In [1]:
```
library(caretEnsemble)
library(RColorBrewer)
library(tm)
library(datarium)
library(leaps)
library(glmnet)
library(pls)
library(gam)
library(splines)
library(MVA)
library(nortest)
library(mvnormtest)
library(pastecs)
library(mvtnorm)
library(igraph)
library(dplyr)
library(ggplot2)
library(ggraph)
library(caret)
library(car)
library(mlbench)
library(tidyverse)
library(MASS)
library(ISLR)
library(psych)
library(faraway)
library(pls)
library(Matrix)
library(stats)
library(biotools)
library(ggpubr)
library(broom)
library(leaps)
library(tidyverse)
library(funModeling)
library(Hmisc)
library(rpart)
library(readr)
library(party)
library(partykit)
library(rpart.plot)
library(stringr)
library(reshape2)
```

```
Loading required package: NLP

Loading required package: Matrix

Loaded glmnet 4.1-2


Attaching package: ‘pls’


The following object is masked from ‘package:stats’:

    loadings
```

```
Loading required package: splines

Loading required package: foreach

Loaded gam 1.20


Loading required package: HSAUR2

Loading required package: tools


Attaching package: ‘igraph’


The following objects are masked from ‘package:stats’:

    decompose, spectrum


The following object is masked from ‘package:base’:

    union


Attaching package: ‘dplyr’


The following objects are masked from ‘package:igraph’:

    as_data_frame, groups, union


The following objects are masked from ‘package:pastecs’:

    first, last


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union


Attaching package: ‘ggplot2’


The following object is masked from ‘package:NLP’:

    annotate


The following object is masked from ‘package:caretEnsemble’:
```

```
    autoplot


Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:pls’:

    R2


Loading required package: carData


Attaching package: ‘car’


The following object is masked from ‘package:dplyr’:

    recode


── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

✓ tibble  3.1.3     ✓ purrr   0.3.4
✓ tidyr   1.1.3     ✓ stringr 1.4.0
✓ readr   2.0.1     ✓ forcats 0.5.1

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✕ purrr::accumulate()     masks foreach::accumulate()
✕ ggplot2::annotate()     masks NLP::annotate()
✕ tibble::as_data_frame() masks dplyr::as_data_frame(), igraph::as_data_frame()
✕ ggplot2::autoplot()     masks caretEnsemble::autoplot()
✕ purrr::compose()        masks igraph::compose()
✕ tidyr::crossing()       masks igraph::crossing()
✕ tidyr::expand()         masks Matrix::expand()
✕ tidyr::extract()        masks pastecs::extract()
✕ dplyr::filter()         masks stats::filter()
✕ dplyr::first()          masks pastecs::first()
✕ dplyr::groups()         masks igraph::groups()
✕ dplyr::lag()            masks stats::lag()
✕ dplyr::last()           masks pastecs::last()
✕ purrr::lift()           masks caret::lift()
✕ tidyr::pack()           masks Matrix::pack()
✕ car::recode()           masks dplyr::recode()
✕ purrr::simplify()       masks igraph::simplify()
✕ purrr::some()           masks car::some()
✕ tidyr::unpack()         masks Matrix::unpack()
✕ purrr::when()           masks foreach::when()


Attaching package: ‘MASS’


The following object is masked from ‘package:dplyr’:

    select
```

```
Attaching package: ‘psych’

The following object is masked from ‘package:car’:

    logit

The following objects are masked from ‘package:ggplot2’:

    %+%, alpha


Attaching package: ‘faraway’

The following object is masked from ‘package:psych’:

    logit

The following objects are masked from ‘package:car’:

    logit, vif

The following object is masked from ‘package:lattice’:

    melanoma

The following objects are masked from ‘package:HSAUR2’:

    epilepsy, toenail

---
biotools version 4.2

Loading required package: Hmisc

Loading required package: survival

Attaching package: ‘survival’

The following objects are masked from ‘package:faraway’:

    rats, solder

The following object is masked from ‘package:caret’:

    cluster

Loading required package: Formula
```

```
Attaching package: ‘Hmisc’


The following object is masked from ‘package:psych’:

    describe


The following objects are masked from ‘package:dplyr’:

    src, summarize


The following objects are masked from ‘package:base’:

    format.pval, units


funModeling v.1.9.4 :)
Examples and tutorials at livebook.datascienceheroes.com
 / Now in Spanish: librovivodecienciadedatos.ai


Attaching package: ‘rpart’


The following object is masked from ‘package:faraway’:

    solder


Loading required package: grid

Loading required package: modeltools

Loading required package: stats4


Attaching package: ‘modeltools’


The following object is masked from ‘package:car’:

    Predict


The following object is masked from ‘package:igraph’:

    clusters


Loading required package: strucchange

Loading required package: zoo


Attaching package: ‘zoo’
```

```
The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric


Loading required package: sandwich


Attaching package: ‘strucchange’


The following object is masked from ‘package:stringr’:

    boundary


Loading required package: libcoin


Attaching package: ‘partykit’


The following objects are masked from ‘package:party’:

    cforest, ctree, ctree_control, edge_simple, mob, mob_control,
    node_barplot, node_bivplot, node_boxplot, node_inner, node_surv,
    node_terminal, varimp


Attaching package: ‘reshape2’


The following object is masked from ‘package:tidyr’:

    smiths
```

In [2]:

```
data01 <- read.csv('HR.csv', header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
head(data01)
```

A data.frame: 6 × 10

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_a
	<dbl>	<dbl>	<int>	<int>	<int>	
1	0.38	0.53	2	157	3	
2	0.80	0.86	5	262	6	
3	0.11	0.88	7	272	4	
4	0.72	0.87	5	223	5	
5	0.37	0.52	2	159	3	
6	0.41	0.50	2	153	3	

