Lab 8 - Correlation

Nick Sumpter (Edited by Eddie-Williams Owiredu & Guy Twa)

2022-09-13

Today’s Lab

Today we will learn how to perform a basic correlation analysis using the built-in `cor` function in base R. The output of this function will be a correlation coefficient that will allow us to assess how well two variables are associated with one another. There are 3 types of correlations that we will review: Pearson’s, Spearman’s and Kendall’s.

Loading Packages and Data

We will be introducing one new package today called `ggcorrplot` , which allows us to plot our correlation results more easily. You will need to install this prior to loading it in.

We will be returning to the `iris` dataset from earlier in the semester, but we will remove the “setosa” Species using the `filter` function first. We will be assessing the correlation between Sepal Length and Sepal Width. For your independent practice, you will need to download the `microbiome3.RData` file from Canvas.

```
install.packages("ggcorrplot")
```

```
library(car)
library(pastecs)
library(ggcorrplot)
library(tidyverse)

setwd("/Users/eddie-williamsowiredu/Desktop/grd770_23/Lab8")

theme_set(theme_bw())

load("microbiome3.RData")

source("functions.R")

iris <- iris

iris2 <- iris %>%
  filter(Species != "setosa")
```

Assumptions of Each Correlation Test

A correlation is defined as the association between two variables, where a stronger correlation means the two variables are more closely related. Correlations can be both positive (as one variable increases, so does the other) and negative (as one variable increases, the other decreases).

The 3 types of correlation tests we will address are:

1. Pearson’s (parametric, correlation of raw values)

2. Spearman’s (non-parametric, correlation of rank values)

3. Kendall’s (non-parametric, correlation of rank values)

Essentially, you would conduct a parametric (Pearson’s) correlation if our assumptions are met, or a non-parametric (Spearman’s or Kendall’s) if our assumptions are not met. But what are the assumptions?

The assumptions we need to assess for a Pearson’s correlation test are the following:

