EDUC 641 Lab: Applied Statistics in Education and Human Services I

Lab 3: 10/16 and 10/17

**Installing packages, exploring tidyverse/ggplot2, and creating visualizations using geom\_bar()**

**Goals:**

1. Installing and loading packages
2. Using ggplot() to describe one categorical variable
3. Using ggplot() to describe two categorical variable

**Packages background**

**Tidyverse** is a collection of R packages designed to make data manipulation and analysis more efficient, consistent, and user-friendly. It's not a single package but a suite of packages that work together cohesively.

1. **ggplot2:** is one of the packages of the "tidyverse" ecosystem in R, which provides a consistent and efficient way to work with data and create visualizations.
2. **dplyr:** Used for data manipulation, including filtering, summarizing, and arranging data.
3. **tidyr:** Designed for tidying and reshaping data, such as converting data from wide to long format.
4. **readr:** A package for efficient reading and writing of data from various file formats.
5. **purrr:** Provides tools for working with functions and vectors.
6. **tibble:** A modern data frame package with improved features over the default R data frames.
7. **stringr:** Offers functions for working with strings and text data.
8. **forcats:** Useful for working with categorical data and factors.

**Today’s Lab**

1. **Installing and loading packages:**
2. *Install* package using “ quotes around package name ” - do this only once.

install.packages(“package\_name”) *# or delete this line of code after it has been run*

1. Load your package using ` library() ` - run this every time you need to use a specific package. Ex. `library(tidyverse) ` or ` library(here) `.

# install.packages(“tidyverse”)

library(tidyverse)

A screenshot of a computer

Description automatically generated

**Try yourself:** Install and load the package “modelsummary.” We will use this later in the term.

1. **Describe one categorical variable from the cat.csv dataset**
2. Select one categorical variable from your dataset and use a bar chart to visualize the counts of each category.

ggplot(name\_of\_data, aes(x = name\_of\_categorical\_variable)) +   
 geom\_bar()+  
 labs(  
 x = "Give a descriptive x-axis label",  
 y = "Give a descriptive y-axis label",  
 title = "Give a descriptive title"  
 ) +  
 theme\_minimal() *# theme\_minimal() is a function that improves your visualization of your graph*

**Try yourself:** with another categorical variable.

1. **Describe two categorical variables from the cat.csv dataset**
2. Let's say you have a dataset with two categorical variables, "Category" and "Group," and you want to create a grouped bar chart to visualize the counts of each category within each group. You can do this as follows:

ggplot(name\_of\_data, aes(x = name\_of\_categorical\_variable, fill = name\_of\_group)) +   
 geom\_bar(position = “dodge”) + *# When you set* ***position = "dodge"****, bars are positioned side by side*  
 labs(  
 x = "Give a descriptive x-axis label",  
 y = "Give a descriptive y-axis label",  
 title = "Give a descriptive title"  
 ) +  
 theme\_minimal() *# theme\_minimal() is a function that improves your visualization of your graph*

NO (position = “dodge”) example

A graph with a red and blue rectangle

Description automatically generated  
YES (position= “dodge”)

A graph with different colored squares

Description automatically generated with medium confidence

*Note.* Plots with bars plotted side by side are more readily interpretable to the viewer and are the preferred visualization.

**Reminder:** To export, use the following code. The files will be saved in the path you specify.

write.csv(counts, "tables/counts.csv")

ggsave(filename = "figures/barplot.jpeg") *#uses default width and height, which can be changed. This*

*must come right after the plot you want to save*