Jeetscase2

Ben

May 16, 2018

```r
###Read in the Data


setwd("C:/Users/Meow/Documents/Bootcamp/PeopleAnalytics")


library(readxl)

PeopleAnalyticsA <- read_excel("~/Bootcamp/PeopleAnalytics/PeopleAnalyticsA.xlsx")

PeopleAnalytica = data.frame(PeopleAnalyticsA)
```

```r
###Load packages


library(latexpdf)
```

```
## Warning: package 'latexpdf' was built under R version 3.4.2
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## 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(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.2
```

```
## Loading required package: lattice
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.1

## Loaded glmnet 2.0-16
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.2

##
## Attaching package: 'gridExtra'

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

```r
library(tabplot)
```

```
## Warning: package 'tabplot' was built under R version 3.4.2

## Loading required package: bit

## Warning: package 'bit' was built under R version 3.4.1

## Attaching package bit

## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)

## creators: bit bitwhich

## coercion: as.logical as.integer as.bit as.bitwhich which

## operator: ! & | xor != ==

## querying: print length any all min max range sum summary

## bit access: length<- [ [<- [[ [[<-

## for more help type ?bit

##
## Attaching package: 'bit'

## The following object is masked from 'package:base':
##
##     xor

## Loading required package: ff

## Warning: package 'ff' was built under R version 3.4.2

## Attaching package ff

## - getOption("fftempdir")=="C:/Users/Meow/AppData/Local/Temp/RtmpYxi2ze"

## - getOption("ffextension")=="ff"

## - getOption("ffdrop")==TRUE

## - getOption("fffinonexit")==TRUE
## - getOption("ffpagesize")==65536
## - getOption("ffcaching")=="mmnoflush"  -- consider "ffeachflush" if your system stalls on large write
## - getOption("ffbatchbytes")==127506841.6 -- consider a different value for tuning your system
## - getOption("ffmaxbytes")==6375342080 -- consider a different value for tuning your system
##
## Attaching package: 'ff'
## The following objects are masked from 'package:bit':
##
##     clone, clone.default, clone.list
## The following objects are masked from 'package:utils':
##
##     write.csv, write.csv2
## The following objects are masked from 'package:base':
##
##     is.factor, is.ordered
## Loading required package: ffbase
## Warning: package 'ffbase' was built under R version 3.4.2
##
## Attaching package: 'ffbase'
## The following objects are masked from 'package:ff':
##
##     [.ff, [.ffdf, [<-.ff, [<-.ffdf
## The following objects are masked from 'package:base':
##
##     %in%, table
```

```r
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.4.1
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```r
library("ROCR")
```

```
## Warning: package 'ROCR' was built under R version 3.4.4
## Loading required package: gplots
## Warning: package 'gplots' was built under R version 3.4.4
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(vioplot)
```

```
## Warning: package 'vioplot' was built under R version 3.4.4
## Loading required package: sm

## Warning: package 'sm' was built under R version 3.4.3

## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.4.3

##
## Attaching package: 'MASS'

## The following object is masked from 'package:sm':
##
##     muscle

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

```r
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 3.4.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.4.1

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(gvlma)
```

```
## Warning: package 'gvlma' was built under R version 3.4.1
```

```r
###Missing or nonsensical values


apply(PeopleAnalyticsA, 2, function(x) any(is.na(x)))
```

```
##                            Employee ID
##                                  FALSE
##                                 School
##                                  FALSE
##                            Accredited?
##                                   TRUE
##                      Years in Fin Svcs
##                                  FALSE
##                            College GPA
##                                  FALSE
##                   Finance / Math major
##                                  FALSE
##        Previously employed at Vanguard
##                                  FALSE
##                                 Gender
##                                  FALSE
##                                   Age
##                                 FALSE
##                           US Citizen?
##                                 FALSE
##                     Internal Referral
##                                 FALSE
##                           Hiring Site
##                                 FALSE
##                             Recruiter
##                                 FALSE
##                           Career Fair
##                                 FALSE
##                    Campus  Recruiting
##                                 FALSE
##                    External Recruiter
##                                 FALSE
##             FINRA Series 7 test score
##                                 FALSE
## Num. of tries before passing Series 7
##                                 FALSE
##                       Training Rating
##                                 FALSE
##                     Learning Aptitude
##                                 FALSE
##                            motivation
##                                  TRUE
##          Training practice test score
##                                 FALSE
```

```
PeopleAnalytica[117, 3] = "No"

PeopleAnalytica[3, 21 ] = "7"

PeopleAnalytica[128, 21 ] = "8"
```

```
###More imputation


PeopleAnalytica$motivation <- as.numeric(PeopleAnalytica$motivation)


People = data.frame(PeopleAnalytica)

People$motivation[is.na(People$motivation)] = mean(People$motivation, na.rm=TRUE)

People$Learning.Aptitude[is.na(People$Learning.Aptitude)] = mean(People$Learning.Aptitude, na.rm=TRUE)

People$Learning.Aptitude[is.na(People$Num..of.tries.before.passing.Series.7)] = median(People$Num..of.t

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

```
##                           Employee.ID
##                                     0
##                                School
##                                     0
##                              Accredited.
##                                        0
##                        Years.in.Fin.Svcs
##                                        0
##                              College.GPA
##                                        0
##                     Finance...Math.major
##                                        0
##          Previously.employed.at.Vanguard
##                                        0
##                                   Gender
##                                        0
##                                      Age
##                                        0
##                              US.Citizen.
##                                        0
##                        Internal.Referral
##                                        0
##                              Hiring.Site
##                                        0
##                                Recruiter
##                                        0
##                              Career.Fair
##                                        0
##                       Campus..Recruiting
##                                        0
##                       External.Recruiter
##                                        0
##                FINRA.Series.7.test.score
##                                        0
## Num..of.tries.before.passing.Series.7
##                                        0
##                          Training.Rating
##                                        0
##                        Learning.Aptitude
##                                        0
##                               motivation
##                                        0
##             Training.practice.test.score
##                                        0
```

```
#### converting data into a more usable form. Y and y are now an equivalent value (1) and N (1)


People1 = data.frame(People)


People1$GenderN = ifelse(People$Gender == 'M', 0,
                  ifelse(People$Gender== 'F', 1, 2))


People1$AccreditedN  = ifelse(People$Accredited. == 'No', 0,
                       ifelse(People$Accredited. == 'Yes', 1, 2))
```

```
People1$FinanceMathN =  ifelse(People$Finance...Math.major == 'N', 0,
                          ifelse(People$Finance...Math.major == 'y',1,
                               ifelse(People1$Finance...Math.major == 'Y', 1, 2)))

People1$Previously.employedN = ifelse(People1$Previously.employed.at.Vanguard == 'N', 0,
                                     ifelse(People1$Previously.employed.at.Vanguard == 'Y', 1, 2))


People1$US.CitizenN = ifelse(People1$US.Citizen. == 'N', 0,
                          ifelse(People1$US.Citizen. == 'Y', 1,
                                  ifelse(People1$US.Citizen == 'y' ,1 , 2)))


People1$Internal.ReferralN= ifelse(People1$Internal.Referral == 'N', 0,
                               ifelse(People1$Internal.Referral == 'Y', 1,
                                 ifelse(People1$Internal.Referral == 'y', 1, 2)))



People1$Career.FairN = ifelse(People1$Career.Fair == 'N', 0,
                              ifelse(People1$Career.Fair == 'Y', 1,
                                     ifelse(People$Career.Fair =='y', 1, 2)))

People1$External.RecruiterN = ifelse(People$External.Recruiter == 'N', 0,
                                ifelse(People1$External.Recruiter == 'Y', 1,
                                  ifelse(People$External.Recruiter==  'y', 1, 2)))


People1$Training.RatingN = ifelse(People$Training.Rating == 'Low',0,
                                  ifelse(People$Training.Rating == 'Medium', 1,
                                         ifelse(People$Training.Rating == 'High', 2, 3)))



People1$SchoolN = ifelse(People$School == 'Arizona State University',0,
                                  ifelse(People$School == 'Drexel University', 1,
                                         ifelse(People$School == 'Florida State University',2,
                                              ifelse(People1$School == 'Penn State',3,
                                                ifelse(People$School == 'Penn State University' ,3,
                                                  ifelse(People$School == 'Strayer Univeristy' ,4,
                                                    ifelse(People$School == "St.Joseph's University" ,9
                                                      ifelse(People$School == "St. Josephs University",
                                                        ifelse(People$School == "St. Joseph's Univ",9,
                                                           ifelse(People$School == "St. Joseph's Univ."
                                                              ifelse(People$School == 'Strayer Universi
                                                                ifelse(People$School == 'University of 
                                                                  ifelse(People$School == 'University o
                                                                    ifelse(People$School == "Villanova 


People1$Hiring.SiteN = ifelse(People$Hiring.Site == 'AZ',0,
                          ifelse(People$Hiring.Site == 'NC', 1,
                           ifelse(People1$Hiring.Site== 'PA', 2, 2)))
```

```
People1$RecruiterN = ifelse(People$Recruiter == 'AL',0,
                      ifelse(People$Recruiter ==  'DE', 1,
                           ifelse(People$Recruiter == 'DT', 2,
                                  ifelse(People$Recruiter == 'GJ',3,
                                         ifelse(People$Recruiter == 'OK' ,4,
                                                ifelse(People$Recruiter== 'GG' ,5,
                                                       ifelse(People$Recruiter == "ER" ,6,
                                                           ifelse(People$Recruiter== "TR",7,
                                                                     ifelse(People$Recruiter== "LK",8,
                                                                            ifelse(People$R == "PL",9, 9

People1$Campus.RecruitingN = ifelse(People$Campus..Recruiting == 'N',0,
                                    ifelse(People$Campus..Recruiting == 'Y', 1,
                                           ifelse(People$Campus..Recruiting== 'y', 1, 2)))
```

```
###Finding the cost per test taker (Recruiting Costs here)  External Recruiter = 5000$, Career Fair = 50


People3 = People1




People3$External.Cost = ifelse(People3$External.RecruiterN == '0', 0,
                        ifelse(People3$External.RecruiterN ==  '1', 5000, 0))


People3$Career.FairCost = ifelse(People3$Career.FairN == '0', 0,
                          ifelse(People3$Career.FairN == '1',500,0))


People3$Campus.Recruiting.Cost = ifelse(People3$Campus.RecruitingN == '0' ,0,
                                  ifelse(People1$Campus.RecruitingN== '1',3000,0))
```

```
##### getting Total Costs and a few more variables/features converted into usable form

People4 = People3



People4$TotalCost <- paste0(People4$External.Cost,People4$Career.FairCost, People4$Campus.Recruiting.Cos



People4$TotalCost = ifelse(People4$TotalCost == '000' ,1000,
                                ifelse(People4$TotalCost == '003000', 3000,
                                  ifelse(People4$TotalCost == '05000', 500,
                                   ifelse(People4$TotalCost == '500000', 5000, 0))))




People4$US.CitizenN = ifelse(People4$US.Citizen. == 'N' ,0,
                           ifelse(People4$US.Citizen. == 'Y', 1, 2))


People4$AccreditedN = ifelse(People4$Accredited. == 'No' ,0,
                           ifelse(People4$Accredited. == 'Yes', 1, 2))
```

```
### Imputing values for costs that don't automatically convert (Recruiting costs from more than one for


People4$TotalRecruitingCost = People4$TotalCost


People4[183, 40] =  "5500"
People4[105, 40] = "3500"
People4[62, 40] = "3500"


People4$TotalCost = NULL

People4$TotalTrainingCost = ifelse(People4$Num..of.tries.before.passing.Series.7 == '1', 1500,
                             ifelse(People4$Num..of.tries.before.passing.Series.7 == '2', 3000,
                               ifelse(People4$Num..of.tries.before.passing.Series.7 == '3', 4500, 0)))






People4[,39] <- as.numeric(as.character(People4[,39]))

People4[,40] <- as.numeric(as.character(People4[,40]))


People4$TotalCost = People4$TotalTrainingCost + People4$TotalRecruitingCost
```

```
## new data frame with all the use variables for analysis

People5 = cbind(People4$Years.in.Fin.Svcs, People4$Age, People4$FINRA.Series.7.test.score, People4$Num.
                People4$Learning.Aptitude, People4$motivation, People4$Training.practice.test.score, Pe
                People4$Training.RatingN,People4$FinanceMathN,People4$Previously.employedN, People4$Inte
                People4$Campus.RecruitingN,People4$External.RecruiterN, People4$Career.FairN, People4$T
                People4$TotalCost,People4$Hiring.SiteN, People4$RecruiterN, People4$College.GPA, People
                People4$AccreditedN)




People6 = data.frame(People5)





names(People6) <- c("Years.in.Fin", "Age", "Test.Score", "Number.Tries","Learning Aptitude", "Motivatio
                    "Gender", "Train.RatingN", "Finance.Math", "Previously.Employed","Internal.Referral
                    "Campus.Recruiting","External Recruiting","Career.Fair.Recruiting", "Total.Training
                    "Hiring Site", "The.Recruiter", "College.GPA", "School.Type","Citizen", "Accredited
```

```
## a few more issues, get rid of useless columns (iNternal referal has only one value)

People7 = People6

People7 = cbind(People6, People4$Training.practice.test.score)

colnames(People7)[colnames(People7)=="People4$Training.practice.test.score"] <- "People7$Training.Practi



People6$`People7$Training.Practice` = NULL


People6 = People7

People6$Internal.ReferralN = NULL

People6$Training.Practice = NULL
```

```
##Convert features into correct form (nominal, ordinal, numeric)

People6$`Learning Aptitude` = as.ordered(People6$`Learning Aptitude`)
People6$`Motivation` = as.ordered(People6$`Motivation`)
People6$Train.RatingN = as.ordered(People6$Train.RatingN)

People6$Previously.Employed = as.factor(People6$Previously.Employed)
People6$Finance.Math = as.factor(People7$Finance.Math)
People6$Gender = as.factor(People6$Gender)
People6$Internal.Referral =as.factor(People6$Internal.Referral)
People6$Campus.Recruiting = as.factor(People6$Campus.Recruiting)


People6$`External Recruiting` = as.factor(People6$`External Recruiting`)
People6$Career.Fair.Recruiting = as.factor(People6$Career.Fair.Recruiting)
People6$`Hiring Site` = as.factor(People6$`Hiring Site`)

People6$School.Type = as.factor(People6$School.Type)
People6$Citizen = as.factor(People6$Citizen)
People6$Accredited = as.factor(People6$Accredited)
People6$The.Recruiter = as.factor(People6$The.Recruiter)
```

```
### Convert back to numeric for Regression/other analysis (only for ordinal values)

People6$Accredited = NULL
People6$Total.Training.Cost = NULL
People6$Total.Recruiting.Cost = NULL
People6$Total.Cost = NULL

####No value to the regression model


People6$`Learning Aptitude` = as.numeric(People6$`Learning Aptitude`)
```

```
People6$`Motivation` = as.numeric(People6$`Motivation`)
People6$Train.RatingN = as.numeric(People6$Train.RatingN)
```

```
### Split the data into Training and Testing Set

set.seed(321)
trainIdx <- createDataPartition(People6$Test.Score,
                                p = .8,
                                list = FALSE,
                                times = 1)
subTrain <- People6[trainIdx,]
subTest <- People6[-trainIdx,]
print(head(subTrain))
```

```
##   Years.in.Fin Age Test.Score Number.Tries Learning Aptitude Motivation
## 1            0  24         90            2                 6          9
## 2            0  24         71            1                 4          3
## 3            1  26         78            3                 4          7
## 4            0  22         77            3                 3          8
## 5            0  22         87            2                 5          1
## 6            3  23         91            2                 5          9
##   Gender Train.RatingN Finance.Math Previously.Employed Internal.Referral
## 1      0             1            1                   0                 0
## 2      0             3            0                   0                 0
## 3      0             3            1                   0                 0
## 4      0             3            1                   0                 0
## 5      0             3            1                   0                 0
## 6      0             3            1                   1                 0
##   Campus.Recruiting External Recruiting Career.Fair.Recruiting Hiring Site
## 1                 0                   1                      0           2
## 2                 0                   0                      1           2
## 3                 0                   0                      1           1
## 4                 1                   0                      0           2
## 5                 0                   1                      0           0
## 6                 1                   0                      0           2
##   The.Recruiter College.GPA School.Type Citizen People7$Training.Practice
## 1             0         3.5           6       1                  91.62605
## 2             1         3.3           0       1                  70.51248
## 3             2         2.9           0       1                  74.06434
## 4             3         3.3           0       1                  66.25515
## 5             4         2.9           1       1                  85.00000
## 6             5         3.0           1       1                  87.84757
```

```
###Set up for machine learning

set.seed(456)
fitCtrl <- trainControl(method = "repeatedcv",
                        number = 5,
                        repeats = 3,
                        verboseIter = FALSE,
                        summaryFunction = defaultSummary)
```

```
### simple linear regression (all variables/features included)

lmFit <- train(Test.Score ~., data = subTrain,
               method = "lm")
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading
```

```
#### find Variable importance and summarize
### Recruiter 10 = NA - Recruiter only recruited one guy. Not useful for prediction. Lazy recruiter.
### Acreddited must be highly correlated with schoo?

lmImp <- varImp(lmFit, scale = FALSE)


summary(lmFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4657 -2.0439 -0.0637  2.2560  6.6381
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       40.14473    7.56829   5.304 5.30e-07 ***
## Years.in.Fin                       1.82912    0.38574   4.742 5.93e-06 ***
## Age                               -0.11162    0.18226  -0.612   0.5414
## Number.Tries                      -0.98874    0.42418  -2.331   0.0214 *
## `\\`Learning Aptitude\\``         -0.12645    0.82747  -0.153   0.8788
## Motivation                         0.12384    0.12471   0.993   0.3227
## Gender1                            0.47603    0.70261   0.678   0.4994
## Train.RatingN                      0.47466    0.46416   1.023   0.3086
## Finance.Math1                      3.53829    0.77134   4.587 1.12e-05 ***
## Previously.Employed1               0.30294    2.78882   0.109   0.9137
## Internal.Referral2                -0.24420    3.79809  -0.064   0.9488
## Campus.Recruiting1                 0.98419    0.72789   1.352   0.1789
## `\\`External Recruiting\\`1`       0.50582    0.95034   0.532   0.5955
## Career.Fair.Recruiting1            1.97363    1.07815   1.831   0.0697 .
## Career.Fair.Recruiting2           -0.23159    3.76499  -0.062   0.9511
## `\\`Hiring Site\\`1`              -0.08831    1.60725  -0.055   0.9563
## `\\`Hiring Site\\`2`               0.80332    1.08092   0.743   0.4588
## The.Recruiter1                    -2.28994    2.58523  -0.886   0.3775
## The.Recruiter2                    -2.72176    2.47621  -1.099   0.2739
## The.Recruiter3                    -1.14816    2.55446  -0.449   0.6539
## The.Recruiter4                    -2.98413    2.19989  -1.356   0.1775
## The.Recruiter5                    -2.22122    2.16795  -1.025   0.3076
## The.Recruiter6                    -3.41497    2.24059  -1.524   0.1301
## The.Recruiter7                    -4.26971    2.50178  -1.707   0.0905 .
## The.Recruiter8                    -4.51936    2.49156  -1.814   0.0722 .
## The.Recruiter9                    -2.64826    2.48296  -1.067   0.2883
## College.GPA                        0.08836    0.65822   0.134   0.8934
## School.Type1                       2.46722    1.71139   1.442   0.1520
## School.Type2                       2.27068    1.30127   1.745   0.0836 .
## School.Type3                       1.53190    1.31964   1.161   0.2480
## School.Type6                       2.68702    1.66305   1.616   0.1088
## School.Type7                       0.07316    1.44408   0.051   0.9597
## School.Type8                      -0.55556    1.68297  -0.330   0.7419
## School.Type9                       1.13302    1.17611   0.963   0.3373
## Citizen1                          -0.29288    1.77953  -0.165   0.8696
## `\\`People7$Training.Practice\\``  0.50559    0.09458   5.346 4.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.466 on 119 degrees of freedom
## Multiple R-squared:  0.8209, Adjusted R-squared:  0.7682
## F-statistic: 15.58 on 35 and 119 DF,  p-value: < 2.2e-16
```

```
####Multiple Regression ASsumptions

fit <- lm(People6$Test.Score ~ ., data= People6)

gvmodel <- gvlma(fit)

summary(gvmodel)
```

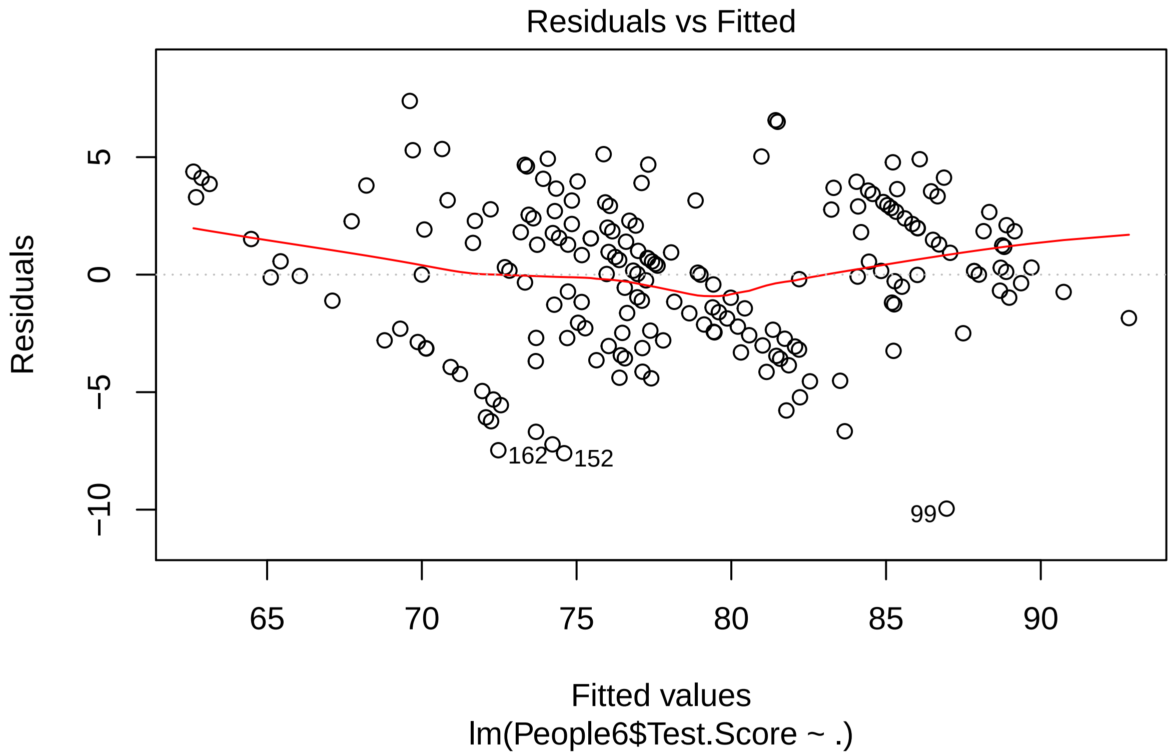
```
##
## Call:
## lm(formula = People6$Test.Score ~ ., data = People6)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.956 -2.395  0.156  2.288  7.390
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 40.28669    6.46040   6.236 4.03e-09 ***
## Years.in.Fin                 1.45216    0.35563   4.083 7.07e-05 ***
## Age                         -0.06330    0.16138  -0.392   0.6954
## Number.Tries                -0.90803    0.37567  -2.417   0.0168 *
## `Learning Aptitude`          0.14823    0.75932   0.195   0.8455
## Motivation                   0.18081    0.10803   1.674   0.0962 .
## Gender1                     -0.36053    0.61365  -0.588   0.5577
## Train.RatingN                0.26463    0.41287   0.641   0.5225
## Finance.Math1                3.48816    0.66587   5.238 5.17e-07 ***
## Previously.Employed1         1.20593    2.75635   0.438   0.6623
## Internal.Referral2          -0.41027    3.76305  -0.109   0.9133
## Campus.Recruiting1           0.73185    0.61515   1.190   0.2360
## `External Recruiting`1       0.39922    0.84419   0.473   0.6369
## Career.Fair.Recruiting1      1.99199    1.02605   1.941   0.0540 .
## Career.Fair.Recruiting2     -1.00915    2.68912  -0.375   0.7080
## `Hiring Site`1              -2.41734    1.41568  -1.708   0.0897 .
## `Hiring Site`2               0.48126    0.94230   0.511   0.6103
## The.Recruiter1              -1.64607    1.88094  -0.875   0.3828
## The.Recruiter2              -0.56720    1.82682  -0.310   0.7566
## The.Recruiter3              -0.29329    1.83491  -0.160   0.8732
## The.Recruiter4              -1.75149    1.49774  -1.169   0.2440
## The.Recruiter5              -1.33728    1.49345  -0.895   0.3719
## The.Recruiter6              -2.68377    1.55909  -1.721   0.0872 .
## The.Recruiter7              -2.21391    1.74498  -1.269   0.2064
## The.Recruiter8              -3.39966    1.94289  -1.750   0.0821 .
## The.Recruiter9              -1.38755    1.82069  -0.762   0.4471
## College.GPA                  0.01872    0.60708   0.031   0.9754
## School.Type1                 2.35330    1.51065   1.558   0.1213
## School.Type2                 3.01098    1.18156   2.548   0.0118 *
## School.Type3                 1.89022    1.16682   1.620   0.1073
## School.Type6                 1.44006    1.49102   0.966   0.3356
## School.Type7                 0.37319    1.21960   0.306   0.7600
## School.Type8                 0.41061    1.40642   0.292   0.7707
## School.Type9                 1.45948    1.02969   1.417   0.1584
## Citizen1                    -0.82898    1.58327  -0.524   0.6013
## `People7$Training.Practice`  0.47997    0.08567   5.603 9.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.503 on 156 degrees of freedom
## Multiple R-squared:  0.8055, Adjusted R-squared:  0.7619
## F-statistic: 18.46 on 35 and 156 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit)
##
##                      Value   p-value                   Decision
## Global Stat        15.2837 0.0041476 Assumptions NOT satisfied!
## Skewness            3.2991 0.0693175    Assumptions acceptable.
## Kurtosis            0.3294 0.5659899    Assumptions acceptable.
## Link Function      10.9577 0.0009321 Assumptions NOT satisfied!
## Heteroscedasticity  0.6974 0.4036530    Assumptions acceptable.
```

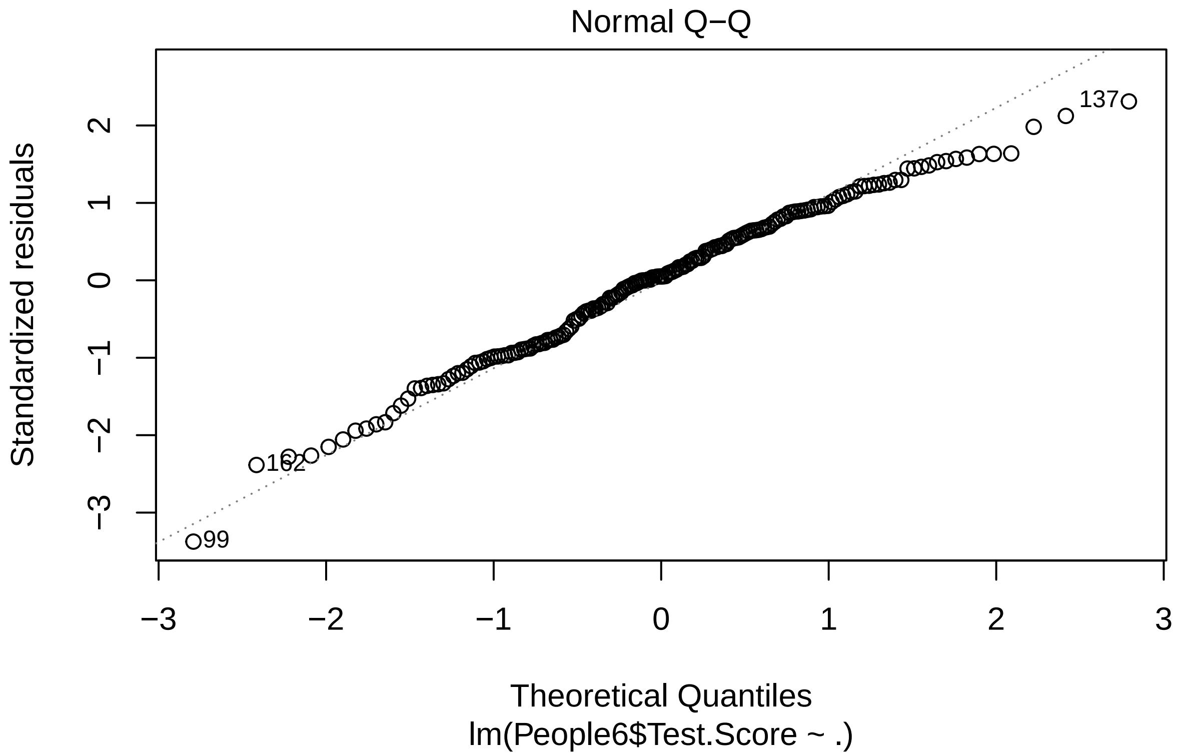
```
class(People6$Test.Score)
```

```
## [1] "numeric"
```

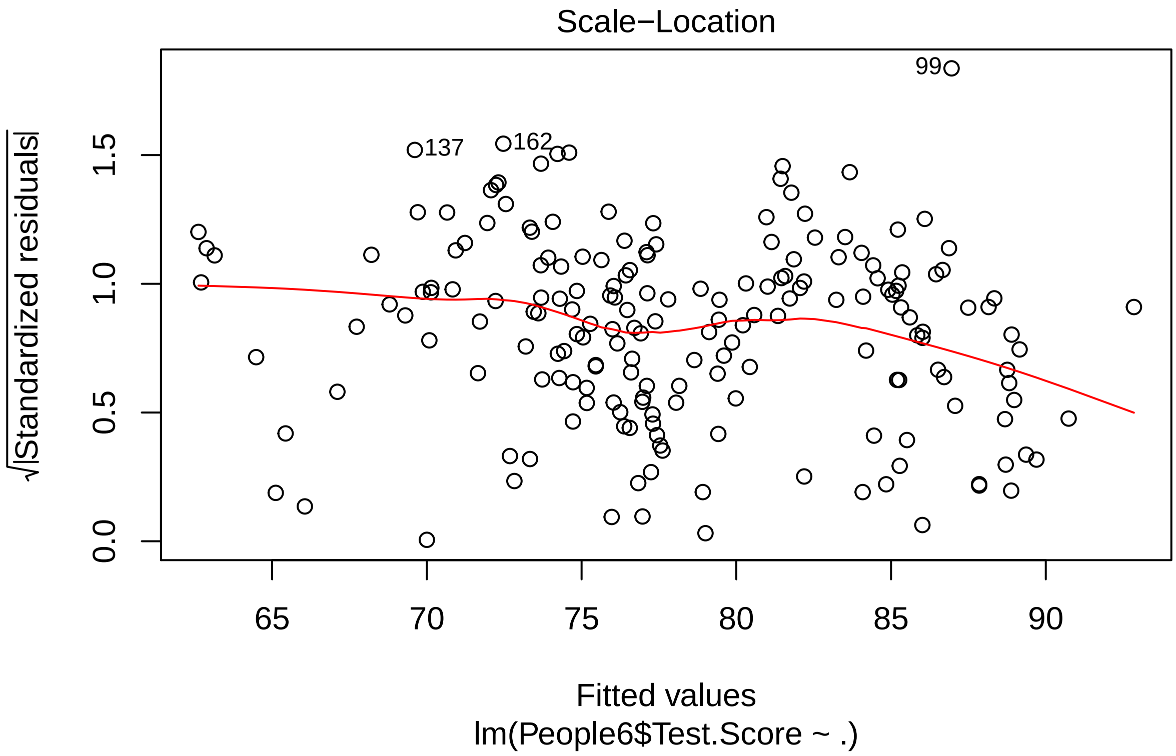
```
plot(fit)
```

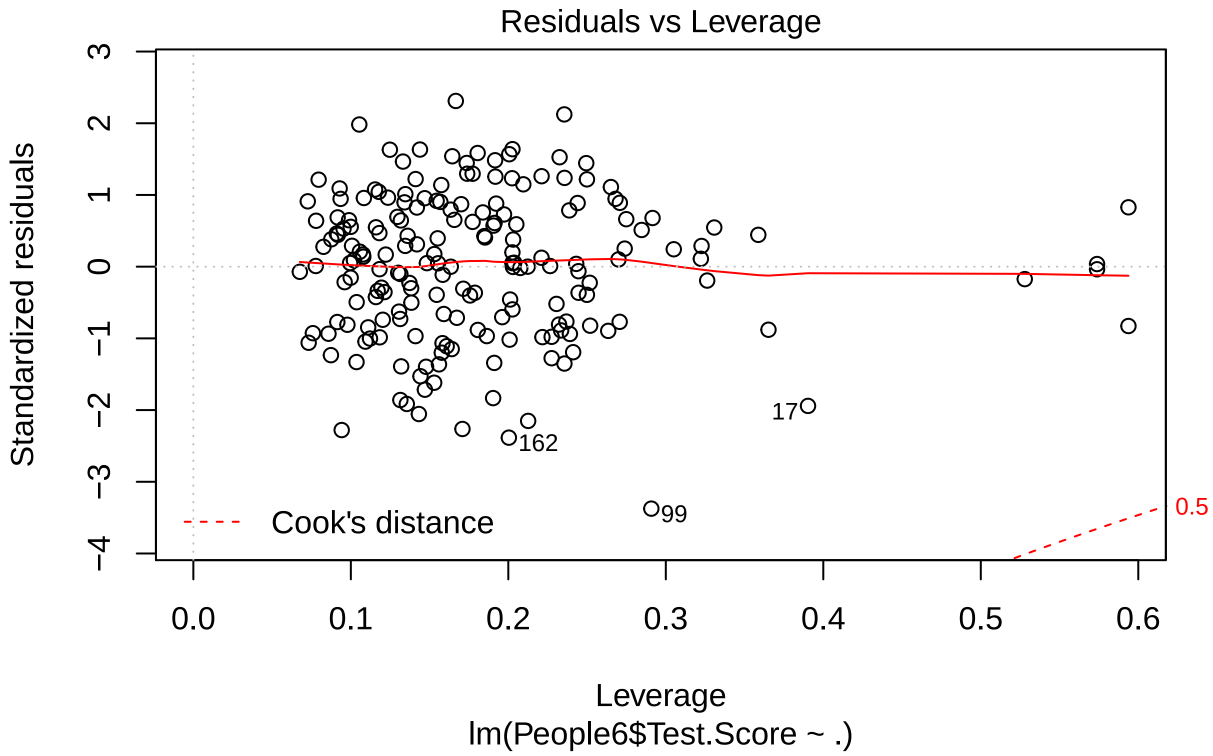
```
## Warning: not plotting observations with leverage one:
##   44
```


Residuals vs Fitted
Residuals
5
0
−5
−10
65
70
75
80
85
90
Fitted values
lm(People6$Test.Score ~ .)
162
152
99

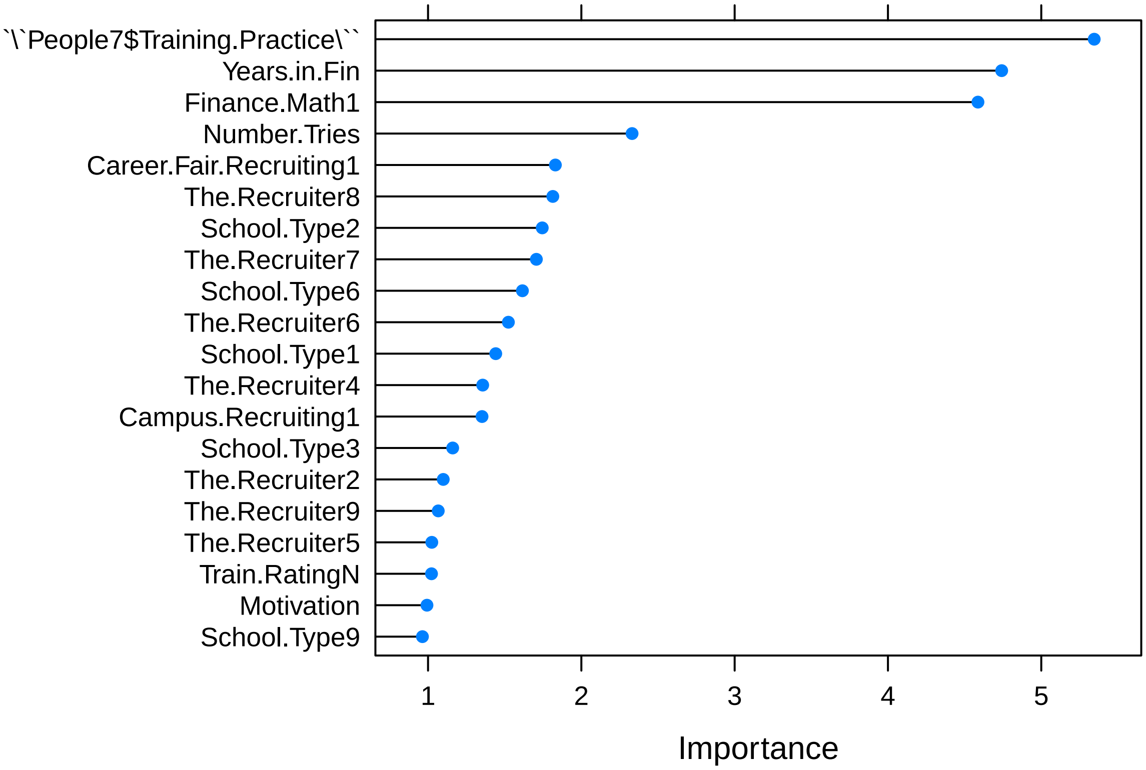


```
## Warning: not plotting observations with leverage one:
##   44
```


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
65
70
75
80
85
90
Fitted values
lm(People6$Test.Score ~ .)
99
137
162



```
### Ploting importance based upon T values for machine learning model

plot(lmImp, top = 20)
```



```
summary(People6$Accredited)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
plot
```

```
## standardGeneric for "plot" defined from package "graphics"
##
## function (x, y, ...)
## standardGeneric("plot")
## <environment: 0x000000001dbfaad0>
## Methods may be defined for arguments: x, y
## Use  showMethods("plot")  for currently available ones.
```

```
### Resampled Root Mean Squared Error

mean(lmFit$resample$RMSE)
```

```
## [1] 4.31444
```

```
###Final Prediction for test Score
## 4.267952 is pretty high

predicted <- predict(lmFit, subTest)
RMSE(pred = predicted, obs = subTest$Test.Score)
```

```
## [1] 4.278157
```

```
###Elastic Regression (added hyperparameters to fix issues related to multicollinearity )


enetGrid <- expand.grid(alpha = seq(0, 2, .1 ),
                        lambda = seq(0, .6, .01 ))

set.seed(1234)  # for reproducibility
enetFit <- train(Test.Score ~ .,
                 data = subTrain,
                 method="glmnet",
                 metric="RMSE",
                 trControl=fitCtrl,
                 tuneGrid=enetGrid)
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

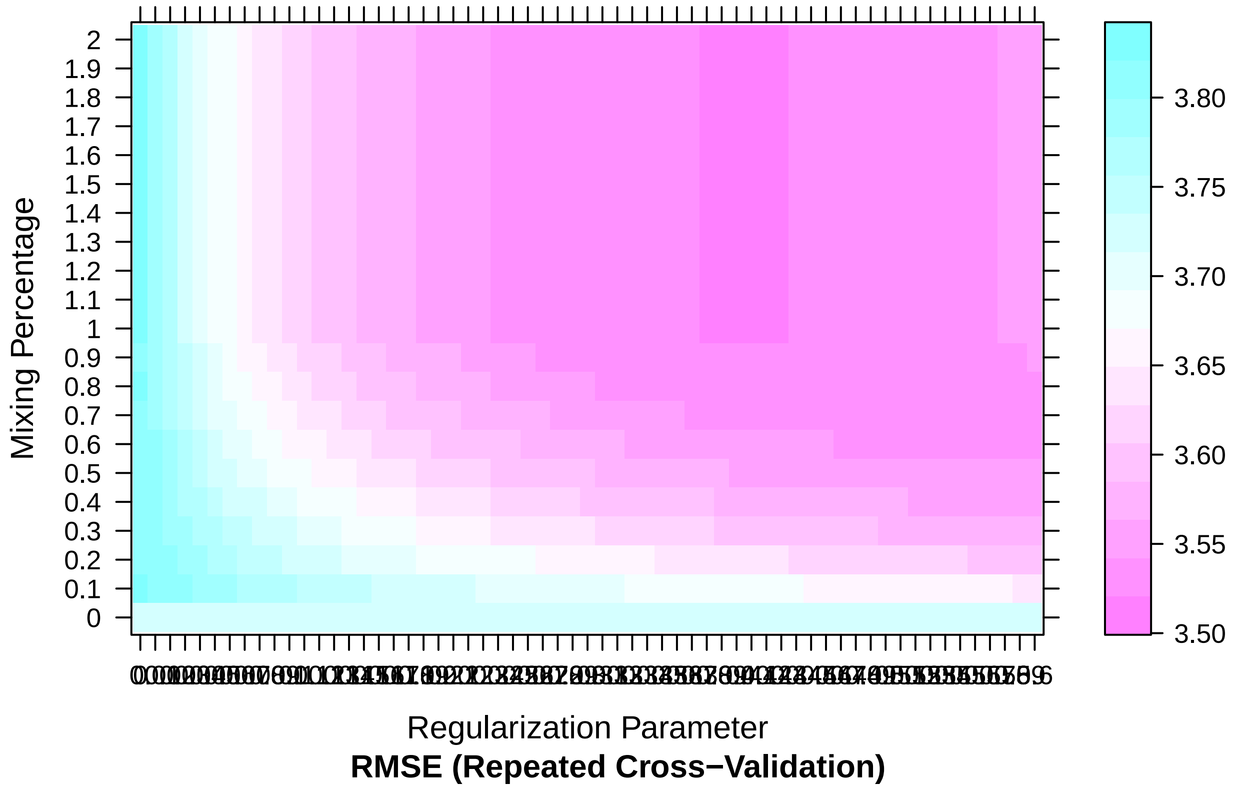
```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

```
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1

## Warning in (function (x, y, family = c("gaussian", "binomial", "poisson", :
## alpha >1; set to 1
```

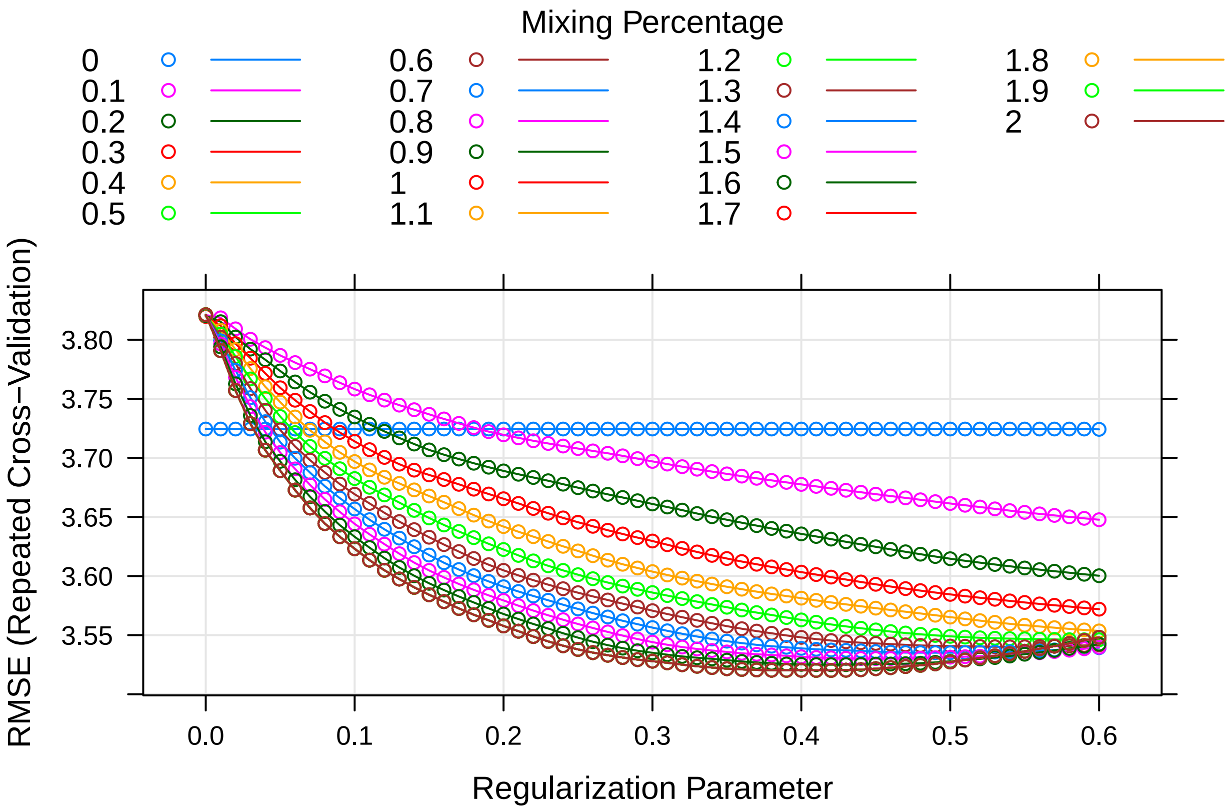
```
print(enetFit$bestTune)
```

```
##     alpha lambda
## 653     1   0.42
```

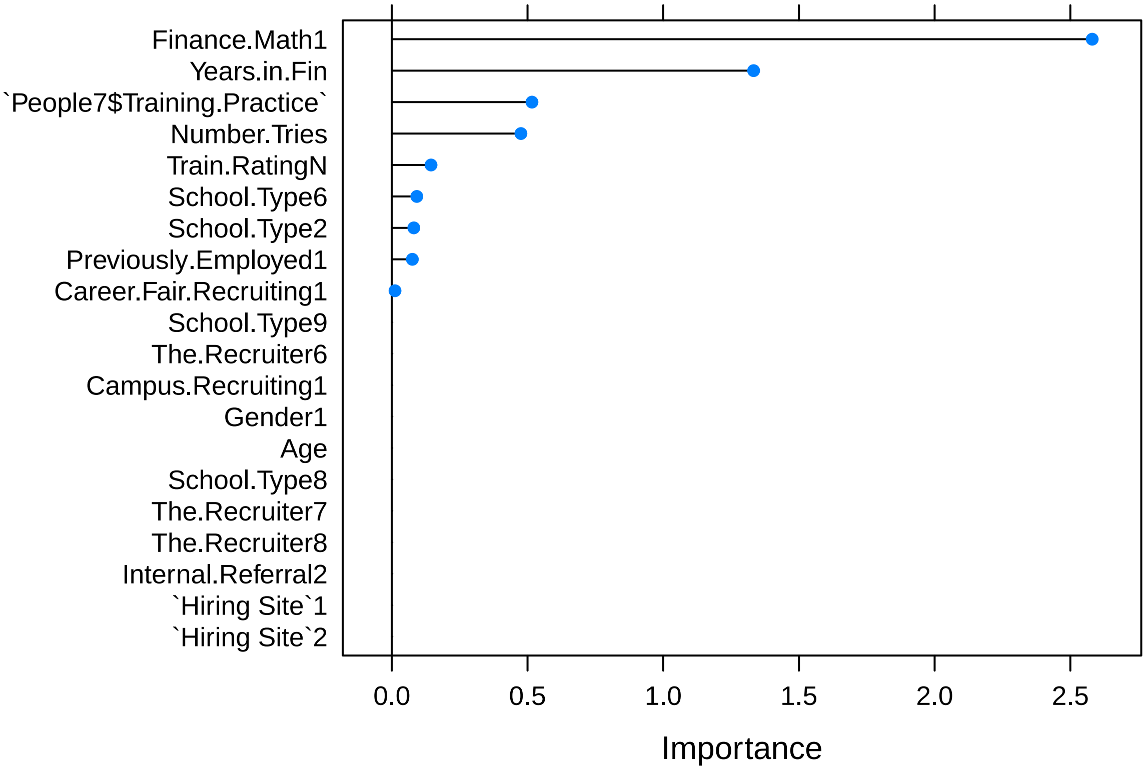
```
### Ploting optimal hyper parameters (tradeoff between bias and variance )
plot(enetFit, plotType = "level")
```