In [3]:

```
# Check the dimensions
dim(data01)
```

14999 · 10

In [4]:
```
str(data01)
```

```
'data.frame':   14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
 $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
 $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
 $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
 $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
 $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Department           : chr  "sales" "sales" "sales" "sales" ...
 $ salary               : chr  "low" "medium" "medium" "low" ...
```

In [5]:
```
sum(is.na(data01))
```

0

In [6]:
```
data01_eda <- function(data01)
{
  glimpse(data01)
  print(status(data01))
  freq(data01)
  print(profiling_num(data01))
  plot_num(data01)
  describe(data01)
}
```

In [7]:
```
data01_eda(data01)
```

```
Rows: 14,999
Columns: 10
$ satisfaction_level    <dbl> 0.38, 0.80, 0.11, 0.72, 0.37, 0.41, 0.10, 0.92, …
$ last_evaluation       <dbl> 0.53, 0.86, 0.88, 0.87, 0.52, 0.50, 0.77, 0.85, …
$ number_project        <int> 2, 5, 7, 5, 2, 2, 6, 5, 5, 2, 2, 6, 4, 2, 2, 2, …
$ average_montly_hours  <int> 157, 262, 272, 223, 159, 153, 247, 259, 224, 142…
$ time_spend_company    <int> 3, 6, 4, 5, 3, 3, 4, 5, 5, 3, 3, 4, 5, 3, 3, 3, …
$ Work_accident         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ left                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ promotion_last_5years <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ Department            <chr> "sales", "sales", "sales", "sales", "sales", "sa…
$ salary                <chr> "low", "medium", "medium", "low", "low", "low", …
                                variable q_zeros   p_zeros q_na p_na q_inf
satisfaction_level       satisfaction_level       0 0.0000000    0    0     0
last_evaluation             last_evaluation       0 0.0000000    0    0     0
number_project               number_project       0 0.0000000    0    0     0
average_montly_hours   average_montly_hours       0 0.0000000    0    0     0
time_spend_company       time_spend_company       0 0.0000000    0    0     0
Work_accident                 Work_accident   12830 0.8553904    0    0     0
left                                   left   11428 0.7619175    0    0     0
promotion_last_5years promotion_last_5years   14680 0.9787319    0    0     0
```

```
Department                      Department       0 0.0000000    0    0     0
salary                              salary       0 0.0000000    0    0     0
                     p_inf      type unique
satisfaction_level       0   numeric     92
last_evaluation          0   numeric     65
number_project           0   integer      6
average_montly_hours     0   integer    215
time_spend_company       0   integer      8
Work_accident            0   integer      2
left                     0   integer      2
promotion_last_5years    0   integer      2
Department               0 character     10
salary                   0 character      3
```

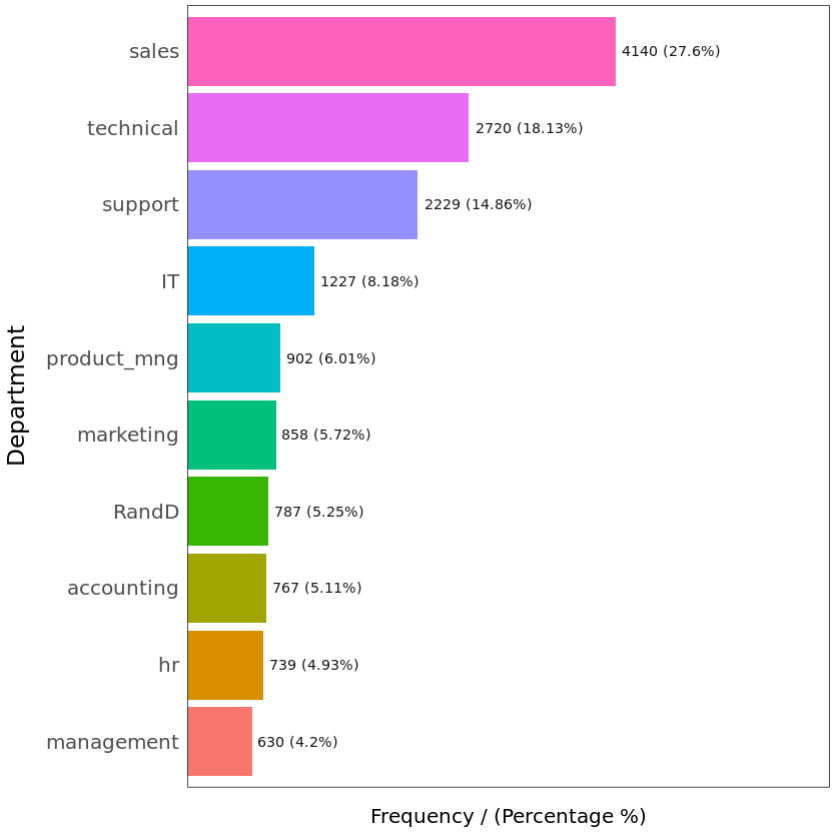
```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
    Department frequency percentage cumulative_perc
1        sales      4140      27.60           27.60
2    technical      2720      18.13           45.73
3      support      2229      14.86           60.59
4           IT      1227       8.18           68.77
5  product_mng       902       6.01           74.78
6    marketing       858       5.72           80.50
7        RandD       787       5.25           85.75
8   accounting       767       5.11           90.86
9           hr       739       4.93           95.79
10  management       630       4.20          100.00
```

