predicting the employee turnover data experiment

```r
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0

## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
org<-read.csv("org.csv")
glimpse(org)
```

```
## Rows: 2,291
## Columns: 14
## $ emp_id            <fct> E11061, E1031, E6213, E5900, E3044, E4008, E...
## $ status            <fct> Inactive, Inactive, Inactive, Inactive, Inac...
## $ turnover          <int> 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1,...
## $ location          <fct> New York, New York, New York, New York, Flor...
## $ level             <fct> Analyst, Analyst, Analyst, Analyst, Analyst,...
## $ date_of_joining   <fct> 22/03/2012, 09/03/2012, 06/01/2012, 22/03/20...
## $ date_of_birth     <fct> 22/03/1992, 10/01/1992, 06/02/1992, 19/12/19...
## $ last_working_date <fct> 11/09/2014, 05/06/2014, 30/04/2014, 09/04/20...
## $ gender            <fct> Male, Female, Female, Female, Female, Female...
## $ department        <fct> Customer Operations, Customer Operations, Cu...
## $ mgr_id            <fct> E1712, E10524, E4443, E3638, E3312, E13933, ...
## $ cutoff_date       <fct> 31/12/2014, 31/12/2014, 31/12/2014, 31/12/20...
## $ generation        <fct> Millennials, Millennials, Millennials, Mille...
## $ emp_age           <dbl> 22.5, 22.4, 22.2, 22.3, 22.1, 23.0, 23.0, 23...
```

```r
dim(org)
```

```
## [1] 2291   14
```

Turnover rate = Number of employees who left / Total number of employees

```r
org%>%
  count(status) # see how many employees still be active
```

```
##     status    n
## 1   Active 1881
## 2 Inactive  410
```

```r
org%>%
  summarise(turnover_rate=mean(turnover)) # Since 1 is inactive employee, so the rate is approxmation 1
```

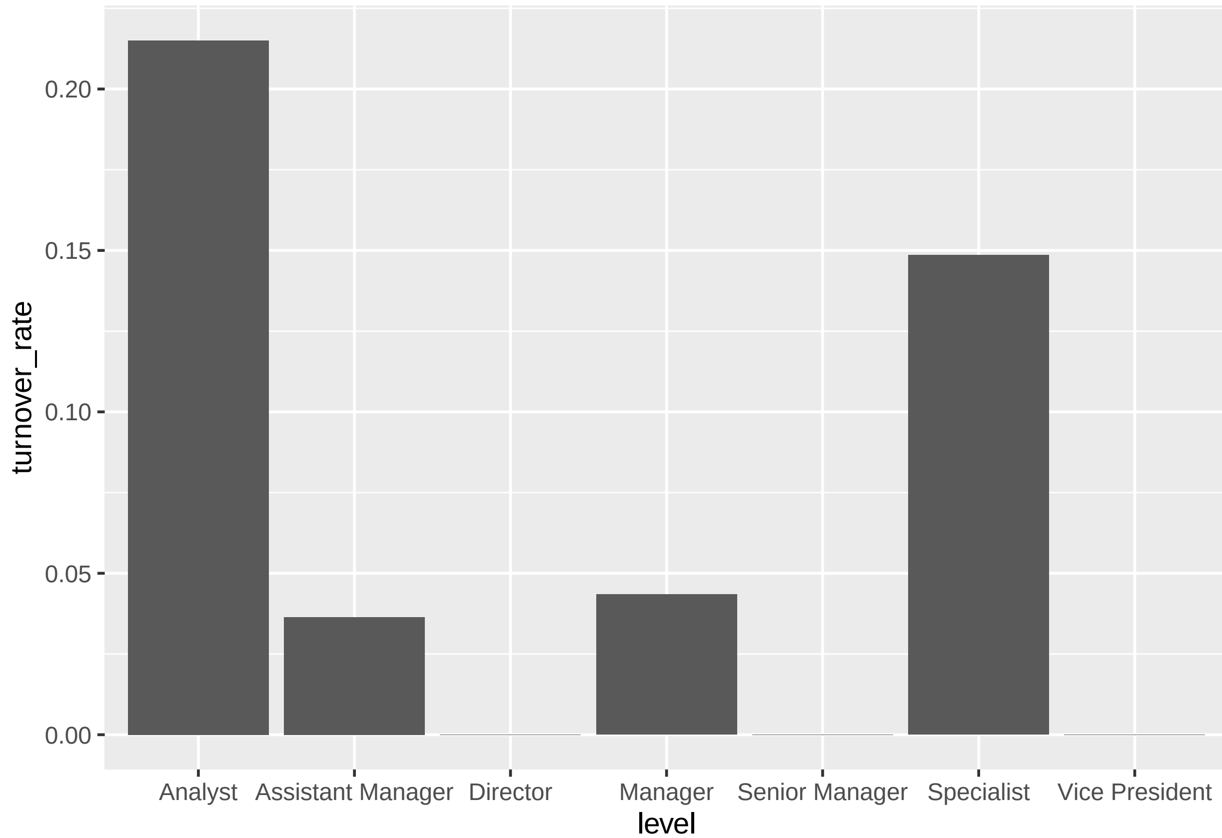
```
##   turnover_rate
## 1     0.1789612
```

```r
level<- org%>% #checking the rate of turnover between different level
  group_by(level)%>%
  summarise(turnover_rate=mean(turnover)) ; level
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 2
##   level             turnover_rate
##   <fct>                     <dbl>
## 1 Analyst                  0.215
## 2 Assistant Manager        0.0365
## 3 Director                 0
## 4 Manager                  0.0435
## 5 Senior Manager           0
## 6 Specialist               0.149
## 7 Vice President           0
```

```r
library(ggplot2) #plot the histogram to see the turnover_rate of level
library(broom)
level%>%
  ggplot(aes(level, turnover_rate))+geom_col()
```

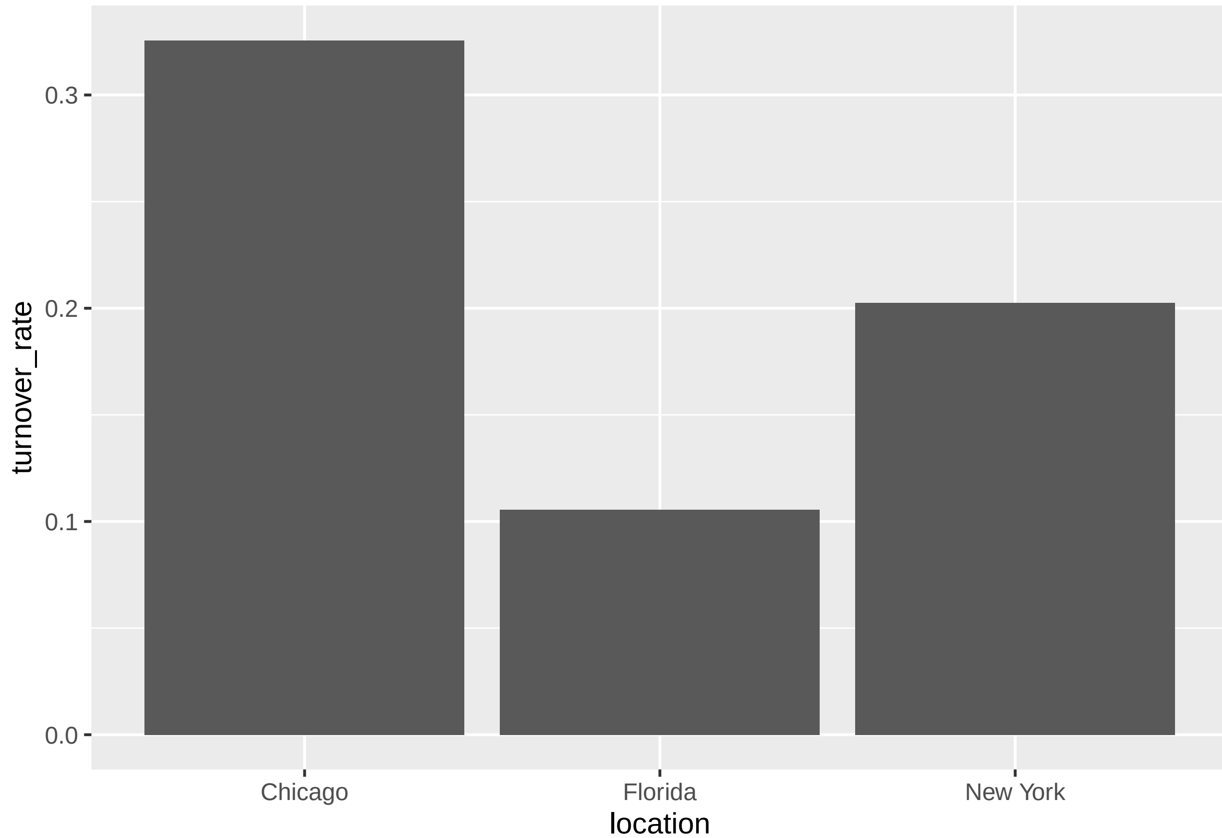



```
location<- org%>%  # checking the turnover_rate of location
  group_by(location)%>%
  summarise(turnover_rate=mean(turnover)) ;  location
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   location turnover_rate
##   <fct>            <dbl>
## 1 Chicago          0.326
## 2 Florida          0.106
## 3 New York         0.203
```

```
location%>%  #histogram to data visulization
  ggplot(aes(location,turnover_rate))+ geom_col()
```




```r
org1<-org%>% # first subset the job level in Analyst and Specialist
  filter(level %in% c("Analyst","Specialist"))
dim(org1)
```

```
## [1] 1954   14
```

```r
org%>%
  count(level)  # total number for each level
```

```
##               level    n
## 1           Analyst 1604
## 2 Assistant Manager  192
## 3          Director    1
## 4           Manager  138
## 5    Senior Manager    5
## 6        Specialist  350
## 7    Vice President    1
```

```r
org1%>%
  count(level) #total number between Analyst and Specialist
```

```
##        level    n
## 1    Analyst 1604
## 2 Specialist  350
```

```
rating<-read.csv("rating.csv")
dim(rating)
```

```
## [1] 1954    2
```

```
org2 <- left_join(org1,rating, by = "emp_id") # combine two table to see if the rate is effect for turn
dim(org2)
```

```
## [1] 1954   15
```

```
org2%>%   # calculate the turnover_rate in rating
  group_by(rating)%>%
  summarise(turnover_rate=mean(turnover))
```

