Week 5 Lab

AUTHOR
Moksha Sharma

Data Frames in R

A data frame is a two-dimensional, table-like of data structure in R. So far, the data structure with which we have been working has been vectors, and we can think of a data frame as a collection of equal-length vectors: each column is a different vector.

Recall that when we first started working with vectors, we discussed that you can only store on type of objects in one vector. In other words, a vector of characters could not also contain integers. Data frames, however, can store different classes of objects.

There are multiple ways of importing datasets into R so we can work with them. For example, if you have CSV file (CSV stands for "comma separated values", and it is basically a giant Excel sheet), you can use the `read.csv()` command. Of course, you have to be careful in specifying the path to the specific file (which can be a big pain to learn when you are starting with R!).

Various packages in `R` also contain data frames for illustrative purposes. We used the `iris` dataset in the first lab session, which is in the `tidyverse` package. For such datasets, we don't need to import them into our `R` session, because they are already loaded into R when we load the relevant package. As a result, we can just start working with them as needed.

The dataset we will used today can also be found in an `r` package (called `car`), but to maintain consistency with the problem set code, I uploaded it to Github as an `RData` file.

```
#run this code to load the dataset
data_location <- "https://github.com/UChicago-pol-methods/Intro

load(url(paste0(data_location, "prestige.Rdata")))
```

Let's look at what exactly we did here. First, we created an object to store

the link to the class Github page (specifically, the page where all the data have been uploaded). The other important thing to note here is the `load()` function, which allows us to read an external file into the current `r` workspace.

Finally, the file we read in is an `RData` file, which is a file format in `r` that allows us to store multiple objects in one file. I stored only the dataset in this file, but on the problem set, when you load in the `RData` file provided by Andy, you will see that it loads in not just the dataset but also 2 vectors. These vectors correspond to vectors that are in the dataset you use in the problem set. As they are separate objects in your `R` workspace for the problem set, you can use them directly without having to reference the dataset.

Prestige Dataset

Now, let's turn our attention to the dataset we are using today. It is called the "Prestige" dataset and is widely used for teaching linear regressions (in fact, in Linear Models, Mark Hansen Bobby Gulotty teach the original paper). The data is originally from 1970 and was used to measure how the perceived prestige of an occupation was related to how educated the employees in that industry were, how much money they made, and how many women worked in the industry.

1. **How many variables are in the dataset?**
2. **What does each row in the dataset represent?**
    1. (note to self: compare with the package version of the dataset)
3. **How many observations are there?**
4. **What information do we have about each observation?**

Linear Regression

A simple (perhaps too simplistic) way of defining a linear regression is as follows: when we linearly regress one variable (DV) on another (IV), we try to model the relationship between them as a straight line. In other words, we take the observed data and try to find a linear equation that best fits it ie minimises the sum of squared residuals .

Let's connect this to the class discussion of best linear predictor (BLP) towards the end of week 4. Just as it sounds, BLP is the linear predictor that best models the linear relationship between 2 variables. And since at least the 1800s, we have known that minimising the sum of the squared residuals - also know as the ordinary least squares (OLS) - gives us the BLP in the single predictor case. In other words, we obtain the BLP of *Y* given *X* by calculating the model parameters that minimise the sum of the squared residuals.

Let's try to appreciate visually why OLS is the BLP (iPad exercise).

Computation of BLP:

Let's start by looking at what the linear model of a relationship between 2 variables would be. We know that some part of the IV would be explained by the DV and some of it would be explained by factors other than the DV. So our model will look as follows:

$$Y = g(X) = a + bX$$

Of course, there are all sorts of things we don't know about this relationship, and to account for that, we add an error term. So the general form of an equation actually is $Y = a + bX + e$. However, we can't know what the error is, and by making a bunch of assumptions (that will be discussed in Linear Models), we can ignore it in the linear model that we estimate.

