# Hypothesis Testing & Statistical Tests Review

This worksheet will review statistical tests and models that are used for data analysis. We will not go into the theory but explain what the tests are used for and how to interpret them. We will be coding in R. Most of this should be STA258 review, but we will not cover one-sided tests. You can think of this as a statistical cookbook.

**Disclaimer:** for demonstration purposes, the data here is randomly generated and I have included the code for how I randomly generated data for learning purposes only. However, you should not randomly generate any data for your assignment.

## Contents

# 1 Hypothesis Testing

Statistical hypothesis testing is a fundamental concept in inferential statistics used to make decisions or draw conclusions about a population based on a sample of data. The two primary hypotheses involved in statistical hypothesis testing are:

1. **Null Hypothesis:** The initial assumption that there is no effect, no difference, or no relationship between variables. In other words, it suggests that any observed differences or relationships in the sample data are due to randomness within the sample.

2. **Alternative Hypothesis:** Proposes a specific effect, difference, or relationship that is being investigated.

P-values are frequently misinterpreted. Some people like saying "It's the evidence against the null hypothesis" and call it a day. Naturally, my monkey brain thinks, "what evidence?" So let me give you a more specific definition: p-value is the probability of the event occurring given that the null hypothesis is true. A primitive example would be to do a coin toss. Suppose I am suspicious that I have an unfair coin. (I.e. the probability of heads is not equal to the probability of tails.)

(Null hypothesis) $H_0 : p_{\text{heads}} = 0.5$

(Alternative hypothesis) $H_A : p_{\text{heads}} \neq 0.5$

Now suppose I flip this coin 40 times. Out of these 40 times, I observe that I have obtained heads 10 times. If I test this in R, I'll get the following result:

```
> binom.test(x = 10, n = 40, p = 0.5)
Exact binomial test
data:  10 and 40
number of successes = 10, number of trials = 40, p-value = 0.002221
```

In this context, the interpretation is that "under the assumption that the null hypothesis holds true, there exists an approximate 0.2% probability that out of 40 coin tosses, exactly 10 of them would result in heads." Usually, a significance threshold of alpha = 5% is employed as a cutoff for p-values. The selection of alpha = 5% is somewhat arbitrary but it aligns conceptually: if the likelihood of such an outcome is less than 5%, it suggests that the null hypothesis is fairly improbable.

Naturally, some people may believe in lowering or increasing the level of significance, but we will not discuss the consequences in this note. If you're interested, you can revisit your notes on concepts like statistical power, Type I, and Type II errors.

# 2 Verifying Assumptions

Assumptions are necessary when performing statistical tests. Some of them don't need to be verified as they are implied by the design of the study, i.e., they assume randomness, independence, etc. However, normality and homogeneity of variances are common assumptions for most statistical tests.

## 2.1 Normality Assumption

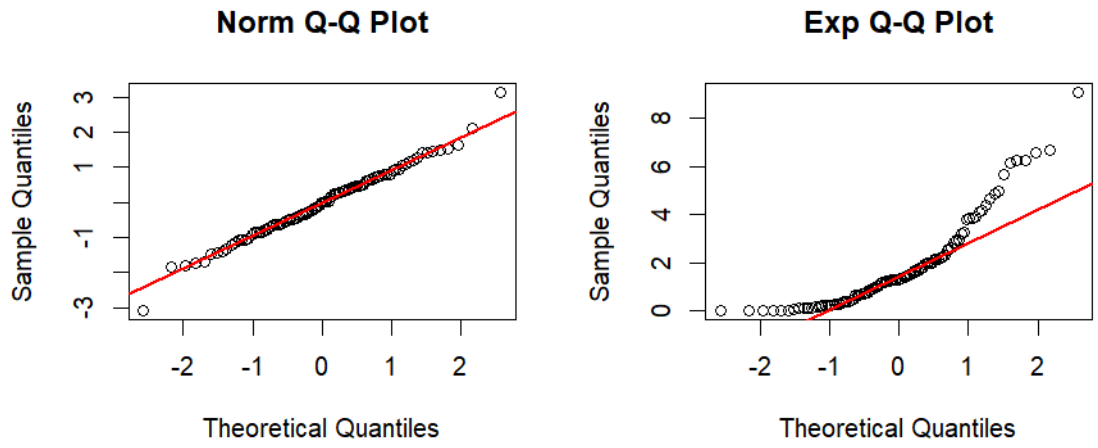
### 2.1.1 Central Limit Theorem

If you have a large sample size ($n \geq 30$) then you may claim that through the central limit theorem, you may use tests for normality.

### 2.1.2 Q-Q Plots

Elementary ways to verify for normality involve looking at a histogram or a box-plot. However, these are less precise. The best visual test for normality is the Q-Q plot (quantile–quantile plot). This can be checked using base R.

```
> X1 = rnorm(100, mean = 0, sd = 1)
> X2 = rexp(100, rate = 0.5)
> par(mfrow = c(1, 2))
> qqnorm(X1, main = "Norm Q-Q Plot"); qqline(X1, col = "red", lwd = 2)
> qqnorm(X2, main = "Exp Q-Q Plot"); qqline(X2, col = "red", lwd = 2)
```




As you can see, $X1$ which is generated from a normal distribution has a Q-Q plot which follows the straight $y = x$ line, whereas $X2$ which is generated from an exponential distribution has a Q-Q plot which doesn't follow the straight line.

### 2.1.3 Shapiro-Wilk test

Instead of a visual, there are many quantitative tests for normality. Here we will discuss the Shapiro-Wilk test as it is one of the most commonly used. Here:

> Null hypothesis: the data comes from a normal distribution.
>
> Alternative hypothesis: the data does not come from a normal distribution.

Using $X1$ and $X2$ as defined earlier in R, we obtain the following results:

```
# WARNING: p-values may vary since I didn't include a seed.
> shapiro.test(X1)
    Shapiro-Wilk normality test
data:  X1
W = 0.99073, p-value = 0.7238
> shapiro.test(X2)
    Shapiro-Wilk normality test
data:  X2
W = 0.84249, p-value = 6.324e-09
```

Which has the same results as the Q-Q Plots. There are some instances where the Q-Q Plots will not have the same results as the Shapiro-Wilk test. There's lots of online discussion about what to do. Personally, if I find the Q-Q Plots to be obvious (either closely following the $y = x$ line or extremely far from it) then I will take preference of the Q-Q Plots. Otherwise, I will go with the Shapiro-wilk test results.

In the linear regression case, you may still use shapiro.test but make sure you apply it explicitly on the residuals of the linear model. I.e., an appropriate command would be "shapiro.test(residuals(model))"

## 2.2 Homogeneity of Variances

### 2.2.1 Categorical Explanatory Variables

In introductory applied statistics courses, instructors usually recommend looking at a box-plot and consider the size of the interquartile range. I personally don't like vague visualizations, so let's introduce Bartlett's test. Here:

> Null hypothesis: the variances amongst the groups are equal.
>
> Alternative hypothesis: the variances amongst the groups are not equal.

Unlike the tests for normality, Bartlett's test in R requires two variables: a categorical variable and a quantitative variable. Below I created a sample data frame and performed the test.

```
# WARNING: p-values may vary since I didn't include a seed.
> data = data.frame(
+   Gender = sample(c("Male", "Female"), size = 50, replace = TRUE),
+   Score = sample(1:100, size = 50, replace = TRUE)
+ )
> bartlett.test(Score ~ Gender, data = data)
Bartlett test of homogeneity of variances
data:  Score by Gender
Bartlett's K-squared = 0.94008, df = 1, p-value = 0.3323
```

The p-value above the $\alpha = 0.05$ significance level indicates that the variances are approximately the same.

Note that if you failed to prove normality or equality of variances, for this course you may do the following:

1. Perform the test anyways, but note the limitation that one of the assumptions were violated. (Note: you cannot do this in formal research, but we understand you may have not encountered non-parametric tests.)
2. Perform the non-parametric version. I will briefly name some of them in certain sections.

