Tutorial 6: Bivariate regression

Johannes Karreth

RPOS 517, Week 6

This tutorial shows you:

- how to fit a bivariate regression model
- how to create a bivariate scatterplot with a line of best fit
- how to create a residual plot
- how to present the output from a regression model in a table

Note on copying & pasting code from the PDF version of this tutorial: Please note that you may run into trouble if you copy & paste code from the PDF version of this tutorial into your R script. When the PDF is created, some characters (for instance, quotation marks or indentations) are converted into non-text characters that R won't recognize. To use code from this tutorial, please type it yourself into your R script or you may copy & paste code from the *source file* for this tutorial which is posted on my website.

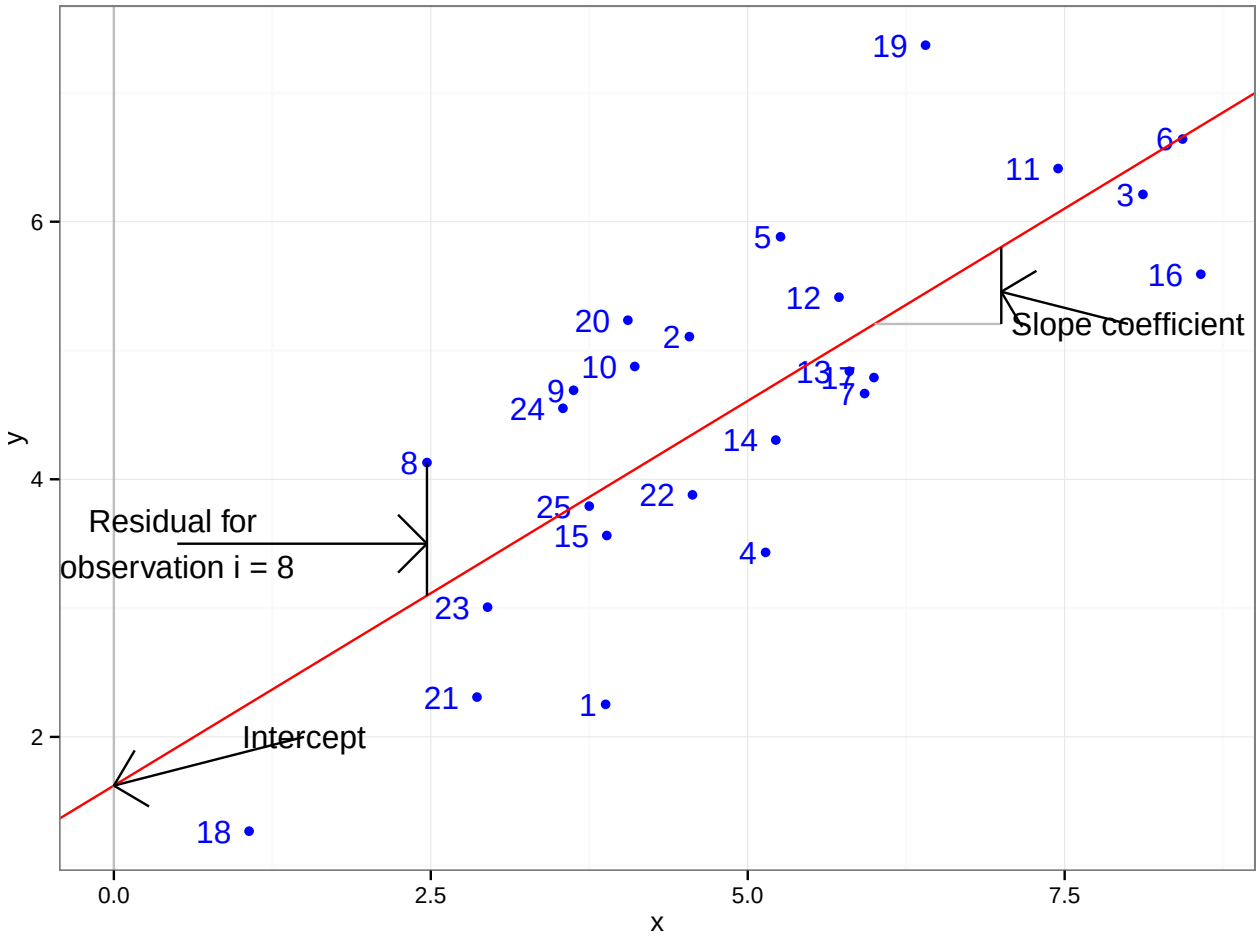
Regression: Basic concepts

So far, you have examined relationships between continuous and binary variables (differences of means & differences of proportions). In your textbook sections for today (*Applied Regression Analysis sections 5.1 and 5.2*, henceforth *AR*), you first encounter regression analysis as a technique to express a **linear relationship** between two **continuous variables**. We typically refer to these variables as **outcome variable** y and **explanatory** or **predictor** variable x. Linear regression allows you to express the relationship between x and y through **coefficients** in the following equation:

