Normality Testing

Normality Testing for Validity Conditions

The validity conditions for our various tests are an important check to ensure that we are properly using the corresponding theory-based approaches. At their heart, many of these conditions are to establish that we are either drawing from a (roughly) normal population for our response variable. For our introductory statistics course, we frequently pose these conditions as "not strongly skewed" or other statements that leave some students unsatisfied by the lack of specificity in the condition. To this end, I wanted to take a little bit of time to discuss checking for normality in the distributions of samples of data.

```
library(tidyverse)
```

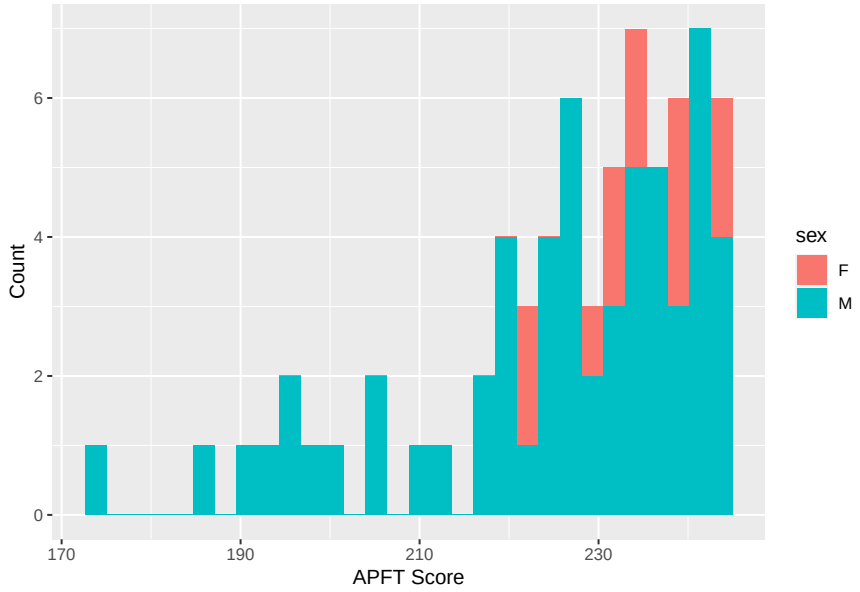
Two-sample t-test (Chapter 6)

Validity Conditions:

- The quantitative variable should have a symmetric distribution in both groups, or
- You should have at least 20 observation in each group and the sample distributions should not be strongly skewed.

In this example I'm interested to see if the average APFT is different between males and females. I'm going to take a subset of our data where the sample size of one group is too small (n = 12 for females) to demonstrate the issues with the central limit theorem when our validity conditions aren't met.

```
IOCT <- read_csv("IOCT_tab_data.csv")

subset <- IOCT %>%
  filter(APFT_Score < 245)%>%
  select(sex, APFT_Score)

subset %>%
  ggplot(aes(x = APFT_Score, fill = sex))+
  labs(x = "APFT Score", y = "Count")+
  geom_histogram()
```



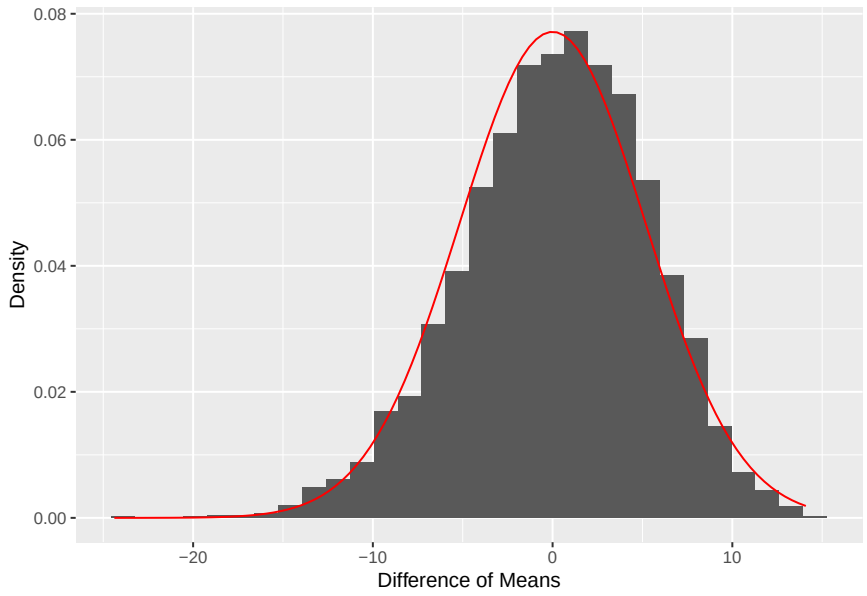
I'm satisfied this is about as skewed a distribution as I can get with this dataset and we know it doesn't meet the "20 in each group" part of the validity conditions. Let's first calculate the sample statistic (the difference between the average APFT scores of males and females in our subset).

```
males_subset <- subset %>%
  filter(sex == "M")

females_subset <- subset %>%
  filter(sex == "F")

sd_m = sd(males_subset$APFT_Score)
sd_f = sd(females_subset$APFT_Score)
n_m = nrow(males_subset)
n_f = nrow(females_subset)

#This is, admittedly, kind of a mess but I didn't want to use t.test()
#It is the Welch T-test DF equation
df = ((sd_m^2/n_m)+(sd_f^2/n_f))^2 /
  ((sd_m^4/(n_m^2*(n_m-1)))+(sd_f^4/(n_f^2*(n_f-1))))
```

