Simple linear regression

Contents

Introduction

> "All models are wrong, but some are useful."
>
> — **George E. P. Box**

After reading this chapter you will be able to:

- Understand the concept of a model.
- Describe two ways in which regression coefficients are derived.
- Estimate and visualize a regression model using `R`.
- Interpret regression coefficients and statistics in the context of real-world problems.
- Use a regression model to make predictions.

Data

This is our second lab when we are still considering 2 dimensions and instead of calculating univariate statistics by groups (or factors) of other variable - we will measure their common relationships based on co-variance and correlation coefficients and construct a simple linear function of regression.

Our dataset contains information about a random sample of apartments. This data can be used for a lot of purposes such as price prediction to exemplify the use of linear regression in Machine Learning.

The columns in the given dataset is as follows:

- price_PLN
- price_EUR
- rooms
- size (sq meters)
- district
- building_type

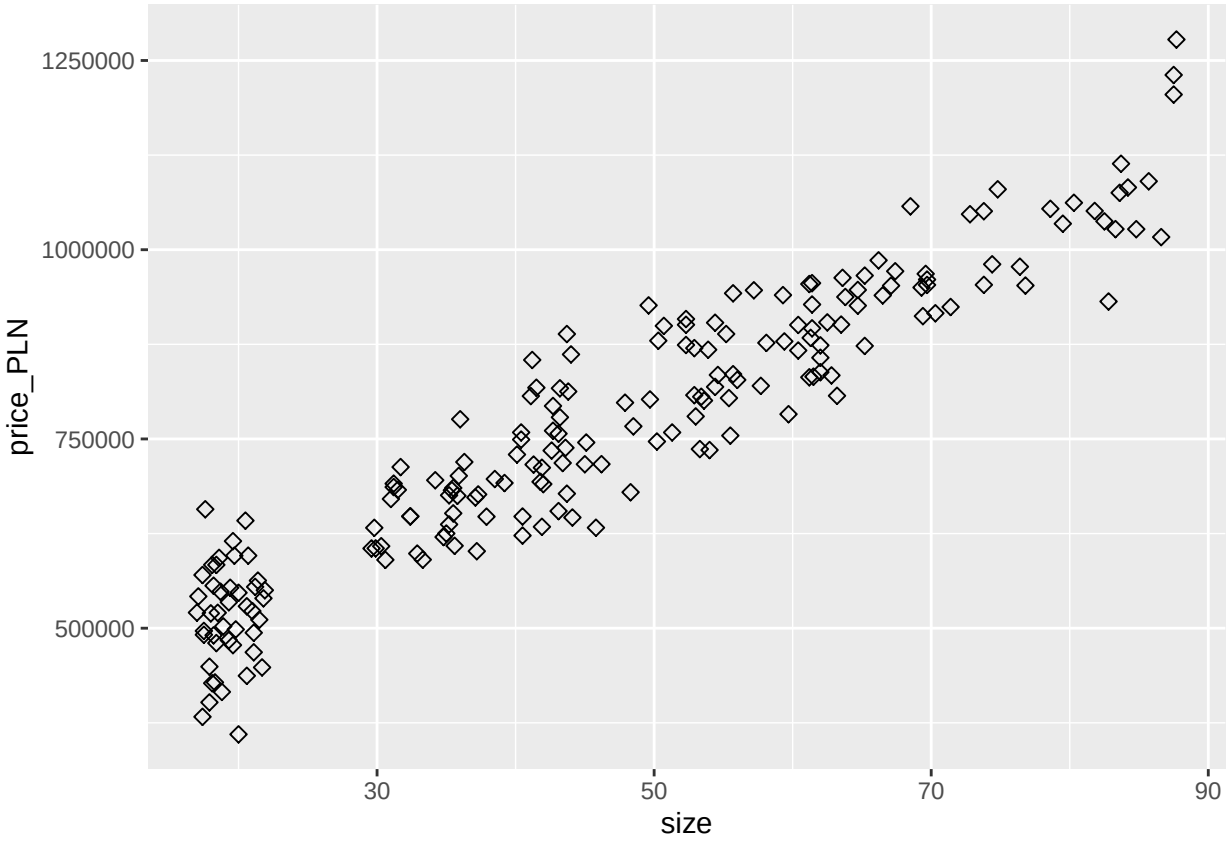
Univariate analysis

Recall that we discussed characteristics such as center, spread, and shape. It's also useful to be able to describe the relationship of two numerical variables, such as `price_PLN` and `size` above.

We can also describe the relationship between these two variables using the univariate analysis of the response variable by groups of the independent variable (explanatory). Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.

Scatterplots

First let's visualize our quantitative relationships using scatterplots.



*If the distribution of the response variable is skewed - don't forget about the necessary transformations (i.e. log function).

