CaseStudy2

Yucheol Shin

8/7/2020

Data Prepartion

```
caseStudy2 = read.csv("data/CaseStudy2-data.csv", header = TRUE)

caseStudy2$Attrition = as.factor(caseStudy2$Attrition)
caseStudy2$BusinessTravel = as.factor(caseStudy2$BusinessTravel)
caseStudy2$Department = as.factor(caseStudy2$Department)
caseStudy2$EducationField = as.factor(caseStudy2$EducationField)
caseStudy2$Gender = as.factor(caseStudy2$Gender)
caseStudy2$JobRole = as.factor(caseStudy2$JobRole)
caseStudy2$MaritalStatus = as.factor(caseStudy2$MaritalStatus)
caseStudy2$Over18 = as.factor(caseStudy2$Over18)
caseStudy2$OverTime = as.factor(caseStudy2$OverTime)
caseStudy2$EnvironmentSatisfaction  = as.factor(caseStudy2$EnvironmentSatisfaction )
caseStudy2$JobLevel = as.factor(caseStudy2$JobLevel)
caseStudy2$JobSatisfaction = as.factor(caseStudy2$JobSatisfaction)
caseStudy2$PerformanceRating = as.factor(caseStudy2$PerformanceRating)
caseStudy2$RelationshipSatisfaction = as.factor(caseStudy2$RelationshipSatisfaction)
caseStudy2$StockOptionLevel = as.factor(caseStudy2$StockOptionLevel)
caseStudy2$WorkLifeBalance = as.factor(caseStudy2$WorkLifeBalance)
caseStudy2$Attrition = ifelse(caseStudy2$Attrition=="No", 0, 1)
caseStudy2$Attrition = as.factor(caseStudy2$Attrition)
caseStudy2$OverTime = ifelse(caseStudy2$OverTime=="No", 0, 1)
caseStudy2$OverTime = as.factor(caseStudy2$OverTime)
str(caseStudy2)
```

```
## 'data.frame':    870 obs. of  36 variables:
##  $ ID                      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age                     : int  32 40 35 32 24 27 41 37 34 34 ...
##  $ Attrition               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 3 2 3 2 2 3 3
##  $ DailyRate               : int  117 1308 200 801 567 294 1283 309 1333 653 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 3 2 2 2 3 3 2 ...
##  $ DistanceFromHome        : int  13 14 18 1 2 10 5 10 10 10 ...
##  $ Education               : int  4 3 2 4 1 2 5 4 4 4 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 4 2 3 6 2 4 2 2 6 ...
##  $ EmployeeCount           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : int  859 1128 1412 2016 1646 733 1448 1105 1055 1597 ...
##  $ EnvironmentSatisfaction : Factor w/ 4 levels "1","2","3","4": 2 3 3 3 1 4 2 4 3 4 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 2 2 1 1 2 ...
##  $ HourlyRate              : int  73 44 60 48 32 32 90 88 87 92 ...
```

```
##  $ JobInvolvement          : int  3 2 3 3 3 3 4 2 3 2 ...
##  $ JobLevel                : Factor w/ 5 levels "1","2","3","4",..: 2 5 3 3 1 3 1 2 1 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 6 5 8 7 5 7 8 9 1
##  $ JobSatisfaction         : Factor w/ 4 levels "1","2","3","4": 4 3 4 4 4 1 3 4 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 1 3 3 2 3 1 2 1 2 2 ...
##  $ MonthlyIncome           : int  4403 19626 9362 10422 3760 8793 2127 6694 2220 5063 ...
##  $ MonthlyRate             : int  9250 17544 19944 24032 17218 4809 5561 24223 18410 15332 ...
##  $ NumCompaniesWorked      : int  2 1 2 1 1 1 2 2 1 1 ...
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 2 2 1 ...
##  $ PercentSalaryHike       : int  11 14 11 19 13 21 12 14 19 14 ...
##  $ PerformanceRating       : Factor w/ 2 levels "3","4": 1 1 1 1 1 2 1 1 1 1 ...
##  $ RelationshipSatisfaction: Factor w/ 4 levels "1","2","3","4": 3 1 3 3 3 3 1 3 4 2 ...
##  $ StandardHours           : int  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel        : Factor w/ 4 levels "0","1","2","3": 2 1 1 3 1 3 1 4 2 2 ...
##  $ TotalWorkingYears       : int  8 21 10 14 6 9 7 8 1 8 ...
##  $ TrainingTimesLastYear   : int  3 2 2 3 2 4 5 5 2 3 ...
##  $ WorkLifeBalance         : Factor w/ 4 levels "1","2","3","4": 2 4 3 3 3 2 2 3 3 2 ...
##  $ YearsAtCompany          : int  5 20 2 14 6 9 4 1 1 8 ...
##  $ YearsInCurrentRole      : int  2 7 2 10 3 7 2 0 1 2 ...
##  $ YearsSinceLastPromotion : int  0 4 2 5 1 1 0 0 0 7 ...
##  $ YearsWithCurrManager    : int  3 9 2 7 3 7 3 0 0 7 ...
```

Missing Data

```
sapply(caseStudy2, function(x) sum(is.na(x)))
```

```
##                       ID                      Age                Attrition
##                        0                        0                        0
##           BusinessTravel                DailyRate               Department
##                        0                        0                        0
##         DistanceFromHome                Education           EducationField
##                        0                        0                        0
##            EmployeeCount           EmployeeNumber  EnvironmentSatisfaction
##                        0                        0                        0
##                   Gender               HourlyRate           JobInvolvement
##                        0                        0                        0
##                 JobLevel                  JobRole          JobSatisfaction
##                        0                        0                        0
##            MaritalStatus            MonthlyIncome              MonthlyRate
##                        0                        0                        0
##       NumCompaniesWorked                   Over18                 OverTime
##                        0                        0                        0
##        PercentSalaryHike        PerformanceRating RelationshipSatisfaction
##                        0                        0                        0
##            StandardHours         StockOptionLevel        TotalWorkingYears
##                        0                        0                        0
##    TrainingTimesLastYear          WorkLifeBalance           YearsAtCompany
##                        0                        0                        0
##       YearsInCurrentRole  YearsSinceLastPromotion     YearsWithCurrManager
##                        0                        0                        0
```

There is no missing data.

EDA Numeric Summary

```
#Overtime is factor of 0 and 1. So when we make it to numeric, 0 becomes 1 and 1 becomes 2. Thus we min
caseStudy2 %>% group_by(Attrition) %>%
  summarize(
    Mean_Income = mean(MonthlyIncome),
    Mean_Years = median(YearsAtCompany),
    Mean_OverTime = mean(as.numeric(OverTime) - 1),
    Mean_Job_Satisfication = mean(as.numeric(JobSatisfaction)),
    count = n())
```

```
## # A tibble: 2 x 6
##   Attrition Mean_Income Mean_Years Mean_OverTime Mean_Job_Satisfication count
##   <fct>           <dbl>      <dbl>         <dbl>                  <dbl> <int>
## 1 0               6702           6         0.236                   2.76   730
## 2 1               4765.          3         0.571                   2.44   140
```

