Q6

Ana Paula Felix de Queiroz

## We-R-Finished

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':  
##   
## recode

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

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

### Question 6

When looking over at the three given questions, there was a common sub-question:

**Will the answer change if I take standard of living into account?**

While it leans more hypothetical, it did bring thoughts of perhaps we need to quantify it to answer it. In this question, we’re going to dive a bit deeper on that question using the Cost of Living and the salary dataset.

### What is standard/cost of living?

From <https://worldpopulationreview.com/state-rankings/cost-of-living-index-by-state>:

“Cost of living refers to the amount needed to cover basic expenses, such as food, shelter, transportation, and healthcare.”

Of course, each state will vary in their cost of living. From what we know, California is one of the most expensive states to live in. Let’s take the shelter expense for example. We know that housing prices have risen to insane numbers in this state. For example, in San Mateo, a single family home can cost upwards to 2 million USD. If one goes to Hillsborough, known to be an expensive area, is double that. Rent is also extremely high, a one bedroom apartment can range between 2 thousand to 4 thousand USD.

There could be many reasons for this such as the population density, tax rates, etc. When moving out of California, or at least the highly populated areas of the state, the costs will go down. Let’s say perhaps somewhere in the Midwest or the South. Housing prices are more likely to be less than 1 million USD, therefore the cost of living index may lower.

### Why is it important?

In our particular project, having data on the cost of living for each region helps employers determine the wage of the job. Of course, certain *genre* of jobs will have higher pay than others, and specific *types* of jobs within that genre will have better wage than the others. When we take the salary dataset into account, we can use this data to find out if one would be able to live in a state with the salary they will receive. Of course, I will only use the top 10 states for data-related jobs I found in Q2.

## Warning: Coercing text to numeric in Y146963 / R146963C25: '45870'

## Warning: Coercing text to numeric in Y164631 / R164631C25: '76700'

### About the dataset

The Cost of Living dataset is from the same site I have mentioned in the previous question, which was found by Russell Chan.

## # A tibble: 6 × 16  
## state densi…¹ pop2023 pop2022 pop2020 pop2019 pop2010 growth…² growth growth…³  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Miss… 63.1 2.96e6 2.96e6 2.96e6 2.96e6 2967297 -0.0002 -602 -0.00264  
## 2 Kans… 36.2 2.96e6 2.95e6 2.94e6 2.93e6 2853118 0.00287 8476 0.0386   
## 3 Alab… 101. 5.10e6 5.07e6 5.02e6 5.00e6 4779736 0.00482 24454 0.0665   
## 4 Okla… 58.6 4.02e6 4.00e6 3.96e6 3.94e6 3751351 0.0052 20800 0.0721   
## 5 Geor… 192. 1.10e7 1.09e7 1.07e7 1.06e7 9687653 0.00938 102426 0.137   
## 6 Tenn… 172. 7.08e6 7.02e6 6.91e6 6.85e6 6346105 0.00804 56474 0.116   
## # … with 6 more variables: costIndex <dbl>, groceryCost <dbl>,  
## # housingCost <dbl>, utilitiesCost <dbl>, transportationCost <dbl>,  
## # miscCost <dbl>, and abbreviated variable names ¹​densityMi, ²​growthRate,  
## # ³​growthSince2010

Within this dataset, there are 50 entries, one for each state in the USA. It contains data on the population, the growth, and various cost indexes for each state. The main columns we are interested in are the cost and the state, because we are planning to merge the CoL and salary dataset.

### Prepping the dataset

We first create a subset in coldf with only the state and costIndex columns before merging it with the salary dataset.

Next, we will make a new column to create an adjusted wage, where we divide the paid wage per year by the cost index to even the field. We will also have a specific subset that only includes: employer name, job title sub, work state, paid wage per year, adjusted year, and Cost index.

### The Question

What I aim to answer with the merged dataset is this:

With the adjusted wage in mind, could one be able to move and live in the work state they desire? Which job subcategory would allow you to do so?

