Data Wrangling

John Benedict A. Monfero

2025-02-16

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
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggrepel)    # for scatter plot point labels
library(kableExtra) # for printing tables
```

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

```
library(cowplot)    # for side by side plots
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

Import

• Import the data into a `tibble` called `mlb_raw` and print it.

```
# Load the dataset
load("ml_pay.rdata")

# Convert the dataset into a tibble
mlb_raw <- as_tibble(ml_pay)

# Print the tibble
print(mlb_raw)
```

```
## # A tibble: 30 × 54
##    payroll avgwin Team.name.2014 p1998 p1999 p2000 p2001 p2002 p2003 p2004 p2005
##      <dbl>  <dbl> <fct>          <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1   1.12   0.490 Arizona Diamo…  31.6  70.5  81.0  81.2 103.   80.6  70.2  63.0
##  2   1.38   0.553 Atlanta Braves  61.7  74.9  84.5  91.9  93.5 106.   88.5  85.1
##  3   1.16   0.454 Baltimore Ori…  71.9  72.2  81.4  72.4  60.5  73.9  51.2  74.6
##  4   1.97   0.549 Boston Red Sox  59.5  71.7  77.9 110.  108.   99.9 125.  121.
##  5   1.46   0.474 Chicago Cubs    49.8  42.1  60.5  64.0  75.7  79.9  91.1  87.2
##  6   1.32   0.511 Chicago White…  35.2  24.5  31.1  62.4  57.1  51.0  65.2  75.2
##  7   1.02   0.486 Cincinnati Re…  20.7  73.3  46.9  45.2  45.1  59.4  43.1  59.7
##  8   0.999  0.496 Cleveland Ind…  59.5  54.4  75.9  92.0  78.9  48.6  34.6  41.8
##  9   1.03   0.463 Colorado Rock…  47.7  55.4  61.1  71.1  56.9  67.2  64.6  47.8
## 10   1.43   0.482 Detroit Tigers  19.2  35.0  58.3  49.8  55.0  49.2  46.4  69.0
## # ℹ 20 more rows
## # ℹ 43 more variables: p2006 <dbl>, p2007 <dbl>, p2008 <dbl>, p2009 <dbl>,
## #   p2010 <dbl>, p2011 <dbl>, p2012 <dbl>, p2013 <dbl>, p2014 <dbl>,
## #   X2014 <int>, X2013 <int>, X2012 <int>, X2011 <int>, X2010 <int>,
## #   X2009 <int>, X2008 <int>, X2007 <int>, X2006 <int>, X2005 <int>,
## #   X2004 <int>, X2003 <int>, X2002 <int>, X2001 <int>, X2000 <int>,
## #   X1999 <int>, X1998 <int>, X2014.pct <dbl>, X2013.pct <dbl>, …
```

• How many rows and columns does the data have? • Does this match up with the data description given above?

```
# Get the dimensions of the dataset
dimensions <- dim(mlb_raw)

# Print the number of rows and columns
cat("Number of rows:", dimensions[1], "\n")
```

```
## Number of rows: 30
```

```
cat("Number of columns:", dimensions[2], "\n")
```

```
## Number of columns: 54
```

Solution: - The dataset contains rows and columns as provided above.

- The description of the case study above, states that the dataset should include *team payroll*, *average win percentage*, *team names*, and *yearly payroll* and *win records from 1998 to 2014*.
- For the next Chapter: we will verify whether these variables match the description by examining the column names in the next steps.

Tidy

> 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.

[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.]

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

mlb_yearly <- mlb_raw %>%
  select(Team.name.2014, starts_with("p"), starts_with("X")) %>%
  pivot_longer(cols = -Team.name.2014, names_to = "column_name", values_to = "value") %>%

  # Extract year using regex to capture 4-digit numbers
  mutate(year = str_extract(column_name, "\\d{4}")) %>%

  # Determine column type using manual checks
  mutate(tidy_col_name = case_when(
    str_starts(column_name, "p") ~ "payroll",
    str_ends(column_name, "pct") ~ "pct_wins",
    TRUE ~ "num_wins"
  )) %>%

  # Drop the original column_name and reshape
  select(Team.name.2014, year, tidy_col_name, value) %>%
  pivot_wider(names_from = tidy_col_name, values_from = value) %>%
  rename(team = Team.name.2014) %>%
  drop_na()
```

- Print these two tibbles. How many rows do mlb_aggregate and mlb_yearly contain, and why?

