

# Data 607 Project 2

Joao De Oliveira

2025-10-05

## Overview

This report intends to use 3 “wide” datasets and tidy them into long formats before analyzing them. I selected a dataset with all English Premier League results from 2024/2025 season. I analyzed Man City’s performance by studying their performance by month and the difference between home and away games performance. The second dataset is a Human Development Index by country that I cleaned and analyzed the changes between 1990 and 2022. Finally, the third dataset is a dataset with Glassdoor jobs posts across the US. After cleaning and formatting it, I found the highest paying companies per city and per city based on Glassdoor job posts.

## Load Libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.2      v tibble    3.3.0
## v lubridate  1.9.4      v tidyr     1.3.1
## v purrr      1.1.0
## -- 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(lubridate)
library(dplyr)
library(stringr)
library(readr)
library(purrr)
```

## Load data

```
premier_league_raw <- readr::read_csv("https://raw.githubusercontent.com/JDO-MSDS/DATA-607/refs/heads/main/premier_league_raw.csv")
```

```
## Rows: 380 Columns: 120
```

```
## -- Column specification -----
## Delimiter: ","
## chr      (7): Div, Date, HomeTeam, AwayTeam, FTR, HTR, Referee
## dbl     (112): FTHG, FTAG, HTHG, HTAG, HS, AS, HST, AST, HF, AF, HC, AC, HY, AY...
## time     (1): Time
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
premier_league <- premier_league_raw %>%
  mutate(Date = lubridate::dmy(Date)) %>%
  arrange(Date)

# glimpse(premier_league)
```

## Tidy Data with one team game per row

```
home <- premier_league %>%
  transmute(
    date = Date,
    team = HomeTeam,
    opponent = AwayTeam,
    venue = "Home",
    gf = FTHG,
    ga = FTAG,
    result = case_when(FTR == "H" ~ "W",
                      FTR == "D" ~ "D",
                      TRUE ~ "L"),

    shots = HS,
    shots_target = HST,
    fouls = HF,
    corners = HC,
    yellow = HY,
    red = HR
  )

away <- premier_league %>%
  transmute(
    date = Date,
    team = AwayTeam,
    opponent = HomeTeam,
    venue = "Away",
    gf = FTAG,
    ga = FTHG,
    result = case_when(FTR == "A" ~ "W",
                      FTR == "D" ~ "D",
                      TRUE ~ "L"),

    shots = AS,
    shots_target = AST,
    fouls = AF,
    corners = AC,
    yellow = AY,
```

```

    red = AR
  )

# table
team_long <- bind_rows(home, away) %>%
  arrange(date)

# save
readr::write_csv(team_long, "premier_league_long.csv")

glimpse(team_long)

```

```

## Rows: 760
## Columns: 13
## $ date      <date> 2024-08-16, 2024-08-16, 2024-08-17, 2024-08-17, 2024-08-~
## $ team      <chr> "Man United", "Fulham", "Ipswich", "Arsenal", "Everton", ~
## $ opponent  <chr> "Fulham", "Man United", "Liverpool", "Wolves", "Brighton"~
## $ venue     <chr> "Home", "Away", "Home", "Home", "Home", "Home", "Home", "~
## $ gf        <dbl> 1, 0, 0, 2, 0, 1, 1, 1, 2, 0, 3, 0, 1, 2, 2, 0, 1, 2, 1, ~
## $ ga        <dbl> 0, 1, 2, 0, 3, 0, 1, 2, 0, 2, 0, 1, 1, 1, 1, 2, 2, 0, 1, ~
## $ result    <chr> "W", "L", "L", "W", "L", "W", "D", "L", "W", "L", "W", "L~
## $ shots     <dbl> 14, 10, 7, 18, 9, 3, 14, 14, 18, 9, 10, 19, 13, 15, 9, 10~
## $ shots_target <dbl> 5, 2, 2, 6, 1, 1, 8, 3, 5, 3, 5, 4, 4, 3, 5, 3, 6, 5, 3, ~
## $ fouls     <dbl> 12, 10, 9, 17, 8, 15, 17, 18, 18, 14, 8, 16, 8, 11, 6, 12~
## $ corners   <dbl> 7, 8, 2, 8, 1, 3, 2, 5, 10, 2, 5, 12, 6, 3, 4, 4, 7, 3, 2~
## $ yellow    <dbl> 2, 3, 3, 2, 1, 2, 1, 1, 1, 2, 1, 4, 3, 2, 1, 1, 5, 1, 1, ~
## $ red       <dbl> 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~

```

## Data Analysis