Previously in Q2, I have used the average and max to answer it. Cost of living dataset does use averages to then make a baseline, and I was planning to use averages to answer this question. But, by Ana’s suggestion, median would be far more stable than using the mean. The mean tends to be skewed if there’s more or less of something, and knowing that software engineer has major outliers, this made much more sense.

Here we will look at the top 5 states median wage, and see how much the wages change when taking into consideration the cost of living in these countries.

## State Job\_Title\_Sub Med\_Paid\_Wage  
## 11 California data scientist 119800.0  
## 21 California software engineer 105972.5  
## 16 California management consultant 97446.0  
## 6 California data analyst 75046.0  
## 1 California business analyst 70000.0  
## 12 Massachusetts data scientist 102500.0  
## 17 Massachusetts management consultant 97000.0  
## 22 Massachusetts software engineer 88691.0  
## 2 Massachusetts business analyst 67000.0  
## 7 Massachusetts data analyst 65000.0  
## 18 New York management consultant 119995.2  
## 13 New York data scientist 105000.0  
## 23 New York software engineer 93621.0  
## 3 New York business analyst 68997.0  
## 8 New York data analyst 65000.0  
## 14 Oregon data scientist 104826.5  
## 24 Oregon software engineer 82846.0  
## 4 Oregon business analyst 68640.0  
## 9 Oregon data analyst 64043.0  
## 19 Oregon management consultant 54080.0  
## 15 Washington data scientist 115000.0  
## 20 Washington management consultant 108000.0  
## 25 Washington software engineer 105000.0  
## 5 Washington business analyst 75000.0  
## 10 Washington data analyst 69000.0

##   
## Attaching package: 'scales'

## The following object is masked from 'package:readr':  
##   
## col\_factor

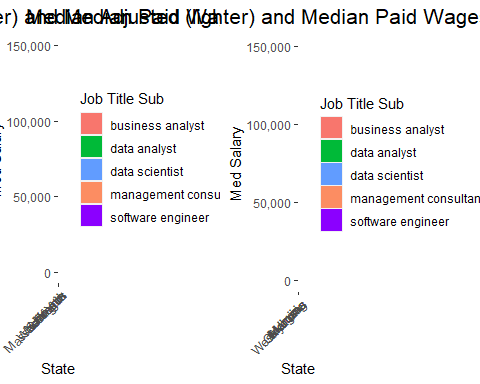
## State Job\_Title\_Sub Med\_Adjusted\_Wage  
## 8 Arkansas data scientist 85258.53  
## 11 Arkansas software engineer 71507.15  
## 1 Arkansas business analyst 66006.60  
## 6 Arkansas data analyst 60506.05  
## 9 Arkansas management consultant 56655.67  
## 2 Montana business analyst 84409.14  
## 12 Montana software engineer 59582.92  
## 10 Oklahoma management consultant 161373.72  
## 3 Oklahoma business analyst 73947.67  
## 13 Oklahoma software engineer 68812.29  
## 7 Oklahoma data analyst 64732.65  
## 14 West Virginia software engineer 63535.91  
## 4 West Virginia business analyst 60356.91  
## 5 Wyoming business analyst 85896.08  
## 15 Wyoming software engineer 58324.50

## State Job\_Title\_Sub Med\_Paid\_Wage  
## 8 Arkansas data scientist 77500.0  
## 11 Arkansas software engineer 65000.0  
## 1 Arkansas business analyst 60000.0  
## 6 Arkansas data analyst 55000.0  
## 9 Arkansas management consultant 51500.0  
## 2 Montana business analyst 85000.0  
## 12 Montana software engineer 60000.0  
## 10 Oklahoma management consultant 141847.5  
## 3 Oklahoma business analyst 65000.0  
## 13 Oklahoma software engineer 60486.0  
## 7 Oklahoma data analyst 56900.0  
## 14 West Virginia software engineer 57500.0  
## 4 West Virginia business analyst 54623.0  
## 5 Wyoming business analyst 81000.0  
## 15 Wyoming software engineer 55000.0