```
# Print mlb_aggregate
print(mlb_aggregate)
```

```
## # A tibble: 30 × 3
##    team                 payroll_aggregate pct_wins_aggregate
##    <fct>                            <dbl>              <dbl>
##  1 Arizona Diamondbacks             1.12               0.490
##  2 Atlanta Braves                   1.38               0.553
##  3 Baltimore Orioles                1.16               0.454
##  4 Boston Red Sox                   1.97               0.549
##  5 Chicago Cubs                     1.46               0.474
##  6 Chicago White Sox                1.32               0.511
##  7 Cincinnati Reds                  1.02               0.486
##  8 Cleveland Indians                0.999              0.496
##  9 Colorado Rockies                 1.03               0.463
## 10 Detroit Tigers                   1.43               0.482
## # ℹ 20 more rows
```

```
# Print mlb_yearly
print(mlb_yearly)
```

```
## # A tibble: 510 × 5
##    team                 year  payroll num_wins pct_wins
##    <fct>                <chr>   <dbl>    <dbl>    <dbl>
##  1 Arizona Diamondbacks 1998     31.6       65    0.399
##  2 Arizona Diamondbacks 1999     70.5      100    0.613
##  3 Arizona Diamondbacks 2000     81.0       85    0.525
##  4 Arizona Diamondbacks 2001     81.2       92    0.568
##  5 Arizona Diamondbacks 2002    103.        98    0.605
##  6 Arizona Diamondbacks 2003     80.6       84    0.519
##  7 Arizona Diamondbacks 2004     70.2       51    0.315
##  8 Arizona Diamondbacks 2005     63.0       77    0.475
##  9 Arizona Diamondbacks 2006     59.7       76    0.469
## 10 Arizona Diamondbacks 2007     52.1       90    0.552
## # ℹ 500 more rows
```

```
# Get the dimensions of both datasets
cat("Number of rows in mlb_aggregate:", nrow(mlb_aggregate), "\n")
```

```
## Number of rows in mlb_aggregate: 30
```

```
cat("Number of rows in mlb_yearly:", nrow(mlb_yearly), "\n")
```

```
## Number of rows in mlb_yearly: 510
```

Solution: - The `mlb_aggregate` tibble contains 30 rows, representing each MLB team. - The `mlb_yearly` tibble contains 510 rows, representing team-year observations. - The increase in rows in `mlb_yearly` occurs because we have multiple observations for each team across multiple years (from 1998 to 2014).

Quality Control

> 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` .

```
mlb_aggregate_computed <- mlb_yearly %>%
  group_by(team) %>%
  summarise(
    payroll_aggregate_computed = sum(payroll, na.rm = TRUE)/1000,
    #since the payroll_aggregate in the `mlb_aggregate` is in terms of billions; in mlb_yearly, each p##
## was in terms of millions, we will divide each computed by 1000

    pct_wins_aggregate_computed = mean(pct_wins, na.rm = TRUE)
  )
#print result
mlb_aggregate_computed
```

```
## # A tibble: 30 × 3
##    team                 payroll_aggregate_computed pct_wins_aggregate_computed
##    <fct>                                     <dbl>                       <dbl>
##  1 Arizona Diamondbacks                       1.22                       0.492
##  2 Atlanta Braves                             1.52                       0.563
##  3 Baltimore Orioles                          1.31                       0.457
##  4 Boston Red Sox                             2.10                       0.551
##  5 Chicago Cubs                               1.55                       0.475
##  6 Chicago White Sox                          1.38                       0.507
##  7 Cincinnati Reds                            1.12                       0.491
##  8 Cleveland Indians                          1.11                       0.505
##  9 Colorado Rockies                           1.13                       0.463
## 10 Detroit Tigers                             1.48                       0.474
## # ℹ 20 more rows
```

- 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`.)