$$y_i = \alpha + \beta x_i + \varepsilon$$

α is the so-called **intercept**; β is the **slope coefficient** for your explanatory variable x; ε is the **residual**. Other ways to express these are β_0 for the intercept and β_1 for the slope coefficient, or to use Roman instead of Greek letters. The index i stands for the individual observations of your data. Also see the summary in *AR*, top of p. 82.

The following picture illustrates these concepts:



Basic workflow

As you just read, linear regression is a useful tool to identify a linear relationship between two continuous variables. To do this, we typically perform the following steps. We'll expand these steps in the next few weeks and supply the logic behind them, but the underlying routine remains the same.

1. Based on your research question and theory, operationalize your outcome and explanatory variables.
2. If your outcome variable is continuous, linear regression may be an appropriate tool to examine data and evaluate your hypothesis.
3. Specify your linear regression equation in the form $y_i = \alpha + \beta x_i + \varepsilon$
    - β will express the relationship between your explanatory variable x and your outcome y.
    - As you will see later today, β can be interpreted as: "a one-unit change in x results in a β-unit change in y."
    - What does your hypothesis predict for β? Do you expect β be positive or negative? How large do you expect β to be?
4. Create a scatterplot of your explanatory and predictor variables.
5. Fit a linear regression model by identifying the straight line that best fits the above scatterplot.

6. Check how well this model fits your data and whether it violates any of the regression assumptions. Note: not all of these assumptions can be directly checked by inspecting your data.
    - Linearity (A1)
    - Constant error variance (A2)
    - Normality of the errors (A3)
    - Independence of observations (A4)
    - No correlation between x and ε (A5)
7. Interpret β in a meaningful way and evaluate how it relates to your hypothesized value of β.

The remainder of this tutorial explains these concepts with some example data about various performance indicators of baseball teams.

Batter up

The movie "Moneyball" focuses on the "quest for the secret of success in baseball". It follows a low-budget team, the Oakland Athletics, who believed that underused statistics, such as a player's ability to get on base, better predict the ability to score runs than typical statistics like home runs, RBIs (runs batted in), and batting average. Obtaining players who excelled in these underused statistics turned out to be much more affordable for the team.

In this tutorial we'll be looking at data from all 30 Major League Baseball teams and examining the linear relationship between runs scored in a season and a number of other player statistics. Our aim will be to summarize these relationships both graphically and numerically in order to find which variable, if any, helps us best predict a team's runs scored in a season.

The data

Let's load up the data for the 2011 season.

```
mlb11 <- read.csv("http://www.jkarreth.net/files/mlb11.csv")
head(mlb11)
```

```
##                 team runs at_bats hits homeruns bat_avg strikeouts
## 1       Texas Rangers  855    5659 1599      210   0.283        930
## 2      Boston Red Sox  875    5710 1600      203   0.280       1108
## 3      Detroit Tigers  787    5563 1540      169   0.277       1143
## 4  Kansas City Royals  730    5672 1560      129   0.275       1006
## 5 St. Louis Cardinals  762    5532 1513      162   0.273        978
## 6       New York Mets  718    5600 1477      108   0.264       1085
##   stolen_bases wins new_onbase new_slug new_obs
## 1          143   96      0.340    0.460   0.800
## 2          102   90      0.349    0.461   0.810
## 3           49   95      0.340    0.434   0.773
## 4          153   71      0.329    0.415   0.744
## 5           57   90      0.341    0.425   0.766
## 6          130   77      0.335    0.391   0.725
```

In addition to runs scored, there are seven traditionally used variables in the data set: at-bats, hits, home runs, batting average, strikeouts, stolen bases, and wins. There are also three newer variables: on-base percentage, slugging percentage, and on-base plus slugging. For the first portion of the analysis we'll consider the seven traditional variables. At the end of the tutorial, you'll work with the newer variables on your own.

