



FINAL TERM PROJECT - FALL (2022-23)

PROJECT TITLE: WEB SCRAPPING

COURSE: INTRODUCTION TO DATA SCIENCE

INSTRUCTOR: AKINUL ISLAM JONY

SUBMITTED BY:

NAME: HIMEL DATTA

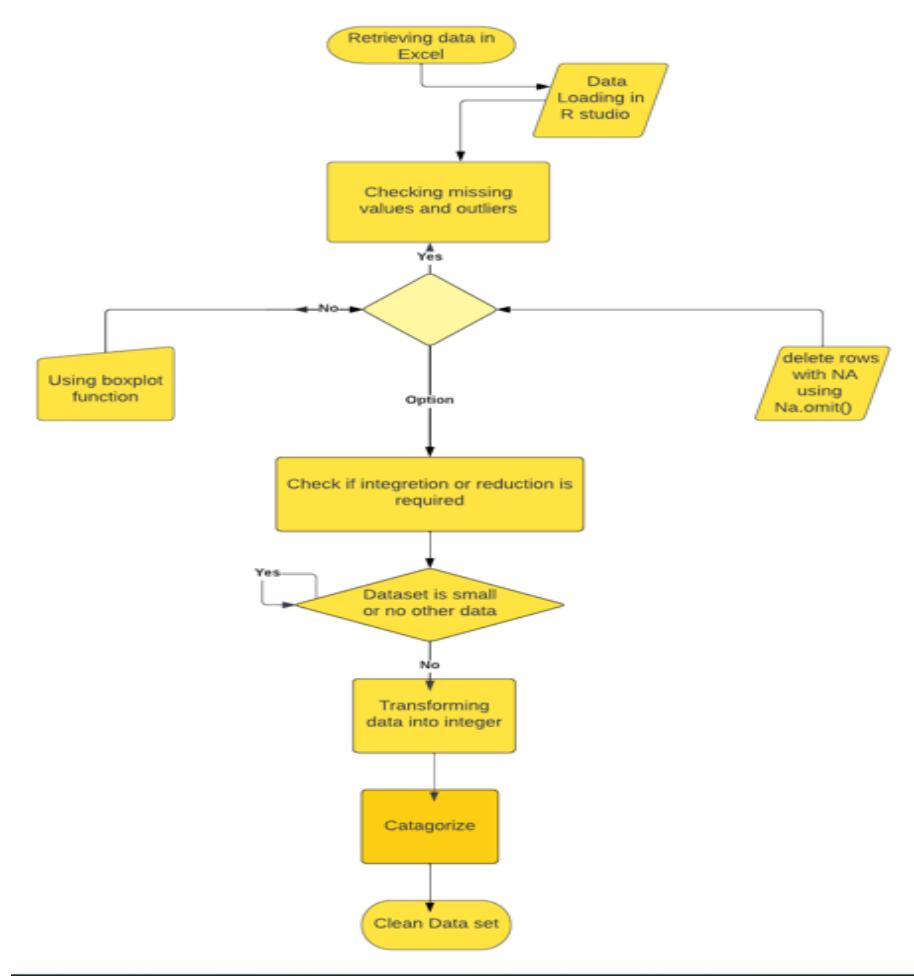
ID: 19-41576-3

SEC:C

Project Overview:

In this project I have worked on web-scraping, Data pre-processing, Descriptive statistics and Data visualization. So, first of all I have scrapped a data table from a website by using regarding R codes on R studio. Then I have loaded this data table as in csv format. After that I have used Data Cleaning methods to process those data and to clean them. Then I have calculated different statistics values for different variables. Finally, I have graphed bar charts to do Data Visualization. For those process I have to install several libraries.

Project Design:



Web Scrapping:

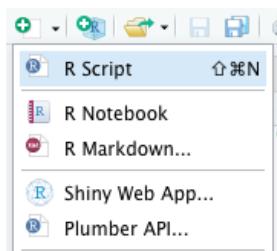
Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, Stack Overflow, etc. have API's that allow to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Scraping tabular data is one of the most powerful skills data scientists can have to gather relevant information at scale. Nowadays, almost every set of data shown on the internet uses HTML tables to organize and display complex set of data in a more understandable format.

Here I'm going to use the Rvest package to extract data from HTML tables and send it to a data frame for further analysis, and import it to a CSV file.

Step 1: Set Up Your Development Environment

The first thing I need to do is creating a new directory for our project. Open RStudio and click on “create a project”. Inside the new folder, click on “new file” and create a new R Script. Let's named Scrapping_5 (it'll automatically save it as a .R file).



On our new file, we'll install the Rvest, Dplyr, xml2, tidyverse packages using the following commands:

```
install.packages('rvest')
install.packages('dplyr')
installed.packages('xml2')
install.packages("tidyverse")

library(rvest)
library(dplyr)
library(xml2)
library(tidyverse)
```

Rvest: rvest helps to scrape information from web pages.

Dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation.

Xml: helps to read files.

It can take a couple of minutes to install, but with this last step, we're ready to create our scraper.

Step 2: Select the website and send initial request to the web

So first, let's take a look at our target site:

I have selected my target website and its URL is: <https://www.patriotsoftware.com> . Now, I have chosen a Data Table named “U.S. cost of living comparison by state” from this website.

Here is Data Table which I have selected for scraping.

The screenshot shows a web browser window with the URL <https://www.patriotsoftware.com/blog/accounting/average-cost-living-by-state/>. The page title is "Patriot". The main content is a data table titled "U.S. cost of living comparison by state". The table has four columns: State, Annual Mean Wage (All Occupations), Average Monthly Rent, and Value of \$100. The table lists data for various states, with a scroll bar indicating more rows below. Below the table, there is another smaller table with similar columns and data for different states.

State	Annual Mean Wage (All Occupations)	Average Monthly Rent	Value of \$100
Alabama	\$48,110	\$914.08	\$114.20
Alaska	\$63,480	\$1,288.42	\$94.90
Arizona	\$55,170	\$1,208.50	\$103.70
Arkansas	\$46,500	\$778.92	\$115.30
California	\$68,510	\$1,777.67	\$83.60
Colorado	\$62,900	\$1,404.33	\$98.10
Connecticut	\$66,130	\$1,315.42	\$95.00
Delaware	\$59,820	\$1,340.08	\$100.60
Florida	\$51,950	\$1,329.92	\$99.00

State	Annual Mean Wage (All Occupations)	Average Monthly Rent	Value of \$100
Georgia	\$53,940	\$1,083.75	\$106.80
Hawaii	\$59,760	\$1,880.08	\$80.70
Idaho	\$47,940	\$901.75	\$107.80
Illinois	\$59,650	\$1,049.75	\$102.60
Indiana	\$50,440	\$889.25	\$111.30
Iowa	\$51,140	\$831.17	\$111.00
Kansas	\$49,680	\$879.83	\$110.80
Kentucky	\$48,170	\$859.25	\$112.60
Louisiana	\$47,740	\$869.75	\$112.10
Maine	\$53,230	\$1,024.67	\$100.70
Maryland	\$65,900	\$1,505.33	\$92.30

A screenshot of a Microsoft Windows desktop. At the top, there's a taskbar with various icons for programs like File Explorer, Google Chrome, and others. Below the taskbar is a window showing a table from a website. The table has two main sections. The first section contains data for states starting with M through N, and the second section contains data for states starting with S through W. Each row in the table includes four columns: state name, average cost of living, median household income, and a third column that appears to be a ratio or percentage.

