Challenge_4: Intro to Visulization: Univariate and Multivariate Graphs

AUTHOR
Mehak Nargotra

PUBLISHED
March 18, 2023

Make sure you change the author's name in the above YAML header.

Setup

If you have not installed the following packages, please install them before loading them.

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.4     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(readxl)
library(haven) #for loading other datafiles (SAS, STATA, SPSS, etc.)
library(stringr) # if you have not installed this package, please install it.
library(ggplot2) # if you have not installed this package, please install it.
```

Challenge Overview

In this challenge, we will practice with the data we worked on in the previous challenges and the data you choose to do some simple data visualizations using the `ggplot2` package.

There will be coding components and writing components. Please read the instructions for each part and complete your challenges.

Datasets

- Part 1 the ESS_Polity Data (created in Challenge#3) ⭐⭐
- Part 2: the Australia Data (from Challenge#2) ⭐⭐
- Part 3: see [Part 3. Practice plotting with a dataset of your choice (25%)]. For online platforms of free data, see Appendix: sources for data to be used in Part 3.

Find the `_data` folder, then read the datasets using the correct R command.

Part 1. Univariate and Multivariate Graphs (45%)

We have been working with these two data in the previous three challenges. Suppose we have a research project that studies European citizens' social behaviors and public opinions, and we are interested in how the countries that respondents live in influence their behavior and opinion. In this challenge, let's work with the combined dataset *ESS_Polity* and create some visualizations.

1. **Read the combined data you created last time. (2.5%)**

```
#type of your code/command here.
ESS_Polity <- read_csv("~/Desktop/DACSS 601/DACSS_601_datasets/ESS_Polity.csv")
```

```
Rows: 52458 Columns: 18
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (3): cntry, Country, scode
dbl (15): idno, essround, male, age, edu, eth_major, income_10, vote, p5, cy...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ESS_Polity
```

idno <dbl>	essround <dbl>	male <dbl>	a... <dbl>	e... <dbl>	eth_major <dbl>	income_10 <dbl>	cntry <chr>	vote <dbl>
15906	2010	0	14	1	1	2	GR	3
21168	2010	0	14	1	1	2	IE	3
40	2010	0	14	1	NA	8	LT	3
2108	2010	0	14	1	1	NA	RU	3
519	2010	0	14	1	1	NA	IL	2
2304	2010	0	14	1	1	NA	ES	3
290	2010	0	14	1	1	NA	PT	2
3977	2010	0	14	1	1	NA	BG	3
23244	2010	0	14	1	1	NA	IE	2
19417	2010	0	14	1	1	NA	IE	3

1-10 of 10,000 rows | 1-9 of 18 columns   Previous 1 2 3 4 5 6 ... 1000 Next

2. **Suppose we are interested in the central tendencies and distributions of the following variables. At the individual level:** *age, male, edu, income_10,* and *vote*. **At the country level**: *democ*.

   (1) Recode the "vote" column: if the value is 1, recode it as 1; if the value is 2, recode it as 0; if the value is 3, recode it as NA. **Make sure to include a sanity check for the recoded data. (2.5%)**

```
#type of your code/command here.
ESS_Polity<-ESS_Polity%>%
mutate(vote = case_when(
vote == 1 ~ 1,
vote == 2 ~ 0,
vote == 3 ~ NA,
TRUE ~ vote))
#Sanity check for if vote is correctly coded: 1%
unique(ESS_Polity$vote)
```

```
[1] NA  0  1
```

```
  sum_stat <- function(x){
stat <- tibble(
range=paste(range(x, na.rm = T)[1],"-",range(x, na.rm = T)[2]),
mean=mean(x, na.rm = T),
sd=sd(x,na.rm=T),
na = sum(is.na(x)),
unique = length(unique(x)),
class = typeof(x)
)
return(stat)
}
sum_stat_table <- rbind(
age = c(sum_stat(ESS_Polity$age)),
edu = c(sum_stat(ESS_Polity$edu)),
income = c(sum_stat(ESS_Polity$income_10)),
vote = c(sum_stat(ESS_Polity$vote)),
democ = c(sum_stat(ESS_Polity$democ)))
sum_stat_table
```

	range	mean	sd	na	unique	class
age	"14 - 101"	47.91529	18.79573	137	88	"double"
edu	"1 - 4"	2.767531	0.9181334	150	5	"double"
income	"1 - 10"	5.048622	2.787532	12620	11	"double"
vote	"0 - 1"	0.7629986	0.4252476	4222	3	"double"
democ	"6 - 10"	9.452663	1.043149	4451	6	"double"

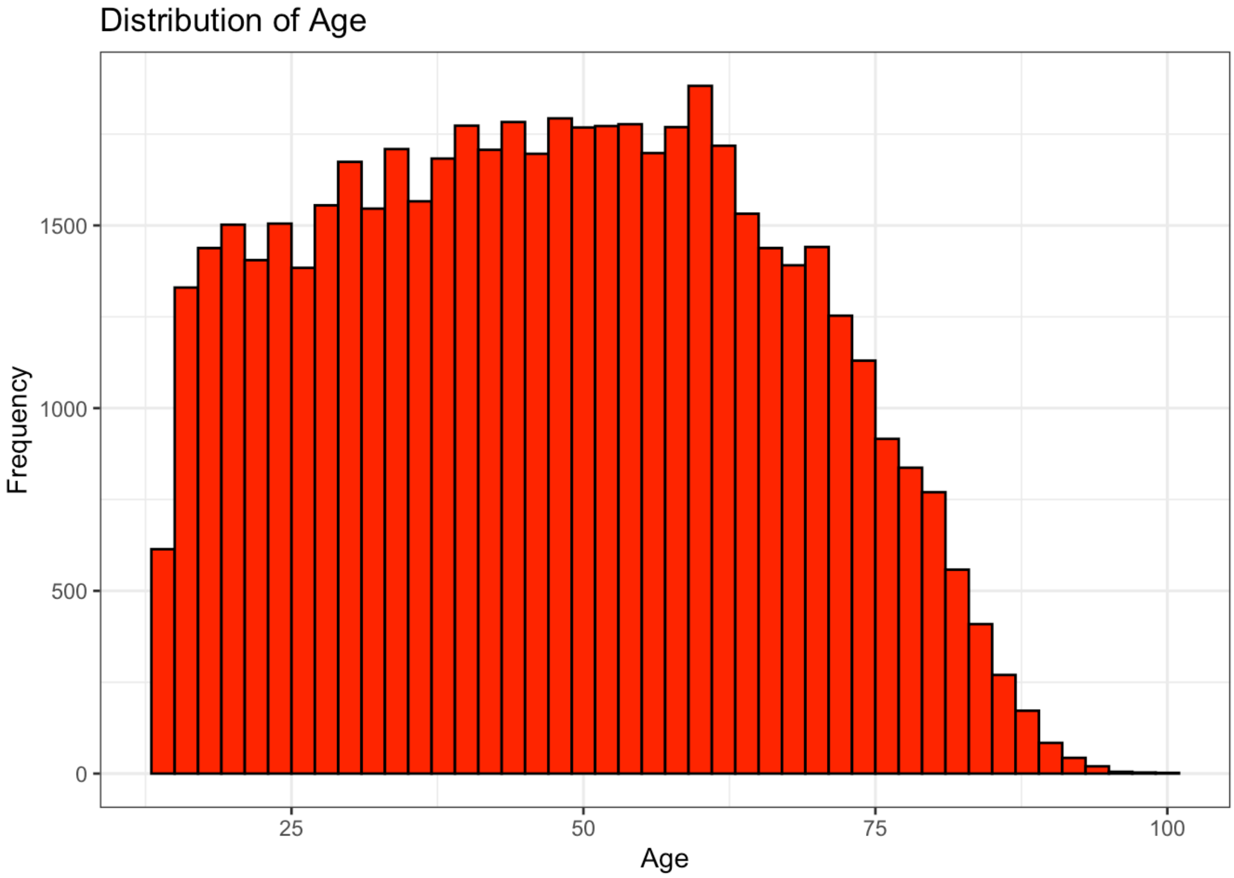
(2) For each of the five variables (*age, edu, income_10, vote,* and *democ)*, please choose an appropriate type of univariate graph to plot the central tendencies and distribution of the variables. Explain why you choose this type of graph to present a particular variable (for example: “I use a histogram to plot *age* because it is a continuous numeric variable”). **(25%)**

(Note: You should use at least two types of univariate graphs covered in the lecture.)