1. What type of plot would you use to display the relationship between `runs` and one of the other numerical variables? Plot this relationship using the variable `at_bats` as the predictor. Does the relationship look linear? If you knew a team's `at_bats`, would you be comfortable using a linear model to predict the number of runs?

If the relationship looks linear, we can quantify the strength of the relationship with the correlation coefficient (see *AR*, bottom of p. 85, equation 5.4).

```
cor(mlb11$runs, mlb11$at_bats)
```

```
## [1] 0.610627
```

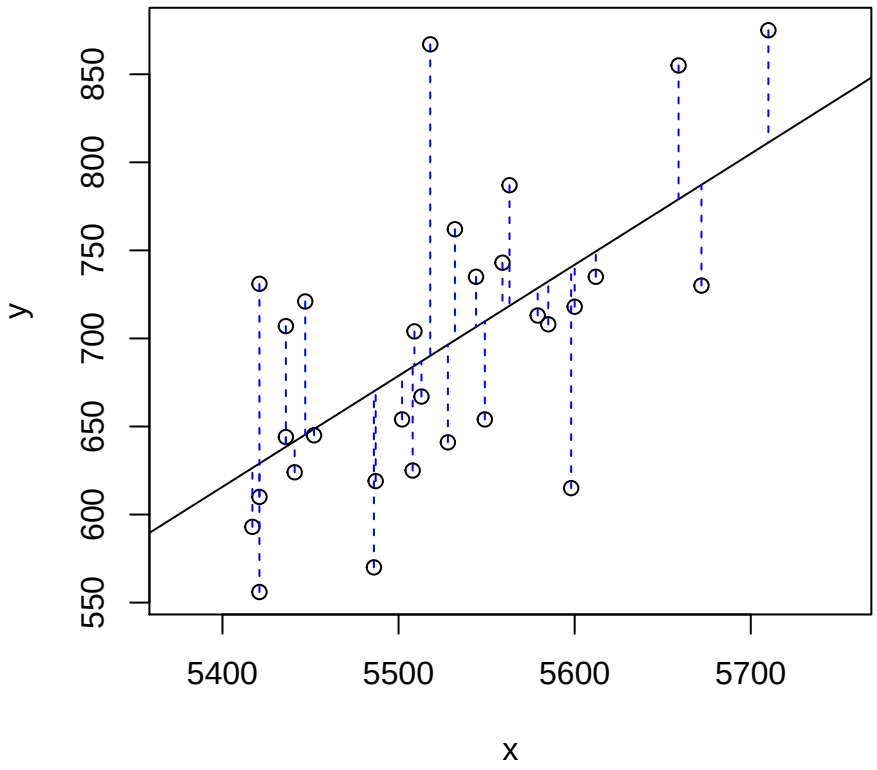
Sum of squared residuals

Think back to the way that we described the distribution of a single variable. 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 `runs` and `at_bats` above.

2. Looking at your plot from the previous exercise, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.

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. **Note: you need to read this function into R using the `source` command. I uploaded a copy of it to my website; the function was originally written by the authors of the OpenIntro book. This function is interactive: it will prompt you to click two points in the RStudio plot window.**

```
source("http://www.jkarreth.net/files/plot_ss.R")
plot_ss(x = mlb11$at_bats, y = mlb11$runs)
```



