PHW251 Take-home Midterm Exam

ANSWER KEY

November 2023

Instructions

Where to put your code and answers

This R Markdown document gives the instructions for the take-home portion of your midterm exam. Submit your completed exam by filling in the "responses.rmd" file. When completing this Rmd, please include your answer for each question where it says **ANSWER HERE:**. Please also provide your code in the indicated code chunks. The code portion will be used to assign partial credit.

How to answer questions

Please carefully read the **Necessary Coding Steps** instructions for each question.

All questions below should be answered using R. Unless otherwise specified, you may use any method (base R, tidyverse, or other) to answer these questions. Please type out your answers in the specified area **ANSWER HERE:** (even if the answer is also available in your code chunk). Code will be used to give partial credit for incorrect answers.

Submitting

When you are done, knit the "responses.rmd" file to PDF and load it into Gradescope before the due date. Additionally, keep in mind best practices including ensuring code does not run of the page and not printing entire dataframes within your final PDF. **When knitted, your PDF should be approximately one page per question** (points will be deducted if too long with unnecessary output).

Scoring

Your exam will be worth **15 points total**. Each question indicates the number of points for that question.

Data

For all questions below, use the exam_data_2023_version1.csv file that is saved on DataHub/GitHub at PHW251_Fall2023/midterm/data/exam_data_2023_version1.csv. This is a real dataset from the California Health and Human Services Open Data Portal, but has been altered slightly for the purpose of this exam. The dataset contains counts of emergency department encounters at California medical facilities.

The file includes the following columns:

- **Year**

- **OSHPD ID**
- **Facility Name**
- **County Name**
- **ER Service Level Desc:** Level of ER service. Options include: BASIC, COMPREHENSIVE, STANDBY, NOT APPLICABLE
- **Type:** Specifies encounter type. Options include: ED_Visit (Encounter in which patient is treated in the Emergency Department and then released), ED_Admit (Encounter in which the patient is initially treated in the Emergency Department and then admitted to the same hospital for continued inpatient care). Categories are mutually exclusive.
- **Count**

Exam Questions

Question 1

Necessary Coding Steps

Import the csv data file. Check what data type each column is.

```
# students:
# to run code, please put the correct file path for your setup on
# the right side in quotations.
file_path <- paste0("./data/randomized_exam_data/exam_data_", params$year,
                    "_version", params$version, ".csv")

ed <- read_csv(file_path, show_col_types = FALSE)

# one option - save types in a vector
col_types <- unlist(lapply(ed, class))
col_types
```

```
##                  Year              OSHPD ID         Facility Name           County Name
##             "numeric"             "numeric"           "character"           "character"
## ER Service Level Desc                  Type                 Count
##           "character"           "character"             "numeric"
```

```
# can also use str or glimpse
# (used glimpse here because str goes off the page)
glimpse(ed)
```

```
## Rows: 1,690
## Columns: 7
## $ Year                    <dbl> 2013, 2013, 2014, 2014, 2015, 2015, 2016, 2016, 2017, 201~
## $ ‘OSHPD ID‘              <dbl> 106034002, 106034002, 106034002, 106034002, 106034002, 10~
## $ ‘Facility Name‘         <chr> "SUTTER AMADOR HOSPITAL", "SUTTER AMADOR HOSPITAL", "SUTT~
## $ ‘County Name‘           <chr> "AMADOR", "AMADOR", "AMADOR", "AMADOR", "AMADOR", "AMADOR~
## $ ‘ER Service Level Desc‘ <chr> "BASIC", "BASIC", "BASIC", "BASIC", "BASIC", "BASIC", "BA~
## $ Type                    <chr> "ED_Admit", "ED_Visit", "ED_Admit", "ED_Visit", "ED_Admit~
## $ Count                   <dbl> 1834, 17696, 1826, 18813, 1943, 20422, 1860, 22002, 1887,~
```

1A. What are the data types of each column when you read the data into R (numeric, factor, logical, character, etc)? [1 pt]

ANSWER:

- Year: numeric
- YOSHPD ID: numeric
- Facility Name: character
- County Name: character
- ER Service Level Desc: character
- Type: character
- Count: numeric

Question 2

Necessary Coding Steps

Notice the column names are not reading in in a very user-friendly way. Rename all columns to align with best practices for naming columns (lowercase with underscores in place of spaces).

2A. What are the new column names? Please list the new column names below. [1 pt]

```r
ed <- ed %>%
  rename_with(~ tolower(gsub(" ", "_", .x, fixed = TRUE)))

# another method using stringr
# colnames(ed) <- str_to_lower(gsub(" ", "_", colnames(ed)))
```

ANSWER: year, oshpd_id, facility_name, county_name, er_service_level_desc, type, count

Question 3

Questions 3-7 are designed to build off each other.