```
plot(enetFit)
```



```
### Variable importance has shifted

enetVarImp <- varImp(enetFit, scale = FALSE)

plot(enetVarImp, top = 20)
```



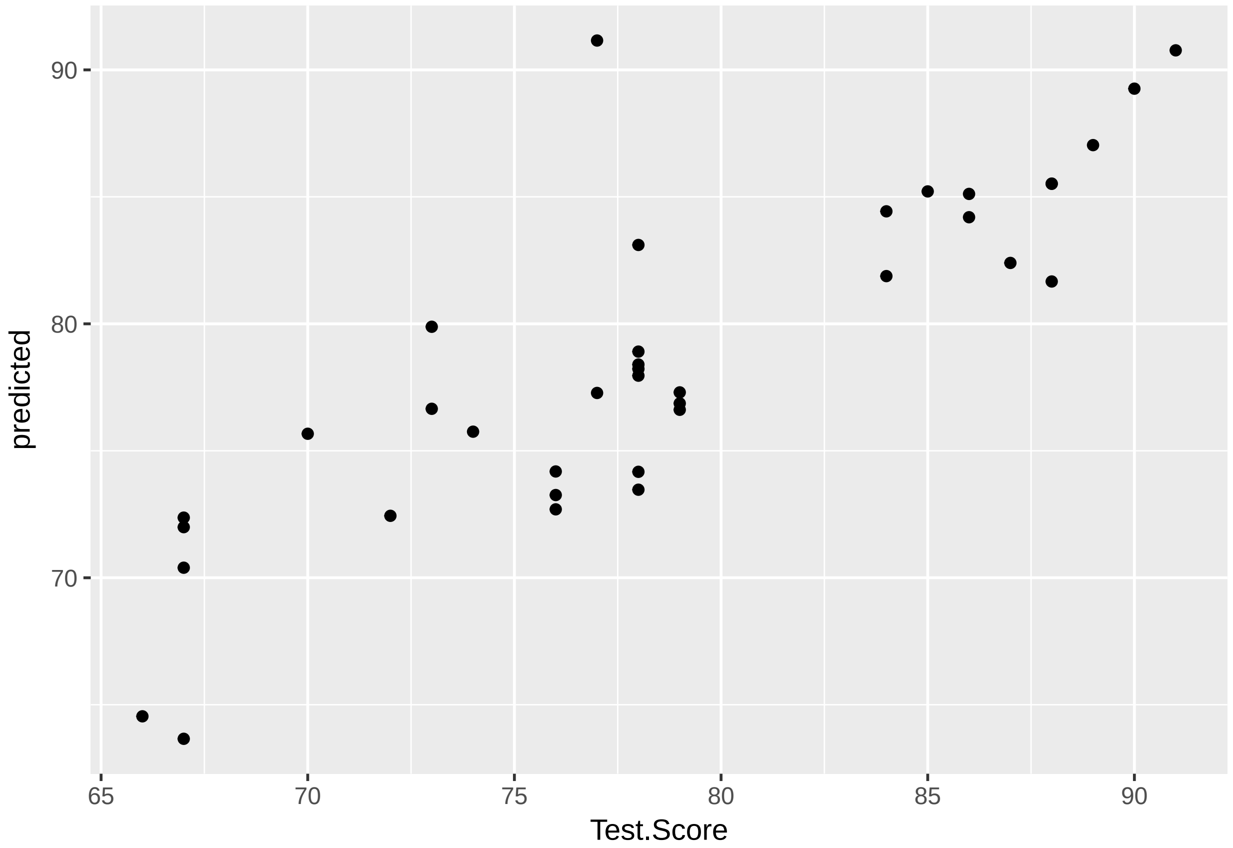
```
mean(enetFit$resample$RMSE)
```

```
## [1] 3.520153
```

```
## Final prediction has improved but still fairly high
predicted <- predict(enetFit, subTest)
RMSE(pred = predicted, obs = subTest$Test.Score)
```

```
## [1] 3.887607
```

```
subTest$predicted <- predict(enetFit, subTest)
ggplot(subTest, aes(x = Test.Score, y = predicted)) + geom_point()
```



```
### Random Forest

fitCtrl <- trainControl(method = "cv",
                        number = 5,
                        verboseIter = TRUE,
                        summaryFunction=defaultSummary)
```

```
### finding the optimal hyperpameter (splitting trees for random forest)

rf_Grid <- expand.grid(mtry = c(1, 10, 20, 50))
```

```
set.seed(678)
RandomFit <- train(Test.Score ~ .,
                data = subTrain,
                method = "rf",
                trControl = fitCtrl,
                tuneGrid= rf_Grid,
                metric='RMSE',
                importance = TRUE,
                maximize=FALSE)
```

```
## Warning: package 'randomForest' was built under R version 3.4.2

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

```
## + Fold1: mtry= 1
## - Fold1: mtry= 1
## + Fold1: mtry=10
## - Fold1: mtry=10
## + Fold1: mtry=20
## - Fold1: mtry=20
## + Fold1: mtry=50
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## - Fold1: mtry=50
## + Fold2: mtry= 1
## - Fold2: mtry= 1
## + Fold2: mtry=10
## - Fold2: mtry=10
## + Fold2: mtry=20
## - Fold2: mtry=20
## + Fold2: mtry=50
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## - Fold2: mtry=50
## + Fold3: mtry= 1
## - Fold3: mtry= 1
## + Fold3: mtry=10
## - Fold3: mtry=10
## + Fold3: mtry=20
## - Fold3: mtry=20
## + Fold3: mtry=50
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## - Fold3: mtry=50
## + Fold4: mtry= 1
## - Fold4: mtry= 1
## + Fold4: mtry=10
## - Fold4: mtry=10
## + Fold4: mtry=20
## - Fold4: mtry=20
## + Fold4: mtry=50
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
## - Fold4: mtry=50
## + Fold5: mtry= 1
## - Fold5: mtry= 1
## + Fold5: mtry=10
## - Fold5: mtry=10
## + Fold5: mtry=20
## - Fold5: mtry=20
## + Fold5: mtry=50
```

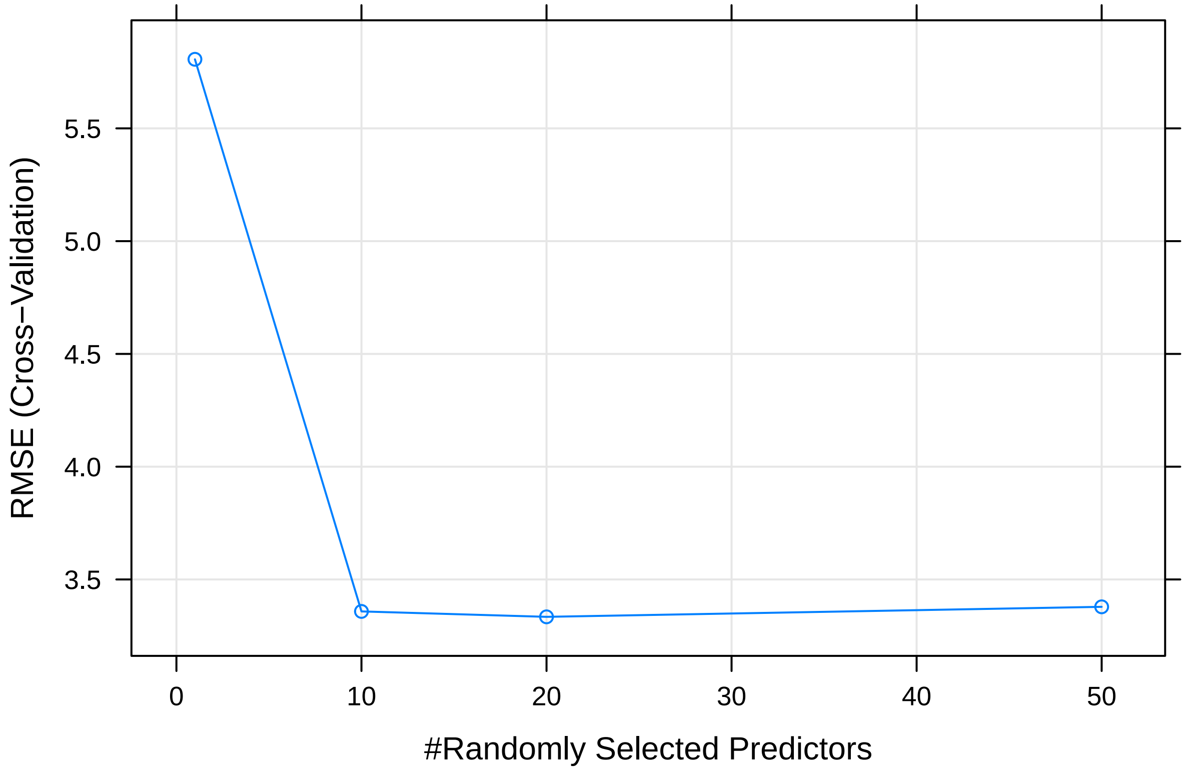
```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## - Fold5: mtry=50
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 20 on full training set
```

```
### Best number of splits is 20

RandomFit$bestTune
```

```
##   mtry
## 3   20
```

```
## viz optimal split (tradeoff between bias and variance in the training set)
plot(RandomFit)
```



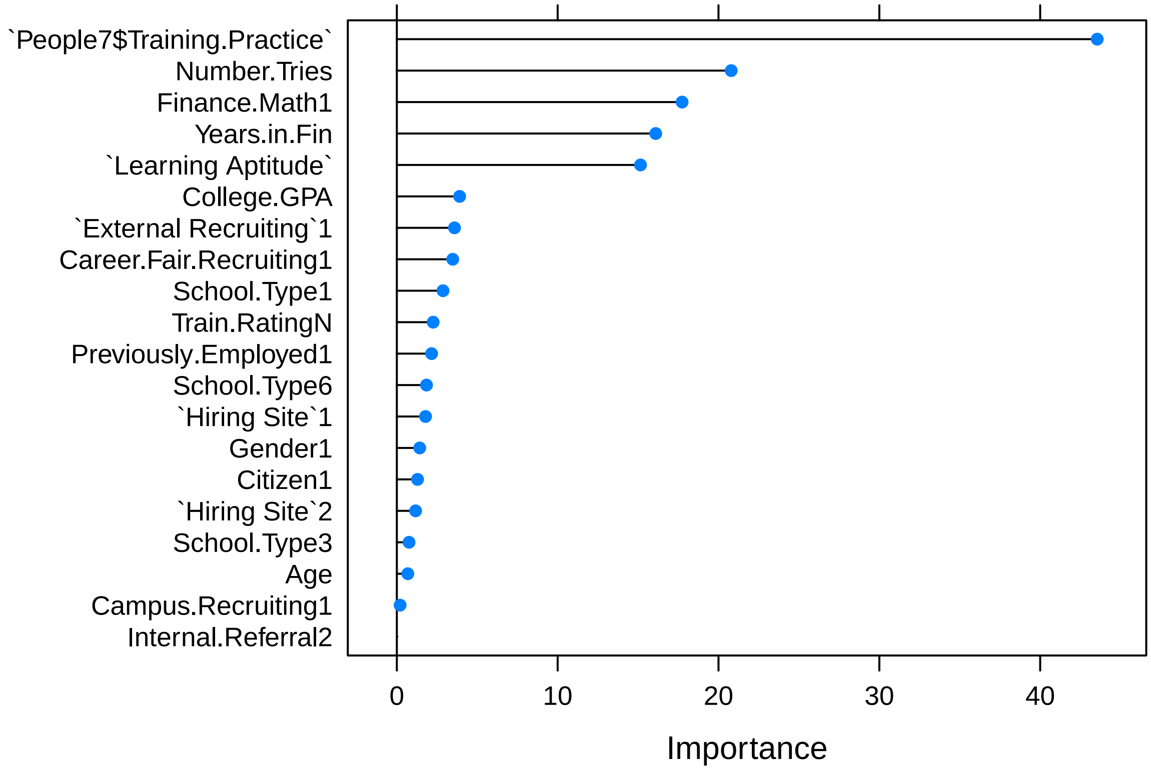
```
mean(RandomFit$resample$RMSE)
```

```
## [1] 3.334225
```

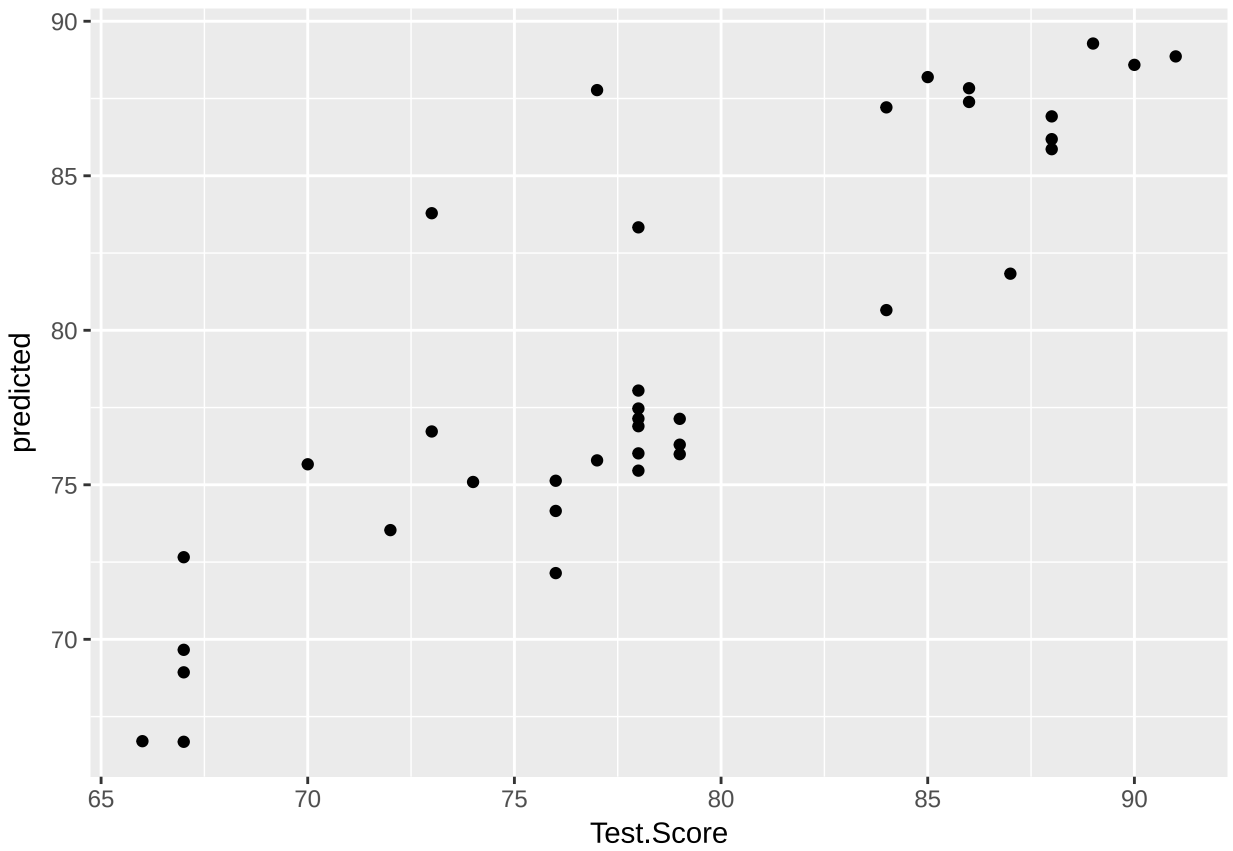
```
###Final and best prediction


predicted <- predict(RandomFit, subTest)
RMSE(pred = predicted, obs = subTest$Test.Score)
```

```
## [1] 3.624769
```

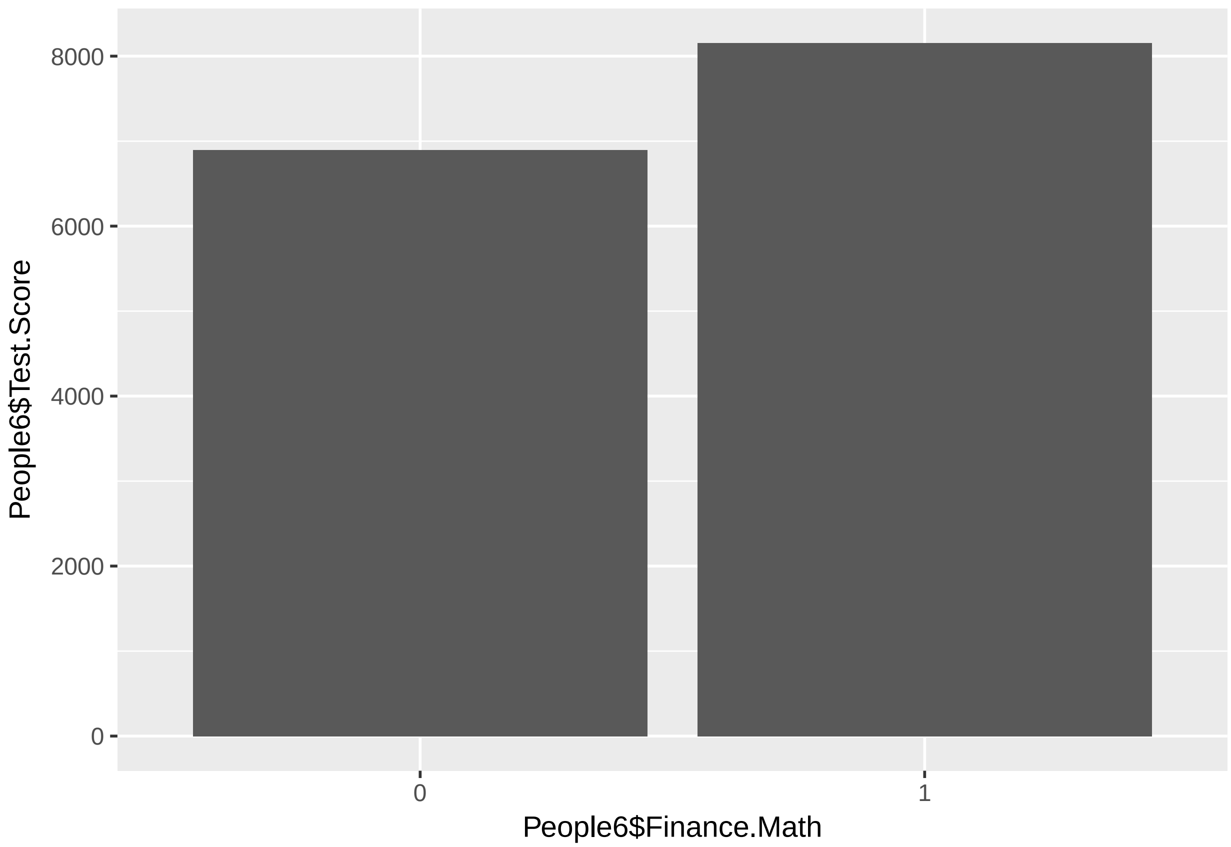
```
###Plotting Variable Importance (Still learning how to interpret)

RandomFitVarImp <- varImp(RandomFit, scale = FALSE)

plot(RandomFitVarImp, top = 20)
```



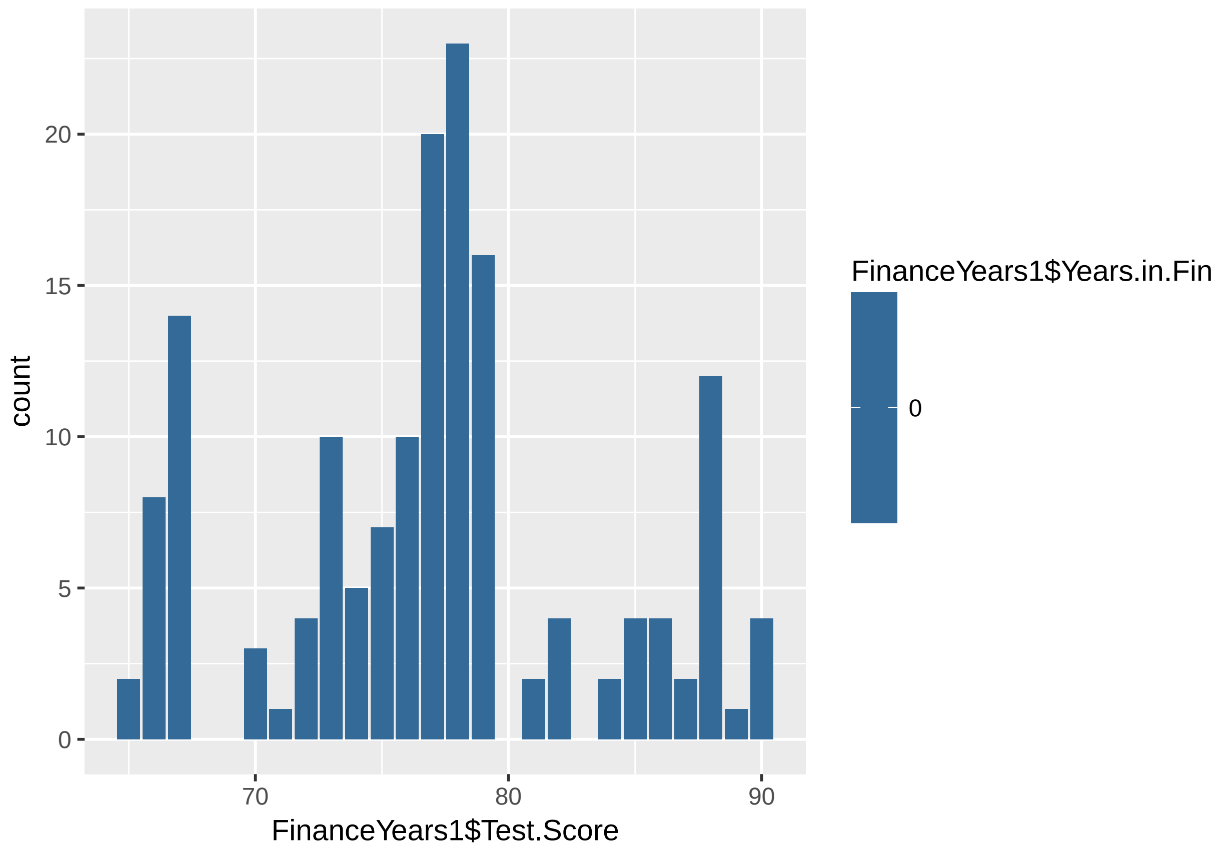
```
###Mapping predictions

subTest$predicted <- predict(RandomFit, subTest)
ggplot(subTest, aes(x = Test.Score, y = predicted)) + geom_point()
```



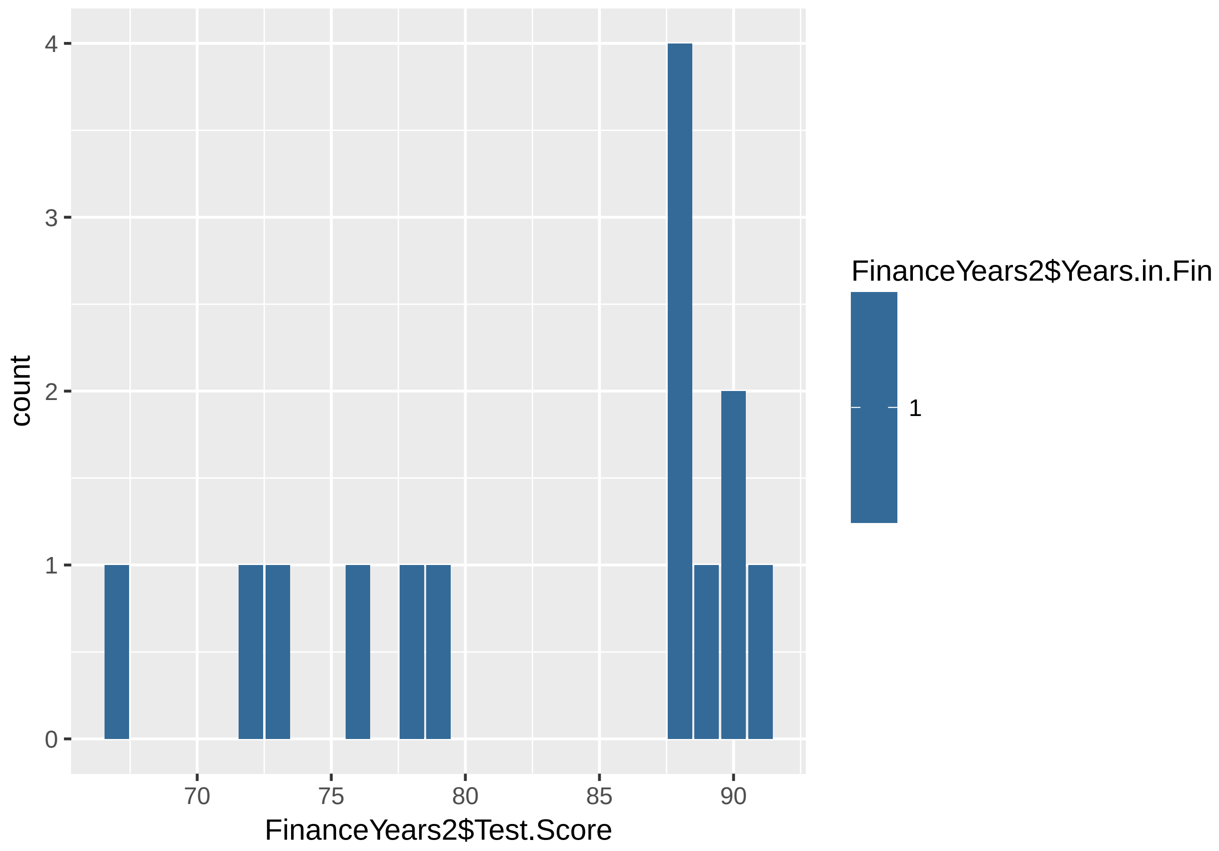
```
###Looking at the difference in scores between math/finance majors and those who didn't study it.
##looks like there is a difference


ggplot(People6, aes(x=People6$Finance.Math,y = People6$Test.Score)) +geom_bar(stat = "identity")
```



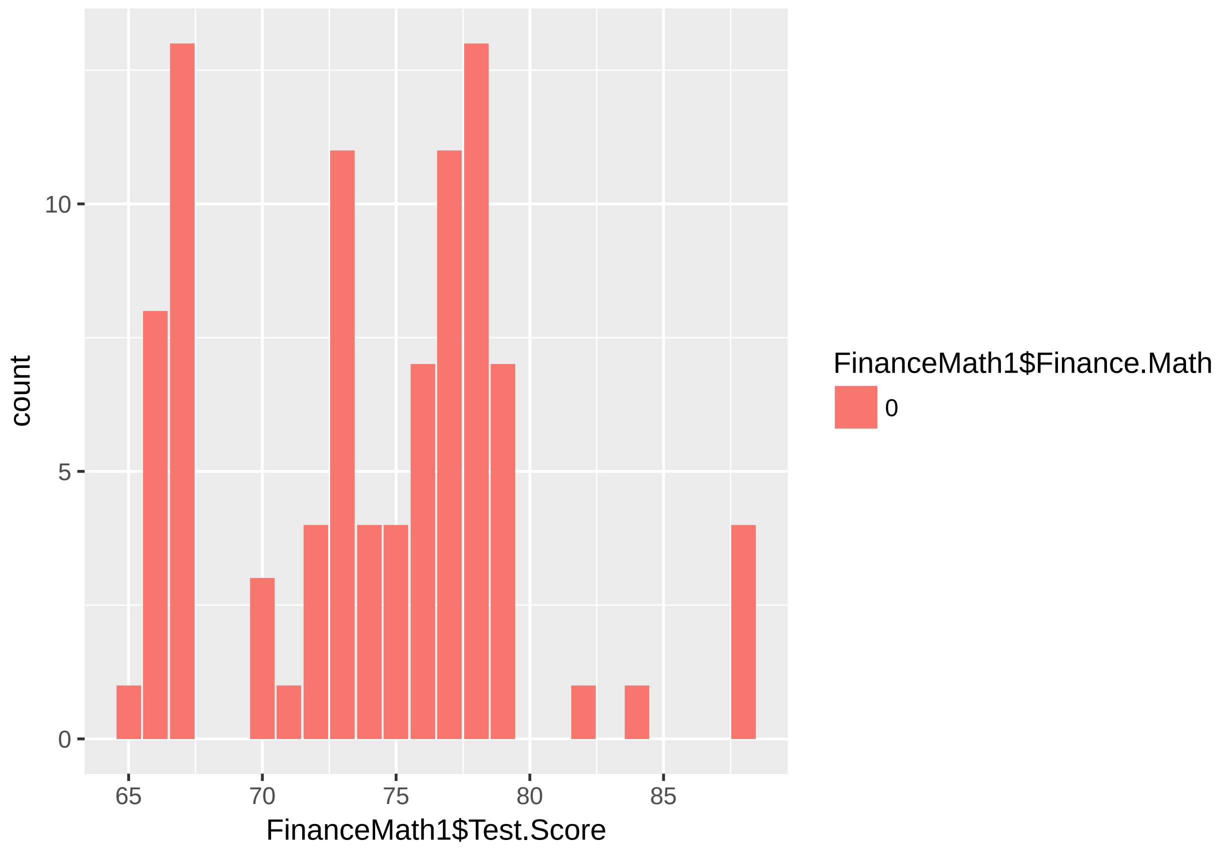
```
FinanceYears1 = dplyr::filter(People6, People6$Years.in.Fin == 0)

FinanceYears2 = dplyr::filter(People6, People6$Years.in.Fin == 1)


ggplot(data = FinanceYears1, aes(x = FinanceYears1$Test.Score))+
                  geom_bar(aes(fill = FinanceYears1$Years.in.Fin))
```



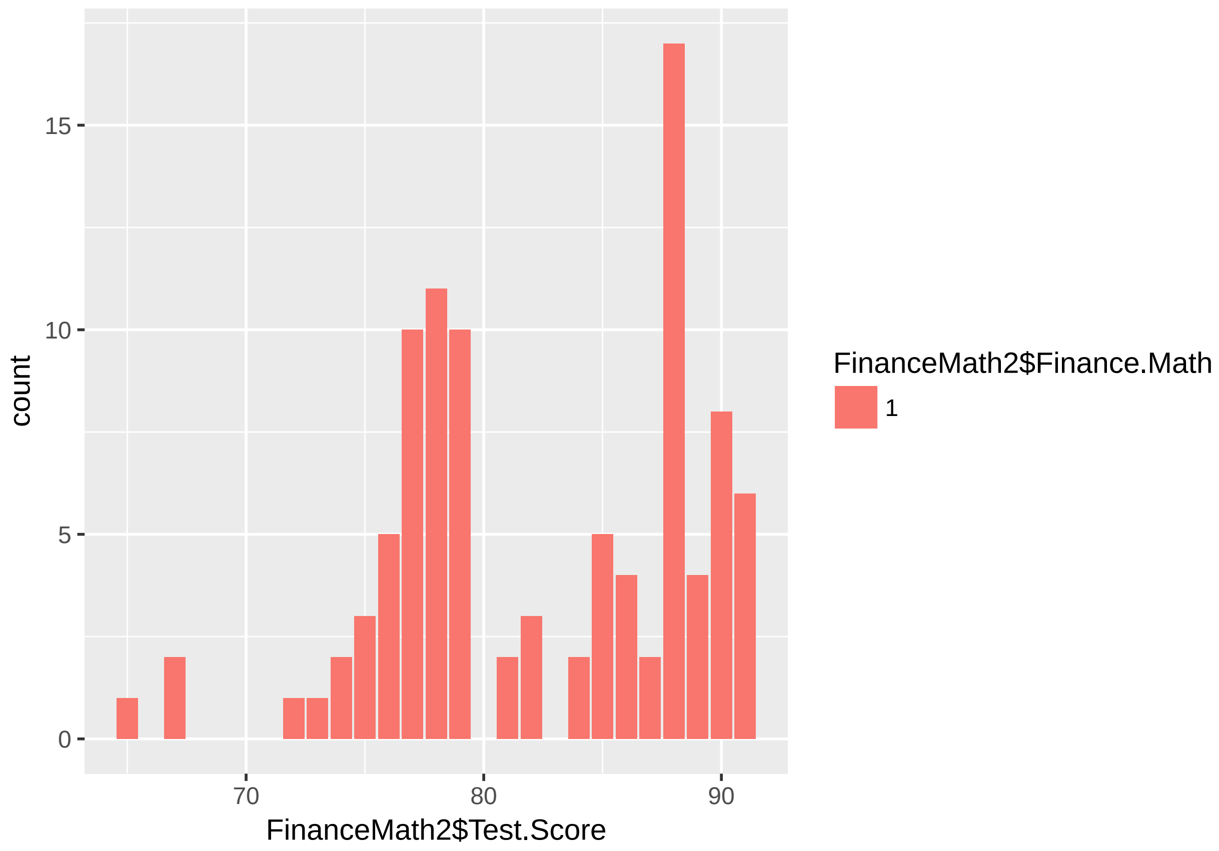
```r
ggplot(data = FinanceYears2, aes(x = FinanceYears2$Test.Score))+
                  geom_bar(aes(fill = FinanceYears2$Years.in.Fin))
```



```
FinanceMath1 = dplyr::filter(People6, People6$Finance.Math == 0)

FinanceMath2 = dplyr::filter(People6, People6$Finance.Math == 1)



ggplot(data = FinanceMath1, aes(x = FinanceMath1$Test.Score))+
geom_bar(aes(fill = FinanceMath1$Finance.Math))
```

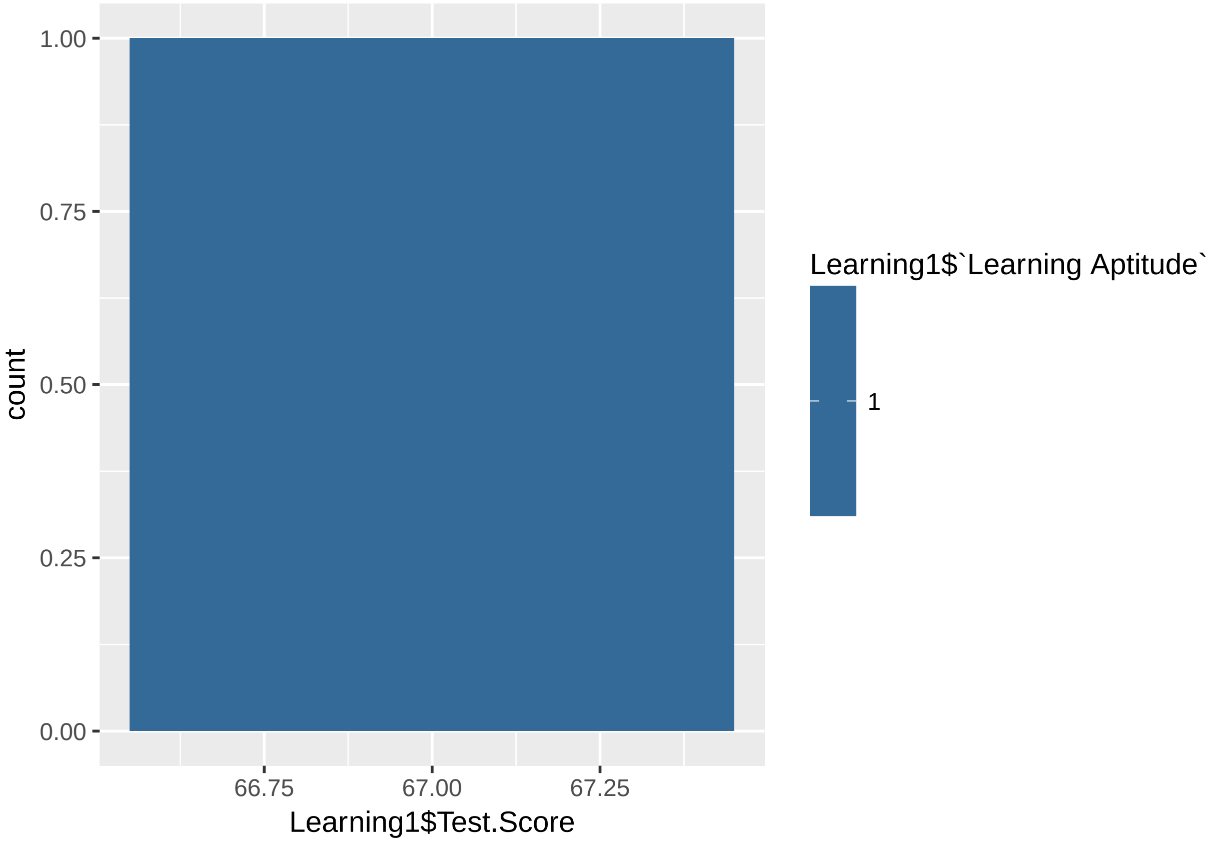


```
ggplot(data = FinanceMath2, aes(x = FinanceMath2$Test.Score))+
geom_bar(aes(fill = FinanceMath2$Finance.Math))
```

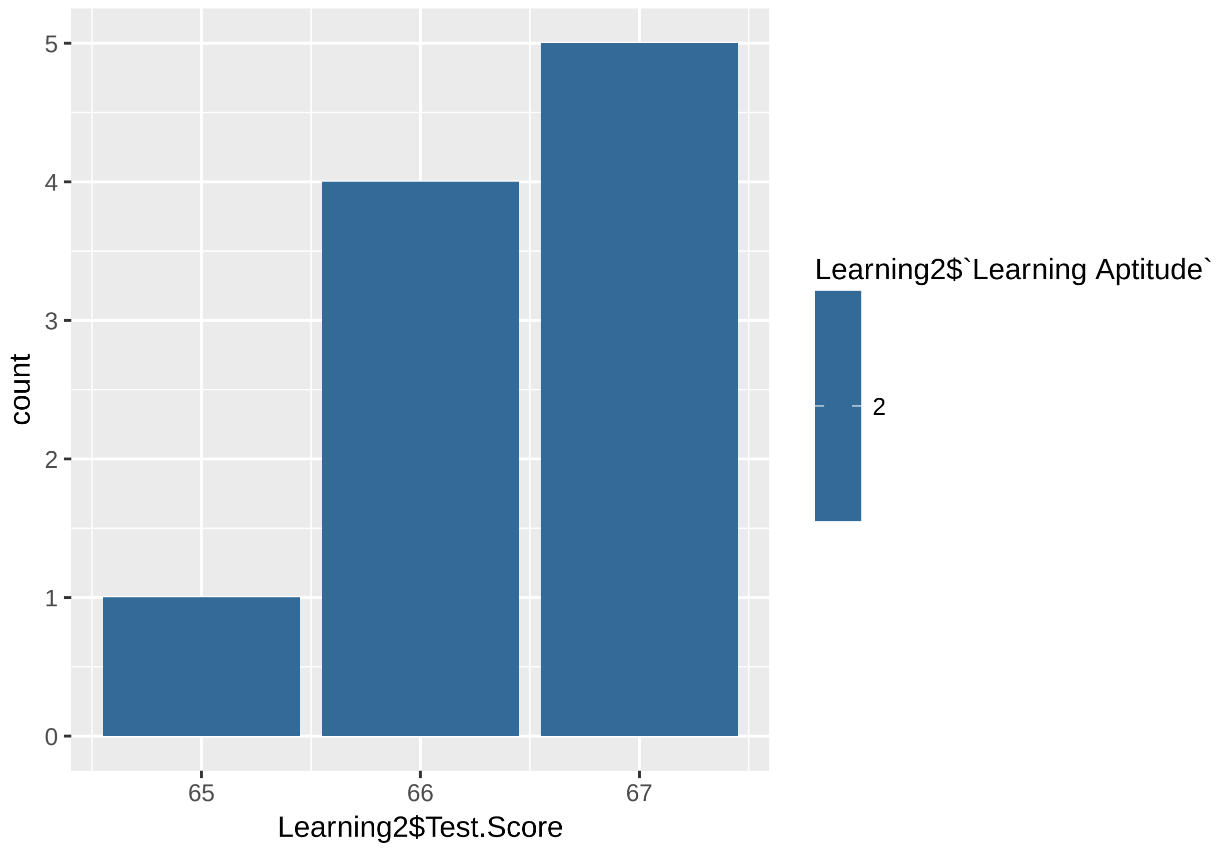


```
# 0 = didn't study finance/math in college
```

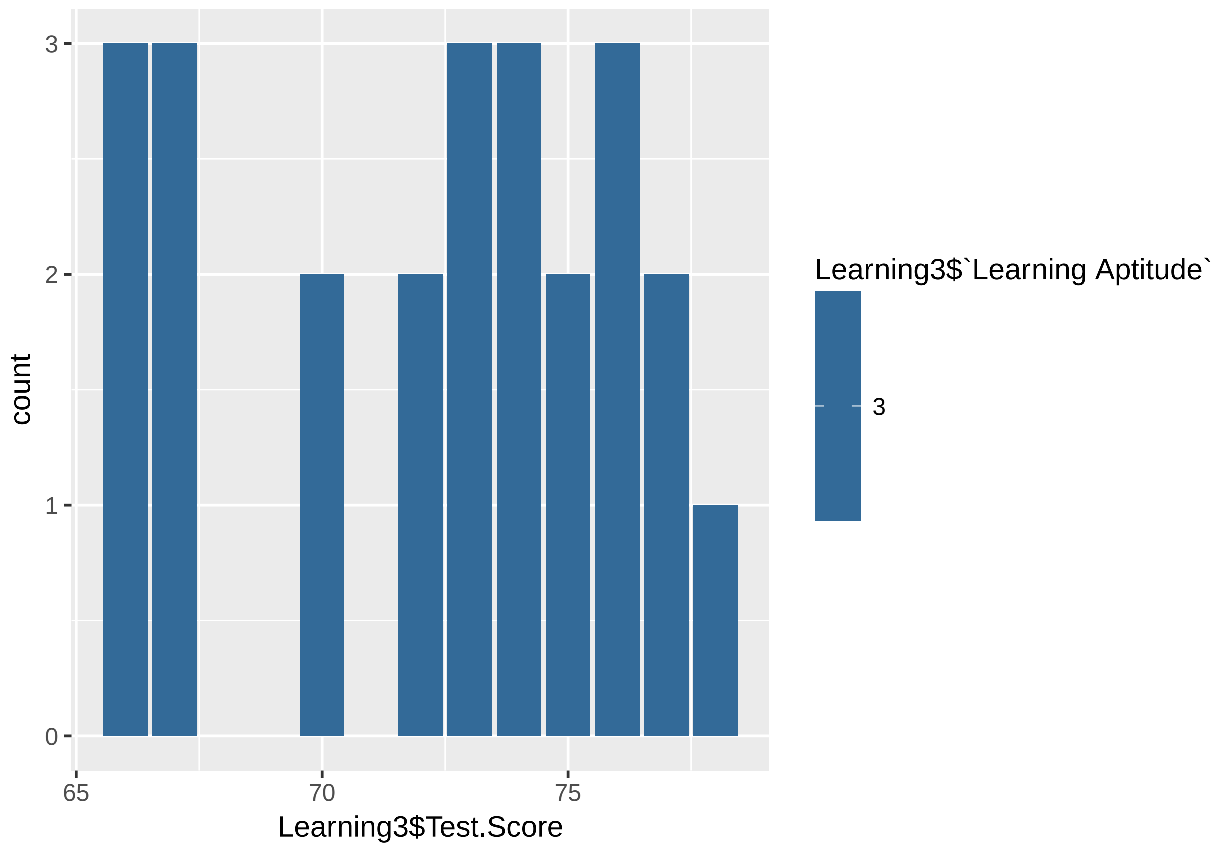
```
Learning1 = dplyr::filter(People6, People6$`Learning Aptitude` == 1)

Learning2 = dplyr::filter(People6, People6$`Learning Aptitude` == 2)


Learning3 = dplyr::filter(People6, People6$`Learning Aptitude` == 3)



Learning4 = dplyr::filter(People6, People6$`Learning Aptitude` == 4)


Learning5 = dplyr::filter(People6, People6$`Learning Aptitude` == 5)


Learning6 = dplyr::filter(People6, People6$`Learning Aptitude` == 6)



ggplot(data = Learning1, aes(x = Learning1$Test.Score))+
geom_bar(aes(fill = Learning1$`Learning Aptitude`))
```



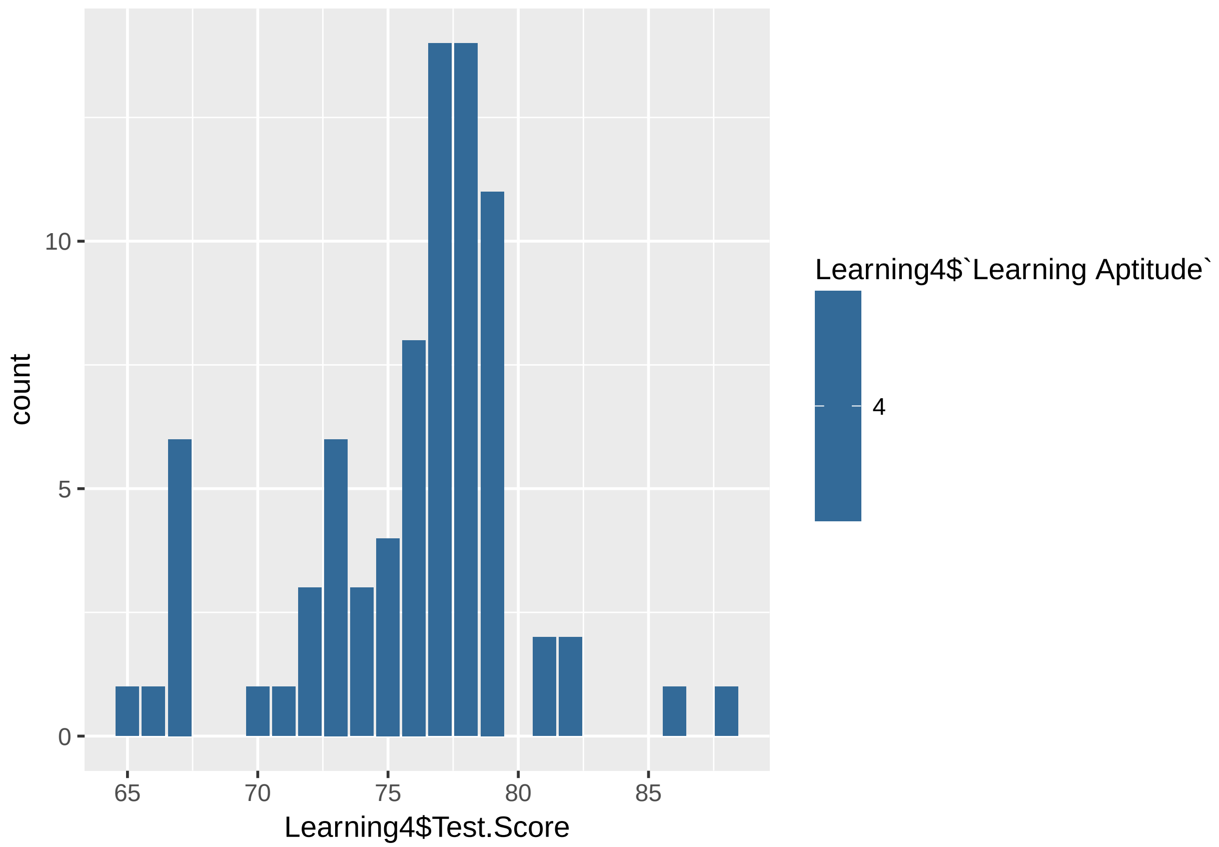
```
ggplot(data = Learning2, aes(x = Learning2$Test.Score))+
geom_bar(aes(fill = Learning2$`Learning Aptitude`))
```



