## Dummy2017                                                           -0.006
##                                                                     (0.032)
##
## Dummy2018                                                            0.023
##                                                                     (0.028)
##
## Dummy2019                                                           -0.005
##                                                                     (0.039)
##
## Dummy2020                                                          -0.092**
##                                                                     (0.039)
##
## Constant                  -0.043              -0.044               -0.084**
##                          (0.032)              (0.036)               (0.042)
##
## -------------------------------------------------------------------------------------------
## Observations               175                  175                   175
## R2                        0.135                0.136                 0.207
## Adjusted R2               0.094                0.083                 0.122
## Residual Std. Error    0.091 (df = 166)     0.092 (df = 164)      0.090 (df = 157)
## F Statistic         3.252*** (df = 8; 166) 2.575*** (df = 10; 164) 2.417*** (df = 17; 157)
## ===========================================================================================
## Note:                                                        *p<0.1; **p<0.05; ***p<0.01
```

```
## Regressions for Brussels

# Model 1

Model1_Brussels = lm(MAR ~ LN_age + LN_OfferSize + Tech + Technique
                     + Sentiment + HC + VC, data = Brussels)
```

```r
Model2_Brussels = lm(MAR ~ LN_age + LN_OfferSize + Tech + Technique
                     + Sentiment + HC + VC + Marketreturn + Marketvolatility, data = Brussels)

Model3_Brussels = lm(MAR ~ LN_age + LN_OfferSize + Tech + Technique
                     + Sentiment + HC + VC + Marketreturn + Marketvolatility
                     + Dummy2014 + Dummy2015 + Dummy2016 + Dummy2017 + Dummy2018 + Dummy2019, data = Bru


# Heteroscedasticity test --> if p-value > 0.05 == homosked

bptest(Model1_Brussels)     # p-value = 0.3092
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model1_Brussels
## BP = 8.2728, df = 7, p-value = 0.3092
```

```r
white_lm(Model1_Brussels)   # p-value = 0.783
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      9.02   0.830        14 White's Test greater
```

```r
bptest(Model2_Brussels)     # p-value = 0.4537
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model2_Brussels
## BP = 8.8233, df = 9, p-value = 0.4537
```

```r
white_lm(Model2_Brussels)   # p-value = 0.570
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      16.3   0.571        18 White's Test greater
```

```r
bptest(Model3_Brussels)     # p-value = 0.2061
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model3_Brussels
## BP = 19.17, df = 15, p-value = 0.2061
```

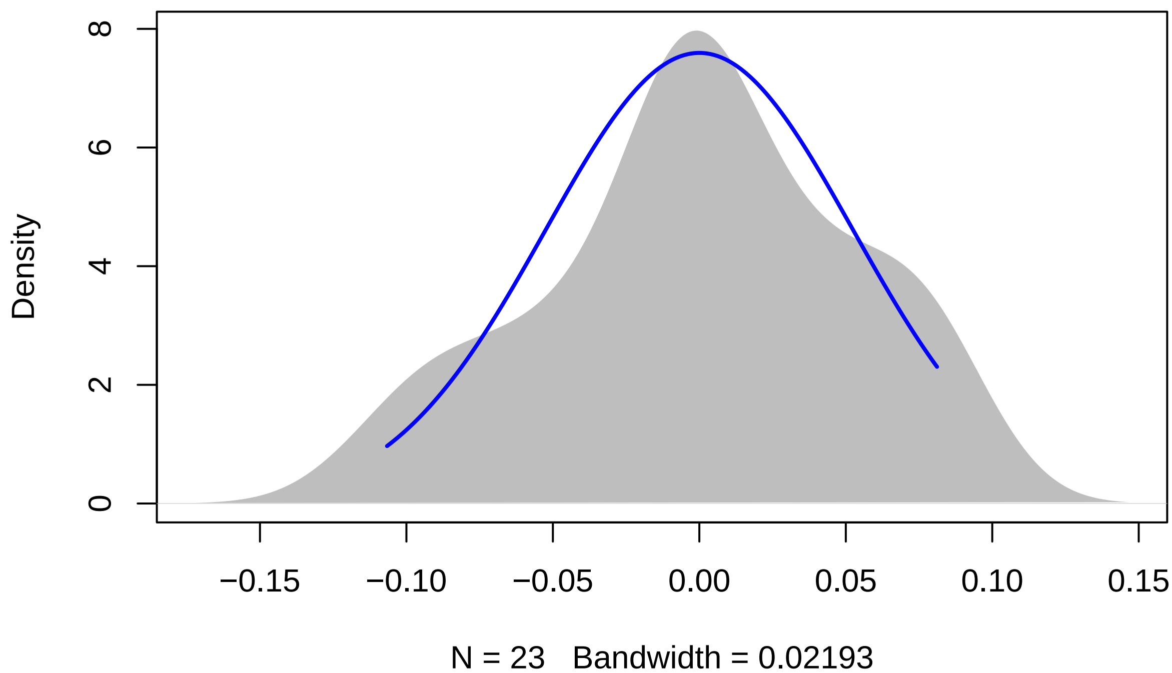
```r
white_lm(Model3_Brussels)   # p-value = 0.924
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      21.3   0.878        30 White's Test greater
```

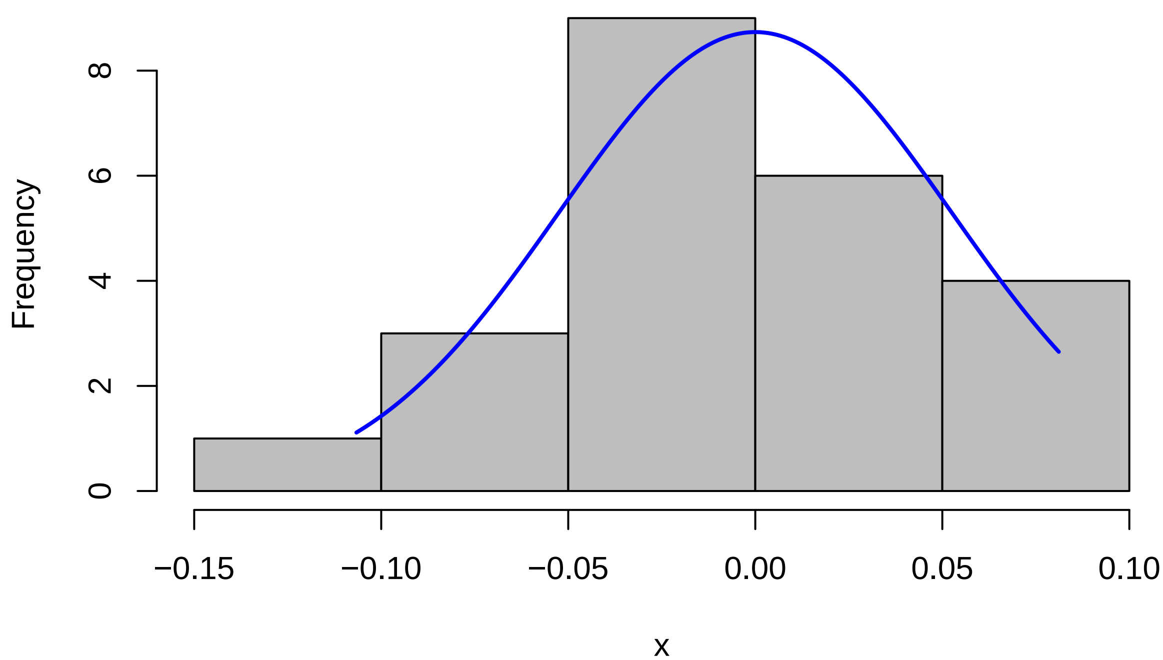
```r
# Normality of Residuals

jarque.bera.test(Model1_Brussels$residuals) # P-value = 0.7565 --> Normally distributed
```

```
##
##  Jarque Bera Test
##
## data:  Model1_Brussels$residuals
## X-squared = 0.5581, df = 2, p-value = 0.7565
```

```
plotNormalDensity(Model1_Brussels$residuals)
```




```
plotNormalHistogram(Model1_Brussels$residuals)
```

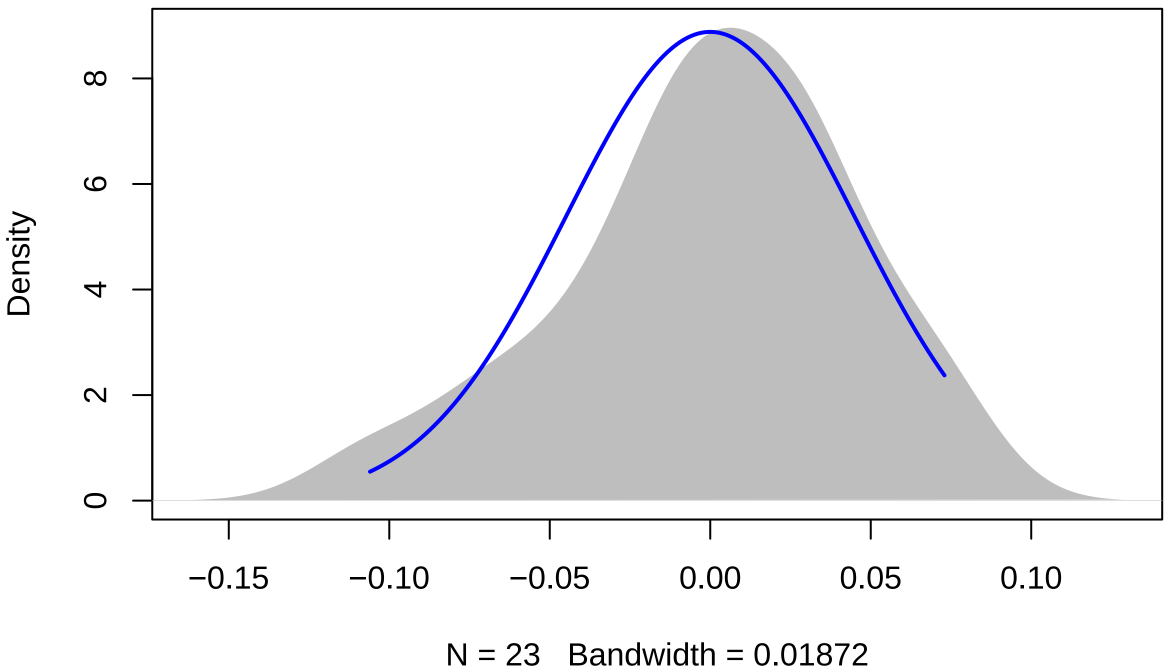



```
jarque.bera.test(Model2_Brussels$residuals) # P-value = 0.6202 --> Normally distributed
```

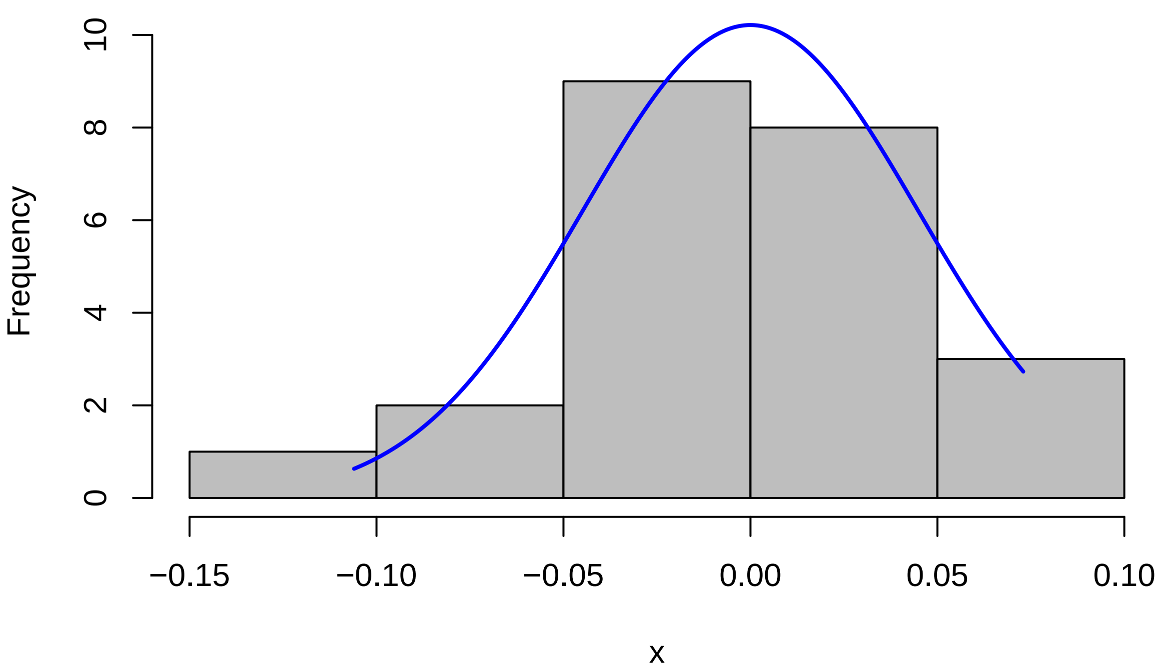
```
##
```