Finally, let's build a simulated null distribution of simulated average differences based on randomly assigned "new" values for the sex of each cadet. Remember we do this because we are assuming the null hypothesis is true when building the null distribution (thus there is no association between sex and score so it shouldn't matter what sex the cadets are).

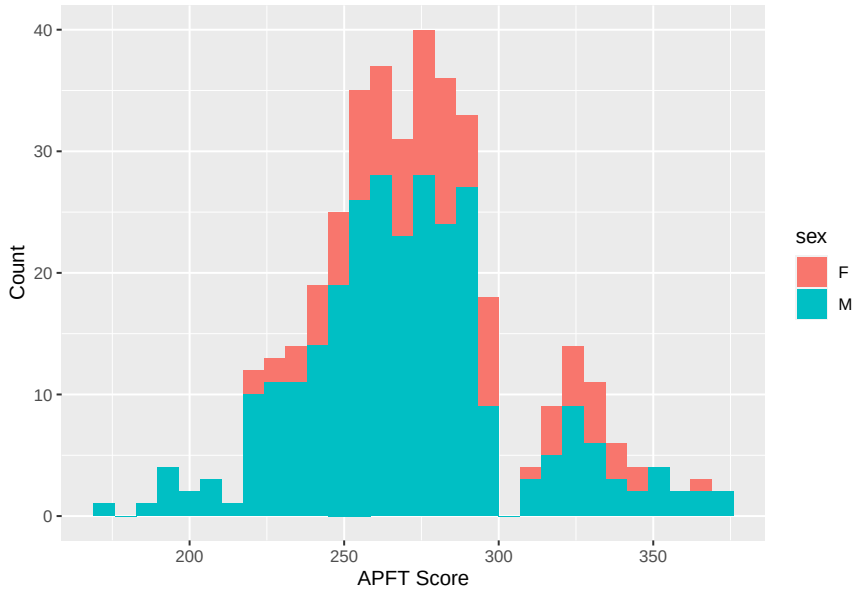
```
null_mean = 0

prop_male = n_m / (n_m + n_f)

sim_data = NULL

reps = 5000

for(i in 1:reps){

  trial_subset <- subset %>%
```

```
    mutate(New_Sex = sample(c('M','F'),
                            size = n(),
                            replace = TRUE,
                            prob = c(prop_male, 1-prop_male)))

  trial_means = trial_subset %>%
    group_by(New_Sex) %>%
    summarise(mean = mean(APFT_Score))

  trial_stat = trial_means[[2]][[1]] - trial_means[[2]][[2]]

  sim_data <- rbind(sim_data, as.data.frame(trial_stat))

}

#Standard deviation of our simulated trial statistics
stand_err = sd(sim_data$trial_stat)

sim_data %>%
  ggplot(aes(x = trial_stat)) +
  geom_histogram(aes(y = ..density..)) +
  labs(x = "Difference of Means", y = "Density") +
  stat_function(fun = dnorm,
    args = list(mean = null_mean, sd = stand_err),
    color = "red")
```



Clearly this isn't the world's worst fit becuase we were using a subset of the data that wasn't horrendously skewed in the first place; however, you will notice a slight skewness in the simulated null distribution. I'm going to repeat the process but now I'm going to use the entire dataset to ensure we meet the validity conditions. Here is what the distribution of our groups looks like now.