1. Normality of the two variables

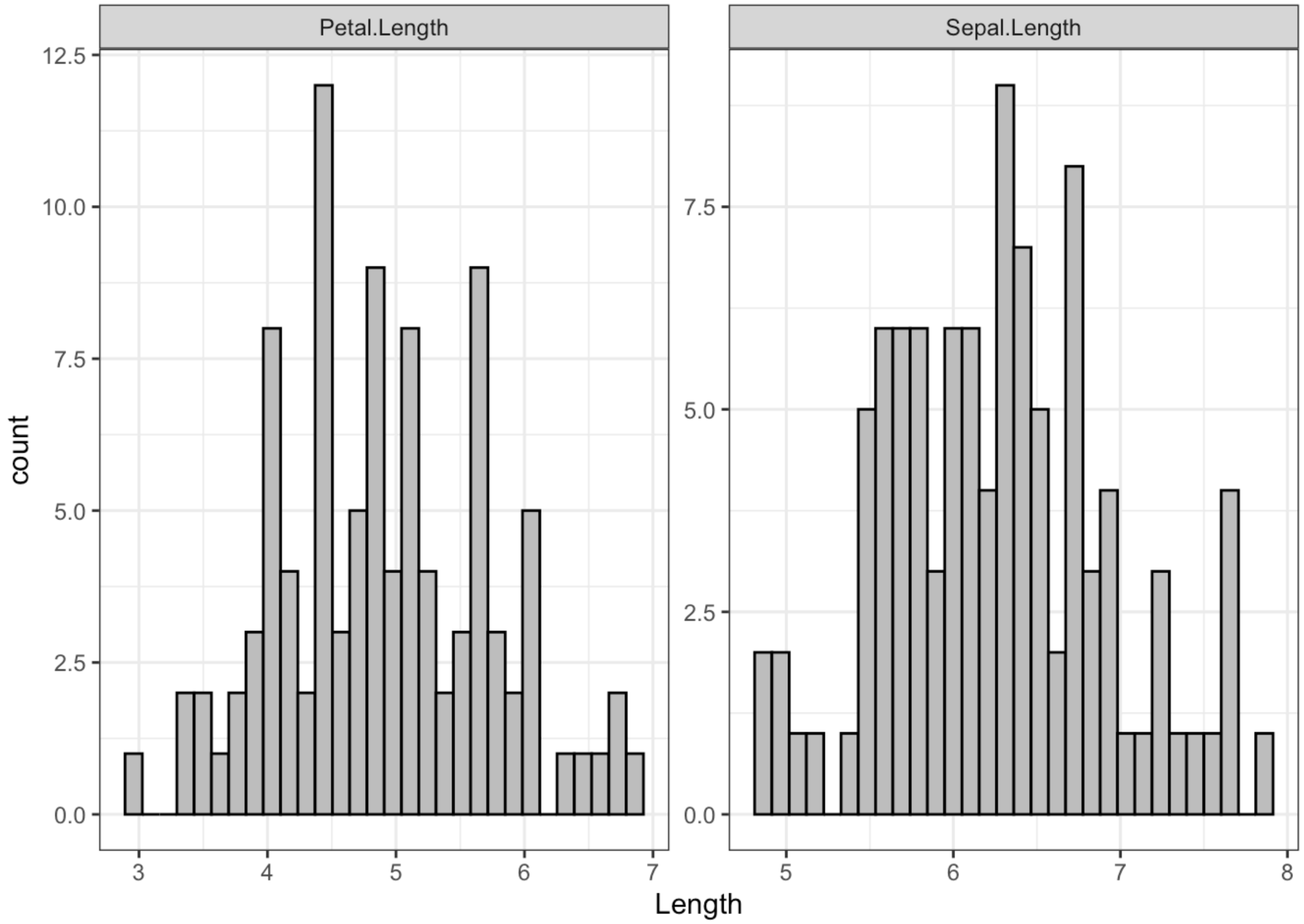
2. Linearity

For a Spearman’s or Kendall’s correlation test, the normality assumption is not required, but a variant of the linearity assumption, monotonicity, is. This assumption essentially says that the relationship can be non-linear as long as it doesn’t flip directions along the length of each variable. For example, if a variable went up at low levels of the other variable, but decreased at higher levels, this assumption would be violated.

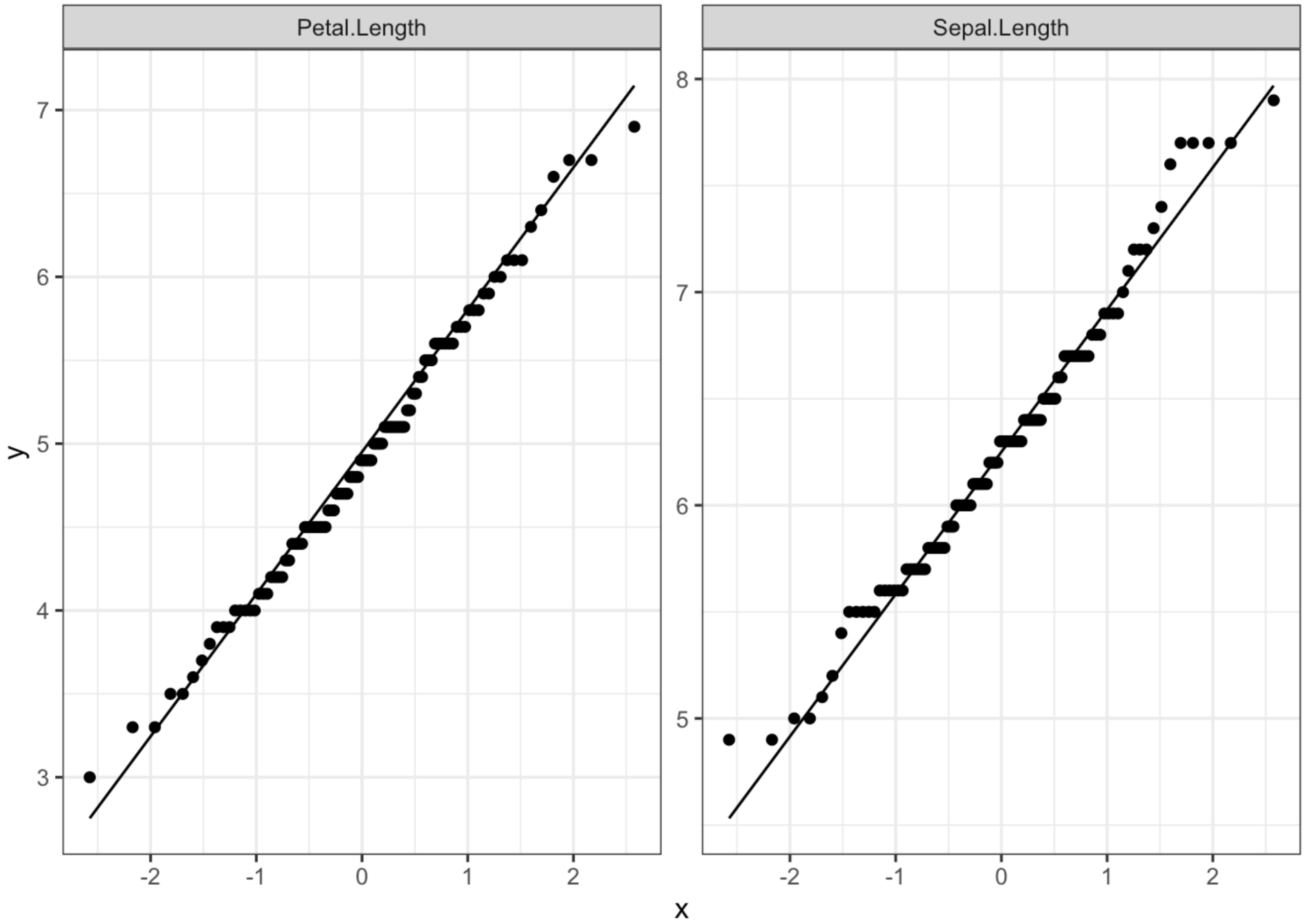
Normality

Assessing the normality assumption requires the testing of normality in both variables. This should always be done with all three graphical tests and the mathematical tests. We can assess these simultaneously as below:

