Dis 4: Stata Review

Stata handout offered by Dr. Pac can be accessed here: Handout
This handout reorganizes the official handout's information, and adds more Stata resources.

1 A Probability Question from Exam 1

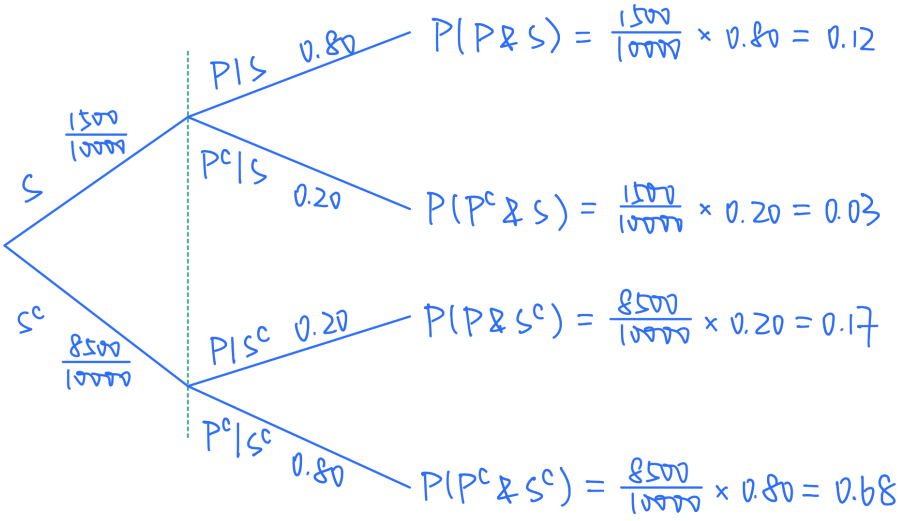
Since our section today is mainly introducing you to Stata, which shouldn't take up the entire duration of the section, let's first go through one fairly challenging question on your first midterm together.

- A UW Madison professor is interested in learning about and detecting the effects of long-haul Covid on individuals' health. The professor examines 10,000 long-haul Covid cases (patients) and learns that 1500 of the cases had serious illness due to long-haul Covid and the other 8500 had no health impact from long-haul Covid. The UW professor develops a new test which is accurate 80% of time in detecting whether long-haul Covid will lead to a serious illness or whether it will have no impact on the patient's health.

  The professor administers the test on a new patient with long-haul Covid. What is the probability that the patient has serious illness due to long-haul Covid if she tests negative using the professor's test?

  This is a question about probability, which is very similar to Dis 3 exercise 3 and 4. To solve any probability related question, let's first draw the tree diagram to present all the probability information given to us in the question.

  Denote the event of having serious illness as S, and the event of being tested positive as P. The tree diagram looks like the following:


S
1500/10000
S^c
8500/10000
P|S 0.80
P^c|S 0.20
P|S^c 0.20
P^c|S^c 0.80
P(P & S) = 1500/10000 × 0.80 = 0.12
P(P^c & S) = 1500/10000 × 0.20 = 0.03
P(P & S^c) = 8500/10000 × 0.20 = 0.17
P(P^c & S^c) = 8500/10000 × 0.80 = 0.68

  Here, the challenging bit might be figuring out the conditional probability $P(P|S)$ and $P(P^C|S^C)$. Since the question says that the test is "accurate 80% of time", the accurate in this context means that if someone has serious illness, then the test is positive 80% of the time (i.e. $P(P|S) = 0.80$), and if someone doesn't have serious illness, then the test is negative 80% of the time (i.e. $P(P^C|S^C) = 0.80$).

Now, the question asks for "the probability that the patient has serious illness . . . if she tests negative". This means that tested negative is the event that has already occurred, so P^C is the event that we condition on. Now, given that someone is tested negative, the probability that the person has serious illness is written as $P(S|P^C)$.

Solving this conditional probability using the conditional probability formula:

$$P(S|P^C) = \frac{P(P^C \text{ and } S)}{P(P^C)} = \frac{P(P^C \text{ and } S)}{P(P^C \text{ and } S) + P(P^C \text{ and } S^C)}$$
$$= \frac{0.03}{0.03 + 0.68} = 0.042$$

Thus, the probability that the patient has serious illness due to long-haul Covid if she tests negative using the professor's test is 0.042.

