Correlation and Linear Regression

In this lab, we will learn how to calculate covariance and correlation using base R and how to estimate and view linear regression models using OLS. Finally, we will use our estimated regression model to generate predictions. Let's get started!

For fun, we are going to evaluate Douglas Hibbs' Bread and Peace Model of U.S. presidential voting. I pulled the data together from several sources, including the Census Bureau, the FEC, Douglas Hibbs, and the Bureau of Economic Analysis.

First, let's load our packages and then get the data from GitHub:

```r
# Install pacman only if needed.
install.packages("pacman")

pacman::p_load(tidyverse)
```

```r
# load the data from my GitHub repository:
hibbs <- read_csv("https://raw.githubusercontent.com/bowendc/510_labs/main/hibbs_1960_2024.cs
```

First, we can create scatterplot between the change in real disposable income per capita (logged) and the incumbent party's vote share:

```r
ggplot(data = hibbs, aes(x = chng.lnrdipc.last, y = inc.voteshare)) +
    geom_point()
```

What do you think? Are change in real disposable logged income and presidential vote share still correlated, as Hibbs' work suggests they should be? If so, is the correlation positive or negative, weak or strong? We can evaluate your reasoning using Pearson's r correlation coefficient. The code below examines just change in logged real disposable income in the election year (late in the term).

```
# the cor() function will calculate the correlation coefficient, with Pearson's correlation c

# the pairwise.complete.obs argument will tell R to use all observations that are complete or
cor(x = hibbs$chng.lnrdipc.last, y = hibbs$inc.voteshare,
    use = "pairwise.complete.obs")
```

What happens if we look at change in real disposable income per capita early in the president's term?

```
cor(x = hibbs$chng.lnrdipc.early, y = hibbs$inc.voteshare,
    use = "pairwise.complete.obs")
```

How would you describe the difference between the relation between the economy and the presidential vote share for the incumbent in the election year verses earlier in the president's term?

```
# the lm() function conducts an OLS regression analysis.
# syntax: lm(y ~ x, data = df)
# model results can be named and stored to call up later
m1 <- lm(inc.voteshare ~ chng.lnrdipc.last,
            data = hibbs, # defines which data frame to use
            na.action = na.exclude) # excludes missing data

m1 # provides quick access to regression parameters

summary(m1)
```

Look at the first table above. The key information for now is presented in the "estimate" column. The value in the "estimate" column for `chng.lnrdipc.last` row is the *coefficient*, or slope, of the regression line: the among the incumbent vote share should go up for every one-unit increase in change in real disposable income (logged). Notice also that R has included an intercept term even though we didn't need to specify one in the function. That value is included in the `(Intercept)` row.

Now let's see what happens if we predict vote share from income change early in the president's term.

```
m2 <- lm(inc.voteshare ~ chng.lnrdipc.early, data = hibbs, na.action = na.exclude)

summary(m2)
```

What if we account for both early and late changes in income while holding constant the other part of Hibbs' model: fatalities in foreign wars?

```
m3 <- lm(inc.voteshare ~ chng.lnrdipc.last + chng.lnrdipc.early + fatalities, data = hibbs,

summary(m3)
```

Prediction and residuals

We can calculate $\hat{Y}$ by plugging in values for our predictor variables (`cng.lnrdipc.last`, `chng.lnrdipc.early`, and `fatalities`). For example, let's see what the predicted incumbent vote share would be for a president with 3.43% recent income change, 1.17% early change, and no fatalities. These were Biden's values heading into the 2024 election.

```
# the estimated slopes from your model are stored by R and accessible:

m3$coefficients["(Intercept)"] + m3$coefficients["chng.lnrdipc.last"]*3.43 + m3$coefficients
```

We can also generate predicted values for all existing observations in our data frame:

```
postest <- hibbs |> mutate(yhat.m3 = fitted(m3))
```

And if we know the predicted values and the original values of Y, then we can easily calculate the residuals (the difference between the actual values and the predicted values):

```
postest <- postest |> mutate(resid.m3 = resid(m3))
# of course, you could do this manually as well:
# postest <- postest |> mutate(resid.m3 = inc.voteshare - yhat.m3)

head(postest |> filter(elecyear>2007))
```