Factors Impacting Number of LinkedIn Job Applications per Post View

JRLK - Jessie Ringness, Rebekah Kim, Laura Cai, Karen Dong

2023-11-14

```
linkedin_subset <- linkedin_subset |>
  filter(between(hourly_max_salary, 19.31, 114.04)) |>
  filter(between(follower_count, 328.15, 3326613.00)) |>
  filter(views>5)
```

Introduction and data

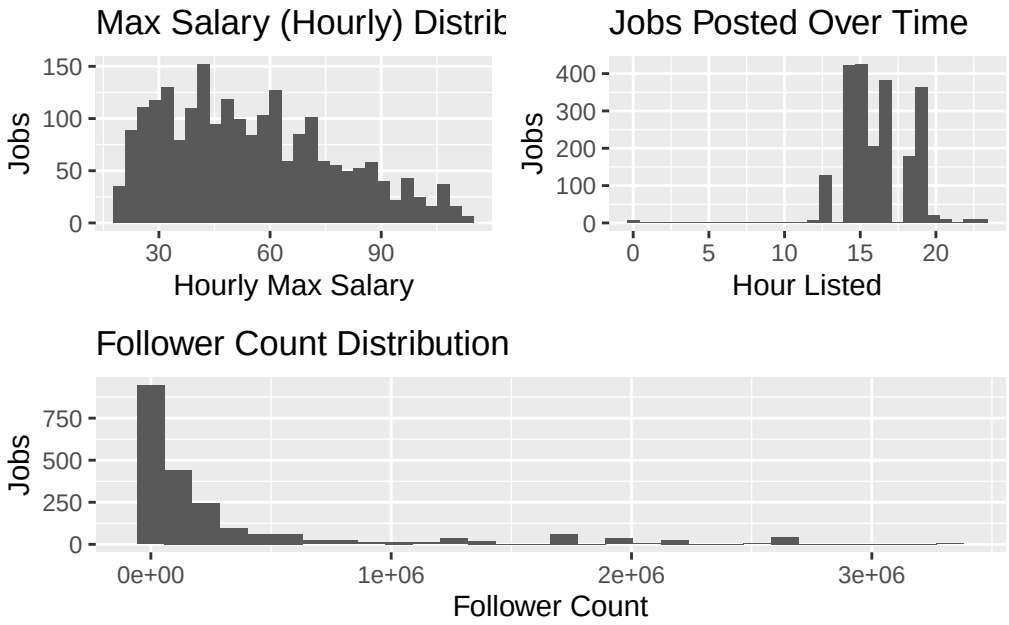
LinkedIn is a popular platform that connects companies and professionals spanning various levels of experience, and there are thousands of active job postings available on LinkedIn. Moreover, online job search services and platforms are now considered equally important for people to access a wide variety of opportunities compared to in-person job postings. With the sheer amount of postings, applicants may be overwhelmed by the vast amount of postings, and are less likely to come across some postings over others. Our primary research question is - what variables about job postings increase popularity among applicants?

This data set was created by Arsh Koneru-Ansari in July 2023, who used Python to scrape data directly from linkedin.com. The scraper code is published in their GitHub.

The data dictionary for the variable definitions can be found in the ReadMe for the data. The variables we will focus on are:

- **applies:** number of applications that have been submitted
- **views:** number of times the job posting has been viewed
- **max_salary:** maximum salary offered in the position
- **remote_allowed:** whether job permits remote work (1 = yes)
- **follower_count:** number of company followers on LinkedIn

- **`listed_time`:** time when the job was listed, in UNIX time
- **`formatted_experience_level`:** job experience level (entry level, associate, mid-senior level, director, executive, internship)
- **`type`**: type of benefit provided (Medical insurance, Dental insurance, 401(k), Paid maternity leave, Disability insurance, Vision insurance, Tuition assistance, Pension plan, Child care support, Commuter benefits, Student loan assistance)




```
sal_quantile<-quantile(linkedin_subset$hourly_max_salary, probs = c(0.05, 0.5, 0.95), na.r
follower_quantile<-quantile(linkedin_subset$follower_count, probs = c(0.05, 0.5, 0.95),na.
views_quantile<-quantile(linkedin_subset$views, probs = c(0.1,0.5, 0.95), na.rm = TRUE)
```

```
linkedin_subset_small <- linkedin_subset[, c(3,8,10)]
summary_stats <- summary(linkedin_subset_small)
print(summary_stats)
```

