Linear Regression using R

Simeon Hristov

Employee Performance Prediction

Context

Given data about employee answers on different selection tools, predict their performance.

Description of the features

The features and their meaning are provided below.

- `ID`: Unique identifier
- `SJT`: Situational judgement test (a type of selection tool that many organizations use)
- `EmotionalIntelligence`: inventory or test
- `Proactivity`: personality assessment test
- `Performance`: job evaluation ratings
- `Turnover`: whether an employee has left the company

Higher scores indicate high level and is assumed to have a positive impact.

Prerequisites

The following libraries need to be installed in order to run the script. You can install them in case you don't have them.

```
install.packages('readr')
install.packages('lessR')
```

When all is set up, include them in the project.

```
library(readr)
library(lessR)
```

Getting the data

Let's first create a variable that we'll treat as the path to our data.

```
DATA_PATH <- './SelectionExercise.csv'
```

Next, we load the data in a data frame by using the `read_csv()` function from the `readr` library.

```
df <- read_csv(DATA_PATH)
```

```
## Rows: 300 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): ID, SJT, EmotionalIntelligence, Proactivity, Performance, Turnover
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

We have a total of 300 samples with no missing values. Here are the first 10 observations.

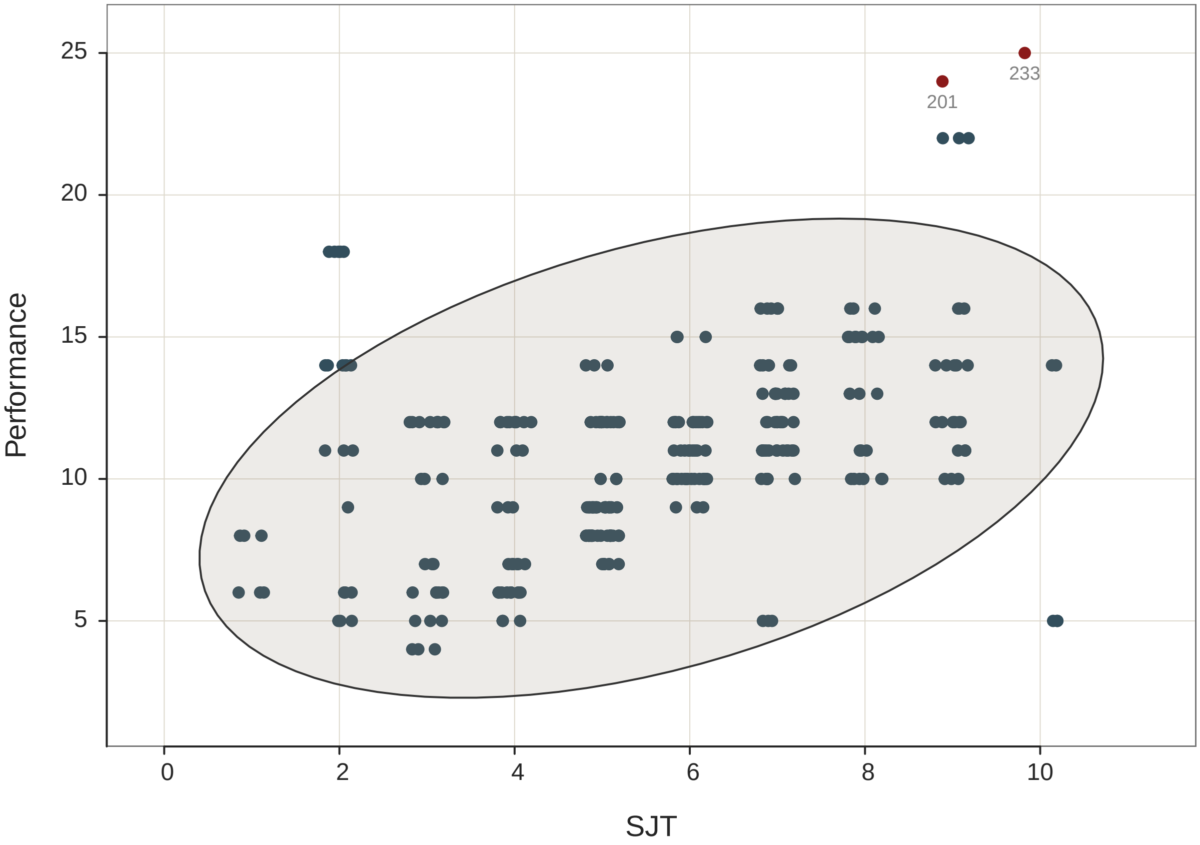
```
df
```

```
## # A tibble: 300 x 6
##       ID   SJT EmotionalIntelligence Proactivity Performance Turnover
##    <dbl> <dbl>                 <dbl>       <dbl>       <dbl>    <dbl>
##  1     1     9                     8           2          22        1
##  2     2     8                     6           3          11        1
##  3     3     7                     6           4           5        0
##  4     4     6                     5           5          11        1
##  5     5     6                     5           6          12        0
##  6     6     5                     5           7          12        1
##  7     7     5                     4           7          12        0
##  8     8     4                     2           8          12        1
##  9     9     3                     2           9          12        1
## 10    10     8                     7           2          10        1
## # ... with 290 more rows
```

Simple Linear Regression

Let's check whether we meet the assumptions of a linear regression model.

```
## [Ellipse with Murdoch and Chow's function ellipse from their ellipse package]
```



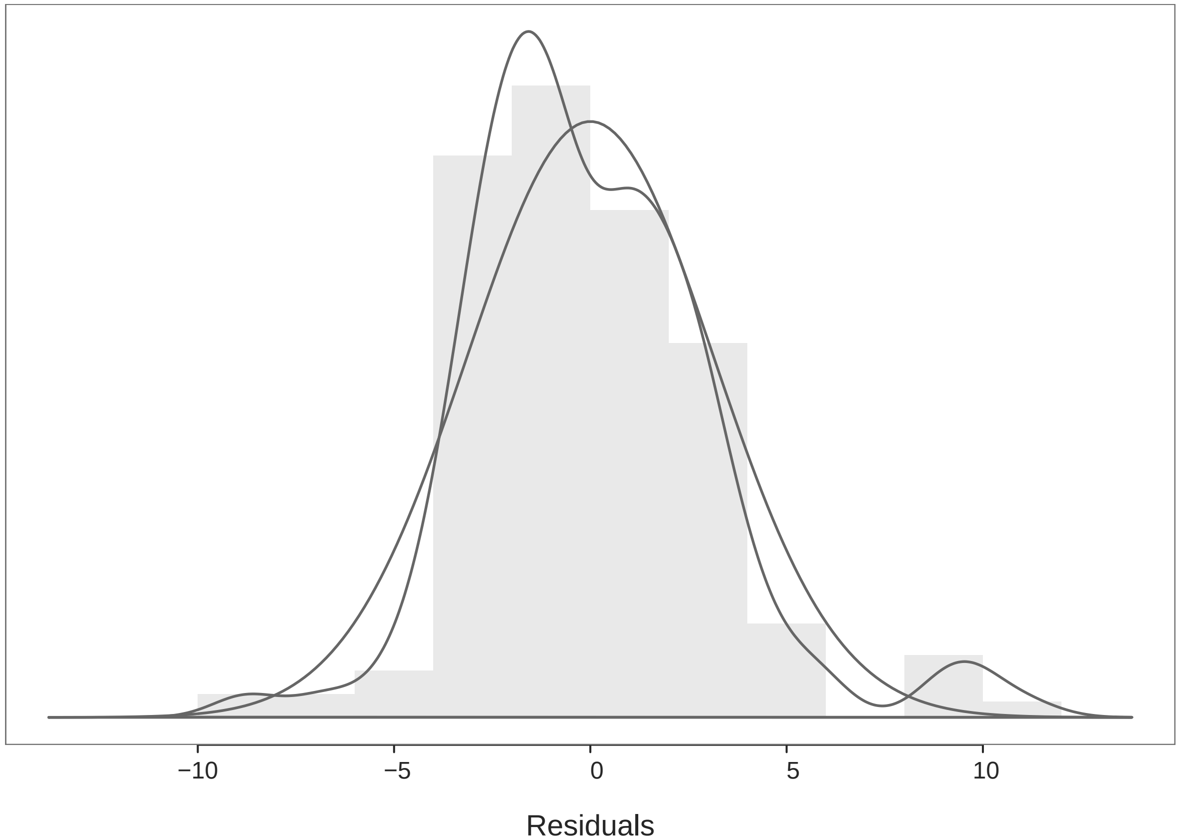
```
## >>> Suggestions
## Plot(SJT, Performance, enhance=TRUE)  # many options
## Plot(SJT, Performance, color="red")  # exterior edge color of points
## Plot(SJT, Performance, fit="lm", fit_se=c(.90,.99))  # fit line, stnd errors
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 300
## Sample Correlation of SJT and Performance: r = 0.417
##
## Hypothesis Test of 0 Correlation:  t = 7.915,  df = 298,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.319 to 0.506
```

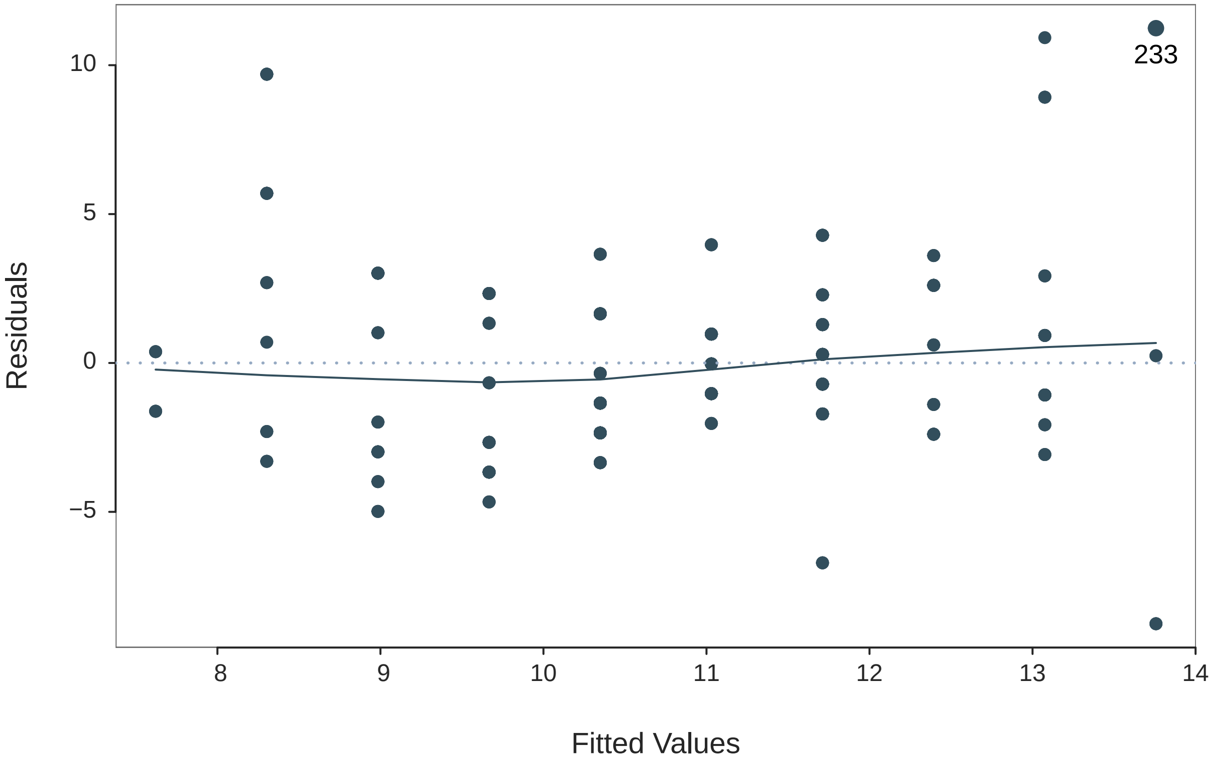
```
## >>> Outlier analysis with Mahalanobis Distance
##
##    MD  ID
## ----- -----
## 17.31 233
## 14.82 201
##
## 12.43  20
## 12.43  69
## 12.43 120
##  ... ...
##
## Some Parameter values (can be manually set)
## -------------------------------------------------------
## fill: #324E5C   filled color of the points
## color: #324E5C  edge color of the points
## size: 0.80  size of plotted points
## jitter_y: 0.00  random vertical movement of points
## jitter_x: 1.00  random horizontal movement of points
```

We meet the assumption of bivariate normal distribution. There is a linear relationship between SJT and Performance. The correlation coefficient is 0.417 which signals good relationship. The p-value is 0 which means that this is a statistically significant association. Nevertheless, there are some outliers (for example 201 and 233) and we will see how the model will perform if we remove them.

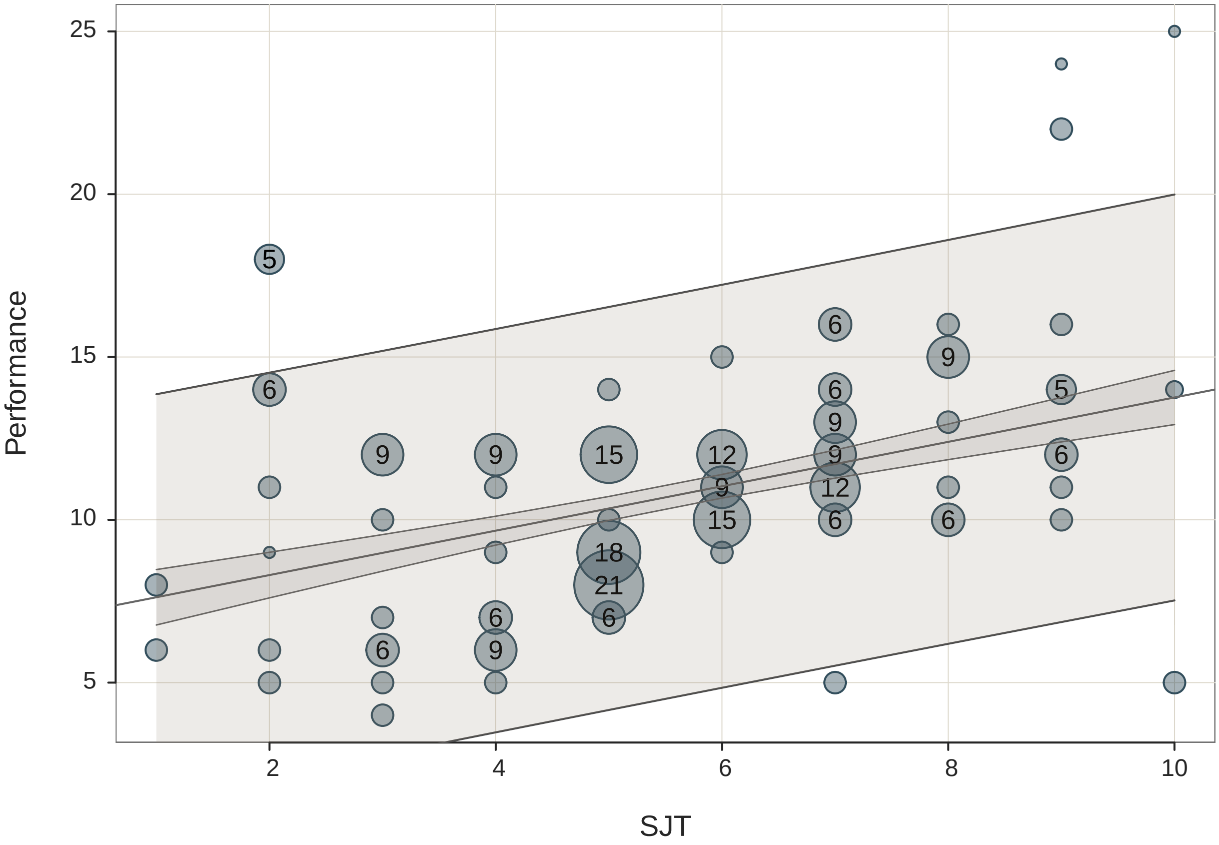
Simple Regression from lessR

We perform linear regression by using the Regression function that comes with the lessR package. Conveniently, it also useful plots and analysis results.





Point with largest Cook's Distance of 0.12 is labeled



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Performance ~ SJT, data=df, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable: SJT
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     6.939      0.512   13.552    0.000       5.932       7.947
##         SJT     0.682      0.086    7.915    0.000       0.512       0.851
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  3.139 for 298 degrees of freedom
## 95% range of residual variation:  12.354 = 2 * (1.968 * 3.139)
##
```

```
## R-squared:  0.174    Adjusted R-squared:  0.171    PRESS R-squared:  0.157
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 62.654     df: 1 and 298     p-value:  0.000
##
## -- Analysis of Variance
##
##               df    Sum Sq   Mean Sq   F-value   p-value
## Model          1   617.264   617.264    62.654     0.000
## Residuals    298  2935.866     9.852
## Performance  299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##               Performance  SJT
##   Performance        1.00 0.42
##           SJT        0.42 1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 300 rows of data, or do res_rows="all"]
## ----------------------------------------------------------
##          SJT Performance fitted  resid rstdnt dffits cooks
##   233 10.000      25.000 13.757 11.243  3.691  0.502 0.121
##   201  9.000      24.000 13.075 10.925  3.570  0.398 0.076
##    70 10.000       5.000 13.757 -8.757 -2.849 -0.388 0.073
##   170 10.000       5.000 13.757 -8.757 -2.849 -0.388 0.073
##   270 10.000       5.000 13.757 -8.757 -2.849 -0.388 0.073
##    20  2.000      18.000  8.303  9.697  3.156  0.360 0.063
##    69  2.000      18.000  8.303  9.697  3.156  0.360 0.063
##   120  2.000      18.000  8.303  9.697  3.156  0.360 0.063
##   169  2.000      18.000  8.303  9.697  3.156  0.360 0.063
##   269  2.000      18.000  8.303  9.697  3.156  0.360 0.063
##     1  9.000      22.000 13.075  8.925  2.896  0.322 0.051
##   101  9.000      22.000 13.075  8.925  2.896  0.322 0.051
##   283  9.000      22.000 13.075  8.925  2.896  0.322 0.051
##    82  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##    97  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##   182  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##   197  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##   282  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##   297  2.000      14.000  8.303  5.697  1.834  0.209 0.022
##     3  7.000       5.000 11.712 -6.712 -2.157 -0.151 0.011
##
##
##   PREDICTION ERROR
##
```

```
## Data, Predicted, Standard Error of Forecast,
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
##
##          SJT Performance   pred    sf pi.lwr pi.upr  width
##    31  1.000       6.000  7.621 3.168  1.386 13.856 12.471
##    41  1.000       8.000  7.621 3.168  1.386 13.856 12.471
## ...
##   299  5.000       8.000 10.348 3.144  4.160 16.536 12.376
##     4  6.000      11.000 11.030 3.144  4.842 17.218 12.375
##     5  6.000      12.000 11.030 3.144  4.842 17.218 12.375
## ...
##   170 10.000       5.000 13.757 3.167  7.524 19.990 12.466
##   233 10.000      25.000 13.757 3.167  7.524 19.990 12.466
##   270 10.000       5.000 13.757 3.167  7.524 19.990 12.466
##
## ---------------------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Reg Line, Confidence & Prediction Intervals
## ---------------------------------------------------
```

