

Exam2

Kavya Sethi

6/28/2021

Contents

1. Clearing environment in r.	1
2. Loading the college_scorecard and name it "college_scorecard"	2
3. Summary statistics for the college_scorecard dataset.	2
4. Creating small_scorecard	2
5. Collapse small_card into even_smaller_scorecard"	3
6. even_smaller_scorecard bar graph	4
7. On the basis of the Graph	4
8. Load avocado dataset	5
9. Capture Year Avocados were sold	6
10. Deflated Data	6
11. Collapse Data into collapsed_avocados	7
12. Reshape collapsed_avocado wide	7
Label your variables on wide_avocados	8
14. Load training dataset	8
15. Reshaping training data long	8
16 Load the titanic in R and call the resulting data frame titanic	9
17. Summary Statistics of Titanic	9
18. Correlation between gender and survival	9
BONUS	31

1. Clearing environment in r.

This code clears the global environment where loaded data appears.

```
rm(list=ls(all=TRUE))
```

2. Loading the college_scorecard and name it “college_scorecard”

To load data in R, I must first call on package rio and then upload data.

```
library(rio)

college_scorecard = import("2021_exam2_data.xlsx", which = 4)
```

Note that I designated which tab of the exam sheet through, “which = 4” in the previous code chunk

3. Summary statistics for the college_scorecard dataset.

Simply use summary() which is a part of r base.

```
summary(college_scorecard)
```

##	unitid	inst_name	state_abbr
##	Min. :100654	Length:48445	Length:48445
##	1st Qu.:163532	Class :character	Class :character
##	Median :212115	Mode :character	Mode :character
##	Mean :260438		
##	3rd Qu.:409120		
##	Max. :490009		
##			
##	pred_degree_awarded_ipeds	year	earnings_med
##	Min. :1.000	Min. :2007	Min. : 8400
##	1st Qu.:1.000	1st Qu.:2011	1st Qu.: 24700
##	Median :2.000	Median :2012	Median : 31600
##	Mean :1.913	Mean :2012	Mean : 33348
##	3rd Qu.:3.000	3rd Qu.:2014	3rd Qu.: 39800
##	Max. :3.000	Max. :2016	Max. :186500
##			NA's :15706
##	count_working		count_not_working
##	Min. : 8		Min. : 0.0
##	1st Qu.: 210		1st Qu.: 46.0
##	Median : 594		Median : 115.0
##	Mean : 2073		Mean : 369.4
##	3rd Qu.: 1477		3rd Qu.: 300.0
##	Max. :94724		Max. :15960.0
##	NA's :14772		NA's :15801

NICE!

4. Creating small_scorecard

small_scorecard includes data measured in 2014 and 2015 on former students who graduated from four-year+ colleges and universities located in Texas (state_abbr: “TX”) and Louisiana (state_abbr: “LA”).

```
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
```

```
#Check class of year so that I know is 2014/2014 should be put in "".
class(college_scorecard$year)
```

```
## [1] "numeric"
```

```
# Create a vector to filter both TX and LA/2014 and 2015 through the code below
state_names <- c("TX", "LA")
small_years <- c(2015,2014)

small_scorecard <- filter(college_scorecard, year== small_years)
```

```
## Warning in year == small_years: longer object length is not a multiple of
## shorter object length
```

```
small_scorecard <- filter(small_scorecard,pred_degree_awarded_ipeds==3)
small_scorecard <- filter(small_scorecard,state_abbr == state_names)
```

```
## Warning in state_abbr == state_names: longer object length is not a multiple of
## shorter object length
```

Seems to be some values missing, but after filtering with excel data there are values missing with specified variables. Note call dplyr because it includes filter().

5.Collapse small_card into even_smaller_scorecard”

Get average of number people working who graduated from universities in Texas and Lousiana and total number of people working who graduated from universities in Texas and Lousiana.