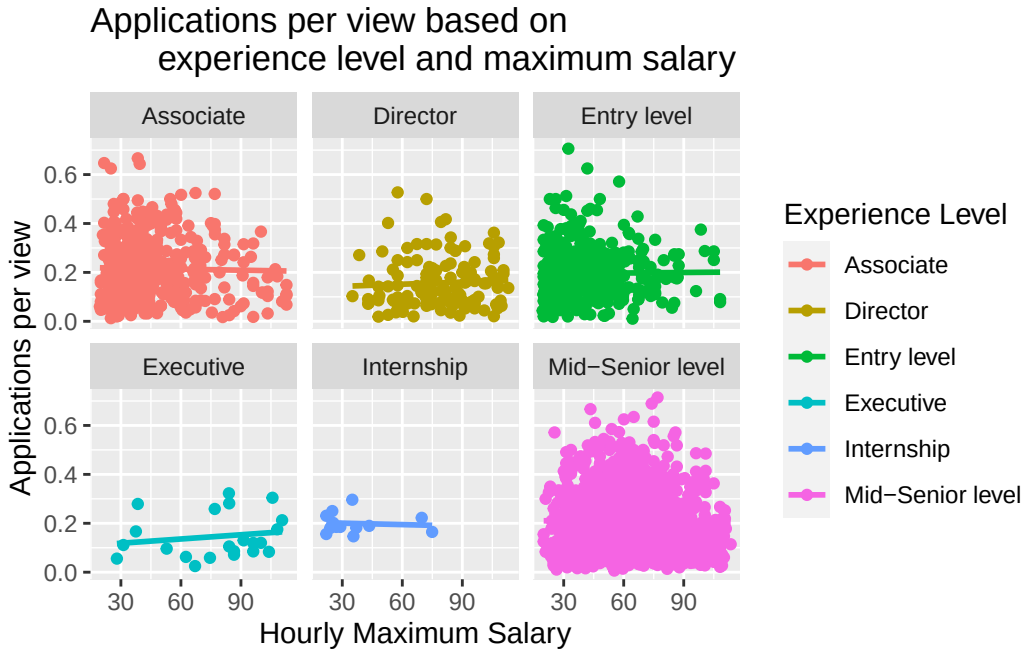
```
 hour_listed    follower_count    hourly_max_salary
Min.   : 0.00   Min.   :    333   Min.   : 19.50
1st Qu.:14.00   1st Qu.:  19296   1st Qu.: 36.06
Median :16.00   Median :  81743   Median : 52.88
Mean   :16.13   Mean   : 344736   Mean   : 55.08
3rd Qu.:18.00   3rd Qu.: 286286   3rd Qu.: 70.00
```

```
 Max.   :23.00    Max.   :3326613    Max.   :113.46
```

Fig 1.1. The distribution of hourly maximum salary is right skewed with its median around 50 dollars an hour and an IQR of 33 dollars an hour, demonstrating how the variability it relatively high. The histogram shows the middle 90% of the data to filter out the significant outliers.

Fig 1.2. The distribution of the hour listed is left skewed and is concentrated mainly between 2:00 PM and 7:00 PM, with a median around 4:00 PM and an IQR of 4 hours, showing how the variability is relatively high

Fig 1.3. The distribution of follower count is, with a median of around 70 thousand followers. The histogram shows the middle 90% of the data to filter out the significant outliers. The IQR is over 280,000 followers, which means there is high variability.




```
tapply(linkedin_subset$per_applies, linkedin_subset$formatted_experience_level, summary)
```

```
$Associate
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01282 0.11697 0.19017 0.21537 0.30047 0.66667

$Director
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
0.01887 0.08069 0.13636 0.15725 0.20840 0.52717

$`Entry level`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01075 0.11470 0.17752 0.19601 0.25266 0.70588

$Executive
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02500 0.08374 0.11534 0.14591 0.20313 0.32258

$Internship
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1467  0.1810  0.1852  0.1993  0.2222  0.2963

$`Mid-Senior level`
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.006579 0.111111 0.184624 0.207172 0.282971 0.714286
```

Fig 2. The distribution of applications per view based on experience level and maximum salary is mostly concentrated when the applications per view is less than 0.50 and the hourly maximum salary is less than 200 for each experience level. The mid-senior level has apparent outliers when the hourly maximum salary is greater than 200 and the NA level has outliers when the applications per view is above 0.75. There is also not as much data for some of the experience levels, including internship and executive.