```
#type of your code/command here.
library(ggplot2)
#age
ggplot(ESS_Polity, aes(x = age)) +
  geom_histogram(binwidth = 2, fill = "red", color = "black") +
```

```
  labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
  theme_bw()
```

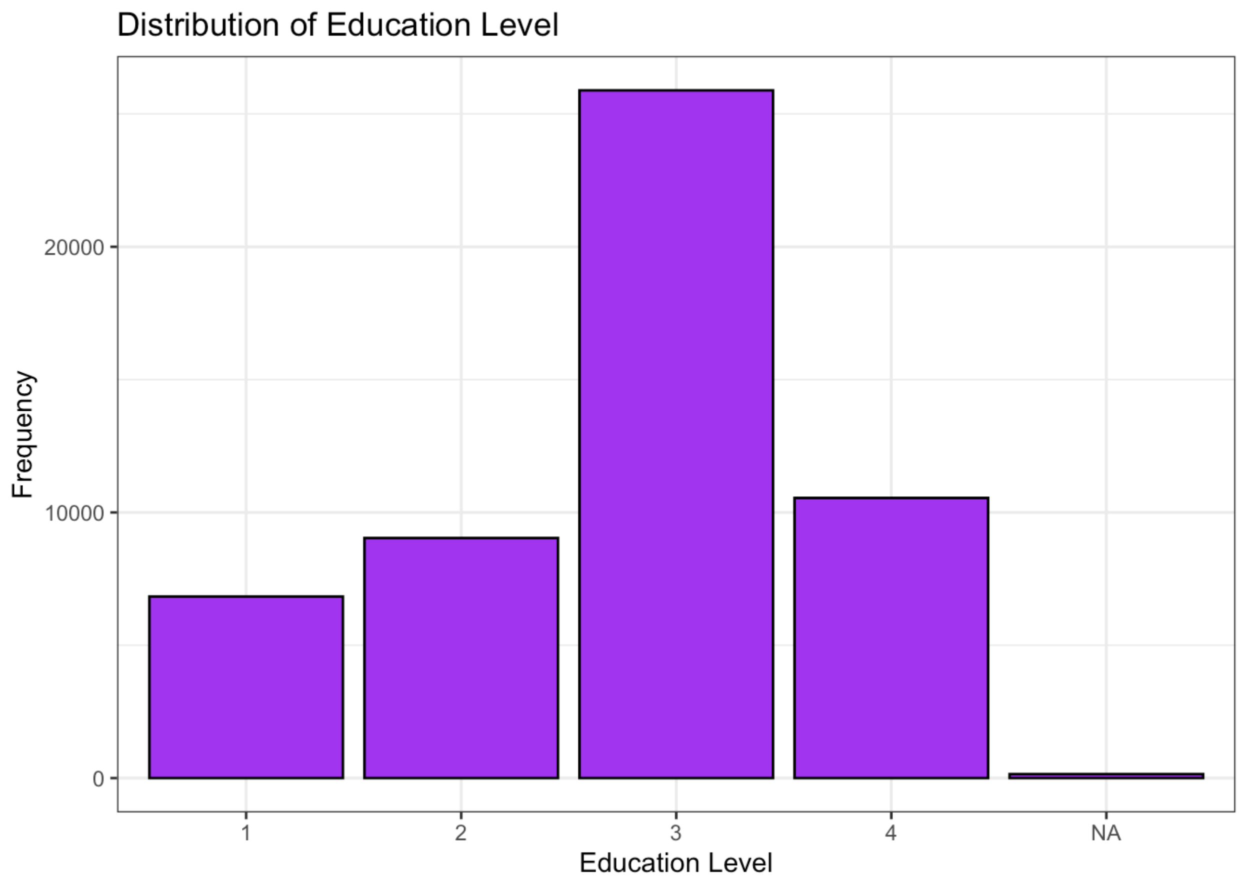
```
Warning: Removed 137 rows containing non-finite values (`stat_bin()`).
```




```
print("I use a histogram to plot 'age' because it is a continuous numeric variab
```

```
[1] "I use a histogram to plot 'age' because it is a continuous numeric
variable."
```

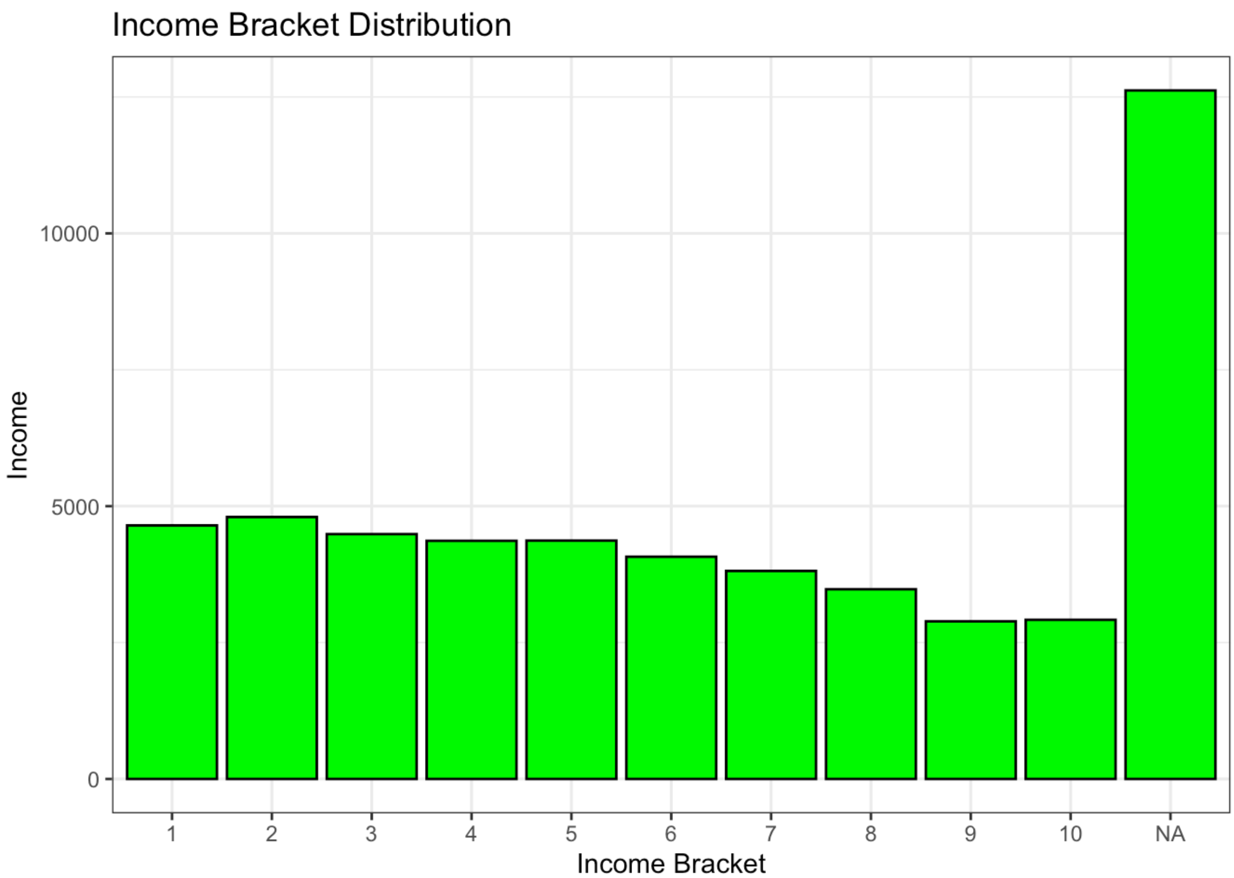
```
#edu
ggplot(ESS_Polity, aes(x = factor(edu))) +
  geom_bar(fill = "purple", color = "black") +
  labs(title = "Distribution of Education Level", x = "Education Level", y = "Fr
  theme_bw()
```




```
print("I use a barchart to plot 'edu' because it is a categorical variable conta
```

