Project 3: Data Exploration

Alice Ding

2023-03-14

Overview

The dataset our team has chosen to use for this project is a table of job postings on https://data.cityofnewyork.us/ provided by the Department of Citywide Administrative Services (DCAS). A description taken from the source is below:

`This dataset contains current job postings available on the City of New York's official jobs site (http://www.nyc.gov/html/careers/html/search/search.shtml). Internal postings available to city employees and external postings available to the general public are included.`

The original table is comprised of 30 columns, but in order to make the data more digestible and easier to read, we've broken this table down into 4 smaller ones; below is an ERD of our relational database model.

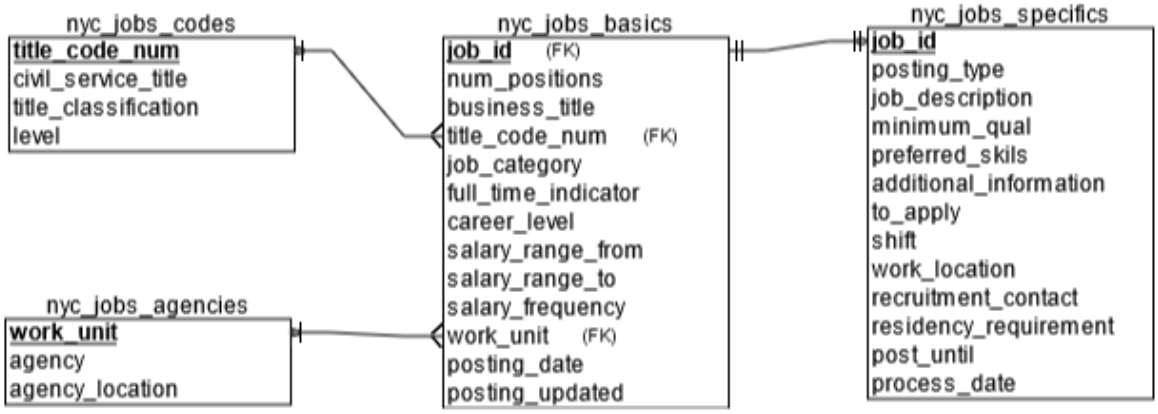


Figure 1: ERD

To keep things relevant, here are the fields we'll be going through as part of this exploratory analysis:

- `nyc_jobs_basics.job_id`: The job opening identification ("Job ID") number that corresponds to and represents a job posting notice published on behalf of a New York City agency.
- `nyc_jobs_basics.work_unit` (agency): Name of the New York City agency ("agency" or "hiring agency") where a job vacancy exists.
- `nyc_jobs_specifics.posting_type`: Identifies whether a job posting is an Internal or External posting. Internal postings are available to City employees only and external postings are available to the general public.

- `nyc_jobs_basics.job_category`: The occupational group in which the posted job belongs, such as: Administration & Human Resources; Communications & Intergovernmental Affairs; Constituent Services & Community Programs; Engineering, Architecture, & Planning; Finance, Accounting, & Procurement; Health; Technology, Data & Innovation; Legal Affairs; Building Operations & Maintenance; Policy, Research & Analysis; Public Safety, Inspections, & Enforcement; Social Services
- `nyc_jobs_basics.full_time_indicator`: "This denotes whether the job is a full time or part time employment; F - Full time; P - Part time"
- `nyc_jobs_basics.career_level`: "This denotes the career level of the job. The possible career levels are:; Student; Entry-level; Experienced (non-manager); Manager; Executive"
- `nyc_jobs_basics.salary_range_from`: The lowest salary on a job posting for a position within the salary band for the related civil service title.
- `nyc_jobs_basics.salary_range_to`: The highest salary on a job posting for a position within the salary band for the related civil service title.
- `nyc_jobs_basics.salary_frequency`: The frequency of proposed salary. Possible salary frequency values include "hourly", "daily", and "annual".
- `nyc_jobs_basics.posting_date`: The date and time that a job vacancy was posted in MM/DD/YY format.

