Midterm: Data Science

CS3072 Spring 2023

Aicha Sidiya - S20106146

9 April 2023

Packages

```
library(tidyverse)
```

Data

```
people <- read_csv("data/people.csv")
pitching <- read_csv("data/pitching.csv")
salaries <- read_csv("data/salaries.csv")
teams <- read_csv("data/teams.csv")
```

Exercise 1

1. Using group_by(), find the top 3 players in pitching dataset who got the maximum total number of earned_runs for the years 2014 through 2019.

```
pitching %>% filter (year_id >= 2014, year_id <= 2019) %>%
  group_by(player_id, year_id) %>%
  summarise(total_earned_runs = sum(earned_runs))%>%
  arrange(desc(total_earned_runs)) %>%
  head(3)
```

```
## `summarise()` has grouped output by 'player_id'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3 x 3
## # Groups:   player_id [3]
##   player_id year_id total_earned_runs
##   <chr>       <dbl>             <dbl>
## 1 giolilu01    2018               118
## 2 samarje01    2015               118
## 3 shielja02    2016               118
```

Exercise 2

2. Join the people data to the salaries data and mutate() a new variable with each player's approximate age. Call this dataset player_income_age. This dataset will have 26428 rows.

```
player_income_age <- inner_join(people,salaries,by="player_id") %>%
  mutate(age = year_id - birth_year)
```

```
## Warning in inner_join(people, salaries, by = "player_id"): Each row in `x` is expected to match at m
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

Exercise 3

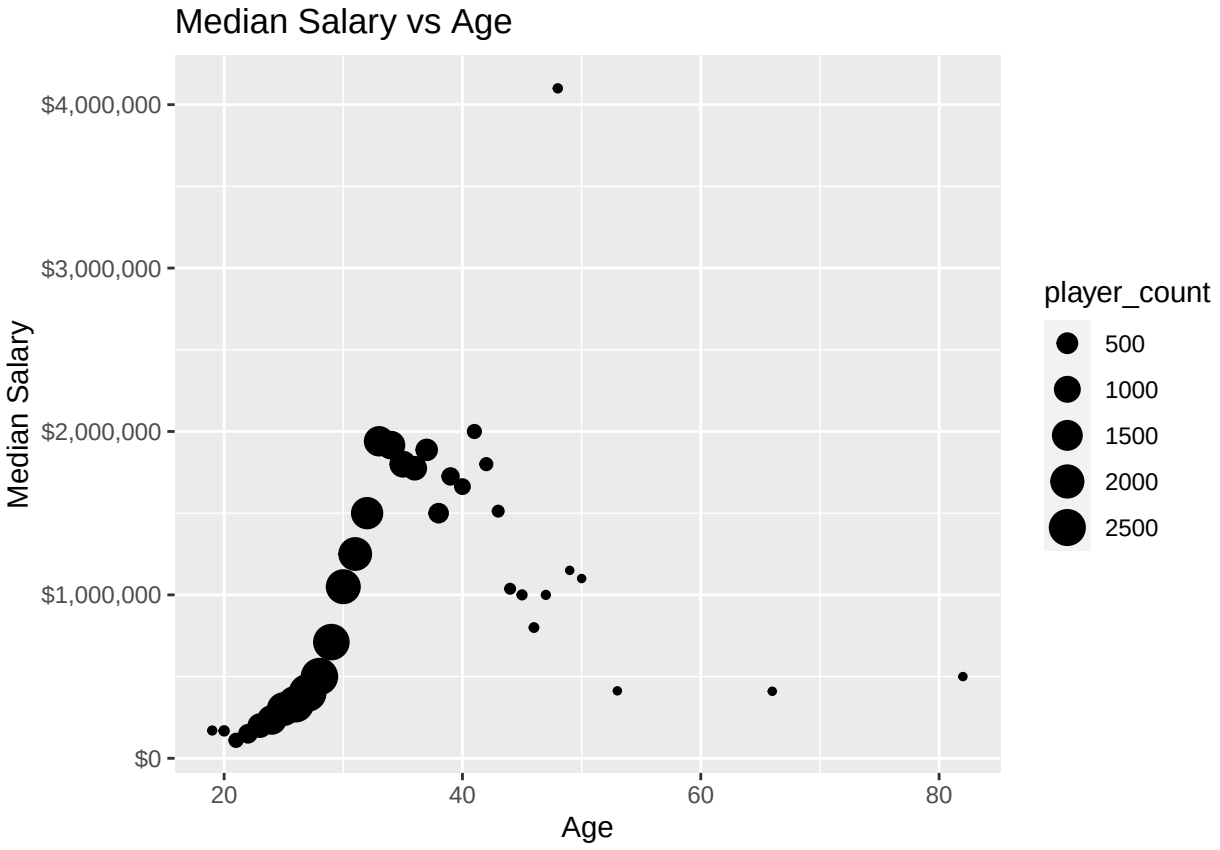
3. Based on the player_income_age dataset, create a new dataset called player_stats_by_age with the median_salary and count of players for all possible ages using group_by() and summarize(). The new dataset will have 35 rows.