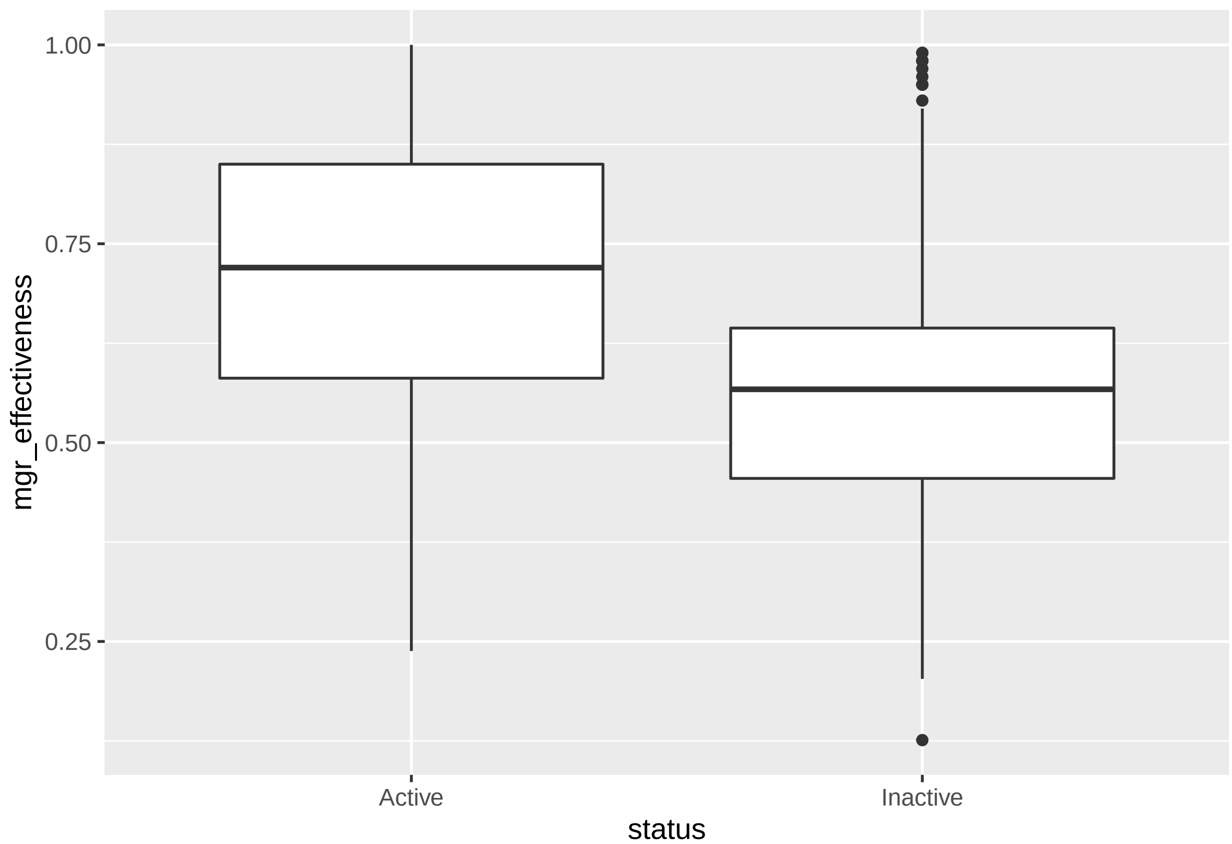
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   rating        turnover_rate
##   <fct>                 <dbl>
## 1 Above Average        0.131
## 2 Acceptable           0.221
## 3 Below Average        0.385
## 4 Excellent            0.0305
## 5 Unacceptable         0.633
```

```
survey<-read.csv("survey.csv")
glimpse(survey)
```

```
## Rows: 350
## Columns: 5
## $ mgr_id              <fct> E1003, E10072, E10081, E10234, E1026, E104...
## $ mgr_effectiveness   <dbl> 0.760, 0.650, 0.800, 0.650, 0.700, 0.980, ...
## $ career_satisfaction <dbl> 0.76, 0.67, 0.82, 0.63, 1.00, 0.91, 0.56, ...
## $ perf_satisfaction   <dbl> 0.71, 0.56, 0.73, 0.75, 1.00, 0.91, 0.50, ...
## $ work_satisfaction   <dbl> 0.82, 0.84, 0.84, 0.70, 0.92, 0.77, 0.81, ...
```

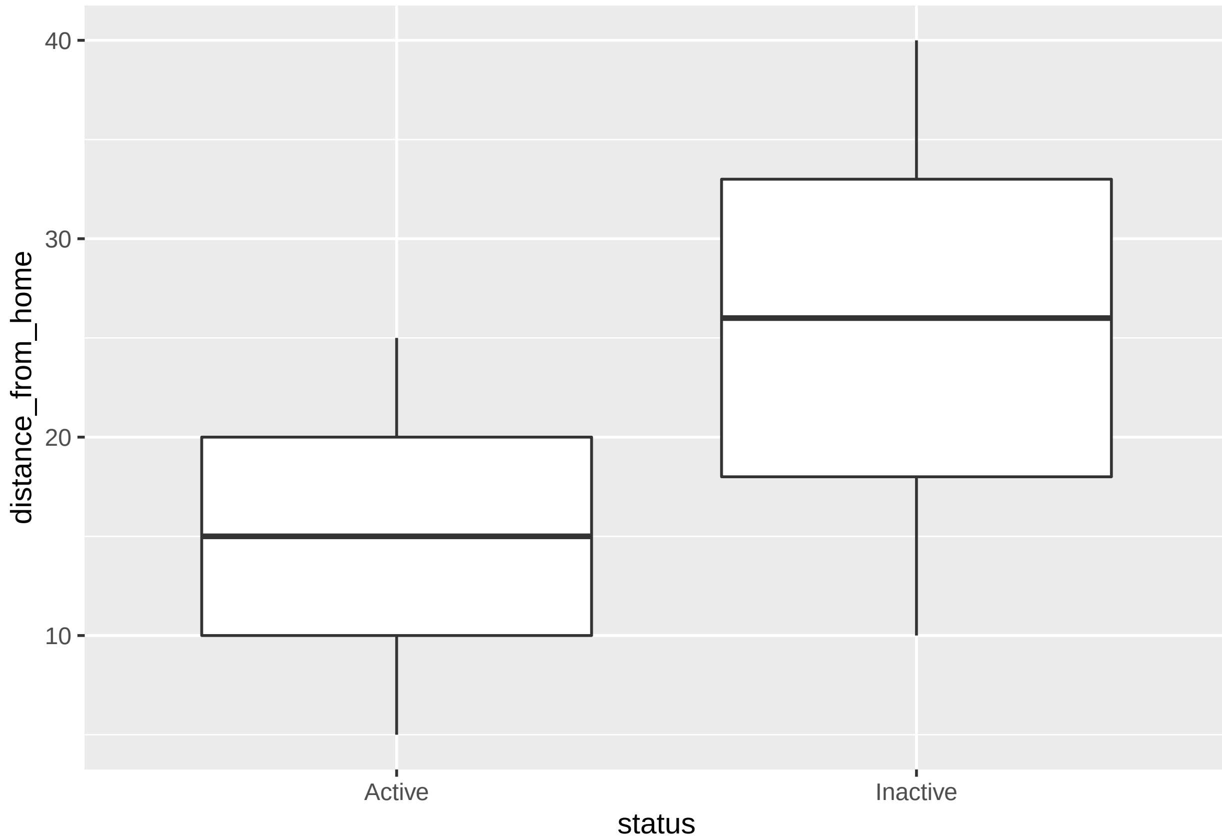
```
org3 <- left_join(org2, survey, by = "mgr_id") # combine the table between the org2 and survey
org3%>%
  ggplot(aes(status,mgr_effectiveness))+geom_boxplot() #graph to show if the effectiveness relationship
```