Exploration

First, we must connect to the database; we've chosen to use Azure as our hosting service.

```
library(plyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
library(odbc)
library(DBI)
library(keyring)
library(ggplot2)

if (!("Overview8909" %in% as_vector(key_list()[2]))) {
  key_set("Project3SQL","Overview8909")
}
```

```
my_connection <- dbConnect(drv = odbc::odbc(),
                           Driver = "ODBC Driver 18 for SQL Server",
                           server = "tcp:data607project3server.database.windows.net,1433",
                           database = "Data 607 Project 3 Database",
                           uid = "Overview8909",
                           pwd = key_get("Project3SQL","Overview8909"),
                           encoding = "latin1"
                           )

nyc_jobs_basics_sql <- "select * from nyc_jobs_basics"
nyc_jobs_agencies_sql <- "select * from nyc_jobs_agencies"
nyc_jobs_codes_sql <- "select * from nyc_jobs_codes"
nyc_jobs_specifics_sql <- "select * from nyc_jobs_specifics"

nyc_jobs_basics <- dbGetQuery(my_connection, nyc_jobs_basics_sql)
nyc_jobs_agencies <- dbGetQuery(my_connection, nyc_jobs_agencies_sql)
nyc_jobs_codes <- dbGetQuery(my_connection, nyc_jobs_codes_sql)
nyc_jobs_specifics <- dbGetQuery(my_connection, nyc_jobs_specifics_sql)

dbDisconnect(my_connection)
```

To start, we'll first just get an overall count of the data.

```
overall_count <- nyc_jobs_basics |>
  dplyr::summarise(count = n())

sprintf("There are %i jobs currently posted.", overall_count$count)
```

```
## [1] "There are 3260 jobs currently posted."
```

As of Thursday, March 17, there are 3,260 jobs currently available.

How can we break this down further?

Agency Posted

What agencies are posting these jobs?

```
agency_count <- nyc_jobs_basics |>
  dplyr::summarise(count = n_distinct(work_unit))

sprintf("There are %i agencies.", agency_count$count)
```

```
## [1] "There are 1048 agencies."
```

```
counts_by_agency <- nyc_jobs_basics |>
  group_by(work_unit) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_agency, 10)
```

```
## # A tibble: 10 x 3
##    work_unit                       count percent
##    <chr>                           <int>   <dbl>
##  1 BWT - ADMINISTRATION/PERSONNEL     43   1.32
##  2 Information Technology             38   1.17
##  3 Manhattan Property Management      36   1.10
##  4 PROJECT MANAGEMENT AND CONSTR.     35   1.07
##  5 FIA Operations-NM                  34   1.04
##  6 Office of Energy Conservatio       30   0.920
##  7 Support Staff                      30   0.920
##  8 Commissioner                       29   0.890
##  9 Mgmt Information System-NM         29   0.890
## 10 Adult Offender Pgms.               28   0.859
```

In total, there are 1,048 different agencies and no agency is more prevalent in job postings than another. The fact that the largest one only has 43 current job postings out of 3k+ (~1%) shows that it's an extremely diverse collection of jobs.

Posting Type

Are a majority of these listings internal or external?

```
counts_by_posting_type <- nyc_jobs_specifics |>
  group_by(posting_type) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_posting_type)
```

```
## # A tibble: 2 x 3
##   posting_type count percent
##   <chr>        <int>   <dbl>
## 1 Internal      1779    54.6
## 2 External      1481    45.4
```

Pretty close to 50/50, although internal is slightly higher at 55% vs. external's 45%.

