R AND TABLEAU

PREDICTING EMPLOYEE CHURN

A data driven analysis to help design employee retention strategies

19073155

WORD COUNT : 1997

*EXCLUDING FIGURE LABELS

The Problem

The growth of an organization depends on employee retention. Employee turnover/attrition/churn refers to an employee ending their relationship with an organization. Over the last few years, the problem of employee churn has intensified, with ~one-quarter of all U.S. workers quitting their jobs every year[i]. Society of Human Research Management (SHRM) estimates that replacing an employee can cost between 50-60% of that employee's salary with costs ranging anywhere from 90% to 200%.[ii]

Turnover may be involuntary (employee is let go by the organization due to contractual obligations, business restructuring and disciplinary actions) or voluntary (employee chooses to resign, for reasons like better opportunities, negative working environment, health, etc). The HR department benefits from a data-driven analysis of key factors driving employee attrition. To design employee retention strategies, they want to use data to answer questions like "Which employee is likely to leave us? Why?" and helps with business decision, "What can we do to prevent them from leaving?"

This analysis will aim to predict whether an employee will churn and identify the potential reasons for, as well as early warning signals before doing so.

The data was found on Datacamp[iii], and was created for the purpose of analyzing employee churn in 2015. Despite not being "real" data, it is representative of organizational HR data and can be deemed reliable. Target attribute is a categorical variable *Status*, "Active" indicating that an employee has stayed within the organization, "Inactive" indicating that they have churned. Scope of this analysis is limited to voluntary churn; HR likely already knows why an involuntary attrition took place.

Understanding the Data

The size of our dataset is formed by 1954 records and 34 attributes. There is no missing data. It is open source. 80% of the observations represent active employees as opposed to 20% of inactive employees: the data is imbalanced.

"turnover" is an int binary variable which exactly reflects employee status. Where Status == "active", *turnover* = 0. For inactive employees, *turnover* = "1". It is useful to have this available, as it helps with numeric calculations, correlations, and predictions.

Date columns in string format were converted to dates using the dmy() function from the lubridate package.

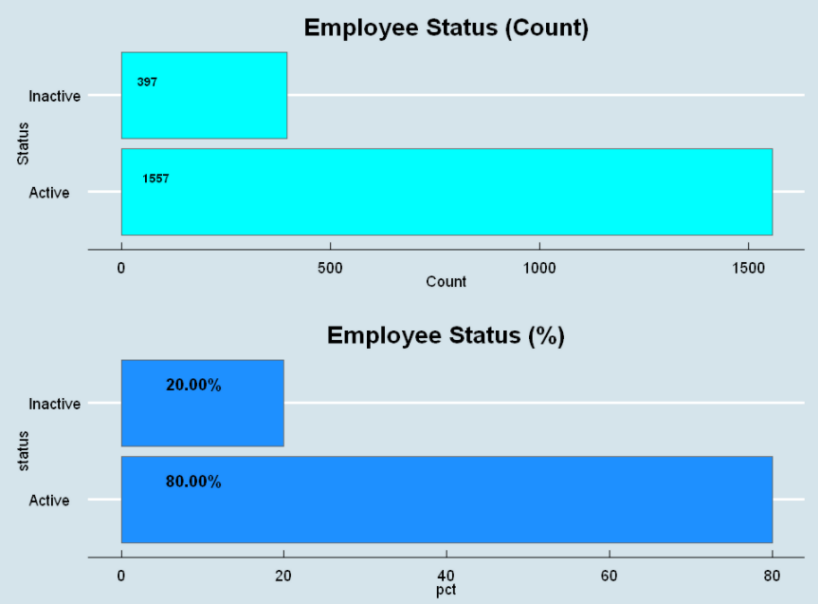


Figure 1: Composition of imbalanced dataset

Exploratory Data Analysis (EDA) was conducted to understand the distribution of attributes, especially categorical ones which cannot be represented through a correlation plot.

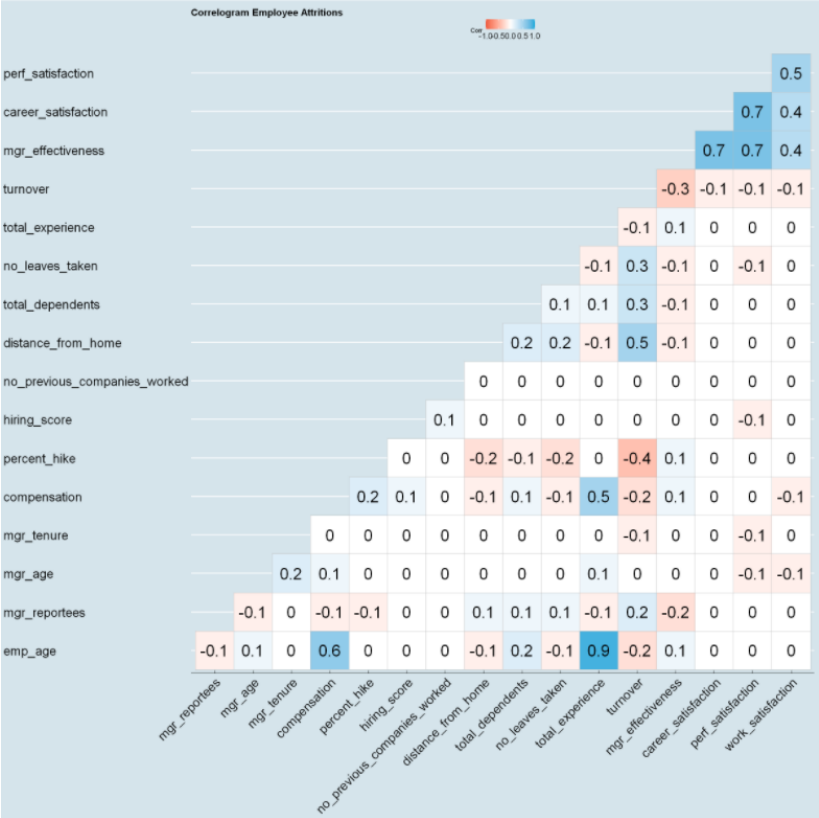


Figure 2: Correlation plot of variables

EDA and Correlation plot helped select the relevant attributes for the analysis summarized below.

Level

It was important to see which job role had the highest attrition. *Turnover_rate* (derived) is the percentage of employees who left from each level. It is the mean of the turnover variable for each level. Managers have the highest turnover rate (26%), followed by analysts.

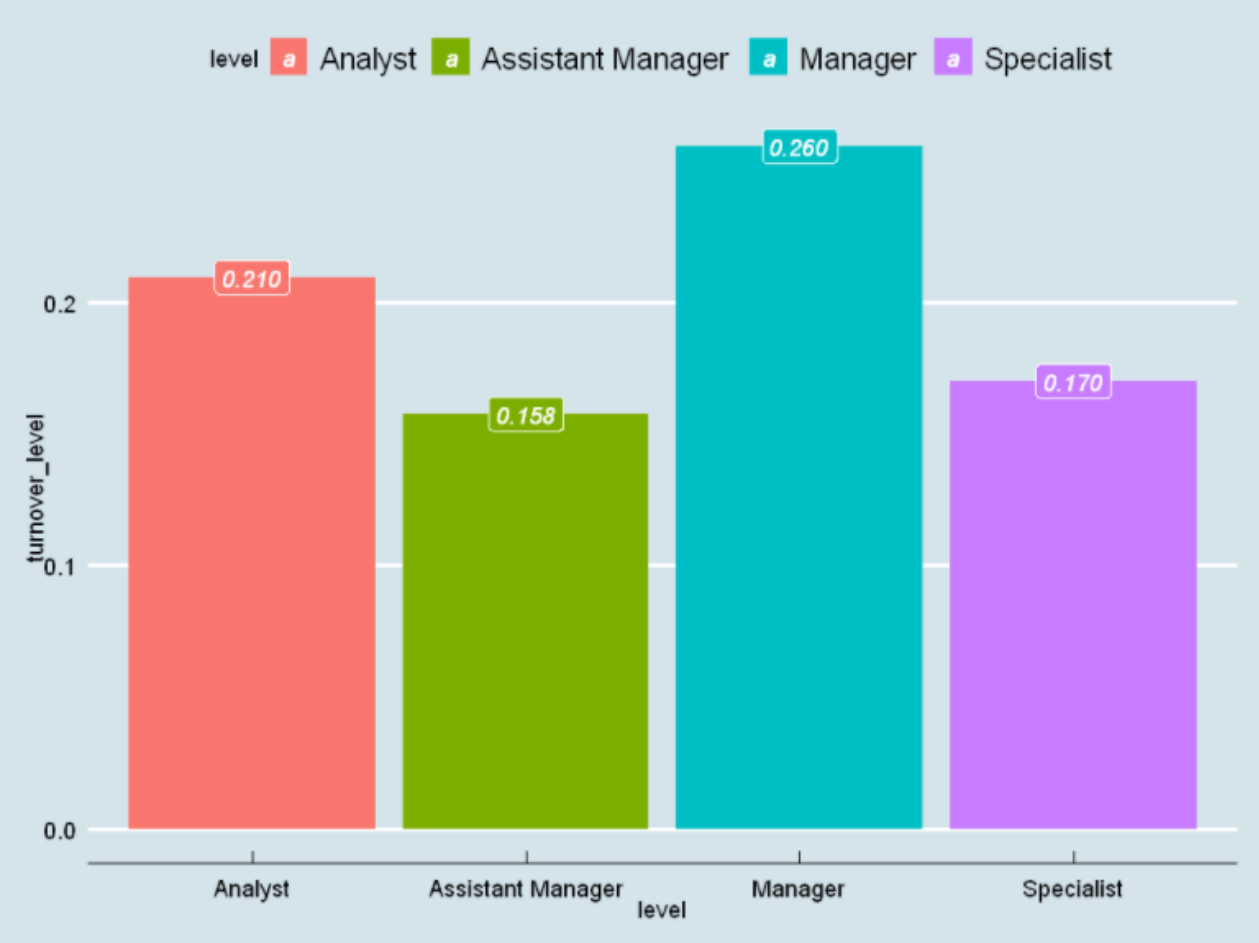


Figure 3:Level wise turnover

Location

Turnover_rate was visualized by location, to see if any location has particularly high attrition. New York has the highest level of turnover (37.5%), followed by Chicago.

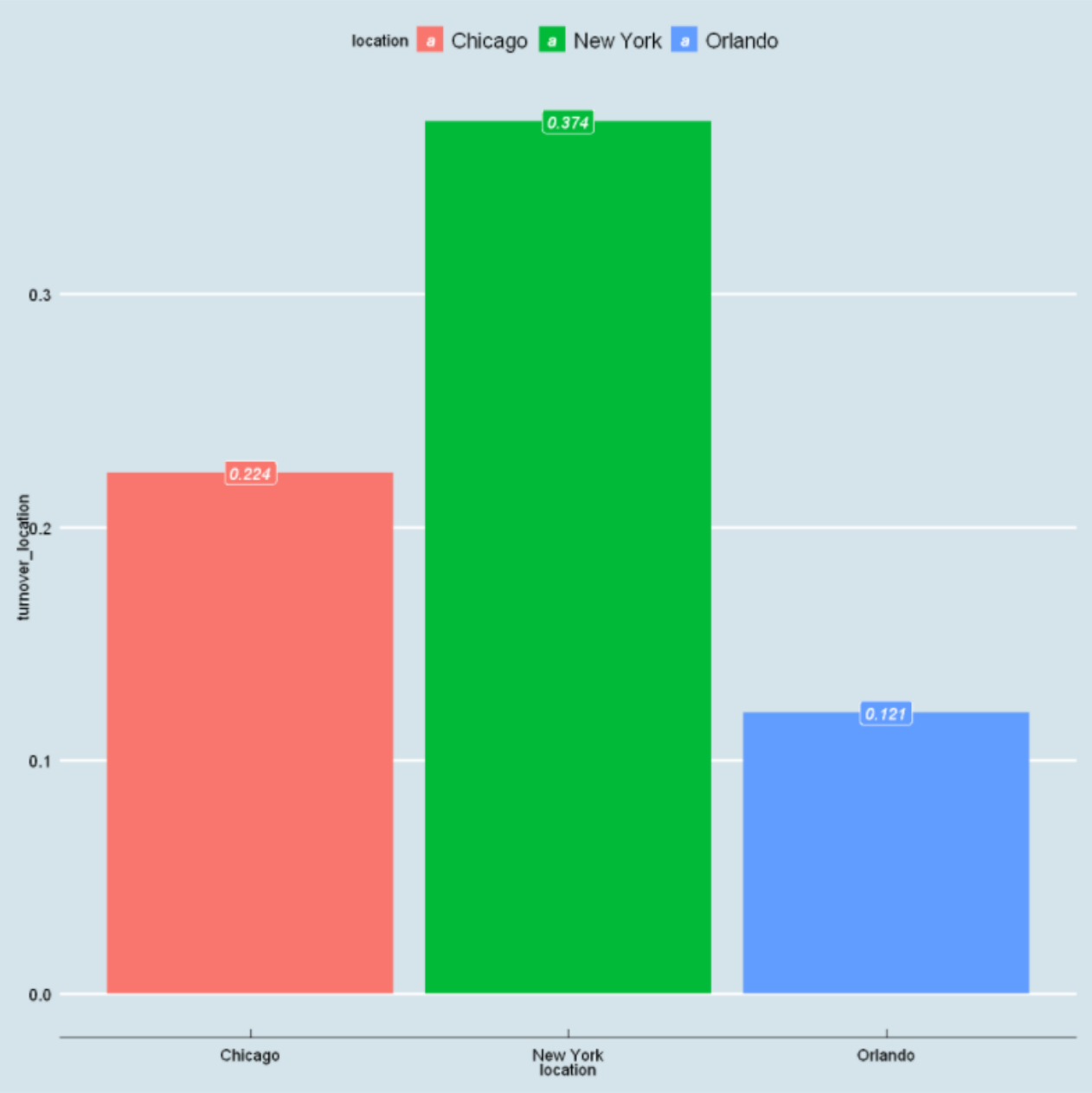


Figure 4: Location-wise turnover

Distance from Home

Inactive employees across all job levels and locations travel significantly longer distances as shown in Figure 5 and 6.

Figure 5: Boxplot showing distribution of distance_from_home

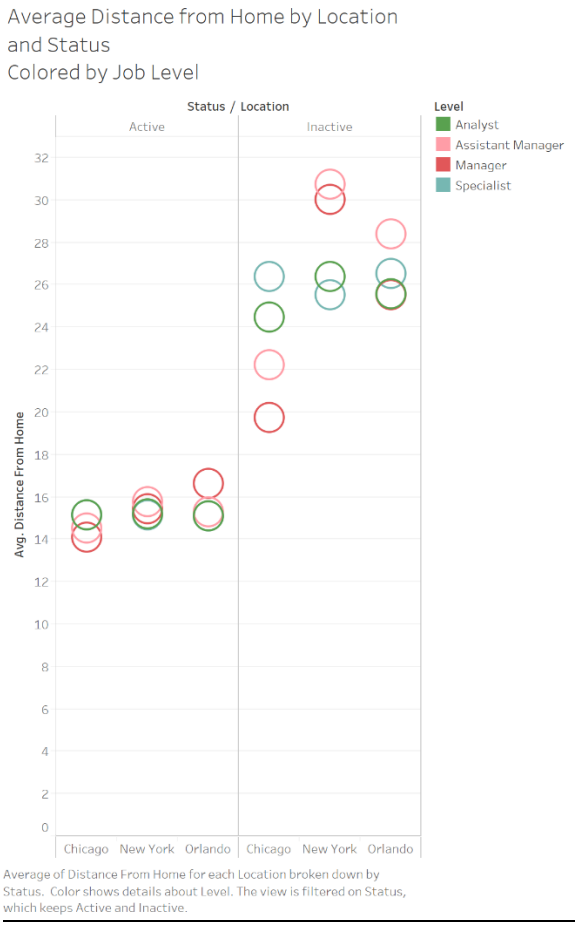


Figure 6: Average Distance for each location by status and job level

Work, Performance and Satisfaction Scores

Scores submitted by employees were compared across levels, as these could be Early Warning Signals of employee churn.

Work satisfaction scores across all levels are generally low. Performance satisfaction among inactive analysts and specialists are lower than other roles. Except for managers, inactive employees all other job roles seem to have been dissatisfied with their career progression.

Figure 7: Average performance satisfaction scores

Figure 8: Average work satisfaction scores

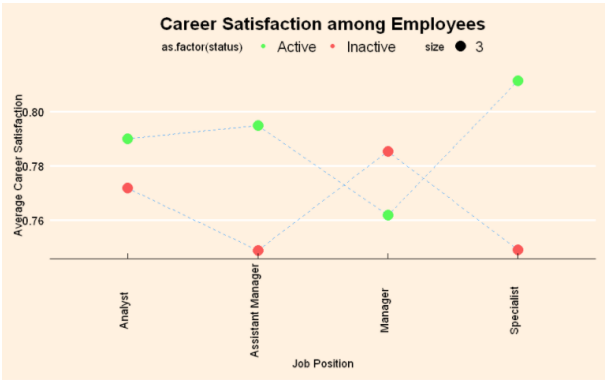


Figure 9: Average Career Satisfaction Scores

Financial Rewards (Compensation, Compa Level (Derived) , Percent Salary Hike and Promotion)

50% of the sample population has a salary hike of less than 10% and a compensation of less than 60,000 USD. In figure 10, most attritions belong to the lower left of the scatterplot, indicating lower percent hike and compensation. This is reaffirmed by the considerably lower compensation values for inactive employees, going as low as 43,350 for Managers, a senior position within the organization (fig 13).

Inactive employees have an average of about 7% percent hike, significantly lower than their active counterparts (fig 11).

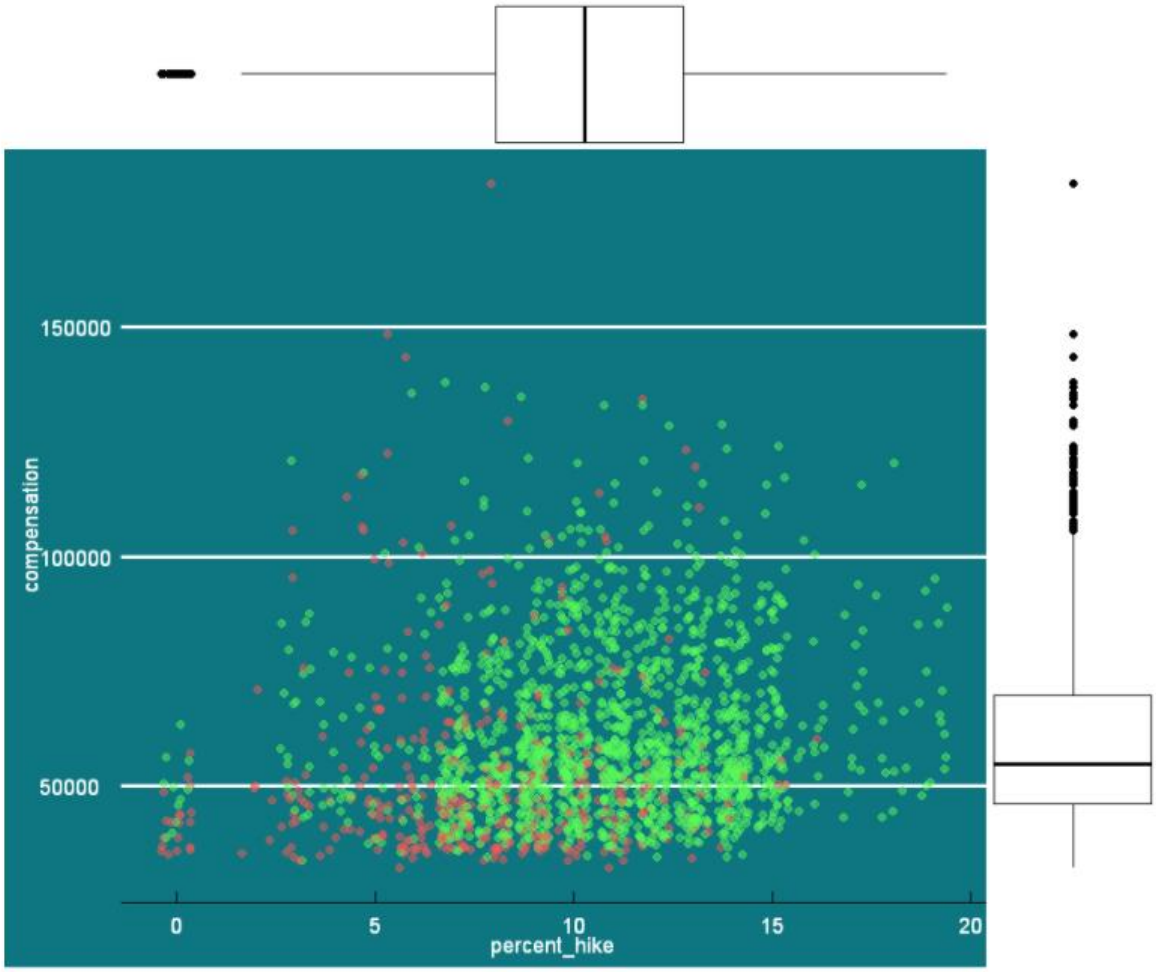


Figure 10: Compensation and Percent Hike

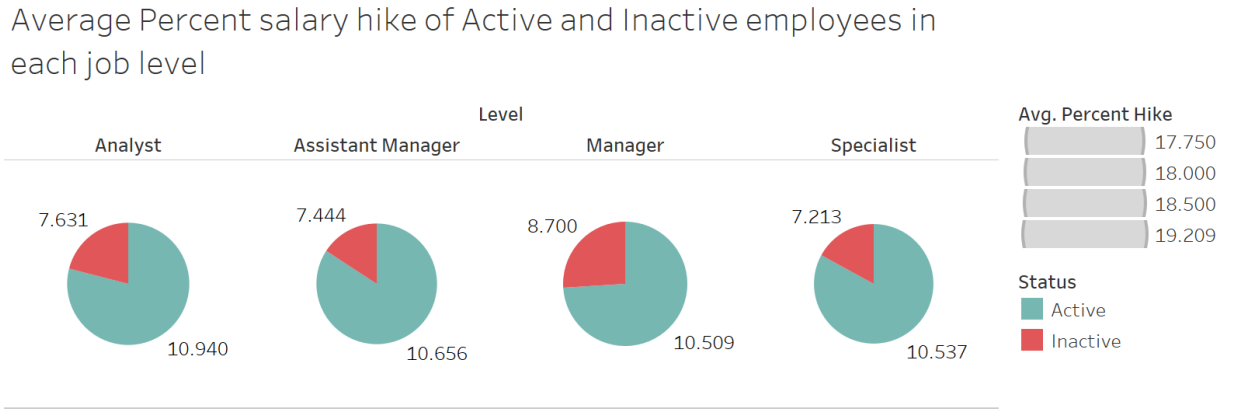


Figure 11

Due to variation within each level, it is important to derive a new variable which provides a direction to understand how an employee is paid compared to the other. *compa_ratio* was derived by dividing actual compensation by median compensation. Employees whose *compa-ratio* is more than one was classified as "Above" and "Below" otherwise. Almost 75% of inactive employees fall below the median compensation level.

25% of employees who did not receive a promotion in the last two years churned.

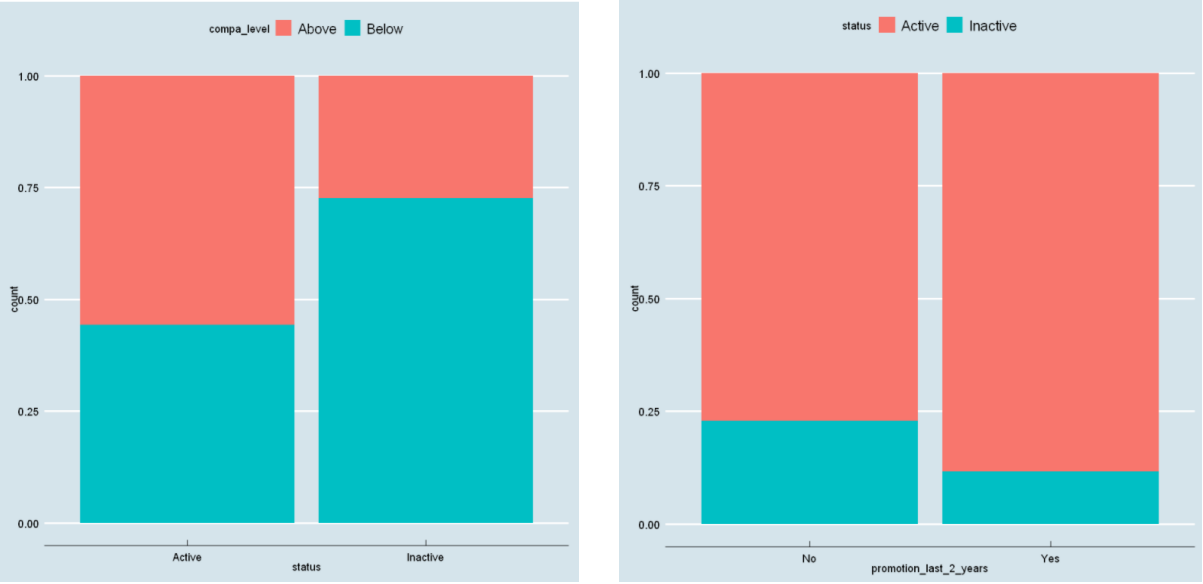


Figure 12: Compa_level and promotion_last two years by status

Average compensation by Location and Job Level

Status	Level	Location: Chicago	Location: New York	Location: Orlando
Active	Analyst	56,639	68,532	64,075
	Assistant Manager	56,348	57,071	63,924
	Manager	56,250	67,800	65,866
	Specialist	54,860	61,566	65,627
Inactive	Analyst	48,593	56,909	55,296
	Assistant Manager	51,724	46,470	47,314
	Manager	53,022	60,120	43,350
	Specialist	44,524	60,594	62,754

Avg. Compensation
43,350 68,532

Average of Compensation broken down by Location vs. Status and Level. Color shows average of Compensation. The marks are labeled by average of Compensation.

Figure 13

Monthly Overtime Hours

Inactive employees have larger bubbles (i.e., greater number of hours), and there also seems to be a group of people working ~12 hours overtime, while rating their Managers "unacceptable". This is for HR to investigate.

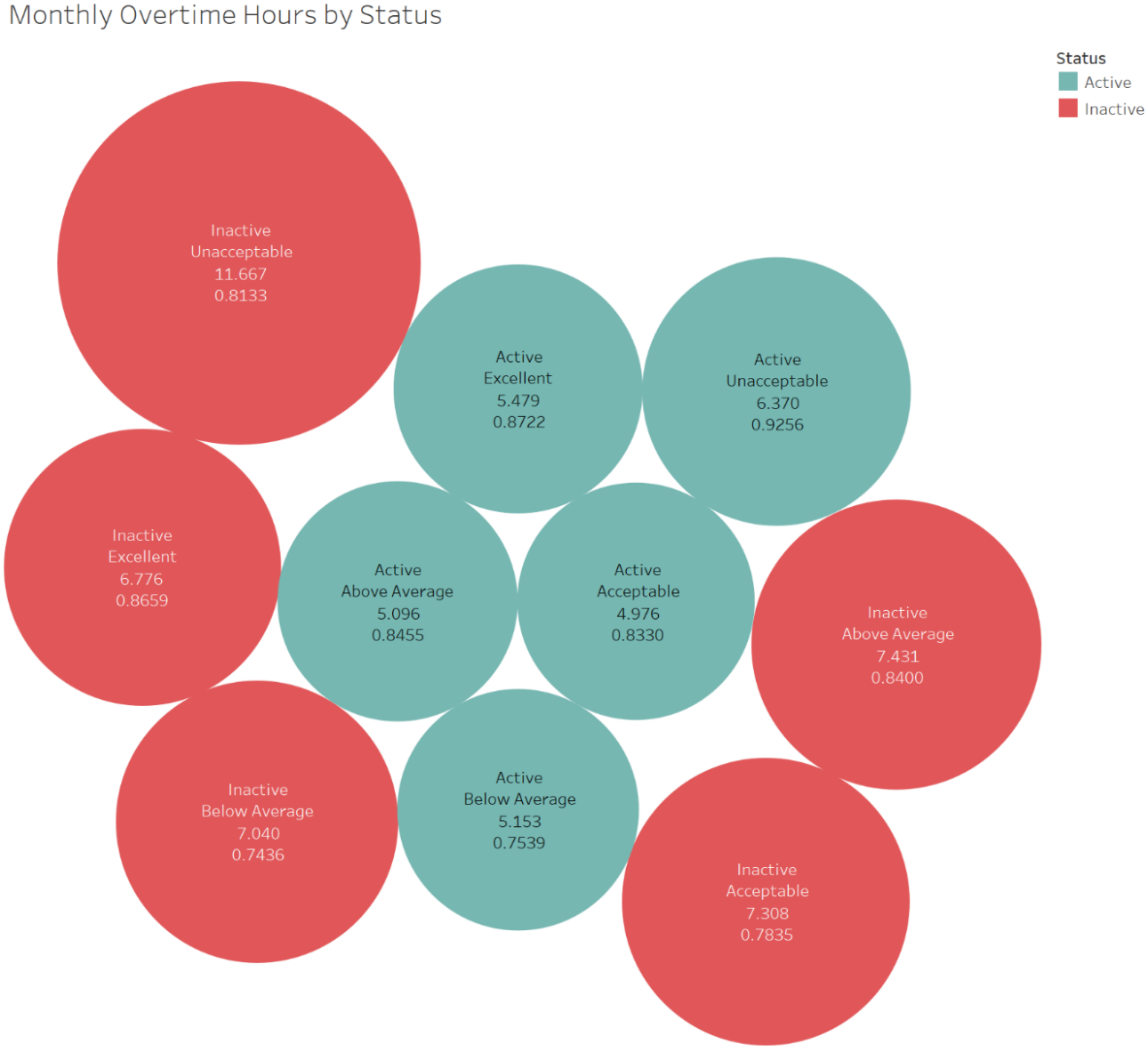


Figure 14

Figure 15:Most active employees work less than 10 hrs overtime and receive better financial rewards for it

Manager Effectiveness and Reportees

Inactive employees report lower manager effectiveness scores, and their managers tend to have a higher number of reportees, possibly overallocated.

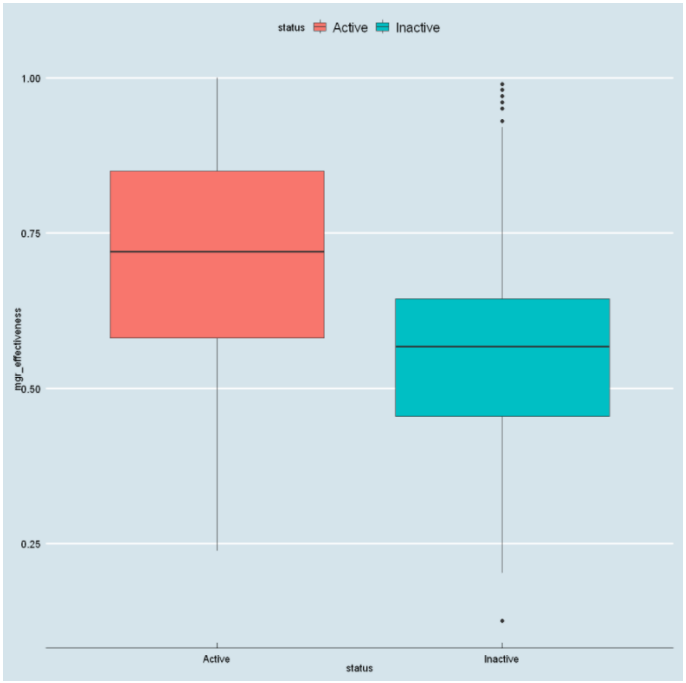


Figure 16: Distribution of Manager Effectiveness scores

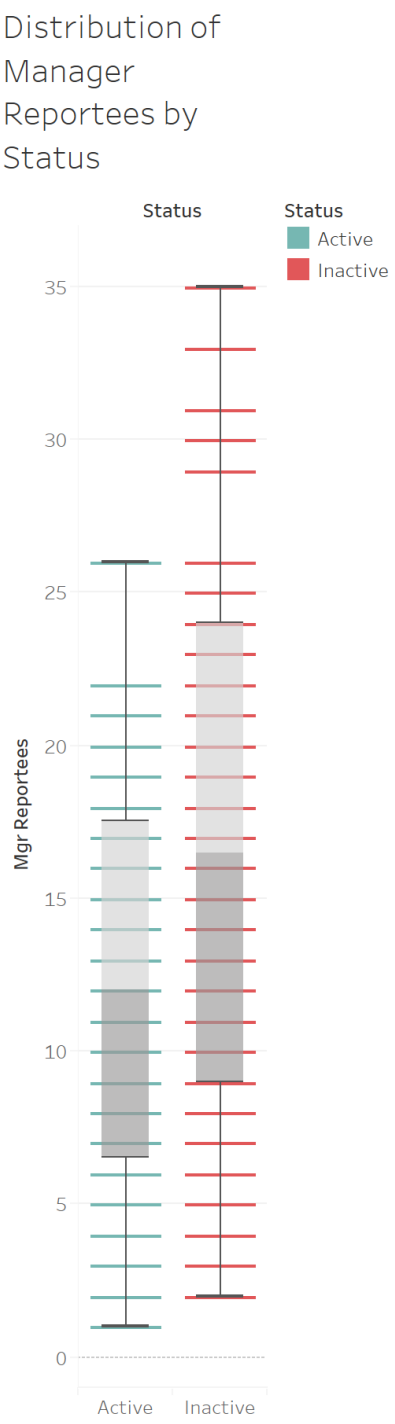

Distribution of Manager Reportees by Status
Status
Status
Active
Inactive
35
30
25
20
15
10
5
0
Mgr Reportees
Active
Inactive
Mgr Reportees for each Status. Color shows details about Status.