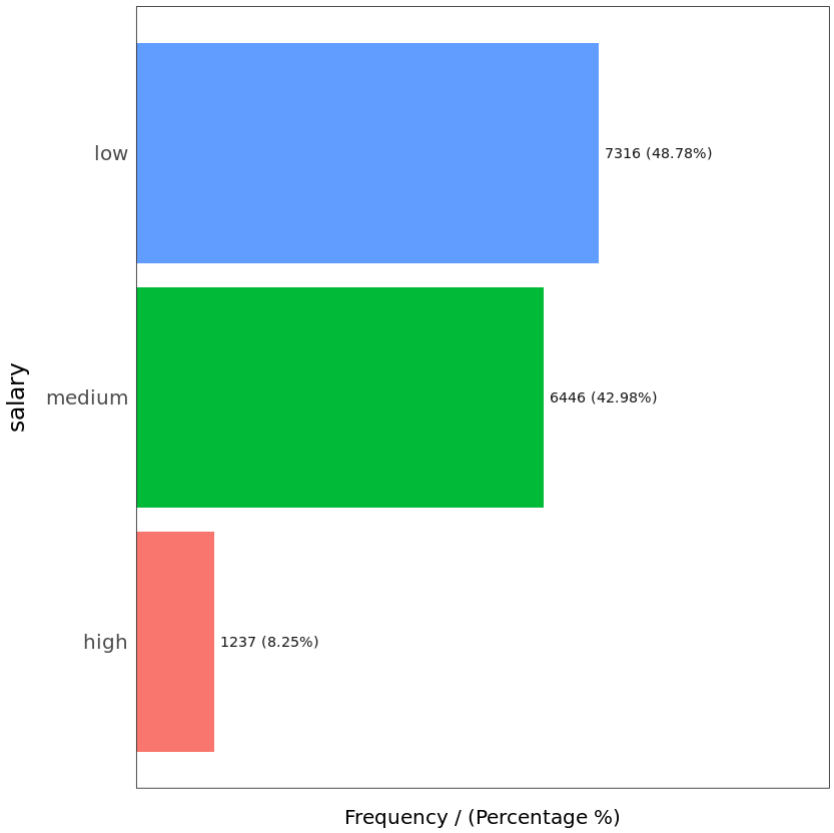
```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
  salary frequency percentage cumulative_perc
1    low      7316      48.78           48.78
2 medium      6446      42.98           91.76
3   high      1237       8.25          100.00
```

```
               variable         mean    std_dev variation_coef   p_01   p_05
1    satisfaction_level   0.61283352  0.2486307      0.4057067   0.09   0.11
2       last_evaluation   0.71610174  0.1711691      0.2390290   0.39   0.46
3        number_project   3.80305354  1.2325924      0.3241060   2.00   2.00
4  average_montly_hours 201.05033669 49.9430994      0.2484109 104.00 130.00
5    time_spend_company   3.49823322  1.4601362      0.4173925   2.00   2.00
6         Work_accident   0.14460964  0.3517186      2.4321930   0.00   0.00
7                  left   0.23808254  0.4259241      1.7889766   0.00   0.00
8 promotion_last_5years   0.02126808  0.1442815      6.7839426   0.00   0.00
    p_25   p_50   p_75   p_95   p_99    skewness  kurtosis   iqr     range_98
1   0.44   0.64   0.82   0.96   0.99 -0.47631270  2.328965  0.38 [0.09, 0.99]
2   0.56   0.72   0.87   0.98   1.00 -0.02661909  1.760973  0.31    [0.39, 1]
3   3.00   4.00   5.00   6.00   7.00  0.33767184  2.504287  2.00       [2, 7]
4 156.00 200.00 245.00 275.00 301.00  0.05283670  1.864997 89.00   [104, 301]
5   3.00   3.00   4.00   6.00  10.00  1.85313370  7.771220  1.00      [2, 10]
6   0.00   0.00   0.00   1.00   1.00  2.02094660  5.084225  0.00       [0, 1]
7   0.00   0.00   0.00   1.00   1.00  1.22991957  2.512702  0.00       [0, 1]
8   0.00   0.00   0.00   0.00   1.00  6.63630462 45.040539  0.00       [0, 1]
      range_80
1 [0.21, 0.92]
2 [0.49, 0.95]
3       [2, 5]
4   [137, 267]
5       [2, 5]
6       [0, 1]
```

```
7       [0, 1]
8       [0, 0]
```

```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
```






```
data01

 10  Variables      14999  Observations
```

```
---------------------------------------------------------------------------
satisfaction_level
       n  missing distinct     Info     Mean      Gmd      .05      .10
   14999        0       92        1   0.6128   0.2823     0.11     0.21
     .25      .50      .75      .90      .95
    0.44     0.64     0.82     0.92     0.96

lowest : 0.09 0.10 0.11 0.12 0.13, highest: 0.96 0.97 0.98 0.99 1.00
---------------------------------------------------------------------------
last_evaluation
       n  missing distinct     Info     Mean      Gmd      .05      .10
   14999        0       65        1   0.7161   0.1973     0.46     0.49
     .25      .50      .75      .90      .95
    0.56     0.72     0.87     0.95     0.98

lowest : 0.36 0.37 0.38 0.39 0.40, highest: 0.96 0.97 0.98 0.99 1.00
---------------------------------------------------------------------------
number_project
       n  missing distinct     Info     Mean      Gmd
   14999        0        6    0.945    3.803    1.367

lowest : 2 3 4 5 6, highest: 3 4 5 6 7

Value          2     3     4     5     6     7
Frequency   2388  4055  4365  2761  1174   256
Proportion 0.159 0.270 0.291 0.184 0.078 0.017
---------------------------------------------------------------------------
average_montly_hours
       n  missing distinct     Info     Mean      Gmd      .05      .10
   14999        0      215        1    201.1    57.48      130      137
     .25      .50      .75      .90      .95
     156      200      245      267      275

lowest :  96  97  98  99 100, highest: 306 307 308 309 310
---------------------------------------------------------------------------
time_spend_company
       n  missing distinct     Info     Mean      Gmd
   14999        0        8    0.905    3.498     1.43

lowest :  2  3  4  5  6, highest:  5  6  7  8 10

Value          2     3     4     5     6     7     8    10
Frequency   3244  6443  2557  1473   718   188   162   214
Proportion 0.216 0.430 0.170 0.098 0.048 0.013 0.011 0.014
---------------------------------------------------------------------------
Work_accident
       n  missing distinct     Info      Sum     Mean      Gmd
   14999        0        2    0.371     2169   0.1446   0.2474

---------------------------------------------------------------------------
left
       n  missing distinct     Info      Sum     Mean      Gmd
   14999        0        2    0.544     3571   0.2381   0.3628

---------------------------------------------------------------------------
promotion_last_5years
       n  missing distinct     Info      Sum     Mean      Gmd
   14999        0        2    0.062      319  0.02127  0.04163

---------------------------------------------------------------------------
Department
```