```
# Join the computed and provided aggregate data
mlb_aggregate_joined <- mlb_aggregate %>%
  left_join(mlb_aggregate_computed, by = "team") %>%
  select(team, payroll_aggregate_computed, payroll_aggregate, pct_wins_aggregate_computed, pct_wins_aggr
egate)

# Print joined tibble
mlb_aggregate_joined
```

```
## # A tibble: 30 × 5
##    team          payroll_aggregate_co…¹ payroll_aggregate pct_wins_aggregate_c…²
##    <fct>                          <dbl>             <dbl>                  <dbl>
##  1 Arizona Diam…                   1.22              1.12                  0.492
##  2 Atlanta Brav…                   1.52              1.38                  0.563
##  3 Baltimore Or…                   1.31              1.16                  0.457
##  4 Boston Red S…                   2.10              1.97                  0.551
##  5 Chicago Cubs                    1.55              1.46                  0.475
##  6 Chicago Whit…                   1.38              1.32                  0.507
##  7 Cincinnati R…                   1.12              1.02                  0.491
##  8 Cleveland In…                   1.11              0.999                 0.505
##  9 Colorado Roc…                   1.13              1.03                  0.463
## 10 Detroit Tige…                   1.48              1.43                  0.474
## # ℹ 20 more rows
## # ℹ abbreviated names: ¹payroll_aggregate_computed,
## #   ²pct_wins_aggregate_computed
## # ℹ 1 more variable: pct_wins_aggregate <dbl>
```

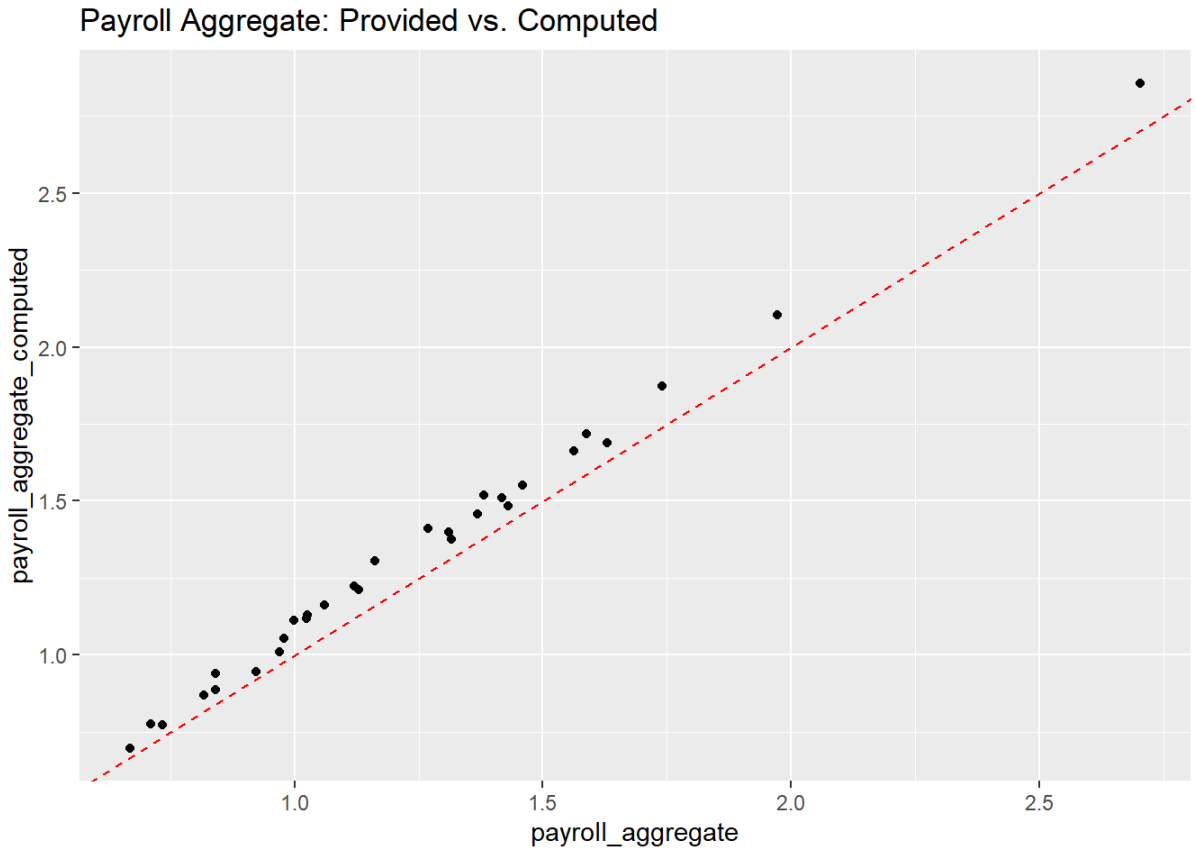
- Create scatter plots of `payroll_aggregate_computed` versus `payroll_aggregate` and `pct_wins_aggregate_computed` versus `pct_wins_aggregate`, including a `45◦` line in each. Display these scatter

plots side by side, and comment on the relationship between the computed and provided aggregate statistics.

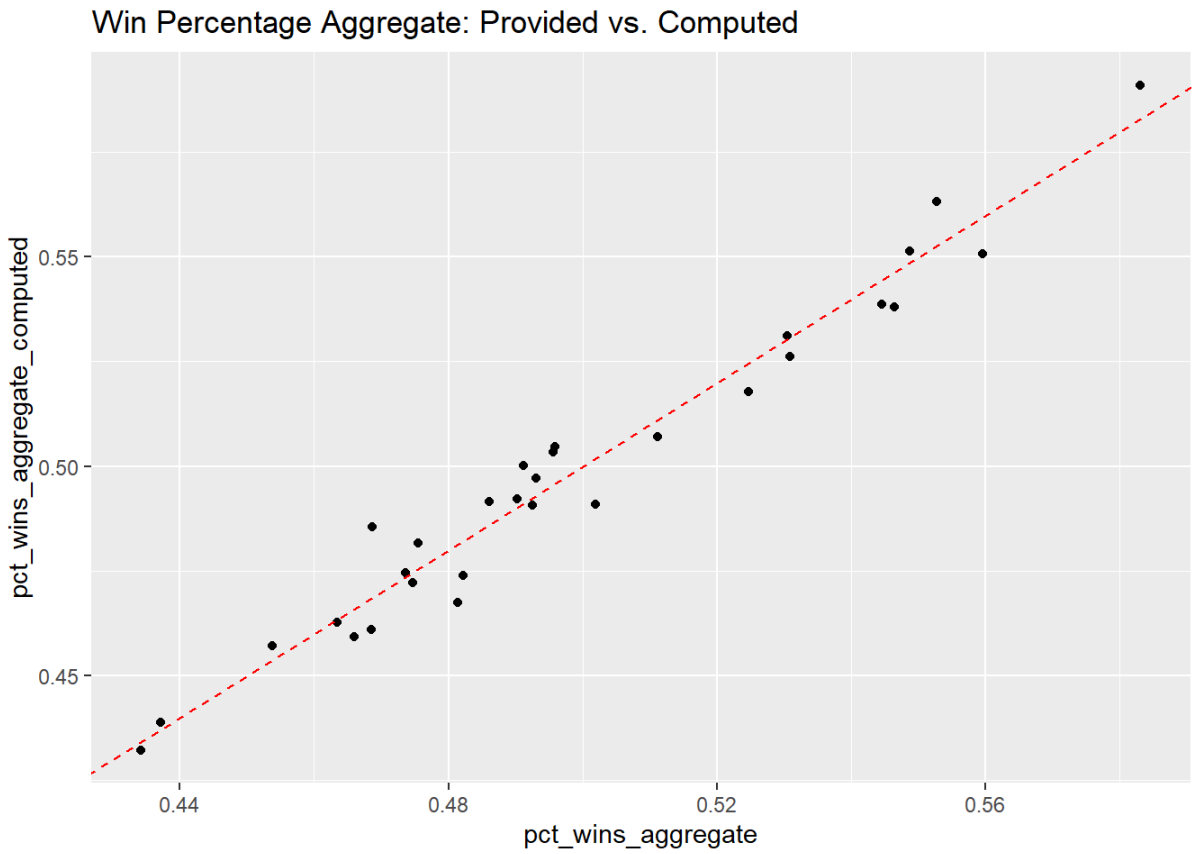
```
plot1 <- ggplot(mlb_aggregate_joined, aes(x = payroll_aggregate, y = payroll_aggregate_computed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  ggtitle("Payroll Aggregate: Provided vs. Computed")