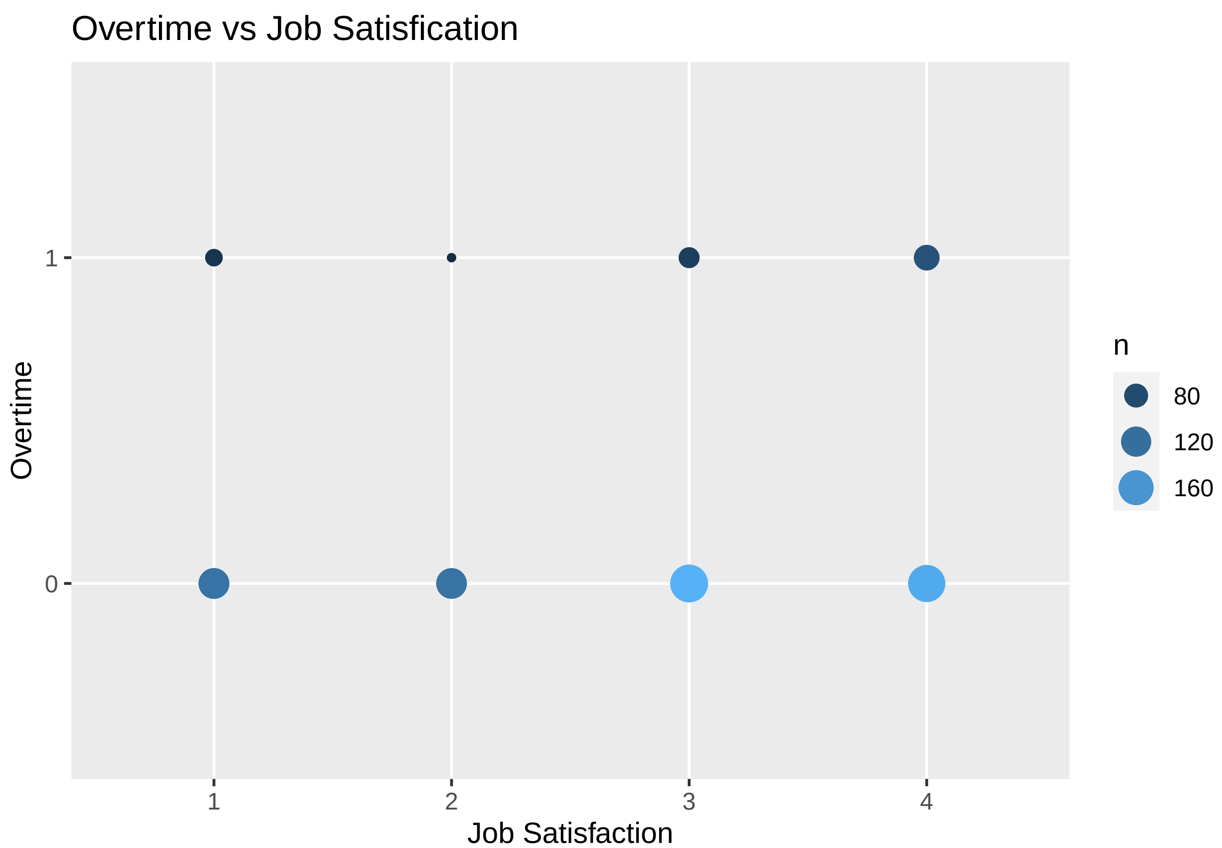
```
caseStudy2 %>% group_by(JobSatisfaction) %>%
  summarize(
    Mean_Income = mean(MonthlyIncome),
    Mean_Years = median(YearsAtCompany),
    Mean_OverTime = mean(as.numeric(OverTime) - 1),
    Mean_Attrition = mean(as.numeric(Attrition) - 1),
    count = n())
```

```
## # A tibble: 4 x 6
##   JobSatisfaction Mean_Income Mean_Years Mean_OverTime Mean_Attrition count
##   <fct>                 <dbl>      <dbl>         <dbl>          <dbl> <int>
## 1 1                     6698.          5         0.302          0.212   179
## 2 2                     6680.          5         0.253          0.187   166
## 3 3                     6291.          5         0.260          0.169   254
## 4 4                     6102.          6         0.332          0.103   271
```

EDA Graph

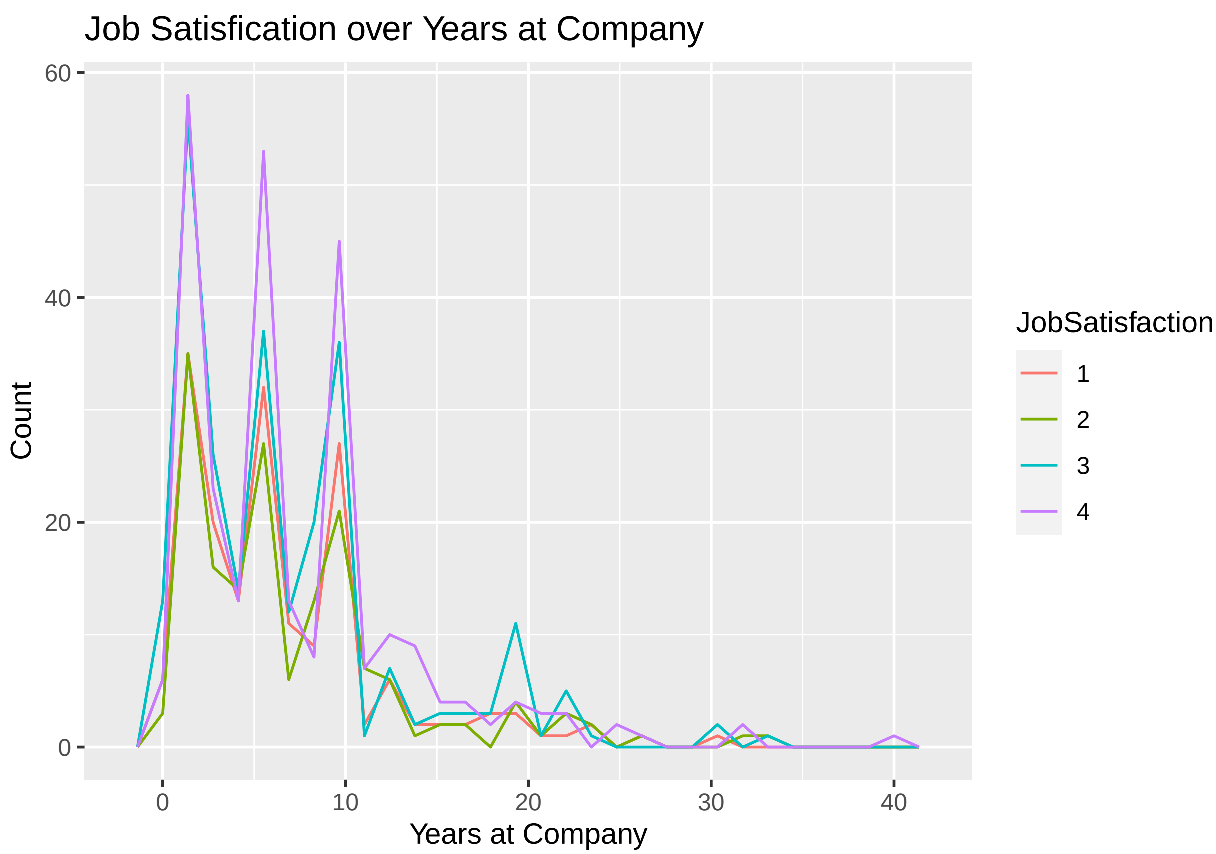
```
caseStudy2 %>% ggplot(aes(JobSatisfaction, JobRole)) + geom_count(aes(color = ..n.., size = ..n..)) + g
  labs(y="Job Role",
       x="Job Satisfaction",
       title="Job Role vs Satisfication")
```


```
caseStudy2 %>% ggplot(aes(JobSatisfaction, OverTime)) + geom_count(aes(color = ..n.., size = ..n..)) + g
  labs(y="Overtime",
       x="Job Satisfaction",
       title="Overtime vs Job Satisfication")
```