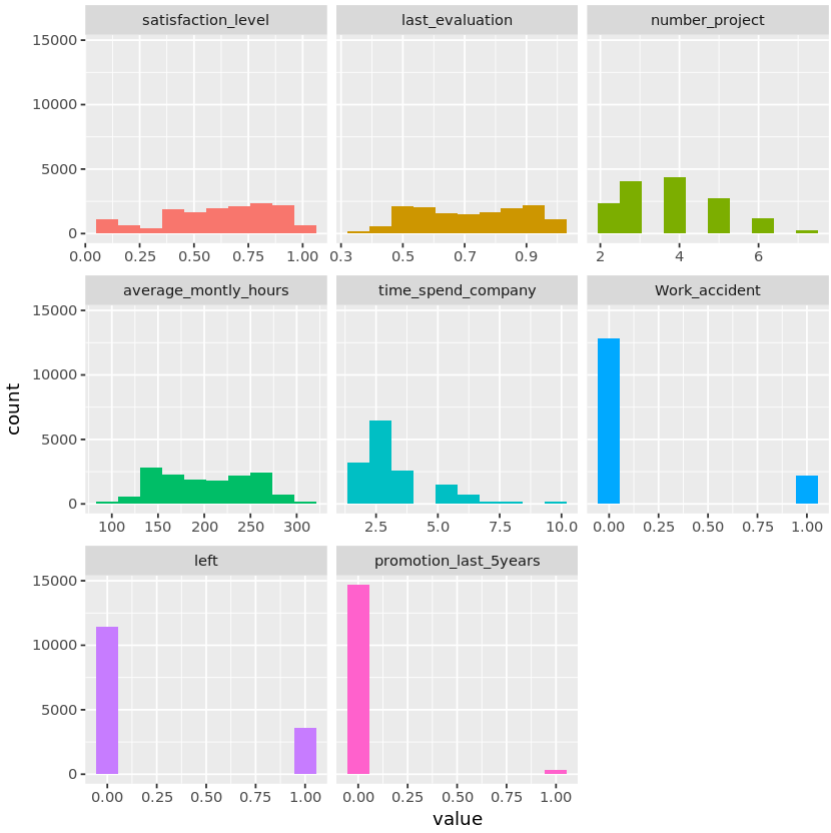
```
       n  missing distinct
   14999        0       10

lowest : accounting  hr          IT          management  marketing
highest: product_mng RandD       sales       support     technical

Value       accounting          hr          IT  management   marketing
Frequency          767         739        1227         630         858
Proportion       0.051       0.049       0.082       0.042       0.057

Value      product_mng       RandD       sales     support   technical
Frequency          902         787        4140        2229        2720
Proportion       0.060       0.052       0.276       0.149       0.181
--------------------------------------------------------------------------------
salary
       n  missing distinct
   14999        0        3

Value        high    low medium
Frequency    1237   7316   6446
Proportion  0.082  0.488  0.430
--------------------------------------------------------------------------------
```



In [9]:

```
# Subsetting the data and keeping the required variables
df <- data01[ ,c("satisfaction_level", "average_montly_hours", "promotion_last_5years", "
# Checking the dim
dim(df)
```

14999 · 5

In [10]:

```
# Generating the frequency table
table(df$salary)
```

```
 high    low medium
 1237   7316   6446
```

In [11]:

```
# Generating the frequency table
table(df$Department)
```

```
accounting         hr         IT  management   marketing product_mng
       767        739       1227        630        858        902
     RandD      sales    support  technical
       787       4140       2229       2720
```

In [12]:

```
# The data is not uniformly distributed. Some of the levels have very few observations an
# to combine similar looking levels.
df$Department[df$Department == "product_mng" | df$Department == "management"] <- "manager
df$Department[df$Department == "accounting" | df$Department == "marketing"] <- "staff"
# Checking the table again
table(df$Department)
```

```
      hr        IT  managers     RandD     sales     staff   support technical
     739      1227      1532       787      4140      1625      2229      2720
```

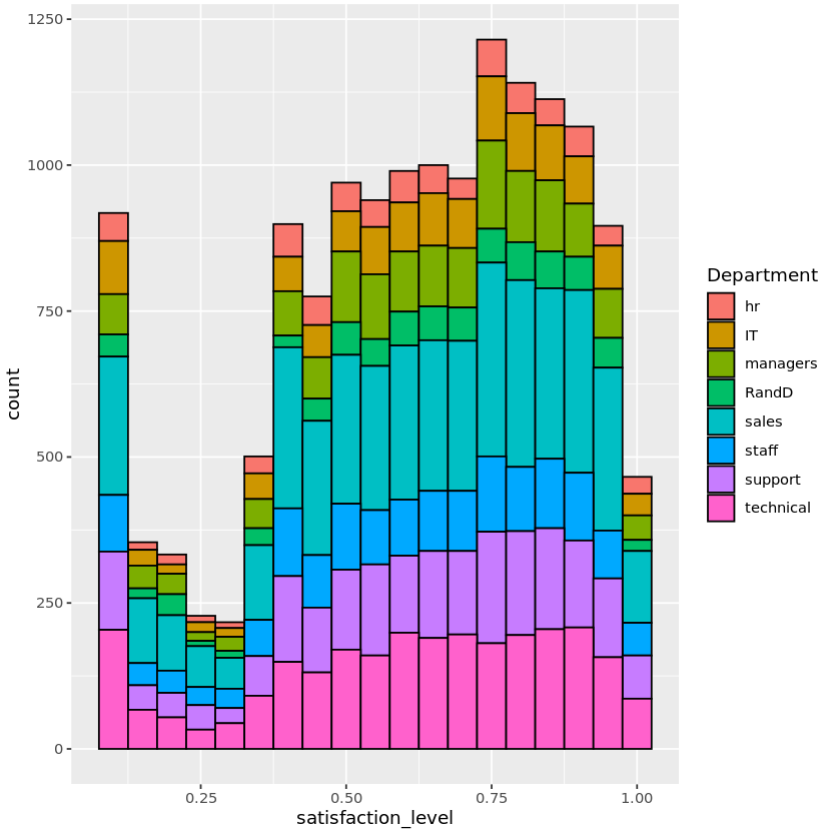
In [13]:

```
# Converting to factor variables
df$Department <- as.factor(df$Department)
df$salary <- as.factor(df$salary)
str(df)
```

```
'data.frame':   14999 obs. of  5 variables:
 $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
 $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Department           : Factor w/ 8 levels "hr","IT","managers",..: 5 5 5 5 5 5 5 5 5 5
...
 $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2
...
```

