Descriptive statistics with dplyr, stringr and ggplot2

Joseph S. Tabadero, Jr.

February 19, 2017

## Some introductions

It's been a while since my first blog post about R. Here in the Philippines, R is not as widely used as SAS in the industries and many pirated versions of SPSS in the academe. People I show R to are very much used to the ease of point-and-click workflow that SPSS has to offer and shudder at the sight of lines of R scripts. If you are one of these people I have described, but have been using a spreadsheet such as Microsoft Excel for a while, then you should know that R will not be that difficult for you to learn. In fact, there are a lot of resources that will show you that you can write R functions just as you would write your Excel functions. You can find some of resources on learning R from an Excel background from [here](https://www.r-bloggers.com/introduction-to-r-for-excel-users/) and [here](https://www.rforexcelusers.com/excel-r-function-formula-list/).

One of my friends asked me to start from the most basic lessons on R. However, I don't want to repeat what can already be found easily with a simple Google search. Nevertheless, I would like to start at something that someone starting with statistics and R can easily find helpful--tables and plots.

For this tutorial, I will be using the following packages:

* dplyr for structuring data
* stringr for the text search function str\_detect
* ggplot2 for the plots
* knitr for the kable function

If you have not installed any of these packages, install them first by opening your R console and typing and entering the following:

After installation, you can use these packages by the following script:

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(stringr)  
library(ggplot2)

## The data

The data is a list of some science and tech institutions in the Philippines known to our imaginary respondents. The rows may not be unique. Our goals are to present the data using a frequency table and some bar graphs, which are topics that are taught in an intro high school statistics class. You can download the data from this [link](https://dl.dropboxusercontent.com/u/28713619/statechniques/institutions.csv). Download and save this into your working directory. (If you don't know where your working directory is, just type: getwd() in your R console).

There are a lot of ways to import data into R. Here, we are going to use the read.csv function from the base package of R. To see how to use the read.csv function, just type ?read.csv in your R console.

For this tutorial, I am going to import the data and save it to an R object named dataset.

dataset = read.csv("institutions.csv")

To view the dataset, type:

View(dataset)

To see what type of object dataset is, type:

str(dataset)

## 'data.frame': 66 obs. of 1 variable:  
## $ Institution: Factor w/ 18 levels "BENGUET STATE UNIVERSITY",..: 6 15 15 6 4 9 10 9 14 8 ...

If you just want to see the column headers, just type:

names(dataset)

## [1] "Institution"

If you want to see the first 6 rows or entries of the data, type:

head(dataset)

## Institution  
## 1 UPB-BIO  
## 2 UPD-NIP  
## 3 UPD-NIP  
## 4 UPB-BIO  
## 5 NOTRE DAME-BAGUIO CITY  
## 6 UPD-INST BIO

You can view the first 10 rows by typing instead:

head(dataset, 10)

## Institution  
## 1 UPB-BIO  
## 2 UPD-NIP  
## 3 UPD-NIP  
## 4 UPB-BIO  
## 5 NOTRE DAME-BAGUIO CITY  
## 6 UPD-INST BIO  
## 7 UPD-INST CHEM ENGG  
## 8 UPD-INST BIO  
## 9 UPD-NIGS  
## 10 UPD-DMMME

For the next parts, we will be making use of the piping operator (%>%) in order to connect data and R functions. There is a basic way of doing this in R, but it doesn't mean that it is easier. There are a lot of materials written about subsetting your data in R using the basic approach so I will no longer do that here. The piping operator is a tidy way to clean, structure, and subset your data in R.

The mutate function is used here in order to create a new column man which stands for "Manila", by which we Filipinos loosely name our National capital Region. The ifelse function is similar to your Excel if function. The "OR" binary operator in R is the same as the "OR" binary operator in Excel (|). The same is true about the "AND" binary operator (&).

