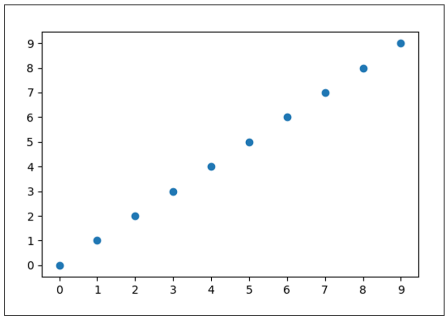
15.7.1 The Correlation Conundrum

Jeremy has finally started to make the connections between his programming experience with some statistical concepts. But this is only the beginning; comparing and contrasting data is only one statistical concept. Another big component to his new job will be to identify patterns in data and generate predictive models. Jeremy has a little experience in generating trendlines in plots, but he has no way to quantify how well these trend lines will perform when it comes time for decision making. Jeremy realizes that he must go back and learn more statistical tests that will help him quantify the patterns and models in his data.

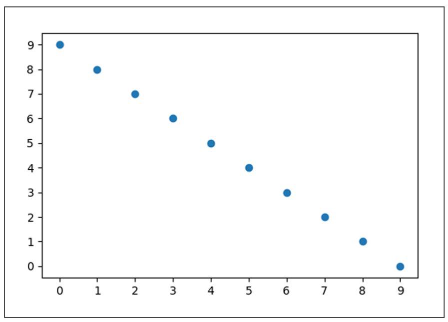
In data analytics, we'll often ask the question "is there any relationship between variable A and variable B?" This concept is known in statistics as correlation. **Correlation analysis** is a statistical technique that identifies how strongly (or weakly) two variables are related.

Correlation is quantified by calculating a **correlation coefficient**, and the most common correlation coefficient is the Pearson correlation coefficient. The **Pearson correlation coefficient** is denoted as "r" in mathematics and is used to quantify a linear relationship between two numeric variables. The Pearson correlation coefficient ranges between -1 and 1, depending on the direction of the linear relationship.

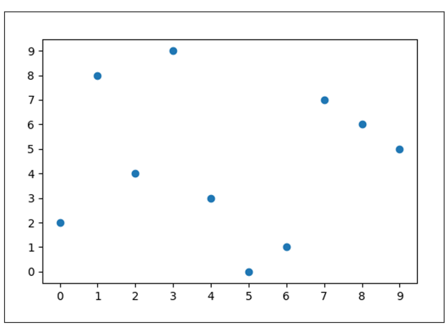
The following image is an example of an **ideal positive correlation** where r = 1. When two variables are positively correlated, they move in the same direction. In other words, when the variable on the x-axis increases, the variable on the y-axis increases as well:



The following image is an example of an **ideal negative correlation** where r = -1. When two variables are negatively correlated, they move in opposite directions. In other words, when the variable on the x-axis increases, the variable on the y-axis decreases.



The following image is an example of two variables with **no correlation** where r ≈ 0. When two variables are not correlated, their values are completely independent between one another.

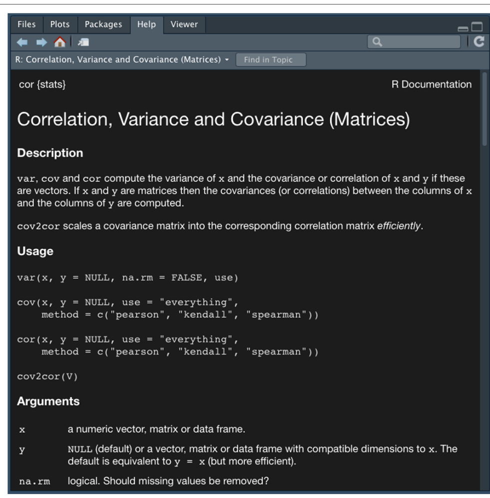


For real-world data, it can be very difficult to determine if two variables are correlated, so we must use the Pearson correlation coefficient to calculate the correlation strength. Refer to the table below.

Absolute Value of r	Strength of Correlation
r < 0.3	None or very weak
0.3 ≤ r < 0.5	Weak
0.5 ≤ r < 0.7	Moderate
r ≥ 0.7	Strong

In R, we can use our `geom_point()` plotting function combined with the `cor()` function to quantify the correlation between variables. Type the following code into the R console to look at the `cor()` documentation in the Help pane:

```
>?cor()
```

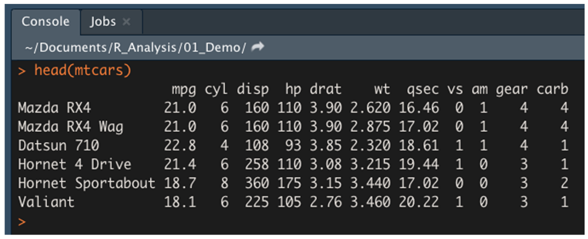


To use the `cor()` function to perform a correlation analysis between two numeric variables, we need to provide the following arguments:

- **x** is the first variable, which would be plotted on the x-axis.
- **y** is the second variable, which would be plotted on the y-axis.