Job Category

What job categories do these postings fall under?

```
category_count <- nyc_jobs_basics |>
  dplyr::summarise(count = n_distinct(job_category))

sprintf("There are %i job categories", category_count$count)
```

```
## [1] "There are 192 job categories"
```

```
counts_by_job_category <- nyc_jobs_basics |>
  group_by(job_category) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_job_category, 10)
```

```
## # A tibble: 10 x 3
##    job_category                                 count percent
##    <chr>                                        <int>   <dbl>
##  1 Engineering, Architecture, & Planning          433   13.3
##  2 Technology, Data & Innovation                  277    8.50
##  3 Legal Affairs                                  205    6.29
##  4 Administration & Human Resources               189    5.80
##  5 Social Services                                180    5.52
##  6 Building Operations & Maintenance              179    5.49
##  7 Finance, Accounting, & Procurement             165    5.06
##  8 Constituent Services & Community Programs      161    4.94
##  9 Public Safety, Inspections, & Enforcement      153    4.69
## 10 Health                                         152    4.66
```

In total, there are 192 distinct job categories. Of these, we can see that the most common categories are `Engineering, Architecture, & Planning` (~13%), `Technology, Data, & Innovation` (~8%), `Legal Affairs` (7%), and `Administration & Human Resources` (~6%).

Circling back to our goal of finding data science related jobs and skills, one way to hone in on this is to use category as a way to see what how many of these postings fall into categories with the words `analysis`, `analytics`, `data`, or `statistics`?

```
to_find <- c("Data", "Analysis", "Analytics", "Statistics")
matches <- unique(grep(paste(to_find, collapse="|"), nyc_jobs_basics$job_category, value=TRUE))

relevant_categories <- filter(nyc_jobs_basics, job_category %in% matches)

overall_relevant_count <- relevant_categories |>
  dplyr::summarise(count = n())

counts_by_relevant_job_category <- relevant_categories |>
  group_by(job_category) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(relevant_categories),
            percent_of_total = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

sprintf("There are %i jobs currently posted within a relevant category.", overall_relevant_count$count)
```

```
## [1] "There are 770 jobs currently posted within a relevant category."
```

```r
head(counts_by_relevant_job_category, 20)
```

```
## # A tibble: 20 x 4
##    job_category                                          count percent perce~1
##    <chr>                                                 <int>   <dbl>   <dbl>
##  1 Technology, Data & Innovation                           277  36.0     8.50
##  2 Policy, Research & Analysis                              85  11.0     2.61
##  3 Finance, Accounting, & Procurement Policy, Research & ~    62   8.05    1.90
##  4 Technology, Data & Innovation Social Services            36   4.68    1.10
##  5 Engineering, Architecture, & Planning Policy, Research~    21   2.73    0.644
##  6 Health Policy, Research & Analysis                       20   2.60    0.613
##  7 Legal Affairs Policy, Research & Analysis                18   2.34    0.552
##  8 Technology, Data & Innovation Policy, Research & Analy~    18   2.34    0.552
##  9 Policy, Research & Analysis Public Safety, Inspections~    16   2.08    0.491
## 10 Policy, Research & Analysis Social Services              16   2.08    0.491
## 11 Administration & Human Resources Policy, Research & An~    14   1.82    0.429
## 12 Engineering, Architecture, & Planning Technology, Data~    12   1.56    0.368
## 13 Communications & Intergovernmental Affairs Policy, Res~    11   1.43    0.337
## 14 Administration & Human Resources Technology, Data & In~    10   1.30    0.307
## 15 Constituent Services & Community Programs Policy, Rese~     8   1.04    0.245
## 16 Constituent Services & Community Programs Health Polic~     7   0.909   0.215
## 17 Engineering, Architecture, & Planning Policy, Research~     7   0.909   0.215
## 18 Technology, Data & Innovation Building Operations & Ma~     6   0.779   0.184
## 19 Constituent Services & Community Programs Communicatio~     5   0.649   0.153
## 20 Legal Affairs Policy, Research & Analysis Public Safet~     5   0.649   0.153
## # ... with abbreviated variable name 1: percent_of_total
```

Here, we can see there are 784 jobs currently posted with a category containing one of our keywords with `Policy, Research & Analysis` bubbling up as the second most popular relevant category at 85 postings (~11% of this subset, ~3% of total).