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

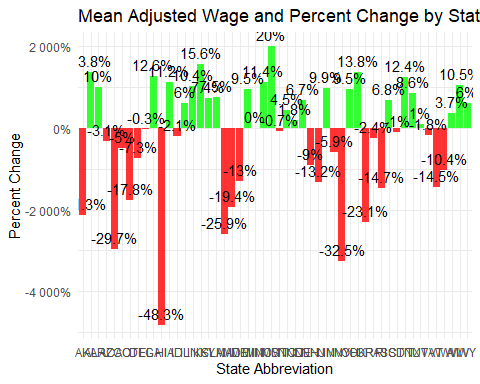
## Warning: Removed 1 rows containing missing values (`geom\_bar()`).



looking at cost index by state

A look into the percentage change in Wages after cost of living is taken into account. We see that Hawaii has the highest percentage change with an adjusted salary of almost 48%, followed by

## State Mean\_Paid\_Wage\_Per\_Year Mean\_Adjusted\_Wage  
## 24 Mississippi 67566.38 81112.10  
## 16 Kansas 71375.30 82514.80  
## 1 Alabama 75558.06 85959.11  
## 36 Oklahoma 65121.79 74086.23  
## 10 Georgia 73190.31 82421.52  
## 42 Tennessee 71280.25 80090.16  
## 25 Missouri 72204.62 80406.04  
## 15 Iowa 70530.69 78454.61  
## 48 West Virginia 67361.66 74432.78  
## 14 Indiana 67881.47 74924.36  
## 4 Arkansas 65246.26 71778.06  
## 31 New Mexico 72902.82 80112.99  
## 35 Ohio 72649.49 79572.28  
## 22 Michigan 71967.91 78825.75  
## 43 Texas 75678.58 82170.01  
## 18 Louisiana 76327.89 82073.00  
## 17 Kentucky 69526.99 74679.91  
## 40 South Carolina 69351.69 74093.68  
## 27 Nebraska 72127.52 76977.08  
## 13 Illinois 76156.11 80759.39  
## 50 Wyoming 66474.17 70492.22  
## 33 North Carolina 76715.01 80161.97  
## 49 Wisconsin 72361.90 75064.21  
## 34 North Dakota 67724.88 68966.28  
## 44 Utah 80210.03 81020.23  
## 23 Minnesota 77631.98 77631.98  
## 9 Florida 69775.20 69566.50  
## 26 Montana 68423.74 67948.10  
## 41 South Dakota 65169.12 64523.89  
## 46 Virginia 76994.09 75632.70  
## 12 Idaho 70834.12 69377.20  
## 38 Pennsylvania 77917.83 76017.40  
## 3 Arizona 75396.53 73058.65  
## 6 Colorado 78685.09 74724.68  
## 28 Nevada 80084.35 75338.05  
## 8 Delaware 72022.69 66749.48  
## 29 New Hampshire 77822.03 70811.67  
## 47 Washington 103432.54 92681.49  
## 19 Maine 70789.61 61556.19  
## 30 New Jersey 76989.80 66831.43  
## 45 Vermont 81494.74 69653.62  
## 39 Rhode Island 74259.37 63361.24  
## 7 Connecticut 77894.12 64057.66  
## 20 Maryland 81110.47 65411.67  
## 2 Alaska 68908.09 54215.65  
## 37 Oregon 83427.39 64125.59  
## 21 Massachusetts 87257.53 64635.21  
## 5 California 105531.88 74213.70  
## 32 New York 91040.63 61430.92  
## 11 Hawaii 72260.59 37382.61  
## WORK\_STATE\_ABBREVIATION Percent\_Change Positive\_Change  
## 24 MS 20.0480192 Yes  
## 16 KS 15.6069364 Yes  
## 1 AL 13.7656428 Yes  
## 36 OK 13.7656428 Yes  
## 10 GA 12.6126126 Yes  
## 42 TN 12.3595506 Yes  
## 25 MO 11.3585746 Yes  
## 15 IA 11.2347052 Yes  
## 48 WV 10.4972376 Yes  
## 14 IN 10.3752759 Yes  
## 4 AR 10.0110011 Yes  
## 31 NM 9.8901099 Yes  
## 35 OH 9.5290252 Yes  
## 22 MI 9.5290252 Yes  
## 43 TX 8.5776330 Yes  
## 18 LA 7.5268817 Yes  
## 17 KY 7.4113856 Yes  
## 40 SC 6.8376068 Yes  
## 27 NE 6.7235859 Yes  
## 13 IL 6.0445387 Yes  
## 50 WY 6.0445387 Yes  
## 33 NC 4.4932079 Yes  
## 49 WI 3.7344398 Yes  
## 34 ND 1.8329939 Yes  
## 44 UT 1.0101010 Yes  
## 23 MN 0.0000000 No  
## 9 FL -0.2991027 No  
## 26 MT -0.6951341 No  
## 41 SD -0.9900990 No  
## 46 VA -1.7681729 No  
## 12 ID -2.0568071 No  
## 38 PA -2.4390244 No  
## 3 AZ -3.1007752 No  
## 6 CO -5.0332384 No  
## 28 NV -5.9266228 No  
## 8 DE -7.3215941 No  
## 29 NH -9.0081893 No  
## 47 WA -10.3942652 No  
## 19 ME -13.0434783 No  
## 30 NJ -13.1944444 No  
## 45 VT -14.5299145 No  
## 39 RI -14.6757679 No  
## 7 CT -17.7631579 No  
## 20 MD -19.3548387 No  
## 2 AK -21.3217939 No  
## 37 OR -23.1360492 No  
## 21 MA -25.9259259 No  
## 5 CA -29.6765120 No  
## 32 NY -32.5236167 No  
## 11 HI -48.2669426 No