```
org_final<-read.csv("org_final.csv")
dim(org_final)
```

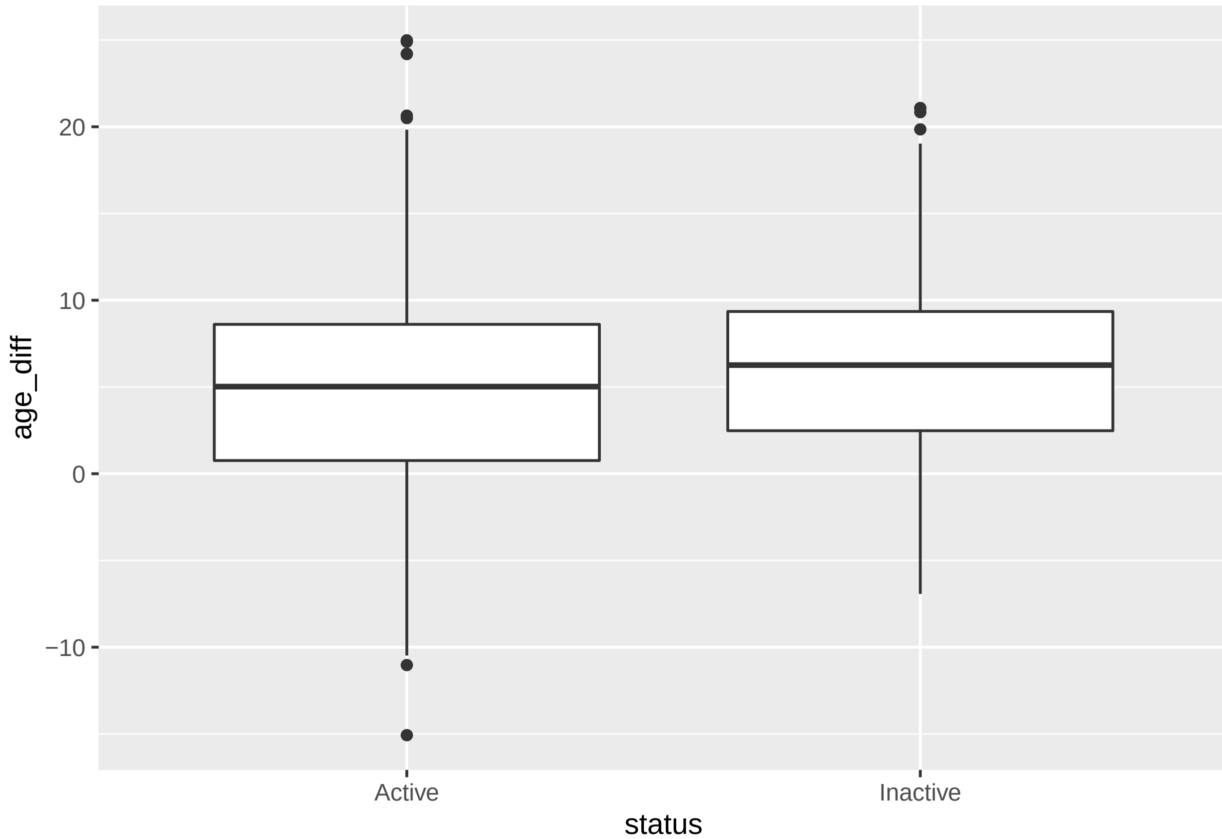
```
## [1] 1954   34
```

```
org_final%>%
  ggplot(aes(status, distance_from_home))+geom_boxplot() # graph for the relationship between distance
```



Job-hop Index = Total experience / Number of companies

```
emp_age_diff<-org_final%>%
  mutate(age_diff= mgr_age-emp_age)
emp_age_diff%>%
  ggplot(aes(status,age_diff))+geom_boxplot()
```



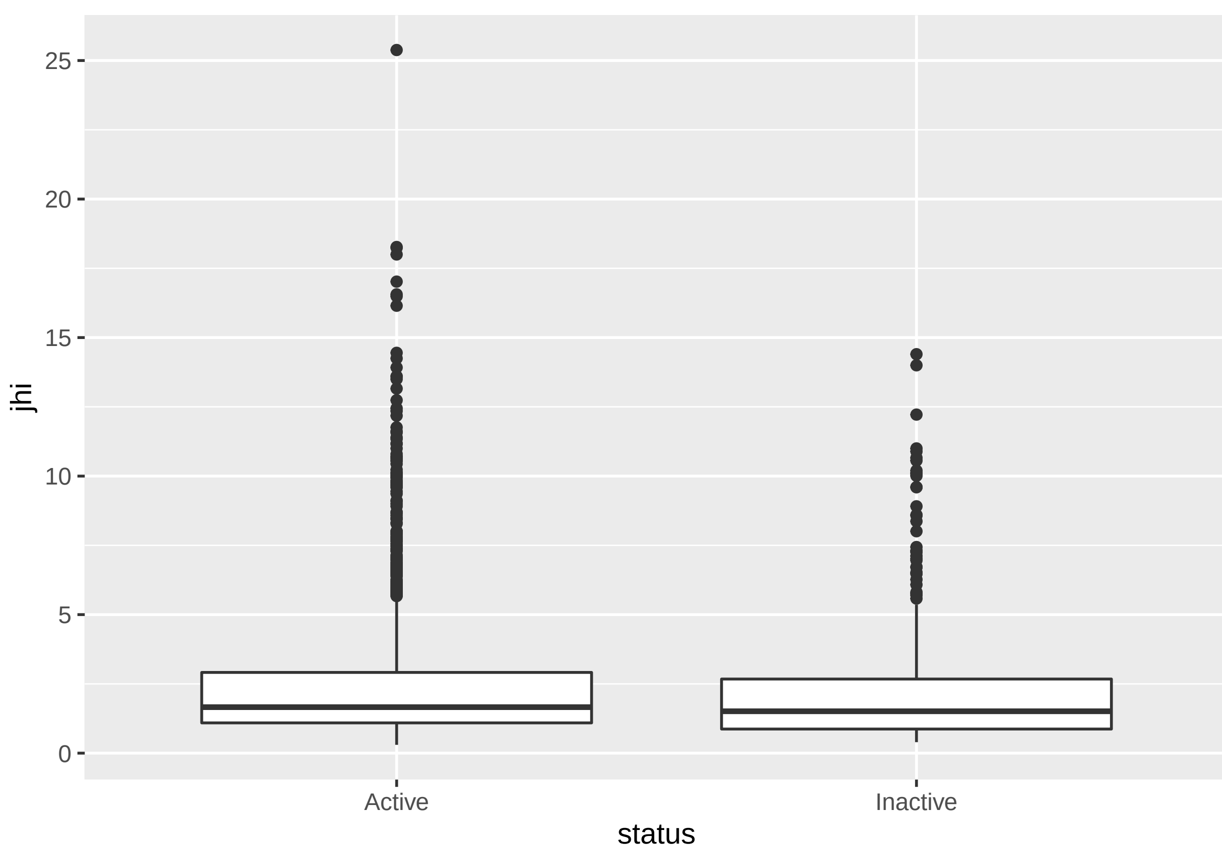