Full/Part Time

What is the breakdown of these jobs by full vs. part time?

```r
counts_by_full_part_time <- nyc_jobs_basics |>
  group_by(full_time_indicator) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_full_part_time)
```

```
## # A tibble: 3 x 3
##   full_time_indicator count percent
##   <chr>               <int>   <dbl>
## 1 F                    3020   92.6
## 2 <NA>                  123    3.77
## 3 P                     117    3.59
```

It seems like there are a few rows where this isn't filled out, but a vast majority of the jobs posted are full-time listings.

Career Level

What is the breakdown of these jobs by career level?

```
counts_by_career_level <- nyc_jobs_basics |>
  group_by(career_level) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_career_level)
```

```
## # A tibble: 6 x 3
##   career_level              count percent
##   <chr>                     <int>   <dbl>
## 1 Experienced (non-manager)  2349 72.1
## 2 Entry-Level                 365 11.2
## 3 Manager                     329 10.1
## 4 Student                     168  5.15
## 5 Executive                    48  1.47
## 6 <NA>                          1  0.0307
```

Interestingly, 72% (2.3k) of these jobs are for `Experienced (non-manager)` roles.

Salary Information

How do these jobs look in terms of salaray?

Frequency We can start with frequency – how often are folks paid for these roles?

```
counts_by_salary_frequency <- nyc_jobs_basics |>
  group_by(salary_frequency) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(counts_by_salary_frequency)
```

```
## # A tibble: 3 x 3
##   salary_frequency count percent
##   <chr>            <int>   <dbl>
## 1 Annual            2897  88.9
## 2 Hourly             332  10.2
## 3 Daily               31   0.951
```

A vast majority at ~89% (2.9k) are annual, although in order to properly compare compensation, we'll have to adjust the hourly and daily rates up to an annual value. We'll assume 40 hour weeks and 52 weeks in a year for the hourly folks and 365 days a year for the daily ones. I'll also round these adjusted salary ranges to the nearest $10,000.

```
salary_from_adjusted <- c()
salary_to_adjusted <- c()

freq <- ""

for(i in 1:nrow(nyc_jobs_basics)) {
    freq <- nyc_jobs_basics$salary_frequency[i]
    if (freq == "Annual") {
      salary_from_adjusted <- append(salary_from_adjusted, round_any(nyc_jobs_basics$salary_range_from[
      salary_to_adjusted <- append(salary_to_adjusted, round_any(nyc_jobs_basics$salary_range_to[i], 10
    } else if (freq == "Hourly") {
      salary_from_adjusted <- append(salary_from_adjusted, round_any(nyc_jobs_basics$salary_range_from[
      salary_to_adjusted <- append(salary_to_adjusted, round_any(nyc_jobs_basics$salary_range_to[i] * 4
    } else { # this means it's daily
      salary_from_adjusted <- append(salary_from_adjusted, round_any(nyc_jobs_basics$salary_range_from[
      salary_to_adjusted <- append(salary_to_adjusted, round_any(nyc_jobs_basics$salary_range_to[i] * 3
    }
}

nyc_jobs_basics$salary_from_adjusted <- salary_from_adjusted
nyc_jobs_basics$salary_to_adjusted <- salary_to_adjusted

head(nyc_jobs_basics)
```

```
##   job_id num_positions                                        business_title
## 1  97899             1           EXECUTIVE DIRECTOR, BUSINESS DEVELOPMENT
## 2 137433             1                                    Contract Analyst
## 3 152738             1                                      Office Manager
## 4 167179             1          CERTIFIED IT ADMINISTRATOR (WAN), Level 4
## 5 171040             1 Clerical Associate, Bureau of Communicable Diseases
## 6 175362             1      Clerical Associate, Bureau of Vital Statistics
##   title_code_num level                                job_category
## 1          10009    M3                                        <NA>
## 2          12158    03          Finance, Accounting, & Procurement
## 3          10251    03           Clerical & Administrative Support
## 4          13642    04 Information Technology & Telecommunications
## 5          10251    03           Clerical & Administrative Support
## 6          10251    03           Clerical & Administrative Support
##   full_time_indicator              career_level salary_range_from
## 1                   F                      <NA>             60740
## 2                   F Experienced (non-manager)             50598
## 3                   F Experienced (non-manager)             30683
## 4                   F Experienced (non-manager)             87203
## 5                   F               Entry-Level             32086
## 6                   F               Entry-Level             32086
##   salary_range_to salary_frequency                    work_unit posting_date
## 1          162014           Annual         Tech Talent Pipeline   01/26/2012
## 2           85053           Annual          BHHS Administration   12/09/2013
## 3           49707           Annual                      Appeals   06/26/2014
## 4          131623           Annual         Executive Management   11/19/2014
## 5           51981           Annual        Communicable Diseases   10/08/2014
## 6           51981           Annual Vital Statistics/Vital Recor   11/18/2014
##   posting_updated salary_from_adjusted salary_to_adjusted
```