Maryland	\$65,900	\$1,505.33	\$92.30
Massachusetts	\$72,940	\$1,553.75	\$89.60
Michigan	\$55,160	\$954.25	\$107.70
Minnesota	\$60,480	\$1,054.33	\$102.00
Mississippi	\$42,700	\$868.42	\$115.60
Missouri	\$51,390	\$909.00	\$111.30
Montana	\$49,340	\$919.67	\$106.50
Nebraska	\$52,110	\$899.83	\$110.50
Nevada	\$51,080	\$1,245.17	\$102.60
New Hampshire	\$59,270	\$1,213.50	\$93.50
New Jersey	\$67,120	\$1,541.08	\$84.00
North Carolina	\$53,100	\$1,008.92	\$108.30
North Dakota	\$53,380	\$853.83	\$110.70
Ohio	\$53,170	\$865.17	\$111.60
Oklahoma	\$48,360	\$847.92	\$112.80
Oregon	\$59,070	\$1,226.58	\$97.80
Pennsylvania	\$55,490	\$1,001.42	\$103.00
Rhode Island	\$62,120	\$1,178.08	\$98.70
South Carolina	\$47,490	\$1,004.50	\$108.50
South Dakota	\$46,810	\$804.83	\$112.20
Tennessee	\$49,330	\$979.92	\$110.30
Texas	\$54,230	\$1,090.83	\$103.50
Utah	\$53,400	\$1,171.08	\$103.50
Vermont	\$55,450	Not provided	\$96.90
Virginia	\$62,330	\$1,351.33	\$98.70
Washington	\$68,740	\$1,487.58	\$91.60
West Virginia	\$46,490	\$786.83	\$112.90
Wisconsin	\$53,120	\$943.58	\$108.10
Wyoming	\$52,110	\$876.92	\$107.20

A screenshot of a Microsoft Windows desktop, showing a second instance of a web browser window. This window also displays the same table of average cost of living by state. The table structure is identical to the one in the first screenshot, listing states from South Dakota to Wyoming with their respective average costs of living, median household incomes, and ratios.

South Dakota	\$46,810	\$804.83	\$112.20
Tennessee	\$49,330	\$979.92	\$110.30
Texas	\$54,230	\$1,090.83	\$103.50
Utah	\$53,400	\$1,171.08	\$103.50
Vermont	\$55,450	Not provided	\$96.90
Virginia	\$62,330	\$1,351.33	\$98.70
Washington	\$68,740	\$1,487.58	\$91.60
West Virginia	\$46,490	\$786.83	\$112.90
Wisconsin	\$53,120	\$943.58	\$108.10
Wyoming	\$52,110	\$876.92	\$107.20

So now for retrieving this table first I need to copy the URL link and send it through `read_html()` function and also store it on a variable. Then, by using `html_table()` function I have stored the link in a new variable `table`. After that I have pointed that as I'm going to retrieve table number 1 and store it in `first_table` variable. And now I have viewed that table.

```
content <- read_html("https://www.patriotsoftware.com/blog/accounting/average-cost-living-by-state/")

table <- content %>% html_table(fill = TRUE)

first_table <- table[[1]]

View(first_table)
```

The screenshot shows the RStudio interface. In the top-left pane, there is a code editor with the following R script:

```

1 install.packages("rvest")
2 install.packages("dplyr")
3 installed.packages("xml2")
4 install.packages("tidyverse")
5
6 library(rvest)
7 library(dplyr)
8 library(xml2)
9 library(tidyverse)
10
11 content <- read_html("https://www.patriotsoftware.com/blog/accounting/average-cost-living-by-state/")
12
13 table <- content %>% html_table(fill = TRUE)
14 first_table <- table[[1]]
15
16 View(first_table)
17
18 <--
```

In the bottom-left pane, the Console tab is active, showing the execution of the script and its output:

```

> Content <- read_html("https://www.patriotsoftware.com/blog/accounting/average-cost-living-by-state/")
> table <- Content %>% html_table(fill = TRUE)
Error in html_table(., fill = TRUE) : object 'Content' not found
> table <- Content %>% html_table(fill = TRUE)
> first_table <- table[[1]]
> View(first_table)
> setwd("E:/10th semester/Introduction To Data Science/Final/project")
> write.csv(first_table, file = "T4_dataset.csv")
> first_table <- read.csv("T4_dataset.csv")
> View(first_table)
```

The right side of the interface features a file browser with a tree view of files and a list view of selected files. The list view includes:

- content: List of 2
- df: 49 obs. of 6 variables
- first_table: 56 obs. of 6 variables
- New_dataset: 49 obs. of 5 variables
- table: List of 1

Retrieved table with lots of NA (Not Available) or missing values.

The screenshot shows the RStudio interface. In the top-left pane, there is a data grid displaying a table with columns: Index, States, Annual_mean_wage, Average_monthly_rent, Value_of_\$100, and X. The data consists of 56 rows of US state information. The 'X' column contains many NA values.

Index	States	Annual_mean_wage	Average_monthly_rent	Value_of_\$100	X
1	Alabama	48110	914	114	NA
2	Alaska	63460	1288	95	NA
3	Arizona	55170	1209	104	NA
4	Arkansas	46500	779	115	NA
5	California	68510	1778	84	NA
6	Colorado	62900	1404	98	NA
7	Connecticut	66130	1315	95	NA
8	Delaware	59820	1346	101	NA
9	Florida	51950	1330	99	NA
10	Georgia	53940	1084	107	NA
11	Hawaii	59760	1880	81	NA
12	Idaho	47940	902	108	NA
13	Illinois	59650	1050	103	NA
14	Indiana	50440	889	111	NA
15	Iowa	51140	831	111	NA
16	Kansas	49600	880	111	NA
17	Kentucky	48170	859	113	NA
18	Louisiana	47740	870	112	NA
19	Maine	53230	1025	101	NA
20	Maryland	65900	1505	92	NA
21	Massachusetts	72940	1554	90	NA
22	Michigan	55200	1150	104	NA
23	Minnesota	60400	1150	104	NA
24	Mississippi	47700	1025	101	NA
25	Missouri	50400	1025	101	NA
26	Montana	47700	1025	101	NA
27	Nebraska	49600	1025	101	NA
28	Nevada	55200	1150	104	NA
29	New Hampshire	59820	1346	101	NA
30	New Jersey	68510	1778	84	NA
31	New Mexico	47700	1025	101	NA
32	New York	68510	1778	84	NA
33	Pennsylvania	62900	1404	98	NA
34	Rhode Island	59820	1346	101	NA
35	Tennessee	50440	889	111	NA
36	Vermont	53230	1025	101	NA
37	Virginia	65900	1505	92	NA
38	Washington	72940	1554	90	NA
39	West Virginia	47700	1025	101	NA
40	Wisconsin	55200	1150	104	NA
41	Wyoming	47700	1025	101	NA

The right side of the interface features a file browser with a tree view of files and a list view of selected files. The list view includes:

- content: List of 2
- df: 49 obs. of 6 variables
- first_table: 56 obs. of 6 variables
- New_dataset: 49 obs. of 5 variables
- table: List of 1

Showing 21 to 43 of 56 entries, 6 total columns

Index	States	Annual_mean_wage	Average_monthly_rent	Value_of_\$100	X
22	Michigan	55160	954	108	NA
23	Minnesota	60480	1054	102	NA
24	Mississippi	42700	866	116	NA
25	Missouri	51390	909	111	NA
26	Montana	49340	920	107	NA
27	Nebraska	52110	900	111	NA
28	Nevada	51080	1245	103	NA
29	New Hampshire	59270	1214	94	NA
30	New Jersey	67120	1541	84	NA
31	New Mexico	51860	901	109	NA
32	New York	70460	1432	84	NA
33	North Carolina	53100	1009	108	NA
34	North Dakota	53380	854	111	NA
35	Ohio	53170	865	112	NA
36	Oklahoma	48360	848	113	NA
37	Oregon	59070	1227	98	NA
38	Pennsylvania	55490	1001	103	NA
39	Rhode Island	62120	1178	99	NA
40	South Carolina	47490	1005	109	NA
41	South Dakota	46810	805	112	NA
42	Tennessee	49330	980	110	NA