Batter up

Our aim today is to summarize relationships both graphically and numerically in order to find which variable, if any, helps us best predict the total amount of $ spent in our restaurant. If you are already interested in the qualitative predictors, please be patient - we still need 2-3 weeks to discuss the multivariate regression functions ;-)

Broadly speaking, we would like to model the relationship between X and Y using the form

$$Y = f(X) + \epsilon.$$

The function f describes the functional relationship between the two variables, and the ϵ term is used to account for error. This indicates that if we plug in a given value of X as input, our output is a value of Y, within a certain range of error. You could think of this a number of ways:

- Response = Prediction + Error
- Response = Signal + Noise

- Response = Model + Unexplained
- Response = Deterministic + Random
- Response = Explainable + Unexplainable

Ok, if the relationship looks linear, we can quantify the strength of the relationship with the Pearson's correlation coefficient.

```
cor(apartments$price_PLN, apartments$size)
```

```
## [1] 0.9453427
```

Sum of squared residuals

Think back to the way that we described the distribution of a single variable.

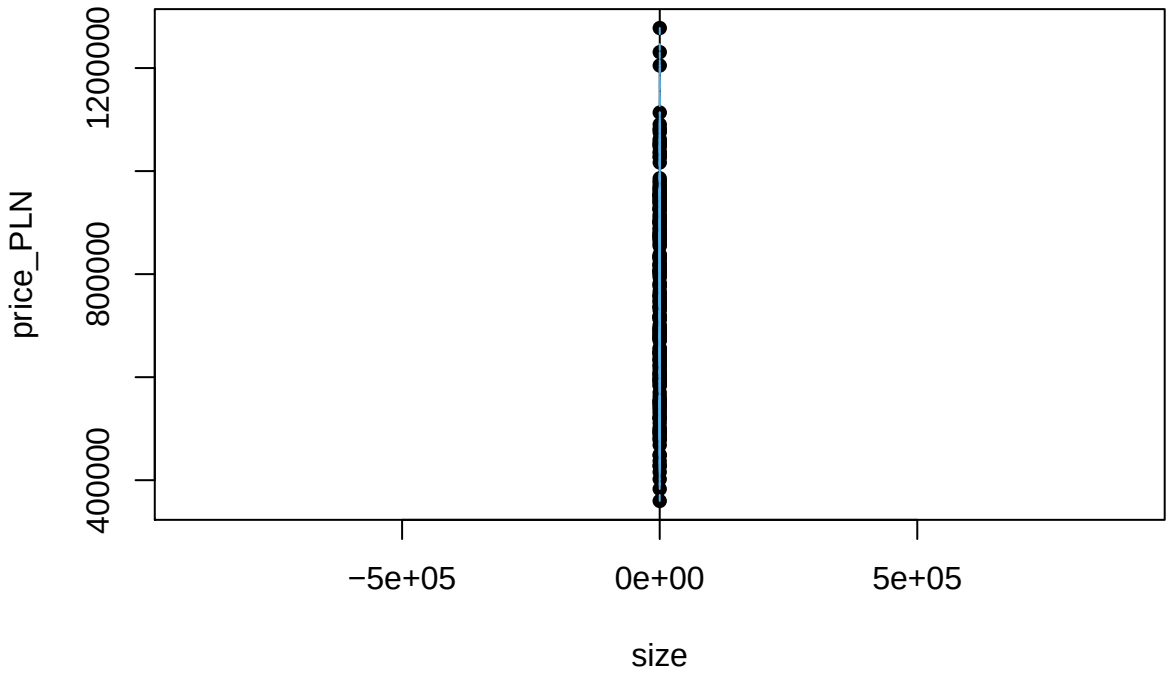
Just as we used the mean and standard deviation to summarize a single variable, we can summarize the relationship between these two variables by finding the line that best follows their association. Use the following interactive function to select the line that you think does the best job of going through the cloud of points.

```
## Loading required package: BayesFactor
```

```
## Loading required package: coda
```

```
## Loading required package: Matrix
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```



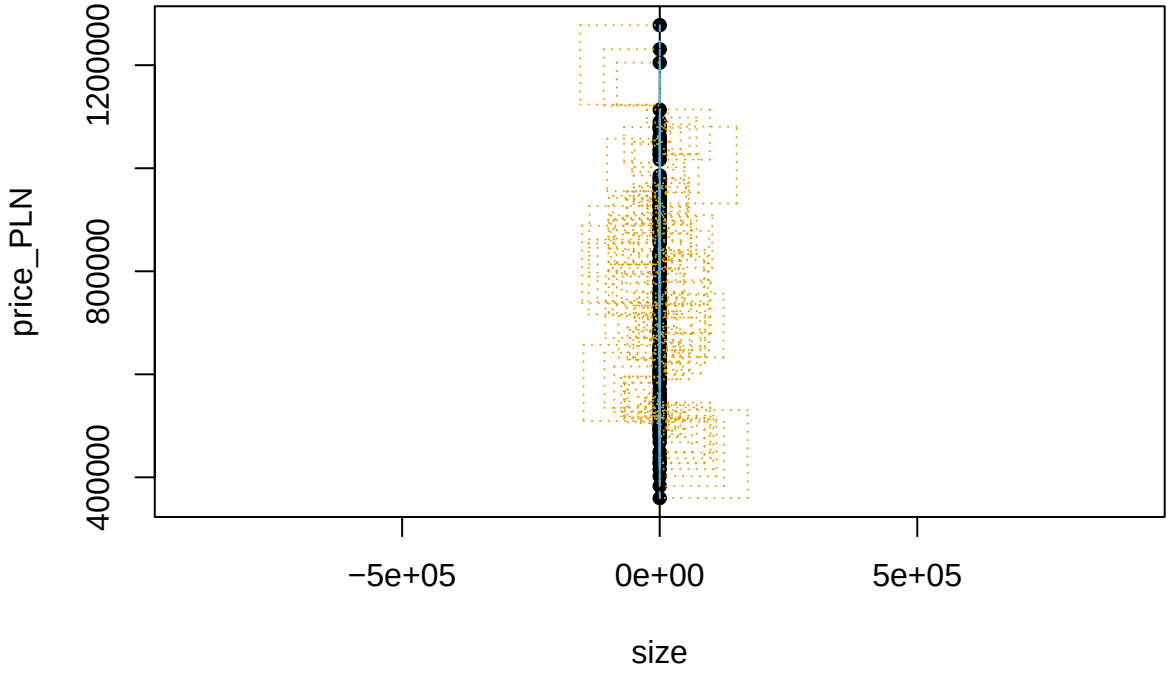