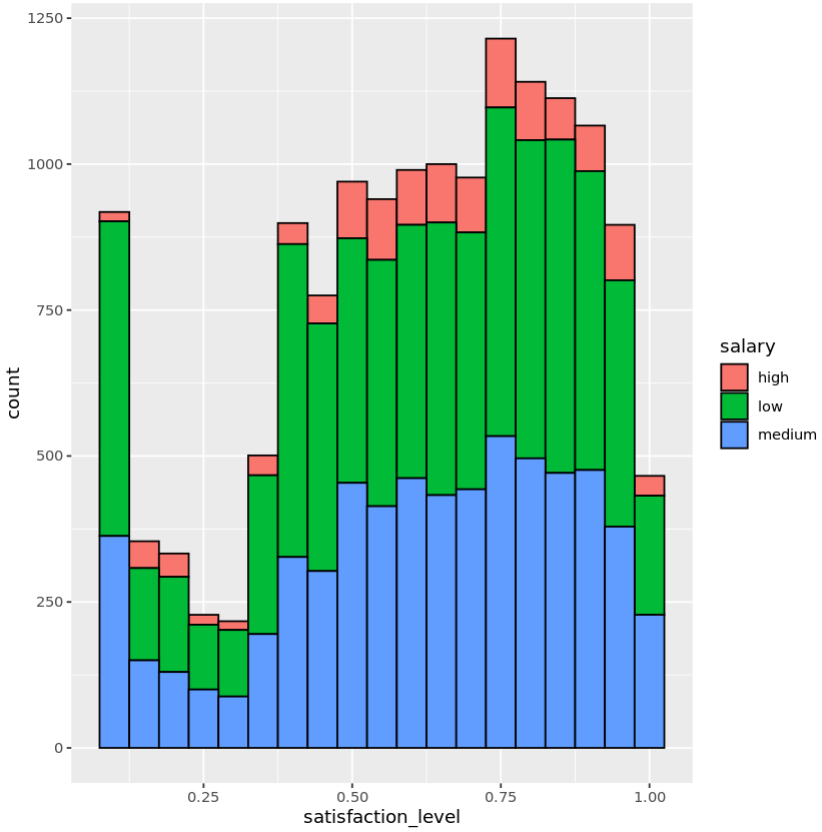
In [22]:

```
ggplot(df, aes(satisfaction_level)) +
  geom_histogram(aes(fill = Department), color = "black", binwidth = 0.05)
```



In [23]:

```
ggplot(df, aes(satisfaction_level)) +
  geom_histogram(aes(fill = salary), color = "black", binwidth = 0.05)
```



In [24]:

```
# Splitting the data into train and test
index <- createDataPartition(df$salary, p = .70, list = FALSE)
```

```
train <- df[index, ]
test <- df[-index, ]
```

In [25]:
```
# Training the model
logistic_model <- glm(salary ~ ., family = binomial(), train)
```

In [26]:
```
# Checking the model
summary(logistic_model)
```

```
Call:
glm(formula = salary ~ ., family = binomial(), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5252   0.3429   0.3740   0.4054   0.9947

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)             3.0892118  0.2523314  12.243  < 2e-16 ***
satisfaction_level     -0.6069238  0.1515486  -4.005 6.21e-05 ***
average_montly_hours    0.0003864  0.0007322   0.528   0.5976
promotion_last_5years -0.9190340  0.1742804  -5.273 1.34e-07 ***
DepartmentIT           -0.1792383  0.2296817  -0.780   0.4352
Departmentmanagers     -1.2982641  0.2011882  -6.453 1.10e-10 ***
DepartmentRandD         0.0261622  0.2586268   0.101   0.9194
Departmentsales        -0.0780850  0.2003958  -0.390   0.6968
Departmentstaff        -0.3741680  0.2134073  -1.753   0.0795 .
Departmentsupport      -0.0763770  0.2130674  -0.358   0.7200
Departmenttechnical    -0.3238723  0.2042962  -1.585   0.1129
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5980.5  on 10500  degrees of freedom
Residual deviance: 5777.4  on 10490  degrees of freedom
AIC: 5799.4

Number of Fisher Scoring iterations: 5
```

In [30]:
```
# Null deviance suggests the response by the model if we only consider the intercept; low
exp(3.0892118) # Intercept
```

21.9597624678411

In [32]:
```
exp(-0.6069238) # satisfaction_level
```

0.545024898522528

In [33]:
```
# satisfaction_level: The number indicates that the odds of an individual being in the hi
1-0.545024898522528
```