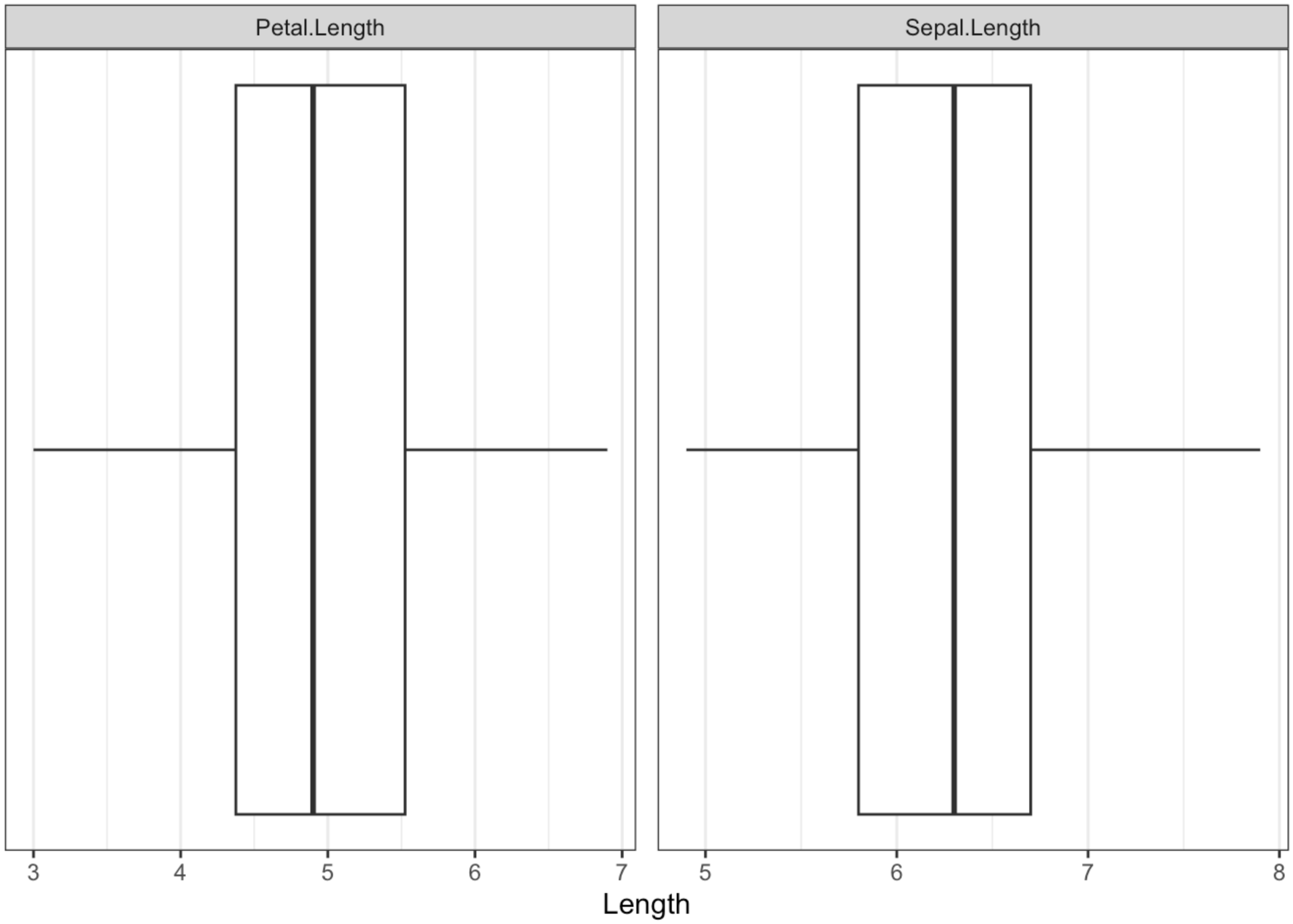
```
# Transform the dataset into a long format with your variables of interest
iris3 <- iris2 %>%
  mutate(ID = factor(1:nrow(.))) %>%
  select(ID, Sepal.Length, Petal.Length) %>%
  pivot_longer(cols = Sepal.Length:Petal.Length,
               names_to = "Measurement",
               values_to = "Length")

# Histogram
ggplot(data = iris3, mapping = aes(x = Length)) +
  geom_histogram(bins = 30, fill = 'gray', color = 'black') +
  facet_wrap(~ Measurement, scales = "free")
```



```
# Q-Q plot
ggplot(data = iris3, mapping = aes(sample = Length)) +
  geom_qq() +
  geom_qq_line() +
  facet_wrap(~ Measurement, scales = "free")
```



```
# Boxplot
ggplot(data = iris3, mapping = aes(x = Length)) +
  geom_boxplot() +
  theme(axis.ticks.y = element_blank(),
        axis.text.y = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +
  facet_wrap(~ Measurement, scales = "free")
```



```
stat.desc.clean(dataset = iris3, variable = Length, Measurement)
```