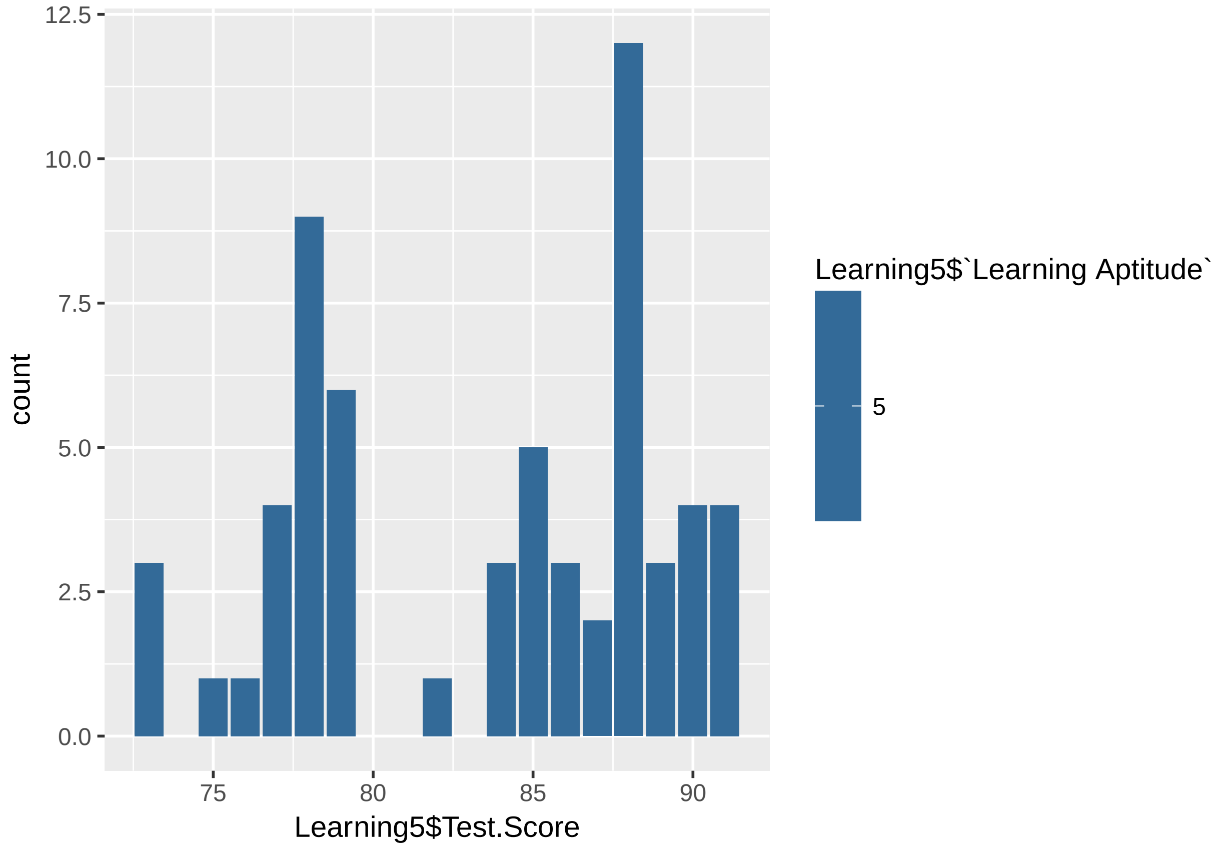
```
ggplot(data = Learning3, aes(x = Learning3$Test.Score))+
geom_bar(aes(fill = Learning3$`Learning Aptitude`))
```



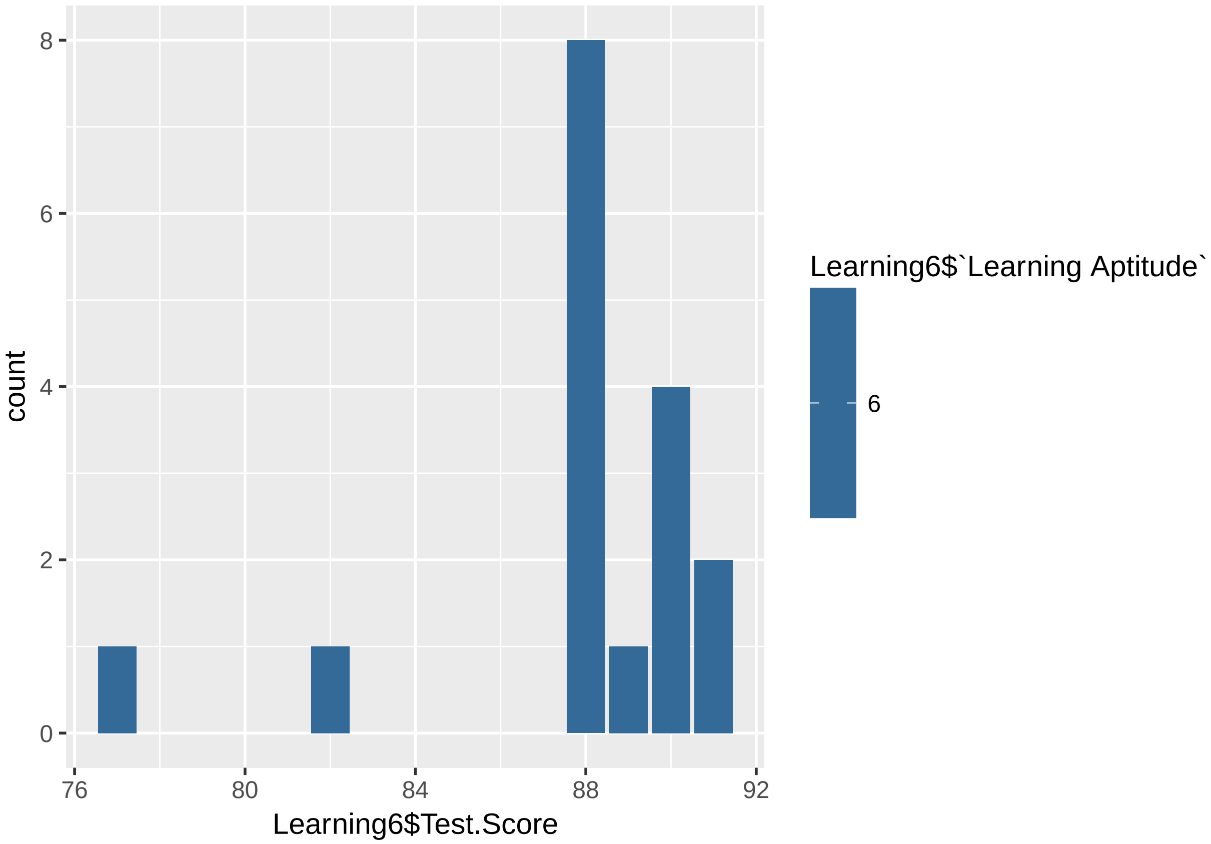
```
ggplot(data = Learning4, aes(x = Learning4$Test.Score))+
geom_bar(aes(fill = Learning4$`Learning Aptitude`))
```



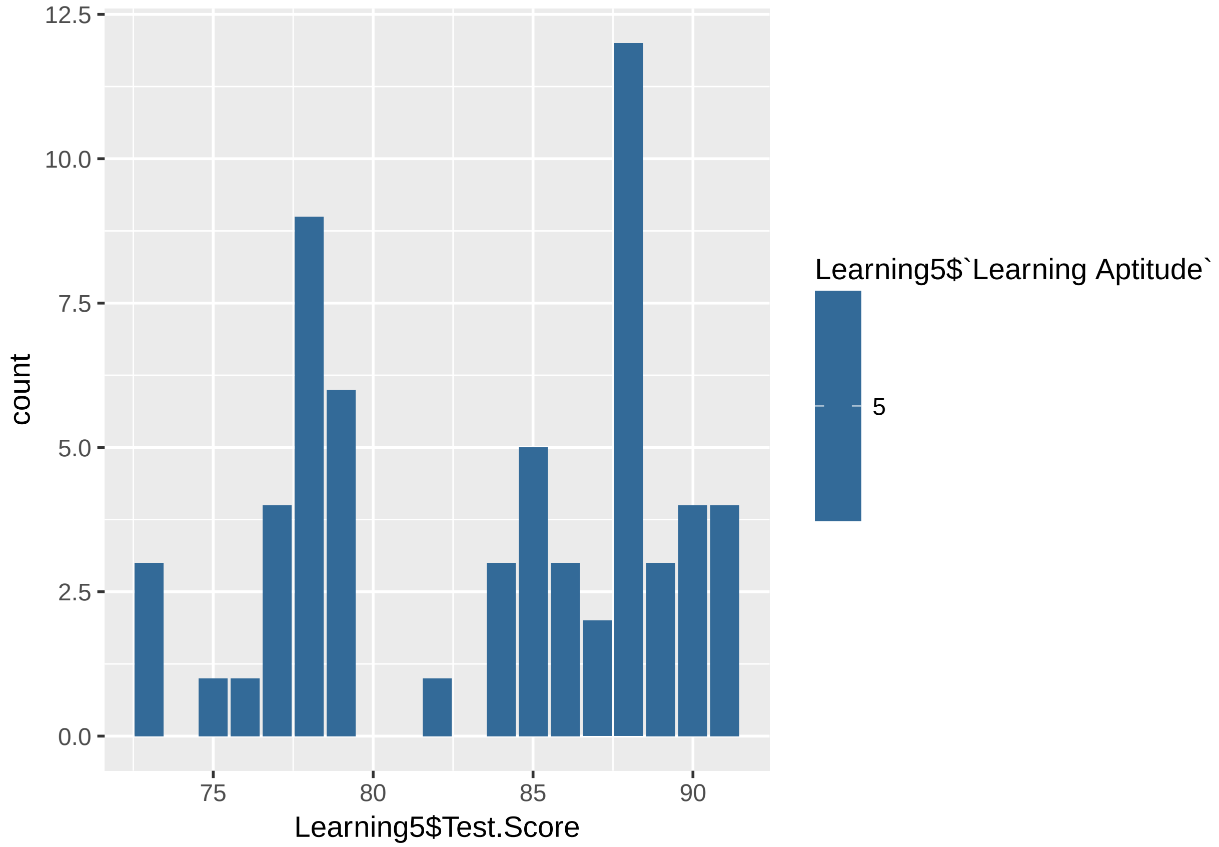
```
ggplot(data = Learning5, aes(x = Learning5$Test.Score))+
geom_bar(aes(fill = Learning5$`Learning Aptitude`))
```



```
ggplot(data = Learning6, aes(x = Learning6$Test.Score))+
geom_bar(aes(fill = Learning6$`Learning Aptitude`))
```

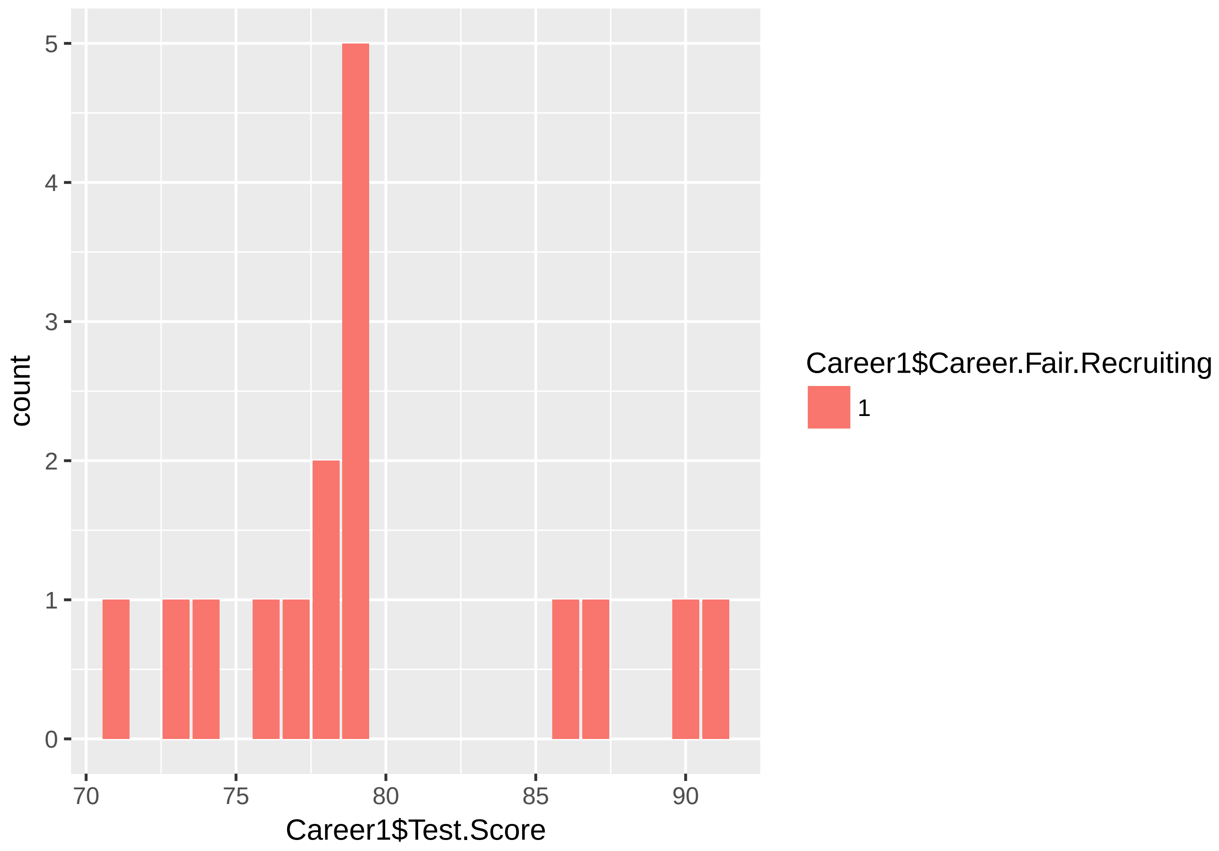


```
ggplot(data = Learning5, aes(x = Learning5$Test.Score))+
geom_bar(aes(fill = Learning5$`Learning Aptitude`))
```

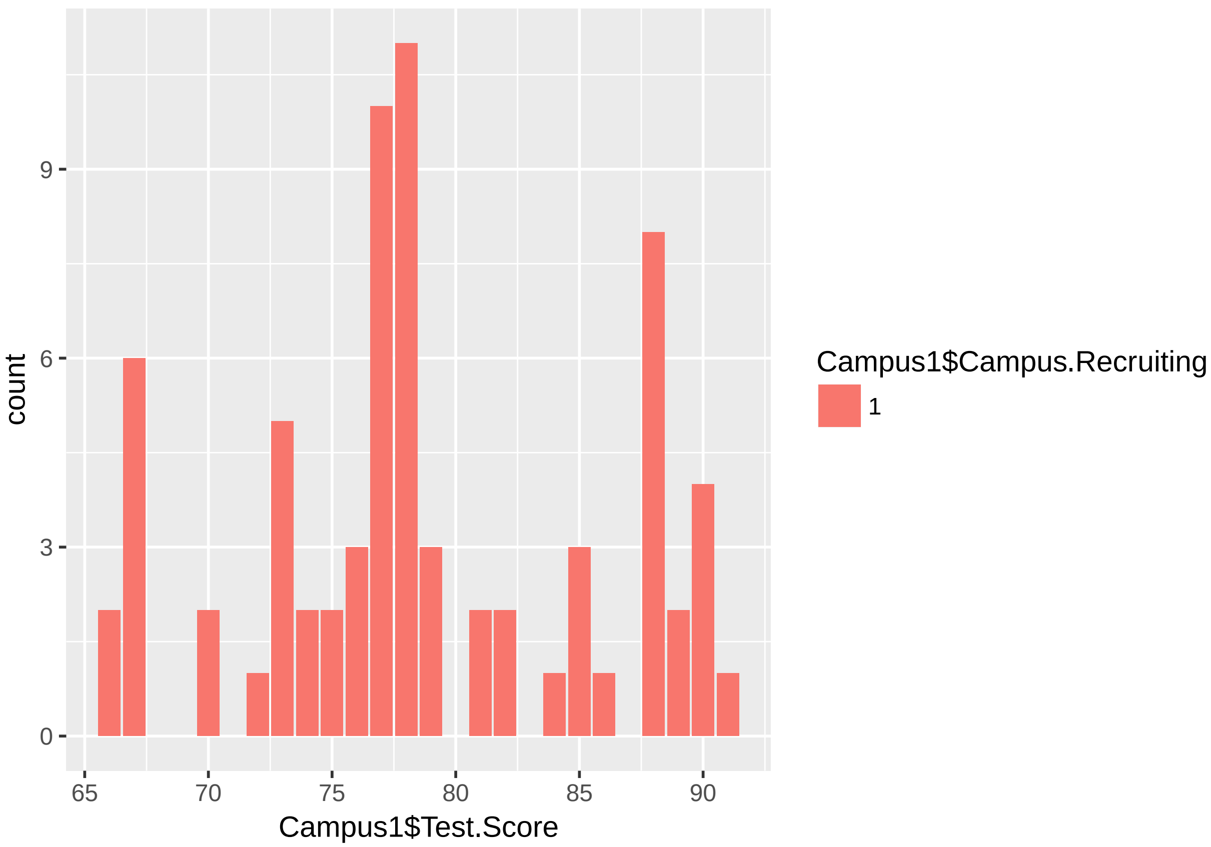


```r
View(People6)
```

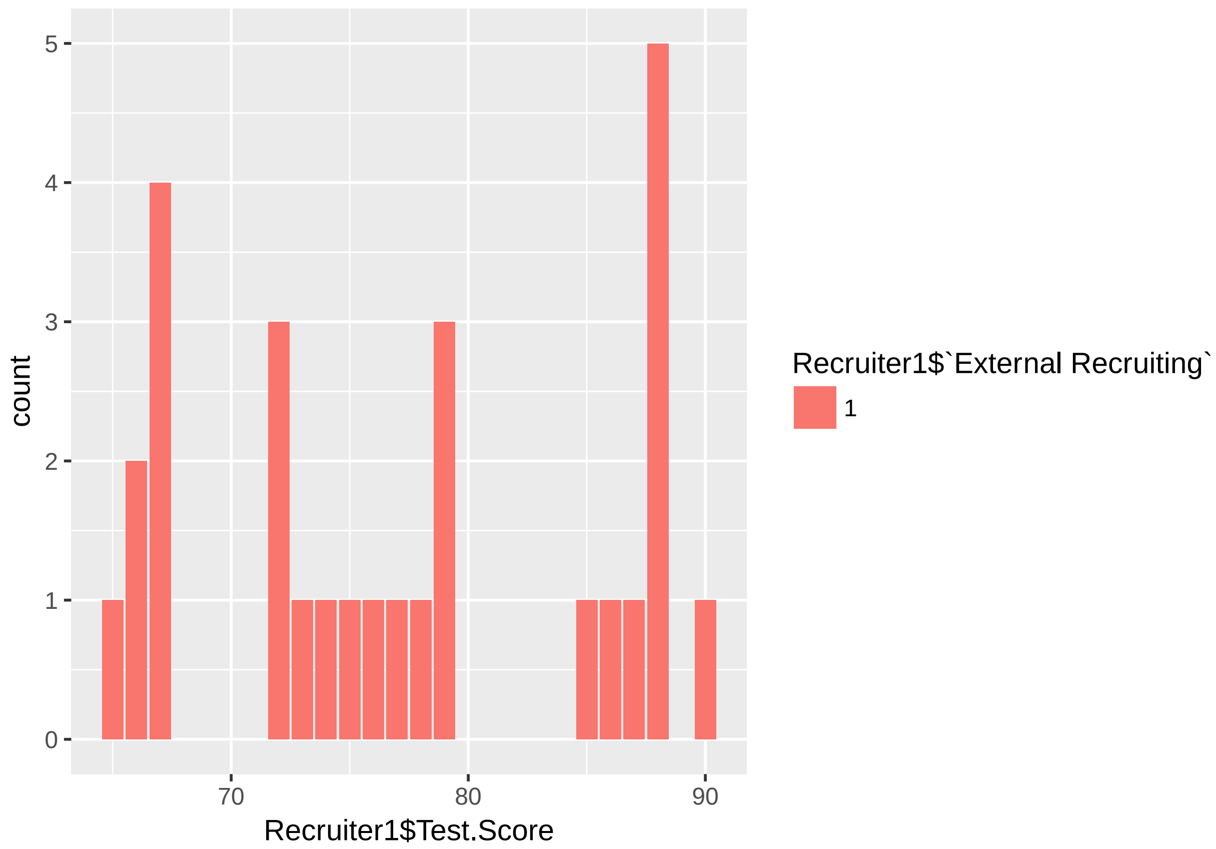
```r
Career1 = dplyr::filter(People6, People6$Career.Fair.Recruiting == 1)

Campus1 = dplyr::filter(People6, People6$Campus.Recruiting == 1)

Recruiter1 = dplyr::filter(People6, People6$`External Recruiting` == 1)








ggplot(data = Career1 , aes(x = Career1$Test.Score))+
  geom_bar(aes(fill = Career1$Career.Fair.Recruiting))
```



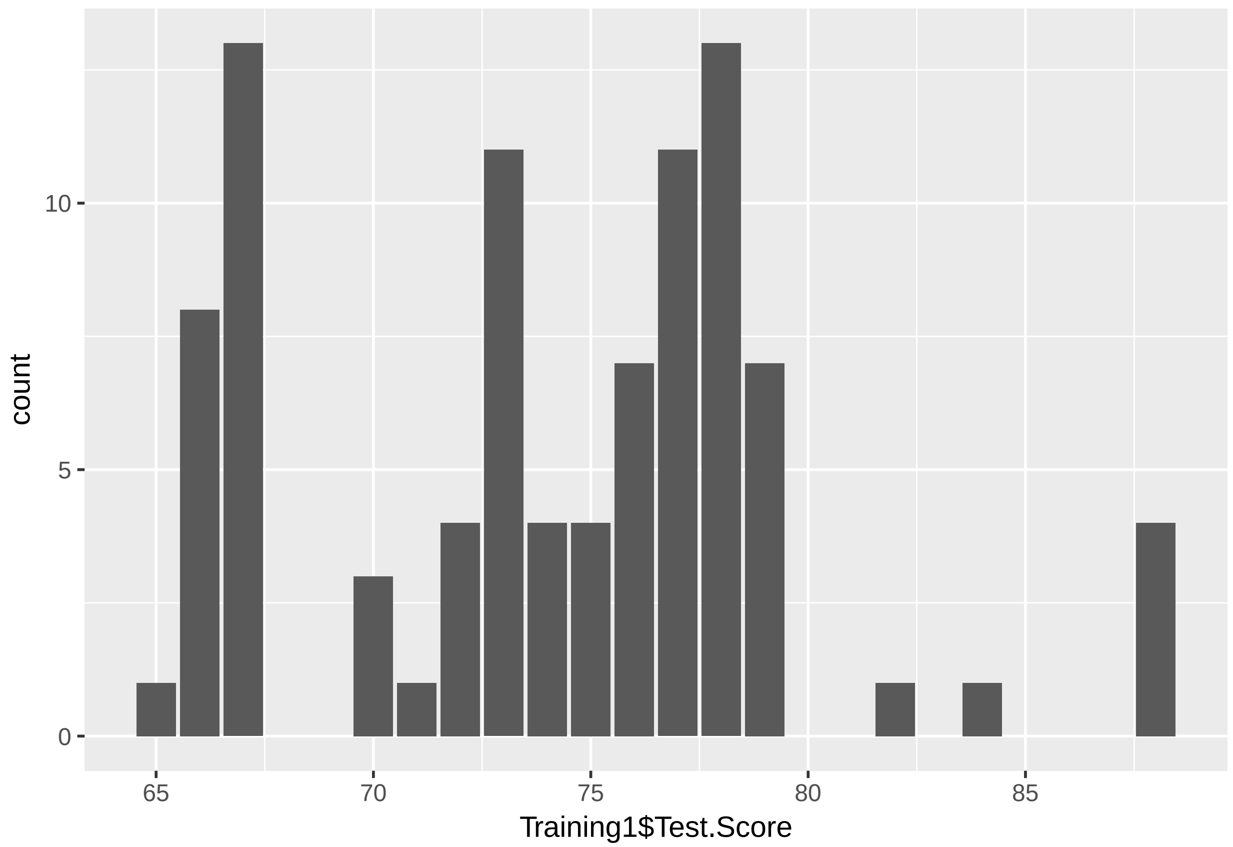
```
ggplot(data = Campus1, aes(x = Campus1$Test.Score))+
geom_bar(aes(fill = Campus1$Campus.Recruiting))
```



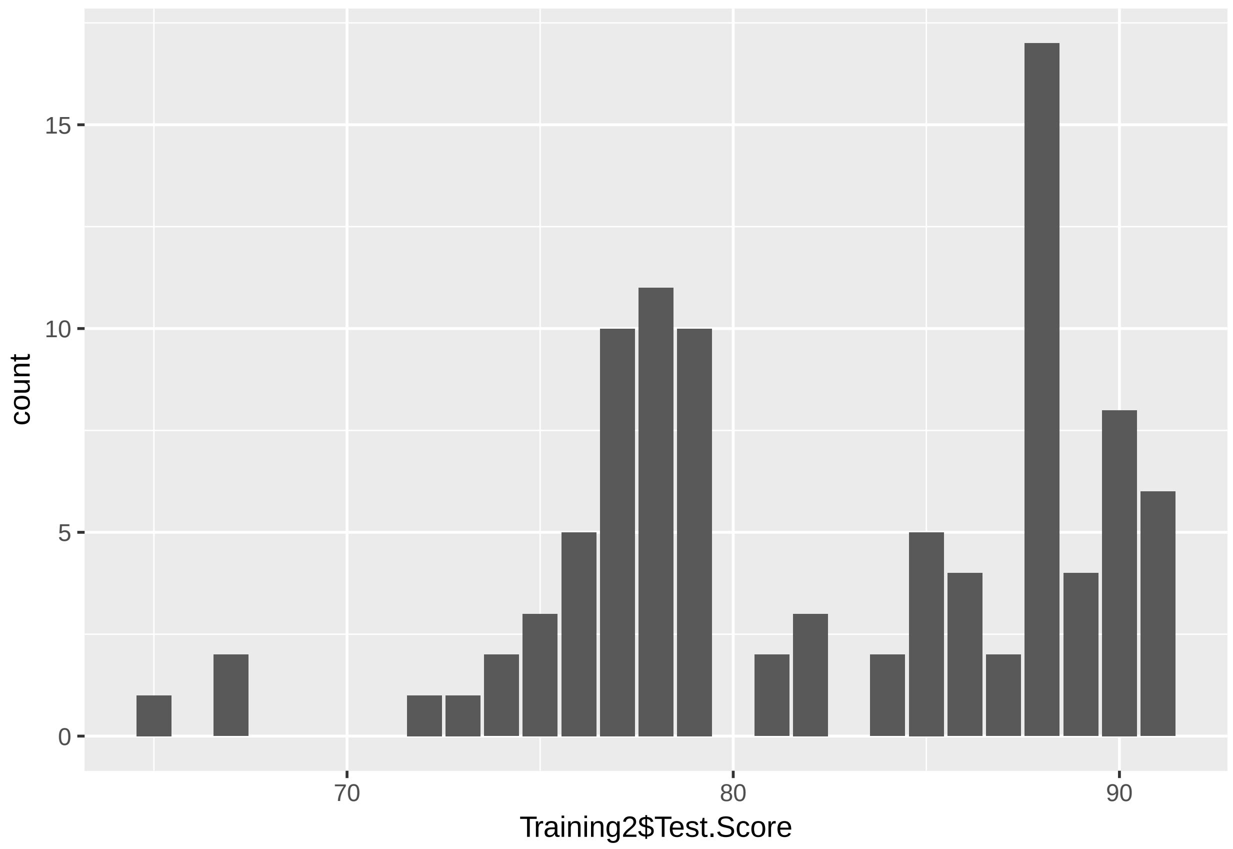
```
ggplot(data = Recruiter1, aes(x = Recruiter1$Test.Score))+
geom_bar(aes(fill = Recruiter1$`External Recruiting`))
```



```
Training1 = dplyr::filter(People6, People6$Finance.Math == 0)

Training2 = dplyr::filter(People6, People6$Finance.Math == 1)



ggplot(data = Training1, aes(x = Training1$Test.Score))+
geom_bar(aes(fill = Training1$`People7$Training.Practice`))
```

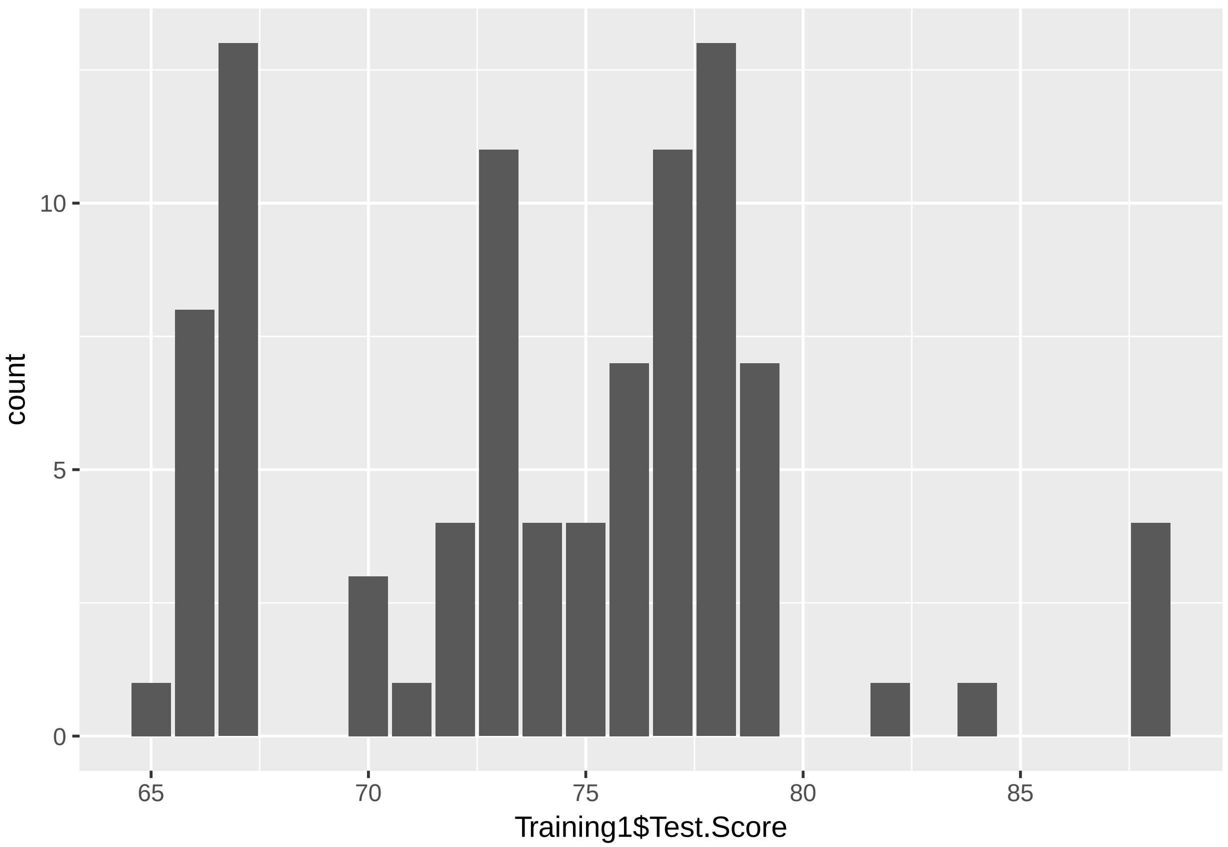


```
ggplot(data = Training2, aes(x = Training2$Test.Score))+
geom_bar(aes(fill = Training2$`People7$Training.Practice`))
```

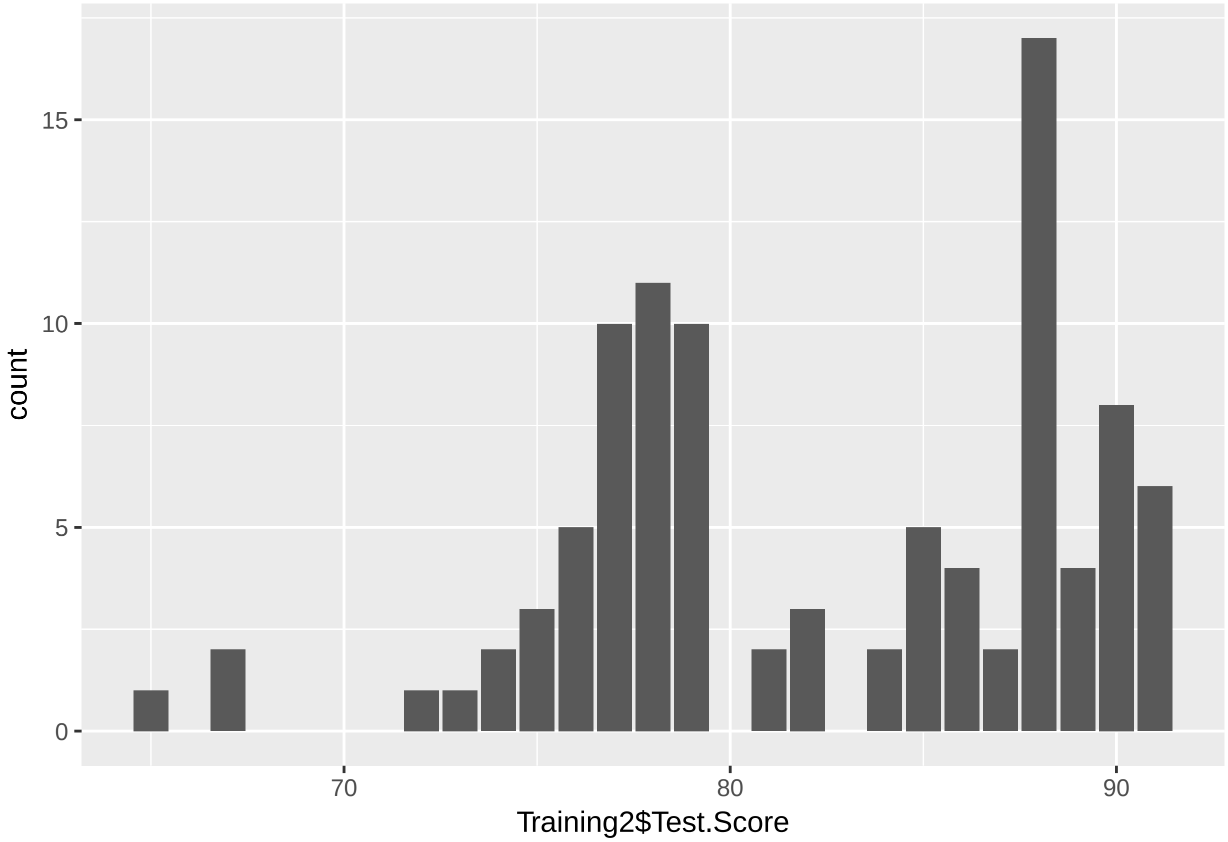


```
Learning1 = dplyr::filter(People6, People6$`Learning Aptitude` == 1)

Learning2 = dplyr::filter(People6, People6$`Learning Aptitude` == 2)

Learning3 = dplyr::filter(People6, People6$`Learning Aptitude` == 3)
```

```
ggplot(data = Training1, aes(x = Training1$Test.Score))+
geom_bar(aes(fill = Training1$`People7$Training.Practice`))
```



```
ggplot(data = Training2, aes(x = Training2$Test.Score))+
geom_bar(aes(fill = Training2$`People7$Training.Practice`))
```

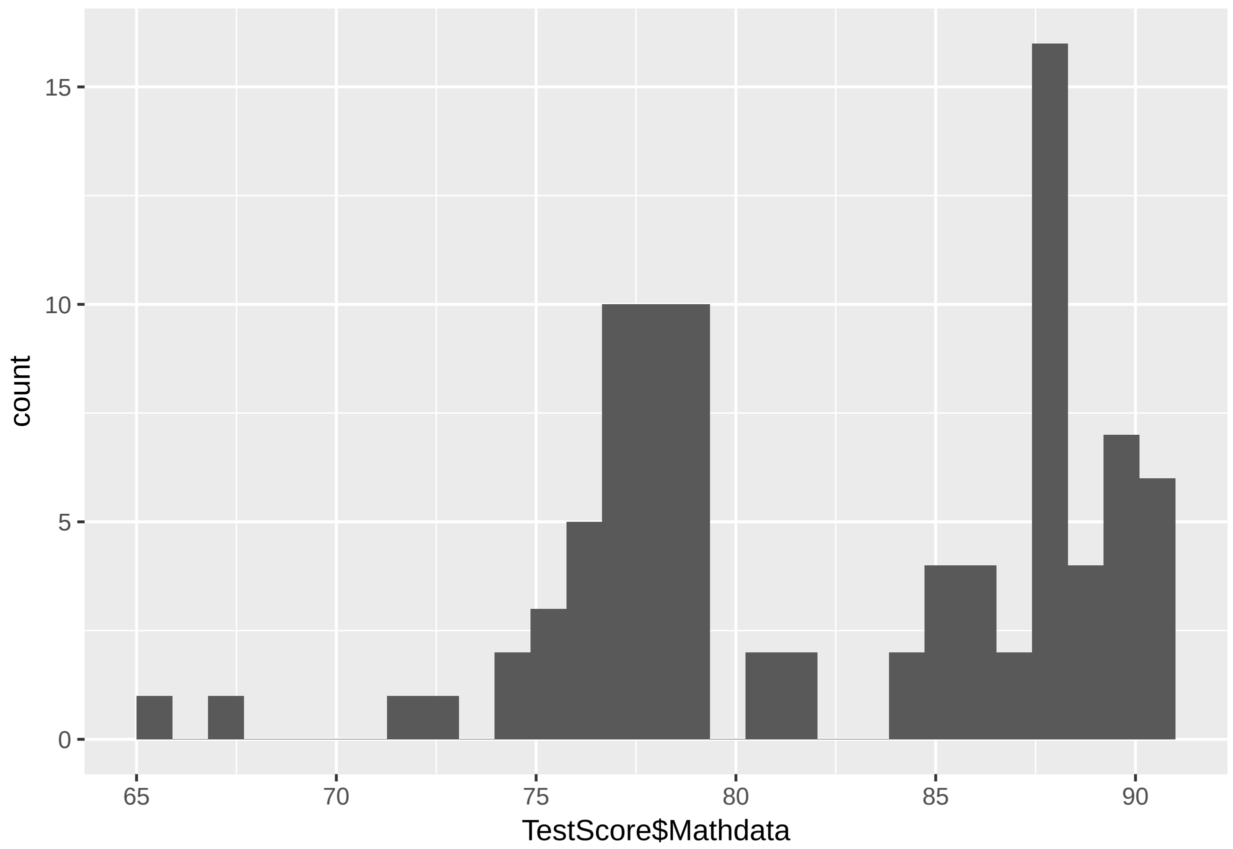


```
###Testing the differences between scores between people who had a math/finance background and those whe


Mathdata = dplyr::filter(People6, People6$Finance.Math == 1)

Mathdata2 = dplyr::filter(People6, People6$Finance.Math == 0)

summary(Mathdata$Finance.Math)
```

```
##  0  1
##  0 99
```

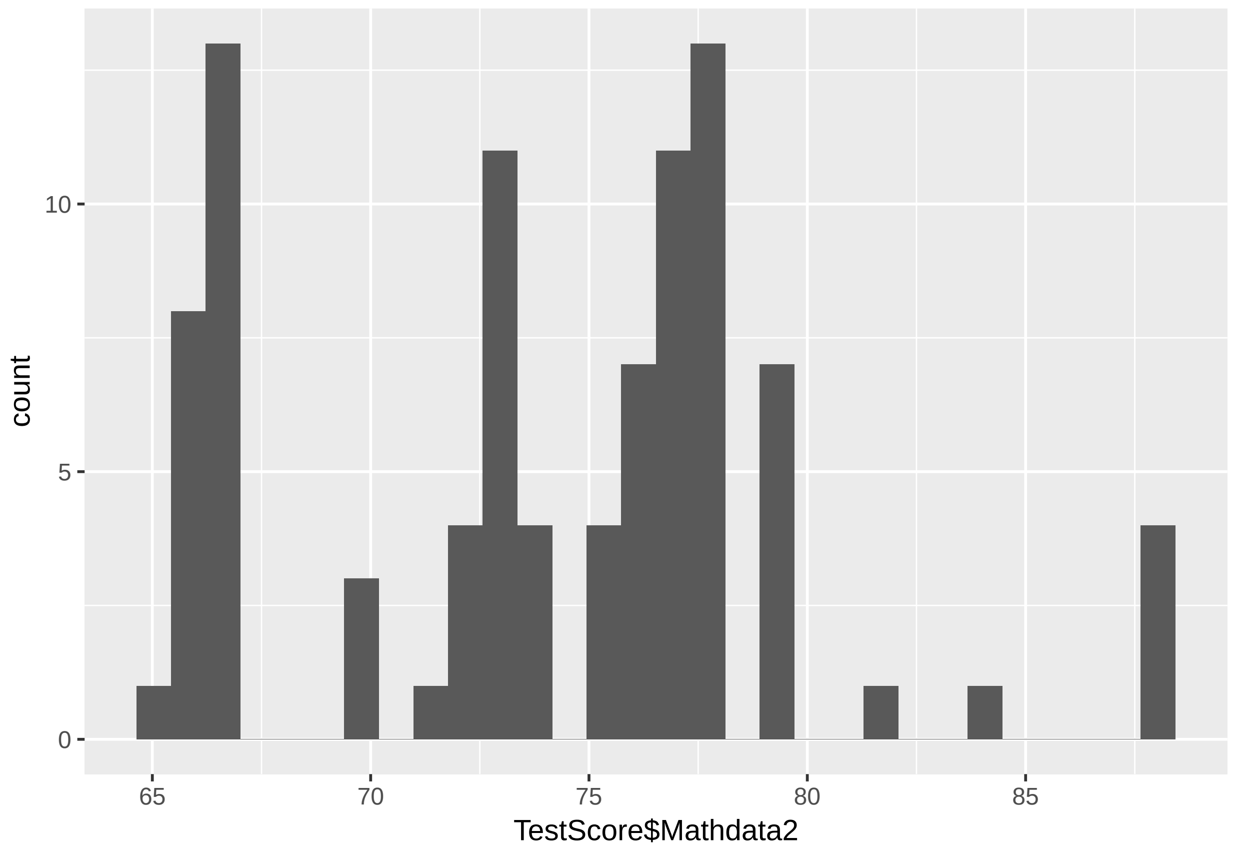
```
Mathdata = sample(Mathdata$Test.Score , 93 , replace = FALSE, prob = NULL)

Mathdata2 = Mathdata2$Test.Score


TestScore = cbind(Mathdata,Mathdata2)

TestScore = data.frame(TestScore)


#### Data isn't normal

ggplot(TestScore, aes(x= TestScore$Mathdata)) + geom_histogram()
```

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



```
ggplot(TestScore, aes(x= TestScore$Mathdata2)) + geom_histogram()
```

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



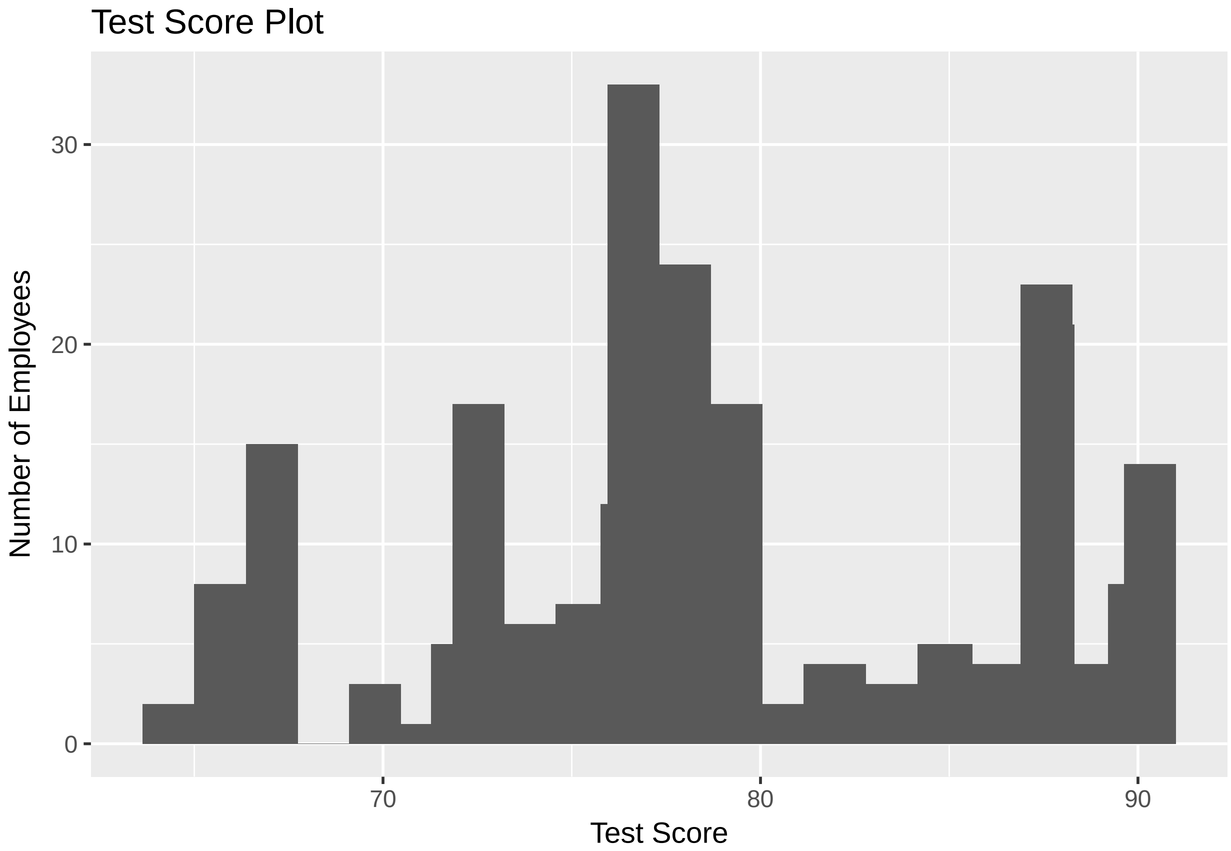
```
wilcox.test(TestScore$Mathdata, TestScore$Mathdata2, alternative = "two.sided")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  TestScore$Mathdata and TestScore$Mathdata2
## W = 7207, p-value = 3.375e-15
## alternative hypothesis: true location shift is not equal to 0
```

```
####W = 7147.5, p-value = 1.227e-14

###There is a sig. difference between the sample of Test Scores for those with and without a math/finan
```