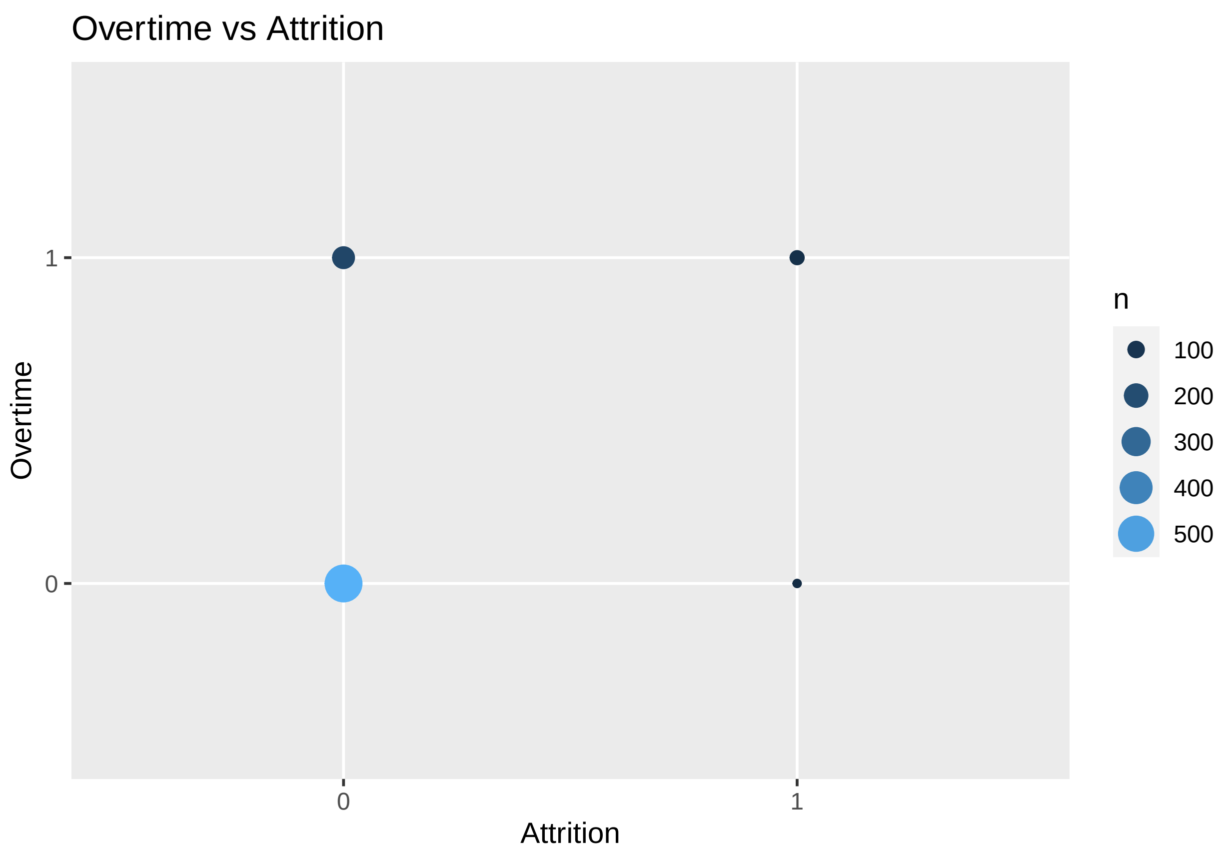



```r
caseStudy2 %>% ggplot(aes(YearsAtCompany, color=JobSatisfaction)) + geom_freqpoly()+
  labs(y="Count",
       x="Years at Company",
       title="Job Satisfication over Years at Company")
```

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

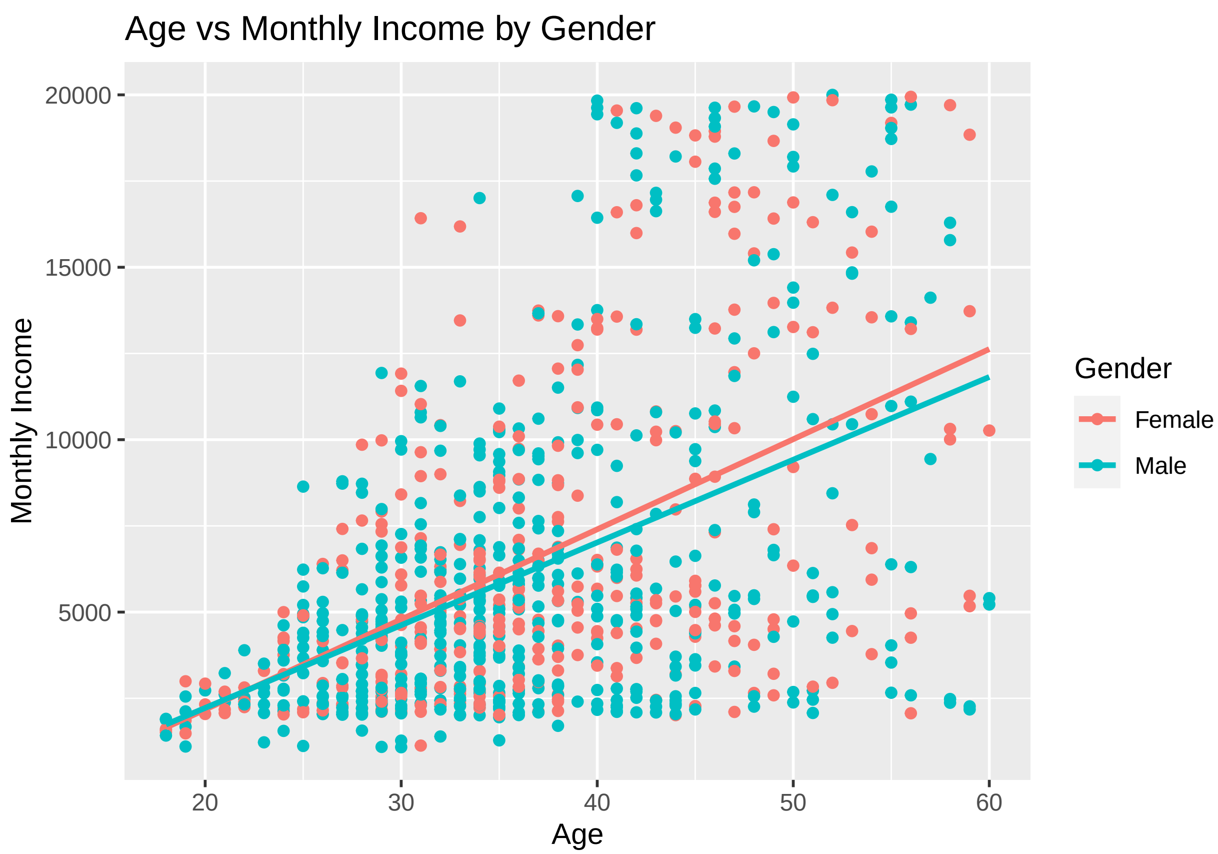



```
caseStudy2 %>% ggplot(aes(Attrition, OverTime)) + geom_count(aes(color = ..n.., size = ..n..)) + guides
  labs(y="Overtime",
       x="Attrition",
       title="Overtime vs Attrition")
```



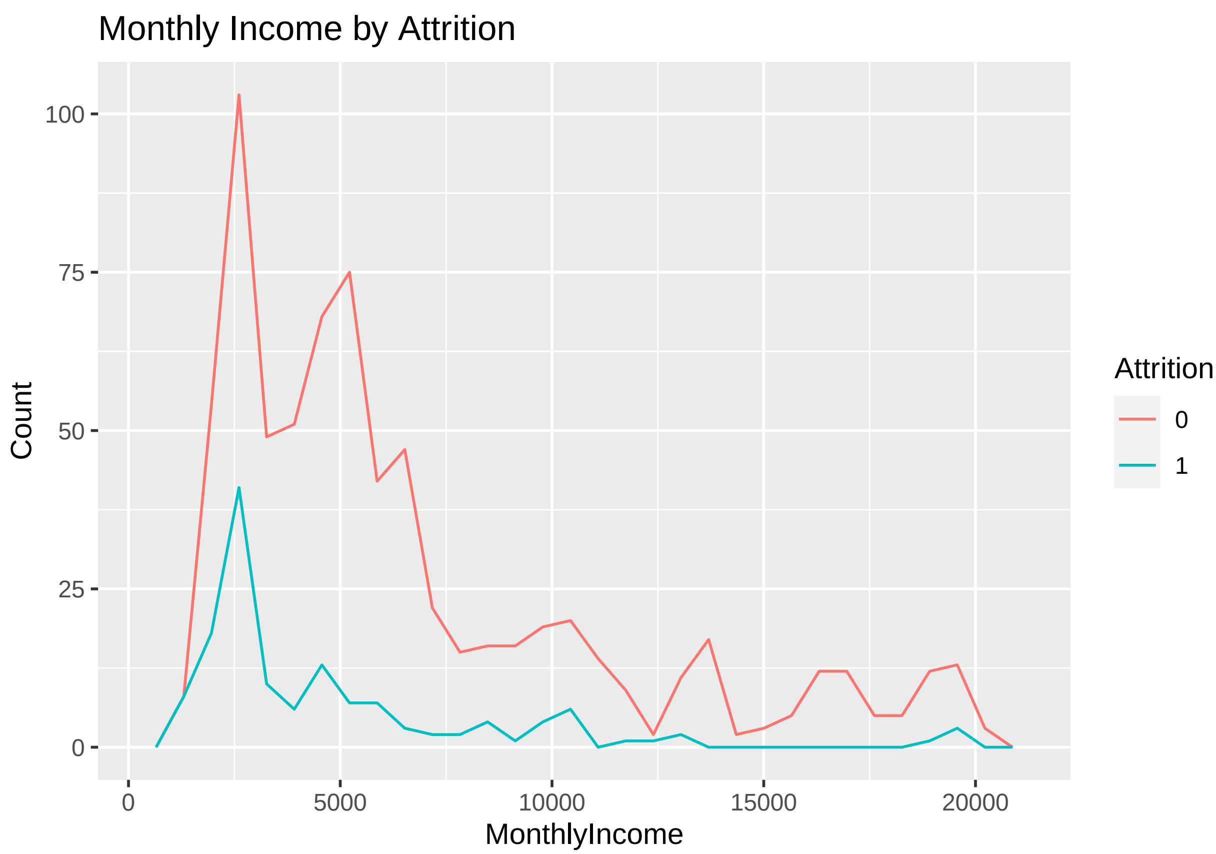

```
caseStudy2 %>% ggplot(aes(Age, MonthlyIncome, color=Gender)) + geom_point() + geom_smooth(method="lm", s
  labs(y="Monthly Income",
       x="Age",
       title="Age vs Monthly Income by Gender")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
caseStudy2 %>% ggplot(aes(MonthlyIncome, color=Attrition)) + geom_freqpoly()+
  labs(y="Count",
       x="MonthlyIncome",
       title="Monthly Income by Attrition")
```