### 2.2.2 Linear Regression Case

Bartlett's test is not appropriate for testing homogeneity of variances amongst residuals for linear regression. Here, the Breusch–Pagan test is often used instead. This has a similar null and alternative hypothesis as bartlett's test, meaning if you have a $p$-value below the $\alpha$ threshold then you assume the variances are unequal.

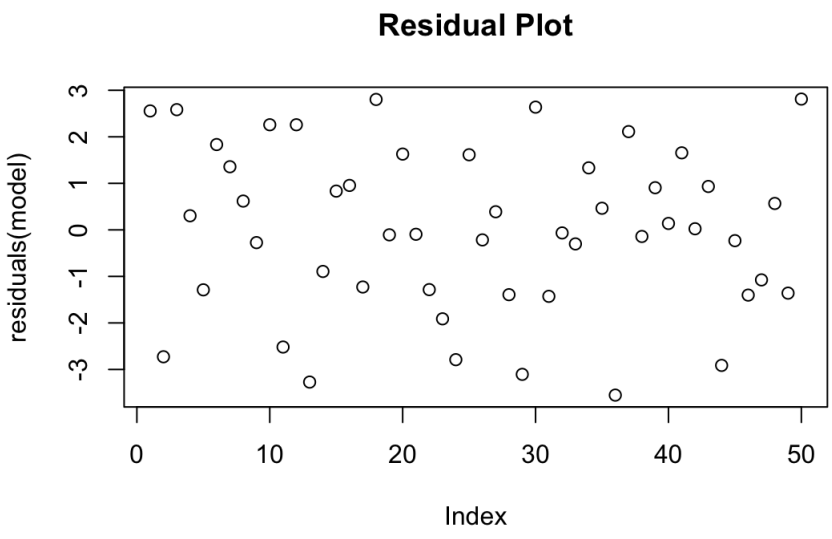
Consider the following model:

```
> library(lmtest) # use install.packages("lmtest") if you don't have it installed
> data = data.frame(
+   Sleep = sample(seq(3,9,0.25), size = 50, replace = TRUE),
+   CGPA = sample(seq(1,4,0.05), size = 50, replace = TRUE)
+ )
> model = lm(Sleep ~ CGPA, data = data)
> bptest(model)
studentized Breusch-Pagan test
data:  model
BP = 0.58744, df = 1, p-value = 0.4434
```

Here, since $p$-value of 0.4434 is greater than $\alpha = 0.05$, we would conclude that the variances are the same amongst the residuals.

## 2.3 Independence Observations (Regression)

After creating a regression model, use "plot(residuals(model), main = "Residual Plot")" in R. Notice that if you have a plot where the residuals seem spread out in a random fashion (similar to the image below), then you likely have independent residuals. If you notice a shape similar to a polynomial, like a quadratic, or perhaps a shape of a sinusoidal function, this is likely a sign to use a different regression method.



## 2.4 No Multicollinearity

Multicollinearity happens when two (or more) of the explanatory variables are highly correlated to each other. It isn't a good idea to use multiple redundant variables for a model, or it makes it less effective. There's also the problem with over-fitting. We will not get into the technical details. A common way to check multicollinearity is to compute the variance inflation factor (VIF). In general, if the VIF of a variable is greater than 5 then you have an issue.

Consider the following data, which by design clearly has an issue with multicollinearity:

```
> library(car) # use install.packages(car) if you don't have it
> test = rnorm(100, mean = 0, sd = 10)
> data = data.frame(
+   X1 = test,
+   X2 = test + rnorm(100, mean = 0, sd = 0.1),
+   Y = rnorm(100, mean = 0, sd = 10)
+ )
> model = lm(Y ~ X1 + X2, data = data)
> vif(model)
      X1       X2
15236.48 15236.48
```

Here, both X1 and X2 have values much greater than 5, indicating they're highly correlated with each other. I would suggest just getting rid of one of them.

# 3 Simple Statistical Tests

## 3.1 Comparing Proportions

### 3.1.1 One Sample Test for Proportion

This is used to test the sample proportion between an expected proportion. Usually, this expected proportion is 0.5.

**Assumptions:**

1. The data was randomly sampled and responses are independent from each other.
2. The population follows a binomial distribution (binary options, such as success/failure, yes/no, etc.)
3. $np_e \geq 5, n(1-p_e) \geq 5$ where $n$ represents the sample and $p_e$ represents the expected proportion.

   (Null hypothesis) $H_0 : p = p_e$ (typically $p_e = 0.5$)

   (Alternative hypothesis) $H_A : p \neq p_e$

There are different ways to compute this test, which depends on the sample size and the continuity correction. Here we use prop.test() as I prefer the continuity correction. Assuming we have 65 out of 100 successes, and assuming our expected proportion is 0.5, we obtain the following test:

```
> prop.test(x = 65, n = 100, p = 0.5, correct = FALSE)
1-sample proportions test without continuity correction
data:  65 out of 100, null probability 0.5
X-squared = 9, df = 1, p-value = 0.0027
alternative hypothesis: true p is not equal to 0.5
```

Here, our $p$-value is less than $\alpha = 0.05$, so we claim that at the $\alpha = 0.05$ significance level, the true proportion cannot be equal to 0.5. Obviously, our sample proportion is greater than 0.05 so we can be more ambitious and in plain language claim that the true proportion is greater than 0.5.

If the assumptions fail, you can try an exact binomial test. This was used in the first section to explain what hypothesis testing is.

### 3.1.2 Two Sample Test for Proportion

The two-sample test for proportion is used to compare two proportions (or percentages) from independent samples to determine if there is a statistically significant difference between them. This type of test is particularly useful when you want to assess whether the proportions in two different groups are significantly different. I.e. Suppose we want to observe the differences between the proportion of students who regularly attend lectures and whether

or not they have a part time job. Let $p_1$ be the proportion for the first group, and $p_2$ be the proportion for the second group. Then, $n_1$ denotes the sample size of the first group, and $n_2$ denotes the sample size for the second group.

**Assumptions:**

1. The data was randomly sampled and responses are independent from each other.
2. The population follows a binomial distribution (binary options, such as success/failure, yes/no, etc.)
3. $n_1\hat{p}_1, n_1(1-\hat{p}_1), n_2\hat{p}_2, n_2(1-\hat{p}_2) \geq 5$

   (Null hypothesis) $H_0 : p_1 - p_2 = 0$

   (Alternative hypothesis) $H_A : p_1 - p_2 \neq 0$

We can compute the this test for proportion using the following command in R (with an amazing randomly generated sample):

```
> data = data.frame(
+   LectureAttend = sample(c("Y", "N"), size = 50, replace = TRUE),
+   PartTimeJob = sample(c("Y", "N"), size = 50, replace = TRUE)
+ )
> contingency_table = table(data$LectureAttend, data$PartTimeJob)
> prop.test(contingency_table)
2-sample test for equality of proportions with continuity correction
data:  contingency_table
X-squared = 4.808e-31, df = 1, p-value = 1
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.3458311  0.2753183
sample estimates:
   prop 1    prop 2
0.4230769 0.4583333
```

Here we have that the $p$-value is greater than $\alpha = 0.05$ (it is in fact, equal to 1). Hence, there is no significant differences between the proportions of two groups.

If the assumptions fail, you can use Fisher's exact test:

```
> fisher.test(contingency_table)
Fisher's Exact Test for Count Data
data:  contingency_table
p-value = 1
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.2456448 3.0560151
sample estimates:
odds ratio
  0.869156
```

Here, we also obtain a similar result, albeit it is uncommon for this to occur.

## 3.2 Comparing Means

### 3.2.1 One Sample Test for Mean

A one-sample test for the mean is used to determine if the mean of a single sample is significantly different from a supposedly known population mean, we'll denote as $\mu_0$.