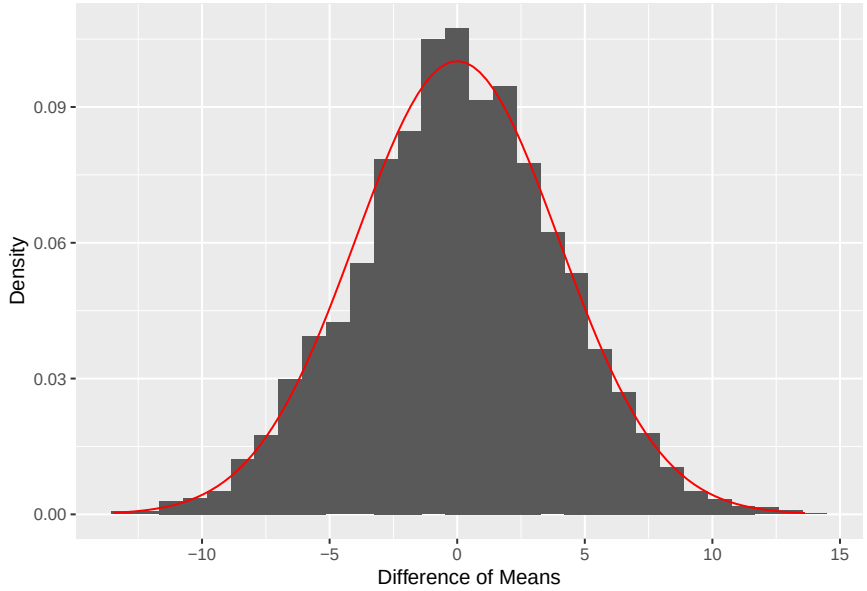
```
IOCT %>%
  ggplot(aes(x = APFT_Score, fill = sex))+
  labs(x = 'APFT Score', y = 'Count')+
  geom_histogram()
```



Now our simulated null distribution and normal distribution overlay looks like this:

```
males_IOCT <- IOCT %>%
  filter(sex == "M")

females_IOCT <- IOCT %>%
  filter(sex == "F")

sd_m = sd(males_IOCT$APFT_Score)
sd_f = sd(females_IOCT$APFT_Score)
n_m = nrow(males_IOCT)
n_f = nrow(females_IOCT)

df = ((sd_m^2/n_m)+(sd_f^2/n_f))^2 /
  ((sd_m^4/(n_m^2*(n_m-1)))+(sd_f^4/(n_f^2*(n_f-1))))

prop_male = n_m / (n_m + n_f)

sim_data = NULL

reps = 5000

for(i in 1:reps){

  trial_IOCT <- IOCT %>%
    mutate(New_Sex = sample(c('M','F'),
                            size = n(),
                            replace = TRUE,
                            prob = c(prop_male, 1-prop_male)))

  trial_means = trial_IOCT %>%
    group_by(New_Sex) %>%
    summarise(mean = mean(APFT_Score))
```

```
  trial_stat = trial_means[[2]][[1]] - trial_means[[2]][[2]]

  sim_data <- rbind(sim_data, as.data.frame(trial_stat))

}

#Standard deviation of our simulated trial statistics
stand_err = sd(sim_data$trial_stat)

sim_data %>%
  ggplot(aes(x = trial_stat)) +
  geom_histogram(aes(y = ..density..)) +
  labs(x = "Difference of Means", y = "Density") +
  stat_function(fun = dnorm,
    args = list(mean = null_mean, sd = stand_err),
    color = "red")
```



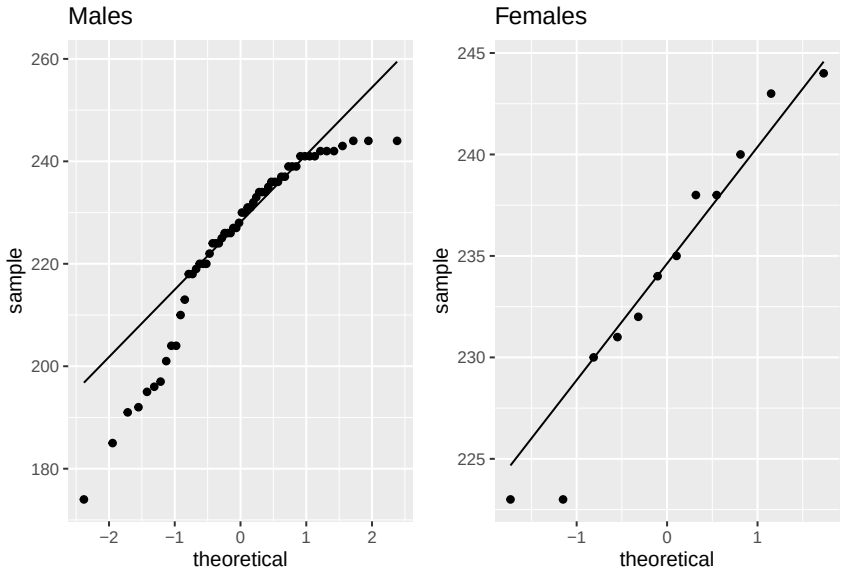
That's much better. So now let's take a look at some ways we judge whether our samples are "not strongly skewed" or otherwise distributed normally "enough" to utilize our theory based methods.