Showing 21 to 43 of 56 entries, 6 total columns

Showing 37 to 56 of 56 entries, 6 total columns

Index	States	Annual_mean_wage	Average_monthly_rent	Value_of_\$100	X
37	Oregon	59070	1227	98	NA
38	Pennsylvania	55490	1001	103	NA
39	Rhode Island	62120	1178	99	NA
40	South Carolina	47490	1005	109	NA
41	South Dakota	46810	805	112	NA
42	Tennessee	49330	980	110	NA
43	Texas	54230	1091	104	NA
44	Utah	53400	1171	104	NA
45	Vermont	55450	NA	97	NA
46	Virginia	62330	1351	99	NA
47	Washington	68740	1488	92	NA
48	West Virginia	46490	787	113	NA
49	Wisconsin	53120	944	108	NA
50	Wyoming	52110	877	107	NA
51	NA	NA	NA	NA	NA
52	NA	NA	NA	NA	NA
53	NA	NA	NA	NA	NA
54	NA	NA	NA	NA	NA
55	NA	NA	NA	NA	NA
56	NA	NA	NA	NA	NA

Showing 37 to 56 of 56 entries, 6 total columns

Step 3:

Now it's time for save the scrapped table or the dataset in Csv format. I have set directory first and named the csv file as T5_dataset.csv

```
write.csv(first_table, file = "T5_dataset.csv")
```

```
first_table = read.csv("T5_dataset.csv")
```

Data Preprocessing

Data in the real world is often dirty; that is it is in need of being cleaned up before it can be used for a desired purpose. This is often called data pre-processing.

There some process for it:

1. Data cleaning:
 - a. Smooth Noisy Data
 - b. Handling Missing Data
 - c. Data Wrangling or Munging
2. Data Integration
3. Data Transformation
4. Data Reduction
5. Data Discretization

First of all I have renamed all column names.

```
#renaming tables  
names(first_table)  
names(first_table)[1] <- "Index"  
names(first_table)[2] <- "States"  
names(first_table)[3] <- "Annual_mean_wage"  
names(first_table)[4] <- "Average_monthly_rent"  
names(first_table)[5] <- "Value_of_$100"  
names(first_table)[6] <- "X"
```

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows variables: content (List of 2), df (49 obs. of 6 variables), first_table (56 obs. of 6 variables), New_dataset (49 obs. of 5 variables), and table (List of 1).
- Data View:** Displays the first 9 rows of the 'first_table' dataset.
- Code Editor:** Shows the R code used to read the CSV file, name the columns, and subset the data to remove the 'X' column.
- Console:** Shows the output of the R code, including the resulting data frame 'df'.
- File Bar:** Includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins.
- System Tray:** Shows system icons like battery level, network, and date/time (Dec 11, 2022, 12:11 PM).

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows variables: content (List of 2), df (49 obs. of 6 variables), first_table (56 obs. of 6 variables), New_dataset (49 obs. of 5 variables), and table (List of 1).
- Data View:** Displays the cleaned dataset 'df' with 49 rows and 6 columns.
- Code Editor:** Shows the R code used to create a bar chart comparing the U.S. cost of living by state, using 'coord_flip()' and 'geom_bar()'.
- Console:** Shows the output of the R code, including the ggplot command.
- File Bar:** Includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins.
- System Tray:** Shows system icons like battery level, network, and date/time (Dec 11, 2022, 12:11 PM).

Handling missing data:

Then started data cleaning process. By using `is.Na()` I have found out which columns and rows have NA values. I saw that "X" named column had full of NA values. So, I have used `subset` function to remove this entire column. And then I have used `NA.omit()` to delete an entire row having NA value.

```
#delete column with NA value  
df = subset(first_table, select = -c(X))  
is.na(df[,1:5])
```

```
#delete NA row
```

```
df <- na.omit(df)
```

As my dataset values are already in integer format so I didn't have to perform data integration, data transformation and reduction here. All I need to categorize the values of dataset. So, I have set Annual mean wage (all occupations) into three categories.

```
library(dplyr)
```

```
df<-df %>% mutate(Type =  
  case_when(Annual_mean_wage < 40000 ~ "Poor wage",  
            Annual_mean_wage < 59999 ~ "Good wage ",  
            Annual_mean_wage >= 60050 ~ "Excellent wage"))
```

```
View(df)
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Scraping.R Untitled1.R scrapping_2.R Scraping_1.R scrapping_4.R scrapping_5.R first_table df

Index States Annual_mean_wage Average_monthly_rent Value_of_5100 Type

1	Alabama	48110	914	114	Good wage
2	Alaska	63480	1288	95	Excellent wage
3	Arizona	55170	1209	104	Good wage
4	Arkansas	46500	779	115	Good wage
5	California	68510	1778	84	Excellent wage
6	Colorado	62900	1404	98	Excellent wage
7	Connecticut	66130	1315	95	Excellent wage
8	Delaware	59820	1346	101	Good wage
9	Florida	51950	1330	99	Good wage
10	Georgia	53940	1084	107	Good wage
11	Hawaii	59760	1880	81	Good wage
12	Idaho	47940	902	108	Good wage
13	Illinois	59650	1050	103	Good wage
14	Indiana	50440	889	111	Good wage
15	Iowa	51140	831	111	Good wage

Showing 1 to 15 of 49 entries. 6 total columns

Console Terminal Background Jobs

```
R 4.2.2 - E:\10th semester\Introduction To Data Science\Final\project> Annual_mean_wage ~ 1~~~~~ ~ 40000 ~ "Poor wage";
+ Annual_mean_wage ~ 59999 ~ "Good wage";
+ Annual_mean_wage >= 61000 ~ "Excellent wage")
> view(df)
> df<-df %>% mutate(Type =
+   case_when(Annual_mean_wage < 40000 ~ "Poor wage",
+             Annual_mean_wage < 59999 ~ "Good wage",
+             Annual_mean_wage >= 61000 ~ "Excellent wage"))
> View(df)
> library(ggplot2)
> View(df)
```

Windows Taskbar: File Explorer, Edge, Google Chrome, R, Xcode, Python, WPS Office, Settings, Cloudflare, 95, 71°F, ENG, 8:11 AM, INTEL, 12/11/2022

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Scraping.R Untitled1.R scrapping_2.R Scraping_1.R scrapping_4.R scrapping_5.R first_table df

Index States Annual_mean_wage Average_monthly_rent Value_of_5100 Type

19	Maine	53230	1025	101	Good wage
20	Maryland	65900	1505	92	Excellent wage
21	Massachusetts	72940	1554	90	Excellent wage
22	Michigan	55160	954	108	Good wage
23	Minnesota	60480	1054	102	Excellent wage
24	Mississippi	42700	868	116	Good wage
25	Missouri	51390	909	111	Good wage
26	Montana	49340	920	107	Good wage
27	Nebraska	52110	900	111	Good wage
28	Nevada	51080	1245	103	Good wage
29	New Hampshire	59270	1214	94	Good wage
30	New Jersey	67120	1541	84	Excellent wage
31	New Mexico	51860	901	109	Good wage
32	New York	70460	1432	84	Excellent wage
33	North Carolina	53100	1009	108	Good wage

Showing 19 to 33 of 49 entries. 6 total columns

Console Terminal Background Jobs

```
R 4.2.2 - E:\10th semester\Introduction To Data Science\Final\project> Annual_mean_wage ~ 1~~~~~ ~ 40000 ~ "Poor wage";
+ Annual_mean_wage ~ 59999 ~ "Good wage";
+ Annual_mean_wage >= 61000 ~ "Excellent wage")
> view(df)
> df<-df %>% mutate(Type =
+   case_when(Annual_mean_wage < 40000 ~ "Poor wage",
+             Annual_mean_wage < 59999 ~ "Good wage",
+             Annual_mean_wage >= 61000 ~ "Excellent wage"))
> View(df)
> library(ggplot2)
> View(df)
```

Windows Taskbar: File Explorer, Edge, Google Chrome, R, Xcode, Python, WPS Office, Settings, Cloudflare, 95, 71°F, ENG, 8:11 AM, INTEL, 12/11/2022

Session 1 (Left):