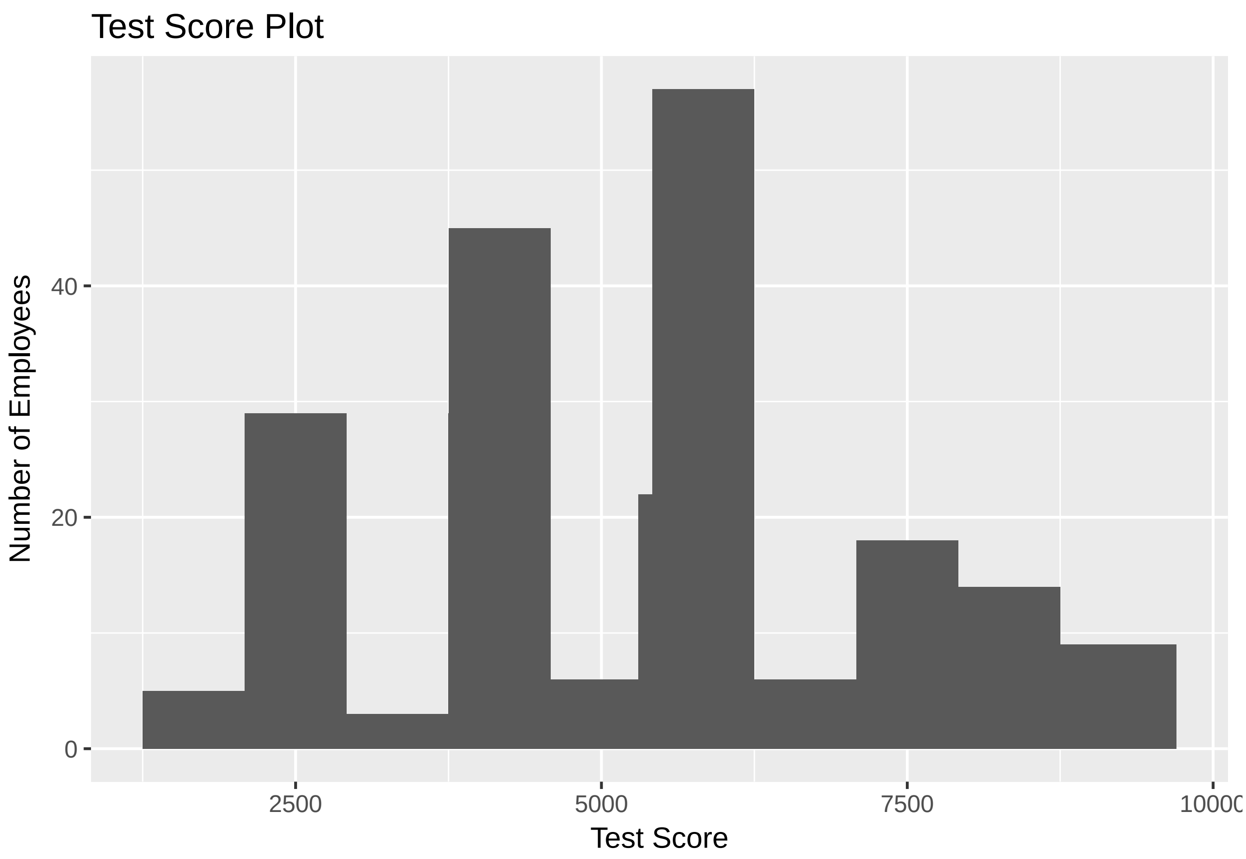
```
### Other Viz.

People6 ["Total.Cost"] <- People7$Total.Cost
People6 ["Total.Recruiting.Cost"]<- People7$Total.Recruiting.Cost
People6 ["Total.Training.Cost"]<- People7$Total.Training.Cost


ggplot(data = People6, aes(x = People6$Test.Score))+
  geom_histogram()+ stat_bin(bins = 20)+
  ggtitle("Test Score Plot")+
  labs(x = "Test Score", y = "Number of Employees")
```

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

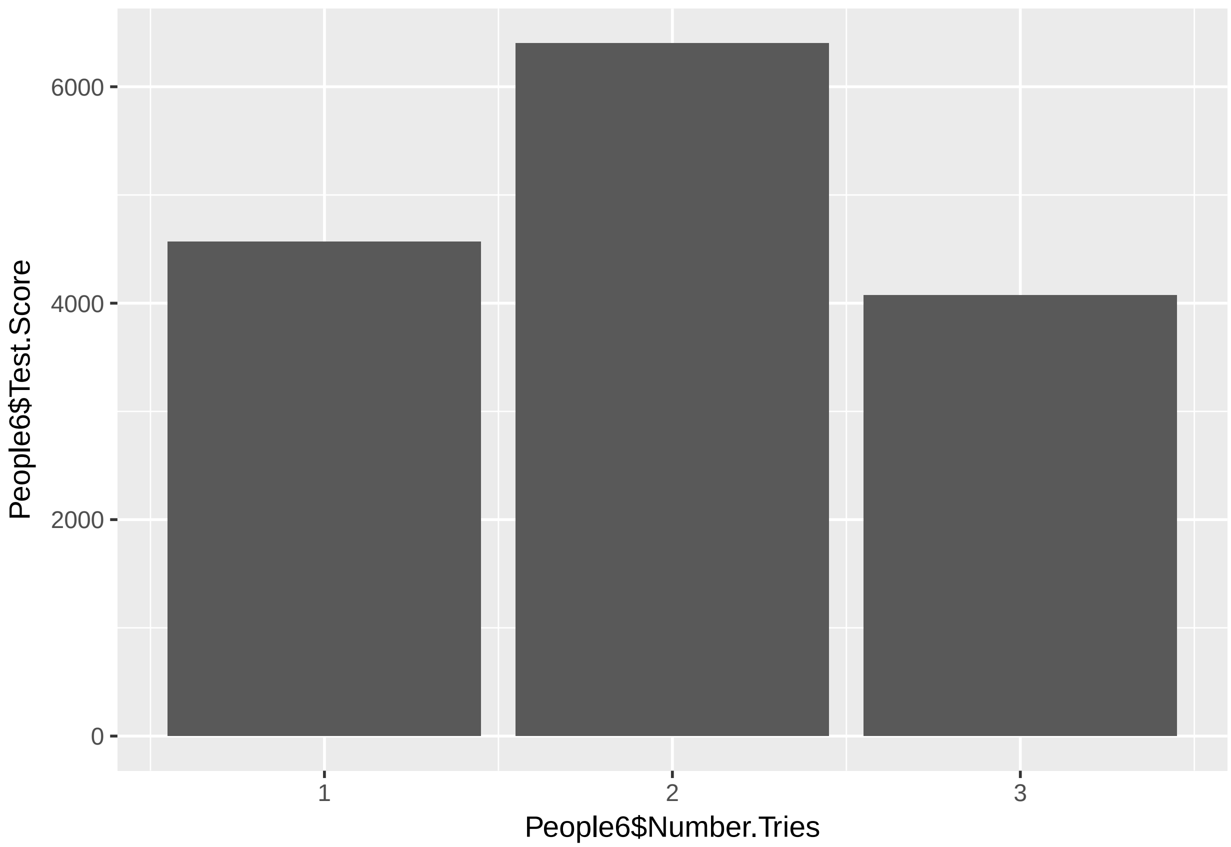


```
ggplot(data = People6, aes(x = People6$Total.Cost))+
  geom_histogram()+ stat_bin(bins = 10)+
  ggtitle("Test Score Plot")+
  labs(x = "Test Score", y = "Number of Employees")
```

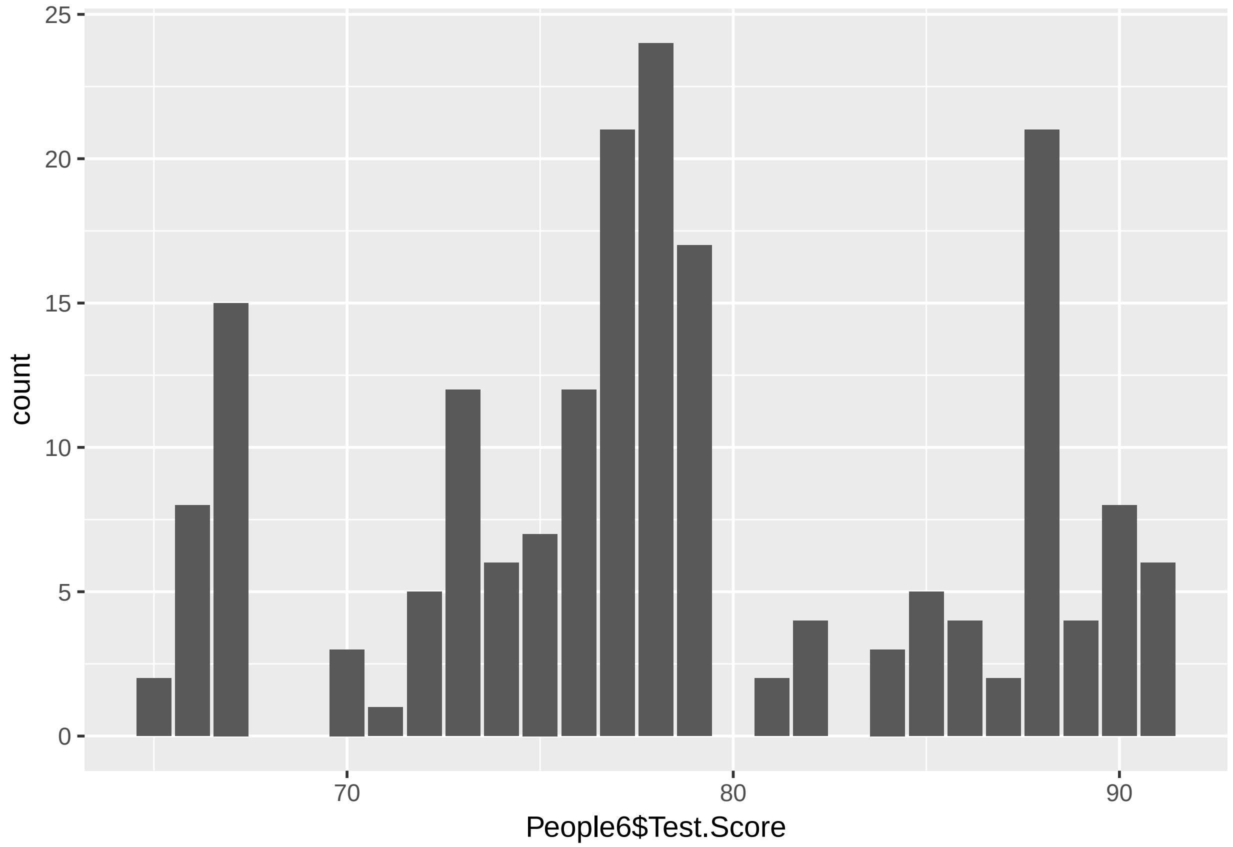
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



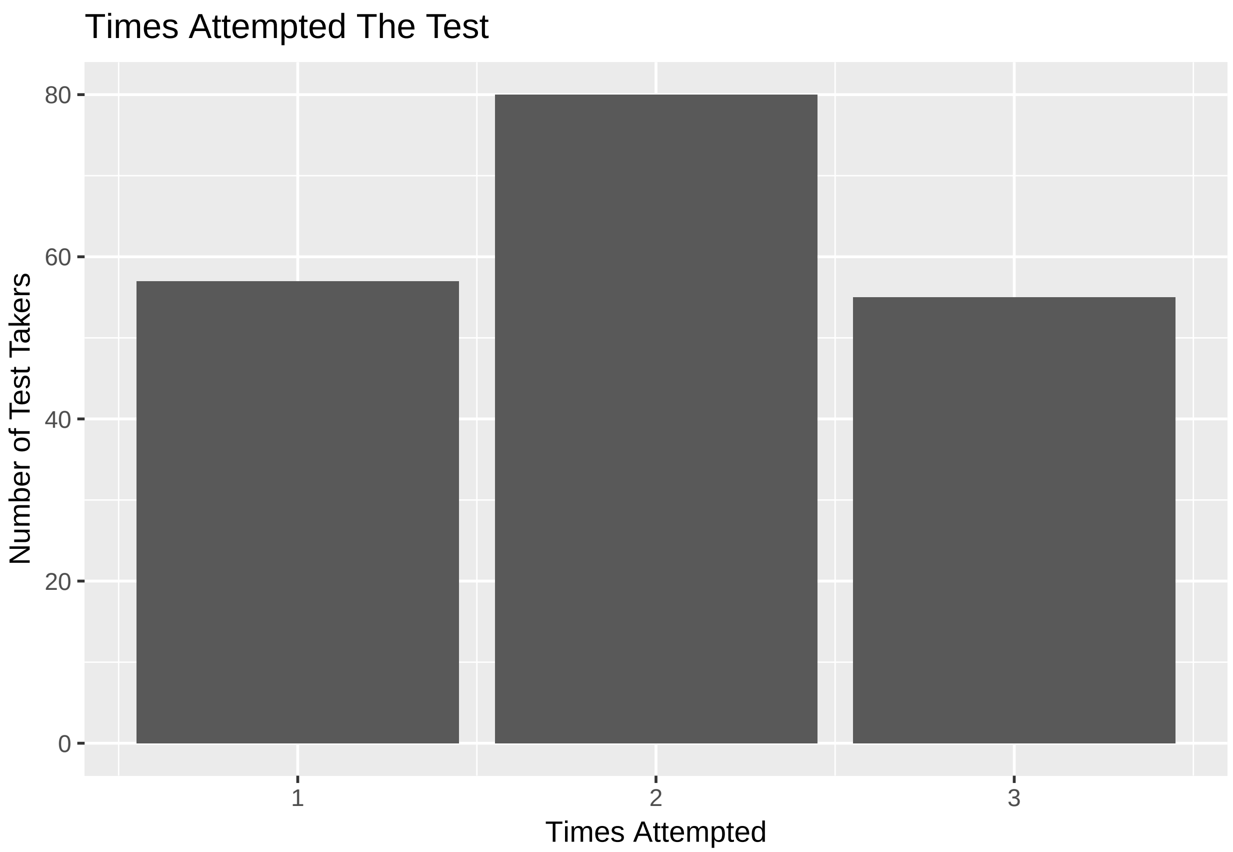
```
NumberTries = ggplot(data = People6, aes(x = People6$Test.Score))+
              geom_bar(aes(fill = People6$Number.Tries))


ggplot(People6, aes(x=People6$Number.Tries,y = People6$Test.Score)) +geom_bar(stat = "identity")
```



```
NumberTries
```



```
ggplot(data = People6, aes(x = People6$Number.Tries)) +
  geom_bar(aes(fill = People6$Number.Tries), position = "dodge")+
  ggtitle("Times Attempted The Test")+
  labs(x = "Times Attempted", y = "Number of Test Takers")
```



```
#### number of failed individuals and deadweight loss associated with their failure



Number3 = dplyr::filter(People6, People6$Number.Tries == 3)

Number2 = dplyr::filter(People6, People6$Number.Tries == 2)

Number1 = dplyr::filter(People6, People6$Number.Tries == 1)


Failed = dplyr::filter(Number3, Number3$Test.Score < 70 )



nrow(Failed)
```

```
## [1] 25
```

```
nrow(Number2)
```

```
## [1] 80
```

```
nrow(Number1)
```

```
## [1] 57
```

```r
nrow(Number3)
```

```
## [1] 55
```

```r
sum(People6$Total.Cost)
```

```
## [1] 1014000
```

```r
sum(Failed$Total.Cost)
```

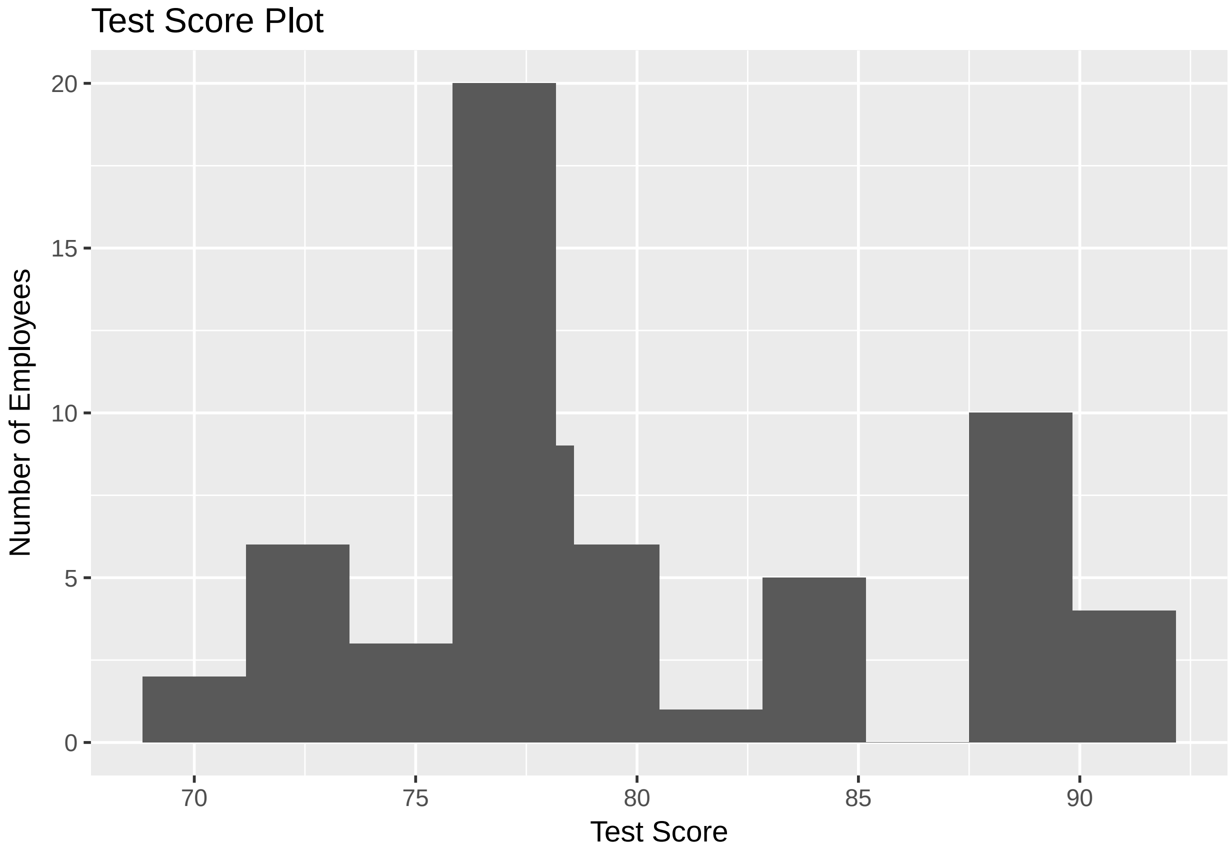
```
## [1] 181500
```

```r
sum(Failed$Total.Training.Cost)
```

```
## [1] 112500
```

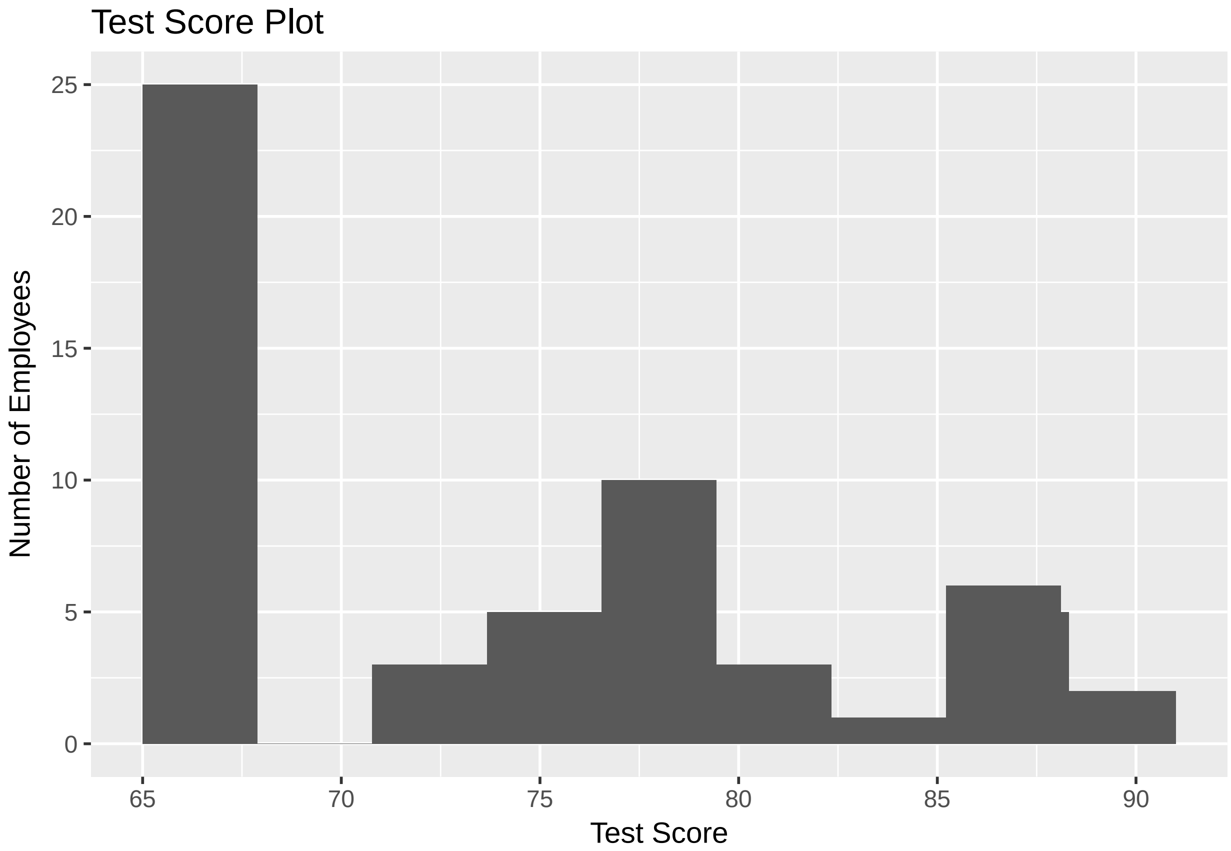
```r
ggplot(data = Number1, aes(x =Number1$Test.Score))+
  geom_histogram()+ stat_bin(bins = 10)+
  ggtitle("Test Score Plot")+
  labs(x = "Test Score", y = "Number of Employees")
```

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



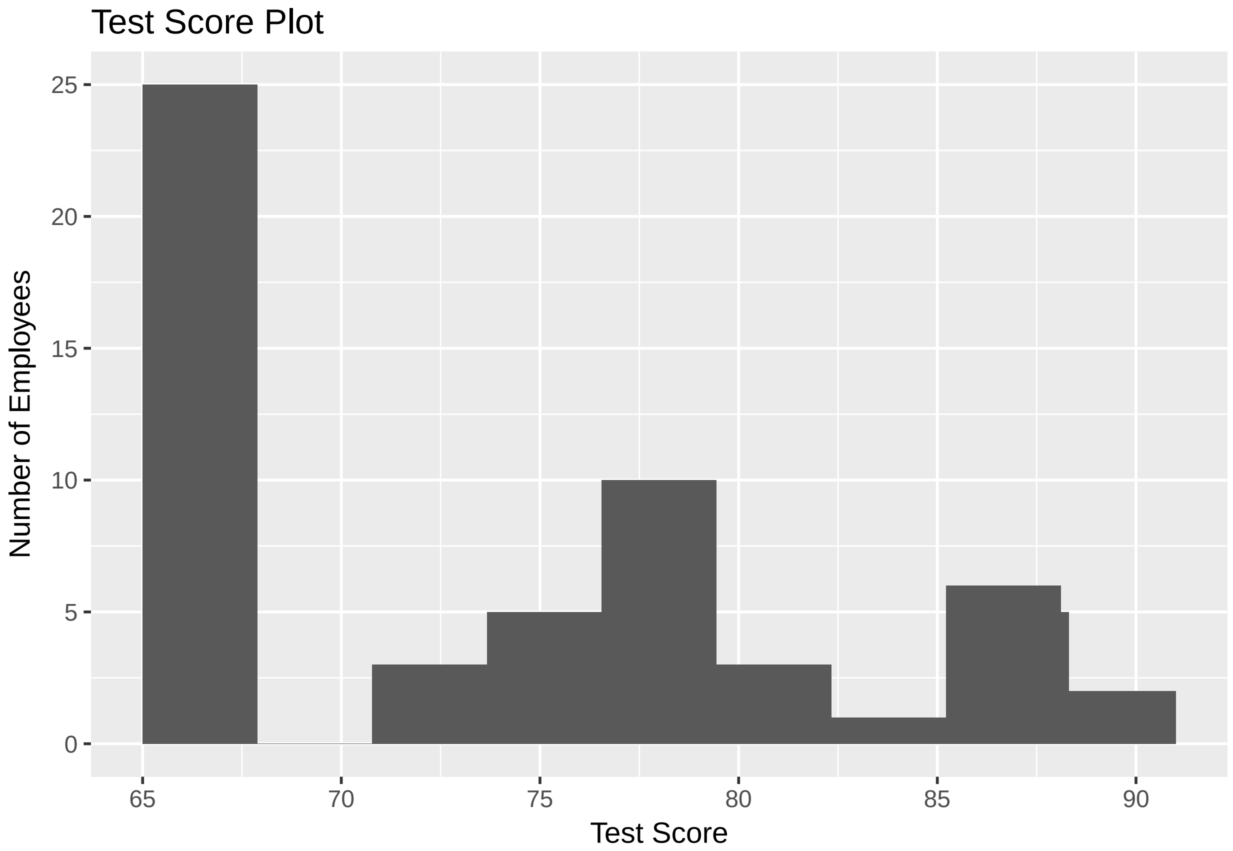
```r
ggplot(data = Number3, aes(x = Number3$Test.Score))+
  geom_histogram()+ stat_bin(bins = 10)+
  ggtitle("Test Score Plot")+
  labs(x = "Test Score", y = "Number of Employees")
```

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



```r
ggplot(data = Number3, aes(x = Number3$Test.Score))+
  geom_histogram()+ stat_bin(bins = 10)+
  ggtitle("Test Score Plot")+
  labs(x = "Test Score", y = "Number of Employees")
```

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



```
kruskal.test(People6$Test.Score ~ People6$Number.Tries, data = People6)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  People6$Test.Score by People6$Number.Tries
## Kruskal-Wallis chi-squared = 22.123, df = 2, p-value = 1.57e-05
```

```
wilcox.test(Number1$Test.Score, Number2$Test.Score, alternative = "two.sided")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Number1$Test.Score and Number2$Test.Score
## W = 2340, p-value = 0.7941
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Number1$Test.Score, Number3$Test.Score, alternative = "two.sided")
```

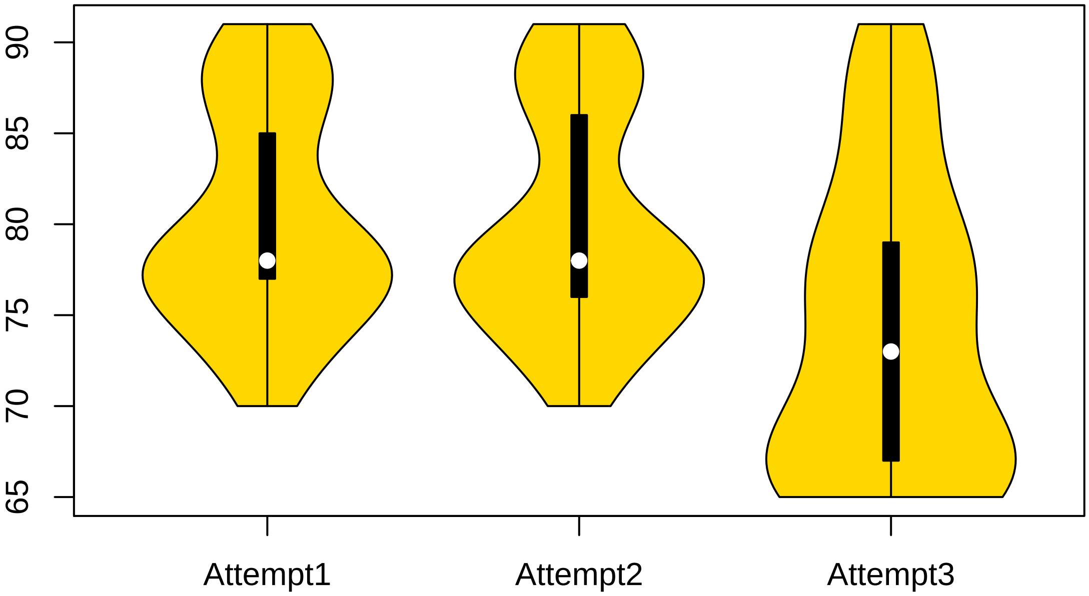
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Number1$Test.Score and Number3$Test.Score
## W = 2261, p-value = 5.136e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Number2$Test.Score, Number3$Test.Score, alternative = "two.sided")
```

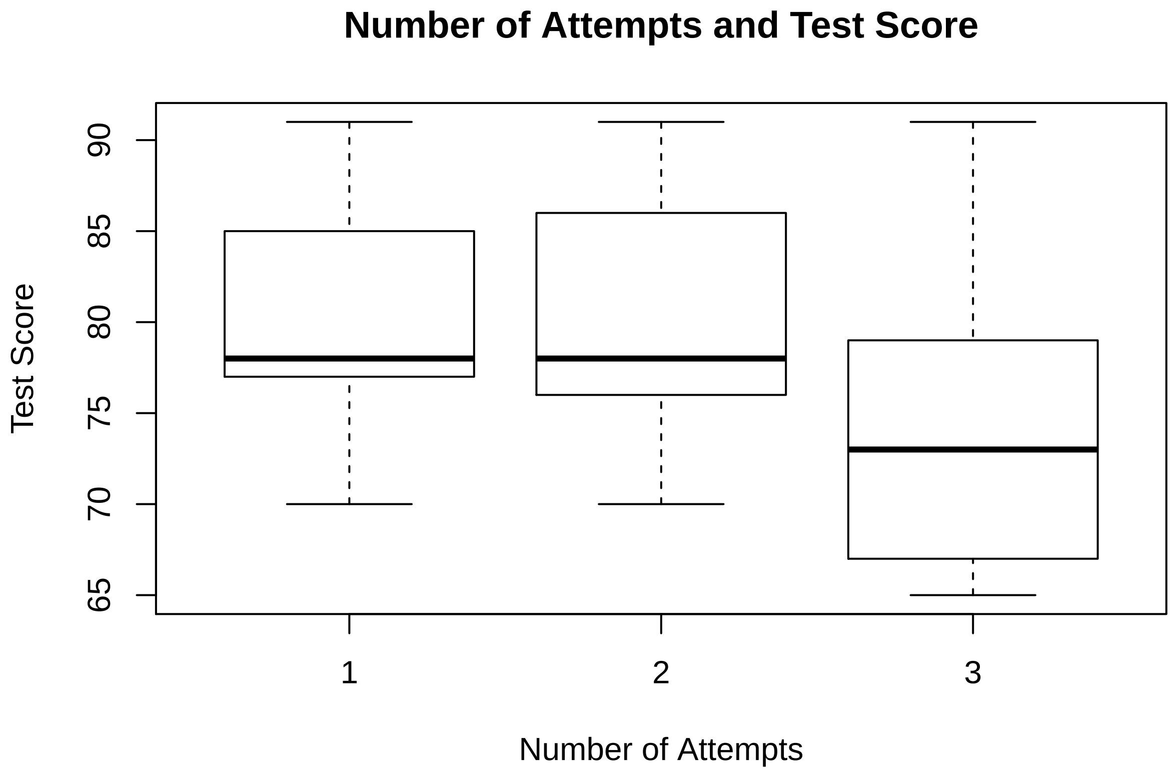
```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  Number2$Test.Score and Number3$Test.Score
## W = 3136.5, p-value = 2.622e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
vioplot(Number1$Test.Score, Number2$Test.Score, Number3$Test.Score, names=c("Attempt1", "Attempt2", "At
   col="gold")
```



```
boxplot(People6$Test.Score ~ People6$Number.Tries, data=People6, main="Number of Attempts and Test Score
    xlab="Number of Attempts", ylab="Test Score")
```



```
People6$Gender
```

```
##   [1] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0
##  [36] 0 0 1 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0
##  [71] 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1
## [106] 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0
## [141] 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0
## [176] 0 1 0 0 1 0 1 0 0 1 1 1 0 0 0 1 1
## Levels: 0 1
```

```
summary(People6$Gender)
```

```
##   0   1
## 113  79
```

```
summary(Number3$Gender)
```

```
##  0  1
## 34 21
```

```
summary(Failed$Gender)
```

```
##  0  1
## 12 13
```

```
kruskal.test(People6$Test.Score ~ People6$Number.Tries, data = People6)
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  People6$Test.Score by People6$Number.Tries
## Kruskal-Wallis chi-squared = 22.123, df = 2, p-value = 1.57e-05
```

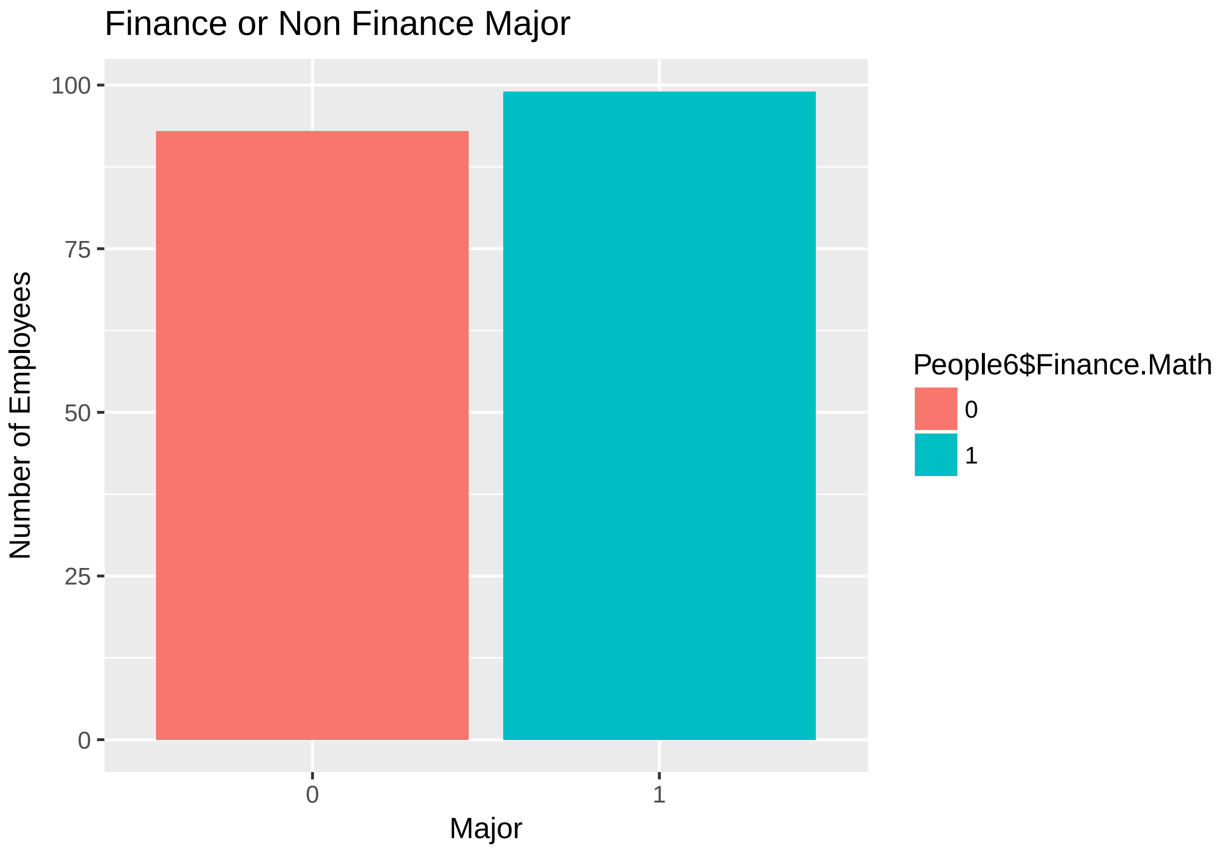
```r
kruskal.test(People6$Test.Score ~ People6$Number.Tries, data = People6)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  People6$Test.Score by People6$Number.Tries
## Kruskal-Wallis chi-squared = 22.123, df = 2, p-value = 1.57e-05
```

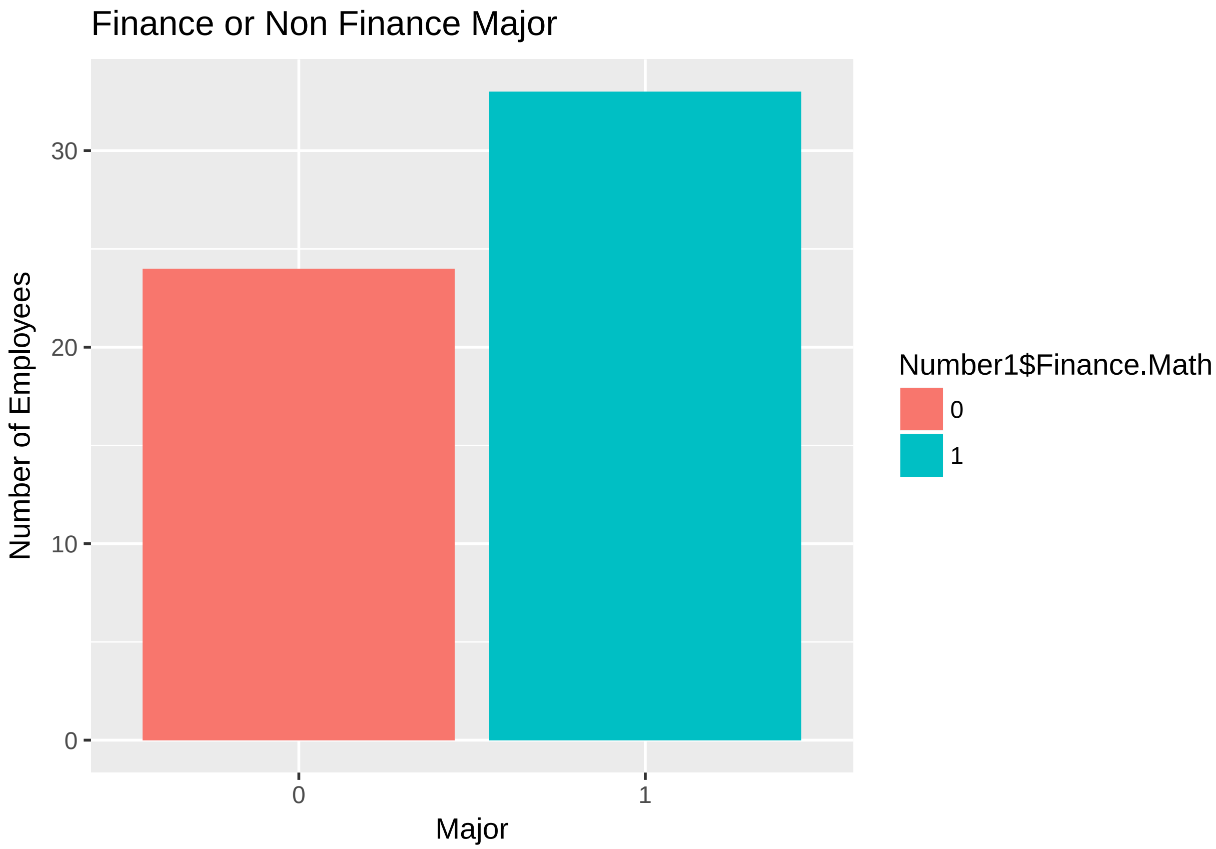
```r
kruskal.test(People6$Test.Score ~ People6$Number.Tries, data = People6)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  People6$Test.Score by People6$Number.Tries
## Kruskal-Wallis chi-squared = 22.123, df = 2, p-value = 1.57e-05
```

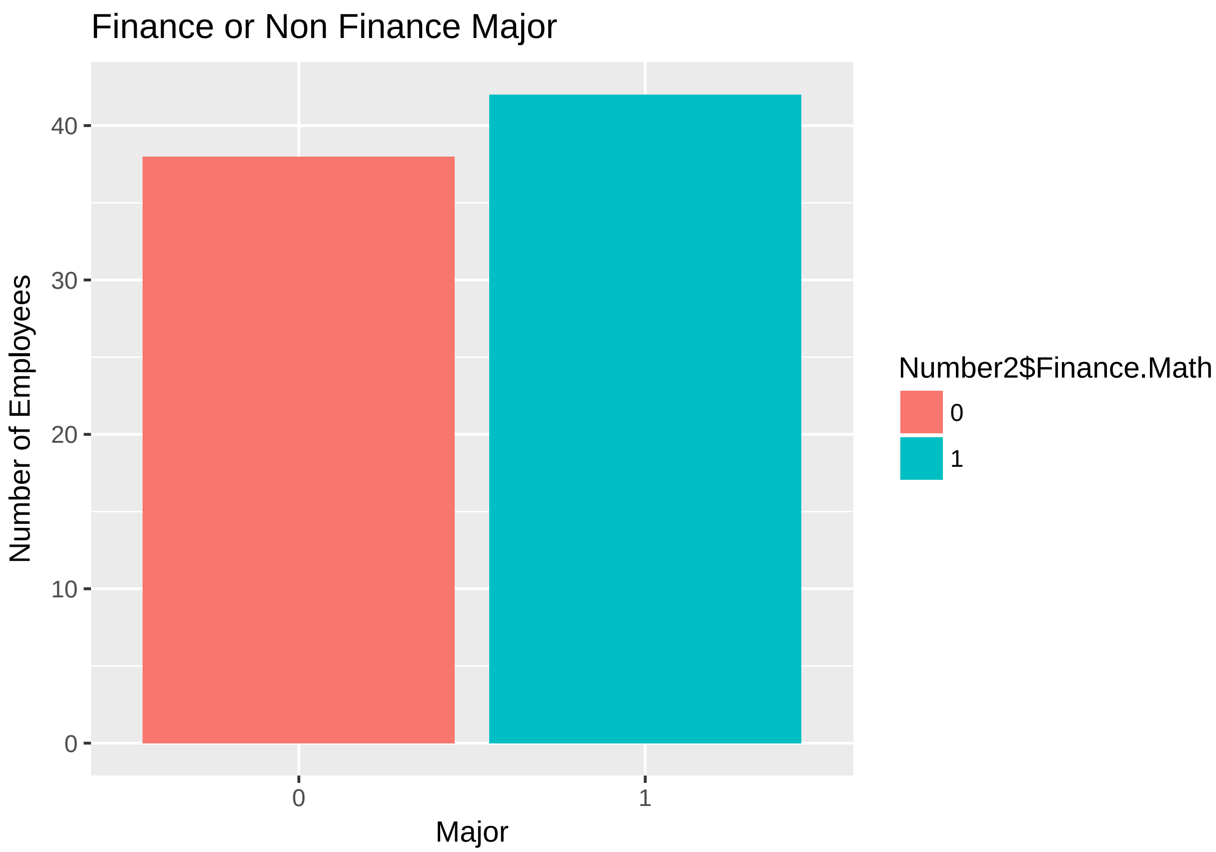
```r
ggplot(data = People6, aes(x = People6$Finance.Math)) +
  geom_bar(aes(fill = People6$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



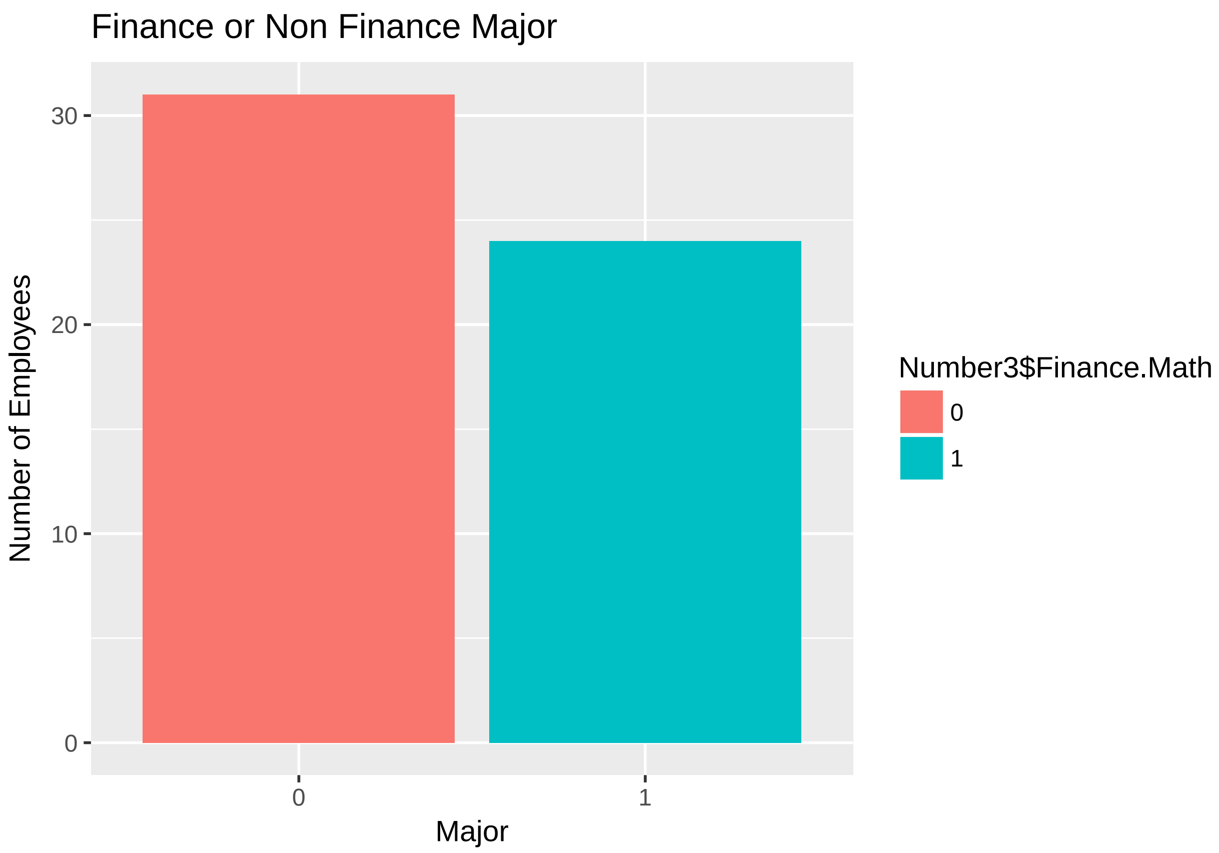
```r
ggplot(data = Number1, aes(x = Number1$Finance.Math)) +
  geom_bar(aes(fill = Number1$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



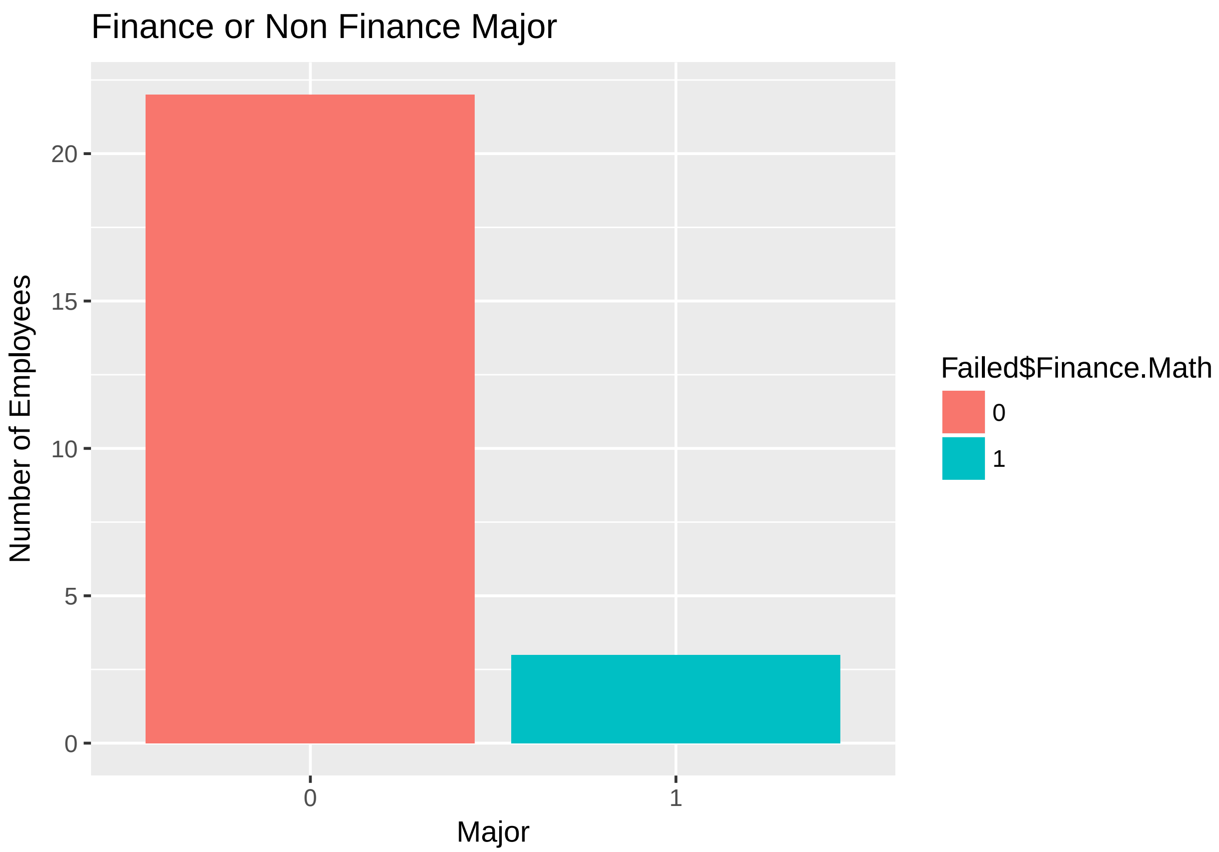
```
ggplot(data = Number2, aes(x = Number2$Finance.Math)) +
  geom_bar(aes(fill = Number2$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