```
##  Jarque Bera Test
##
## data:  Model2_Brussels$residuals
## X-squared = 0.95555, df = 2, p-value = 0.6202
```

```r
plotNormalDensity(Model2_Brussels$residuals)
```




```r
plotNormalHistogram(Model2_Brussels$residuals)
```

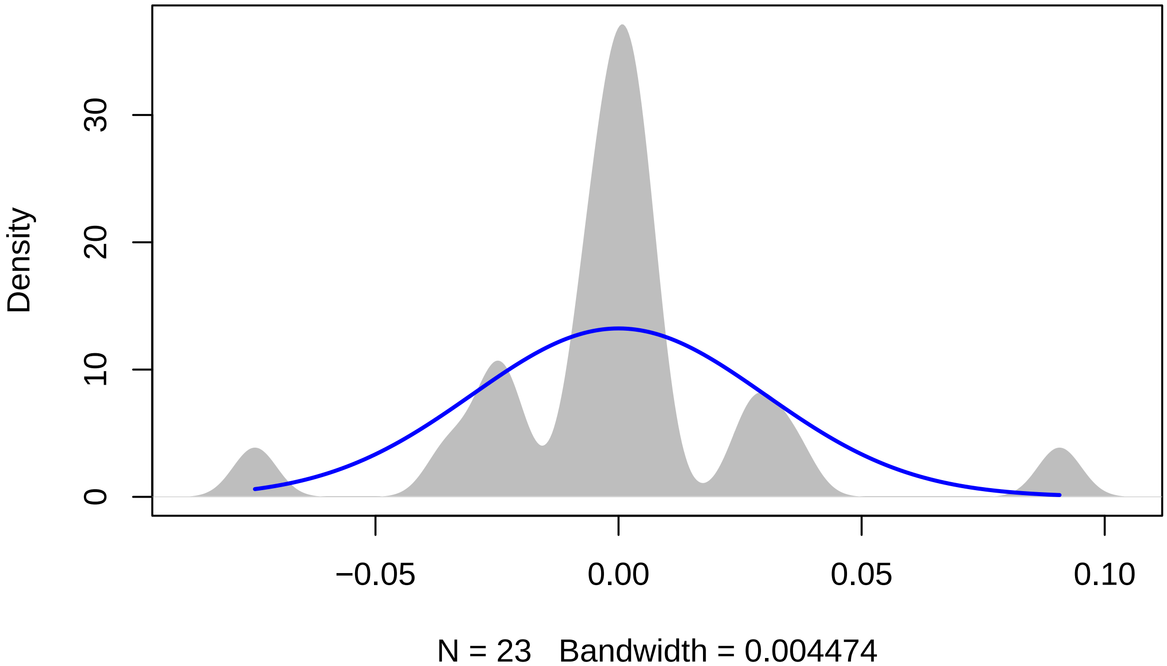



```r
jarque.bera.test(Model3_Brussels$residuals) # P-value = 0.007059 non-normal
```

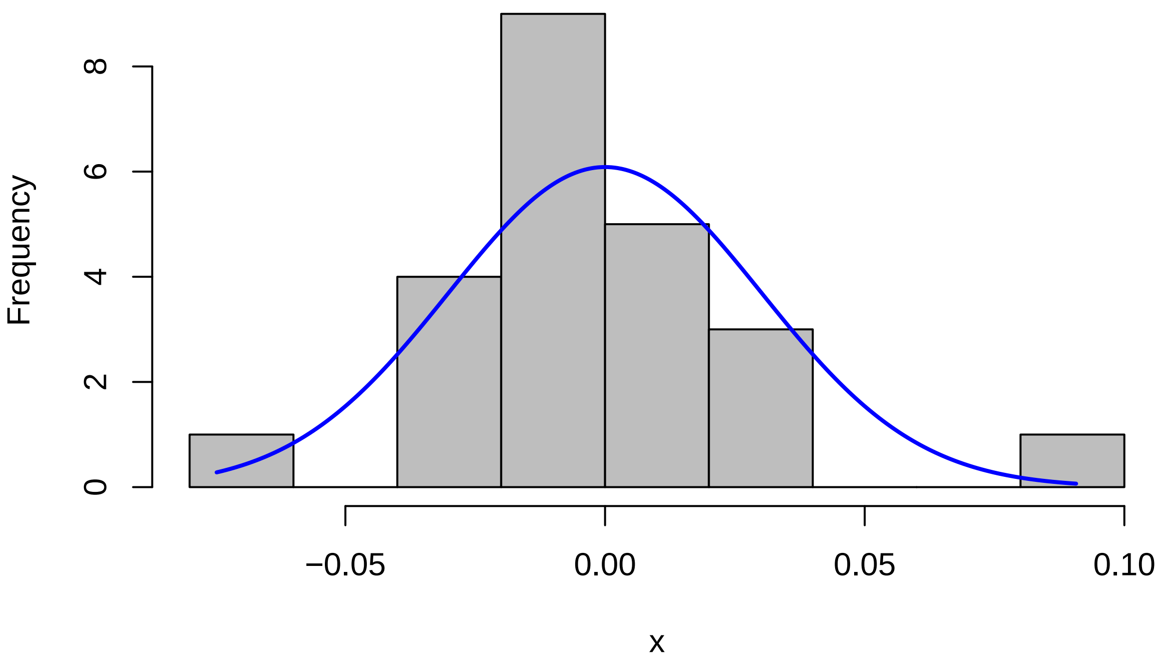
```
##
##  Jarque Bera Test
```

```
##
## data:  Model3_Brussels$residuals
## X-squared = 9.9069, df = 2, p-value = 0.007059
```

```
plotNormalDensity(Model3_Brussels$residuals)
```




```
plotNormalHistogram(Model3_Brussels$residuals)
```




```
# multicollinearity

ols_vif_tol(Model1_Brussels) # all good
```

```
##      Variables Tolerance      VIF
```

```
## 1       LN_age 0.8025892 1.245967
## 2 LN_OfferSize 0.7647422 1.307630
## 3         Tech 0.7111738 1.406126
## 4    Technique 0.8527585 1.172665
## 5    Sentiment 0.7388220 1.353506
## 6           HC 0.8368064 1.195020
## 7           VC 0.7989974 1.251568
```

```
ols_vif_tol(Model2_Brussels) # all good
```

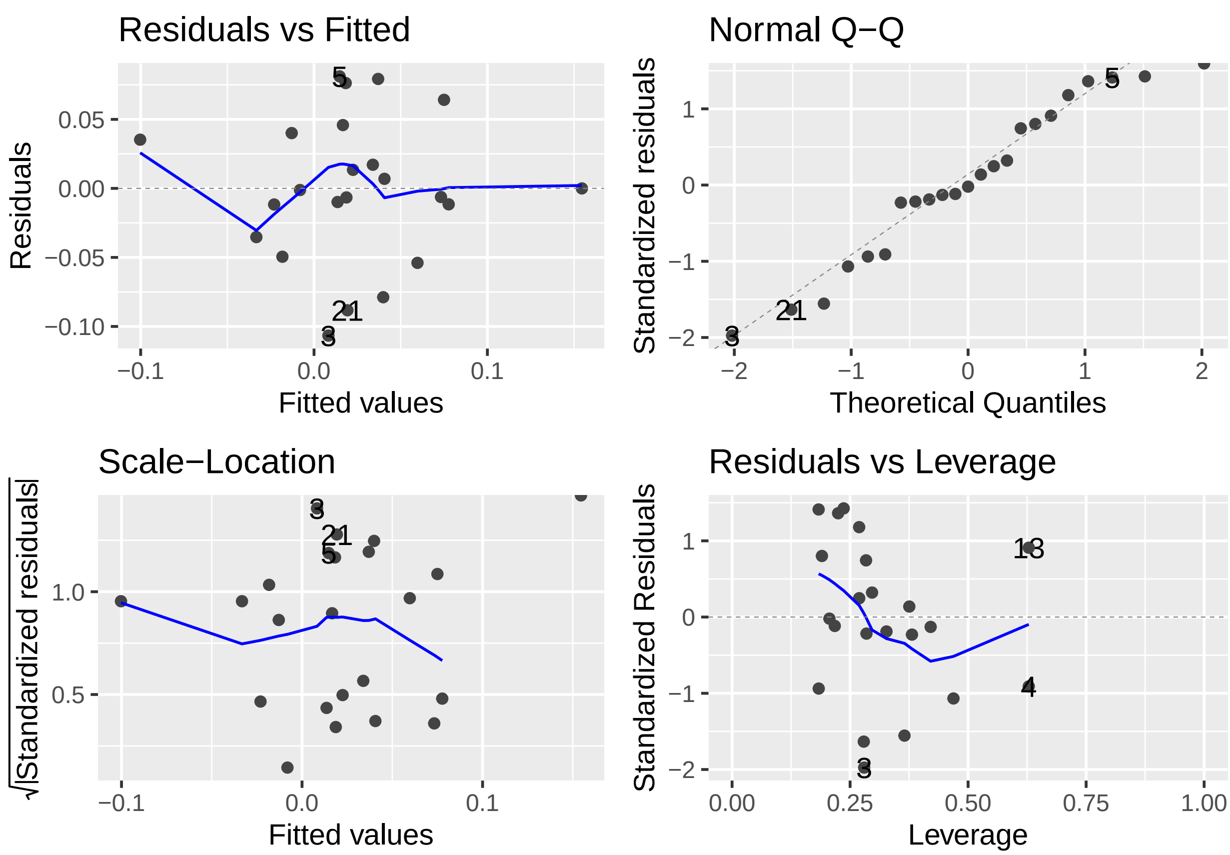
```
##          Variables Tolerance      VIF
## 1           LN_age 0.7042833 1.419883
## 2     LN_OfferSize 0.6562820 1.523735
## 3             Tech 0.6734732 1.484840
## 4        Technique 0.6519290 1.533909
## 5        Sentiment 0.6224381 1.606585
## 6               HC 0.7035693 1.421324
## 7               VC 0.5913819 1.690955
## 8     Marketreturn 0.4810467 2.078800
## 9 Marketvolatility 0.5538648 1.805495
```

```
ols_vif_tol(Model3_Brussels) # all good
```

```
##           Variables Tolerance      VIF
## 1            LN_age 0.3292165 3.037515
## 2      LN_OfferSize 0.4888681 2.045541
## 3              Tech 0.5412341 1.847629
## 4         Technique 0.3703445 2.700189
## 5         Sentiment 0.2907515 3.439364
## 6                HC 0.2551813 3.918783
## 7                VC 0.3594358 2.782138
## 8      Marketreturn 0.3309995 3.021152
## 9  Marketvolatility 0.3263181 3.064494
## 10        Dummy2014 0.3603907 2.774766
## 11        Dummy2015 0.4594982 2.176287
## 12        Dummy2016 0.4829814 2.070473
## 13        Dummy2017 0.2242209 4.459888
## 14        Dummy2018 0.3335248 2.998278
## 15        Dummy2019 0.7114904 1.405500
```

```
# Regression Model Diagnostics
autoplot(Model1_Brussels)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```