```

team_name <- "Man City"

team_month <- team_long %>%
  filter(team == team_name) %>%
  mutate(
    month = floor_date(date, "month"),
    points = case_when(result == "W" ~ 3L,
                       result == "D" ~ 1L,
                       TRUE ~ 0L),
    gd = gf - ga
  ) %>%
  group_by(month) %>%
  summarise(
    matches = n(),
    pts = sum(points),
    pts_per_game = mean(points),
    avg_gd = mean(gd),
    win_rate = mean(result == "W"),
    .groups = "drop"
  )

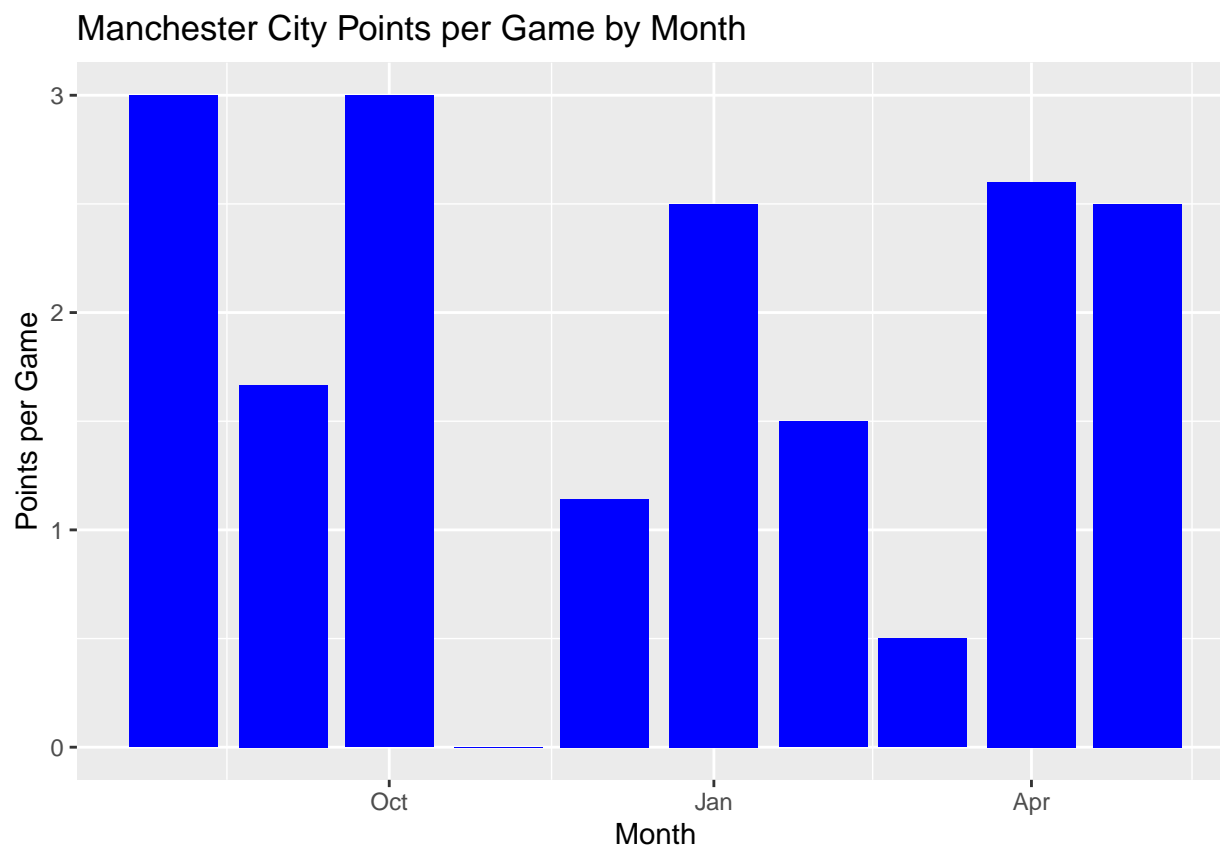
```

```
team_month
```

```
## # A tibble: 10 x 6
##   month      matches    pts pts_per_game avg_gd win_rate
##   <date>      <int> <int>      <dbl>  <dbl>  <dbl>
## 1 2024-08-01         3     9         3    2.33     1
## 2 2024-09-01         3     5        1.67  0.333  0.333
## 3 2024-10-01         3     9         3     1     1
## 4 2024-11-01         3     0         0    -2     0
## 5 2024-12-01         7     8        1.14  0.143  0.286
## 6 2025-01-01         4    10         2.5    2.75   