```
[1] "I use a barchart to plot 'edu' because it is a categorical variable
containing 5 variables (i.e., 1,2,3,4 and NA)."
```

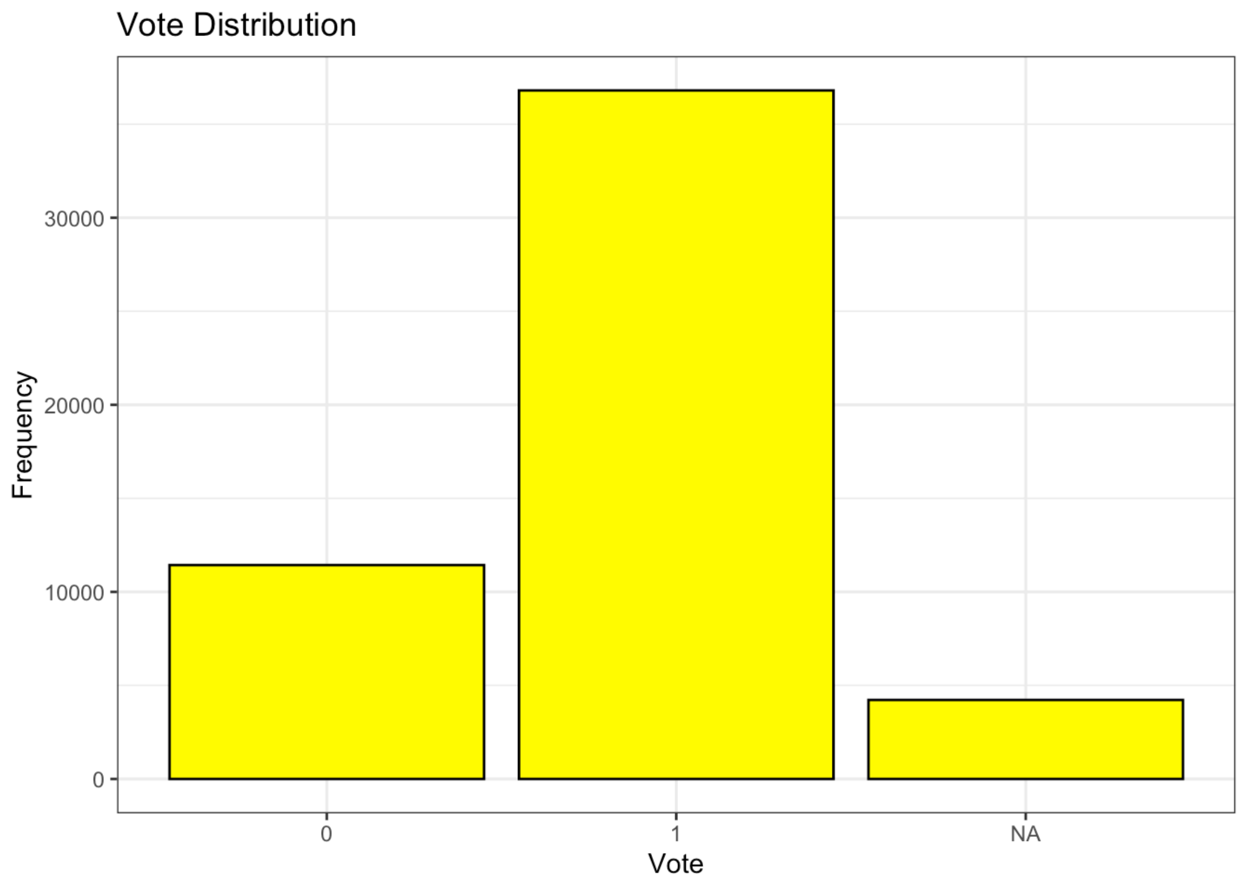
```
#income_10
ggplot(ESS_Polity, aes(x = factor(income_10))) +
  geom_bar(fill = "green", color = "black") +
  labs(title = "Income Bracket Distribution", x = "Income Bracket", y = "Income"
  theme_bw()
```




```
print("I use a barchart to plot 'income_10' because it is a categorical variable
```

```
[1] "I use a barchart to plot 'income_10' because it is a categorical variable
as there are income brackets to tell information about the income of a person."
```

```
#vote
ggplot(ESS_Polity, aes(x = factor(vote))) +
  geom_bar(fill = "yellow", color = "black") +
  labs(title = "Vote Distribution", x = "Vote", y = "Frequency") +
  theme_bw()
```

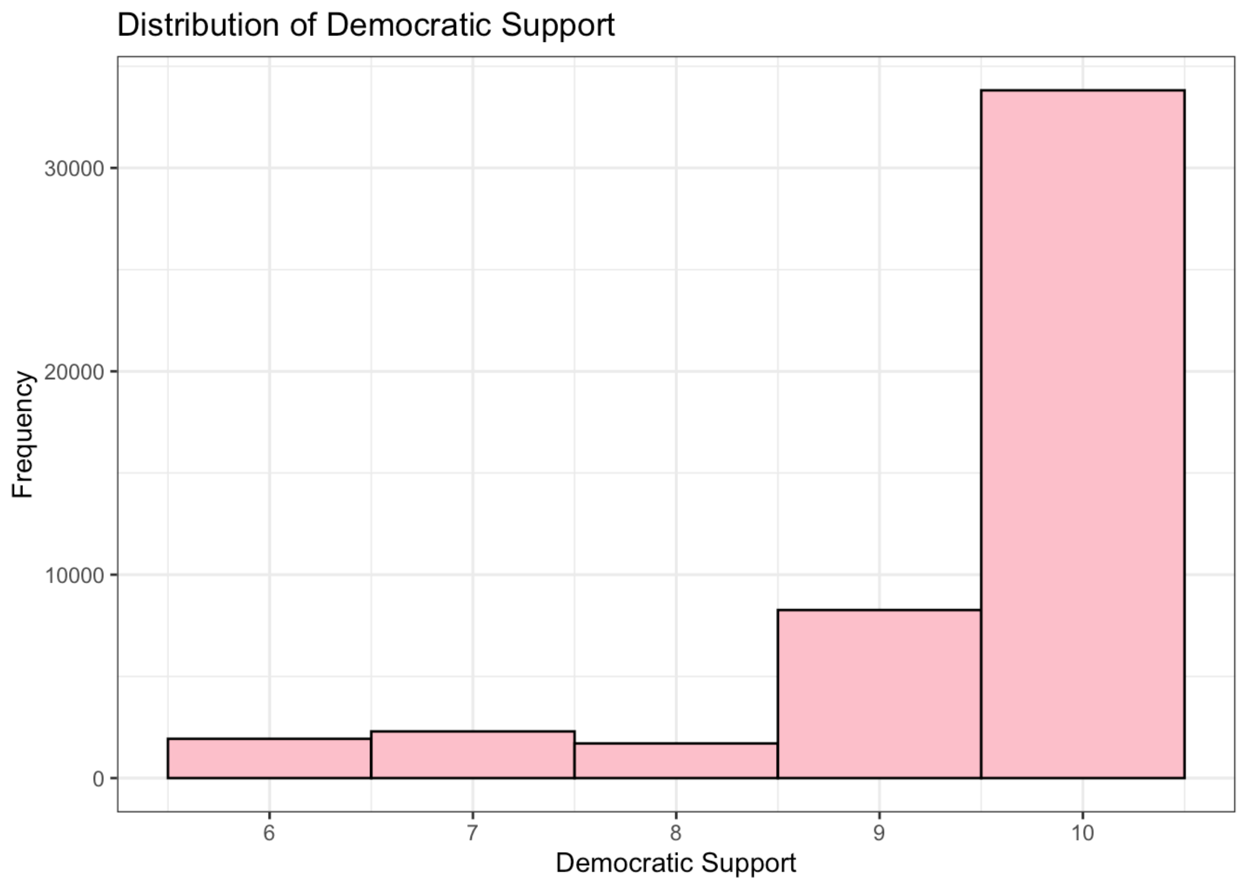



```
print("I use a barchart to plot 'vote' because it is a categorical variable with
```

```
[1] "I use a barchart to plot 'vote' because it is a categorical variable with
distinct categories (i.e., 1, 2, 3)."
```

```
#democ
ggplot(ESS_Polity, aes(x = democ)) +
  geom_histogram(binwidth = 1, fill = "pink", color = "black") +
  labs(title = "Distribution of Democratic Support", x = "Democratic Support", y
  theme_bw()
```