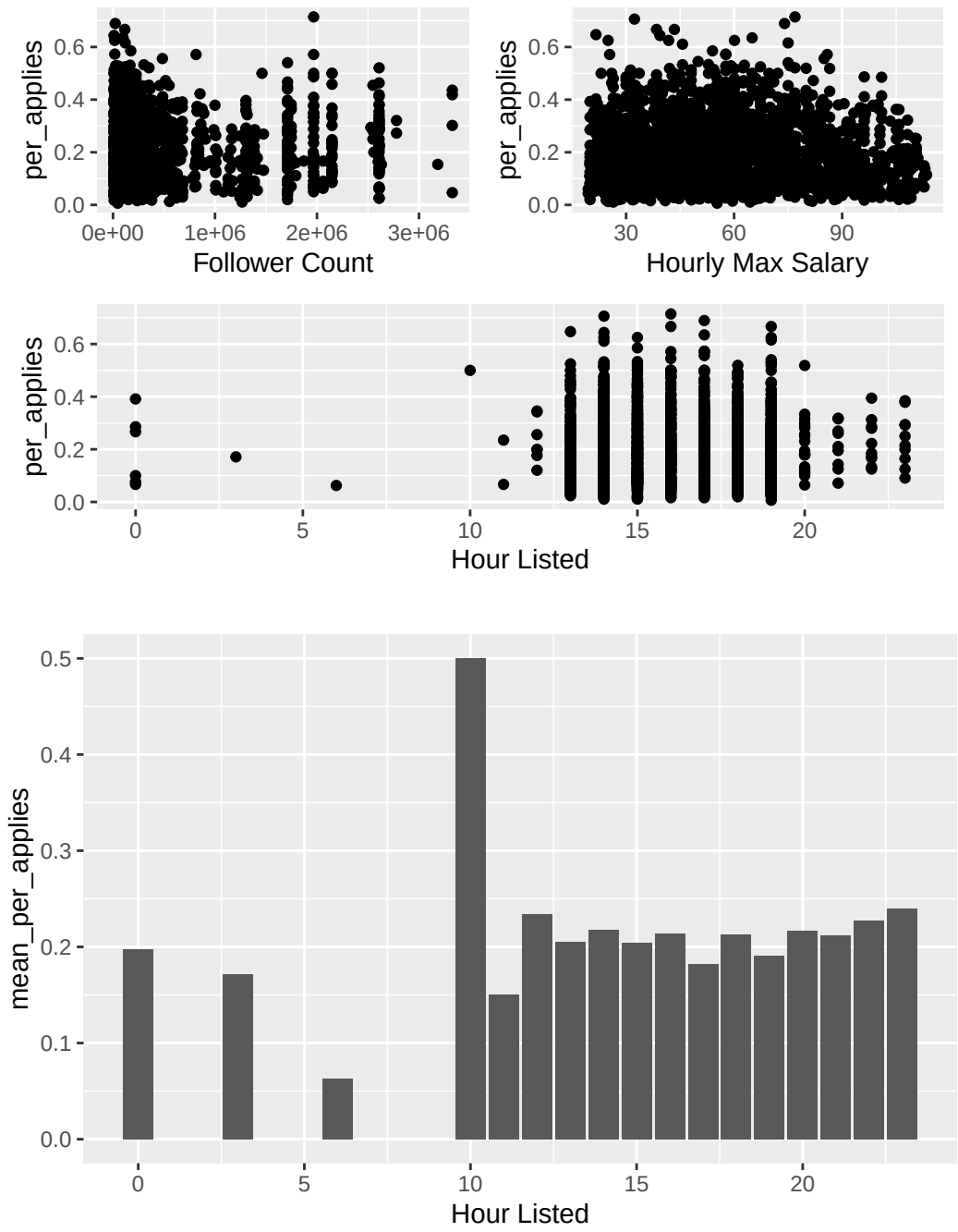


Fig 3.1. There is a slight linear correlation between follower count of a company and percentage of viewers who apply to the job. Most of the observations are concentrated to have less than 1,000,000 followers, so it would be beneficial to remove outliers that have more followers than that.