**Assumptions:**

1. The samples are independent from each other & are obtained randomly.
2. The samples are normally distributed.

   (Null hypothesis) $H_0 : \mu = \mu_0$

   (Alternative hypothesis) $H_A : \mu \neq \mu_0$

Note that for small samples ($< 30$) and if the variance is unknown, typically a t-test is employed. If the sample is large and you know the variance, one can directly compute it using the z-test. However, I doubt you'll actually know the variance for your study, so for now we'll just use the t-test, regardless:

```
> data = rnorm(100, mean = 20, sd = 5)
> t.test(data, mu = 25)
One Sample t-test
data:  data
t = -8.4669, df = 99, p-value = 2.374e-13
alternative hypothesis: true mean is not equal to 25
95 percent confidence interval:
 19.15906 21.37694
sample estimates:
mean of x
   20.268
```

Here I just generated data from a normal distribution of a different mean (20) instead of $\mu_0 = 25$. Thankfully, we've rejected the null hypothesis at the $\alpha = 0.05$ with a $p$-value of $2.374e - 13$.

### 3.2.2 Two Sample Test for Mean

The two-sample test for mean is similar to the two-sample test for proportion, except we are computing for the mean of two separate groups. For example, suppose we are testing the average grades of students who regularly attend lectures versus those who don't regularly attend lectures. Let $\mu_1$ represent the mean grades of the first group (students who regularly attend lectures), and let $\mu_2$ represent the mean grades of the second group (students who don't regularly attend lectures).

**Assumptions:**

1. The samples are independent from each other & are obtained randomly.
2. The samples are normally distributed.
3. The variances for the two independent groups are equal.

   (Null hypothesis) $H_0 : \mu_1 - \mu_2 = 0$

   (Alternative hypothesis) $H_A : \mu_1 - \mu_2 \neq 0$

In R, here's how we can conduct the test with some dummy data:

```
> data = data.frame(
+   LectureAttend = sample(c("Y", "N"), size = 50, replace = TRUE),
+   Grades = rnorm(50, mean = 60, sd = 18)
+ )
> t.test(Grades ~ LectureAttend, data = data, var.equal = TRUE)
Two Sample t-test
data:  Grades by LectureAttend
t = 0.13874, df = 48, p-value = 0.8902
alternative hypothesis: true difference in means between group N and group Y is not equal to 0
95 percent confidence interval:
 -9.555542 10.972023
sample estimates:
mean in group N mean in group Y
       58.22754        57.51930
```

Here, $p$-value is 0.8902 which is above the $\alpha = 0.05$ level of significance, and thus we would say there's no difference between lecture attendance and grades (although since this data is fake, you shouldn't let this dictate your lecture attendance).

### 3.2.3 Alternatives for Failing Assumptions

Note that if the homogeneity of variance assumption is violated, you can use "var.equal = FALSE". (Arguably, if the variances of the two means are different then the comparison of means doesn't make sense. Even if the means are the same, the performance of the two treatments could be quite different due to the difference in the variances. So in such a case saying the means are the same does not mean the treatments are the same which is the goal of

such an analysis. You won't get deducted for using these tests if the variances are unequal, but it was an interesting point brought up by my supervisor.)

If the normality assumption is violated (and you have a small sample), we have a larger problem. I recommend the Mann–Whitney U test (also called the Wilcoxon rank-sum test). They have two different names because they created the same statistical test around the same time frame, but I suppose people didn't realise until it was too late. This was back in the 1940's before they had the glorious internet, so it's unlikely one just re-wrote a paper after reading the first-published edition. We can use the following command in R:

```
> wilcox.test(Grades ~ LectureAttend, data = data, var.equal = TRUE)

Wilcoxon rank sum exact test

data:  Grades by LectureAttend
W = 315, p-value = 0.9616
alternative hypothesis: true location shift is not equal to 0
```

Similarly, if if the homogeneity of variance assumption is violated alongside the violation of normality you can just say "var.equal = FALSE".

**Remark:** I have omitted the explanation of paired tests since I don't think your data in STA304 will be paired.

## 3.3 Testing for Independence (Categorical Variables)

The chi-square test of independence assesses whether there is a relationship or association between two categorical variables. It is often applied when you have a contingency table (also known as a cross-tabulation or a two-way table) that displays the frequencies or counts of observations for the two categorical variables.

**Assumptions:**

1. The data was randomly sampled and responses are independent from each other.
2. The expected value of each cell is greater than 5.

Supposing that A and B denote two distinct categorical variables, then the hypothesis test being performed is:

(Null hypothesis) $H_0 : A \perp\!\!\!\perp B$ (A and B are independent of each other.)

(Alternative hypothesis) $H_A : A \not\perp\!\!\!\perp B$ (A and B are not independent of each other.)

There are multiple ways to write the "not independent" symbol.

Here's an example: suppose you have two categorical variables, i.e, gender and whether they are enrolled in a statistics program. I'm sure you've noticed that the statistics department at UTM is male dominated. However, does this also apply for students? The data is fabricated below.

```
> data = data.frame(
+   Gender = sample(c("M", "F"), size = 50, replace = TRUE, prob = c(0.7, 0.3)),
+   Program = sample(c("Stats", "NoStats"), size = 50, replace = TRUE, prob = c(0.8, 0.2))
+ )
> contingency_table = table(data$Gender, data$Program)
> test = chisq.test(contingency_table)
Warning message:
In chisq.test(contingency_table) :
  Chi-squared approximation may be incorrect
```

Notice the last warning message; this is an indication that we should likely use a different test. Let's ignore the error for now and see the results of the test:

```
> test
Pearson's Chi-squared test with Yates' continuity correction
data:  contingency_table
X-squared = 4.3434e-31, df = 1, p-value = 1
```

Here, our $p$-value is 1, wow! Hence gender and stats program are not related to each other. However, we were given a warning error. If we look at the expected frequencies of the cells, we'll notice one of them was definitely less than 1:

```
> test$expected
    NoStats Stats
  F     1.8  13.2
  M     4.2  30.8
```

### 3.3.1 Alternatives for Failing Assumptions

A different test we can use is Fisher's exact test, which is the non-parametric version of the Chi-square test for independence, commonly used for small sample sizes. We obtain the following:

```
> fisher.test(contingency_table)
Fisher's Exact Test for Count Data
data:  contingency_table
p-value = 1
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.09625652 9.53686983
sample estimates:
odds ratio
  1.187987
```

We obtain a similar result, but at least there's no error message! (Note: Fisher's exact test is the non-parametric version for independence and the two sample test for proportions.)

## 3.4 Simple Linear Regression

Simple linear regression is a statistical method used to model the relationship between two variables, typically referred to as the independent variable (X) and the dependent variable (Y). The relationship is represented by a straight line equation:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- $\beta_0$ is the intercept.
- $\beta_1$ is the slope.
- $\epsilon$ represents the error term, accounting for the variability in Y that is not explained by the linear relationship with X.

In simple linear regression, the goal is to estimate the values $\beta_0$ and $\beta_1$ that best fit the observed data. This is often done using the method of least squares, which minimizes the sum of the squared differences between the observed values of Y and the values predicted by the linear equation. Once the regression coefficients $\beta_0$ and $\beta_1$ are estimated, you can use the regression equation to make predictions for Y based on different values of Y. This is further discussed in STA302, but it was briefly mentioned in STA258/STA260.

**Assumptions:**

1. **Linearity:** The relationship between X and Y is linear. (You don't really need to verify this one, but sometimes linear regression is not the best way to model a response variable from explanatory variables.)
2. **Independence of errors:** X is independent from the residuals. (If you plot the residuals and notice a pattern, this would imply you should use a different model.)
3. **Normality of errors:** the residuals must be approximately normally distributed.
4. **Homogeneity of variances amongst errors:** the variance of the residuals are the same for all values of X.