As long as we are using two numeric variables, there are no other assumptions regarding our input data. To practice calculating the Pearson correlation coefficient, we'll use the mtcars dataset. Type the following in the R console:

```
> head(mtcars)
```

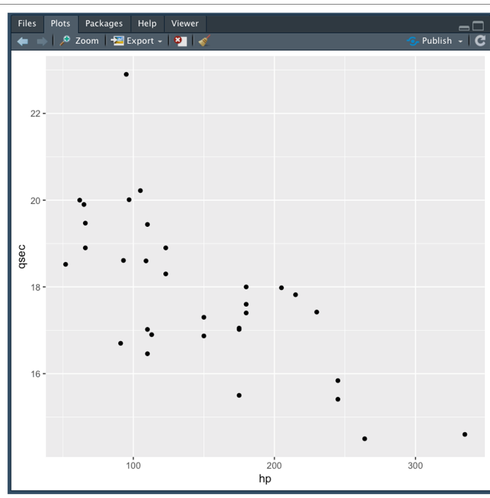


```
Console   Jobs
~/Documents/R_Analysis/01_Demo/
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
>
```

In the mtcars dataset, there are a number of numeric columns that we can use to test for correlation such as `mpg`, `disp`, `hp`, `drat`, `wt`, and `qsec`. For our example, we'll test whether or not horsepower (`hp`) is correlated with quarter-mile race time (`qsec`).

First, let's plot our two variables using the `geom_point()` function as follows:

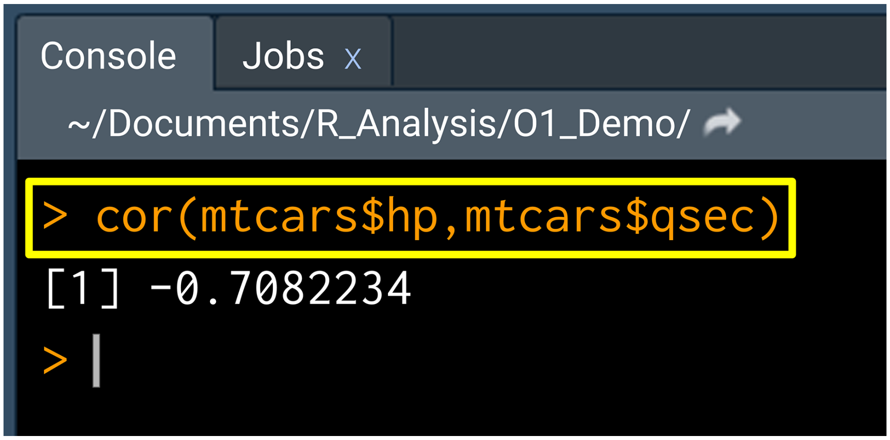
```
> plt <- ggplot(mtcars,aes(x=hp,y=qsec)) #import dataset into ggplot2
> plt + geom_point() #create scatter plot
```



Looking at our plot, it appears that the quarter-mile time is negatively correlated with horsepower. In other words, as vehicle horsepower increases, vehicle quarter-mile time decreases.

Next, we'll use our `cor()` function to quantify the strength of the correlation between our two variables:

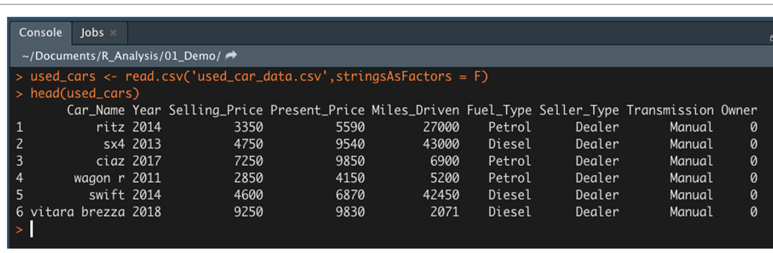
```
> cor(mtcars$hp,mtcars$qsec) #calculate correlation coefficient
```



From our correlation analysis, we have determined that the r-value between horsepower and quarter-mile time is -0.71, which is a strong negative correlation.