knitr::opts\_chunk$set(echo = FALSE)  
knitr::opts\_knit$set(root.dir = "C:/Users/pauli/OneDrive/Documentos/Classes 2023/311\_tech")  
  
library(car)  
  
library(readr)  
library(readxl)  
library(dplyr)  
# reading salary dataset  
salary <- read\_excel("salary\_data\_states.xlsx")  
#names(salary)  
#dim(salary)  
colnames(salary)[19] = "Work\_State"  
colnames(salary)[26] = "Job\_Title\_Sub"  
# filtering  
sal <- salary %>%  
 filter(!grepl("professor", `Job\_Title\_Sub`, ignore.case = TRUE) &  
 !grepl("attorney", `Job\_Title\_Sub`, ignore.case = TRUE) &  
 !grepl("assistant professor", `Job\_Title\_Sub`, ignore.case = TRUE) &  
 !grepl("teacher", `Job\_Title\_Sub`, ignore.case = TRUE))  
datsal <- sal %>%  
 filter(!grepl("Guam", `Work\_State`, ignore.case = TRUE) &  
 !grepl("Guamam", `Work\_State`, ignore.case = TRUE) &  
 !grepl("Palau", `Work\_State`, ignore.case = TRUE) &  
 !grepl("Northern Mariana Islands", `Work\_State`, ignore.case = TRUE) &  
 !grepl("Puerto Rico", `Work\_State`, ignore.case = TRUE) &  
 !grepl("Virgin Islands", `Work\_State`, ignore.case = TRUE))  
# reading CoL dataset  
COL <- read\_csv("CostOfLiving2023.csv", show\_col\_types = FALSE)  
coldf <- COL[, !(names(COL) %in% c("fips"))]  
head(coldf)  
# subset  
costate <- subset(coldf, select = c(state, costIndex))  
# merging  
salcost <- merge(datsal, costate, by.x = "Work\_State", by.y = "state")  
#head(salcost)  
  
salcost$ADJUSTED\_WAGE <- (salcost$PAID\_WAGE\_PER\_YEAR / salcost$costIndex)\*100  
  
# new subset  
salcost <- salcost[, c("EMPLOYER\_NAME", "Job\_Title\_Sub", "Work\_State", "PAID\_WAGE\_PER\_YEAR", "ADJUSTED\_WAGE", "costIndex", "WORK\_STATE\_ABBREVIATION")]  
  
  
#unique(salcost$Job\_Title\_Sub)  
  
