



Lab 1. Why William & Mary?

Due by noon on Friday, September 11th



"I look to the diffusion of light and education as the resource most to be relied on for ameliorating the condition, promoting the virtue, and advancing the happiness of man.

Letter from Thomas Jefferson to C.C. Blatchley, 1822



Laboratory in Brief:

The purpose of this laboratory is to think about ways to justify a decision using data that is openly accessible via a web browser. We will continue your introduction to the statistical programming language R and some of its basic methods. You may be asking yourself, "Will I really need to defend my choice to attend William & Mary for another two weeks?" The goal of this assignment is not only to encourage you to consider the College you have chosen, but also to begin thinking about how to obtain, import, manage and use data in support of your (very good) decision.

Goals of this Laboratory:

1. To retrieve data from an open access server via a web browser and create a .csv (comma-separated values) file for saving to your William & Mary H:\ drive.
2. To begin understanding how to think about and describe data, while also continuing our introduction to the statistical programming language R.
3. To create plots for incorporation into a visualization that supports a decision using a given data set.

Session 1: Monday, August 31st

Step-by-step instructions: first, lets acquire the data by using our web browser



1. With your browser go to the National Center for Education Statistics' Data Center website. You should be able to simply click on the following link.
<https://nces.ed.gov/ipeds/datacenter/>
2. On the left side of the webpage find the **Download Custom Data Files** link and select it.

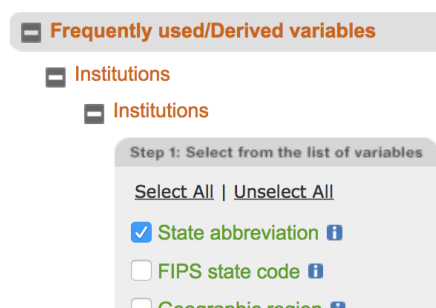



3. After selecting the **Download Custom Data Files** link you will be asked *What data would you like to access?* Under the *Provisional Release Data* and below the



For additional years of data: go ahead and select **Use final release data** and then select **CONTINUE**.

4. Next you will be asked "How would you like to select institutions to include in your data file/report?" Of the three options available, place your cursor over the link for **By Groups**, so that a sub-window appears, then choose **EZ Group**.
5. After selecting **EZ Group** you will be presented with options under the heading *Data Collection: 2013*. The first line will present you with four options, choose **U.S. only**. Once you select **U.S. only**, the grey bar to the right should automatically update to indicate that 7597 institutions will be selected. Then go ahead and choose **Search**.
6. Now your institutions have been selected and you should have a list of the first 20 of the 7597 U.S. higher education institutions that you will be selecting. On this page, you only need to select the **CONTINUE** button that is found following the statement *When you have finished selecting institutions* **CONTINUE** to Step 2 - Select Variables.
7. On the next page you will select the variables that will be downloaded to your local machine as a .csv file. Scroll down past *Available Years* and find the  to the left of **Frequently used/Derived variables** and click on it so all the sub options appear. Then expand the  to the left of **Institutions**. This same field appears again (*seems like it might be a website typo*) so you will need to expand it a second time. Then once you have fully expanded the field, choose **State abbreviation**.



8. Once you have selected the **State abbreviation** box go back to the top level **Frequently used/Derived variables** and select the  to the left of the **Total cost of attendance** field. This field should expand and present you with several checkbox options under the title *Price*. Select the first four fields named **Tuition and fees**, each one for a different year: **2010-11**, **2011-12**, **2012-13**, **2013-14**.



☒ **Total cost of attendance**

☒ **Price**

Step 1: Select from the list of variables

Select All | Unselect All


☒ Tuition and fees, 2010-11 ⓘ

☒ Tuition and fees, 2011-12 ⓘ

☒ Tuition and fees, 2012-13 ⓘ

☒ Tuition and fees, 2013-14 ⓘ

☐ Total price for in-district students

9. Now you will want to go back to the top level again **Frequently used/Derived variables** and this time expand the  next to **Revenues and expenditures: Fiscal year 2013**. Then under that heading expand **Expenses for salaries, wages and nbenefits as a percent of total expenses, by function** where you will need to choose the first four options **Salaries, wages and benefit expenses for core expenses, instruction, research and public service**