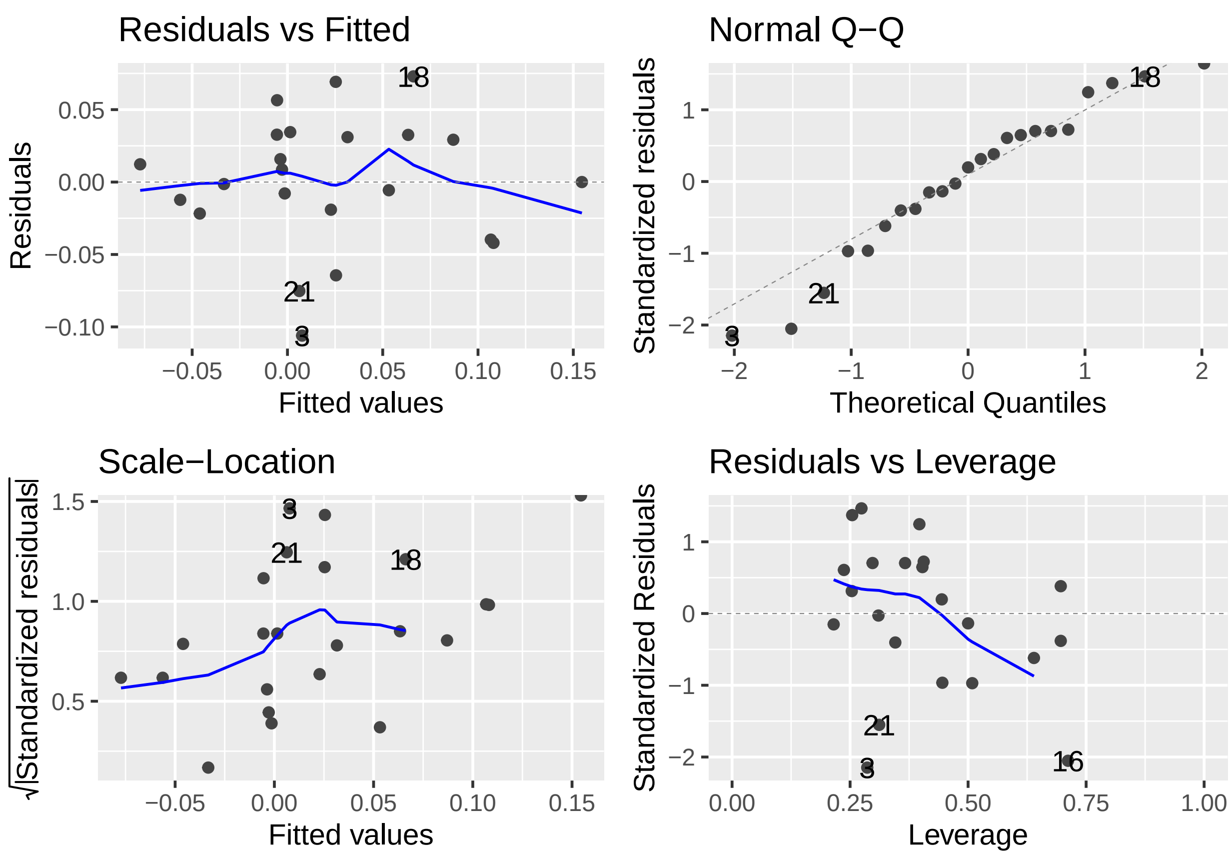



```
autoplot(Model2_Brussels)
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```

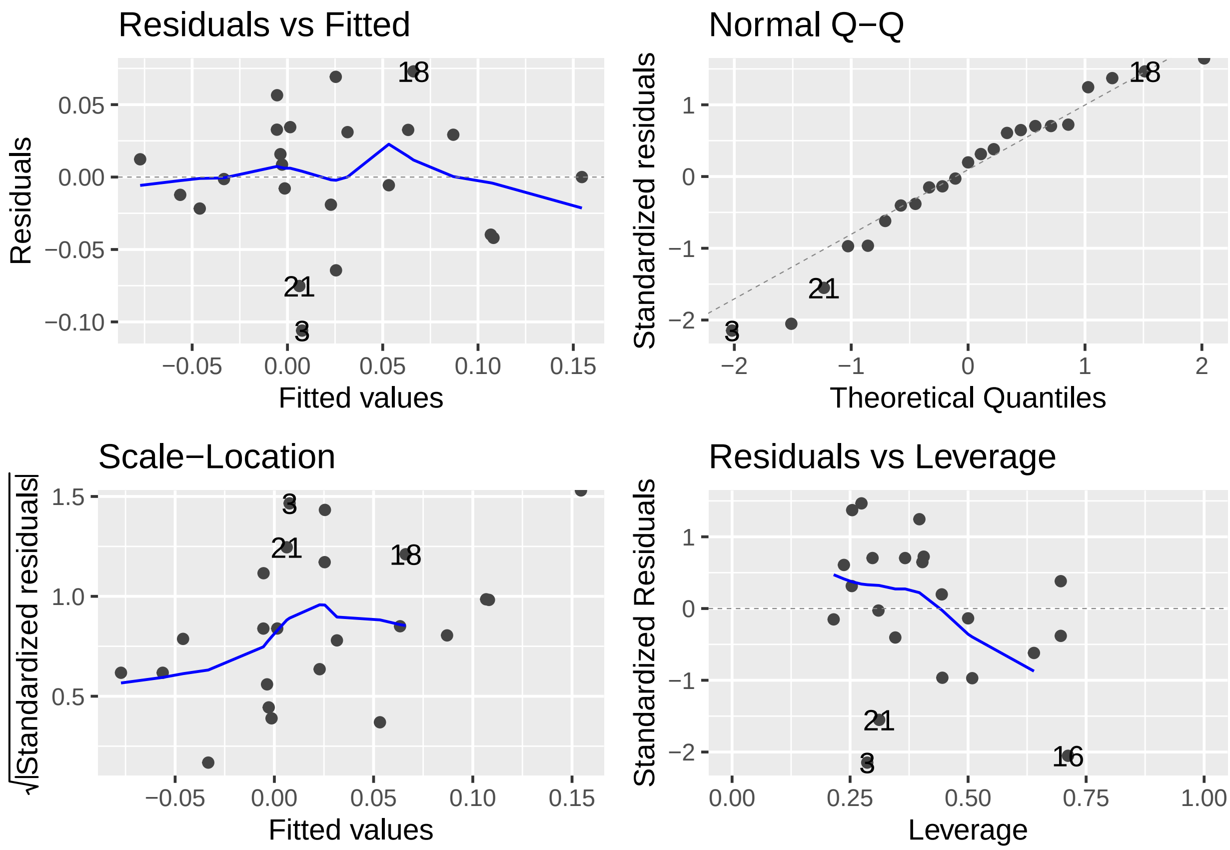
```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```




```
autoplot(Model2_Brussels)
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 4 row(s) containing missing values (geom_path).
```




```r
# List of models

stargazer(Model1_Brussels, Model2_Brussels, Model3_Brussels, type = "text")
```

```
##
## ===========================================================================
##                                    Dependent variable:
##                          --------------------------------------------------
##                                           MAR
##                                 (1)              (2)              (3)
## ---------------------------------------------------------------------------
## LN_age                        -0.020           -0.008           -0.005
##                              (0.023)          (0.022)          (0.030)
##
## LN_OfferSize                  0.022            0.030**          0.028*
##                              (0.013)          (0.013)          (0.014)
##
## Tech                          0.084            0.121            0.120
##                              (0.077)          (0.073)          (0.074)
##
## Technique                     -0.018           -0.002           -0.019
##                              (0.030)          (0.032)          (0.038)
##
## Sentiment                     -0.006           -0.003          -0.015**
##                              (0.005)          (0.005)          (0.006)
##
## HC                            0.019            0.044            0.056
```

```
##                          (0.029)             (0.029)            (0.044)
##
## VC                        -0.070              -0.013              0.012
##                          (0.053)             (0.056)            (0.066)
##
## Marketreturn                                  1.050**             1.138*
##                                              (0.486)            (0.536)
##
## Marketvolatility                              -0.759              -9.571
##                                              (4.945)            (5.893)
##
## Dummy2014                                                         -0.060
##                                                                 (0.049)
##
## Dummy2015                                                          0.080*
##                                                                 (0.037)
##
## Dummy2016                                                          0.022
##                                                                 (0.079)
##
## Dummy2017                                                          0.024
##                                                                 (0.115)
##
## Dummy2018                                                          0.107
##                                                                 (0.068)
##
## Dummy2019                                                          0.019
##                                                                 (0.065)
##
## Constant                  -0.072              -0.140              -0.183
##                          (0.101)             (0.116)            (0.115)
##
## ----------------------------------------------------------------------------
## Observations                23                  23                  23
## R2                         0.465               0.609               0.824
## Adjusted R2                0.215               0.337               0.446
## Residual Std. Error  0.064 (df = 15)     0.058 (df = 13)     0.053 (df = 7)
## F Statistic        1.859 (df = 7; 15) 2.245* (df = 9; 13) 2.179 (df = 15; 7)
## ============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

```r
## Regressions for Amsterdam

# Model 1

Model1_Amsterdam = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                      + Sentiment + HC + VC, data = Amsterdam)

Model2_Amsterdam = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                      + Sentiment + HC + VC + Marketreturn + Marketvolatility, data = Amsterdam)

Model3_Amsterdam = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                      + Sentiment + HC + VC + Marketreturn + Marketvolatility
                      + Dummy2014 + Dummy2015 + Dummy2016 + Dummy2017 + Dummy2018 + Dummy2019 + Dummy202
```

```r
# Heteroscedasticity test --> if p-value > 0.05 == homosked

bptest(Model1_Amsterdam)     # p-value = 0.5693
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model1_Amsterdam
## BP = 6.7001, df = 8, p-value = 0.5693
```

```r
white_lm(Model1_Amsterdam)   # p-value = 0.954
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      7.81   0.954        16 White's Test greater
```

```r
bptest(Model2_Amsterdam)     # p-value = 0.5535
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model2_Amsterdam
## BP = 8.7759, df = 10, p-value = 0.5535
```

```r
white_lm(Model2_Amsterdam)   # p-value = 0.660
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      16.9   0.660        20 White's Test greater
```

```r
bptest(Model3_Amsterdam)     # p-value = 0.2895
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model3_Amsterdam
## BP = 19.707, df = 17, p-value = 0.2895
```

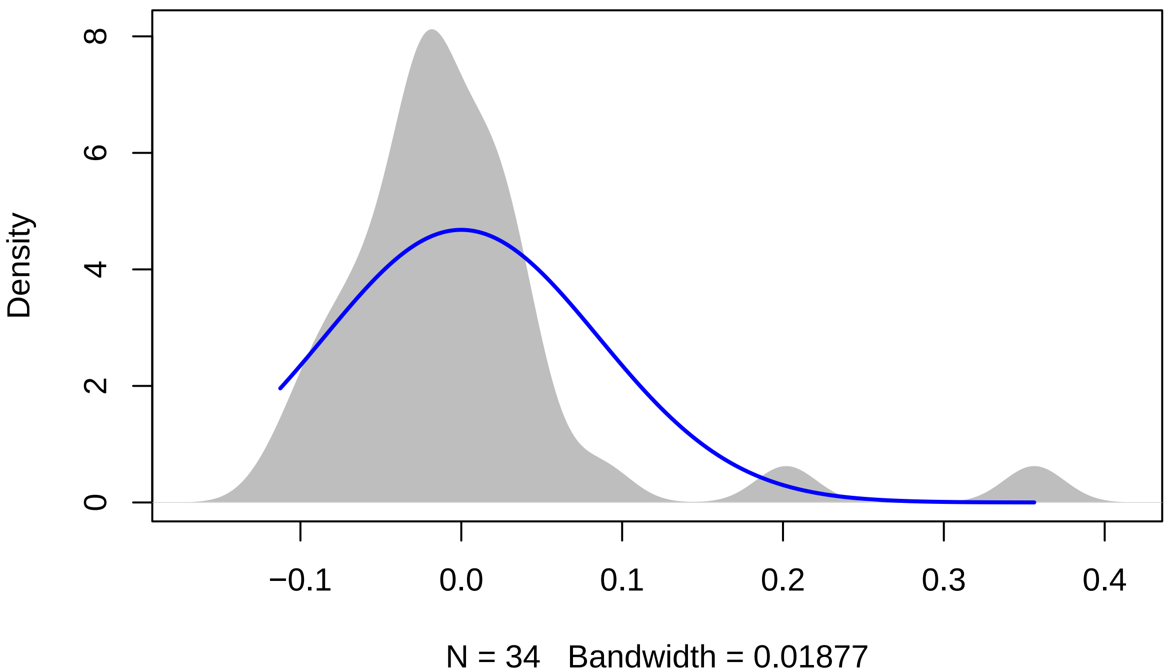
```r
white_lm(Model3_Amsterdam)   # p-value = 0.796
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      27.0   0.796        34 White's Test greater
```

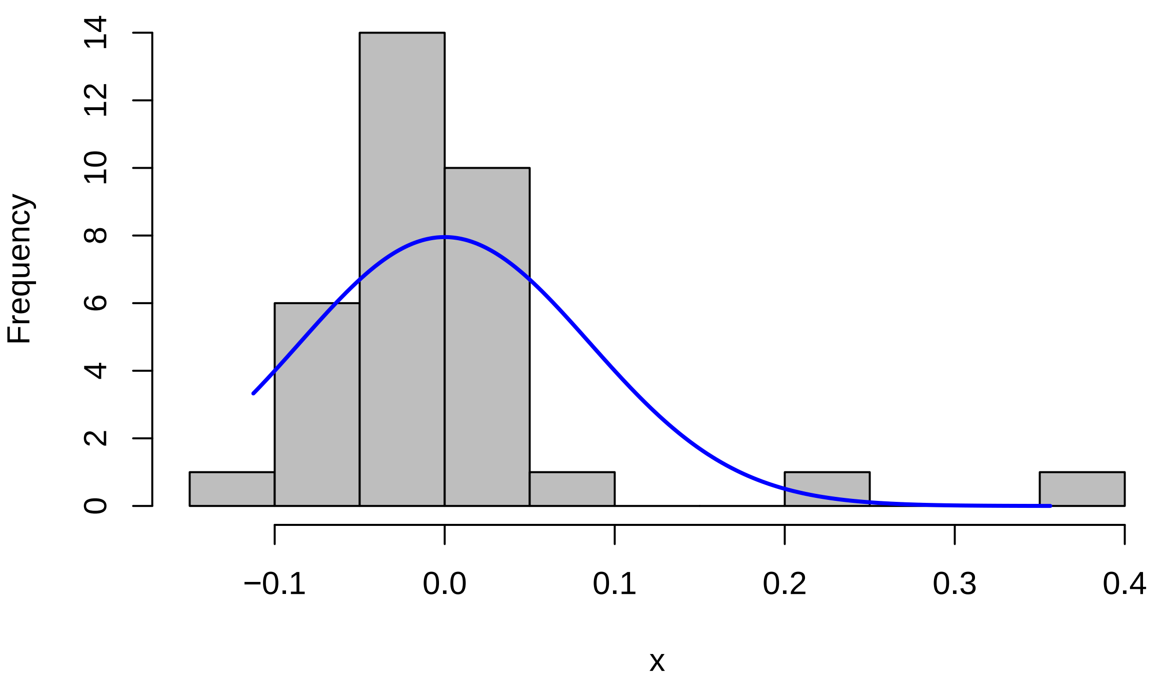
```r
# Normality of Residuals

jarque.bera.test(Model1_Amsterdam$residuals) # P-value < 2.2e-16 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model1_Amsterdam$residuals
## X-squared = 120.1, df = 2, p-value < 2.2e-16
```

