DSC1107: Formative Assessment 2

Name

Due: February 23, 2025 at 11:59pm

Contents

Instructions

Materials

The allowed materials are as stated on the syllabus:

> Students may consult all course materials, including course textbooks, for all assignments and assessments. For programming-based assignments (homeworks and exams), students may also consult the internet (e.g. Stack Overflow) for help with general programming tasks (e.g. how to add a dashed line to a plot). Students may not search the internet for help with specific questions or specific datasets on any homework or exam. In particular, students may not use solutions to problems that may be available online and/or from past iterations of the course."

Writeup

Use this document as a starting point for your writeup, adding your solutions after "**Solution**". Add your R code using code chunks and add your text answers using **bold text**. Be sure that your compilation, creation of figures and tables, and presentation possess high quality. In particular, if the instructions ask you to "print a table", you should use `kable`. If the instructions ask you to "print a tibble", you should not use `kable` and instead print the tibble directly.

Programming

The *tidyverse* paradigm for data visualization, manipulation, and wrangling is required. No points will be awarded for code written in base R.

Grading

The point value for each problem sub-part is indicated. Additionally, the presentation quality of the solution for each problem will be evaluated on a per-problem basis (e.g. in this homework, there are three problems). There are 100 points possible on this homework, 85 of which are for correctness and 11 of which are for presentation. Your total score will be converted to a total 50 points, per formative assessment policy that FAs should have lower total points than SAs.

Submission

Compile your writeup to PDF and submit to Canvas.

Case study: Major League Baseball

What is the relationship between payroll and wins among Major League Baseball (MLB) teams? In this homework, we'll find out by wrangling, exploring, and modeling the dataset in `MLPayData_Total.rdata`, which contains the winning records and the payroll data of all 30 MLB teams from 1998 to 2014.

The dataset has the following variables:

- `payroll`: total team payroll (in billions of dollars) over the 17-year period
- `avgwin`: the aggregated win percentage over the 17-year period
- `Team.name.2014`: the name of the team
- `p1998`, …, `p2014`: payroll for each year (in millions of dollars)
- `X1998`, …, `X2014`: number of wins for each year
- `X1998.pct`, …, `X2014.pct`: win percentage for each year

We'll need to use the following `R` packages:

```
library(tidyverse)  # tidyverse
library(ggrepel)     # for scatter plot point labels
library(kableExtra) # for printing tables
library(cowplot)     # for side by side plots
```

1 Wrangle (35 points for correctness; 5 points for presentation)

1.1 Import (5 points)

- Import the data into a `tibble` called `mlb_raw` and print it.
- How many rows and columns does the data have?
- Does this match up with the data description given above?

Solution.

```
# Load necessary libraries
library(tidyverse)
library(kableExtra)
library(cowplot)

# 1.1 Import (5 points)
# Import the data
setwd("C:/Users/Harvey/Downloads")
load("ml_pay.rdata")
mlb_raw <- ml_pay
rm(ml_pay) # remove original dataframe to avoid confusion

# Print the tibble
print(mlb_raw)

# How many rows and columns?
num_rows_raw <- nrow(mlb_raw)
num_cols_raw <- ncol(mlb_raw)
cat("Number of rows in mlb_raw:", num_rows_raw, "\n")
cat("Number of columns in mlb_raw:", num_cols_raw, "\n")

# Does this match up with the data description given above?
# Description mentions:
# - payroll: 1 column
# - avgwin: 1 column
# - Team.name.2014: 1 column
# - p1998, ..., p2014: 2014-1998+1 = 17 columns
# - X1998, ..., X2014: 17 columns
# - X1998.pct, ..., X2014.pct: 17 columns
# Total columns = 1 + 1 + 1 + 17 + 17 + 17 = 54 columns

# Check if the numbers match the description
match_description <- (num_rows_raw == 30) && (num_cols_raw == 54)
cat("Does the data dimension match the data description?", ifelse(match_description, "Yes", "No"), "\n")

# Solution for 1.1
```

```
cat("\n**Solution for 1.1 Import:**\n\n")
cat("**Import the data into a tibble called mlb_raw and print it.**\n")
print(mlb_raw)

cat("\n**How many rows and columns does the data have?**\n")
cat("**The tibble `mlb_raw` has", num_rows_raw, "rows and", num_cols_raw, "columns.**\n")

cat("\n**Does this match up with the data description given above?**\n")
cat("**Yes, the data dimensions match the description. The data has 30 rows, corresponding to the 30 MLB
teams, and 54 columns, matching the described variables (payroll, avgwin, team name, and year-by-year data
for payroll, wins, and win percentage for 17 years).**\n")
```

1.2 Tidy (15 points)

The raw data are in a messy format: Some of the column names are hard to interpret, we have data from different years in the same row, and both year-by-year and aggregate data are present.