```
Warning: Removed 4451 rows containing non-finite values (`stat_bin()`).
```



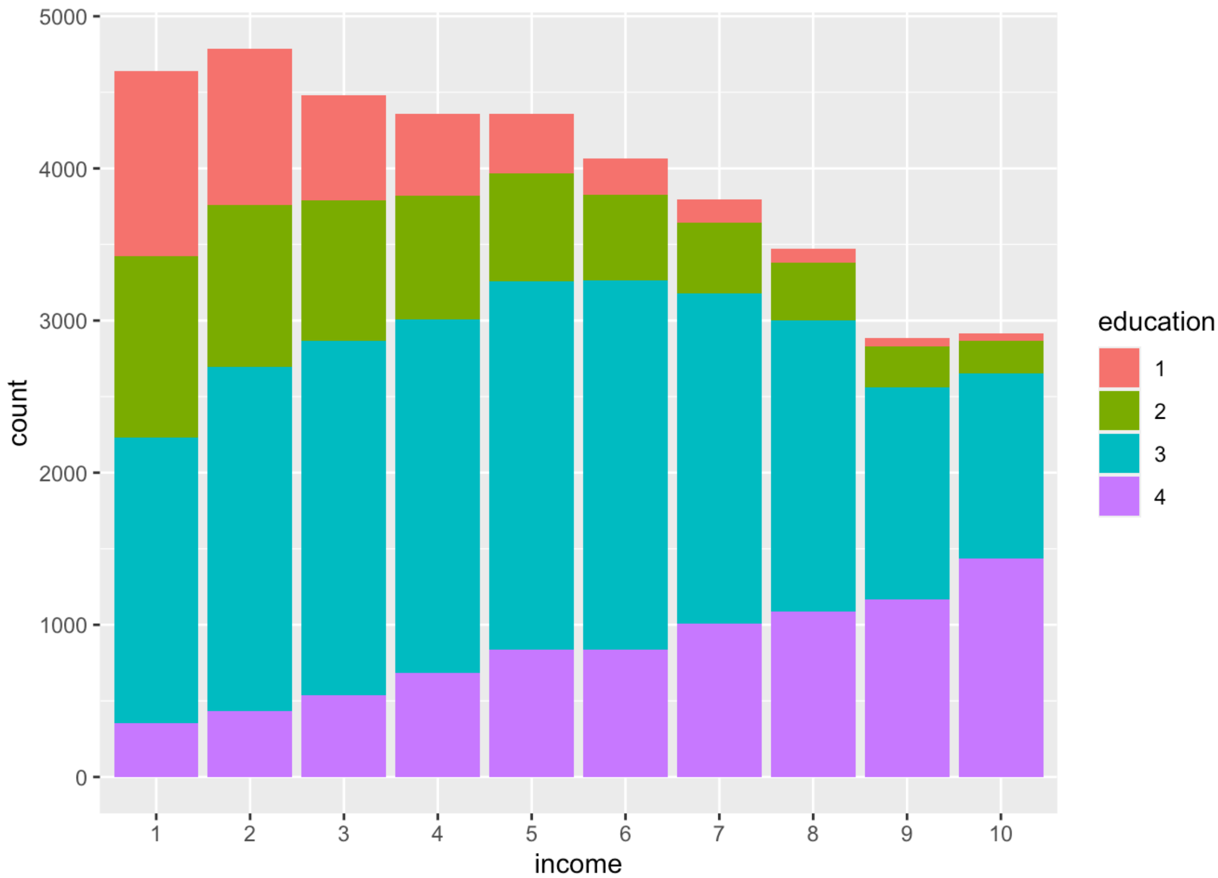

```r
print("I use a histogram to plot 'democ' because it is a numeric variable.")
```

```
[1] "I use a histogram to plot 'democ' because it is a numeric variable."
```

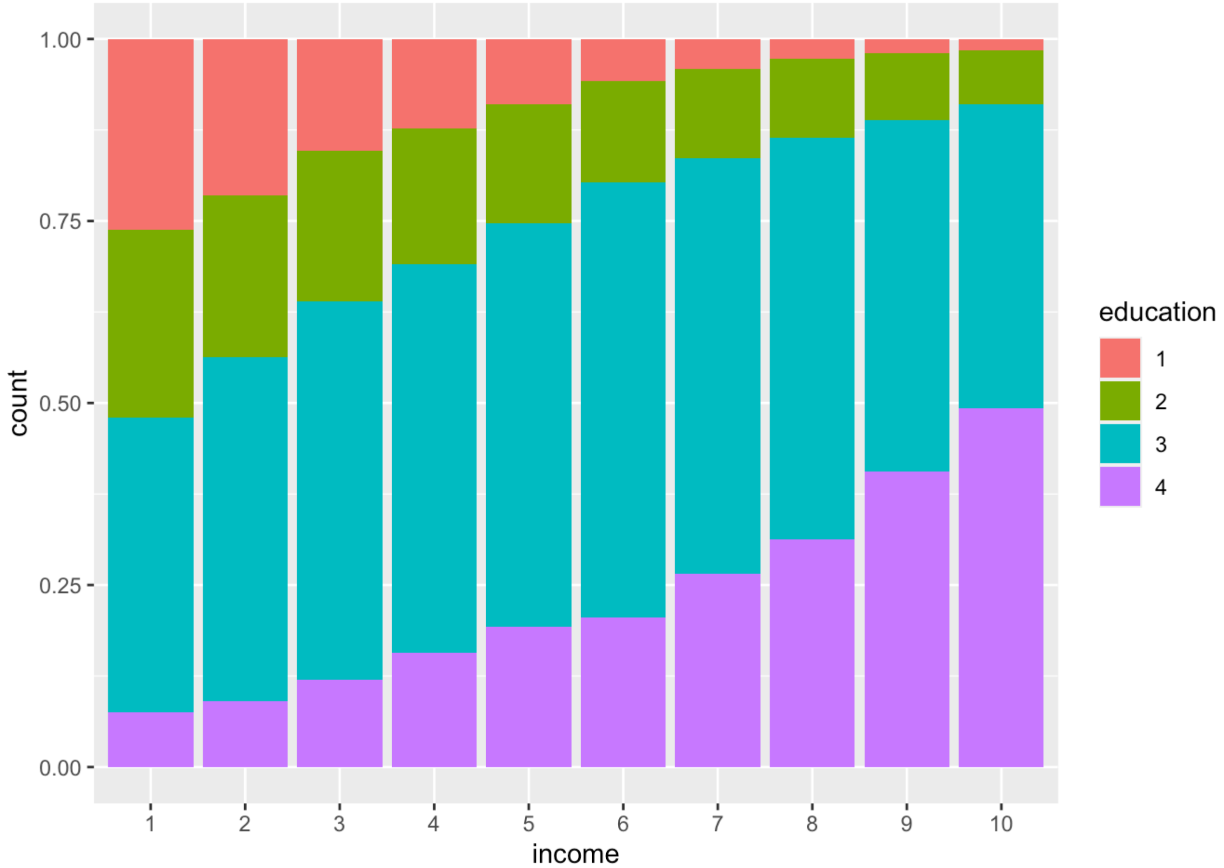
3. **Suppose we want to test two hypotheses on the relationships of two pairs of variables**. **Please use the appropriate type of graphs we learned to visualize these two pairs of variables. Briefly describe the graph you plot, and answer: Does the graph we create from the data support the hypothesis?**

   (1) Hypothesis#1: The more years of education (edu) a person completed, the higher income (income_10) they earn. **(7.5%)**

```r
ESS_Polity|>
subset(!is.na(income_10))|> #remove na in income
subset(!is.na(edu))|> #remove na in edu
ggplot(aes(x = as.factor(income_10), fill = as.factor(edu))) +
geom_bar(position="stack") +
labs(x="income",fill="education")
```




```
ESS_Polity|>
subset(!is.na(income_10))|> #remove na in income
subset(!is.na(edu))|> #remove na in edu
ggplot(aes(x = as.factor(income_10), fill = as.factor(edu))) +
geom_bar(position="fill") +
labs(x="income",fill="education")
```

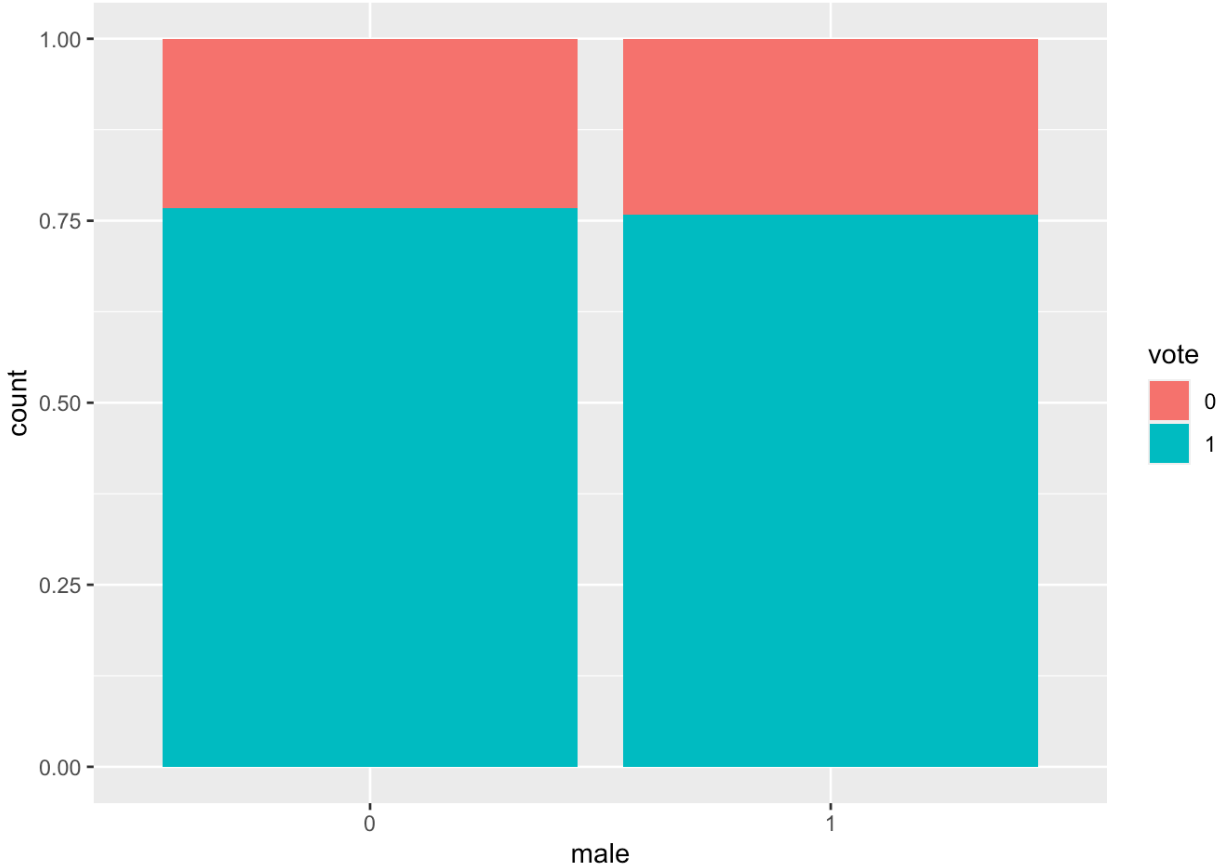