Figure 17

Number of Leaves Taken

Inactive employees take 20 leaves on average, compared to 12 for active employees. This can be an early warning signal to alert HR of possible occurrence of attrition.

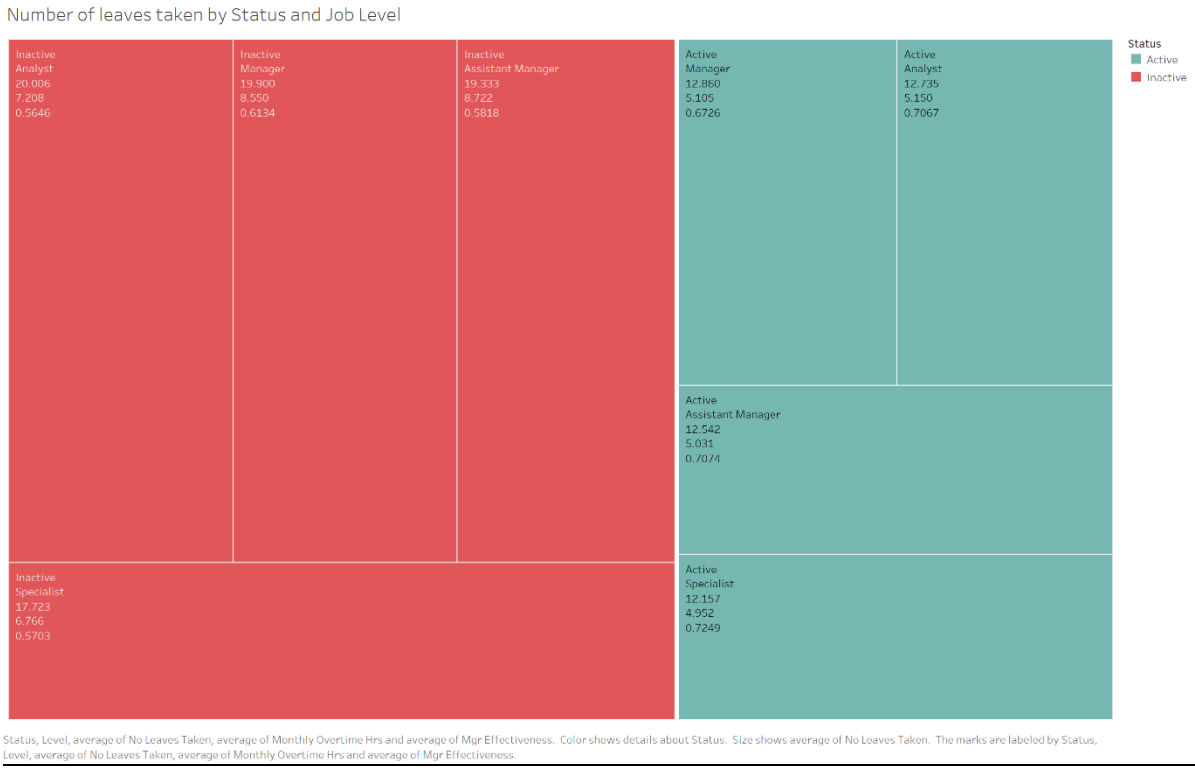


Figure 18

Tenure (Derived)

"Tenure" refers to the duration spent by an employee at the company. For inactive employees, it is the difference of last_working_date and start_date. For active employees, last working date is set to a cutoff_date. In R, Tenure is computed in years using the interval () and time_length() functions from the lubridate package. In Tableau, the DATEDIFF and DATETRUNC functions were used.

In figure 18, most inactive employees left within 2-6 years of joining the company, indicating a lack of career progression or training(which aligns with results of the career satisfaction analysis in Fig 9)

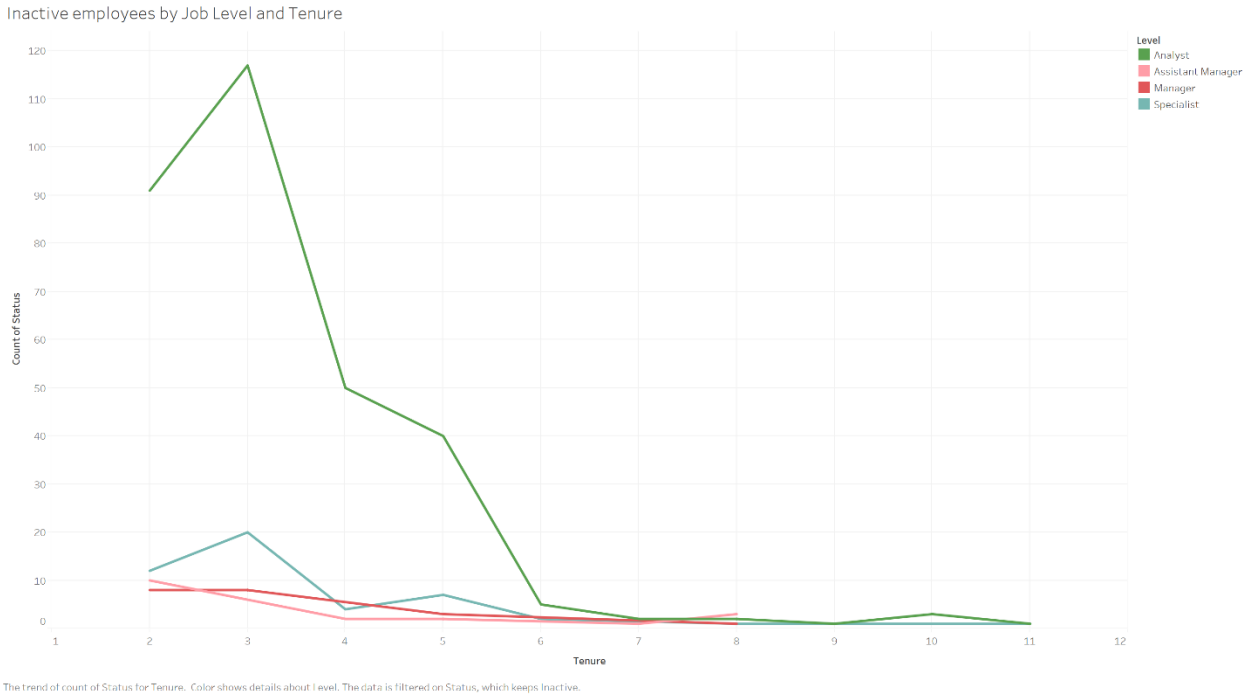


Figure 19

Clustering

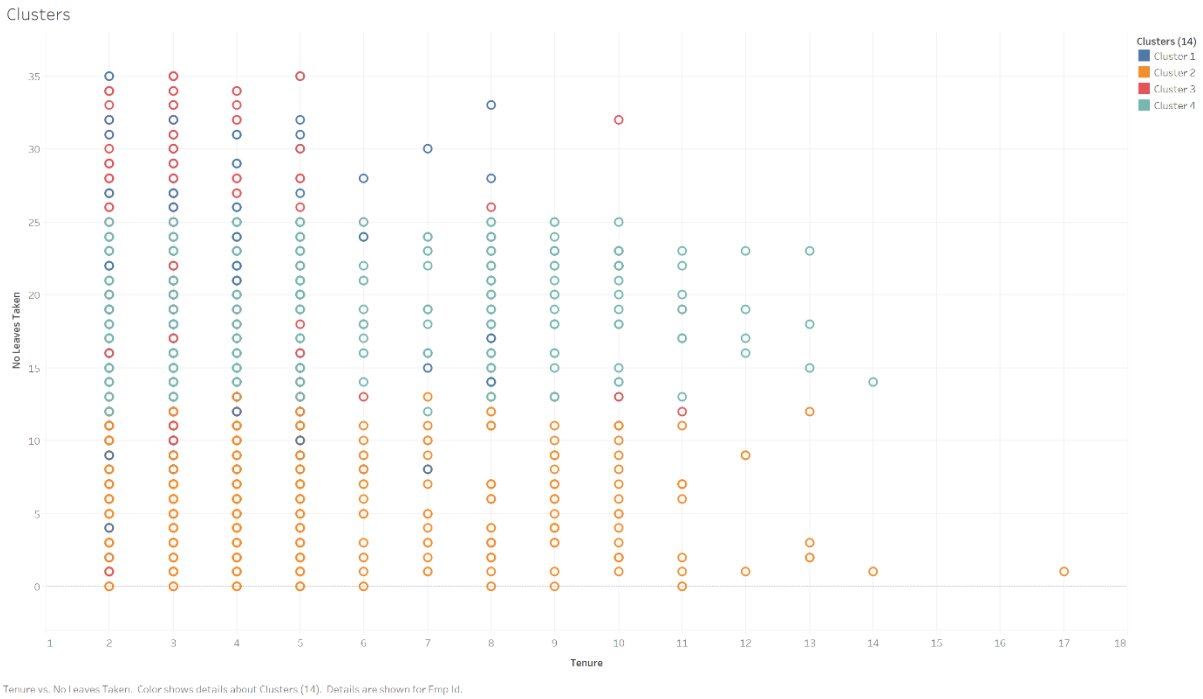


Figure 20

K- Means Clustering was performed while exploring the relationship between tenure and number of leaves taken (inactive employees take more leaves in recent years) to get

insights regarding different segments of employees, grouped based on similarities in attributes (Fig 20).

Two clusters of inactive employees are evident:

- Those who work very long hours, but face an imbalance with financial rewards, and a lower salary hike than others who work considerably lesser hours (Cluster 3)
- People who do not have as much to do in terms of projects and hours, who are relatively dissatisfied with their situation and managers (Cluster 4).

Summary Diagnostics

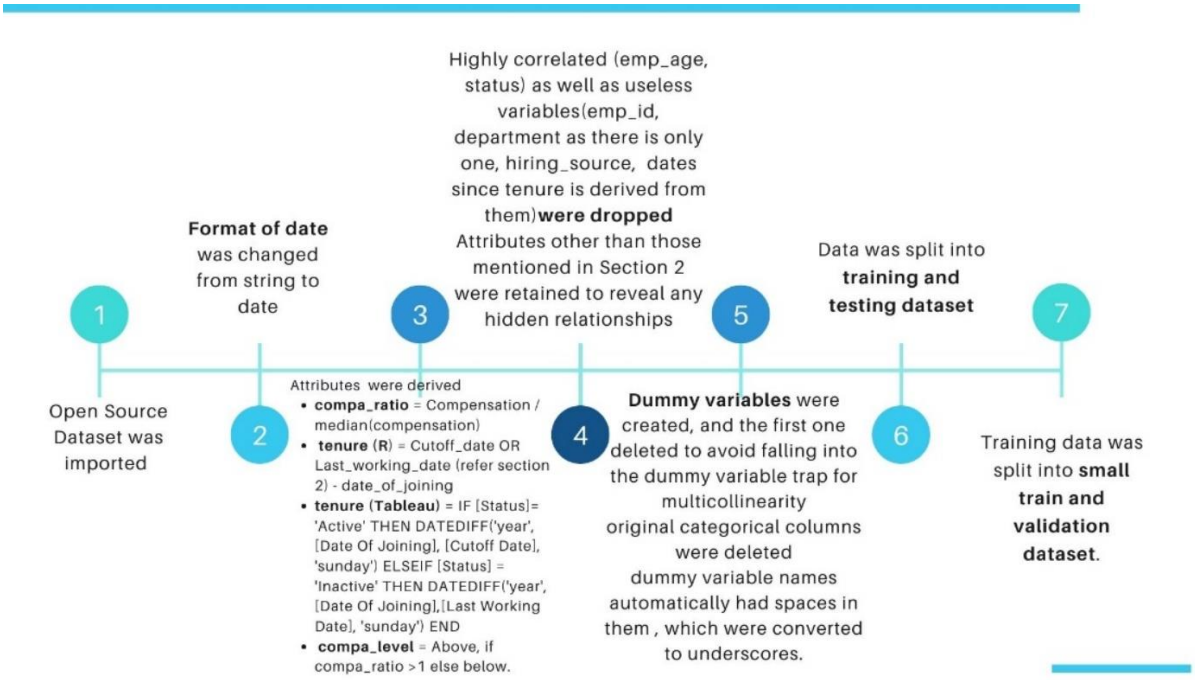
Number of Clusters:	4
Number of Points:	1954
Between-group Sum of Squares:	438.17
Within-group Sum of Squares:	322.61
Total Sum of Squares:	760.78

		Centers						
Clusters	Number of Items	Avg. No Leaves Taken	Avg. Compensation	Turnover	Avg. Tenure	Avg. Mgr Effectiveness	Avg. Monthly Overtime Hrs	Avg. Percent Hike
Cluster 1	188	4.2805	72233.0	0.0	7.1341	0.71263	5.2669	10.946
Cluster 2	761	17.831	82169.0	0.0	9.0462	0.72949	4.9755	10.204
Cluster 3	209	29.111	55537.0	1.0	4.1852	0.60785	9.0546	6.9873
Cluster 4	796	28.222	56135.0	1.0	4.4444	0.49531	3.8356	8.3139
Not Clustered	0							

Figure 21: Key attributes of different clusters. Clusters 1 and 2 refer to no attrition

Preparing the Data

The target attribute for model prediction is already available, *"turnover"*, a dummy variable representing "status_inactive". Since some models only use binary int variables, this is a useful variable to have.



Generating and Testing Prediction Models

Logistic Regression

The Brier Skill Score achieved was 0.78349. While predicting, type was set to "response", which gives the probability rather than the outcome variable. The most statistically significant attributes were tenure, percent_hike, distance_from_home, total_dependents, no_of_leaves_taken, monthly_overtime_hrs, 'rating_unacceptable", "rating_below_average" and mgr_effectiveness. Career_satisfaction and mgr_reportees were also significant. marital_status_single and education_masters also played an important role.

When interactions were carried out, the BSS dropped to -0.3. Hence, the Variance Inflation Factor method was used to detect multicollinearity and optimize the model. "level_Specialist", a perfectly multicollinear dummy variable and "compensation", with a VIF factor of ~15000 were removed. BSS slightly improved to 0.78463.

```
Call:
glm(formula = turnover ~ . - level_Specialist, family = "binomial",
    data = train_set)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.58673  -0.16539  -0.04680  -0.00647   3.04322

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                   9.781e+01  4.354e+01   2.246 0.024685 *
mgr_reportees                 9.537e-02  3.031e-02   3.146 0.001656 **
mgr_tenure                   -2.588e-02  4.401e-02  -0.588 0.556429
compensation                  1.718e-03  7.351e-04   2.338 0.019404 *
percent_hike                 -5.964e-01  8.058e-02  -7.402 1.34e-13 ***
hiring_score                  7.140e-02  4.337e-02   1.646 0.099708 .
no_previous_companies_worked -1.828e-02  5.294e-02  -0.345 0.729903
distance_from_home            2.114e-01  2.288e-02   9.240  < 2e-16 ***
total_dependents              8.422e-01  1.193e-01   7.061 1.65e-12 ***
no_leaves_taken               1.070e-01  1.995e-02   5.363 8.20e-08 ***
total_experience             -9.578e-03  6.612e-02  -0.145 0.884824
monthly_overtime_hrs          2.457e-01  4.254e-02   5.777 7.62e-09 ***
mgr_effectiveness            -1.005e+01  1.480e+00  -6.791 1.11e-11 ***
career_satisfaction           4.149e+00  1.466e+00   2.830 0.004660 **
perf_satisfaction             1.766e+00  1.270e+00   1.391 0.164268
work_satisfaction             2.066e+00  1.521e+00   1.358 0.174472
age_diff                      6.278e-02  3.739e-02   1.679 0.093146 .
tenure                       -3.437e-01  9.889e-02  -3.476 0.000509 ***
median_compensation          -2.044e-03  7.989e-04  -2.559 0.010502 *
compa_ratio                  -9.422e+01  4.025e+01  -2.341 0.019242 *
location_New_York             9.884e-01  4.560e-01   2.168 0.030195 *
location_Orlando             -8.768e-01  3.854e-01  -2.275 0.022911 *
level_Assistant_Manager      -5.082e-01  6.785e-01  -0.749 0.453860
level_Manager                -6.955e-01  6.594e-01  -1.055 0.291506
gender_Male                   4.146e-01  3.216e-01   1.289 0.197391
rating_Acceptable            -1.892e-02  3.778e-01  -0.050 0.960053
rating_Below_Average         -2.463e+00  6.836e-01  -3.604 0.000314 ***
rating_Excellent             -3.913e-01  8.946e-01  -0.437 0.661842
rating_Unacceptable          -4.518e+00  1.189e+00  -3.800 0.000144 ***
mgr_rating_Acceptable        -9.561e-02  3.582e-01  -0.267 0.789539
mgr_rating_Below_Average     -1.201e+00  6.615e-01  -1.816 0.069402 .
mgr_rating_Excellent         -7.318e-01  5.250e-01  -1.394 0.163327
mgr_rating_Unacceptable       1.163e+00  1.247e+00   0.933 0.350776
marital_status_Single         2.573e+00  5.588e-01   4.604 4.14e-06 ***
education_Masters             2.138e+00  6.059e-01   3.528 0.000419 ***
promotion_last_2_years_Yes    3.825e-01  4.607e-01   0.830 0.406375
compa_level_Below             3.031e-01  4.432e-01   0.684 0.494089
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1370.21  on 1367  degrees of freedom
Residual deviance:  349.93  on 1331  degrees of freedom
AIC: 423.93

Number of Fisher Scoring iterations: 8

0.783485700798038
```

Figure 22: Output of Logistic Regression

mgr_reportees	1.35465413264449
mgr_tenure	1.2779311375067
compensation	11515.0787271547
percent_hike	3.108013596304
hiring_score	1.17719921905059
no_previous_compan...	1.12793644689199
distance_from_home	1.28667350813173
total_dependents	2.10550099246543
no_leaves_taken	1.15179968432142
total_experience	2.34432936757264
monthly_overtime_hrs	1.33758473375535
mgr_effectiveness	3.11404000506252
career_satisfaction	3.05682674887363
perf_satisfaction	2.77786802487066
work_satisfaction	1.98815235374584
age_diff	2.00108606691097
tenure	1.51113901586985
median_compensation	13.0709029607094
compa_ratio	11452.4954887182
location_New_York	1.79788936036612
location_Orlando	1.69356702186501
level_Assistant_Mana...	1.26477007179645
level_Manager	1.14706286137602
gender_Male	1.18532927311811
rating_Acceptable	1.76961583025623
rating_Below_Average	2.40320665352125
rating_Excellent	1.16959529392162
rating_Unacceptable	2.42727714101789
mgr_rating_Acceptable	1.63244367702597
mgr_rating_Below_A...	1.62430871834306
mgr_rating_Excellent	1.47060538712122
mgr_rating_Unaccept...	1.22171218154367
marital_status_Single	2.28895137654004
education_Masters	1.3245159685606
promotion_last_2_ye...	1.66228189686249
compa_level_Below	2.40004390239926

Figure 23: VIF for each variable

Regression Tree

Regression Tree was created for better results. Initial BSS was 0.86213. Pruning was done to reduce overfitting by verifying the predictive utility of all nodes of the tree. Nodes that did not improve the expected prediction quality on new data were replaced by leaves. This slightly increased BSS to 0.86262.

Most important features are consistent with that from the logistic regression, with distance_from_home, no_leaves_taken, and monthly_overtime_hrs taking the lead.

However, a small change in the data can cause a large change in the structure of the decision tree causing instability. The model may also be making wrong assumptions due to the data being imbalanced.

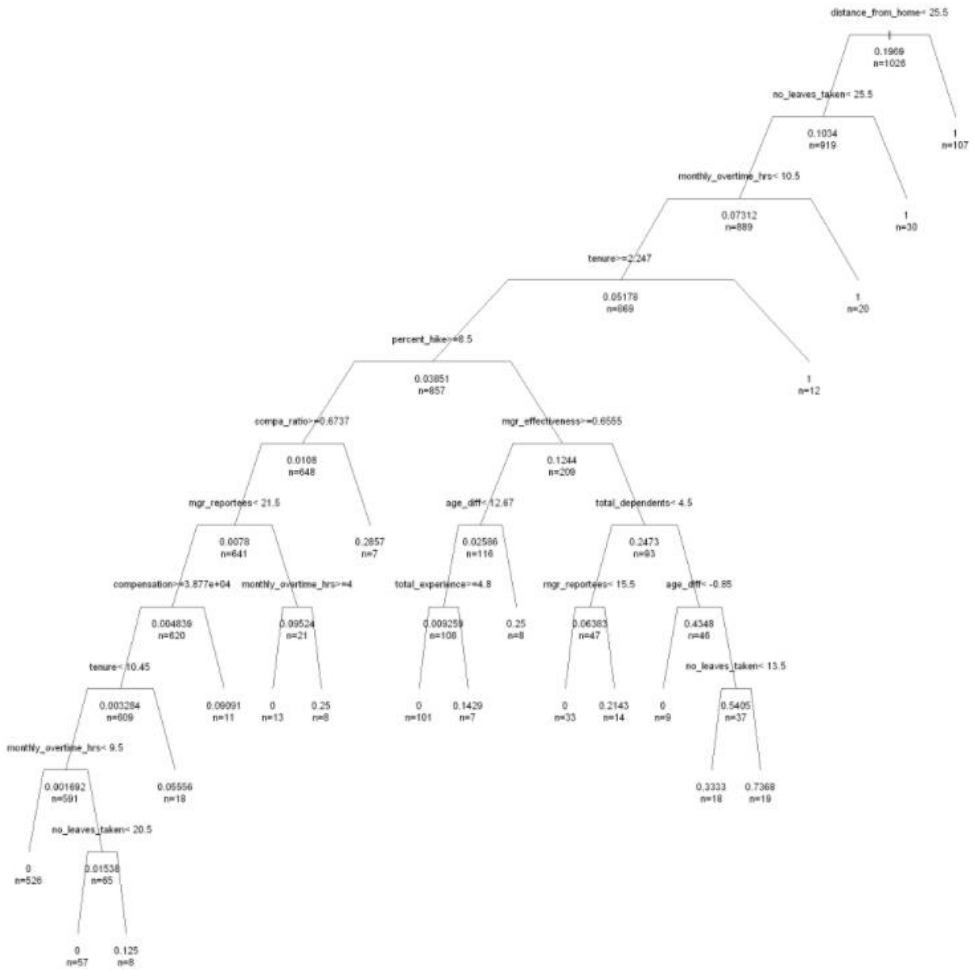


Figure 24:Output of Regression Tree (fancy tree using rparty() is too small)

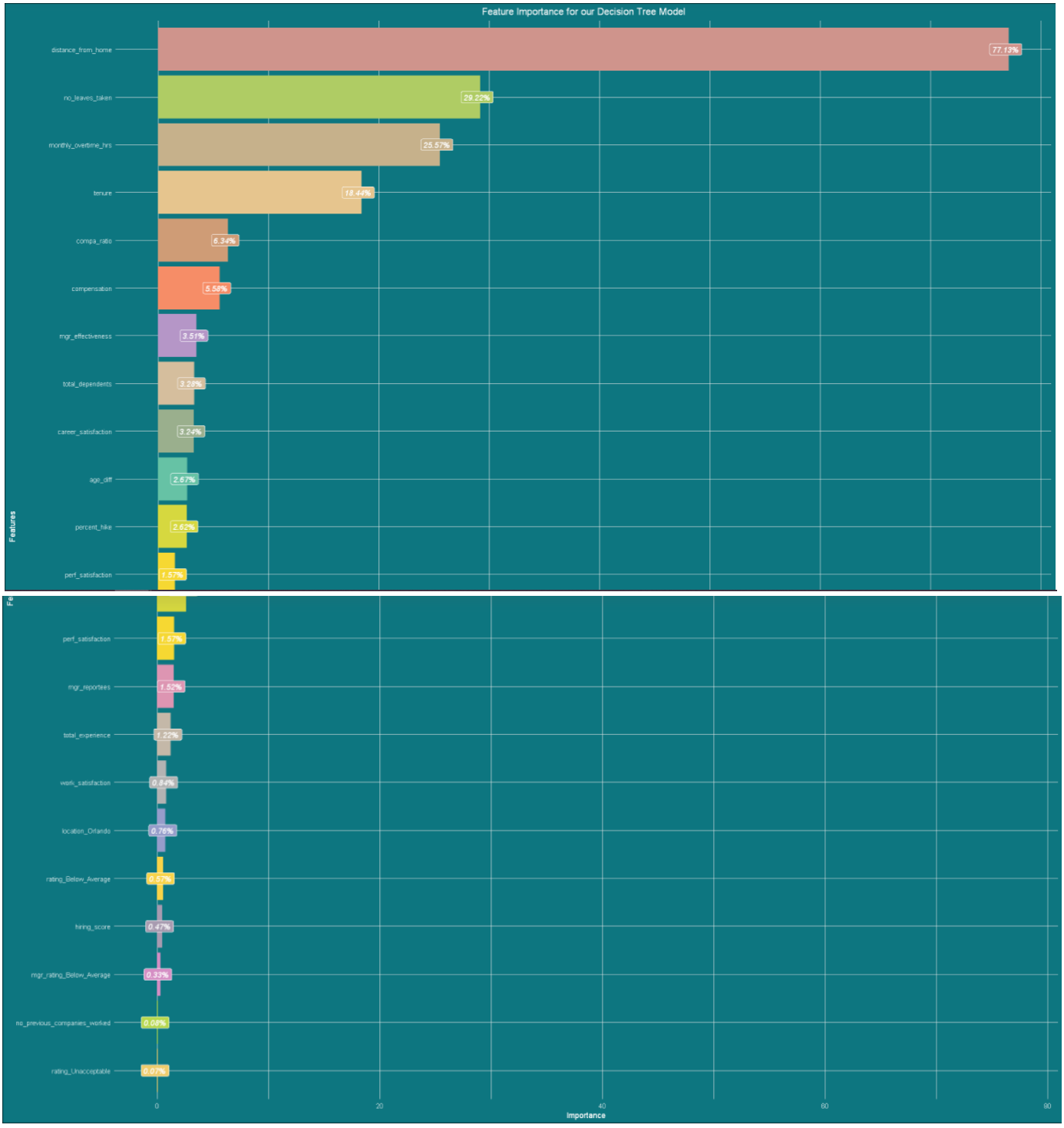


Figure 25: Most important factors according to regression tree

XGBoost

With a BSS of 0.84705, Importance of Variables is consistent with linear regression and regression tree, however we see that percent_hike has gained higher importance.

Feature	Gain	Cover	Frequency
distance_from_home	0.4149547449	0.170869534	0.077519380
no_leaves_taken	0.1402546383	0.124040481	0.062015504
monthly_overtime_hrs	0.0961164591	0.093698996	0.041343669
tenure	0.0787274736	0.105896838	0.090439276
percent_hike	0.0563820600	0.083401995	0.056847545
total_dependents	0.0502352916	0.074938713	0.060723514
mgr_effectiveness	0.0445843437	0.064264440	0.077519380
mgr_reportees	0.0185189962	0.043992881	0.054263566
compensation	0.0169101864	0.028145563	0.042635659
age_diff	0.0151529889	0.025482847	0.058139535
work_satisfaction	0.0105355451	0.033281073	0.055555556
rating_Below_Average	0.0080026844	0.007932401	0.012919897
location_Orlando	0.0063650859	0.011541697	0.015503876
career_satisfaction	0.0056920639	0.010161843	0.032299742
hiring_score	0.0050438104	0.013527786	0.028423773
compa_ratio	0.0045152889	0.010451574	0.020671835
total_experience	0.0044639534	0.015391581	0.040051680
no_previous_companies_worked	0.0039621243	0.011631638	0.027131783
marital_status_Single	0.0039592454	0.015114391	0.021963824
mgr_tenure	0.0039203739	0.016919750	0.054263566
location_New_York	0.0030702767	0.004239540	0.007751938
perf_satisfaction	0.0030077159	0.016457387	0.033591731
education_Masters	0.0029041462	0.007389926	0.005167959
gender_Male	0.0019317887	0.003559097	0.007751938
mgr_rating_Acceptable	0.0005313089	0.003477023	0.005167959
rating_Acceptable	0.0002574053	0.004191005	0.010335917

0.847051822834058

Figure 26: XGBoost Most important attributes

Random Forest

Random forest model was chosen as it generates more accurate results than regression trees by introducing a random component in the tree building process, which lowers the variance of a single tree's prediction.

Parallel processing was carried out to improve the model. The OOB was found to be lowest with mtry =12, which was added to the model. Ultimately, the BSS was 0.86123.

Importance of features took almost the same form as that from XGBoost.

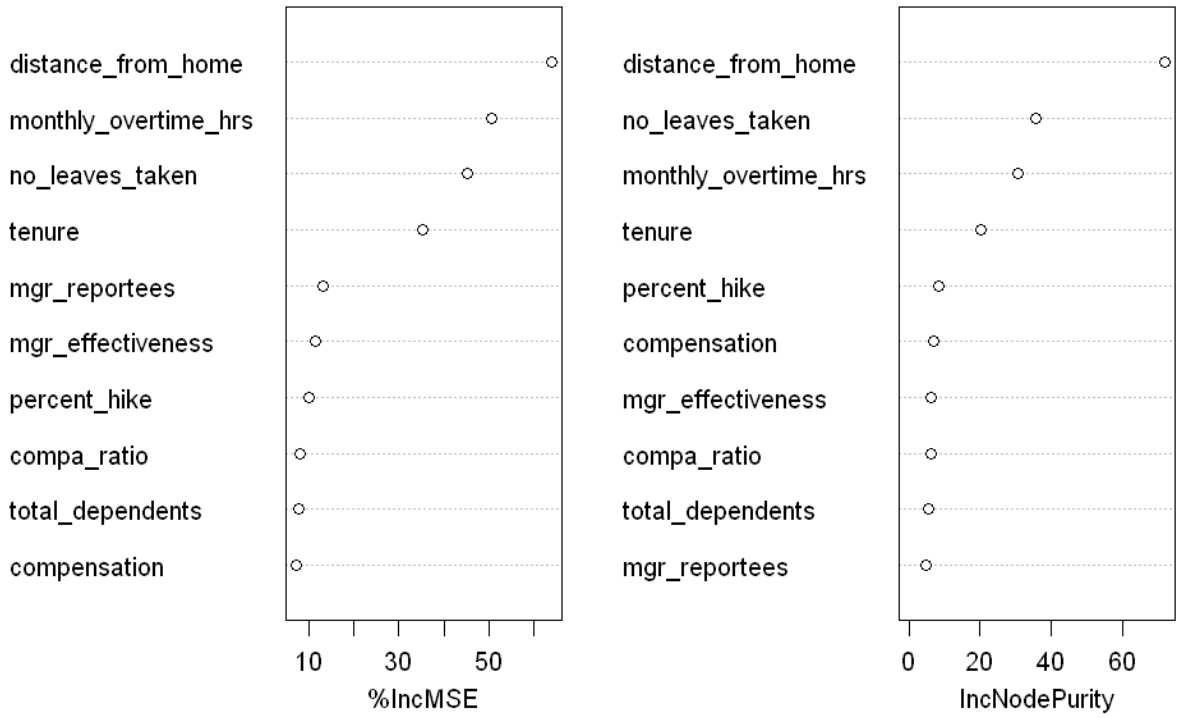


Figure 27: Most important attributes according to random forest

Ensemble Models

Ensemble models were formed with Logistic Regression, Regression tree and XGBoost to incorporate models with varying BSS.

Weighted Average (BSS 0.88583)
A weighted Average model was formed by giving stepwise weights to base models, resulting in 81 combinations. The highest BSS was selected from the matrix at Row 4, Column 5.

Stacking Model – Regression Tree (BSS 0.90003)
m1, m2, and m3 predictions were generated from the logistic regression, regression tree, and XGBoost respectively and stored along with true values in a dataframe. All three

predictions (m1_pred, m2_pred and m3_pred) were fed as predictor variables to a regression tree, with turnover as the output variable. The BSS is higher than the Weighted Average.

Stacking Model – XGBoost (BSS 0.95143)
The above process was repeated by feeding predictor variables to XGBoost instead of regression tree. The dataframe was converted to a matrix to be able to use in XGBoost.

Comparison of Brier Skill Scores of all models:

Model	Brier Skill Score
Logistic Regression	0.78463
Regression Tree	0.86262
XGBoost	0.84705
Random Forest	0.86123
Weighted Average	0.88583
Stacker- Regression Tree	0.90003
Stacker- XGBoost	0.95143

The Stacker XG Boost model has the best BSS, due to many reasons including meta-learner stacking generalizing better than a single model, as well as regularization by XGBoost to combat overfitting. It was used to generate final prediction model on the testing data trained on the entire train set.

The final BSS is 0.94716.