```r
glimpse(emp_age_diff)
```

```
## Rows: 1,954
## Columns: 35
## $ emp_id                       <fct> E10012, E10025, E10027, E10048, E...
## $ status                       <fct> Active, Active, Active, Active, A...
## $ location                     <fct> New York, Chicago, Orlando, Chica...
## $ level                        <fct> Analyst, Analyst, Specialist, Spe...
## $ gender                       <fct> Female, Female, Female, Male, Mal...
## $ emp_age                      <dbl> 25.09, 25.98, 33.40, 24.55, 31.23...
## $ rating                       <fct> Above Average, Acceptable, Accept...
## $ mgr_rating                   <fct> Acceptable, Excellent, Above Aver...
## $ mgr_reportees                <int> 9, 4, 6, 10, 11, 19, 21, 9, 12, 2...
## $ mgr_age                      <dbl> 44.07, 35.99, 35.78, 26.70, 34.28...
## $ mgr_tenure                   <dbl> 3.17, 7.92, 4.38, 2.87, 12.95, 10...
## $ compensation                 <int> 64320, 48204, 85812, 49536, 75576...
## $ percent_hike                 <int> 10, 8, 11, 8, 12, 8, 12, 9, 9, 6,...
## $ hiring_score                 <int> 70, 70, 77, 71, 70, 75, 72, 70, 7...
## $ hiring_source                <fct> Consultant, Job Fairs, Consultant...
## $ no_previous_companies_worked <int> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 3, ...
## $ distance_from_home           <int> 14, 21, 15, 9, 25, 23, 17, 16, 22...
## $ total_dependents             <int> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, ...
## $ marital_status               <fct> Single, Single, Single, Single, S...
## $ education                    <fct> Bachelors, Bachelors, Bachelors, ...
## $ promotion_last_2_years       <fct> No, No, Yes, Yes, No, No, No, No,...
```

```
## $ no_leaves_taken             <int> 2, 10, 18, 19, 25, 15, 10, 20, 22...
## $ total_experience            <dbl> 6.86, 4.88, 8.55, 4.76, 8.06, 13....
## $ monthly_overtime_hrs        <int> 1, 5, 3, 8, 1, 7, 2, 10, 2, 10, 8...
## $ date_of_joining             <fct> 06/03/2011, 23/09/2009, 02/11/200...
## $ last_working_date           <fct> NA, NA, NA, NA, NA, 11/12/2014, N...
## $ department                  <fct> Customer Operations, Customer Ope...
## $ mgr_id                      <fct> E9335, E6655, E13942, E7063, E566...
## $ cutoff_date                 <fct> 31/12/2014, 31/12/2014, 31/12/201...
## $ turnover                    <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, ...
## $ mgr_effectiveness           <dbl> 0.730, 0.581, 0.770, 0.240, 0.710...
## $ career_satisfaction         <dbl> 0.73, 0.72, 0.85, 0.42, 0.78, 0.8...
## $ perf_satisfaction           <dbl> 0.73, 0.84, 0.80, 0.33, 0.67, 0.8...
## $ work_satisfaction           <dbl> 0.75, 0.85, 0.87, 0.85, 0.80, 0.8...
## $ age_diff                    <dbl> 18.98, 10.01, 2.38, 2.15, 3.05, 2...
```

```
emp_JHI<-emp_age_diff%>%
  mutate(jhi=total_experience / no_previous_companies_worked) #calcultate the Job hop for each emplyees
emp_JHI%>%
  ggplot(aes(status,jhi))+geom_boxplot() # box-plot to demonstrate the outliers and mean
```

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

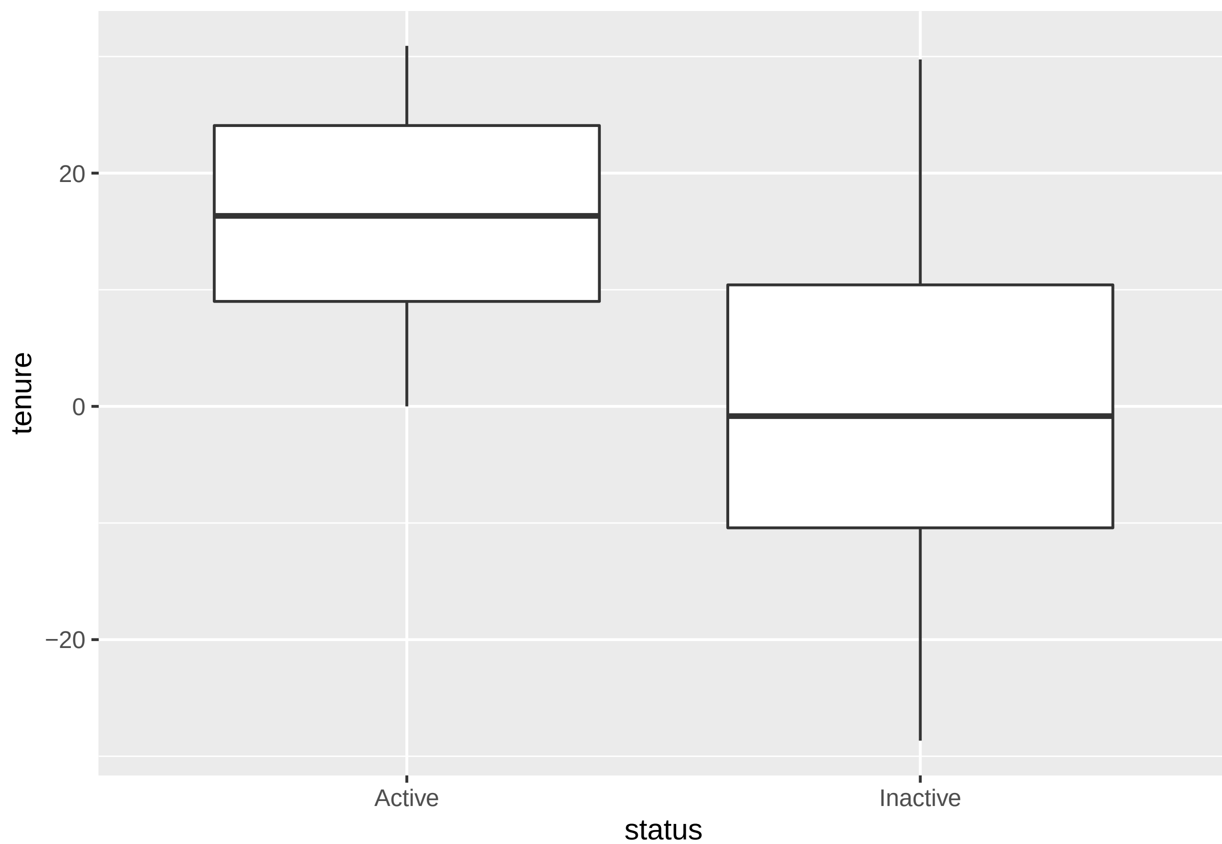