```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
```

```
## (Intercept)            x
##      355269         8762
##
## Sum of Squares:  7.32806e+11
```

After running this command, you'll be prompted to click two points on the plotto define a line. Once you've done that, the line you specified will be shown in black and the residuals in blue. Note that there are 50 residuals, one for each of the 50 customers. Recall that the residuals are the difference between the observed values and the values predicted by the line:

$$e_i = y_i - \hat{y}_i$$

The most common way to do linear regression is to select the line that minimizes the sum of squared residuals. To visualize the squared residuals, you can rerun the plot command and add the argument `showSquares = TRUE`.




```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##      355269         8762
##
## Sum of Squares:  7.32806e+11
```

Note that the output from the `plot_ss` function provides you with the slope and intercept of your line as well as the sum of squares.

Using `plot_ss`, choose a line that does a good job of minimizing the sum of squares. Run the function several times. What was the smallest sum of squares that you got? How does it compare to your friends (send a message on chat).

The linear model

It is rather cumbersome to try to get the correct least squares line, i.e. the line that minimizes the sum of squared residuals, through trial and error. Instead we can use the `lm` function in R to fit the linear model (a.k.a. regression line).

```
model1 <- lm(price_PLN ~ size, data = apartments)
```

The first argument in the function `lm` is a formula that takes the form `y ~ x`. Here it can be read that we want to make a linear model of `price_PLN` as a function of `size`. The second argument specifies that R should look in the `apartments` data frame to find the `price_PLN` and `size` variables.

The output of `lm` is an object that contains all of the information we needabout the linear model that was just fit. We can access this information using the summary function.