☒ **Revenues and expenditures: Fiscal year 2013**

☒ Percent distribution of core revenues, by source

☒ Core revenues per FTE enrollment, by source

☒ Percent distribution of core expenses, by function

☒ Core expenses per FTE enrollment, by function

☒ **Expenses for salaries, wages, and benefits as a percent of total expenses, by function**

Step 1: Select from the list of variables

Select All | Unselect All

☒ Salaries, wages, and benefit expenses for core expenses as a percent of total core expenses (GASB) ⓘ

☒ Salaries, wages, and benefit expenses for instruction as a percent of total expenses for instruction (GASB) ⓘ


☒ Salaries, wages, and benefit expenses for research as a percent of total expenses for research (GASB) ⓘ

☒ Salaries, wages, and benefit expenses for public service as a percent of total expenses for public service (GASB) ⓘ

☐ Salaries, wages, and benefit expenses for academic support as a percent of total expenses for academic support

10. Finally, scroll back up to the top of the page and click .

11. On this final page, the IPEDS Data Center web page will ask in what format do you want your data. In the box labeled *Year 2013*, and under the heading *Download as single file for:*, off to the right select the **CSV button**.

Year 2013 

Download as single file for:

Frequently used/Derived variables/Total cost of attendance

Frequently used/Derived variables/Institutions

Frequently used/Derived variables/Revenues and expenditures: Fiscal year 2013

CSV **SAS**

STATA **SPSS**

12. Selecting the **CSV button** will download a folder that is named CSV_ followed by a series of numbers to your download folder. Go ahead and move that folder to your desktop and open it up. Inside this folder you will find a .csv file and .html file. Go ahead and move the .csv file to your William & Mary H:\\ drive, then rename it to lab1_data.csv or something similar.



13. Then we can open your `.csv` file in Microsoft Excel and have a brief look.
14. Finally, let's import your `.csv` file into R and create an object.
 - A. In your William & Mary `H:\` drive, create a New Folder and name it `lab1` or something similar.
 - B. Open R or R Studio
 - C. Create a new script file. Go to the **File** menu and choose **New Document**. Then save this `.R` script file to your William & Mary `H:\lab1` folder, and name it `lab1_script.R` or something similar.
 - D. One of the first lines of code in your script file should tell R where your working directory is located, which in this case is `H:\lab1`. The working directory is the default location where R will look for a file that has been identified as part of a command. While we could specify the full path of our file location, once we set the working directory, R will automatically know where to find our `.csv` file.

```
R> setwd("H:\\lab1")
```

- E. After setting your working directory, now we should be able to import our `.csv` file into R by using the `read()` command in R and typing the following.

```
R> data <- read.csv("lab1.csv")
```

Now we have created an object named `data`. By keeping our source data external to R and then importing it and creating a new object, we have greatly enhanced the reproducibility of our work and advancing one of the fundamental principals of the scientific method. From this point forward our work is in our code, while the data remains in its original state and does not change.

- F. Finally we can think about the structure of the R object we have created. The `str()` command is used to determine the type of object, and several other characteristics about that object. Rather than saving the `str()` command in our code, we can simply type it directly into the console.

```
R> str(data)
```

R informs us that our object is a `data.frame` with 7597 obs. of 12 variables. Then R lists the name of each variable in our `data.frame`, tells us the variable name, the data type for that variable and then prints the first few observations found in that variable.



Session 2: Wednesday, September 2nd

Step-by-step instructions: now let's modify & describe our data

1. For starting out on day 2, let's think about the R object we have created a bit more. We can type in `str(data)` as we did at the end of Session 1, or we could also use the `names()` command, and R will give us the names of each of the 12 variables.

```
R> names(data)
```

Many of these names are long and confusing, so it would be helpful to rename our variables and make them more manageable. To do this, we will create a new R object that contains each of the new names that will be used to replace the 12 existing ones. When doing this, it is important that the new names are listed in the same order as the existing ones as contained in the `data.frame`.

```
R> data_names <- c("id", "name", "year", "tuition10_11",
  "tuition11_12", "tuition12_13", "tuition13_14", "state",
  "core_exp", "instructional_exp", "research_exp",
  "publicservice_exp")
```

We can then type `data_names` directly into the console and R will return the same names we just typed when creating that object.