Here, $g(X)$ captures that *Y* is a function of *X*. What is the error in this context?

$$\begin{aligned} \epsilon &= Y - (a + bX) \\ \implies \epsilon^2 &= [Y - (a + bX)]^2 \\ \implies E[\epsilon^2] &= E[Y - (a + bX)]^2 \end{aligned}$$

Since we are trying to minimise this error, we try to find values of a and b that would make the error as small as possible. Does the equation below look familiar?

$$(\alpha, \beta) = \underset{(a,b)\in\mathbb{R}^2}{\arg\min}\, \mathrm{E}\left[(Y - (a + bX))^2\right]$$

To minimise a function, we differentiate it (with respect to what?) and set it to 0. Doing so here gives us:

- $\alpha = \mathrm{E}[Y] - \beta\mathrm{E}[X]$

- $\beta = \frac{\mathrm{Cov}[X,Y]}{\mathrm{V}[X]}$

Without getting into too much detail, let's try to intuitively understand these parameters.

Regression in R

There is a base R function that performs the OLS regression for us. Pick any variable from the Prestige dataset and use `lm()` to determine the linear relationship between your chosen variable and the perceived prestige of the occupations.

BEFORE you start, think about what the linear relationship should be. Then, get help from the function documentation to write your code.

```
# your code here
summary(lm(prestige~women, prestige))
```

```
Call:
lm(formula = prestige ~ women, data = prestige)

Residuals:
    Min      1Q  Median      3Q     Max 
-33.444 -12.391  -4.126  13.034  39.185 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 48.69300    2.30760  21.101   <2e-16 ***
women       -0.06417    0.05385  -1.192    0.236    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.17 on 100 degrees of freedom
Multiple R-squared:  0.014, Adjusted R-squared:  0.004143 
F-statistic:  1.42 on 1 and 100 DF,  p-value: 0.2362
```

Fill out the equation below to get the functional form of this model:

$$\text{prestige} = \beta_0 + \beta_1()$$

Interpretation:

2 notes here:

1. If the variables you are using for your regression already exist as vectors in your environment, you don't have to call on the dataset to reference them.
2. Or, you can also save the variables as vectors to run the regression.
3. Or, you can also subset them out of the dataset using $$ signs.

```
#your code here
summary(lm(prestige$prestige~prestige$women))
```

```
Call:
lm(formula = prestige$prestige ~ prestige$women)

Residuals:
    Min      1Q  Median      3Q     Max 
-33.444 -12.391  -4.126  13.034  39.185 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    48.69300    2.30760  21.101   <2e-16 ***
prestige$women -0.06417    0.05385  -1.192    0.236    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.17 on 100 degrees of freedom
Multiple R-squared:  0.014, Adjusted R-squared:  0.004143 
F-statistic:  1.42 on 1 and 100 DF,  p-value: 0.2362
```

Preview of Things to Come

Controls

Of course, prestige is affected by a lot of other factors, many of which are in the dataset. So we can regress "prestige" on multiple variables at the same time to see how they impact it individually. For instance, the presence of women does have a notable impact on the perceived prestige of the job, but it is confounded by the income associated with it. It is possible that if an industry is female-dominated, then that industry might be perceived as less prestigious. But industries dominated by women also tend to industries that pay less, so we would want to isolate the impact of women's presence by **controlling** for income. Such a regression is called

a **multiple linear regression.**

$$\text{prestige} = \beta_0 + \beta_1() + \beta_2()$$

```
#your code here
summary(lm(prestige~income + women, prestige))
```

```
Call:
lm(formula = prestige ~ income + women, data = prestige)

Residuals:
    Min      1Q  Median      3Q     Max 
-38.037  -7.109  -1.560   6.464  36.302 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.033e+01  2.996e+00   6.785 8.58e-10 ***
income      3.334e-03  3.012e-04  11.067  < 2e-16 ***
women       1.326e-01  4.032e-02   3.289  0.00139 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.54 on 99 degrees of freedom
Multiple R-squared:  0.5593,    Adjusted R-squared:  0.5504 
F-statistic: 62.81 on 2 and 99 DF,  p-value: < 2.2e-16
```

Interaction Terms

You may also be interested in the **interaction** of 2 variables. This means that the effect of one DV on the IV may depend on the levels of another IV.