```
summary(model1)
```

```
##
## Call:
## lm(formula = price_PLN ~ size, data = apartments)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170735  -40337   -5216   40794  154018
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 355269.3    10814.5   32.85   <2e-16 ***
## size          8761.7      214.8   40.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60840 on 198 degrees of freedom
## Multiple R-squared:  0.8937, Adjusted R-squared:  0.8931
## F-statistic:  1664 on 1 and 198 DF,  p-value: < 2.2e-16
```

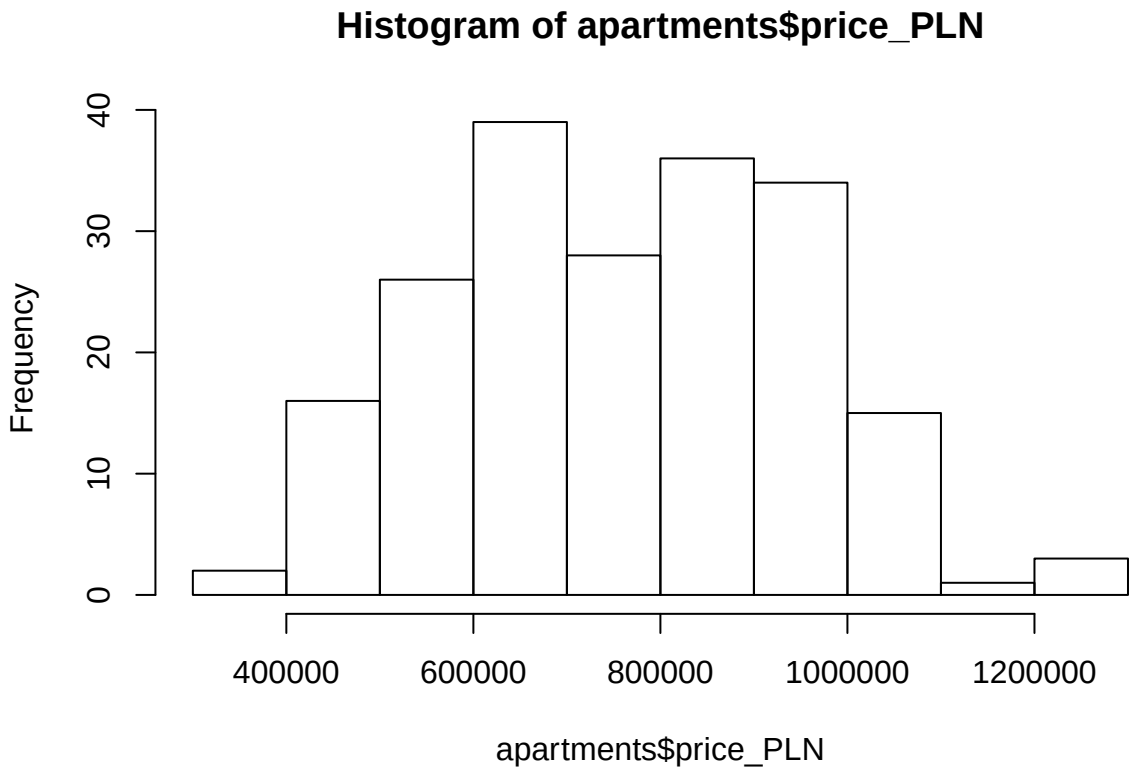
Let's consider this output piece by piece. First, the formula used to describe the model is shown at the top. After the formula you find the five-number summary of the residuals. The "Coefficients" table shown next is key; its first column displays the linear model's y-intercept and the coefficient of `log_income`. With this table, we can write down the least squares regression line for the linear model:

$$\hat{y} = 355269.3 + 8761.7 * size$$

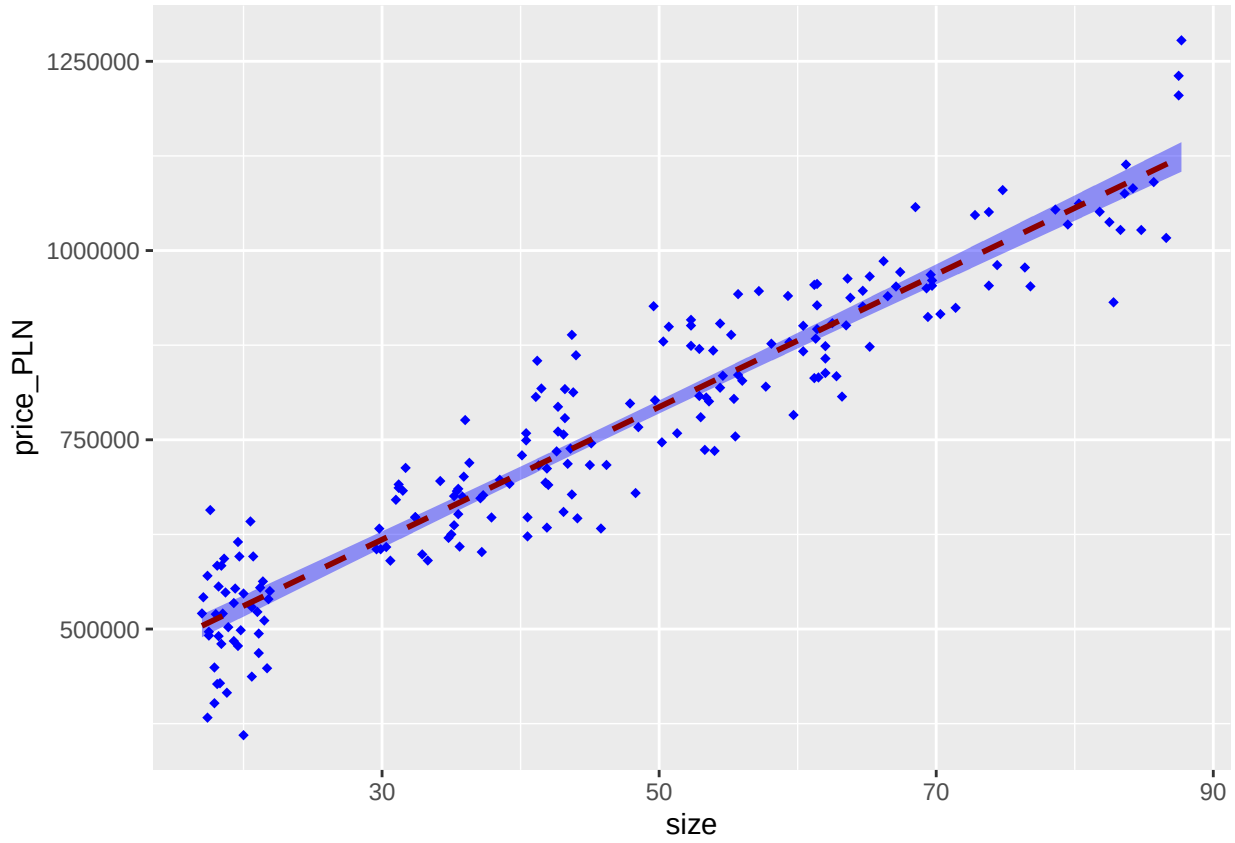
One last piece of information we will discuss from the summary output is the Multiple R-squared, or more simply, R^2. The R^2 value represents the proportion of variability in the response variable that is explained by the explanatory variable. For this model, 89.37% of the variability of apartments in prices (in PLN) is explained by size in square meters.

Prediction and prediction errors

Let's create a scatterplot with the least squares line laid on top.