The assumption of normally distributed residuals is met based on `Plot 1: Distribution of Residuals`. The assumptions of average residual error being (almost) 0 and homoscedasticity of variances are both met based on `Plot 2: Residuals vs Fitted Values`.

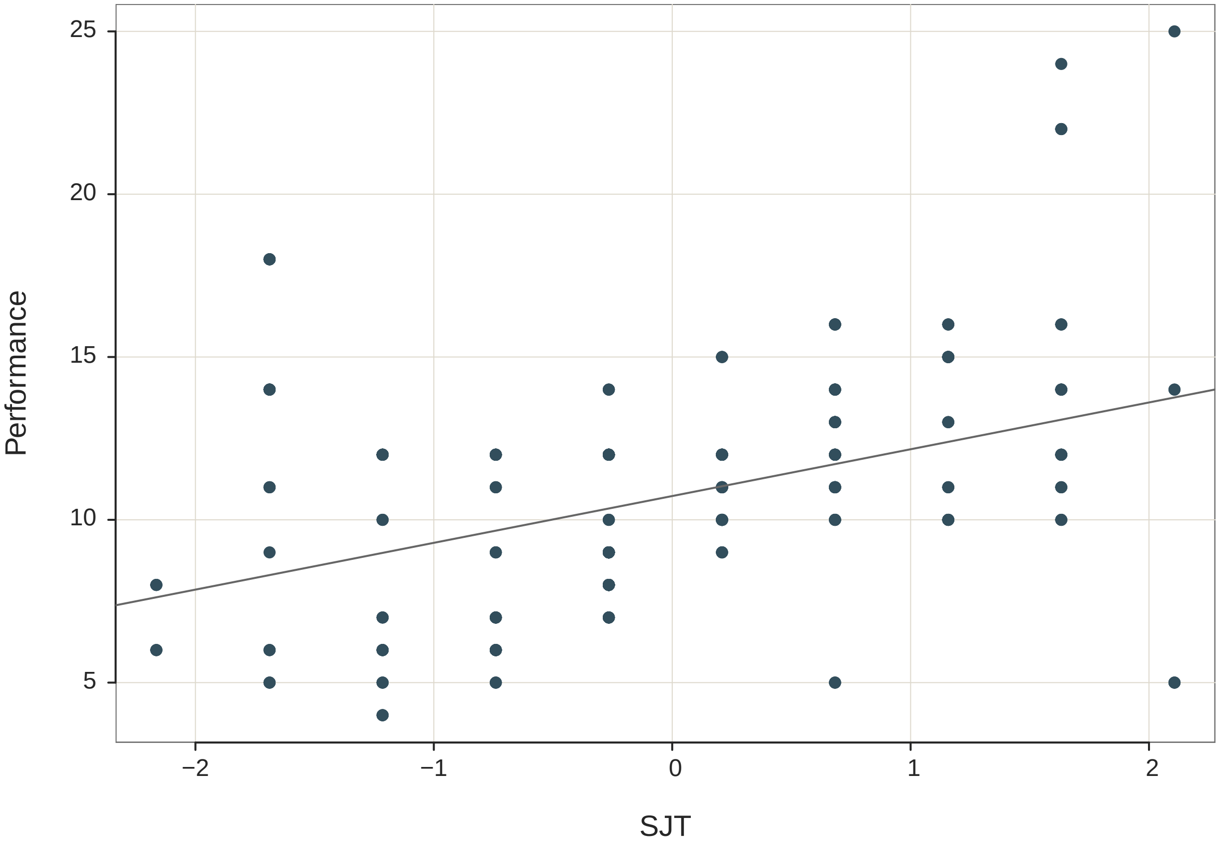
The `BASIC ANALYSIS` part of the output shows that for every `1` unit increase in `SJT`, `Performance` increases by `0.682` units. We can also construct the equation of the line: `Performance = 6.939 + 0.682 * SJT`.

The `adjusted R-squared` value is `0.171`, i.e. `SJT` explains about `17%` of the variability in `Performance`. The `F-statistic` shows that a model using `SJT` will outperform a null model, i.e. `SJT` is significant.

We can again see the potential outliers by looking at the part. The `Cook's Distance` of sample with id `233` is noticeably higher than the other samples: `0.121`.

If we standardized our data, the results will not differ by much.