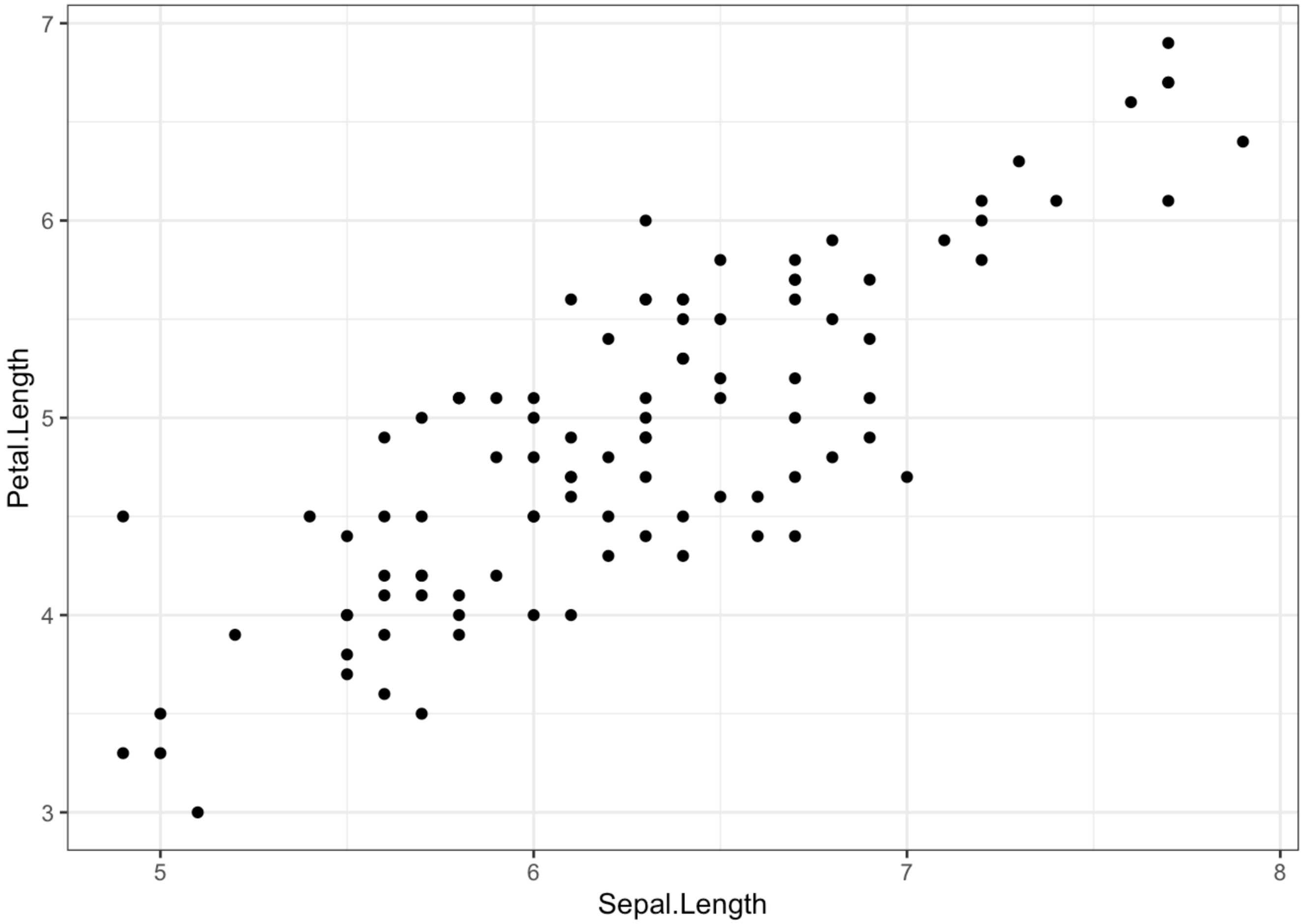
```
## # A tibble: 2 × 7
## # Groups:   Measurement [2]
##   Measurement  skewness skew.2SE kurtosis kurt.2SE normtest.W normtest.p
##   <chr>           <dbl>    <dbl>    <dbl>    <dbl>      <dbl>      <dbl>
## 1 Petal.Length    0.166    0.344   -0.421   -0.440      0.991      0.745
## 2 Sepal.Length    0.299    0.620   -0.197   -0.206      0.981      0.146
```

Based on all of these plots and mathematical tests, both variables clearly meet the assumption of normality.

Linearity/Monotonicity

The linearity assumption means that the relationship between variables is linear, which we can assess using a scatter plot. The scatter plot requires the `geom_point` function.

```
ggplot(data = iris2, mapping = aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point()
```