And the hypothesis test performed would be:

(Null hypothesis) $H_0 : \beta_1 = 0$

(Alternative hypothesis) $H_A : \beta_1 \neq 0$

Below is an example, where X and Y were just randomly generated from a normal distribution:

```
> data = data.frame(
+   X = rnorm(100, mean = 0, sd = 10),
+   Y = rnorm(100, mean = 0, sd = 10)
+ )
> model = lm(Y ~ X, data = data)
> summary(model)
Call:
lm(formula = Y ~ X, data = data)
Residuals:
     Min       1Q   Median       3Q      Max
-21.8468  -7.1242  -0.8884   8.0431  23.1106
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.4948     1.0380   0.477    0.635
X             0.1086     0.1021   1.064    0.290
Residual standard error: 10.38 on 98 degrees of freedom
Multiple R-squared:  0.01142,Adjusted R-squared:  0.001337
F-statistic: 1.133 on 1 and 98 DF,  p-value: 0.2899
```

Few things to note:

- For the coefficient of determination ($R^2$), you check multiple R-squared, not adjusted R-squared.
- The F-statistic and $p$-value at the end is testing for significance of the entire model (both the intercept and the explanatory variable).
- To check if X is significant, check the value under the "$\Pr(>|t|)$" column of the coefficients. Note that since this data was randomly generated, we obviously won't see any significant effects.

## 3.5 Correlation

If you know about simple linear regression then you should know about correlation. For some reason, from personal experience (and what it seems like based off of what students report), Pearson's correlation coefficient $\rho$ doesn't receive much attention in pre-requisite courses. However, it is one of the most simplest measures: the linear association between two variables. It is a standarized version of covariance. A correlation coefficient of...

- $\rho = -1$ indicates a perfectly negative linear correlation between two variables.
- $-1 < \rho \leq -0.8$ indicates a strong negative correlation between two variables.
- $-0.8 < \rho \leq -0.5$ indicates a moderate-to-strong negative correlation between two variables.
- $-0.5 < \rho \leq -0.3$ indicates a moderate negative correlation between two variables.
- $-0.3 < \rho < 0$ indicates a weak negative correlation between two variables.
- 0 indicates no linear correlation between two variables.
- $0 < \rho \leq 0.3$ indicates a weak positive correlation between two variables.
- $0.3 < \rho \leq 0.5$ indicates a moderate positive correlation between two variables.
- $0.5 < \rho \leq 0.8$ indicates a moderate-to-strong positive correlation between two variables.
- $0.8 < \rho < 1$ indicates a strong positive correlation between two variables.
- 1 indicates a perfectly positive linear correlation between two variables.

Now these are just suggestions and are not absolute. For example, psychology students that take a statistics course for psychology majors may learn that $|\rho| < 0.5$ indicates a weak relationship. Don't get too caught into this guide and just treat it as some suggestions.

**Assumptions:**

1. **Linearity:** The relationship between X and Y is linear.
2. **Normality of errors:** $X$ and $Y$ follow a normal distribution. (But it allegedly doesn't matter much if this is violated...)

And the hypothesis test performed would be:

(Null hypothesis) $H_0 : \rho = 0$

(Alternative hypothesis) $H_A : \rho \neq 0$

We will make dummy data and then compute Pearson's correlation coefficient using the following command in R:

```
> cor.test(data$X, data$Y, method='pearson')

Pearson's product-moment correlation

data:  data$X and data$Y
t = 1.2156, df = 98, p-value = 0.2271
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.07637137  0.31085177
sample estimates:
     cor
0.121875
```

Here, we have that the correlation $\rho$ is approximately 0.1219 which indicates a weak positive correlation. However, because $p = 0.2271 > 0.05$, this suggests that at the $\alpha = 0.05$ level, we still say there's no correlation. Teehee!

# 4 Advanced Statistical Tests

When we say "advanced" we just mean things that may not be taught in pre-requisite courses. In general, these aren't that advanced. Don't feel bad if you never learned any of these in depth though, because I self-learned most of these.

## 4.1 Bootstrapping

Bootstrapping has a dramatic backstory... A desperate statistician repeatedly tried to find a journal to publish his findings, and they kept on rejecting it because seemed too similar to existing methods. Anyways, he's successful now so let's not pity Bradley Efron. Bootstrapping is used when you have a small sample size but you still want to make estimations. There's a good StatQuest video on Youtube about this. But to summarize, the steps are as follows:

1. Make a new dataset (this is called a bootstrapped dataset) by sampling with replacement $m$ values of the original sample.
2. Make a calculation. Common ones involve the mean, standard deviation, median, coefficient of determination, coefficients of a linear model, etc.
3. Store the calculation from part 2.
4. Repeat steps 1-3 some $r$ amount of times.

Below we will use bootstrapping to compute the mean, with $r = 1000$ replications.

```
> library(boot) # use install.packages("boot") if you don't have it.
> data = data.frame(
+   Question1 = sample(1:7, size = 29, replace = TRUE),
+   Question2 = sample(1:7, size = 29, replace = TRUE),
+   Question3 = sample(1:7, size = 29, replace = TRUE)
+ )
> statistic_info = function(data, indices){
+   # Here we are just computing the mean.
+   return(mean(data[indices]))
+ }
> results = boot(data = data$Question1, statistic = statistic_info, R = 1000)
> results
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = data$Question1, statistic = statistic_info, R = 1000)
Bootstrap Statistics :
    original       bias    std. error
t1* 3.482759 -0.003551724   0.4023246
```

Note that the value of t1* gives you the mean of the original dataset. To actually find the mean from the bootstrapped sample, one must do the following:

```
> mean(results$t)
[1] 3.588379
```

For significance testing, I would personally use a one-sample test using the bootstrapped values.

## 4.2 Cronbach's Alpha

Cronbach's alpha is used to test the consistency amongst participant's responses, commonly for Likert-scale data. **Do not use Cronbach's alpha for unordered categories.** It is best explained through an example: suppose there is a test for mental illness diagnosis, where participants have to state how much they strongly agree to strongly disagree with certain statements. The instrument is designed in a way where those who tend to strongly agree with a statement display more symptoms. In general, we expect those who strongly agree to symptom "A" to also strongly agree with the rest of the symptoms.

In general, Cronbach's alpha may signify one (or both) of two things:

1. The questions were related to one another.
2. The participants were consistent with their responses.

The following is a table that provides the general guidelines of the reliability level and the interpretation:

| Cronbach's alpha | Reliability level |
|---|---|
| $\geq 0.9$ | Excellent |
| 0.80 - 0.89 | Good |
| 0.70 - 0.79 | Acceptable |
| 0.60 - 0.69 | Questionable |
| 0.50 - 0.59 | Poor |
| $\leq 0.49$ | Unacceptable |

Table 1: Range of reliability and its coefficient of Cronbach's alpha. As expected, the higher Cronbach's alpha indicates a higher reliability.

Below is an example for computing Cronbach's alpha. Note that you'll need to compute characters, such as "strongly agree", "agree", "slightly agree", "neutral", ..., "strongly disagree" into numerical values, typically values from 1 to 7. I randomly generated some data here, in which I should expect a poor reliability level:

```
> library(ltm) # if you do not have this installed, try install.packages("ltm")
> data = data.frame(
+   Question1 = sample(1:7, size = 50, replace = TRUE),
+   Question2 = sample(1:7, size = 50, replace = TRUE),
+   Question3 = sample(1:7, size = 50, replace = TRUE),
+   Question4 = sample(1:7, size = 50, replace = TRUE),
+   Question5 = sample(1:7, size = 50, replace = TRUE),
+   Question6 = sample(1:7, size = 50, replace = TRUE),
+   Question7 = sample(1:7, size = 50, replace = TRUE)
+ )
> cronbach.alpha(data)
Cronbach's alpha for the 'data' data-set
Items: 7
Sample units: 50
alpha: 0.231
```

As expected, Cronbach's alpha is 0.231 which is below 0.49, indicating an unacceptable reliability score. Researchers would interpret that either there's something wrong with the questionnaire, the population, or both.