0.454975101477472

In [34]:

```
exp(-0.9190340) # promotion_last_5years
```

0.398904196478306

In [35]:

```
# promotion_last_5years: satisfaction_level decreases by 60%
1-0.398904196478306
```

0.601095803521694

In [36]:

```
exp(-1.2982641) # Departmentmanagers
```

0.273005291827917

In [37]:

```
# Departmentmanagers satisfaction_level decreases by 72%
1-0.273005291827917
```

0.726994708172083

In [40]:

```
# Predicting in the test dataset: The output of the predict function is the probability.
pred_prob <- predict(logistic_model, test, type = "response")
pred_prob
```

10: 0.943272373561077 **13:** 0.930327985911969 **14:** 0.943719541212998 **16:** 0.944577764540113 **17:** 0.942667285600173 **20:** 0.934079542993393 **21:** 0.954926595757654 **22:** 0.944658631002276 **28:** 0.943775044148324 **32:** 0.946235570002106 **34:** 0.946353406923187 **35:** 0.935502705440271 **45:** 0.94360752680508 **46:** 0.910995916035757 **47:** 0.94411080155907 **52:** 0.942565385912993 **54:** 0.944125183572775 **55:** 0.955671243951228 **56:** 0.944923196771434 **57:** 0.95454951909964 **58:** 0.94396211221203 **60:** 0.911847457027695 **62:** 0.950838632463216 **66:** 0.922769321989487 **67:** 0.8304889097724 **68:** 0.793595684295969 **78:** 0.925890822978724 **79:** 0.943257785229245 **96:** 0.945135598455986 **99:** 0.954959850438359 **101:** 0.944799880082393 **103:** 0.943437574212054 **112:** 0.958761523310445 **114:** 0.910422970711186 **123:** 0.916339374799032 **124:** 0.955298040661158 **126:** 0.954633274087573 **128:** 0.934545630202898 **129:** 0.954990266526266 **138:** 0.830037186565627 **143:** 0.949890744576257 **144:** 0.835500746204278 **153:** 0.927287026462823 **158:** 0.943557931697523 **162:** 0.905741930561319 **169:** 0.946296665685707 **172:** 0.870465955961986 **175:** 0.936562763367495 **177:** 0.943719541212998 **179:** 0.944571804356984 **180:** 0.955357087284834 **186:** 0.947624556984754 **189:** 0.930830492340125 **194:** 0.943049734767766 **197:** 0.937549846353624 **201:** 0.928686931304425 **202:** 0.944186309924408 **203:** 0.933155554957605 **206:** 0.944868761035771 **207:** 0.932993740706178 **216:** 0.939991215821058 **217:** 0.940056577543319 **220:** 0.792117240441645 **221:** 0.863040689344915 **230:** 0.911654278470196 **232:** 0.927660792682525 **235:** 0.929455531050649 **237:** 0.926261200270283 **244:** 0.943631329120126 **247:** 0.945215698191051 **248:** 0.942085892925461 **251:** 0.932925325878335 **252:** 0.932755854147555 **255:** 0.939725281108155 **258:** 0.943561177801042 **259:** 0.9064131713208 **264:** 0.957828549689626 **265:** 0.947251004793877 **267:** 0.943165625672601 **271:** 0.927396719706929 **273:**

0.915139736958232 **279:** 0.955056663011106 **281:** 0.955028366268685 **282:** 0.950980933385246 **286:** 0.944131188940403 **287:** 0.955396951389758 **288:** 0.869027250019653 **289:** 0.91351404901313 **290:** 0.930848041474881 **293:** 0.93805463467277 **296:** 0.939104250319142 **299:** 0.82783894493195 **301:** 0.936743443874452 **303:** 0.958959614376363 **305:** 0.935460094875622 **306:** 0.947570310671079 **312:** 0.862962771222828 **316:** 0.934792553728784 **322:** 0.931358143133302 **323:** 0.939270467658413 **327:** 0.945175662103467 **328:** 0.929450694302924 **330:** 0.942744625936698 **332:** 0.955258092294398 **333:** 0.955264876863742 **336:** 0.924161514160821 **337:** 0.912974459223114 **343:** 0.909600180170662 **345:** 0.930905098512665 **346:** 0.943325032310725 **352:** 0.928152291621257 **353:** 0.928361074961415 **357:** 0.943933025320833 **361:** 0.955392099769771 **371:** 0.939362422663759 **372:** 0.939553763250393 **373:** 0.833235803383818 **375:** 0.796919254369862 **376:** 0.809866249667885 **380:** 0.937681311721596 **384:** 0.944900564383423 **385:** 0.928031444293388 **391:** 0.912089050555961 **394:** 0.954926595757654 **395:** 0.943085980819011 **411:** 0.932726754405258 **412:** 0.909411945443297 **416:** 0.947559099095205 **417:** 0.935837907528714 **420:** 0.913238871431369 **429:** 0.944640953236712 **437:** 0.954650007435396 **438:** 0.943411055140495 **440:** 0.928843397051003 **447:** 0.936919658461113 **455:** 0.95925826599448 **456:** 0.947718048052049 **460:** 0.943816042253696 **463:** 0.915160904190114 **464:** 0.803909047069783 **466:** 0.927521184122598 **472:** 0.942765481462796 **480:** 0.954843356677846 **483:** 0.933224820357864 **491:** 0.94665389752899 **493:** 0.957578088070645 **496:** 0.914455843978412 **498:** 0.924762625197209 **499:** 0.927811940762608 **500:** 0.90847236540032 **503:** 0.943082709032995 **511:** 0.942795078031923 **512:** 0.940169343627712 **513:** 0.955139527367817 **517:** 0.942681149410447 **520:** 0.938377447635122 **529:** 0.95127772054896 **531:** 0.925890822978724 **533:** 0.957855162272016 **536:** 0.945155633725744 **545:** 0.94280103204528 **548:** 0.944854379860646 **550:** 0.943513937713743 **552:** 0.934262657672481 **553:** 0.931456898633806 **563:** 0.926498399548978 **567:** 0.958602161489536 **576:** 0.929464363766495 **580:** 0.930220649546293 **587:** 0.942263161829667 **597:** 0.922285239244769 **602:** 0.796882402605079 **607:** 0.95011962679535 **610:** 0.959398410009112 **616:** 0.86217048988565 **617:** 0.926490646394772 **620:** 0.942925942493877 **623:** 0.95460457426065 **626:** 0.933352210136 **631:** 0.955253226277667 **634:** 0.944799880082393 **635:** 0.930887252339101 **636:** 0.943932843124246 **638:** 0.943050596641484 **640:** 0.925061491654772 **642:** 0.93867013304337 **643:** 0.939173318586216 **646:** 0.929370428471584 **648:** 0.927203654600432 **651:** 0.92959093531636 **653:** 0.943894726245869 **655:** 0.943941666795415 **657:** 0.932401585416918 **658:** 0.942788937747864 **659:** 0.944727677900716 **664:** 0.954828776666797 **665:** 0.955160958842432 **666:** 0.93467488601315 **668:** 0.910432254717681 **669:** 0.942910406609525 **670:** … **671:** 0.945135598455986 **673:** 0.954905057554324 **674:** 0.94704390911039 **675:** 0.910422970711186 **676:** 0.912611867274326 **677:** 0.912537539295434 **678:** 0.942436241099819 **685:** 0.929218077863382 **686:** 0.955298040661158 **690:** 0.934545630202898 **693:** 0.805658876576541 **697:** 0.930979541333895 **702:** 0.923645301119143 **709:** 0.932391361772787 **710:** 0.942675864841217 **715:**