0.75
## 7 2025-02-01         4     6         1.5   -0.25   0.5
## 8 2025-03-01         2     1         0.5   -0.5     0
## 9 2025-04-01         5    13         2.6    1.6    0.8
## 10 2025-05-01        4    10         2.5    1.25   0.75
```

Plot of points per game each month

```
ggplot(team_month, aes(x = month, y = pts_per_game)) +
  geom_col(fill = "blue") +
  scale_x_date(date_labels = "%b") +
  labs(title = "Manchester City Points per Game by Month", x = "Month", y = "Points per Game")
```



## Home vs Away performance comparison

```
home_away <- team_long %>%
  filter(team == team_name) %>%
  mutate(points = case_when(result == "W" ~ 3L,
                             result == "D" ~ 1L,
                             TRUE ~ 0L
                           )) %>%

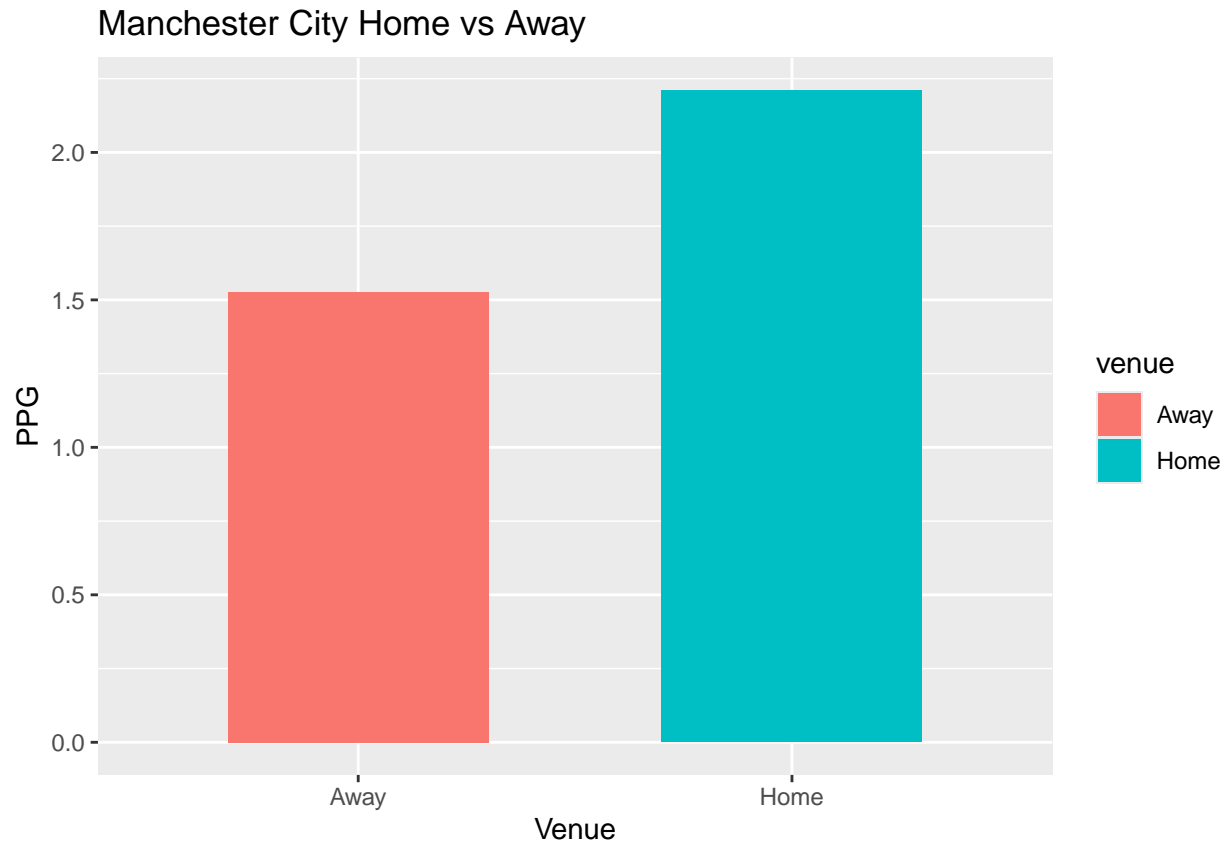
  group_by(venue) %>%
  summarise(
    matches = n(),
    ppg = mean(points),
    avg_gf = mean(gf),
    avg_ga = mean(ga),
    win_rate = mean(result == "W"),
    .groups = "drop"
  )
```

home\_away

