Announcements

- Make sure to sign in on the google form (linked here)
- Midterm October 11
- Review session October 10 at 11 am

The case of the disappearing significance

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

```
countries <- read.csv("data/countries.csv")
```

A few useful columns:

- `spi_ospi`: Overall social progress index on 0-100 scale
- `mad_gdppc`: GDP per capita
- `wdi_expedu`: Government expenditure on education as percent of GDP
- `wvs_fight`: Proportion of people who answered affirmatively to "Of course, we all hope that there will not be another war, but if it were to come to that, would you be willing to fight for your country?"
- `wvs_jabribe`: Average response on scale of 1 (never justifiable) to 10 (always justifiable) to the question: "Do you think the following can be justified: someone accepting a bribe in the course of their duties."

1. Fit four linear models to predict a country's overall social progress index: one based on each variable. Interpret each result. Do these make sense?

```
# TODO: 4 models
```

2. Fit a linear model to predict a country's overall social progress index from GDP per capita and education expenditures. Find the correlation between the coefficients for GDP and education expenditures.

```
# TODO: fit model

# This is a pretty good general purpose command for getting the correlation of predictors
# Divides the variance covariance matrix by the standard deviation of the rows and columns
# to get the pairwise correlations
lm2_vcov <- vcov(lm2)
t(t(lm2_vcov/sqrt(diag(lm2_vcov))) /
    sqrt(diag(lm2_vcov)))[2:nrow(lm2_vcov), 2:nrow(lm2_vcov)]
```

3. Fit a linear model to predict a country's overall social progress index from all the variables we have considered so far and print the summary. What happens to the `wdi_expedu` significance? Print the pairwise correlation between the fit coefficients. What's explaining the change in `wdi_expedu` significance? How can you test this?

```
# TODO: Fit the model
# TODO: Find the coefficient correlation
```

4. Seychelles is an archipelagic country consisting of 115 islands in the Indian Ocean off the east coast of Africa. It has a GDP per capita and education expenditure percentage listed but no OSPI. Use the GDP per capita and education expenditures model to provide a 95% confidence and prediction interval for OSPI. What is each interval trying to capture?

```
# TODO: Confidence interval
# TODO: Prediction interval
```

Contrast test and limiting cases

Recall the setup for a contrast test: $H_0 : \vec{C}^T\vec{\beta} = \gamma_0$ vs. $H_a : \vec{C}^T\vec{\beta} \neq \gamma_0$. Under the null, the following random variable has a $t_{n-(p+1)}$ distribution.

$$T = \frac{\vec{C}^T\hat{\vec{\beta}} - \gamma_0}{\hat{\sigma}\sqrt{C^T(X^TX)^{-1}C}}$$

1. Name two situations in which we would take γ_0 to be 0.
2. Write a function that will take a linear model and two named vectors and run a contrast test for a difference between the response variable. Hint: `vcov(fit_lm)` is the same as $\hat{\sigma^2}(X^TX)^{-1}$ and `coef(fit_lm)` gets the fit coefficeints of the model.

```
contrast.test <- function(fit_lm, vec1, vec2) {
  # TODO: Write function
}
```

3. Perform a formal contrast test based on the GDP per capita plus education expenditures model to determine whether the mean OSPI for (mythical) countries like Seychelles is significantly different from the mean OSPI for (mythical) countries like Madagascar.

```
# Contrast vector
c("(Intercept)" = 1, unlist(countries[countries$cname == "Seychelles",
                                      c("mad_gdppc", "wdi_expedu")])) -
  c("(Intercept)" = 1, unlist(countries[countries$cname == "Madagascar",
                                        c("mad_gdppc", "wdi_expedu")]))
```

```
## (Intercept)   mad_gdppc  wdi_expedu
##     0.00000 28103.46021     1.59553
```

```
# TODO: Contrast test
```

4. Name and check two limiting cases to ensure the function works as intended.

-
-

```
# Make sure single predictor case collapses to t-test
contrast.test(lm2, c("(Intercept)" = 0, "mad_gdppc" = 1, "wdi_expedu" = 0),
              c("(Intercept)" = 0, "mad_gdppc" = 0, "wdi_expedu" = 0))
summary(lm2)$coefficients
```

```r
# Binary predictor t-test
lm5 <- lm(spi_ospi ~ as.factor(bmr_dem), countries)
summary(lm5)$coefficients
contrast.test(lm5, c("(Intercept)" = 0, "bmr_dem" = 1),
              c("(Intercept)" = 0, "bmr_dem" = 0))
t.test(spi_ospi ~ bmr_dem, countries)
```