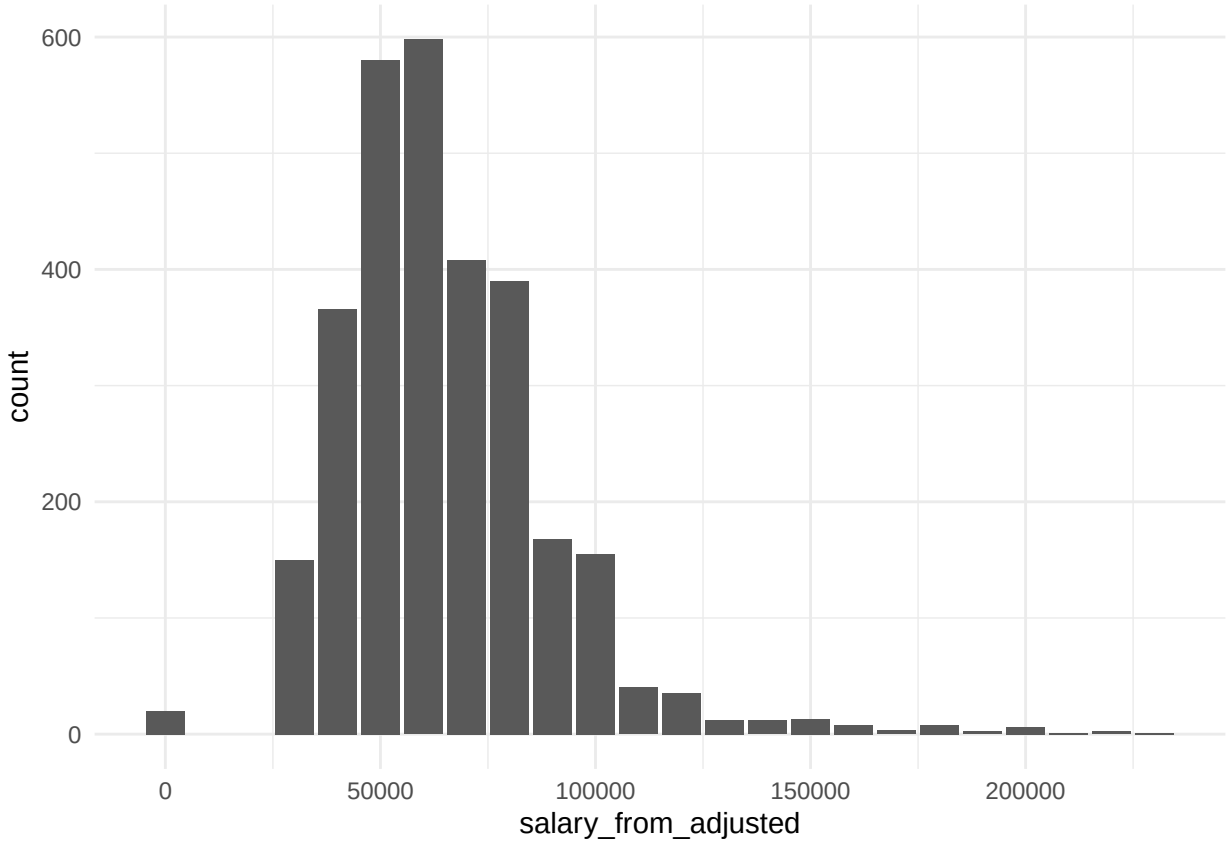
```
## 1      01/26/2012                60000             160000
## 2      12/09/2013                50000              90000
## 3      06/26/2014                30000              50000
## 4      11/19/2014                90000             130000
## 5      10/08/2014                30000              50000
## 6      11/18/2014                30000              50000
```