```
## `geom_smooth()` using formula 'y ~ x'
```



The function `abline` plots a line based on its slope and intercept. Here, we used a shortcut by providing the

model `model1`, which contains both parameter estimates. This line can be used to predict y at any value of x. When predictions are made for values of x that are beyond the range of the observed data, it is referred to as *extrapolation* and is not usually recommended. However, predictions made within the range of the data are more reliable. They're also used to compute the residuals.

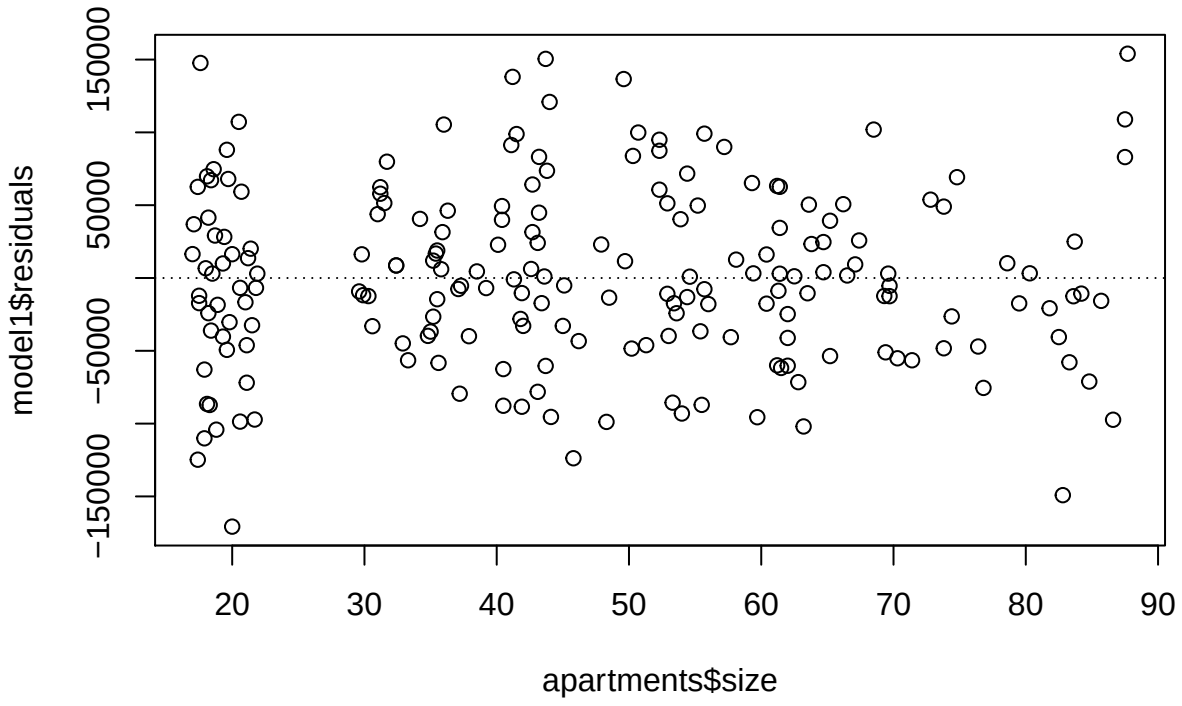
If a real estate manager saw the least squares regression line and not the actual data, what price (in PLN) would he or she predict for an apartment to have 50 sq. meters? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction?

Model diagnostics

To assess whether the linear model is reliable, we need to check for (1) linearity, (2) nearly normal residuals, and (3) constant variability.

Linearity: We already checked if the relationship between price_PLN and size is linear using a scatterplot. We should also verify this condition with a plot of the residuals vs. size. Recall that any code following a # is intended to be a comment that helps understand the code but is ignored by R.