```
## # A tibble: 2 x 6
##   venue matches   ppg avg_gf avg_ga win_rate
##   <chr>   <int> <dbl> <dbl> <dbl>   <dbl>
## 1 Away       19  1.53  1.53  1.11  0.421
## 2 Home       19  2.21  2.26  1.21  0.684
```

## Plot

```
ggplot(home_away, aes(x = venue, y = ppg, fill = venue)) +
  geom_col(width = 0.6) +
  labs(title = "Manchester City Home vs Away", x = "Venue", y = "PPG")
```



## Conclusion

By looking at the Points per Game by Month plot, we can see that August and October were top performing months with only wins (3pts) while September, December, February, and March were clearly the months where Man City underperformed, which can be explained by European competition physical and mental overload. Manchester City had a much stronger performance in home games than in away games, which is not surprising, but the away performance (around 1.5 pts) was clearly below what is expected from a championship contender.

## Untidy dataset Human Development Index

The data that I chose is the Human Development Index (HDI) from 1990 to 2022, which can be found in this link: [worldhdi: Human Development Index Worldwide 1990-2022 INDEX DATA](https://worldhdi.org/) One possible analysis of this dataset would be to explore trends in human development across countries and regions over time. For example, we could examine which countries have experienced the greatest improvements in their HDI from 1990 to 2022, or compare the progress of different regions such as Africa, Asia, and Latin America.

## Load and tidy data

```
path <- "https://raw.githubusercontent.com/JDO-MSDS/DATA-607/main/Project2/HDR23-24_Statistical_Annex_HI
hdr_lines <- readr::read_lines(path, n_max = 80)
```

```

skip_lines <- {
  idx <- which(stringr::str_detect(hdr_lines, "\\bCountry\\b"))[1]
  if (is.na(idx)) 3 else idx - 1
}

hdi <- readr::read_csv(path, skip = skip_lines, show_col_types = FALSE)

## New names:
## * ' ' -> '...4'
## * ' ' -> '...6'
## * ' ' -> '...8'
## * ' ' -> '...10'
## * ' ' -> '...12'
## * ' ' -> '...14'
## * ' ' -> '...16'
## * ' ' -> '...18'
## * ' ' -> '...22'
## * ' ' -> '...24'
## * ' ' -> '...26'

# Ensure the country column is named 'Country'
if (!"Country" %in% names(hdi)) {
  ci <- which(grepl("^\\s*Country\\s*$", names(hdi), ignore.case = TRUE))[1]
  if (!is.na(ci)) names(hdi)[ci] <- "Country" else if (ncol(hdi) >= 2) names(hdi)[2] <- "Country"
}

hdi <- hdi %>% filter(!is.na(Country))

year_cols <- names(hdi)[stringr::str_detect(names(hdi), "(19|20)\\d{2}")]

hdi_long <- hdi %>%
  tidyr::pivot_longer(
    cols = all_of(year_cols),
    names_to = "year_col",
    values_to = "hdi_raw",
    values_transform = list(hdi_raw = as.character)
  ) %>%
  mutate(
    year = readr::parse_integer(stringr::str_extract(year_col, "(19|20)\\d{2}")),
    hdi = suppressWarnings(as.numeric(hdi_raw))
  ) %>%
  filter(!is.na(year), dplyr::between(year, 1990, 2022), !is.na(hdi)) %>%
  group_by(Country, year) %>%
  summarise(hdi = mean(hdi), .groups = "drop")

```

## Improvement analysis between 1990 and 2022