Now with this, how do the salary ranges look? Let's start with the beginning band (from).

```
count_by_salary_from <- nyc_jobs_basics |>
  group_by(salary_from_adjusted, salary_frequency) |>
  dplyr::summarise(count = n(),
                   percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

salary_from_bar <-ggplot(data=count_by_salary_from
                   , aes(x=salary_from_adjusted
                         , y=count)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Paired")+
  theme_minimal()

salary_from_summary <- nyc_jobs_basics |>
  dplyr::summarise(mean = mean(salary_from_adjusted),
                   median = median(salary_from_adjusted),
                   min = min(salary_from_adjusted),
                   max = max(salary_from_adjusted))

salary_from_summary
```

```
##       mean median min    max
## 1 65085.89  60000   0 230000
```

```
salary_from_bar
```

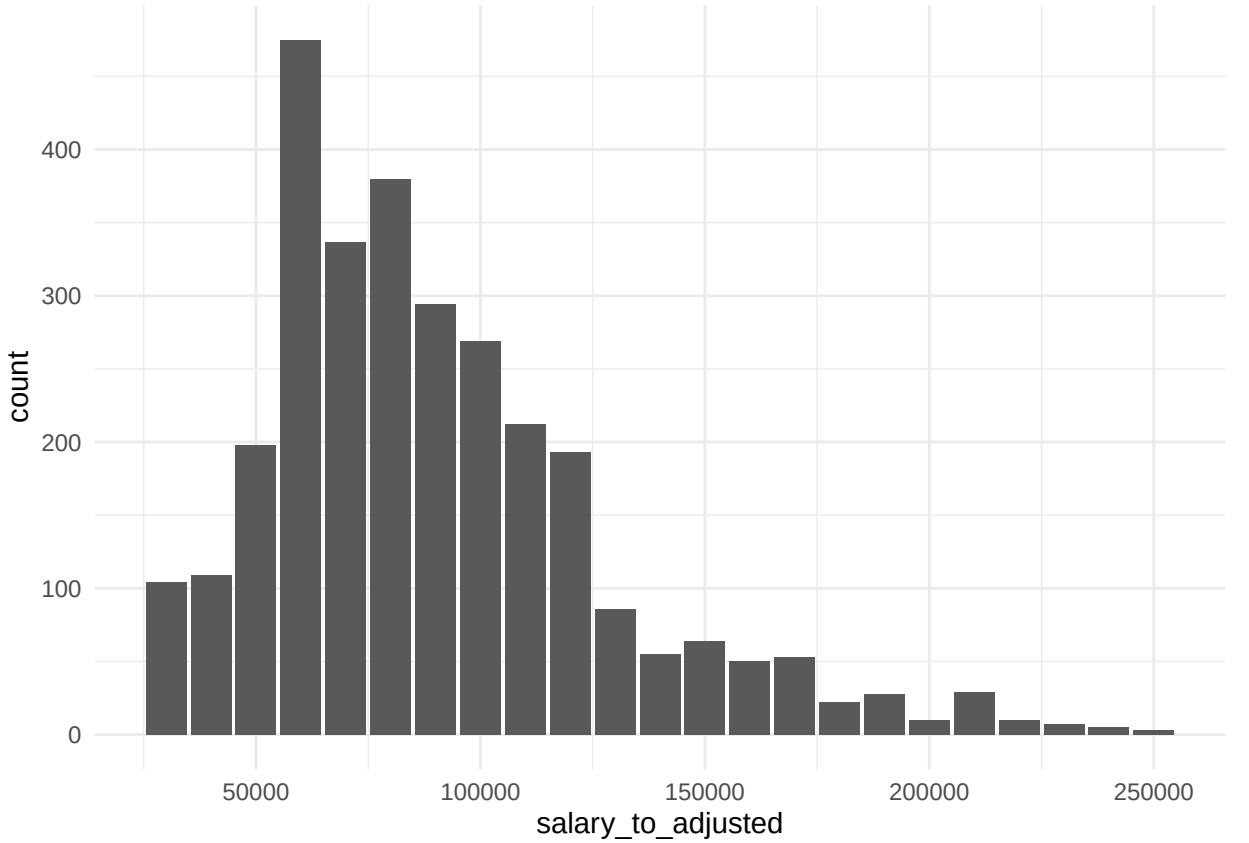


The data looks to be right skewed here with a center around ~$60k and a high of ~$230k. The mean and median are close but not quite the same so it's not a perfectly normal distribution. What about the to (high) portion of the salary range?

```
count_by_salary_to <- nyc_jobs_basics |>
  group_by(salary_to_adjusted, salary_frequency) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

salary_to_bar <-ggplot(data=count_by_salary_to
                    , aes(x=salary_to_adjusted
                          , y=count)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Paired")+
  theme_minimal()

salary_to_summary <- nyc_jobs_basics |>
  dplyr::summarise(mean = mean(salary_to_adjusted),
                   median = median(salary_to_adjusted),
                   min = min(salary_to_adjusted),
                   max = max(salary_to_adjusted))

salary_to_summary
```

```
##       mean median   min    max
## 1 88346.63  80000 30000 250000
```

```
salary_to_bar
```



This one looks less like a bell-curve (still right-skewed though) and seems to have a high frequency at ~$60k as well, however there seems to be more jobs with rates at the tail ends of the spectrum. We can see that the max here goes up to ~$250k with more activity in the ~$100k+ range. The mean and median here have been brought up though at $88k and $80k respectively while the minimum is also at $30k; in general, these rates pay more as they're the at the higher end of the spectrum.