My goal with the following script is to create the man column which indicates whether an institution is located in NCR or outside of NCR. I know beforehand where the institutions are located. Now, we can go crazy and do some advanced text mining stuff. But to demonstrate the operators I described above, I will just use a simple hack based on a simple fact: Those institutions starting with "UPD" are institutions in the University of the Philippines Diliman network of institutions that are located in Quezon City in NCR, except for the Marine Science Institute in Bolinao; "UPM" stands for University of the Philippines Manila, also in NCR; the rest are outside of NCR. We can use the base function grep for identifying these characters in each entry in the data set (data frame in R lingo), but for this tutorial, we will use the str\_detect function of the stringr package. To find out more about the str\_detect function, type ?str\_detect. Can you tell what the following script is doing?

dataset = dataset %>% mutate(man = ifelse(str\_detect(Institution, c("UPD")) | str\_detect(Institution, "UPM") | str\_detect(Institution, "SPARKLAB") & !str\_detect(Institution, "BOLINAO"), "yes", "no"))

Going back to our question above, the above script tells R to create a new column named man which indicates if an institution is in NCR (yes) or outside NCR (no). An institution is NCR if it has the strings "UPD", "UPM" but not "BOLINAO" in it. By the way, what happens if we remove dataset = from the script? Will the man column be saved in the object dataset?

To see the changes, we can type:

head(dataset)

## Institution man  
## 1 UPB-BIO no  
## 2 UPD-NIP yes  
## 3 UPD-NIP yes  
## 4 UPB-BIO no  
## 5 NOTRE DAME-BAGUIO CITY no  
## 6 UPD-INST BIO yes

We now have two columns: Institution and man. Practice with the str and names functions of R. You can also try the glimpse function from the tidyr package. Compare this with the str package.

Suppose want to find out the frequency distribution of institutions based on whether they are located in or outside of NCR. Here, I used the group\_by, and summarise functions of dplyr. n() simply counts the number of incidents of finding each level of the variable as specified by group\_by.

dataset %>% group\_by(man) %>% summarise( Frequency = n())

## # A tibble: 2 × 2  
## man Frequency  
## <chr> <int>  
## 1 no 27  
## 2 yes 39

We can create a frequency and relative frequency distribution by making use of the mutate function.

dataset %>% group\_by(man) %>% summarise(Frequency = n()) %>% mutate(Rel.Frequency = Frequency/sum(Frequency))

## # A tibble: 2 × 3  
## man Frequency Rel.Frequency  
## <chr> <int> <dbl>  
## 1 no 27 0.4090909  
## 2 yes 39 0.5909091

Note the use of the sum function. Which function in Excel do you think makes a similar behavior? The chart below shows the distribution of SSIP participants according to the location of their first choices.

## Exercise 1

Can you write a Percent column instead or write it side by side the existing ones? Something like this? (How did I round the Percent column into two decimal places? Any familiar Excel function?)

## # A tibble: 2 × 4  
## man Frequency Rel.Frequency Percent  
## <chr> <int> <dbl> <dbl>  
## 1 no 27 0.4090909 40.91  
## 2 yes 39 0.5909091 59.09

## Exercise 2

Write an R script that will produce the following table by counting the occurrence of each institution.

## # A tibble: 18 × 4  
## Institution Frequency Rel.Frequency Percent  
## <fctr> <int> <dbl> <dbl>  
## 1 BENGUET STATE UNIVERSITY 1 0.01515152 1.52  
## 2 BFAR-DAGUPAN 10 0.15151515 15.15  
## 3 DOST-CAR 3 0.04545455 4.55  
## 4 NOTRE DAME-BAGUIO CITY 6 0.09090909 9.09  
## 5 SPARKLAB INNOVATION CENTRE 1 0.01515152 1.52  
## 6 UPB-BIO 7 0.10606061 10.61  
## 7 UPD-COMP SCI AND RES INST 1 0.01515152 1.52  
## 8 UPD-DMMME 2 0.03030303 3.03  
## 9 UPD-INST BIO 9 0.13636364 13.64  
## 10 UPD-INST CHEM ENGG 3 0.04545455 4.55  
## 11 UPD-INST CIVIL ENGG 7 0.10606061 10.61  
## 12 UPD-INST MECH ENGG 3 0.04545455 4.55  
## 13 UPD-MSI BOLINAO 1 0.01515152 1.52  
## 14 UPD-NIGS 1 0.01515152 1.52  
## 15 UPD-NIP 5 0.07575758 7.58  
## 16 UPM-INST. OF EPIDEMEOLOGY 2 0.03030303 3.03  
## 17 UPM-INST. OF PHARM. SCIENCES 3 0.04545455 4.55  
## 18 UPM-PHIL. EYE RESEARCH INST 1 0.01515152 1.52

## Bar Graphs

Bar graphs are used to show, graphically, the distribution of categorical variables. We can as easily produce the plots using the base graphics capabilities of R. But I will use ggplot2 package of Hadley Wickham, which uses the *grammar of graphics* approach. If you want to know more about ggplot2, just head on to the github page of the updated book [here](https://github.com/hadley/ggplot2-book). You can also read the old version of the book at <http://ggplot2.org/>.