Note that it doesn't make sense to use Cronbach's alpha for questions that don't have an overarching theme. I.e., if the Likert questions were all independent of each other (suppose participants were just signifying how strongly they feel about random hobbies) then you can expect there to be a small reliability.

## 4.3 ANOVA

ANOVA (analysis of variance) is used to test the differences between the means, not variances. This is used for when you are testing between more than two groups. I.e., you want to compare the CGPAs between students enrolled in the stats minor, the stats major, and the stats specialist, and see if they are significantly different from each other. Although you could use a t-test multiple times, the issue is that it usually leads to Type I errors (rejecting the null hypothesis when the null hypothesis is actually true). Hence, a new method was invented to deal with this problem: ANOVA.

**Assumptions:**

1. The samples between the groups are independent of each other and are randomly selected.
2. Each group has a normal population distribution.
3. Each group has the same variance.

For our example, Let $\mu_1$ represent the mean CGPA for students enrolled in the stats minor, $\mu_2$ represent the mean CGPA for students enrolled in the stats major, and $\mu_3$ represent the mean CGPA for students enrolled in the stats specialist. For this example, we would perform the following hypothesis test:

(Null hypothesis) $H_0 : \mu_1 = \mu_2 = \mu_3$

(Alternative hypothesis) $H_A : \exists i, j \in \{1, 2, 3\}$ such that $\mu_i \neq \mu_j$

Now let's test out this hypothesis with some randomly generated data:

```
> data = data.frame(
+   CGPA = sample(seq(0.7, 4.0, by = 0.02), size = 50, replace = TRUE),
+   Group = sample(c("Spec", "Major", "Minor"), size = 50, replace = TRUE)
+ )
> model = lm(CGPA ~ Group, data = data)
> anova(model)
Analysis of Variance Table
Response: CGPA
          Df Sum Sq Mean Sq F value Pr(>F)
Group      2  0.843 0.42153  0.4641 0.6315
Residuals 47 42.686 0.90822
```

Since the data was randomly generated, the $p$-value being above the $\alpha = 0.05$ significance level is expected. But suppose there was a significant difference; naturally, the question would be where the difference lies. For example, there could be cases where there is a difference between the statistics minors versus the majors, there is a difference between the statistics minors versus the specialists, but no difference between the statistics majors and the specialists. For cases like these, post-hoc tests are employed.

However, post-hoc tests are complicated. More advanced types of ANOVA will be covered in STA305. For now, it is sufficient to compute means or use visuals such as a box-plot to demonstrate the differences between the groups, if there are any.

### 4.3.1 Alternatives for Failing Assumptions

Now suppose the normality assumption is violated. Fortunately, there's the non-parametric version of one-way ANOVA called the Kruskal–Wallis test. We can still perform this in R:

```
> data = data.frame(
+   CGPA = sample(seq(0.7, 4.0, by = 0.02), size = 50, replace = TRUE),
+   Group = sample(c("Spec", "Major", "Minor"), size = 50, replace = TRUE)
+ )
> kruskal.test(CGPA ~ Group, data = data)
Kruskal-Wallis rank sum test
data:  CGPA by Group
Kruskal-Wallis chi-squared = 14.774, df = 2, p-value = 0.0006193
```

Here, it seems as though there's a difference between the cGPA of those enrolled in the specialist, major, and minor programs at the $\alpha = 0.05$ level! (I regenerated the random data, hence the result.) If you want to try to make directional conclusions, Let's look at the means here:

```
> tapply(data$CGPA, data$Group, mean)
   Major    Minor     Spec
2.390476 3.004286 1.585333
```

Wow, these specialists are doing awful! It must be because they have to take STA348.

## 4.4 Multiple Linear Regression

In linear regression, there is one quantitative explanatory variable predicting one quantitative response variable. In multiple linear regression, there are multiple quantitative explanatory variables predicting the singular quantitative response variable. (Now, of course there are multiple variations where you can combine categorical explanatory variables and have multiple response variables, but we'll not get into those details here.)

The assumptions for multiple linear regression are similar to linear regression, but heavily extended:

1. **Linearity:** the relationship between the explanatory variables and the response variable is linear. (You don't really need to verify this one, but sometimes linear regression is not the best way to model a response variable from explanatory variables.)
2. **Independence of errors:** the response variable is independent of the residuals. (If you plot the residuals and notice a pattern, this would imply you should use a different model.)
3. **Normality of errors:** the residuals must be approximately normally distributed.
4. **Homogeneity of variances amongst errors:** the variance of the residuals are the same for all explanatory variables.
5. **No multi-collinearity:** the independent variables are not too highly correlated with each other.

For example, suppose you are trying to predict the numbers of hours a student will sleep in a day. There are multiple variables to consider, such as their cGPA, time spent studying, time spent partying, etc. Consider the following make-shift data:

```
> data = data.frame(
+   Sleep = sample(seq(3,9,0.25), size = 50, replace = TRUE),
+   CGPA = sample(seq(1,4,0.05), size = 50, replace = TRUE),
+   TimeStudy = rnorm(50, mean = 10, sd = 2.5),
+   TimeGame = rnorm(50, mean = 5, sd = 1)
+ )
> model = lm(Sleep ~ CGPA + TimeStudy + TimeGame, data = data)
> summary(model)
Call:
lm(formula = Sleep ~ CGPA + TimeStudy + TimeGame, data = data)
Residuals:
    Min      1Q  Median      3Q     Max
-3.5515 -1.2877 -0.0211  1.3508  2.8107
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.43532    1.87870   2.361   0.0225 *
CGPA         0.15323    0.34546   0.444   0.6594
TimeStudy    0.04011    0.10270   0.391   0.6979
TimeGame     0.19931    0.23413   0.851   0.3990
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.815 on 46 degrees of freedom
Multiple R-squared:  0.02118,Adjusted R-squared:  -0.04266
F-statistic: 0.3318 on 3 and 46 DF,  p-value: 0.8024
```

Few things to note:

- For the coefficient of determination ($R^2$), you check multiple R-squared, not adjusted R-squared.
- The F-statistic and $p$-value at the end is testing for significance of the entire model. Since we have a $p$-value of 0.8024, this implies that time spent sleeping per day is not associated with cGPA, time spent studying per day, and time spent gaming per day.

- For multiple linear regression, R automatically finds the coefficients by minimizing the residual sum of squares. (You learn this more in detail in STA302/STA314). As a result, $R^2$ (the coefficient of determination) will always increase as you include more variables. However, it is also not good to overuse irrelevant variables due to the issues with over-fitting. (Over-fitting is discussed in STA314 so we'll not go into depth.)
- To check whether any of the explanatory variables are significant, check the value under the "$\Pr(>|t|)$" column of the coefficients. Note that since this data was randomly generated, we obviously won't see any significant effects.
- Naturally, you may be thinking of the optimal way to pick explanatory variables. This is taught in statistical learning (STA314). We will not go into detail here. I would obviously prefer a model with all significant variables, but the art of model selection isn't trivial. That being said, if you're taking STA314 you're allowed (and even encouraged) to use those techniques. If you haven't taken STA314, then just pick variables you felt would be intuitive as predictors.

Note that sometimes there are interaction effects; i.e., perhaps time spent studying per day is related to time spent gaming per day. Hence, you would use a different symbol (* instead of +):

```
> model2 = lm(Sleep ~ CGPA + TimeStudy * TimeGame, data = data)
> summary(model2)
Call:
lm(formula = Sleep ~ CGPA + TimeStudy * TimeGame, data = data)
Residuals:
    Min      1Q  Median      3Q     Max
-3.2459 -1.0709  0.2011  1.1137  2.6078
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -5.93835    4.25110  -1.397  0.16930
CGPA               0.23804    0.32586   0.731  0.46886
TimeStudy          1.21053    0.44693   2.709  0.00952 **
TimeGame           2.23397    0.78985   2.828  0.00696 **
TimeStudy:TimeGame -0.23384    0.08719  -2.682  0.01020 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.704 on 45 degrees of freedom
Multiple R-squared:  0.1561,Adjusted R-squared:  0.08106
F-statistic: 2.081 on 4 and 45 DF,  p-value: 0.09913
```

If you didn't care about the individual effects of TimeStudy and TimeGame, you would use ":" instead of "*".