Necessary Coding Steps

Using the data frame from question 2, create a new data frame that limits the data frame to only contain rows where the type of service was "basic" and year is between 2014 and 2018 (inclusive of these years).

3A. How many records are in the new subsetted dataset? [1 pt]

```r
# students: To run the code,
# replace the code on the right sides of the arrows with the year you were
# given in the problem
min_yr <- years_min + 1
max_yr <- years_max

ed3a <- ed %>%
    filter(er_service_level_desc == "BASIC" & year %in% min_yr:max_yr)

# another filter method to accomplish the same thing
# ed13 <- ed %>%
#    filter(er_service_level_desc=="BASIC" & year >= min_year, year <= max_year)
nrow(ed3a)
```

```
## [1] 1154
```

ANSWER: 1,154

Question 4

Necessary Coding Steps

Using the data frame created in question 3, create a new column called `total_encounters` by grouping OSHPD ID and Year and then summing the values in the `count` column to get total encounters. *(Hint: After adding this column your data frame should contain the same number of rows that it had before you added the column.)*

```
# students: to run the code
# replace the code on the right sides of the arrows with the year and
# the hospital you were given in the problem
single_year <- single_year1
single_hosp <- single_hosp1

ed4a <- ed3a %>%
  group_by(oshpd_id,year) %>%
  mutate(total_encounters = sum(count)) %>%
  ungroup()

temp <- ed4a %>% filter(facility_name == single_hosp1 &
                          year == single_year1)


total_encounters <- temp[["total_encounters"]]

total_encounters
```

4A. What is the value of `total_encounters` for LOS ALAMITOS MEDICAL CENTER in 2018? [1 pt]

```
## [1] 34547 34547
```

Answer: 34,547

Necessary Coding Steps

Create another new column called `pct_encounter_type` that calculates the percent of ED encounters that were visits or admits. Display the percentage as multiplied by 100 and rounded to 1 decimal (for example, 35.1% would be displayed as 35.1).

```
# students: to run the code,
# replace the code on the right sides of the arrows with the year and
# the hospital you were given in the problem
single_year <- single_year2
single_hosp <- single_hosp2
```

```r
ed4b <- ed4a %>%
  mutate(pct_encounter_type = round(100*count/total_encounters,1))

pct_encounter_type <- ed4b %>%
  filter(facility_name == single_hosp2 &
         year == single_year2 & type == "ED_Admit") %>%
  pull(pct_encounter_type)

pct_encounter_type
```

4B. What is the value of `pct_encounter_type` for ED admits at SUTTER DAVIS HOSPITAL in 2016? [1 pt]

```
## [1] 6.3
```

Answer: 6.3%

Question 5

Necessary Coding Steps

Using the data frame created in question 4, first create a subset table that only includes rows for ED admits. Then use the arrange function to order the data frame to display rows first by lowest to highest year and then by highest to lowest value of `pct_encounter_type` (at the same time).

```r
ed5a <- ed4b %>%
  filter(type=="ED_Admit") %>%
  arrange(year, desc(pct_encounter_type))
```

5A. Show a single line of code that can be used for this arrange step. [1 pt]

ANSWER: arrange(year, desc(pct_encounter_type))

5B. What code would you use to obtain only the facility names for the first 5 rows of the dataset created in question 5A (facilities with highest values in the `pct_encounter_type` column for the first year in the data frame)? [1 pt]

ANSWER: df[1:5,3] is one way to get this. This gives us the result ed5a[1:5,3]. See code below for more ways to do this.

```r
ed5b <- ed5a %>%
  select(facility_name) %>%
  head(5)

ed5b
```

```
## # A tibble: 5 x 1
##   facility_name
##   <chr>
## 1 LOMA LINDA UNIVERSITY MEDICAL CENTER
## 2 MISSION HOSPITAL LAGUNA BEACH
## 3 MISSION HOSPITAL REGIONAL MEDICAL CENTER
## 4 ORANGE COAST MEMORIAL MEDICAL CENTER
## 5 JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS
```

```r
# Different ways of getting correct answer
ed5b_2 <- ed5a[1:5,3]
ed5b_2
```

```
## # A tibble: 5 x 1
##   facility_name
##   <chr>
## 1 LOMA LINDA UNIVERSITY MEDICAL CENTER
## 2 MISSION HOSPITAL LAGUNA BEACH
## 3 MISSION HOSPITAL REGIONAL MEDICAL CENTER
## 4 ORANGE COAST MEMORIAL MEDICAL CENTER
## 5 JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS
```

```
ed5b_3 <- head(ed5a[,3], 5)
ed5b_3
```

```
## # A tibble: 5 x 1
##   facility_name
##   <chr>
## 1 LOMA LINDA UNIVERSITY MEDICAL CENTER
## 2 MISSION HOSPITAL LAGUNA BEACH
## 3 MISSION HOSPITAL REGIONAL MEDICAL CENTER
## 4 ORANGE COAST MEMORIAL MEDICAL CENTER
## 5 JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS
```

```
ed5b_4 <- ed5a[1:5,"facility_name"]
ed5b_4
```

```
## # A tibble: 5 x 1
##   facility_name
##   <chr>
## 1 LOMA LINDA UNIVERSITY MEDICAL CENTER
## 2 MISSION HOSPITAL LAGUNA BEACH
## 3 MISSION HOSPITAL REGIONAL MEDICAL CENTER
## 4 ORANGE COAST MEMORIAL MEDICAL CENTER
## 5 JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS
```