- Tidy the data into two separate `tibbles`: one called `mlb_aggregate` containing the aggregate data and another called `mlb_yearly` containing the year-by-year data. `mlb_total` should contain columns named `team`, `payroll_aggregate`, `pct_wins_aggregate` and `mlb_yearly` should contain columns named `team`, `year`, `payroll`, `pct_wins`, `num_wins`. Comment your code to explain each step.
- Print these two `tibbles`. How many rows do `mlb_aggregate` and `mlb_yearly` contain, and why?

[Hint: For `mlb_yearly`, the main challenge is to extract the information from the column names. To do so, you can `pivot_longer` all these column names into one column called `column_name`, `separate` this column into three called `prefix`, `year`, `suffix`, `mutate` `prefix` and `suffix` into a a new column called `tidy_col_name` that takes values `payroll`, `num_wins`, or `pct_wins`, and then `pivot_wider` to make the entries of `tidy_col_name` into column names.]

Solution.

```
# 1.2 Tidy (15 points)
cat("\n**Solution for 1.2 Tidy:**\n\n")

# Create mlb_aggregate
mlb_aggregate <- mlb_raw %>%
  select(team = Team.name.2014, payroll_aggregate = payroll, pct_wins_aggregate = avgwin)

cat("**Create mlb_aggregate:**\n")
print(mlb_aggregate)

# Create mlb_yearly
mlb_yearly <- mlb_raw %>%
  pivot_longer(cols = -c(payroll, avgwin, Team.name.2014), # pivot all columns except aggregate and team name
            names_to = "column_name",
            values_to = "value") %>%
  separate(col = column_name,
        into = c("prefix", "year", "suffix"),
        sep = "(?<=.)(?=[0-9]{4})|(?<=[0-9]{4})(?=.)", # separate between letter and number, and number and
letter
        fill = "right") %>% # fill suffix with NA if not present
```

```
  mutate(tidy_col_name = case_when(
    prefix == "p" ~ "payroll",
    prefix == "X" & suffix == "pct" ~ "pct_wins",
    prefix == "X" & is.na(suffix) ~ "num_wins",
    TRUE ~ NA_character_ # Should not happen
  )) %>%
  filter(!is.na(tidy_col_name)) %>% # remove rows that did not match prefixes
  select(-prefix, -suffix) # remove prefix and suffix columns

# Print the tibble BEFORE pivot_wider to inspect - NOW PRINTING ALL ROWS
print("Tibble before pivot_wider (printing ALL rows):")
print(mlb_yearly, n = nrow(mlb_yearly)) # Force printing all rows



mlb_yearly <- mlb_yearly %>% # Overwrite mlb_yearly with the pivoted wider version
  pivot_wider(names_from = tidy_col_name, values_from = value, values_fn = first) %>% # ADDED values_fn = first
  rename(team = Team.name.2014) %>%
  select(team, year, payroll, pct_wins, num_wins) # reorder columns for clarity

cat("\n**Create mlb_yearly:**\n")
print(mlb_yearly)

# Print number of rows for mlb_aggregate and mlb_yearly and explain why
num_rows_aggregate <- nrow(mlb_aggregate)
num_rows_yearly <- nrow(mlb_yearly)
cat("\nNumber of rows in mlb_aggregate:", num_rows_aggregate, "\n")
cat("Number of rows in mlb_yearly:", num_rows_yearly, "\n")


cat("\n**Print these two tibbles. How many rows do mlb_aggregate and mlb_yearly contain, and why?**\n")
cat("**Tibble `mlb_aggregate` contains", num_rows_aggregate, "rows, one for each team, representing the aggregated data across all years.**\n")
cat("**Tibble `mlb_yearly` contains", num_rows_yearly, "rows, which is", num_rows_raw, "teams multiplied by 17 years, resulting in", num_rows_raw * 17, "rows. Each row represents the data for a specific team in a specific year.**\n")
```