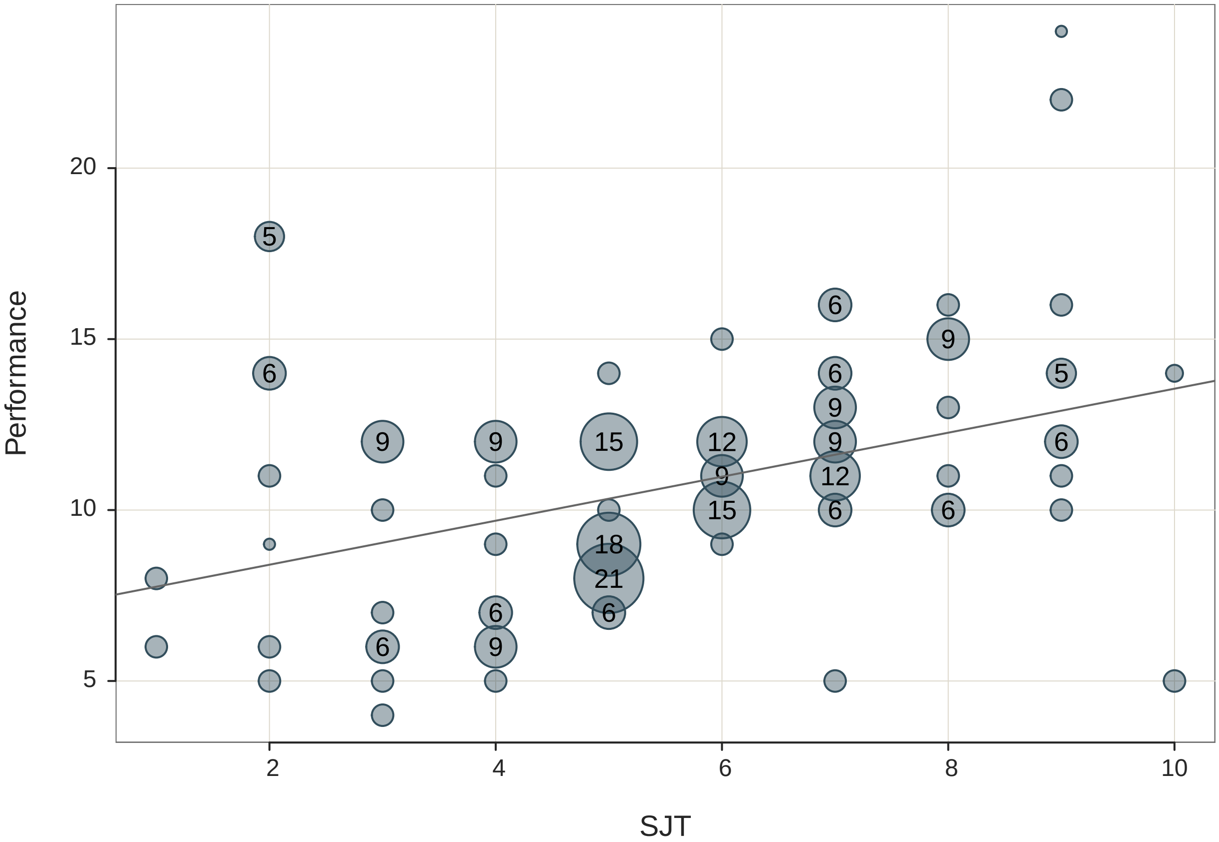
```
##
## Rescaled Data, First Six Rows
##     Performance    SJT
## 31            6 -2.164
## 41            8 -2.164
## 131           6 -2.164
## 141           8 -2.164
## 231           6 -2.164
## 241           8 -2.164
```



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## reg(Performance ~ SJT, data=df, new_scale="z", Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable: SJT
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
## Data are Standardized
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)    10.730      0.181   59.212    0.000      10.373      11.087
##         SJT     1.437      0.182    7.916    0.000       1.080       1.794
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  3.139 for 298 degrees of freedom
## 95% range of residual variation:  12.354 = 2 * (1.968 * 3.139)
##
## R-squared:  0.174    Adjusted R-squared:  0.171    PRESS R-squared:  0.157
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 62.661     df: 1 and 298     p-value:  0.000
##
## -- Analysis of Variance
##
##                df    Sum Sq   Mean Sq   F-value   p-value
## Model           1   617.320   617.320    62.661     0.000
## Residuals     298  2935.810     9.852
## Performance   299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##
##   RESIDUALS AND INFLUENCE
##
##
##   PREDICTION ERROR
```

Inspecting outliers

A model without the observation with id 233 does not perform that well. Here’s what we would get.



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## reg(Performance ~ SJT, data=df, rows=(ID != 233), Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable: SJT
##
## Number of cases (rows) of data:  299
## Number of cases retained for analysis:  299
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     7.114      0.504   14.122    0.000       6.123       8.105
##         SJT     0.644      0.085    7.570    0.000       0.476       0.811
##
## Standard deviation of Performance: 3.352
##
## Standard deviation of residuals:  3.074 for 297 degrees of freedom
## 95% range of residual variation:  12.101 = 2 * (1.968 * 3.074)
##
```

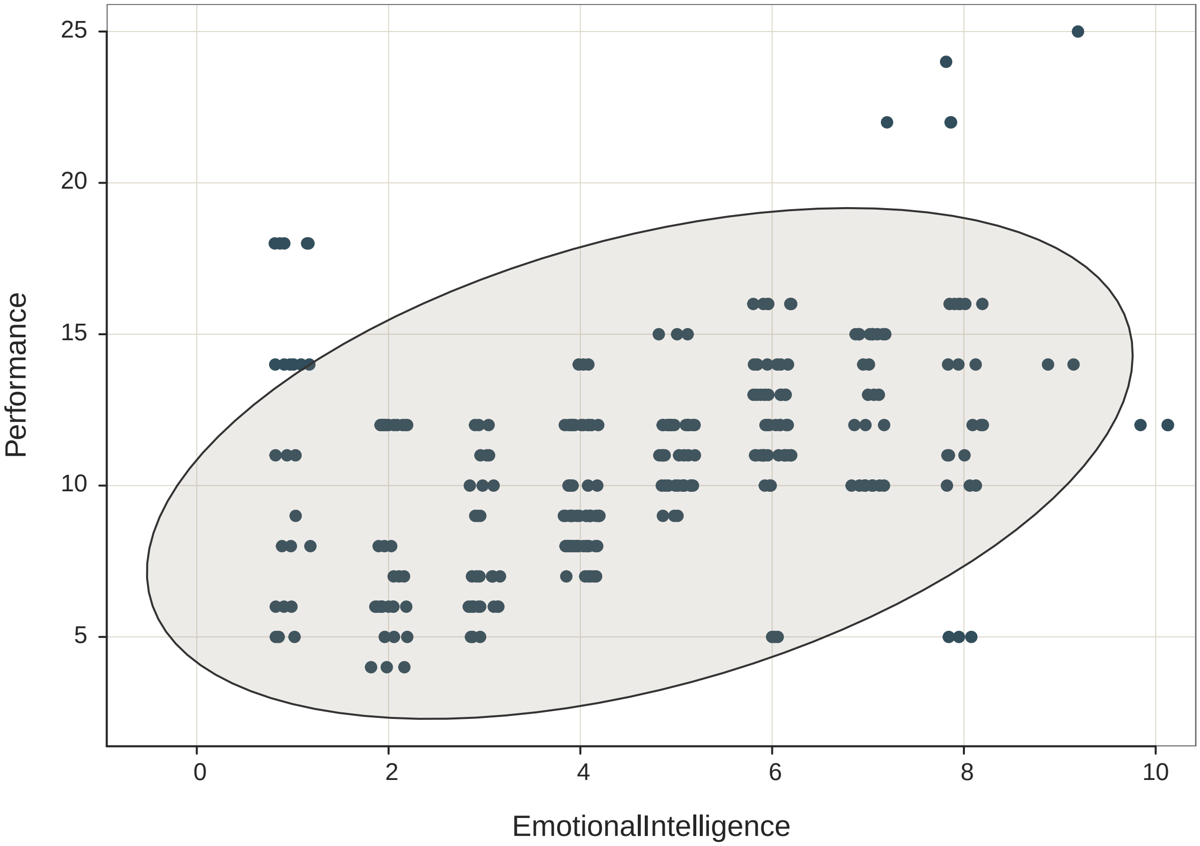
```
## R-squared:  0.162    Adjusted R-squared:  0.159    PRESS R-squared:  0.145
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 57.312     df: 1 and 297     p-value:  0.000
##
## -- Analysis of Variance
##
##              df    Sum Sq   Mean Sq   F-value   p-value
## Model         1   541.692   541.692    57.312     0.000
## Residuals   297  2807.124     9.452
## Performance  298  3348.816    11.238
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##
##   RESIDUALS AND INFLUENCE
##
##
##   PREDICTION ERROR
```

The new `Adjusted R-squared` value is `0.159` which is a slight decrease from `0.171` and therefore we have no reason to remove that observation.

Multiple Linear Regression

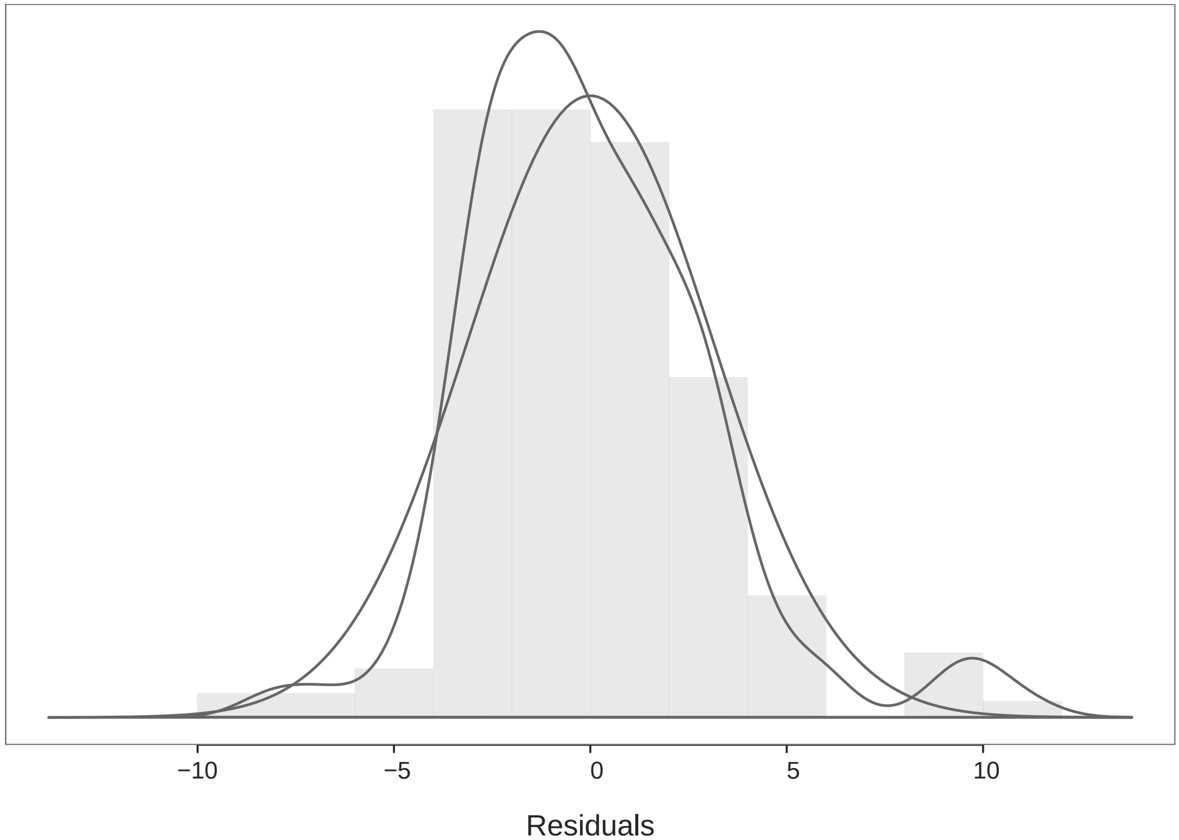
Firstly, we'll check how well the other features predict `Performance`. We start with `EmotionalIntelligence`.