```
player_stats_by_age <- player_income_age %>%
  group_by(age) %>%
  summarise(median_salary = median(salary), player_count = n())
```

Exercise 4

4. Construct a plot of median salary versus age with points sized by how many pitchers are at that age. Describe what you observe.

```
ggplot(data=player_stats_by_age,
       mapping=aes(x=age, y=median_salary)) +
  geom_point(mapping = aes(size = player_count)) +
  scale_y_continuous(labels = scales :: dollar_format())+
  labs(x = "Age", y = "Median Salary",
       title = "Median Salary vs Age")
```



Exercise 5

5. Use the teams dataset to find each team's win percentage for the years 2011 through 2016. Save the result as a tibble named team_stats. This tibble should have three columns (year_id, team_id, and win_pct) and 180 rows. Steps:
6. Filter the data for the given years
7. Add a new variable (win_pct)
8. Select only the three columns
9. Save the data in a new dataframe (team_stats)

```
team_stats <- teams %>%
  filter(year_id >= 2011, year_id <= 2016) %>%
  mutate(win_pct = (wins / games)) %>%
  select(year_id, team_id, win_pct)
```

Exercise 6

6. Use an appropriate join function to add salary information to players in the pitching data. You should only include observations that appear in both pitching and salaries. Then, using a group_by() paired with a summarize(), create a new column giving the total amount of money (sum()) spent on pitching salaries by each team in each year. Save the result as a tibble named team_spending. This dataset should have three columns (year_id, team_id, and pitching_salaries) and 918 rows. Steps:

7. Join pitching and salaries by player, year, and team ().
8. Group by year and team
9. Create a new variable (pitching_Salaries) using summarize
10. Save the data in new dataframe (team_spending) Note: when you join two datasets based on more than one variable, the syntax is: inner_join(dataframe1, dataframe2, by = c("var1", "var2", "var3"))