2. Again we use the `names()` function to replace the existing names with the new ones.

```
names(data) <- data_names
```

Now we can type `str(data)` to verify that our variables have been renamed.

3. Next we will use R to examine our data. First, let's think about how our data set can be described spatially. To do this we can consider the state where each institution is located by identifying the appropriate variable that provides this information. Using the `str(data)` command again should begin to facilitate our inquiry.

```
> str(data)
'data.frame':      7597 obs. of  12 variables:
 $ id              : int  100654 100663 100690 100706 100724 100733 ...
 $ name            : Factor w/ 7457 levels "A & W Healthcare Educato...
```



```

$ year      : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
$ tuition10_11 : int  5800 5806 8360 7492 7164 NA 7900 2700 NA 6000 2700 NA 6000
$ tuition11_12 : int  6828 6264 8720 8094 8082 NA 8600 2700 NA 7500 2700 NA 7500
$ tuition12_13 : int  7182 6798 6800 8794 7932 NA 9200 4140 NA 8000 4140 NA 8000
$ tuition13_14 : int  7182 7206 6870 9192 8720 NA 9450 4200 NA 8000 4200 NA 8000
$ state       : Factor w/ 51 levels "Alabama","Alaska",...: 1 1 1 1 1 1 1 1 1 1 1 1
$ core_exp    : int  48 68 NA 68 44 99 67 43 61 59 ...
$ instructional_exp: int  78 83 NA 80 73 NA 77 83 77 92 ...
$ research_exp : int  52 60 NA 69 33 NA 52 NA NA 56 ...
$ publicservice_exp: int  60 54 NA 45 14 NA 70 17 NA 49 ...

```

We can see this `data.frame` object has 12 variables, the name for each one listed (as we had previously renamed them), following a `$` sign. The variable `$state` appears to be the correct one to determine the location for each of the 7597 obs. In order to return a table that aggregates all institutions according to state, we use the `table()` command by executing the following code.

```
R> table(data$state)
```

Now we have introduced an additional wrinkle to our code. Instead of simply executing a function that addresses only an object, we have also qualified that object by using the `$` sign. The `$` sign is used to indicate which of the variables within the object is being used as part of the executed function. In this case when we use `table(data$state)` R totals the number of institution located in each state.

Alabama	Alaska	Arizona	
93	12	146	
Colorado	Connecticut	Delaware	District of Columbia
140	100	21	
Georgia	Hawaii	Idaho	
199	28	44	
Iowa	Kansas	Kentucky	
96	96	117	
Maryland	Massachusetts	Michigan	
105	197	204	
Missouri	Montana	Nebraska	
226	31	53	
New Jersey	New Mexico	New York	
173	52	481	
Ohio	Oklahoma	Oregon	
383	149	99	



<i>South Carolina</i>	<i>South Dakota</i>	<i>Tennessee</i>
113	31	189
<i>Vermont</i>	<i>Virginia</i>	<i>Washington</i>
28	178	127
<i>Wyoming</i>		
12		

4. We can also ask R to give us the summary statistics of all observations for a single variable. To do this, we use the `summary()` command and again identify that variable by using the `$` operator.

```
R> summary(data$tuit13_14)
```

<i>Min.</i>	<i>1st Qu.</i>	<i>Median</i>	<i>Mean</i>	<i>3rd Qu.</i>	<i>Max.</i>	<i>NA's</i>
80	5242	12110	14140	18050	64900	3424

In this case, R has returned basic descriptive statistics, including the mean, median, 1st and 3rd quartiles, as well as the minimum and the maximum for the variable total tuition and costs for the school year 2013 to 2014. The last column NA's is a special designation in R that counts the total number of observations within a variable that do not have an outcome. It literally stands for *Not Available*. Keep in mind that an observation outcome NA is different from a 0 or an indication such as None.

5. We can also do the same thing, but instead of calculating the summary statistics for all U.S. educational institutions, we can subset only those schools that are located in the state of Virginia. The simplest way to do this is by using the `subset()` command.

```
R> subset(data, state == "Virginia")
```

Notice the `subset()` command introduces another, new wrinkle to our code. Rather than using the `$` operator as before, this time, first we indicate the object, then a second qualifying statement designates the variable, in this case `state == "Virginia"`.