0.946545637734893 **716:** 0.946296665685707 **721:** 0.955357087284834 **725:** 0.940387031385258 **726:** 0.958883583058914 **727:** 0.947930589389725 **730:** 0.928147694227638 **734:** 0.943751298901213 **736:** 0.943049734767766 **739:** 0.937549846353624 **741:** 0.943469481476556 **742:** 0.929757325392076 **747:** 0.944868761035771 **748:** 0.940056577543319 **749:** 0.921175053887593 **750:** 0.937004234059206 **751:** 0.863040689344915 **757:** 0.922680253583626 **760:** 0.926208393726225 **761:** 0.949102741241277 **762:** 0.939492597676852 **763:** 0.929455531050649 **774:** 0.799911904780307 **778:** 0.92457785688567 **782:** 0.942562768448828 **787:** 0.942085892925461 **788:** 0.932925325878335 **789:** 0.932755854147555 **790:** 0.9512006650009 **796:** 0.9064131713208 **803:** 0.946484809276689 **804:** 0.913134378269817 **811:** 0.943165625672601 **815:** 0.927039297218188 **817:** 0.942436241099819 **821:** 0.94401739011833 **822:** 0.955011765949537 **825:** 0.934146863644313 **827:** 0.931195396419147 **831:** 0.955375627441483 **836:** 0.944687306040632 **845:** 0.944131188940403 **847:** 0.955396951389758 **865:** 0.91351404901313 **872:** 0.950838632463216 **873:** 0.834039714107059 **877:** 0.832713993867989 **881:** 0.955309680124925 **883:** 0.925943838611089 **884:** 0.934792553728784 **885:** 0.954826691296615 **900:** 0.939270467658413 **902:** 0.954921695389514 **906:** 0.955258092294398 **908:** 0.930556297961535 **909:** 0.930905098512665 **914:** 0.943325032310725 **921:** 0.914628160757603 **925:** 0.944180549542442 **931:** 0.928152291621257 **937:** 0.947706904891936 **939:** 0.943563752491061 **941:** 0.955403715885814 **942:** 0.934208388874335 **946:** 0.943748673864458 **947:** 0.938070472898188 **948:** 0.950361462793972 **951:** 0.939362422663759 **958:** 0.835803549639104 **967:** 0.944900564383423 **971:** 0.93441485741116 **975:** 0.943736826385724 **979:** 0.924323862110775 **980:** 0.931001262942351 **987:** 0.941895841846178 **990:** 0.927963758977117 **991:** 0.935128466557279 **992:** 0.927270836265905 **994:** 0.934698267241282 **1002:** 0.935837907528714 **1003:** 0.943442792774151 **1005:** 0.943510689055855 **1006:** 0.91235230378981 **1007:** 0.927440988668387 **1010:** 0.930009287120284 **1013:** 0.954650007435396 **1015:** 0.933210844682133 **1016:** 0.804098812252848 **1026:** 0.80006097430155 **1030:** 0.935811879535931 **1032:** 0.943816042253696 **1036:** 0.803909047069783 **1044:** 0.909520472733794 **1046:** 0.943944892140616 **1048:** 0.954821780580758 **1050:** 0.942765481462796 **1051:** 0.929341792177121 **1058:** 0.92666693068767 **1063:** 0.935470389899766 **1064:** 0.924762625197209 **1065:** 0.940169343627712 **1068:** 0.955139527367817 **1069:** 0.931843594226985 **1072:** 0.9442822968754 **1075:** 0.829523516375568 **1076:** 0.950779084389495 **1081:** 0.805520001130496 **1091:** 0.957855162272016 **1092:** 0.923563499728598 **1093:** 0.910649560976284 **1095:** 0.912048927867417 **1096:** 0.94280103204528 **1099:** 0.932828533486796 **1100:** 0.955170562083474 **1102:** 0.926498399548978 **1104:** 0.938390853026236 **1106:** 0.946543500386548 **1107:** 0.927656490119273 **1109:** 0.918314901361268 **1110:** 0.917135790677985 **1112:** 0.930805607049006 **1115:** 0.925921544105684 **1124:** 0.942774232635811 **1127:** 0.934455234277101 **1131:** 0.870397081923928 **1132:** 0.950592574092 **1133:** 0.799292675283335 **1136:** 0.95011962679535 **1141:** 0.959398410009112 **1142:** 0.926490646394772 **1149:** 0.943883517105802 **1151:** 0.954559247063655 **1155:** 0.943932843124246 **1156:**