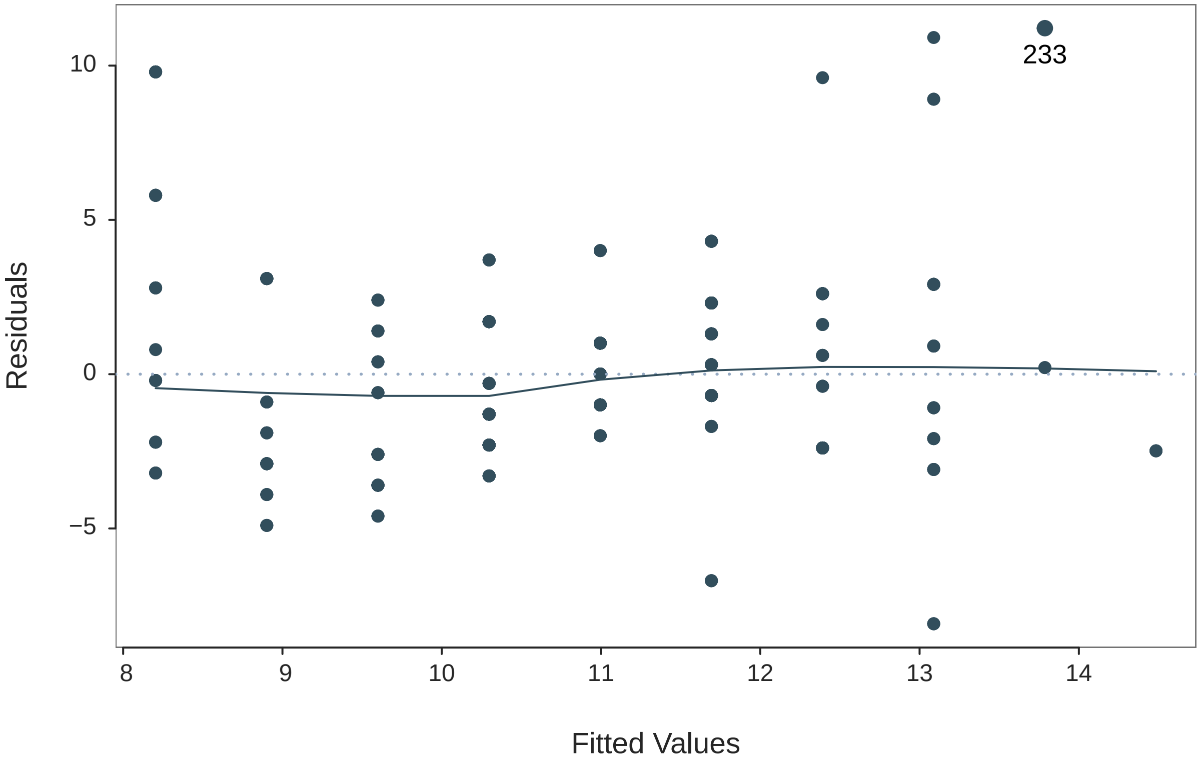
```
## [Ellipse with Murdoch and Chow's function ellipse from their ellipse package]
```



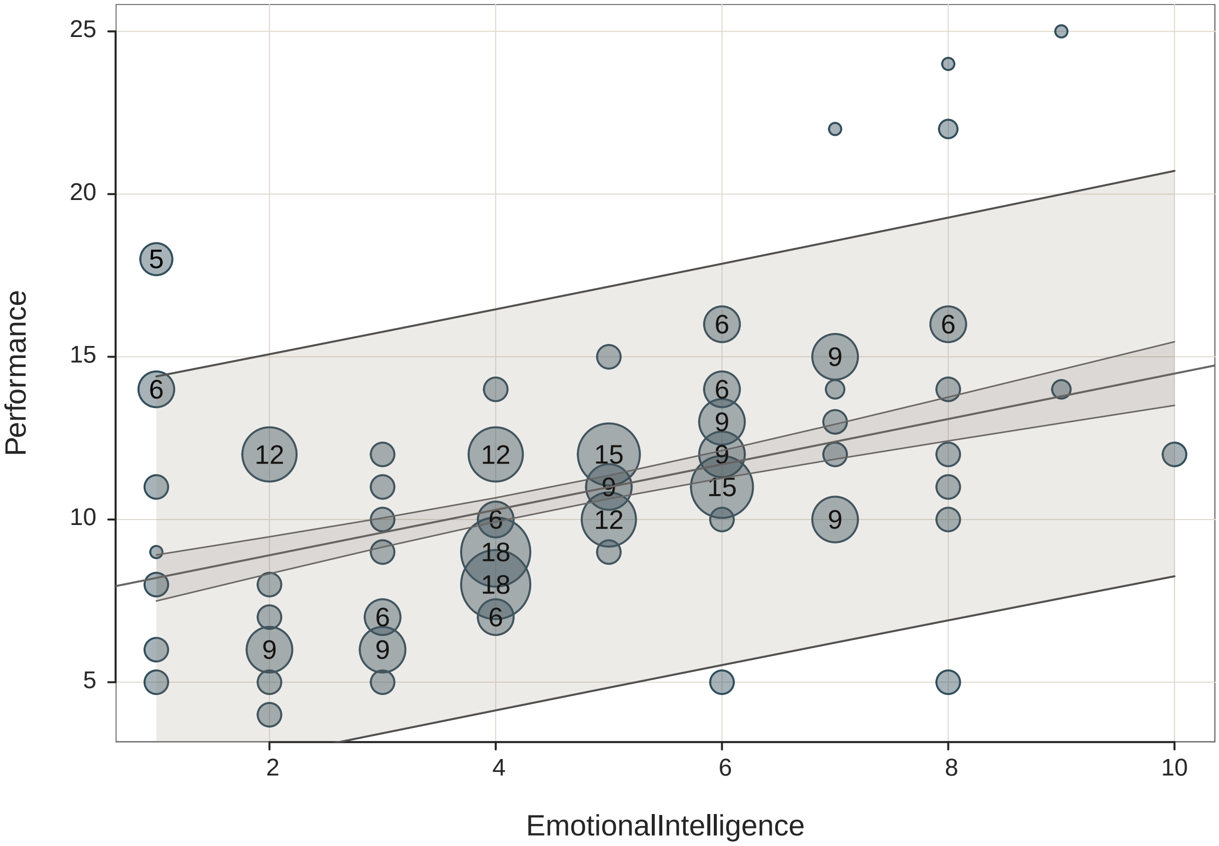
```
## >>> Suggestions
## Plot(EmotionalIntelligence, Performance, enhance=TRUE)  # many options
## Plot(EmotionalIntelligence, Performance, color="red")  # exterior edge color of points
## Plot(EmotionalIntelligence, Performance, fit="lm", fit_se=c(.90,.99))  # fit line, stnd errors
## Plot(EmotionalIntelligence, Performance, out_cut=.10)  # label top 10% from center as outliers
## 
## >>> Pearson's product-moment correlation
## 
## Number of paired values with neither missing, n = 300
## Sample Correlation of EmotionalIntelligence and Performance: r = 0.425
## 
## Hypothesis Test of 0 Correlation:  t = 8.108,  df = 298,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.328 to 0.514
```

```
## 
## Some Parameter values (can be manually set) 
## -------------------------------------------------------
## fill: #324E5C   filled color of the points
## color: #324E5C  edge color of the points
## size: 0.80  size of plotted points
## jitter_y: 0.00  random vertical movement of points
## jitter_x: 1.00  random horizontal movement of points
```





Point with largest Cook's Distance of 0.12 is labeled



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Performance ~ EmotionalIntelligence, data=df, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable: EmotionalIntelligence
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
##
##   BASIC ANALYSIS
##
##                         Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
##           (Intercept)     7.506      0.437   17.190    0.000       6.647       8.365
## EmotionalIntelligence     0.698      0.086    8.108    0.000       0.528       0.867
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  3.125 for 298 degrees of freedom
## 95% range of residual variation:  12.301 = 2 * (1.968 * 3.125)
##
```

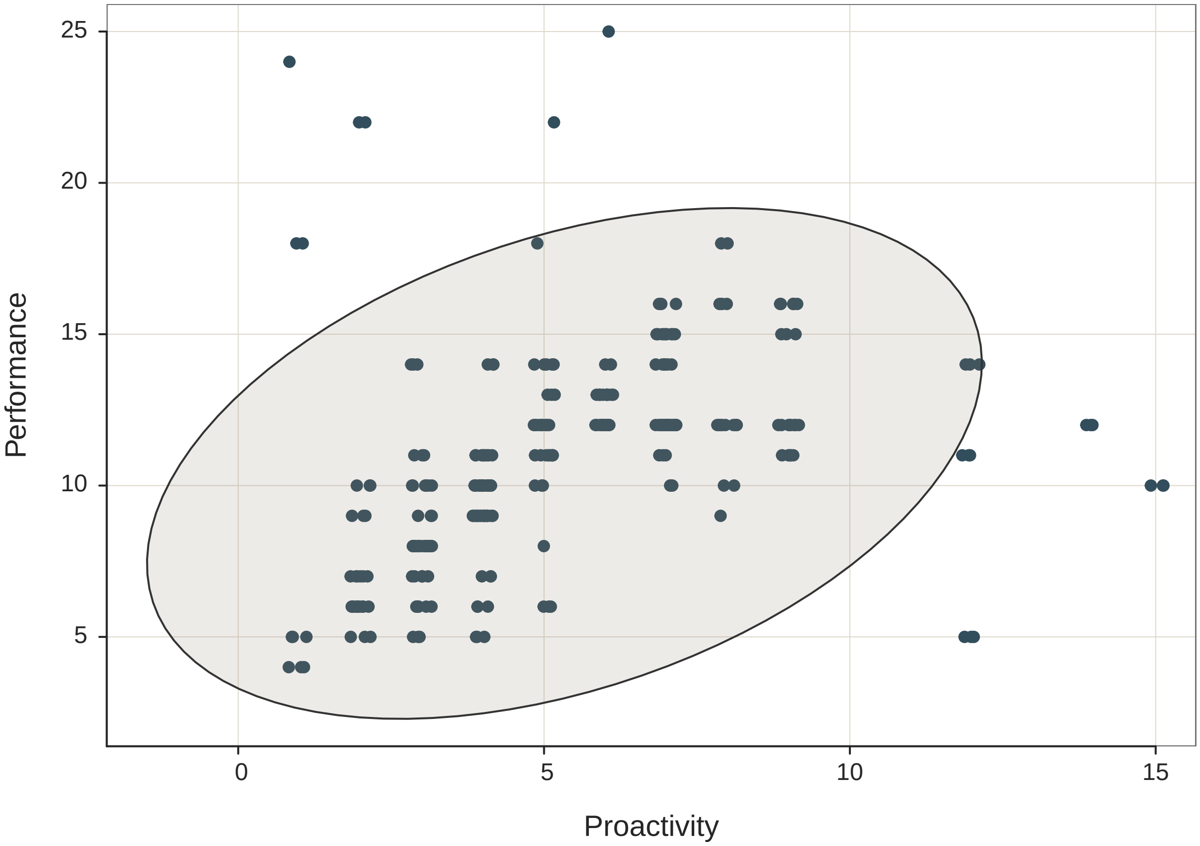
```
## R-squared:  0.181    Adjusted R-squared:  0.178    PRESS R-squared:  0.165
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 65.739     df: 1 and 298     p-value:  0.000
##
## -- Analysis of Variance
##
##                         df    Sum Sq   Mean Sq   F-value   p-value
## Model                    1   642.161   642.161    65.739     0.000
## Residuals              298  2910.969     9.768
## Performance            299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##                         Performance EmotionalIntelligence
##             Performance        1.00                  0.43
##   EmotionalIntelligence        0.43                  1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 300 rows of data, or do res_rows="all"]
## -------------------------------------------------------------------------
##       EmotionalIntelligence Performance fitted  resid rstdnt dffits cooks
##   233                 9.000      25.000 13.787 11.213  3.696  0.499 0.119
##   201                 8.000      24.000 13.089 10.911  3.581  0.395 0.075
##    20                 1.000      18.000  8.204  9.796  3.204  0.372 0.067
##    69                 1.000      18.000  8.204  9.796  3.204  0.372 0.067
##   120                 1.000      18.000  8.204  9.796  3.204  0.372 0.067
##   169                 1.000      18.000  8.204  9.796  3.204  0.372 0.067
##   269                 1.000      18.000  8.204  9.796  3.204  0.372 0.067
##     1                 8.000      22.000 13.089  8.911  2.904  0.320 0.050
##   101                 8.000      22.000 13.089  8.911  2.904  0.320 0.050
##    70                 8.000       5.000 13.089 -8.089 -2.629 -0.290 0.041
##   170                 8.000       5.000 13.089 -8.089 -2.629 -0.290 0.041
##   270                 8.000       5.000 13.089 -8.089 -2.629 -0.290 0.041
##   283                 7.000      22.000 12.391  9.609  3.132  0.275 0.037
##    82                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##    97                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##   182                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##   197                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##   282                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##   297                 1.000      14.000  8.204  5.796  1.875  0.217 0.023
##     3                 6.000       5.000 11.693 -6.693 -2.160 -0.150 0.011
##
##
##   PREDICTION ERROR
##
```