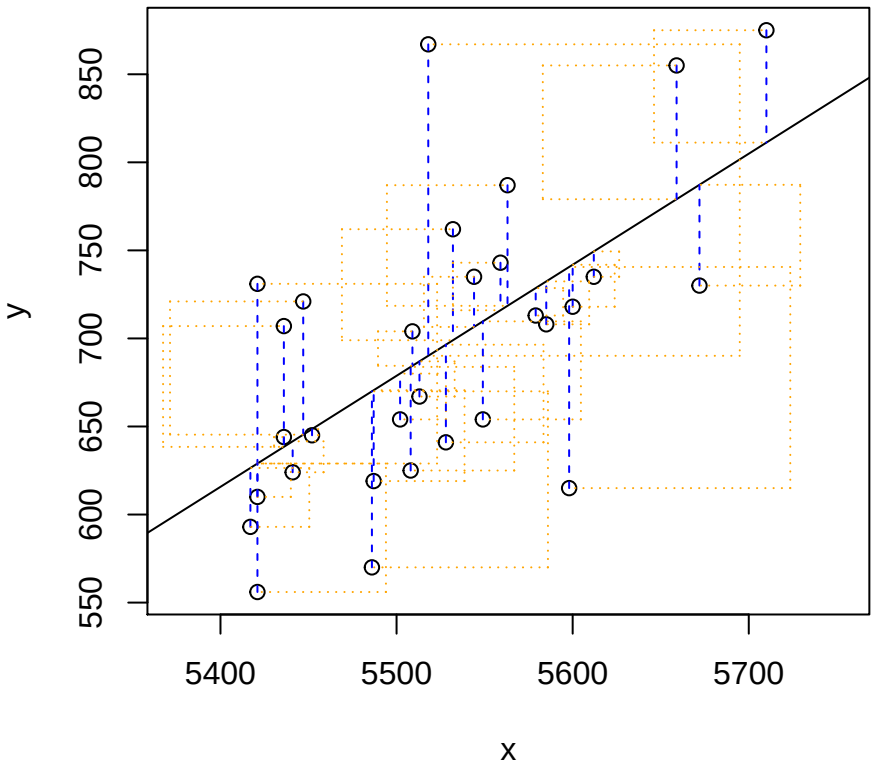

```
## Click two points to make a line.

## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##  -2789.2429       0.6305
##
## Sum of Squares:  123721.9
```

After running this command, you'll be prompted to click two points on the plot to 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 30 residuals, one for each of the 30 observations. Recall that the residuals are the difference between the observed values and the values predicted by the line (*AR*, p. 78):

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

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`.

```
plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)
```




```
## Click two points to make a line.

## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##  -2789.2429       0.6305
##
## Sum of Squares:  123721.9
```

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.

3. 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 colleagues?

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).

```
m1 <- lm(runs ~ at_bats, data = mlb11)
```

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 `runs` as a function of `at_bats`. The second argument specifies that R should look in the `mlb11` data frame to find the `runs` and `at_bats` variables.

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

```
summary(m1)
```

```
##
## Call:
## lm(formula = runs ~ at_bats, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125.58  -47.05  -16.59   54.40  176.87
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2789.2429   853.6957  -3.267 0.002871 **
## at_bats         0.6305     0.1545   4.080 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.47 on 28 degrees of freedom
## Multiple R-squared:  0.3729, Adjusted R-squared:  0.3505
## F-statistic: 16.65 on 1 and 28 DF,  p-value: 0.0003388
```

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 `at_bats`. With this table, we can write down the least squares regression line for the linear model:

$$\hat{y} = -2789.2429 + 0.6305 * \text{at\_bats}$$

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 (AR, bottom of p. 83. For this model, 37.3% of the variability in runs is explained by at-bats.

4. Fit a new model that uses `homeruns` to predict `runs`. Using the estimates from the R output, write the equation of the regression line. What does the slope tell us in the context of the relationship between success of a team and its home runs?

Summarizing the results of a regression model

For your applied work, you will need to present the results of a regression model in the form of a table or a graph. R offers convenient functions for both. Here, we briefly look at the `screenreg` function to create a nicely formatted table in the R console, and then its relatives, `texreg` for a a PDF file and `htmlreg` function, which you can use to create a table for a Word document.

First, load the texreg package.

```
library(texreg)
```

Then, simply feed the model object, `m1`, to the `texreg` function:

```
screenreg(m1)
```

```
##
## ========================
##              Model 1
## ------------------------
## (Intercept)  -2789.24 **
##               (853.70)
## at_bats          0.63 ***
##                 (0.15)
## ------------------------
## R^2              0.37
## Adj. R^2         0.35
## Num. obs.       30
## ========================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

We can customize this table in many ways. For your work, you at least want to simplify the number of stars that are printed (if any) and provide custom coefficient names. The `stars` and `custom.coef.names` arguments do this job:

```
screenreg(m1, stars = 0.05, custom.coef.names = c("Intercept", "At-bats"))
```

```
##
## =====================
##             Model 1
## ---------------------
## Intercept  -2789.24 *
##             (853.70)
## At-bats        0.63 *
##               (0.15)
## ---------------------
## R^2            0.37
## Adj. R^2       0.35
## Num. obs.     30
## =====================
## * p < 0.05
```

To create a table in your R data analysis notebook in PDF format, simply use `texreg` and make sure to add `results = "asis"` to your R Markdown code chunk parameters:

```
texreg(m1, stars = 0.05, custom.coef.names = c("Intercept", "At-bats"),
  caption = "Regression table", caption.above = TRUE)
```

And if you'd like to export a regression table to Word, use `htmlreg`, save the file in your working directory, and open it with Word or Libre Office, and then copy and paste the table in your manuscript. Use the `file =` argument to specify the name of the file. R will automatically save it to your current working directory.

