Simple Linear Regression in RStudio

Before beginning to read this tutorial, be sure to have read Chapter 3 in the Lilja text.

The Setup

An antique dealer who specializes in antique clocks would like to predict how much some of the clocks in her inventory will sell for at auction. She collects recent sales data from 32 grandfater clocks and would like to use the data to create a model for determining a selling price. The data can be found in `auction.csv`[1].

Let's try to help the dealer create a model.

Tasks

a. Set up a scatter plot: Review the distribution of data and pay close attention to see if the data may be linear.
b. Generate the model: What are the regression coefficients (m and b)? Write the model you found.
c. Evaluate the regression for fit (Rsquare, ANOVA, F-statistic): Is the model a good fit for the data?
d. Check: Make sure all the assumptions for linear regression are met.
e. Predict: If the equation is a good fit for the data, what is your prediction for the selling price of a 160 year old clock?

The process

As always, I'll begin by loading our standard helper library

```
library(tidyverse)
```

Now let's read the data from the given data file. Use the Files tab, navigate to the where you downloaded the file and click it to import. You can look at the data in the tab that pops up. I'll use `head` here to display the first few rows. I'll also `attach` the data set for convenience.

```
auction <- read_csv("auction.csv")
```

```
Rows: 32 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (3): Age, Bidders, Price

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.
```

```
head(auction)
```

```
# A tibble: 6 x 3
    Age Bidders Price
  <dbl>   <dbl> <dbl>
1   127      13  1235
2   115      12  1080
```

[1] Mendenhall, W, and Sincich, TL (1993). Selling Price of Antique Grandfather Clocks [Data File]. Accessed at http://www.statsci.org/data/general/auction.txt

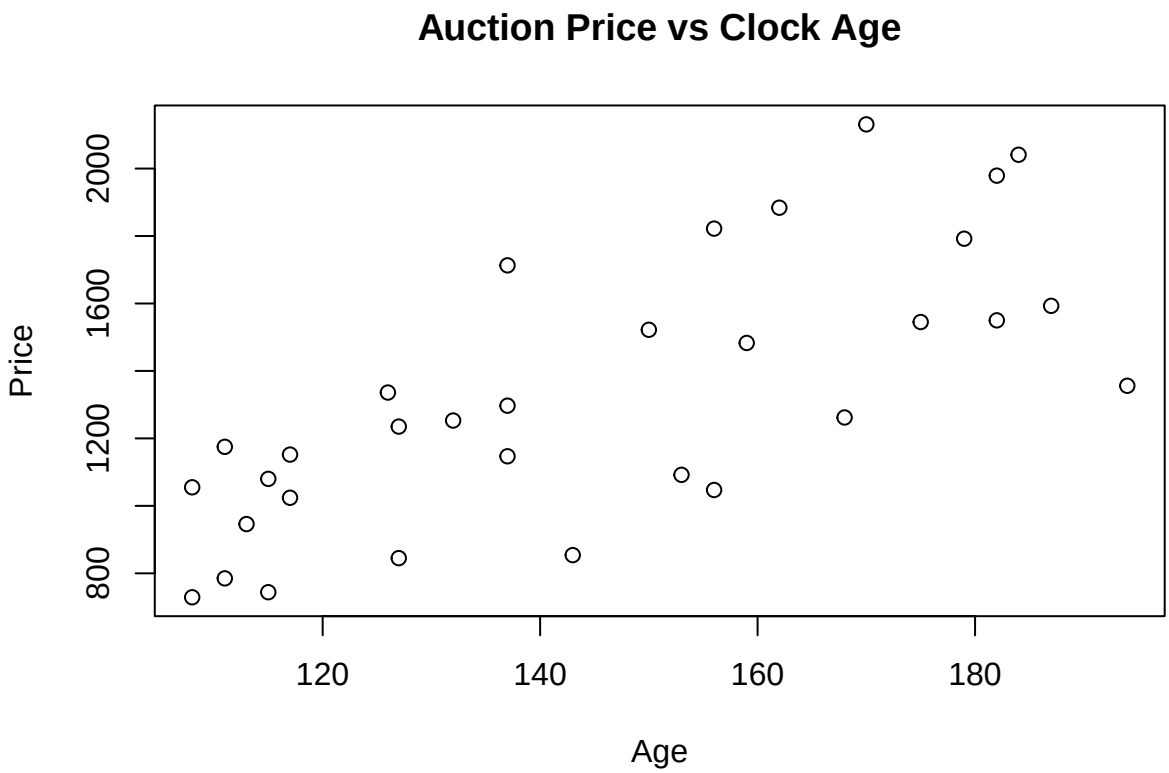
```
3   127       7   845
4   150       9  1522
5   156       6  1047
6   182      11  1979
```

This looks like every row is one day, and each column is one variable. That's the format we want. The variables we want in particular are called `Age` and `Price`.