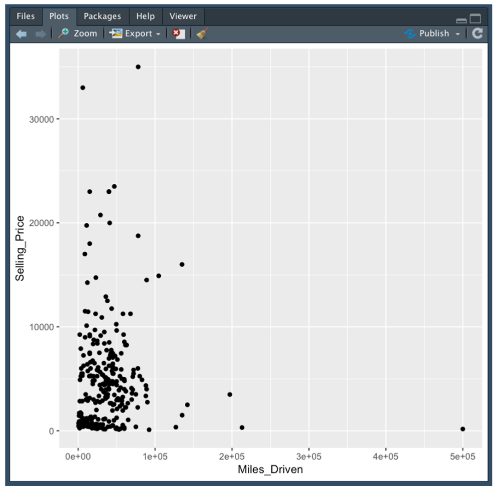
For another example, let's reuse our `used_cars` dataset:

```
> used_cars <- read.csv('used_car_data.csv',stringsAsFactors = F) #read in d
> head(used_cars)
```



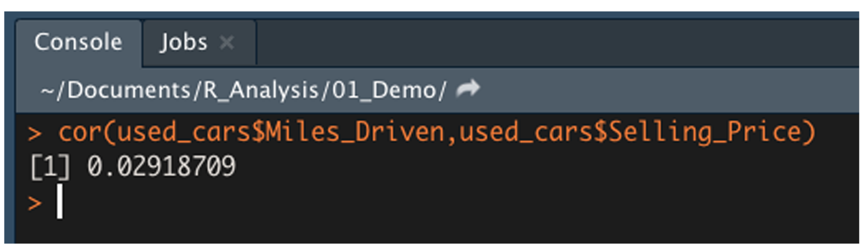
For this example, we'll test whether or not vehicle miles driven and selling price are correlated. Once again, we'll plot our two variables using the `geom_point()` function:

```
> plt <- ggplot(used_cars,aes(x=Miles_Driven,y=Selling_Price)) #import datas
> plt + geom_point() #create a scatter plot
```



Compared to our previous example, our scatter plot did not help us determine whether or not our two variables are correlated. However, let's see what happens if we calculate the Pearson correlation coefficient using the `cor()` function:

```
> cor(used_cars$Miles_Driven,used_cars$Selling_Price) #calculate correlation
```



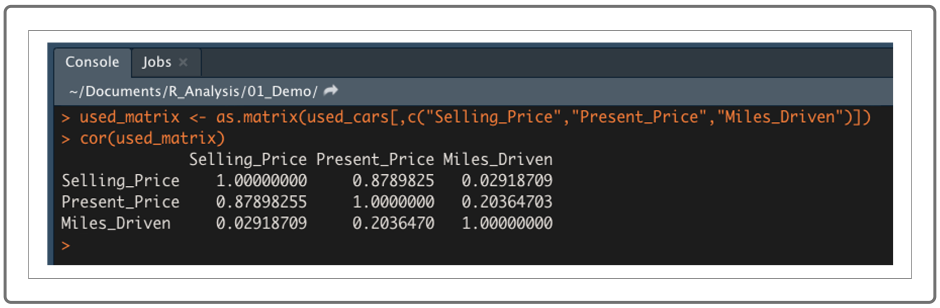
Our calculated r-value is 0.02, which means that there is a negligible correlation between miles driven and selling price in this dataset.

In most cases, we'll use correlation analysis as a means of exploring data and looking for trends. Although we can calculate the correlation of each pair of numerical variables in a dataset, this process can be highly time-consuming.

Instead of computing each pairwise correlation, we can use the `cor()` function to produce a correlation matrix. A **correlation matrix** is a lookup table where the variable names of a data frame are stored as rows and columns, and the intersection of each variable is the corresponding Pearson correlation coefficient. We can use the `cor()` function to produce a correlation matrix by providing a matrix of numeric vectors.

For example, if we want to produce a correlation matrix for our used_cars dataset, we would first need to select our numeric columns from our data frame and convert to a matrix. Then we can provide our numeric matrix to the `cor()` function as follows:

```
> used_matrix <- as.matrix(used_cars[,c("Selling_Price","Present_Price","Mil
> cor(used_matrix)
```



If we look at the correlation matrix using either rows or columns, we can identify pairs of variables with strong correlation (such as selling price versus present price), or no correlation (like our previous example of miles driven versus selling price).

The correlation matrix is a very powerful data exploration tool that allows an analyst to scan large numerical datasets for variables of interest. Once the variables of interest have been identified, the analyst can move on to more rigorous data analysis and hypothesis testing.