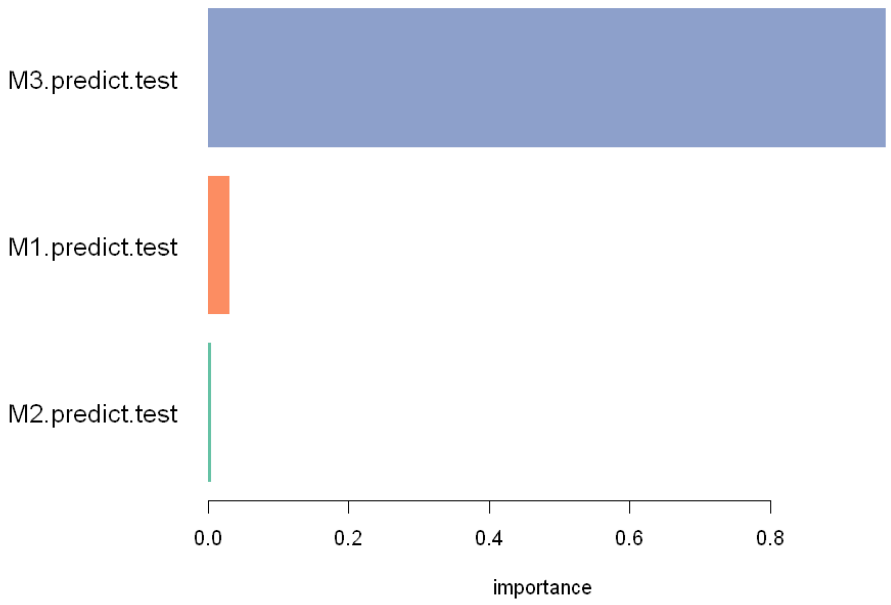


Figure 28: Most important predictions from stacker model, XGBoost is the most important

Conclusions

Based on this analysis, top reasons for leaving an organization are:

- **Distance from Home**: Employees who must travel long distances are likely to churn
- **Monthly Overtime Hours**: Long work hours can be exhausting for employees and deteriorate their performance.
- **Lack of financial rewards** results in declined motivation to work and stay in the organization.
- **Tenure**: Employees who have joined the organization in the last 2-5 years are likely to leave due to **lack of career progression and opportunities.**
- **Manager's ineffectiveness and overburden** contributes to employees wanting to leave.

Turnover cost includes the cost of offboarding an employee, and other hidden costs such as the transition of work, training, and improving performance and business loss. HR and managers can take following measures to retain their employees and ultimately save costs.

- **Increase in financial Rewards:** It was noted that most employees have a hike of below 10%. *percent_hike* values were grouped into intervals and turnover rate for each *hike_range* was computed.

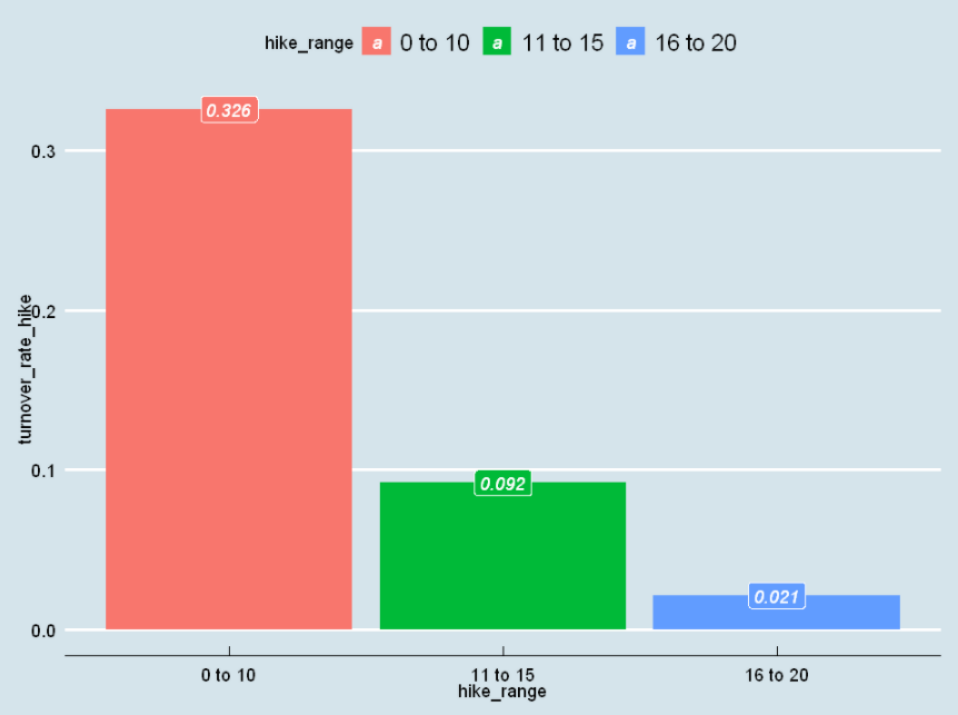


Figure 29: Turnover rate for each hike range

Turnover rate is highest for the 0-10% category. If all employees who received a salary hike between 0 and 10% were offered a hike between 10 and 15%, chances are that the organization would be able to retain most of the employees.

It is estimated that retention of an employee at the Analyst level saves ~$40,000 USD to the company.[iv]

Turnover Overview	Scenario 1	Scenario 2	Change
Total Turnover	300	200	33%
Average Cost of Turnover	40,000	40,000	0
Total Cost of Turnover	12,000,000	8,000,000	4,000,000

Using these facts, Return on Investment (ROI) can be calculated:

Median Salary Analyst	54684	From fig below
Analyst Turnover Original(%)	21	From fig 2
Analyst Turnover New(%)	12	Assumption
Median Salary Manager	54456	From fig below
Manager Turnover Original(%)	26	Fig 2
Manager Turnover New(%)	13	Assumption
Turnover Cost (Both analyst and Managers)	40,000	From above calculation, we assume managers avaerage turnover cost same as analyst in this case
Extra cost to the Organization	5457	(B1 * 0.05) + (B4 * 0.05), as we are aiming for 5% increase
Savings	8800	((B2-B3)/100)*B7 + ((B5-B6)/100)*B7
Return on Investment (%)	249.2211838	(B9/B8) *100

Median Salary across Job Levels

Level

Analyst	Assistant Mana..	Manager	Speciali..
54,684	54,210	54,456	56,442

Thus, though the organization will incur a cost equivalent to "extra cost to the organization", cost savings significantly make up for it and ROI of increasing salary by 5% is 249%.

- The organization should allow employees who travel distances greater than 20km to **work from home** few days a week. This practice is coming to the forefront, with Facebook declaring that its employees can carry on WFH after the pandemic.[v]

- Once **early warning signals like greater absence and ominous survey** results are found, HR and managers should conduct regular **one on one meetings** with employees to understand and relieve their issues.

- If organization requires overtime work, this should be made manageable by **assigning equal overtime hours** on a roster basis to each employee and avoid overburdening some. Financial incentives could be introduced.

- **Career progression** where deserved should be offered, internal job postings, and investments in training and development should be made.

- Reportees should be reallocated to **Managers**, to ensure equitable distribution. They should also undertake Leadership Development and Managerial Effectiveness Training.

What-If Analysis in Tableau can further be used to assess the impact of these changes.

Appendix

Understanding the Data

In [1]:
```
#Install and Load packages
library(rlang)
library(dplyr)
library(readxl)
library(ggplot2)
library(tidyverse)
install.packages('Information')
library(Information)
library(cowplot)
install.packages('ggthemes')
library(ggthemes)
library(lubridate)
install.packages('ggcorrplot')
library(ggcorrplot)
library("GGally")
install.packages("partykit")
install.packages("party")
install.packages("rattle")
library(rattle)
library(party)
library(partykit)
install.packages("rpart.plot")
library(rpart.plot)
library(ggthemes)
library(extrafont)
install.packages("hrbrthemes")
library(hrbrthemes)
install.packages("ggExtra")
library(ggExtra)
install.packages("xgboost")
library(xgboost)
library(randomForest)
library(lubridate)
install.packages("doSNOW")
library(doSNOW)
library(foreach)
library(parallel)
library(car)
```

```
    importance

The following object is masked from 'package:ggplot2':

    margin

The following object is masked from 'package:dplyr':

    combine

Installing package into 'C:/Users/bhavy/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
package 'doSNOW' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\bhavy\AppData\Local\Temp\Rtmpag78e6\downloaded_packages
Warning message:
"package 'doSNOW' was built under R version 3.6.3"Loading required package: foreach
```

In [2]:
```
# Import the data
data <- read_csv("C:/Users/bhavy/Downloads/org_final.csv")

# Check the structure of the dataset, the dplyr way
glimpse(data)
```

```
Parsed with column specification:
cols(
```

```
  education = col_character(),
  promotion_last_2_years = col_character(),
  date_of_joining = col_character(),
  last_working_date = col_character(),
  department = col_character(),
  mgr_id = col_character(),
  cutoff_date = col_character()
)
See spec(...) for full column specifications.
```

```
Rows: 1,954
Columns: 34
$ emp_id                      <chr> "E10012", "E10025", "E10027", "E10048"...
$ status                      <chr> "Active", "Active", "Active", "Active"...
$ location                    <chr> "New York", "Chicago", "Orlando", "Chi...
$ level                       <chr> "Analyst", "Analyst", "Analyst", "Anal...
$ gender                      <chr> "Female", "Female", "Female", "Male", ...
$ emp_age                     <dbl> 25.09, 25.98, 33.40, 24.55, 31.23, 31....
$ rating                      <chr> "Above Average", "Acceptable", "Accept...
$ mgr_rating                  <chr> "Acceptable", "Excellent", "Above Aver...
$ mgr_reportees               <dbl> 9, 4, 6, 10, 11, 19, 21, 9, 12, 22, 17...
$ mgr_age                     <dbl> 44.07, 35.99, 35.78, 26.70, 34.28, 34....
$ mgr_tenure                  <dbl> 3.17, 7.92, 4.38, 2.87, 12.95, 10.88, ...
$ compensation                <dbl> 64320, 48204, 85812, 49536, 75576, 569...
$ percent_hike                <dbl> 10, 8, 11, 8, 12, 8, 12, 9, 9, 6, 11, ...
$ hiring_score                <dbl> 70, 70, 77, 71, 70, 75, 72, 70, 70, 70...
$ hiring_source               <chr> "Consultant", "Job Fairs", "Consultant...
$ no_previous_companies_worked <dbl> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 3, 6, 2,...
$ distance_from_home          <dbl> 14, 21, 15, 9, 25, 23, 17, 16, 22, 22,...
$ total_dependents            <dbl> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, 5, 4,...
$ marital_status              <chr> "Single", "Single", "Single", "Single"...
$ education                   <chr> "Bachelors", "Bachelors", "Bachelors",...
$ promotion_last_2_years      <chr> "No", "No", "Yes", "Yes", "No", "No", ...
$ no_leaves_taken             <dbl> 2, 10, 18, 19, 25, 15, 10, 20, 22, 23,...
$ total_experience            <dbl> 6.86, 4.88, 8.55, 4.76, 8.06, 13.72, 5...
$ monthly_overtime_hrs        <dbl> 1, 5, 3, 8, 1, 7, 2, 10, 2, 10, 8, 3, ...
$ date_of_joining             <chr> "6/3/2011", "23/09/2009", "2/11/2005",...
$ last_working_date           <chr> NA, NA, NA, NA, NA, "11/12/2014", NA, ...
$ department                  <chr> "Customer Operations", "Customer Opera...
$ mgr_id                      <chr> "E9335", "E6655", "E13942", "E7063", "...
$ cutoff_date                 <chr> "31/12/2014", "31/12/2014", "31/12/201...
$ turnover                    <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1,...
$ mgr_effectiveness           <dbl> 0.730, 0.581, 0.770, 0.240, 0.710, 0.5...
$ career_satisfaction         <dbl> 0.73, 0.72, 0.85, 0.42, 0.78, 0.88, 0....
$ perf_satisfaction           <dbl> 0.73, 0.84, 0.80, 0.33, 0.67, 0.81, 0....
$ work_satisfaction           <dbl> 0.75, 0.85, 0.87, 0.85, 0.80, 0.86, 0....
```

In [3]:
```
# Count Active and Inactive employees
data %>%
  count(status)


# Calculate turnover rate
data %>%
  summarize(avg_turnover_rate = mean(turnover))
```

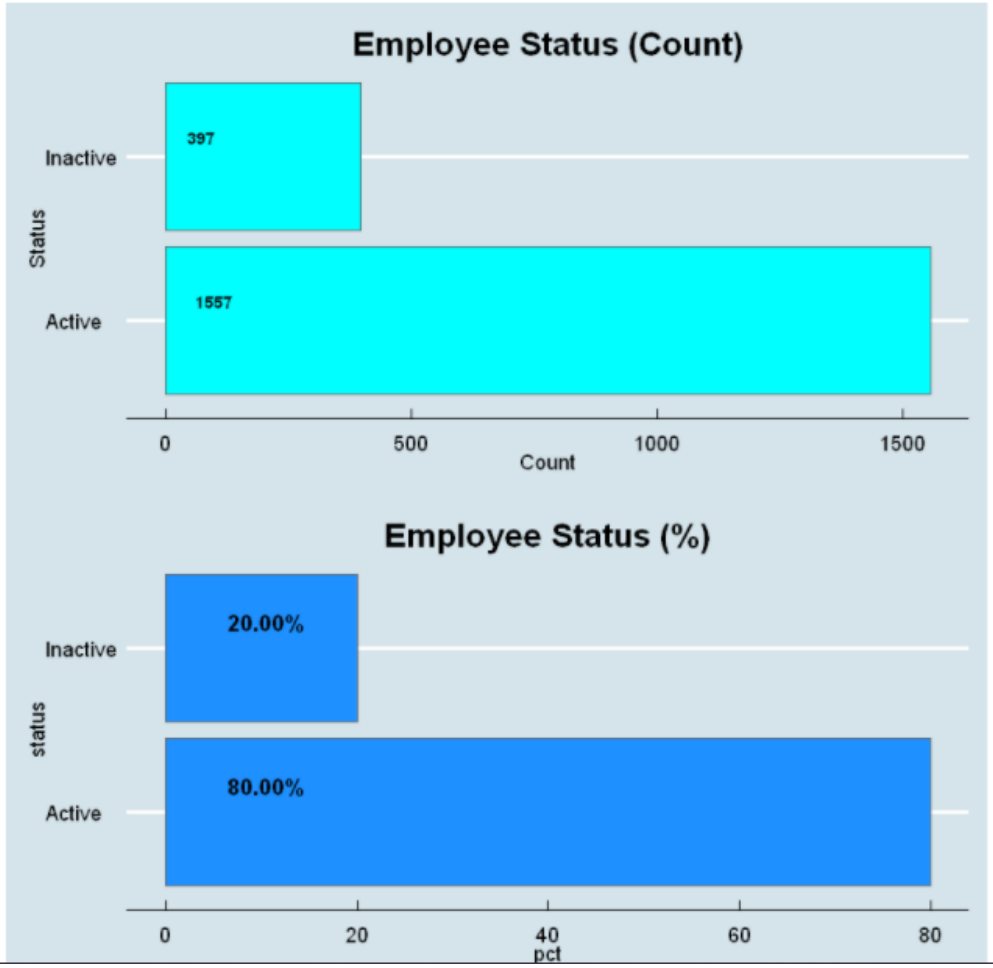
status	n
Active	1557
Inactive	397

avg_turnover_rate
0.203173

In [4]:
```
# Let us understand the distribution of our labels (employee turnover)
#Calculate and visualize the number of employees who leave the organization vs those who stay
attritions_number <- data %>% group_by(status) %>% summarise(Count=n()) %>%
ggplot(aes(x=status, y=Count)) + geom_bar(stat="identity", fill="cyan", color="grey40") + theme_economist() + coord_flip() +
geom_text(aes(x=status, y=0.01, label= Count),
          hjust=-0.8, vjust=-1, size=3,
```

In [4]:

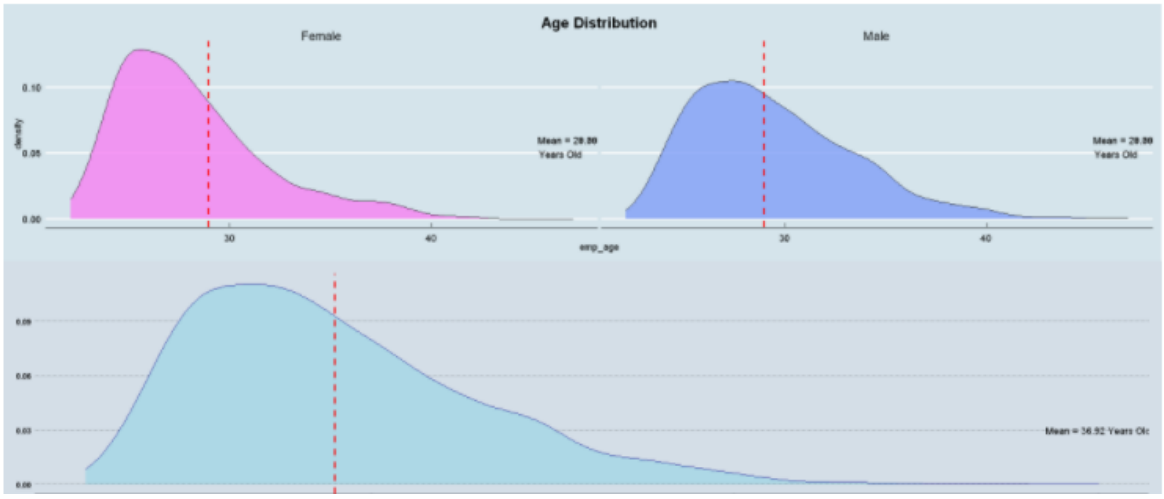
```
# Let us understand the distribution of our labels (employee turnover)
#Calculate and visualize the number of employees who leave the organization vs those who stay
attritions_number <- data %>% group_by(status) %>% summarise(Count=n()) %>%
ggplot(aes(x=status, y=Count)) + geom_bar(stat="identity", fill="cyan", color="grey40") + theme_economist() + coord_flip() +
geom_text(aes(x=status, y=0.01, label= Count),
            hjust=-0.8, vjust=-1, size=3,
            colour="black", fontface="bold",
        angle=360) + labs(title="Employee Status (Count)", x="Status",y="Count") + theme(plot.title=element_text(hjust=0.5))

#Calculate and visualize the percentage relative to each other and visualize
attrition_percentage <- data %>% group_by(status) %>% summarise(Count=n()) %>%
mutate(pct=round(prop.table(Count),2) * 100) %>%
ggplot(aes(x=status, y=pct)) + geom_bar(stat="identity", fill = "dodgerblue", color="grey40") + coord_flip() +
geom_text(aes(x=status, y=0.01, label= sprintf("%.2f%%", pct)),
            hjust=-0.8, vjust=-1, size=4,
            colour="black", fontface="bold") + theme_economist() +
labs(title="Employee Status (%)") + theme(plot.title=element_text(hjust=0.5))

#plot number and percentage together
plot_grid(attritions_number, attrition_percentage, align="h", nrow=2)
```



In [5]:
```
#analysis  by gender
#Calculate the age distribution between males and females
avg.age <- data %>% dplyr::select(gender, emp_age) %>% group_by(gender) %>% summarize(avg=mean(emp_age))

avg.age
```

gender	avg
Female	28.06934
Male	29.39449

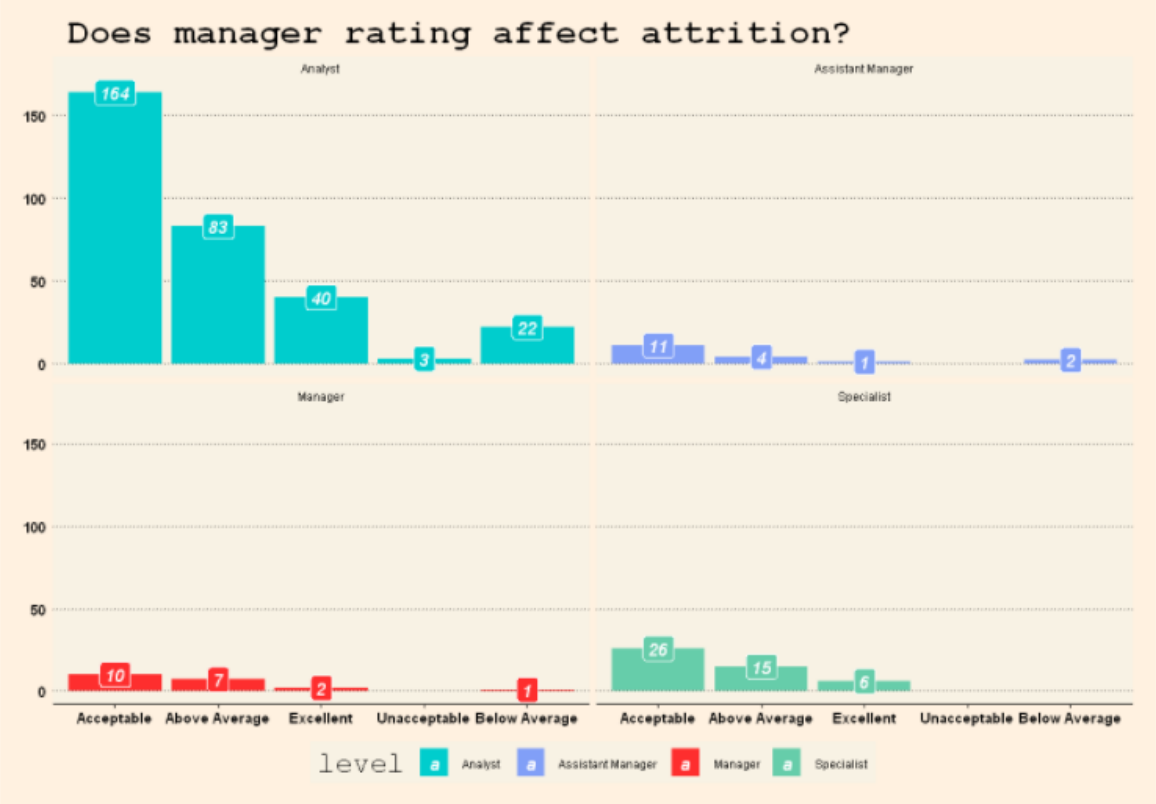
In [6]:
```
# Let's look at the distribution of the Age of our employees
# Unlike the older generation, millenials tend to switch workplaces more and that could
#be an explanation of why we have the current levels of attrition

options(repr.plot.width=18, repr.plot.height=8)

dat_text <- data.frame(
  label = c("Mean = 28.06 \n Years Old", "Mean = 29.39 \n Years Old"), #from the previous cell
  Gender   = c("Female", "Male")
)

gender.dist <- data %>% dplyr::select(gender, emp_age) %>% filter(gender == 'Male' | gender== "Female") %>%
filter(!is.na(emp_age)) %>% group_by(gender) %>%
ggplot(aes(x=emp_age)) + geom_density(aes(fill=gender), alpha=0.8, show.legend=FALSE) + facet_wrap(~gender) +theme_economist() +
geom_vline(aes(xintercept=mean(emp_age)),
           color="red", linetype="dashed", size=1) + labs(title="Age Distribution") +
theme(plot.title=element_text(hjust=0.5)) + scale_fill_manual(values=c("#F781F3", "#819FF7")) +
geom_text(
  data    = dat_text,
  mapping = aes(x = 45, y = 0.03, label = label),
  hjust   = -0.1,
  vjust   = -1
)

overall.dist <- data %>% dplyr::select(gender, emp_age) %>% filter(!is.na(emp_age)) %>%
ggplot(data, mapping=aes(x=emp_age)) + geom_density(color="darkblue", fill="lightblue") +
geom_vline(aes(xintercept=mean(emp_age)),
           color="red", linetype="dashed", size=1) +  theme_wsj(base_size = 9, color = 'blue') + labs(x="Overall Age") +
annotate("text", label = "Mean = 36.92 Years Old", x = 50, y = 0.03, color = "black")


plot_grid(gender.dist, overall.dist, nrow=2)
```



There are not many insights regarding gender or age distribution.

In [7]:
```
#Now we see whether the rating employees give to their managers| managers is an indication of attrition
library(extrafont)
#Set plot size
options(repr.plot.width=10, repr.plot.height=7)

#attritions are the inactive employees in the organization
attritions <- data %>% dplyr::filter(status == "Inactive")

#calculate and visualize attritions by manager rating for each job level
attritions$mgr_rating <- as.factor(attritions$mgr_rating)

by.level <- attritions %>% dplyr::select(level, mgr_rating) %>% group_by(level, mgr_rating) %>%
summarize(count=n()) %>%
ggplot(aes(x=fct_reorder(mgr_rating, -count), y=count, fill=level)) + geom_bar(stat='identity') + facet_wrap(~level) +
theme_wsj(base_size = 9) + theme(legend.position="bottom", plot.title=element_text(hjust=0.05), plot.background=element_rect(fil
scale_fill_manual(values=c("cyan3", "#819FF7", "#FE2E2E", "aquamarine3" )) +
geom_label(aes(label=count, fill = level), colour = "white", fontface = "italic") +
labs(title="Does manager rating affect attrition?", x="Manager Rating", y="Number of Employees")

by.level
```

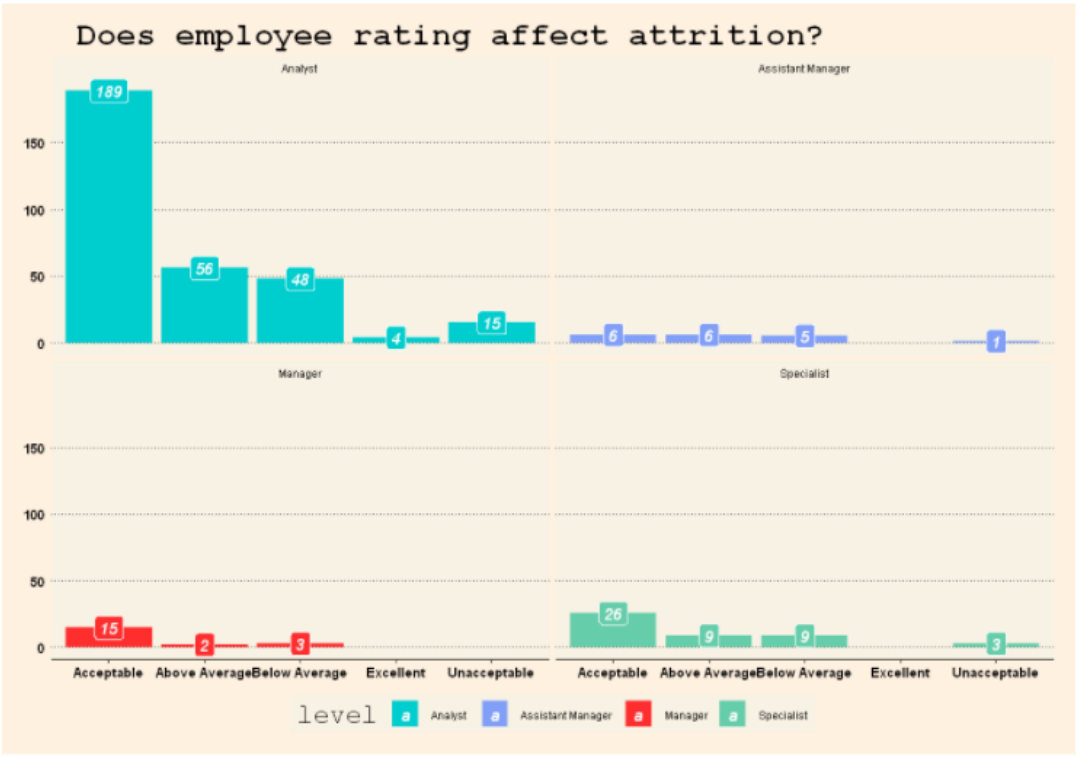
```
`summarise()` has grouped output by 'level'. You can override using the `.groups` argument.
```



This does not really establish any relationship, as most employees give managers a rating “Acceptable”, which is quite a neutral sentiment. The same is noticed with Employee rating below.

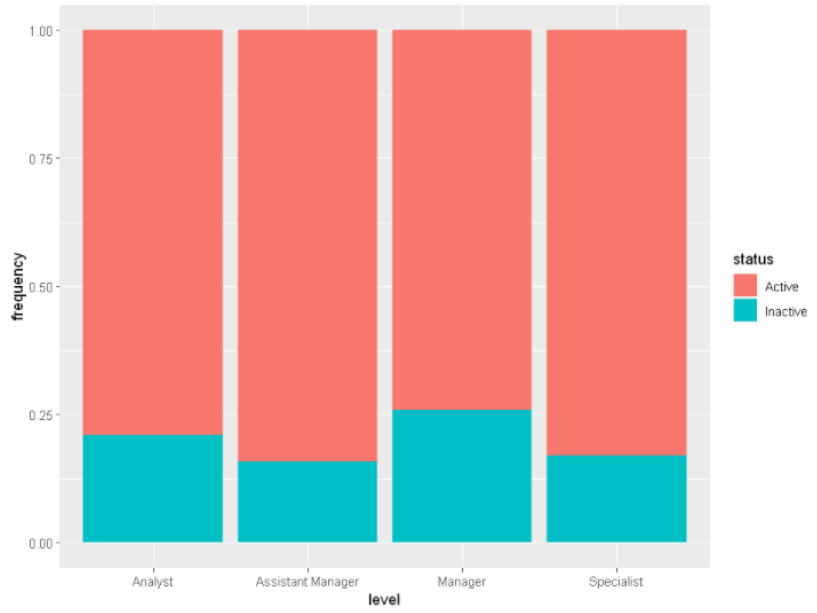
In [8]:
```
#Wee see whether the rating given to the employee is an indication of attrition

options(repr.plot.width=10, repr.plot.height=7)

attritions <- data %>% dplyr::filter(status == "Inactive")

attritions$mgr_rating <- as.factor(attritions$rating)

by.level <- attritions %>% dplyr::select(level, rating) %>% group_by(level, rating) %>%
summarize(count=n()) %>%
ggplot(aes(x=fct_reorder(rating, -count), y=count, fill=level)) + geom_bar(stat='identity') + facet_wrap(~level) + theme_wsj(bas
theme(legend.position="bottom", plot.title=element_text(hjust=0.1), plot.background=element_rect(fill="#FFF1E0")) +
scale_fill_manual(values=c("cyan3", "#819FF7", "#FE2E2E", "aquamarine3" )) +
geom_label(aes(label=count, fill = level), colour = "white", fontface = "italic") +
labs(title="Does employee rating affect attrition?", x="Manager Rating", y="Number of Employees")

by.level
```

```
`summarise()` has grouped output by 'level'. You can override using the `.groups` argument.
```



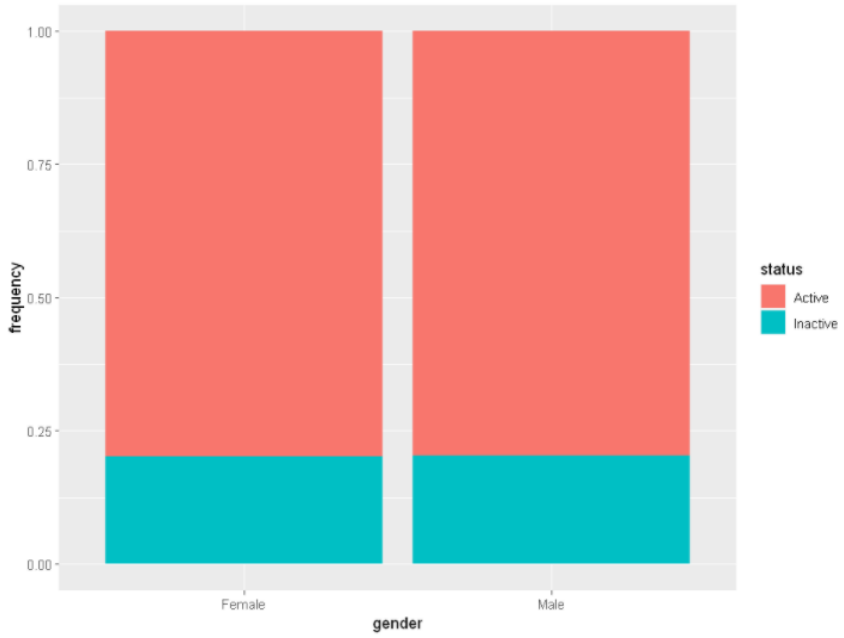
In [9]:
```
options(repr.plot.width=8, repr.plot.height=6)
#Let us visualize frequency of each job level in the organization
#Group by status to see the proportion of employees in each job level which stay in and leave the organization
#it is more reasonable to view by position "fill" as it shows us the propportion of
#inactive employees as a total of employees in each level
#otherwise we may get the illusion that a job level has more attrition as there may be more employees in that level
by_level <- data %>% group_by(status, level) %>% summarise(frequency=n()) %>%
ggplot(aes(x = level, y = frequency, fill = status)) +
  geom_col(position = "fill") #+ geom_bar(position="fill", stat="identity")
plot_grid(by_level)
```

```
`summarise()` has grouped output by 'status'. You can override using the `.groups` argument.
```