```r
plotNormalDensity(Model1_Amsterdam$residuals)
```



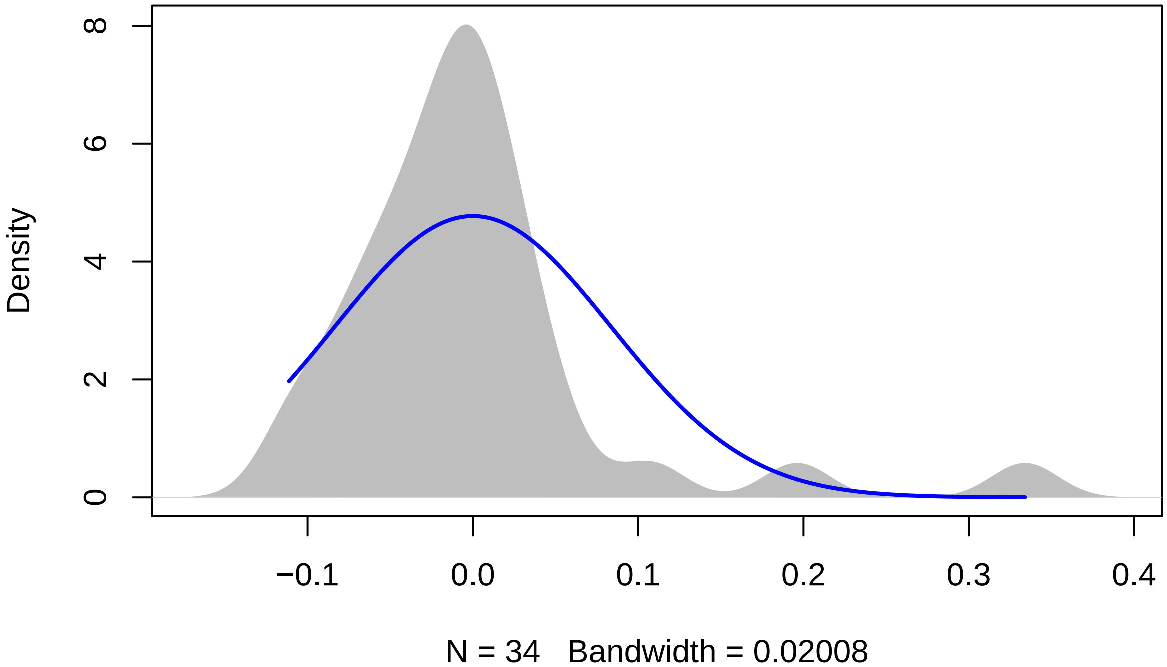

```r
plotNormalHistogram(Model1_Amsterdam$residuals)
```



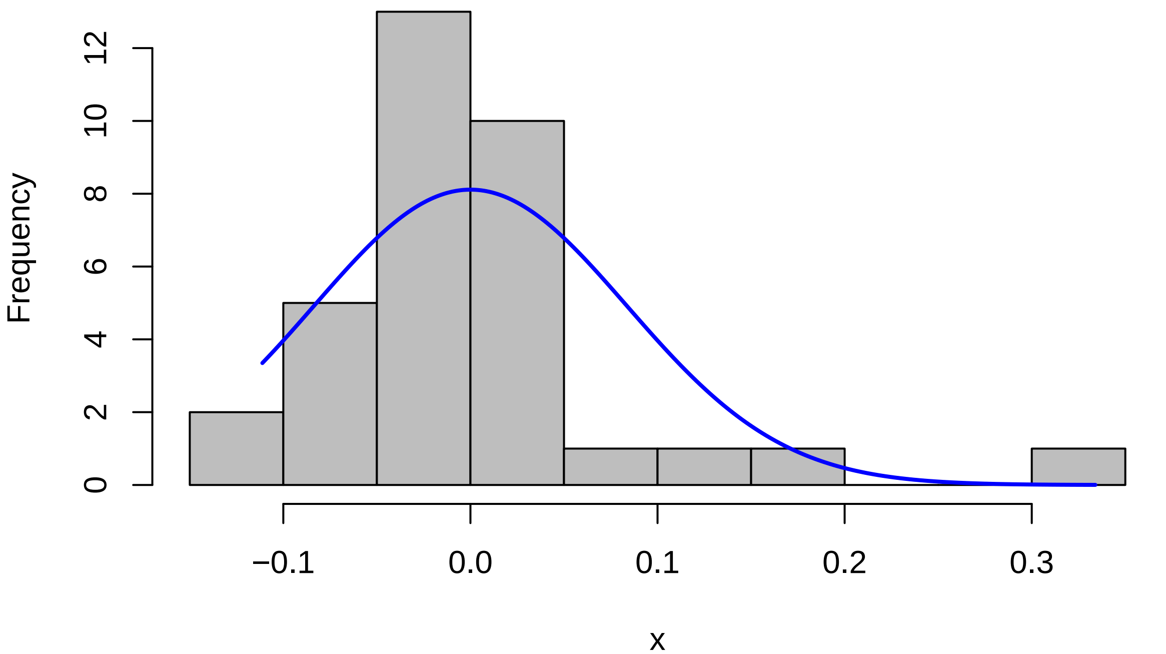

```r
jarque.bera.test(Model2_Amsterdam$residuals) # P-value < 2.2e-16 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model2_Amsterdam$residuals
## X-squared = 82.907, df = 2, p-value < 2.2e-16
```

```
plotNormalDensity(Model2_Amsterdam$residuals)
```



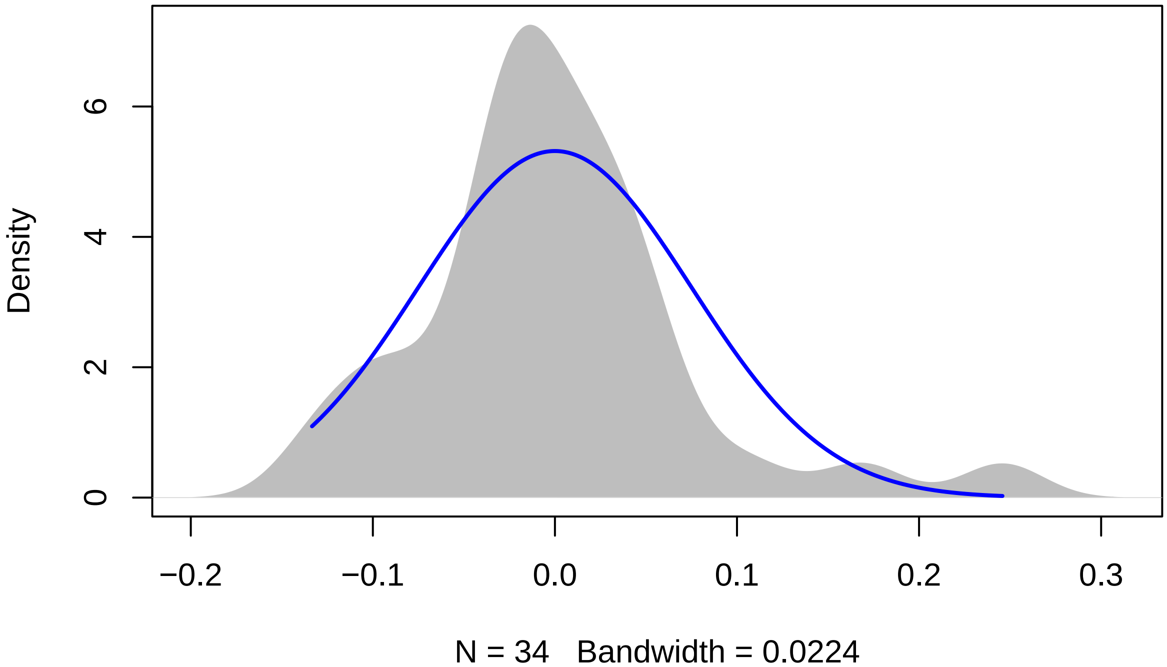

```
plotNormalHistogram(Model2_Amsterdam$residuals)
```



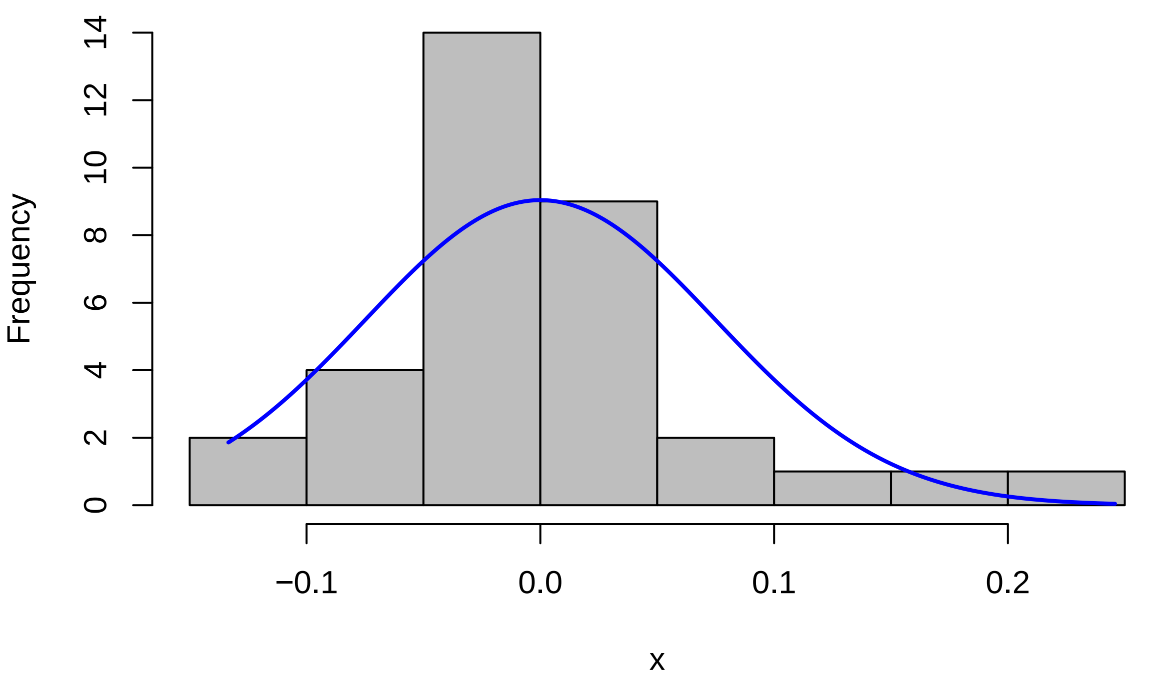

```
jarque.bera.test(Model3_Amsterdam$residuals) # P-value = 0.0008986 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model3_Amsterdam$residuals
## X-squared = 14.029, df = 2, p-value = 0.0008986
```

```
plotNormalDensity(Model3_Amsterdam$residuals)
```




```
plotNormalHistogram(Model3_Amsterdam$residuals)
```




```
# multicollinearity

ols_vif_tol(Model1_Amsterdam) # all good
```

```
##      Variables Tolerance      VIF
## 1       LN_age 0.8033542 1.244781
## 2 LN_OfferSize 0.7629029 1.310783
```