Make a Scatterplot

A good first step is to look at a scatter plot and decide if the relationship looks in any way linear. Note the nice title added to the plot:

```
plot(Price ~ Age, main="Auction Price vs Clock Age", data=auction)
```



The plot is somewhat linear with no obvious curved pattern, so it makes sense to try and use a linear model.

Generate the model

Let's move on to generating the linear regression model. The first line will generate a linear model with `Price` as the dependent variable (y), and `Age` as the predictor variable (x). The second line prints a summary of the model.

```
auction_lm <- lm(Price ~ Age, data=auction)
summary(auction_lm)
```

```

Call:
lm(formula = Price ~ Age, data = auction)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-485.29 -192.66   30.75  157.21  541.21

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -191.66     263.89  -0.726    0.473
Age            10.48       1.79   5.854  2.1e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 273 on 30 degrees of freedom
Multiple R-squared:  0.5332,    Adjusted R-squared:  0.5177
F-statistic: 34.27 on 1 and 30 DF,  p-value: 2.096e-06
```

The last F-statistic line shows a very small p-value, so we have very strong evidence that the coefficient on `Invoices_Processed` should not be 0. In other words, we have strong evidence there is a linear relationship!

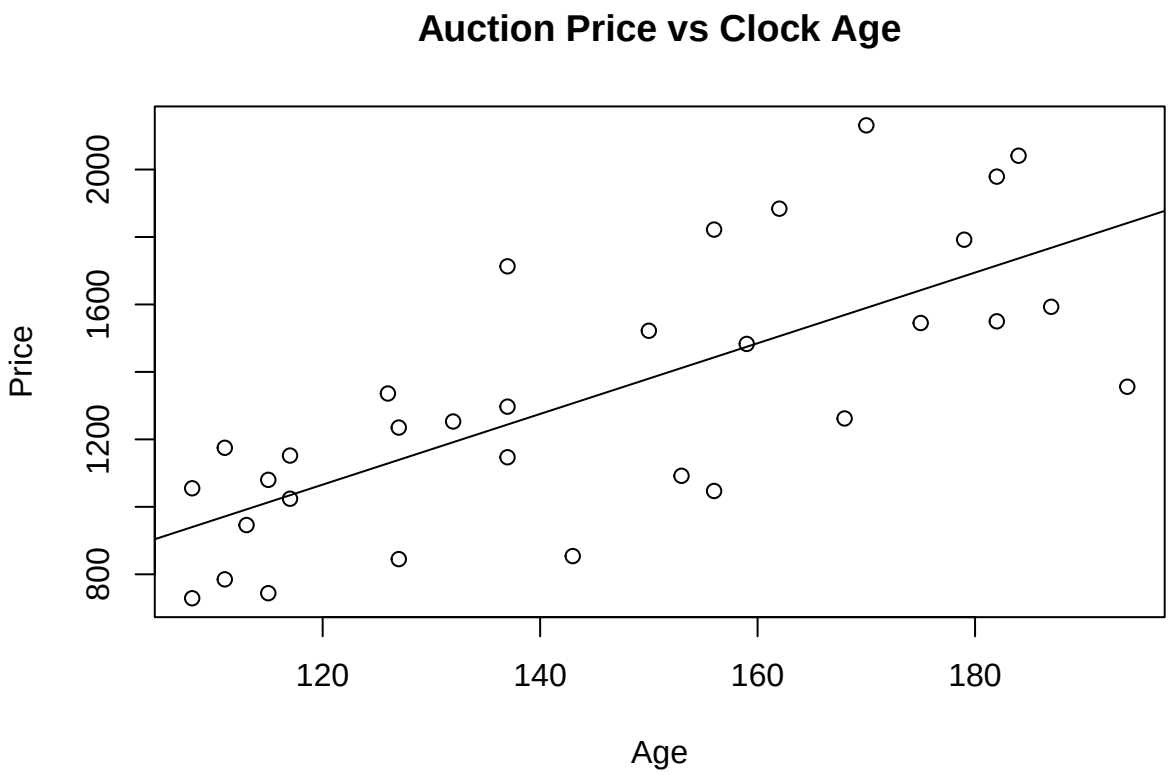
The second to last line gives us an R^2 of 0.5332 indicating that approximately 53% of the variation in time is explained by the number of invoices to be processed. This is a moderately strong relationship.