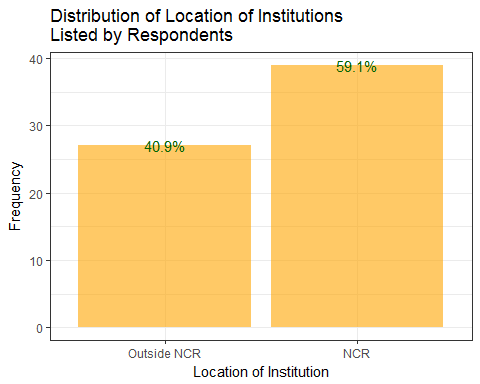
To produce the bar graph for the distribution of institutions according to whether they are located in NCR or not, we simply type:

ggplot(dataset, aes(man)) + geom\_bar()



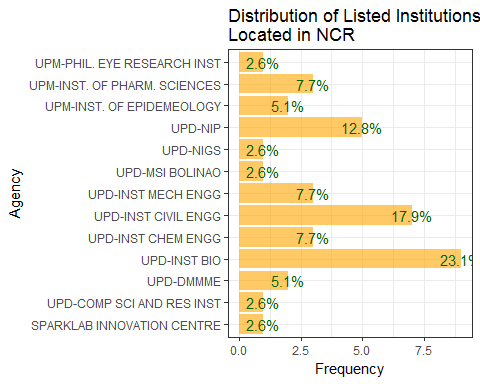
This is the default bar graph that you will get. For data exploration, this is fine. But if you want to present this and make it comprehensible to readers, you have to add some information in the labels and the title.

ggplot(dataset, aes(man)) +   
 geom\_bar(aes(y = (..count..)),fill="orange", alpha=0.6) +   
 geom\_text(aes(y = (..count..),label = ifelse((..count..)==0,"",scales::percent((..count..)/sum(..count..)))), stat="count",colour="darkgreen") +   
 theme\_bw() +  
 xlab("Location of Institution") +  
 ylab("Frequency") +   
 scale\_x\_discrete(labels=c("yes"="NCR", "no"="Outside NCR")) +   
 ggtitle("Distribution of Location of Institutions\nListed by Respondents")



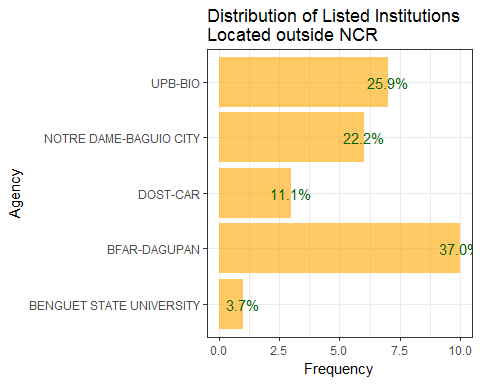
The following is the bar graph for the distribution of listed institutions located within NCR.

dataset %>% filter(man=="yes") %>% ggplot(., aes(x=Institution)) +   
 geom\_bar(aes(y = (..count..)),fill="orange", alpha=0.6) +   
 geom\_text(aes(y = (..count..) ,label = ifelse((..count..)==0,"",scales::percent((..count..)/sum(..count..)))), stat="count",colour="darkgreen") +   
 theme\_bw() +  
 xlab("Agency") +  
 ylab("Frequency") +   
 ggtitle("Distribution of Listed Institutions\nLocated in NCR") +  
 coord\_flip()



## Exercise 3

Produce the following bar graph with ggplot.



And there we have it, folks. Don't forget to subscribe to this blog for more articles about statistics, R, data science, and what-nots.