## 4.5 Logistic Regression

In logistic regression, the explanatory variable is quantitative and the response is qualitative and binary (yes/no, success/failure, etc.) If you want to know some of the intuition/theory, there's a great StatQuest video about it on Youtube.

At first this seems unintuitive for how an explanatory variable could be quantitative but for the response to be binary. Imagine the following scenario: employee's preferences for hybrid verses complete remote work could depend on commute time. In general, I would expect the longer the commute, the higher preference for remote work.

The assumptions for logistic regression are as follows:

1. The data was randomly sampled and responses are independent from each other.
2. **No multi-collinearity:** the independent variables are not too highly correlated with each other.
3. Linear relationship between independent variables and log odds (logit). Note that this is NOT the same as saying there's a linear relationship between the explanatory and response variable. Most online tutorials recommend the Box-Tidwell Transformation method, however I've experienced a plethora of bugs/issues with the function in R (and there are also issues with just relying on the Box-Tidwell transformation.) For the course project, I am fine with you omitting this assumption.

Note that in R, for the logistic regression model to work you need to use values "0" and "1". This is why I suggest using a code book; so you'd be able to remember what "0" and "1" mean exactly. For now we can pretend in this makeshift example that "0" means remote and "1" means hybrid.

```
> data = data.frame(
+   Preference = sample(c(0, 1), size = 50, replace = TRUE),
+   CommuteTime = sample(1:120, size = 50, replace = TRUE)
+ )
> log_model = glm(Preference ~ CommuteTime, data = data, family = "binomial")
> summary(log_model)
Call:
glm(formula = Preference ~ CommuteTime, family = "binomial",
    data = data)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.370693   0.571206   0.649    0.516
CommuteTime -0.009395   0.008807  -1.067    0.286
```

As expected, since the data was uniformly randomly generated, the $p$-value of commute time is 0.286, which is above the $\alpha = 0.05$ significance level, meaning that commute time is not a predictor (or related) for whether someone prefers hybrid or remote work. That being said, if the data wasn't randomly generated then I would expect commute time to be an indicator.

But suppose that the results were significant for a second, in which we would pay attention to the value of the estimate (-0.009395). Without getting into too much theory, note that the linear model will provide you with the log odds. Hence to get the odds, we need to take the exponential (recall that $x = e^{\log(x)} = \log(e^x)$). Odds is another way to represent or express probabilities that is common in casinos and card games. I personally find it less intuitive, so we will not be going into detail about how odds work. Instead we'll address the exact translation from odds to probability:

$$\text{odds} = \frac{\text{probability}}{1 - \text{probability}} \qquad \text{probability} = \frac{\text{odds}}{1 + \text{odds}}$$

Hence, suppose that someone only commutes for 30 minutes. Using the estimate we will obtain the following calculation:

```
> odds = exp(0.370693 - 0.009395*30)
> prob = odds / (1 + odds)
> prob
[1] 0.5221962
```

Given that "0" means remote, this tells us that the probability that someone wants to work hybrid given their commute time is 30 minutes is approximately 52.22%.

There is obviously multiple logistic regression as well. Consider the following output:

```
> summary(log_model2)
Call:
glm(formula = Preference ~ NumKids + Salary + CommuteTime, family = "binomial",
    data = data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.208e+00  2.972e+00  -0.743    0.458
NumKids      2.629e-01  2.185e-01   1.204    0.229
Salary       2.458e-05  2.834e-05   0.867    0.386
CommuteTime -1.003e-02  1.304e-02  -0.769    0.442
```

We call this model "log_model2". Suppose we want to know the probability of someone preferring hybrid work given they have 2 kids, their salary is $ 100,000, and their commute time is 30 minutes. Then:

```
> intercept_coef = as.numeric(coefficients(log_model2)[1])
> numkids_coef = as.numeric(coefficients(log_model2)[2])
> salary_coef = as.numeric(coefficients(log_model2)[3])
> commutetime_coef = as.numeric(coefficients(log_model2)[4])
> odds = exp(intercept_coef + numkids_coef*2 + salary_coef*100000 + commutetime_coef*30)
> prob = odds / (1 + odds)
> prob
[1] 0.6166885
```

Hence this probability is approximately 61.67%.

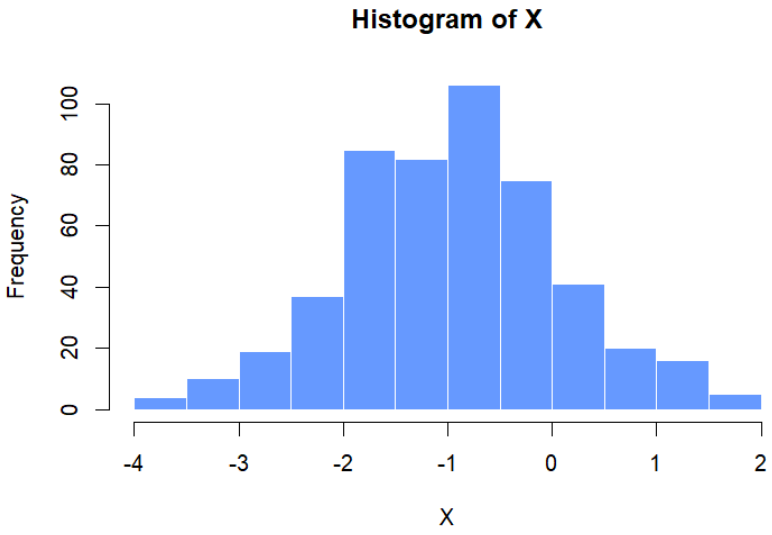
# 5 Data Visualization

This is a simple guide as to how you can make data visualizations for certain plots in base R. Note that if you want to level-up your data visualization game, you can learn how to use ggplot2. Nonetheless, with fine colour tuning, you can make excellent plots in base R! (Much better than matplotlib in Python...) There were even instances where people thought I was using ggplot2 when I was just using base R. ;)

## 5.1 Histogram

Histograms are good for the tests of the mean. Histograms visualize the distribution of continuous data by dividing the data range into intervals (bins) and displaying the frequency or count of data points within each bin. To make a histogram of just one data point, one can use the hist command in R:

```
> data = data.frame(
+   X = rnorm(500, mean = -1, sd = 1),
+   Y = rnorm(500, mean = 1, sd = 1)
+ )
> X_colour = "#6699FF"
> Y_colour = "#FF6666"
> hist(data$X, main = "Histogram of X", xlab = "X", col = X_colour, border = "white")
```



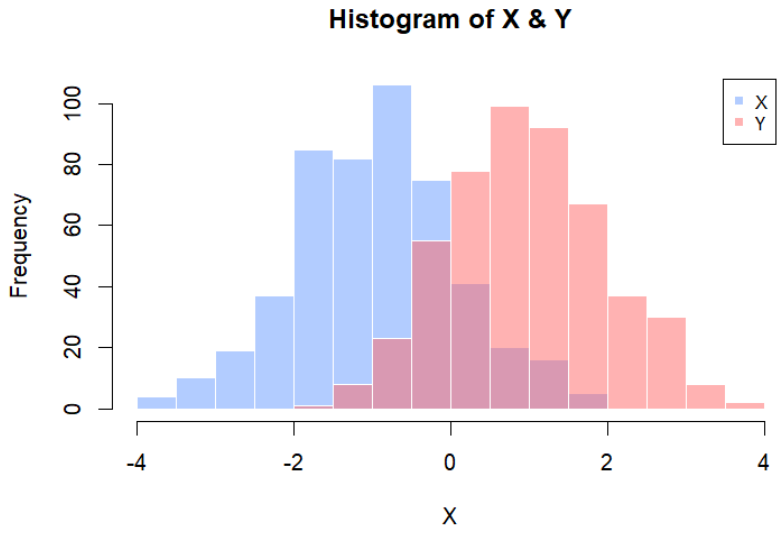

Some of the commands I've used are:

- "main": the title of the plot.
- "xlab": the x-axis label. Note that if you want to change the y-axis, you can use "ylab".
- "col": the colour of the bars.
- "border": the colour of the boarder of the bars. From personal experience, a white (or non-existent) boarder on base R looks great.