```

hdi_1990 <- hdi_long %>% filter(year == 1990) %>% select(Country, hdi_1990 = hdi)
hdi_2022 <- hdi_long %>% filter(year == 2022) %>% select(Country, hdi_2022 = hdi)

improve_1990_2022 <- inner_join(hdi_1990, hdi_2022, by = "Country") %>%

```

```
mutate(delta = hdi_2022 - hdi_1990) %>%
  arrange(desc(delta))

cat("HDI Improvement 1990 → 2022 (Top 20):\n")
```

```
## HDI Improvement 1990 → 2022 (Top 20):
```

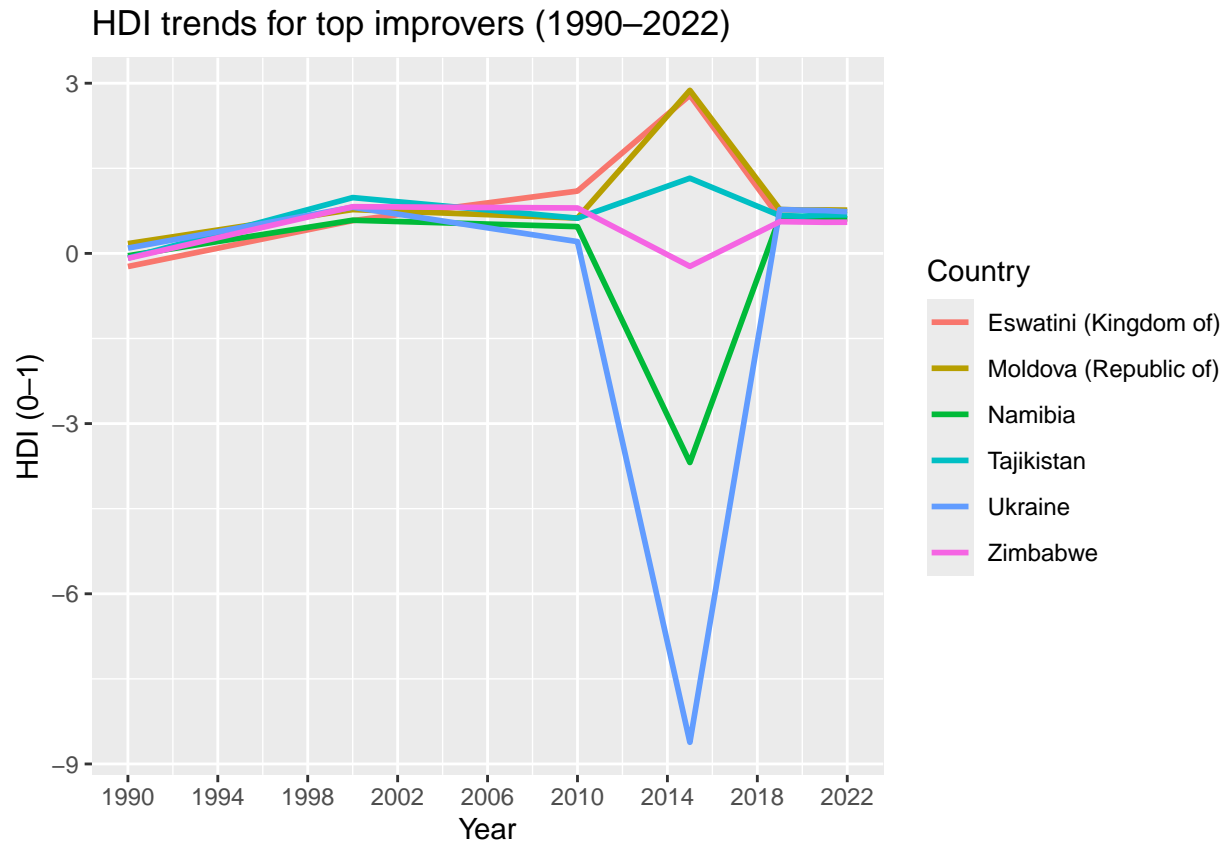
```
print(improve_1990_2022 %>% slice_head(n = 20))
```