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



1 WorkLifeBalance 2 NumCompaniesWorked 3 OverTimeYes 4 JobSatisfaction Attrition MonthlyIncome Gender
YearsAtCompany

Clean Constant feature

```
# count number of unqiue values in column (1 is row, 2 is column)
apply(caseStudy2, 2, function(x) length(unique(x)))
```

```
##                       ID                      Age                Attrition
##                      870                       43                        2
##           BusinessTravel                DailyRate               Department
##                        3                      627                        3
##         DistanceFromHome                Education           EducationField
##                       29                        5                        6
##            EmployeeCount           EmployeeNumber  EnvironmentSatisfaction
##                        1                      870                        4
##                   Gender               HourlyRate           JobInvolvement
##                        2                       71                        4
##                 JobLevel                  JobRole          JobSatisfaction
##                        5                        9                        4
##            MaritalStatus            MonthlyIncome              MonthlyRate
##                        3                      826                      852
##       NumCompaniesWorked                   Over18                 OverTime
##                       10                        1                        2
##        PercentSalaryHike        PerformanceRating RelationshipSatisfaction
##                       15                        2                        4
```

```
##           StandardHours         StockOptionLevel        TotalWorkingYears
##                       1                        4                       39
##    TrainingTimesLastYear           WorkLifeBalance           YearsAtCompany
##                       7                        4                       32
##       YearsInCurrentRole  YearsSinceLastPromotion     YearsWithCurrManager
##                      19                       16                       17
```

```
cleanData <- subset(caseStudy2, select = -c(EmployeeCount, Over18, StandardHours))
```

We are dropping EmployeeCount, Over18, StandardHours features as there is only one unique value.

Multicollinearity for continous variables We are using pearson correlation to find correlation between numeric data. Data with 0.5 < perason correlation has strong correlation.

```
numericColumns <- unlist(lapply(cleanData, is.numeric))
numericData = cleanData[ , numericColumns]
correlation = cor(numericData, method = c("pearson"))

round_df <- function(x, digits) {
    numeric_columns <- sapply(x, mode) == 'numeric'
    x[numeric_columns] <-  round(x[numeric_columns], digits)
    x
}

correlation = round_df(correlation, 4)
```

Strong Correlation TotalWorkingYears - Age TotalWorkingYears - MonthlyIncome TotalWorkingYears - YearsAtCompany YearsInCurrentRole - YearsAtCompany YearsInCurrentRole - YearsSinceLastPromotion YearsInCurrentRole - YearsWithCurrManager YearsAtCompany - YearsSinceLastPromotion YearsAtCompany - YearsWithCurrManager YearsWithCurrManager - YearsSinceLastPromotion

Chi-squared Test Null = variables are independent Alternative = there is a relationship

```
tbl = table(cleanData$TotalWorkingYears, cleanData$Attrition)
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 74.121, df = 38, p-value = 0.0004072
```

```
tbl = table(cleanData$YearsAtCompany, cleanData$Attrition)
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 69.893, df = 31, p-value = 7.894e-05
```

```r
tbl = table(cleanData$YearsSinceLastPromotion, cleanData$Attrition)
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 21.239, df = 15, p-value = 0.1294
```

```r
tbl = table(cleanData$YearsWithCurrManager, cleanData$Attrition)
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 43.229, df = 16, p-value = 0.0002581
```

```r
tbl = table(cleanData$YearsInCurrentRole, cleanData$Attrition)
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 43.028, df = 18, p-value = 0.0007929
```

```r
cleanData <- subset(cleanData, select = -c(ID, TotalWorkingYears, YearsSinceLastPromotion, YearsWithCur
```

YearsSinceLastPromotion and YearsAtCompany has strong correlation. Since chi-squared test with YearsAtCompany reject the null, we can assume YearsAtCompany is more relate to Attrition than YearsSinceLastPromotion. Thus we choose YearsAtCompany for our model. Since YearsAtCompany has lower p value and has strong correlation with other years variables, we choose YearsAtCompany for our variable. Thus from multicorrlinearity, we got Age, MontlyIncome, YearsAtCompany. Finally, we drop TotalWorkingYears, YearsSinceLastPromotion, YearsWithCurrManager, and YearsInCurrentRole. We also drop ID as that is not feature.

Stepwise Feature Selection

```r
model <- glm(Attrition ~., data = cleanData, family = binomial)
stepwise <- model %>% stepAIC(trace = FALSE)
summary(stepwise)
```