QQ Plot

One visual method is to check the QQ plot for each of our groups. Ideally we'd like the dots on the plot of follow the line. There will always be a little veering off of the line at the end but the "males" plot below is particularly bad. Also of note is the fact that having a small sample size, as in the case of the "female" plot, always makes it difficult to judge the distribution of data. Note: *Patchwork* library is only needed if you want to do sweet side-by-side plots. The "stat_qq" functions are included in ggplot.

```
library(patchwork)
maleQQ <- males_subset %>%
  ggplot(aes(sample = APFT_Score))+
  labs(title = "Males")+
  stat_qq()+
  stat_qq_line()
```

```
femaleQQ <- females_subset %>%
  ggplot(aes(sample = APFT_Score))+
  labs(title = "Females")+
  stat_qq()+
  stat_qq_line()

maleQQ + femaleQQ
```



The issue with this is we are still relying on a visual judgement call for whether or not our samples are normally distributed. Fortunately we have a few analytic tests that can be used for this so let's take a look at the Shapiro-Wilk test for normality.

Shapiro-Wilk Test

```
shapiro.test(males_subset$APFT_Score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  males_subset$APFT_Score
## W = 0.88355, p-value = 4.588e-05
```

```
shapiro.test(females_subset$APFT_Score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  females_subset$APFT_Score
## W = 0.94646, p-value = 0.5859
```

The null hypothesis of the Shapiro-Wilk test is that your sample is drawn from the normal distribution so a low p-value (typically "0.05" is used) means that we have significant evidence against the null hypothesis. So if we have a p-value less than 0.05 we should assess that our sample is likely not distributed normally enough to meet our validity conditions. Here we see that our p-value for the "male" subset is 4.588e-05 which provides significant evidence that our data is not normally distributed and the p-value for the "female" subset

is 0.5859 which doesn't not provide this evidence. The issue with the "female" subset is that we have such a low sample size, it's difficult to know if it's because the sample is normally distributed or it's because we just don't have enough to tell.

Residual Standard Error

One of the statistics that is available to us in the output of the linear model (should the problem call for one) is the residual standard error (RSE). If we are checking the validity conditions of a linear regression model, we need to ensure that our residuals are distributed normally. In this document we have discussed several ways of checking this normality condition but a relatively simple way is to use the "68-95-99 Rule" to see what proportion of our residuals fall within one, two, and three standard deviations of zero.

```
model = lm(IOCT$weight ~ IOCT$height)

summary(model)
```

```
##
## Call:
## lm(formula = IOCT$weight ~ IOCT$height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.583 -13.468   0.516  11.966  80.150
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -195.3394    18.2652  -10.70   <2e-16 ***
## IOCT$height    5.2665     0.2641   19.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.29 on 382 degrees of freedom
## Multiple R-squared:  0.5101, Adjusted R-squared:  0.5088
## F-statistic: 397.7 on 1 and 382 DF,  p-value: < 2.2e-16
```

```
RSE = summary(model)[[6]][[1]]

sd_count = as.data.frame(model$residuals) %>%
  mutate(sd_away = ceiling(abs(model$residuals / RSE))) %>%
  group_by(sd_away) %>%
  summarise(count = n())

paste("Within 1SD:", sd_count[1,2] / nrow(IOCT))
```

```
## [1] "Within 1SD: 0.669270833333333"
```

```
paste("Within 2SD:", (sd_count[1,2] + sd_count[2,2]) / nrow(IOCT))
```

```
## [1] "Within 2SD: 0.963541666666667"
```

```
paste("Within 3SD:", (sd_count[1,2] + sd_count[2,2] + sd_count[3,2]) / nrow(IOCT))
```

```
## [1] "Within 3SD: 0.994791666666667"
```

These values are very close to our "68-95-99 Rule" which leads me to believe that, for this linear model, our residuals are distributed normally.

There are certainly other normality tests but the purpose here is to provide some concrete analysis tools to,

hopefully, help alleviate concerns about judging whether or not sample are “strongly skewed.”