Fig 3.2. There is a slight positive linear correlation between a job's adjusted hourly maximum salary and percentage of viewers who apply to the job. Most of the observations are concentrated to have less than $200,000 for adjusted hourly max salary, so it would be beneficial to filter out outliers that are above this threshold. Additionally, it would be beneficial to remove the outlier where 100% of viewers applied to the job (per_applies = 1.0)

Fig 3.3. Convert to categorical variable, make a histogram that's colored by morning, afternoon, evening

Fig 3.4. This figure could be statistically misleading since the intervals are not consistent, and mean_per_applies is not a helpful response variable for the model. We should convert hour_listed to a categorical variable and create a pie chart instead to show the frequency of each level.

Methodology

~~ We are provided with data regarding benefits, where it listed types of benefits provided for each company, and data regarding job postings, which lists the posted jobs scraped from LinkedIn on July 23 and 24, 2023. To prepare our data for our model: We counted the total benefits offered by each company to prepare out "if_benefits" predictor. We joined the datasets to use the benefits predictor with other predictors. We made the assumption that if there were NAs for "remote_allowed" and "tot_benefits" for jobs/companies that didn't allow remote and had no listed benefits. Therefore, we imputed them to 0's. We filtered out the other values that were NA. We created an "hourly_max_salary" predictor variable to compare the salaries even if they were listed as monthly or annually. To use the data relating to the time the job was posted, we converted the time format to EST time and kept only kept the hour. To remove significant outliers that may affect the model and its precision, we filtered to keep only the middle 90% of data for the "hourly_max_salary" and "follower_count" variables. In order to generalize our results to jobs with not too few views, we filtered out the jobs that had less than 5 views. ~~

While the bulk of our data was found in the 'job_postings' data set, we also wanted to include employee count and follower count data in the 'employee' data set and the type and number of benefits listed on the post found in the 'benefits' data set. We needed to manipulate the data in 'benefits' to create a useful predictor as the majority of the data was NA, meaning no benefits could be found from the scraped data. To make the data from 'benefits' a useful predictor, we created a new categorical variable called ''if_benefits' where if a benefit (such as paid maternity leave or a 401k plan) was listed on the post, then the post was considered as having benefits 'listed', and otherwise was listed as 'none' listed. We joined all data sets together by 'company_id', and saved the data set as 'linkedin'.

We also made some assumptions about other variables to normalize predictor variables. The categorical variable 'pay_period' contained data on when the job would pay its worker the

‘max_salary‘ or ‘min_salary‘ amount, with hourly, monthly, and annual payments as the different levels. To normalize the ‘max_salary‘ amount, we calculated the hourly wage given the maximum pay for hourly, monthly, and yearly pay periods. We assumed 160 hours for the monthly payments (40 hour work week for 4 weeks), and 2080 hours for the annual payments (40 hour work week for 52 weeks). We saved the new data in a variable called ‘hourly_max_salary‘.

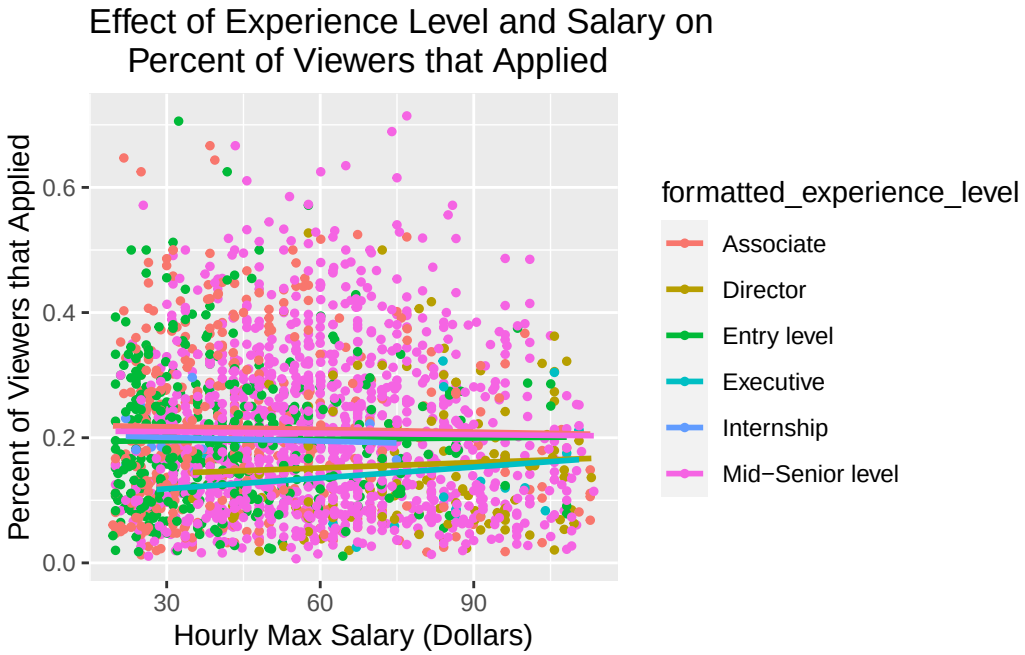
We then dropped all ‘NA‘ values for all predictors we wanted to observe so that we could keep our dataset consistent when testing different models, meaning NA values for ‘hourly_max_salary‘, ‘per_applies‘, ‘follower_count‘, ‘formatted_experience_level‘, ‘original_listed_time‘, and ‘remote_allowed‘.