Those last two paragraphs taken together tell me we have very strong evidence the relationship exists, but it is only a moderate relationship. We will need to tell the antique dealer that there are likely other factors that influence the amount of time needed. (Hopefully this makes sense to you. Likely condition, or rarity would be useful variables, though perhaps harder to quantify.)

The model is `Price = -191.66 + 10.48 * Age`.

If you'd like a plot with the 'best fit' line you can execute these next two lines in the same code block.

```
plot(Price ~ Age, main="Auction Price vs Clock Age", data=auction)
abline(auction_lm)
```



Checking Assumptions

Before we confidently use this model however, let's check all our requirements:

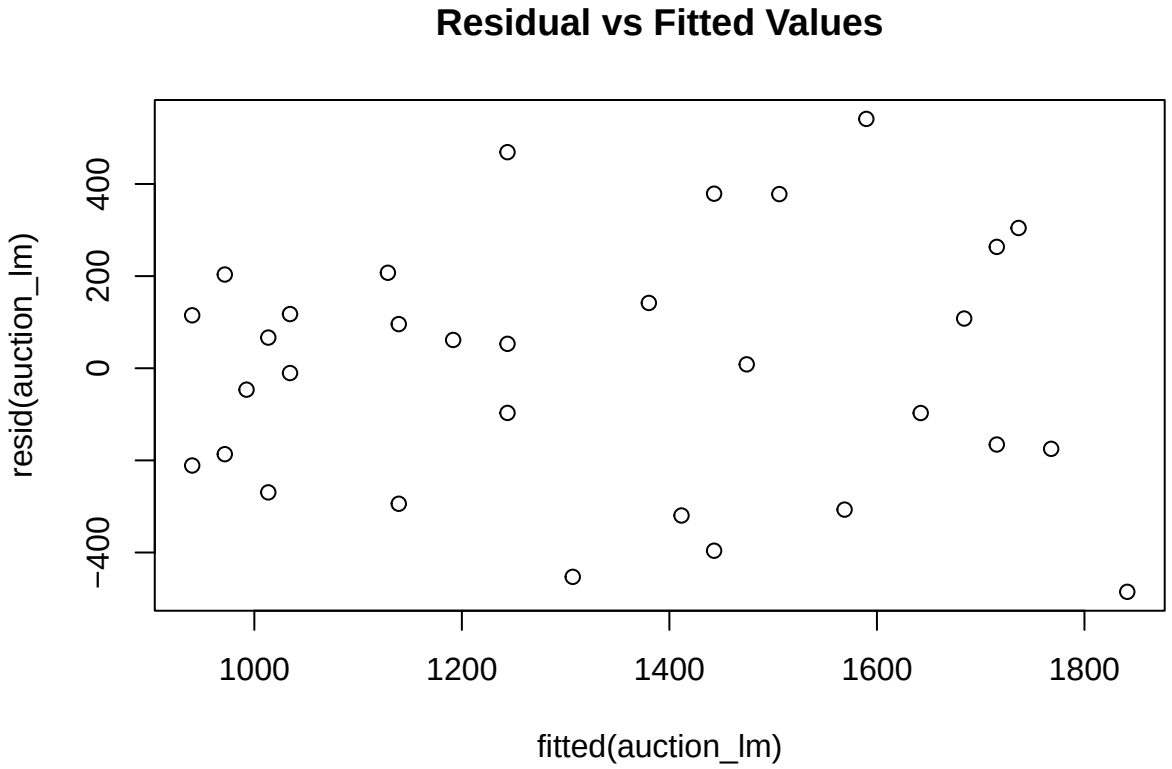
1. The relationship is linear
2. The errors are independent
3. The errors at each predictor value are normally distributed
4. The errors have equal variance across predictors (homoscedasticity)

We'll check these by analyzing the residuals from the model.

Residual vs Fitted Plot for #1 and #4

We will do a residual vs fitted values plot. Remember, a residual is the error when trying to use the model to predict a particular value. We would like to see a scatter plot here with no systematic pattern to the plot (to verify #1) and we would like to see a consistent spread (to verify #4).

```
plot(fitted(auction_lm), resid(auction_lm), main="Residual vs Fitted Values")
```



Looking across the plot we see the average is close to 0, and we don't see any excessive outlying points. A clear pattern is not evident, so #1 is satisfied. The vertical spread of the data is perhaps a little less on the left hand side, but not enough to worry about, so that supports #4. This is good!