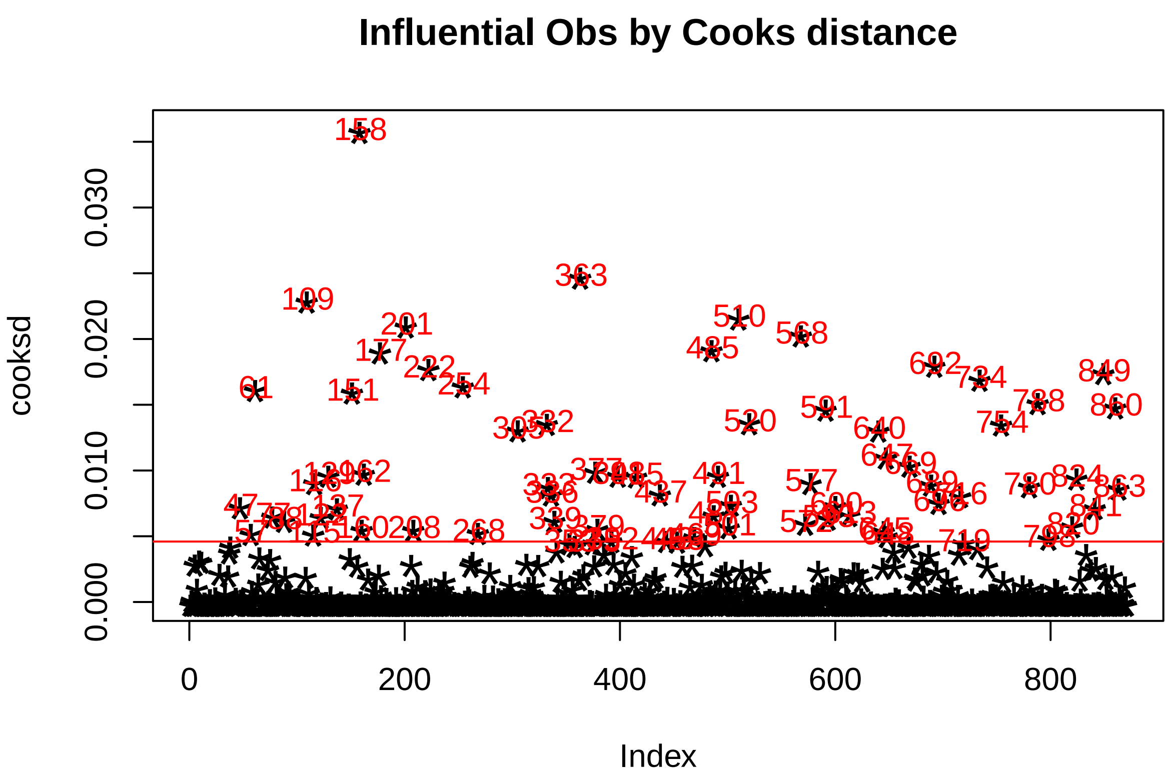
```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + EnvironmentSatisfaction + HourlyRate +
##     JobInvolvement + JobLevel + JobSatisfaction + MaritalStatus +
##     MonthlyIncome + NumCompaniesWorked + OverTime + RelationshipSatisfaction +
##     StockOptionLevel + TrainingTimesLastYear + WorkLifeBalance,
##     family = binomial, data = cleanData)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0649  -0.4306  -0.1952  -0.0629   3.5664
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      4.1202225  1.3353246   3.086 0.002032 **
## Age                             -0.0492943  0.0170466  -2.892 0.003831 **
## BusinessTravelTravel_Frequently  1.5833244  0.5040234   3.141 0.001682 **
## BusinessTravelTravel_Rarely      0.6210157  0.4538110   1.368 0.171173
## DepartmentResearch & Development -0.2626898  0.6259558  -0.420 0.674732
## DepartmentSales                  1.3716289  0.6625081   2.070 0.038419 *
## DistanceFromHome                 0.0651053  0.0154843   4.205 2.62e-05 ***
## EnvironmentSatisfaction2        -1.2071386  0.3901604  -3.094 0.001975 **
## EnvironmentSatisfaction3        -1.0662878  0.3516253  -3.032 0.002426 **
## EnvironmentSatisfaction4        -0.8762661  0.3486895  -2.513 0.011970 *
## HourlyRate                       0.0110300  0.0063034   1.750 0.080145 .
## JobInvolvement                  -0.8678779  0.1723596  -5.035 4.77e-07 ***
## JobLevel2                       -1.8029732  0.4616386  -3.906 9.40e-05 ***
## JobLevel3                       -0.2281937  0.8485820  -0.269 0.787998
## JobLevel4                       -0.3622194  1.5335482  -0.236 0.813280
## JobLevel5                        2.3106841  1.9670956   1.175 0.240128
## JobSatisfaction2                -0.3885410  0.3745585  -1.037 0.299582
## JobSatisfaction3                -0.2978990  0.3338599  -0.892 0.372239
## JobSatisfaction4                -1.3064754  0.3698582  -3.532 0.000412 ***
## MaritalStatusMarried             0.8676151  0.4300445   2.018 0.043643 *
## MaritalStatusSingle              0.8004723  0.5423559   1.476 0.139966
## MonthlyIncome                   -0.0001830  0.0001133  -1.615 0.106259
## NumCompaniesWorked               0.1802377  0.0492492   3.660 0.000253 ***
## OverTime1                        2.1053652  0.2685831   7.839 4.55e-15 ***
## RelationshipSatisfaction2       -0.8682301  0.4085931  -2.125 0.033593 *
## RelationshipSatisfaction3       -0.8395659  0.3488345  -2.407 0.016094 *
## RelationshipSatisfaction4       -0.7969704  0.3369416  -2.365 0.018015 *
## StockOptionLevel1               -1.4821999  0.3932960  -3.769 0.000164 ***
## StockOptionLevel2               -1.7741646  0.7264714  -2.442 0.014599 *
## StockOptionLevel3                0.1347374  0.5433672   0.248 0.804160
## TrainingTimesLastYear           -0.2584145  0.1034204  -2.499 0.012466 *
## WorkLifeBalance2                -1.4372296  0.4925135  -2.918 0.003521 **
## WorkLifeBalance3                -1.7925027  0.4590947  -3.904 9.44e-05 ***
## WorkLifeBalance4                -2.1513158  0.6063368  -3.548 0.000388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 767.67  on 869  degrees of freedom
## Residual deviance: 452.28  on 836  degrees of freedom
## AIC: 520.28
##
## Number of Fisher Scoring iterations: 6
```

Cook's Distance for OutLiers

```
cooksd <- cooks.distance(stepwise)

sample_size <- nrow(cleanData)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4/sample_size, col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample_size, names(cooksd),""), col="red")
```




```
influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])
cleanData <- cleanData[-influential, ]

model2 <- glm(Attrition ~., data = cleanData, family = binomial)
stepwise2 <- model2 %>% stepAIC(trace = FALSE)
summary(stepwise2)
```

```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + DailyRate +
##     Department + DistanceFromHome + Gender + HourlyRate + JobInvolvement +
##     JobLevel + JobSatisfaction + NumCompaniesWorked + OverTime +
##     PercentSalaryHike + RelationshipSatisfaction + StockOptionLevel +
##     TrainingTimesLastYear + WorkLifeBalance + YearsAtCompany,
##     family = binomial, data = cleanData)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -0.004790   0.000000   0.000000   0.000000   0.004938
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       1.049e+04  2.810e+06   0.004    0.997
## Age                              -1.525e+02  2.388e+03  -0.064    0.949
## BusinessTravelTravel_Frequently   6.361e+03  7.032e+04   0.090    0.928
## BusinessTravelTravel_Rarely       1.722e+03  2.555e+04   0.067    0.946
## DailyRate                        -9.196e-01  1.520e+01  -0.060    0.952
## DepartmentResearch & Development -5.874e+02  2.805e+06   0.000    1.000
## DepartmentSales                   5.470e+03  2.805e+06   0.002    0.998
## DistanceFromHome                  3.026e+02  2.521e+03   0.120    0.904
## GenderMale                        5.113e+02  1.953e+04   0.026    0.979
## HourlyRate                        4.882e+01  1.190e+03   0.041    0.967
## JobInvolvement                   -3.263e+03  3.171e+04  -0.103    0.918
## JobLevel2                        -8.104e+03  7.400e+04  -0.110    0.913
## JobLevel3                        -4.855e+03  4.080e+04  -0.119    0.905
## JobLevel4                        -5.305e+03  2.522e+05  -0.021    0.983
## JobLevel5                         1.001e+02  2.804e+06   0.000    1.000
## JobSatisfaction2                 -4.881e+02  2.288e+04  -0.021    0.983
## JobSatisfaction3                  5.976e+02  2.144e+04   0.028    0.978
## JobSatisfaction4                 -5.149e+03  4.105e+04  -0.125    0.900
## NumCompaniesWorked                6.992e+02  5.137e+03   0.136    0.892
## OverTime1                         8.375e+03  6.467e+04   0.130    0.897
## PercentSalaryHike                -3.850e+01  1.869e+03  -0.021    0.984
## RelationshipSatisfaction2        -5.026e+03  5.150e+04  -0.098    0.922
## RelationshipSatisfaction3        -3.226e+03  3.290e+04  -0.098    0.922
## RelationshipSatisfaction4        -5.045e+03  5.598e+04  -0.090    0.928
## StockOptionLevel1                -5.313e+03  4.268e+04  -0.124    0.901
## StockOptionLevel2                -5.762e+03  2.454e+05  -0.023    0.981
## StockOptionLevel3                 4.245e+02  1.535e+04   0.028    0.978
## TrainingTimesLastYear            -4.899e+02  2.292e+04  -0.021    0.983
## WorkLifeBalance2                 -4.781e+03  4.385e+04  -0.109    0.913
## WorkLifeBalance3                 -6.600e+03  4.250e+04  -0.155    0.877
## WorkLifeBalance4                 -6.589e+03  4.430e+04  -0.149    0.882
## YearsAtCompany                   -6.422e+01  2.076e+03  -0.031    0.975
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.2822e+02  on 796  degrees of freedom
## Residual deviance: 2.7625e-04  on 765  degrees of freedom
## AIC: 64
##
## Number of Fisher Scoring iterations: 25
```