```
print("I have plotted a box plot to support this hypothesis as it is a graph of
```

```
[1] "I have plotted a box plot to support this hypothesis as it is a graph of
categorical values vs the numerical values. As we can see from the graph, as the
levels of education increase, the income f the individuals increases."
```

(2) Hypothesis#2: There is a gender disparity (male) in voting behavior (vote). (Either men are more likely to vote, or women are more likely to vote). **(7.5%)**

```
#type of your code/command here.
ESS_Polity|>
subset(!is.na(male))|> #remove na in income
subset(!is.na(vote))|>
ggplot(aes(x = as.factor(male), fill = as.factor(vote))) +
geom_bar(position="fill") +
labs(x = "male", fill = "vote")
```




```
ESS_Polity|>
group_by(male)|>
subset(!is.na(male))|>
subset(!is.na(vote))|>
summarise(mean(vote))
```

male <dbl>	mean(vote) <dbl>
0	0.7667045
1	0.7585367

2 rows

```
print("I have plotted a graph of categorical values vs count of the votes. This
```

```
[1] "I have plotted a graph of categorical values vs count of the votes. This is
a bar graph which is a univariate graph. The average voter turnout for both
males and females is quite similar. This suggests that a person's gender doesn't
influence their decision to vote. Using group_by and summarise(), we find that
the turnout rate for females is 0.767 and for males it's 0.759. Therefore, it
seems that the second hypothesis is correct."
```

Part 2. Comparing between Partial and Whole, and among Groups (30%)

In this part, we will use the clean version of the Australian public opinion poll on Same-Sex Marriage to generate graphs and plots. **You may need to do the data transformation or mutation needed to help graphing.**

1. Read in data. **(2.5%)**

```
#type of your code/command here.
australian_data <- read_csv("~/Desktop/DACSS 601/DACSS_601_datasets/australian_d
```

```
New names:
Rows: 150 Columns: 7
── Column specification
──────────────────────────────────────────────────────── Delimiter: "," chr
(2): District, Division dbl (5): ...1, Yes, No, Illegible, No Response
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
• `` -> `...1`
```

```
head(australian_data)
```

...1	District	Yes	No	Illegible	No Response
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Banks	37736	46343	247	20928
2	Barton	37153	47984	226	24008
3	Bennelong	42943	43215	244	19973
4	Berowra	48471	40369	212	16038
5	Blaxland	20406	57926	220	25883
6	Bradfield	53681	34927	202	17261

6 rows | 1-6 of 7 columns

2. Use a barplot to graph the Australian data based on their responses: yes, no, illegible, and no response. The y-axis should be the count of responses, and each response should be represented by one individual bar (so there should be four bars). **(7.5%)**

   (you can use either geom_bar() or geom_col())

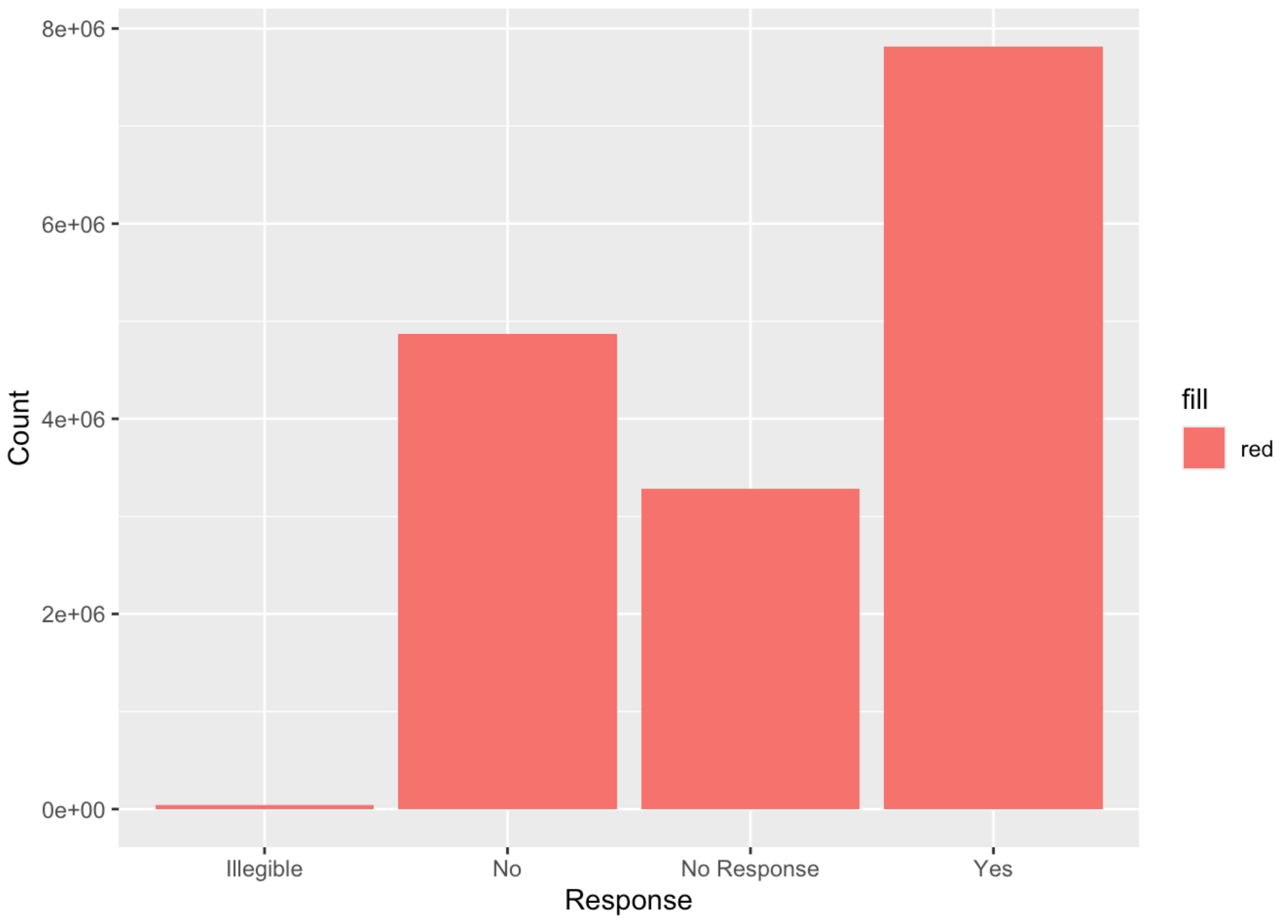
```
aus_data <- australian_data %>%
  pivot_longer(cols = c(Yes, No, Illegible, `No Response`),
               names_to = "Response",
               values_to = "Count")

aus_data
```

...1	District	Division	Response	Cou
<dbl>	<chr>	<chr>	<chr>	<db
1	Banks	New South Wales Divisions	Yes	3773
1	Banks	New South Wales Divisions	No	4634
1	Banks	New South Wales Divisions	Illegible	24

...1	District	Division	Response	Cou
<dbl>	<chr>	<chr>	<chr>	<db
1	Banks	New South Wales Divisions	No Response	2092
2	Barton	New South Wales Divisions	Yes	3715
2	Barton	New South Wales Divisions	No	4798
2	Barton	New South Wales Divisions	Illegible	22
2	Barton	New South Wales Divisions	No Response	2400
3	Bennelong	New South Wales Divisions	Yes	4294
3	Bennelong	New South Wales Divisions	No	4321

1-10 of 600 rows Previous 1 2 3 4 5 6 ... 60 Next

```
ggplot(aus_data, aes(x = Response, y = Count, fill = "red")) +
  geom_col()
```



3. The previous graph only shows the difference in amount. Let's create a stacked-to-100% barplot to show the proportion of each of the four responses (by % of the total response). **(7.5%)**

   (you can use either geom_bar() or geom_col())