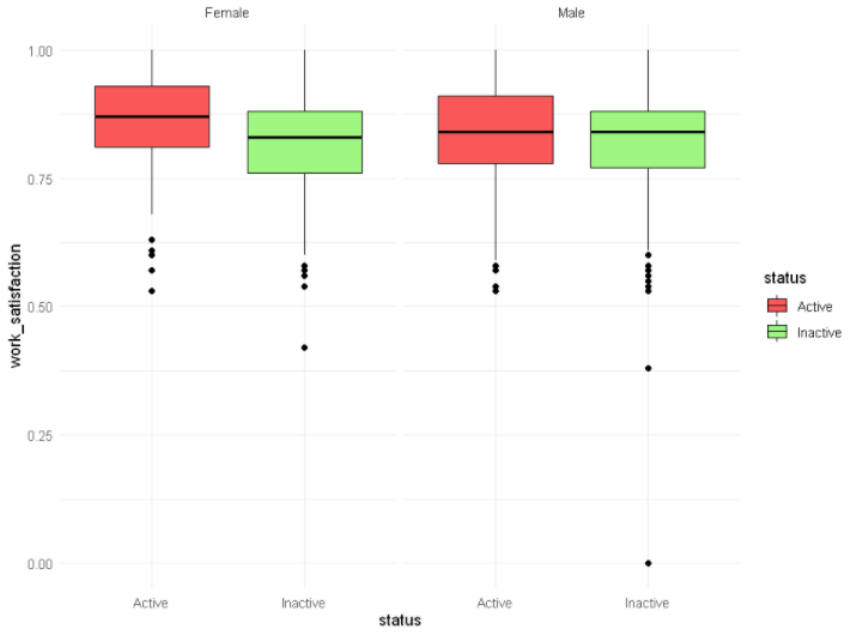


In [10]:
```
options(repr.plot.width=8, repr.plot.height=6)

#visualize attrition by gender
by_level <- data %>% group_by(status, gender) %>% summarise(frequency=n()) %>%
ggplot(aes(x = gender, y = frequency, fill = status)) +
  geom_col(position = "fill") #+ geom_bar(position="dodge", stat="identity")
plot_grid(by_level)
```

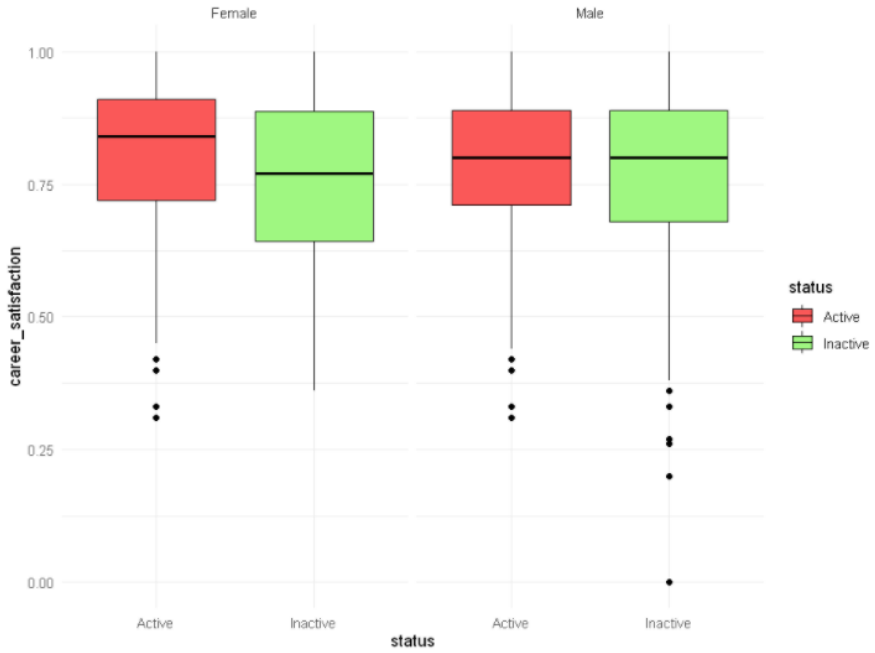
```
`summarise()` has grouped output by 'status'. You can override using the `.groups` argument.
```



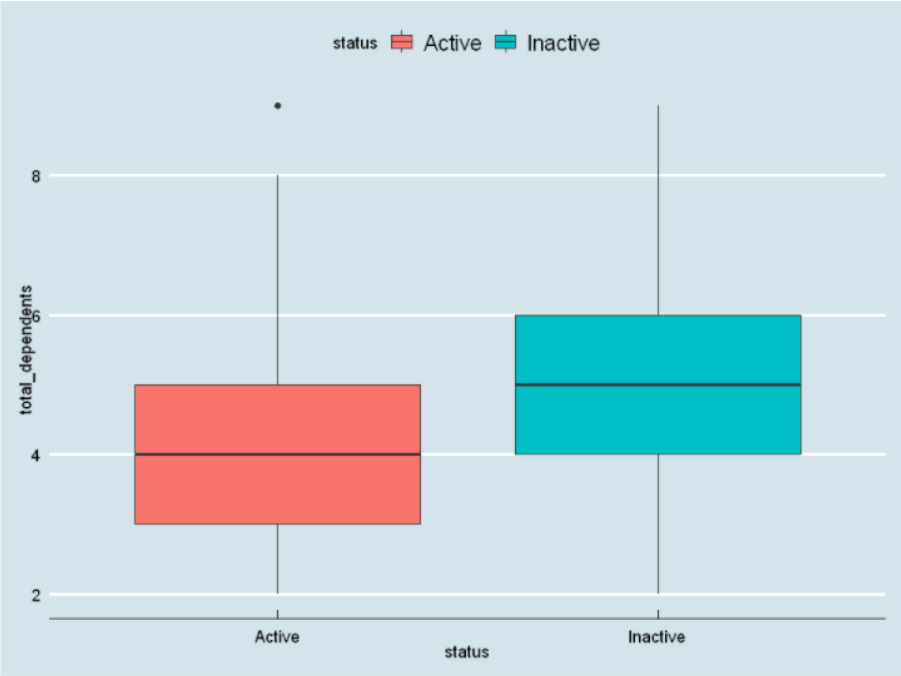
In [11]:
```
# Boxplot with attrition in the X-axis and Work Satisfaction in the y-Axis
options(repr.plot.width=8, repr.plot.height=6)


box.attrition <- data %>% dplyr::select(status, work_satisfaction, gender) %>%
ggplot(aes(x=status, y= work_satisfaction, fill=status)) + geom_boxplot(color="black")  +theme_minimal() +  facet_wrap(~gender) +
scale_fill_manual(values=c("#FA5858", "#9FF781"))

plot_grid(box.attrition)
```



In [12]:
```
# Boxplot with attrition in the X-axis and Career Satisfaction in the y-Axis
options(repr.plot.width=8, repr.plot.height=6)


box.attrition <- data %>% dplyr::select(status, career_satisfaction, gender) %>%
ggplot(aes(x=status, y= career_satisfaction, fill=status)) + geom_boxplot(color="black") + theme_minimal() + facet_wrap(~gender) +
scale_fill_manual(values=c("#FA5858", "#9FF781"))

plot_grid(box.attrition)
```

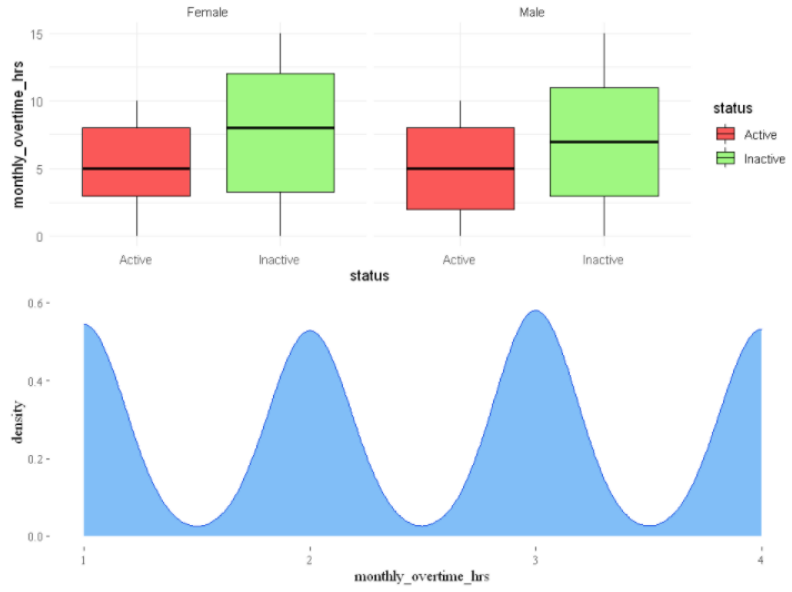


In [13]:
```
# Boxplot with attrition in the X-axis and Total Dependents in the y-Axis
options(repr.plot.width=8, repr.plot.height=6)
# Compare the total dependents of Active and Inactive employees
ggplot(data, aes(x = status, y = total_dependents, fill = status)) +
  geom_boxplot() + theme_economist()
```



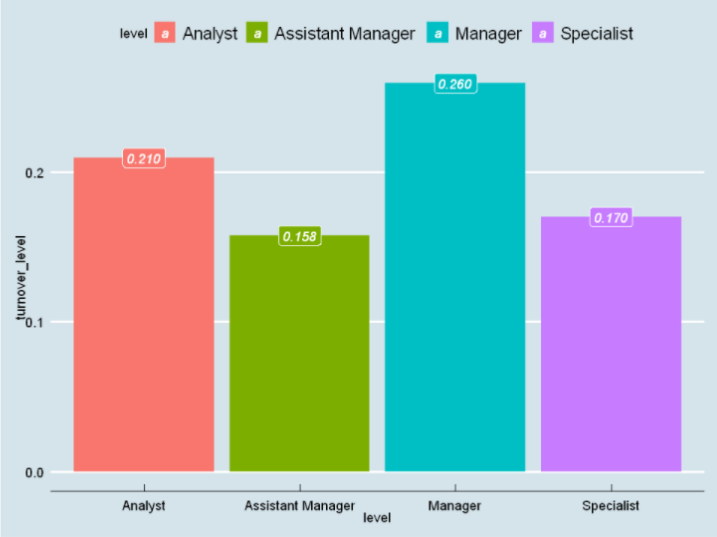
In [14]:
```
# Boxplot with attrition in the X-axis and Monthly Overtime Hours in the y-Axis
options(repr.plot.width=8, repr.plot.height=6)


box.attrition <- data %>% dplyr::select(status, monthly_overtime_hrs, gender) %>%
ggplot(aes(x=status, y= monthly_overtime_hrs, fill= status)) + geom_boxplot(color="black") + theme_minimal() + facet_wrap(~gende
scale_fill_manual(values=c("#FA585S", "#9FF781"))

# Distribution of Monthly overtime hours
dist.satisfaction <- data %>% dplyr::select(monthly_overtime_hrs) %>%
ggplot(aes(x=monthly_overtime_hrs)) + geom_density(color="#013ADF", fill="#81BEF7", trim=TRUE) + theme_tufte() + xlim(range(c(1,


plot_grid(box.attrition, dist.satisfaction, nrow=2)
```

```
Warning message:
"Removed 1280 rows containing non-finite values (stat_density)."
```



In [15]:
```
# Calculate level wise turnover rate
df_level <- data %>%
  group_by(level) %>%
  summarize(turnover_level = mean(turnover))

# Check the results
df_level

# Visualize the results
ggplot(df_level, aes(x = level, y = turnover_level, fill = level)) +
  geom_col() + geom_label(aes(label=sprintf("%0.3f", round(turnover_level, digits = 3)), fill = level), colour = "white", fontfa
theme_economist()
```

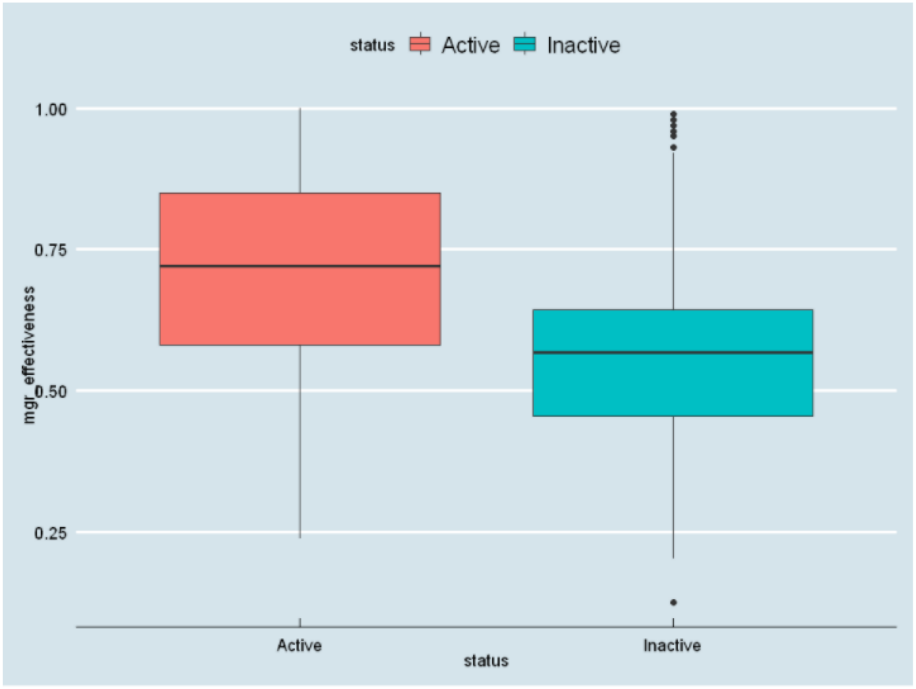
level	turnover_level
Analyst	0.2098184
Assistant Manager	0.1578947
Manager	0.2597403
Specialist	0.1702899



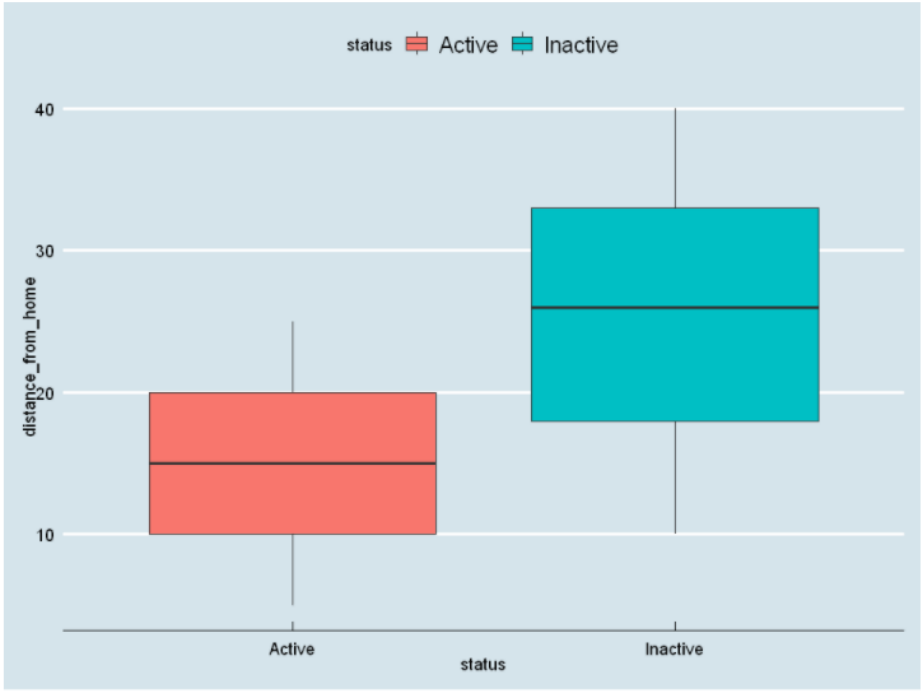
In [16]:
```
# Calculate location wise turnover rate
df_location <- data %>%
  group_by(location) %>%
  summarize(turnover_location = mean(turnover))

# Check the results
df_location

# Visualize the results
ggplot(df_location, aes(x = location, y = turnover_location, fill = location)) +
  geom_col() + geom_label(aes(label=sprintf("%0.3f", round(turnover_location, digits = 3)), fill = location), colour = "white",
theme_economist()
```

location	turnover_location
Chicago	0.2235294
New York	0.3742331
Orlando	0.1205098

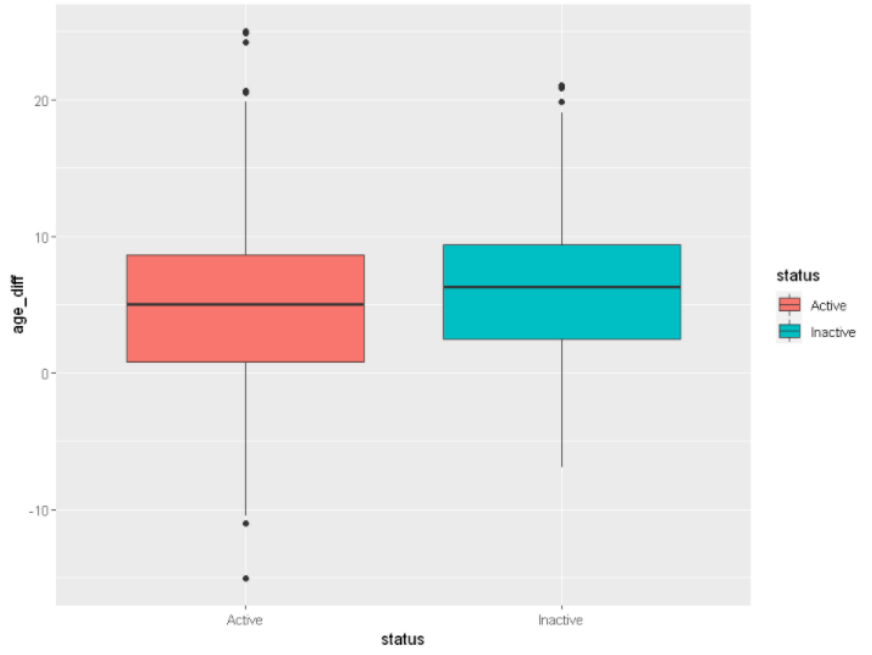
In [17]:
```
# Compare manager effectiveness scores across status
ggplot(data, aes(x = status, y = mgr_effectiveness, fill = status)) +
  geom_boxplot() + theme_economist()
```



In [18]:
```
options(repr.plot.width=8, repr.plot.height=6)
# Compare the travel distance of Active and Inactive employees
ggplot(data, aes(x = status, y = distance_from_home, fill = status)) +
  geom_boxplot() +theme_economist()
```

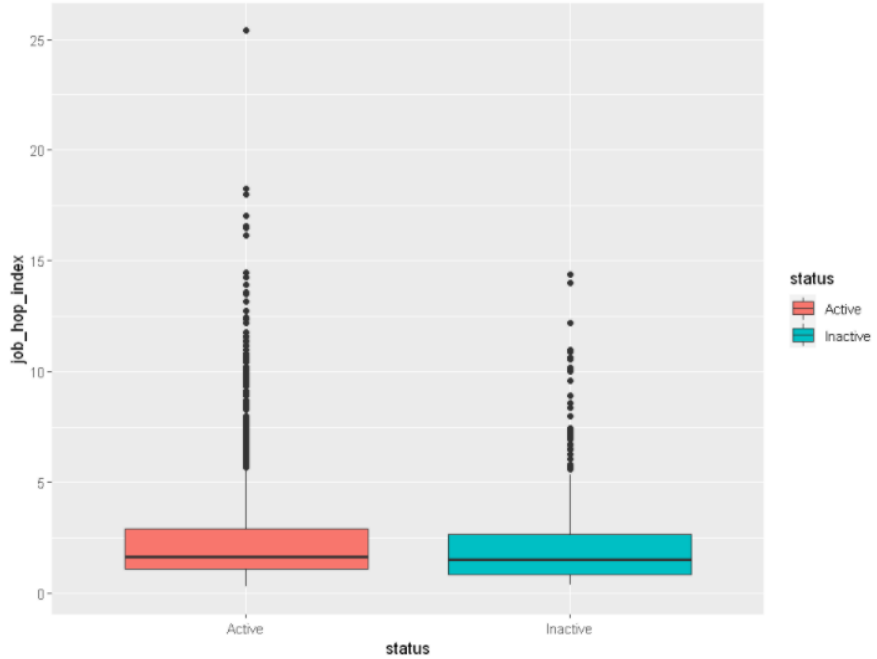


In [20]:
```
#feature engineering
# Add age_diff between employees and managers
emp_age_diff <- data %>%
  mutate(age_diff = mgr_age - emp_age)

# Plot the distribution of age difference
ggplot(emp_age_diff, aes(x = status, y = age_diff, fill = status)) +
  geom_boxplot()
```

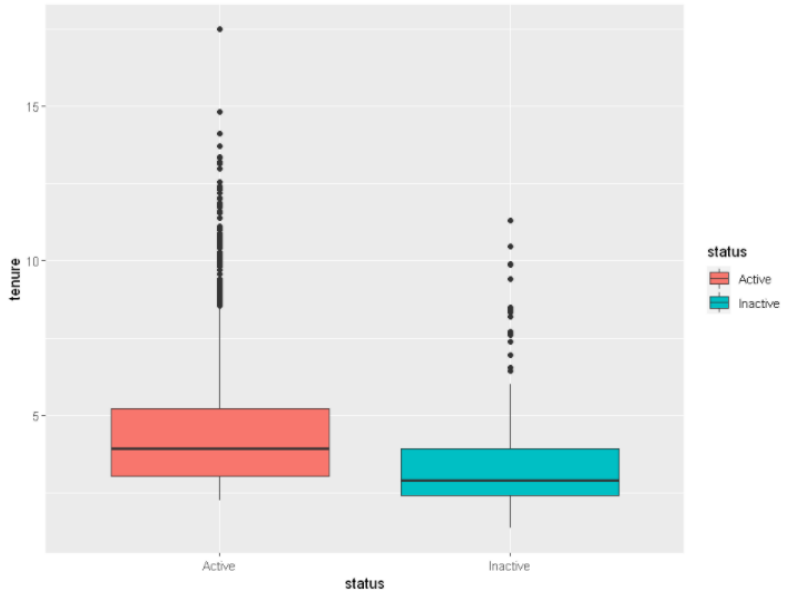


In [21]:
```
# Add job_hop_index
emp_jhi <- emp_age_diff %>%
  mutate(job_hop_index =  total_experience / no_previous_companies_worked)

# Compare job hopping index of Active and Inactive employees
ggplot(emp_jhi, aes(x = status, y = job_hop_index, fill = status)) +
  geom_boxplot()
```

```
Warning message:
"Removed 186 rows containing non-finite values (stat_boxplot)."
```

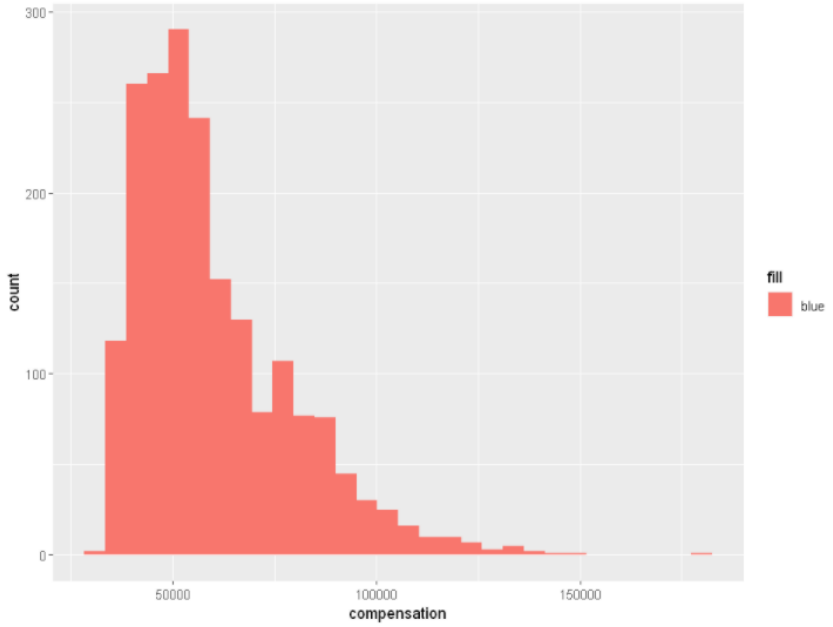


In [22]:
```
# Add tenure (time spent at the company as of now)
emp_tenure <- emp_jhi %>%
  mutate(tenure = ifelse(status == "Active",
                         time_length(interval(dmy(`date_of_joining`), dmy(`cutoff_date`)),
                                     "years"),
                         time_length(interval(dmy(`date_of_joining`), dmy(`last_working_date`)),
                                     "years")))

# Compare tenure of active and inactive employees
ggplot(emp_tenure, aes(x = status, y = tenure, fill = status)) +
  geom_boxplot()
```

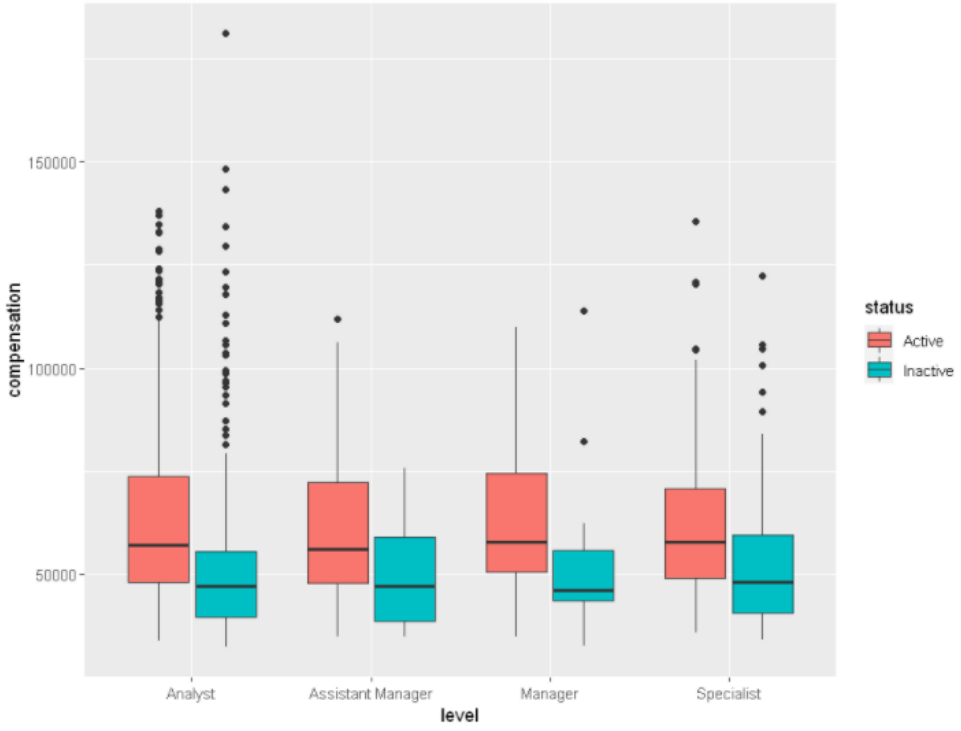


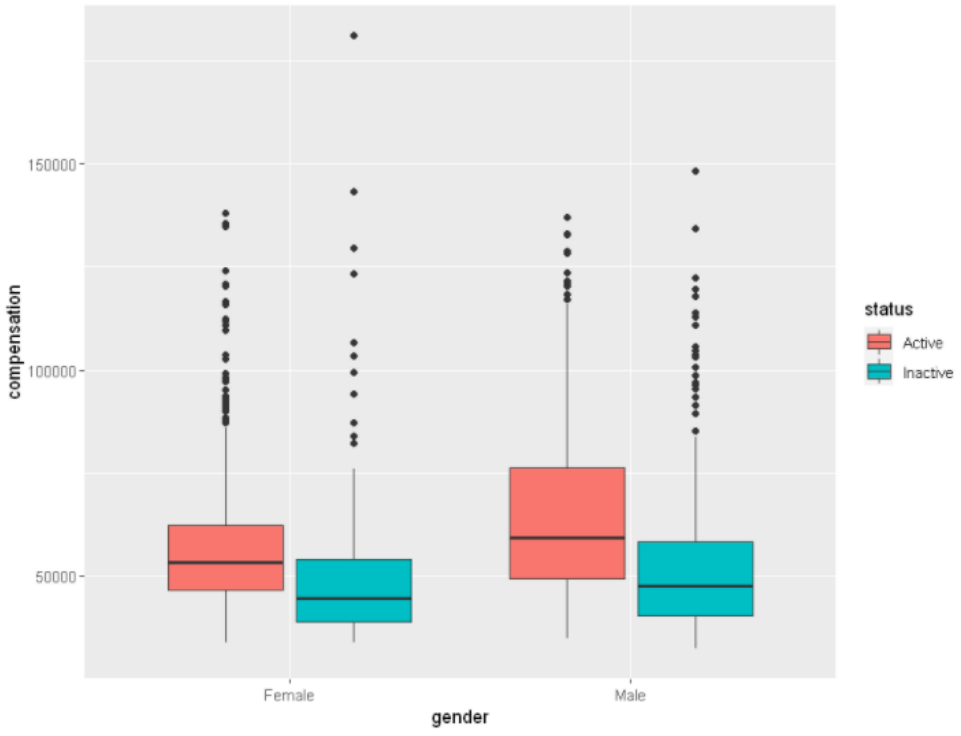
In [23]:
```
# Plot the distribution of compensation
ggplot(emp_tenure, aes(x = compensation, fill= "blue")) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

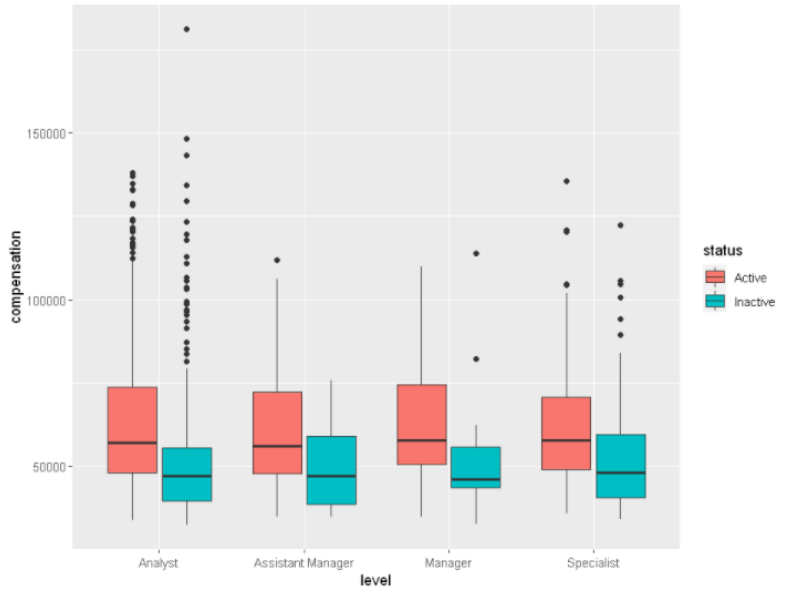


In [24]:
```
# Plot the distribution of compensation across levels
ggplot(emp_tenure,
       aes(x = level, y = compensation, fill = status)) +
  geom_boxplot()