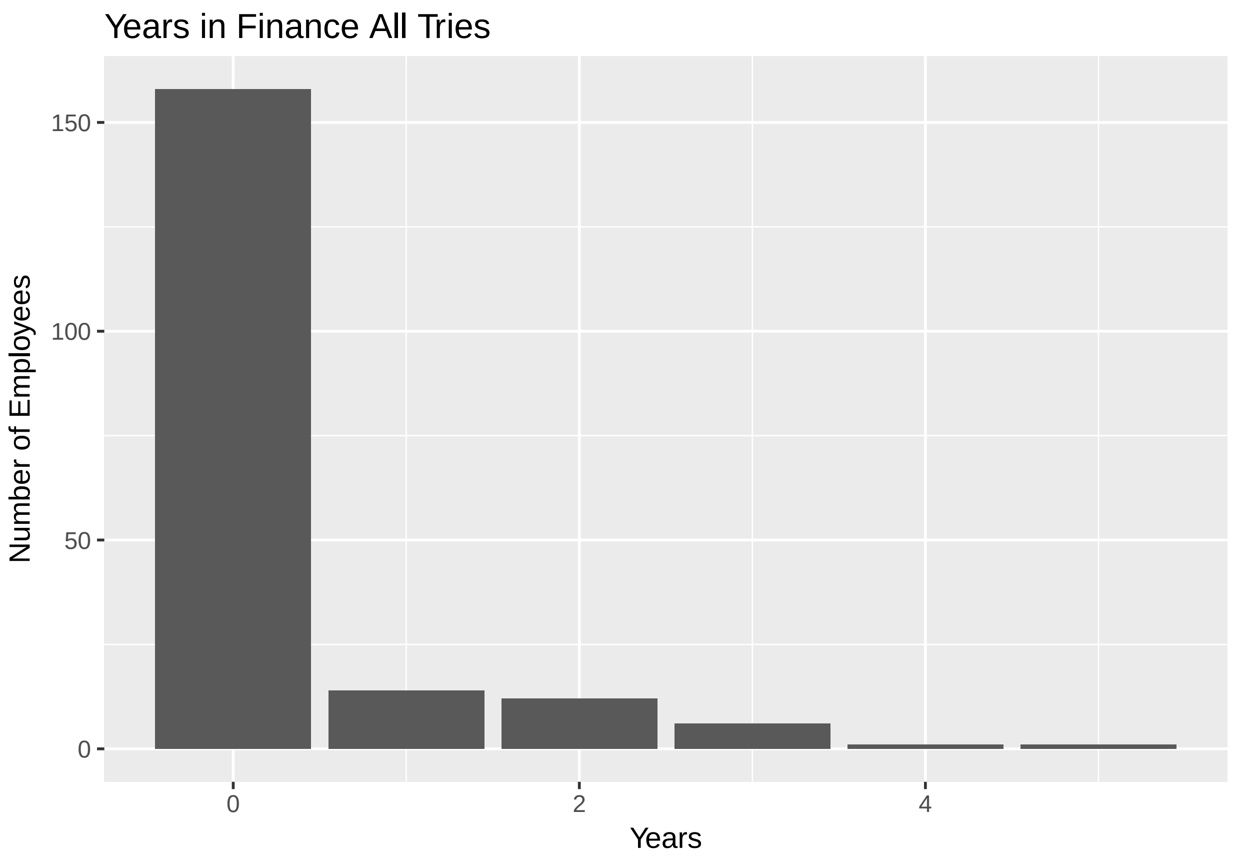
```
ggplot(data = Number3, aes(x = Number3$Finance.Math)) +
  geom_bar(aes(fill = Number3$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



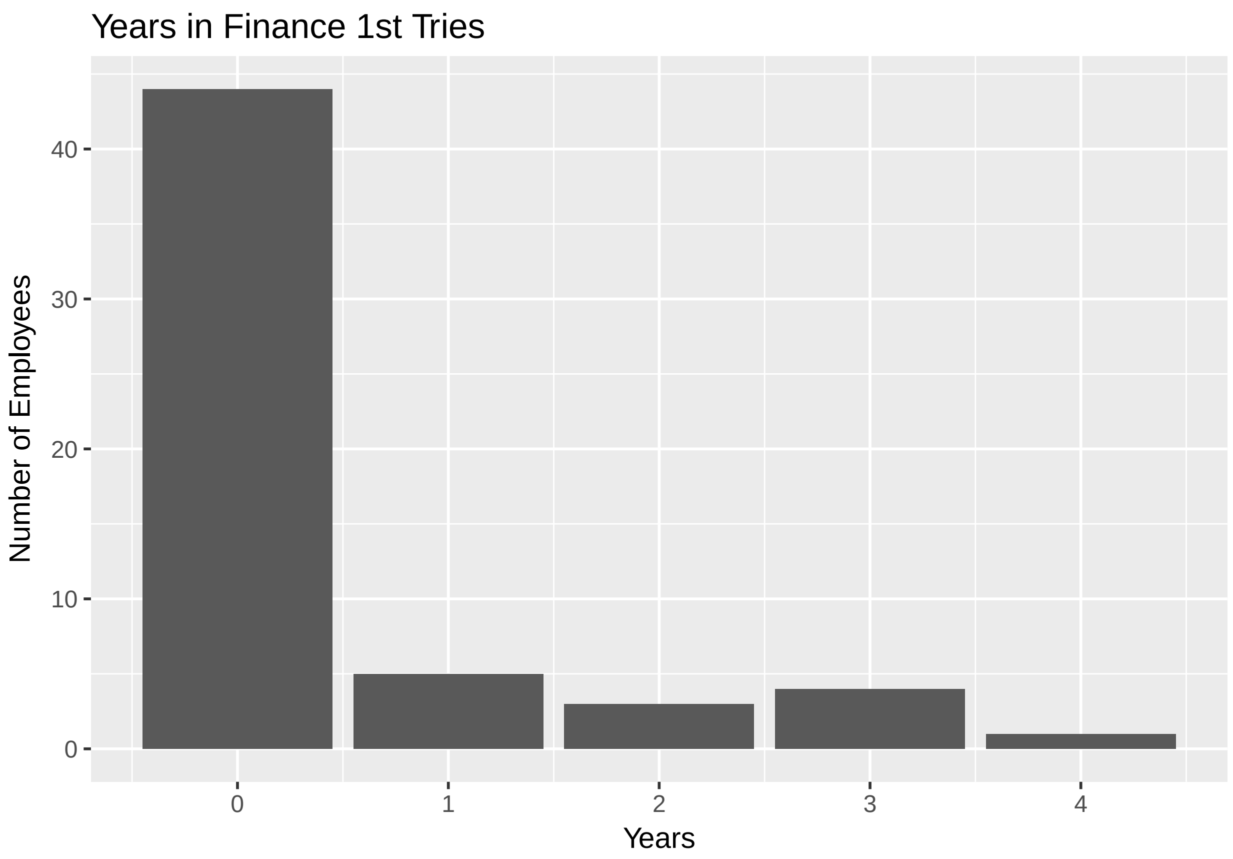
```
ggplot(data = Failed, aes(x = Failed$Finance.Math)) +
  geom_bar(aes(fill = Failed$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



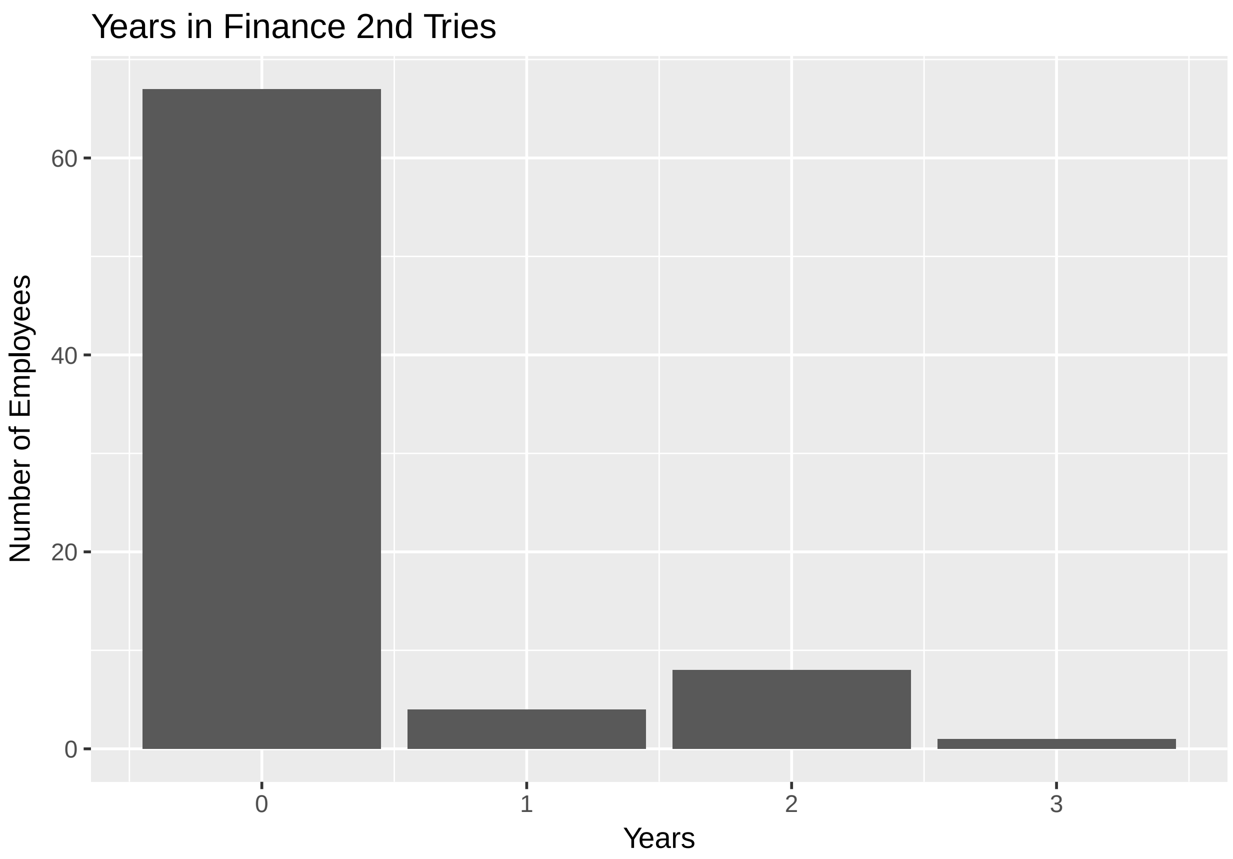
```
ggplot(data = People6, aes(x = People6$Years.in.Fin)) +
  geom_bar(aes(fill = People6$Years.in.Fin), position = "dodge")+
  ggtitle("Years in Finance All Tries")+
  labs(x = "Years", y = "Number of Employees")
```



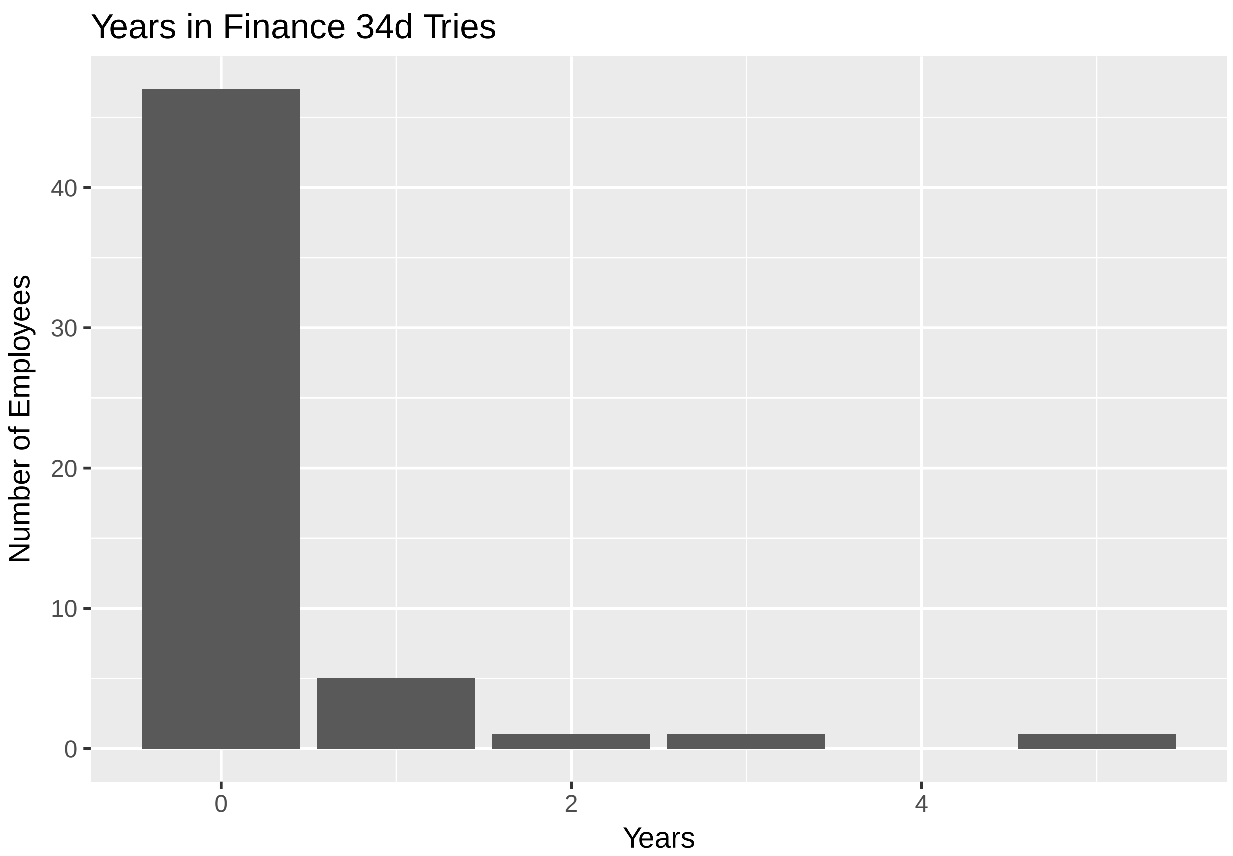
```r
ggplot(data = Number1, aes(x = Number1$Years.in.Fin)) +
  geom_bar(aes(fill = Number1$Years.in.Fin), position = "dodge")+
  ggtitle("Years in Finance 1st Tries")+
  labs(x = "Years", y = "Number of Employees")
```



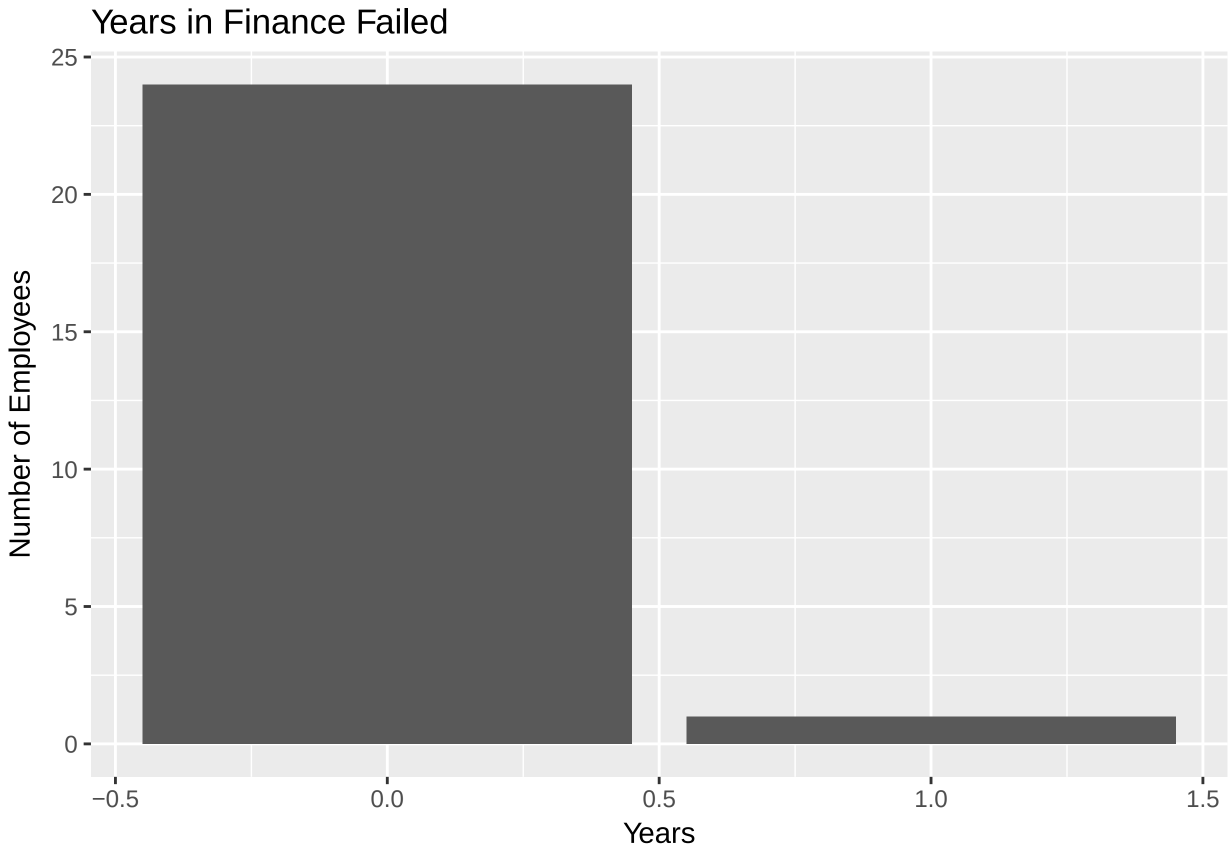
```r
ggplot(data = Number2, aes(x = Number2$Years.in.Fin)) +
  geom_bar(aes(fill = Number2$Years.in.Fin), position = "dodge")+
  ggtitle("Years in Finance 2nd Tries")+
  labs(x = "Years", y = "Number of Employees")
```



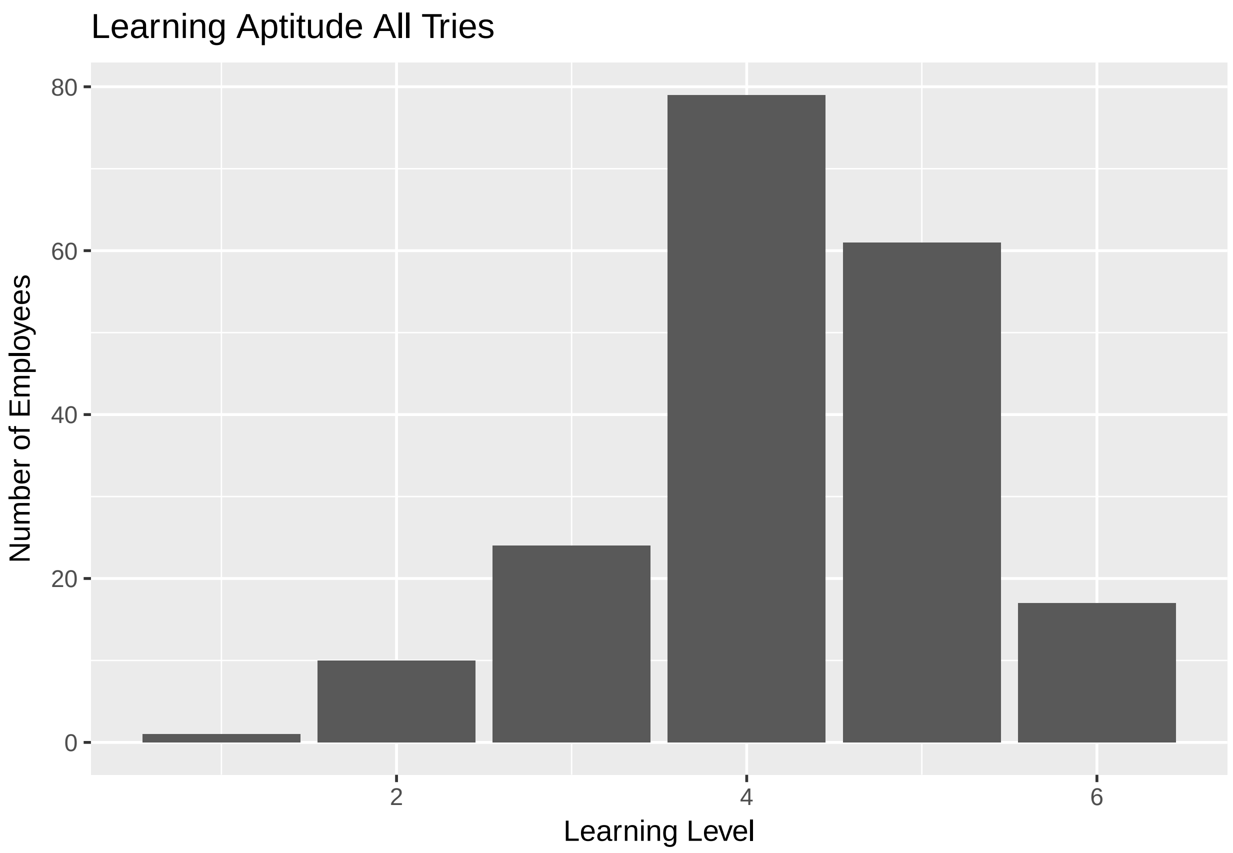
```
ggplot(data = Number3, aes(x = Number3$Years.in.Fin)) +
  geom_bar(aes(fill = Number3$Years.in.Fin), position = "dodge")+
  ggtitle("Years in Finance 34d Tries")+
  labs(x = "Years", y = "Number of Employees")
```



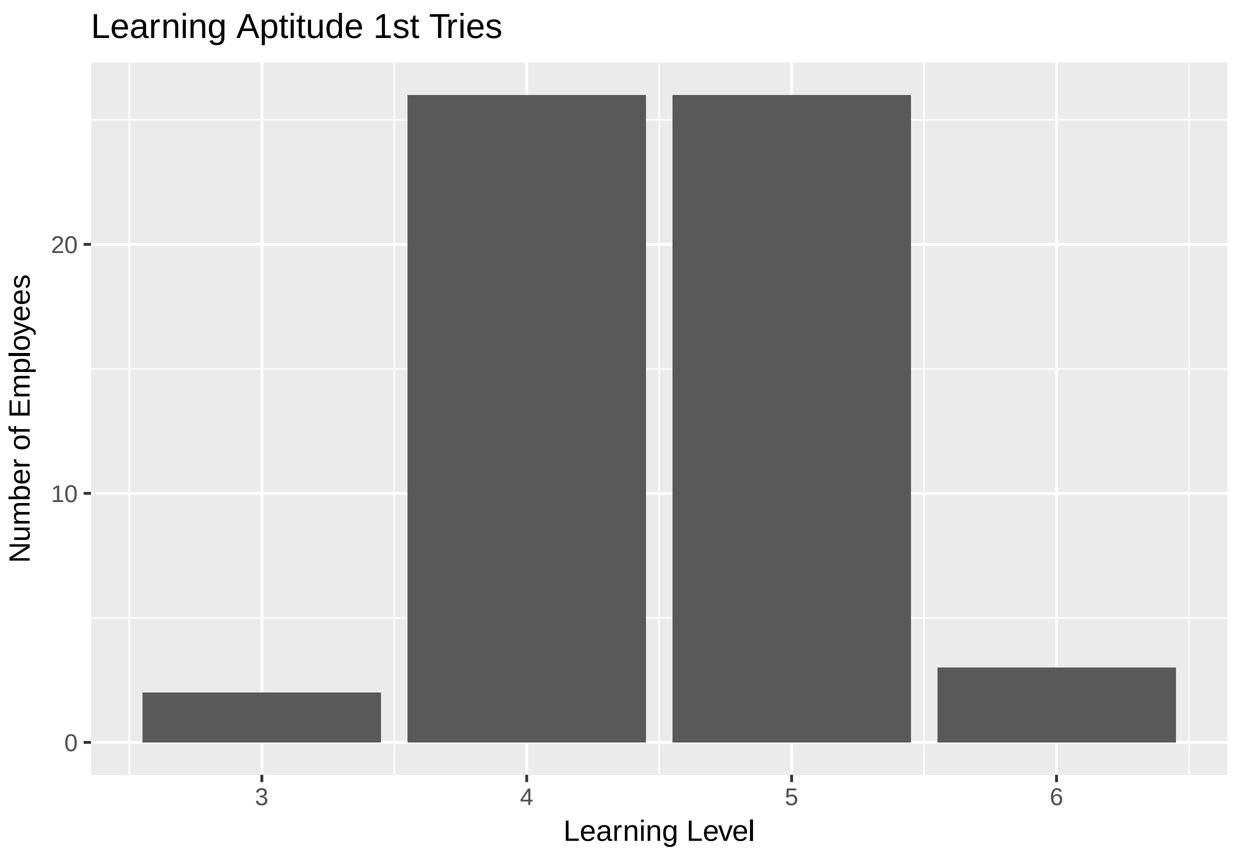
```
ggplot(data = Failed, aes(x = Failed$Years.in.Fin)) +
  geom_bar(aes(fill = Failed$Years.in.Fin), position = "dodge")+
  ggtitle("Years in Finance Failed")+
  labs(x = "Years", y = "Number of Employees")
```



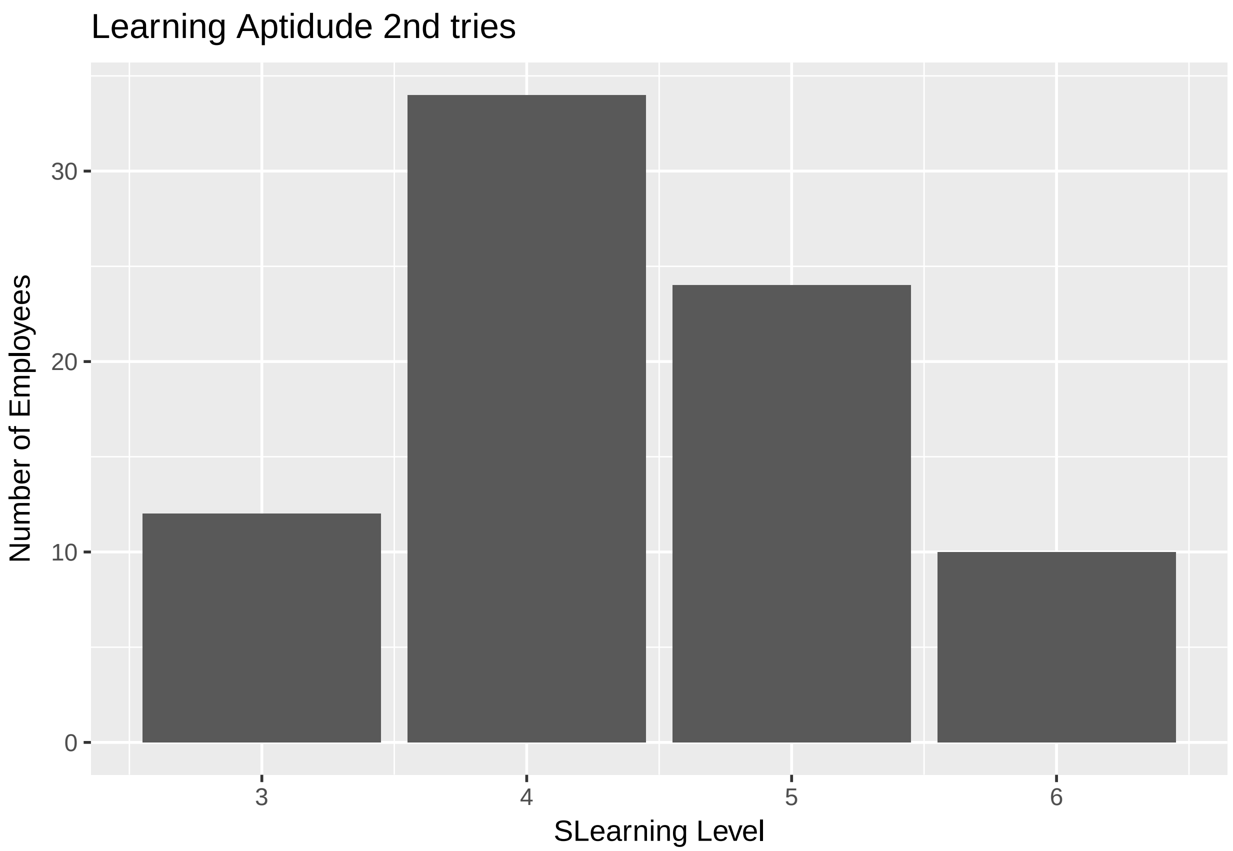
```
ggplot(data = People6, aes(x = People6$`Learning Aptitude`)) +
  geom_bar(aes(fill = People6$`Learning Aptitude`), position = "dodge")+
  ggtitle("Learning Aptitude All Tries")+
  labs(x = "Learning Level", y = "Number of Employees")
```



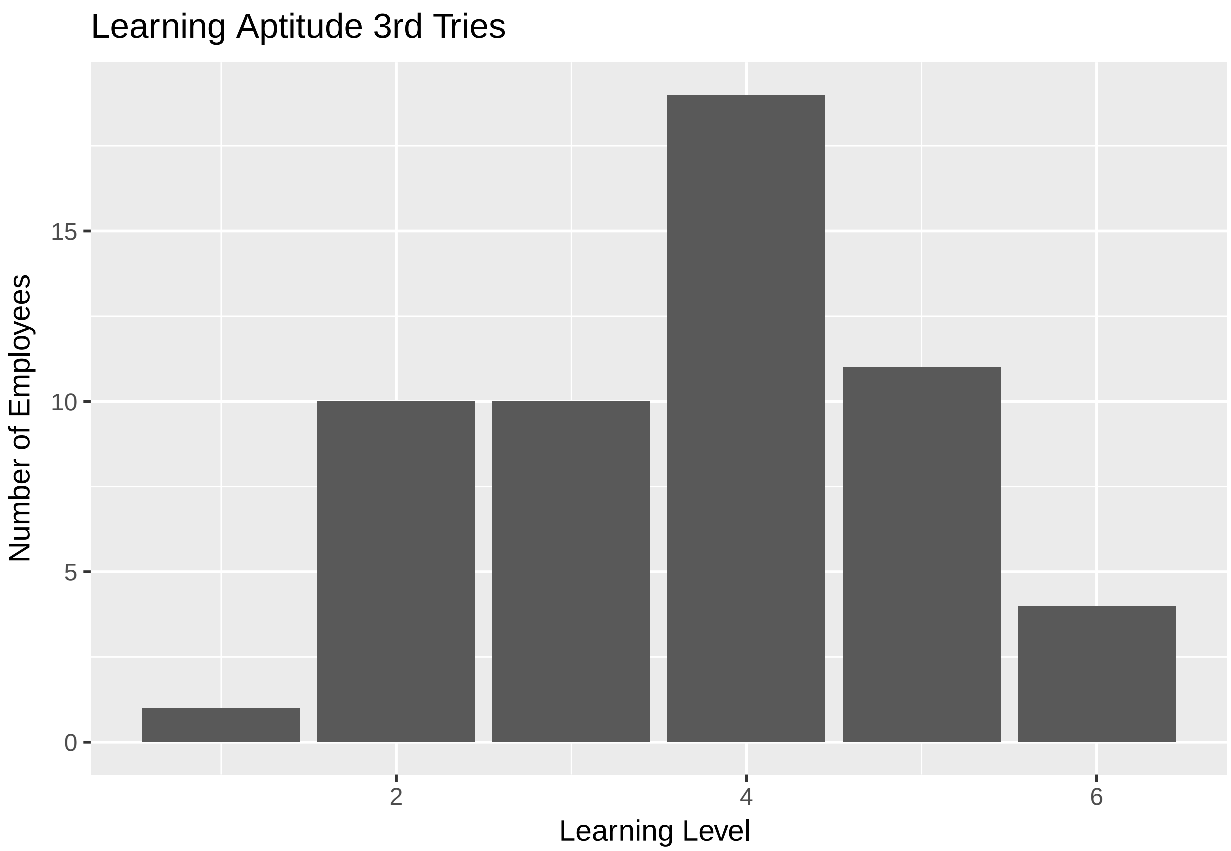
```
ggplot(data = Number1, aes(x = Number1$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number1$`Learning Aptitude`), position = "dodge")+
  ggtitle("Learning Aptitude 1st Tries")+
  labs(x = "Learning Level", y = "Number of Employees")
```



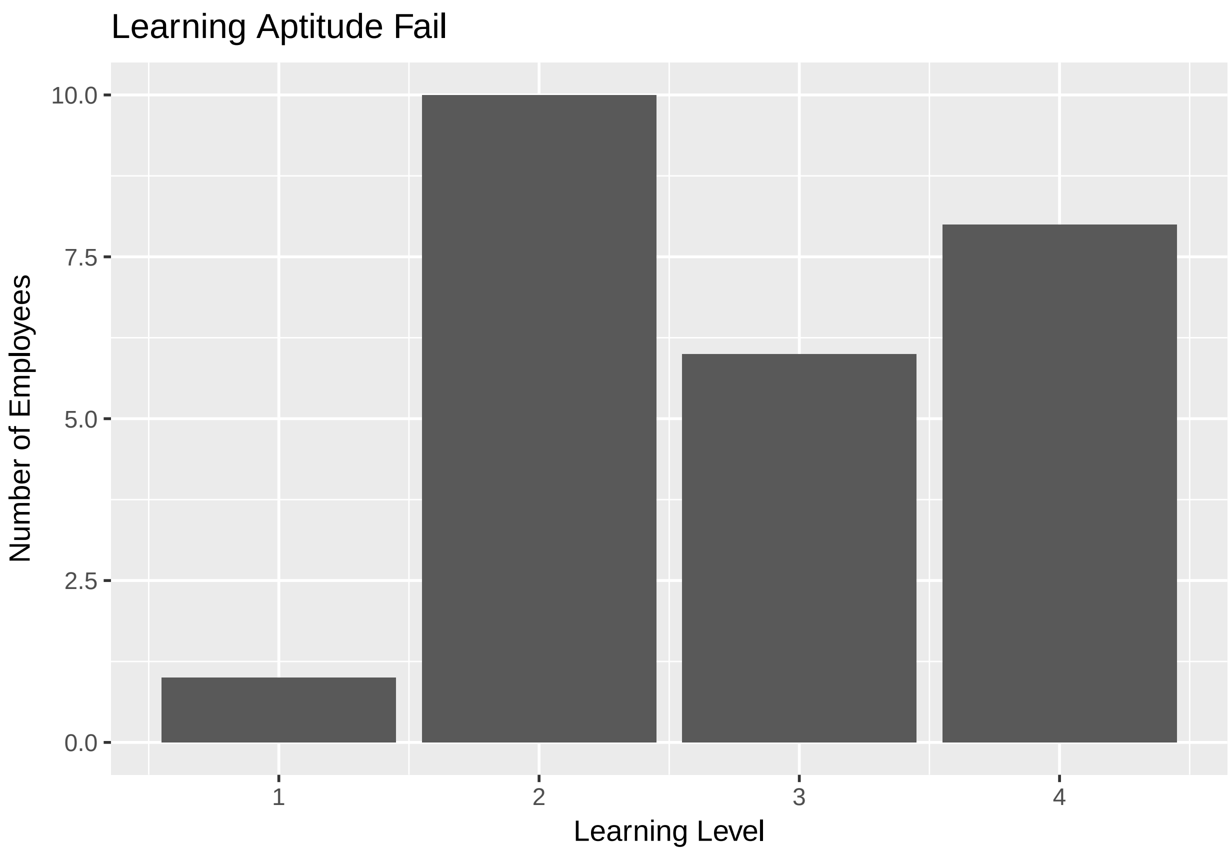
```
ggplot(data = Number2, aes(x = Number2$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number2$`Learning Aptitude`), position = "dodge")+
  ggtitle("Learning Aptidude 2nd tries")+
  labs(x = "SLearning Level", y = "Number of Employees")
```



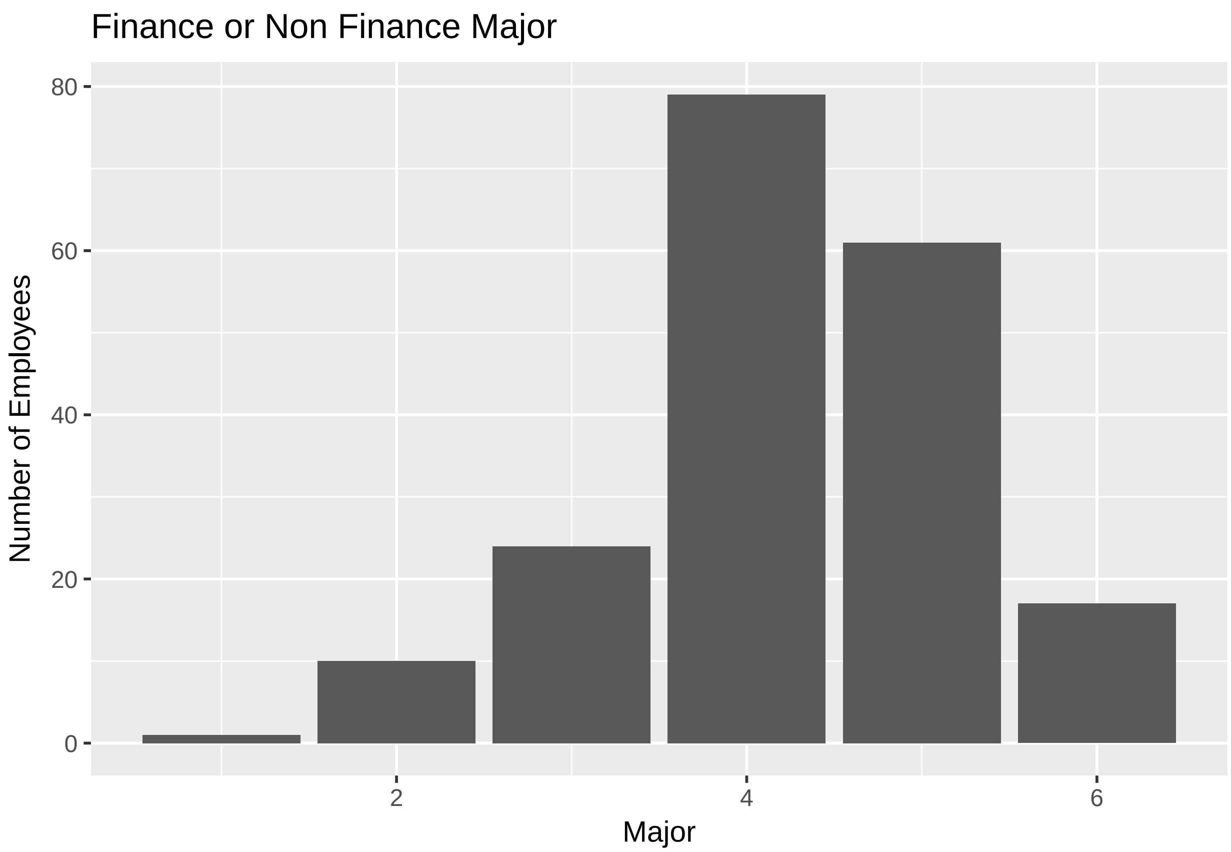
```
ggplot(data = Number3, aes(x = Number3$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number3$`Learning Aptitude`), position = "dodge")+
  ggtitle("Learning Aptitude 3rd Tries")+
  labs(x = "Learning Level ", y = "Number of Employees")
```



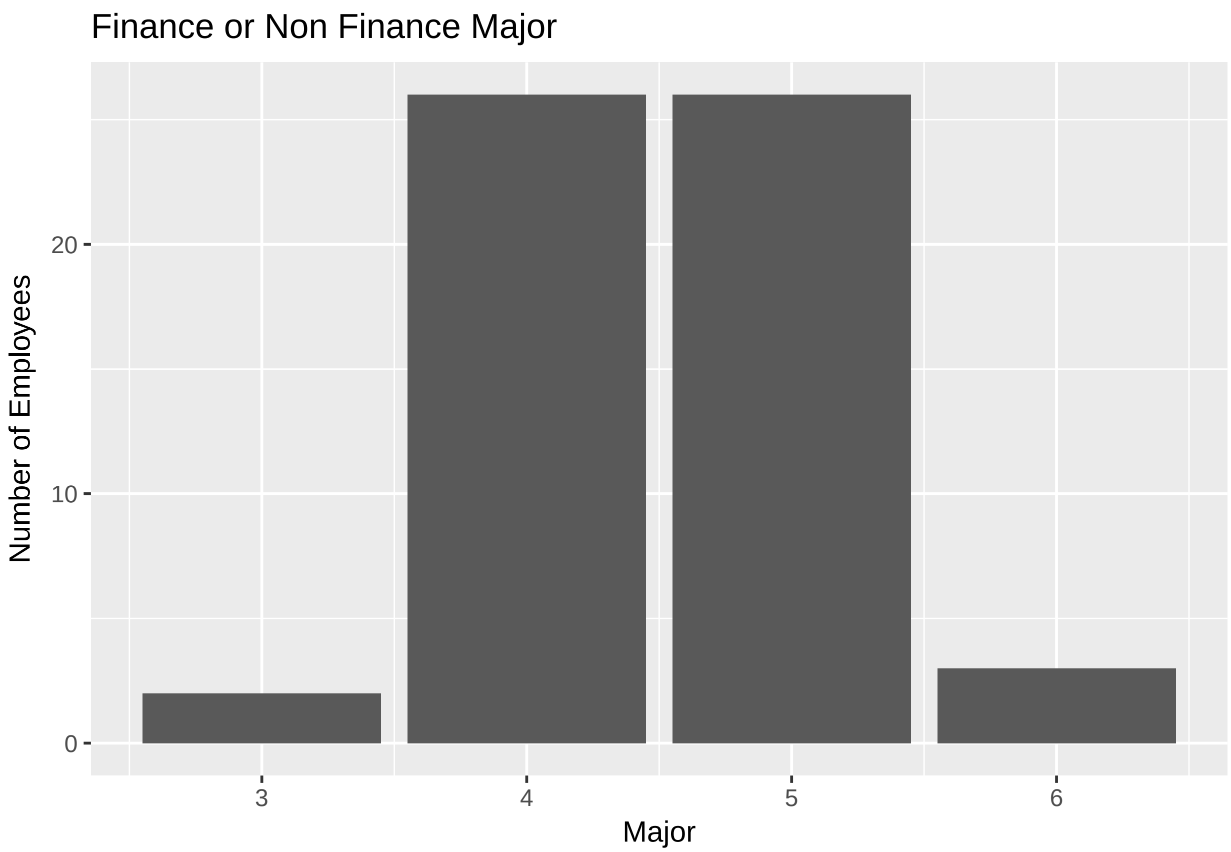
```
ggplot(data = Failed, aes(x = Failed$`Learning Aptitude`)) +
  geom_bar(aes(fill = Failed$`Learning Aptitude`), position = "dodge")+
  ggtitle("Learning Aptitude Fail")+
  labs(x = "Learning Level", y = "Number of Employees")
```



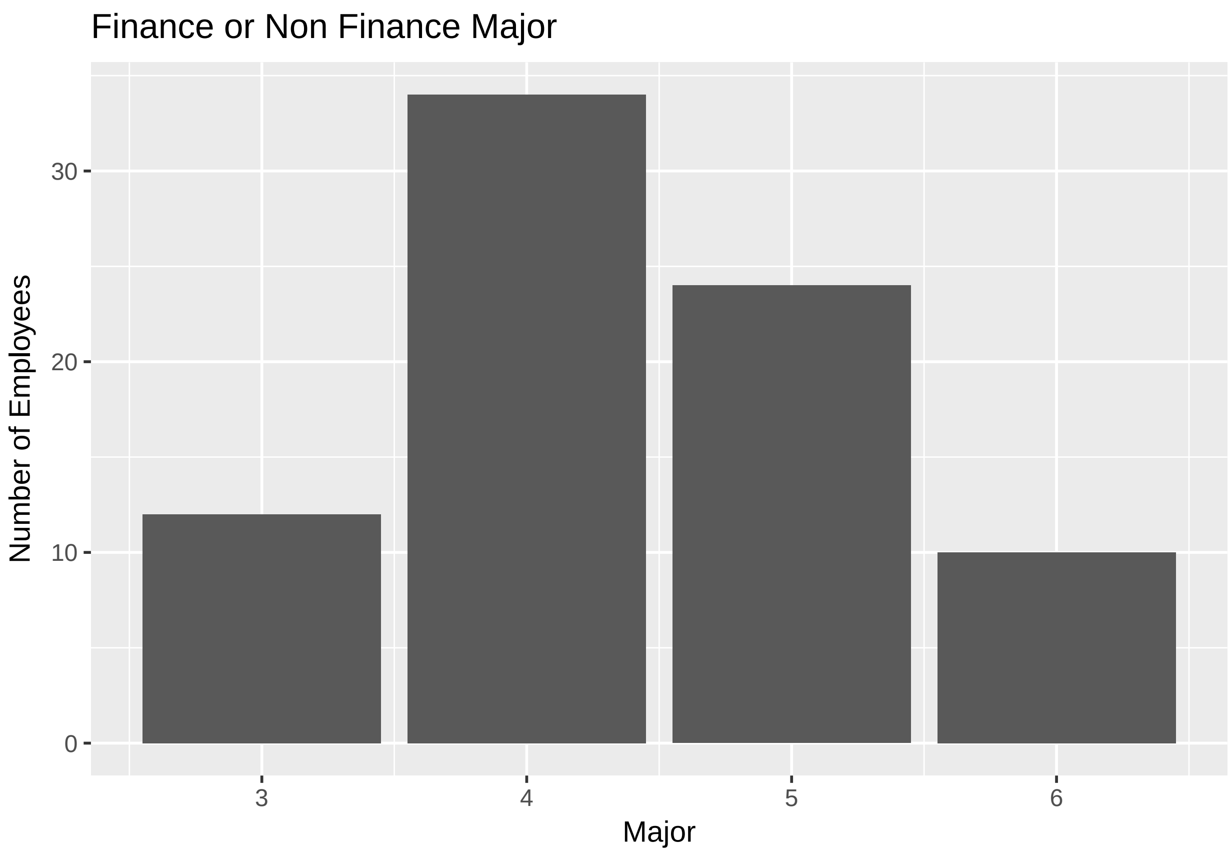
```
ggplot(data = People6, aes(x = People6$`Learning Aptitude`)) +
  geom_bar(aes(fill = People6$`Learning Aptitude`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



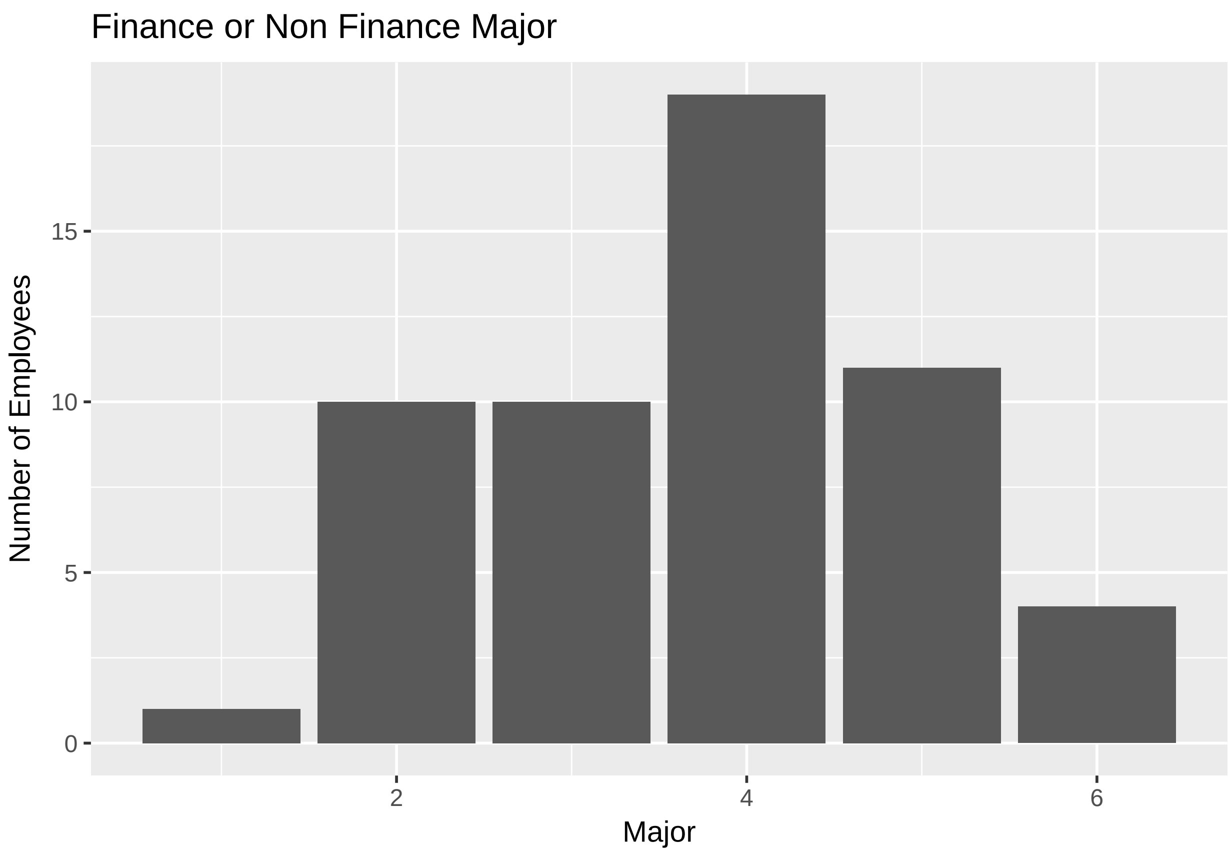
```
ggplot(data = Number1, aes(x = Number1$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number1$`Learning Aptitude`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