```
## Data, Predicted, Standard Error of Forecast,
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
## 
##       EmotionalIntelligence Performance   pred    sf pi.lwr pi.upr  width
##    20                 1.000      18.000  8.204 3.146  2.012 14.395 12.383
##    40                 1.000       5.000  8.204 3.146  2.012 14.395 12.383
##    55                 1.000      11.000  8.204 3.146  2.012 14.395 12.383
## ...
##   299                 4.000       8.000 10.297 3.131  4.135 16.459 12.324
##     4                 5.000      11.000 10.995 3.131  4.834 17.156 12.323
##     5                 5.000      12.000 10.995 3.131  4.834 17.156 12.323
## ...
##   233                 9.000      25.000 13.787 3.153  7.581 19.992 12.411
##    84                10.000      12.000 14.484 3.165  8.256 20.712 12.456
##   184                10.000      12.000 14.484 3.165  8.256 20.712 12.456
## 
## ---------------------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Reg Line, Confidence & Prediction Intervals
## ---------------------------------------------------
```

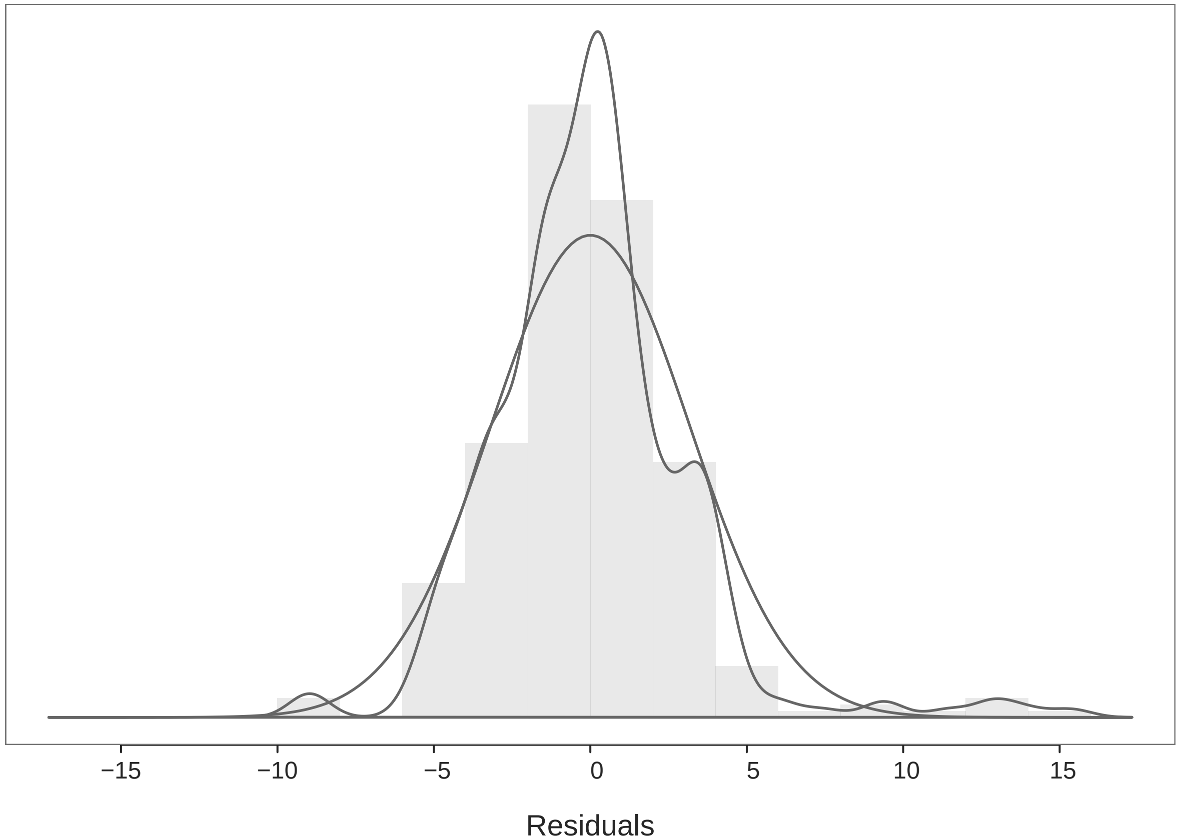
The `EmotionalIntelligence` feature is statistically significantly associated in a positive direction with `Performance`.

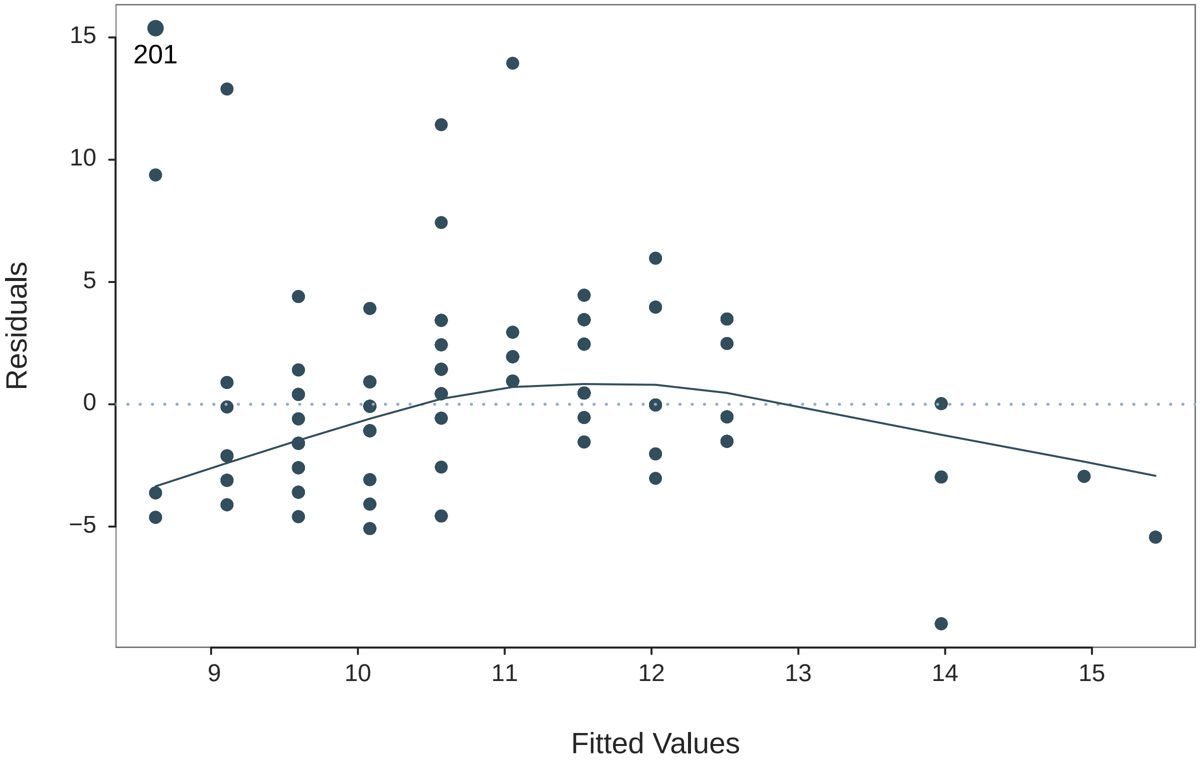
The same can be said for `Proactivity`.

```
## [Ellipse with Murdoch and Chow's function ellipse from their ellipse package]
```

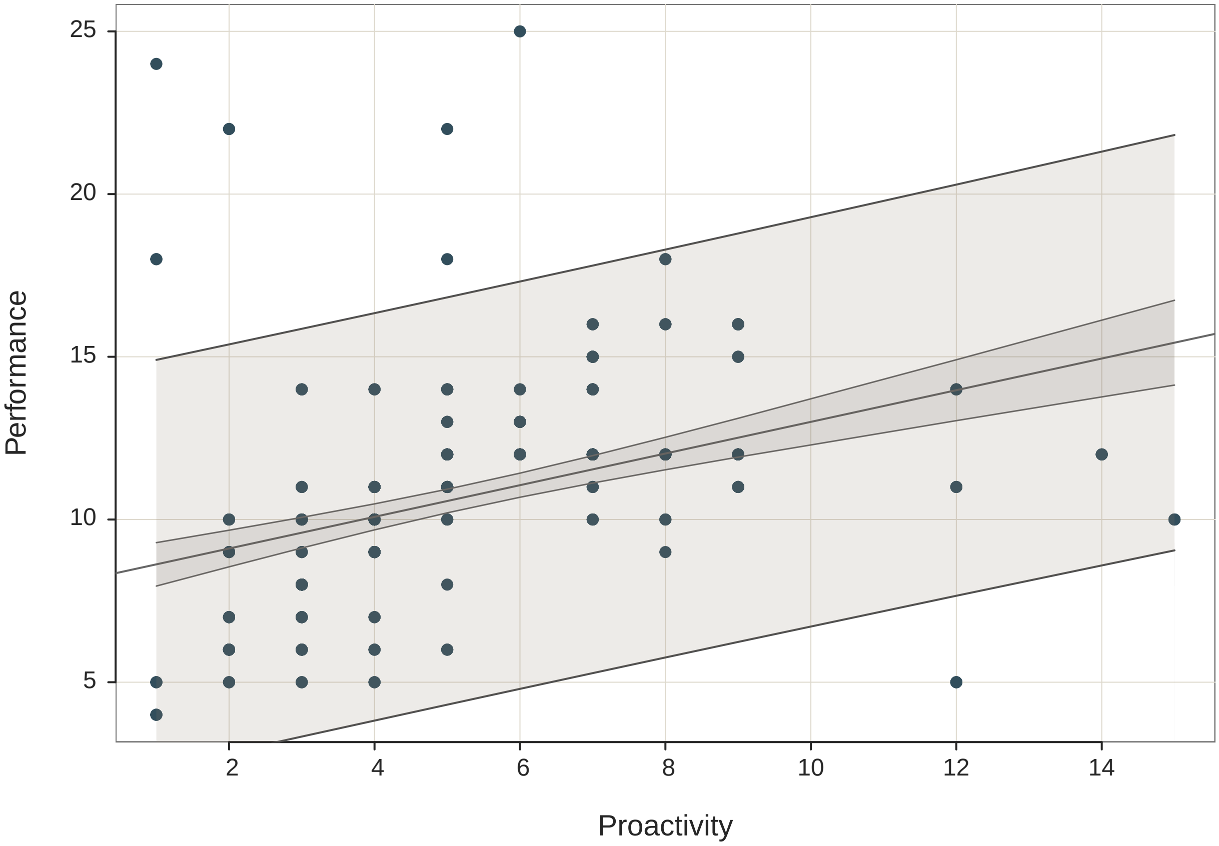


```
## >>> Suggestions
## Plot(Proactivity, Performance, enhance=TRUE)  # many options
## Plot(Proactivity, Performance, color="red")  # exterior edge color of points
## Plot(Proactivity, Performance, fit="lm", fit_se=c(.90,.99))  # fit line, stnd errors
## Plot(Proactivity, Performance, out_cut=.10)  # label top 10% from center as outliers
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 300
## Sample Correlation of Proactivity and Performance: r = 0.394
##
## Hypothesis Test of 0 Correlation:  t = 7.390,  df = 298,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.293 to 0.485
##
## Some Parameter values (can be manually set)
## -------------------------------------------------------
## fill: #324E5C   filled color of the points
## color: #324E5C  edge color of the points
## size: 0.80  size of plotted points
## jitter_y: 0.00  random vertical movement of points
## jitter_x: 0.87  random horizontal movement of points
```





Point with largest Cook's Distance of 0.14 is labeled



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Performance ~ Proactivity, data=df, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable: Proactivity
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
##
##   BASIC ANALYSIS
##
##               Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)      8.135      0.396   20.539    0.000       7.356       8.915
## Proactivity      0.487      0.066    7.390    0.000       0.357       0.616
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  3.174 for 298 degrees of freedom
## 95% range of residual variation:  12.494 = 2 * (1.968 * 3.174)
##
```

