FA2 Data wrangling

2024-02-26

```
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
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.4.4     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(tidyr)
library(ggrepel)
```

```
load("C:/Users/yohan gacasa/Downloads/ml_pay.RData")
mlb_data <- ml_pay
```

```
payroll_long <- mlb_data %>%
  pivot_longer(cols = starts_with("p"), names_to = "year", values_to = "payroll") %>%
  mutate(year = as.numeric(gsub("p", "", year)))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `year = as.numeric(gsub("p", "", year))`.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
wins_long <-  mlb_data %>%
  pivot_longer(cols = starts_with("X"), names_to = "year", values_to = "wins") %>%
  mutate(year = as.numeric(gsub("X", "", year)))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `year = as.numeric(gsub("X", "", year))`.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
win_percentage_long <-  mlb_data %>%
  pivot_longer(cols = starts_with("X"), names_to = "year", values_to = "win_percentage") %>%
  mutate(year = as.numeric(gsub("X|\\.pct", "", year)))

tidy_data <- payroll_long %>%
  left_join(wins_long, by = c("Team.name.2014", "year")) %>%
  left_join(win_percentage_long, by = c("Team.name.2014", "year"))


head(tidy_data)
```

```
## # A tibble: 6 x 78
##   avgwin.x Team.name.2014  X2014 X2013 X2012 X2011 X2010 X2009 X2008 X2007 X2006
##      <dbl> <fct>           <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## 2    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## 3    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## 4    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## 5    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## 6    0.490 Arizona Diamon~    59    81    81    94    65    70    82    90    76
## # i 67 more variables: X2005 <int>, X2004 <int>, X2003 <int>, X2002 <int>,
## #   X2001 <int>, X2000 <int>, X1999 <int>, X1998 <int>, X2014.pct <dbl>,
## #   X2013.pct <dbl>, X2012.pct <dbl>, X2011.pct <dbl>, X2010.pct <dbl>,
## #   X2009.pct <dbl>, X2008.pct <dbl>, X2007.pct <dbl>, X2006.pct <dbl>,
## #   X2005.pct <dbl>, X2004.pct <dbl>, X2003.pct <dbl>, X2002.pct <dbl>,
## #   X2001.pct <dbl>, X2000.pct <dbl>, X1999.pct <dbl>, X1998.pct <dbl>,
## #   year <dbl>, payroll.x <dbl>, payroll.y <dbl>, avgwin.y <dbl>, ...
```

```r
missing_values <- sapply( mlb_data, function(x) sum(is.na(x)))
print("Missing values:")
```

```
## [1] "Missing values:"
```

```r
print(missing_values)
```

```
##        payroll         avgwin Team.name.2014          p1998          p1999
##              0              0              0              0              0
##          p2000          p2001          p2002          p2003          p2004
##              0              0              0              0              0
##          p2005          p2006          p2007          p2008          p2009
##              0              0              0              0              0
##          p2010          p2011          p2012          p2013          p2014
```

```
##             0             0             0             0             0
##         X2014         X2013         X2012         X2011         X2010
##             0             0             0             0             0
##         X2009         X2008         X2007         X2006         X2005
##             0             0             0             0             0
##         X2004         X2003         X2002         X2001         X2000
##             0             0             0             0             0
##         X1999         X1998     X2014.pct     X2013.pct     X2012.pct
##             0             0             0             0             0
##     X2011.pct     X2010.pct     X2009.pct     X2008.pct     X2007.pct
##             0             0             0             0             0
##     X2006.pct     X2005.pct     X2004.pct     X2003.pct     X2002.pct
##             0             0             0             0             0
##     X2001.pct     X2000.pct     X1999.pct     X1998.pct
##             0             0             0             0
```