Name it “even_smaller_scorecard”

```
small_scorecard$total = small_scorecard$count_not_working + small_scorecard$count_working

even_smaller_scorecard <-
  small_scorecard%>%
  group_by(state_abbr) %>% # tell R the unique IDs
  summarize(across(where(is.numeric), sum, na.rm = TRUE))%>% # summarize numeric vars by
  select(-c("unitid", "pred_degree_awarded_ipeds", "year", "earnings_med", "count_not_working"))

print(even_smaller_scorecard)
```

```
## # A tibble: 2 x 3
##   state_abbr count_working total
##   <chr>         <dbl>   <dbl>
## 1 LA             8223    9386
## 2 TX            157765 182538
```

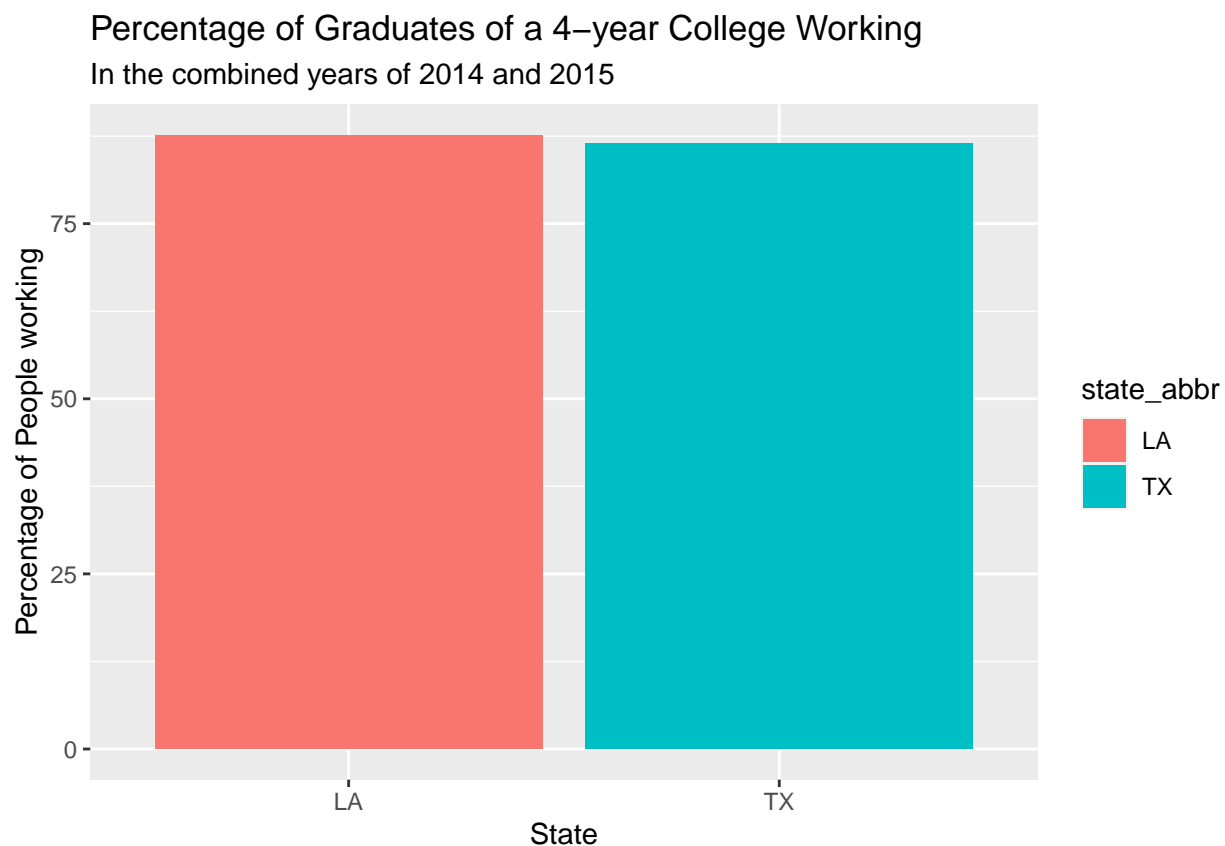
6. even_smaller_scorecard bar graph

use ggplot2

```
even_smaller_scorecard$percentage <- (even_smaller_scorecard$count_working/even_smaller_scorecard$total*100)

library(ggplot2)

even_smaller_scorecardgraph <- ggplot(even_smaller_scorecard, aes(x=state_abbr,y=percentage, fill = state_abbr)) +
  geom_col() + labs( x = "State", y = "Percentage of People working",
  title ="Percentage of Graduates of a 4-year College Working", subtitle = "In the combined years of 2014 and 2015")
print(even_smaller_scorecardgraph)
```