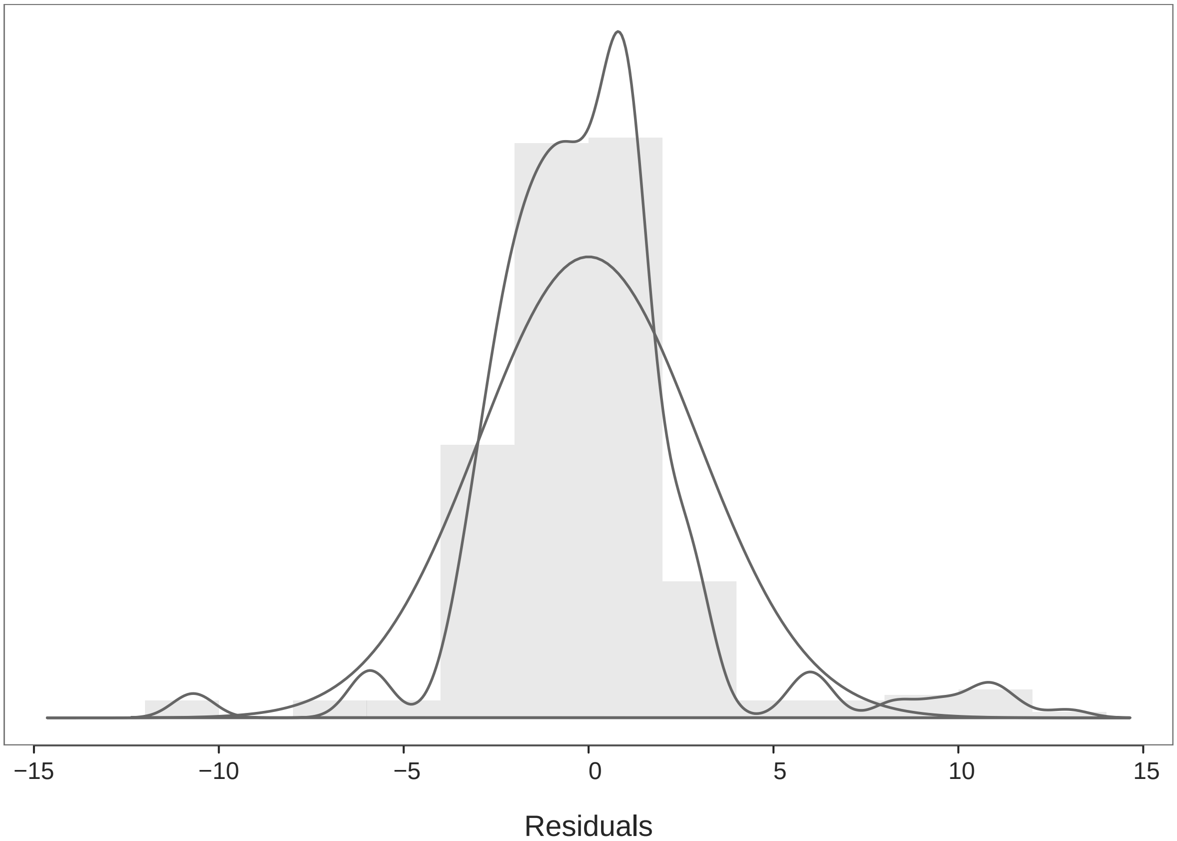
```
## R-squared:  0.155    Adjusted R-squared:  0.152    PRESS R-squared:  0.138
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 54.606     df: 1 and 298     p-value:  0.000
##
## -- Analysis of Variance
##
##              df    Sum Sq   Mean Sq   F-value   p-value
## Model          1   550.256   550.256    54.606     0.000
## Residuals    298  3002.874    10.077
## Performance  299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##              Performance Proactivity
##   Performance        1.00        0.39
##   Proactivity        0.39        1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 300 rows of data, or do res_rows="all"]
## -------------------------------------------------------------
##       Proactivity Performance fitted  resid rstdnt dffits cooks
##   201       1.000      24.000  8.622 15.378  5.070  0.545 0.137
##    70      12.000       5.000 13.973 -8.973 -2.894 -0.439 0.094
##   170      12.000       5.000 13.973 -8.973 -2.894 -0.439 0.094
##   270      12.000       5.000 13.973 -8.973 -2.894 -0.439 0.094
##    92      15.000      10.000 15.433 -5.433 -1.756 -0.375 0.070
##   192      15.000      10.000 15.433 -5.433 -1.756 -0.375 0.070
##   292      15.000      10.000 15.433 -5.433 -1.756 -0.375 0.070
##     1       2.000      22.000  9.108 12.892  4.189  0.379 0.068
##   101       2.000      22.000  9.108 12.892  4.189  0.379 0.068
##    69       1.000      18.000  8.622  9.378  3.011  0.324 0.051
##   169       1.000      18.000  8.622  9.378  3.011  0.324 0.051
##   233       6.000      25.000 11.054 13.946  4.544  0.270 0.034
##   283       5.000      22.000 10.568 11.432  3.683  0.215 0.022
##    28      14.000      12.000 14.947 -2.947 -0.945 -0.182 0.017
##   128      14.000      12.000 14.947 -2.947 -0.945 -0.182 0.017
##   228      14.000      12.000 14.947 -2.947 -0.945 -0.182 0.017
##    68       1.000       4.000  8.622 -4.622 -1.467 -0.158 0.012
##   168       1.000       4.000  8.622 -4.622 -1.467 -0.158 0.012
##   268       1.000       4.000  8.622 -4.622 -1.467 -0.158 0.012
##    20       8.000      18.000 12.027  5.973  1.896  0.152 0.011
##
##
##   PREDICTION ERROR
##
```

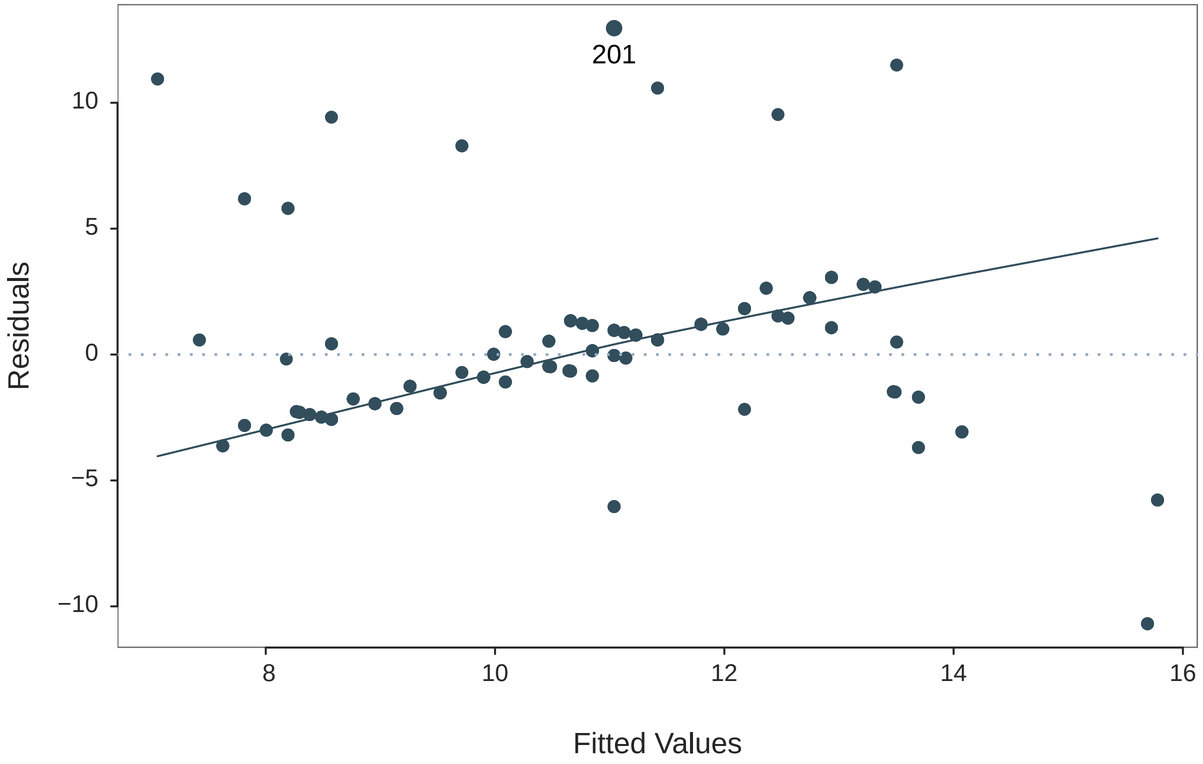
```
## Data, Predicted, Standard Error of Forecast,
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
##
##       Proactivity Performance   pred    sf  pi.lwr pi.upr  width
##    67       1.000       5.000  8.622 3.192   2.339 14.904 12.565
##    68       1.000       4.000  8.622 3.192   2.339 14.904 12.565
##    69       1.000      18.000  8.622 3.192   2.339 14.904 12.565
## ...
##   298       4.000       9.000 10.081 3.181   3.821 16.341 12.520
##     4       5.000      11.000 10.568 3.180   4.310 16.825 12.515
##    13       5.000      12.000 10.568 3.180   4.310 16.825 12.515
## ...
##   228      14.000      12.000 14.947 3.230   8.589 21.304 12.715
##    92      15.000      10.000 15.433 3.243   9.051 21.815 12.763
##   192      15.000      10.000 15.433 3.243   9.051 21.815 12.763
##
## ---------------------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Reg Line, Confidence & Prediction Intervals
## ---------------------------------------------------
```

`Proactivity` is also statistically significantly associated in a positive direction with `Performance`. The residuals when using `Proactivity` don't have strong homoscedasticity though. Also, on average they are not 0.

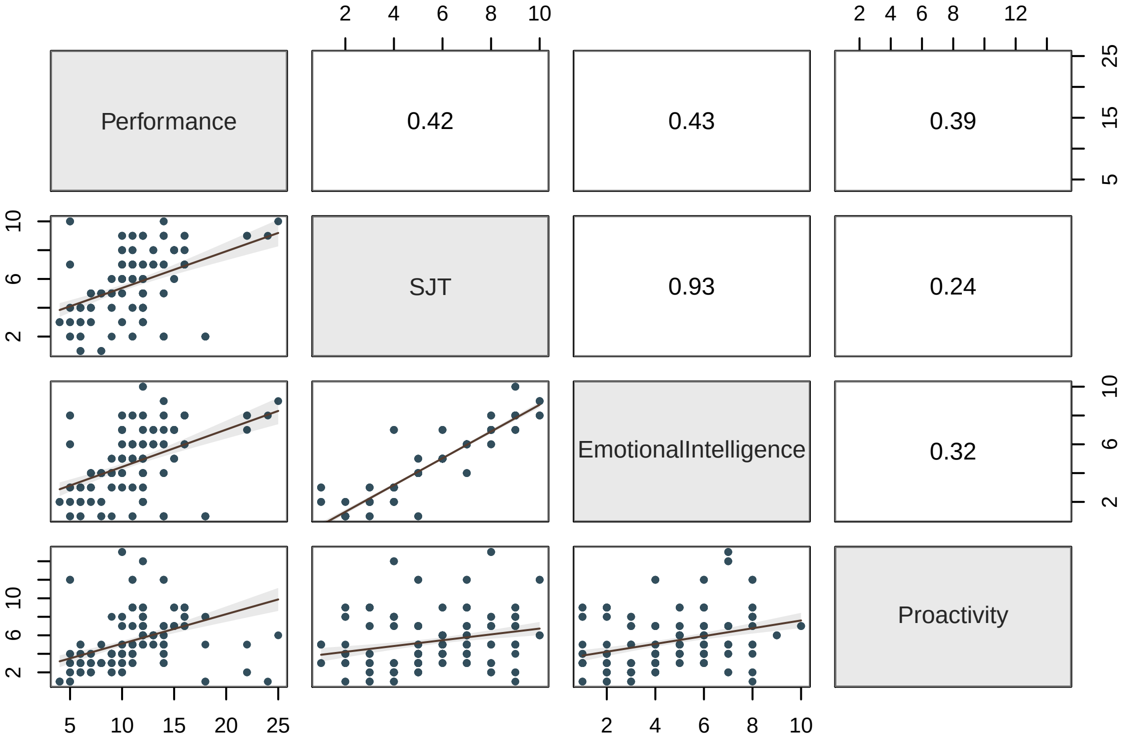
Nevertheless, we do see a linear relationship between those features and `Performance`, so we'll try out a model by using them.

```
Regression(Performance ~ SJT + EmotionalIntelligence + Proactivity, data=df)
```