```
plot(model1$residuals ~ apartments$size)
abline(h = 0, lty = 3)  # adds a horizontal dashed line at y = 0
```

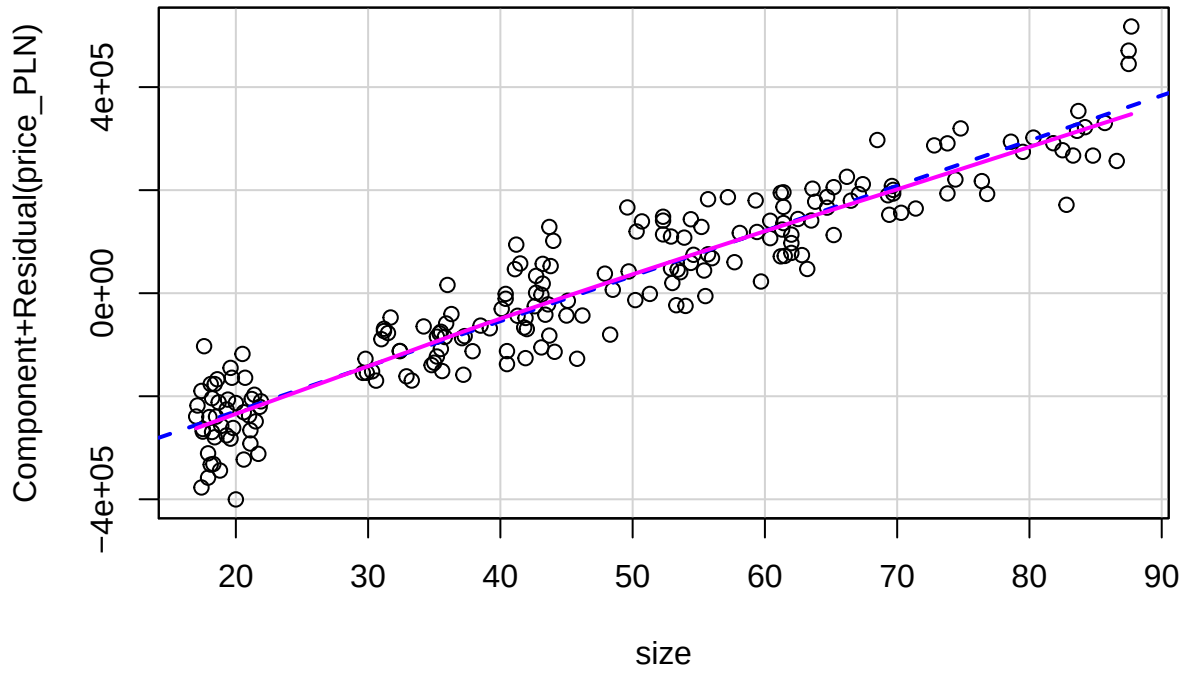


We can also verify linearity using some external libraries.