# Plot the distribution of compensation across gender
ggplot(emp_tenure,
       aes(x = gender, y = compensation, fill = status)) +
  geom_boxplot()
```

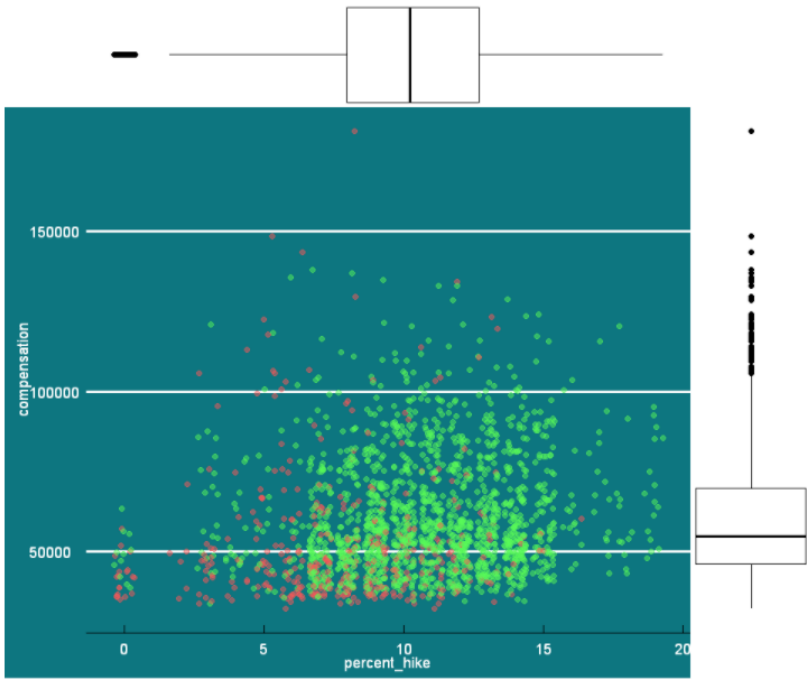




In [25]:
```
# Compare compensation of Active and Inactive employees across levels
ggplot(emp_tenure,
       aes(x = level, y = compensation, fill = status)) +
  geom_boxplot()
```



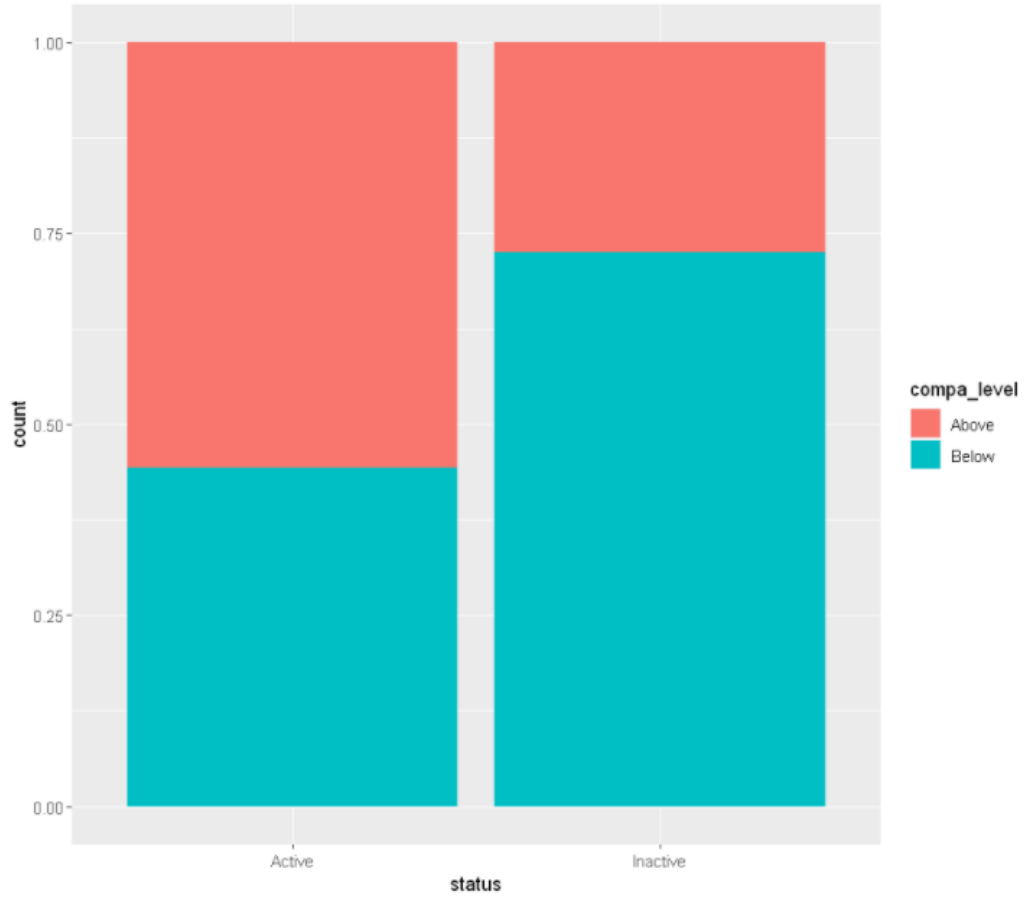
In [26]:
```
options(repr.plot.width=8, repr.plot.height=7)

per.sal <- emp_tenure %>% select(status, percent_hike, compensation) %>%
ggplot(aes(x=percent_hike, y=compensation)) + geom_jitter(aes(col=status), alpha=0.5) +
theme_economist() + theme(legend.position="none") + scale_color_manual(values=c("#58FA58", "#FA5858")) +
labs(title="Income and its Impact on Attrition") + theme(plot.title=element_text(hjust=0.5, color="white"), plot.background=elem
                                                        axis.text.x=element_text(colour="white"), axis.text.y=element_text(colou
                                                        axis.title=element_text(colour="white"))

#plot_grid(per.sal)

p3 <- ggMarginal(per.sal, type="boxplot")

p3
```



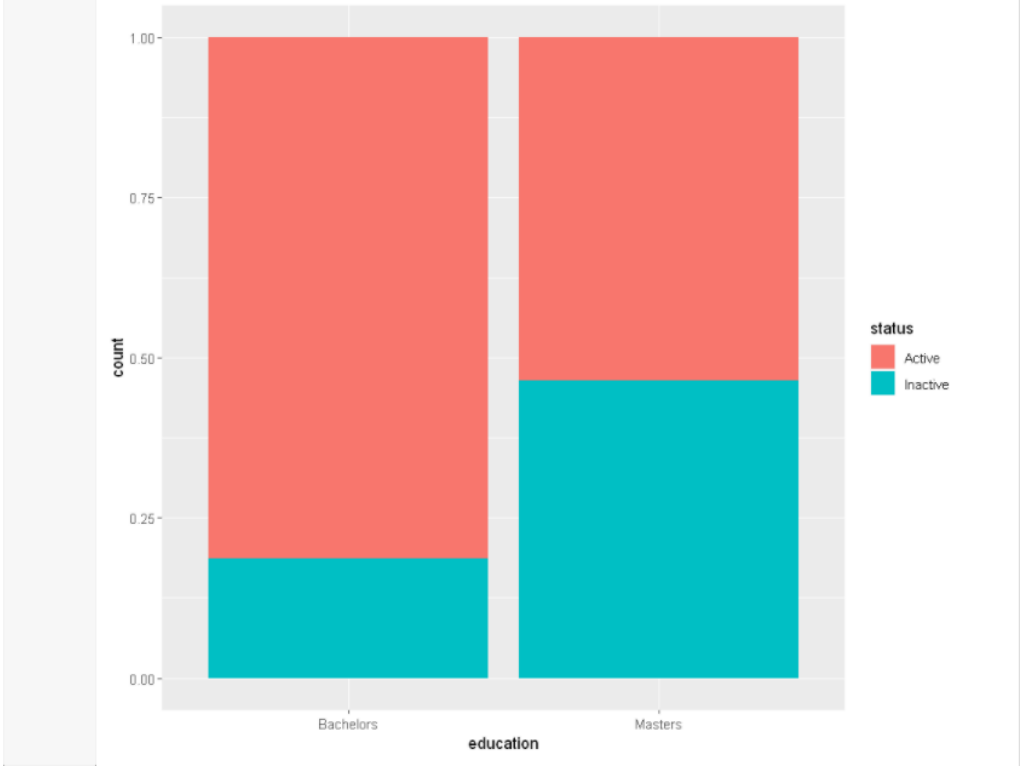
In [27]:
```
# Add median_compensation and compa_ratio
emp_compa_ratio <- emp_tenure %>%
  group_by(level) %>%
  mutate(median_compensation = median(compensation),
         compa_ratio = compensation/ median_compensation)

# Look at the median compensation for each level
emp_compa_ratio %>%
  distinct(level, median_compensation)
```

level	median_compensation
Analyst	54684
Assistant Manager	54210
Specialist	56442
Manager	54456

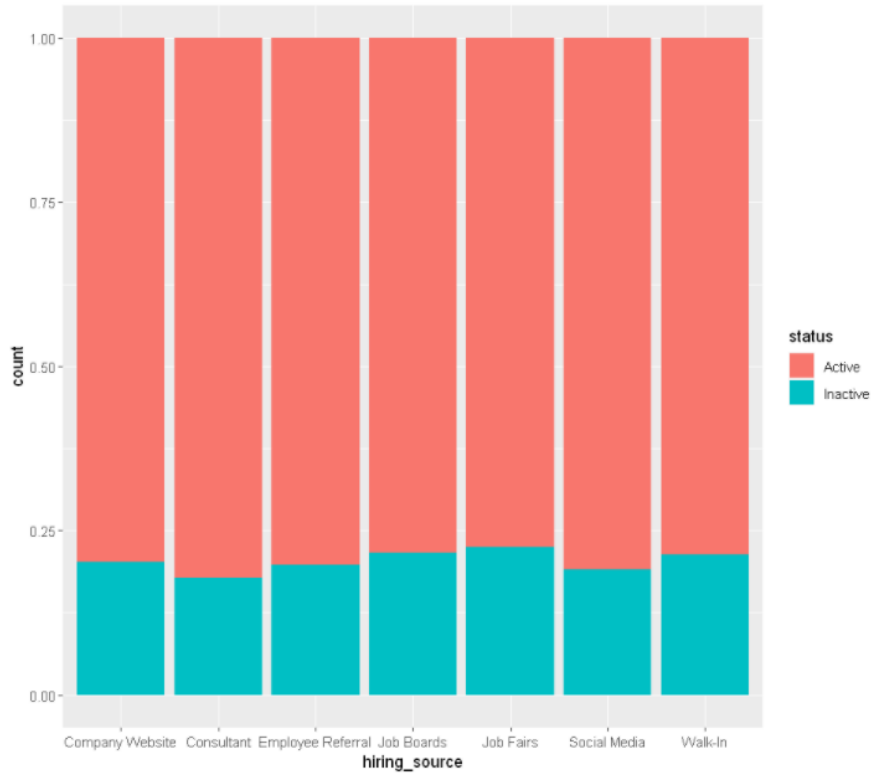
In [28]:
```
# Add compa_level for an understanding of relative distribution of compensation
emp_final <- emp_compa_ratio %>%
  mutate(compa_level = if_else(compa_ratio > 1, "Above", "Below"))
# Compare compa_level for Active & Inactive employees
ggplot(emp_final, aes(x = status, fill = compa_level)) +
  geom_bar(position = "fill")
```



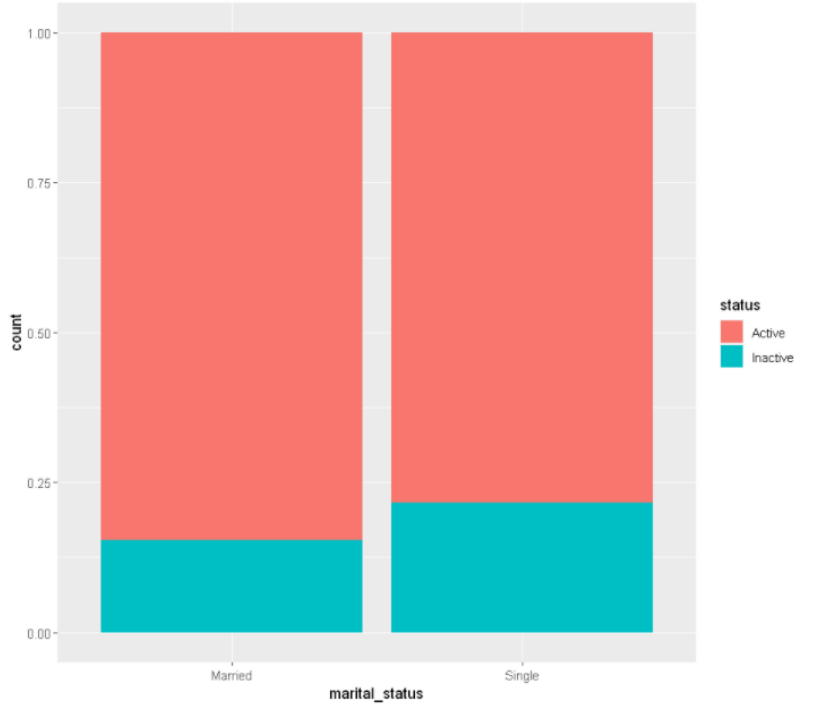
In [29]:
```
# Compare education for Active & Inactive employees
ggplot(emp_final, aes(x = education, fill = status)) +
  geom_bar(position = "fill")
```



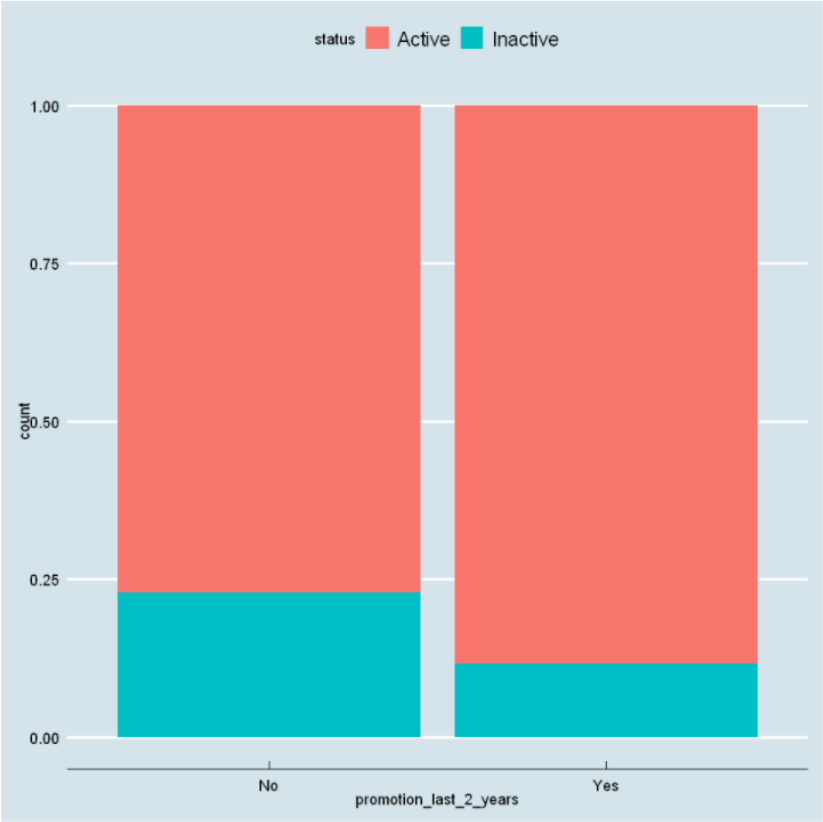
In [30]:
```
# Compare hiring source for Active & Inactive employees
ggplot(emp_final, aes(x = hiring_source, fill = status)) +
  geom_bar(position = "fill")
```



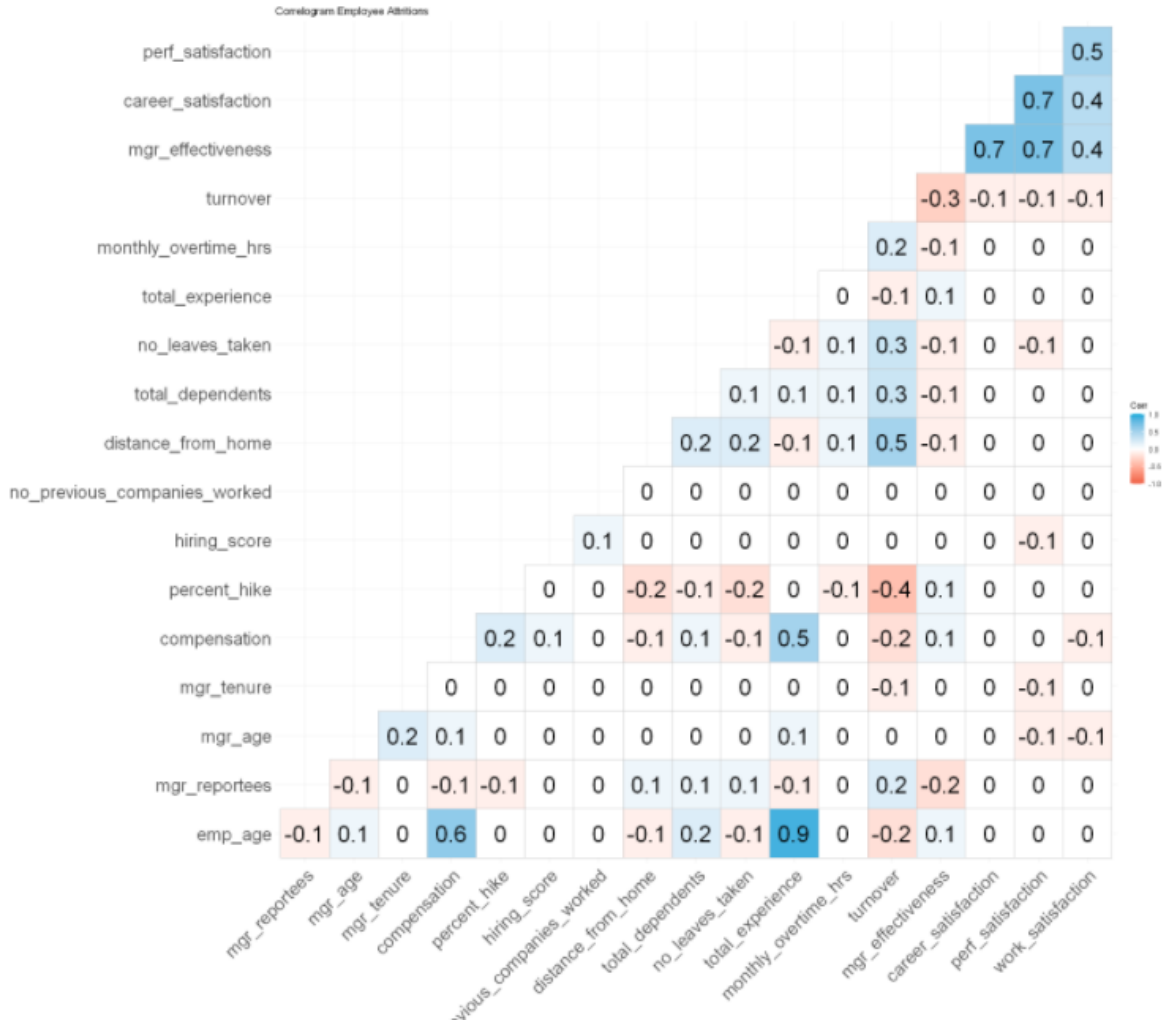
In [31]:
```
# Compare marital status for Active & Inactive employees
ggplot(emp_final, aes(x = marital_status, fill = status)) +
  geom_bar(position = "fill")
```



In [32]:
```
options(repr.plot.width=8, repr.plot.height=8)
# Find out whether Active & Inactive employees have had a promotion in the last 2 years
ggplot(emp_final, aes(x = promotion_last_2_years, fill = status)) +
  geom_bar(position = "fill") + theme_economist()
```

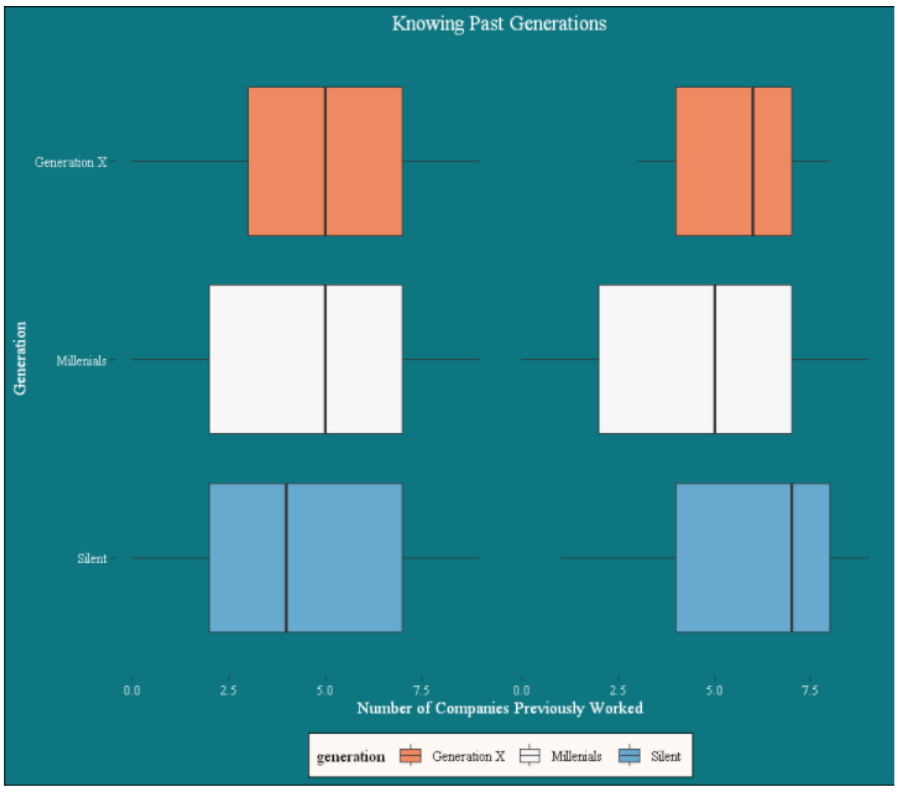


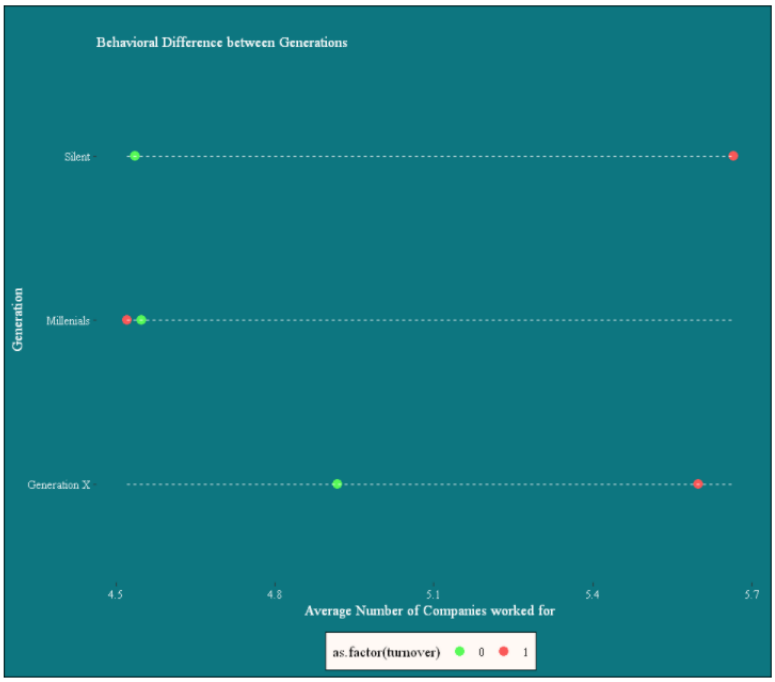
```
# # Let's have a better understanding about each feature through a correlation plot
options(repr.plot.width=20, repr.plot.height=20)

nums <- select_if(data, is.numeric)

corr <- round(cor(nums), 1)

ggcorrplot(corr,
           type = "lower",
           lab = TRUE,
           lab_size = 10,
           method="square",
           colors = c("tomato2", "white", "#01A9DB"),
           title="Correlogram Employee Attritions",
           ggtheme=theme_minimal(),
           tl.cex = 22, tl.col = "black")
```



In [34]:
```
# Distribution of Number of Companies Worked by Attrition and Age
# We want to see if young people have worked in more companies than the older generation
# This might prove that the millenials tend to be more picky with regards to jobs than the older generation.
options(repr.plot.width=8, repr.plot.height=7)

# First we must create categoricals variables based on Age
data$generation <- ifelse(data$emp_age<37,"Millenials",
ifelse(data$emp_age>=38 & data$emp_age<54,"Generation X",
ifelse(data$emp_age>=54 & data$emp_age<73,"Boomers","Silent"
)))


# Let's see the distribution by generation now
generation.dist <- data %>% dplyr::select(generation, no_previous_companies_worked, turnover) %>%
ggplot() + geom_boxplot(aes(x=reorder(generation, no_previous_companies_worked, FUN=median),
                            y=no_previous_companies_worked, fill=generation)) +
theme_tufte() + facet_wrap(~turnover) +
scale_fill_brewer(palette="RdBu") + coord_flip() +
labs(title="Knowing Past Generations",x="Generation", y="Number of Companies Previously Worked") +
theme(legend.position="bottom", legend.background = element_rect(fill="#FFF9F5",
                                size=0.5, linetype="solid",
                                colour ="black")) + theme(strip.background = element_blank(), strip.text.x = element_blank(),
    plot.title=element_text(hjust=0.5, color="white"), plot.background=element_rect(fill="#0D7680"),
                                          axis.text.x=element_text(colour="white"), axis.text.y=element_text(colou
                                          axis.title=element_text(colour="white"))


# 2.69
overall.avg <- data%>% dplyr::select(generation, no_previous_companies_worked) %>% summarize(avg_ov=mean(no_previous_companies_w
```

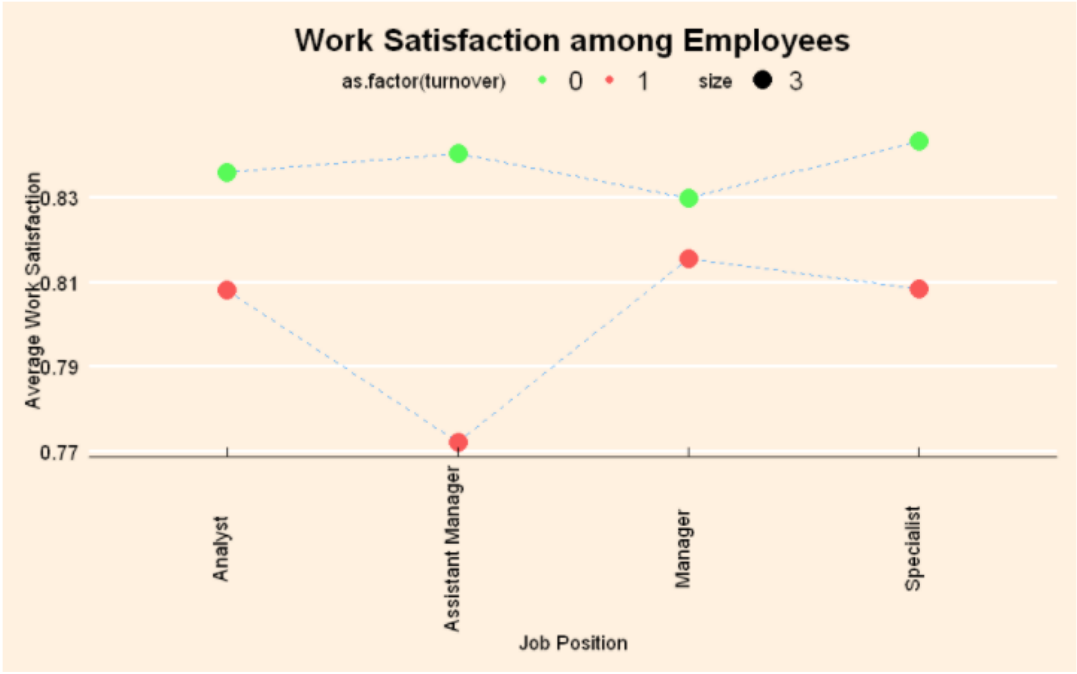
In [35]:
```
# Let's find the Average Numbers of Companies worked by Generation
avg.comp <- data %>% dplyr::select(generation, no_previous_companies_worked, turnover) %>% group_by(generation, turnover) %>%
summarize(avg=mean(no_previous_companies_worked)) %>%
ggplot(aes(x=generation, y=avg, color= as.factor(turnover))) +
  geom_point(size=3) + theme_tufte() +  # Draw points
  geom_segment(aes(x=generation,
                   xend=generation,
                   y=min(avg),
                   yend=max(avg)),
               linetype="dashed",
               size=0.1,
              color="white") +
  labs(title="",
       subtitle="Behavioral Difference between Generations",
       y="Average Number of Companies worked for",
      x="Generation") +
  coord_flip() + scale_color_manual(values=c("#58FA58", "#FA5858")) +
theme(legend.position="bottom", legend.background = element_rect(fill="#FFF9F5",
                                size=0.5, linetype="solid",
                                colour ="black")) + theme(strip.background = element_blank(), strip.text.x = element_blank(),
    plot.title=element_text(hjust=0.5, color="white"),plot.subtitle=element_text(color="white"), plot.background=element_rect(f
                                          axis.text.x=element_text(colour="white"), axis.text.y=element_text(colou
                                          axis.title=element_text(colour="white"))


#plot_grid(generation.dist, avg.comp, nrow=2)

plot_grid(generation.dist)
plot_grid(avg.comp)
```

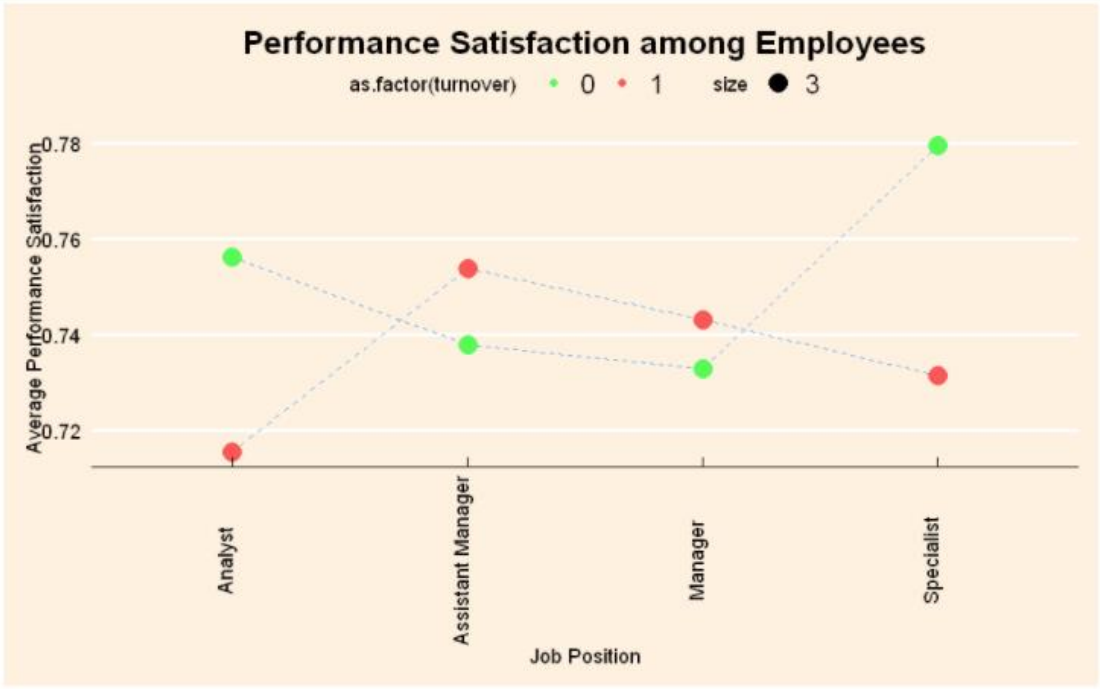


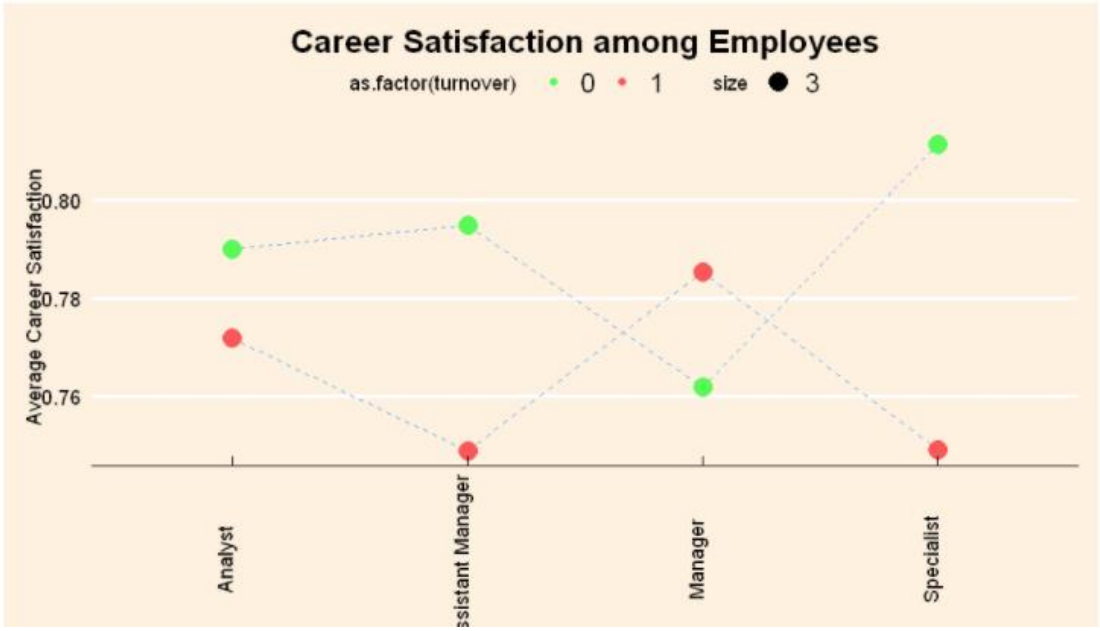

Behavioral Difference between Generations
Silent
Millenials
Generation X
Generation
4.5
4.8
5.1
5.4
5.7
Average Number of Companies worked for
as.factor(turnover)
0
1

```r
options(repr.plot.width=8, repr.plot.height=5)

env.attr <- data %>% dplyr::select(work_satisfaction, level, turnover) %>% group_by(level, turnover) %>%
summarize(avg.env=mean(work_satisfaction))

ggplot(env.attr, aes(x=level, y=avg.env)) + geom_line(aes(group=turnover), color="#58ACFA", linetype="dashed") +
geom_point(aes(color=as.factor(turnover), size=3)) +  theme_economist() + theme(plot.title=element_text(hjust=0.5), axis.text.x=
                                                      plot.background=element_rect(fill="#FFF1E0")) +
labs(title="Work Satisfaction among Employees", y="Average Work Satisfaction", x="Job Position") + scale_color_manual(values=c("