From stepwise model we found Age + BusinessTravel + DailyRate + Department + DistanceFromHome + Gender + HourlyRate + JobInvolvement + JobLevel + JobSatisfaction + NumCompaniesWorked + OverTime + PercentSalaryHike + RelationshipSatisfaction + StockOptionLevel + TrainingTimesLastYear + WorkLifeBalance + YearsAtCompany are the features we should use to build the model

Feature Importance

```
importance <- varImp(stepwise2, scale=FALSE)
head(arrange(importance,desc(Overall)), n = 5)
```

```
##     Overall
## 1 0.1552940
## 2 0.1487197
## 3 0.1361088
## 4 0.1295050
## 5 0.1254394
```

```
head(importance)
```

```
##                                       Overall
## Age                              0.0638393637
## BusinessTravelTravel_Frequently  0.0904617135
## BusinessTravelTravel_Rarely      0.0673745217
## DailyRate                        0.0604838440
## DepartmentResearch & Development 0.0002094097
## DepartmentSales                  0.0019500908
```

Top 5 important features 1 WorkLifeBalance 2 NumCompaniesWorked 3 OverTimeYes 4 JobSatisfaction 5 StockOptionLevel

Prediction and Confusion Matrix

```
set.seed(4)
splitPerc = .70
trainIndices = sample(1:dim(cleanData)[1],round(splitPerc * dim(cleanData)[1]))
train = cleanData[trainIndices,]
test = cleanData[-trainIndices,]


trainFit <- glm(Attrition ~., data = train, family = binomial)
trainModel <- trainFit %>% stepAIC(trace = FALSE)
pred <- predict(trainModel,test)
pred <- as.factor(as.numeric(pred>0.5))
confusionMatrix(pred, reference = test$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 206   4
##          1   8  21
##
##                Accuracy : 0.9498
##                  95% CI : (0.9139, 0.9738)
##     No Information Rate : 0.8954
##     P-Value [Acc > NIR] : 0.002091
##
##                   Kappa : 0.7497
##
##  Mcnemar's Test P-Value : 0.386476
##
##             Sensitivity : 0.9626
##             Specificity : 0.8400
```

```
##          Pos Pred Value : 0.9810
##          Neg Pred Value : 0.7241
##              Prevalence : 0.8954
##          Detection Rate : 0.8619
##    Detection Prevalence : 0.8787
##       Balanced Accuracy : 0.9013
##
##        'Positive' Class : 0
##
```

Use original data to get the accuracy

```r
trainFit <- glm(formula = Attrition ~ Age + BusinessTravel + DailyRate +
    Department + DistanceFromHome + Gender + HourlyRate + JobInvolvement +
    JobLevel + JobSatisfaction + NumCompaniesWorked + OverTime +
    PercentSalaryHike + RelationshipSatisfaction + StockOptionLevel +
    TrainingTimesLastYear + WorkLifeBalance + YearsAtCompany,
    family = binomial, data = train)
trainModel <- trainFit %>% stepAIC(trace = FALSE)

pred <- predict(trainModel,caseStudy2)
pred <- as.factor(as.numeric(pred>0.5))
confusionMatrix(pred, reference = caseStudy2$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 710  51
##          1  20  89
##
##                Accuracy : 0.9184
##                  95% CI : (0.8982, 0.9357)
##     No Information Rate : 0.8391
##     P-Value [Acc > NIR] : 3.583e-12
##
##                   Kappa : 0.6681
##
##  Mcnemar's Test P-Value : 0.0003704
##
##             Sensitivity : 0.9726
##             Specificity : 0.6357
##          Pos Pred Value : 0.9330
##          Neg Pred Value : 0.8165
##              Prevalence : 0.8391
##          Detection Rate : 0.8161
##    Detection Prevalence : 0.8747
##       Balanced Accuracy : 0.8042
##
##        'Positive' Class : 0
##
```

KNN

```r
set.seed(4)
splitPerc = .70
knnData <- caseStudy2
knnData[1:36] = sapply(knnData[,1:36], as.numeric)

trainIndices = sample(1:dim(knnData)[1],round(splitPerc * dim(knnData)[1]))
train = knnData[trainIndices,]
test = knnData[-trainIndices,]

knnModel = knn(train, test, train$Attrition, prob = TRUE, k = 5)
table(test$Attrition,knnModel)
```

```
##    knnModel
##       1   2
##   1 224   3
##   2  33   1
```

```r
CM = confusionMatrix(table(test$Attrition ,knnModel))
CM
```

```
## Confusion Matrix and Statistics
##
##    knnModel
##       1   2
##   1 224   3
##   2  33   1
##
##                Accuracy : 0.8621
##                  95% CI : (0.8142, 0.9015)
##     No Information Rate : 0.9847
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0259
##
##  Mcnemar's Test P-Value : 1.343e-06
##
##             Sensitivity : 0.87160
##             Specificity : 0.25000
##          Pos Pred Value : 0.98678
##          Neg Pred Value : 0.02941
##              Prevalence : 0.98467
##          Detection Rate : 0.85824
##    Detection Prevalence : 0.86973
##       Balanced Accuracy : 0.56080
##
##        'Positive' Class : 1
##
```

KNN with original Data

```r
set.seed(4)
splitPerc = .70
```

```r
knnData <- caseStudy2
knnData[1:36] = sapply(knnData[,1:36], as.numeric)

knnModel = knn(knnData, knnData, knnData$Attrition, prob = TRUE, k = 5)
table(knnData$Attrition, knnModel)
```

```
##    knnModel
##       1   2
##   1 715  15
##   2 113  27
```

```r
CM = confusionMatrix(table(knnData$Attrition ,knnModel))
CM
```

```
## Confusion Matrix and Statistics
##
##    knnModel
##       1   2
##   1 715  15
##   2 113  27
##
##                Accuracy : 0.8529
##                  95% CI : (0.8276, 0.8758)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2403
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8635
##             Specificity : 0.6429
##          Pos Pred Value : 0.9795
##          Neg Pred Value : 0.1929
##              Prevalence : 0.9517
##          Detection Rate : 0.8218
##    Detection Prevalence : 0.8391
##       Balanced Accuracy : 0.7532
##
##        'Positive' Class : 1
##
```

KNN with feature selection

```r
knnFeatureData <- subset(caseStudy2, select = c(Attrition, Age, BusinessTravel, DailyRate, Department, 
knnFeatureData[1:19] = sapply(knnFeatureData[,1:19], as.numeric)

set.seed(4)
splitPerc = .70

trainIndices = sample(1:dim(knnFeatureData)[1],round(splitPerc * dim(knnFeatureData)[1]))
train = knnFeatureData[trainIndices,]
```

```r
test = knnFeatureData[-trainIndices,]

knnModel = knn(train, test, train$Attrition, prob = TRUE, k = 5)
table(test$Attrition,knnModel)
```

```
##    knnModel
##       1   2
##   1 223   4
##   2  34   0
```

```r
CM = confusionMatrix(table(test$Attrition ,knnModel))
CM
```

```
## Confusion Matrix and Statistics
##
##    knnModel
##       1   2
##   1 223   4
##   2  34   0
##
##                Accuracy : 0.8544
##                  95% CI : (0.8057, 0.8949)
##     No Information Rate : 0.9847
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0282
##
##  Mcnemar's Test P-Value : 2.546e-06
##
##             Sensitivity : 0.8677
##             Specificity : 0.0000
##          Pos Pred Value : 0.9824
##          Neg Pred Value : 0.0000
##              Prevalence : 0.9847
##          Detection Rate : 0.8544
##    Detection Prevalence : 0.8697
##       Balanced Accuracy : 0.4339
##
##        'Positive' Class : 1
##
```