P-values high and low

1. Explain intuitively how increasing the number of predictors in a model could reduce the significance of a particular predictor relative to a model with only that predictor. Find a pair of matrices X_1 and X_2 that illustrate this point. Hint: Recall that the standard error of $\hat{\beta}_i$ is $\hat{\sigma}\sqrt{C^T(X^TX)^{-1}C}$ where C is a one-hot encoded vector with the i^{th} entry as a 1.
2. Explain intuitively how increasing the number of predictors in a model could increase the significance of a particular predictor relative to a model with only that predictor.
3. Interpret the results of the following simulation:

We will generate data from the model $Y_i = \beta_1 X_{1,i} + \beta_2 X_{2,i} + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Let $\beta_1 = 1$, $\beta_2 = 2$, $\sigma^2 = 2^2$.

```r
library(MASS)
```

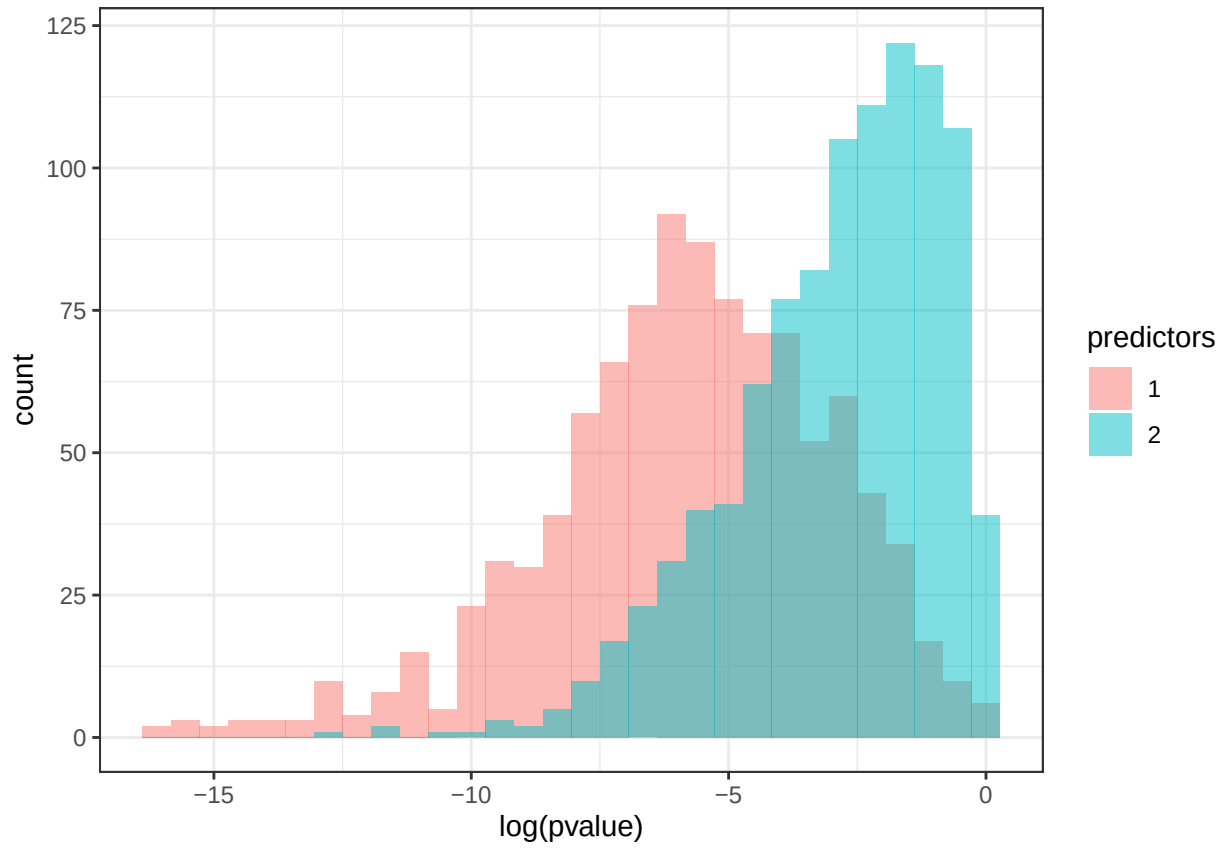
```
## Warning: package 'MASS' was built under R version 4.1.3
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

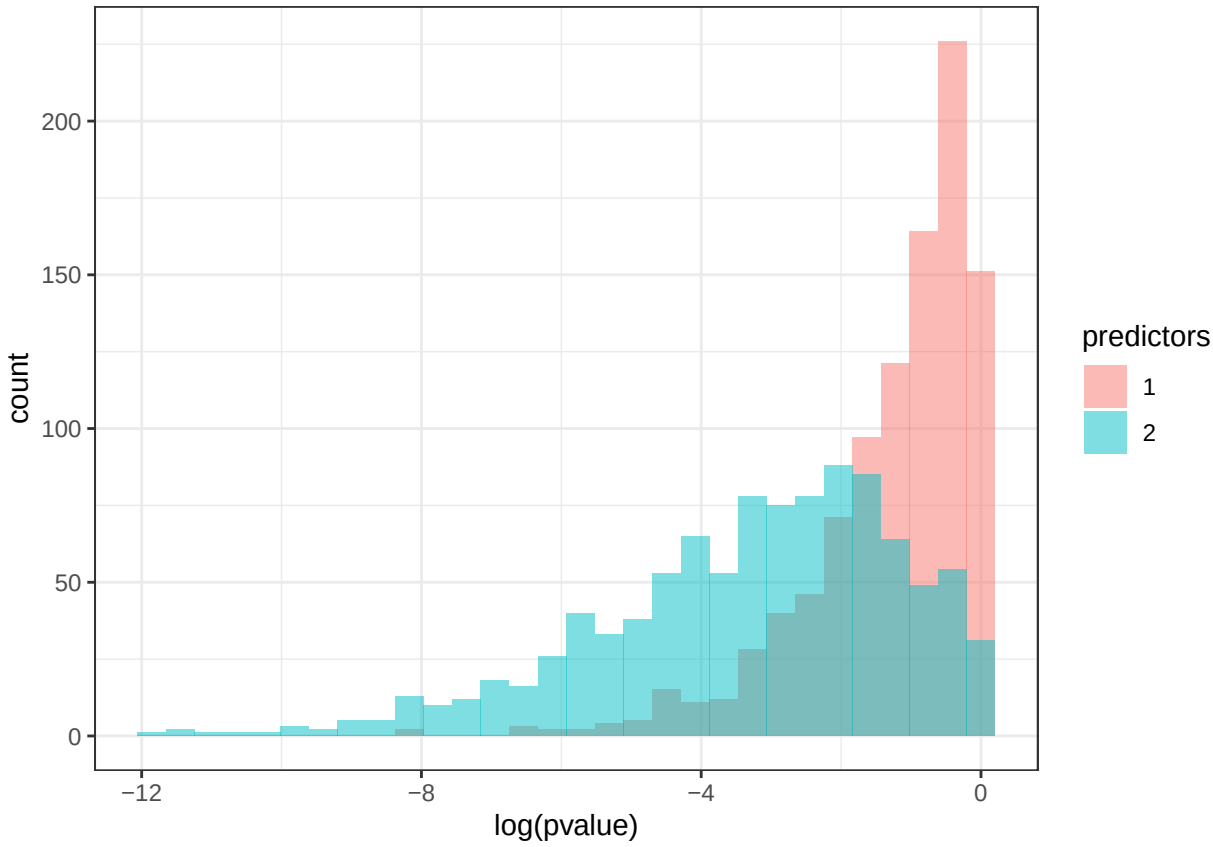
```r
nsims = 1000
n = 20
beta_1 = 1
beta_2 = 2
sigma = 2

# Correlation matrix for Xs
Sigma = cbind(c(1, 0.5), c(0.5, 1))

# Model with predictors for only X_1
pvals_single = vector(length = nsims)
for (i in 1:nsims) {
  x = mvrnorm(n = n, rep(0, 2), Sigma)
  y = x %*% c(beta_1, beta_2) + rnorm(n, 0, sigma)
  pvals_single[i] = summary(lm(y ~ x[,1]))$coefficients[2, 4] # Notice the difference
}