```
australian_data_proportions <- australian_data %>%
  summarise(
    yes_pct = sum(Yes)/sum(Yes, No, Illegible, `No Response`) * 100,
    No_pct = sum(No)/sum(Yes, No, Illegible, `No Response`) * 100,
    Illegible_pct = sum(Illegible)/sum(Yes, No, Illegible, `No Response`) * 100,
```

```r
    `No Response_pct` = sum(`No Response`)/sum(Yes, No, Illegible, `No Response`) *
    na.rm = TRUE
  )
australian_data_proportions
```

yes_pct	No_pct	Illegible_pct	No Response_pct	na.rm
<dbl>	<dbl>	<dbl>	<dbl>	<lgl>
48.83893	30.45066	0.229199	20.48121	TRUE

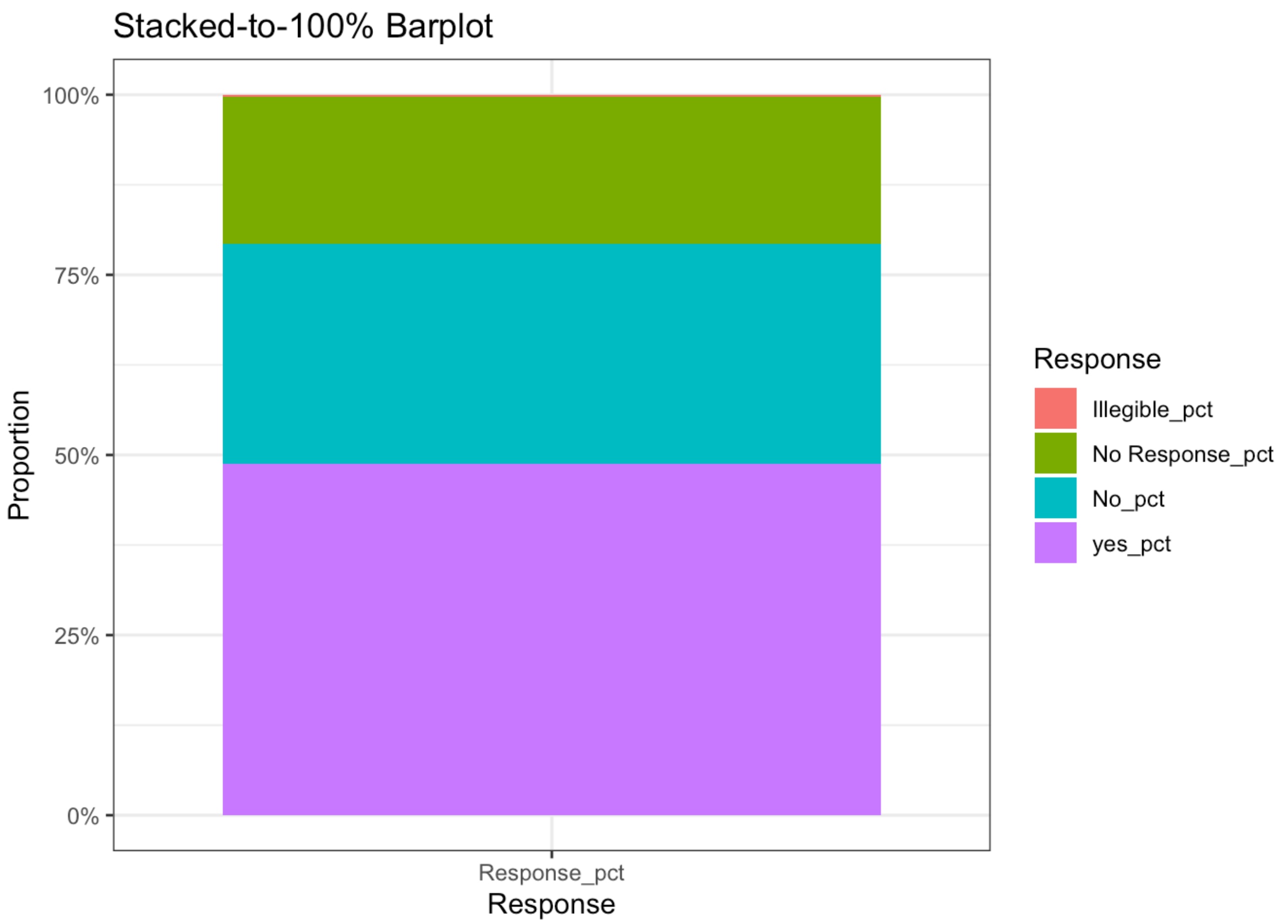
1 row

```r
australian_data_longer <- australian_data_proportions %>%
  pivot_longer(cols = c(yes_pct, No_pct, Illegible_pct, `No Response_pct`),
               names_to = "Response",
               values_to = "Proportion")
australian_data_longer
```

na.rm	Response	Proportion
<lgl>	<chr>	<dbl>
TRUE	yes_pct	48.838930
TRUE	No_pct	30.450657
TRUE	Illegible_pct	0.229199
TRUE	No Response_pct	20.481214

4 rows

```r
ggplot(australian_data_longer, aes(x = "Response_pct", y = Proportion, fill = Respor
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stacked-to-100% Barplot",
       x = "Response",
       y = "Proportion") +
  theme_bw()
```



4. Let's see if there's a relationship between Division and Response - that is, are certain divisions more likely to respond one way compared to other divisions? Again, we will use barplot(s) to present the visualization. **(12.5%)**

(you can use either geom_bar() or geom_col())

```
#type of your code/command here.
australian_data_proportions <- australian_data %>%
  group_by(Division) %>%
  summarise(
yes_pct = sum(Yes)/sum(Yes, No, Illegible, `No Response`) * 100,
No_pct = sum(No)/sum(Yes, No, Illegible, `No Response`) * 100,
Illegible_pct = sum(Illegible)/sum(Yes, No, Illegible, `No Response`) * 100,
`No Response_pct` = sum(`No Response`)/sum(Yes, No, Illegible, `No Response`) *
na.rm = TRUE
  )
australian_data_proportions
```

Division <chr>	yes_pct <dbl>	No_pct <dbl>
Australian Capital Territory Divisions	60.90043	21.35310
New South Wales Divisions	45.76924	33.48005
Northern Territory Divisions	35.25391	22.94697
Queensland Divisions	47.19517	30.49996
South Australia Divisions	49.64292	29.84693
Tasmania Divisions	50.58878	28.90008

Division	yes_pct	No_pct
<chr>	<dbl>	<dbl>
Victoria Divisions	52.82993	28.58869
Western Australia Divisions	49.87959	28.37077

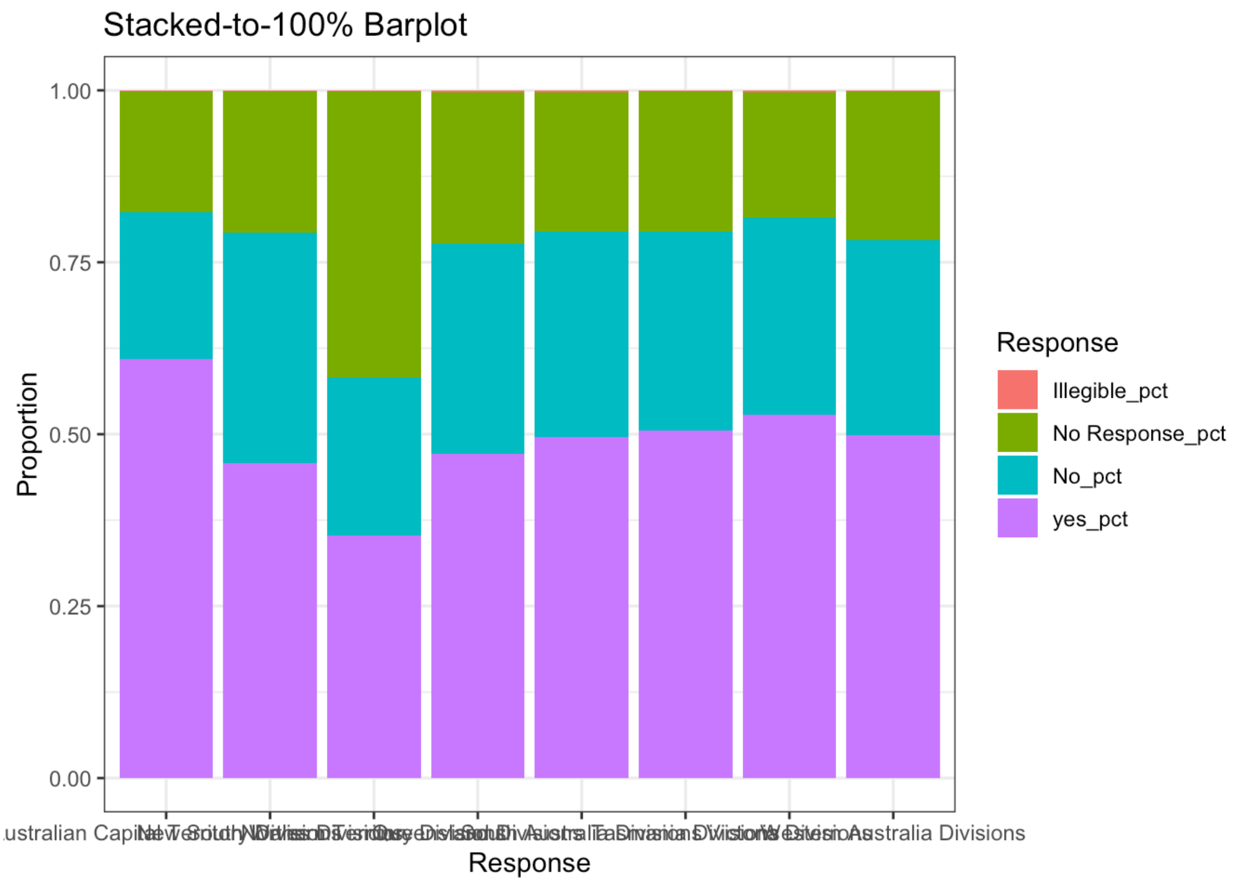
8 rows | 1-3 of 6 columns