```
library(lubridate) #load package for manipulation the time
```

```
##
```

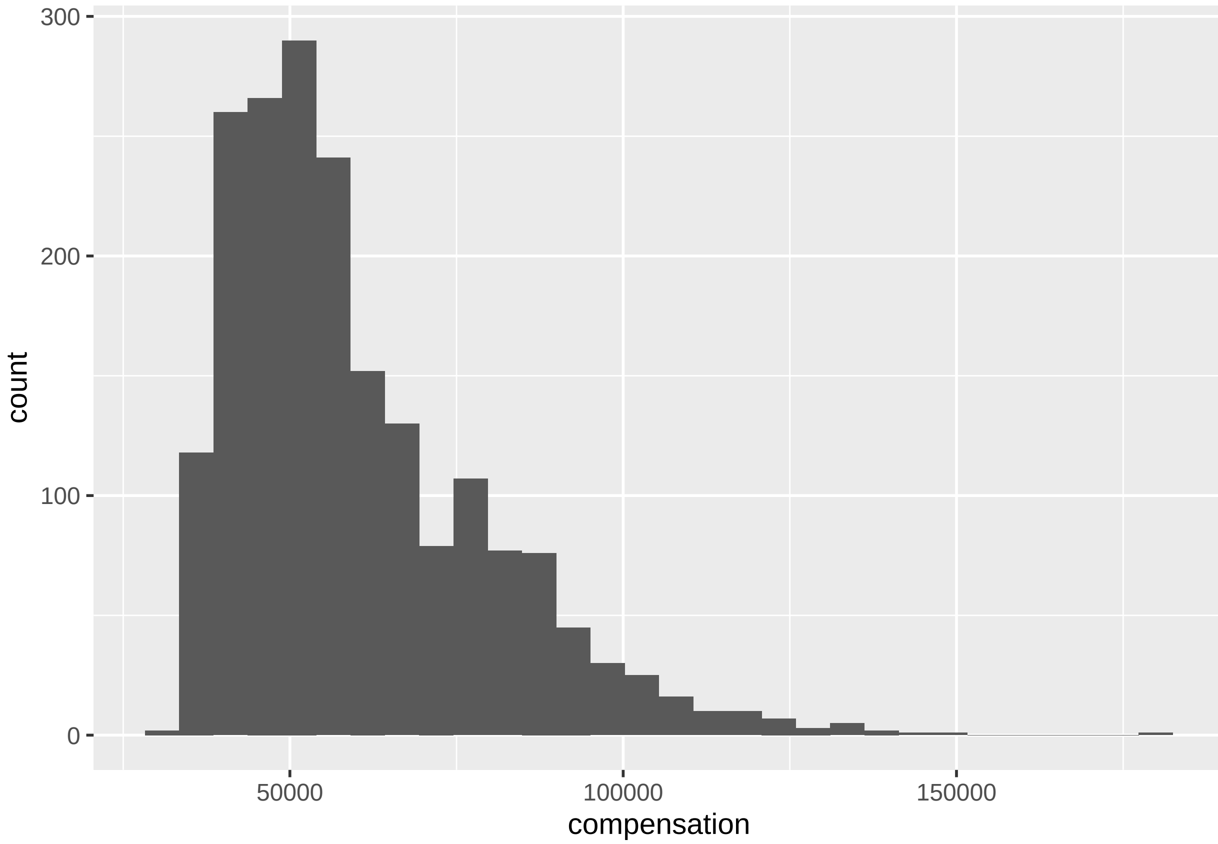
```
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
emp_tenure<- emp_JHI%>%
  mutate(tenure = ifelse(status=="Active",
         time_length(interval(date_of_joining, cutoff_date), "years"),
         time_length(interval(date_of_joining, last_working_date), "years"))) #add column for work dura
emp_tenure%>%
  ggplot(aes(status,tenure))+geom_boxplot() #box plot displaying
```

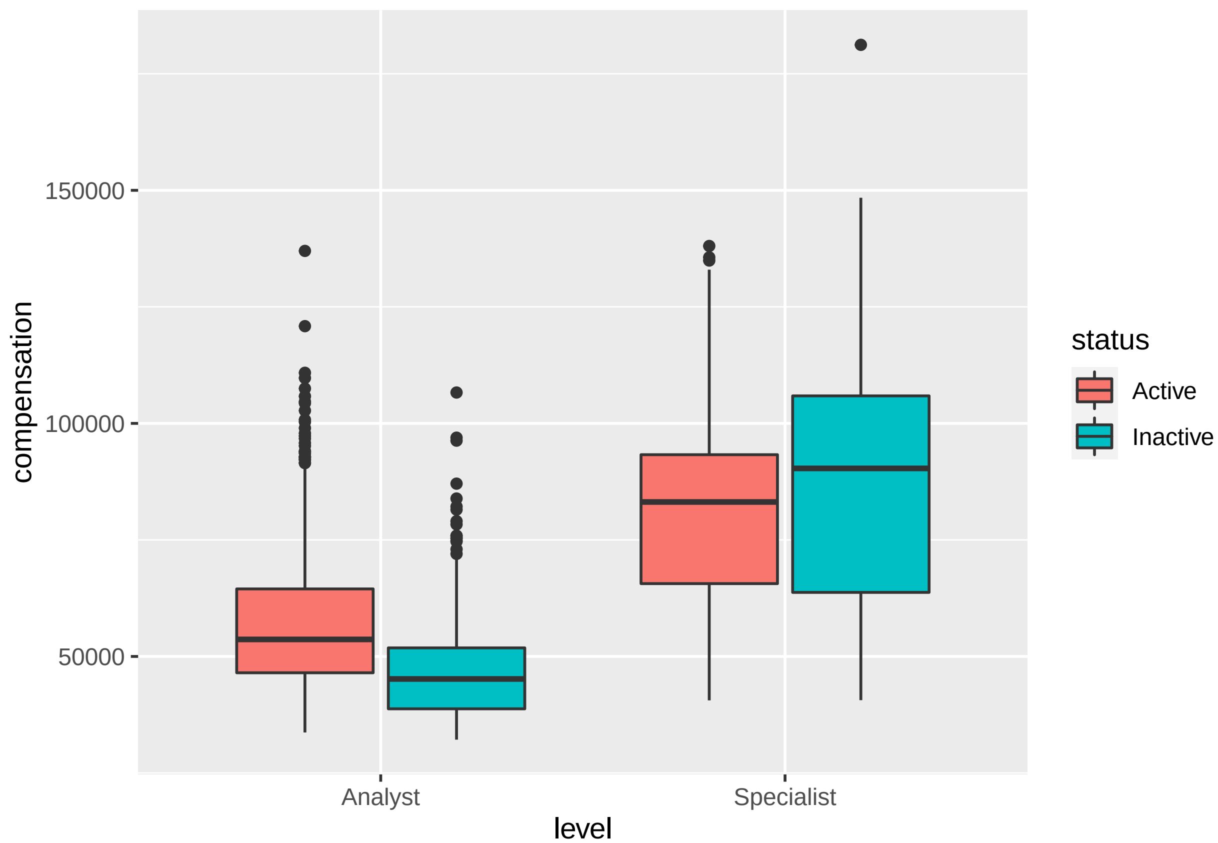



```
emp_tenure%>%
  ggplot(aes(compensation))+geom_histogram() #plot the distribution for compensation
```

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




```
emp_tenure%>%
  ggplot(aes(level, compensation, fill=status))+geom_boxplot() # graph to compare the compensation with
```

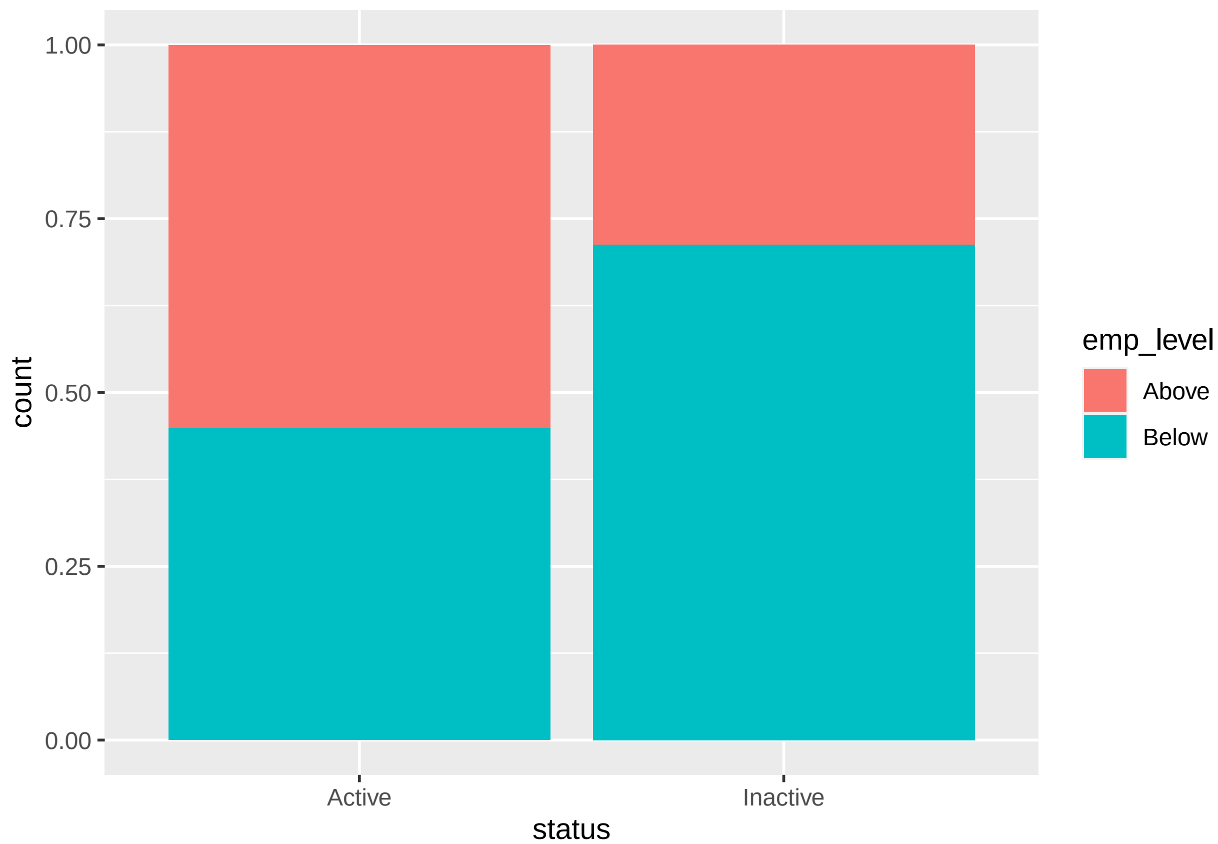


Compa Ratio is estimation to evaluate the employee wage percentage to median pay. ##Compa Ratio = Actual Compensation / Median Compensation

```
emp_ratio<- emp_tenure%>%
  group_by(level)%>%
  mutate(median_compensation = median(compensation),
         compa_ratio = (compensation / median_compensation)) # derive compensation ratio
emp_ratio%>%
  distinct(level,median_compensation) # look at the median compensation for each level
```

```
## # A tibble: 2 x 2
## # Groups:   level [2]
##   level       median_compensation
##   <fct>                     <dbl>
## 1 Analyst                   51840
## 2 Specialist                83496
```

```
emp_final<- emp_ratio%>%
  mutate(emp_level = ifelse( compa_ratio > 1, "Above", "Below")) # add compa level , if compa_ration ge
emp_final%>%
  ggplot(aes(status, fill = emp_level))+geom_bar(position = "fill") #compare compa level between active
```



Unstanding information value : measure of predictive power of independent variable to accurately predict the dependent variable

Information value = sima(% of non-events-% of events))* log(% of non-events/% of events)

information value : less than 0.15 meaning predictive power is poor, if 0.15 < IV < 0.4 id moderate, else greater than 0.4 meaning strong.