plot2 <- ggplot(mlb_aggregate_joined, aes(x = pct_wins_aggregate, y = pct_wins_aggregate_computed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  ggtitle("Win Percentage Aggregate: Provided vs. Computed")

plot1
```




```
plot2
```



Solution:

- The `mlb_aggregate_computed` tibble was created by summing the payroll and averaging win percentage from `mlb_yearly`.
- The `mlb_aggregate_joined` tibble contains the original and computed aggregates for comparison.
- The scatter plots compare the provided vs. computed values, with a 45-degree line as a reference.
- If points align closely with the 45-degree line, the data is consistent. Deviations indicate discrepancies.

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

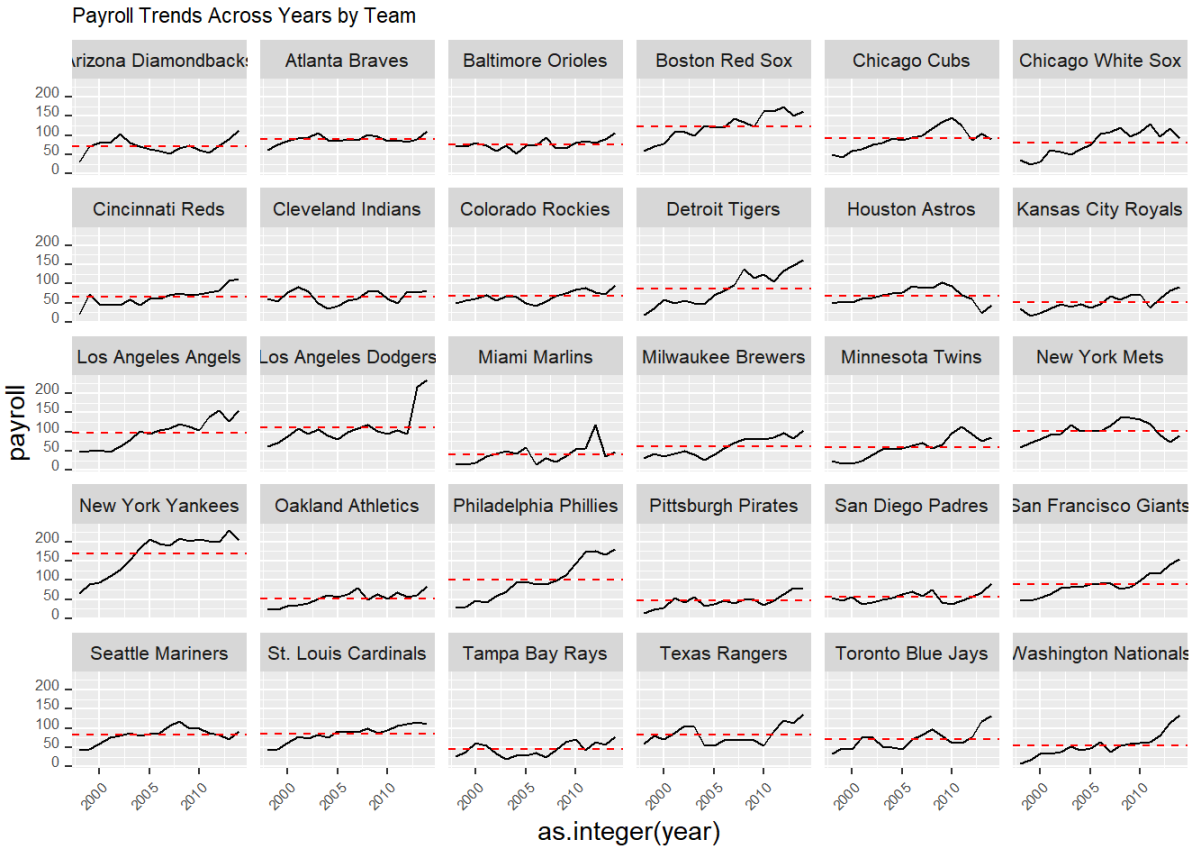
Exploration of the Case Study

Payroll across years

- 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.