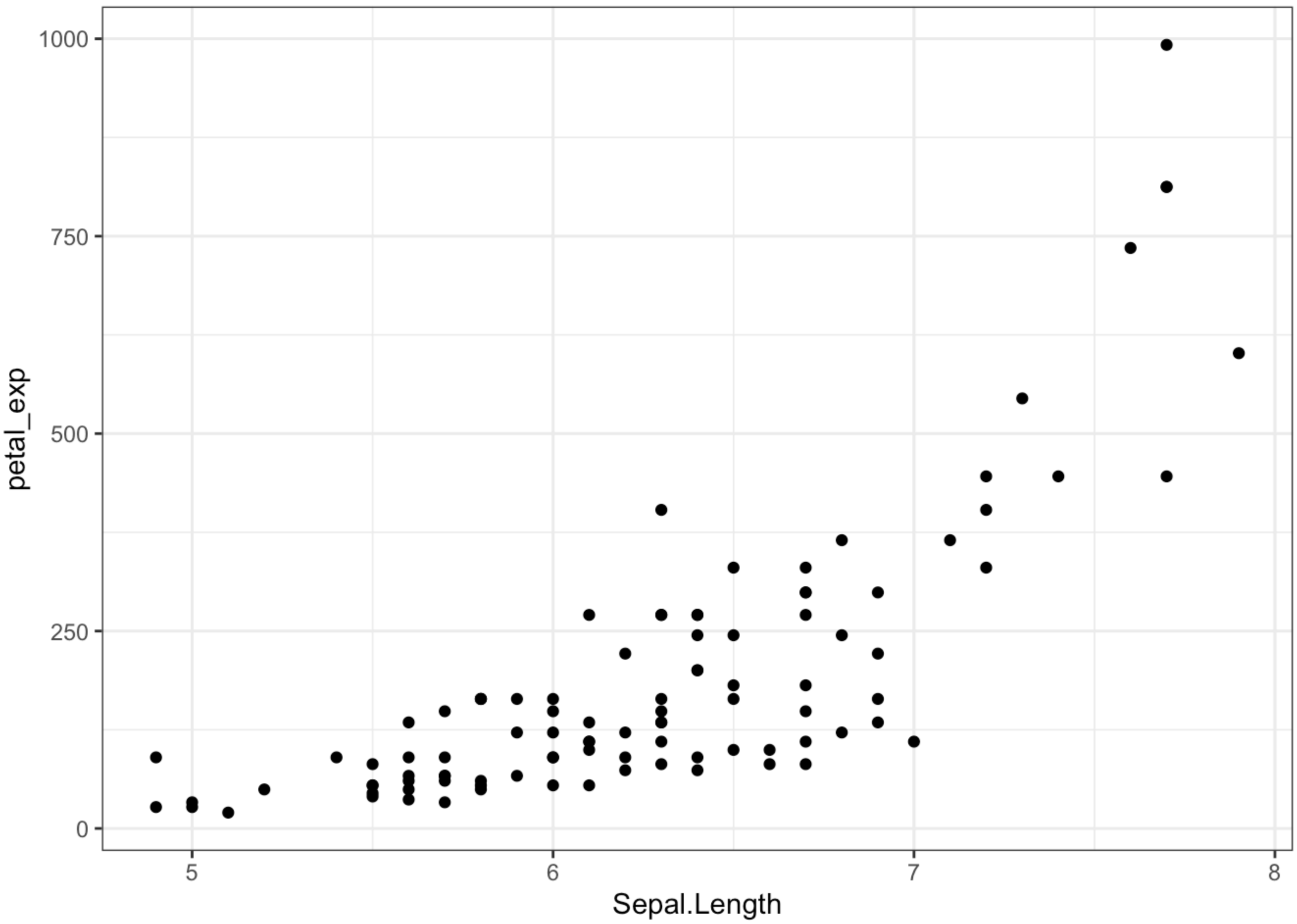


As you can see, petal length seems to linearly increase as sepal length increases, and doesn't have any point on the line with a negative slope. Therefore the linearity and monotonicity assumptions are met.

Corrections for Deviations from Linearity

As we showed last lab, you can transform your variables using either the square root or log transforms if your data doesn't meet the normality assumption. Importantly, these transformations can be extremely useful for correcting data that exhibit non-linear relationships. For example, let's imagine we were testing the relationship between the exponential of the petal length (e ^ Petal.Length) vs sepal length. This will look like:

```
iris4 <- iris2 %>%
  mutate(petal_exp = exp(Petal.Length))

ggplot(data = iris4, mapping = aes(x = Sepal.Length, y = petal_exp)) +
  geom_point()
```



Clearly, the relationship is now non-linear, and if you came across a situation like this in the wild, you would want to look into a log transformation of the petal_exp variable such that the relationship would become linear (because the natural log is the opposite of the exponential function). This would allow you to perform a Pearson correlation as the linearity assumption would now be met.

Running a Correlation Analysis

All three types of correlation can be performed using the `cor` function:

`cor(x, y, method)`

- `x` : your first variable
- `y` : your second variable
- `method` : the correlation method (either "pearson", "kendall", or "spearman")

The output of this function is the correlation coefficient, which is a numeric value between -1 and 1. A negative value implies a negative correlation (as one variable increases, the other decreases) while a positive value implies a positive correlation.

As an example, let's test the correlation between `Sepal.Length` and `Petal.Length` in the `iris` dataset. As both variables meet the assumptions, the appropriate test is a Pearson's correlation test:

```
cor(x = iris2$Sepal.Length, y = iris2$Petal.Length, method = "pearson")
```

```
## [1] 0.8284787
```

Describe the correlation. Is it positive or negative? Strong or weak?

In this example, we have a positive correlation with a correlation coefficient of 0.83. As this coefficient is close to 1, it indicates a strong correlation.

Rank Correlation

Now let's say that the data does not meet the assumptions of the Pearson correlation test, even after you transform the data. The appropriate correlation test would therefore be the `kendall` or `spearman` test. Let's do both by changing the `method` argument to `"spearman"` and `"kendall"` respectively:

```
cor(x = iris2$Sepal.Length, y = iris2$Petal.Length, method = "spearman")
```

```
## [1] 0.7804327
```

```
cor(x = iris2$Sepal.Length, y = iris2$Petal.Length, method = "kendall")
```

```
## [1] 0.6179442
```

Compared to the Pearson's correlation coefficient, the Spearman's rho is close in value at 0.78, but the Kendall's tau is much lower at 0.62. In this case, given we met the assumptions for the Pearson's correlation test, this test is the most appropriate. To illustrate the usage of these non-parametric tests, let's run these same tests on the exponentially transformed data that we made earlier (which doesn't meet the linearity assumption of the Pearson's test).

```
cor(x = iris4$Sepal.Length, y = iris4$petal_exp, method = "spearman")
```

```
## [1] 0.7804327
```

```
cor(x = iris4$Sepal.Length, y = iris4$petal_exp, method = "kendall")
```

```
## [1] 0.6179442
```

Notice how the results are identical to the untransformed data, which is not surprising given that the rank of the petal length variable didn't change when we transformed it, only the values.