```
team_spending <- inner_join(pitching, salaries,
                            by = c("player_id",
                                   "year_id",
                                   "team_id")) %>%
  group_by(year_id, team_id) %>%
  summarise(pitching_salaries = sum(salary))
```

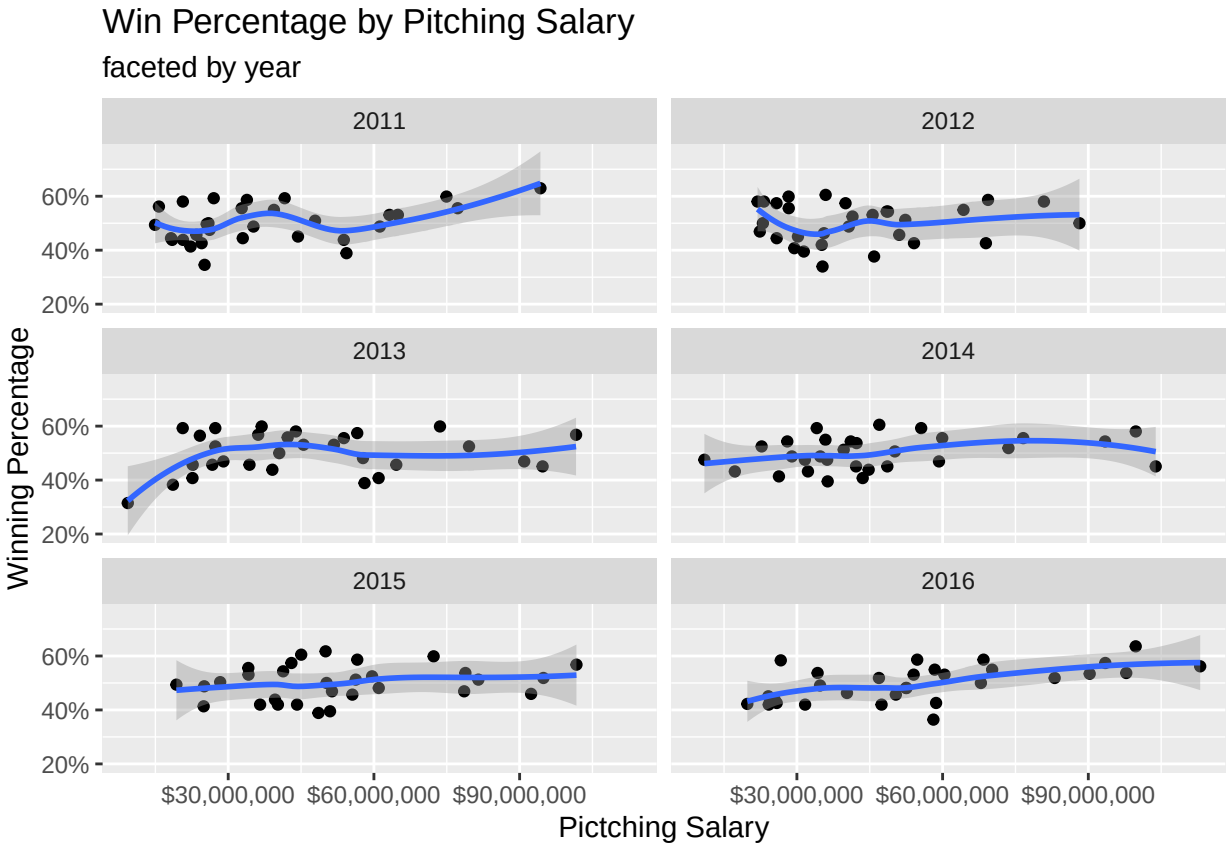
```
## `summarise()` has grouped output by 'year_id'. You can override using the
## `.groups` argument.
```

Exercise 7

7. Use an appropriate function to join team_stats and team_spending and use faceting to create subplots of win percentage versus pitching spending for each year. Steps:
8. Join team_stats and team_spending by year and team
9. Provide data visualization using geom_point and geom_smooth

```
inner_join(team_stats, team_spending,
           by = c("year_id", "team_id")) %>%
  ggplot(mapping=aes(x=pitching_salaries, y=win_pct)) +
  geom_point() + geom_smooth() +
  facet_wrap(.~year_id, nrow = 3)+
  scale_x_continuous(labels = scales :: dollar_format())+
  scale_y_continuous(labels = scales :: percent)+
  labs(title = "Win Percentage by Pitching Salary",
       subtitle = "faceted by year",
       x="Pictching Salary",
       y="Winning Percentage")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Exercise 8

8. Find the pitcher with the most strikeouts (strike_outs in the pitching dataset) in each year, by league, from 1901 to the most recent year in the dataset. Save the result as a tibble named top_so_by_year and glimpse() the dataset. Your final dataset should have 243 rows (there are some ties) and the columns year_id, league_id,first_name, and last_name, where the name information is from the people dataset. Steps:
9. Filter the pitching data from 1901 to the most recent year
10. Join the filtered data with people dataset by player
11. Group the joined data by year and league
12. Filter the data to have only the max strike_outs
13. Save the data in ne dataframe (top_so_by_year) year_id, league_id,first_name, and last_name

```
top_so_by_year <- pitching %>%
  filter(year_id >= 1901) %>%
  inner_join(y = people, by = "player_id") %>%
  group_by(year_id, league_id) %>%
  slice_max(strike_outs, n = 1) %>%
  select(year_id, league_id, first_name, last_name) %>%
  glimpse()
```

```
## Rows: 243
## Columns: 4
## Groups: year_id, league_id [240]
```

```
## $ year_id    <dbl> 1901, 1901, 1902, 1902, 1903, 1903, 1904, 1904, 1905, 1905,~
## $ league_id  <chr> "AL", "NL", "AL", "NL", "AL", "NL", "AL", "NL", "AL", "NL",~
## $ first_name <chr> "Cy", "Noodles", "Rube", "Vic", "Rube", "Christy", "Rube", ~
## $ last_name  <chr> "Young", "Hahn", "Waddell", "Willis", "Waddell", "Mathewson~
```