# Model with predictors for X_1 and X_2
pvals_double = vector(length = nsims)
for (i in 1:nsims) {
```

```
  x = mvrnorm(n = n, rep(0, 2), Sigma)
  y = x %*% c(beta_1, beta_2) + rnorm(n, 0, sigma)
  pvals_double[i] = summary(lm(y ~ x))$coefficients[2, 4] # Notice the difference
}

out_data = data.frame(predictors = as.factor(c(rep(1, nsims), rep(2, nsims))), pvalue = c(pvals_single,

ggplot(out_data, aes(x = log(pvalue), fill = predictors)) +
  geom_histogram(alpha=0.5, position="identity", bins=30) +
  theme_bw()
```




```
nsims = 1000
n = 20
beta_1 = 1
beta_2 = 2
sigma = 2

# Notice the very large variance of X_2
Sigma = cbind(c(1, 0), c(0, 10))

# Model with X_1 as the only predictor
pvals_single = vector(length = nsims)
for (i in 1:nsims) {
  x = mvrnorm(n = n, rep(0, 2), Sigma)
  y = x %*% c(beta_1, beta_2) + rnorm(n, 0, sigma)
```

```
  pvals_single[i] = summary(lm(y ~ x[,1]))$coefficients[2, 4] # Notice the difference
}

# Model with both X_1 and X_2 as predictors
pvals_double = vector(length = nsims)
for (i in 1:nsims) {
  x = mvrnorm(n = n, rep(0, 2), Sigma)
  y = x %*% c(beta_1, beta_2) + rnorm(n, 0, sigma)
  pvals_double[i] = summary(lm(y ~ x))$coefficients[2, 4] # Notice the difference
}

out_data = data.frame(predictors = as.factor(c(rep(1, nsims), rep(2, nsims))), pvalue = c(pvals_single,

ggplot(out_data, aes(x = log(pvalue), fill = predictors)) +
  geom_histogram(alpha=0.5, position="identity", bins = 30) +
  theme_bw()
```

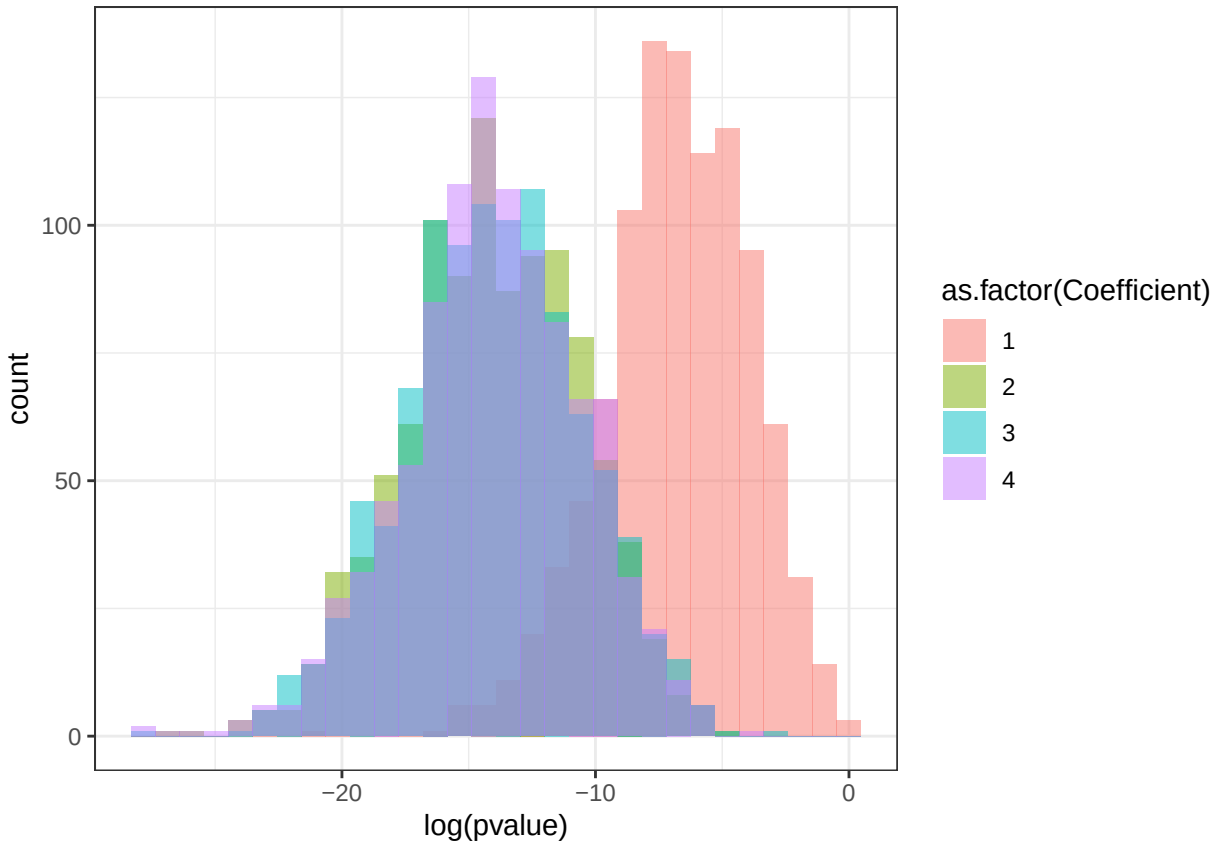


Winner takes all

1. Many times in the real world we have a situation where some particular predictor underlies many other predictors. The following simulation shows a situation in which there is an underlying variable `x_main` and three related predictors `x_2`, `x_3`, and `x_4`. First look at the histograms from when the coefficients and variances are the same for all the Xs. Compare a multiple regression model with a single-variable regression model for each coefficient and comment on what's happening. Then, by changing the values

of the beta coefficients and the variances of `x_main`, `x_2`, `x_3`, and `x_4`, explore what factors drive significance in a multiple regression model.