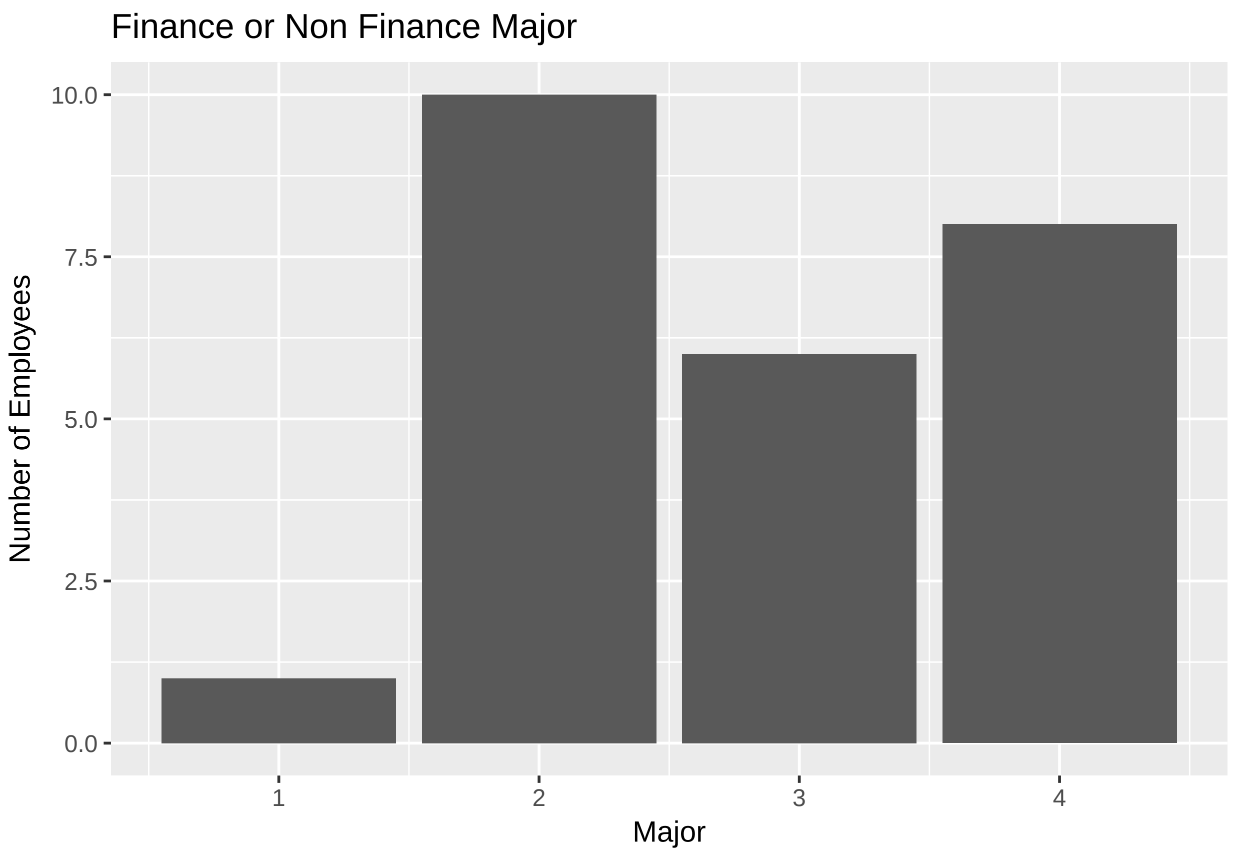
```
ggplot(data = Number2, aes(x = Number2$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number2$`Learning Aptitude`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



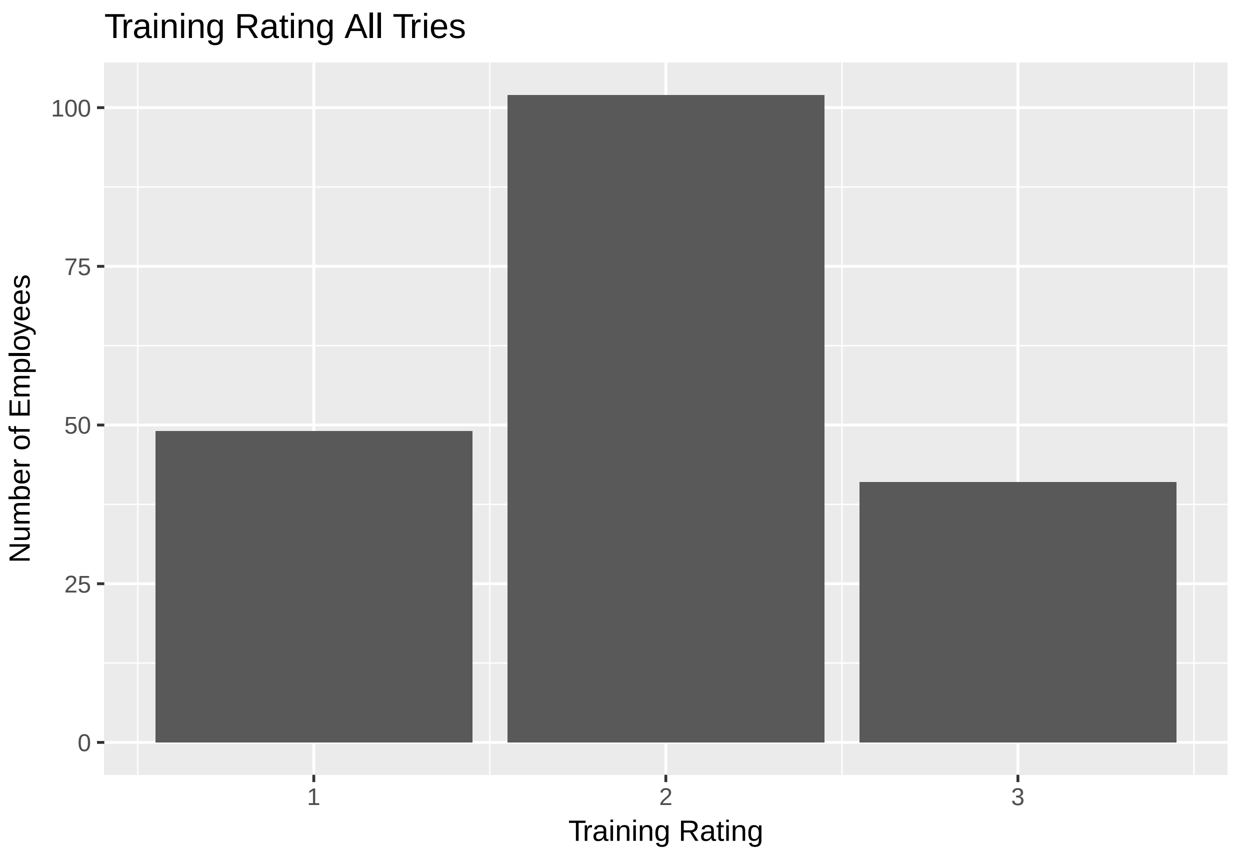
```r
ggplot(data = Number3, aes(x = Number3$`Learning Aptitude`)) +
  geom_bar(aes(fill = Number3$Test.Score), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



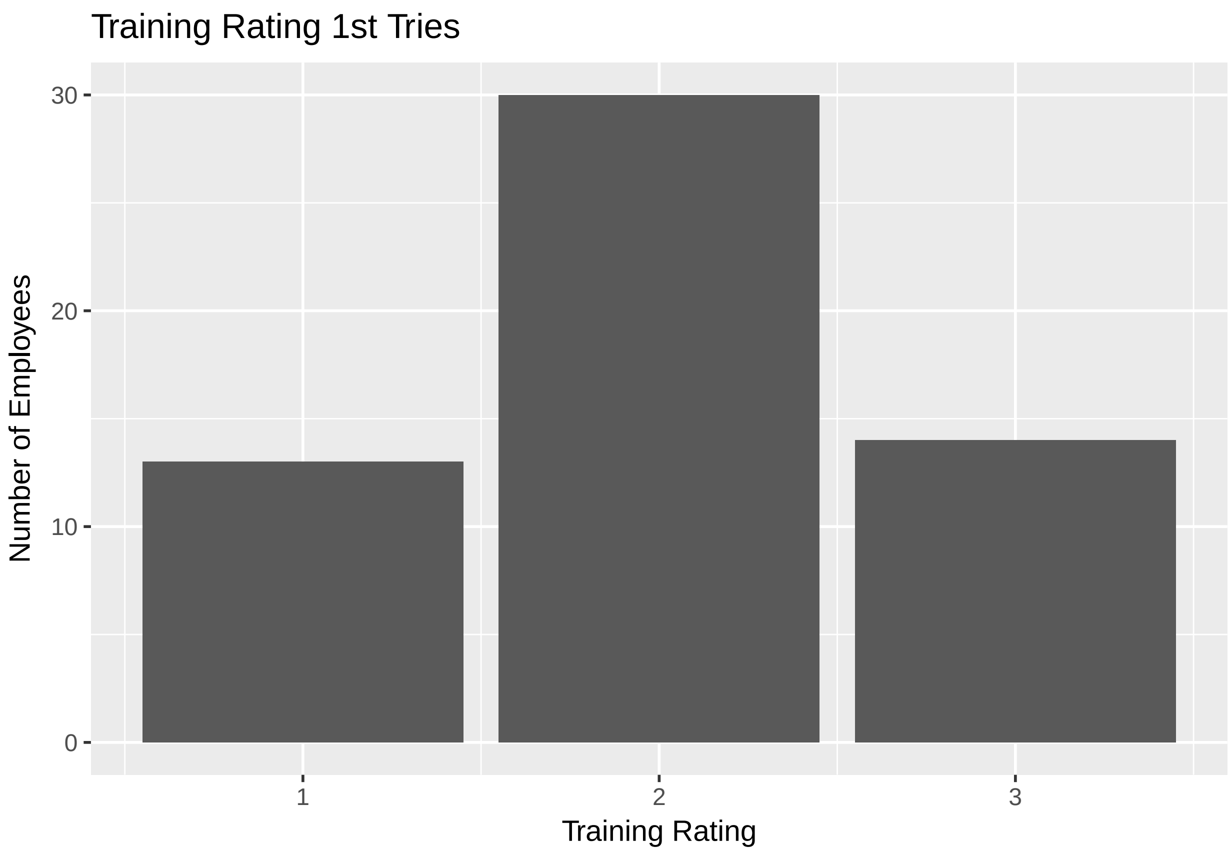
```r
ggplot(data = Failed, aes(x = Failed$`Learning Aptitude`)) +
  geom_bar(aes(fill = Failed$Test.Score), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



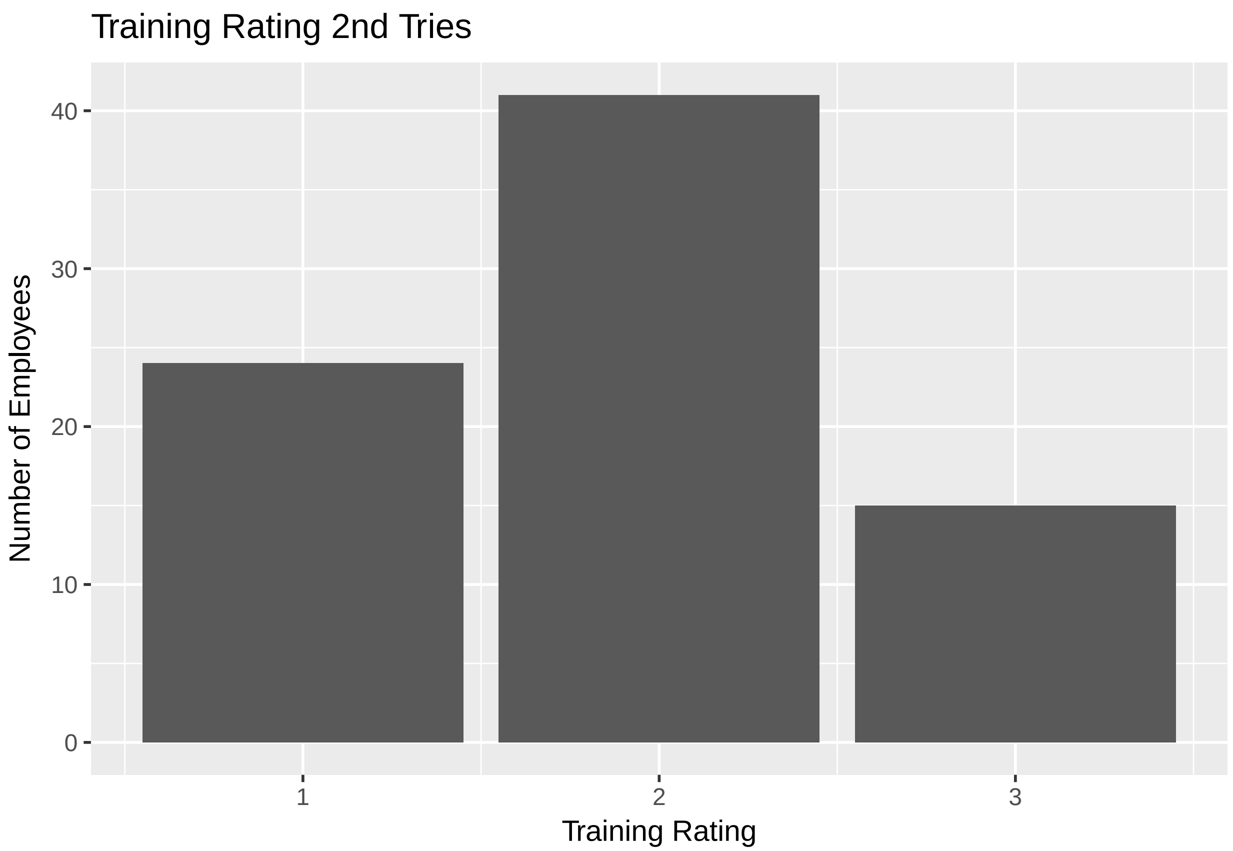
```
ggplot(data = People6, aes(x = People6$Train.RatingN)) +
  geom_bar(aes(fill = People6$Train.RatingN), position = "dodge")+
  ggtitle("Training Rating All Tries")+
  labs(x = "Training Rating", y = "Number of Employees")
```



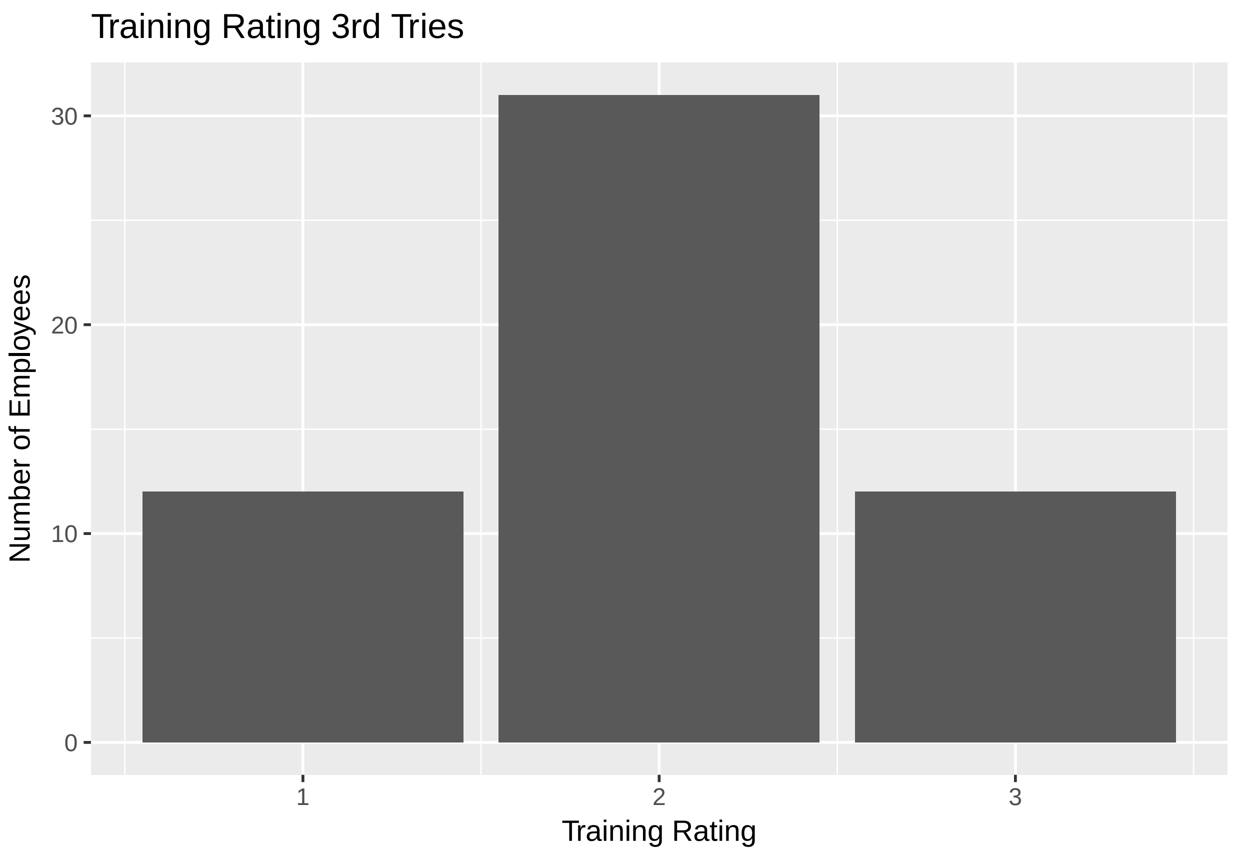
```r
ggplot(data = Number1, aes(x = Number1$Train.RatingN)) +
  geom_bar(aes(fill = Number1$Train.RatingN),position = "dodge")+
  ggtitle("Training Rating 1st Tries")+
  labs(x = "Training Rating", y = "Number of Employees")
```



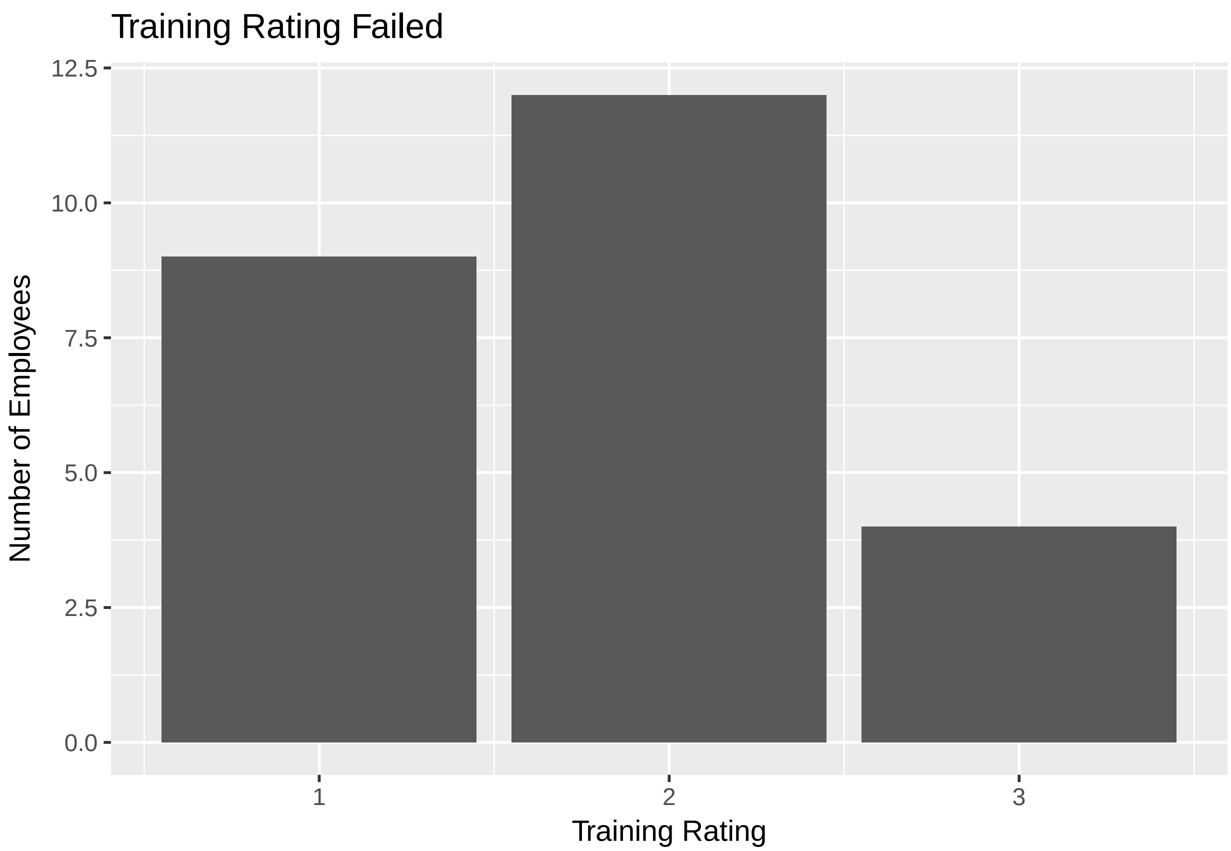
```
ggplot(data = Number2, aes(x = Number2$Train.RatingN)) +
  geom_bar(aes(fill = Number2$Train.RatingN),position = "dodge")+
   ggtitle("Training Rating 2nd Tries")+
  labs(x = "Training Rating", y = "Number of Employees")
```



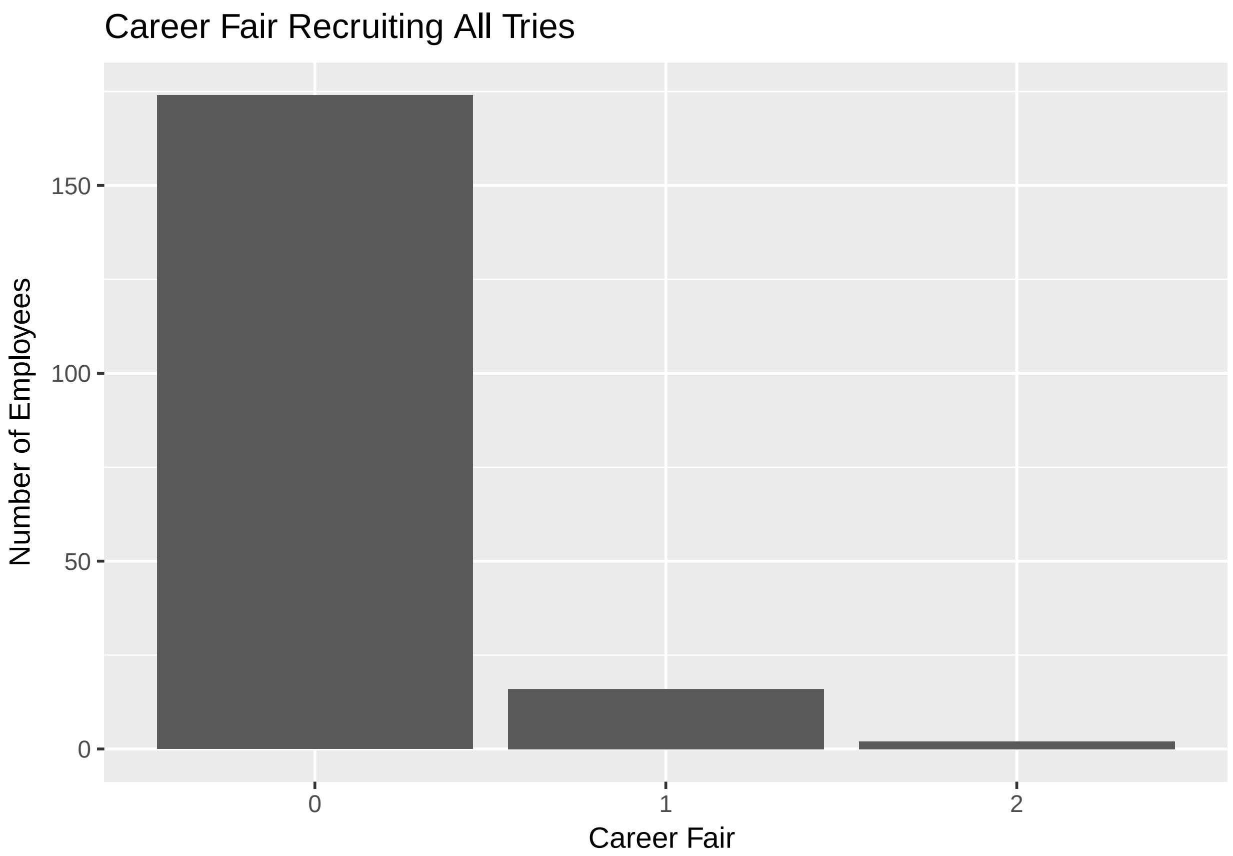
```
ggplot(data = Number3, aes(x = Number3$Train.RatingN)) +
  geom_bar(aes(fill = Number3$Train.RatingN), position = "dodge")+
  ggtitle("Training Rating 3rd Tries")+
  labs(x = "Training Rating", y = "Number of Employees")
```



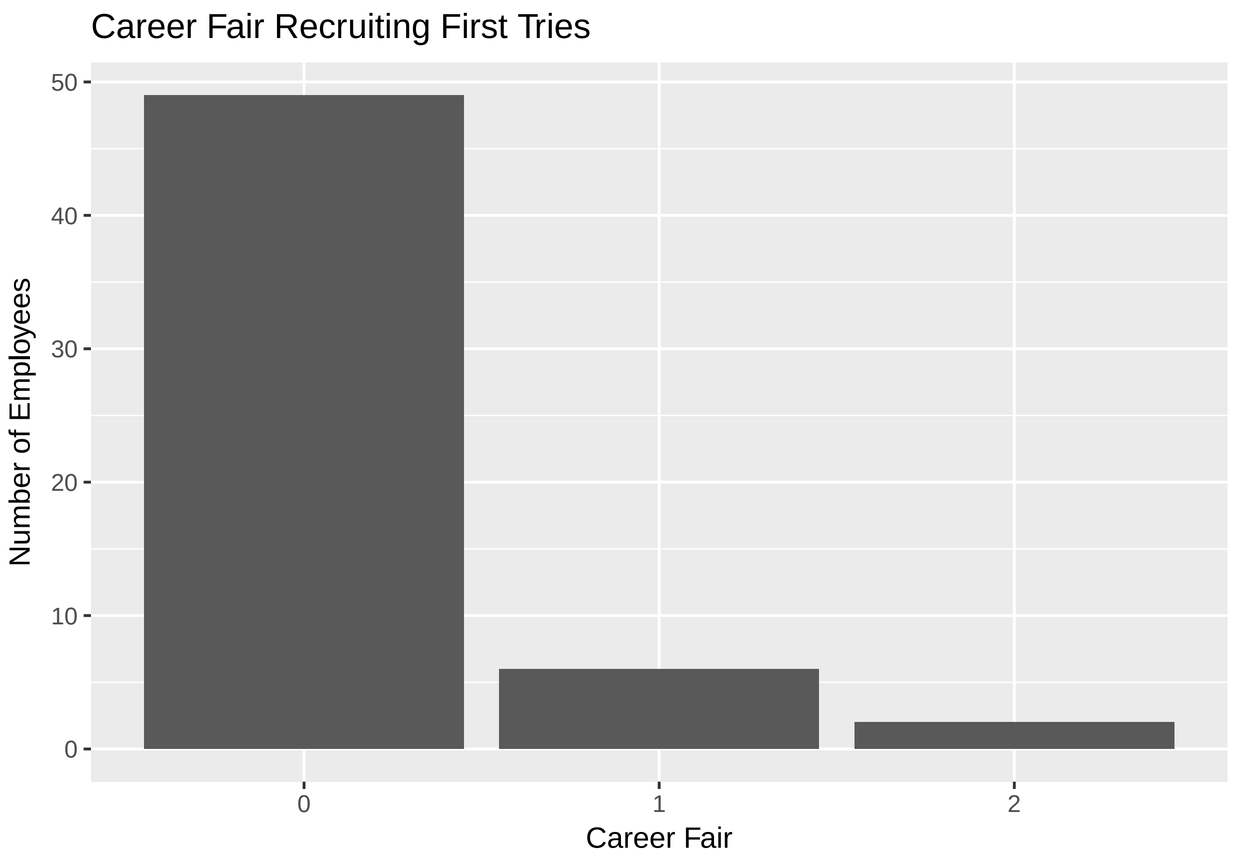
```
ggplot(data = Failed, aes(x = Failed$Train.RatingN)) +
  geom_bar(aes(fill = Failed$Train.RatingN), position = "dodge")+
  ggtitle("Training Rating Failed")+
  labs(x = "Training Rating", y = "Number of Employees")
```



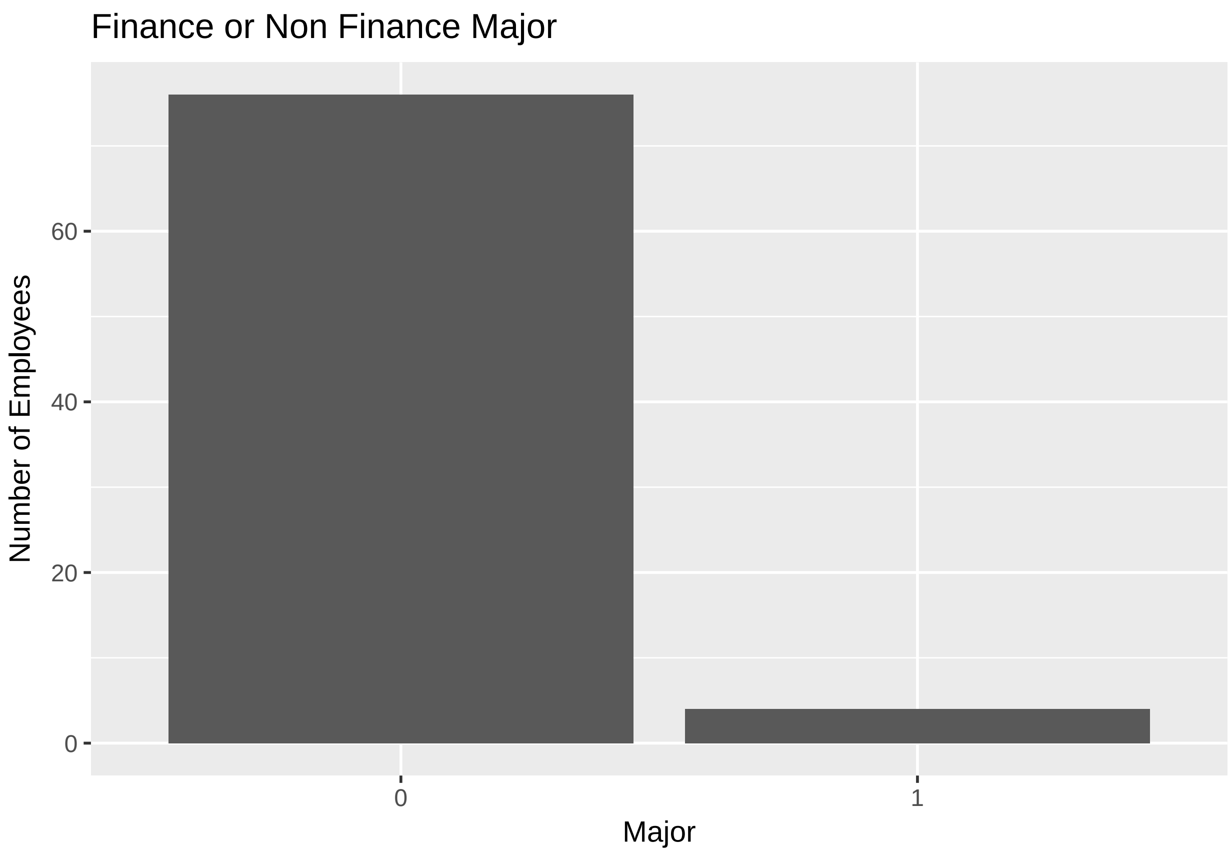
```
ggplot(data = People6, aes(x = People6$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = People6$`Learning Aptitude`), position = "dodge")+
  ggtitle("Career Fair Recruiting All Tries")+
  labs(x = "Career Fair ", y = "Number of Employees")
```



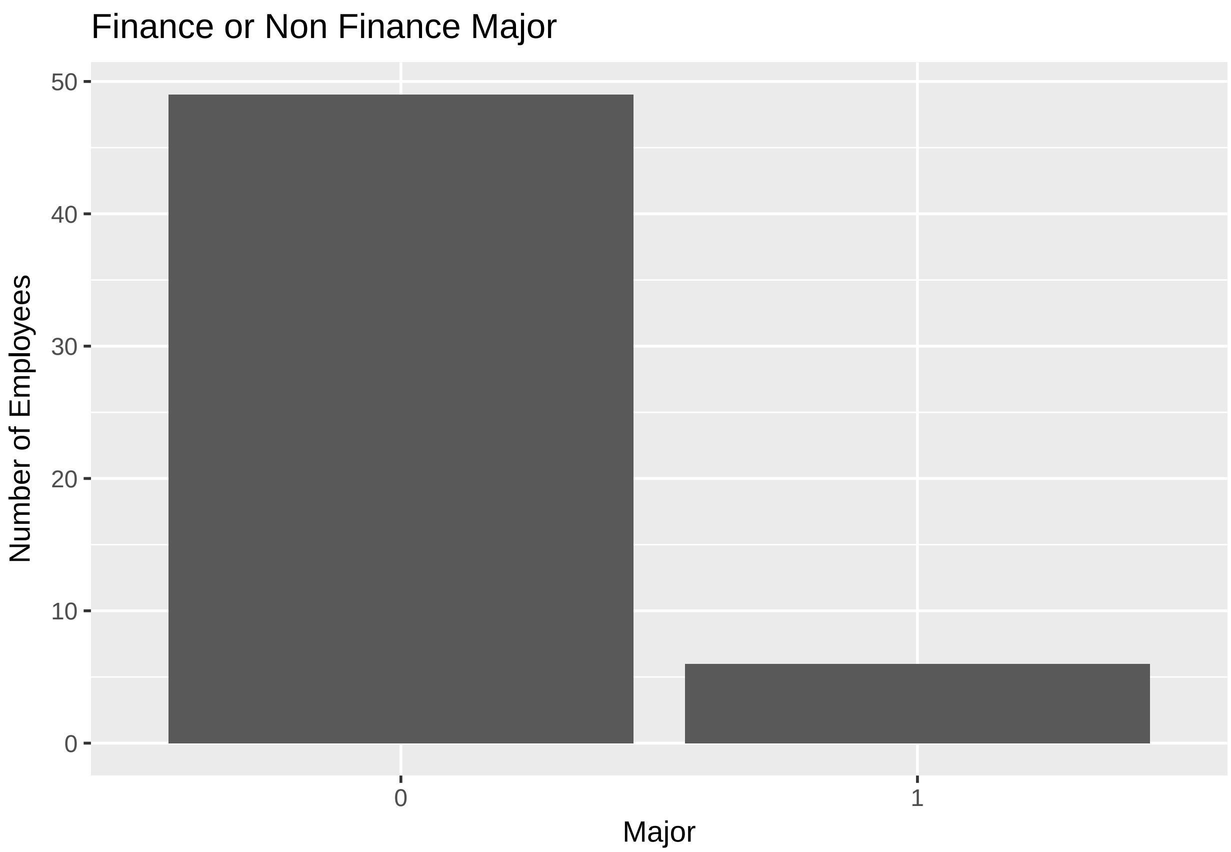
```
ggplot(data = Number1, aes(x = Number1$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number1$`Learning Aptitude`), position = "dodge")+
  ggtitle("Career Fair Recruiting First Tries")+
  labs(x = "Career Fair", y = "Number of Employees")
```



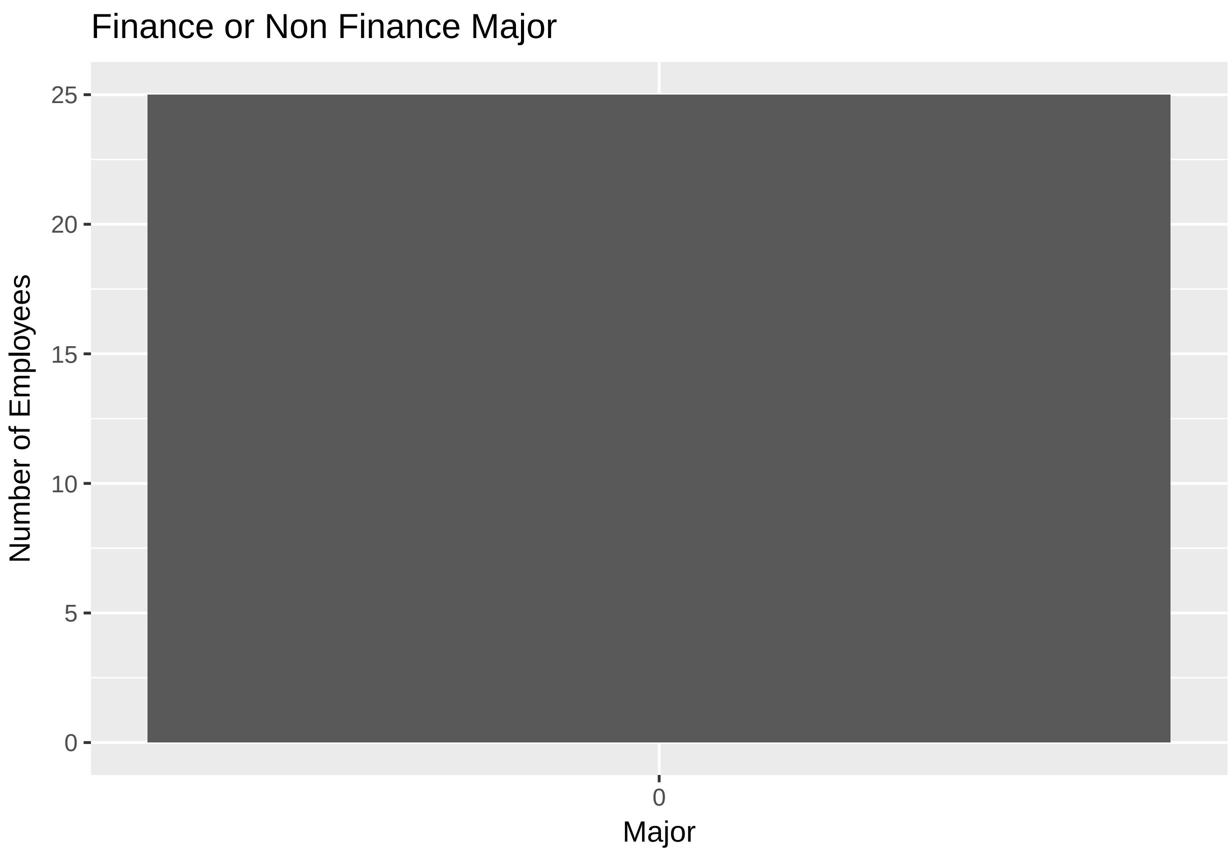
```
ggplot(data = Number2, aes(x = Number2$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number2$Test.Score), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



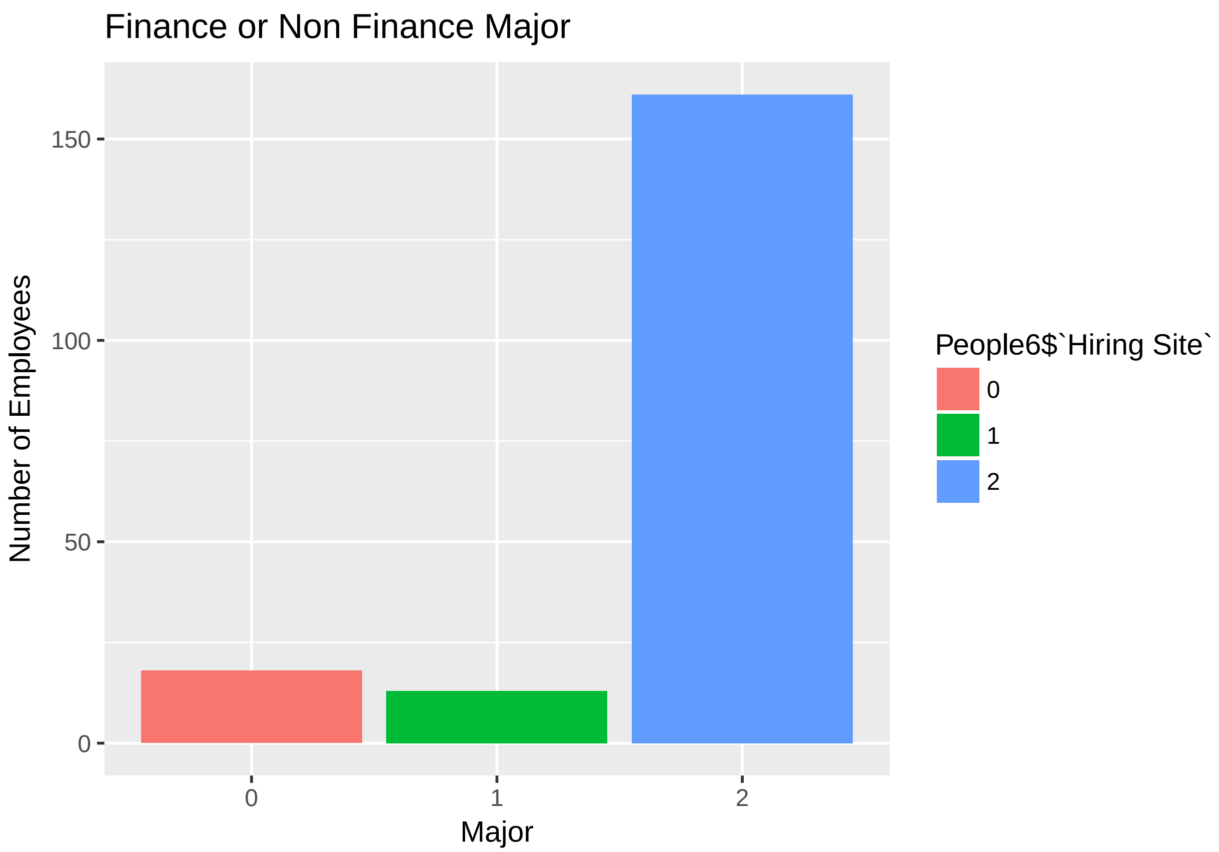
```r
ggplot(data = Number3, aes(x = Number3$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number3$Test.Score), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



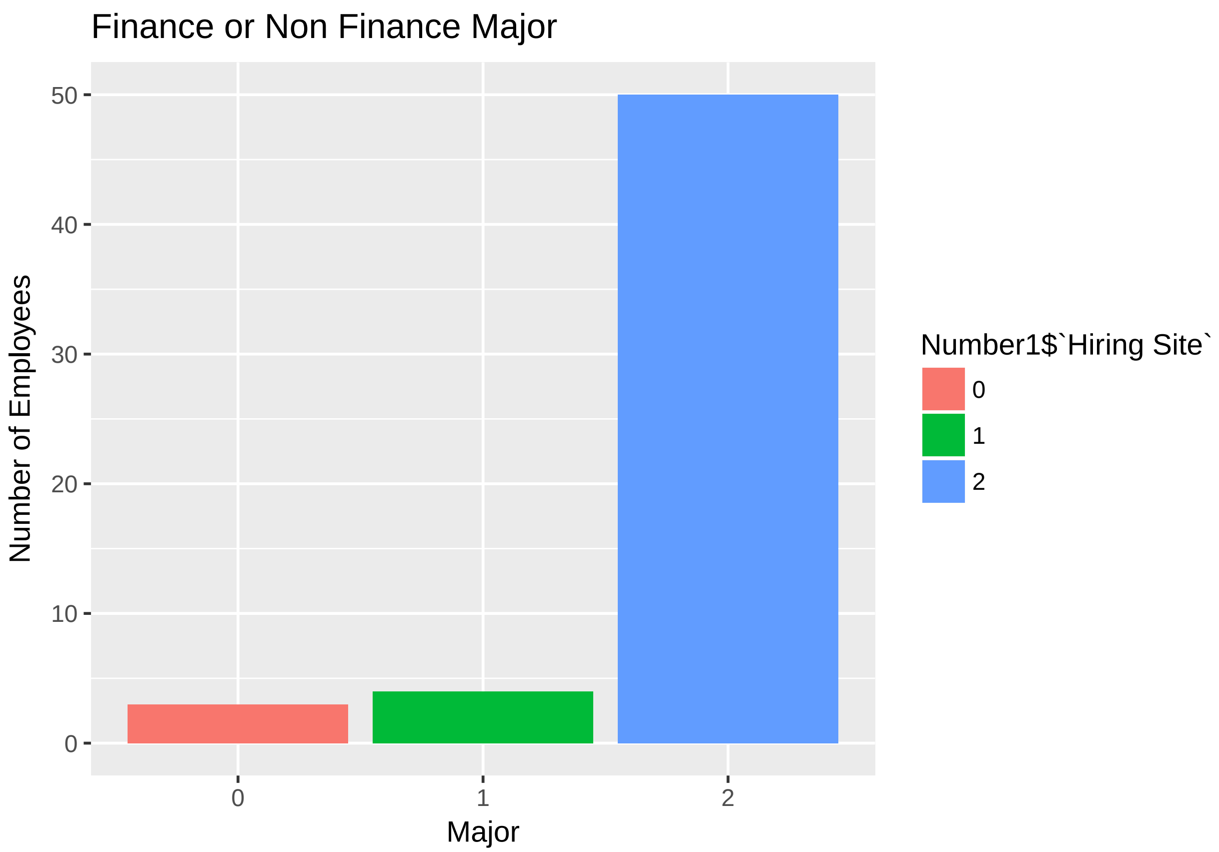
```
ggplot(data = Failed, aes(x = Failed$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Failed$Test.Score), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