```
R 4.2.2 - E:\10th semester\Introduction To Data Science\Final\project> 
+   Annual_mean_wage < 35000 ~ "poor wage" 
+   Annual_mean_wage ~ 35000 ~ "good wage" 
+   Annual_mean_wage >= 61000 ~ "Excellent wage") 
> View(df)
> df<-df %>% mutate(type = 
+   case_when(Annual_mean_wage < 40000 ~ "Poor wage",
+             Annual_mean_wage < 59999 ~ "Good wage",
+             Annual_mean_wage >= 60050 ~ "Excellent wage"))
> View(df)
> library(ggplot2)
> View(df)
> library(mosaicData)
> ggplot(df)+geom_bar(aes(x = df$states, y = df$average_monthly_rent),stat="identity")
> ggplot(df)+geom_bar(aes(x = states , y = Average_monthly_rent),stat="identity")
Error: unexpected '$' in "ggplot(df)+geom_bar(aes(x =`"
> ggplot(df)+geom_bar(aes(x =states , y = Average_monthly_rent),stat="identity")
Error in `geom_bar()`:
```

Session 2 (Right):

```
R 4.2.2 - E:\10th semester\Introduction To Data Science\Final\project> 
+   Annual_mean_wage < 35000 ~ "poor wage" 
+   Annual_mean_wage ~ 35000 ~ "good wage" 
+   Annual_mean_wage >= 61000 ~ "Excellent wage") 
> View(df)
> df<-df %>% mutate(type = 
+   case_when(Annual_mean_wage < 40000 ~ "Poor wage",
+             Annual_mean_wage < 59999 ~ "Good wage",
+             Annual_mean_wage >= 60050 ~ "Excellent wage"))
> View(df)
> library(ggplot2)
> View(df)
```

So, This Dataset df is clean now.

Descriptive Statistics:

Mean:

The mean (technically the arithmetic mean), is a measure of central tendency that is calculated by adding together all of the observations and dividing by the number of observations. I had to use mean() to determine mean.

Annual mean wage (for all occupation) = \$55,363.5

Average monthly rent (mean) = \$1104.306

Value Of \$100(for different states) = \$103.5918

#coding part

```
mean(df$Annual_mean_wage)  
mean(df$Average_monthly_rent)  
mean(df$`Value_of_$100`)
```

Median:

The median is another measure of central tendency but one that cannot be directly calculated. Instead, you make a sorted list of all of the observations in the sample and then go halfway up that list.

- Whatever the value of the observation is at the halfway point, that is the median.
- So, the median is the middle value in a data set ordered from low to high.

Median value for Annual mean wage (for all occupations) = 53230

Median value Average monthly rent (mean) = 1009

Median value Value Of \$100(for different states) = 107

#Coding part

```
median(df$Annual_mean_wage)  
median(df$Average_monthly_rent)  
median(df$`Value_of_$100`)
```

Range:

The range is a measure of dispersion—how spread out a bunch of numbers in a sample are—calculated by subtracting the lowest value from the highest value. The range can only be calculated for numerical data. I Have to Use the R min() and max() functions to find the range.

The Range of Annual mean wage (for all occupations) = 30240

The Range of Average monthly rent (mean) = 1101

The Range of Value of \$100(for different states) = 35

#Coding Part

```
max(df$Annual_mean_wage)-min(df$Annual_mean_wage)
```

```
max(df$Average_monthly_rent)-min(df$Average_monthly_rent)
```

```
max(df`Value_of_$100`)-min(df`Value_of_$100`)
```

Variance:

- The variance is a measure of dispersion.
- Like the range, the variance describes how spread out a sample of numbers is.
- Unlike the range, though, which uses just two numbers to calculate dispersion, the variance is obtained from all of the numbers through a simple calculation that compares each number to the mean.
- Steps to calculate variance:
 - A) Find the mean of the attribute you want to calculate the variance.
 - B) For Each Value of the attribute - Find the Difference from the Mean.
 - C) For Each Difference - Find the Square Value.
 - D) The Variance is the Average Number of These Squared Values.

The variance value of Annual mean wage (for all occupations) = 53329

The variance value of Average monthly rent (mean) = 72567.47

The variance of Value of \$100(for different states) = 81.45493

#Coding Part

```
var(df$Annual_mean_wage)
```

```
var(df$Average_monthly_rent)
```

```
var(df$`Value_of_$100`)
```

Standard Deviation:

The standard deviation is simply the square root of the variance. Standard deviation measures how far a 'typical' observation is from the average of the data.

The Standard Deviation value of Annual mean wage (for all occupations) = 7302.711

The Standard Deviation value of Average monthly rent (mean) = 269.3835

The Standard Deviation value of Value of \$100(for different states) = 9.025239

#Coding Part

```
sd(df$Annual_mean_wage)
```

```
sd(df$Average_monthly_rent)
```

```
sd(df$`Value_of_$100`)
```

Quartiles:

- Quartiles are values that separate the data into four equal parts.
- The quartiles (Q0, Q1, Q2, Q3, Q4) are the values that separate each quarter.
 - Q0 is the smallest value in the data.
 - Q1 is the value separating the first quarter from the second quarter of the data.
 - Q2 is the middle value (median), separating the bottom from the top half.
 - Q3 is the value separating the third quarter from the fourth quarter
 - Q4 is the largest value in the data
- Using the R quantile() function I have to find the Quartiles of the values.

Quartile of Annual mean wage (for all occupations):

```
> quantile(df$Annual_mean_wage)
  0%   25%   50%   75%  100%
42700 49680 53230 59820 72940
>
```

Quartile of value of Average monthly rent (mean):

```
> quantile(df$Average_monthly_rent)
 0% 25% 50% 75% 100%
 779 889 1009 1288 1880
   ...
```

Quartile of Value of \$100(for different states):

```
> quantile(df$value_of_$100`)
 0% 25% 50% 75% 100%
 81  99  107 111 116
 ~ top(df$Annual_mean_wage)
```

#Coding Part

```
quantile(df$Annual_mean_wage)
```

```
quantile(df$Average_monthly_rent)
```

```
quantile(df$`Value_of_$100`)
```

Interquartile:

- Interquartile range is the difference between the first and third quartiles (Q1 and Q3).
- The 'middle half' of the data is between the first and third quartile.
- The first quartile is the value in the data that separates the bottom 25% of values from the top 75%.
- The third quartile is the value in the data that separates the bottom 75% of the values from the top 25%.

Interquartile value of Annual mean wage (for all occupations) = 10140

Interquartile value of Average monthly rent (mean) = 399

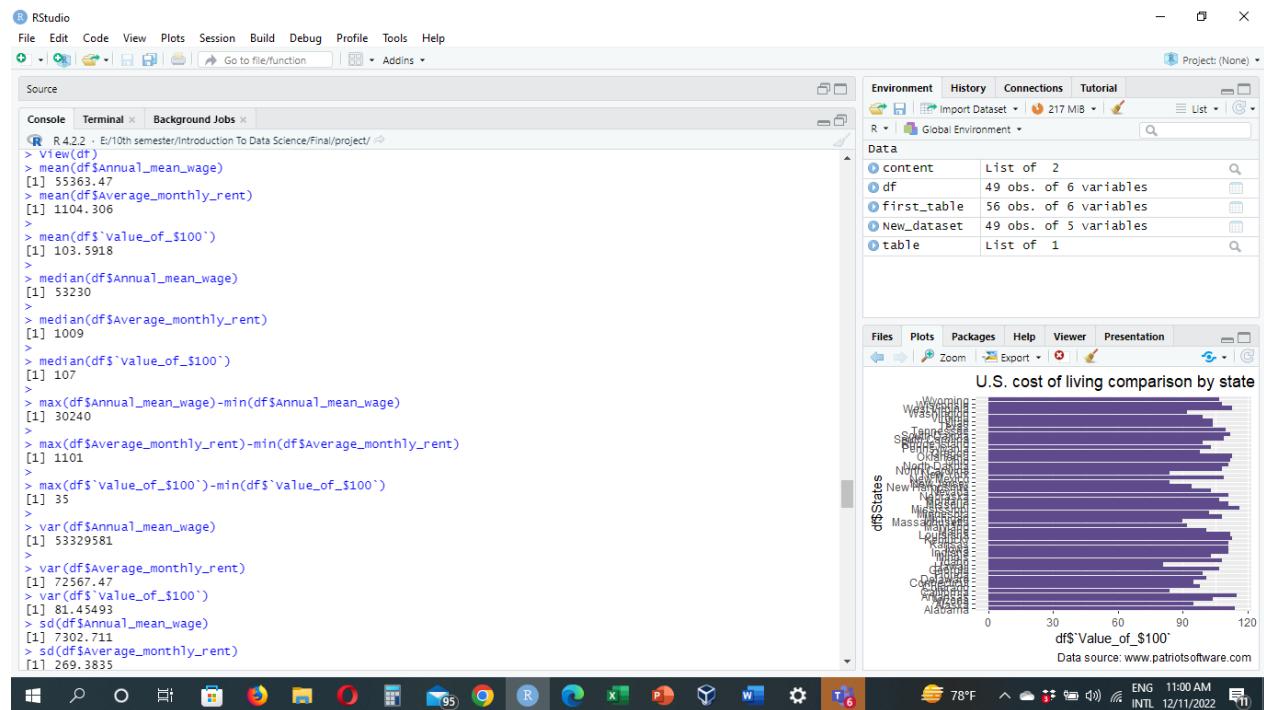
Interquartile value of Value of \$100(for different states) = 12

#Coding Part