```
# Assessing linearity
library(car)
```

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

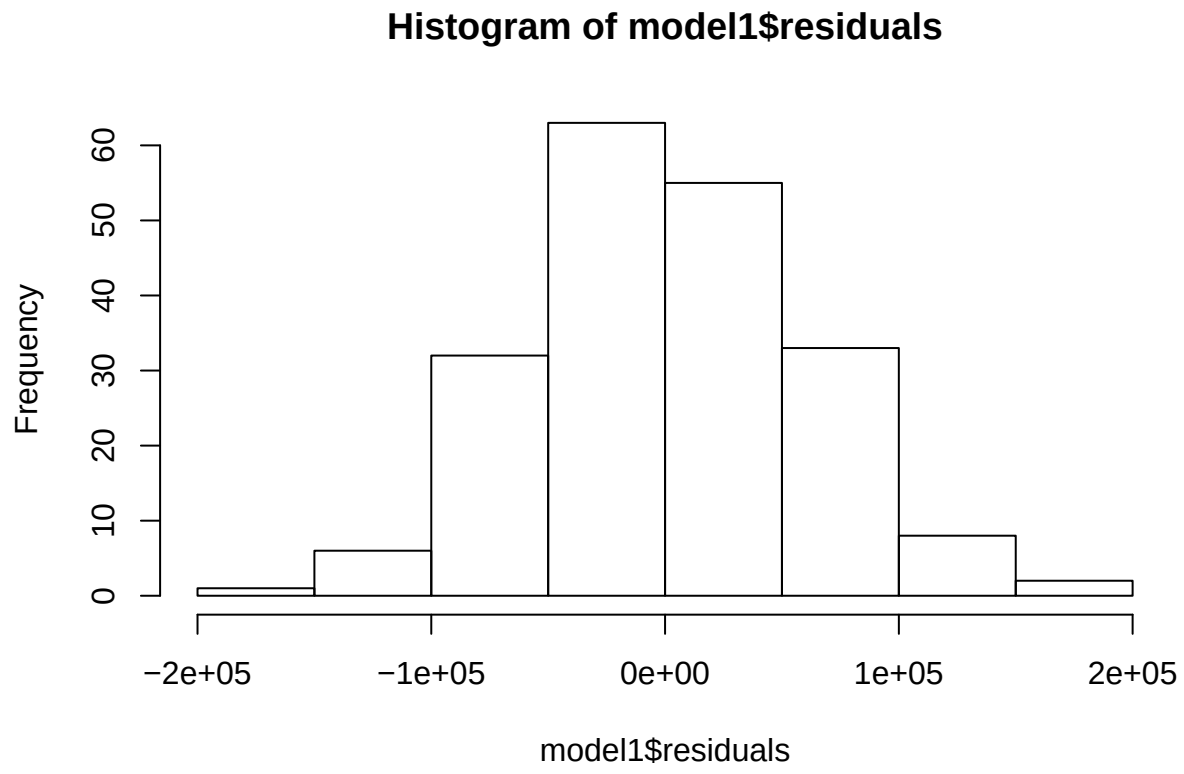
```
crPlots(model1)
```



Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between runs and at-bats?

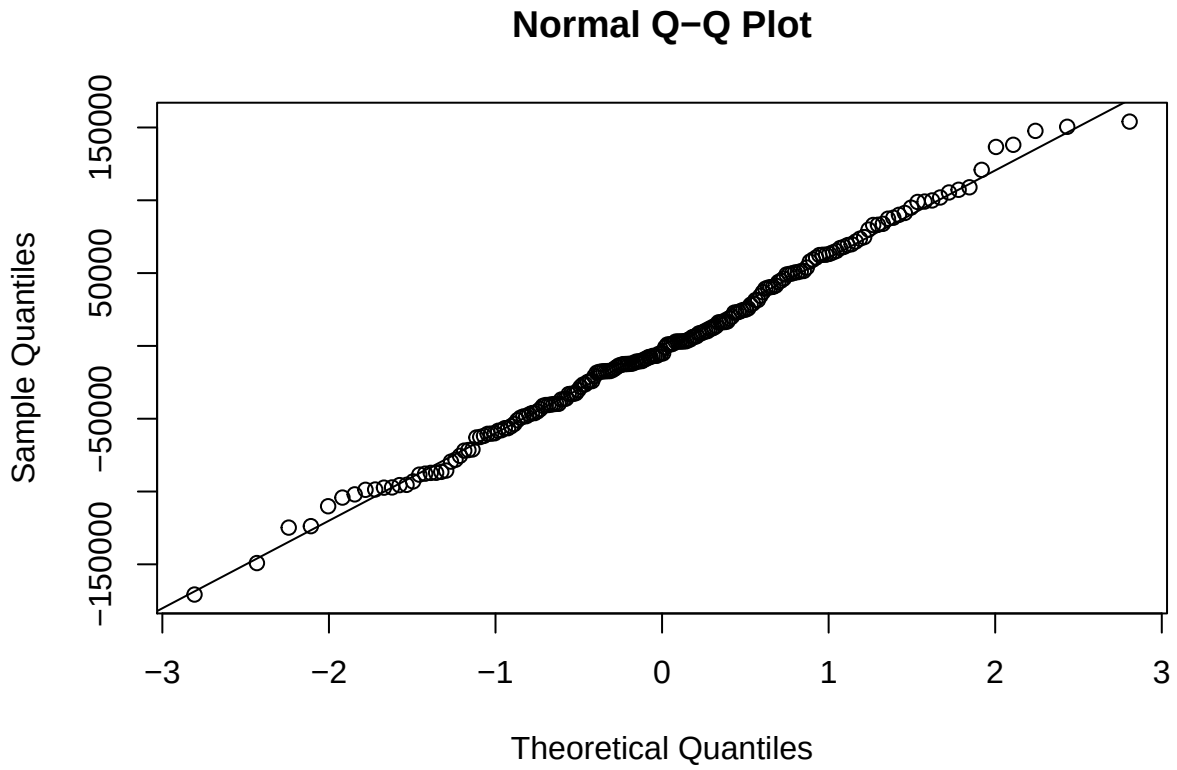
Nearly normal residuals: To check this condition, we can look at a histogram

```
hist(model1$residuals)
```



or a normal probability plot of the residuals.