```
htmlreg(m1, stars = 0.05, custom.coef.names = c("Intercept", "At-bats"),
  caption = "Regression table", caption.above = TRUE, file = "mod1.doc")
```

Table 1: Regression table

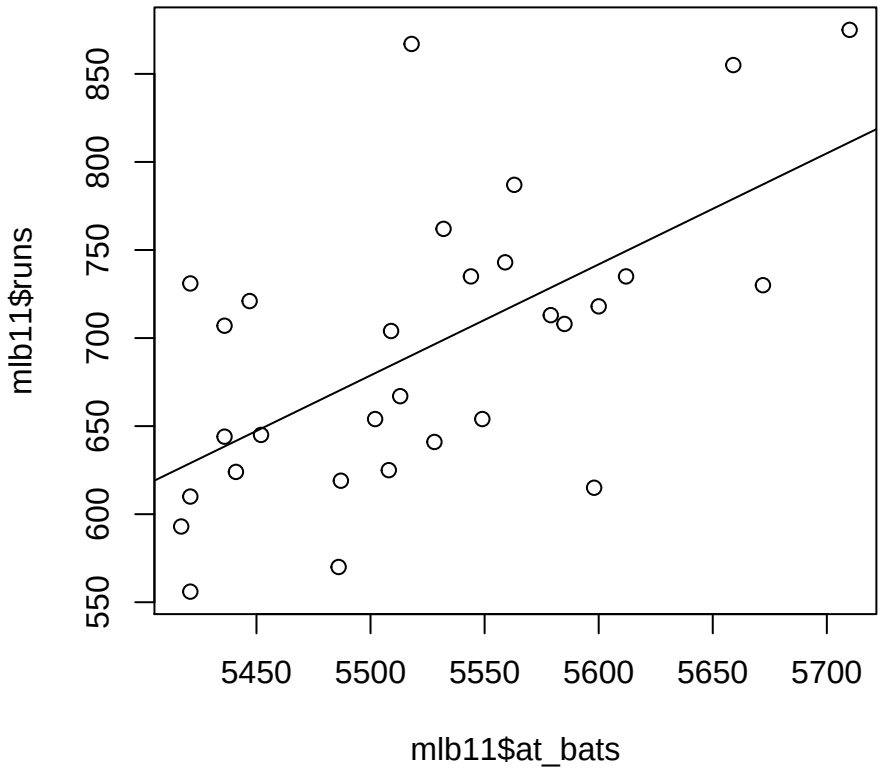
	Model 1
Intercept	−2789.24*
	(853.70)
At-bats	0.63*
	(0.15)
R^2	0.37
Adj. R^2	0.35
Num. obs.	30

*$p < 0.05$

Prediction and prediction errors

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