Because the number of views an application gets is directly related to the number of applications and the jobs have been listed for varying durations of time, we decided to normalize the number of applications with the views. To do so, we created a new variable ‘per_applies‘. We then used ‘per_applies‘ as our response variable.

Because ‘per_applies‘ is a numerical variable, a linear regression model would be most appropriate to predict the number of applications per view. As we addressed in the introduction, a person takes into consider many different factors when applying to a job, so our model takes into consideration multiple predictors, including the hour the job was posted, the number of followers the company has, the job experience level, the maximum salary, ability to work remote, and if benefits are listed.

Checking interaction effect:

```
linkedin_subset |>
  ggplot(aes(x = hourly_max_salary, y = per_applies, color = formatted_experience_level))
  geom_point(size = 1) +
  geom_smooth(method = "lm", formula = y~x, se = F) +
  labs(
    x = "Hourly Max Salary (Dollars)",
    y = "Percent of Viewers that Applied",
    title = "Effect of Experience Level and Salary on
    Percent of Viewers that Applied"
  )
```



We split the 'linkedin' dataset into training and testing data, with 75% of the data in training and 25% in testing. We then used cross-fold validation with 12 folds on the training data set to find the mean summary statistics (AIC, BIC, Adjusted R-Squared) for each model and compared the different values to find the best possible model. This process was repeated for models containing each combination of predictor variables. We set a seed of (2) when splitting and folding the data to ensure reproducibility.

```
set.seed(2)
linkedin_recipe3 <- recipe(per_applies ~ job_id + formatted_experience_level +
                             hour_listed + remote_allowed + hourly_max_salary,
                           data = linkedin_train) |>
  step_interact(terms = ~ hourly_max_salary:remote_allowed) |>
  update_role(job_id, new_role = "ID") |>
  step_dummy(all_nominal_predictors()) |>

  step_zv(all_predictors())
```

```
set.seed(2)
calc_model_stats <- function(x) {
  glance(extract_fit_parsnip(x)) |>
    select(adj.r.squared, AIC, BIC)
```

```
}
```

```
map_df(linkedin_fit_rs_full$.extracts, ~ .x[[1]][[1]]) |>
  summarise(mean_adj_rsq = mean(adj.r.squared),
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1       0.0246   -2100.   -2036.
```

```
map_df(linkedin_fit_rs_red$.extracts, ~ .x[[1]][[1]]) |>
  summarise(mean_adj_rsq = mean(adj.r.squared),
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1       0.0218   -2098.   -2051.
```

```
map_df(linkedin_fit_rs_red_inter$.extracts, ~ .x[[1]][[1]]) |>
  summarise(mean_adj_rsq = mean(adj.r.squared),
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1       0.0216   -2096.   -2038.
```

```
linkedin_fit_full <- linkedin_wflow1 |>
  fit(data = linkedin_test)

tidy(linkedin_fit_full) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.281	0.037	7.553	0.000
hourly_max_salary	0.000	0.000	0.098	0.922
follower_count	0.000	0.000	1.972	0.049
hour_listed	-0.004	0.002	-1.936	0.053
remote_allowed_yes	0.029	0.012	2.325	0.020
formatted_experience_level_Director	-0.067	0.027	-2.513	0.012
formatted_experience_level_Entry.level	-0.020	0.017	-1.192	0.234
formatted_experience_level_Executive	-0.090	0.071	-1.278	0.202
formatted_experience_level_Internship	-0.027	0.061	-0.448	0.654
formatted_experience_level_Mid.Senior.level	-0.012	0.014	-0.843	0.400
if_benefits_listed	-0.011	0.011	-0.998	0.319

The adjusted R^2 value for a model including `hourly_max_salary`, `follower_count`, `remote_allowed`, `formatted_experience_level`, `hour_listed`, and `if_benefits` was 0.0246, and the AIC and BIC was -2099.724 and -2036.034, respectively.

Using a significance level of $\alpha = 0.10$, `follower_count`, `hour_listed`, `remote_allowed`, and if the job requires director level of experience are the only statistically significant variables, with p-values of 0.049, 0.053, 0.020, and 0.012 respectively.

Then, we made a reduced model with these statistically significant variables including `hour_listed`, `follower_count`, `remote_allowed`, and `formatted_experience_level` to compare models. The adjusted R^2, AIC, and BIC were 0.0241, -2100.996, and -2047.921, respectively. Although the adjusted R-squared is lower for the reduced more, signfiying potentially a worse model, the AIC and BIC were lower than those of the full model, indicating that the reduced model is a stronger model. Since the adjusted R-squared for both models aren't significantly different, we decided that the reduced model would work better because of the lower AIC and BIC.

Lastly, we included a model with the variables from the reduced model, plus `hourly_max_salary` to explore the potential interaction effect between `hourly_max_salary` and `remote_allowed`, which we identified earlier has a potential interaction term. The adjusted R^2, AIC, and BIC are 0.0216, -2096.261, and -2037.879. Since this model with the interaction term has a lower adjusted R^2 and higher AIC and BIC, we concluded that the reduced model with `hour_listed`, `follower_count`, `remote_allowed`, and `formatted_experience_level`as predictors works the best.