If we had seen a clear pattern to the data that would indicate the relationship is likely not linear and we would need to try other methods.

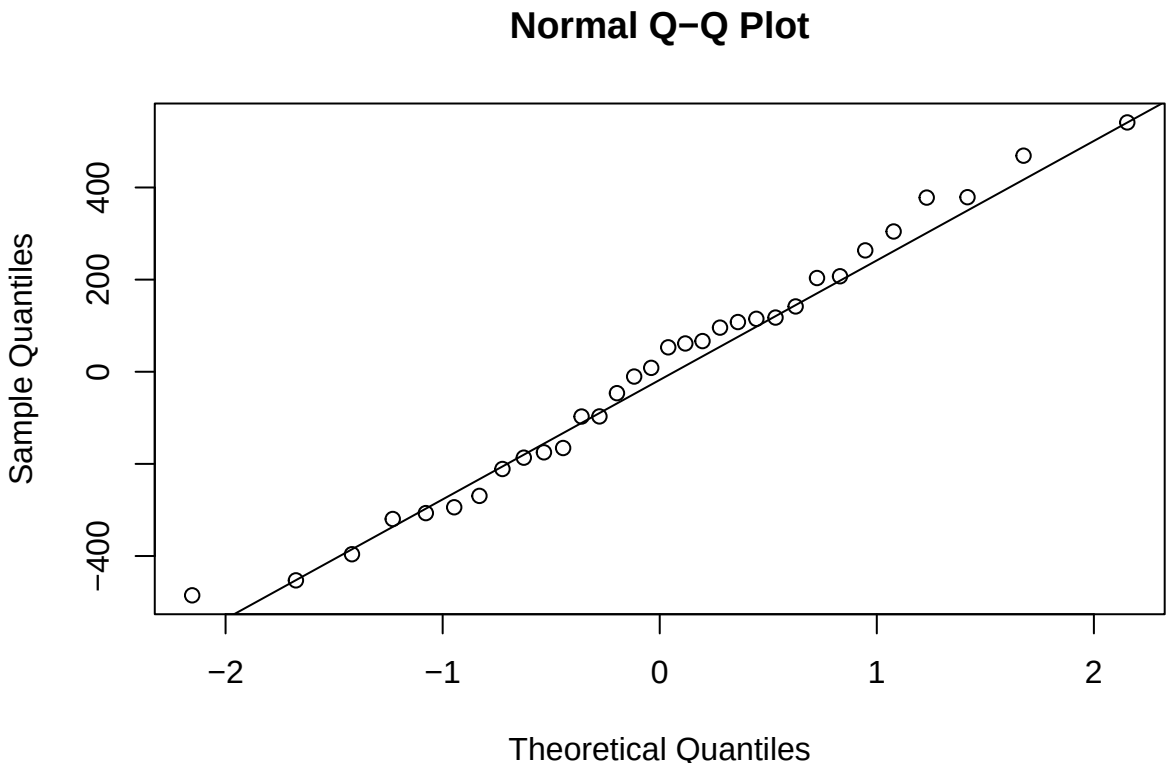
Checking #2

We need to ask ourselves if the data is reasonabley independent. Presumably the data come from sales of unrelated clocks, so this assumption is met.

Normal Quantile Plot for #3

To check that the residuals are normally distributed (#3) we can use a normal quantile plot. The commands below produce the normal quantile plot for the residuals along with a reference line. We hope the points are close to the reference line.

```
qqnorm(resid(auction_lm))
qqline(resid(auction_lm))
```



This normal probability plot is quite good, so #3 is satisfied.

Did you notice there is one point is the lower left that is away from the line? Why wasn't I worried about that? I'm sure you're wondering if there is a way to more precisely quantify when a plot meets these requirements. In fact there is!

We need to install a new library called `car`. The first time you use it you will need to install the package with the command `install.packages("car")`. You only need to run this command once.

Once that is installed, we load it with the `library` command, then create a quantile plot with confidence bands. If all points land within the dashed line then we can say we have evidence that the data is from a normally distributed population.