```r
library(Information)
IV <- create_infotables(data = emp_final, y = "turnover")
```

```
## [1] "Variable emp_id was removed because it is a non-numeric variable with >1000 categories"
## [1] "Variable department was removed because it has only 1 unique value"
## [1] "Variable cutoff_date was removed because it has only 1 unique value"
```

```r
IV$Summary  #after we calculate the information value, we can see which variables are significant stron
```

```
##                     Variable           IV
## 12              percent_hike 1.144784e+00
## 17          total_dependents 1.088645e+00
```

```
## 21                no_leaves_taken 9.404533e-01
## 33                         tenure 7.636901e-01
## 27              mgr_effectiveness 6.830020e-01
## 11                   compensation 6.074885e-01
## 35                    compa_ratio 4.768892e-01
## 24                date_of_joining 4.330804e-01
## 6                          rating 3.869373e-01
## 23           monthly_overtime_hrs 3.786644e-01
## 8                   mgr_reportees 3.620543e-01
## 2                        location 2.963023e-01
## 36                      emp_level 2.940446e-01
## 26                         mgr_id 2.820235e-01
## 5                         emp_age 2.275477e-01
## 16             distance_from_home 1.470549e-01
## 30              work_satisfaction 1.378953e-01
## 22               total_experience 1.345781e-01
## 19                      education 1.253865e-01
## 20         promotion_last_2_years 9.979915e-02
## 9                         mgr_age 9.816205e-02
## 29              perf_satisfaction 7.099511e-02
## 13                   hiring_score 6.684727e-02
## 31                       age_diff 6.634065e-02
## 32                            jhi 6.586588e-02
## 10                     mgr_tenure 5.918048e-02
## 28            career_satisfaction 3.539857e-02
## 3                           level 2.726491e-02
## 34            median_compensation 2.726491e-02
## 18                 marital_status 2.588063e-02
## 7                      mgr_rating 2.172222e-02
## 15 no_previous_companies_worked 1.729893e-02
## 14                  hiring_source 8.773529e-03
## 4                          gender 3.959968e-05
## 1                          status 0.000000e+00
## 25              last_working_date 0.000000e+00
```

split the data 70% into train and 30% into test

```
library(ISLR)
smp_siz <- floor(0.7 * nrow(emp_final) )
smp_siz
```

```
## [1] 1367
```

```
set.seed(1234)
train_ind<-sample(seq_len(nrow(emp_final)), size = smp_siz)
train <- emp_final [train_ind,]
test<- emp_final [-train_ind,]
```

```
train%>%
  count(status)%>%
  mutate(prop=n/sum(n)) #calculate the proportion in train for level and status
```

```
## # A tibble: 4 x 4
## # Groups:   level [2]
##   level      status       n  prop
##   <fct>      <fct>    <int> <dbl>
## 1 Analyst    Active     874 0.780
## 2 Analyst    Inactive   246 0.220
## 3 Specialist Active     212 0.858
## 4 Specialist Inactive    35 0.142
```

```
test%>%
  count(status)%>%
  mutate(prop=n/sum(n)) # calculate the proportion in test for level and status
```

```
## # A tibble: 4 x 4
## # Groups:   level [2]
##   level      status       n  prop
##   <fct>      <fct>    <int> <dbl>
## 1 Analyst    Active     385 0.795
## 2 Analyst    Inactive    99 0.205
## 3 Specialist Active      86 0.835
## 4 Specialist Inactive    17 0.165
```

```
log<- glm(turnover ~ percent_hike,
          family= "binomial",
          data=train) # build a logistic regression using percent_hike to predict turnover
summary(log)
```

```
##
## Call:
## glm(formula = turnover ~ percent_hike, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9286  -0.7093  -0.4514  -0.2808   2.6717
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.69046    0.23039   7.337 2.18e-13 ***
## percent_hike -0.32692    0.02521 -12.967  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1388.9  on 1366  degrees of freedom
## Residual deviance: 1167.1  on 1365  degrees of freedom
## AIC: 1171.1
##
## Number of Fisher Scoring iterations: 5
```

```
mul_log<- glm(turnover~ level+gender+mgr_rating+compensation+hiring_score+marital_status+distance_from_
              family="binomial",
```

```r
             data=train) #bulid a multiple regression model for couple independent variables to predic
summary(mul_log)
```

```
##
## Call:
## glm(formula = turnover ~ level + gender + mgr_rating + compensation +
##     hiring_score + marital_status + distance_from_home + monthly_overtime_hrs +
##     work_satisfaction, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7460  -0.5310  -0.2587  -0.1051   2.9871
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -4.140e+00  2.296e+00  -1.803 0.071379 .
## levelSpecialist          7.524e-01  2.889e-01   2.605 0.009196 **
## genderMale               3.537e-01  1.948e-01   1.816 0.069360 .
## mgr_ratingAcceptable     3.352e-01  2.110e-01   1.588 0.112280
## mgr_ratingBelow Average -2.210e-01  3.808e-01  -0.580 0.561713
## mgr_ratingExcellent      2.069e-01  3.089e-01   0.670 0.502907
## mgr_ratingUnacceptable  -1.212e+00  9.908e-01  -1.223 0.221234
## compensation            -4.466e-05  6.780e-06  -6.587 4.48e-11 ***
## hiring_score             2.765e-02  2.925e-02   0.945 0.344466
## marital_statusSingle    -1.007e-01  2.409e-01  -0.418 0.676056
## distance_from_home       2.141e-01  1.458e-02  14.677  < 2e-16 ***
## monthly_overtime_hrs     1.767e-01  2.491e-02   7.096 1.28e-12 ***
## work_satisfaction       -2.923e+00  8.538e-01  -3.424 0.000617 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1388.91  on 1366  degrees of freedom
## Residual deviance:  848.93  on 1354  degrees of freedom
## AIC: 874.93
##
## Number of Fisher Scoring iterations: 6
```

```r
mul_log1<- glm(turnover~ level+compensation+distance_from_home+monthly_overtime_hrs+work_satisfaction,
               family="binomial",
               data=train)
summary(mul_log1)
```