```
library(rms)
linkedin_full_fit <- linear_reg() |>
  set_engine("lm") |>
  fit(per_applies ~ hourly_max_salary + follower_count + remote_allowed + formatted_experi
```

```
vif(linkedin_full_fit$fit)
```

```
                          hourly_max_salary
                                   1.326296
                             follower_count
                                   1.008260
                          remote_allowedyes
                                   1.032291
        formatted_experience_levelDirector
                                   1.406394
     formatted_experience_levelEntry level
                                   1.617741
       formatted_experience_levelExecutive
                                   1.071177
      formatted_experience_levelInternship
                                   1.027259
formatted_experience_levelMid-Senior level
                                   1.888772
                          if_benefitslisted
                                   1.006216
                                hour_listed
                                   1.030193
```

Multicollinearity occurs where there are very high correlations among two or more predictor variables, and we need to check for multicollinearity because it causes a loss in precision in our estimates of the regression coefficients. It is a concern when VIF is greater than 10 for a predictor, which is not the case in our model.

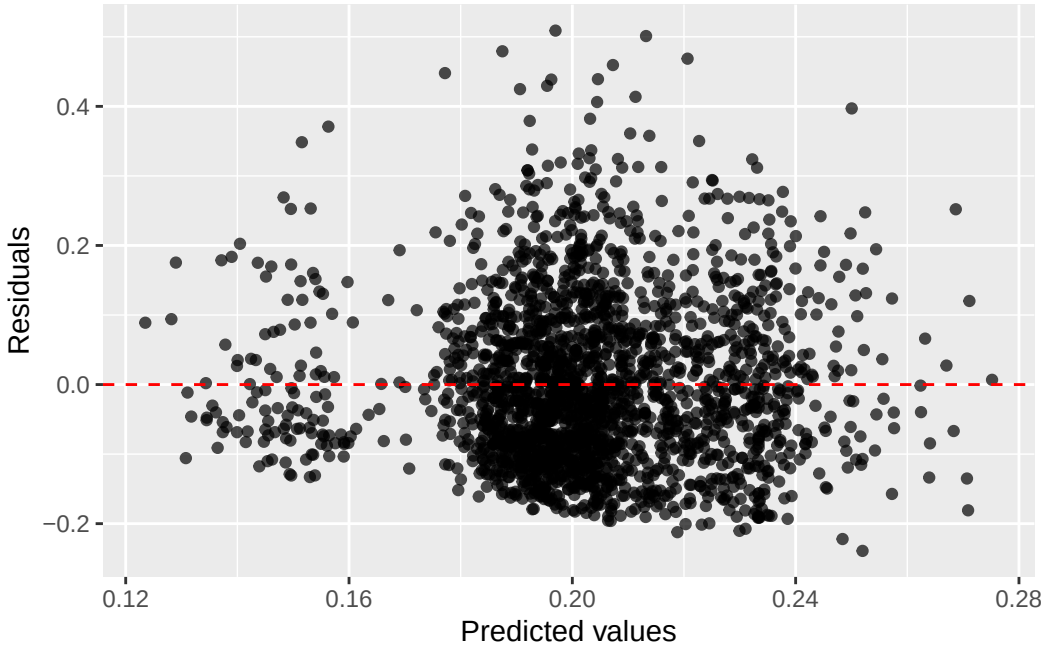
Conditions

We need to check linearity, constant variance, normality, and independence as conditions for inference.

Linearity