```
## 3         Tech 0.8388964 1.192042
## 4         Rank 0.8463132 1.181596
## 5    Technique 0.8290364 1.206220
## 6    Sentiment 0.7889789 1.267461
## 7           HC 0.8139337 1.228601
## 8           VC 0.7212448 1.386492
```

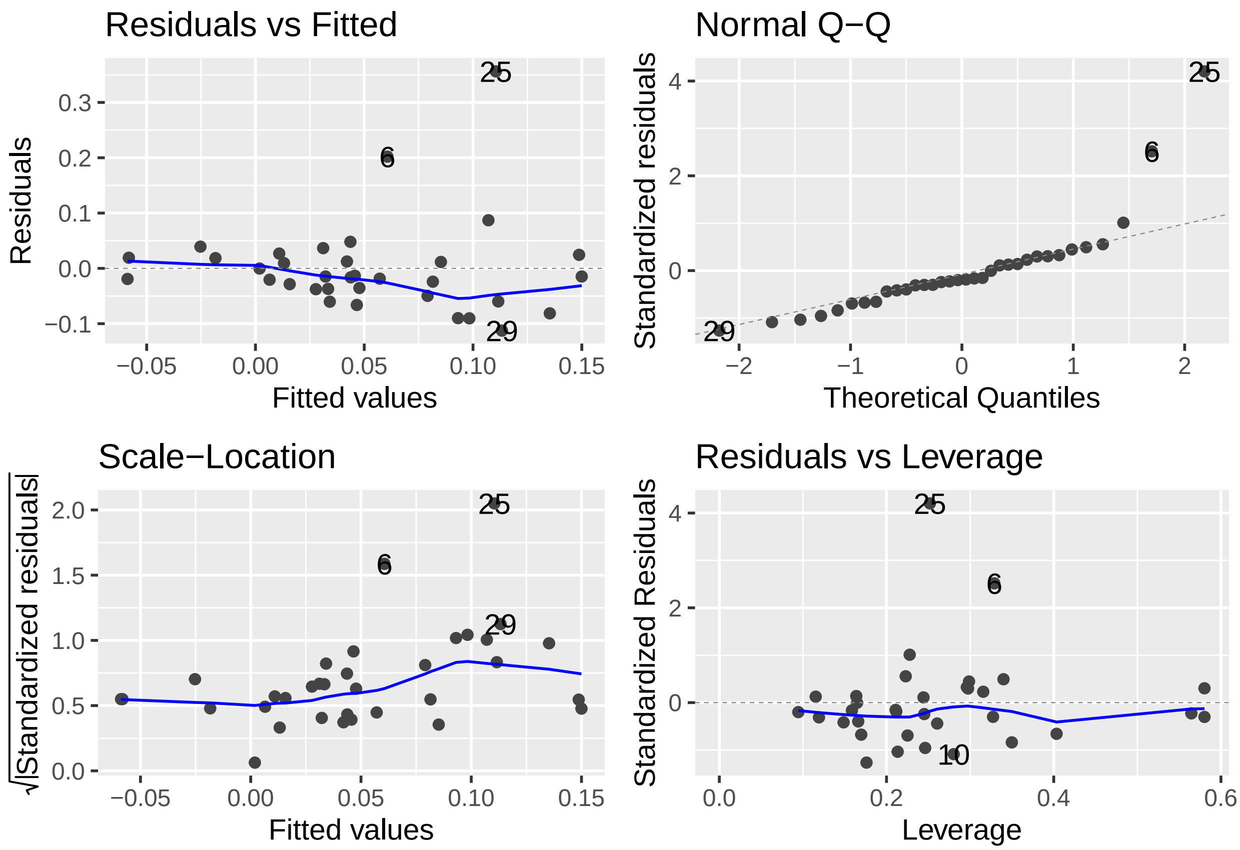
```
ols_vif_tol(Model2_Amsterdam) # all good
```

```
##            Variables Tolerance      VIF
## 1             LN_age 0.7473669 1.338031
## 2       LN_OfferSize 0.6807690 1.468927
## 3               Tech 0.7802284 1.281676
## 4               Rank 0.7897065 1.266293
## 5          Technique 0.5163961 1.936498
## 6          Sentiment 0.5849319 1.709601
## 7                 HC 0.7223560 1.384359
## 8                 VC 0.7027634 1.422954
## 9       Marketreturn 0.8312118 1.203063
## 10 Marketvolatility 0.5009108 1.996364
```

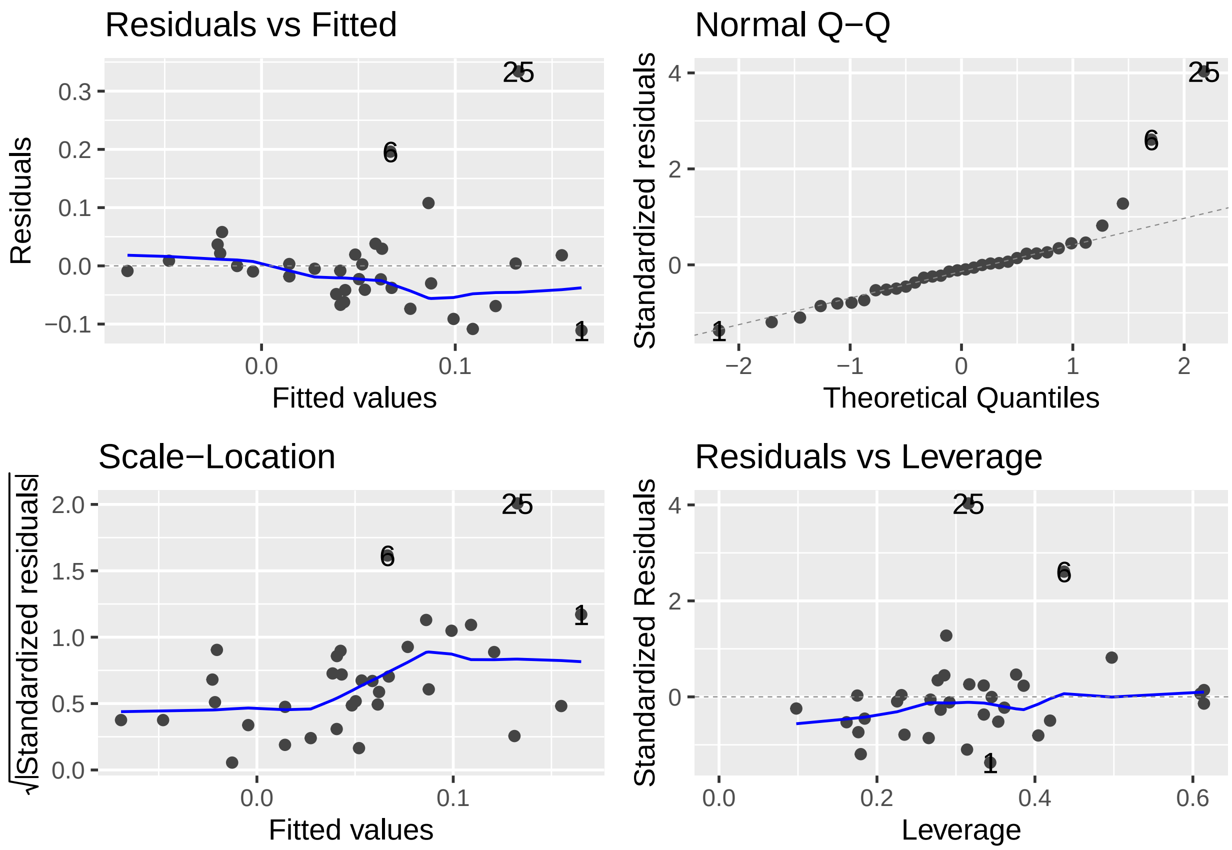
```
ols_vif_tol(Model3_Amsterdam) # all good
```

```
##            Variables Tolerance      VIF
## 1             LN_age 0.4811773 2.078236
## 2       LN_OfferSize 0.4993968 2.002416
## 3               Tech 0.5032893 1.986929
## 4               Rank 0.6070780 1.647235
## 5          Technique 0.3150645 3.173953
## 6          Sentiment 0.2578965 3.877525
## 7                 HC 0.2532064 3.949348
## 8                 VC 0.5130155 1.949259
## 9       Marketreturn 0.3388763 2.950930
## 10 Marketvolatility 0.1985284 5.037064
## 11         Dummy2014 0.2050187 4.877605
## 12         Dummy2015 0.2384610 4.193557
## 13         Dummy2016 0.2268081 4.409013
## 14         Dummy2017 0.4539348 2.202959
## 15         Dummy2018 0.2779447 3.597838
## 16         Dummy2019 0.6894026 1.450531
## 17         Dummy2020 0.3027563 3.302987
```

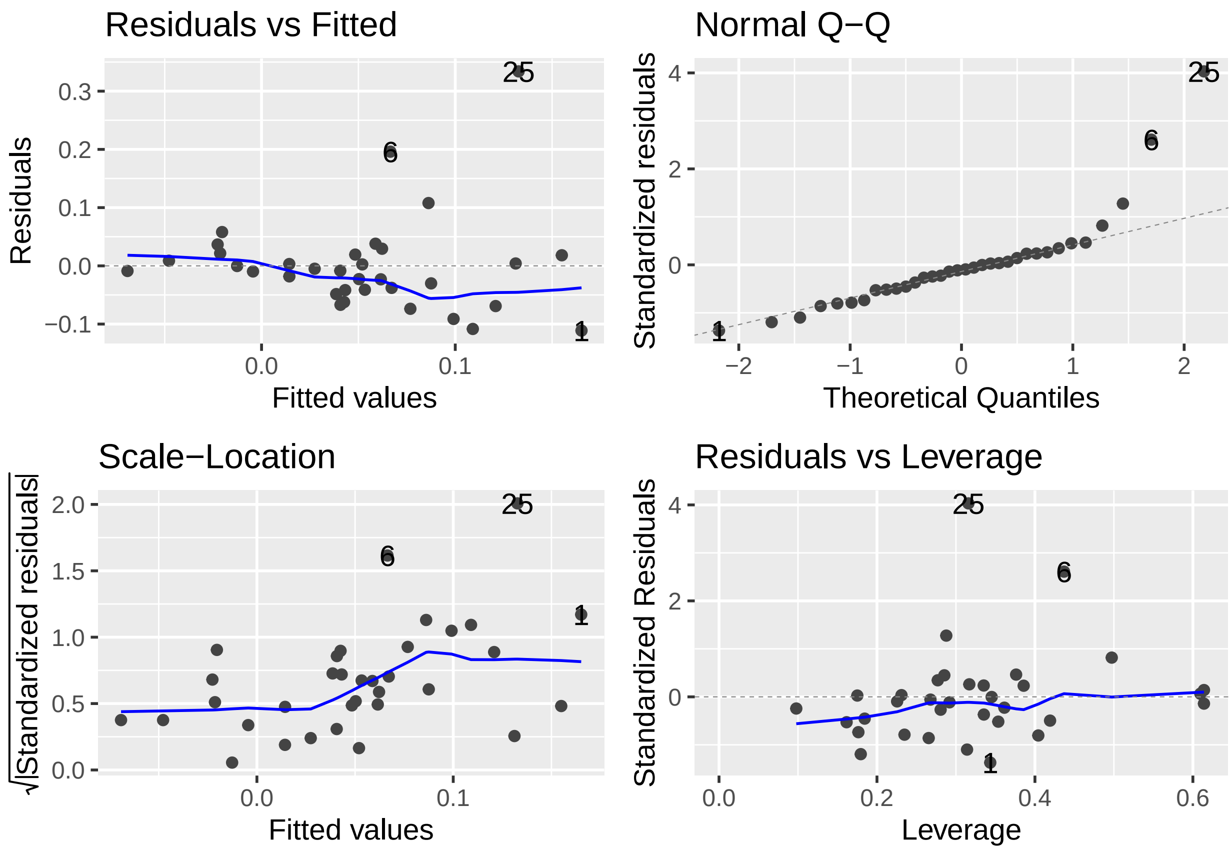
```
# Regression Model Diagnostics
autoplot(Model1_Amsterdam)
```




```
autoplot(Model2_Amsterdam)
```




```
autoplot(Model2_Amsterdam)
```




```r
# List of models

stargazer(Model1_Amsterdam, Model2_Amsterdam, Model3_Amsterdam, type = "text")
```

```
##
## ============================================================================
##                                       Dependent variable:
##                           --------------------------------------------------
##                                              MAR
##                                 (1)              (2)              (3)
## ----------------------------------------------------------------------------
## LN_age                       -0.006           -0.001           -0.010
##                              (0.017)          (0.018)          (0.025)
##
## LN_OfferSize                  0.015            0.014            0.029
##                              (0.013)          (0.015)          (0.018)
##
## Tech                         -0.097           -0.092           -0.016
##                              (0.078)          (0.083)          (0.111)
##
## Rank                         -0.014           -0.009           -0.023
##                              (0.040)          (0.042)          (0.052)
##
## Technique                    -0.021           -0.046           -0.051
##                              (0.048)          (0.063)          (0.086)
##
## Sentiment                    -0.006           -0.009           -0.010
```