0.942577524590425 **1170:** 0.943050596641484 **1172:** 0.925061491654772 **1175:** 0.93867013304337 **1181:** 0.939173318586216 **1184:** 0.944231880824781 **1188:** 0.927039297218188 **1192:** 0.943463409240902 **1195:** 0.942788937747864 **1200:** 0.944727677900716 **1212:** 0.945115556292132 **1222:** 0.942675864841217 **1225:** 0.943127451097428 **1229:** 0.943050596641484 **1230:** 0.954441766596496 **1234:** 0.940380648970544 **1239:** 0.957823950886513 **1241:** 0.92819922055853 **1246:** 0.943537347636259 **1251:** 0.944008759624218 **1253:** 0.955518831391232 **1258:** 0.943162974949498 **1270:** 0.943020446972739 **1271:** 0.912466740940787 **1275:** 0.922348772047625 **1276:** 0.950324989575032 **1277:** 0.829741999078289 **1278:** 0.804324276526299 **1280:** 0.921901539036755 **1284:** 0.959488632718706 **1295:** 0.937971499112262 **1296:** 0.909115608354576 **1299:** 0.831016251445994 **1308:** 0.924531886767694 **1309:** 0.935465479449407 **1316:** 0.937785926055021 **1317:** 0.933460952397524 **1318:** 0.930932527808751 **1324:** 0.955264876863742 **1325:** 0.940926325057961 **1334:** 0.927695189930166 **1336:** 0.92795397539495 **1341:** 0.958842344227781 **1343:** 0.936707290069349 **1346:** 0.915667726537591 **1352:** 0.909799999993502 **1353:** 0.927046997061803 **1354:** 0.928628274632208 **1356:** 0.928294501306143 **1359:** 0.944192309110582 **1361:** 0.945324051806199 **1364:** 0.944983500979409

In [43]:
```
# Converting probability to class values in the training dataset
# Converting from probability to actual output
train$pred_sal <- ifelse(logistic_model$fitted.values >= 0.9, "low", "high")
# Generating the classification table
ctab_train <- table(train$salary, train$pred_sal)
ctab_train
```

```
        high  low
 high    228  638
 low     499 4623
 medium  567 3946
```

In [44]:
```
# Training dataset converting from probability to class values
# Converting from probability to actual output
test$pred_sal <- ifelse(pred_prob >= 0.9, "low", "high")
# Generating the classification table
ctab_test <- table(test$salary, test$pred_sal)
ctab_test
```

```
        high  low
 high     99  272
 low     214 1980
 medium  228 1705
```

In [45]:
```
# Accuracy in Training dataset
accuracy_train <- sum(diag(ctab_train))/sum(ctab_train)*100
accuracy_train
```

46.1956004190077

Our logistics model is able to classify 46.2% of all the observations correctly in the training dataset.

In [46]:
```
# Accuracy in Test dataset
accuracy_test <- sum(diag(ctab_test))/sum(ctab_test)*100
accuracy_test
```

46.220542463317

The overall correct classification accuracy in test dataset is 46.2% which is comparable to train dataset. This shows that our model is performing not so good but could be improved.

In [47]:
```
# Misclassification Rate indicates how often is our predicted values are False.
# Recall or TPR indicates how often does our model predicts actual TRUE from the overall
Recall <- (ctab_train[2, 2]/sum(ctab_train[2, ]))*100
Recall
```

90.2577118313159

In [48]:
```
# True Negative Rate indicates how often does our model predicts actual nonevents from th
TNR <- (ctab_train[1, 1]/sum(ctab_train[1, ]))*100
TNR
```

26.3279445727483

In [49]:
```
# Precision indicates how often does your predicted TRUE values are actually TRUE.
Precision <- (ctab_train[2, 2]/sum(ctab_train[, 2]))*100
Precision
```

50.2117953730857

In [50]:
```
# F-Score is a harmonic mean of recall and precision. The score value lies between 0 and
# The value of 1 represents perfect precision & recall. The value 0 represents the worst
F_Score <- (2 * Precision * Recall / (Precision + Recall))/100
F_Score
```

0.645264847512038

This ROC curve is a line plot that is drawn between the Sensitivity and (1 – Specificity) Or between TPR and TNR. This graph is then used to generate the AUC value. An AUC value of greater than .70 indicates a good model.

In [51]:
```
library(pROC)
roc <- roc(train$salary, logistic_model$fitted.values)
auc(roc)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: ‘pROC’


The following objects are masked from ‘package:stats’:

    cov, smooth, var


Warning message in roc.default(train$salary, logistic_model$fitted.values):
“'response' has more than two levels. Consider setting 'levels' explicitly or using 'mult
```

```
iclass.roc' instead”
Setting levels: control = high, case = low

Setting direction: controls < cases
```

0.629990134483048

The AUC value for this model is 0.63 < 0.7: the model is close to a good model.Concordance In how many pairs does the probability of ones is higher than the probability of zeros divided by the total number of possible pairs. The higher the values better is the model. The value of concordance lies between 0 and 1.

In [53]:
```
install.packages("InformationValue")
library(InformationValue)
```

```
Installing package into ‘/home/mladenoffj/R_libs’
(as ‘lib’ is unspecified)


Attaching package: ‘InformationValue’


The following objects are masked from ‘package:caret’:

    confusionMatrix, precision, sensitivity, specificity
```

In [54]:
```
Concordance(logistic_model$y,logistic_model$fitted.values)
```

$Concordance	0.619826675982843
$Discordance	0.380173324017157
$Tied	0
$Pairs	8343910

In [ ]:
```

```