```
linkedin_full_aug <- augment(linkedin_full_fit$fit)
```

```
ggplot(data = linkedin_full_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(x = "Predicted values", y = "Residuals")
```



Results

```
linkedin_yr_rec <- recipe(per_applies ~ job_id + hourly_max_salary + follower_count + remo
  update_role(job_id, new_role = "ID") |>
  step_naomit(all_predictors()) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors())
```

```
#specify the model
linkedin_yr_spec <- linear_reg() |>
  set_engine("lm")

#build model workflow
linkedin_yr_workflow <- workflow() |>
  add_model(linkedin_yr_spec) |>
  add_recipe(linkedin_yr_rec)

# fit the model
linkedin_yr_fit <- linkedin_yr_workflow |>
  fit(data = linkedin_subset)
```

```
tidy(linkedin_yr_fit) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.249	0.019	12.825	0.000
hourly_max_salary	0.000	0.000	-1.009	0.313
follower_count	0.000	0.000	2.996	0.003
hour_listed	-0.002	0.001	-1.981	0.048
remote_allowed_yes	0.034	0.006	5.282	0.000
formatted_experience_level_Director	-0.051	0.013	-4.039	0.000
formatted_experience_level_Entry.level	-0.016	0.008	-1.859	0.063
formatted_experience_level_Executive	-0.063	0.026	-2.375	0.018
formatted_experience_level_Internship	-0.011	0.034	-0.318	0.750
formatted_experience_level_Mid.Senior.level	-0.006	0.007	-0.921	0.357
if_benefits_listed	-0.007	0.005	-1.379	0.168

```
glance(linkedin_yr_fit)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik    AIC    BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl>  <dbl>  <dbl>
1    0.0323        0.0278 0.119      7.20 2.89e-11    10  1544. -3064. -2996.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Looking at the how the maximum salary, follower count, whether the job allows remote, the experience level, if there were benefits listed, and the hour at which the job was listed all affected the ratio of applications to views, there is no significant relationship between these variables, since the adjusted R^2 s 0.0589, which is very low. This means that about 6% of the variation in the percent of viewers that applied is a result of the variation in the predictor variables we are studying, indicating no relationship.We tried different combinations of variables to see if there was any relationship between the variables with percent of viewers that applied.

Discussion + Conclusion

While we had expected the percent of viewers that applied for each job to increase as the maximum salary and number of followers increase, our model and EDA have demonstrated how there is no relationship between these variables, along with having a remote option, the

job experience level, the hour at which it was posted, and if the benefits were posted, and our response variable of the percent of viewers who applied.

Some of the limitations with our analysis may include assumption errors with the number of hours the job would work for the jobs listed as yearly and monthly, the filtering of outliers and deciding the thresholds for outliers, model complexity where we are fitting many predictor variables, and the lack of data considering how long the jobs were listed.

Some ways our analysis could be improved would be filtering out the outliers better, deciding the thresholds for views, follower count, salaries, etc, that would avoid skewing the data.

Potential issues relating to the data would include the time scraped from LinkedIn, since many of the original listed times and the listed times (the time it was scraped) is exactly the same, which is not meaningful in determining the number of applications over a certain amount of time.