7. On the basis of the Graph

Broadly speaking, the bar graphs between TX and LA appear to be relatively similar. The percentage difference between the states is too small to be significant.

```
summary(even_smaller_scorecard)
```

```
##   state_abbr      count_working      total      percentage
## Length:2         Min.   : 8223   Min.   : 9386   Min.   :86.43
## Class :character  1st Qu.: 45608   1st Qu.: 52674   1st Qu.:86.72
## Mode  :character  Median : 82994   Median : 95962   Median :87.02
##                      Mean   : 82994   Mean   : 95962   Mean   :87.02
##                      3rd Qu.:120380   3rd Qu.:139250   3rd Qu.:87.31
##                      Max.    :157765   Max.    :182538   Max.    :87.61
```

```
summary(small_scorecard)
```

```
##      unitid      inst_name      state_abbr
## Min.   :158802   Length:64   Length:64
## 1st Qu.:223084   Class :character   Class :character
## Median :227287   Mode  :character   Mode  :character
## Mean   :247536
## 3rd Qu.:228920
## Max.   :484756
##
## pred_degree_awarded_ipeds      year      earnings_med      count_not_working
## Min.   :3                     Min.   :2014   Min.   :23400   Min.   : 27.0
## 1st Qu.:3                     1st Qu.:2014   1st Qu.:35400   1st Qu.: 72.0
## Median :3                     Median :2014   Median :40200   Median : 163.0
## Mean   :3                     Mean   :2014   Mean   :40310   Mean   : 894.3
## 3rd Qu.:3                     3rd Qu.:2015   3rd Qu.:47100   3rd Qu.: 315.0
## Max.   :3                     Max.   :2015   Max.   :54900   Max.   :15960.0
##                      NA's   :35   NA's   :35
## count_working      total
## Min.   : 147   Min.   : 174
## 1st Qu.: 491   1st Qu.: 547
## Median :1106   Median : 1297
## Mean   : 5724   Mean   : 6618
## 3rd Qu.:2267   3rd Qu.: 2562
## Max.   :94724   Max.   :110684
## NA's   :35     NA's   :35
```

Based on the data summaries and graph, I do not think there is a significant difference between the state of Texas and Louisiana. I was unable to calculate other states in the college_scoreboard so I would not be comfortable guessing if there is advantage in gaining employment between all 50 states or universities. But considering that the data is so board, it is likely that some particular states or particular universities have a higher percentage of graduates employed.

8. Load avocado dataset

```
library(rio)

avocados <- import("2021_exam2_data.xlsx", which = 2)
```

9. Capture Year Avocados were sold

Use lubridate package to extrapolate the year into a new column

```
library(lubridate)

##
## Attaching package: 'lubridate'

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

library(nycflights13)
library(dplyr)
avocado_dates <- avocados

avocados<-
avocado_dates %>%
  dplyr::mutate(lubridate::year(avocado_dates$date))
```

10. Deflated Data

Google WDI GDP deflator

```
library(WDI)
# https://data.worldbank.org/indicator/NY.GDP.DEFL.ZS
deflator_data = WDI(country = "all", indicator = c("NY.GDP.DEFL.ZS"),
  start = 2015, # start of foreign aid data
  end = 2018, # end of of foreign aid data
  extra = FALSE, cache = NULL)

# rename variables so they are understandable using the data.table package
library(data.table)

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##   hour, isoweek, mday, minute, month, quarter, second, wday, week,
##   yday, year

## The following objects are masked from 'package:dplyr':
##
##   between, first, last
```

```

setnames(deflator_data,"NY.GDP.DEFL.ZS", "deflator")

setnames(avocados,"lubridate::year(avocado_dates$date)","year")

# select only the United States data
usd_deflator = subset(deflator_data, country=="United States")

# To determine base year
subset(usd_deflator, deflator==100)