```
duplicate_rows <- mlb_data[duplicated( mlb_data), ]
print("Duplicate rows:")
```

```
## [1] "Duplicate rows:"
```

```
print(duplicate_rows)
```

```
##  [1] payroll        avgwin         Team.name.2014 p1998          p1999
##  [6] p2000          p2001          p2002          p2003          p2004
## [11] p2005          p2006          p2007          p2008          p2009
## [16] p2010          p2011          p2012          p2013          p2014
## [21] X2014          X2013          X2012          X2011          X2010
## [26] X2009          X2008          X2007          X2006          X2005
## [31] X2004          X2003          X2002          X2001          X2000
## [36] X1999          X1998          X2014.pct      X2013.pct      X2012.pct
## [41] X2011.pct      X2010.pct      X2009.pct      X2008.pct      X2007.pct
## [46] X2006.pct      X2005.pct      X2004.pct      X2003.pct      X2002.pct
## [51] X2001.pct      X2000.pct      X1999.pct      X1998.pct
## <0 rows> (or 0-length row.names)
```

```
print("Data types:")
```

```
## [1] "Data types:"
```

```
print(sapply( mlb_data, class))
```

```
##        payroll         avgwin Team.name.2014         p1998         p1999
##      "numeric"      "numeric"       "factor"      "numeric"      "numeric"
##         p2000         p2001         p2002         p2003         p2004
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##         p2005         p2006         p2007         p2008         p2009
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##         p2010         p2011         p2012         p2013         p2014
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##         X2014         X2013         X2012         X2011         X2010
```

```
##      "integer"      "integer"      "integer"      "integer"      "integer"
##          X2009          X2008          X2007          X2006          X2005
##      "integer"      "integer"      "integer"      "integer"      "integer"
##          X2004          X2003          X2002          X2001          X2000
##      "integer"      "integer"      "integer"      "integer"      "integer"
##          X1999          X1998      X2014.pct      X2013.pct      X2012.pct
##      "integer"      "integer"      "numeric"      "numeric"      "numeric"
##      X2011.pct      X2010.pct      X2009.pct      X2008.pct      X2007.pct
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##      X2006.pct      X2005.pct      X2004.pct      X2003.pct      X2002.pct
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##      X2001.pct      X2000.pct      X1999.pct      X1998.pct
##      "numeric"      "numeric"      "numeric"      "numeric"
```

```
ml_pay_long <-  mlb_data %>%
  pivot_longer(cols = starts_with("p"), names_to = "year", values_to = "payroll") %>%
  mutate(year = as.numeric(gsub("p", "", year)))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `year = as.numeric(gsub("p", "", year))`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
aggregated_computed <- ml_pay_long %>%
  group_by(Team.name.2014, year) %>%
  summarise(payroll = mean(payroll))
```

```
## `summarise()` has grouped output by 'Team.name.2014'. You can override using
## the `.groups` argument.
```

```
aggregated_computed
```

```
## # A tibble: 540 x 3
## # Groups:   Team.name.2014 [30]
##    Team.name.2014        year payroll
##    <fct>                <dbl>   <dbl>
##  1 Arizona Diamondbacks  1998    31.6
##  2 Arizona Diamondbacks  1999    70.5
##  3 Arizona Diamondbacks  2000    81.0
##  4 Arizona Diamondbacks  2001    81.2
##  5 Arizona Diamondbacks  2002   103.
##  6 Arizona Diamondbacks  2003    80.6
##  7 Arizona Diamondbacks  2004    70.2
##  8 Arizona Diamondbacks  2005    63.0
##  9 Arizona Diamondbacks  2006    59.7
## 10 Arizona Diamondbacks  2007    52.1
## # i 530 more rows
```