Salary Prediction

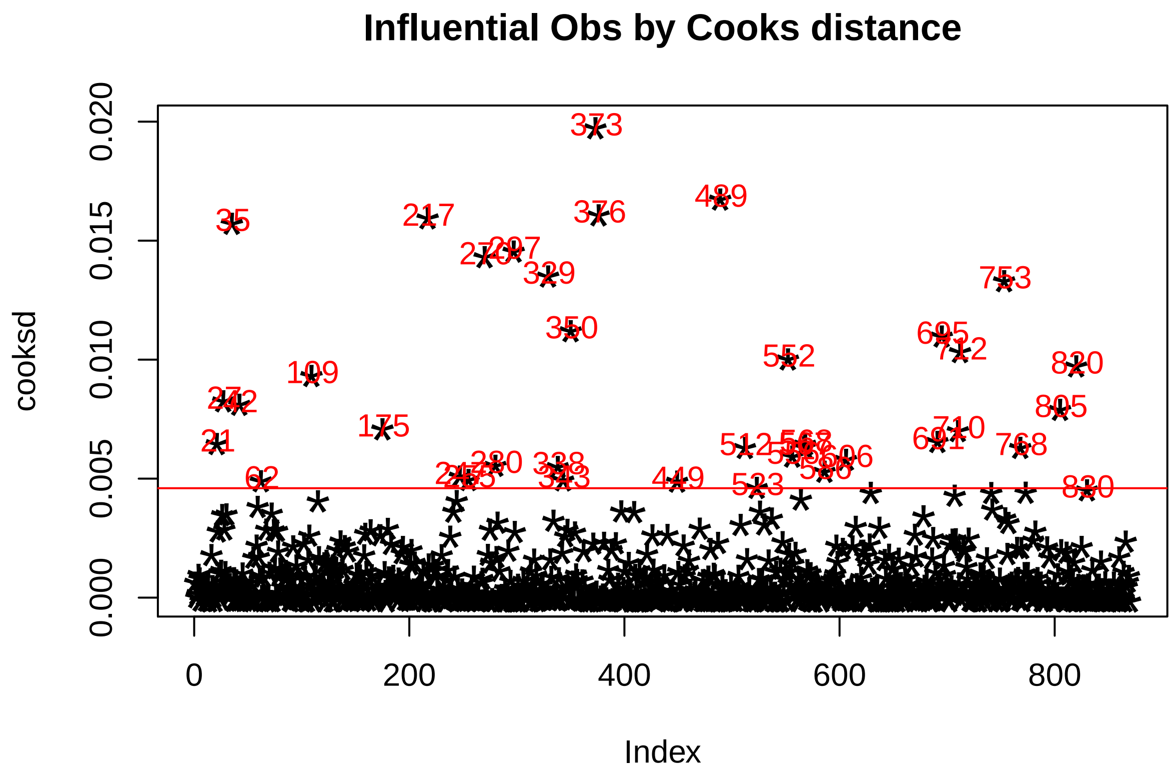
```r
data3 <- subset(caseStudy2, select = -c(ID, EmployeeCount, Over18, StandardHours, TotalWorkingYears, Yea
str(data3)
```

```
## 'data.frame':    870 obs. of  28 variables:
##  $ Age                     : int  32 40 35 32 24 27 41 37 34 34 ...
##  $ Attrition               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 3 2 3 2 2 3 3
##  $ DailyRate               : int  117 1308 200 801 567 294 1283 309 1333 653 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 3 2 2 2 3 3 2 ...
```

```
##  $ DistanceFromHome        : int  13 14 18 1 2 10 5 10 10 10 ...
##  $ Education               : int  4 3 2 4 1 2 5 4 4 4 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 4 2 3 6 2 4 2 2 6 ...
##  $ EmployeeNumber          : int  859 1128 1412 2016 1646 733 1448 1105 1055 1597 ...
##  $ EnvironmentSatisfaction : Factor w/ 4 levels "1","2","3","4": 2 3 3 3 1 4 2 4 3 4 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 2 2 1 1 2 ...
##  $ HourlyRate              : int  73 44 60 48 32 32 90 88 87 92 ...
##  $ JobInvolvement          : int  3 2 3 3 3 3 4 2 3 2 ...
##  $ JobLevel                : Factor w/ 5 levels "1","2","3","4",..: 2 5 3 3 1 3 1 2 1 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 6 5 8 7 5 7 8 9 1
##  $ JobSatisfaction         : Factor w/ 4 levels "1","2","3","4": 4 3 4 4 4 1 3 4 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 1 3 3 2 3 1 2 1 2 2 ...
##  $ MonthlyIncome           : int  4403 19626 9362 10422 3760 8793 2127 6694 2220 5063 ...
##  $ MonthlyRate             : int  9250 17544 19944 24032 17218 4809 5561 24223 18410 15332 ...
##  $ NumCompaniesWorked      : int  2 1 2 1 1 1 2 2 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 2 2 1 ...
##  $ PercentSalaryHike       : int  11 14 11 19 13 21 12 14 19 14 ...
##  $ PerformanceRating       : Factor w/ 2 levels "3","4": 1 1 1 1 1 2 1 1 1 1 ...
##  $ RelationshipSatisfaction: Factor w/ 4 levels "1","2","3","4": 3 1 3 3 3 3 1 3 4 2 ...
##  $ StockOptionLevel        : Factor w/ 4 levels "0","1","2","3": 2 1 1 3 1 3 1 4 2 2 ...
##  $ TrainingTimesLastYear   : int  3 2 2 3 2 4 5 5 2 3 ...
##  $ WorkLifeBalance         : Factor w/ 4 levels "1","2","3","4": 2 4 3 3 3 2 2 3 3 2 ...
##  $ YearsAtCompany          : int  5 20 2 14 6 9 4 1 1 8 ...
```

```r
linearModel <- lm(MonthlyIncome ~., data=data3)  # build linear regression model on full data
stepwiseLinear <- linearModel %>% stepAIC(trace = FALSE)
cooksd <- cooks.distance(stepwiseLinear)

sample_size <- nrow(data3)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4/sample_size, col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample_size, names(cooksd),""), col="red")
```




```r
influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])
data3 <- data3[-influential, ]

linearModel <- lm(MonthlyIncome ~., data=data3)  # build linear regression model on full data
stepwiseLinear <- linearModel %>% stepAIC(trace = FALSE)
summary(stepwiseLinear)
```

```
##
## Call:
## lm(formula = MonthlyIncome ~ Age + Attrition + BusinessTravel +
##     JobLevel + JobRole + NumCompaniesWorked + OverTime + PercentSalaryHike +
##     PerformanceRating + YearsAtCompany, data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2192.6  -577.9   -64.2   549.9  3191.2
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2366.334    286.162   8.269 5.51e-16 ***
## Age                              11.541      3.928   2.938 0.003400 **
## Attrition1                     -168.472     87.274  -1.930 0.053907 .
## BusinessTravelTravel_Frequently  276.005    114.510   2.410 0.016161 *
## BusinessTravelTravel_Rarely      380.245     96.378   3.945 8.66e-05 ***
## JobLevel2                      1959.185    120.728  16.228  < 2e-16 ***
```

```
## JobLevel3                         5512.279    161.041  34.229  < 2e-16 ***
## JobLevel4                         9298.208    221.011  42.071  < 2e-16 ***
## JobLevel5                        11921.487    262.945  45.338  < 2e-16 ***
## JobRoleHuman Resources           -1050.348    222.599  -4.719 2.80e-06 ***
## JobRoleLaboratory Technician      -873.223    156.758  -5.571 3.46e-08 ***
## JobRoleManager                    3156.156    209.620  15.057  < 2e-16 ***
## JobRoleManufacturing Director      -75.236    139.022  -0.541 0.588532
## JobRoleResearch Director          3297.477    187.544  17.582  < 2e-16 ***
## JobRoleResearch Scientist         -798.503    158.260  -5.046 5.59e-07 ***
## JobRoleSales Executive            -131.138    119.607  -1.096 0.273227
## JobRoleSales Representative       -993.390    195.141  -5.091 4.44e-07 ***
## NumCompaniesWorked                  43.203     12.748   3.389 0.000736 ***
## OverTime1                          101.817     67.730   1.503 0.133157
## PercentSalaryHike                   26.436     12.722   2.078 0.038019 *
## PerformanceRating4                -354.138    129.768  -2.729 0.006490 **
## YearsAtCompany                      24.037      6.275   3.831 0.000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 839.8 on 810 degrees of freedom
## Multiple R-squared:  0.9673, Adjusted R-squared:  0.9664
## F-statistic:  1140 on 21 and 810 DF,  p-value: < 2.2e-16
```

```r
importanceIncome <- varImp(stepwiseLinear, scale=FALSE)
head(arrange(importanceIncome,desc(Overall)), n = 5)
```

```
##    Overall
## 1 45.33828
## 2 42.07122
## 3 34.22899
## 4 17.58241
## 5 16.22813
```

```r
head(importanceIncome)
```

```
##                                    Overall
## Age                               2.937734
## Attrition1                        1.930389
## BusinessTravelTravel_Frequently   2.410313
## BusinessTravelTravel_Rarely       3.945352
## JobLevel2                        16.228126
## JobLevel3                        34.228991
```