To make a histogram of two data points (which can be used to compare the distributions between two quantitative variables), you can make two histograms and include the "add = TRUE" command to the second one. You should also extend the x-axis for the first histogram to make space for the second histogram (this was done by changing the "xlim" values). You also need the colour to be slightly transparent. In R, to deal with transparency you need to use rgb for colour instead of a hex colour code or colour name. You change how transparent an item is using the "alpha" argument in "rgb()".

```
> X_rgb = col2rgb(X_colour)
> Y_rgb = col2rgb(Y_colour)
> X_hist_col = rgb(X_rgb[1]/255, X_rgb[2]/255, X_rgb[3]/255, alpha = 0.5)
> Y_hist_col = rgb(Y_rgb[1]/255, Y_rgb[2]/255, Y_rgb[3]/255, alpha = 0.5)
> x_axis_max = ceiling(max(data$X, data$Y))
> x_axis_min = floor(min(data$X, data$Y))
```

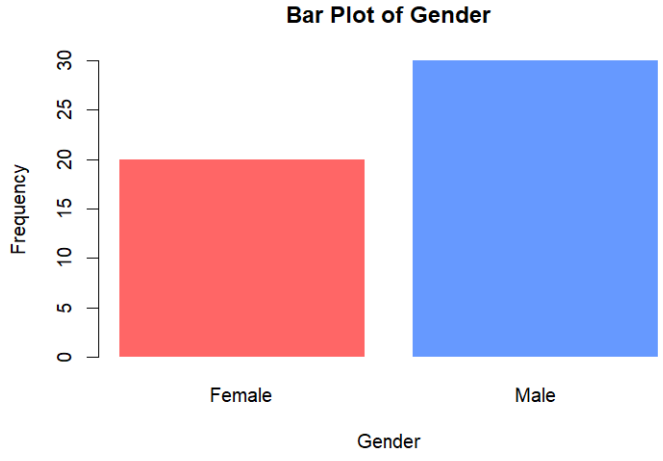
```
> hist(data$X, main = "Histogram of X & Y", xlab = "X", xlim = c(x_axis_min, x_axis_max),
+ col = X_hist_col, border = "white")
> hist(data$Y, col = Y_hist_col, border = "white", add = TRUE)
> legend("topright", legend = c("X", "Y"), pch = 15, col = c(X_hist_col, Y_hist_col),
+ inset = 0.02, cex = 0.8)
```




## 5.2 Bar plot

This is good for comparing categories or groups. Each bar represents a category, and the height (or length) of the bar corresponds to the value or frequency of that category. In R, you may use the barplot function.

```
> data = data.frame(
+   Gender = sample(c("M", "F"), size = 50, replace = TRUE, prob = c(0.7, 0.3)),
+   Program = sample(c("Stats", "NoStats"), size = 50, replace = TRUE, prob = c(0.8, 0.2))
+ )
>
> barplot(table(data$Gender), main="Bar Plot of Gender",
+         xlab="Gender", ylab="Frequency",
+         border=NA, col=c("#FF6666", "#6699FF"),
+         names.arg = c("Female", "Male"))
```




Few things to note:

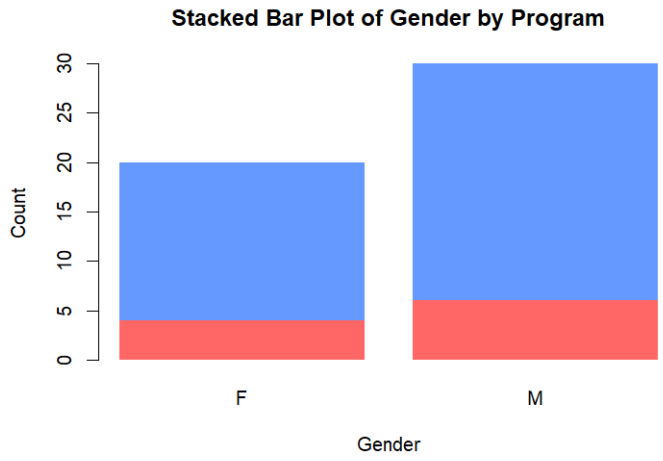
- The order of the bars are determined by "table()", which usually does it by alphabetical order. "names.arg" is to just rename the categorical variables "F" to "Female" and "M" to "Male".

- "main": the title of the plot.
- "xlab": the x-axis label. Similarly, "ylab": the y-axis label.
- "col": the colour of the bars; the order of the colour corresponds to the order of the bars.

### 5.2.1 Stacked Bar Plot

This is good for the two-sample proportion test or the test of independence of categorical variables. Again, using the same function but with some alternations to the table, we can achieve a stacked bar plot:

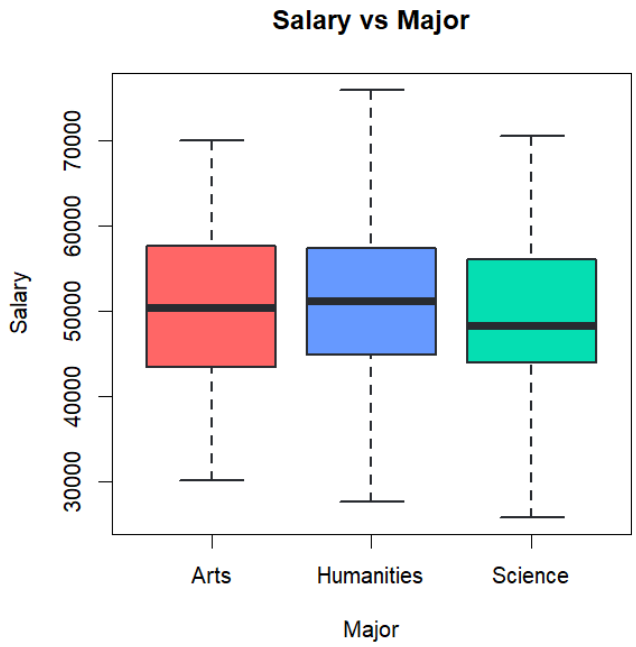
```
> barplot(table_data, main = "Stacked Bar Plot of Gender by Program",
+         xlab = "Gender", ylab = "Count", border = NA,
+         col = c("#FF6666", "#6699FF"))
> legend("topleft", legend = c("Statistics Major", "Non Statistics Major"), pch = 15,
+        col = c("#FF6666", "#6699FF"), inset = 0.02, cex = 0.8)
```




## 5.3 Box plot (or Box and Whisker Plots)

This is good for the two-sample test of the means and ANOVA. Box plots provide a visual summary of the minimum, first quartile, median, third quartile, and maximum values. This can be done using the boxplot option:

```
> data = data.frame(
+   Major = sample(c("Science", "Arts", "Humanities"), size = 200, replace = TRUE),
+   Salary = rnorm(200, mean = 50000, sd = 10000)
+ )
> boxplot(Salary ~ Major, data=data,
+         main = "Salary vs Major",
+         col = c("#FF6666", "#6699FF", "#05DEB2"),
+         border = c("#282B30"), lwd = 2)
```