2 Stata Overview

- **What is Stata?**
  - Stata is a software that allows you to analyze data statistically. You can think about it as the advanced version of Microsoft Excel.
  - Some comparable software / programming languages out there that can do what Stata does (and maybe some more) include SAS, SPSS, R, Python, MATLAB, Julia.
    - Because Stata is currently the most popular statistical package amongst economists, it is what we'll be learning and using for this class.
    - If you go on to take Econ 400 or 410, you will continue to use Stata in that class as well.
    - The Social Science Computing Cooperative (SSCC) here at UW-Madison offers some training classes in Stata, R, and Python. If you're interested in these classes, you can find more information on this website: `https://sscc.wisc.edu/sscc_jsp/training/`
- **How to access Stata?**

  You can access Stata using either one of the two following methods:

  - **Installing it onto your personal laptop (recommended)**:

    Visit UW Software Library (`software.wisc.edu`) for installation guide and license & activation key. The version of Stata to install is **Stata/SE**.
  - **Logging into Winstat (i.e. a remote server; great alternative for people with Chromebook or using unsupported OS)**:

    Check out the following link for information on logging into Winstat: `https://kb.wisc.edu/sscc/using-winstat`

3 Get Started Using Stata

Let's go through the following steps to get you started on using Stata.

1. Launch Stata, and let's go through how the Stata program looks like. Specifically, identify

  - where results show up
  - where can you find the list of variables
  - where can you find more information about the variables
  - where to run your commands (via either the Command panel or the Do-file editor)

2. Before running any commands, let's set up the working directory to tell Stata which folder on your laptop should Stata read data and save graphs or results to. The easiest way to do so is to go to the menu bar, and select

   ```
   File > Change working directory...
   ```

3. Just to make sure your name is somewhere in your results, use the display command to write your name in the log. For example, assuming you happen to be Lindsey Lohan, you would type the following command:

   ```
   display "Lindsey Lohan"
   ```

4. Let's now load a set of data to do some simple statistical analysis. On your Stata problem set, we tell you exactly what command you should use to load the appropriate dataset. But for today, let's just load a sample dataset known as auto:

   ```
   sysuse auto
   ```

5. Use the describe command to determine which variable in this dataset contains "Price" and which contains "Trunk space in cubic feet":

   ```
   describe
   ```

6. Use the histogram command to graph a histogram of price and assess whether the distribution is symmetric or skewed:

   ```
   histogram price
   ```

7. Save the histogram created by clicking on the "Save" button in the graph window. Make sure you save the graph as a `.png` file.

8. Use the summarize command to calculate the mean, median, and standard deviation of trunk space:

   ```
   summarize trunk
   ```

9. Did the previous command have all the information you needed? If not, let's now try the same command with the detail option:

   ```
   summarize trunk, detail
   ```

10. Use the correlate command to calculate the correlation coefficient between price and trunk space:

    ```
    correlate price trunk
    ```

11. Use the scatter command to graph a scatterplot of price (on the y-axis) and trunk space (on the x-axis) and assess the relationship between the two variables (note that the order of variables in the following command matters):

    ```
    scatter price trunk
    ```

12. We've now finished running all practice commands. To save the output printed in the Results panel, go to

    ```
    File > Print > Results
    ```

    and then save your Stata output as a PDF document using your OS's printing dialogue.

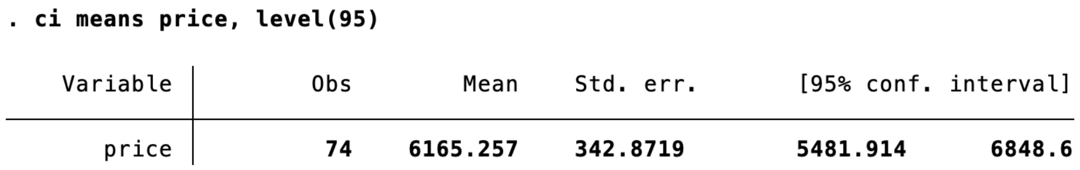
4 Some More Commands For Future Use

The previous section covered all the Stata commands for the statistical operations that we have learned so far in class. There are some other Stata commands that could be helpful for your Stata problem set due around the end of the semester (April 30 at 11pm). They are listed in here for you to reference to when doing your Stata problem set.