```
## # A tibble: 20 x 4
##   Country          hdi_1990 hdi_2022 delta
##   <chr>          <dbl>    <dbl> <dbl>
## 1 Eswatini (Kingdom of) -0.231    0.61  0.841
## 2 Tajikistan          -0.0813   0.679 0.760
## 3 Namibia             -0.0417   0.61  0.652
## 4 Ukraine              0.0937   0.734 0.640
## 5 Zimbabwe            -0.087    0.55  0.637
## 6 Moldova (Republic of) 0.173    0.763 0.590
## 7 Congo                0.0737   0.593 0.519
## 8 Russian Federation    0.317    0.821 0.504
## 9 United States         0.425    0.927 0.502
## 10 Kyrgyzstan           0.206    0.701 0.495
## 11 Brunei Darussalam     0.360    0.823 0.463
## 12 Jamaica              0.248    0.706 0.458
## 13 Canada                0.484    0.935 0.451
## 14 Australia             0.501    0.946 0.445
## 15 Lesotho               0.083    0.521 0.438
## 16 San Marino            0.447    0.867 0.42
## 17 Switzerland          0.55     0.967 0.417
## 18 Japan                 0.512    0.92  0.408
## 19 Armenia               0.396    0.786 0.39
## 20 South Africa         0.328    0.717 0.389
```

Plot of top 6 improvements

```
top_countries <- improve_1990_2022 %>%
  slice_head(n = 6) %>%
  pull(Country)

hdi_long %>%
  filter(Country %in% top_countries) %>%
  ggplot(aes(x = year, y = hdi, color = Country, group = Country)) +
  geom_line(linewidth = 1) +
  scale_x_continuous(breaks = seq(1990, 2022, by = 4)) +
  labs(
    title = "HDI trends for top improvers (1990-2022)",
    x = "Year",
    y = "HDI (0-1)"
  )
```



## Conclusion 2

HDI seems to have an upward trend across countries. The improvement table highlights a set of countries with especially large gains, being most of them countries that started with low levels of HDI in 1990. A few countries show pauses or small dips, with emphasis in cases like Ukraine where there is a sharp dip around 2014/2015 (Crimea invasion).

## Uncleaned Data Science Job Postings on Glassdoor

This section of the projects intends to tidy the job postings data from Glassdoor and analyze which company pays the highest by city and by state.

```
jobs_raw <- readr::read_csv("https://raw.githubusercontent.com/JDO-MSDS/DATA-607/refs/heads/main/Project2/jobs_raw.csv")

clean_company <- function(a) {
  a %>% str_remove("\\s+\\d+\\.?\\d*$") %>% str_squish()
}

# parse city and state
parse_city <- function(a) str_squish(str_extract(a, "[^,]+"))
parse_state <- function(a) str_squish(str_replace_na(str_extract(a, "(?<=,)\\s*.*$")))

# parse salary in thousands
parse_salary_k <- function(txt) {
```

```

if (is.na(txt) || txt == "-1") return(NA_real_)
s <- tolower(txt)
is_hourly <- str_detect(s, "hour")
nums <- str_extract_all(s, "\\d+")[[1]]
if (length(nums) == 0) return(NA_real_)
minv <- as.numeric(nums[1])
maxv <- as.numeric(ifelse(length(nums) >= 2, nums[2], nums[1]))
if (is_hourly) {
  conv <- 2080/1000
  midk <- (minv + maxv)/2 * conv
} else {
  midk <- (minv + maxv)/2
}
midk
}

# jobs
jobs <- jobs_raw %>%
  mutate(
    company      = clean_company(`Company Name`),
    rating       = suppressWarnings(as.numeric(Rating)),
    city         = parse_city(Location),
    state        = parse_state(Location),
    salary_text  = `Salary Estimate`,
    avg_salary_k = map_dbl(salary_text, parse_salary_k)
  ) %>%
  filter(!is.na(avg_salary_k) | !is.na(rating))

jobs %>% select(company, city, state, rating, salary_text, avg_salary_k) %>% slice_head(n = 8)

```

```

## # A tibble: 8 x 6
##   company      city      state rating salary_text      avg_salary_k
##   <chr>      <chr>    <chr>  <dbl> <chr>          <dbl>
## 1 Healthfirst New York  NY      3.1 $137K-$171K (Glass~ 154
## 2 ManTech     Chantilly VA       4.2 $137K-$171K (Glass~ 154
## 3 Analysis Group Boston    MA       3.8 $137K-$171K (Glass~ 154
## 4 INFICON     Newton    MA       3.5 $137K-$171K (Glass~ 154
## 5 Affinity Solutions New York  NY       2.9 $137K-$171K (Glass~ 154
## 6 HG Insights  Santa Barbara CA       4.2 $137K-$171K (Glass~ 154
## 7 Novartis    Cambridge MA       3.9 $137K-$171K (Glass~ 154
## 8 iRobot      Bedford   MA       3.5 $137K-$171K (Glass~ 154

```

### Highest paying company per city and state