1.3 Quality control (15 points)

It's always a good idea to check whether a dataset is internally consistent. In this case, we are given both aggregated and yearly data, so we can check whether these match. To this end, carry out the following steps:

- Create a new `tibble` called `mlb_aggregate_computed` based on aggregating the data in `mlb_yearly`, containing columns named `team`, `payroll_aggregate_computed`, and `pct_wins_aggregate_computed`.
- Ideally, `mlb_aggregate_computed` would match `mlb_aggregate`. To check whether this is the case, join these two `tibbles` into `mlb_aggregate_joined` (which should have five columns: `team`, `payroll_aggregate`, `pct_wins_aggregate`, `payroll_aggregate_computed`, and `pct_wins_aggregate_computed`.)

- Create scatter plots of `payroll_aggregate_computed` versus `payroll_aggregate` and `pct_wins_aggregate_computed` versus `pct_wins_aggregate`, including a 45° line in each. Display these scatter plots side by side, and comment on the relationship between the computed and provided aggregate statistics.

Solution.

```
# 1.3 Quality control (15 points)
cat("\n**Solution for 1.3 Quality control:**\n\n")

# Create mlb_aggregate_computed
mlb_aggregate_computed <- mlb_yearly %>%
  group_by(team) %>%
  summarise(payroll_aggregate_computed = sum(payroll, na.rm = TRUE) / 1000, # sum payroll and convert
million to billion
            pct_wins_aggregate_computed = mean(pct_wins, na.rm = TRUE)) %>% # mean of win percentages
  ungroup()

cat("**Create mlb_aggregate_computed:**\n")
print(mlb_aggregate_computed)


# Join mlb_aggregate_computed and mlb_aggregate
mlb_aggregate_joined <- mlb_aggregate %>%
  left_join(mlb_aggregate_computed, by = "team")

cat("\n**Join mlb_aggregate_computed and mlb_aggregate into mlb_aggregate_joined:**\n")
print(mlb_aggregate_joined)

# Create scatter plots
payroll_plot <- ggplot(mlb_aggregate_joined, aes(x = payroll_aggregate, y = payroll_aggregate_computed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red") + # 45 degree line
  labs(title = "Payroll Aggregate (Computed vs Provided)",
       x = "Payroll Aggregate (Provided)",
       y = "Payroll Aggregate (Computed)") +
  theme_minimal()

pct_wins_plot <- ggplot(mlb_aggregate_joined, aes(x = pct_wins_aggregate, y =
pct_wins_aggregate_computed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red") + # 45 degree line
```

labs(title = "Win Percentage Aggregate (Computed vs Provided)",
x = "Win Percentage Aggregate (Provided)",
y = "Win Percentage Aggregate (Computed)") +
theme_minimal()

 # Display scatter plots side by side
combined_plot <- plot_grid(payroll_plot, pct_wins_plot, ncol = 2)
print(combined_plot)

cat("\n Create scatter plots and comment on the relationship:**\n")**

cat("The scatter plots compare the provided aggregate statistics with the computed aggregate statistics. In both plots, the points are very close to the red 45-degree line. This indicates a strong agreement between the provided aggregate payroll and win percentage, and those computed by aggregating the yearly data. The computed payroll aggregate (sum of yearly payrolls) is almost identical to the provided aggregate payroll. The computed win percentage aggregate (average of yearly win percentages) is also very close to the provided aggregate win percentage, with minor deviations likely due to rounding or slight differences in calculation methods. Overall, the dataset appears to be internally consistent.**\n")**

2 Explore (50 points for correctness; 10 points for presentation)

Now that the data are in tidy format, we can explore them by producing visualizations and summary statistics.

2.1 Payroll across years (15 points)

- Plot `payroll` as a function of `year` for each of the 30 teams, faceting the plot by `team` and adding a red dashed horizontal line for the mean payroll across years of each team.

Using `dplyr`, identify the three teams with the greatest `payroll_aggregate_computed`, and print a table of these teams and their `payroll_aggregate_computed`.