# Performance satisfaction by Job Role
options(repr.plot.width=8, repr.plot.height=5)

env.attr <- data %>% dplyr::select(perf_satisfaction, level, turnover) %>% group_by(level, turnover) %>%
summarize(avg.env=mean(perf_satisfaction))

ggplot(env.attr, aes(x=level, y=avg.env)) + geom_line(aes(group=turnover), color="#58ACFA", linetype="dashed") +
geom_point(aes(color=as.factor(turnover), size=3)) +  theme_economist() + theme(plot.title=element_text(hjust=0.5), axis.text.x=
                                                      plot.background=element_rect(fill="#FFF1E0")) +
labs(title="Performance Satisfaction among Employees", y="Average Performance Satisfaction", x="Job Position") + scale_color_man
# Career Satisfaction  by JobRole
options(repr.plot.width=8, repr.plot.height=5)

env.attr <- data %>% dplyr::select(career_satisfaction, level, turnover) %>% group_by(level, turnover) %>%
summarize(avg.env=mean(career_satisfaction))

ggplot(env.attr, aes(x=level, y=avg.env)) + geom_line(aes(group=turnover), color="#58ACFA", linetype="dashed") +
geom_point(aes(color=as.factor(turnover), size=3)) +  theme_economist() + theme(plot.title=element_text(hjust=0.5), axis.text.x=
                                                      plot.background=element_rect(fill="#FFF1E0")) +
labs(title="Career Satisfaction among Employees", y="Average Career Satisfaction", x="Job Position") + scale_color_manual(values
```

```
`summarise()` has grouped output by 'level'. You can override using the `.groups` argument.
`summarise()` has grouped output by 'level'. You can override using the `.groups` argument.
```

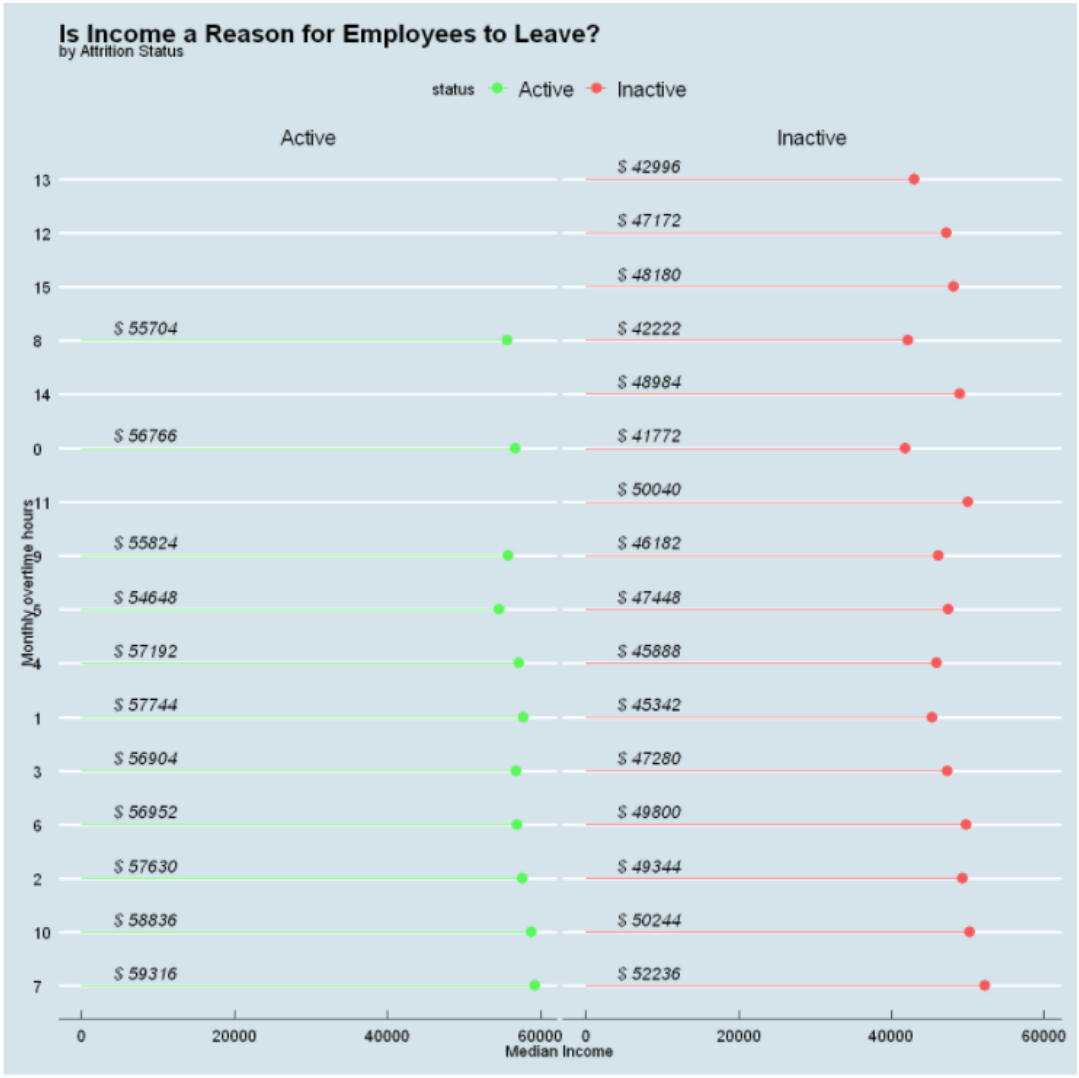



Performance Satisfaction among Employees
as.factor(turnover) 0 1 size 3
Average Performance Satisfaction
0.78
0.76
0.74
0.72
Analyst
Assistant Manager
Manager
Specialist
Job Position

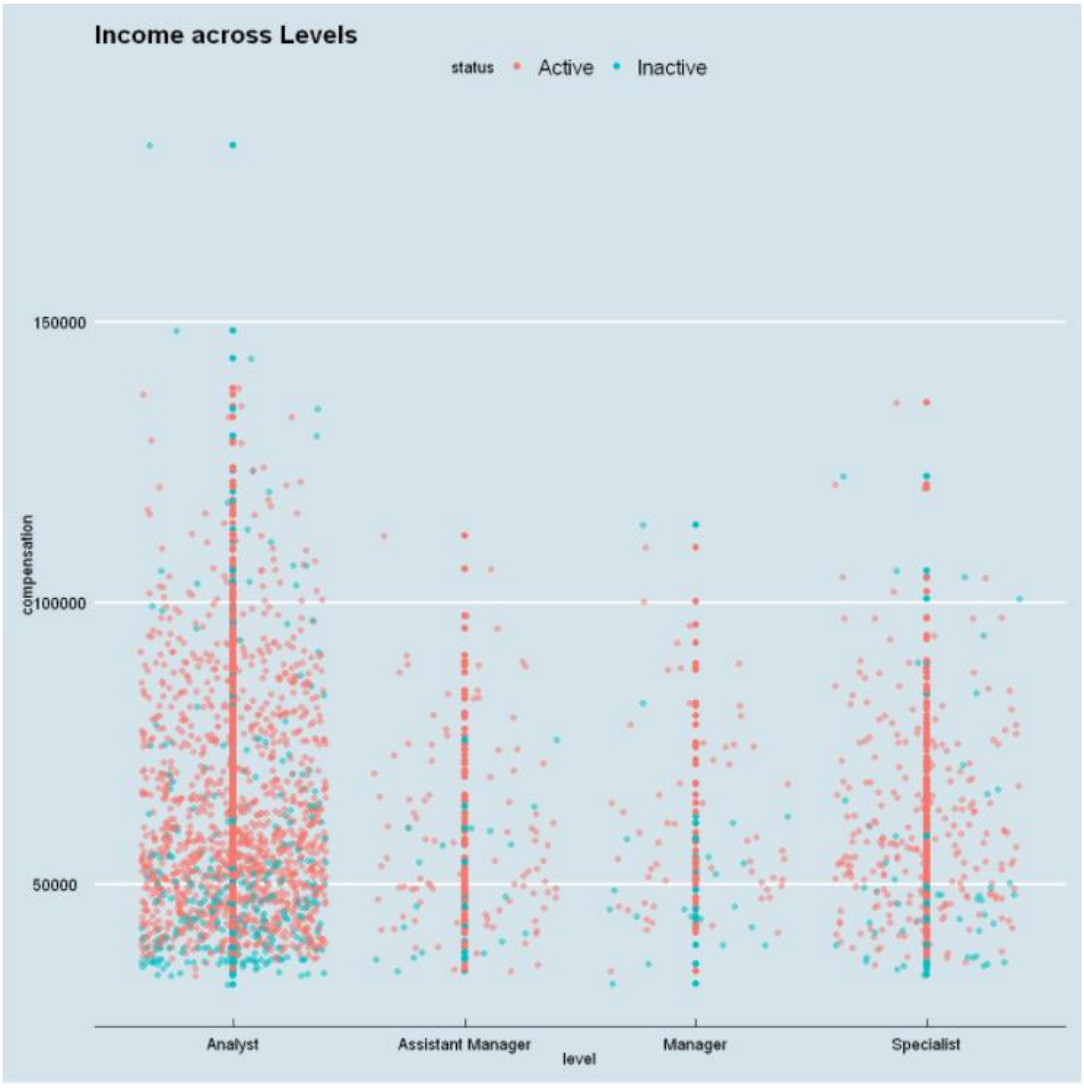

Career Satisfaction among Employees
as.factor(turnover) 0 1 size 3
Average Career Satisfaction
0.80
0.78
0.76
Analyst
Manager
Specialist

In [37]:
```
#Let us visualize average compensation for each category of monthly overtime hours worked

options(repr.plot.width=10, repr.plot.height=10)

# Turn the column to factor: One because it should not be considered an integer
# Two: Will help us sort in an orderly manner.
data$monthly_overtime_hrs <- as.factor(data$monthly_overtime_hrs)

high.inc <- data %>% dplyr::select(monthly_overtime_hrs, compensation, status) %>% group_by(monthly_overtime_hrs, status) %>%
summarize(med=median(compensation)) %>%
ggplot(aes(x=fct_reorder(monthly_overtime_hrs, -med), y=med, color=status)) +
  geom_point(size=3) +
  geom_segment(aes(x=monthly_overtime_hrs,
                   xend=monthly_overtime_hrs,
                   y=0,
                   yend=med)) + facet_wrap(~status) +
  labs(title="Is Income a Reason for Employees to Leave?",
       subtitle="by Attrition Status",
      y="Median Income",
      x="Monthly overtime hours") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6), plot.title=element_text(hjust=0.5), strip.background = element_blank(),
  strip.text = element_blank()) +
coord_flip() + theme_economist()  + scale_color_manual(values=c("#58FA58", "#FA5858")) +
geom_text(aes(x=monthly_overtime_hrs, y=0.01, label= paste0("$ ", round(med,2))),
            hjust=-0.5, vjust=-0.5, size=4,
            colour="black", fontface="italic",
          angle=360)


high.inc
```

```
`summarise()` has grouped output by 'monthly_overtime_hrs'. You can override using the `.groups` argument.
```



In [38]:
```
#category plot of salary per job role
ggplot(data, aes(x=level, y=compensation, color=status)) + geom_jitter(aes(col=status), alpha=0.5)+ labs(title="Income across Le
    geom_point(size=2) + theme_economist()

# per.sal <- emp_tenure %>% select(status, percent_hike, compensation) %>%
# ggplot(aes(x=percent_hike, y=compensation)) + geom_jitter(aes(col=status), alpha=0.5) +
# theme_economist() + theme(legend.position="none") + scale_color_manual(values=c("#58FA58", "#FA5858")) +
 #theme(plot.title=element_text(hjust=0.5, color="white"), plot.background=element_rect(fill="#0D7680"),
#                                                    axis.text.x=element_text(colour="white"), axis.text.y=element_text(col
#                                                    axis.title=element_text(colour="white"))
```



In [41]:
```
# Calculate salary level
df_salary <- data %>%
  group_by(level) %>%
  summarize(salary_level = mean(compensation))

# Check the results
df_salary
```

level	salary_level
Analyst	59979.50
Assistant Manager	58486.95
Manager	60007.48
Specialist	60113.43

Preparing the Data

```
$ hiring_source               <chr> "Consultant", "Job Fairs", "Consultant...
$ no_previous_companies_worked <dbl> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 3, 6, 2,...
$ distance_from_home          <dbl> 14, 21, 15, 9, 25, 23, 17, 16, 22, 22,...
$ total_dependents            <dbl> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, 5, 4,...
```

In [43]:
```
#remove highly correlated and useless variables
drops <- c("emp_id","status", "emp_age", "mgr_age", "hiring_source", "job_hop_index", "date_of_joining",
           "last_working_date", "cutoff_date", "department", "mgr_id")
emp_final <- emp_final[ , !(names(emp_final) %in% drops)]
```

In [44]:
```
#make all categorical variables into binary
install.packages("fastDummies")
library(fastDummies)
#remove first dummy to avoid dummy variable trap for multicollinearity
emp_final <- dummy_cols(emp_final, remove_first_dummy = TRUE)
```

```
Installing package into 'C:/Users/bhavy/Documents/R/win-library/3.6'
(as 'lib' is unspecified)

package 'fastDummies' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\bhavy\AppData\Local\Temp\Rtmpag78e6\downloaded_packages

Warning message:
"package 'fastDummies' was built under R version 3.6.3"
```

In [45]:
```
emp_final %>% dplyr::rename_all(list(~make.names(.)))
```

location	level	gender	rating	mgr_rating	mgr_reportees	mgr_tenure	compensation	percent_hike	hiring_score	...	rating_Excellent	rating_Una
New York	Analyst	Female	Above Average	Acceptable	9	3.17	64320	10	70	...	0	
Chicago	Analyst	Female	Acceptable	Excellent	4	7.92	48204	8	70	...	0	
Orlando	Analyst	Female	Acceptable	Above Average	6	4.38	85812	11	77	...	0	
Chicago	Analyst	Male	Acceptable	Acceptable	10	2.87	49536	8	71	...	0	
Orlando	Analyst	Male	Acceptable	Acceptable	11	12.95	75576	12	70	...	0	
Orlando	Assistant Manager	Male	Below Average	Above Average	19	10.88	56904	8	75	...	0	
Chicago	Specialist	Male	Acceptable	Above Average	21	4.01	38772	12	72	...	0	
Orlando	Analyst	Male	Above Average	Above Average	9	4.21	52320	9	70	...	0	
New York	Analyst	Female	Acceptable	Acceptable	12	1.27	50940	9	70	...	0	

In [46]:
```
#remove original categorical variable
emp_final <- emp_final %>% dplyr::select(-location)
emp_final <- emp_final %>% dplyr::select(-rating)
emp_final <- emp_final %>% dplyr::select(-mgr_rating)
emp_final <- emp_final %>% dplyr::select(-level)
emp_final <- emp_final %>% dplyr::select(-gender)
emp_final <- emp_final %>% dplyr::select(-marital_status)
emp_final <- emp_final %>% dplyr::select(-education)
emp_final <- emp_final %>% dplyr::select(-promotion_last_2_years)
emp_final <- emp_final %>% dplyr::select(-compa_level)

#remove spaces in between automatically generated column names
names(emp_final)[names(emp_final) == "mgr_rating_Below Average"] <- "mgr_rating_Below_Average"
names(emp_final)[names(emp_final) == "mgr_rating_Above Average"] <- "mgr_rating_Above_Average"
names(emp_final)[names(emp_final) == "rating_Below Average"] <- "rating_Below_Average"
names(emp_final)[names(emp_final) == "rating_Above Average"] <- "rating_Above_Average"
names(emp_final)[names(emp_final) == "level_Assistant Manager"] <- "level_Assistant_Manager"
names(emp_final)[names(emp_final) == "location_New York"] <- "location_New_York"
emp_final
```

mgr_reportees	mgr_tenure	compensation	percent_hike	hiring_score	no_previous_companies_worked	distance_from_home	total_dependents	no_leaves_take
9	3.17	64320	10	70	0	14	2	

In [47]:
```
# Load caret
library(caret)

# Set seed of 567
set.seed(567)

# Store row numbers for training dataset: index_train
index_train <- createDataPartition(emp_final$turnover, p = 0.7, list = FALSE)

# Create training dataset: train_set
train_set <- emp_final[index_train, ]

# Create testing dataset: test_set
test_set <- emp_final[-index_train, ]
```

```
Loading required package: lattice

Attaching package: 'caret'

The following object is masked from 'package:purrr':

    lift
```

In [48]:
```
# Calculate turnover proportion in train_set
train_set %>%
  count(turnover) %>%
  mutate(prop = n / sum(n))

# Calculate turnover proportion in test_set
test_set %>%
  count(turnover) %>%
  mutate(prop = n / sum(n))

#they should be the same proportion
```

turnover	n	prop
0	1094	0.7997076
1	274	0.2002924

turnover	n	prop
0	483	0.7901024
1	123	0.2098976

In [49]:
```
glimpse(train_set)
```

```
Rows: 1,368
Columns: 38
$ mgr_reportees                <dbl> 9, 4, 6, 10, 11, 19, 21, 9, 12, 17, 13...
$ mgr_tenure                   <dbl> 3.17, 7.92, 4.38, 2.87, 12.95, 10.88, ...
$ compensation                 <dbl> 64320, 48204, 85812, 49536, 75576, 569...
$ percent_hike                 <dbl> 10, 8, 11, 8, 12, 8, 12, 9, 9, 11, 7, ...
$ hiring_score                 <dbl> 70, 70, 77, 71, 70, 75, 72, 70, 70, 70...
$ no_previous_companies_worked <dbl> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 6, 6, 0,...
$ distance_from_home           <dbl> 14, 21, 15, 9, 25, 23, 17, 16, 22, 18,...
$ total_dependents             <dbl> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, 5, 2,...
$ no_leaves_taken              <dbl> 2, 10, 18, 19, 25, 15, 10, 20, 22, 24,...
$ total_experience             <dbl> 6.86, 4.88, 8.55, 4.76, 8.06, 13.72, 5...
$ monthly_overtime_hrs         <dbl> 1, 5, 3, 8, 1, 7, 2, 10, 2, 8, 3, 3, 4...
$ turnover                     <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,...
$ mgr_effectiveness            <dbl> 0.730, 0.581, 0.770, 0.240, 0.710, 0.5...
$ career_satisfaction          <dbl> 0.73, 0.72, 0.85, 0.42, 0.78, 0.88, 0....
$ perf_satisfaction            <dbl> 0.73, 0.84, 0.80, 0.33, 0.67, 0.81, 0....
$ work_satisfaction            <dbl> 0.75, 0.85, 0.87, 0.85, 0.80, 0.86, 0....
$ age_diff                     <dbl> 18.98, 10.01, 2.38, 2.15, 3.05, 2.84, ...
$ tenure                       <dbl> 3.821918, 5.271233, 9.161644, 3.616438...
$ median_compensation          <dbl> 54684, 54684, 54684, 54684, 54684, 542...
$ compa_ratio                  <dbl> 1.1762124, 0.8815010, 1.5692341, 0.905...
$ location_New_York            <int> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...
```

Prediction Models

In [50]:
```
# Set up multiple cores as separate workers and then make them a cluster.
workers <- detectCores()
cluster <- makeCluster(workers, type = "SOCK")
registerDoSNOW(cluster)

workers
```

8

In [51]:
```
# Function to calculate the average Brier score.
brier.score <- function(predictions, realizations) {
     return(mean((predictions - realizations)^2))
}

# Establish a reference brier score
# Naive forecast: use the propotion of default loans in the traning set as the prediction
# for all the loans in the testing set.
train.churn.rate <- mean(train_set$turnover)
test.realizations <- test_set$turnover
naive.pred <- rep(train.churn.rate, length(test.realizations))
brier.ref <- brier.score(naive.pred, test.realizations)

# Function to calculate the Brier skill score
skill.score <- function(predictions, realizations, brier.ref) {
    # calculate the Brier score for your predictions.
    brier.score <- brier.score(predictions, realizations)
    return(1 - brier.score/ brier.ref)
}
```

In [52]:
```
## Step 4: Apply different models to predict default rates
(nrowTrain <- nrow(train_set))
(nrowSmallTrain <- round(nrowTrain*0.75))
(nrowvalid <- nrowTrain - nrowSmallTrain)

set.seed(201)

# generate row numbers of the training set.
rowIndicesSmallTrain <- sample(1:nrowTrain, size = nrowSmallTrain, replace = FALSE)
smalltrain.df <- train_set[rowIndicesSmallTrain, ]
valid.df <- train_set[-rowIndicesSmallTrain, ]


# print out column names
print(colnames(smalltrain.df))
```

1368

1026

342

```
 [1] "mgr_reportees"               "mgr_tenure"
 [3] "compensation"                "percent_hike"
 [5] "hiring_score"                "no_previous_companies_worked"
 [7] "distance_from_home"          "total_dependents"
 [9] "no_leaves_taken"             "total_experience"
[11] "monthly_overtime_hrs"        "turnover"
[13] "mgr_effectiveness"           "career_satisfaction"
[15] "perf_satisfaction"           "work_satisfaction"
[17] "age_diff"                    "tenure"
[19] "median_compensation"         "compa_ratio"
[21] "location_New_York"           "location_Orlando"
[23] "level_Assistant_Manager"     "level_Manager"
[25] "level_Specialist"            "gender_Male"
[27] "rating_Acceptable"           "rating_Below_Average"
[29] "rating_Excellent"            "rating_Unacceptable"
[31] "mgr_rating_Acceptable"       "mgr_rating_Below_Average"
[33] "mgr_rating_Excellent"        "mgr_rating_Unacceptable"
[35] "marital_status_Single"       "education_Masters"
[37] "promotion_last_2_years_Yes"  "compa_level_Below"
```

In [53]:
```
### Model1: Logistic Regression
# Train logistic regression model
# Remember to set family = "binomial", which fit the logistic regression rather than linear regression.
# Build a multiple logistic regression model
logit.reg <- glm(turnover ~ . - level_Specialist,  family = "binomial",
                 data = train_set)

# Print summary
summary(logit.reg)

# Generate predictions in the validation set
# set type = "response", which gives the probability rather than the outcome variable
pred.logit.reg <- predict(logit.reg, valid.df, type = "response")

# Calculate the Brier Skill Score
skill.score(pred.logit.reg, valid.df$turnover, brier.ref)
```

```
Call:
glm(formula = turnover ~ . - level_Specialist, family = "binomial",
    data = train_set)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.58673  -0.16539  -0.04680  -0.00647   3.04322

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   9.781e+01  4.354e+01   2.246 0.024685 *
mgr_reportees                 9.537e-02  3.031e-02   3.146 0.001656 **
mgr_tenure                   -2.588e-02  4.401e-02  -0.588 0.556429
compensation                  1.718e-03  7.351e-04   2.338 0.019404 *
percent_hike                 -5.964e-01  8.058e-02  -7.402 1.34e-13 ***
hiring_score                  7.140e-02  4.337e-02   1.646 0.099708 .
no_previous_companies_worked -1.828e-02  5.294e-02  -0.345 0.729903
distance_from_home            2.114e-01  2.288e-02   9.240  < 2e-16 ***
total_dependents              8.422e-01  1.193e-01   7.061 1.65e-12 ***
no_leaves_taken               1.070e-01  1.995e-02   5.363 8.20e-08 ***
total_experience             -9.578e-03  6.612e-02  -0.145 0.884824
monthly_overtime_hrs          2.457e-01  4.254e-02   5.777 7.62e-09 ***
mgr_effectiveness            -1.005e+01  1.480e+00  -6.791 1.11e-11 ***
career_satisfaction           4.149e+00  1.466e+00   2.830 0.004660 **
perf_satisfaction             1.766e+00  1.270e+00   1.391 0.164268
work_satisfaction             2.066e+00  1.521e+00   1.358 0.174472
age_diff                      6.278e-02  3.739e-02   1.679 0.093146 .
tenure                       -3.437e-01  9.889e-02  -3.476 0.000509 ***
median_compensation          -2.044e-03  7.989e-04  -2.559 0.010502 *
compa_ratio                  -9.422e+01  4.025e+01  -2.341 0.019242 *
location_New_York             9.884e-01  4.560e-01   2.168 0.030195 *
location_Orlando             -8.768e-01  3.854e-01  -2.275 0.022911 *
level_Assistant_Manager      -5.082e-01  6.785e-01  -0.749 0.453860
level_Manager                -6.955e-01  6.594e-01  -1.055 0.291506
gender_Male                   4.146e-01  3.216e-01   1.289 0.197391
rating_Acceptable            -1.892e-02  3.778e-01  -0.050 0.960053
rating_Below_Average         -2.463e+00  6.836e-01  -3.604 0.000314 ***
rating_Excellent             -3.913e-01  8.946e-01  -0.437 0.661842
rating_Unacceptable          -4.518e+00  1.189e+00  -3.800 0.000144 ***
mgr_rating_Acceptable        -9.561e-02  3.582e-01  -0.267 0.789539
mgr_rating_Below_Average     -1.201e+00  6.615e-01  -1.816 0.069402 .
mgr_rating_Excellent         -7.318e-01  5.250e-01  -1.394 0.163327
mgr_rating_Unacceptable       1.163e+00  1.247e+00   0.933 0.350776
marital_status_Single         2.573e+00  5.588e-01   4.604 4.14e-06 ***
education_Masters             2.138e+00  6.059e-01   3.528 0.000419 ***
promotion_last_2_years_Yes    3.825e-01  4.607e-01   0.830 0.406375
compa_level_Below             3.031e-01  4.432e-01   0.684 0.494089
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1370.21  on 1367  degrees of freedom
Residual deviance:  349.93  on 1331  degrees of freedom
AIC: 423.93
```

```
Number of Fisher Scoring iterations: 8
```

0.78348570079804

In [54]:
```
# # Train logistic regression model with interactions
# logit.reg <- glm(turnover ~ . + .^2, data = smalltrain.df, family = "binomial")
# summary(logit.reg)

# # Generate predictions in the validation set
# pred.logit.reg <- predict(logit.reg, valid.df, type = 'response')

# # Calculate the Brier Skill Score based on the interaction logistic regression model
# skill.score(pred.logit.reg, valid.df$turnover, brier.ref)

# # The score is way worse than the model without interactions, so we do not use it and this is commented out
```

In [55]:
```
# Load the car package
library(car)

# Check for multicollinearity
vif(logit.reg)
```

mgr_reportees	1.35465413264967
mgr_tenure	1.27793113750991
compensation	11515.0787271118
percent_hike	3.108013596304
hiring_score	1.17719921905046
no_previous_compan...	1.12793644689198
distance_from_home	1.28667350813174
total_dependents	2.10550099246495
no_leaves_taken	1.15179968432141
total_experience	2.34432936757099
monthly_overtime_hrs	1.33758473375529
mgr_effectiveness	3.1140400050624 6
career_satisfaction	3.05682674887363
perf_satisfaction	2.77786802487066
work_satisfaction	1.98815235374587
age_diff	2.00108606691049
tenure	1.51113901586954
median_compensation	13.0709029607094
compa_ratio	11452.4954886754
location_New_York	1.79788936036609
location_Orlando	1.69356702186495
level_Assistant_Mana...	1.26477007179647
level_Manager	1.14706286137604
gender_Male	1.18532927311813
rating_Acceptable	1.76961583025622
rating_Below_Average	2.40320665352128
rating_Excellent	1.16959529392164
rating_Unacceptable	2.42727714101788
mgr_rating_Acceptable	1.63244367702598
mgr_rating_Below_A...	1.62430871834307
mgr_rating_Excellent	1.47060538712121
mgr_rating_Unaccept...	1.22171218154367
marital_status_Single	2.28895137653917
education_Masters	1.32451596856058
promotion_last_2_ye...	1.66228189686086
compa_level_Below	2.40004390239475

In [56]:
```
# Build a multiple logistic regression model
#remove level_specialist as it is perfectly multicollinear, found using the alias() function
#remove compensation as it has the highest vif
logit.reg <- glm(turnover ~ . - level_Specialist - compensation,  family = "binomial",
                 data = train_set)

# Print summary
summary(logit.reg)

# Generate predictions in the validation set
# Remember to set type = "response", which gives you the probability rather than the outcome variable
pred.logit.reg <- predict(logit.reg, valid.df, type = "response")

# Calculate the Brier Skill Score
skill.score(pred.logit.reg, valid.df$turnover, brier.ref)
```

```
Call:
glm(formula = turnover ~ . - level_Specialist - compensation,
    family = "binomial", data = train_set)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.52844  -0.17094  -0.05113  -0.00718   3.03726

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                   3.008e+00  1.431e+01   0.210 0.833514
mgr_reportees                 9.471e-02  2.986e-02   3.172 0.001515 **
mgr_tenure                   -1.706e-02  4.361e-02  -0.391 0.695724
percent_hike                 -5.930e-01  7.983e-02  -7.428 1.10e-13 ***
hiring_score                  6.496e-02  4.358e-02   1.490 0.136102
no_previous_companies_worked -1.522e-02  5.250e-02  -0.290 0.771875
distance_from_home            2.088e-01  2.268e-02   9.208  < 2e-16 ***
total_dependents              8.086e-01  1.139e-01   7.102 1.23e-12 ***
no_leaves_taken               1.050e-01  1.970e-02   5.328 9.95e-08 ***
total_experience             -4.216e-03  6.693e-02  -0.063 0.949779
monthly_overtime_hrs          2.474e-01  4.253e-02   5.818 5.94e-09 ***
mgr_effectiveness            -1.013e+01  1.462e+00  -6.925 4.37e-12 ***
career_satisfaction           3.933e+00  1.465e+00   2.685 0.007250 **
perf_satisfaction             2.002e+00  1.266e+00   1.582 0.113632
work_satisfaction             2.043e+00  1.528e+00   1.337 0.181130
age_diff                      5.971e-02  3.716e-02   1.607 0.108042
tenure                       -3.423e-01  9.970e-02  -3.433 0.000597 ***
median_compensation          -2.983e-04  2.493e-04  -1.197 0.231453
compa_ratio                  -1.912e-01  6.552e-01  -0.292 0.770474
location_New_York             9.781e-01  4.559e-01   2.146 0.031893 *
location_Orlando             -7.868e-01  3.795e-01  -2.073 0.038138 *
level_Assistant_Manager      -5.361e-01  6.562e-01  -0.817 0.413940
level_Manager                -6.287e-01  6.450e-01  -0.975 0.329639
gender_Male                   3.798e-01  3.182e-01   1.194 0.232654
rating_Acceptable            -1.978e-02  3.724e-01  -0.053 0.957656
rating_Below_Average         -2.536e+00  6.825e-01  -3.715 0.000203 ***
rating_Excellent             -3.385e-01  8.613e-01  -0.393 0.694335
rating_Unacceptable          -4.548e+00  1.204e+00  -3.777 0.000159 ***
mgr_rating_Acceptable        -2.960e-02  3.570e-01  -0.083 0.933923
mgr_rating_Below_Average     -1.084e+00  6.583e-01  -1.646 0.099721 .
mgr_rating_Excellent         -6.119e-01  5.087e-01  -1.203 0.228990
mgr_rating_Unacceptable       1.270e+00  1.251e+00   1.016 0.309797
marital_status_Single         2.453e+00  5.476e-01   4.480 7.45e-06 ***
education_Masters             2.121e+00  5.879e-01   3.607 0.000310 ***
promotion_last_2_years_Yes    4.957e-01  4.541e-01   1.092 0.275007
compa_level_Below             2.049e-01  4.415e-01   0.464 0.642586
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1370.21  on 1367  degrees of freedom
Residual deviance:  355.06  on 1332  degrees of freedom
AIC: 427.06

Number of Fisher Scoring iterations: 8
```

0.7846324814790355

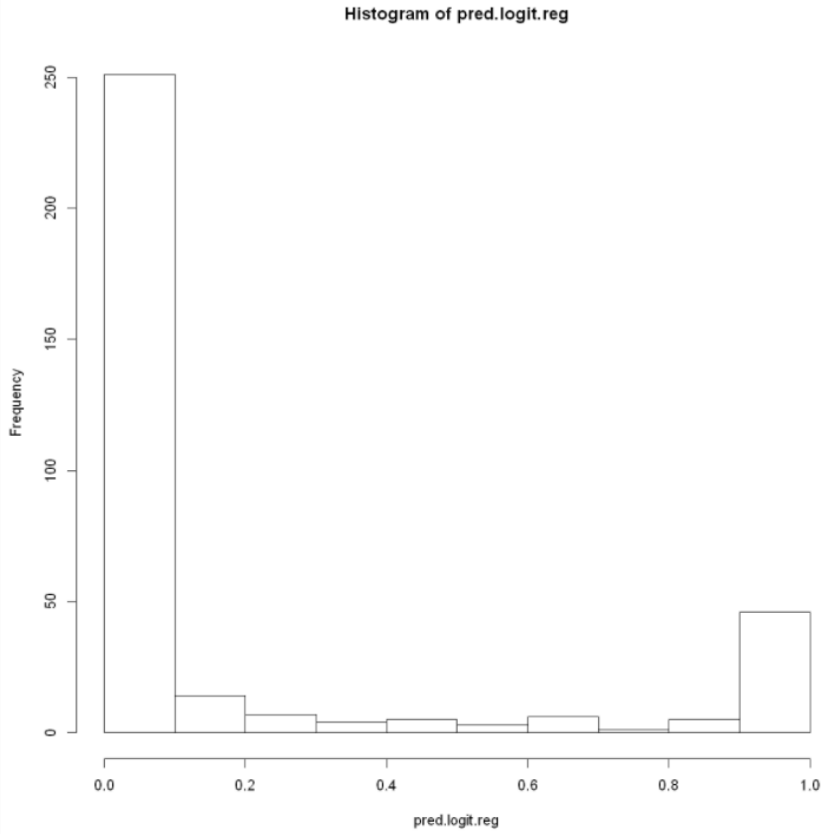
In [57]:
```
vif(logit.reg)
#Large difference between null and residual deviance shows that model is good
```

mgr_reportees	1.32819248689043
mgr_tenure	1.28045573144961
percent_hike	3.04337250260656
hiring_score	1.17851523146325
no_previous_compan...	1.12527611395984
distance_from_home	1.25840979115379
total_dependents	1.9845771569687
no_leaves_taken	1.14841192406386
total_experience	2.34410944451954
monthly_overtime_hrs	1.34980374098852
mgr_effectiveness	3.05686462555275
career_satisfaction	3.04997993515422
perf_satisfaction	2.77762799339796
work_satisfaction	1.98968035045944
age_diff	1.96215125905416
tenure	1.4599594022388
median_compensation	1.22950069935458
compa_ratio	2.98508247405225
location_New_York	1.82446203998291
location_Orlando	1.67354059483787
level_Assistant_Mana...	1.26978137230864
level_Manager	1.14476390524031
gender_Male	1.17943957656571
rating_Acceptable	1.74242033326323
rating_Below_Average	2.40943980964231
rating_Excellent	1.18628853421636
rating_Unacceptable	2.3590785957391
mgr_rating_Acceptable	1.6429507340811
mgr_rating_Below_A...	1.6178967051491
mgr_rating_Excellent	1.46291586595034
mgr_rating_Unaccept...	1.21401476905171
marital_status_Single	2.17283400213824
education_Masters	1.33758955157086
promotion_last_2_ye...	1.64050213588387
compa_level_Below	2.41352153685

In [58]:
```
# Look at the prediction range
hist(pred.logit.reg)
prediction_categories <- ifelse(pred.logit.reg > 0.5, 1, 0)