```
##                          (0.006)            (0.007)            (0.011)
##
## HC                        -0.045             -0.033              0.013
##                          (0.038)            (0.042)            (0.076)
##
## VC                         0.045              0.054              0.094
##                          (0.052)            (0.054)            (0.068)
##
## Marketreturn                                 -0.440              0.369
##                                             (0.597)            (1.006)
##
## Marketvolatility                             -5.787              7.527
##                                             (8.061)           (13.778)
##
## Dummy2014                                                        0.094
##                                                                (0.107)
##
## Dummy2015                                                       -0.043
##                                                                (0.089)
##
## Dummy2016                                                       -0.005
##                                                                (0.110)
##
## Dummy2017                                                        0.047
##                                                                (0.117)
##
## Dummy2018                                                        0.149
##                                                                (0.109)
##
## Dummy2019                                                        0.049
##                                                                (0.132)
##
## Dummy2020                                                       -0.078
##                                                                (0.199)
##
## Constant                  -0.043             -0.015             -0.261
##                          (0.108)            (0.123)            (0.227)
##
## ----------------------------------------------------------------------------
## Observations                34                 34                 34
## R2                         0.280              0.307              0.442
## Adjusted R2                0.049              0.006             -0.151
## Residual Std. Error  0.098 (df = 25)     0.100 (df = 23)     0.108 (df = 16)
## F Statistic         1.213 (df = 8; 25) 1.021 (df = 10; 23) 0.746 (df = 17; 16)
## ============================================================================
## Note:                                        *p<0.1; **p<0.05; ***p<0.01
```

```
## Regressions for Paris

# Model 1

Model1_Paris = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                  + Sentiment + HC + VC, data = Paris)
```

```r
Model2_Paris = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                  + Sentiment + HC + VC + Marketreturn + Marketvolatility, data = Paris)

Model3_Paris = lm(MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique
                  + Sentiment + HC + VC + Marketreturn + Marketvolatility
                  + Dummy2014 + Dummy2015 + Dummy2016 + Dummy2017 + Dummy2018 + Dummy2019 + Dummy2020, d


# Heteroscedasticity test --> if p-value > 0.05 == homosked

bptest(Model1_Paris)     # p-value = 0.0358
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model1_Paris
## BP = 16.496, df = 8, p-value = 0.0358
```

```r
white_lm(Model1_Paris)    # p-value = 0.0563
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      25.8  0.0563        16 White's Test greater
```

```r
bptest(Model2_Paris)     # p-value = 0.04007
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model2_Paris
## BP = 19.015, df = 10, p-value = 0.04007
```

```r
white_lm(Model2_Paris)    # p-value = 0.0168
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      35.7  0.0168        20 White's Test greater
```

```r
bptest(Model3_Paris)     # p-value = 0.02306
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Model3_Paris
## BP = 30.486, df = 17, p-value = 0.02305
```

```r
white_lm(Model3_Paris)    # p-value = 0.337
```

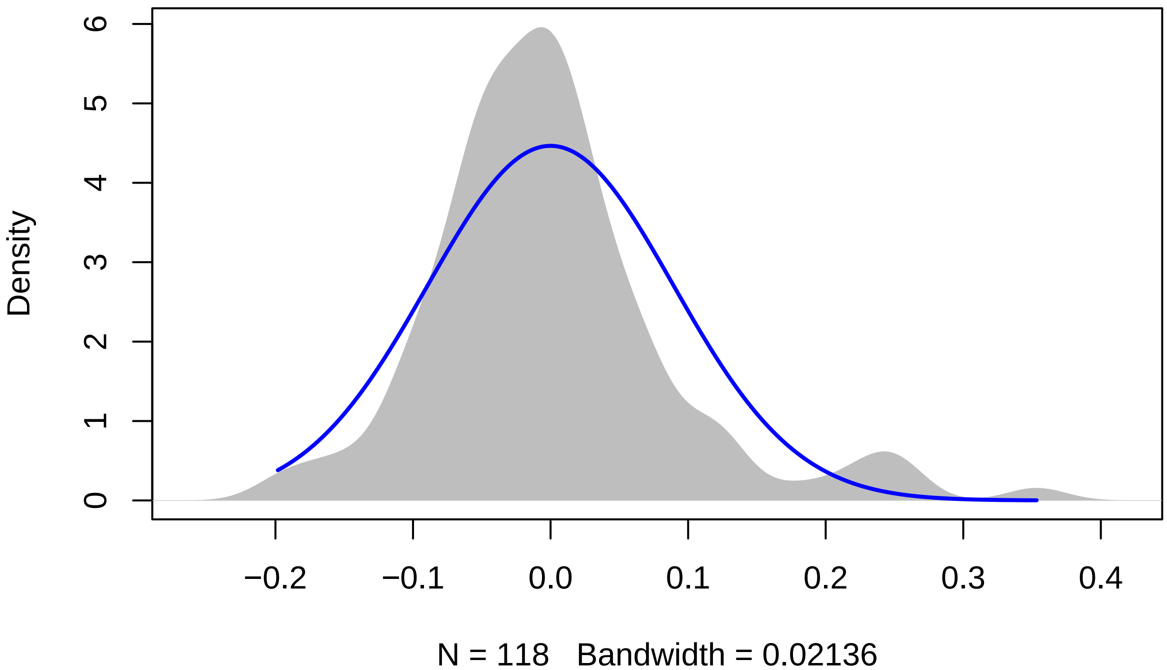
```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      36.9   0.337        34 White's Test greater
```

```r
# Heteroscedasticity robust models

Model1_Paris_robust = coeftest(Model1_Paris, vcov = vcovHC(Model1_Paris, type = 'HC0'))
```

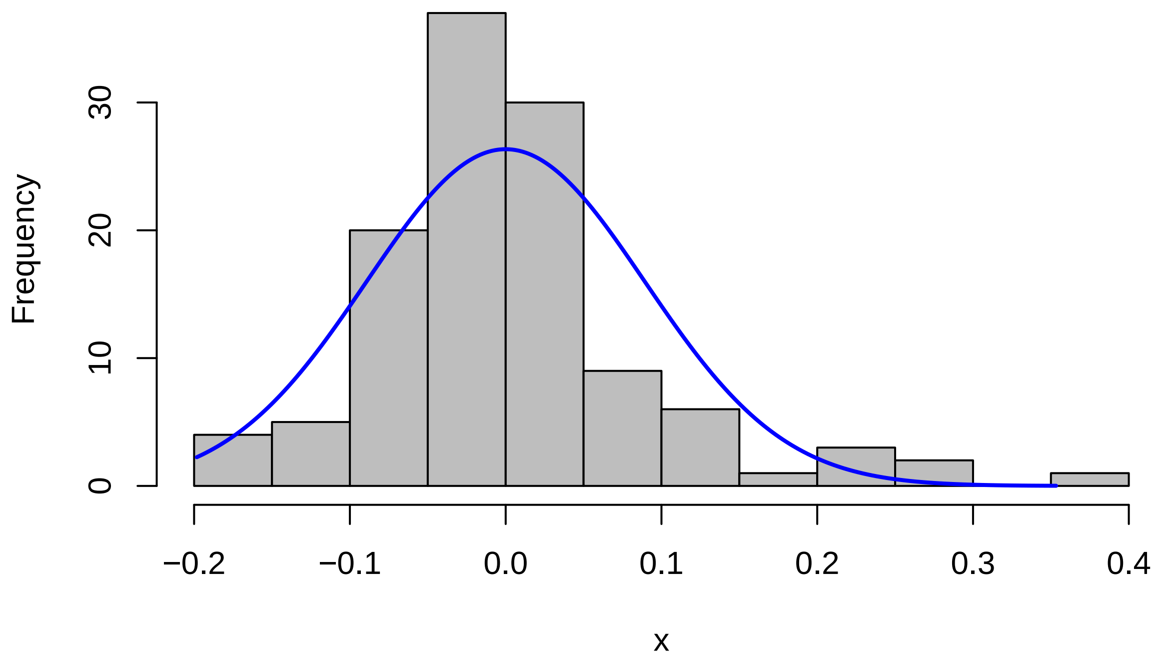
```
Model2_Paris_robust = coeftest(Model2_Paris, vcov = vcovHC(Model2_Paris, type = 'HC0'))
Model3_Paris_robust = coeftest(Model3_Paris, vcov = vcovHC(Model3_Paris, type = 'HC0'))

# Normality of Residuals

jarque.bera.test(Model1_Paris$residuals) # P-value < 1.454e-12 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model1_Paris$residuals
## X-squared = 54.513, df = 2, p-value = 1.454e-12
```

```
plotNormalDensity(Model1_Paris$residuals)
```



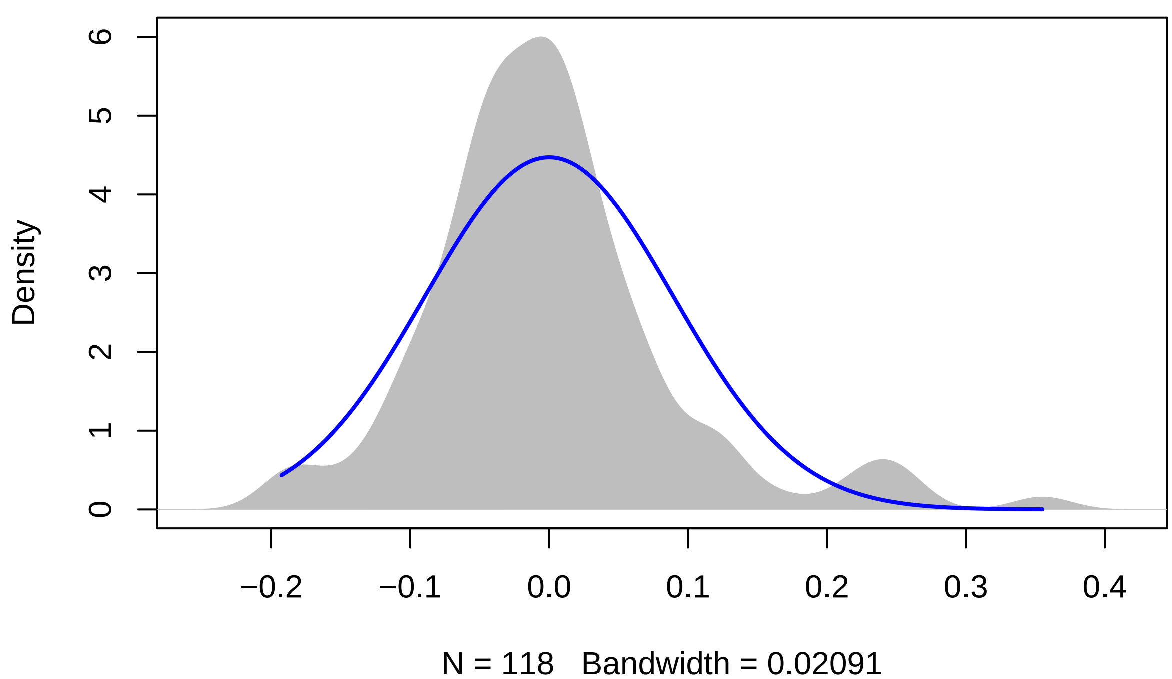