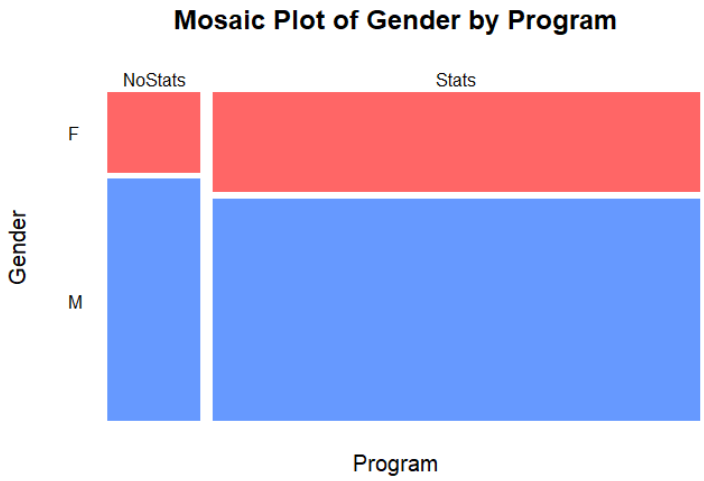

Few things to note:

- In the first argument ($Y \sim X$), the first entry $Y$ should be the quantitative variable and the second should be the qualitative variable $X$.
- "main": the title of the plot.
- "xlab": the x-axis label. Similarly, "ylab": the y-axis label.
- "col": the colour of the boxes; the order of the colour corresponds to the order of the boxes.
- "border": the colour of the border of the boxes. Something I learned when I took art classes that it's better to use colours close to black instead of just black (which is what I did.)
- "lwd": the thickness of the border of the boxes as well as the median line.

## 5.4 Mosaic Plot

This is good for the test of independence between categorical variables. They display the data in a matrix format where the size of each tile represents the proportion or frequency of the corresponding combination of categories. This can be done using the mosaicplot function.

```
> data = data.frame(
+   Gender = sample(c("M", "F"), size = 50, replace = TRUE, prob = c(0.7, 0.3)),
+   Program = sample(c("Stats", "NoStats"), size = 50, replace = TRUE, prob = c(0.8, 0.2))
+ )
> table_data = table(data$Program, data$Gender)
> mosaicplot(table_data,
+            main = "Mosaic Plot of Gender by Program",
+            xlab = "Program", ylab = "Gender",
+            col = c("#FF6666", "#6699FF"), border = NA,
+            las = 1, cex.axis = 0.8)
```



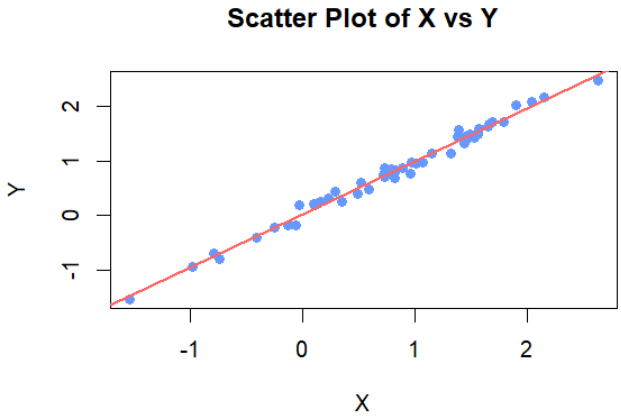

Few things to note:

- "main": the title of the plot.
- "xlab": the x-axis label. Similarly, "ylab": the y-axis label.
- "col": the colour of the rectangles; corresponds to the first item in the "table" argument (in this case, I had program over gender.)
- "border": the colour of the border of the rectangles. In my opinion, it looks better without a boarder!

## 5.5 Scatter Plot

This is good if you're using simple linear regression or just a test for correlation. Each point on the plot represents an observation, with its position determined by the values of the two variables. This can easily be done using the plot function in R. To add the linear regression line, it is suggested to add abline.

```
> X = rnorm(50, mean = 1, sd = 1)
> Y = X + rnorm(50, mean = 0, sd = 0.1)
> model = lm(Y~X)
> plot(X, Y, main = "Scatter Plot of X vs Y", pch = 19, col = "#6699FF")
> abline(model, lwd = 2, col = "#FF6666")
```

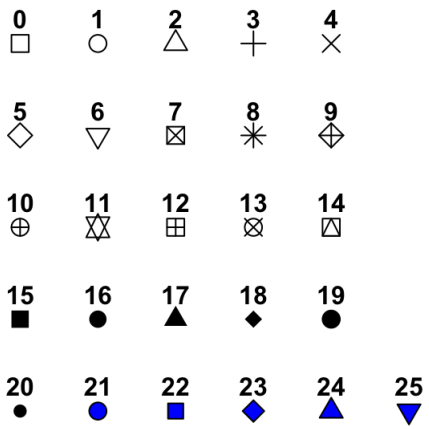



Few things to note:

- "main": the title of the plot.
- "pch": the different plotting shapes available in R. There are 26 in total.

- "col": if using pch from 0 to 20, it determines the colour of the plot symbol. If using 21 to 25, it is the border. Use "bg" for the filling. (In abline(), it is the colour of the linear regression line.)
- "lwd": the width of the linear regression line.

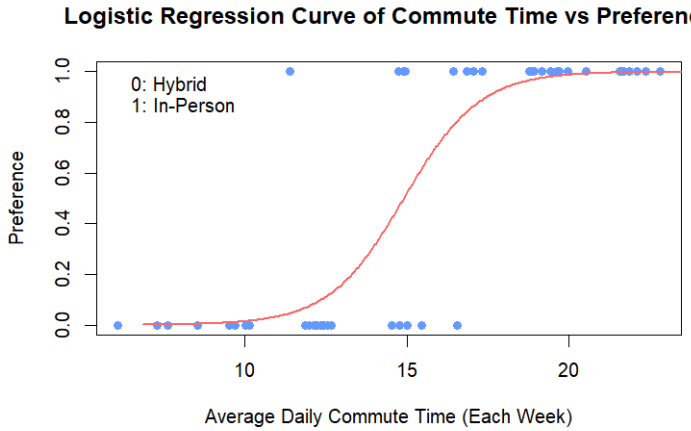
Below are the different types of plotting symbols (image sourced from sthda).




## 5.6 Logistic Regression Curve

As the name implies, this is strictly for logistic regression. The method for creating the plot (in base R) is quite similar to the scatter plot method. See the following code:

```
> data = data.frame(
+   Preference = c(rep(0, 20), sample(c(0, 1), size = 10, replace = TRUE), rep(1, 20)),
+   CommuteTime = c(rnorm(20, 10, 2), rnorm(10, 15, 2), rnorm(20, 20, 2))
+ )
> log_model = glm(Preference ~ CommuteTime, data = data, family = "binomial")
> newdata = data.frame(CommuteTime=seq(min(CommuteTime), max(CommuteTime),len=500))
> newdata$Preference = predict(log_model, newdata, type="response")
> plot(Preference ~ CommuteTime, data = data,
+      main = "Logistic Regression Curve of Commute Time vs Preference",
+      xlab = "Average Daily Commute Time (Each Week)", ylab = "Preference",
+      pch = 19, col="#6699FF")
> lines(Preference ~ CommuteTime, data = newdata, lwd=2, col = "#FF6666")
> legend("topleft", legend = c("0: Hybrid", "1: In-Person"), border = "white", bty = "n")
```




Few things to note:

- "main": the title of the plot.
- "xlab": the x-axis label. Similarly, "ylab": the y-axis label.
- "pch": the different plotting shapes available in R. There are 26 in total. Details regarding this is discussed in the scatter plot section.

- "col": if using pch from 0 to 20, it determines the colour of the plot symbol. If using 21 to 25, it is the border. Use "bg" for the filling. (In line(), it is the colour of the logistic regression line.)
- "lwd": the width of the logistic regression line.
- I don't do this often, but in the legend I got rid of the border since in base R, there is spade made for the shape of a symbol.

# 6 End

Thank you for reading through this statistical cookbook and I hope you found this guide to be useful and at times, humourous. Please email annahuynh.ly@mail.utoronto.ca for typos and possible suggestions. You may follow me on GitHub and connect with me on Linkedin. I'm trying to get more GitHub followers than my ex (yes I am toxic.)