```
australian_data_longer <- australian_data_proportions %>%
  pivot_longer(cols = c(yes_pct, No_pct, Illegible_pct, `No Response_pct`),
           names_to = "Response",
           values_to = "Proportion")
australian_data_longer
```

Division	na.rm	Response
<chr>	<lgl>	<chr>
Australian Capital Territory Divisions	TRUE	yes_pct
Australian Capital Territory Divisions	TRUE	No_pct
Australian Capital Territory Divisions	TRUE	Illegible_pct
Australian Capital Territory Divisions	TRUE	No Response_pct
New South Wales Divisions	TRUE	yes_pct
New South Wales Divisions	TRUE	No_pct
New South Wales Divisions	TRUE	Illegible_pct
New South Wales Divisions	TRUE	No Response_pct
Northern Territory Divisions	TRUE	yes_pct
Northern Territory Divisions	TRUE	No_pct

1-10 of 32 rows | 1-3 of 4 columns Previous **1** [2](#) [3](#) [4](#) [Next](#)

```
ggplot(australian_data_longer, aes(x = Division, y = Proportion, fill = Response
  geom_bar(stat = "identity", position = "fill") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Stacked-to-100% Barplot",
   x = "Response",
   y = "Proportion") +
  theme_bw()
```



Part 3. Practice plotting with a dataset of your choice (25% of the total grade)

In this part, you will choose data of your interests for graphing and plotting. This data can be tidy/ready-to-be-used or raw data that needs cleaning. If the data is very large (for example, more than 20 columns), you should definitely subset the data by selecting less than 10 variables of your interests to avoid taking too much room in your R memory.

1. Include a link to the data page (this page should include the introduction or description and the link to download this dataset). **(2%)**

2. Read the data you choose and briefly answer the following questions. (Optional: you may need to subset, clean, and transform the data if necessary). **(8%)**

```
#type of your code/command here.
titanic <- read_csv("~/Desktop/DACSS 601/DACSS_601_datasets/titanic.csv")
```

```
Rows: 418 Columns: 12
── Column specification ────────────────────────────────────────────────
Delimiter: ","
chr (5): Name, Sex, Ticket, Cabin, Embarked
dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(titanic)
```

	PassengerId <dbl>	Survived <dbl>	Pclass <dbl>
	892	0	3
	893	1	3
	894	0	2
	895	0	3
	896	1	3
	897	0	3

6 rows | 1-3 of 12 columns

(1) What is the structure (dimension) of the data;

(2) What is the unit of observation?

(3) What does each column mean in this data?

```r
#\(1\) What is the structure (dimension) of the data;
    dimensions <- dim(titanic)
    print(dimensions)
```

```
[1] 418  12
```

```r
    print("The titanic dataset contains 418 rows and 12 columns")
```

```
[1] "The titanic dataset contains 418 rows and 12 columns"
```

```r
#\(2\) What is the unit of observation?
print("The unit of observation in this dataset is each passenger.")
```

```
[1] "The unit of observation in this dataset is each passenger."
```

```r
#\(3\) What does each column mean in this data?
print("Each column in the data tells about whether a particular passenger survived c
```

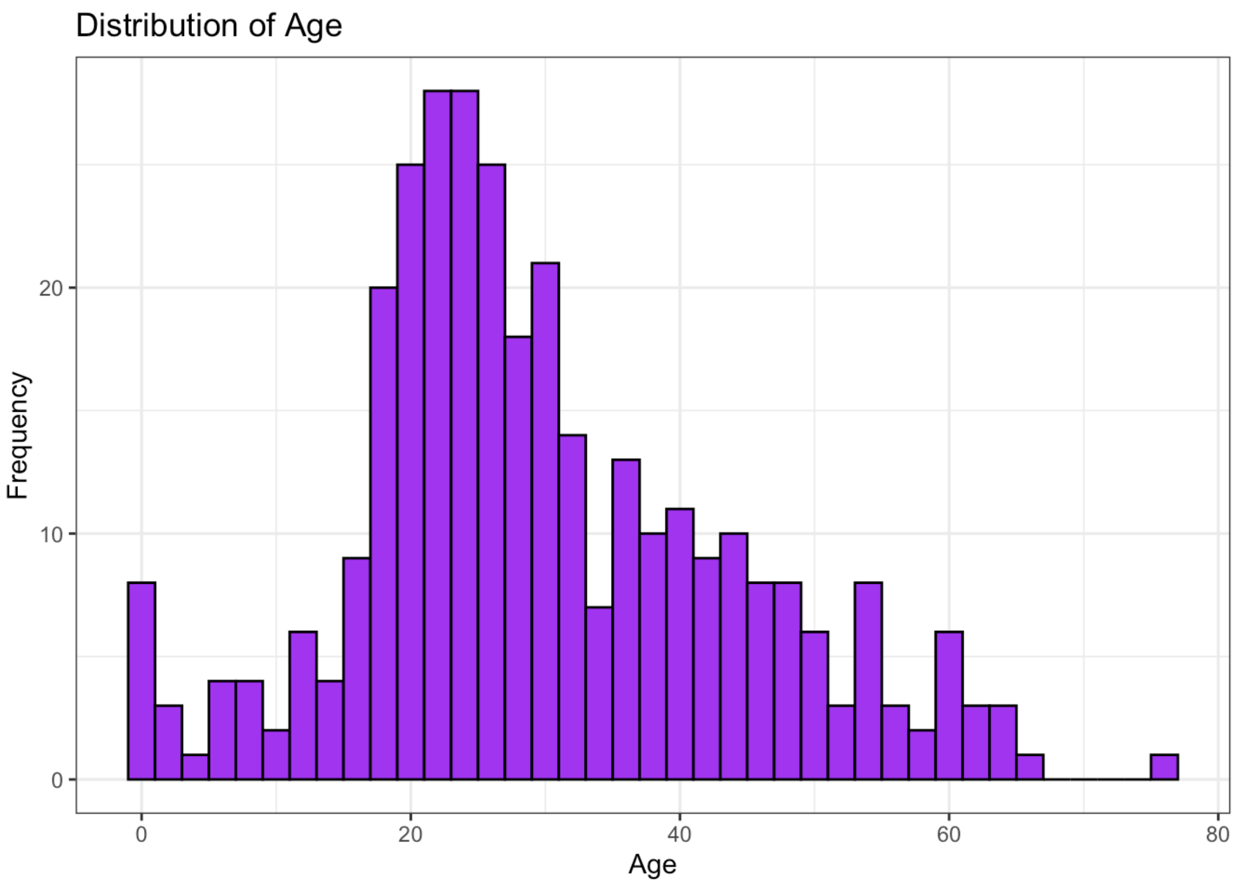
```
[1] "Each column in the data tells about whether a particular passenger survived or
not, their age, their ticket number, fare of their ticket, which class they belonged
to and where their cabin was located."
```

3. Choose two columns/variables of your interests. Plot one univariate graph for each of the variables. **(5%)**

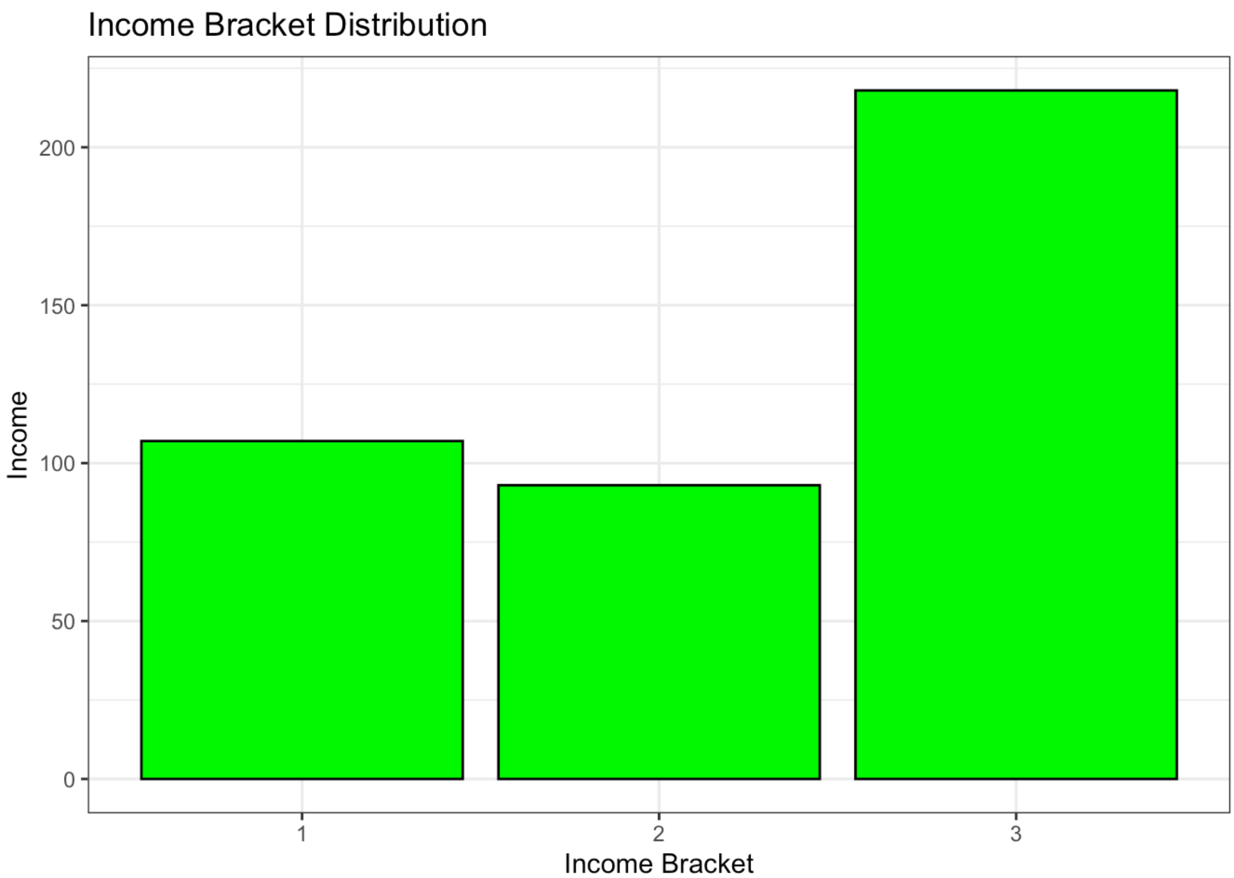
```r
#type of your code/command here.
#column 1: AGE
ggplot(titanic, aes(x = Age)) +
  geom_histogram(binwidth = 2, fill = "purple", color = "black") +
```