Additionally, it would be helpful to have the summary statistics for our subset of the data. First, we create a new object using the assignment operator and then execute our `summary()` command once again for our subset of the data we have just created.



```
R> va_data <- subset(data, state == "Virginia")
R> summary(va_data$tuit13_14)
```

```
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
 3169   7922  14670  14920  18050  45320    60
```

Now we can compare the various summary statistics for all the institutions in Virginia with those for the entire United States of America. How does tuition in Virginia compare with the rest of the country?

6. We can also isolate those institutions in Virginia that are above the national mean? We do this by first creating an object that represents the national average of total tuition and costs for the year 2013 to 2014. Notice the qualifying statement `na.rm = TRUE`, which stands for NA's remove. This statement removes all of the observations that have unavailable outcomes and permits R to return the correct mean for the chosen variable.

```
R> us_mean <- mean(data$tuition13_14, na.rm = TRUE)
R> va_above_avg <- subset(va_data, tuition13_14 > us_mean)
```

7. To see if William & Mary is above the national average, just type the object name `va_above_avg` and look through the output or you can also type

```
R> subset(data, name == "College of William and Mary")
```

8. We can also use the `subset()` command to find the least expensive school in the country, while using the `class()` command will inform us of the object type. In this case, our institution name is returned as a factor, when all we need is the school's name. We can use `as.character()` to convert the name of the school from factor to character.

```
R> min(data$tuition13_14, na.rm = TRUE)
R> min_cost <- min(data$tuition13_14, na.rm = TRUE)
R> min_obs <- subset(data, tuition13_14 == min_cost)
R> min_obs$name
R> class(min_obs$name)
R> min_name <- as.character(min_obs$name)
R> min_name
```



```
R> class(min_name)
```

Sometimes `subset()` functions requirement of using a logical argument can be a nuisance. A more advanced approach to solving the same problem is to extract observations using brackets. The `[` and `]` brackets are also called subscripting operators.

```
R> data[which.max(data$tuition13_14),]  
R> data[which.max(data$tuition13_14),]$name  
R> as.character(data[which.max(data$tuition13_14),]$name)
```

As we can see, we accomplished the same result with only two lines of code using the `[` and `]` brackets instead of the `subset()` command. Subscripting is one of the most powerful tools available in R due to its speed and widespread applicability. Type `?Subscript` or `?Extract` to read the help page on how to use the `[` and `]` brackets as operators.

Session 3: Monday, September 7th

Step by Step Instructions: now let's analyze & plot our results

1. Now, we're going to do some basic analysis that you can include on your revised infographic. Note the goal of this R tutorial is not to create beautiful visualisations though R can do that. Rather, we want to extract data you can then use in visme to improve your original infographic. Let's do something very simple to start: how does the percentage increase of William and Mary's tuition compare to (a) all schools, and (b) Virginia schools from 2010 to the 2013-2014 school year?
2. First, let's calculate this for William and Mary (type in `wm_change`) at the end.

```
R> wm_tuition1314 <- data[which(data$name == "College of  
William and Mary"), ]$tuition13_14  
R> wm_tuition1213 <- data[which(data$name == "College of  
William and Mary"), ]$tuition12_13  
R> wm_tuition1112 <- data[which(data$name == "College of  
William and Mary"), ]$tuition11_12  
R> wm_tuition1011 <- data[which(data$name == "College of  
William and Mary"), ]$tuition10_11  
  
R> wm_change14 <- wm_tuition1314 / wm_tuition1213  
R> wm_change13 <- wm_tuition1213 / wm_tuition1112
```



```
R> wm_change12 <- wm_tuition1112 / wm_tuition1011

R> wm_change <- mean(c(wm_change14, wm_change13, wm_change12))

R> wm_change
[1] 1.083435
```

This gives us the *average annual change* in tuition costs for the time period from the school year 2010 and 2011 until the school year 2013 and 2014. Note there are only three annual intervals to average since we have not included the tuition costs for the years before 2010 or after 2014. The `c()` operator is another new command that literally means to combine. Here we use the `c()` to find the mean of the three combined objects.