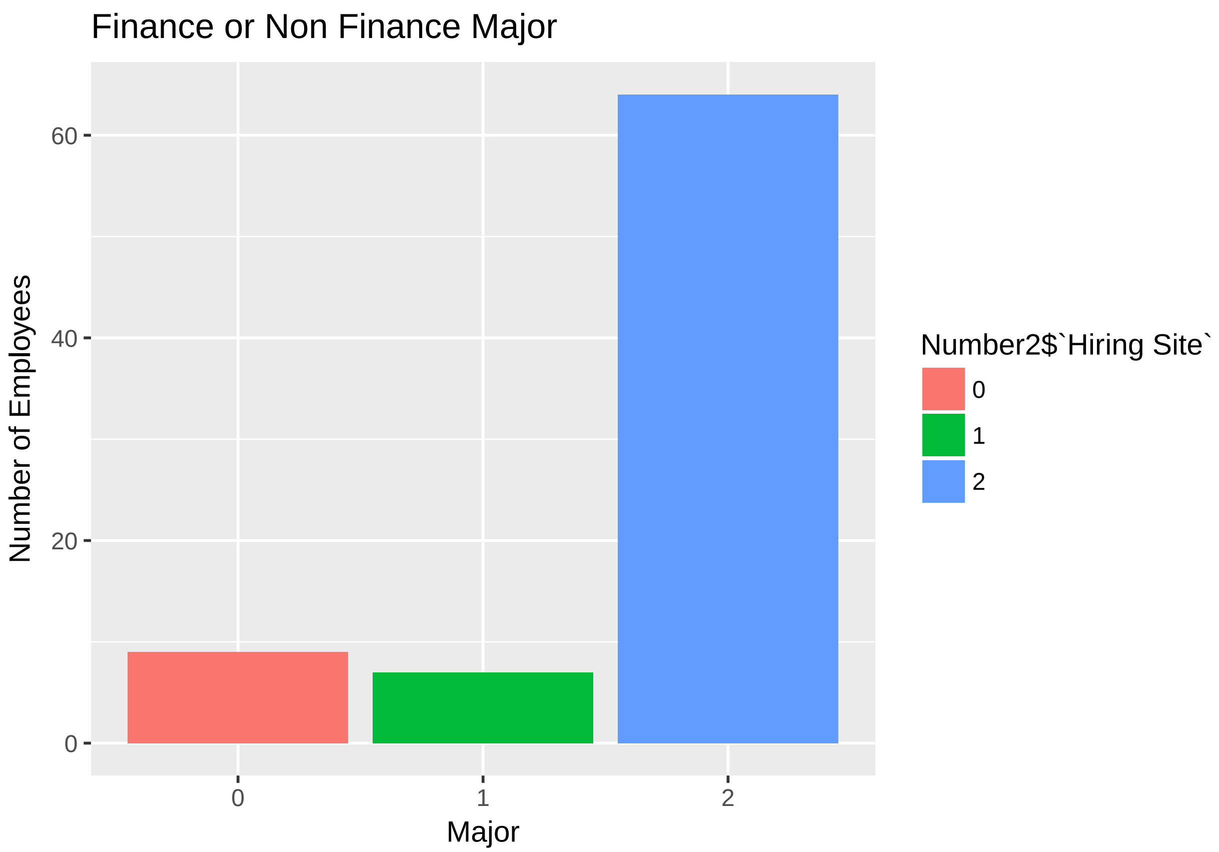
```
ggplot(data = People6, aes(x = People6$`Hiring Site`)) +
  geom_bar(aes(fill = People6$`Hiring Site`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



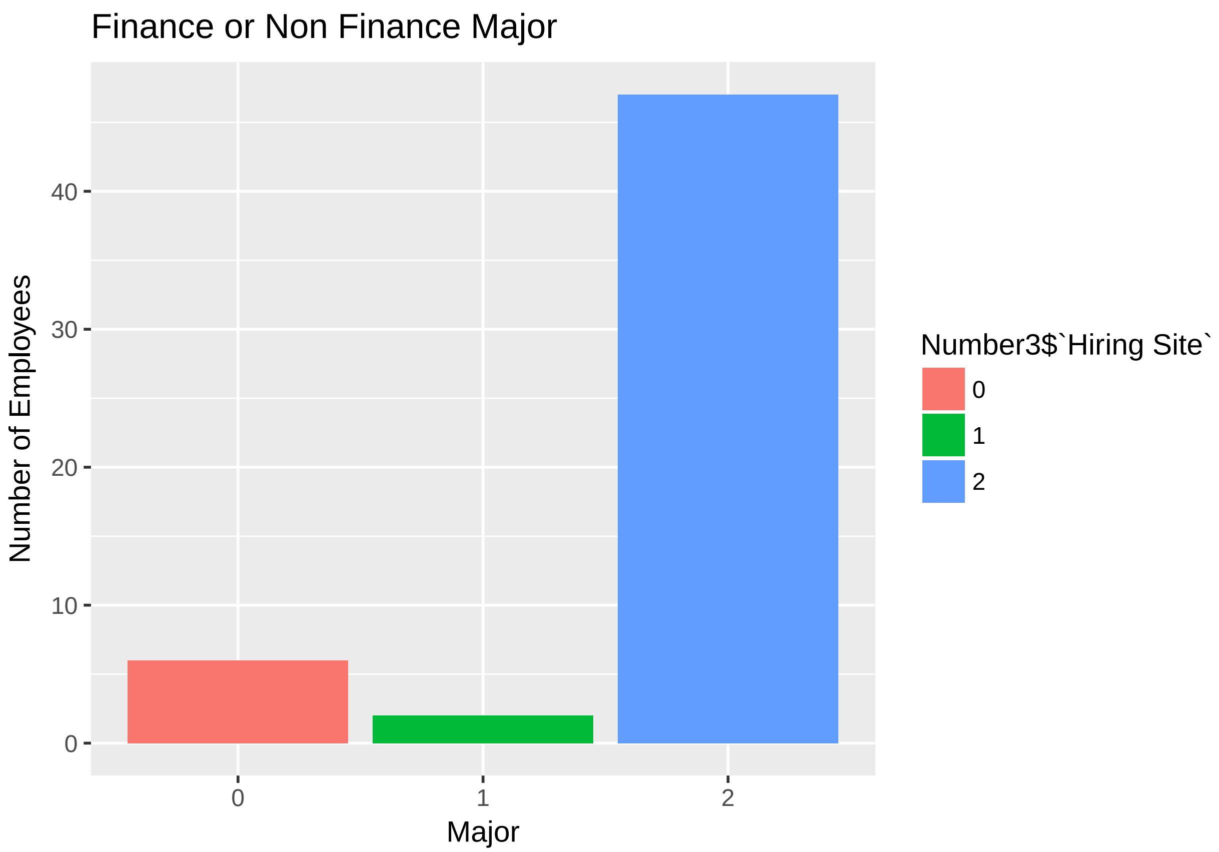
```
ggplot(data = Number1, aes(x = Number1$`Hiring Site`)) +
  geom_bar(aes(fill = Number1$`Hiring Site`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



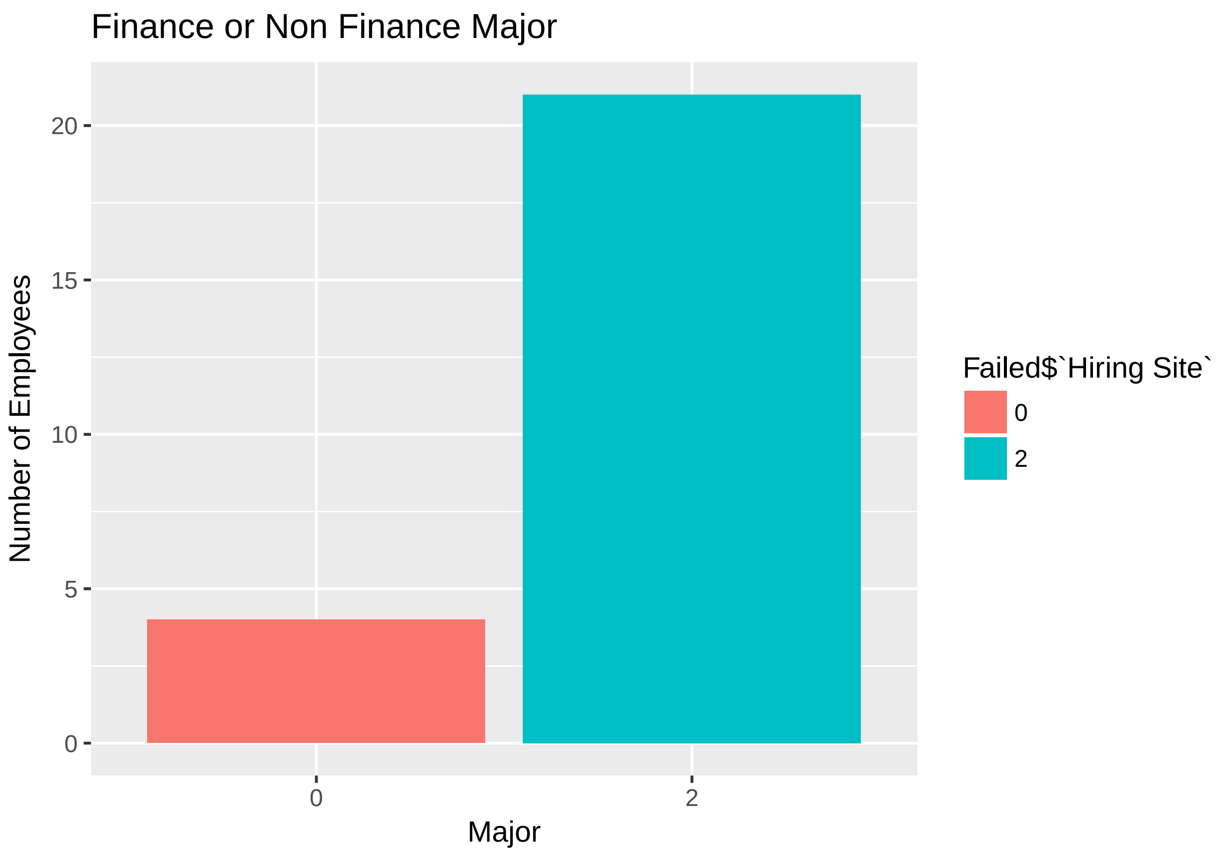
```r
ggplot(data = Number2, aes(x = Number2$`Hiring Site`)) +
  geom_bar(aes(fill = Number2$`Hiring Site`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



```
ggplot(data = Number3, aes(x = Number3$`Hiring Site`)) +
  geom_bar(aes(fill = Number3$`Hiring Site`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



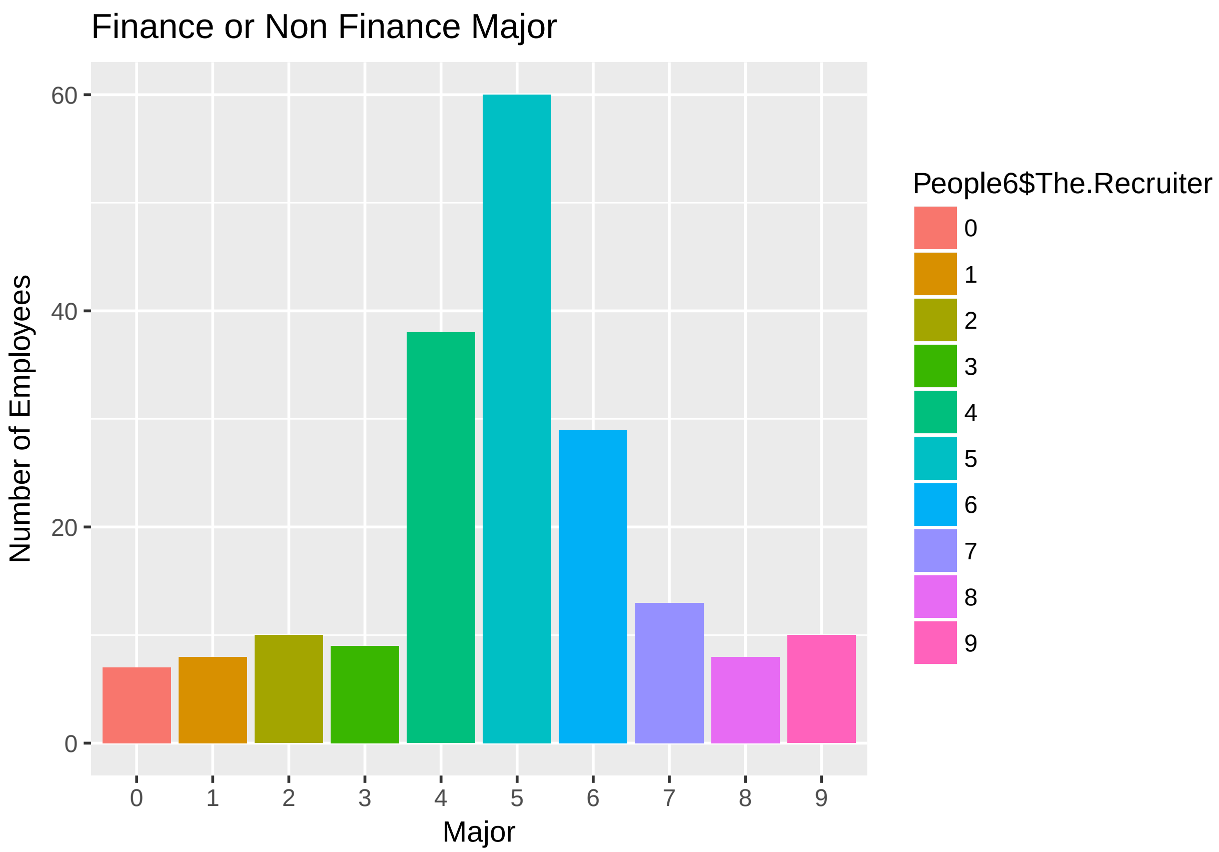
```r
ggplot(data = Failed, aes(x = Failed$`Hiring Site`)) +
  geom_bar(aes(fill = Failed$`Hiring Site`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



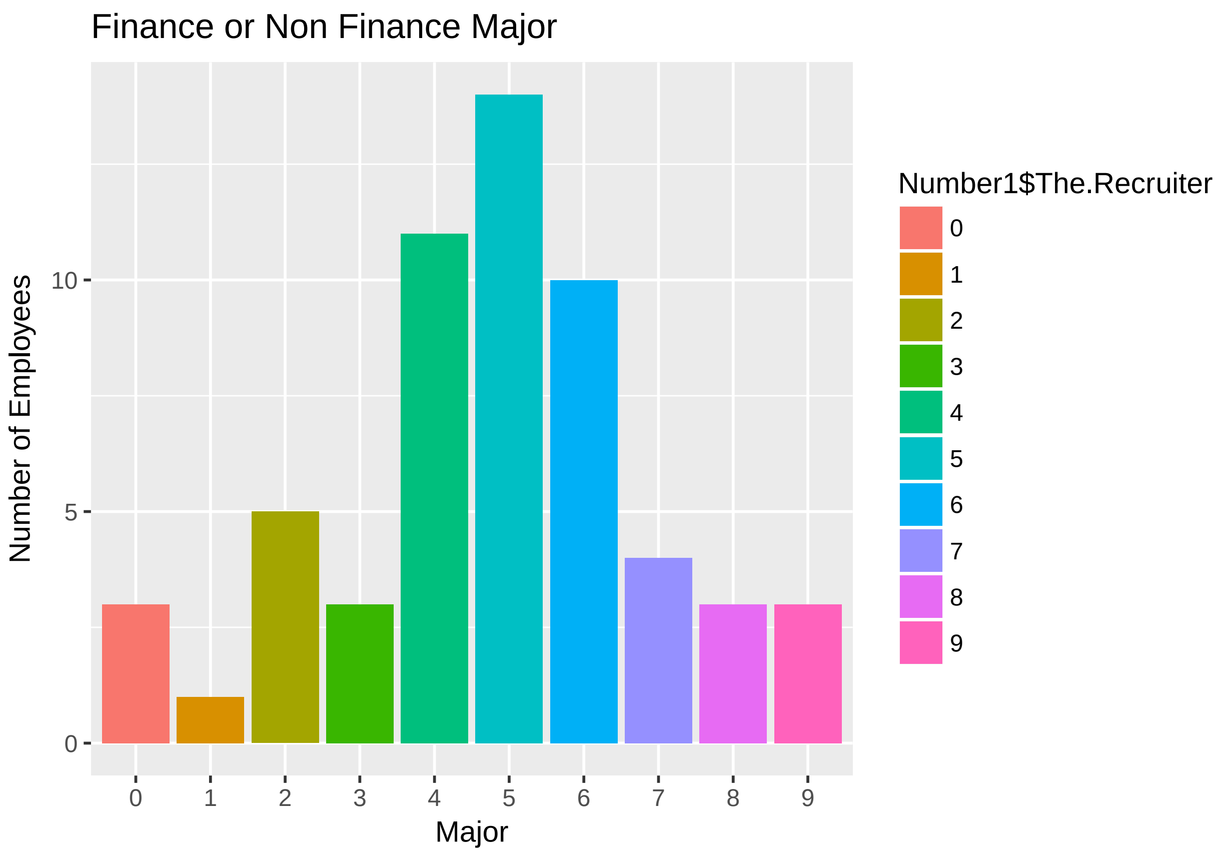
```
summary(People6$`Hiring Site`)
```

```
##   0   1   2 
##  18  13 161
```

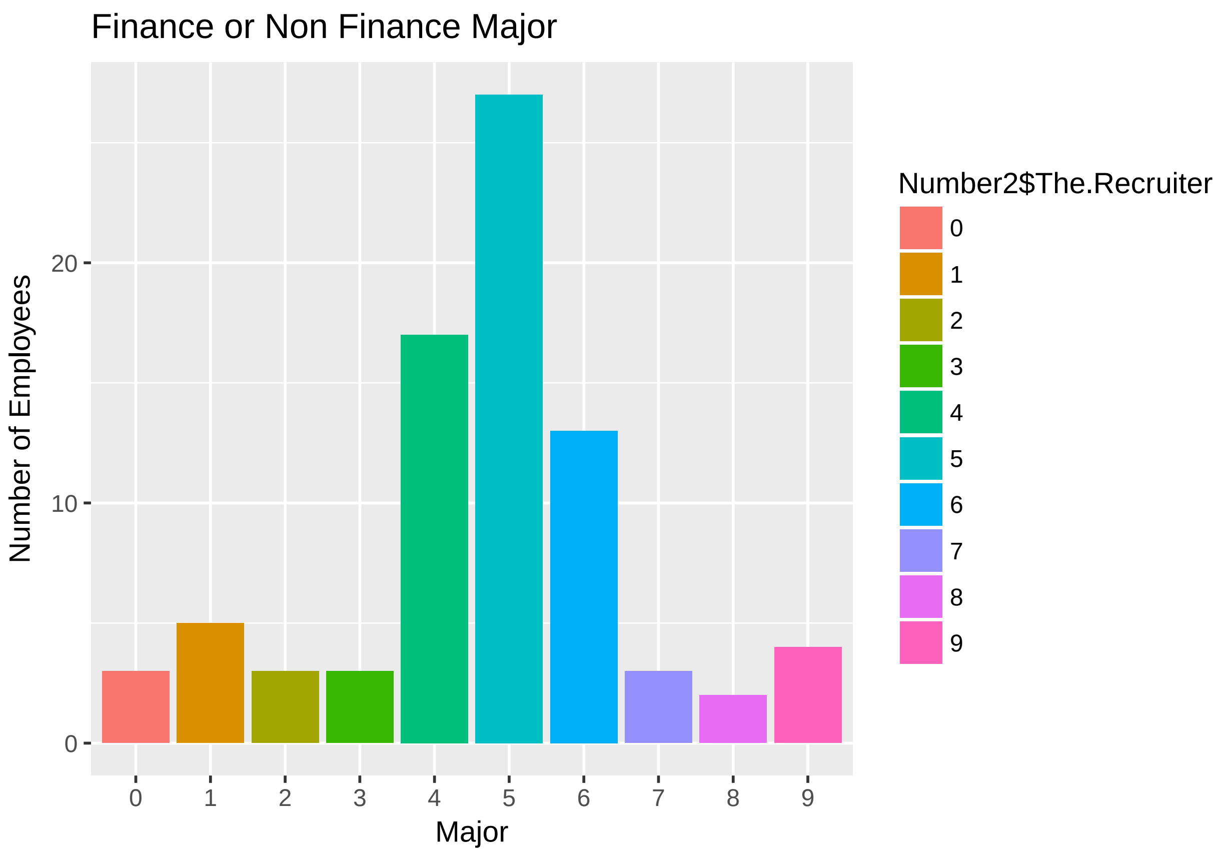
```
ggplot(data = People6, aes(x = People6$The.Recruiter)) +
  geom_bar(aes(fill = People6$The.Recruiter), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



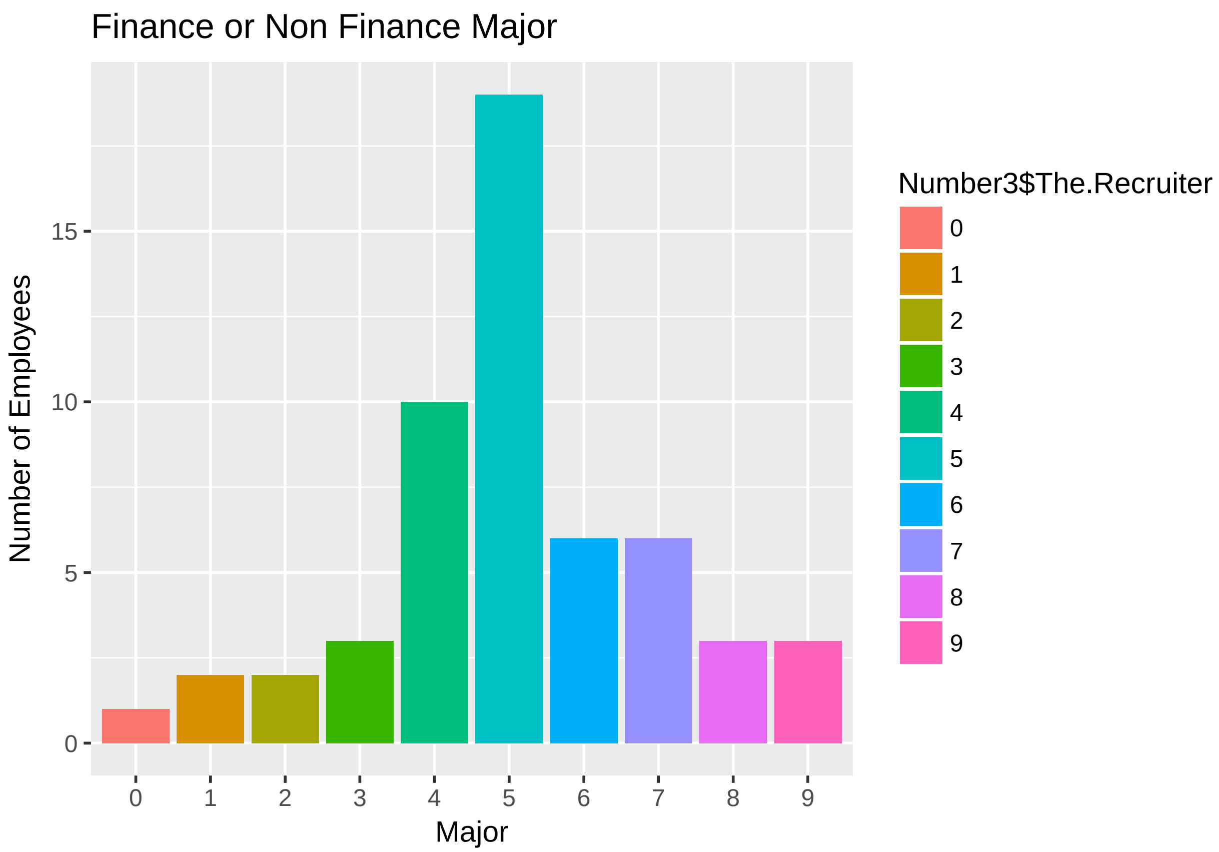
```
ggplot(data = Number1, aes(x = Number1$The.Recruiter)) +
  geom_bar(aes(fill = Number1$The.Recruiter), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



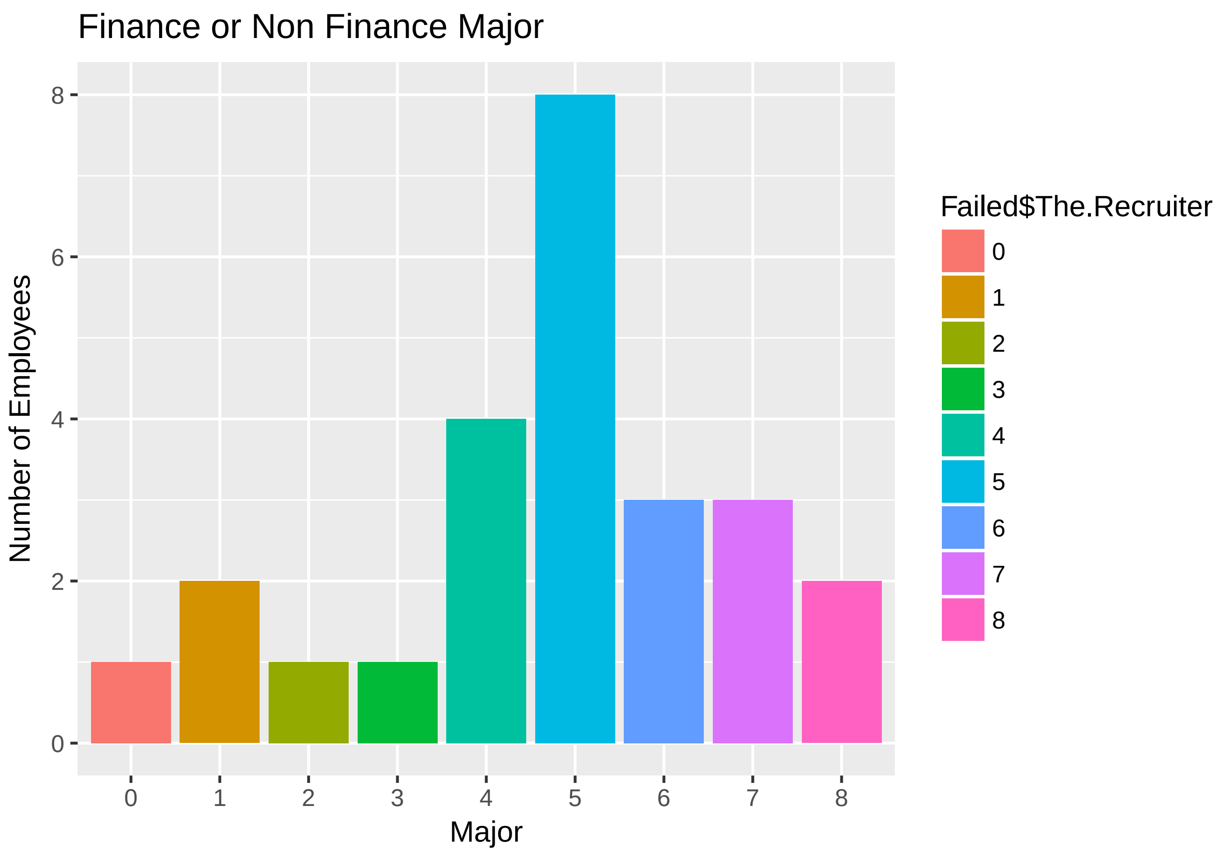
```r
ggplot(data = Number2, aes(x = Number2$The.Recruiter)) +
  geom_bar(aes(fill = Number2$The.Recruiter), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



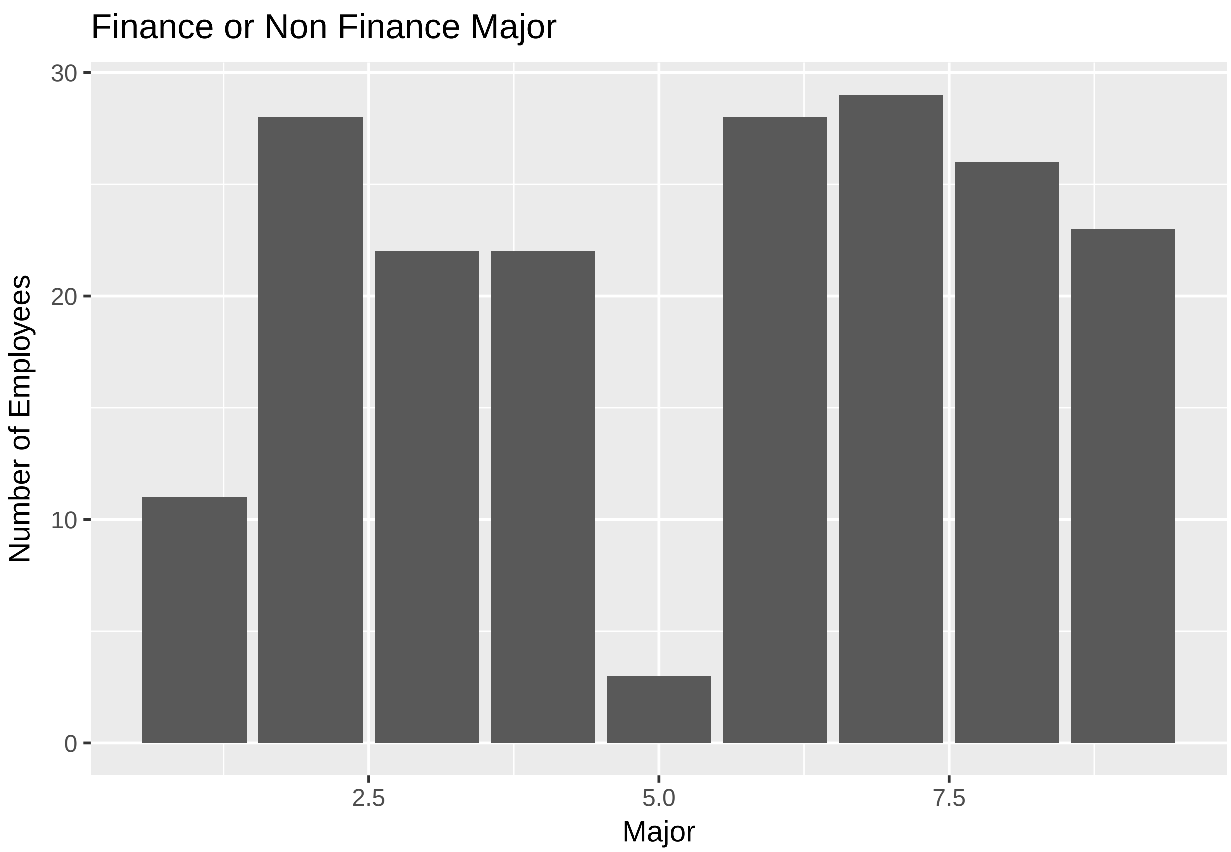
```
ggplot(data = Number3, aes(x = Number3$The.Recruiter)) +
  geom_bar(aes(fill = Number3$The.Recruiter), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



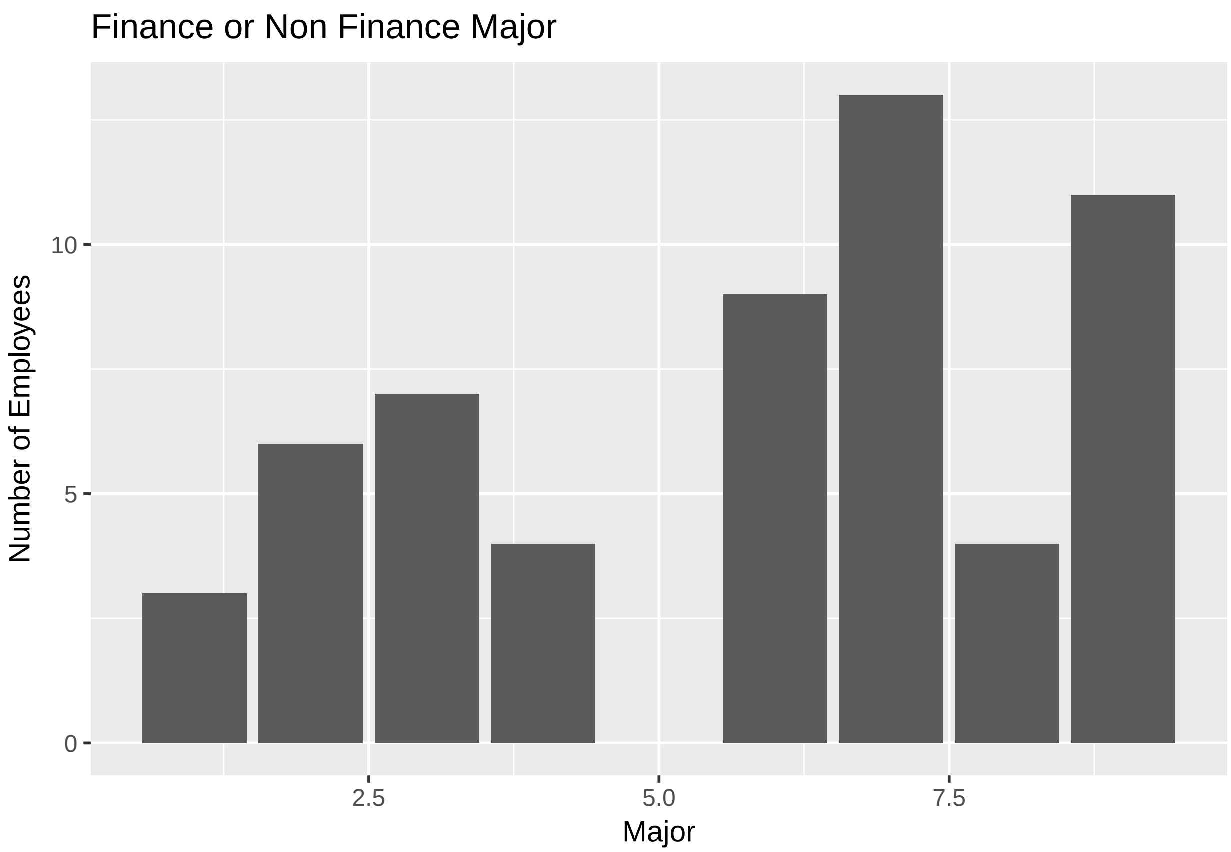
```
ggplot(data = Failed, aes(x = Failed$The.Recruiter)) +
  geom_bar(aes(fill = Failed$The.Recruiter), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



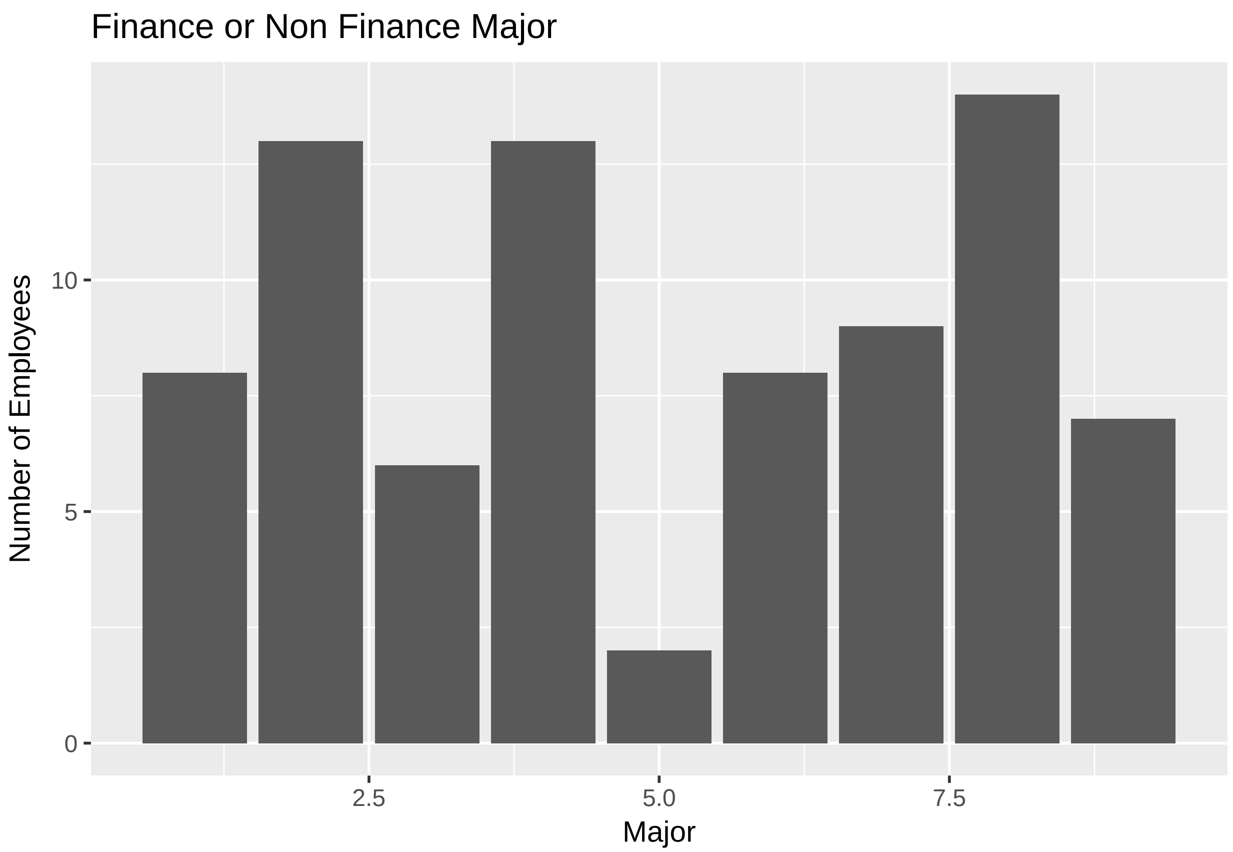
```r
ggplot(data = People6, aes(x = People6$Motivation)) +
  geom_bar(aes(fill = People6$Motivation), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



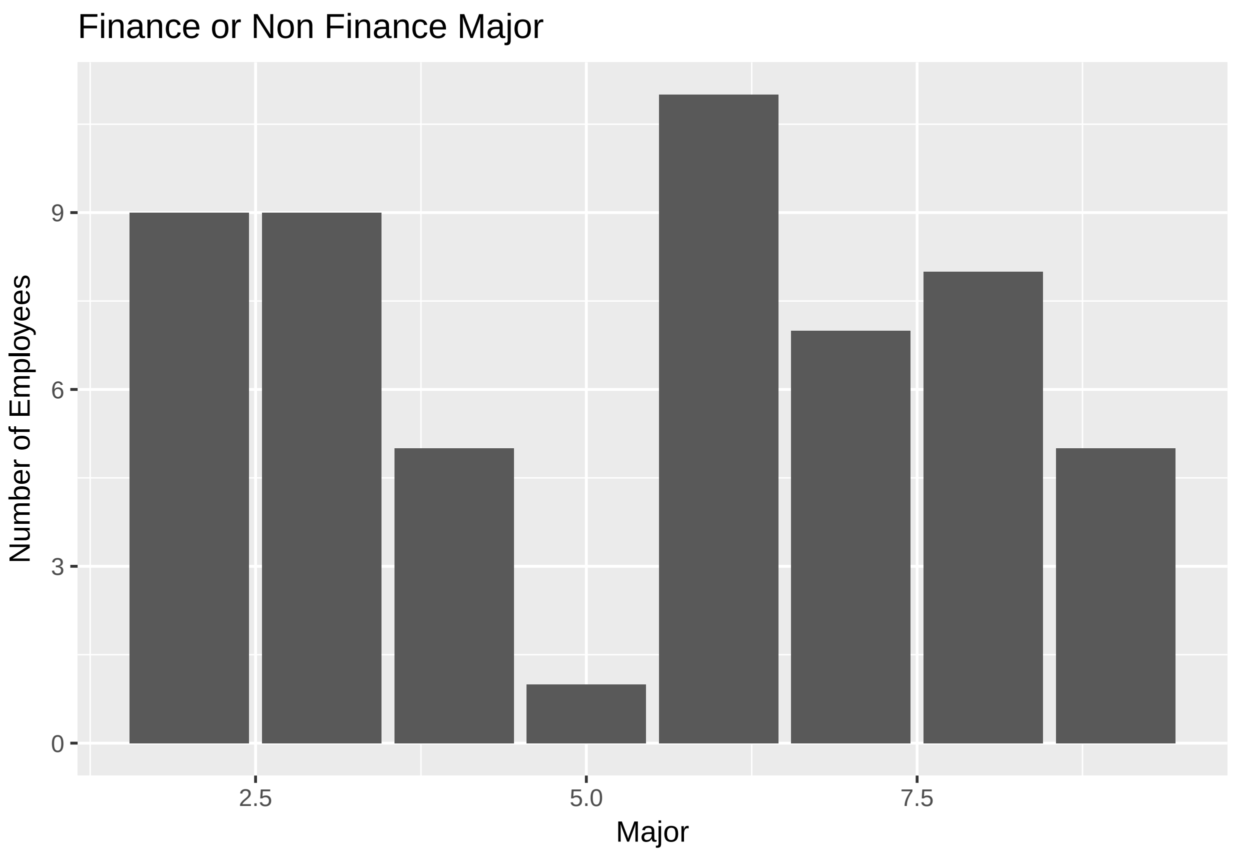
```
ggplot(data = Number1, aes(x = Number1$Motivation)) +
  geom_bar(aes(fill = Number1$Motivation), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



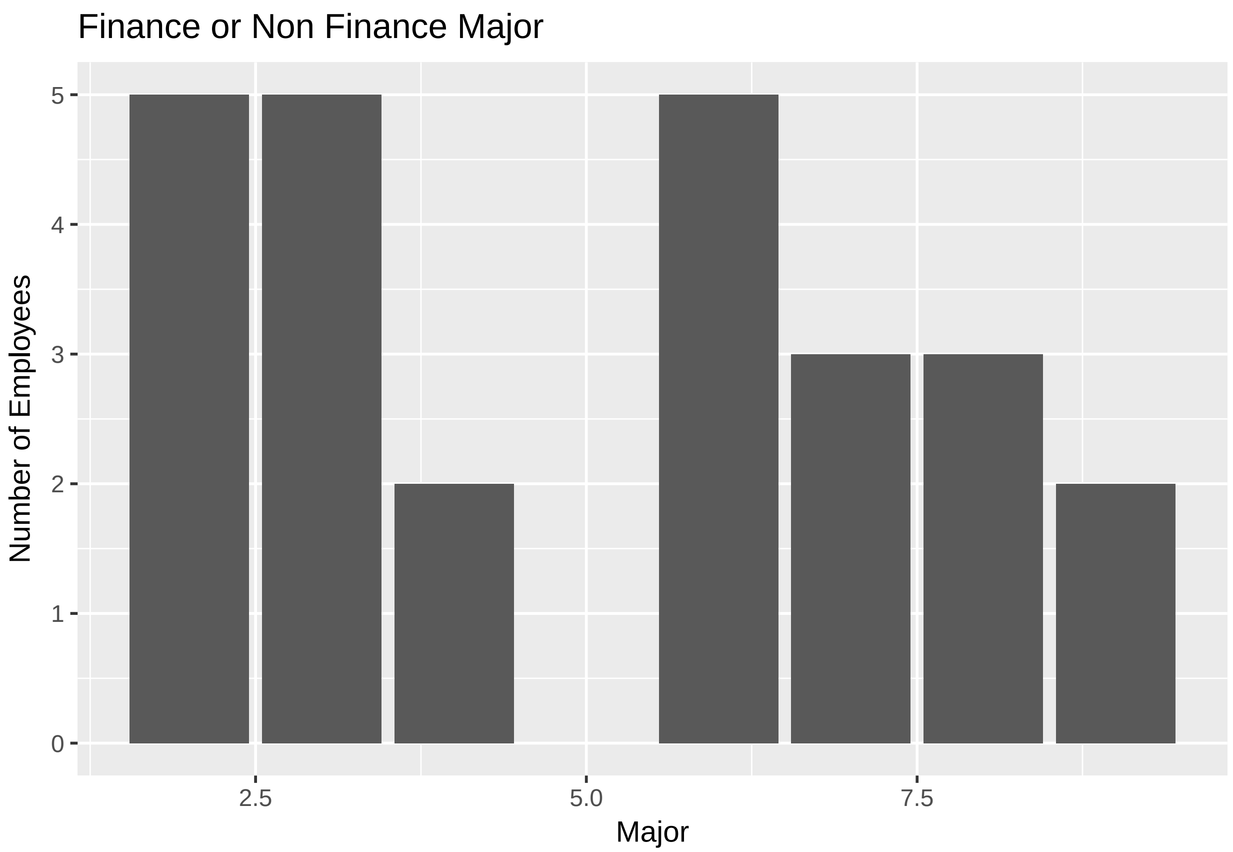
```
ggplot(data = Number2, aes(x = Number2$Motivation)) +
  geom_bar(aes(fill = Number2$Motivation), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



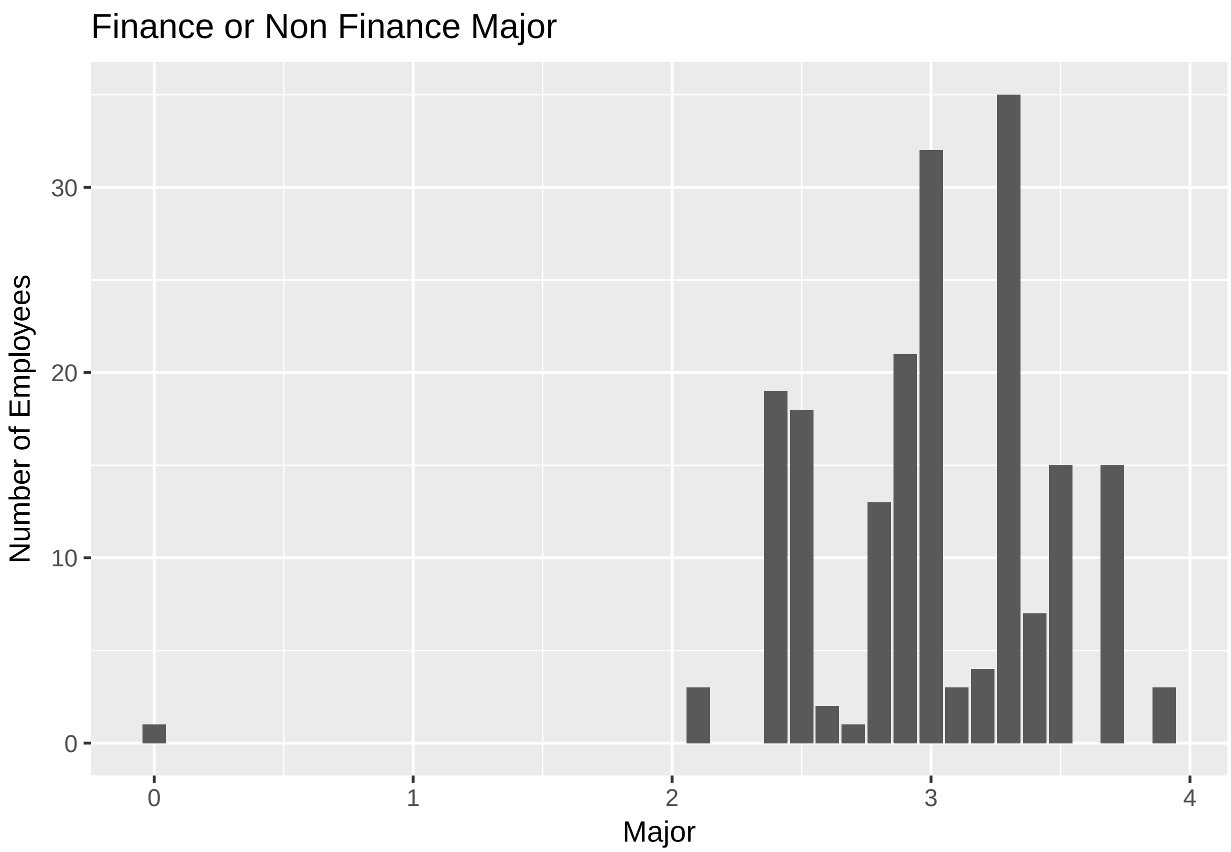
```
ggplot(data = Number3, aes(x = Number3$Motivation)) +
  geom_bar(aes(fill = Number3$Motivation), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