Point with largest Cook's Distance of 0.14 is labeled



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Performance ~ SJT + EmotionalIntelligence + Proactivity, data=df, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable 1: SJT
## Predictor Variable 2: EmotionalIntelligence
## Predictor Variable 3: Proactivity
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
##
##   BASIC ANALYSIS
##
##                          Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
##            (Intercept)      5.627      0.569    9.893    0.000       4.507       6.746
##                    SJT      0.481      0.229    2.104    0.036       0.031       0.932
## EmotionalIntelligence      0.088      0.235    0.372    0.710      -0.375       0.550
##            Proactivity      0.379      0.066    5.755    0.000       0.250       0.509
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  2.968 for 296 degrees of freedom
## 95% range of residual variation:  11.683 = 2 * (1.968 * 2.968)
```

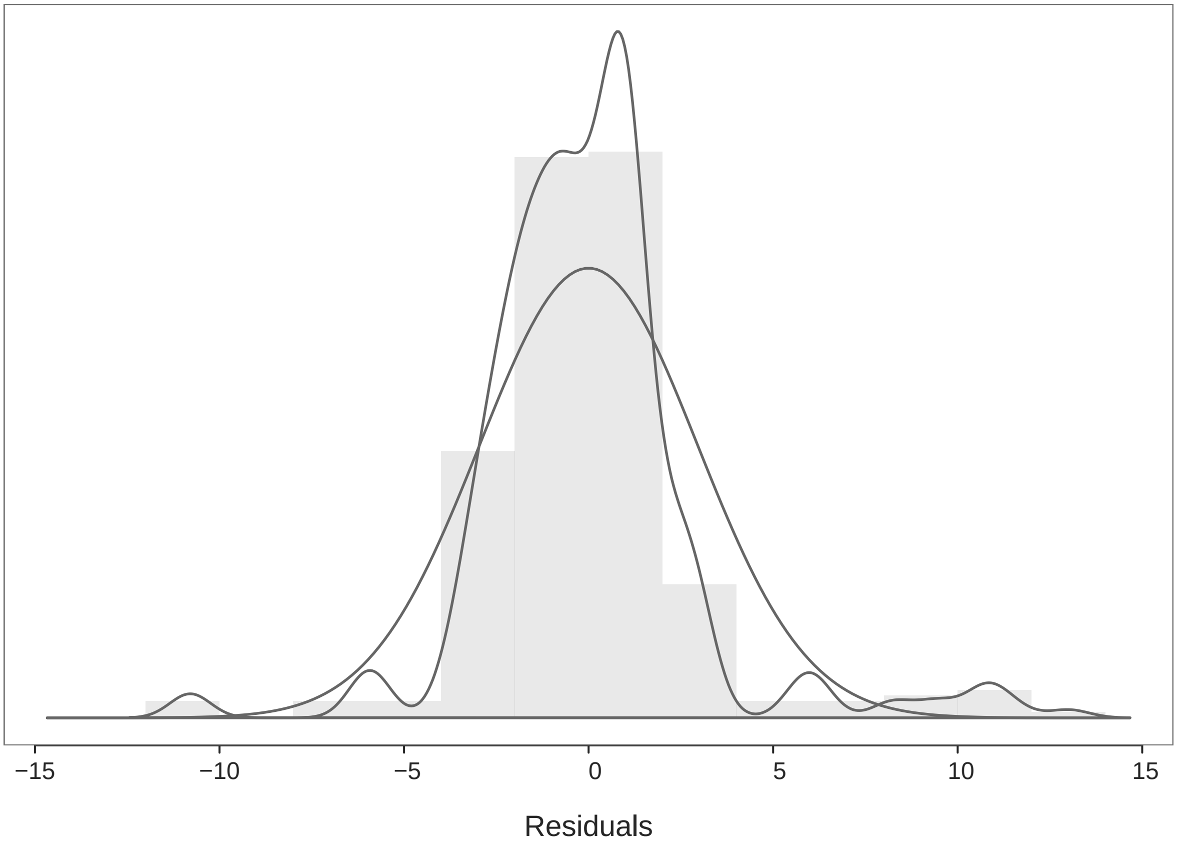
```
##
## R-squared:  0.266    Adjusted R-squared:  0.259    PRESS R-squared:  0.234
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 35.764     df: 3 and 296     p-value:  0.000
##
## -- Analysis of Variance
##
##                         df    Sum Sq   Mean Sq   F-value   p-value
##                    SJT    1   617.264   617.264    70.062     0.000
## EmotionalIntelligence    1    36.234    36.234     4.113     0.043
##            Proactivity    1   291.781   291.781    33.118     0.000
##
## Model                    3   945.278   315.093    35.764     0.000
## Residuals              296  2607.852     8.810
## Performance            299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##                         Performance  SJT EmotionalIntelligence Proactivity
##             Performance        1.00 0.42                  0.43        0.39
##                     SJT        0.42 1.00                  0.93        0.24
##   EmotionalIntelligence        0.43 0.93                  1.00        0.32
##             Proactivity        0.39 0.24                  0.32        1.00
##
##                         Tolerance       VIF
##                     SJT     0.127     7.887
##   EmotionalIntelligence     0.121     8.278
##             Proactivity     0.873     1.146
##
##  SJT EmotionalIntelligence Proactivity    adjr2    X's
##    1                     0           1    0.261      2
##    1                     1           1    0.259      3
##    0                     1           1    0.250      2
##    1                     1           0    0.178      2
##    0                     1           0    0.178      1
##    1                     0           0    0.171      1
##    0                     0           1    0.152      1
##
## [based on Thomas Lumley's leaps function from the leaps package]
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 300 rows of data, or do res_rows="all"]
## ----------------------------------------------------------------------------------------------
##          SJT EmotionalIntelligence Proactivity Performance fitted   resid rstdnt dffits cooks
##   201  9.000                 8.000       1.000      24.000 11.038  12.962  4.576  0.775 0.141
##    70 10.000                 8.000      12.000       5.000 15.691 -10.691 -3.755 -0.758 0.138
##   170 10.000                 8.000      12.000       5.000 15.691 -10.691 -3.755 -0.758 0.138
##   270 10.000                 8.000      12.000       5.000 15.691 -10.691 -3.755 -0.758 0.138
##     1  9.000                 8.000       2.000      22.000 11.418  10.582  3.682  0.563 0.076
##   101  9.000                 8.000       2.000      22.000 11.418  10.582  3.682  0.563 0.076
##   233 10.000                 9.000       6.000      25.000 13.503  11.497  4.009  0.558 0.074
##    69  2.000                 1.000       1.000      18.000  7.056  10.944  3.804  0.509 0.062
##   169  2.000                 1.000       1.000      18.000  7.056  10.944  3.804  0.509 0.062
##    92  8.000                 7.000      15.000      10.000 15.779  -5.779 -2.003 -0.440 0.048
##   192  8.000                 7.000      15.000      10.000 15.779  -5.779 -2.003 -0.440 0.048
##   292  8.000                 7.000      15.000      10.000 15.779  -5.779 -2.003 -0.440 0.048
##    20  2.000                 1.000       8.000      18.000  9.711   8.289  2.857  0.425 0.044
##   120  2.000                 1.000       8.000      18.000  9.711   8.289  2.857  0.425 0.044
##   283  9.000                 7.000       5.000      22.000 12.468   9.532  3.291  0.424 0.043
##   269  2.000                 1.000       5.000      18.000  8.573   9.427  3.250  0.388 0.036
##    97  2.000                 1.000       3.000      14.000  7.815   6.185  2.111  0.250 0.015
##   197  2.000                 1.000       3.000      14.000  7.815   6.185  2.111  0.250 0.015
##   297  2.000                 1.000       3.000      14.000  7.815   6.185  2.111  0.250 0.015
##    82  2.000                 1.000       4.000      14.000  8.194   5.806  1.979  0.231 0.013
##
##
##   PREDICTION ERROR
##
## Data, Predicted, Standard Error of Forecast,
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
##
##        SJT EmotionalIntelligence Proactivity Performance    pred    sf pi.lwr pi.upr  width
##    69 2.000                 1.000       1.000      18.000   7.056 2.994  1.163 12.949 11.785
##   169 2.000                 1.000       1.000      18.000   7.056 2.994  1.163 12.949 11.785
##    41 1.000                 2.000       3.000       8.000   7.421 3.023  1.472 13.370 11.898
## ...
##   238 4.000                 3.000       8.000      12.000  10.849 2.984  4.976 16.721 11.746
##     4 6.000                 5.000       5.000      11.000  10.849 2.974  4.997 16.701 11.704
##    47 6.000                 5.000       5.000      11.000  10.849 2.974  4.997 16.701 11.704
## ...
##   256 8.000                 8.000       8.000      16.000  13.212 2.991  7.326 19.098 11.772
##    28 4.000                 7.000      14.000      12.000  13.474 3.120  7.335 19.614 12.279
##   128 4.000                 7.000      14.000      12.000  13.474 3.120  7.335 19.614 12.279
##
## ----------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: ScatterPlot Matrix
## ----------------------------------
```

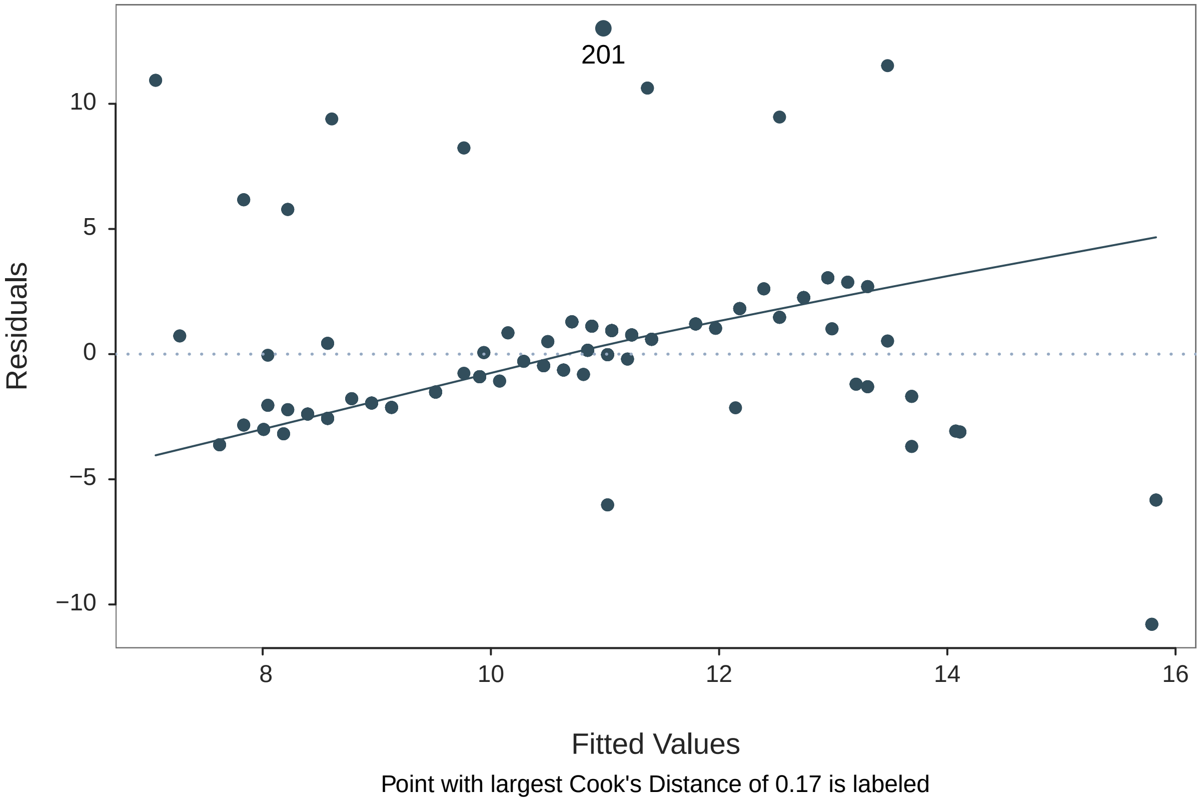
From the `Collinearity` part of the output we see that there is multicollinearity! `SJT` and `EmotionalIntelligence` are `0.93` correlated. Also the tolerance for them is below `0.2` (`1.0` is optimal) and therefore we have to drop one of them in order to do proper regression. Because the `p-value` for `EmotionalIntelligence` is largest, we remove it.

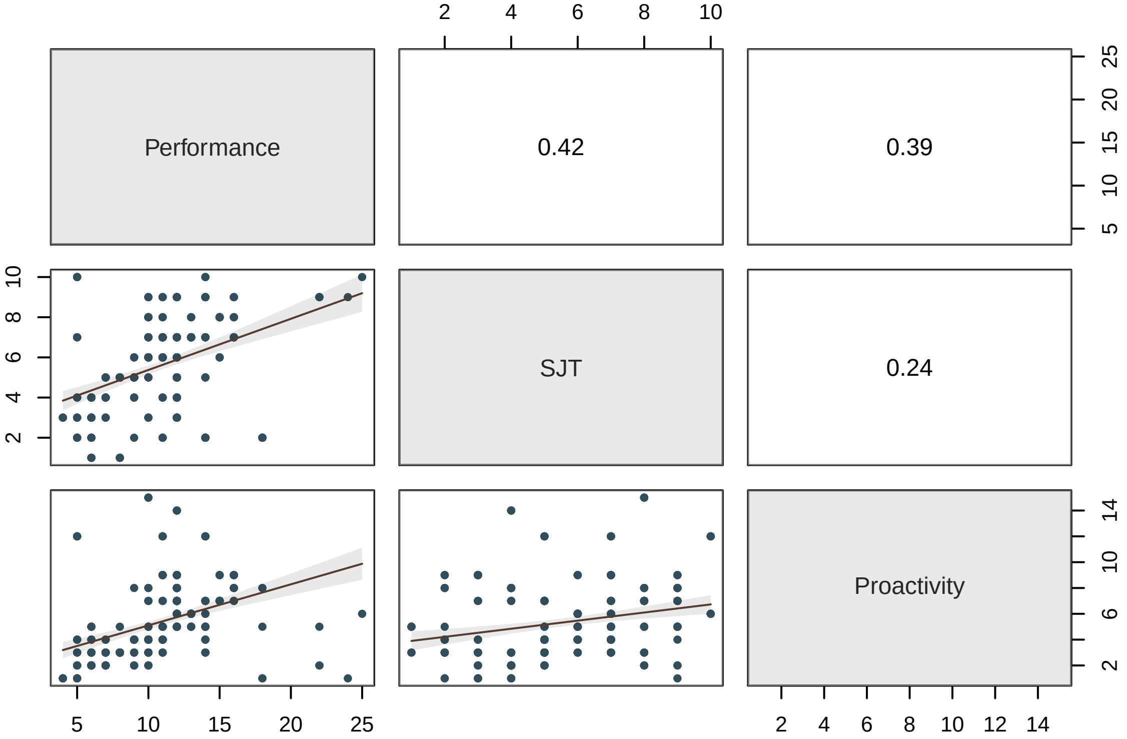
We can also notice the higher `Adjusted R-squared` score of `0.259`. We could say that this model performs better than the model with `SJT` only.

Dropping the `EmotionalIntelligence` feature