```
plot(mlb11$runs ~ mlb11$at_bats)
abline(m1)
```



The function `abline` plots a line based on its slope and intercept. Here, we used a shortcut by providing the model `m1`, 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.

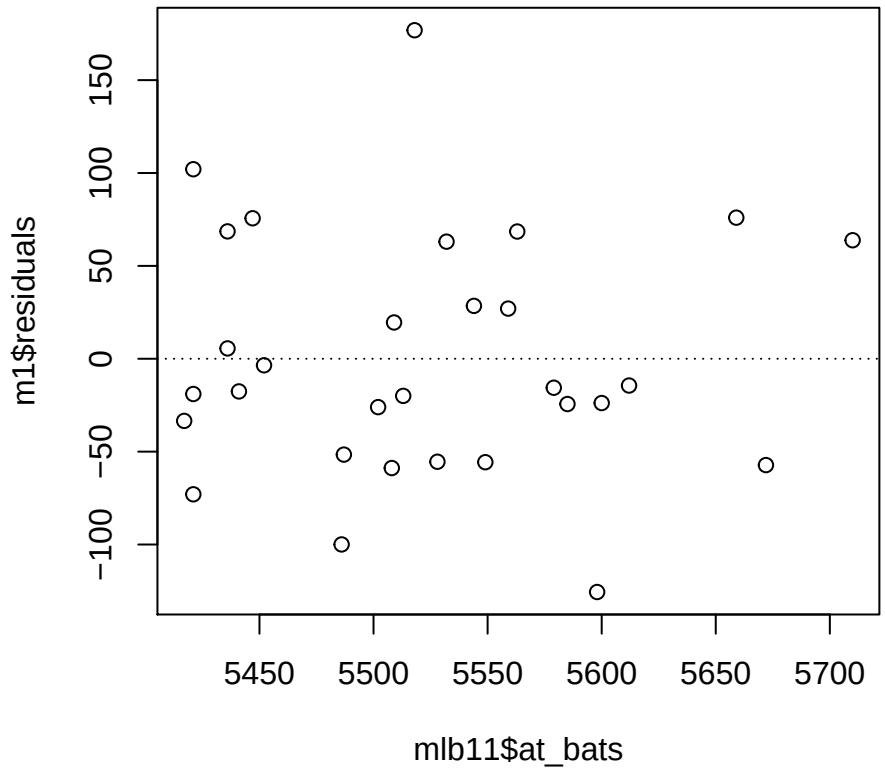
5. If a team manager saw the least squares regression line and not the actual data, how many runs would he or she predict for a team with 5,578 at-bats? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction?

Model diagnostics

To assess whether the linear model is reliable, we need to check for the assumptions (A1) linearity, (A2) constant error variance, (3) normality of the errors, (A4) independence of observations, and (A5) no correlation between x_i and ε_i. (A4) and (A5) cannot be directly tested, and we will discuss them further in class.

- **Linearity (A1)**: You already checked if the relationship between runs and at-bats is linear using a scatterplot. We should also verify this condition with a plot of the residuals vs. at-bats. Recall that any code following a # is intended to be a comment that helps understand the code but is ignored by R.