```
IQR(df$Annual_mean_wage)
```

```
IQR(df$Average_monthly_rent)
```

```
IQR(df`Value_of_$100`)
```



The screenshot shows the RStudio interface. The console window displays R code and its output, including statistical calculations like mean and standard deviation for 'df\$Value_of_\$100'. The environment browser shows objects like 'content', 'df', 'first_table', 'New_dataset', and 'table'. A bar chart titled 'U.S. cost of living comparison by state' is displayed, comparing the value of \$100 across different US states. The x-axis represents the value of \$100, ranging from 0 to 120. The y-axis lists states from West Virginia at the top to Alabama at the bottom. The chart shows a general downward trend from west to east.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

Source
Console Terminal Background Jobs
[R 4.2.2 · 6/10th semester/Introduction To Data Science/Final/project/ · 53329581]
> var(df$Average_monthly_rent)
[1] 72567.47
> var(df$value_of_$100')
[1] 81.45493
> sd(df$Annual_mean_wage)
[1] 7302.711
> sd(df$Average_monthly_rent)
[1] 269.3835
> sd(df$value_of_$100')
[1] 9.025239
> quantile(df$Annual_mean_wage)
 0% 25% 50% 75% 100%
42700 49680 53230 59820 72940
> quantile(df$Average_monthly_rent)
 0% 25% 50% 75% 100%
779 889 1009 1288 1880
> quantile(df$value_of_$100')
 0% 25% 50% 75% 100%
8 99 107 111 116
> IQR(df$Annual_mean_wage)
[1] 10140
> IQR(df$Average_monthly_rent)
[1] 399
> IQR(df$value_of_$100')
[1] 12
> install.packages(c("mosaicdata", "ggplot2"))
Error in install.packages : Updating loaded packages
Restarting R session...
> install.packages(c("mosaicdata", "ggplot2"))
Installing packages into 'C:/Users/USER/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'vctrs'

Environment History Connections Tutorial
Project: (None)

Data
content List of 2
df 49 obs. of 6 variables
first_table 56 obs. of 6 variables
New_dataset 49 obs. of 5 variables
table List of 1

Files Plots Packages Help Viewer Presentation
U.S. cost of living comparison by state
df$States
West Virginia
Tennessee
South Carolina
Pennsylvania
North Carolina
New Jersey
New York
New Hampshire
Mississippi
Massachusetts
Louisiana
Georgia
Connecticut
Rhode Island
Alabama
Data source: www.patriotsoftwre.com
df$Value_of_$100
0 30 60 90 120

```

Data Visualization

The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics.

Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.

Bar chart: Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.

When to Use a Bar Chart

- 1) When I need to compare a large set of categorical values.
- 2) When required to compare multiple categories or sub-categories simultaneously
- 3) When you need to visualize two data sets on a single chart
- 4) When you need to gather insights on deviations in data

Here I'm going to use bar chart for data visualization.

First, I have installed required packages and libraries.

```
install.packages(c("mosaicData","ggplot2"))
```

```
library(ggplot2)
```

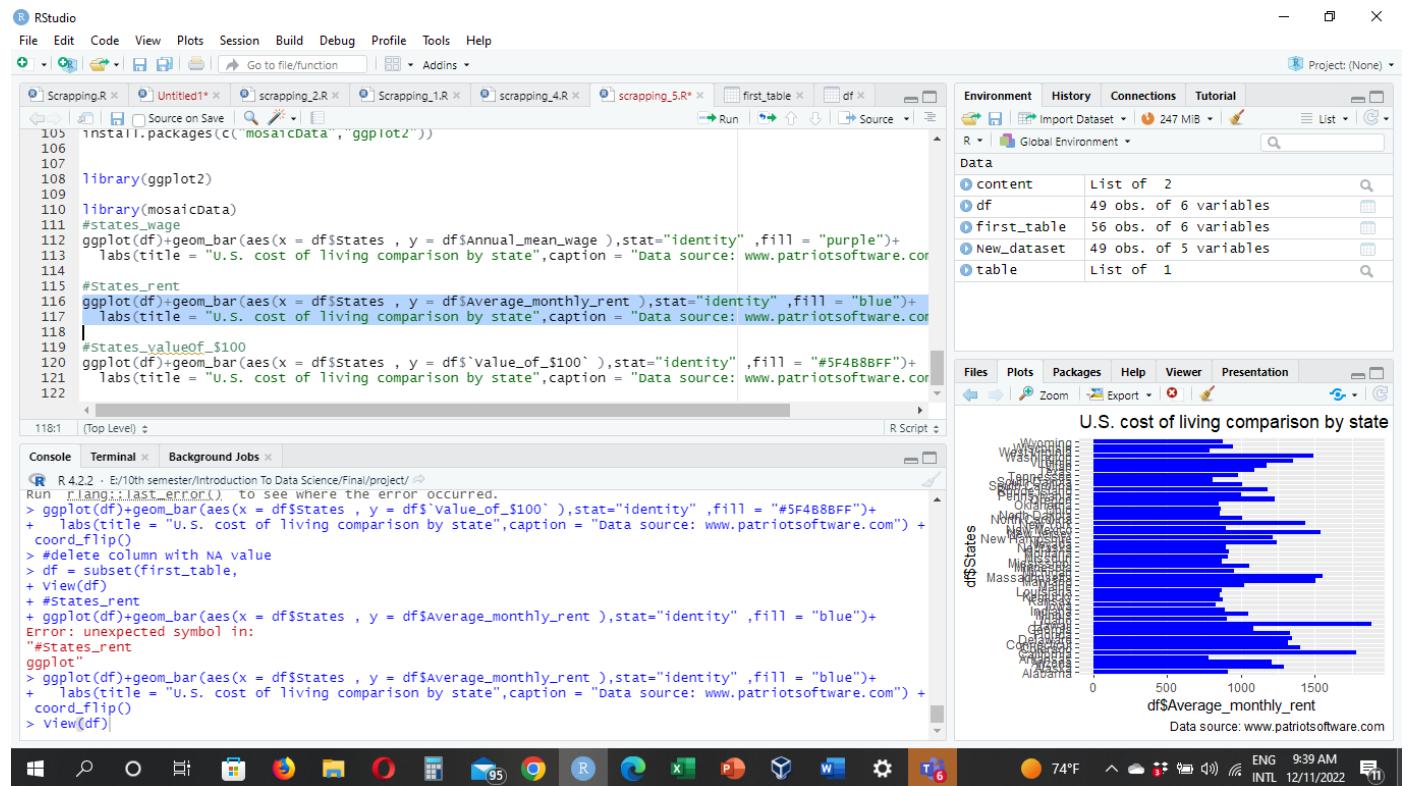
```
library(mosaicData)
```

***Graph plotting for Annual mean wage(all occupations) on different states:**

Then I have called ggplot function and set df dataset there. Then called geombar function for using bar chart and set x axis as States variable and Y axis as Annual_mean_wage variable. Then set the color as purple. Then used labs() function to set a suitable title for the graph and caption to mention source. Finally I have used coord_flip() function to flip the bar chart and to plot the data as a horizontal bar chart.