```
qqnorm(model1$residuals)
qqline(model1$residuals)  # adds diagonal line to the normal prob plot
```



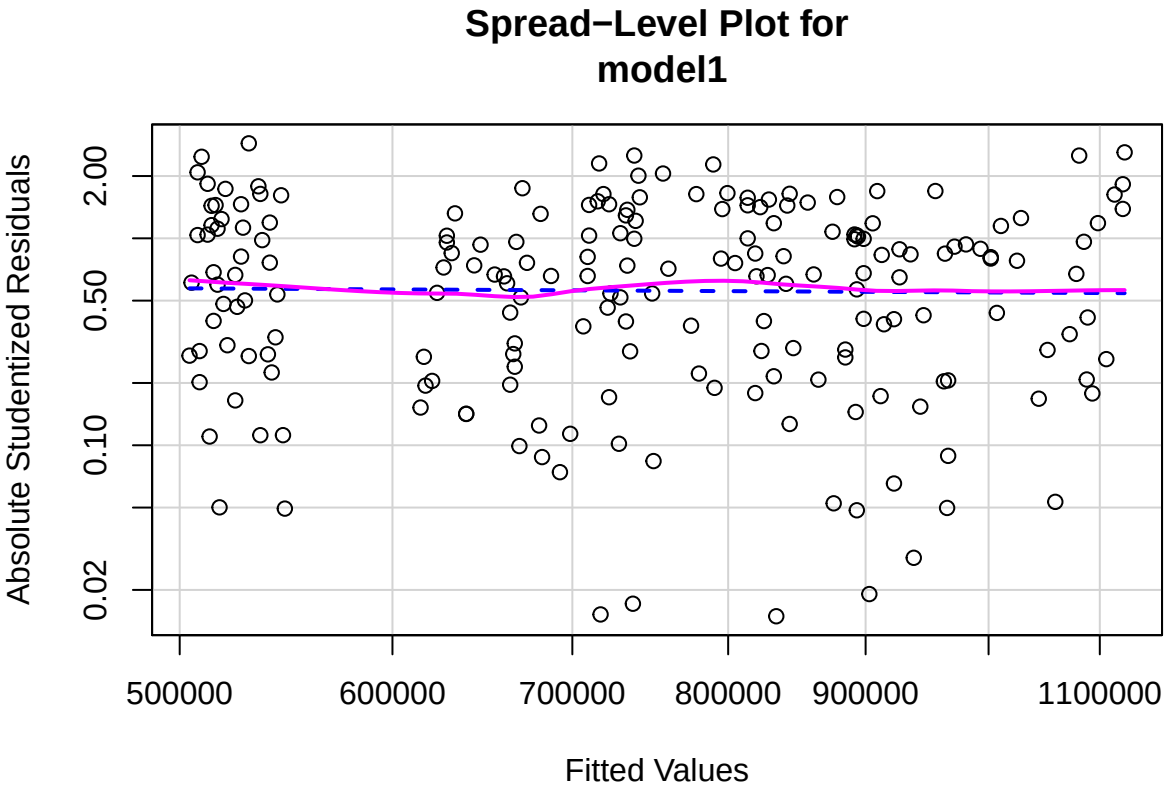
Based on the histogram and the normal probability plot, does the nearly normal residuals condition appear to be met?

Constant variability:

```
# Assessing homoscedasticity
library(car)
ncvTest(model1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.02459333, Df = 1, p = 0.87538
```

```
spreadLevelPlot(model1)
```




```
##
## Suggested power transformation:  1.066677
```

Based on the plot in (1), does the constant variability condition appear to be met?

```r
# Global test of linear model assumptions
#install.packages("gvlma")
library(gvlma)
gvmodel <- gvlma(model1)
summary(gvmodel)
```

```
##
## Call:
## lm(formula = price_PLN ~ size, data = apartments)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170735  -40337   -5216   40794  154018
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 355269.3    10814.5   32.85   <2e-16 ***
## size          8761.7      214.8   40.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60840 on 198 degrees of freedom
```

```
## Multiple R-squared:  0.8937, Adjusted R-squared:  0.8931
## F-statistic:  1664 on 1 and 198 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = model1)
##
##                     Value p-value                Decision
## Global Stat        1.9143  0.7515 Assumptions acceptable.
## Skewness           0.3701  0.5429 Assumptions acceptable.
## Kurtosis           0.1054  0.7454 Assumptions acceptable.
## Link Function      1.1400  0.2856 Assumptions acceptable.
## Heteroscedasticity 0.2988  0.5846 Assumptions acceptable.
```