Introductions

- Name
- Year
- Make sure to sign in on the google form (linked here)

Hypothesis testing on real data

This section will deal with a data set of country-level statistics from this source with an explanation of the data encoding found here.

```
# Read in the data
countries <- read.csv("data/countries.csv", check.names = F)
```

1. Perform a formal *t*-test for the difference in GDP per capita (encoded as `mad_gdppc`) by democracy status (encoded as `br_dem`) and provide a 95% confidence interval for the difference. Comment on the assumptions of the test. Note that a formal hypothesis test should state the two hypotheses, state the test statistic (and degrees of freedom), state the p-value, and draw a conclusion in the context of the question.

Inference:

Assumptions:

```
# T test
t.test() # TODO

# Distribution of GDP by democracy status
boxplot(mad_gdppc~as.factor(br_dem), countries)

# Countries in each group
table(as.factor(countries$br_dem))
```

2. Perform a formal analysis of variance for the difference in GDP per capita (encoded as `mad_gdppc`) by type of democracy (encoded as `gol_inst`). Comment on the assumptions of the test.

Inference:

Assumptions:

```
summary(aov()) # TODO
boxplot(mad_gdppc~factor(gol_inst), countries)
by(countries$mad_gdppc, factor(countries$gol_inst), var,na.rm=T)/10^6
table(countries$gol_inst[!is.na(countries$mad_gdppc)])
```

3. Formally test whether there is a difference in the proportion of countries that are democracies (`br_dem`) by whether they are part of the British commonwealth (`br_cw`). Comment on the assumptions of the test.

Inference:

Assumptions:

```
prop.test(), correct = F) # TODO
table(factor(countries$br_dem),factor(countries$br_cw))
```

Manipulating new distributions

Let $T_n \sim t_n$. Find the following:

1. Distribution of T_n^2. Hint: Consider the story of the t distribution.
2. Distribution of T^{-2}

Bonferroni

Let $H_1, H_2, \ldots, H_n$ be a series of hypotheses, and let $p_1, p_2, \ldots, p_n$ be their corresponding p-values. Let m_0 be the number of true null hypotheses, and let α be the false discovery rate.

1. Show that the Bonferroni correction limits the family wise error rate to α. Hint: Bound the family-wise error rate using the first term of inclusion-exclusion.
2. Find the exact family-wise error rate if $P(p_i \leq \alpha/m | p_j \leq \alpha/m) = \rho P(p_j \leq \alpha/m), P(p_i \leq \alpha/m | p_j \leq \alpha/m, p_k \leq \alpha/m) = \rho P(p_j \leq \alpha/m, p_k \leq \alpha/m)$, etc. Hint: use the full inclusion-exclusion.
3. Write a chunk of code to calculate the family-wise error rate if $m_0 = m = 40$ (what does this mean in context?), $\rho = 0.05$ (what does this mean in context?), and $\alpha = 0.05$. Comment on the rate when the errors are perfectly correlated, when the errors are mutually exclusive, and when when the errors are independent.

Interpretation:

```
m_0 = 40
m = m_0
alpha = 0.05
rho = 1
# TODO: Write code to calculate the family-wise error rate

# Bonferroni correction
print(alpha/m)
```

```
## [1] 0.00125
```

Simulations

1. Let $X_1, X_2, \ldots, X_n \sim \mathcal{N}(10, 3^2)$. Then, let $X_{i,1} = X_i + \epsilon_{i,1}$ and $X_{i,2} = X_i + \beta + \epsilon_{i,2}$ with $\epsilon_{i,j} \sim \mathcal{N}(0, 3^2)$. What is the best way to test whether $\beta = 0$? Let $n = 10$, $\beta = 1$, and the number of simulations be 1000.

```
library(ggplot2)

n = 10
beta = 1
nsim = 1000

paired_vec = vector(length = nsim)
unpaired_vec = vector(length = nsim)
for (i in 1:nsim) {
  x = rnorm(n, 10, 3)
  x_1 = x + rnorm(n, 5, 3)
  x_2 = x + beta + rnorm(n, 0, 3)
  paired_vec[i] = # TODO: Get the paired t-test statistic
  unpaired_vec[i] = # TODO: Get the unpaired t-test statistic
}

df = data.frame("t-stat" = c(paired_vec, unpaired_vec),
                "paired" = c(rep("Paired", nsim), rep("Unpaired", nsim)), check.names = F)

ggplot(df, aes(x=`t-stat`, fill=paired)) +
  geom_histogram(alpha=0.4, position="identity") +
  theme_bw()
```

2. Let $\beta_1 = 1, \beta_2 = 2, \dots, \beta_{n_1} = n_1$. Let $X_{i,j} = \beta_i + \epsilon_{i,j}$ for $j = 1$ to n_2 with $\epsilon_{i,j} \sim \mathcal{N}(0, 5^2)$. What is the best way to test whether $\beta_1 = \beta_2 = \cdots = \beta_{n_1}$? If we break each group up into n_3 subgroups, what happens to our power? (Think of this as accidentally choosing too many categories.) Let $n_1 = 10$, $n_2 = 30$, $n_3 = 6$, and the number of simulations be 1000.

```
nsim = 1000
n_1 = 10
n_2 = 30
n_3 = 6
betas = 1:n_1

f_correct = vector(length = nsim)
f_incorrect = vector(length = nsim)

for (i in 1:nsim) {
  df = data.frame(matrix(nrow = 0, ncol = 0))
  for (beta in betas) {
    x = beta + rnorm(n_2, 0, 5)
    df = rbind(df, cbind(x, beta))
  }
  df$beta <- as.factor(df$beta)
  f_correct[i] = # TODO: Get the F statistic using the correct groups
  df$beta = as.numeric(as.character(df$beta))
  df$beta = (df$beta - 1) * n_3 + rep(rep(0:(n_3-1), n_2/n_3), n_1)
  df$beta <- as.factor(df$beta)
  f_incorrect[i] = # TODO: Get the F statistic using the incorrect groups
}

df = data.frame("f-stat" = c(f_correct, f_incorrect),
                "split" = c(rep("Correct", nsim), rep("Incorrect", nsim)), check.names = F)
```

```r
ggplot(df, aes(x=`f-stat`, fill=split)) +
  geom_histogram(alpha=0.4, position="identity") +
  theme_bw() + xlim(0, 30)
```

3. Let $X_i \sim \mathcal{N}(0, 1)$ for i from 1 to n. Show that the p-values of the test $H_0 : \mu = 0$, $H_a : \mu \neq 0$ are uniformly distributed. Let $X_i \sim \text{Exp}(1) - 1$ for i from 1 to n. Show that the p-values of the test $H_0 : \mu = 0$, $H_a : \mu \neq 0$ are not uniformly distributed for small n. Does the skew of the distribution make sense? Compare this to a large n (~100).

```r
par(mfrow=c(1,2))
set.seed(139)
n = 5
nsim = 100000

normal = vector(length = nsim)
for (i in 1:nsim) {
  x = # TODO: Simulate normals
  normal[i] = # TODO: Get t-test p-values
}

hist(normal)
mean(normal<0.05)

expo = vector(length = nsim)
for (i in 1:nsim) {
  x = # TODO: Simulate exponentials
  expo[i] = # TODO: Get t-test p-values
}

hist(expo)
mean(expo<0.05)
```