  
# ADJUSTED\_WAGE for each state  
all\_states\_med\_adjusted <- aggregate(salcost$ADJUSTED\_WAGE, by = list(salcost$Work\_State), FUN = median)  
names(all\_states\_med\_adjusted) <- c("State", "Median\_Adjusted\_Wage")  
  
# Sort the state\_means data frame by Mean\_Adjusted\_Wage in descending order  
all\_states\_med\_adjusted <- all\_states\_med\_adjusted[order(-all\_states\_med\_adjusted$Median\_Adjusted\_Wage),]  
  
  
# PAID\_WAGE for each state  
all\_states\_med\_paid <- aggregate(salcost$PAID\_WAGE\_PER\_YEAR, by = list(salcost$Work\_State), FUN = median)  
  
names(all\_states\_med\_paid) <- c("State", "Median\_paid\_Wage")  
  
# Sort the state\_means data frame by Mean\_Adjusted\_Wage in descending order  
all\_states\_med\_paid <- all\_states\_med\_paid[order(-all\_states\_med\_paid$Median\_paid\_Wage),]  
  
###################################################  
  
#Looking at the median change of top 5 states by subtitle after adjusted wage  
# First, extract the top 5 states  
top\_states <- head(all\_states\_med\_paid, 5)$State  
  
# Filter the data by the top 5 states  
top\_states\_data <- salcost[salcost$Work\_State %in% top\_states, ]  
  
# Calculate the mean ADJUSTED\_WAGE by job subtitle and state  
state\_job\_medians <- aggregate(top\_states\_data$ADJUSTED\_WAGE,   
 by = list(top\_states\_data$Work\_State, top\_states\_data$Job\_Title\_Sub),  
 FUN = median)  
  
# Rename the columns of the state\_job\_means data frame  
names(state\_job\_medians) <- c("State", "Job\_Title\_Sub", "Median\_Adjusted\_Wage")  
  
# Order the data frame by state and mean adjusted wage in descending order  
state\_job\_medians <- state\_job\_medians[order(state\_job\_medians$State, -state\_job\_medians$Median\_Adjusted\_Wage),]  
  
# Print the state\_job\_means data frame  
#print(state\_job\_medians)  
  
  
  
  
###################comparison paid wage##################  
########### comparison with payed wage ################  
  
  
# Calculate the mean PAID\_WAGE\_PER\_YEAR by job subtitle and state  
state\_job\_med\_paidwage <- aggregate(top\_states\_data$PAID\_WAGE\_PER\_YEAR,   
 by = list(top\_states\_data$Work\_State, top\_states\_data$Job\_Title\_Sub),  
 FUN = median)  
  
# Rename the columns of the state\_job\_means data frame  
names(state\_job\_med\_paidwage) <- c("State", "Job\_Title\_Sub", "Med\_Paid\_Wage")  
  
# Order the data frame by state and mean paid wage in descending order  
state\_job\_med\_paidwage <- state\_job\_med\_paidwage[order(state\_job\_med\_paidwage$State, -state\_job\_med\_paidwage$Med\_Paid\_Wage),]  
  
# Print the state\_job\_means data frame  
print(state\_job\_med\_paidwage)  
  
  
  
  
  
###############################  
  
# Load ggplot2 library  
library(ggplot2)  
library(scales)  
  
# Create a new data frame by merging state\_job\_means and state\_job\_means\_paidwage data frames  
state\_job\_means\_all <- merge(state\_job\_medians, state\_job\_med\_paidwage, by = c("State", "Job\_Title\_Sub"))  
  
  
  
##########We can also look at the bottom states:   
  
  
# First, extract the bottom 5 states  
bottom\_states <- tail(all\_states\_med\_paid, 5)$State  
  
# Filter the data by the bottom 5 states  
bottom\_states\_data <- salcost[salcost$Work\_State %in% bottom\_states, ]  
  
# Calculate the mean ADJUSTED\_WAGE by job subtitle and state  
state\_job\_med\_b <- aggregate(bottom\_states\_data$ADJUSTED\_WAGE,   
 by = list(bottom\_states\_data$Work\_State, bottom\_states\_data$Job\_Title\_Sub),  
 FUN = median)  
  