```

city_company_pay <- jobs %>%
  filter(!is.na(avg_salary_k), !is.na(city)) %>%
  group_by(city, company) %>%
  summarise(mean_salary_k = mean(avg_salary_k, na.rm = TRUE),
            n_postings = n(), .groups = "drop_last") %>%
  arrange(desc(mean_salary_k)) %>%
  ungroup()

```

```

# top company per city
top_by_city <- city_company_pay %>%
  group_by(city) %>%
  slice_max(order_by = mean_salary_k, n = 1, with_ties = TRUE) %>%
  arrange(city, desc(mean_salary_k), desc(n_postings), company) %>%
  ungroup()

top_by_city %>% arrange(desc(mean_salary_k)) %>% slice_head(n = 20)

## # A tibble: 20 x 4
##   city                company                mean_salary_k n_postings
##   <chr>                <chr>                <dbl>         <int>
## 1 Dayton              Southwest Research Institute      272.           1
## 2 Fort Sam Houston    Alaka'ina Foundation Family of Com~      272.           1
## 3 New York            CompuForce              272.           1
## 4 Pleasanton          Roche                   272.           1
## 5 Raleigh             10x Genomics            272.           1
## 6 Seattle             Sharpedge Solutions Inc      272.           1
## 7 United States       Creative Circle           272.           1
## 8 Washington          Aptive                   272.           1
## 9 Wilmington          AstraZeneca              272.           1
## 10 Lexington Park     Hexagon US Federal        204.           2
## 11 Cambridge          Monte Rosa Therapeutics     198.           2
## 12 Atlanta            Cambridge FX              185.           1
## 13 Chicago            Triplebyte               185.           1
## 14 Pittsburgh         Advance Sourcing Concepts    185.           1
## 15 Remote             TrueAccord               185.           1
## 16 Reston             ASRC Federal Holding Company  185.           1
## 17 Richmond          GSK                      185.           1
## 18 San Francisco      Metromile                185.           1
## 19 Woodlawn           ALTA IT Services          185.           1
## 20 Arlington         Leidos                   183.           1

state_company_pay <- jobs %>%
  filter(!is.na(avg_salary_k), !is.na(state)) %>%
  group_by(state, company) %>%
  summarise(mean_salary_k = mean(avg_salary_k, na.rm = TRUE), n_postings = n(),
    .groups = "drop_last") %>%
  arrange(desc(mean_salary_k)) %>%
  ungroup()

top_state <- state_company_pay %>%
  group_by(state) %>%
  slice_max(order_by = mean_salary_k, n = 1, with_ties = TRUE) %>%
  arrange(state, desc(mean_salary_k), desc(n_postings), company) %>%
  ungroup()

top_state %>% arrange(desc(mean_salary_k)) %>% slice_head(n=20)

## # A tibble: 20 x 4
##   state company                mean_salary_k n_postings
##   <chr> <chr>                <dbl>         <int>

```

##	1	CA	Roche	272.	1
##	2	DC	Aptive	272.	1
##	3	DE	AstraZeneca	272.	1
##	4	NA	Creative Circle	272.	1
##	5	NC	10x Genomics	272.	1
##	6	NY	CompuForce	272.	1
##	7	OH	Southwest Research Institute	272.	1
##	8	TX	Alaka'ina Foundation Family of Companies	272.	1
##	9	WA	Sharpedge Solutions Inc	272.	1
##	10	MD	Hexagon US Federal	204.	2
##	11	MA	Monte Rosa Therapeutics	198.	2
##	12	GA	Cambridge FX	185	1
##	13	IL	Triplebyte	185	1
##	14	PA	Advance Sourcing Concepts	185	1
##	15	VA	ASRC Federal Holding Company	185	1
##	16	VA	GSK	185	1
##	17	WI	Colony Brands	183	1
##	18	WI	Oshkosh Corporation	183	1
##	19	MI	Criteo	173	2
##	20	OR	New Relic	164.	1

## Conclusion

We can see that the top highest paying companies per city end up being the highest paying companies in their corresponding states with the curiosity that all of them have an average annual salart of \$272K.