```
Regression(Performance ~ SJT + Proactivity, data=df)
```







```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Performance ~ SJT + Proactivity, data=df, Rmd="eg")
##
##
```

```
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable 1: SJT
## Predictor Variable 2: Proactivity
##
## Number of cases (rows) of data:  300
## Number of cases retained for analysis:  300
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     5.555      0.534   10.399    0.000       4.503       6.606
##         SJT     0.561      0.084    6.695    0.000       0.396       0.725
## Proactivity     0.386      0.063    6.099    0.000       0.261       0.510
##
## Standard deviation of Performance: 3.447
##
## Standard deviation of residuals:  2.964 for 297 degrees of freedom
## 95% range of residual variation:  11.666 = 2 * (1.968 * 2.964)
##
## R-squared:  0.266    Adjusted R-squared:  0.261    PRESS R-squared:  0.238
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 53.733     df: 2 and 297     p-value:  0.000
##
## -- Analysis of Variance
##
##               df    Sum Sq   Mean Sq   F-value   p-value
##         SJT    1   617.264   617.264    70.265     0.000
## Proactivity    1   326.793   326.793    37.200     0.000
##
## Model          2   944.056   472.028    53.733     0.000
## Residuals    297  2609.074     8.785
## Performance  299  3553.130    11.883
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##               Performance  SJT Proactivity
##   Performance        1.00 0.42        0.39
##           SJT        0.42 1.00        0.24
##   Proactivity        0.39 0.24        1.00
##
##               Tolerance       VIF
##           SJT     0.944     1.060
##   Proactivity     0.944     1.060
##
```

```
##  SJT Proactivity    adjr2    X's
##    1           1    0.261      2
##    1           0    0.171      1
##    0           1    0.152      1
## 
## [based on Thomas Lumley's leaps function from the leaps package]
## 
## 
##   RESIDUALS AND INFLUENCE
## 
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 300 rows of data, or do res_rows="all"]
## -----------------------------------------------------------------------
##          SJT Proactivity Performance fitted   resid rstdnt dffits cooks
##   201  9.000       1.000      24.000 10.986  13.014  4.596  0.745 0.173
##    70 10.000      12.000       5.000 15.792 -10.792 -3.781 -0.675 0.145
##   170 10.000      12.000       5.000 15.792 -10.792 -3.781 -0.675 0.145
##   270 10.000      12.000       5.000 15.792 -10.792 -3.781 -0.675 0.145
##   233 10.000       6.000      25.000 13.476  11.524  4.023  0.551 0.096
##     1  9.000       2.000      22.000 11.372  10.628  3.701  0.544 0.094
##   101  9.000       2.000      22.000 11.372  10.628  3.701  0.544 0.094
##    69  2.000       1.000      18.000  7.062  10.938  3.807  0.509 0.083
##   169  2.000       1.000      18.000  7.062  10.938  3.807  0.509 0.083
##    92  8.000      15.000      10.000 15.829  -5.829 -2.022 -0.433 0.062
##   192  8.000      15.000      10.000 15.829  -5.829 -2.022 -0.433 0.062
##   292  8.000      15.000      10.000 15.829  -5.829 -2.022 -0.433 0.062
##    20  2.000       8.000      18.000  9.764   8.236  2.839  0.399 0.052
##   120  2.000       8.000      18.000  9.764   8.236  2.839  0.399 0.052
##   283  9.000       5.000      22.000 12.530   9.470  3.268  0.377 0.046
##   269  2.000       5.000      18.000  8.606   9.394  3.241  0.374 0.045
##    97  2.000       3.000      14.000  7.834   6.166  2.107  0.247 0.020
##   197  2.000       3.000      14.000  7.834   6.166  2.107  0.247 0.020
##   297  2.000       3.000      14.000  7.834   6.166  2.107  0.247 0.020
##    82  2.000       4.000      14.000  8.220   5.780  1.972  0.225 0.017
## 
## 
##   PREDICTION ERROR
## 
## Data, Predicted, Standard Error of Forecast, 
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
## 
##         SJT Proactivity Performance   pred    sf pi.lwr pi.upr  width
##    69 2.000       1.000      18.000  7.062 2.990  1.178 12.946 11.768
##   169 2.000       1.000      18.000  7.062 2.990  1.178 12.946 11.768
##    41 1.000       3.000       8.000  7.273 2.992  1.384 13.162 11.778
## ...
##   210 8.000       2.000      10.000 10.811 2.987  4.933 16.689 11.756
##     4 6.000       5.000      11.000 10.848 2.969  5.005 16.691 11.687
##    47 6.000       5.000      11.000 10.848 2.969  5.005 16.691 11.687
## ...
##   226 7.000      12.000      11.000 14.110 2.997  8.212 20.009 11.796
##    92 8.000      15.000      10.000 15.829 3.028  9.869 21.788 11.919
##   192 8.000      15.000      10.000 15.829 3.028  9.869 21.788 11.919
## 
## ----------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: ScatterPlot Matrix
## ----------------------------------
```

Now the non-collinearity assumption is met based on the `Tolerance` statistics. The residuals are normally distributed based on `Plot 1: Distribution of Residuals`. There is no evidence against heteroscedasticity in `Plot 2: Residuals vs Fitted Values`. Also, although there is a slight bend, the average error is not that far from `0`.

The `Adjusted R-squared` value is `0.261`, i.e. we keep improving the model and explaining more and more variance.

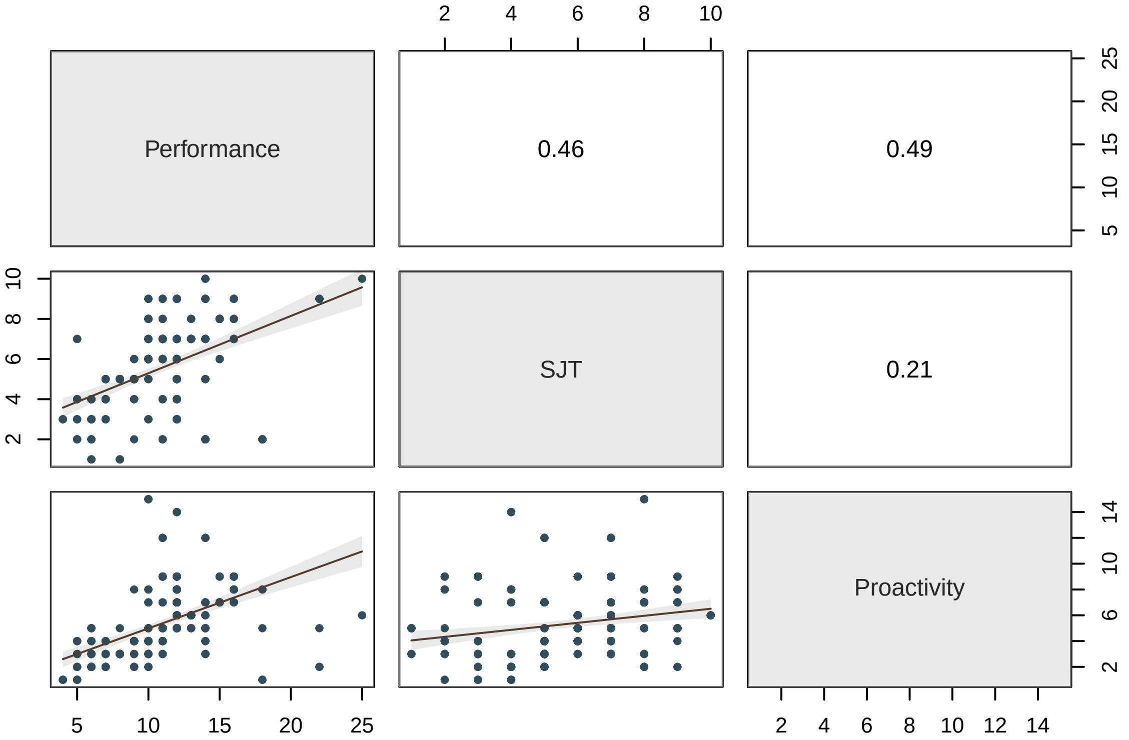
Looking at the `BASIC ANALYSIS` part of the output, we that when `Proactivity` is fixed, for every `1` unit increase in `SJT`, there is a `0.561` increase in `Performance`. When `SJT` is fixed, for every `1` unit increase in `Proactivity`, there is a `0.386` increase in `Performance`. This goes to say that an increase in `Proactivity` is better (more important) than an increase in `SJT`.

This is the line we would get: `Performance = 5.555 + 0.561*SJT + 0.386*Proactivity`.

Removing potential outliers

From the `RESIDUALS AND INFLUENCE` part of the output, we see that observations `201`, `70`, `170`, and `270` have a large `Cook's Distance`, i.e. our model performed poorly on them. We can test to see if removing them yields a better result.

```
reg_brief(Performance ~ SJT + Proactivity, data=df, rows = (!ID %in% c(201, 70, 170, 270)))
```



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## reg(Performance ~ SJT + Proactivity, data=df, rows=(!ID %in% c(201, 70, 170, 270)), Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df
##
## Response Variable: Performance
## Predictor Variable 1: SJT
## Predictor Variable 2: Proactivity
##
## Number of cases (rows) of data:  296
## Number of cases retained for analysis:  296
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     4.734      0.491    9.649    0.000       3.769       5.700
##         SJT     0.608      0.076    7.966    0.000       0.458       0.758
## Proactivity     0.504      0.058    8.688    0.000       0.390       0.619
##
## Standard deviation of Performance: 3.334
##
## Standard deviation of residuals:  2.646 for 293 degrees of freedom
## 95% range of residual variation:  10.417 = 2 * (1.968 * 2.646)
##
## R-squared:  0.374    Adjusted R-squared:  0.370    PRESS R-squared:  0.351
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 87.570     df: 2 and 293     p-value:  0.000
##
## -- Analysis of Variance
##
##               df    Sum Sq   Mean Sq   F-value   p-value
##         SJT    1   697.913   697.913    99.656     0.000
## Proactivity    1   528.628   528.628    75.484     0.000
##
## Model          2  1226.542   613.271    87.570     0.000
## Residuals    293  2051.945     7.003
## Performance  295  3278.486    11.114
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##
##   RESIDUALS AND INFLUENCE
##
##
```

```
##   PREDICTION ERROR
```

The new `Adjusted R-squared` is `0.370` which means that with this data it is better to remove those observations but we should be careful before drawing conclusions. Those samples may not be true outliers, but rather the result of poor sampling.