1. Use the ci command to calculate a 95% confidence interval for price:

   ```
   ci means price, level(95)
   ```

   The result from running this command looks like the following:

   ```
   . ci means price, level(95)

       Variable |        Obs        Mean    Std. err.       [95% conf. interval]
   -------------+---------------------------------------------------------------
          price |         74    6165.257    342.8719        5481.914      6848.6
   ```

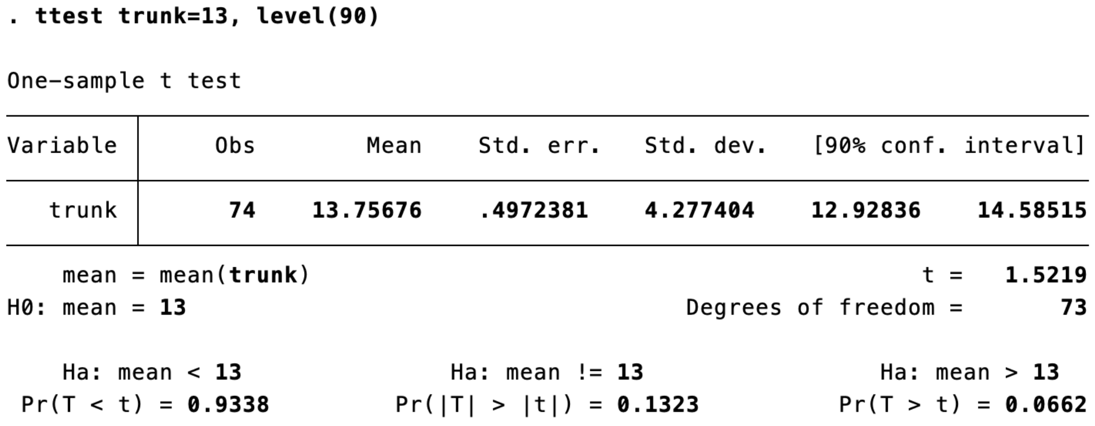
   From the result table, the last two columns under the marker [95% conf. interval] is the lower and upper bound of the 95% confidence interval. Thus, the 95% confidence interval is

   $$[5481.914, \quad 6848.6]$$

2. Use the ttest command to test whether the population mean of trunk space is equal to 13 using a 10% size of test:

   ```
   ttest trunk=13, level(90)
   ```

   The result from running this command looks like the following:

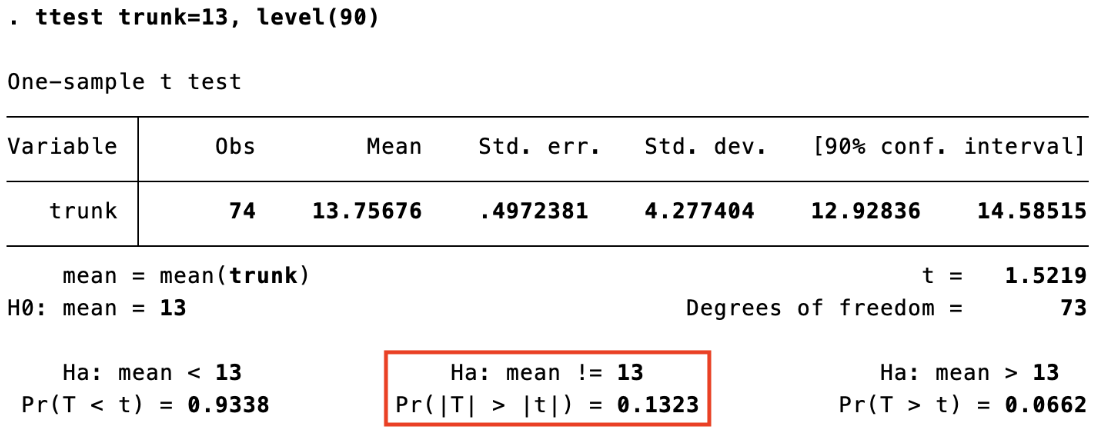
   

   ```
   . ttest trunk=13, level(90)

   One-sample t test
   ------------------------------------------------------------------------------
   Variable |     Obs        Mean    Std. err.   Std. dev.   [90% conf. interval]
   ---------+--------------------------------------------------------------------
      trunk |      74    13.75676    .4972381    4.277404    12.92836    14.58515
   ------------------------------------------------------------------------------
       mean = mean(trunk)                                            t =   1.5219
   H0: mean = 13                                    Degrees of freedom =       73

       Ha: mean < 13               Ha: mean != 13               Ha: mean > 13
    Pr(T < t) = 0.9338         Pr(|T| > |t|) = 0.1323          Pr(T > t) = 0.0662
   ```

   When testing whether the population mean of trunk space is equal to 13, the null and alternative hypotheses are

   $$H_0 : \mu = 13$$
   $$H_1 : \mu \neq 13$$

   This means that we can look at bottom middle part of the result to perform the test:



```
. ttest trunk=13, level(90)

One-sample t test
------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. err.   Std. dev.   [90% conf. interval]
---------+--------------------------------------------------------------------
   trunk |      74    13.75676    .4972381    4.277404    12.92836    14.58515
------------------------------------------------------------------------------
    mean = mean(trunk)                                            t =   1.5219
H0: mean = 13                                    Degrees of freedom =       73

    Ha: mean < 13               Ha: mean != 13               Ha: mean > 13
 Pr(T < t) = 0.9338         Pr(|T| > |t|) = 0.1323          Pr(T > t) = 0.0662
```

(The Ha means "alternative hypothesis", which is the same thing as H_0)

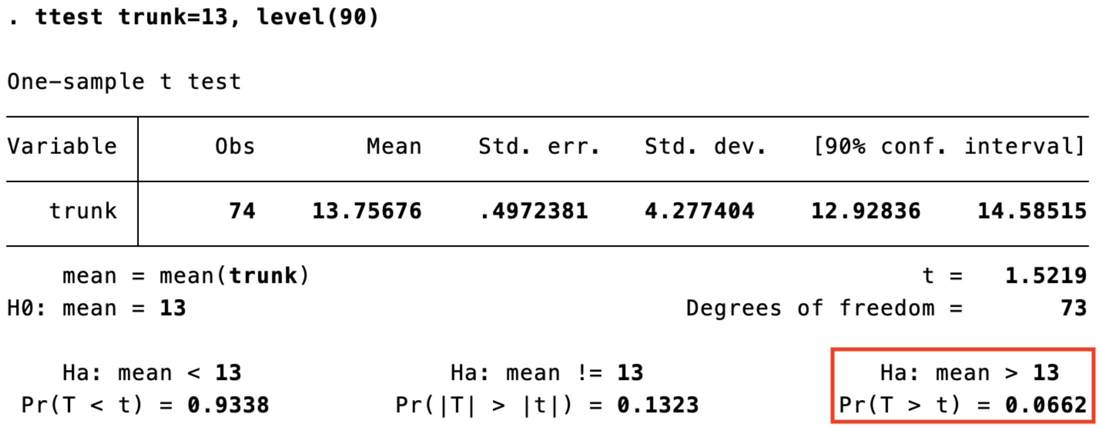
In the red box highlighted, $Pr(|T| > |t|) = 0.1323$ means that the p-value for this test is 0.1323. Given that the p-value is the lowest significance level needed to reject the null, and that our significance level (i.e. size) is $10\% = 0.10 < 0.1323 =$ the lowest significance level needed for rejection of the null, this means that we fail to reject the null at 10% size.

3. Use the ttest command to test whether the population mean of trunk space is greater than 13 using a 10% size of test. Note: Will this command differ from the one above? Why or why not?