```
plotNormalHistogram(Model1_Paris$residuals)
```



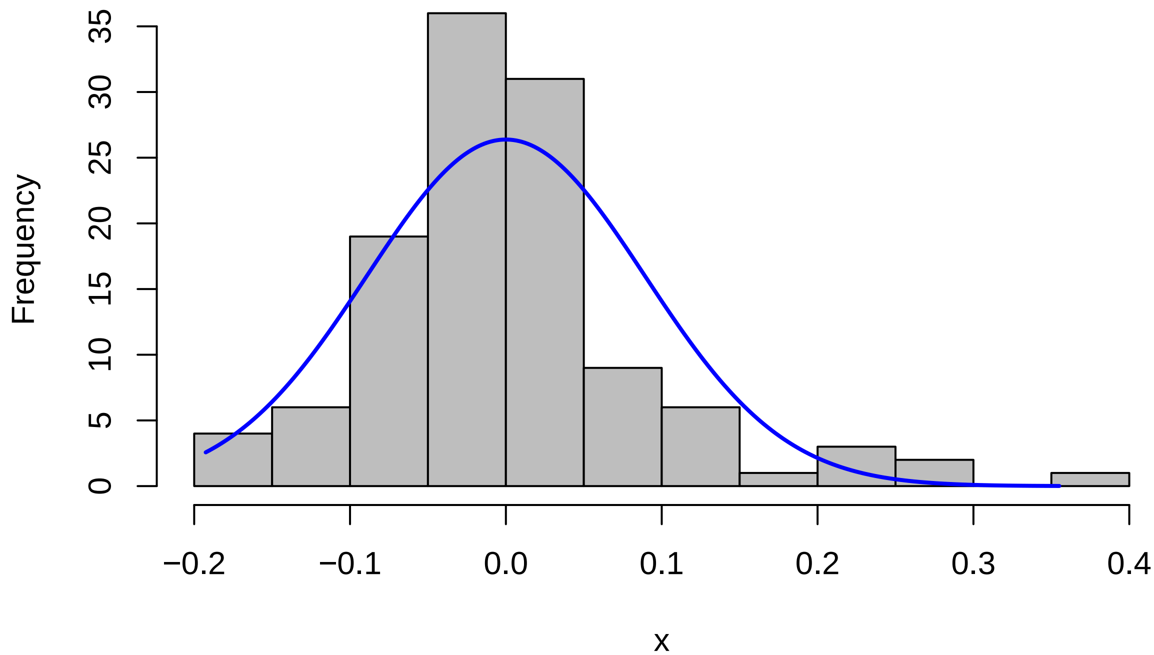

```
jarque.bera.test(Model2_Paris$residuals) # P-value < 1.356e-12 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model2_Paris$residuals
## X-squared = 54.653, df = 2, p-value = 1.356e-12
```

```
plotNormalDensity(Model2_Paris$residuals)
```



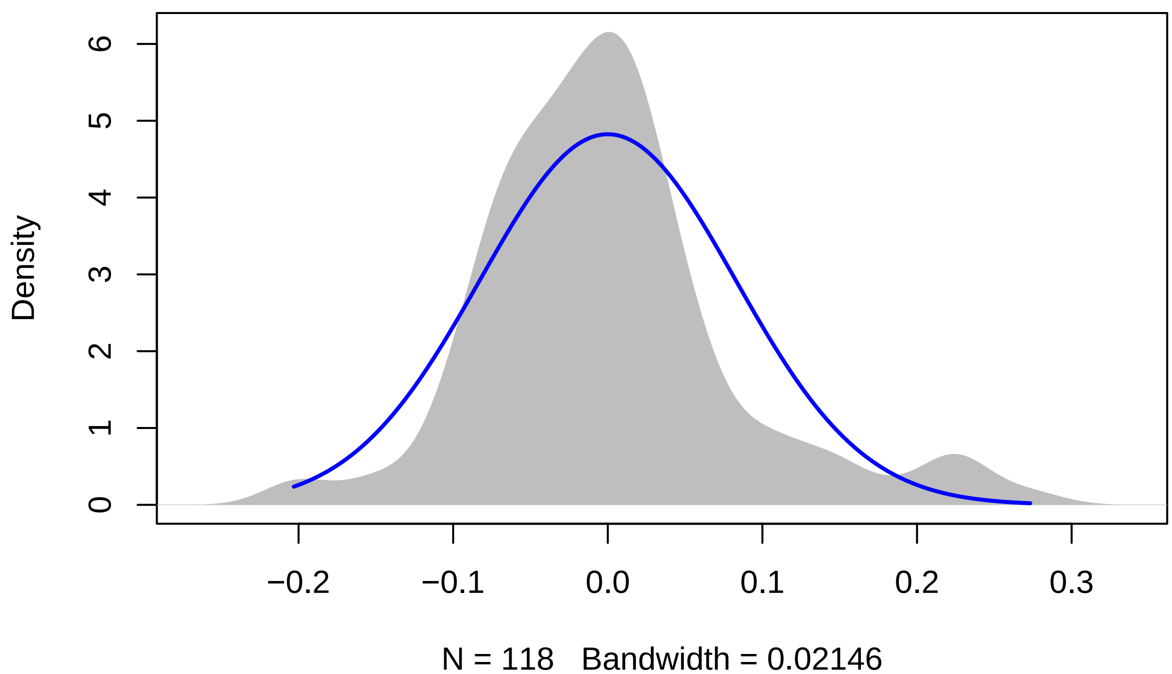
```
plotNormalHistogram(Model2_Paris$residuals)
```



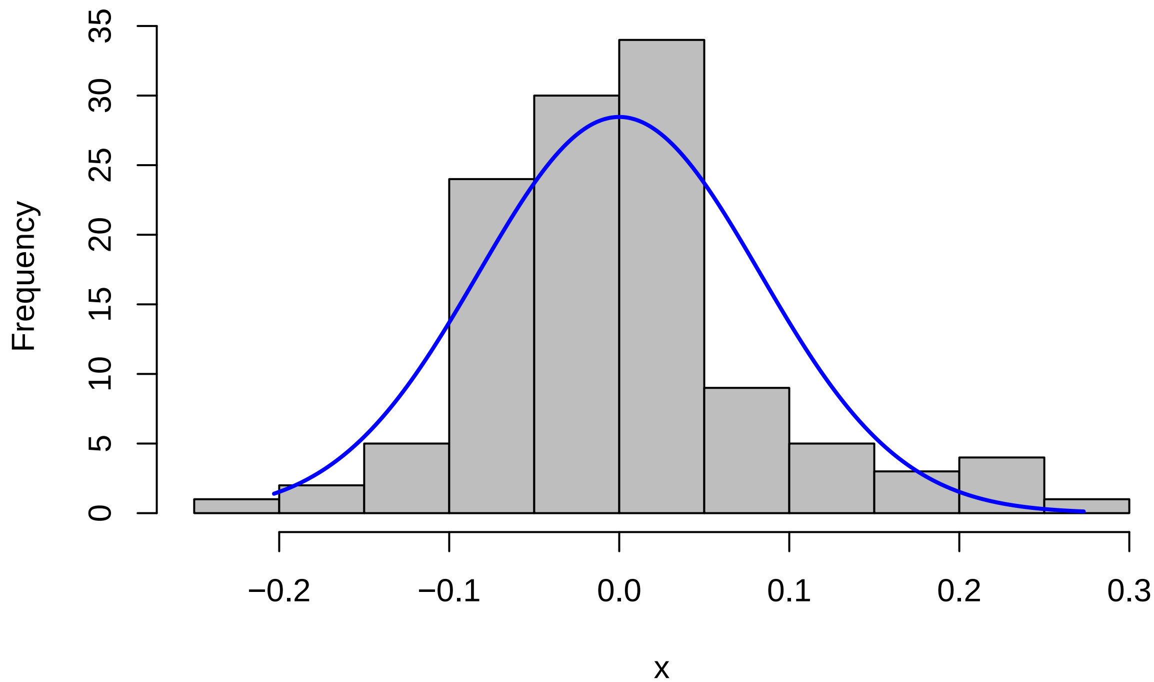

```
jarque.bera.test(Model3_Paris$residuals) # P-value < 1.664e-06 --> non-normal
```

```
##
##  Jarque Bera Test
##
## data:  Model3_Paris$residuals
## X-squared = 26.613, df = 2, p-value = 1.664e-06
```

```
plotNormalDensity(Model3_Paris$residuals)
```



```
plotNormalHistogram(Model3_Paris$residuals)
```




```
# multicollinearity

ols_vif_tol(Model1_Paris) # all good
```

```
##      Variables Tolerance      VIF
## 1       LN_age 0.8401312 1.190290
## 2 LN_OfferSize 0.5685206 1.758951
## 3         Tech 0.9595804 1.042122
## 4         Rank 0.5971266 1.674687
## 5    Technique 0.8258591 1.210860
## 6    Sentiment 0.9594436 1.042271
## 7           HC 0.8259158 1.210777
## 8           VC 0.9251134 1.080949
```

```
ols_vif_tol(Model2_Paris) # all good
```

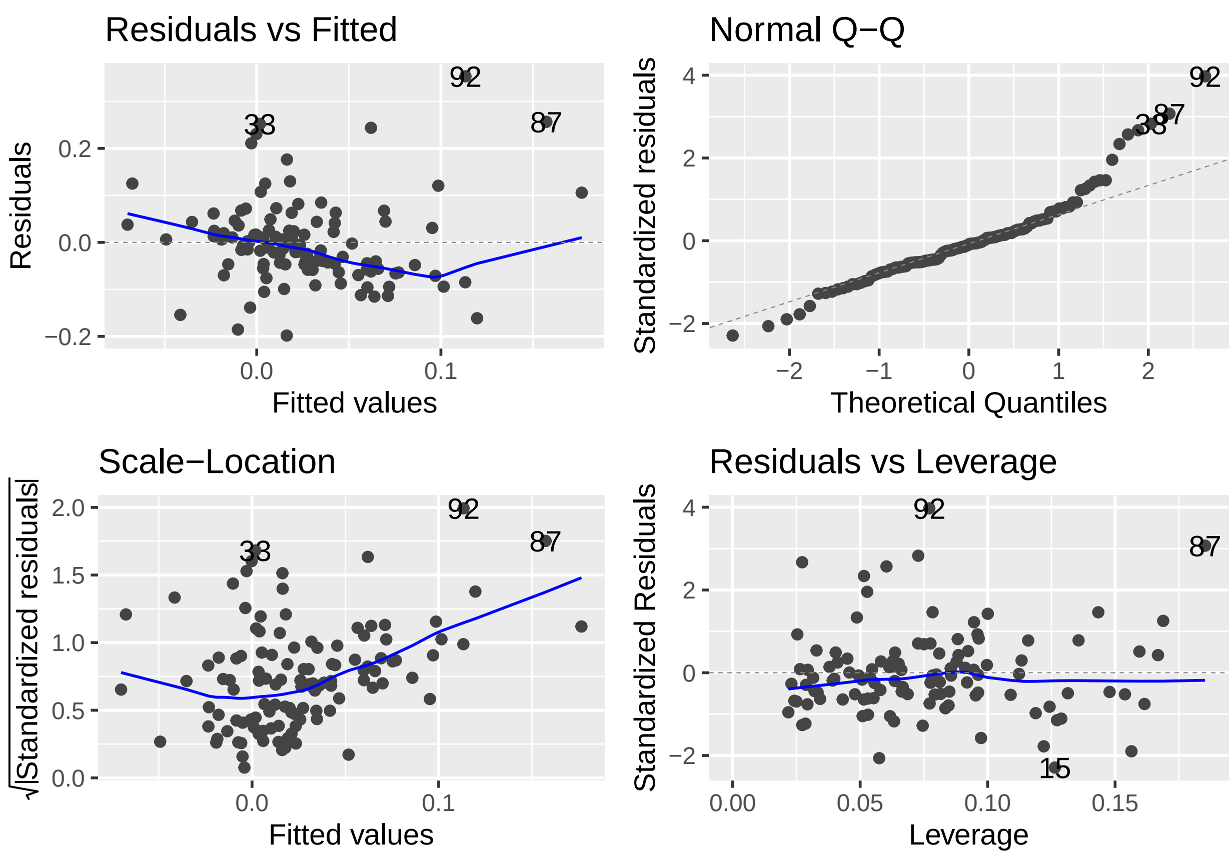
```
##           Variables Tolerance      VIF
## 1            LN_age 0.8261918 1.210373
## 2      LN_OfferSize 0.5558924 1.798909
## 3              Tech 0.9332756 1.071495
## 4              Rank 0.5958242 1.678347
## 5         Technique 0.8162208 1.225159
## 6         Sentiment 0.8478450 1.179461
## 7                HC 0.8001723 1.249731
## 8                VC 0.9080530 1.101257
## 9       Marketreturn 0.9007936 1.110132
## 10 Marketvolatility 0.7918379 1.262885
```