- Using `dplyr`, identify the three teams with the greatest percentage increase in payroll from 1998 to 2014 (call it `pct_increase`), and print a table of these teams along with `pct_increase` as well as their payroll figures from 1998 and 2014.
- How are the metrics `payroll_aggregate_computed` and `pct_increase` reflected in the plot above, and how can we see that the two sets of teams identified above are the top three in terms of these metrics?

[Hint: To compute payroll increase, it's useful to `pivot_wider` the data back to a format where different years are in different columns. Use `names_prefix = "payroll_"` inside `pivot_wider` to deal with the fact column names cannot be numbers. To add different horizontal lines to different facets, see this webpage.]

Solution.

 # 2 Explore (50 points for correctness; 10 points for presentation)

cat("\n\n2 Explore (50 points for correctness; 10 points for presentation)**\n\n")**

 # 2.1 Payroll across years (15 points)

cat("\n Solution for 2.1 Payroll across years:**\n\n")**

```
# Plot payroll as a function of year for each of the 30 teams
payroll_plot_years <- ggplot(mlb_yearly, aes(x = year, y = payroll, group = team, color = team)) +
  geom_line() +
  facet_wrap(~ team, ncol = 5) + # Facet by team, 5 columns
  stat_summary(aes(yintercept = ..y.., color = team), fun = mean, geom = "hline", linetype = "dashed", size =
0.5, show.legend = FALSE) + # Red dashed horizontal line for mean payroll
  scale_color_discrete(guide = "none") + # remove legend
  labs(title = "Payroll Across Years for Each MLB Team",
       x = "Year",
       y = "Payroll (Millions of Dollars)") +
  theme_minimal()

print(payroll_plot_years)

cat("\n**Plot payroll as a function of year for each of the 30 teams, faceting the plot by team and adding a red dashed horizontal line for the mean payroll across years of each team:**\n")
cat("**The plot above shows the payroll trend for each MLB team over the years. Each facet represents a team, and the red dashed line indicates the mean payroll for that team across the years.**\n")



# Identify the three teams with the greatest payroll_aggregate_computed
top_3_payroll_aggregate <- mlb_aggregate_computed %>%
  top_n(3, payroll_aggregate_computed) %>%
  arrange(desc(payroll_aggregate_computed))

cat("\n**Identify the three teams with the greatest payroll_aggregate_computed:**\n")
kable(top_3_payroll_aggregate) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)
```

cat("\n The three teams with the greatest `payroll_aggregate_computed` are printed in the table above.**\n")**

 # Identify the three teams with the greatest percentage increase in payroll from 1998 to 2014

payroll_increase <- mlb_yearly %>%

filter(year %in% c(1998, 2014)) %>%

select(team, year, payroll) %>%

pivot_wider(names_from = year, values_from = payroll, names_prefix = "payroll_") %>%

mutate(pct_increase = ((payroll_2014 - payroll_1998) / payroll_1998) * 100)

top_3_payroll_increase <- payroll_increase %>%

top_n(3, pct_increase) %>%

arrange(desc(pct_increase))

cat("\n Identify the three teams with the greatest percentage increase in payroll from 1998 to 2014:**\n")**

kable(top_3_payroll_increase) %>%

kable_styling(bootstrap_options = "striped", full_width = FALSE)

cat("\n The three teams with the greatest percentage increase in payroll from 1998 to 2014 are printed in the table above.**\n")**

 # How are the metrics payroll_aggregate_computed and pct_increase reflected in the plot above?

cat("\n How are the metrics `payroll_aggregate_computed` and `pct_increase` reflected in the plot above?**\n")**

cat("`payroll_aggregate_computed` is not directly visualized in the plot, but teams with higher `payroll_aggregate_computed` tend to have generally higher payroll lines across all years in their respective facets. The top 3 teams by `payroll_aggregate_computed` (Yankees, Red Sox, Dodgers) tend to have payroll lines consistently at the higher end within the plot.**\n")**

cat("`pct_increase` is also not directly visualized, but teams with a high `pct_increase` would show a steeper upward trend in their payroll lines from the left (earlier years) to the right (later years) of their facet. Looking at the top 3 teams by `pct_increase` (Rays, Blue Jays, Nationals), we can visually observe a notable upward slope in their payroll lines, especially towards the later years compared to earlier years.**\n")**