Prediction and RMSE for Linear Regression

```r
set.seed(4)
splitPerc = .70
trainIndices = sample(1:dim(data3)[1],round(splitPerc * dim(data3)[1]))
train = data3[trainIndices,]
test = data3[-trainIndices,]
```

```r
trainFit <- lm(MonthlyIncome ~., data=train)
trainModel <- trainFit %>% stepAIC(trace = FALSE)

#Predict monthly income
incomePred <- predict(trainModel, test)
head(incomePred)
```

```
##        1        5        7       17       18       19
## 5471.215 2458.770 3033.145 5371.011 5152.543 2773.291
```

```r
#Model Summary
summary (trainModel)
```

```
##
## Call:
## lm(formula = MonthlyIncome ~ Age + BusinessTravel + DailyRate +
##     Gender + JobLevel + JobRole + NumCompaniesWorked + PercentSalaryHike +
##     PerformanceRating + YearsAtCompany, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1878.37  -540.40   -69.66   538.06  3126.15
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2153.0864   349.6626   6.158 1.41e-09 ***
## Age                               12.3033     4.5395   2.710 0.006930 **
## BusinessTravelTravel_Frequently  232.8100   135.6816   1.716 0.086742 .
## BusinessTravelTravel_Rarely      418.8783   116.7227   3.589 0.000362 ***
## DailyRate                          0.1663     0.0872   1.908 0.056940 .
## GenderMale                        98.3148    70.5861   1.393 0.164222
## JobLevel2                       2156.7862   146.4992  14.722  < 2e-16 ***
## JobLevel3                       5589.2095   196.4309  28.454  < 2e-16 ***
## JobLevel4                       9395.5682   266.1947  35.296  < 2e-16 ***
## JobLevel5                      12091.7426   325.7107  37.124  < 2e-16 ***
## JobRoleHuman Resources         -1035.7959   269.7171  -3.840 0.000137 ***
## JobRoleLaboratory Technician    -777.0710   184.9579  -4.201 3.09e-05 ***
## JobRoleManager                  3211.4214   251.6419  12.762  < 2e-16 ***
## JobRoleManufacturing Director   -324.9316   165.4957  -1.963 0.050096 .
## JobRoleResearch Director        3267.3291   229.1539  14.258  < 2e-16 ***
## JobRoleResearch Scientist       -786.2554   189.1537  -4.157 3.73e-05 ***
## JobRoleSales Executive          -199.0814   140.1290  -1.421 0.155960
## JobRoleSales Representative     -976.1543   229.7850  -4.248 2.52e-05 ***
## NumCompaniesWorked                29.8864    14.6989   2.033 0.042499 *
## PercentSalaryHike                 23.5188    15.0010   1.568 0.117489
## PerformanceRating4              -369.6095   152.4621  -2.424 0.015655 *
## YearsAtCompany                    22.3169     7.9667   2.801 0.005266 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 828 on 560 degrees of freedom
## Multiple R-squared:  0.9655, Adjusted R-squared:  0.9642
## F-statistic: 745.4 on 21 and 560 DF,  p-value: < 2.2e-16
```

```
RSS <- c(crossprod(trainModel$residuals))
#Mean squared error:
MSE <- RSS / length(trainModel$residuals)
#Root MSE:
RMSE <- sqrt(MSE)
RMSE
```

```
## [1] 812.1544
```

Predict Attrition with No Attrition Data

```
attritionPredData = read.csv("data/CaseStudy2CompSet No Attrition.csv", header = TRUE)
attritionPredData$OverTime = ifelse(attritionPredData$OverTime=="No", 0, 1)
attritionPredData$OverTime = as.factor(attritionPredData$OverTime)
attritionPredData$EnvironmentSatisfaction  = as.factor(attritionPredData$EnvironmentSatisfaction )
attritionPredData$JobLevel = as.factor(attritionPredData$JobLevel)
attritionPredData$JobSatisfaction = as.factor(attritionPredData$JobSatisfaction)
attritionPredData$PerformanceRating = as.factor(attritionPredData$PerformanceRating)
attritionPredData$RelationshipSatisfaction = as.factor(attritionPredData$RelationshipSatisfaction)
attritionPredData$StockOptionLevel = as.factor(attritionPredData$StockOptionLevel)
attritionPredData$WorkLifeBalance = as.factor(attritionPredData$WorkLifeBalance)

trainFit <- glm(formula = Attrition ~ Age + BusinessTravel + DailyRate +
    Department + DistanceFromHome + Gender + HourlyRate + JobInvolvement +
    JobLevel + JobSatisfaction + NumCompaniesWorked + OverTime +
    PercentSalaryHike + RelationshipSatisfaction + StockOptionLevel +
    TrainingTimesLastYear + WorkLifeBalance + YearsAtCompany,
    family = binomial, data = cleanData)
trainModel <- trainFit %>% stepAIC(trace = FALSE)

pred <- predict(trainModel,attritionPredData)
pred <- as.factor(as.numeric(pred>0.5))
attritionPredData$Attrition = pred
attritionPredResult <- subset(attritionPredData, select = c(ID, Attrition))
attritionPredResult$Attrition = ifelse(attritionPredResult$Attrition==0, "No", "Yes")
attritionPredResult$Attrition = as.factor(attritionPredResult$Attrition)
write.csv(x=attritionPredResult, file="Case2PredictionsShin Attrition.csv", row.names=FALSE,quote=FALSE
```

Predict Salary with No MonthlyIncome Data

```
salaryPredData <- read_excel( "data/CaseStudy2CompSet No Salary.xlsx")
salaryPredData$OverTime = ifelse(salaryPredData$OverTime=="No", 0, 1)
salaryPredData$OverTime = as.factor(salaryPredData$OverTime)
salaryPredData$Attrition = ifelse(salaryPredData$Attrition=="No", 0, 1)
salaryPredData$Attrition = as.factor(salaryPredData$Attrition)
salaryPredData$EnvironmentSatisfaction  = as.factor(salaryPredData$EnvironmentSatisfaction )
salaryPredData$JobLevel = as.factor(salaryPredData$JobLevel)
salaryPredData$JobSatisfaction = as.factor(salaryPredData$JobSatisfaction)
salaryPredData$PerformanceRating = as.factor(salaryPredData$PerformanceRating)
salaryPredData$RelationshipSatisfaction = as.factor(salaryPredData$RelationshipSatisfaction)
salaryPredData$StockOptionLevel = as.factor(salaryPredData$StockOptionLevel)
salaryPredData$WorkLifeBalance = as.factor(salaryPredData$WorkLifeBalance)
```

```r
linearModel = lm(formula = MonthlyIncome ~ Age + Attrition + BusinessTravel +
    JobLevel + JobRole + NumCompaniesWorked + OverTime + PercentSalaryHike +
    PerformanceRating + YearsAtCompany, data = caseStudy2)
predSalary = predict(linearModel, newdata = salaryPredData)
salaryPredData$MonthlyIncome = predSalary
salaryPredResult <- subset(salaryPredData, select = c(ID, MonthlyIncome))
write.csv(x=salaryPredResult, file="Case2PredictionsShin Salary.csv", row.names=FALSE, quote=FALSE)
```