```
plot(m1$residuals ~ mlb11$at_bats)
abline(h = 0, lty = 3)  # adds a horizontal dashed line at y = 0
```



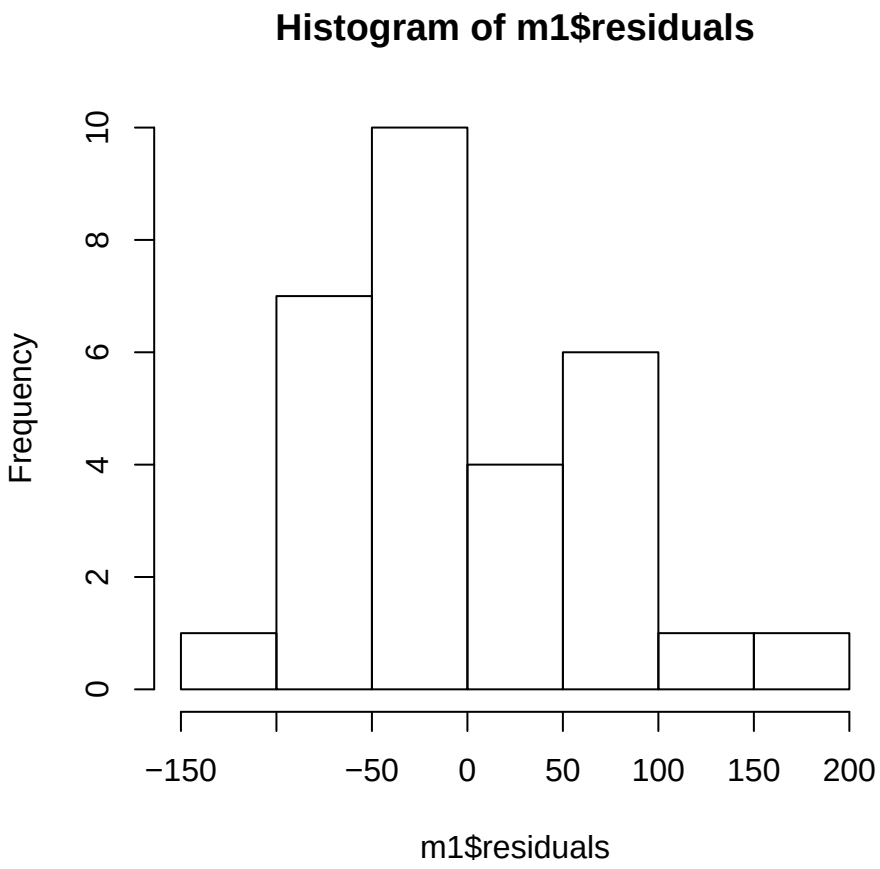
6. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between runs and at-bats?

- **Constant error variance (A2)**: You can use the residuals plot to check whether the variance of the errors is constant or changes across the range of the predictor.

7. Based on the plot in you just created, does the constant error variance condition appear to be met?

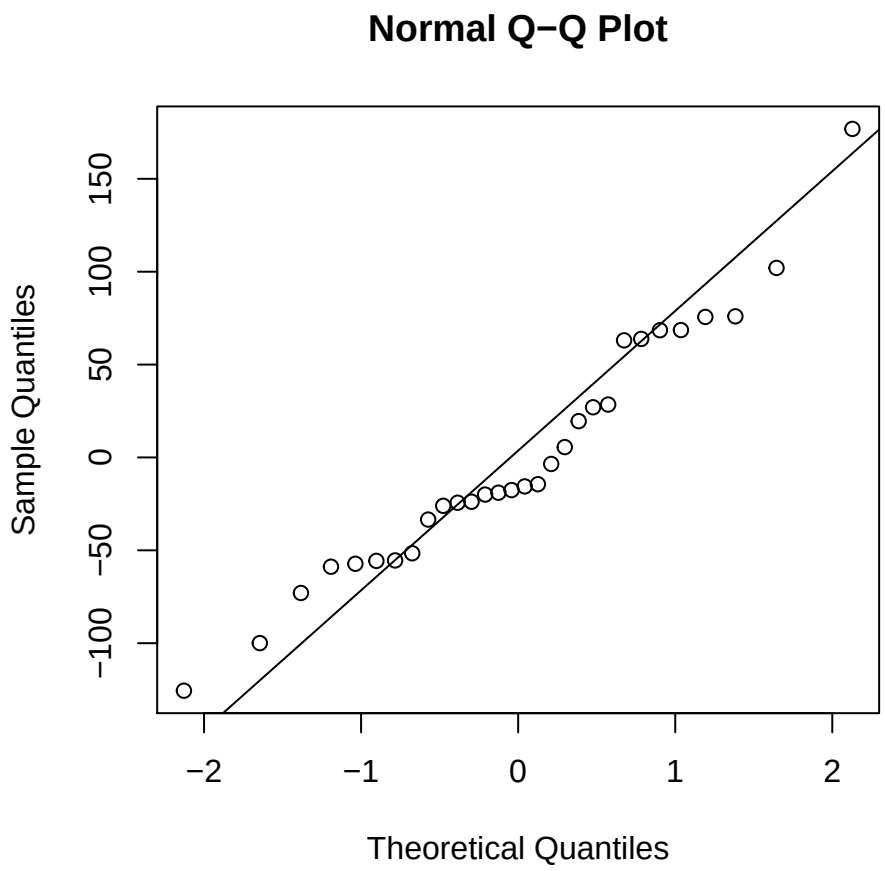
- **Normality of the errors (A3)**: To check this condition, we can look at a histogram

```
hist(m1$residuals)
```



or a normal probability plot of the residuals.