```
plot_payroll <- mlb_yearly %>%
  ggplot(aes(x = as.integer(year), y = payroll, group = team)) +
  geom_line() +
  facet_wrap(~team) +
  geom_hline(data = mlb_yearly %>%
               group_by(team) %>%
               summarise(mean_payroll = mean(payroll, na.rm = TRUE)),
             aes(yintercept = mean_payroll), linetype = "dashed", color = "red") +
  ggtitle("Payroll Trends Across Years by Team") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 6),
        axis.text.y = element_text(angle = 0, vjust = 0, size = 6),
        strip.text = element_text(size = 8),
        plot.title = element_text(size = 8.5))

plot_payroll
```



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

```
# Identify top 3 teams with highest total payroll
mlb_top_payroll <- mlb_aggregate_computed %>%
  arrange(desc(payroll_aggregate_computed)) %>%
  head(3)

kable(mlb_top_payroll, caption = "Top 3 Teams by Payroll Aggregate Computed")
```

Top 3 Teams by Payroll Aggregate Computed

team	payroll_aggregate_computed	pct_wins_aggregate_computed
New York Yankees	2.857093	0.5909819
Boston Red Sox	2.103581	0.5512860
Los Angeles Dodgers	1.874194	0.5261364

- 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`.

[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]

```
# Compute percentage increase from 1998 to 2014
mlb_payroll_wide <- mlb_yearly %>%
  filter(year %in% c(1998, 2014)) %>%
  select(team, year, payroll) %>%
  pivot_wider(names_from = year, values_from = payroll, names_prefix = "payroll_")

mlb_pct_increase <- mlb_payroll_wide %>%

  mutate(pct_increase =
           (payroll_2014 - payroll_1998) / payroll_1998 * 100) %>%
  select(team, payroll_1998, payroll_2014, pct_increase) %>%
  arrange(desc(pct_increase)) %>%
  head(3)