```
##
## Call:
## glm(formula = turnover ~ level + compensation + distance_from_home +
##     monthly_overtime_hrs + work_satisfaction, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.6147  -0.5487  -0.2623  -0.1091   2.9730
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.601e+00  8.249e-01  -1.940   0.0523 .
## levelSpecialist        7.406e-01  2.900e-01   2.554   0.0107 *
## compensation          -4.230e-05  6.456e-06  -6.551 5.70e-11 ***
## distance_from_home     2.121e-01  1.444e-02  14.691  < 2e-16 ***
## monthly_overtime_hrs   1.691e-01  2.454e-02   6.891 5.55e-12 ***
## work_satisfaction     -3.247e+00  8.143e-01  -3.987 6.68e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1388.91  on 1366  degrees of freedom
## Residual deviance:  859.92  on 1361  degrees of freedom
## AIC: 871.92
##
## Number of Fisher Scoring iterations: 6
```

Variance Inflation Factor

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

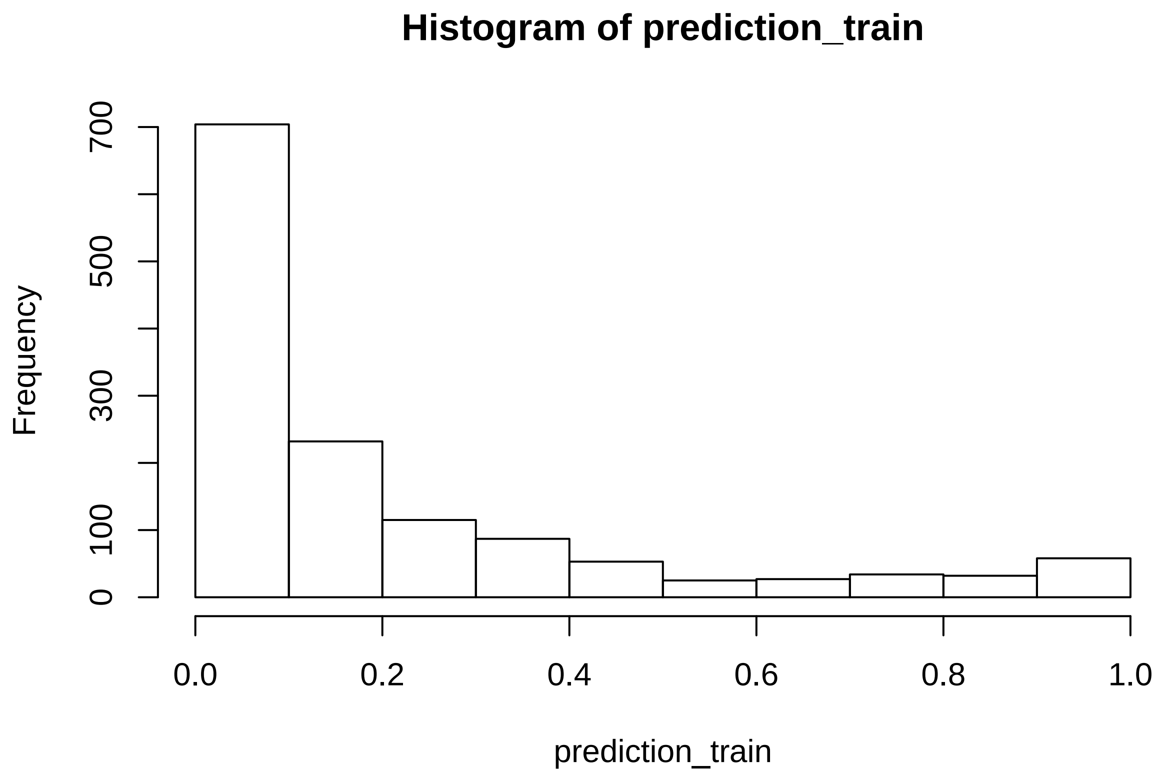
```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

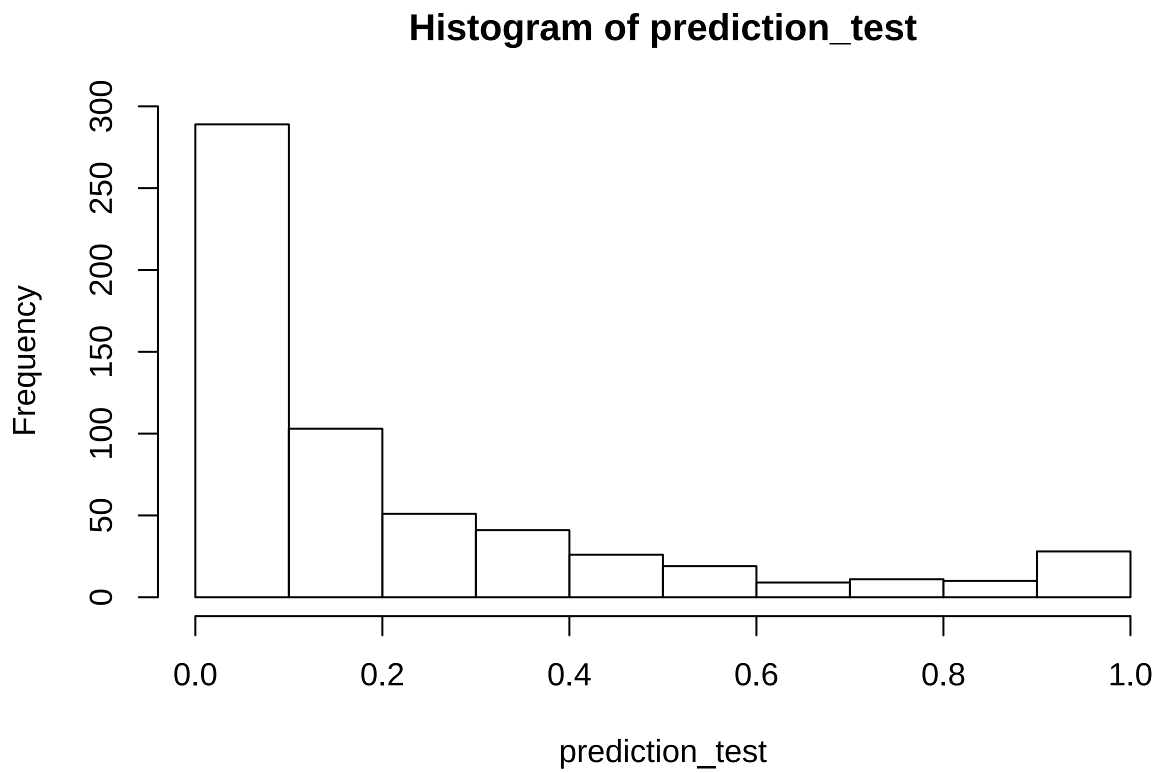
```
vif(mul_log1) #check a multicollinearity, the vif for each variables is greater than 1 but less than 2.
```

```
##                level         compensation   distance_from_home
##             1.483949             1.529480             1.050157
## monthly_overtime_hrs    work_satisfaction
##             1.027776             1.035387
```

```
prediction_train<- predict(mul_log1, newdata= train,
                           type = "response")
hist(prediction_train) # distribution skewed left, and histogram shown the probability to the employees
```




```
prediction_test<-predict(mul_log1 , newdata = test,
                          type = "response")
hist(prediction_test) # check a train data into test data
```



Turn probabilities in categories by using a cut-off

```
pre_cut <- ifelse(prediction_test >0.5 , 1 ,0) #classify predictions using a cut-off of 0.5
conf_matrix<- table(pre_cut , test$turnover)
conf_matrix # 1 means inactive while 0 is active
```

```
##
## pre_cut   0   1
##       0 456  54
##       1  15  62
```

```
n<-sum(conf_matrix) #number of instances
nc<- nrow(conf_matrix) # number of classes
diag <- diag(conf_matrix) # number of correctly classified instances per class
rowsums <- apply(conf_matrix, 1, sum) # number of instances per class
colsums <- apply(conf_matrix, 2, sum) # number of predictions per class
p <- rowsums / n # distribution of instances over the actual classes
q <- colsums / n # distribution of instances over the predicted classes
```

```
accuracy <- sum(diag) / n ; accuracy # the model's accuracy is 0.88
```

```
## [1] 0.8824532
```

```r
precision <- diag/colsums;precision # the model's precision to active is 0.97 and inactive 0.53
```

```
##         0         1 
## 0.9681529 0.5344828
```

create retension strategy

```r
library(tidypredict)
emp_risk<- emp_final %>%
  filter (status == "Active")%>%
  tidypredict_to_column(mul_log1) # calculate probability of turnover and add predictions using the mul
```

```r
emp_risk %>%
  select(emp_id , fit)%>%
  group_by(level)%>%
  top_n(5, wt = fit)%>%
  arrange(desc(fit))  # look at the employee's probability of turnover from high to low
```

```
## Adding missing grouping variables: `level`
```

```
## # A tibble: 10 x 3
## # Groups:   level [2]
##    level      emp_id   fit
##    <fct>      <fct>  <dbl>
##  1 Analyst    E277   0.728
##  2 Analyst    E7328  0.716
##  3 Specialist E10412 0.715
##  4 Analyst    E1800  0.706
##  5 Analyst    E5942  0.704
##  6 Analyst    E6249  0.683
##  7 Specialist E440   0.611
##  8 Specialist E13662 0.569
##  9 Specialist E13617 0.548
## 10 Specialist E10462 0.526
```

```r
emp_risk_bucket <- emp_risk%>%
  mutate(risk_bucket =cut(fit, breaks =c(0,0.3,0.5,0.7,1),
                          labels = c("no-risk", "low-risk", "medium-risk", "high-risk")))
emp_risk_bucket%>%
  count(risk_bucket)%>% #calculate the risk of turnover to the active employee
  group_by(risk_bucket)
```