   The command will not differ from the one above. Now our alternative hypothesis has changed to the following:

$$H_1 : \mu > 13$$

   This means that we can look at bottom right part of the result to perform the test:



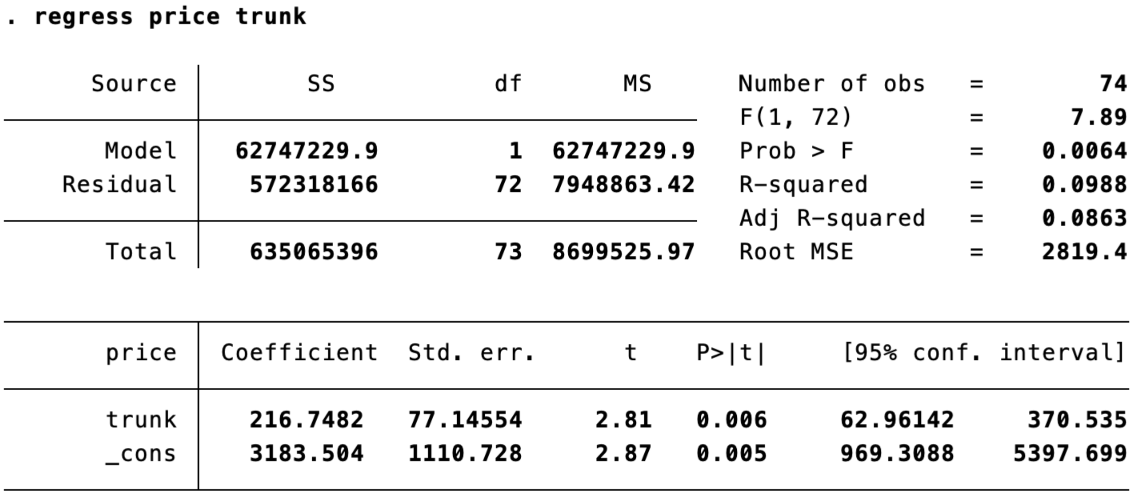
```
. ttest trunk=13, level(90)

One-sample t test
------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. err.   Std. dev.   [90% conf. interval]
---------+--------------------------------------------------------------------
   trunk |      74    13.75676    .4972381    4.277404    12.92836    14.58515
------------------------------------------------------------------------------
    mean = mean(trunk)                                            t =   1.5219
H0: mean = 13                                    Degrees of freedom =       73

    Ha: mean < 13               Ha: mean != 13               Ha: mean > 13
 Pr(T < t) = 0.9338         Pr(|T| > |t|) = 0.1323          Pr(T > t) = 0.0662
```

   In the red box highlighted, $Pr(|T| > |t|) = 0.0662$ means that the p-value for this test is 0.0662. Since our significance level (i.e. size) is $10\% = 0.10 > 0.0662 =$ the lowest significance level needed for rejection of the null, this means that we reject the null at 10% size now.

4. Use the regress command to run a regression of price on trunk space (this language means price should be the dependent variable while trunk should be the explanatory variable, so the order of variables in the following command matters):

```
regress price trunk
```

The result from running this command looks like the following:



```
. regress price trunk

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(1, 72)        =      7.89
       Model |  62747229.9         1  62747229.9   Prob > F        =    0.0064
    Residual |   572318166        72  7948863.42   R-squared       =    0.0988
-------------+----------------------------------   Adj R-squared   =    0.0863
       Total |   635065396        73  8699525.97   Root MSE        =    2819.4

------------------------------------------------------------------------------
       price | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       trunk |   216.7482   77.14554     2.81   0.006     62.96142     370.535
       _cons |   3183.504   1110.728     2.87   0.005     969.3088    5397.699
------------------------------------------------------------------------------
```

(When using the `regress` command, the variable directly following `regress` is the dependent variable y, and whatever follows afterwards are the independent variable(s) x)

From the Coefficient column, we can read the βs estimated. The _cons row is for β_0. In other words, the estimated linear line is

$$\begin{aligned}\widehat{\text{price}} &= \hat{\beta}_0 + \hat{\beta}_1 \text{ trunk} \\ &= 3183.504 + 216.7482 \text{ trunk}\end{aligned}$$