Multiple Correlations

There are cases where you may like to know the correlations between many different variables in a dataframe. Luckily, `cor` can easily take care of this. If we wanted to do the correlations between multiple variables, we can set the `x` argument to a dataset consisting of all the variables of interest and leave out the `y` argument:

```
iris2 %>%
  select(Sepal.Length:Petal.Width) %>%
  cor(method = "pearson")
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.5538548    0.8284787   0.5937094
## Sepal.Width     0.5538548   1.0000000    0.5198023   0.5662025
## Petal.Length    0.8284787   0.5198023    1.0000000   0.8233476
## Petal.Width     0.5937094   0.5662025    0.8233476   1.0000000
```

The output is a table of size n x n, where n is the number of columns in the input. In this case, we input 4 columns, so the output table is a 4 x 4. You can clearly see the correlation values between all of our `iris` variables this way.

Statistically Significant Correlations

During a correlation analysis, you will also want to determine whether the correlations you have are significant or not. The basic way to do this is to use the `cor.test` function. You set it up exactly the same as `cor` .

Let’s see if our `Sepal.Length` and our `Petal.Length` data are significantly correlated:

```
cor.test(x = iris2$Sepal.Length, y = iris2$Petal.Length, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  iris2$Sepal.Length and iris2$Petal.Length
## t = 14.645, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7549049 0.8814586
## sample estimates:
##       cor
## 0.8284787
```

Are `Sepal.Length` and `Petal.Length` significantly correlated?

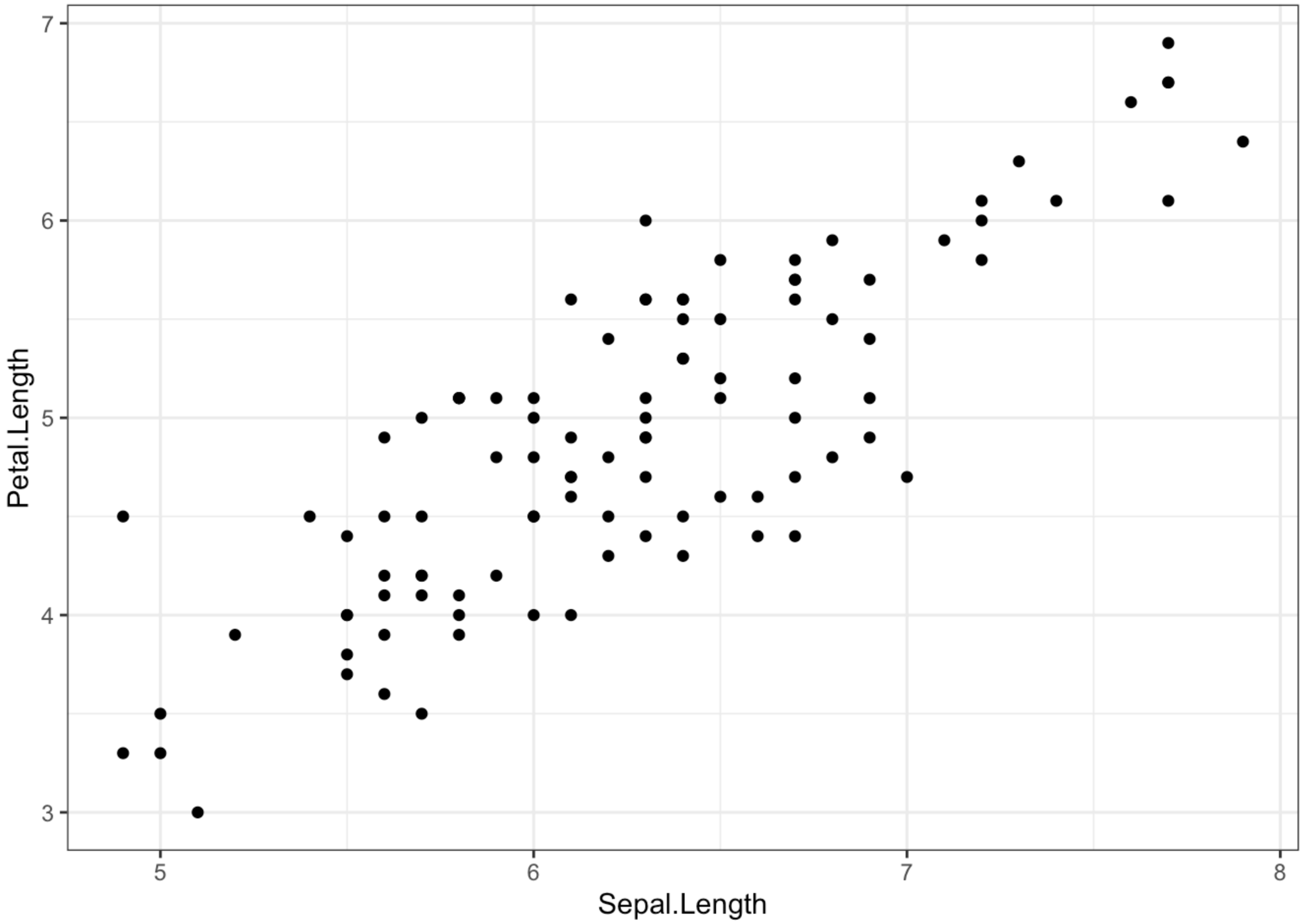
Yes, the correlation coefficient is 0.83 with a 95% confidence interval of 0.75 to 0.88 and p < 2.2e-16.