#Coding Part

```
ggplot(df)+geom_bar(aes(x = df$States , y = df$Annual_mean_wage ),stat="identity" ,fill = "purple")+\n  labs(title = "U.S. cost of living comparison by state",caption = "Data source:\nwww.patriotsoftware.com") + coord_flip()
```



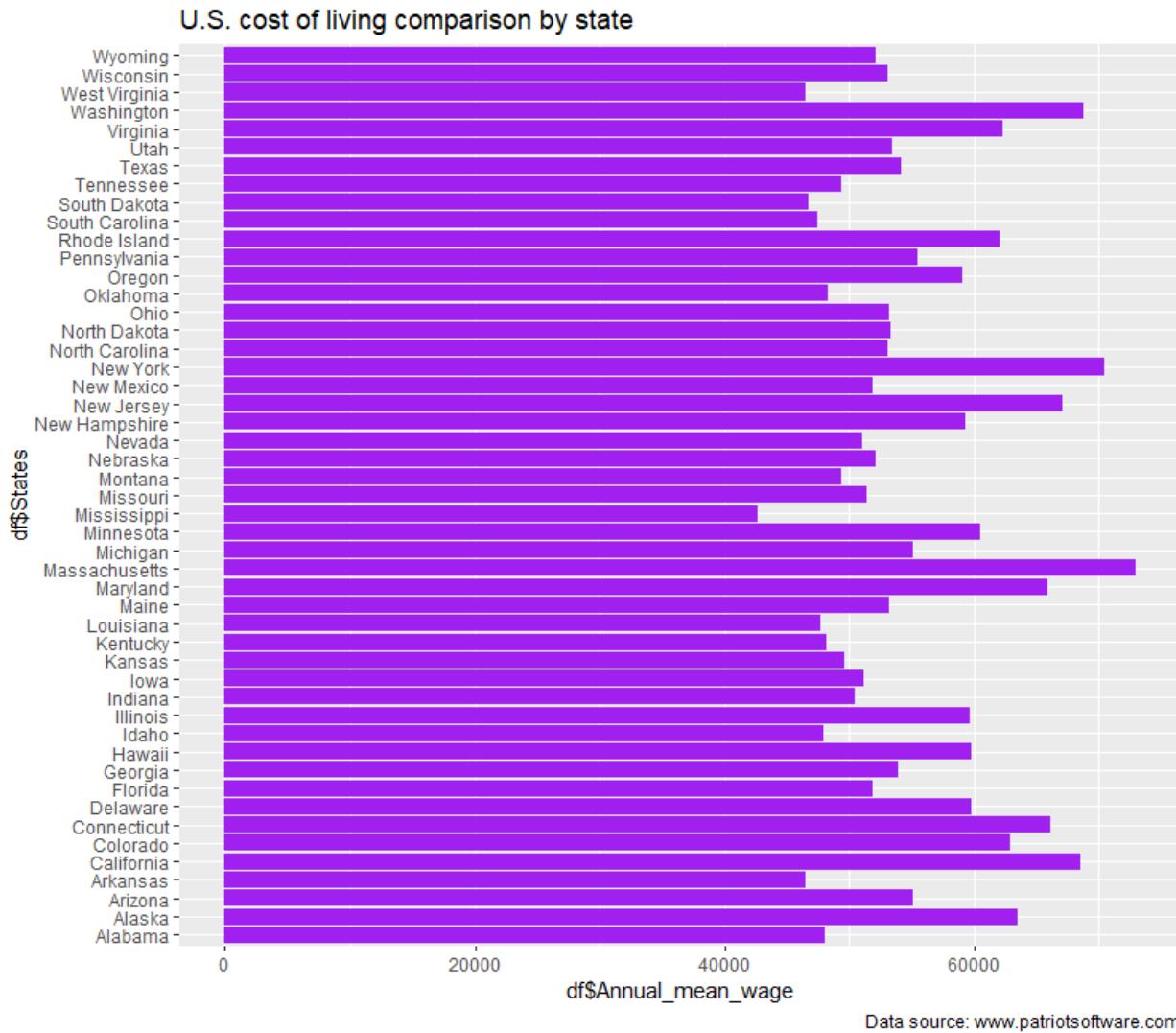


Figure 1:Bar_Chart diagram over Annual_mean_wage for different states

From this bar chart we can see the highest Annual mean wage is on Massachusetts and the lowest Annual mean wage for Mississippi.

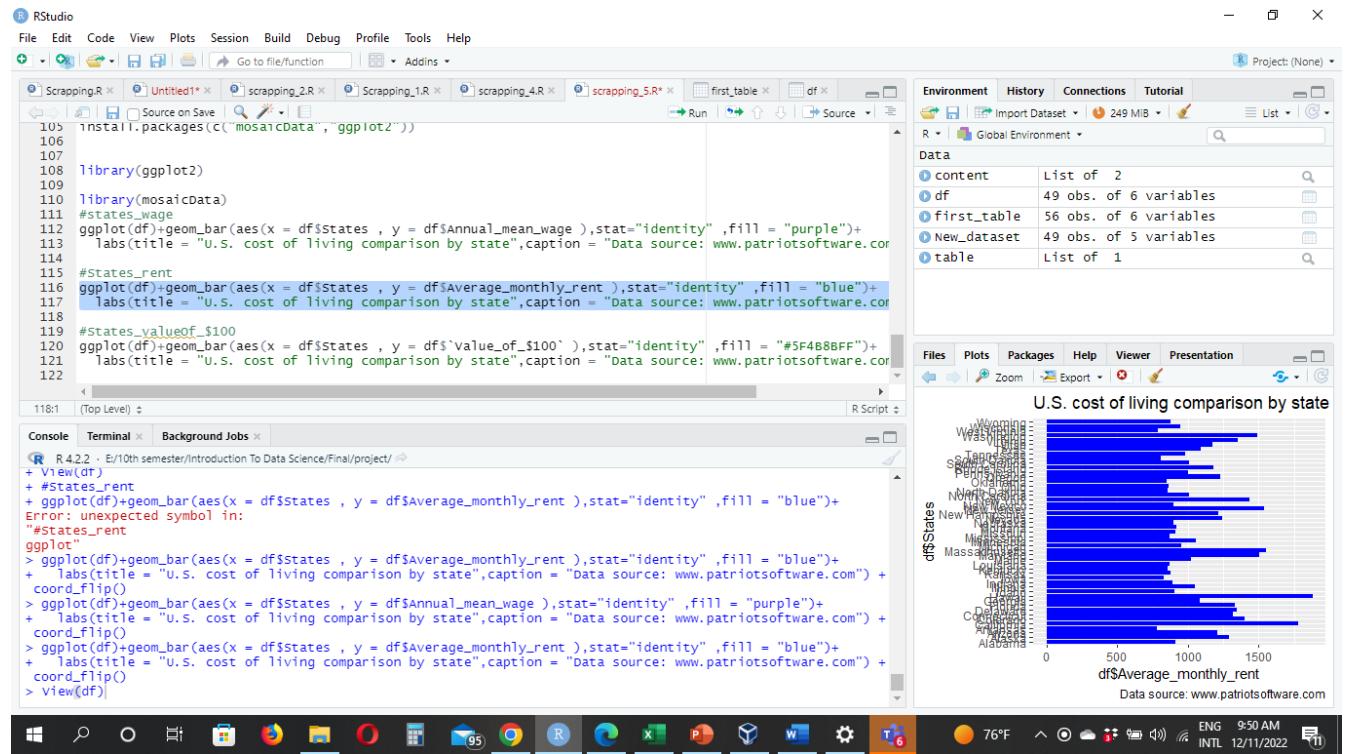
***Graph plotting for Average monthly rent on different states:**

Again we have plotted the graph for Average monthly rent on different states. So, I have to call ggplot function and set df dataset there. Then called goembar function for using bar chart and set x axis as States variable and Y axis as Average monthly rent variable. Then set the color as blue.