```
win_percentage_data <-  mlb_data %>%
  select(Team.name.2014,avgwin, starts_with("X")) %>%
  pivot_longer(cols = starts_with("X"), names_to = "Year", values_to = "Win_Percentage") %>%
  mutate(Year = as.numeric(gsub("X", "", Year))) %>%
  arrange(Year)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Year = as.numeric(gsub("X", "", Year))`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
win_percentage_data
```

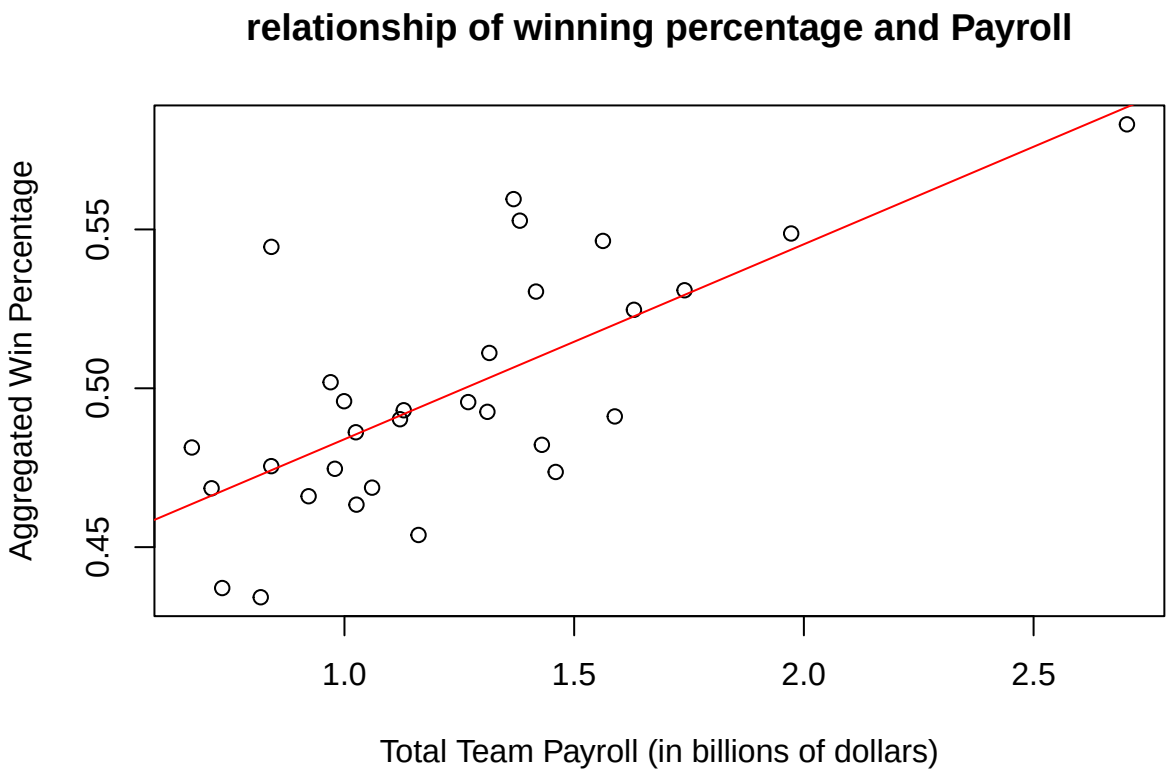
```
## # A tibble: 1,020 x 4
##    Team.name.2014       avgwin  Year Win_Percentage
##    <fct>                 <dbl> <dbl>          <dbl>
##  1 Arizona Diamondbacks  0.490  1998             65
##  2 Atlanta Braves        0.553  1998            106
##  3 Baltimore Orioles     0.454  1998             79
##  4 Boston Red Sox        0.549  1998             92
##  5 Chicago Cubs          0.474  1998             90
##  6 Chicago White Sox     0.511  1998             80
##  7 Cincinnati Reds       0.486  1998             77
##  8 Cleveland Indians     0.496  1998             89
##  9 Colorado Rockies      0.463  1998             77
## 10 Detroit Tigers        0.482  1998             65
## # i 1,010 more rows
```

```
model <- lm(avgwin ~ payroll, data =  mlb_data)

summary(model)
```

```
##
## Call:
## lm(formula = avgwin ~ payroll, data = mlb_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.040034 -0.017492  0.000936  0.010954  0.070302
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.42260    0.01534  27.555  < 2e-16 ***
## payroll      0.06137    0.01173   5.233 1.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02697 on 28 degrees of freedom
## Multiple R-squared:  0.4944, Adjusted R-squared:  0.4763
## F-statistic: 27.38 on 1 and 28 DF,  p-value: 1.469e-05
```

```
plot(mlb_data$payroll,  mlb_data$avgwin,
     xlab = "Total Team Payroll (in billions of dollars)",
     ylab = "Aggregated Win Percentage",
     main = "relationship of winning percentage and Payroll")
abline(model, col = "red")
```




```
###with the redline representing the slope. we can say that the even if the team has a low payroll thei
```

```
mlb_data$efficiency <-  mlb_data$avgwin / mlb_data$payroll

 mlb_data <-  mlb_data[order(mlb_data$efficiency, decreasing = TRUE), ]

head( mlb_data[, c("Team.name.2014", "efficiency")], 30)
```

```
##           Team.name.2014 efficiency
## 15         Miami Marlins  0.7208173
## 27       Tampa Bay Rays  0.6591511
## 20     Oakland Athletics  0.6475023
## 22    Pittsburgh Pirates  0.5956153
## 23      San Diego Padres  0.5656087
## 12    Kansas City Royals  0.5310098
## 17       Minnesota Twins  0.5175197
## 30  Washington Nationals  0.5054638
## 8      Cleveland Indians  0.4963290
## 16     Milwaukee Brewers  0.4847920
## 7        Cincinnati Reds  0.4744038
## 9       Colorado Rockies  0.4515659
## 11        Houston Astros  0.4421262
## 1   Arizona Diamondbacks  0.4373897
## 29     Toronto Blue Jays  0.4367340
## 26   St. Louis Cardinals  0.4089881
```

```
## 2         Atlanta Braves  0.4000549
## 3      Baltimore Orioles  0.3908202
## 28         Texas Rangers  0.3904841
## 6      Chicago White Sox  0.3885666
## 25      Seattle Mariners  0.3756954
## 24  San Francisco Giants  0.3743705
## 13    Los Angeles Angels  0.3496570
## 10        Detroit Tigers  0.3372659
## 5           Chicago Cubs  0.3244735
## 21 Philadelphia Phillies  0.3218793
## 18         New York Mets  0.3091979
## 14   Los Angeles Dodgers  0.3050375
## 4         Boston Red Sox  0.2782035
## 19      New York Yankees  0.2156931
```

```
###you can see that the most efficient team is the miami marlins with an efficient score of 0.7208173 f
```

‘‘‘