Plotting Correlations and Correlation Matrices

Now that we know how to find the correlation coefficient and how to determine if the correlation is significant, we can graph our correlations using a scatterplot.

Let’s plot our `Sepal.Length` and `Petal.Length` to visualize their correlation:

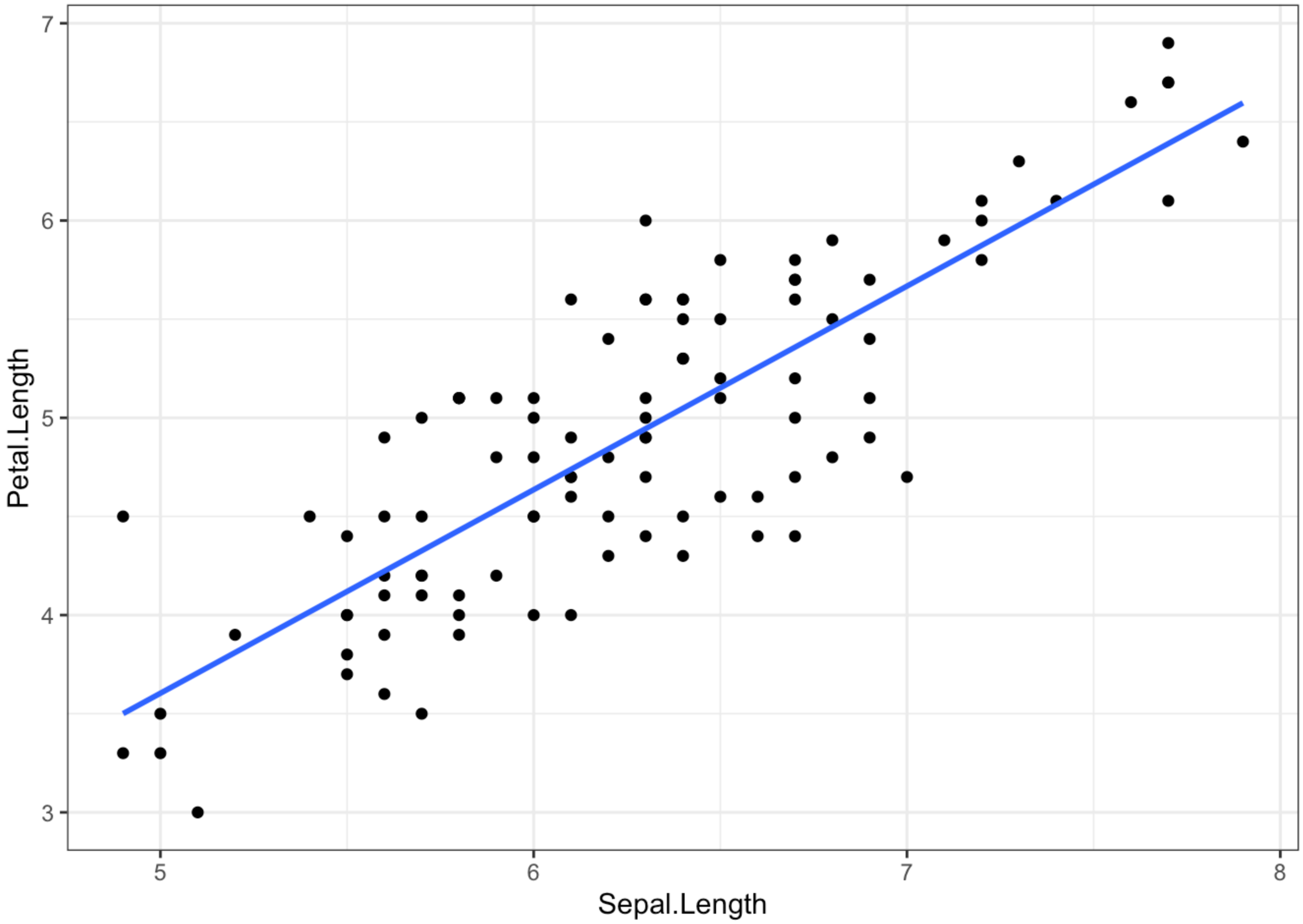
```
ggplot(data = iris2, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point()
```



Notice how the majority of the points fall along a diagonal, contributing to a high correlation (0.83 from above).