Then used labs() function to set a suitable title for the graph and caption to mention source.
Finally I have used coord_flip() function to flip the bar chart and to plot the data as a horizontal bar chart.

#Coding Part

```
ggplot(df)+geom_bar(aes(x = df$States , y = df$Average_monthly_rent ),stat="identity" ,fill = "blue")+
  labs(title = "U.S. cost of living comparison by state",caption = "Data source: www.patriotsoftware.com") + coord_flip()
```



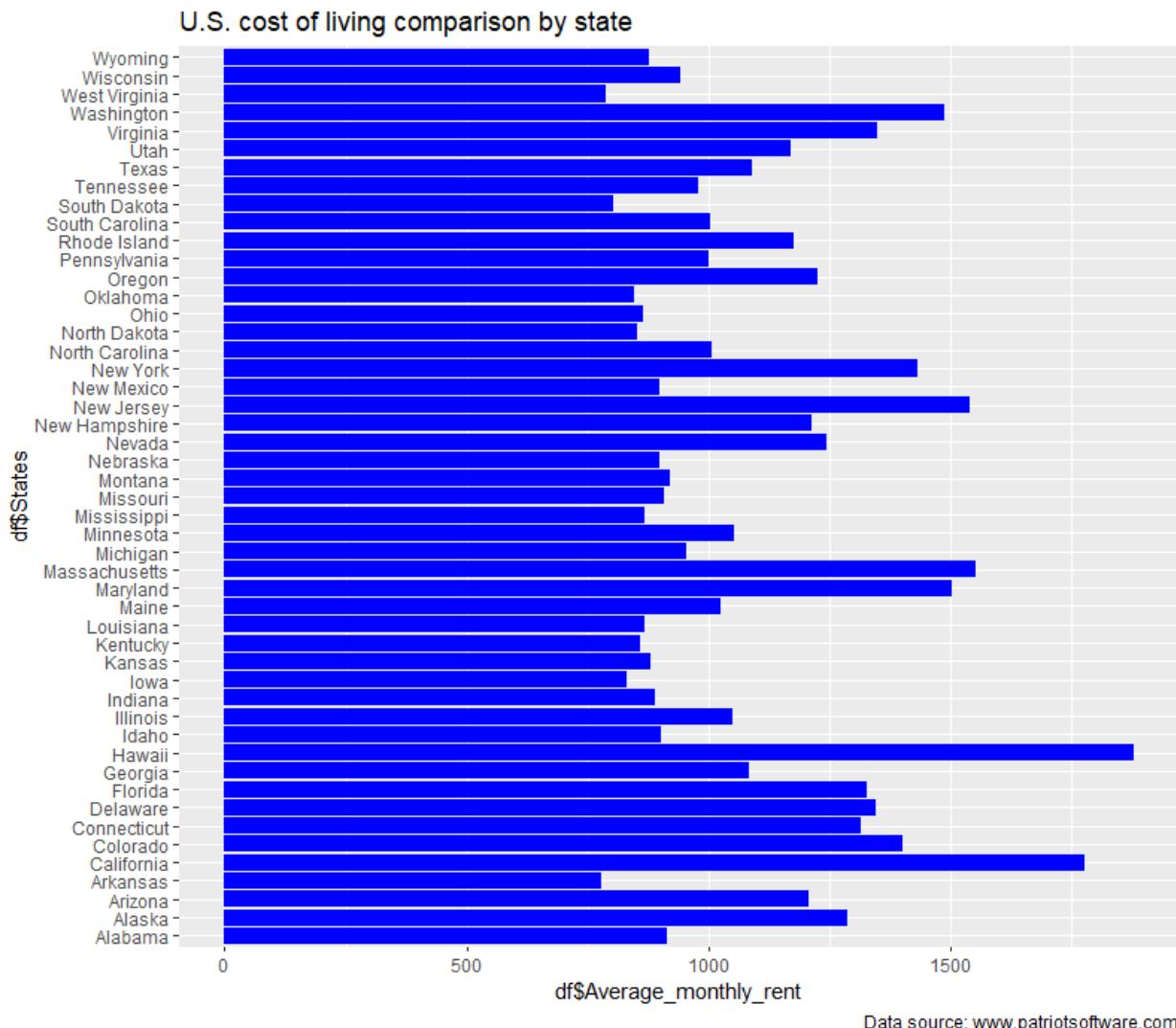


Figure 2: Monthly average rent on different states

From this bar chart we can see the highest Average monthly rent is on Hawaii and the lowest Annual mean wage for Arkansas alongside West Virginia.

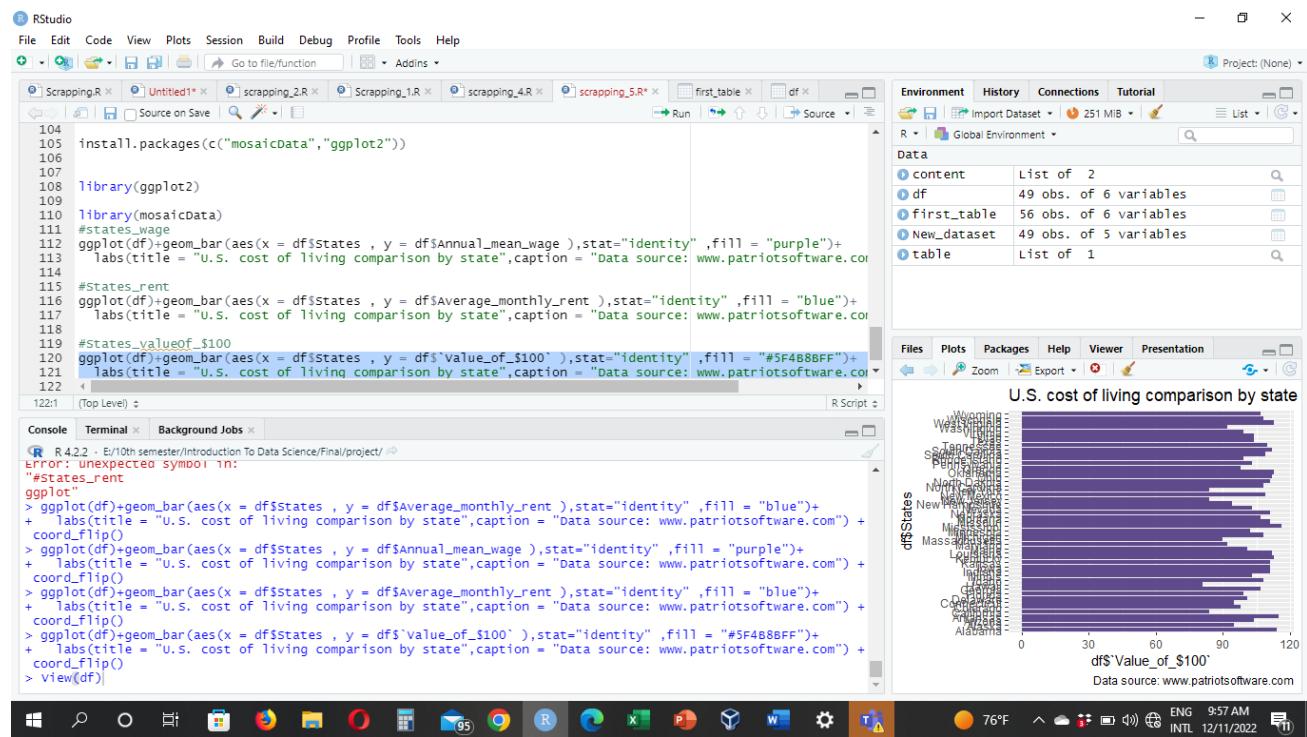
***Graph plotting for Value of \$100 on different states:**

Again, we have plotted the graph for Value of \$100 rent on different states. So, I have to call ggplot function and set df dataset there. Then called goembar() function for using bar chart and

set x axis as States variable and Y axis as Value of \$100 variable. Then set the color as “#5F4B8BFF”. Then used labs() function to set a suitable title for the graph and caption to mention source. Finally I have used coord_flip() function to flip the bar chart and to plot the data as a horizontal bar chart.