# Construct a confusion matrix
conf_matrix <- table(prediction_categories, valid.df$turnover)
conf_matrix
```

```
prediction_categories   0   1
                    0 266  15
                    1   4  57
```

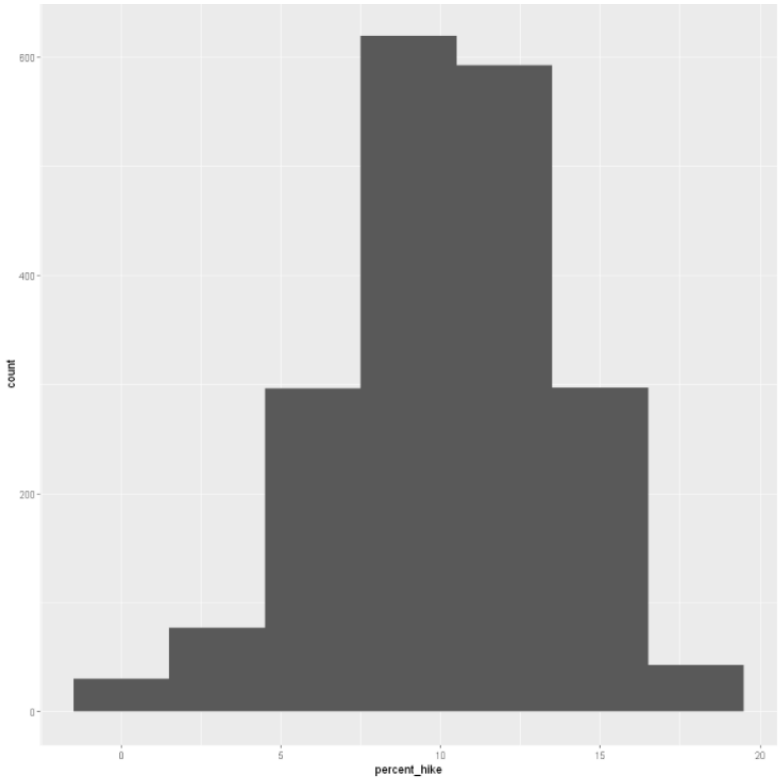
```
prediction_categories   0   1
                    0 266  15
                    1   4  57
```



In [59]:
```
# Load caret
library(caret)

# Call confusionMatrix
confusionMatrix(conf_matrix)
```

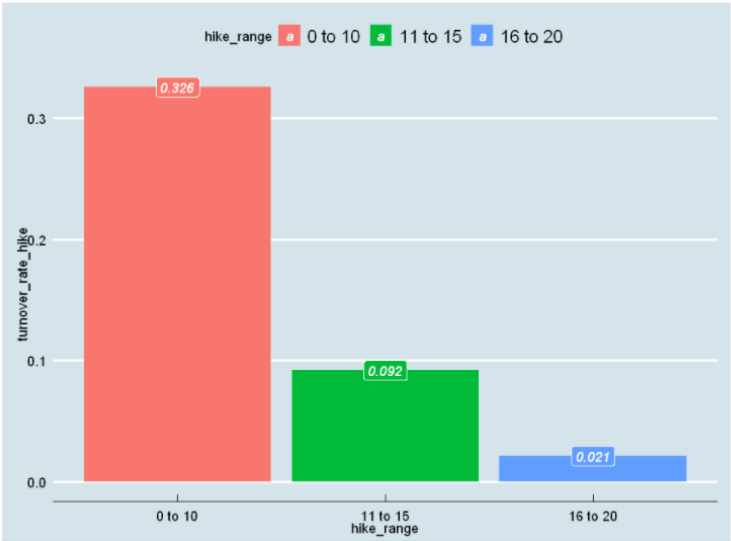
```
Confusion Matrix and Statistics


prediction_categories   0   1
                    0 266  15
                    1   4  57

               Accuracy : 0.9444
                 95% CI : (0.9146, 0.9662)
    No Information Rate : 0.7895
    P-Value [Acc > NIR] : 8.494e-16

                  Kappa : 0.823

 Mcnemar's Test P-Value : 0.02178

            Sensitivity : 0.9852
            Specificity : 0.7917
         Pos Pred Value : 0.9466
         Neg Pred Value : 0.9344
             Prevalence : 0.7895
         Detection Rate : 0.7778
   Detection Prevalence : 0.8216
      Balanced Accuracy : 0.8884

       'Positive' Class : 0
```

In [60]:
```
# Plot histogram of percent hike
ggplot(emp_final, aes(x = percent_hike)) +
  geom_histogram(binwidth = 3)

# Create salary hike_range of Analyst level employees
emp_hike_range <- data %>%
  filter(level == "Analyst") %>%
  mutate(hike_range = cut(percent_hike, breaks = c(0, 10, 15, 20),
                          include.lowest = TRUE,
                          labels = c("0 to 10",
                                     "11 to 15", "16 to 20")))
```



In [79]:
```
# Calculate the turnover rate for each salary hike range
df_hike <- emp_hike_range %>%
  group_by(hike_range) %>%
  summarize(turnover_rate_hike = mean(turnover))

# Check the results
df_hike

# Visualize the results
ggplot(df_hike, aes(x = hike_range, y = turnover_rate_hike, fill = hike_range)) +
  geom_col() +
geom_label(aes(label=sprintf("%0.3f", round(turnover_rate_hike, digits = 3)), fill = hike_range), colour = "white", fontface = "
theme_economist()
```

hike_range	turnover_rate_hike
0 to 10	0.32591823
11 to 15	0.09171598
16 to 20	0.02127660

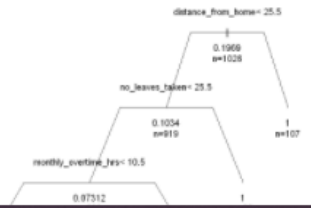


In [63]:

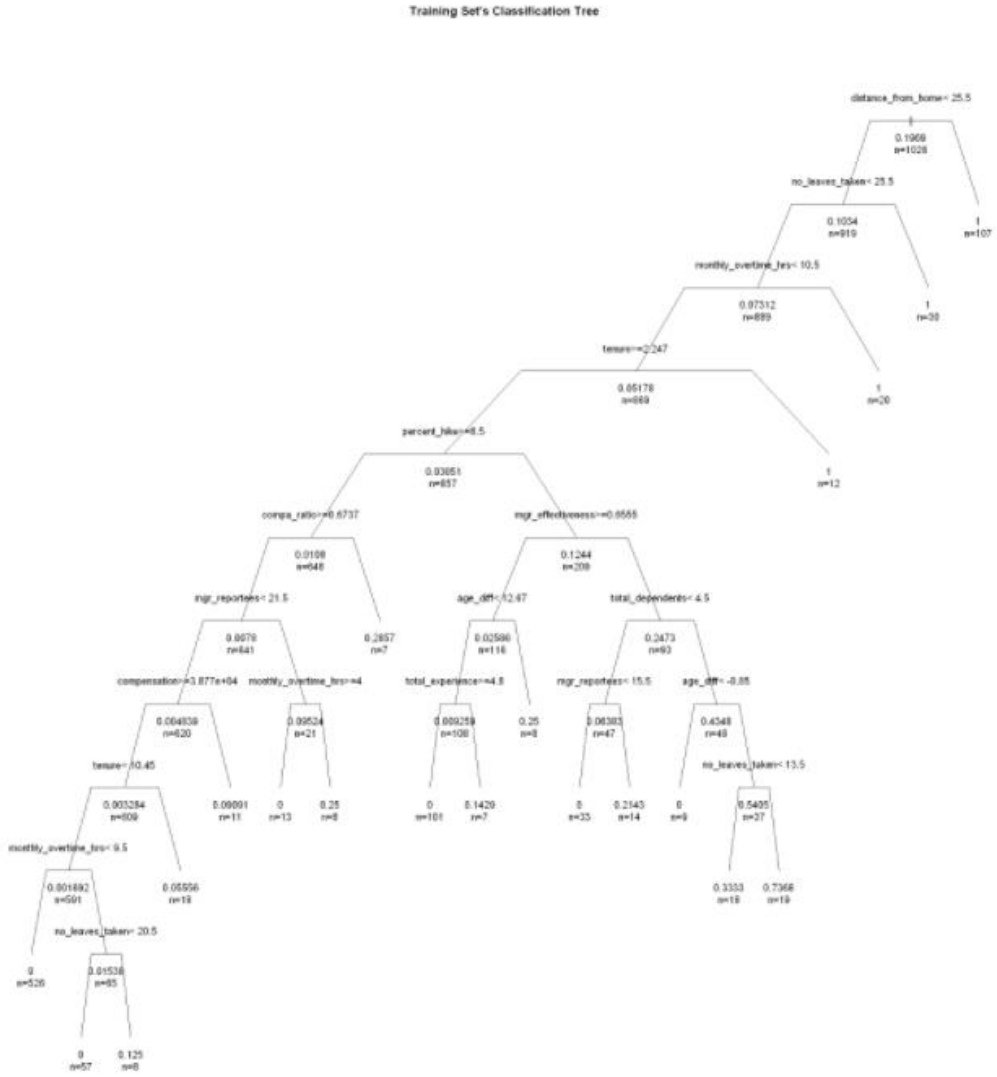
```
### Model2: Regression Tree
# Train regression tree model
library(rpart)
reg.tree <- rpart(turnover ~ ., data = smalltrain.df,
                  control = rpart.control(cp = 0.0002))

#plot the tree
options(repr.plot.width=20, repr.plot.height=20)
plot(reg.tree, uniform=TRUE, branch=0.6, margin=0.05)
text(reg.tree, all=TRUE, use.n=TRUE)
title("Training Set's Classification Tree")

#plot fancy tree
rparty.tree <- as.party(reg.tree)
rparty.tree
options(repr.plot.width=20, repr.plot.height=20)
fancyRpartPlot(reg.tree)

# Print out feature importance
reg.tree$variable.importance

#plot variable importance
library(RColorBrewer)
var_imp <- data.frame(reg.tree$variable.importance)
var_imp$features <- rownames(var_imp)
var_imp <- var_imp[, c(2, 1)]
var_imp$importance <- round(var_imp$reg.tree.variable.importance, 2)
var_imp$reg.tree.variable.importance <- NULL

colorCount <- length(unique(var_imp$features))
feature_importance <- var_imp %>%
ggplot(aes(x=reorder(features, importance), y=importance, fill=features)) + geom_bar(stat='identity') + coord_flip() +
  theme_minimal() + theme(legend.position="none", strip.background = element_blank(), strip.text.x = element_blank(),
     plot.title=element_text(hjust=0.5, color="white"), plot.subtitle=element_text(color="white"), plot.background=element_rect(
                                                        axis.text.x=element_text(colour="white"), axis.text.y=element_text(colou
                                                        axis.title=element_text(colour="white"),
     legend.background = element_rect(fill="#FFF9F5",
                                      size=0.5, linetype="solid",
                                      colour ="black")) + scale_fill_manual(values = colorRampPalette(brewer.pal(24, "Set2"))(colorC
geom_label(aes(label=paste0(importance, "%")), colour = "white", fontface = "italic", hjust=0.6) +
labs(title="Feature Importance for our Decision Tree Model", x="Features", y="Importance")
feature_importance


# Generate predictions in the validation set
reg.tree.pred <- predict(reg.tree,valid.df)

# Calculate the Brier Skill Score
skill.score(reg.tree.pred, valid.df$turnover, brier.ref)

#plot confusion matrix
library(RColorBrewer)
options(repr.plot.width=8, repr.plot.height=6)
prediction_categories <- ifelse(reg.tree.pred > 0.5, 1, 0)

conf_df <- data.frame(table(valid.df$turnover, prediction_categories))

ggplot(data =  conf_df, mapping = aes(x = prediction_categories, y = Var1)) +
  geom_tile(aes(fill = Freq), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "#F3F781", high = "#58FA82") +
  theme_economist() + theme(legend.position="none", strip.background = element_blank(), strip.text.x = element_blank(),
     plot.title=element_text(hjust=0.5, color="white"), plot.subtitle=element_text(color="white"), plot.background=element_rect(
                                                        axis.text.x=element_text(colour="white"), axis.text.y=element_text(colou
                                                        axis.title=element_text(colour="white"),
     legend.background = element_rect(fill="#FFF9F5",
                                      size=0.5, linetype="solid",
                                      colour ="black")) +
labs(title="Confusion Matrix", y="Attrition Status", x="Predictions")
```

```
Model formula:
turnover ~ mgr_reportees + mgr_tenure + compensation + percent_hike +
    hiring_score + no_previous_companies_worked + distance_from_home +
    total_dependents + no_leaves_taken + total_experience + monthly_overtime_hrs +
    mgr_effectiveness + career_satisfaction + perf_satisfaction +
    work_satisfaction + age_diff + tenure + median_compensation +
    compa_ratio + location_New_York + location_Orlando + level_Assistant_Manager +
    level_Manager + level_Specialist + gender_Male + rating_Acceptable +
    rating_Below_Average + rating_Excellent + rating_Unacceptable +
    mgr_rating_Acceptable + mgr_rating_Below_Average + mgr_rating_Excellent +
    mgr_rating_Unacceptable + marital_status_Single + education_Masters +
    promotion_last_2_years_Yes + compa_level_Below

Fitted party:
[1] root
|   [2] distance_from_home < 25.5
|   |   [3] no_leaves_taken < 25.5
|   |   |   [4] monthly_overtime_hrs < 10.5
|   |   |   |   [5] tenure >= 2.24658
|   |   |   |   |   [6] percent_hike >= 8.5
|   |   |   |   |   |   [7] compa_ratio >= 0.6737
|   |   |   |   |   |   |   [8] mgr_reportees < 21.5
|   |   |   |   |   |   |   |   [9] compensation >= 38766
|   |   |   |   |   |   |   |   |   [10] tenure < 10.45479
|   |   |   |   |   |   |   |   |   |   [11] monthly_overtime_hrs < 9.5: 0.000 (n = 526, err = 0.0)
|   |   |   |   |   |   |   |   |   |   [12] monthly_overtime_hrs >= 9.5
|   |   |   |   |   |   |   |   |   |   |   [13] no_leaves_taken < 20.5: 0.000 (n = 57, err = 0.0)
|   |   |   |   |   |   |   |   |   |   |   [14] no_leaves_taken >= 20.5: 0.125 (n = 8, err = 0.9)
|   |   |   |   |   |   |   |   |   [15] tenure >= 10.45479: 0.056 (n = 18, err = 0.9)
|   |   |   |   |   |   |   |   [16] compensation < 38766: 0.091 (n = 11, err = 0.9)
|   |   |   |   |   |   |   [17] mgr_reportees >= 21.5
|   |   |   |   |   |   |   |   [18] monthly_overtime_hrs >= 4: 0.000 (n = 13, err = 0.0)
|   |   |   |   |   |   |   |   [19] monthly_overtime_hrs < 4: 0.250 (n = 8, err = 1.5)
|   |   |   |   |   |   [20] compa_ratio < 0.6737: 0.286 (n = 7, err = 1.4)
|   |   |   |   |   [21] percent_hike < 8.5
|   |   |   |   |   |   [22] mgr_effectiveness >= 0.6555
|   |   |   |   |   |   |   [23] age_diff < 12.67
|   |   |   |   |   |   |   |   [24] total_experience >= 4.8: 0.000 (n = 101, err = 0.0)
|   |   |   |   |   |   |   |   [25] total_experience < 4.8: 0.143 (n = 7, err = 0.9)
|   |   |   |   |   |   |   [26] age_diff >= 12.67: 0.250 (n = 8, err = 1.5)
|   |   |   |   |   |   [27] mgr_effectiveness < 0.6555
|   |   |   |   |   |   |   [28] total_dependents < 4.5
|   |   |   |   |   |   |   |   [29] mgr_reportees < 15.5: 0.000 (n = 33, err = 0.0)
|   |   |   |   |   |   |   |   [30] mgr_reportees >= 15.5: 0.214 (n = 14, err = 2.4)
|   |   |   |   |   |   |   [31] total_dependents >= 4.5
|   |   |   |   |   |   |   |   [32] age_diff < -0.85: 0.000 (n = 9, err = 0.0)
|   |   |   |   |   |   |   |   [33] age_diff >= -0.85
|   |   |   |   |   |   |   |   |   [34] no_leaves_taken < 13.5: 0.333 (n = 18, err = 4.0)
|   |   |   |   |   |   |   |   |   [35] no_leaves_taken >= 13.5: 0.737 (n = 19, err = 3.7)
|   |   |   |   |   [36] tenure < 2.24658: 1.000 (n = 12, err = 0.0)
|   |   |   |   [37] monthly_overtime_hrs >= 10.5: 1.000 (n = 20, err = 0.0)
|   |   |   [38] no_leaves_taken >= 25.5: 1.000 (n = 30, err = 0.0)
|   |   [39] distance_from_home >= 25.5: 1.000 (n = 107, err = 0.0)

Number of inner nodes:    19
Number of terminal nodes: 20
```

Training Set's Classification Tree

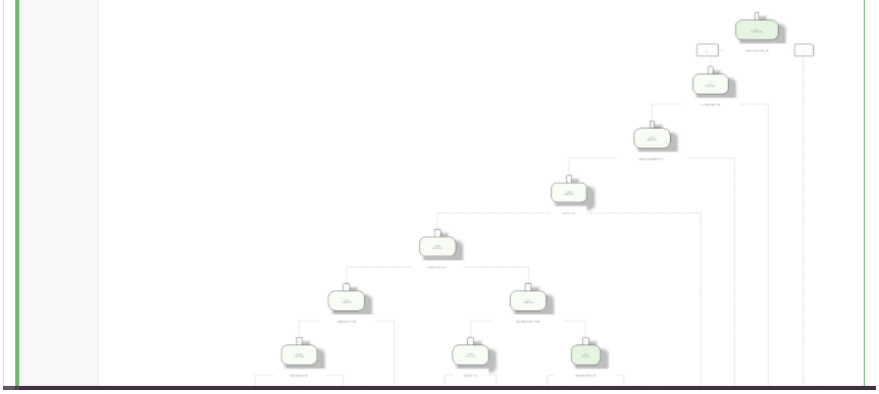


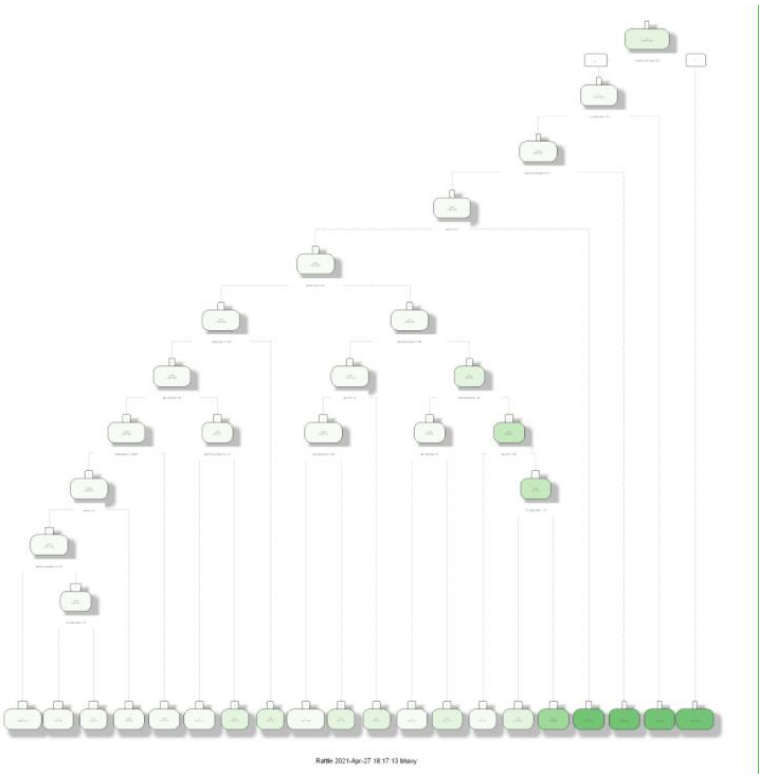
```
|   |   |   [37] monthly_overtime_hrs >= 10.5: 1.000 (n = 20, err = 0.0)
|   |   [38] no_leaves_taken >= 25.5: 1.000 (n = 30, err = 0.0)
|   [39] distance_from_home >= 25.5: 1.000 (n = 107, err = 0.0)

Number of inner nodes:    19
Number of terminal nodes: 20
```



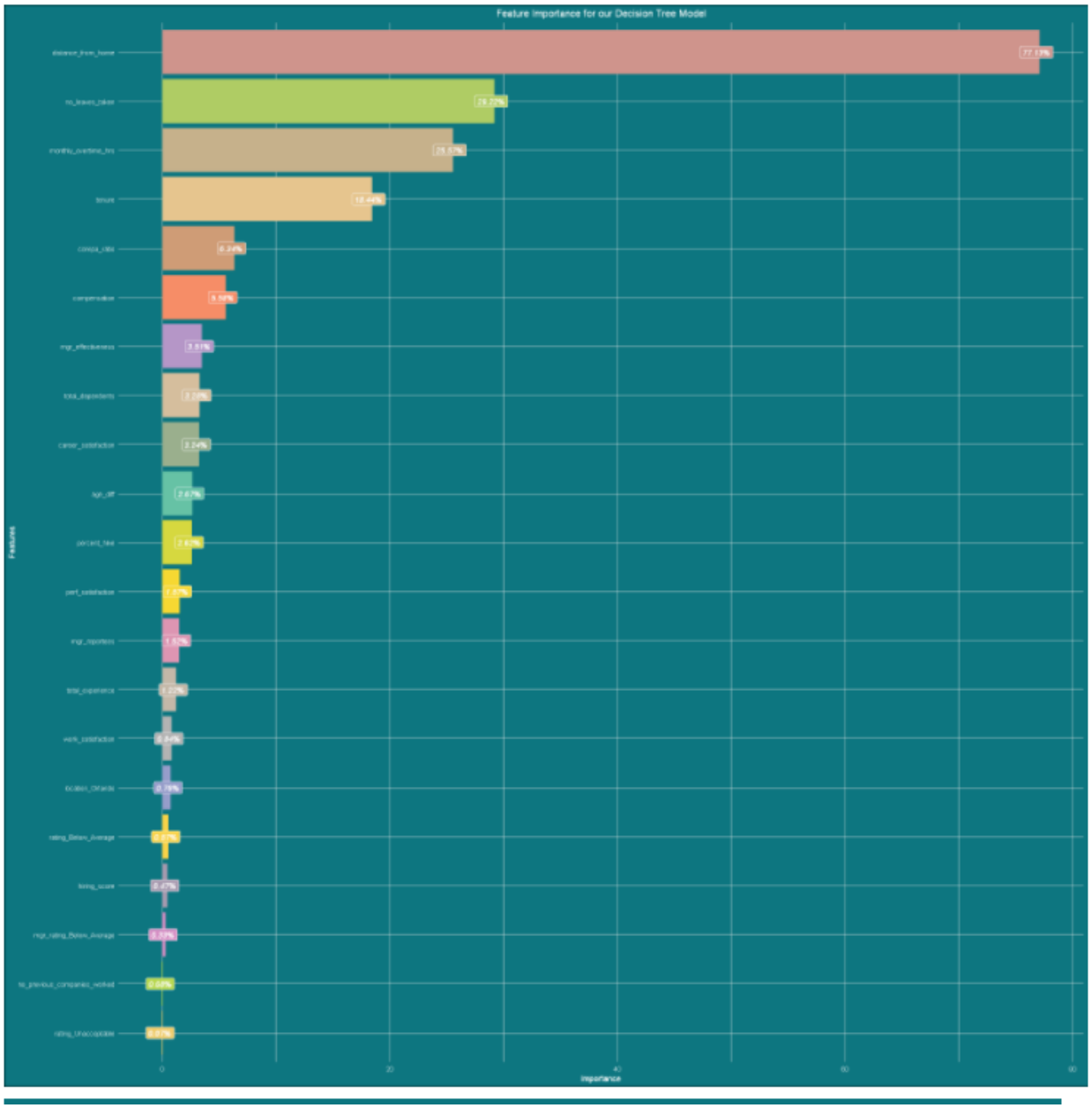
distance_from_home	77.1278574640587
no_leaves_taken	29.2175408312188
monthly_overtime_hrs	25.5719644818194
tenure	18.4360386231774
compa_ratio	6.34115496108647
compensation	5.57703865135097
mgr_effectiveness	3.507091489777
total_dependents	3.27635044498537
career_satisfaction	3.23780308679465
age_diff	2.66869976892207
percent_hike	2.62462473942492
perf_satisfaction	1.57322038737121
mgr_reportees	1.52141513633709
total_experience	1.22116150379738
work_satisfaction	0.841662050258599
location_Orlando	0.764970965622796
rating_Below_Average	0.565945478690329
hiring_score	0.470035252643948
mgr_rating_Below_A...	0.326620450211839
no_previous_compan...	0.0773809523809524
rating_Unacceptable	0.0683037646695223

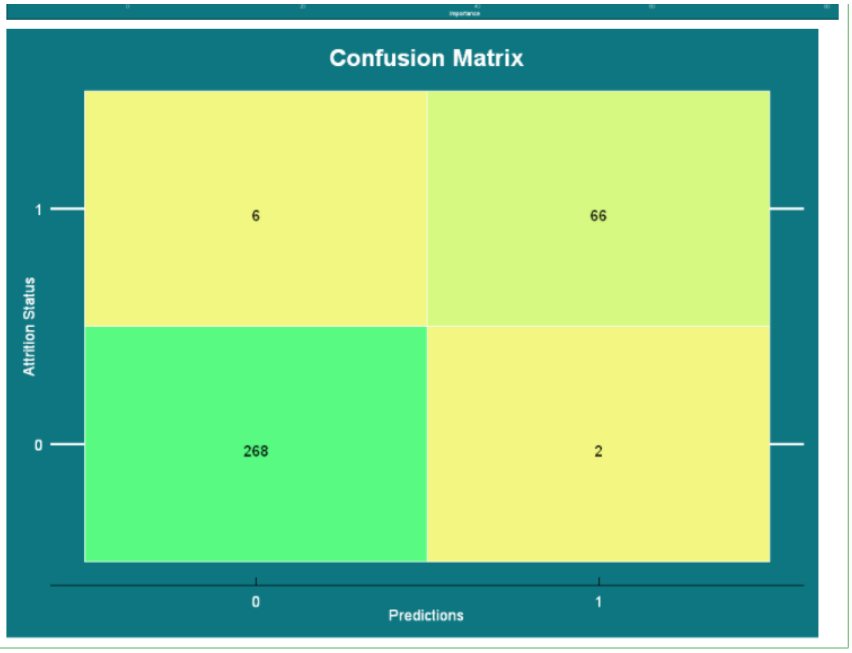
```
Warning message in brewer.pal(24, "Set2"):
"n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
"
```





0.862132252158232



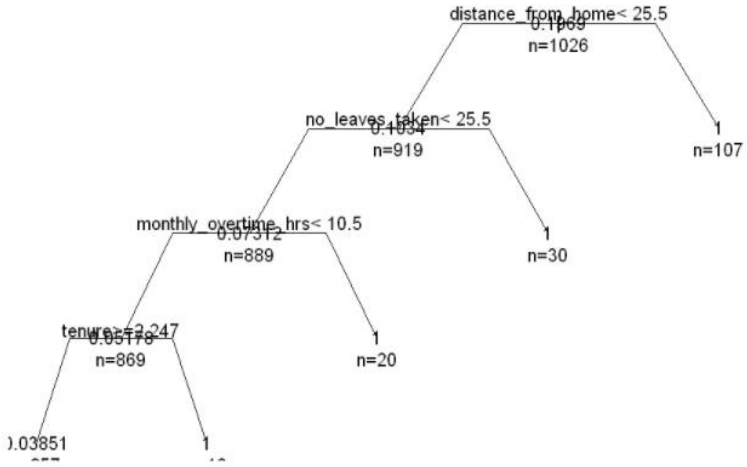


In [64]:
```
# Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances

prune.reg.tree <- prune(reg.tree, cp=0.02) # pruning the tree
plot(prune.reg.tree, uniform=TRUE, branch=0.6)
text(prune.reg.tree, all=TRUE, use.n=TRUE)
prune.reg.tree.pred <- predict(prune.reg.tree,valid.df)
skill.score(prune.reg.tree.pred, valid.df$turnover, brier.ref)
```

0.862621459722464

In [64]:
```
# Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances

prune.reg.tree <- prune(reg.tree, cp=0.02) # pruning the tree
plot(prune.reg.tree, uniform=TRUE, branch=0.6)
text(prune.reg.tree, all=TRUE, use.n=TRUE)
prune.reg.tree.pred <- predict(prune.reg.tree,valid.df)
skill.score(prune.reg.tree.pred, valid.df$turnover, brier.ref)
```

0.862621459722464



In [65]:
```
str(prediction_categories)
str(valid.df$turnover)
```

```
 Named num [1:342] 0 0 1 0 1 1 1 0 0 1 ...
 - attr(*, "names")= chr [1:342] "1" "2" "3" "4" ...
 num [1:342] 0 0 1 0 1 1 1 0 0 1 ...
```

In [66]:
```
### Model3: XGBoost
# XGBoost Model can only deal with numeric numbers/matrix, so we convert all variables to matrix first
xsmalltrain.xgb <- model.matrix(~ 0 + ., data = smalltrain.df[, -12]) #IMPORTANT --> REMOVE TARGET VARIABLE, stores all predicto
ysmalltrain.xgb <- as.vector(smalltrain.df$turnover) #target variables

xvalid.xgb <- model.matrix(~ 0 + ., data = valid.df[, -12])
yvalid.xgb <- as.vector(valid.df$turnover)

#change categorical variables to binary values
#DO NOT INCLUDE TARGET VARIABLE IN PREDICTORS!!!

# Note: Take a while to train
# Train the XGBoost model
t1 <- proc.time()
xgb.trees <- xgboost(xsmalltrain.xgb,
                     ysmalltrain.xgb,
                     max.depth = 3,
                     nthread = workers,
                     nround = 200,
                     objective = "binary:logistic",
                     verbose = 0)
proc.time() - t1
# Tune the model above by increasing/decreasing the depth of each tree
# and/or the number of trees (nround).