We use the `qqPlot` command (note the capital P) on the residuals.

```
library(car)
```

```
Loading required package: carData
```

```
Attaching package: 'car'
```

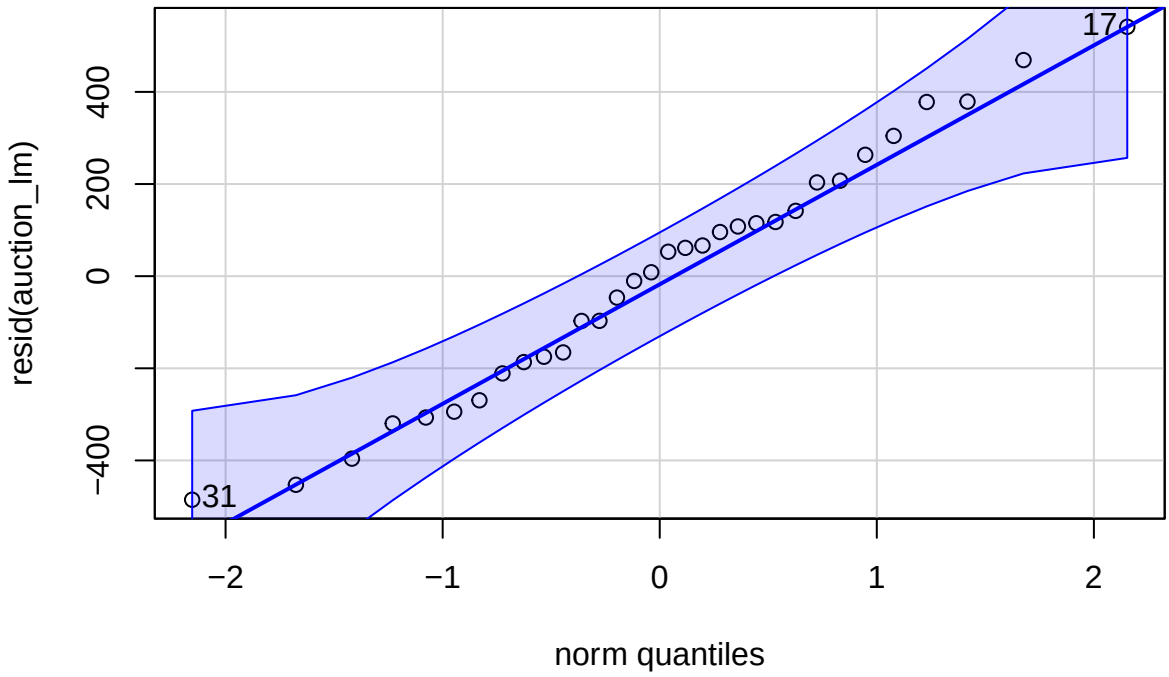
```
The following object is masked from 'package:dplyr':

    recode
```

```
The following object is masked from 'package:purrr':

    some
```

```
qqPlot(resid(auction_lm))
```




```
[1] 17 31
```

Since all the points land within the bands we can say this assumption is met.

Going forward I suggest you always load the `car` package and use the `qqPlot` command when checking for normality. The dashed lines confidence bands are very useful.

Our conclusion

Overall we can use this model, but should expect some uncertainty with our predictions since our R^2 indicates this one predictor does not perfectly explain the variation in price.

Make our prediction

We were asked to predict the selling price of a 160 year old clock, so we need to see what our model says when `Age` = 160. Recall the model was `Price = -191.66 + 10.48 * Age`. The code below computes our answer.

```
-191.66 + 10.48 * 160
```

```
[1] 1485.14
```

So our best estimate is about $1,485.

In relaying this estimate we should note that we expect a lot of variance in that number. Remember, R^2 was only about 0.5, so only 50% of the variation in price needed is explained by the age. Presumably there are other factors at play and if a better model is needed we should identify what else would affect the price.

A final note

What you are seeing here is a very technical document. You would likely not give something like this to your manager or client. Instead you would write something more like the last paragraph above.

HOWEVER it is vital that you do all this technical work on your own. You cannot give an honest interpretation of how reliable the model will be without checking the assumptions through residual analysis, thinking about the overall significance of the model, and evaluating R^2.

So this type of technical evaluation is critical, and mandatory, even if it doesn't end up in your final report.

Brief code summary

```
library(tidyverse)

auction <- read_csv("auction.csv") # load the data
plot(Price ~ Age, main="Auction Price vs Clock Age", data=auction) # make sure the scatterplot looks so

auction_lm <- lm(Price ~ Age, data=auction)
summary(auction_lm)

plot(fitted(auction_lm), resid(auction_lm), main="Residual vs Fitted Values") # check for homoscedastic

library(car)
qqPlot(resid(auction_lm)) # check for normality in residuals
```