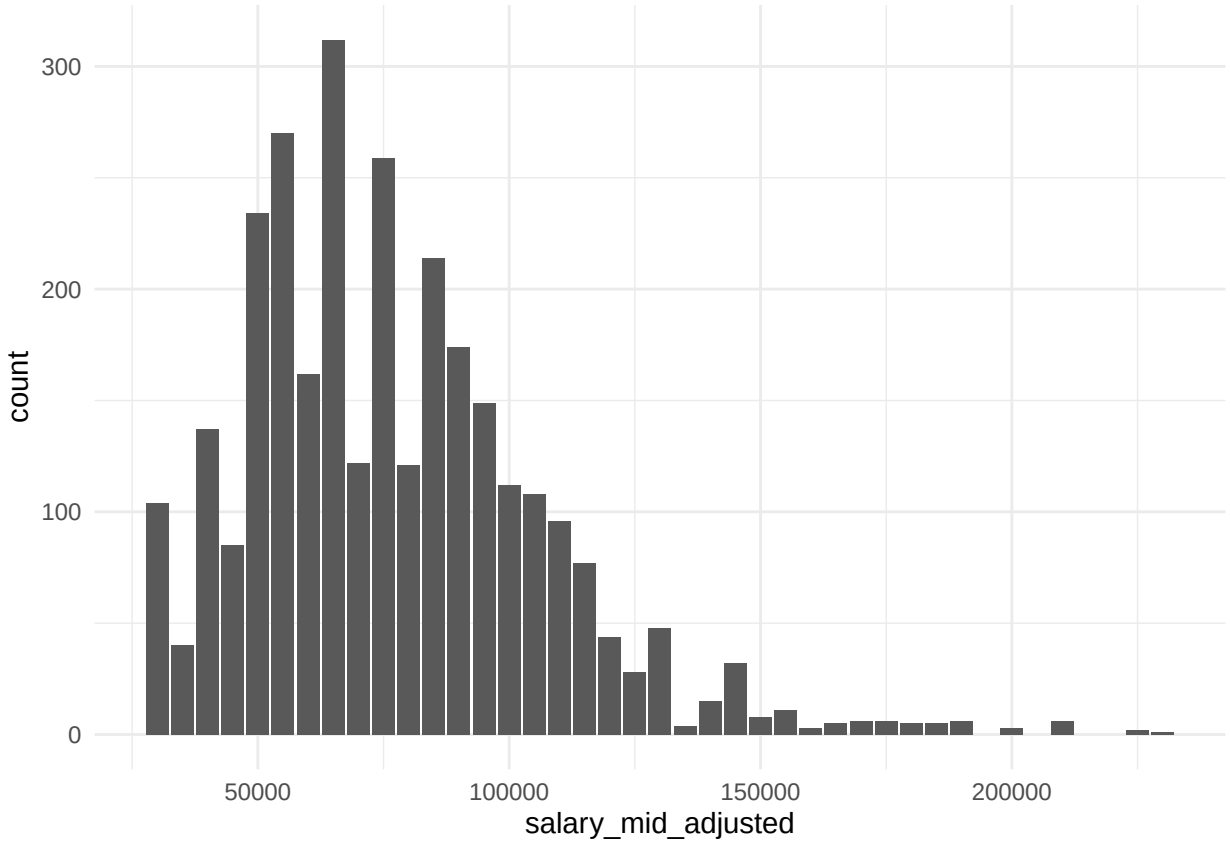
What does salary look like if we assume most folks get the middle point of each posted range?

```
nyc_jobs_basics$salary_mid_adjusted <- (nyc_jobs_basics$salary_to_adjusted + nyc_jobs_basics$salary_fro

count_by_salary_mid <- nyc_jobs_basics |>
  group_by(salary_mid_adjusted, salary_frequency) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

salary_mid_bar <-ggplot(data=count_by_salary_mid
                    , aes(x=salary_mid_adjusted
                          , y=count)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Paired")+
```

```
  theme_minimal()

salary_mid_summary <- nyc_jobs_basics |>
  dplyr::summarise(mean = mean(salary_mid_adjusted),
                   median = median(salary_mid_adjusted),
                   min = min(salary_mid_adjusted),
                   max = max(salary_mid_adjusted))

salary_mid_summary
```

```
##       mean median   min    max
## 1 76716.26  75000 30000 230000
```

```
salary_mid_bar
```



Using the mid-point, we can see that the curve now peaks at ~$65k. It's still a bit right-skewed with a tail trailing off at ~$230k as well while the mean and median now are closer at ~$\77k and ~$75k respectively..

Post Date

In this data set, when were these jobs posted?

```
nyc_jobs_basics$posting_date <- as.Date(nyc_jobs_basics$posting_date, "%m/%d/%Y")
```

```r
count_by_date <- nyc_jobs_basics |>
  group_by(year = format(posting_date, '%Y')) |>
  dplyr::summarise(count = n(),
            percent = 100 * n()/nrow(nyc_jobs_basics),
            .groups = 'drop') |>
  arrange(desc(count))

head(count_by_date, 10)
```

```
## # A tibble: 10 x 3
##    year  count percent
##    <chr> <int>   <dbl>
##  1 2023   1579  48.4
##  2 2022   1481  45.4
##  3 2021    103   3.16
##  4 2018     23   0.706
##  5 2019     19   0.583
##  6 2020     19   0.583
##  7 2016     15   0.460
##  8 2015      8   0.245
##  9 2017      7   0.215
## 10 2014      4   0.123
```

It looks like a majority of these job listings are from this year and last year, however some of them are even from 2014. For the most part though, it seems like they're mostly recent as 48% of them are from this year, but it's only been ~3.5 months as it's only March.