# Print out feature importance
# Gain indicates the contribution of each feature. Higher gain means higher importance.
xgb.importance(colnames(xsmalltrain.xgb), model = xgb.trees)

# Generate predictions in the validation set
xgb.trees.pred <- predict(xgb.trees, xvalid.xgb)

# Calculate the Brier Skill Score
skill.score(xgb.trees.pred, yvalid.xgb, brier.ref)
```

```
[18:17:18] WARNING: amalgamation/../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
   user  system elapsed
   0.83    0.42    0.50
```

Feature	Gain	Cover	Frequency
distance_from_home	0.4149547449	0.170869534	0.077519380
no_leaves_taken	0.1402546383	0.124040481	0.062015504
monthly_overtime_hrs	0.0981164591	0.093898996	0.041343669
tenure	0.0787274736	0.105896838	0.090439276
percent_hike	0.0563820600	0.083401995	0.056847545
total_dependents	0.0502352916	0.074938713	0.060723514
mgr_effectiveness	0.0445843437	0.064264440	0.077519380
mgr_reportees	0.0185189962	0.043992881	0.054263566
compensation	0.0169101864	0.028145563	0.042635659
age_diff	0.0151529889	0.025482847	0.058139535
work_satisfaction	0.0105355451	0.033281073	0.055555556
rating_Below_Average	0.0080026844	0.007932401	0.012919897

```
   user  system elapsed
   0.83    0.42    0.50
```

Feature	Gain	Cover	Frequency
distance_from_home	0.4149547449	0.170869534	0.077519380
no_leaves_taken	0.1402546383	0.124040481	0.062015504
monthly_overtime_hrs	0.0961164591	0.093698996	0.041343669
tenure	0.0787274736	0.105896838	0.090439276
percent_hike	0.0563820600	0.083401995	0.056847545
total_dependents	0.0502352916	0.074938713	0.060723514
mgr_effectiveness	0.0445843437	0.064264440	0.077519380
mgr_reportees	0.0185189962	0.043992881	0.054263566
compensation	0.0169101864	0.028145563	0.042635659
age_diff	0.0151529889	0.025482847	0.058139535
work_satisfaction	0.0105355451	0.033281073	0.055555556
rating_Below_Average	0.0080026844	0.007932401	0.012919897
location_Orlando	0.0063650859	0.011541697	0.015503876
career_satisfaction	0.0056920639	0.010161843	0.032299742
hiring_score	0.0050438104	0.013527786	0.028423773
compa_ratio	0.0045152889	0.010451574	0.020671835
total_experience	0.0044639534	0.015391581	0.040051680
no_previous_companies_worked	0.0039621243	0.011631638	0.027131783
marital_status_Single	0.0039592454	0.015114391	0.021963824
mgr_tenure	0.0039203739	0.016919750	0.054263566
location_New_York	0.0030702767	0.004239540	0.007751938
perf_satisfaction	0.0030077159	0.018457387	0.033591731
education_Masters	0.0029041462	0.007389926	0.005167959
gender_Male	0.0019317887	0.003559097	0.007751938
mgr_rating_Acceptable	0.0005313089	0.003477023	0.005167959
rating_Acceptable	0.0002574053	0.004191005	0.010335917

0.847051822834058

In [68]:
```r
### Model4: Ensemble Model --- Weighted Average
m1_weight <- seq(0.1,0.9,0.1)
m2_weight <- seq(0.1,0.9,0.1)

# set up a matrix to store the Brier skill scores
bss_matrix <- matrix(0,9,9)
for (i in 1:9) {
    for (j in 1:9){
        if (m2_weight[j]+ m1_weight[i] > 1) next
        ensemble_pred <- m1_weight[i]*pred.logit.reg + reg.tree.pred*m2_weight[j] + xgb.trees.pred*(1-m1_weight[i] - m2_weight[i
        bss_matrix[i,j] <- skill.score(ensemble_pred, valid.df$turnover, brier.ref)
    }
}

# print out the Brier skills score matrix
bss_matrix

# Find the element in the matrix corresponds to the highest score?
which(bss_matrix == max(bss_matrix), arr.ind = TRUE)

# According to the table, the weights put on the predictions from the logistic regression,
# regression tree and XGBoost should be 0.2,0.2,and 0.6.
# The highest Brier Skill Score acheived by the weighted average ensemble is 0.0522.
```

0.86228253	0.8688106	0.8515548	0.8105151	0.7456917	0.6570843	0.5446932	0.4085181	0.2485593
0.84324385	0.8730086	0.8789896	0.8611866	0.8195999	0.7542293	0.6650748	0.5521365	0.0000000
0.79701096	0.8500125	0.8792301	0.8846639	0.8663139	0.8241800	0.7582623	0.0000000	0.0000000
0.72358385	0.7998221	0.8522764	0.8809470	0.8858337	0.8669365	0.0000000	0.0000000	0.0000000
0.62296252	0.7224375	0.7981286	0.8500358	0.8781592	0.0000000	0.0000000	0.0000000	0.0000000
0.49514698	0.6178586	0.7167865	0.7919305	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.34013722	0.4860856	0.6082502	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.15793325	0.3271184	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
-0.05146494	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

row	col
4	5

In [69]:

```
### Model5-6: Ensemble Model --- Stacking
# Logistic regression
M1 <- glm(turnover ~  . + .^2, data = smalltrain.df, family = "binomial")
M1.predict <- predict(M1, valid.df, type = "response")

# Regression tree
M2 <- rpart(turnover ~ .,
            data = smalltrain.df,
            control = rpart.control(cp = 0.0001))
M2.predict <-  predict(M2, valid.df)

# XGBoost
xtrain.xgb <- model.matrix(~ 0 + ., data = smalltrain.df[,-12])
ytrain.xgb <- as.vector(smalltrain.df$turnover)

xvalid.xgb <- model.matrix(~ 0 + ., data = valid.df[,-12])
yvalid.xgb <- as.vector(valid.df$turnover)

M3 <- xgboost(xtrain.xgb, ytrain.xgb,
              max.depth = 3,
              nthread = workers,
              nround = 200,
              objective = "binary:logistic",
              verbose = 0)


M3.predict <- predict(M3, xvalid.xgb)

# Construct the stacker dataframe
stacker.df <- data.frame(turnover = valid.df$turnover,
                         M1.predict = M1.predict,
                         M2.predict = M2.predict,
                         M3.predict = M3.predict)

head(stacker.df)
```

```
Warning message:
"glm.fit: algorithm did not converge"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message in predict.lm(object, newdata, se.fit, scale = 1, ty
pe = if (type == :
"prediction from a rank-deficient fit may be misleading"
```

```
[18:17:26] WARNING: amalgamation/../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the
objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old
behavior.
```

turnover	M1.predict	M2.predict	M3.predict
0	1.780418e-06	0	9.807499e-05
0	2.220446e-16	0	1.811919e-03
1	2.220446e-16	1	7.142323e-01
0	2.220446e-16	0	3.389503e-04
1	9.998899e-01	1	9.815139e-01
1	1.000000e+00	1	9.997780e-01

In [70]:
```
### Model 5: Stacking model - Regression Tree
# Train Stacker model 1: regression tree
# Dependent variable is the DEFAULT_FLAG, independent variables are M1, M2, M3 predictions
stackerModel_1 <- rpart(turnover ~ .,
                        data = stacker.df,
                        control = rpart.control(cp = 0.0004))

# Predict from stacker model 1 --- regression tree
predict.variables <- stacker.df[, -1]

stacker.predict.rt <- predict(stackerModel_1, predict.variables)

# Score the stacker model's prediction
skill.score(stacker.predict.rt, valid.df$turnover, brier.ref)
```

```
0.900036561256962
```

In [71]:
```
### Model 6: Stacking model - XGBoost
# Convert stacker.df to matrix format
stacker.x.xgb <- model.matrix(~ 0 + ., data = stacker.df[,-1])
stacker.y.xgb <- as.vector(stacker.df[,1])

stackerModel_2 <- xgboost(stacker.x.xgb,
                          stacker.y.xgb,
                          max.depth = 3,
                          nthread = workers,
                          nround = 20,
                          objective = "binary:logistic",
                          verbose = 0)
# Different settings of max.depth and nround have been examined.
# max.depth = 3 and nround = 20 provide the highest Brier Skill Score in the test set.

# Predict from stacker model 2 --- XGBoost, and calculate the Brier Skill Score
predict.variables.xgb <- model.matrix(~ 0 + ., data = predict.variables)

stacker.predict.xgb <- predict(stackerModel_2, predict.variables.xgb)

# Score the stacker model's prediction
skill.score(stacker.predict.xgb, valid.df$turnover, brier.ref)
```

```
[18:17:27] WARNING: amalgamation/../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

0.954805578842172
```

In [72]:
```
glimpse(train_set)
```

```
Rows: 1,368
Columns: 38
$ mgr_reportees                <dbl> 9, 4, 6, 10, 11, 19, 21, 9, 12, 17, 13...
$ mgr_tenure                   <dbl> 3.17, 7.92, 4.38, 2.87, 12.95, 10.88, ...
$ compensation                 <dbl> 64320, 48204, 85812, 49536, 75576, 569...
$ percent_hike                 <dbl> 10, 8, 11, 8, 12, 8, 12, 9, 9, 11, 7, ...
$ hiring_score                 <dbl> 70, 70, 77, 71, 70, 75, 72, 70, 70, 70...
$ no_previous_companies_worked <dbl> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 6, 6, 0,...
$ distance_from_home           <dbl> 14, 21, 15, 9, 25, 23, 17, 16, 22, 18,...
$ total_dependents             <dbl> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, 5, 2,...
$ no_leaves_taken              <dbl> 2, 10, 18, 19, 25, 15, 10, 20, 22, 24,...
$ total_experience             <dbl> 6.86, 4.88, 8.55, 4.76, 8.06, 13.72, 5...
$ monthly_overtime_hrs         <dbl> 1, 5, 3, 8, 1, 7, 2, 10, 2, 8, 3, 3, 4...
$ turnover                     <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,...
$ mgr_effectiveness            <dbl> 0.730, 0.581, 0.770, 0.240, 0.710, 0.5...
$ career_satisfaction          <dbl> 0.73, 0.72, 0.85, 0.42, 0.78, 0.88, 0....
$ perf_satisfaction            <dbl> 0.73, 0.84, 0.80, 0.33, 0.67, 0.81, 0....
$ work_satisfaction            <dbl> 0.75, 0.85, 0.87, 0.85, 0.80, 0.86, 0....
$ age_diff                     <dbl> 18.98, 10.01, 2.38, 2.15, 3.05, 2.84, ...
$ tenure                       <dbl> 3.821918, 5.271233, 9.161644, 3.616438...
$ median_compensation          <dbl> 54684, 54684, 54684, 54684, 54684, 542...
$ compa_ratio                  <dbl> 1.1762124, 0.8815010, 1.5692341, 0.905...
$ location_New_York            <int> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...
$ location_Orlando             <int> 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0,...
```

In [73]:
```
### Final Step: Retrain this best model using the entire training set and generate predictions for the testing set
# Select OLTV, DTI, CSCORE_B as predictors. turnover is the response variable.

xtrain.xgb <- model.matrix(~ 0 + ., data = train_set[,-12]) #remove variable 12, turnover
ytrain.xgb <- as.vector(train_set$turnover)

xtest.xgb <- model.matrix(~ 0 + ., data = test_set[,-12])
ytest.xgb <- as.vector(test_set$turnover)
```

In [74]:
```
# Train base models
M1.trainAll <- glm(turnover ~ . + .^2,
                   data = train_set, family = "binomial")

M2.trainAll <- rpart(turnover ~ .,
                     data = train_set,
                     control = rpart.control(cp = 0.0001))

M3.trainAll <- xgboost(xtrain.xgb, ytrain.xgb,
                       max.depth = 3,
                       nthread = workers,
                       nround = 200,
                       objective = "binary:logistic",
                       verbose = 0)
# Generate predictions in the testing dataset using each of the base model
M1.predict.test <- predict(M1.trainAll, test_set, type = "response")
M2.predict.test <- predict(M2.trainAll, test_set)
M3.predict.test <- predict(M3.trainAll, xtest.xgb)
```

```
Warning message:
"glm.fit: algorithm did not converge"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

```
[18:17:34] WARNING: amalgamation/../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
"prediction from a rank-deficient fit may be misleading"
```

In [75]:
```
# Construct the stacker dataframe
stacker.df <- data.frame(turnover = test_set$turnover,
                         M1.predict.test = M1.predict.test,
                         M2.predict.test = M2.predict.test,
                         M3.predict.test = M3.predict.test)

head(stacker.df)

# Train our best stacker model (i.e., stacker model 2 --- XGBoost)
stacker.x.xgb <- model.matrix(~ 0 + ., data = stacker.df[,-1])
stacker.y.xgb <- as.vector(stacker.df[,1])

stackerModel_2 <- xgboost(stacker.x.xgb,
                          stacker.y.xgb,
                          max.depth = 3,
                          nthread = workers,
                          nround = 20,
                          objective = "binary:logistic",
                          verbose = 0)

# Predict from our best model (i.e., stacker model 2 --- XGBoost)
predict.variables <- stacker.df[, -1]
predict.variables.xgb <- model.matrix(~ 0 + ., data = predict.variables)

stacker.predict.xgb <- predict(stackerModel_2, predict.variables.xgb)

# Score the stacker model's prediction
skill.score(stacker.predict.xgb, test_set$turnover, brier.ref)
```

turnover	M1.predict.test	M2.predict.test	M3.predict.test
1	1.000000e+00	0.63157895	0.9969930649
1	9.999959e-01	1.00000000	0.9997059703

```
# Score the stacker model's prediction
skill.score(stacker.predict.xgb, test_set$turnover, brier.ref)
```

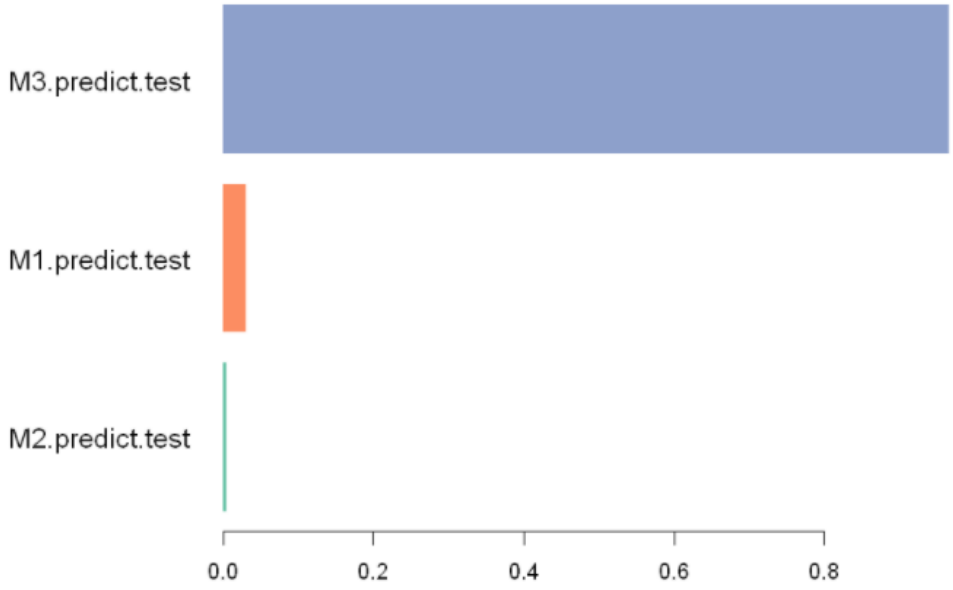
turnover	M1.predict.test	M2.predict.test	M3.predict.test
1	1.000000e+00	0.63157895	0.9969930649
1	9.999959e-01	1.00000000	0.9997059703
0	1.000000e+00	0.00000000	0.0001225407
1	1.000000e+00	1.00000000	0.9997116923
0	1.370081e-03	0.05882353	0.0078845769
0	2.220446e-16	0.00000000	0.0004496651

```
[18:17:34] WARNING: amalgamation/../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the
objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old
behavior.
```

0.947512252114369

In [76]:
```
mycolb <- brewer.pal(5, "Set2")
xgb.importance(model = stackerModel_2)
xgb.plot.importance(xgb.importance(model = stackerModel_2), top_n = 15, xlab= "importance", col = mycolb)
```

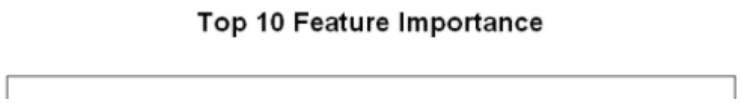
Feature	Gain	Cover	Frequency
M3.predict.test	0.965534376	0.91072771	0.65979381
M1.predict.test	0.030233910	0.07014126	0.27835052
M2.predict.test	0.004231714	0.01913103	0.06185567

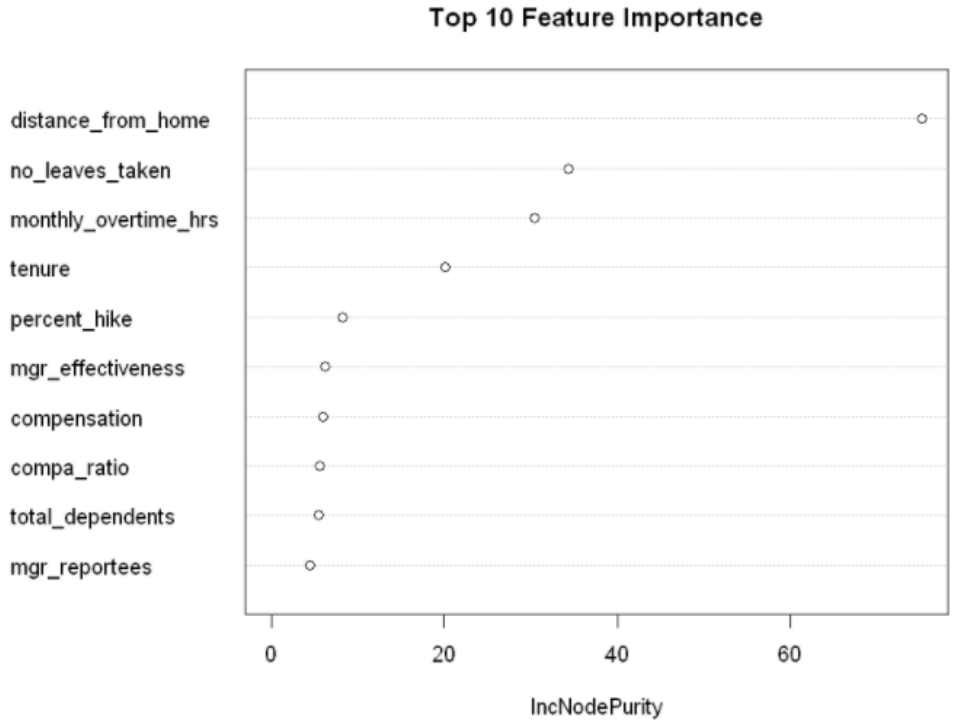


In [77]:
```
#train the random forest
rfModel <- randomForest(turnover ~., data = train_set)
print(rfModel)

# plot the importance of each variable.
varImpPlot(rfModel, sort=T, n.var = 10, main = 'Top 10 Feature Importance')

#predictions and confusion matrix
pred_rf <- predict(rfModel, test_set)
pred_rf_cat= ifelse(pred_rf > 0.5, "1", "0")
pred_rf_cat = as.factor(pred_rf_cat)
confusionMatrix(pred_rf_cat, as.factor(test_set$turnover))

# Calculate the Brier Skill Score
skill.score(pred_rf, test_set$turnover, brier.ref)

##random forest error rate
plot(rfModel)
```

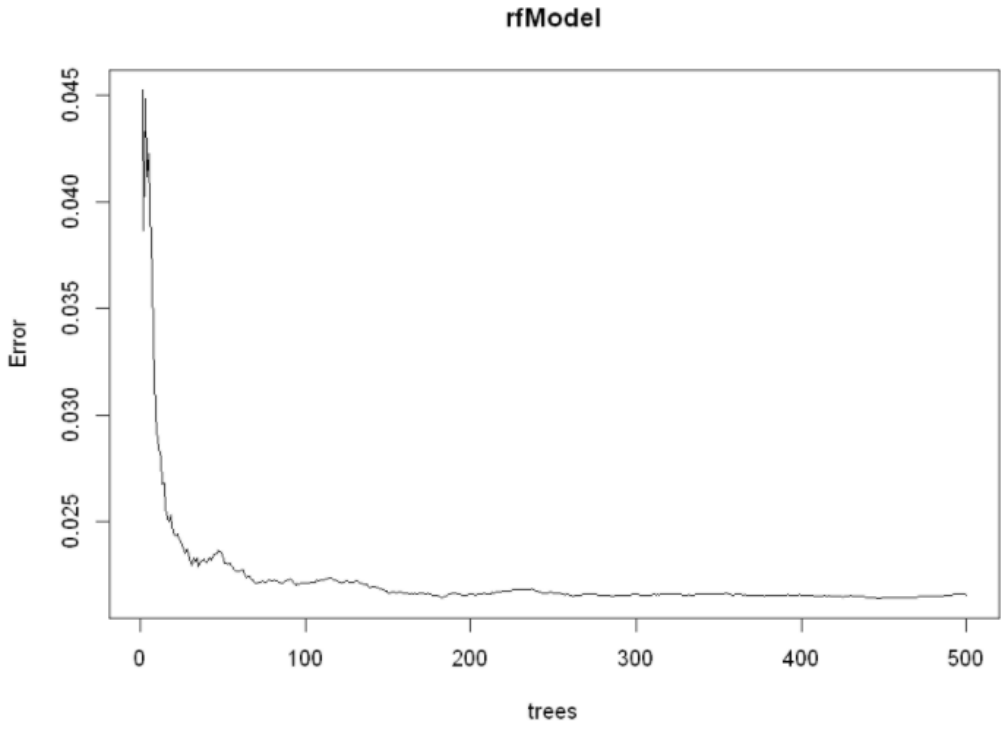
```
Warning message in randomForest.default(m, y, ...):
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

```
Call:
 randomForest(formula = turnover ~ ., data = train_set)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.02158997
                    % Var explained: 86.52

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 462  15
         1   1 108

               Accuracy : 0.9727
                 95% CI : (0.956, 0.9843)
    No Information Rate : 0.7901
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9141

 Mcnemar's Test P-Value : 0.001154

            Sensitivity : 0.9978
            Specificity : 0.8780
         Pos Pred Value : 0.9686
         Neg Pred Value : 0.9908
             Prevalence : 0.7901
         Detection Rate : 0.7884
   Detection Prevalence : 0.8140
      Balanced Accuracy : 0.9379

       'Positive' Class : 0


0.861793392922356
```




Top 10 Feature Importance
distance_from_home
no_leaves_taken
monthly_overtime_hrs
tenure
percent_hike
mgr_effectiveness
compensation
compa_ratio
total_dependents
mgr_reportees
0
20
40
60
IncNodePurity


rfModel
Error
0.030
0.035
0.040
0.045



In [78]:

```r
#Tune Random Forest Model
t <- tuneRF(train_set[, -12], train_set$turnover , stepFactor = 0.5, plot = TRUE, ntreeTry = 200, trace = TRUE, improve = 0.05)

#Fit the Random Forest Model After Tuning
rfModel_new <- randomForest(turnover ~., data = train_set, ntree = 200, mtry = 2, importance = TRUE, proximity = TRUE)
print(rfModel_new)

#Random Forest Predictions and Confusion Matrix After Tuning
pred_rf_new <- predict(rfModel_new, test_set)
pred_rf_new_cat= ifelse(pred_rf_new > 0.5, "1", "0")
pred_rf_new_cat = as.factor(pred_rf_new_cat)
confusionMatrix(pred_rf_new_cat, as.factor(test_set$turnover))

#Random Forest Feature Importance
varImpPlot(rfModel_new, sort=T, n.var = 10, main = 'Top 10 Feature Importance')

#Fit the Random Forest Model With Best mtry
rfModel_final <- randomForest(turnover ~., data = train_set, ntree = 200, mtry = 12, importance = TRUE, proximity = TRUE)
print(rfModel_final)

#Random Forest Predictions and Confusion Matrix After Tuning
pred_rf_final <- predict(rfModel_final, test_set)
pred_rf_final_cat= ifelse(pred_rf_final > 0.5, "1", "0")
pred_rf_final_cat= as.factor(pred_rf_final_cat)
confusionMatrix(pred_rf_final_cat, as.factor(test_set$turnover))

#Random Forest Feature Importance
```

```r
#Random Forest Predictions and Confusion Matrix After Tuning
pred_rf_final <- predict(rfModel_final, test_set)
pred_rf_final_cat= ifelse(pred_rf_final > 0.5, "1", "0")
pred_rf_final_cat= as.factor(pred_rf_final_cat)
confusionMatrix(pred_rf_final_cat, as.factor(test_set$turnover))

#Random Forest Feature Importance
varImpPlot(rfModel_final, sort=T, n.var = 10, main = 'Top 10 Feature Importance')

# Calculate the Brier Skill Score
skill.score(pred_rf_final, test_set$turnover, brier.ref)
```

```
Warning message in randomForest.default(x, y, mtry = mtryStart, ntree = ntreeTry, :
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

```
mtry = 12  OOB error = 0.02144094
Searching left ...
```

```
Warning message in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

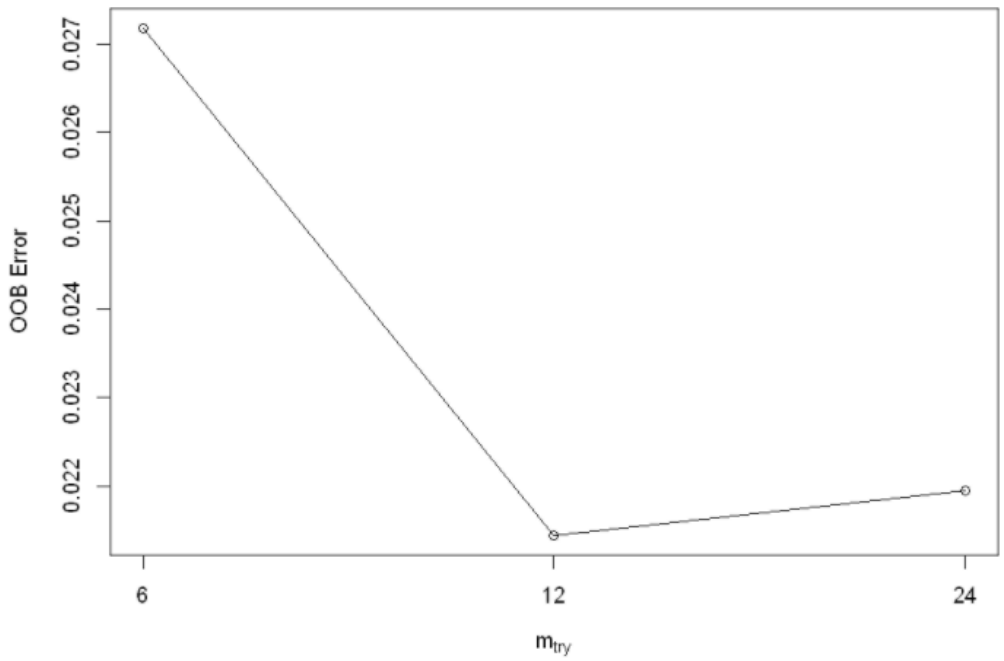
```
mtry = 24       OOB error = 0.02194464
-0.02349264 0.05
Searching right ...
```

```
Warning message in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

```
mtry = 6        OOB error = 0.02717689
-0.2675233 0.05
```

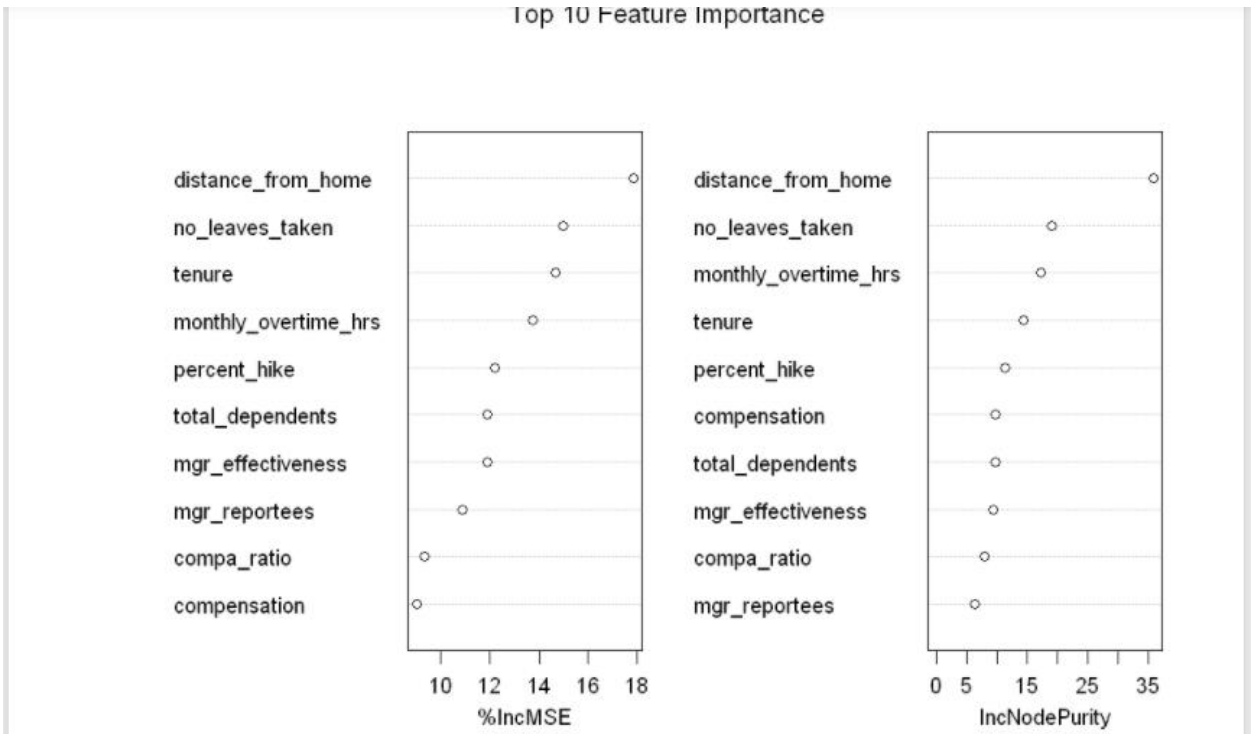
```
Warning message in randomForest.default(m, y, ...):
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

```
Call:
 randomForest(formula = turnover ~ ., data = train_set, ntree = 200,      mtry = 2, importance = TRUE, proximity = TRUE)
               Type of random forest: regression
                     Number of trees: 200
No. of variables tried at each split: 2

          Mean of squared residuals: 0.04583686
                    % Var explained: 71.38

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 463  24
         1   0  99

               Accuracy : 0.959
                 95% CI : (0.9397, 0.9736)
    No Information Rate : 0.7901
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.867

 Mcnemar's Test P-Value : 2.668e-06

            Sensitivity : 1.0000
            Specificity : 0.8049
         Pos Pred Value : 0.9507
         Neg Pred Value : 1.0000
             Prevalence : 0.7901
         Detection Rate : 0.7901
   Detection Prevalence : 0.8311
      Balanced Accuracy : 0.9024

       'Positive' Class : 0
```

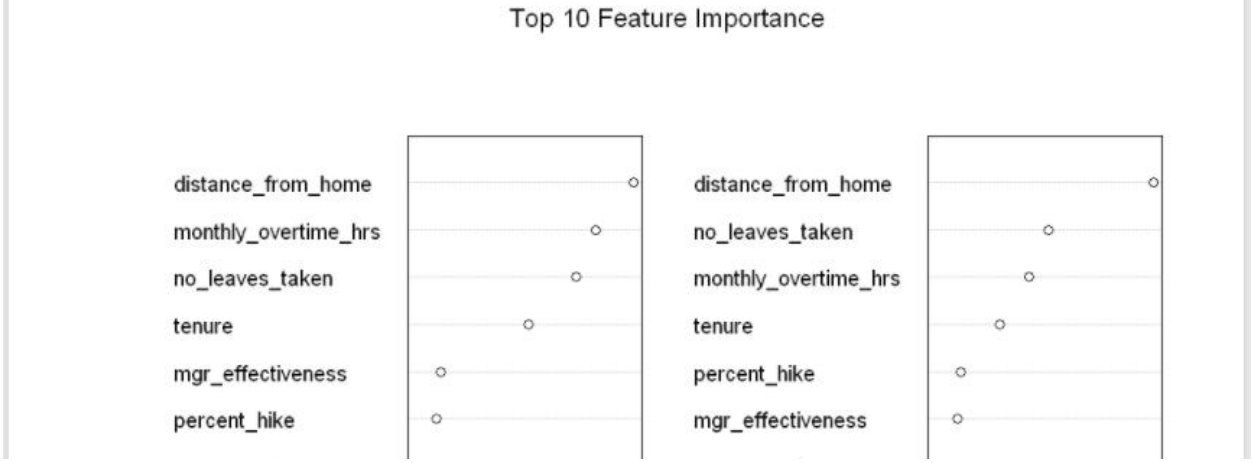



```
Warning message in randomForest.default(m, y, ...):
"The response has five or fewer unique values.  Are you sure you want to do regression?"
```

```
Call:
 randomForest(formula = turnover ~ ., data = train_set, ntree = 200,      mtry = 12, importance = TRUE, proximity = TRUE)
               Type of random forest: regression
                     Number of trees: 200
No. of variables tried at each split: 12

          Mean of squared residuals: 0.02155107
                    % Var explained: 86.55

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 461  14
         1   2 109

               Accuracy : 0.9727
                 95% CI : (0.956, 0.9843)
    No Information Rate : 0.7901
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9146

 Mcnemar's Test P-Value : 0.00596

            Sensitivity : 0.9957
            Specificity : 0.8862
         Pos Pred Value : 0.9705
         Neg Pred Value : 0.9820
```



0.868670232675955



References

[i] Info.workinstitute.com. 2021. [online] Available at: <https://info.workinstitute.com/hubfs/2019%20Retention%20Report/Work%20Institute%202019%20Retention%20Report%20final-1.pdf> [Accessed 27 April 2021].

[ii] Enrich.org. 2021. *The Cost of Replacing an Employee and the Role of Financial Wellness*. [online] Available at: <https://www.enrich.org/blog/The-true-cost-of-employee-turnover-financial-wellness-enrich#:~:text=Research%20by%20SHRM%20suggests%20that,overall%20losses%20to%20the%20company.> [Accessed 27 April 2021].

[iii] 2021. [online] Available at: <https://learn.datacamp.com/courses/human-resources-analytics-predicting-employee-churn-in-r> [Accessed 27 April 2021].

[iv] Enrich.org. 2021. *The Cost of Replacing an Employee and the Role of Financial Wellness*. [online] Available at: <https://www.enrich.org/blog/The-true-cost-of-employee-turnover-financial-wellness-enrich#:~:text=Research%20by%20SHRM%20suggests%20that,overall%20losses%20to%20the%20company.> [Accessed 27 April 2021].

[v] BBC News. 2021. *Facebook: Our staff can carry on working from home after Covid*. [online] Available at: <https://www.bbc.com/news/business-56759151> [Accessed 27 April 2021].