```
library(reshape2)

nsims = 1000
n = 20
beta_1 = 2
beta_2 = 2
beta_3 = 2
beta_4 = 2
sigma = 2

# Model with X_1 as the only predictor
pvals = matrix(nrow = nsims, ncol = 4)
for (i in 1:nsims) {
  x_main = rnorm(n, 0, 2)
  x_2 = rnorm(n, x_main, 2)
  x_3 = rnorm(n, x_main, 2)
  x_4 = rnorm(n, x_main, 2)
  y = beta_1 * x_main + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4 + rnorm(n, 0, sigma)
  pvals[i,] = summary(lm(y ~ x_main + x_2 + x_3 + x_4))$coefficients[2:5, 4]
}

out_data = melt(pvals)[,2:3]
colnames(out_data) <- c("Coefficient", "pvalue")

ggplot(out_data, aes(x = log(pvalue), fill = as.factor(Coefficient))) +
  geom_histogram(alpha=0.5, position="identity", bins = 30) +
  theme_bw()
```




```
library(reshape2)

nsims = 1000
n = 20
beta_1 = 2
beta_2 = 2
beta_3 = 2
beta_4 = 2
sigma = 2

# Model with X_1 as the only predictor
pvals = matrix(nrow = nsims, ncol = 4)
for (i in 1:nsims) {
  x_main = rnorm(n, 0, 2)
  x_2 = rnorm(n, x_main, 2)
  x_3 = rnorm(n, x_main, 2)
  x_4 = rnorm(n, x_main, 2)
  y = beta_1 * x_main + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4 + rnorm(n, 0, sigma)
  pvals[i,1] = summary(lm(y ~ x_main))$coefficients[2, 4]
  pvals[i,2] = summary(lm(y ~ x_2))$coefficients[2, 4]
  pvals[i,3] = summary(lm(y ~ x_3))$coefficients[2, 4]
  pvals[i,4] = summary(lm(y ~ x_4))$coefficients[2, 4]
}

out_data = melt(pvals)[,2:3]
colnames(out_data) <- c("Coefficient", "pvalue")
```

```
ggplot(out_data, aes(x = log(pvalue), fill = as.factor(Coefficient))) +
  geom_histogram(alpha=0.5, position="identity", bins = 30) +
  theme_bw()
```