# Rename the columns of the state\_job\_means data frame  
names(state\_job\_med\_b) <- c("State", "Job\_Title\_Sub", "Med\_Adjusted\_Wage")  
  
# Order the data frame by state and mean adjusted wage in descending order  
state\_job\_med\_b <- state\_job\_med\_b[order(state\_job\_med\_b$State, -state\_job\_med\_b$Med\_Adjusted\_Wage),]  
  
# Print the state\_job\_means data frame  
print(state\_job\_med\_b)  
  
  
########################  
  
# Calculate the median PAID\_WAGE\_PER\_YEAR by job subtitle and state  
state\_job\_med\_paidwage\_b <- aggregate(bottom\_states\_data$PAID\_WAGE\_PER\_YEAR,   
 by = list(bottom\_states\_data$Work\_State, bottom\_states\_data$Job\_Title\_Sub),  
 FUN = median)  
  
# Rename the columns of the state\_job\_means data frame  
names(state\_job\_med\_paidwage\_b) <- c("State", "Job\_Title\_Sub", "Med\_Paid\_Wage")  
  
# Order the data frame by state and mean paid wage in descending order  
state\_job\_med\_paidwage\_b <- state\_job\_med\_paidwage\_b[order(state\_job\_med\_paidwage\_b$State, -state\_job\_med\_paidwage\_b$Med\_Paid\_Wage),]  
  
# Print the state\_job\_means data frame  
print(state\_job\_med\_paidwage\_b)  
  
############################### Here the colors are swaped. we have adjusted to be lighter and paid wage stronger color..   
  
# Load ggplot2 library graph for bottom 5  
library(ggplot2)  
library(dplyr)  
  
# Create a new data frame by merging state\_job\_means and state\_job\_means\_paidwage data frames  
state\_job\_means\_all\_b <- merge(state\_job\_med\_b, state\_job\_med\_paidwage\_b, by = c("State", "Job\_Title\_Sub"))  
  
  
  
###################################new  
library(ggplot2)  
library(dplyr)  
  
# Create the grouped bar plot  
p1 <- ggplot(state\_job\_means\_all, aes(x = State, y = Median\_Adjusted\_Wage, fill = Job\_Title\_Sub)) +  
 geom\_bar(position = "dodge", stat = "identity", width = 0.6) +  
 geom\_bar(aes(y = Med\_Paid\_Wage), position = "dodge", stat = "identity", width = 0.6, alpha = 0.5) +  
 scale\_fill\_manual(values = c("#F8766D", "#00BA38", "#619CFF", "#FC8D62", "#8B00FF")) +  
 labs(x = "State", y = "Med Salary", fill = "Job Title Sub") +  
 ggtitle("Median Adjusted (darker) and Median Paid Wages (lighter)") +  
 theme(plot.title = element\_text(size = 16, hjust = 0.5), axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(labels = scales::comma, limits = c(0, 150000))  
  
# Create the grouped bar plot  
p2 <- ggplot(state\_job\_means\_all\_b, aes(x = State, y = Med\_Paid\_Wage, fill = Job\_Title\_Sub)) +  
 geom\_bar(position = "dodge", stat = "identity", width = 0.6) +  
 geom\_bar(aes(y = Med\_Adjusted\_Wage), position = "dodge", stat = "identity", width = 0.6, alpha = 0.5) +  
 scale\_fill\_manual(values = c("#F8766D", "#00BA38", "#619CFF", "#FC8D62", "#8B00FF")) +  
 labs(x = "State", y = "Med Salary", fill = "Job Title Sub") +  
 ggtitle("Median Adjusted (lighter) and Median Paid Wages (darker)") +  
 theme(plot.title = element\_text(size = 16, hjust = 0.5), axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(labels = scales::comma, limits = c(0, 150000))  
  
# Display plots side-by-side with shared y-axis limits  
library(gridExtra)  
grid.arrange(p1, p2, ncol = 2, widths = c(1, 1))  
  