kable(mlb_pct_increase, caption = "Top 3 Teams by Payroll Percentage Increase (1998-2014)")
```

Top 3 Teams by Payroll Percentage Increase (1998-2014)

team	payroll_1998	payroll_2014	pct_increase
Washington Nationals	8.3170	134.7044	1519.6277
Detroit Tigers	19.2375	162.2285	743.2932
Philadelphia Phillies	28.6225	180.0527	529.0601

- 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?

Solution:

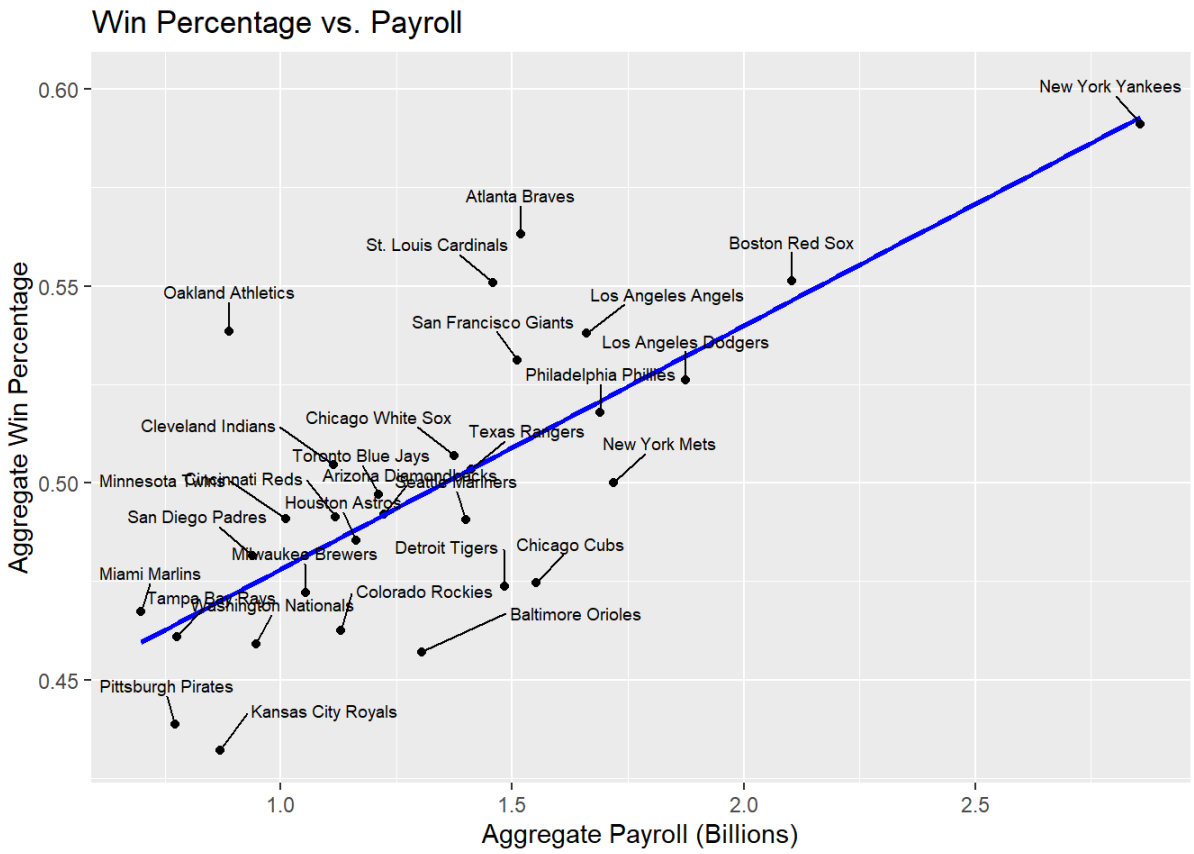
- The payroll trends for each team are plotted, with a red dashed line indicating the mean payroll across years.
- The top 3 teams by total payroll are identified and printed.
- The top 3 teams by payroll percentage increase from 1998 to 2014 are identified and printed.
- These metrics are reflected in the payroll trends plot, showing the top teams with increasing payroll trends over time.

```
# Scatter plot of pct_wins vs. payroll with team labels and regression line
plot3 <- ggplot(mlb_aggregate_joined, aes(x = payroll_aggregate_computed, y = pct_wins_aggregate_compute
d, label = team)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  geom_text_repel(size = 2.5, nudge_y = 0.01, direction = "x") +  # Adjust text position above points
  ggtitle("Win Percentage vs. Payroll") +
  xlab("Aggregate Payroll (Billions)") +
  ylab("Aggregate Win Percentage")

print(plot3)
```

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

```
## Warning: The following aesthetics were dropped during statistical transformation: label.
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
# Compute team efficiency (win percentage per payroll dollar)
mlb_efficiency <- mlb_aggregate_computed %>%
  mutate(efficiency = pct_wins_aggregate_computed / payroll_aggregate_computed) %>%
  arrange(desc(efficiency)) %>%
  select(team, efficiency, pct_wins_aggregate_computed, payroll_aggregate_computed)

# Identify the top three most efficient teams
top_efficient_teams <- mlb_efficiency %>% top_n(3, efficiency)

# Print the top efficient teams
kable(top_efficient_teams, caption = "Top 3 Most Efficient MLB Teams")
```

Top 3 Most Efficient MLB Teams

team	efficiency	pct_wins_aggregate_computed	payroll_aggregate_computed
Miami Marlins	0.6694182	0.4673161	0.6980929
Oakland Athletics	0.6067658	0.5385489	0.8875729
Tampa Bay Rays	0.5939731	0.4610341	0.7761869