Question 6

Necessary Coding Steps

Using the dataset created in question 5A, find the average (mean) value of percent of encounter types (`pct_encounter_type`) that were admits among all facilities from 2014 to 2018. Use this mean value to create another new column called `above_below_avg` that categorizes facilities with `pct_encounter_type` equal to or above average as "above"; otherwise, categorize as "below".

```r
#one way
ed6a <- ed5a %>%
  mutate(avg = mean(pct_encounter_type),
         above_below_avg = if_else(pct_encounter_type>=avg,"above","below"))

# or calculate the average specifically
mean_ed_visits <- mean(ed5a$pct_encounter_type)
mean_ed_visits
```

6A. What is the average (mean) percentage of encounters that are ED admits for all facilities from 2014 to 2018? ***[1 pt]***

```
## [1] 12.57314
```

```r
# then calculate above_below_avg
ed6a <- ed5a %>%
  mutate(above_below_avg = if_else(pct_encounter_type >=
                                     mean_ed_visits,"above","below"))
```

ANSWER HERE: 12.57%

```r
# for students:
# replace the code on the right sides of the arrows with the year and
# the hospital you were given in the problem
single_year <- single_year1
single_hosp <- single_hosp3

value <- ed6a %>%
  filter(year == single_year & facility_name == single_hosp3) %>%
  pull(above_below_avg)

value
```

6B. What is the value of `above_below_avg` for ER admits at SOUTH COAST GLOBAL MEDICAL CENTER for 2018?

```
## [1] "below"
```

ANSWER: below

Question 7

Necessary Coding Steps

Restrict the dataset created in 6A to only include records for TULARE, CONTRA COSTA, and SANTA CLARA facilities for the year 2016.

```r
# for students:
# replace the code on the right sides of the arrows with the year and
# the counties (in quotation marks) that you were given in the problem
single_year <- single_year2
counties <- c(single_county1, single_county2,single_county3)


ed7a <- ed6a %>%
  filter(county_name %in% counties & year == single_year)

nrow(ed7a)
```

```
## [1] 18
```

7A. How many records remain? [1 pt] ANSWER: 18

```r


hosp <- ed7a %>%
          arrange(desc(pct_encounter_type)) %>%
          head(1) %>%
          pull(facility_name)

hosp
```

7B. Using the data frame created in question 7A, what hospital has the highest percent of encounters that are admits? [1 pt]

```
## [1] "JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS"
```

ANSWER: JOHN MUIR MEDICAL CENTER-WALNUT CREEK CAMPUS

Question 8

Necessary Coding Steps

For questions 8 & 9, please use the data frame from the end of question 2. To start, create a new subset data frame that only includes records for encounters that were ED visits (not admits) in the year 2018. Additionally, only include the following columns: Facility Name, County, ER Service Level visits, Type, and Count.

Create a new column called `county_visit_total` that contains the total number of ED visits for each county. *(Hint: this data frame should contain 1 row per county.)* Re-order the table to display the county with the highest number of ED visits at the top of the table.

```
# for students:
# replace the code on the right sides of the arrows with the year
# given in the problem
single_year <- years_max

ed8a <- ed %>%
  filter(year == single_year & type == "ED_Visit") %>%
  select(facility_name, county_name, er_service_level_desc, type, count)

ed8a <- ed8a %>%
  group_by(county_name) %>%
  summarise(county_visit_total = sum(count)) %>%
  ungroup() %>%
  arrange(desc(county_visit_total))

head(ed8a,10)
```

8A What county has the 10th highest total number of ED visits in 2018? [1 pt]

```
## # A tibble: 10 x 2
##    county_name    county_visit_total
##    <chr>                       <dbl>
##  1 SAN DIEGO                  858742
##  2 ORANGE                     840358
##  3 SAN BERNARDINO             775904
##  4 SANTA CLARA                492839
##  5 CONTRA COSTA               312575
##  6 FRESNO                     291371
##  7 STANISLAUS                 236829
##  8 SAN MATEO                  185666
##  9 SONOMA                     160948
## 10 MADERA                     135995
```