  
# Calculate the median cost index by state  
state\_cost\_index <- aggregate(salcost$costIndex, by = list(salcost$Work\_State), FUN = median)  
  
# Rename the columns of the state\_cost\_index data frame  
names(state\_cost\_index) <- c("State", "Median\_Cost\_Index")  
  
# Order the data frame by mean cost index in descending order  
state\_cost\_index <- data.frame(state\_cost\_index[order(-state\_cost\_index$Median\_Cost\_Index), ])  
  
# Print the state\_cost\_index data frame  
#print(state\_cost\_index)  
  
#head(salcost)  
  
# Calculate the mean paid wage per year by state  
state\_all\_means\_paid <- aggregate(PAID\_WAGE\_PER\_YEAR ~ Work\_State, data = salcost, FUN = mean)  
  
# Rename the columns of the state\_means data frame  
names(state\_all\_means\_paid) <- c("State", "Mean\_Paid\_Wage\_Per\_Year")  
  
# Order the data frame by mean paid wage per year in descending order  
state\_all\_means\_paid <- state\_all\_means\_paid[order(-state\_all\_means\_paid$Mean\_Paid\_Wage\_Per\_Year),]  
  
# Print the state\_means data frame  
#print(state\_all\_means\_paid)  
# Print the state medians data frame  
#print(all\_states\_med\_paid)  
  
  
# Calculate the mean adjusted wage by state  
state\_all\_means\_adjusted <- aggregate(ADJUSTED\_WAGE ~ Work\_State, data = salcost, FUN = mean)  
  
# Rename the columns of the state\_means\_adjusted data frame  
names(state\_all\_means\_adjusted) <- c("State", "Mean\_Adjusted\_Wage")  
  
# Order the data frame by mean paid wage per year in descending order  
state\_all\_means\_adjusted <- state\_all\_means\_adjusted[order(-state\_all\_means\_adjusted$Mean\_Adjusted\_Wage),]  
  
# Print the state\_means data frame  
#print(state\_all\_means\_adjusted)  
# Print the state medians data frame  
#print(all\_states\_med\_adjusted)  
  
####################################################################  
#Graph of all states means percentage change in wage after adjusted mean wage  
  
# Merge the state\_means and state\_means\_adjusted data frames by state  
merged\_means <- merge(state\_all\_means\_paid, state\_all\_means\_adjusted, by = "State")  
  
  
# Add the WORK\_STATE\_ABBREVIATION column to merged\_means  
merged\_means$WORK\_STATE\_ABBREVIATION <- salcost$WORK\_STATE\_ABBREVIATION[match(merged\_means$State, salcost$Work\_State)]  
  
# Calculate the percentage change of the adjusted wage  
merged\_means$Percent\_Change <- (merged\_means$Mean\_Adjusted\_Wage - merged\_means$Mean\_Paid\_Wage\_Per\_Year) / merged\_means$Mean\_Paid\_Wage\_Per\_Year \* 100  
  
# Add a column indicating whether the percentage change is positive or negative  
merged\_means$Positive\_Change <- ifelse(merged\_means$Percent\_Change > 0, "Yes", "No")  
  
# Order the data frame by percent change in descending order  
merged\_means <- merged\_means[order(-merged\_means$Percent\_Change),]  
  
# Print the merged\_means data frame  
print(merged\_means)  
  
  
ggplot(data = merged\_means, aes(x = WORK\_STATE\_ABBREVIATION)) +  
 geom\_bar(aes(y = Percent\_Change, fill = Positive\_Change), alpha = 0.8, stat = "identity") +  
 scale\_fill\_manual(values = c("red", "green"), guide = "none" ) +  
 geom\_text(aes(y = Percent\_Change, label = paste0(round(Percent\_Change, 1), "%")), vjust = -0.5) +  
 scale\_y\_continuous(labels = scales::percent\_format()) +  
 labs(title = "Mean Adjusted Wage and Percent Change by State",  
 x = "State Abbreviation",  
 y = "Percent Change",  
 color = "Positive Change") +  
 theme\_minimal()