For instance, you may think that the effect of income on prestige depends on the presence of women in the industry or vice versa. Income and the presence of women obviously impact the prestige of a job. But it is also additionally possible that jobs that have more women are seen as more prestigious when the income associated with these jobs is higher. Working as a secretary is not seen as that prestigious because a lot of secretaries are women AND it is not a highly-paid job. However, a lot of psychologists are women AND they get paid a lot, which would contribute to the perceived prestige of the job.

```
summary(lm(prestige~income + women + income*women, prestige))
```

```
Call:
lm(formula = prestige ~ income + women + income * women, data
= prestige)

Residuals:
    Min      1Q  Median      3Q     Max
-28.576  -6.519  -1.318   5.460  36.042

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.392e+01  2.847e+00   8.402 3.49e-13 ***
income        2.581e-03  3.210e-04   8.038 2.10e-12 ***
women        -1.646e-01  7.495e-02  -2.196   0.0305 *
income:women  7.337e-05  1.612e-05   4.552 1.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.53 on 98 degrees of freedom
Multiple R-squared:  0.6362,    Adjusted R-squared:  0.6251
F-statistic: 57.13 on 3 and 98 DF,  p-value: < 2.2e-16
```

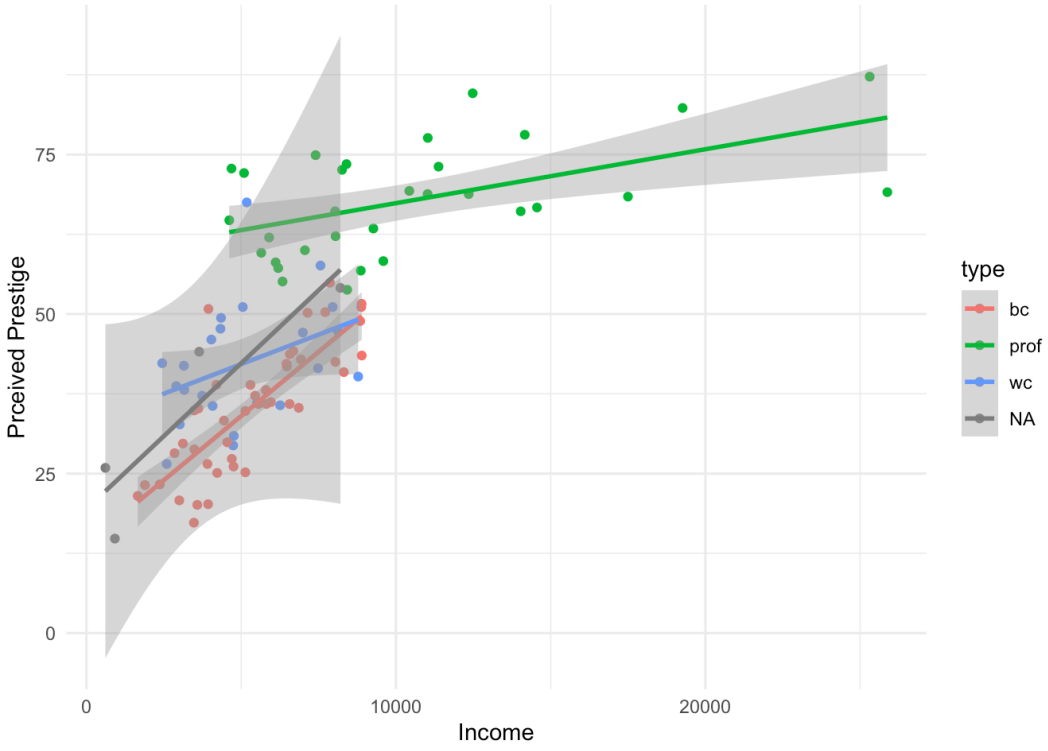
Pretty Plots

Finally, we can make pretty plots in `R` that model these relationships for us.

```
library(ggplot2)
plot_1 <- ggplot(data = prestige,
                 mapping = aes(x = income,
                               y = prestige,
                               color = type)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "Income", y="Prceived Prestige", "Type of Job") +
  theme_minimal()

plot_1
```

```
`geom_smooth()` using formula 'y ~ x'
```


Prceived Prestige
75
50
25
0
0
10000
20000
Income
type
bc
prof
wc
NA