```

```

##      iso2c      country deflator year
## 1016    US United States      100 2015

```

```

deflated_avocados= left_join(avocados, usd_deflator, by=c("year"))

deflated_avocados$deflated_average_price = deflated_avocados$average_price/(deflated_avocados$deflator/

```

11. Collapse Data into collapsed_avocados

```

collapsed_avocados <- deflated_avocados %>%
  group_by(year) %>% # tell R the unique IDs
  summarize(across(where(is.numeric), sum)) %>% # summarize numeric vars by
  select(-c("average_price","total_volume","deflator"))

head(collapsed_avocados)

```

```

## # A tibble: 4 x 2
##   year deflated_average_price
##   <dbl>             <dbl>
## 1  2015                53.1
## 2  2016                53.8
## 3  2017                64.5
## 4  2018                12.3

```

12. Reshape collapsed_avocado wide

```

library(tidyverse)

```

```

## -- Attaching packages ----- tidyverse 1.3.1 --

## v tibble  3.1.2    v purrr  0.3.4
## v tidyr   1.1.3    v stringr 1.4.0
## v readr   1.4.0    v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()

```

```
## x data.table::between()      masks dplyr::between()
## x lubridate::date()         masks base::date()
## x dplyr::filter()           masks stats::filter()
## x data.table::first()       masks dplyr::first()
## x data.table::hour()        masks lubridate::hour()
## x lubridate::intersect()     masks base::intersect()
## x data.table::isoweek()     masks lubridate::isoweek()
## x dplyr::lag()              masks stats::lag()
## x data.table::last()        masks dplyr::last()
## x data.table::mday()        masks lubridate::mday()
## x data.table::minute()      masks lubridate::minute()
## x data.table::month()       masks lubridate::month()
## x data.table::quarter()     masks lubridate::quarter()
## x data.table::second()      masks lubridate::second()
## x lubridate::setdiff()      masks base::setdiff()
## x purrr::transpose()       masks data.table::transpose()
## x lubridate::union()        masks base::union()
## x data.table::wday()        masks lubridate::wday()
## x data.table::week()        masks lubridate::week()
## x data.table::yday()        masks lubridate::yday()
## x data.table::year()        masks lubridate::year()
```

```
wide_avocados <-
  collapsed_avocados %>%
    pivot_wider(id_cols = "year", names_from = year, values_from = deflated_average_price)
head(wide_avocados)
```

```
## # A tibble: 1 x 4
##   '2015' '2016' '2017' '2018'
##   <dbl> <dbl> <dbl> <dbl>
## 1   53.1   53.8   64.5   12.3
```

Label your variables on wide_avocados

```
library(labelled)
var_label(wide_avocados) <- list('2015' = "Year", '2016' = "Year", '2017' = "Year", '2018' = "Year")
```

14. Load training dataset

```
library(rio)

training = import("2021_exam2_data.xlsx", which = 3)
```

15. Reshaping training data long


```
long_training <-
  training %>%
  pivot_longer(cols = starts_with("re_"), names_to = "Year", names_prefix = "re_")

summary(long_training)
```

```
##  training_program      age      educ      black
##  Min.   :0.0000  Min.   :17.00  Min.   : 3.0  Min.   :0.0000
##  1st Qu.:0.0000  1st Qu.:20.00  1st Qu.: 9.0  1st Qu.:1.0000
##  Median :0.0000  Median :24.00  Median :10.0  Median :1.0000
##  Mean   :0.4157  Mean   :25.37  Mean   :10.2  Mean   :0.8337
##  3rd Qu.:1.0000  3rd Qu.:28.00  3rd Qu.:11.0  3rd Qu.:1.0000
##  Max.   :1.0000  Max.   :55.00  Max.   :16.0  Max.   :1.0000
##      hisp      marr      Year      value
##  Min.   :0.00000  Min.   :0.0000  Length:1335  Min.   : 0
##  1st Qu.:0.00000  1st Qu.:0.0000  Class :character  1st Qu.: 0
##  Median :0.00000  Median :0.0000  Mode  :character  Median : 0
##  Mean   :0.08764  Mean   :0.1685           Mean   : 2927
##  3rd Qu.:0.00000  3rd Qu.:0.0000           3rd Qu.: 4045
##  Max.   :1.00000  Max.   :1.0000           Max.   :60308
```