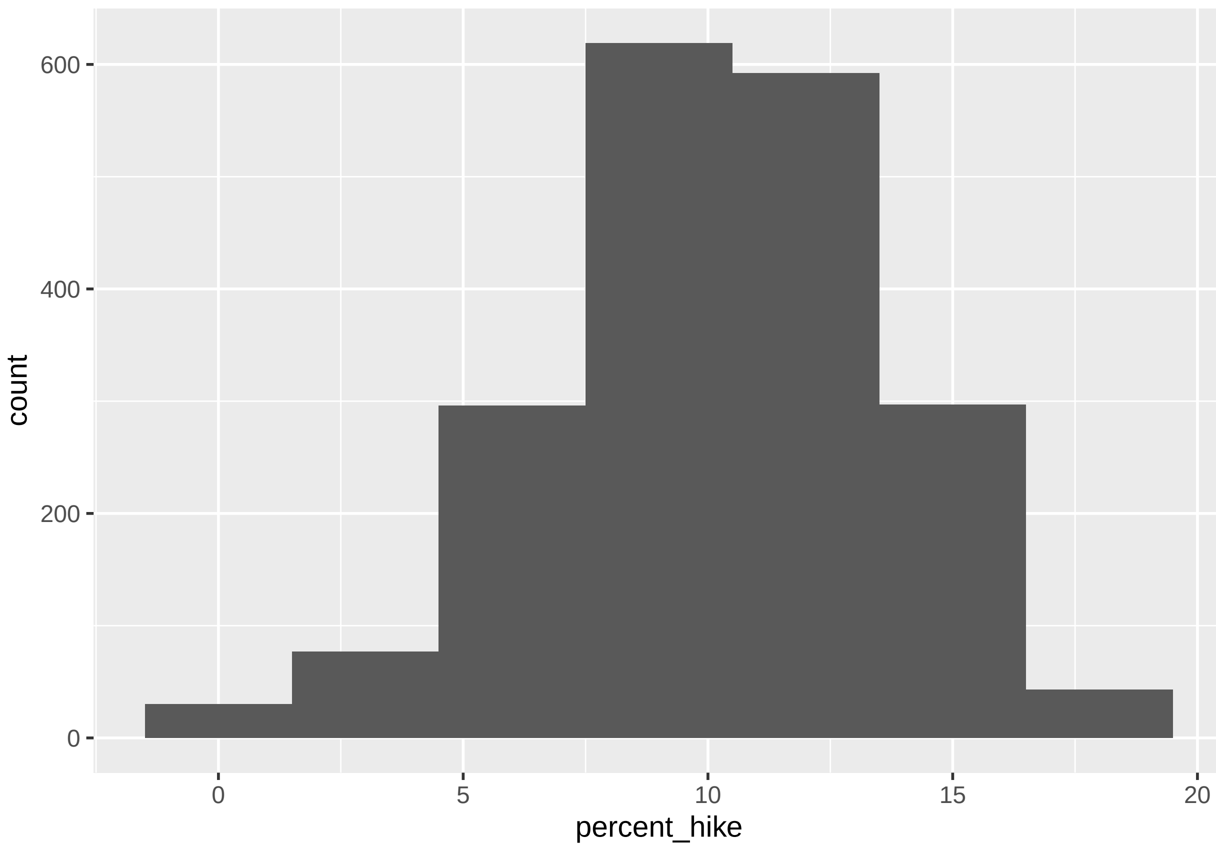
```
## # A tibble: 8 x 3
## # Groups:   risk_bucket [4]
##   level      risk_bucket     n
##   <fct>      <fct>       <int>
## 1 Analyst    no-risk      1089
## 2 Analyst    low-risk      134
```

```
## 3 Analyst    medium-risk    32
## 4 Analyst    high-risk       4
## 5 Specialist no-risk       272
## 6 Specialist low-risk       21
## 7 Specialist medium-risk     4
## 8 Specialist high-risk       1
```

ROI: retun on investment

ROI = Program Benifits / Program Cost

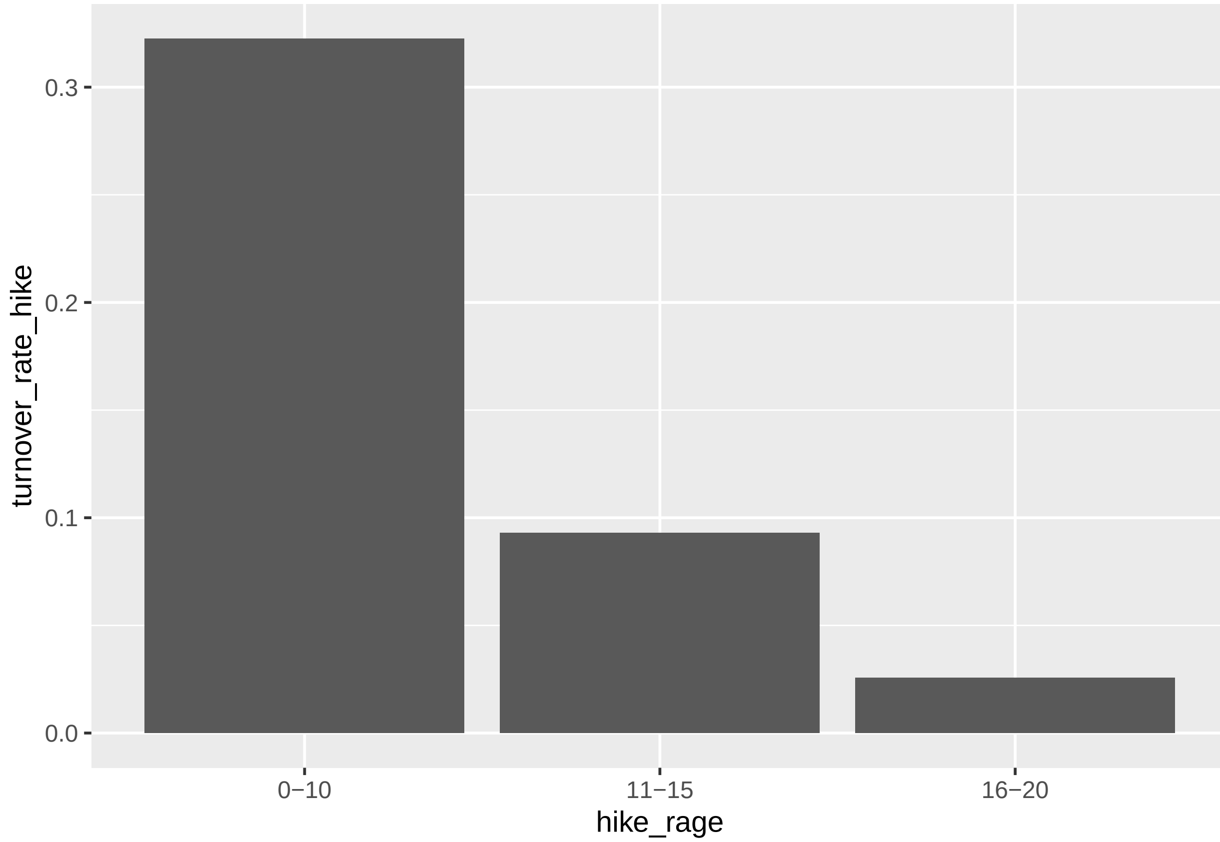
```
emp_final%>%
  ggplot(aes(percent_hike))+geom_histogram(binwidth = 3) #plot histogram of percent hike
```




```
emp_hike_range<- emp_final%>%
  filter(level == "Analyst")%>%
  mutate(hike_rage = cut(percent_hike, breaks = c(0,10,15,20),
                         include.lowest = TRUE,
                         labels = c("0-10","11-15","16-20"))) #create salary hike_rage of analyst level
df_hike<-emp_hike_range%>%
  group_by(hike_rage)%>%
  summarise(turnover_rate_hike = mean(turnover)) # calculate the turnover rate for each salary hike rag
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
df_hike%>%
  ggplot(aes(hike_rage, turnover_rate_hike))+geom_col()
```




```
emp_final%>%
  filter(level == "Analyst")%>%
  count(median_compensation) # after filter we know median_compensation of analyst is 51840
```

```
## # A tibble: 1 x 3
## # Groups:   level [1]
##   level   median_compensation     n
##   <fct>                 <dbl> <int>
## 1 Analyst               51840  1604
```

```
emp_final%>%
  filter(level=="Analyst")%>%
  select(compensation)%>%
  arrange(compensation)%>%
  head()#calculate the minium salary to analyst
```

```
## Adding missing grouping variables: `level`
```

```
## # A tibble: 6 x 2
```

```
## # Groups:   level [1]
##   level   compensation
##   <fct>          <int>
## 1 Analyst        32148
## 2 Analyst        32304
## 3 Analyst        33696
## 4 Analyst        33768
## 5 Analyst        33768
## 6 Analyst        33900
```

```
extra_cost<- 51840 * 0.05 ; extra_cost #increase the salary 5%
```

```
## [1] 2592
```

```
savings <- 40000*0.17 ; savings #assuming the analyst left then hire other one and traning cost
```

```
## [1] 6800
```

```
ROI<-(savings / extra_cost)*100
cat(paste0("The return on investment is ", round(ROI), "%!"))
```

```
## The return on investment is 262%!
```