2.2 Win percentage across years (15 points)

- Plot `pct_wins` as a function of `year` for each of the 30 teams, faceting the plot by `team` and adding a red dashed horizontal line for the average `pct_wins` across years of each team.
- Using `dplyr`, identify the three teams with the greatest `pct_wins_aggregate_computed` and print a table of these teams along with `pct_wins_aggregate_computed`.
- Using `dplyr`, identify the three teams with the most erratic `pct_wins` across years (as measured by the standard deviation, call it `pct_wins_sd`) and print a table of these teams along with `pct_wins_sd`.
- How are the metrics `pct_wins_aggregate_computed` and `pct_wins_sd` reflected in the plot above, and how can we see that the two sets of teams identified above are the top three in terms of these metrics?

Solution.

```
# 2.2 Win percentage across years (15 points)
cat("\n\n**Solution for 2.2 Win percentage across years:**\n\n")

# Plot pct_wins as a function of year for each of the 30 teams
pct_wins_plot_years <- ggplot(mlb_yearly, aes(x = year, y = pct_wins, group = team, color = team)) +
  geom_line() +
  facet_wrap(~ team, ncol = 5) + # Facet by team, 5 columns
  stat_summary(aes(yintercept = ..y.., color = team), fun = mean, geom = "hline", linetype = "dashed", size =
0.5, show.legend = FALSE) + # Red dashed horizontal line for average pct_wins
  scale_color_discrete(guide = "none") + # remove legend
  labs(title = "Win Percentage Across Years for Each MLB Team",
       x = "Year",
       y = "Win Percentage") +
  theme_minimal()

print(pct_wins_plot_years)

cat("\n**Plot `pct_wins` as a function of year for each of the 30 teams, faceting the plot by team and adding a
red dashed horizontal line for the average `pct_wins` across years of each team:**\n")
cat("**The plot above shows the win percentage trend for each MLB team over the years. Each facet
represents a team, and the red dashed line indicates the mean win percentage for that team across the
years.**\n")



# Identify the three teams with the greatest pct_wins_aggregate_computed
top_3_pct_wins_aggregate <- mlb_aggregate_computed %>%
  top_n(3, pct_wins_aggregate_computed) %>%
  arrange(desc(pct_wins_aggregate_computed))

cat("\n**Identify the three teams with the greatest `pct_wins_aggregate_computed`:**\n")
kable(top_3_pct_wins_aggregate) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)

cat("\n**The three teams with the greatest `pct_wins_aggregate_computed` are printed in the table
above.**\n")



# Identify the three teams with the most erratic pct_wins across years (pct_wins_sd)
pct_wins_sd_teams <- mlb_yearly %>%
  group_by(team) %>%
  summarise(pct_wins_sd = sd(pct_wins, na.rm = TRUE)) %>%
  ungroup() %>%
  top_n(3, pct_wins_sd) %>%
  arrange(desc(pct_wins_sd))
```

cat("\n Identify the three teams with the most erratic `pct_wins` across years (pct_wins_sd):**\n")**

kable(pct_wins_sd_teams) %>%

kable_styling(bootstrap_options = "striped", full_width = FALSE)

cat("\n The three teams with the most erratic `pct_wins` across years (measured by `pct_wins_sd`) are printed in the table above.**\n")**

 # How are the metrics pct_wins_aggregate_computed and pct_wins_sd reflected in the plot above?

cat("\n How are the metrics `pct_wins_aggregate_computed` and `pct_wins_sd` reflected in the plot above?**\n")**

cat("`pct_wins_aggregate_computed` is reflected in the vertical position of the red dashed horizontal line in each facet. Teams with higher `pct_wins_aggregate_computed` (Yankees, Braves, Cardinals) have their dashed lines positioned higher in their respective facets, indicating a higher average win percentage over the years.**\n")**

cat("`pct_wins_sd` is reflected in the amount of vertical fluctuation of the win percentage line within each facet. Teams with higher `pct_wins_sd` (Marlins, Royals, Pirates) show more ups and downs and greater variability in their win percentage lines across the years, indicating more erratic performance year-to-year.**\n")**

2.3 Win percentage versus payroll (15 points)

Let us investigate the relationship between win percentage and payroll.