#Coding Part

```
ggplot(df)+geom_bar(aes(x = df$States , y = df`Value_of_$100` ),stat="identity",fill = "#5F4B8BFF")+
  labs(title = "U.S. cost of living comparison by state",caption = "Data source: www.patriotsoftware.com") + coord_flip()
```



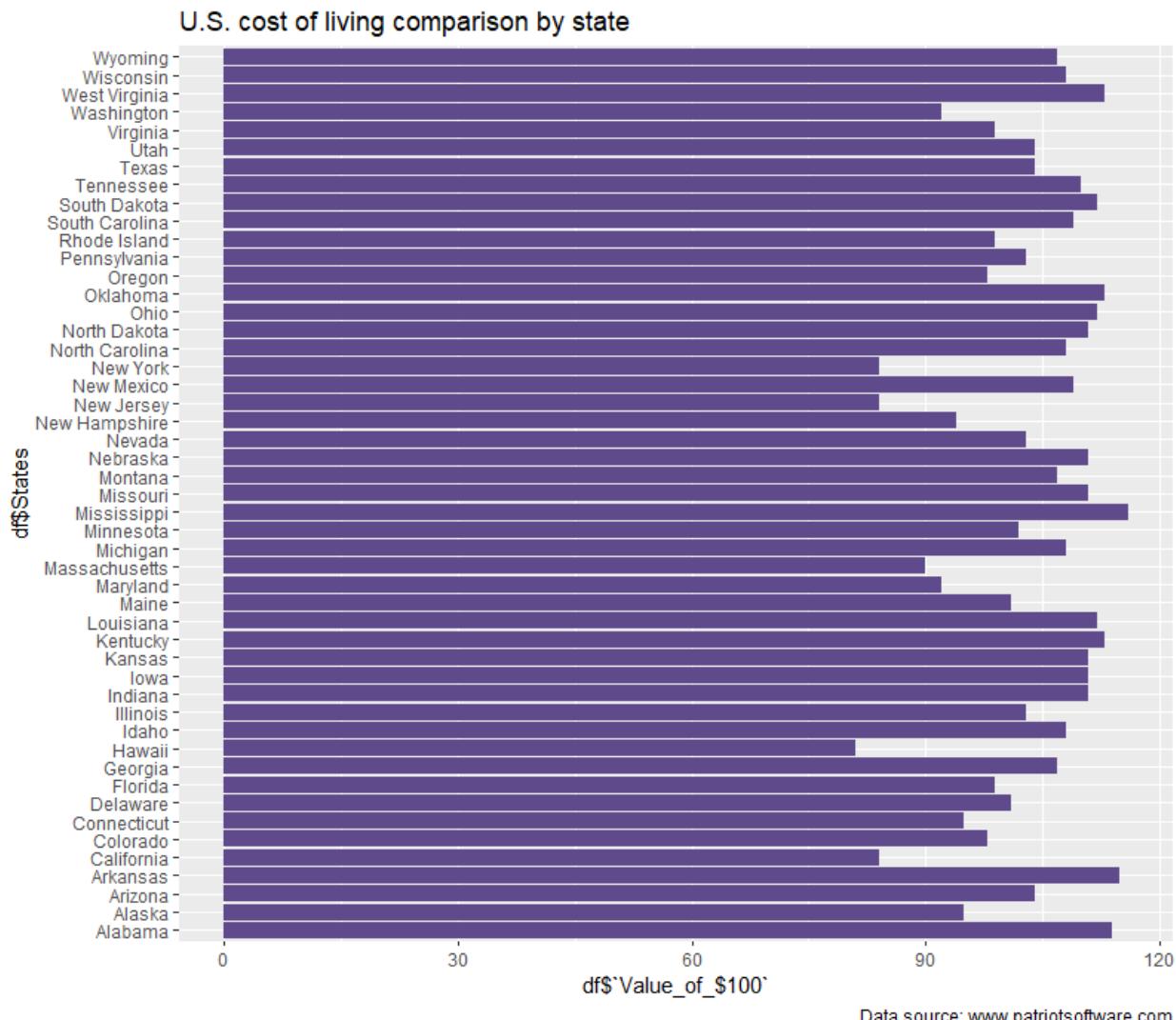


Figure 3: Value Of \$100 on different states

From this bar chart we can see the state where \$100 is most valuable is Mississippi and also to close states are Arkansas, West Virginia. Now, the less valuable place is Hawaii.

Discussion & Conclusion:

In this project, I have worked on scrapping, Data-Preprocessing, Descriptive Statistics and Data Visualization. For this I have Scrapped a table from this URL:

<https://www.patriotsoftware.com/blog/accounting/average-cost-living-by-state/> and this table is based on Cost of living in each state of U.S.A.

Cost of living in each state data can vary depending on what factors are analyzed (e.g., housing) and who conducts the study. So, for cost-of-living comparison by state, this dataset having with the following critical information:

- Annual mean wage for all occupations
- Average monthly rent
- Value of \$100

The mean wages represent the average wages employees working in the state earn per year. This data is from May 2021. Also, I can to divide the annual mean wage by 12 to find the monthly wage amount.

Average monthly rent shows the average overall value of rent in the state (for all unit sizes). This data was compiled by GoBankingRates in 2021.

The value of \$100 captures how much 100 George Washingtons are worth in the state. If the value of a dollar drops below \$100, then it does not go as far in that state. This data was compiled in February 2022.

By looking at the chart, we have to keep in mind that states with a higher mean wage, higher monthly rent, and lower value of a dollar tend to have a higher cost of living.

So, from bar charts we can easily see that there is highest Annual mean wage for Massachusetts which is \$72940, highest Average monthly rent is for Hawaii is \$1880 and highest value of \$100 is for Mississippi with \$116.

Now, coming to lowest. From bar charts we can see there is lowest Annual mean wage for Mississippi which is \$42700, lowest Average monthly rent is for Arkansas which is \$779 and lowest value of \$100 is for Hawaii with \$81.

So, by analyzing those data we can say that Mississippi is the most affordable state by considering its highest value of \$100 and lowest Annual mean wage.

And, Hawaii is most expensive state by considering its highest Average monthly rent and lowest value of \$100.

- **Cost of living per state: Lowest to highest list**

By considering on values from graph and dataset I tried organize those states from most affordable to most expensive order. Here is the list:

1. Mississippi
2. Arkansas
3. Missouri
4. Michigan
5. Tennessee
6. Ohio
7. Kentucky
8. Oklahoma
9. Georgia
10. Alabama
11. Indiana
12. North Carolina
13. West Virginia
14. Texas
15. Louisiana
16. Kansas
17. Iowa
18. South Carolina
19. Illinois
20. Wisconsin
21. Nebraska
22. Idaho
23. South Dakota
24. New Mexico
25. Florida
26. Pennsylvania

27. Arizona
28. Minnesota
29. Montana
30. Virginia
31. Utah
32. Wyoming
33. North Dakota
34. Delaware
35. Nevada
36. Colorado
37. New York
38. Washington
39. Maine
40. Oregon
41. Vermont
42. New Jersey
43. New Hampshire
44. Maryland
45. Rhode Island
46. Connecticut
47. Massachusetts
48. Alaska
49. California
50. Hawaii

Through this list anyone could easily get valuable idea about affordable or expensive states in U.S.A.