3. Now, let's calculate the average change for every other school. This is the first time we will add a new column to our data frame. We can review our existing variable names, as we did at the beginning of this laboratory by again using the `names()` command.

```
R> names(data)
```

4. Now, let's create a new variable that describes the *average annual change* in tuition costs for all institutions during the same period of time (as we have just done for William & Mary).

```
R> data$us_change14 <- data$tuition13_14 / data$tuition12_13
R> data$us_change13 <- data$tuition12_13 / data$tuition11_12
R> data$us_change12 <- data$tuition11_12 / data$tuition10_11

R> data$us_change <- (data$us_change14 + data$us_change13 +
data$us_change12) / 3

R> data$us_change

R> mean(data$us_change, na.rm = TRUE)
[1] 1.046987
```

5. Note, there are several new names in our data.frame when we type `names(data)`. Let's summarize the annual change variable now:



```
R> summary(data$sus_change)
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
0.709    1.023    1.039    1.047    1.057    8.701    3592
```

6. Let's do a comparison between the tuition in 2014 and the percent of that money spent on instructors salary. First, let's look it up for William & Mary:

```
R> wm <- subset(data, name == "College of William and Mary")
R> wm$instructional_exp
```

7. And second, let's compare that to the national average:

```
R> summary(data$instructional_exp)
```

8. Finally, let's plot out all schools tuition in 2014 contrasted to the % of tuition they spend on instructors salary.

```
R> plot(data$instructional_exp, data$tuition13_14)
```

9. That on it's own isn't particularly helpful, as it's hard to see patterns with outliers and without knowing which dot is William & Mary. It's also probably more fair to compare to only Virginia schools, so let's drop out everything that isn't Virginia and label W&M in an appropriate color. First, re-use the old VAcollleges dataset you made:

```
R> plot(va_data$instructional_exp, va_data$tuition13_14)
```

10. Now, let's label just W&M save this figure (click the export button above the chart) as you will be turning it in!

```
R> wm_plot <- data[which(data$name=="College of William and
Mary"),]
R> points(wm_plot$instructional_exp, wm_plot$tuition13_14,
col="green")
```



Session 4: Wednesday, September 9th

Step by Step Instructions: Lab Questions

1. Calculate which two states have the most colleges, and how many does each one have?
2. Calculate the mean tuition cost for the US and for Virginia? Does Virginia have a higher or lower average tuition cost than the rest of the USA? What is this difference?
3. Calculate the 2010-2011, 2011-2012, 2012-2013, and 2013-2014 Tuition and Annual Percent Change for William & Mary, all Universities in Virginia, and all Universities in the USA.
4. Create a graph showing USD invested in instruction as compared with tuition costs and highlight William & Mary. Create a similar chart considering the mean of these two values for all Universities in Virginia and all Universities in the USA.
5. Calculate what are the most and least expensive Universities in Virginia and the USA.
6. Attach a printed copy of your "visme" visualisation to this document, or submit it electronically. What variable did you include that was not explicitly a part of this lab? What challenges did you have in retrieving it, and how did you use it to illustrate William and Mary is an exceptional (or, crazy!) choice of where to get your degree?

Stretch Goals

Made it this far? Want to try to go even farther? Just want to learn? Try any of the following to really impress us, it won't count for your grade, but it'll give you a leg up on future assignments:

- In your download, ask for the "latitude" and "longitude" columns, then plot colleges following this tutorial: <http://www.milanor.net/blog/?p=594>
- Examine how US schools compare to international schools.
- Make similar comparisons with a different dataset ? i.e., try

Final Output for Submission

Due by noon on Friday, November 13th:

... Make certain the Final Report meets the following criteria.

- Single spaced with block paragraph style and 10 to 12 font, preferably New Times Roman, Arial, Courier or a similar font.
- x to x pages in length



- Output from Session 2
- Output from Session 3
- Answers to Questions Considered during Session 4

Grading

This lab will be graded based on the six deliverables identified on the final page of this lab. The first five questions are worth 50% of your grade, and are based on your capability to work through the steps below (i.e., if you follow the steps, you'll get the right answer!). The last question ? question 6 ? asks you to update your visualisation you created last week using not only the datasets referenced in this lab, but asks you to download at least one additional variable to use in your update. The process of critically deciding what data to use, how to analyse it, and how to visualise it will be worth the remaining 50% of your grade.