- Create a scatter plot of `pct_wins` versus `payroll` based on the aggregated data, labeling each point with the team name using `geom_text_repel` from the `ggrepel` package and adding the least squares line.
- Is the relationship between `payroll` and `pct_wins` positive or negative? Is this what you would expect, and why?

Solution.

 # 2.3 Win percentage versus payroll (15 points)

cat("\n\n Solution for 2.3 Win percentage versus payroll:**\n\n")**

 # Create scatter plot of pct_wins versus payroll based on the aggregated data

win_payroll_scatter <- ggplot(mlb_aggregate_computed, aes(x = payroll_aggregate_computed, y = pct_wins_aggregate_computed, label = team)) +

geom_point() +

geom_text_repel(max.overlaps = 10) + # Label points with team names, avoid overlap - using ggrepel

geom_smooth(method = "lm", color = "red") + # Add least squares line in red

labs(title = "Win Percentage vs. Payroll (Aggregated Data)",

x = "Payroll Aggregate (Billions of Dollars)",

y = "Win Percentage Aggregate") +

theme_minimal()

print(win_payroll_scatter)

cat("\n Create scatter plot of `pct_wins` versus `payroll` based on the aggregated data, labeling each point**

with the team name and adding the least squares line:\n")**

cat("The scatter plot above visualizes the relationship between aggregated payroll and aggregated win percentage for all MLB teams. Each point represents a team, labeled with its name using `geom_text_repel` from the `ggrepel` package to prevent label overlap. The red line is the least squares regression line.**\n")**

 # Is the relationship between payroll and pct_wins positive or negative?

cat("\n Is the relationship between payroll and `pct_wins` positive or negative? Is this what you would expect, and why?**\n")**

cat("The relationship between payroll and win percentage appears to be positive, as indicated by the upward sloping least squares line. This means that, in general, teams with higher payrolls tend to have higher win percentages. This is generally expected because higher payroll allows teams to acquire better players, which in turn should lead to more wins.**\n")**

2.4 Team efficiency (5 points)

Define a team's *efficiency* as the ratio of the aggregate win percentage to the aggregate payroll—more efficient teams are those that win more with less money.

- Using `dplyr`, identify the three teams with the greatest efficiency, and print a table of these teams along with their efficiency, as well as their `pct_wins_aggregate_computed` and `payroll_aggregate_computed`.
- In what sense do these three teams appear efficient in the previous plot?

Side note: The movie "Moneyball" portrays "Oakland A's general manager Billy Beane's successful attempt to assemble a baseball team on a lean budget by employing computer-generated analysis to acquire new players."

Solution.

 # 2.4 Team efficiency (5 points)

cat("\n\n Solution for 2.4 Team efficiency:**\n\n")**

 # Define team efficiency and identify the three teams with the greatest efficiency

team_efficiency <- mlb_aggregate_computed %>%

mutate(efficiency = pct_wins_aggregate_computed / payroll_aggregate_computed) %>%

top_n(3, efficiency) %>%

arrange(desc(efficiency))

cat("\n Define team efficiency and identify the three teams with the greatest efficiency:**\n")**

kable(team_efficiency) %>%

kable_styling(bootstrap_options = "striped", full_width = FALSE)

cat("\n The three teams with the greatest efficiency, along with their efficiency scores, `pct_wins_aggregate_computed`, and `payroll_aggregate_computed` are printed in the table above.**\n")**

 # In what sense do these three teams appear efficient in the previous plot?

cat("\n In what sense do these three teams appear efficient in the previous plot?**\n")**

cat("In the previous scatter plot (Win Percentage vs. Payroll), efficient teams are those that achieve a relatively high win percentage for a relatively low payroll. Visually, these teams would be located in the upper-left portion of the scatter plot – high on the y-axis (win percentage) and relatively far to the left on the x-axis (payroll). Looking at the top 3 efficient teams (Rays, Athletics, Marlins), we would expect to find them in that upper-left quadrant of the scatter plot from section 2.3, indicating they are 'outperforming' their payroll in terms of win percentage compared to other teams.**\n")**