```
qqnorm(m1$residuals)
qqline(m1$residuals)  # adds diagonal line to the normal prob plot
```

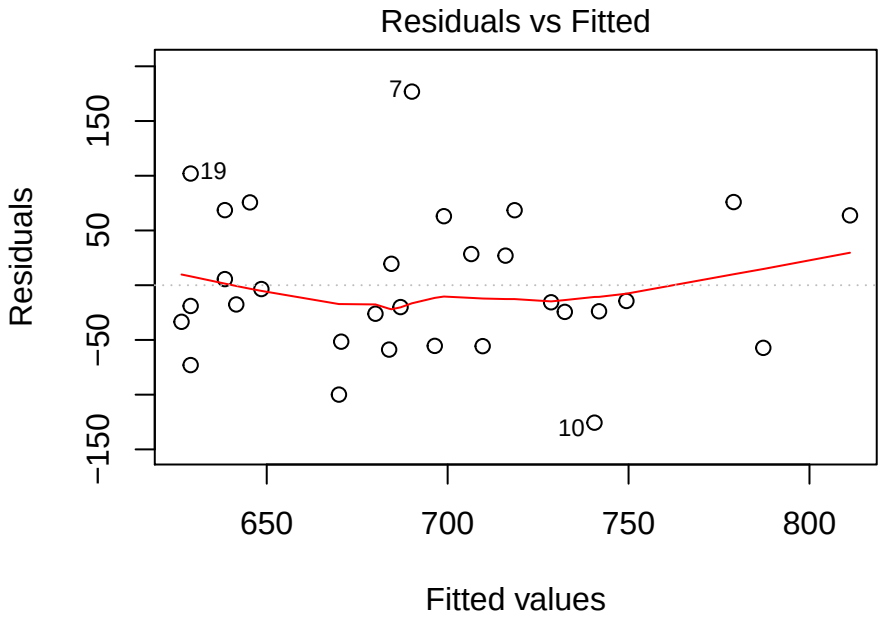


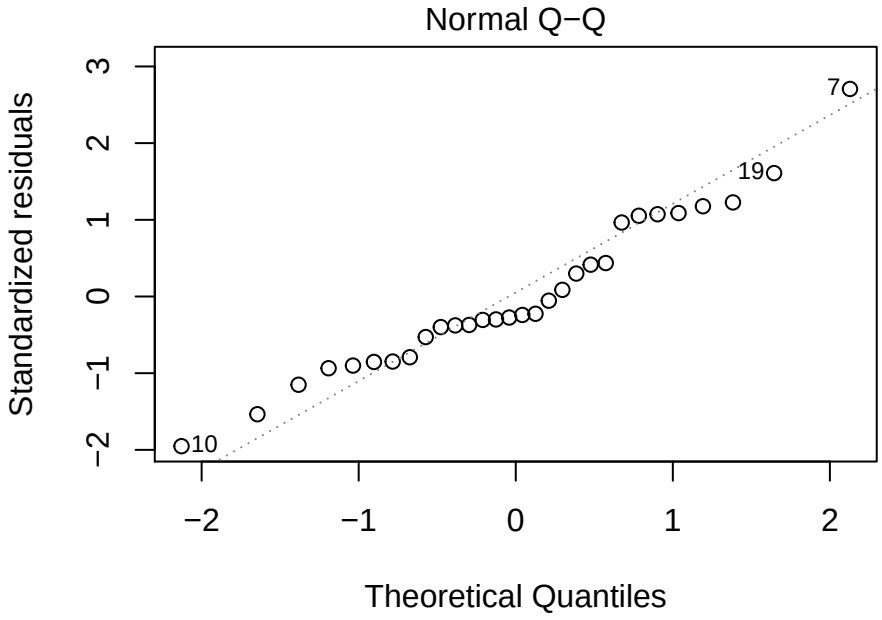
8. Based on the histogram and the normal probability plot, does the nearly normal residuals condition appear to be met?

Diagnostic plots in R

R also offers a convenient way of producing diagnostic plots to assess these assumptions. After creating a linear regression model object, simply `plot()` that object. For now, I recommend using this command in the following way. You can read up more on the options under `?plot.lm`.

```
par(mfrow = c(2, 1))
plot(m1, which = c(1, 2))
```






```
par(mfrow = c(1, 1))
```

On Your Own

- Choose another traditional variable from `mlb11` that you think might be a good predictor of `runs`. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship?

- How does this relationship compare to the relationship between `runs` and `at_bats`? Use the R^2 values from the two model summaries to compare. Does your variable seem to predict `runs` better than `at_bats`? How can you tell?
- Now that you can summarize the linear relationship between two variables, investigate the relationships between `runs` and each of the other five traditional variables. Which variable best predicts `runs`? Support your conclusion using the graphical and numerical methods we've discussed (for the sake of conciseness, only include output for the best variable, not all five).
- Now examine the three newer variables. These are the statistics used by the author of *Moneyball* to predict a teams success. In general, are they more or less effective at predicting runs that the old variables? Explain using appropriate graphical and numerical evidence. Of all ten variables we've analyzed, which seems to be the best predictor of `runs`? Using the limited (or not so limited) information you know about these baseball statistics, does your result make sense?
- Check the model diagnostics for the regression model with the variable you decided was the best predictor for runs.