```
  labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
  theme_bw()
```

```
Warning: Removed 86 rows containing non-finite values (`stat_bin()`).
```

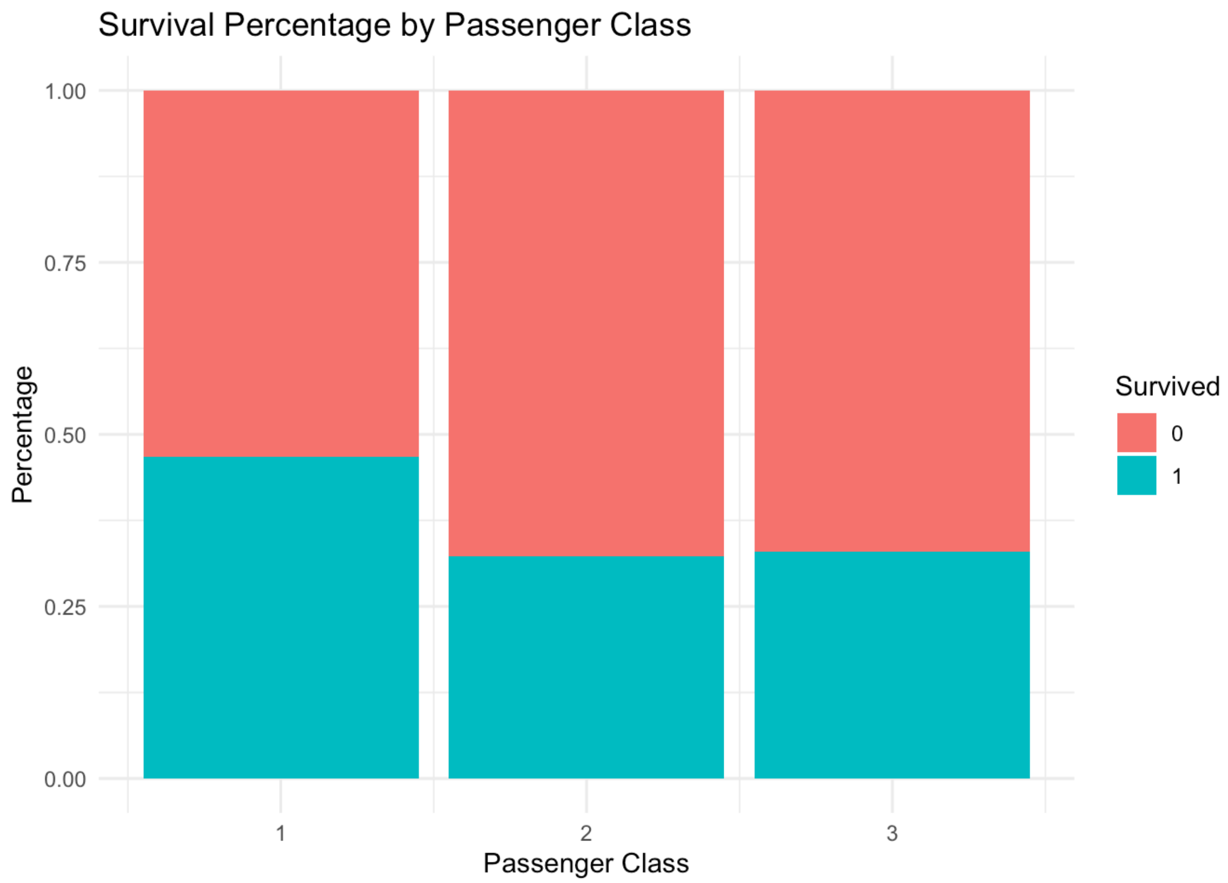



```
#column 2: Passenger's class -> Pclass
ggplot(titanic, aes(x = factor(Pclass))) +
  geom_bar(fill = "green", color = "black") +
  labs(title = "Income Bracket Distribution", x = "Income Bracket", y = "Income"
  theme_bw()
```



4. Choose a pair of variables that may be correlated and make a graph (scatter plot or barplot) using them. Based on the visual evidence, do you see any potential correlation between the two variables **(10%)**

```
ggplot(titanic, aes(x = Pclass, fill = factor(Survived))) +
  geom_bar(position = "fill") +  # Change position to "fill"
  labs(title = "Survival Percentage by Passenger Class",
   x = "Passenger Class",
   y = "Percentage",
   fill = "Survived") +
  theme_minimal()
```




```
print("By looking at the graph I can clearly say that there are more number of s
```

```
[1] "By looking at the graph I can clearly say that there are more number of
survivors which belong to the class 3 as compared to the other classes. There is
no significant pattern observed. There are almost equal number of survivors in
class 1 as the people who did not survive. Comparatively there are more
survivors in class 2."
```

Appendix: sources for data to be used in Part 3

Here are some online sources and popular Online Dataset Hub:

1. Many US governments (usually at the federal and state levels), bureaus, and departments have open data archives on their websites, allowing the public to access, download, and use them. Just use Google to search for them.

2. **The Harvard Dataverse Repository** is a free data repository open to all researchers from any discipline, inside and outside the Harvard community, where you can share, archive, cite, access, and explore research data. Each individual Dataverse collection is a customizable collection of datasets (or a virtual repository) for organizing, managing, and showcasing datasets.

3. **Inter-university Consortium for Political and Social Research (ICPSR)** of the University of Michigan-Ann Arbor provides leadership and training in data access, curation, and methods of analysis for the social science research community.

4. **UN: https://data.un.org/**

5. **OECD Data**: economic and development data of the most developed countries in the world.

6. The five sources above are mainly for social science data; **there exists another very big community and open data archives for machine-learning and data science: Kaggle.**