16 Load the titanic in R and call the resulting data frame titanic

```
library(rio)

titanic = import("2021_exam2_data.xlsx", which = 1)
```

17. Summary Statistics of Titanic

```
summary(titanic)
```

```
##      class      age      female      survived
##  Min.   :1.000  Min.   :0.0000  Min.   :0.0000  Min.   :0.000
##  1st Qu.:2.000  1st Qu.:1.0000  1st Qu.:1.0000  1st Qu.:0.000
##  Median :3.000  Median :1.0000  Median :1.0000  Median :0.000
##  Mean   :2.977  Mean   :0.9505  Mean   :0.7865  Mean   :0.323
##  3rd Qu.:4.000  3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:1.000
##  Max.   :4.000  Max.   :1.0000  Max.   :1.0000  Max.   :1.000
```

18. Correlation between gender and survival

```
titanic$male = NA
titanic$male[titanic$female==0]=1
titanic$male[titanic$female==1]=0

titanic$notsurvived = NA
```

```

titantic$notsurvived[titantic$survived==1]=0
titantic$notsurvived[titantic$survived==0]=1

cor(titantic$female,titantic$survived)

```

```
## [1] -0.4556048
```

```
cor(titantic$male,titantic$survived)
```

```
## [1] 0.4556048
```

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
## order_by
```

```
summaryBy(male ~ survived, data=titantic, FUN=c(mean,length))
```

```
##   survived  male.mean male.length
## 1         0 0.08456376         1490
## 2         1 0.48382560          711
```

The tab shows us that of those that survived (1), the male mean was higher than those that didn't survive. Meaning more men survived. So the correlation between the male gender and surviving 0.4556048, means that more men survived.

19. ifelse first class

```
FirstClass <- titantic$class
```

```
ifelse(test = FirstClass == 1, yes = "Passenger had First Class", no = "Passenger did not have first class")
```

```
##      [1] "Passenger had First Class"      "Passenger had First Class"
##      [3] "Passenger had First Class"      "Passenger had First Class"
##      [5] "Passenger had First Class"      "Passenger had First Class"
##      [7] "Passenger had First Class"      "Passenger had First Class"
##      [9] "Passenger had First Class"      "Passenger had First Class"
##     [11] "Passenger had First Class"      "Passenger had First Class"
##     [13] "Passenger had First Class"      "Passenger had First Class"
##     [15] "Passenger had First Class"      "Passenger had First Class"
##     [17] "Passenger had First Class"      "Passenger had First Class"
##     [19] "Passenger had First Class"      "Passenger had First Class"
##     [21] "Passenger had First Class"      "Passenger had First Class"
##     [23] "Passenger had First Class"      "Passenger had First Class"
##     [25] "Passenger had First Class"      "Passenger had First Class"
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## [2187] "Passenger did not have first class" "Passenger did not have first class"
## [2189] "Passenger did not have first class" "Passenger did not have first class"
## [2191] "Passenger did not have first class" "Passenger did not have first class"
## [2193] "Passenger did not have first class" "Passenger did not have first class"
## [2195] "Passenger did not have first class" "Passenger did not have first class"
## [2197] "Passenger did not have first class" "Passenger did not have first class"
## [2199] "Passenger did not have first class" "Passenger did not have first class"
## [2201] "Passenger did not have first class"
```

BONUS

“My Heart Will Go On” by Céline Dion