```
ols_vif_tol(Model3_Paris) # all good
```

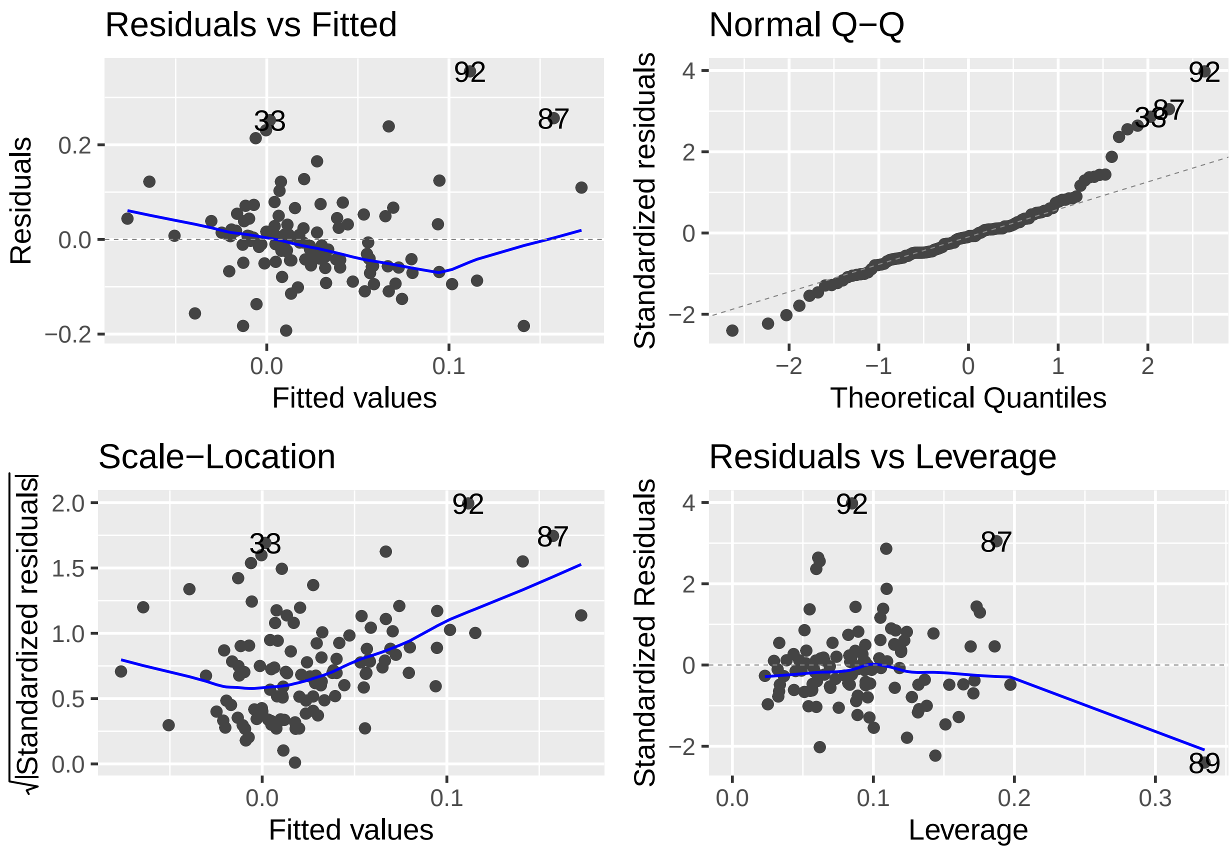
```
##           Variables Tolerance      VIF
## 1            LN_age 0.7764276 1.287950
```

```
## 2       LN_OfferSize 0.5156052 1.939469
## 3               Tech 0.8961922 1.115832
## 4               Rank 0.5380376 1.858606
## 5          Technique 0.7632946 1.310110
## 6          Sentiment 0.3465976 2.885190
## 7                 HC 0.3623296 2.759918
## 8                 VC 0.7799738 1.282094
## 9        Marketreturn 0.7488642 1.335356
## 10 Marketvolatility 0.4217072 2.371314
## 11         Dummy2014 0.3335048 2.998458
## 12         Dummy2015 0.3805231 2.627961
## 13         Dummy2016 0.4843005 2.064834
## 14         Dummy2017 0.4572695 2.186894
## 15         Dummy2018 0.4734628 2.112098
## 16         Dummy2019 0.6832175 1.463663
## 17         Dummy2020 0.4932073 2.027545
```

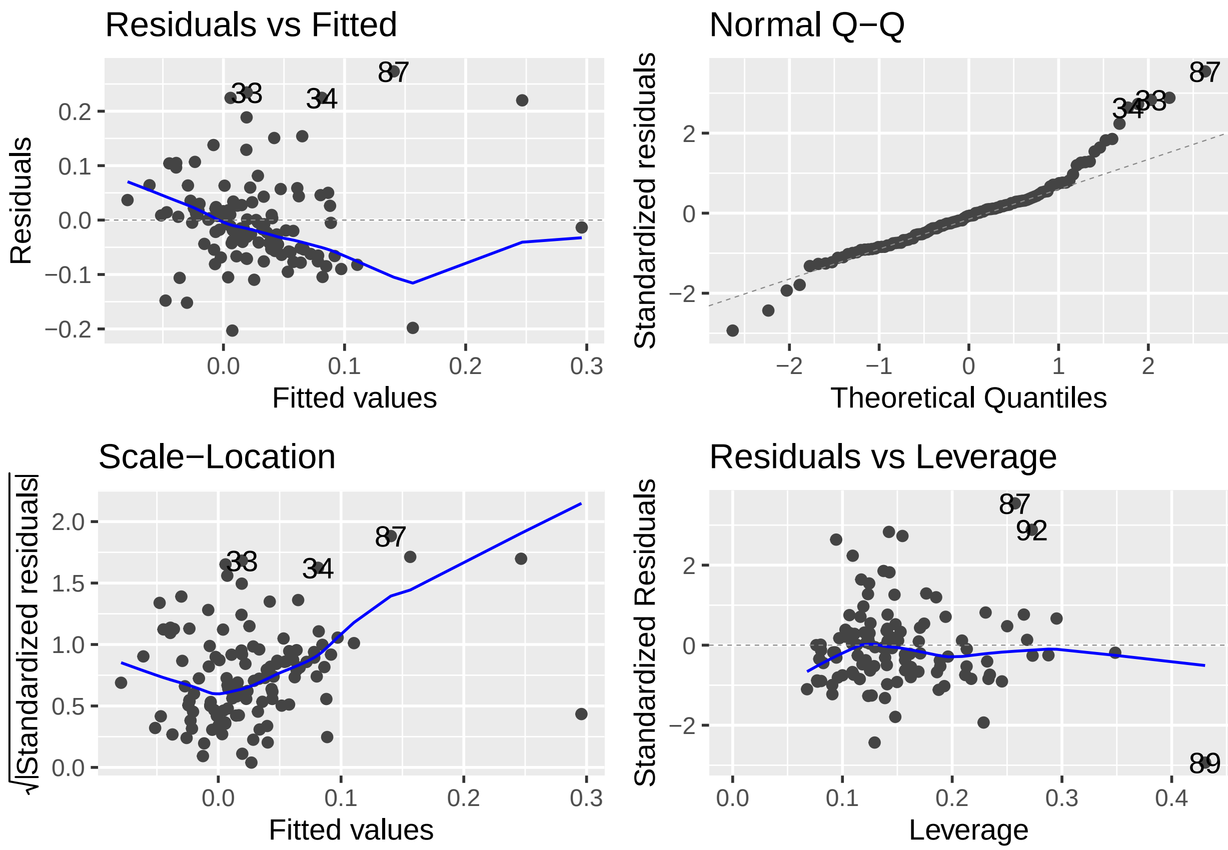
```r
# Regression Model Diagnostics
autoplot(Model1_Paris)
```




```r
autoplot(Model2_Paris)
```




```
autoplot(Model3_Paris)
```




```
# List of models

stargazer(Model1_Paris_robust, Model2_Paris_robust, Model3_Paris_robust, type = "text")
```

```
##
## ==============================================
##                         Dependent variable:
##                    ----------------------------
##
##                       (1)       (2)       (3)
## ----------------------------------------------
## LN_age              -0.014*   -0.013*   -0.008
##                     (0.008)   (0.008)   (0.009)
##
## LN_OfferSize         0.009     0.009    0.011*
##                     (0.007)   (0.007)   (0.006)
##
## Tech                -0.002    -0.001    -0.003
##                     (0.027)   (0.026)   (0.025)
##
## Rank                -0.035    -0.034   -0.047**
##                     (0.022)   (0.021)   (0.022)
##
## Technique            0.041     0.040     0.023
##                     (0.025)   (0.025)   (0.025)
##
## Sentiment          -0.011**  -0.011**  -0.022***
```

```
##                   (0.004)   (0.004)   (0.006)
##
## HC               -0.038**  -0.038**   -0.032
##                   (0.017)   (0.017)   (0.034)
##
## VC                -0.029    -0.030    -0.022
##                   (0.022)   (0.022)   (0.021)
##
## Marketreturn                 0.127    -0.017
##                             (0.268)   (0.284)
##
## Marketvolatility             0.598    -0.499
##                             (2.930)   (3.372)
##
## Dummy2014                            -0.124***
##                                       (0.046)
##
## Dummy2015                             -0.042
##                                       (0.042)
##
## Dummy2016                            -0.099**
##                                       (0.042)
##
## Dummy2017                             -0.014
##                                       (0.041)
##
## Dummy2018                             -0.020
##                                       (0.036)
##
## Dummy2019                             -0.015
##                                       (0.044)
##
## Dummy2020                            -0.130**
##                                       (0.063)
##
## Constant          -0.031    -0.035    -0.077
##                   (0.034)   (0.039)   (0.056)
##
## ===============================================
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
summary(Model1_Paris)
```

```
##
## Call:
## lm(formula = MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique +
##     Sentiment + HC + VC, data = Paris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19826 -0.04790 -0.00811  0.03468  0.35334
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -0.031064   0.041261  -0.753 0.453152
## LN_age       -0.013641   0.011199  -1.218 0.225835
## LN_OfferSize  0.008853   0.007566   1.170 0.244480
## Tech         -0.001967   0.024195  -0.081 0.935343
## Rank         -0.034548   0.023296  -1.483 0.140957
## Technique     0.040758   0.024993   1.631 0.105822
## Sentiment    -0.011053   0.002923  -3.782 0.000255 ***
## HC           -0.038126   0.021534  -1.771 0.079436 .
## VC           -0.028619   0.020578  -1.391 0.167136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09257 on 109 degrees of freedom
## Multiple R-squared:  0.1712, Adjusted R-squared:  0.1104
## F-statistic: 2.814 on 8 and 109 DF,  p-value: 0.007099
```

```
summary(Model2_Paris)
```

```
##
## Call:
## lm(formula = MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique +
##     Sentiment + HC + VC + Marketreturn + Marketvolatility, data = Paris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19260 -0.04878 -0.00784  0.03204  0.35512
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.0350516  0.0469873  -0.746 0.457314
## LN_age           -0.0128760  0.0113820  -1.131 0.260472
## LN_OfferSize      0.0085533  0.0077116   1.109 0.269848
## Tech             -0.0005853  0.0247268  -0.024 0.981160
## Rank             -0.0339847  0.0235057  -1.446 0.151154
## Technique         0.0395980  0.0253383   1.563 0.121059
## Sentiment        -0.0106398  0.0031339  -3.395 0.000963 ***
## HC               -0.0381973  0.0220504  -1.732 0.086106 .
## VC               -0.0301027  0.0209344  -1.438 0.153365
## Marketreturn      0.1268946  0.2398769   0.529 0.597902
## Marketvolatility  0.5979773  2.5520541   0.234 0.815190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0933 on 107 degrees of freedom
## Multiple R-squared:  0.1735, Adjusted R-squared:  0.09627
## F-statistic: 2.246 on 10 and 107 DF,  p-value: 0.02018
```

```
summary(Model3_Paris)
```

```
##
## Call:
## lm(formula = MAR ~ LN_age + LN_OfferSize + Tech + Rank + Technique +
##     Sentiment + HC + VC + Marketreturn + Marketvolatility + Dummy2014 +
##     Dummy2015 + Dummy2016 + Dummy2017 + Dummy2018 + Dummy2019 +
##     Dummy2020, data = Paris)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.203065 -0.053689 -0.005555  0.029276  0.273210
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.077494   0.055642  -1.393  0.16679
## LN_age           -0.008271   0.011257  -0.735  0.46419
## LN_OfferSize      0.010519   0.007677   1.370  0.17368
## Tech             -0.002752   0.024192  -0.114  0.90967
## Rank             -0.047241   0.023715  -1.992  0.04909 *
## Technique         0.022995   0.025121   0.915  0.36220
## Sentiment        -0.022159   0.004699  -4.716 7.83e-06 ***
## HC               -0.032031   0.031416  -1.020  0.31039
## VC               -0.021723   0.021656  -1.003  0.31822
## Marketreturn     -0.017121   0.252231  -0.068  0.94602
## Marketvolatility -0.498637   3.352748  -0.149  0.88207
## Dummy2014        -0.124314   0.040604  -3.062  0.00283 **
## Dummy2015        -0.041867   0.034274  -1.222  0.22476
## Dummy2016        -0.099015   0.039148  -2.529  0.01299 *
## Dummy2017        -0.014416   0.040288  -0.358  0.72124
## Dummy2018        -0.019553   0.035925  -0.544  0.58747
## Dummy2019        -0.014809   0.049454  -0.299  0.76521
## Dummy2020        -0.130086   0.049634  -2.621  0.01014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08945 on 100 degrees of freedom
## Multiple R-squared:   0.29,  Adjusted R-squared:  0.1693
## F-statistic: 2.403 on 17 and 100 DF,  p-value: 0.003685
```