ANSWER HERE: MADERA

Question 9

Necessary Coding Steps

Building on to the data frame created in question 8, create a new column called `visit_category` indicating the categorical level of ER visits utilization (High, Medium, Low, Very Low) in each county. The categories should be defined as:

- "High": > 178649 ED visits
- "Medium": > 66521 ED visits
- "Low": > 22026 ED visits
- "Very Low": <= 22026 ED visits

Create a final table that summarizes the number of counties in each category *(Hint: this table should only have 4 rows).*

9A. How many counties are in the "low" coverage category? [1 pt]

```
ed9a <- ed8a %>%
  mutate(visit_category = case_when(
    county_visit_total > 178649 ~ "High",
    county_visit_total > 66521 ~ "Medium",
    county_visit_total > 22026 ~ "Low",
    TRUE ~ "Very low"
  )) %>%
  group_by(visit_category) %>%
  count()

ed9a
```

```
## # A tibble: 4 x 2
## # Groups:   visit_category [4]
##   visit_category     n
##   <chr>          <int>
## 1 High               8
## 2 Low                6
## 3 Medium             6
## 4 Very low          10
```

ANSWER HERE: 6

Question 10

Necessary Coding Steps

For question 10, please use the data frame from the end of question 2 – NOT the data frames that you created in questions 8 and 9.

Create a subset dataset with only records for 2013 and basic ER service level. Keep only the following columns: Facility Name, County Name, Type, and Count. Pivot the dataset to create columns for each of the ED encounter types; these columns should each contain the counts of encounters for each type (admit and visit).

```r
# for students:
# replace the code on the right sides of the arrows with the year
# given in the problem
single_year <- years_min

ed10a <- ed %>%
  filter(year == years_min & er_service_level_desc=="BASIC") %>%
  select(facility_name, county_name, type, count) %>%
  pivot_wider(names_from = "type", values_from = "count")
  # can also run this last line with no quotations around type
  # and count and it will work.

nrow(ed10a)
```

10A. How many records are in the dataset after the pivot? [1 pt]

```
## [1] 116
```

ANSWER: 116

10B. Include the line of code used to perform the pivot. Make sure to include the function name as well as the arguments used. ***[1 pt]*** **ANSWER HERE:** pivot_wider(names_from = type, values_from = count)

EXTRA CREDIT

[2 points total]

Question 11 (extra credit)

Complete questions 8 and 9 using only one dplyr call. In other words, start with the data frame from the end of question 2, perform the necessary subsetting, grouping, and summarizing with the end goal of producing a table that displays the number of counties in each visit count category.

Please include sufficient code for the teaching team to be able to run your code, if needed. This means either including the import statement for the csv before the dplyr call, or including the import statement as part of your dplyr call.

Hint: Including more than one `group_by()` in a single call may also require the use of `ungroup()`.

Paste the single dplyr call below. **[1 pt]**

```
edq11 <- ed %>%
  # for students, replace years_max with the year you were
  # given in problem 8.
  filter(year == years_max & type=="ED_Visit") %>%
  select(facility_name, county_name, er_service_level_desc, type, count) %>%
  group_by(county_name) %>%
  summarise(county_visit_total = sum(count)) %>%
  ungroup() %>%
  arrange(desc(county_visit_total)) %>%
  mutate(visit_category = case_when(
    county_visit_total > 178649 ~ "High",
    county_visit_total > 66521 ~ "Medium",
    county_visit_total > 22026 ~ "Low",
    TRUE ~ "Very low"
  )) %>%
  group_by(visit_category) %>%
  count()

edq11
```

```
## # A tibble: 4 x 2
## # Groups:   visit_category [4]
##   visit_category     n
##   <chr>          <int>
## 1 High               8
## 2 Low                6
## 3 Medium             6
## 4 Very low          10
```

Question 12 (extra credit)

Include code that uses the `kable` package to print the final table for question 9 in a print-friendly format (easy to read with meaningful column names and rows in descending order from High to Very Low). **[1 pt]**

Table 1: ED Visit Utilization by Visit Category

Visit Category	Total Visits
High	8
Medium	6
Low	6
Very low	10

```r
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

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

```r
ed_bonus2 <- ed9a %>%
  arrange(c("High", "Medium", "Low", "Very low"))

kable(ed_bonus2,
      caption = "ED Visit Utilization by Visit Category",
      col.names = c("Visit Category", "Total Visits"))
```