Another tool you can use along with the scatterplot is to add a regression line to it using `geom_smooth`.

```
ggplot(data = iris2, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```



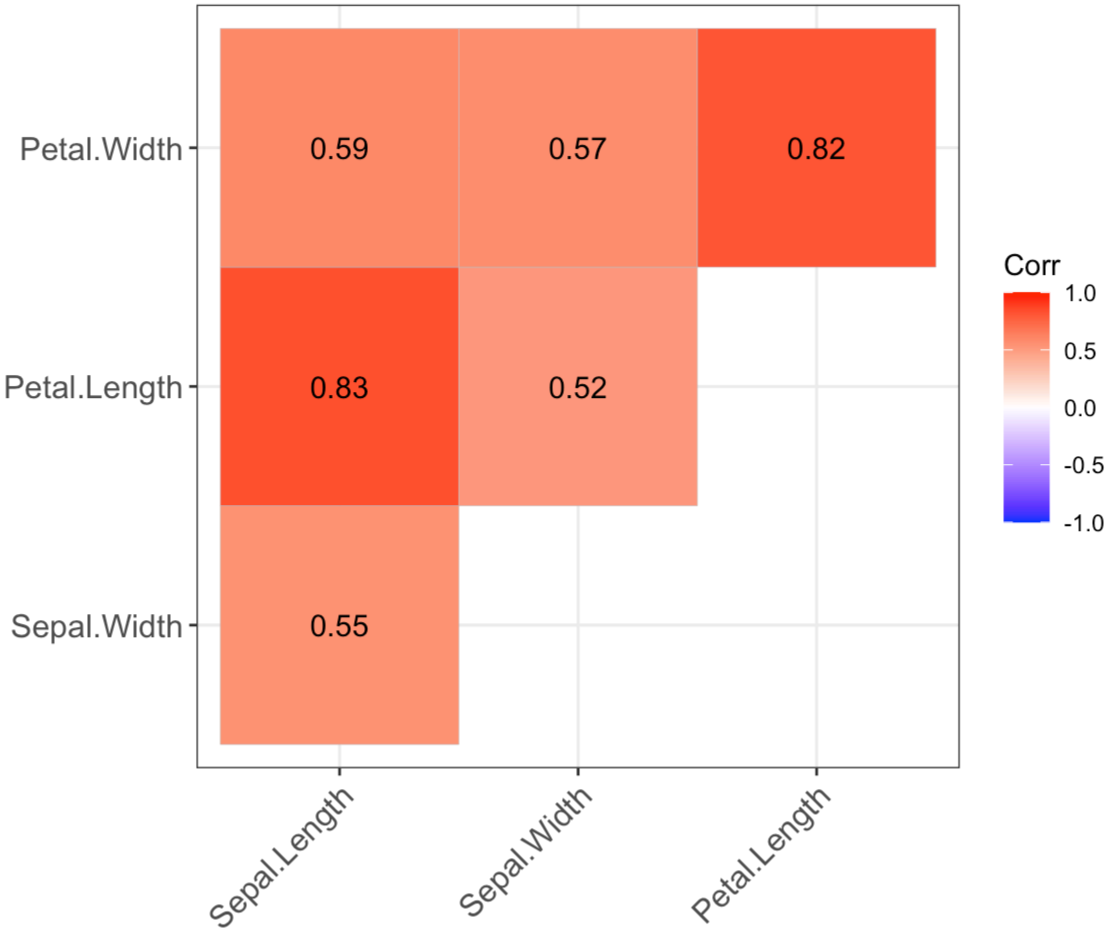
This, however, only plots one correlation. What if you wanted to show multiple correlations?

At it’s most basic, `ggcorrplot` will accept a correlation matrix like we calculated above (see the first example under Multiple Correlations) using `cor` and will plot a heatmap of correlation values.

Other useful inputs for the function are:

- `lab` : add correlation coefficients to each square in the plot
- `ggtheme` : change the theme of the ggplot
- `type` : say whether you want to show the full square or just the upper or lower triangles (‘full’ is default, ‘upper’, ‘lower’)

```
iris2 %>%
  select(Sepal.Length:Petal.Width) %>%
  cor(method = "pearson") %>%
  ggcorrplot(lab = TRUE,
             ggtheme = theme_bw,
             type = "upper")
```



The plot shows three variables on each axis (total of four unique variables), with each intersection between these variables representing their pairwise correlation coefficient. As you can see, all six squares are red (denoting positive correlations), with the strength of correlation shown by the depth of color. Correlation matrix plots are great tools for seeing trends between multiple variables at a time.

Reporting the Results

You can use the following template for reporting a correlation, using the Pearson correlation as an example. The `r` represents the correlation coefficient and the `n` represents the number of observations.

"A Pearson correlation test on the sepal length and petal length of iris flowers showed a significant positive correlation [r = 0.83, n = 100, p < 2.2e-16]."

Independent Practice

For these exercises, we will be using the `microbiome3.RData` file on Canvas. We hypothesize that there will be a significant correlation between the concentrations of Proteobacteria (`Proteo` column) and Bacteroidetes (`Bacter` column) microbes. Investigate this using the appropriate type of correlation.

1. Get to know your data: describe your variables and their distribution
2. Assumptions

   a. Determine whether assumptions are met

   b. If assumptions are not met, describe what you will do to account for this

3. Run the appropriate correlation test

4. Report your findings as you would describe them in a results section