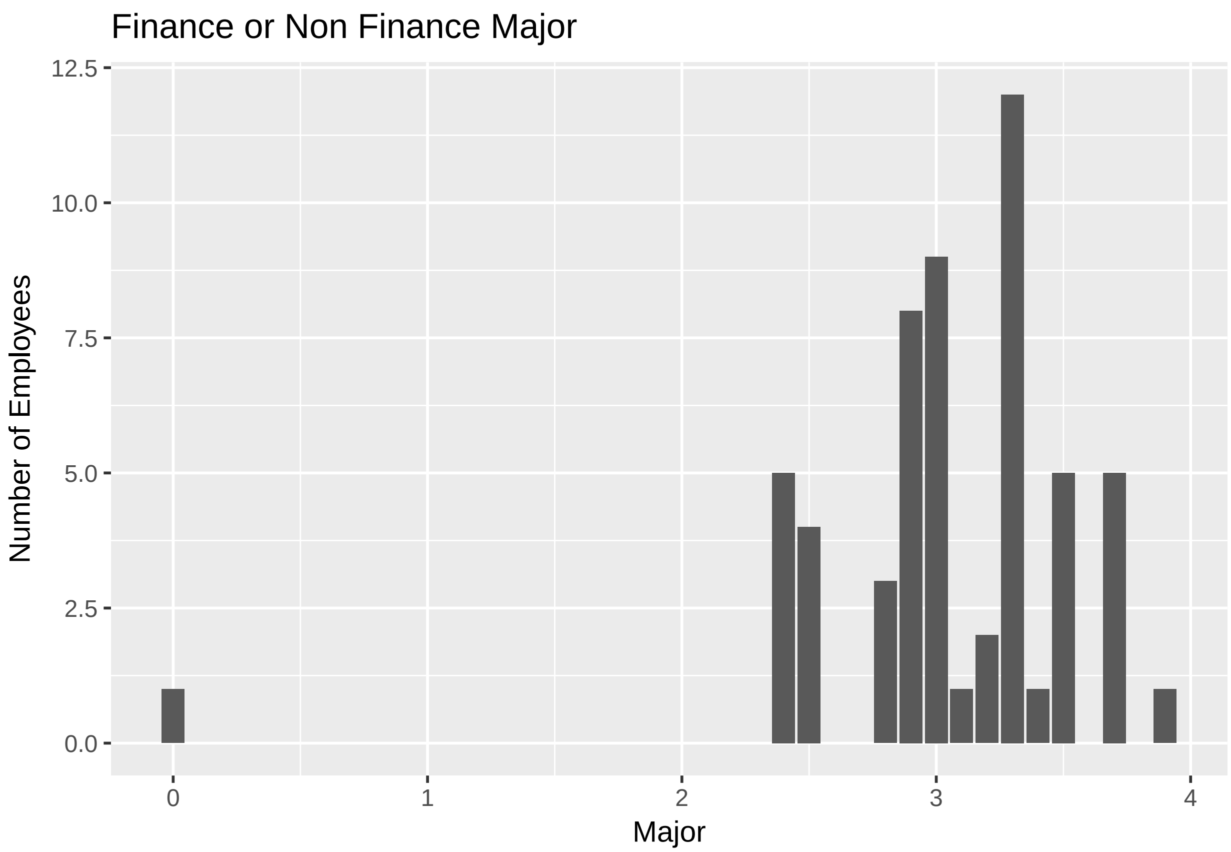
```
ggplot(data = Failed, aes(x = Failed$Motivation)) +
  geom_bar(aes(fill = Failed$Motivation), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



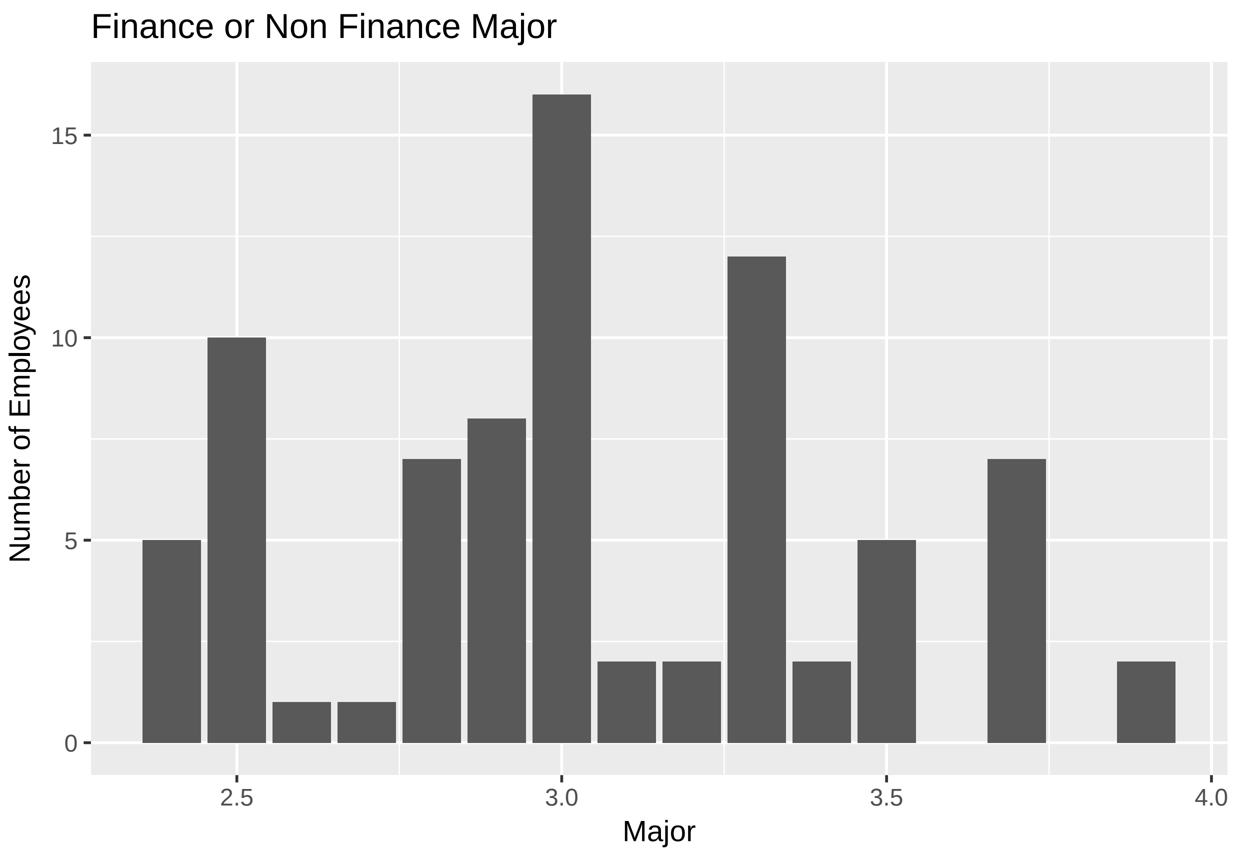
```
ggplot(data = People6, aes(x = People6$College.GPA)) +
  geom_bar(aes(fill = People6$College.GPA), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



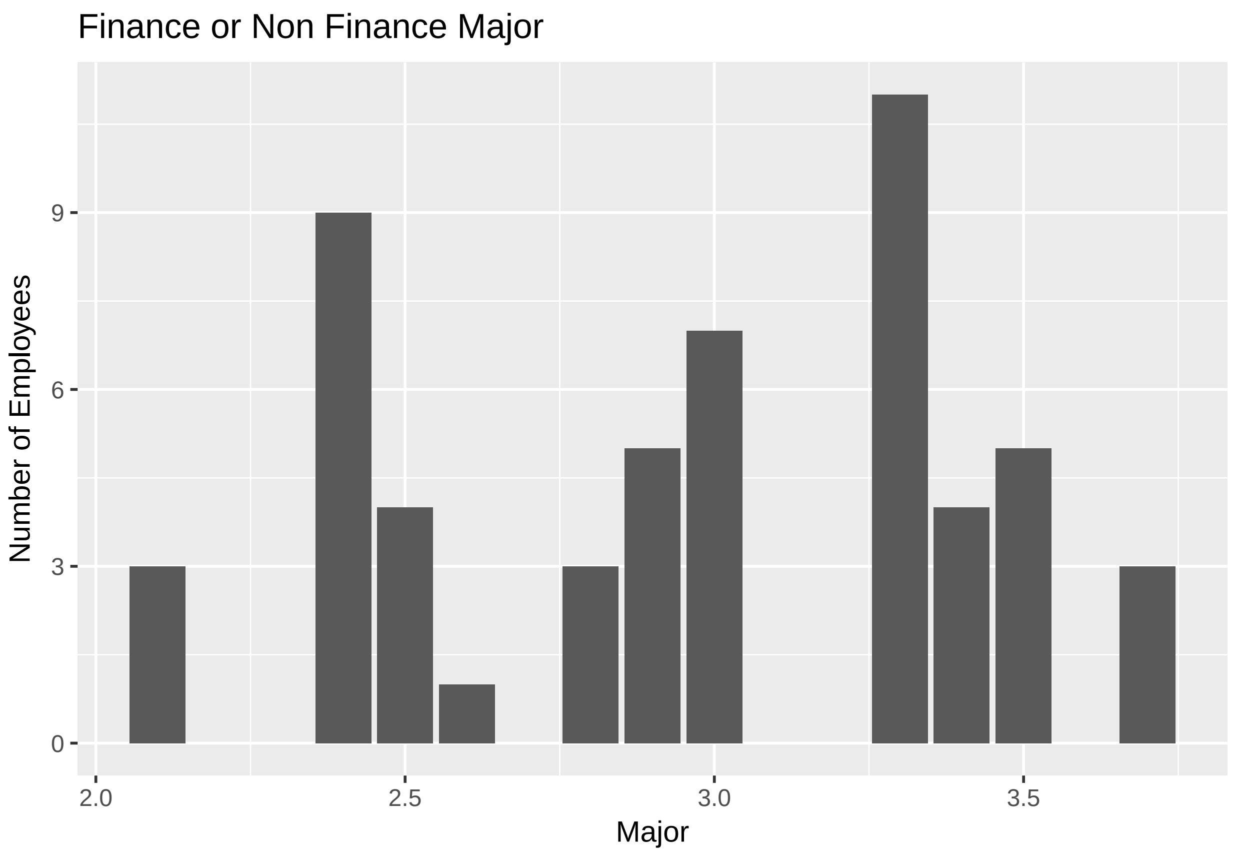
```
ggplot(data = Number1, aes(x = Number1$College.GPA)) +
  geom_bar(aes(fill = Number1$College.GPA), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



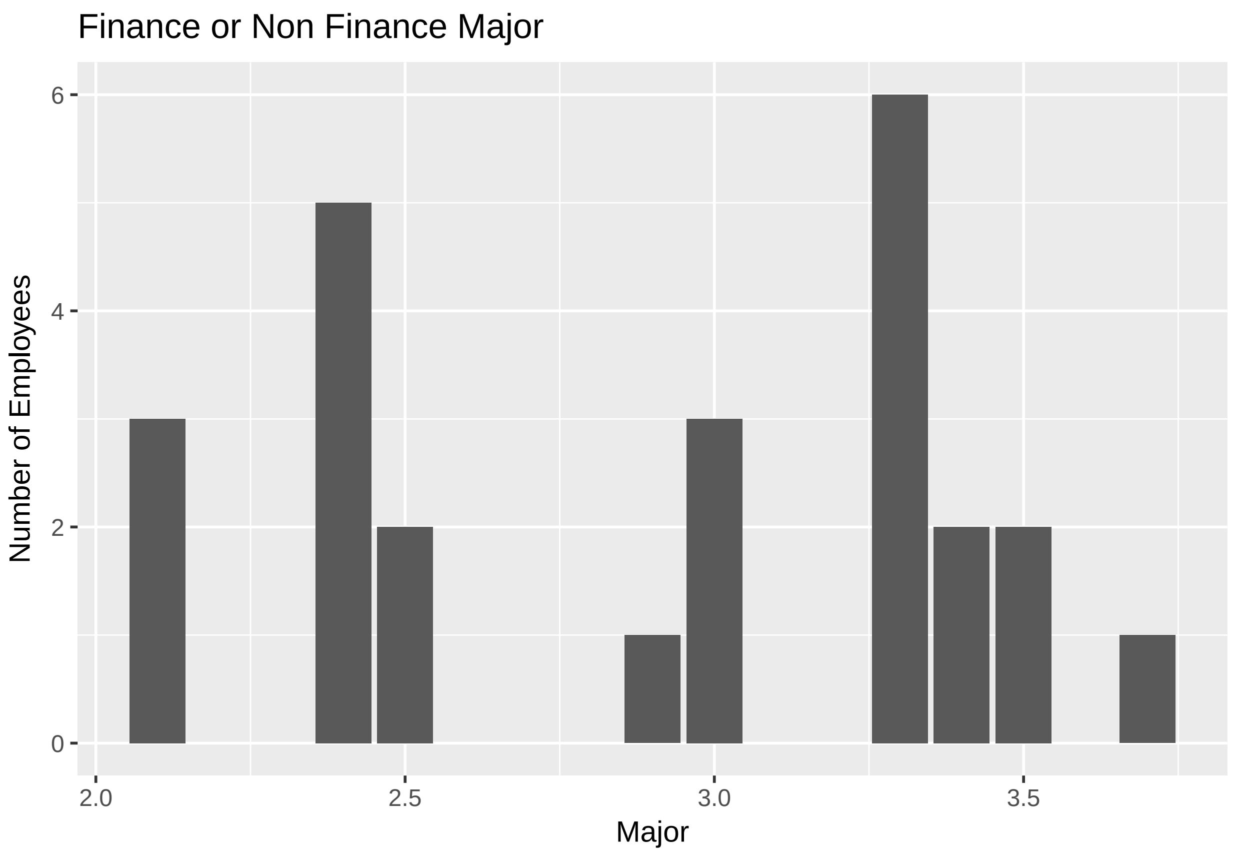
```
ggplot(data = Number2, aes(x = Number2$College.GPA)) +
  geom_bar(aes(fill = Number2$College.GPA), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



```
ggplot(data = Number3, aes(x = Number3$College.GPA)) +
  geom_bar(aes(fill = Number3$College.GPA), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```

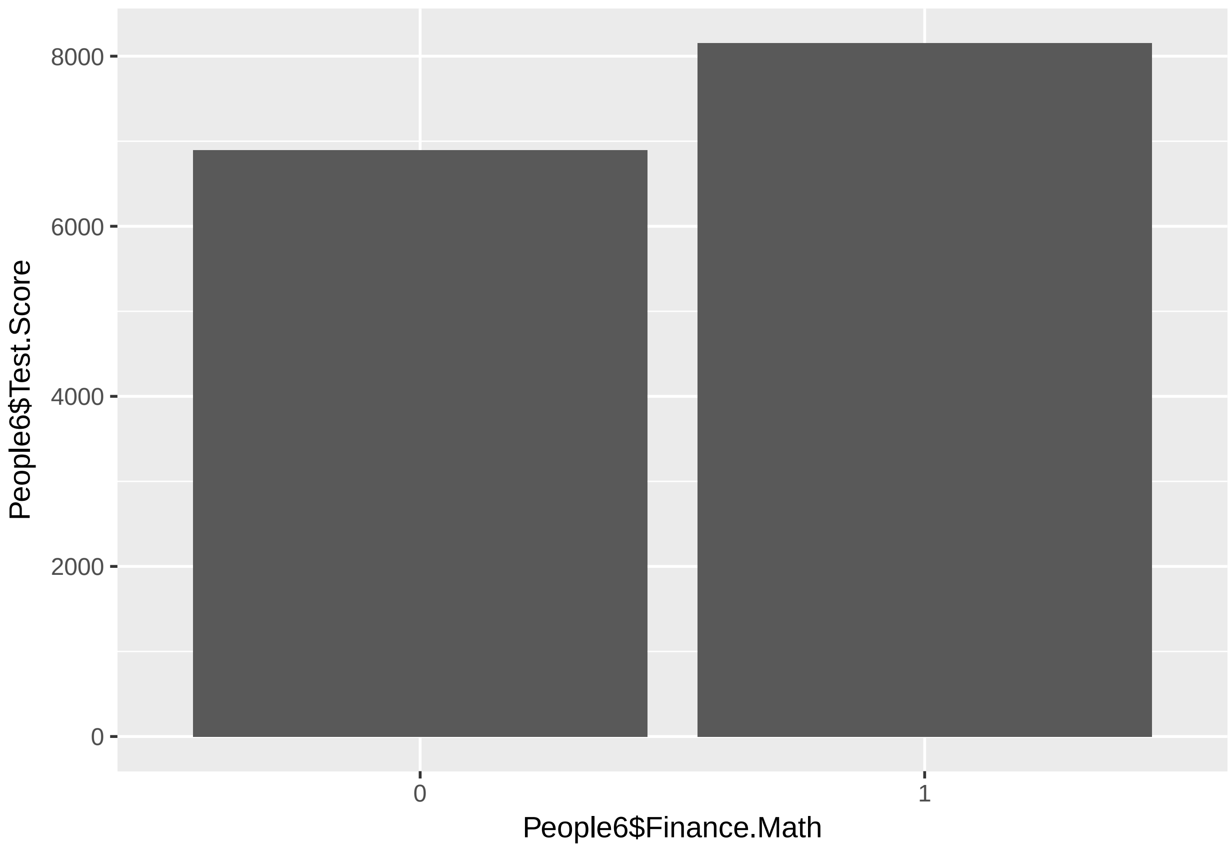


```
ggplot(data = Failed, aes(x = Failed$College.GPA)) +
  geom_bar(aes(fill = Failed$College.GPA), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```

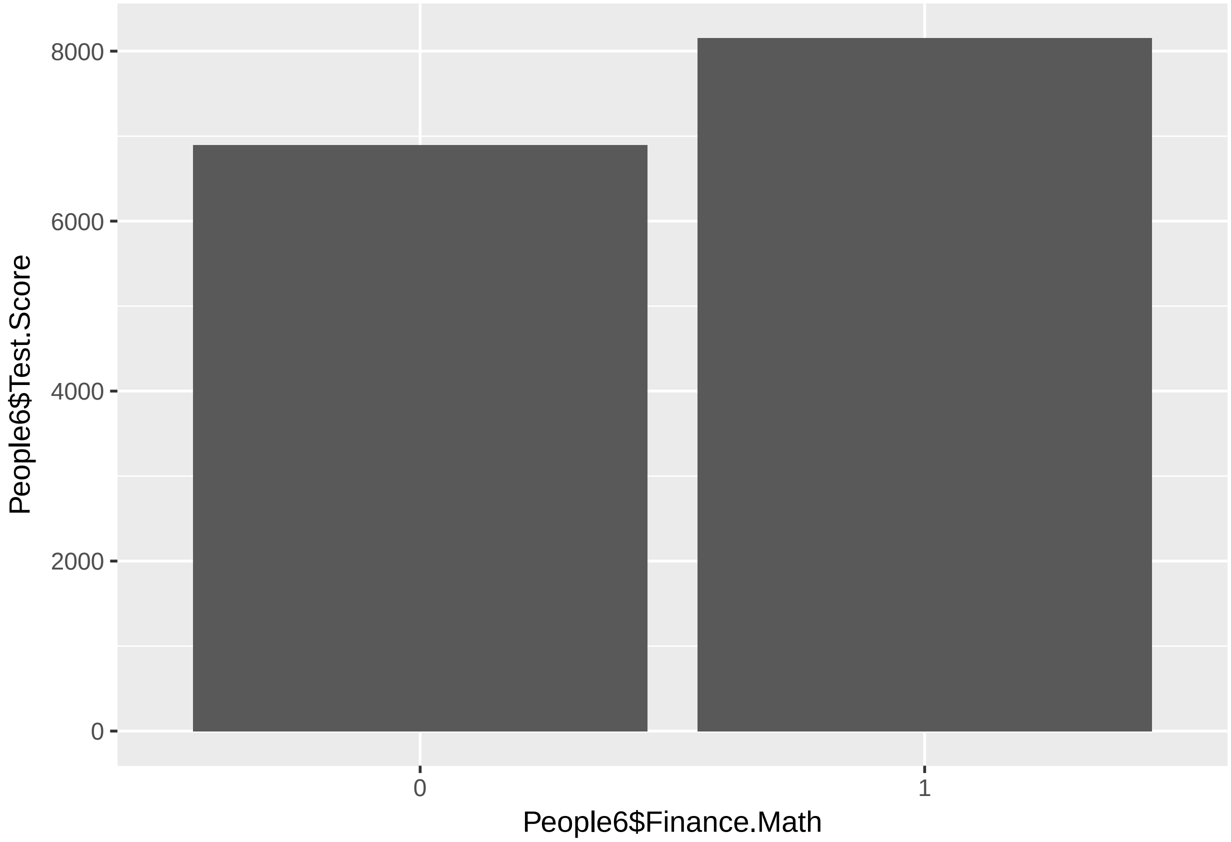


```
Meow  = dplyr::filter(Failed, Failed$College.GPA >= 3.0)
```

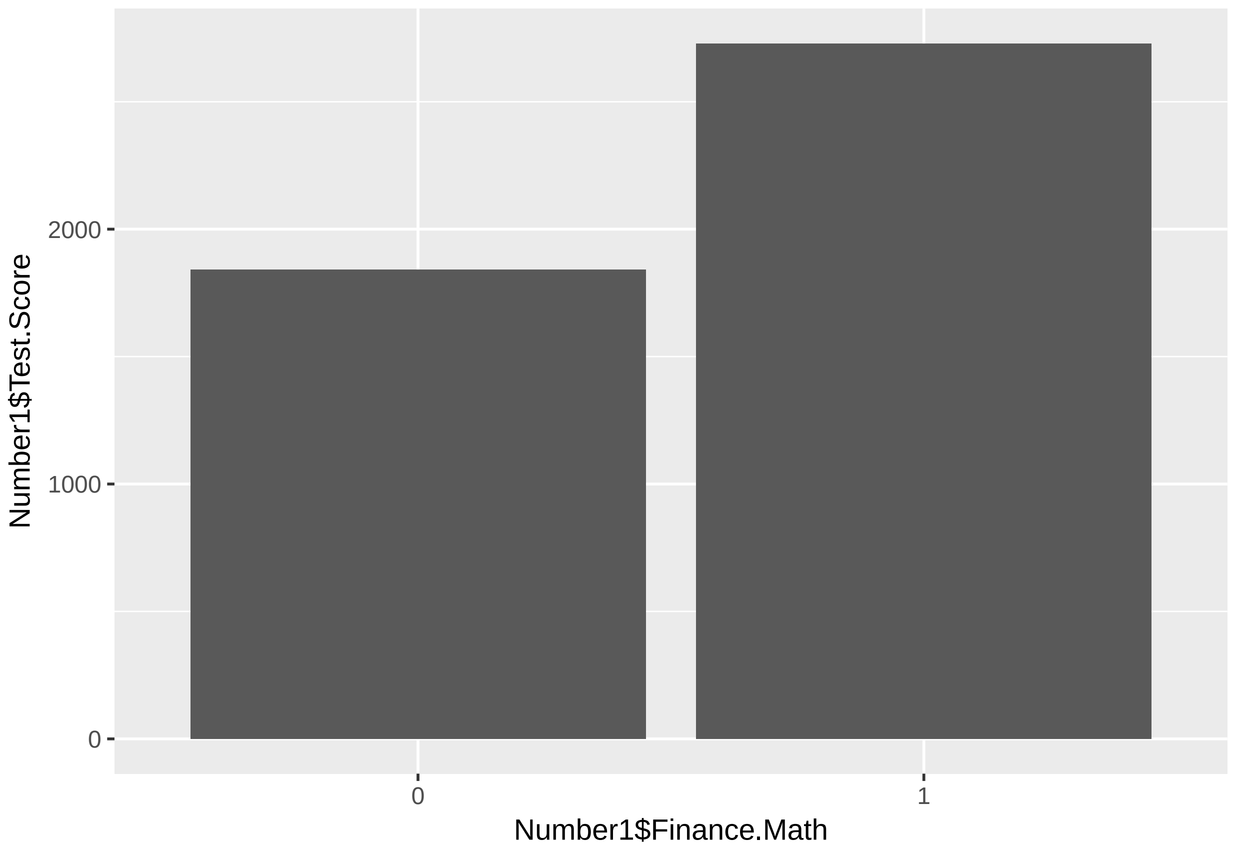
```
ggplot(data = People6, aes(x = People6$Finance.Math, y = People6$Test.Score)) +
  geom_bar(stat = "identity")
```



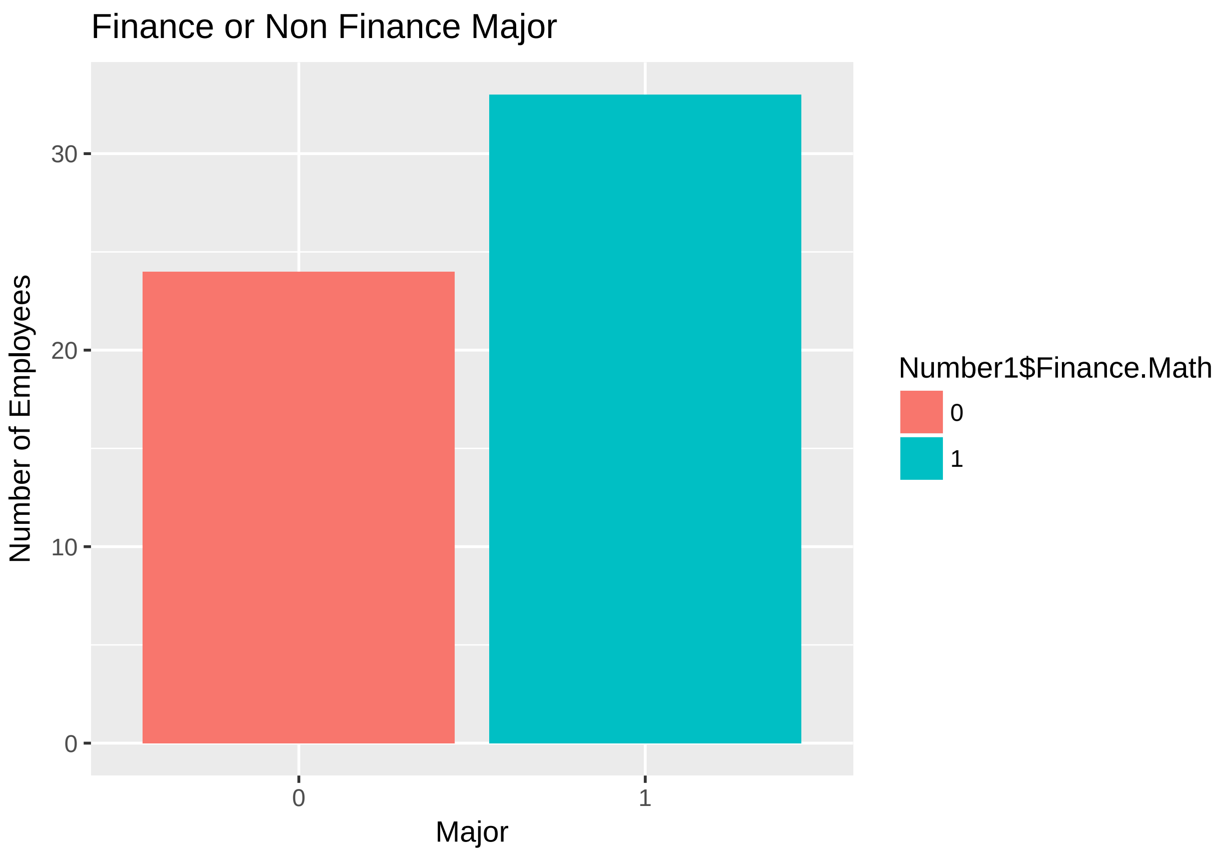
```r
ggplot(data = People6, aes(x = People6$Finance.Math, y = People6$Test.Score)) +
  geom_bar(stat = "identity")
```



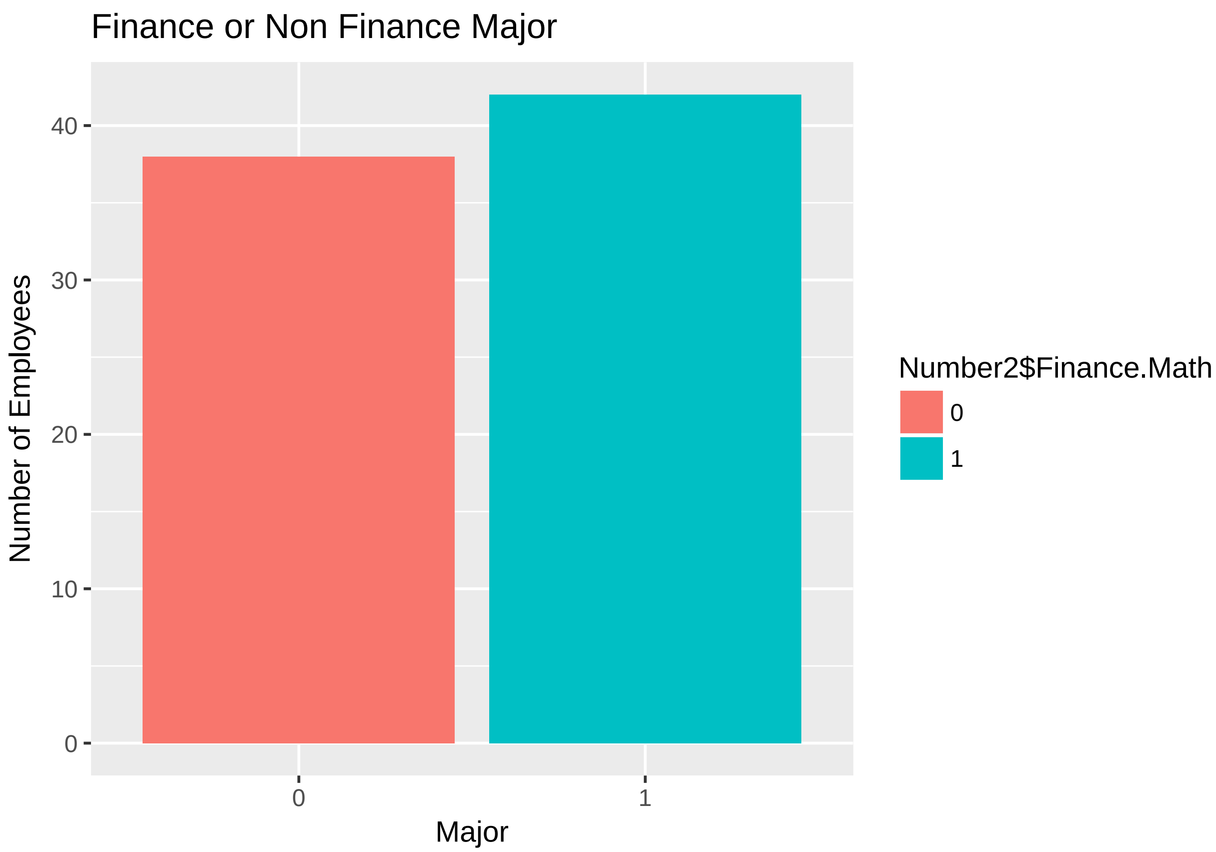
```r
ggplot(data = Number1, aes(x = Number1$Finance.Math, y = Number1$Test.Score)) +
  geom_bar(stat = "identity")
```



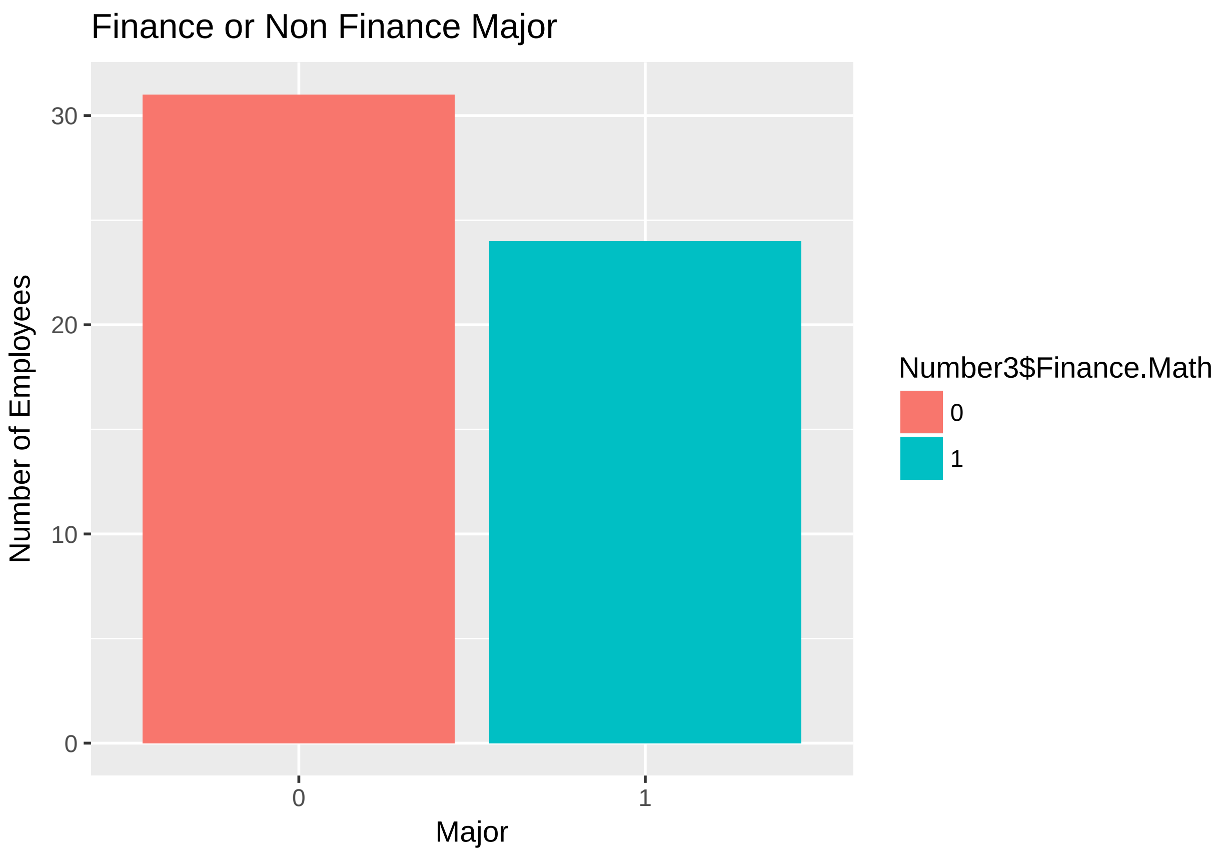
```
ggplot(data = Number1, aes(x = Number1$Finance.Math)) +
  geom_bar(aes(fill = Number1$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



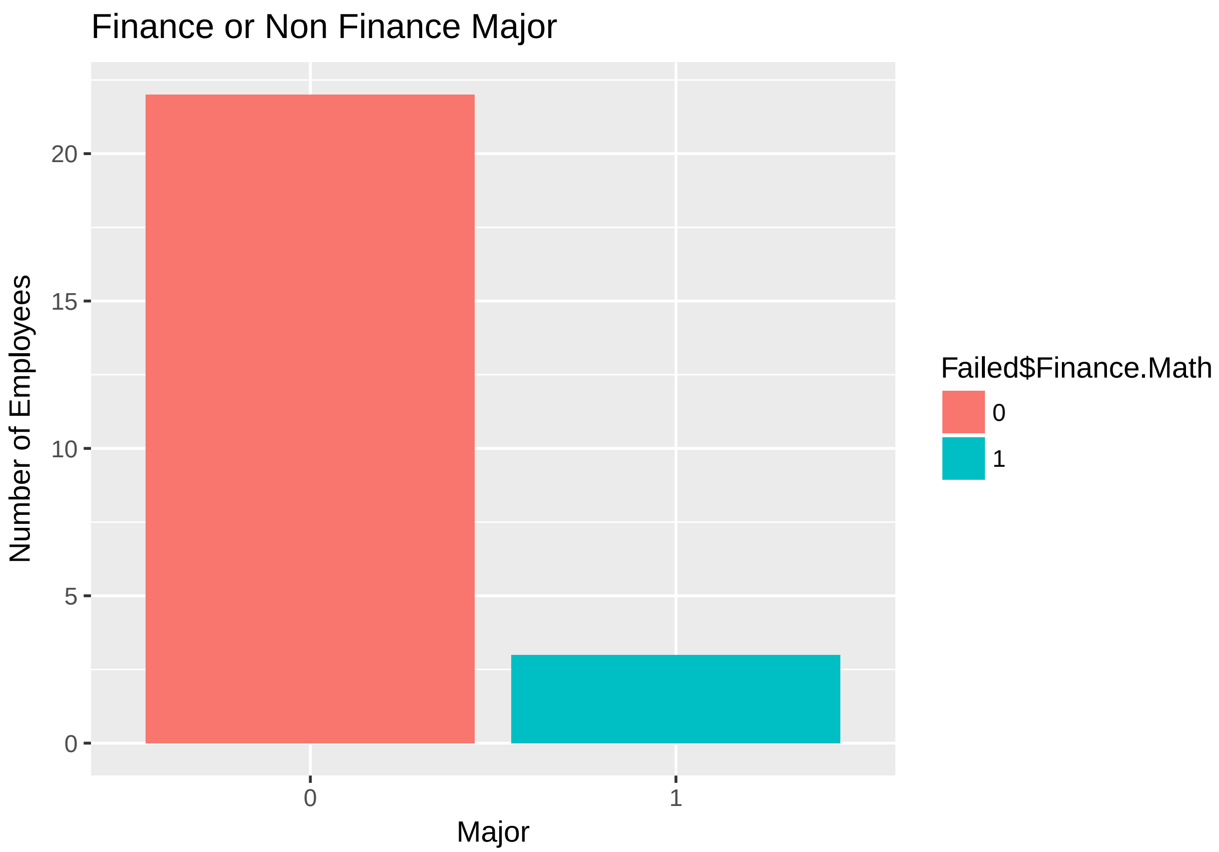
```r
ggplot(data = Number2, aes(x = Number2$Finance.Math)) +
  geom_bar(aes(fill = Number2$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



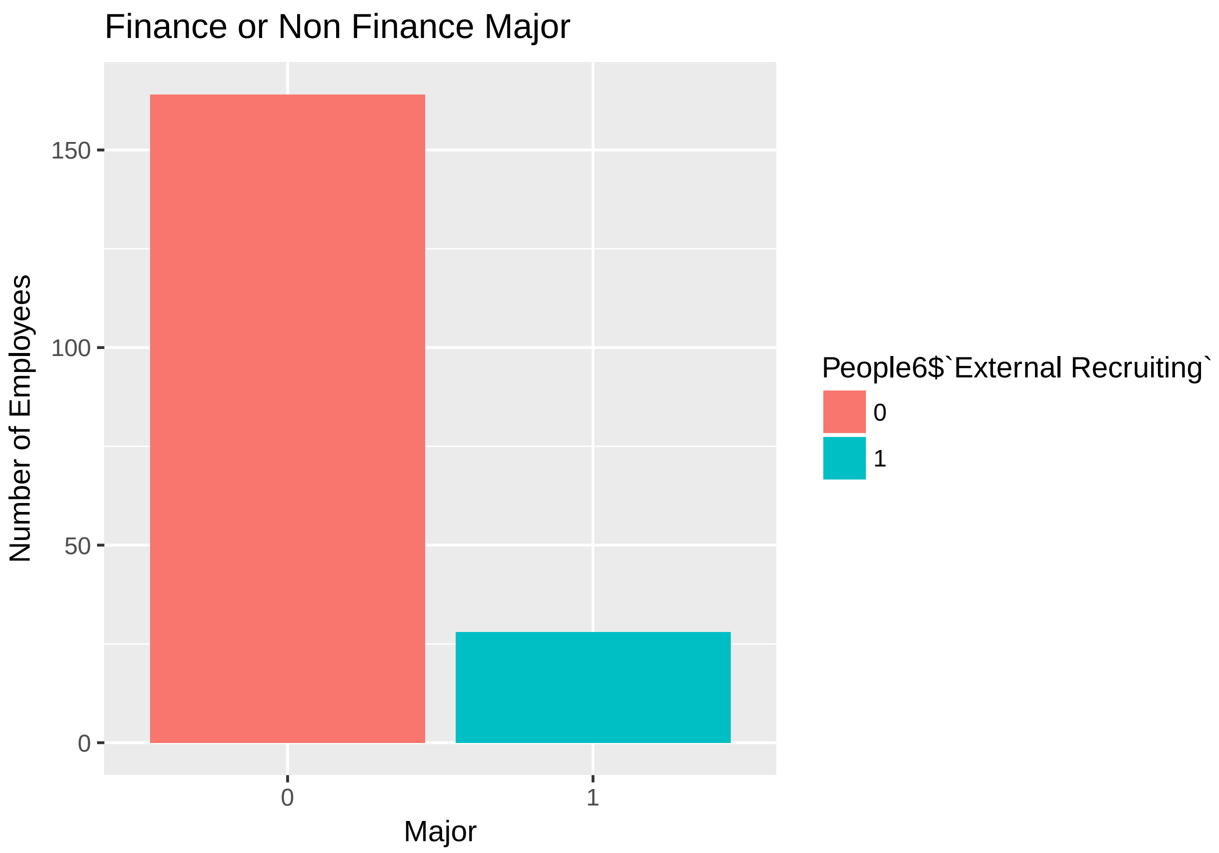
```
ggplot(data = Number3, aes(x = Number3$Finance.Math)) +
  geom_bar(aes(fill = Number3$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



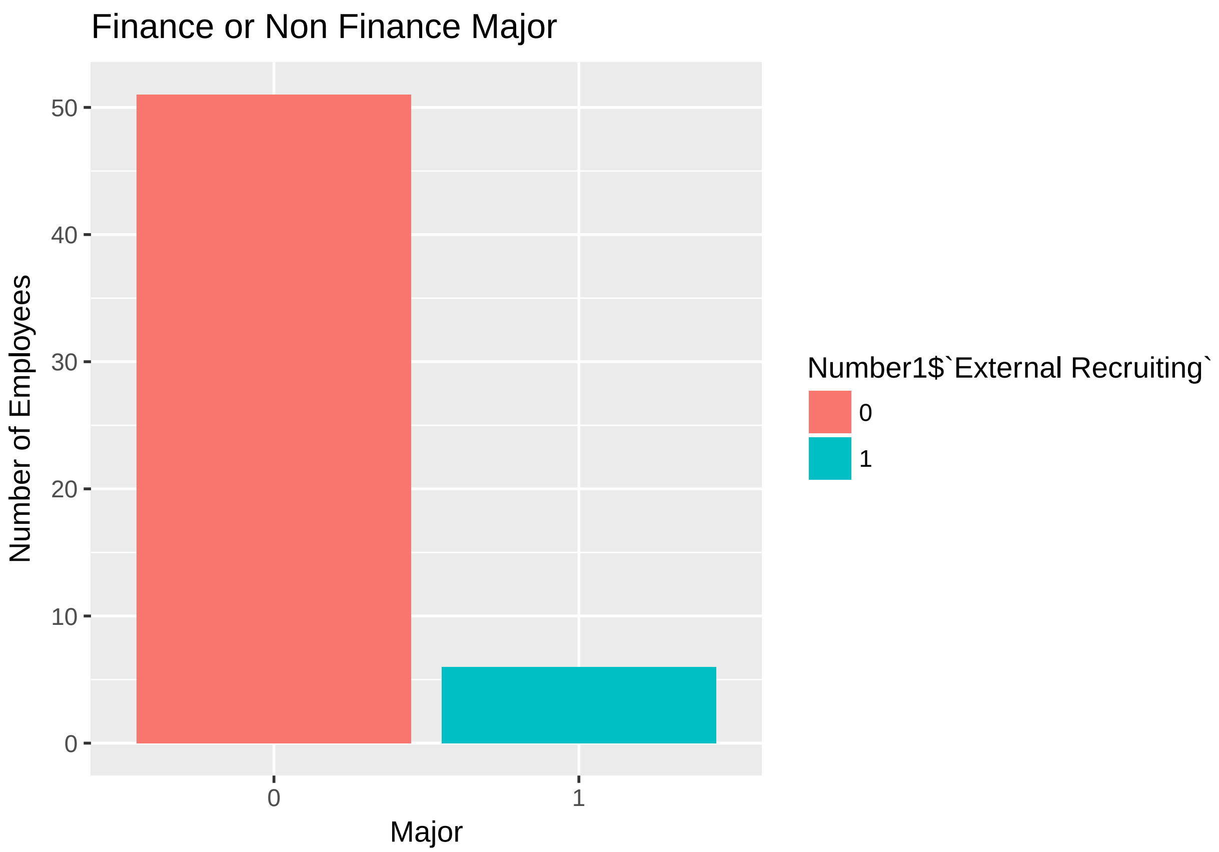
```
ggplot(data = Failed, aes(x = Failed$Finance.Math)) +
  geom_bar(aes(fill = Failed$Finance.Math), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



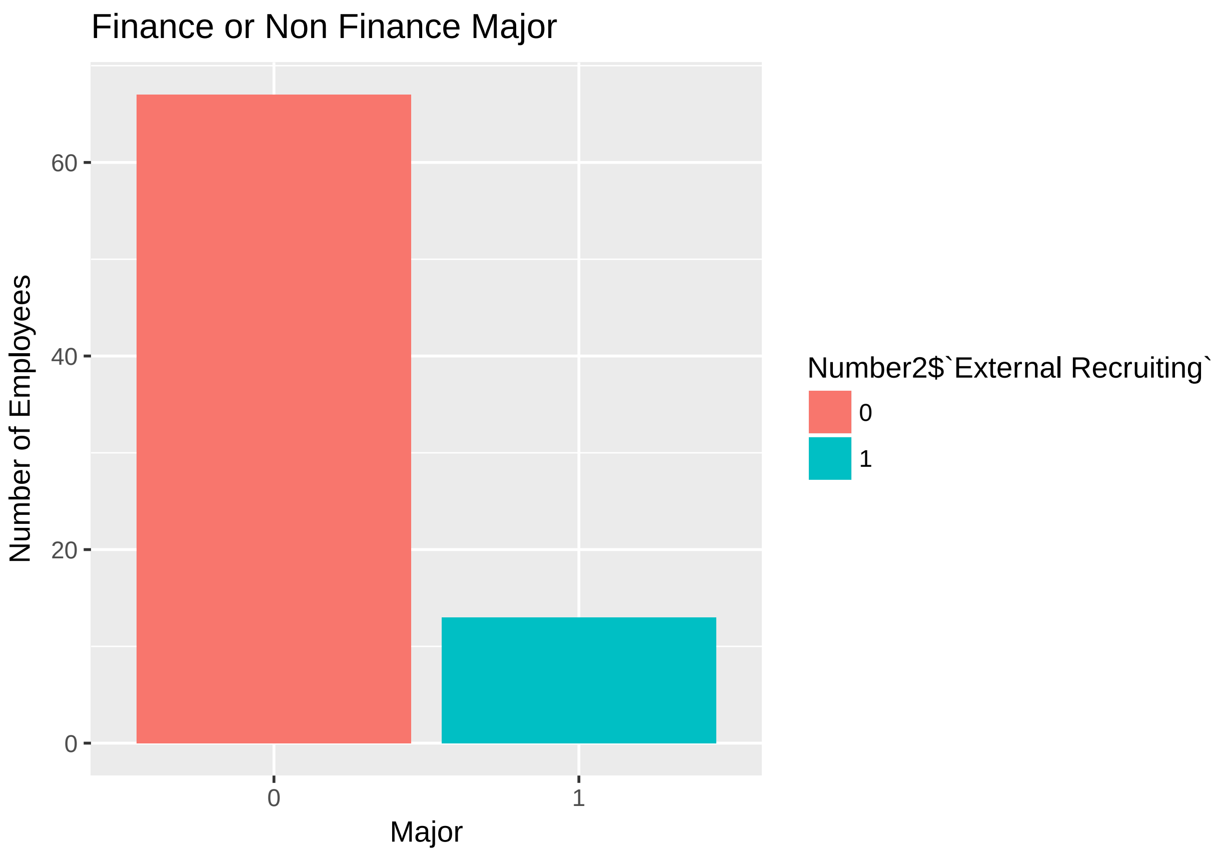
```
ggplot(data = People6, aes(x = People6$`External Recruiting`)) +
  geom_bar(aes(fill = People6$`External Recruiting`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



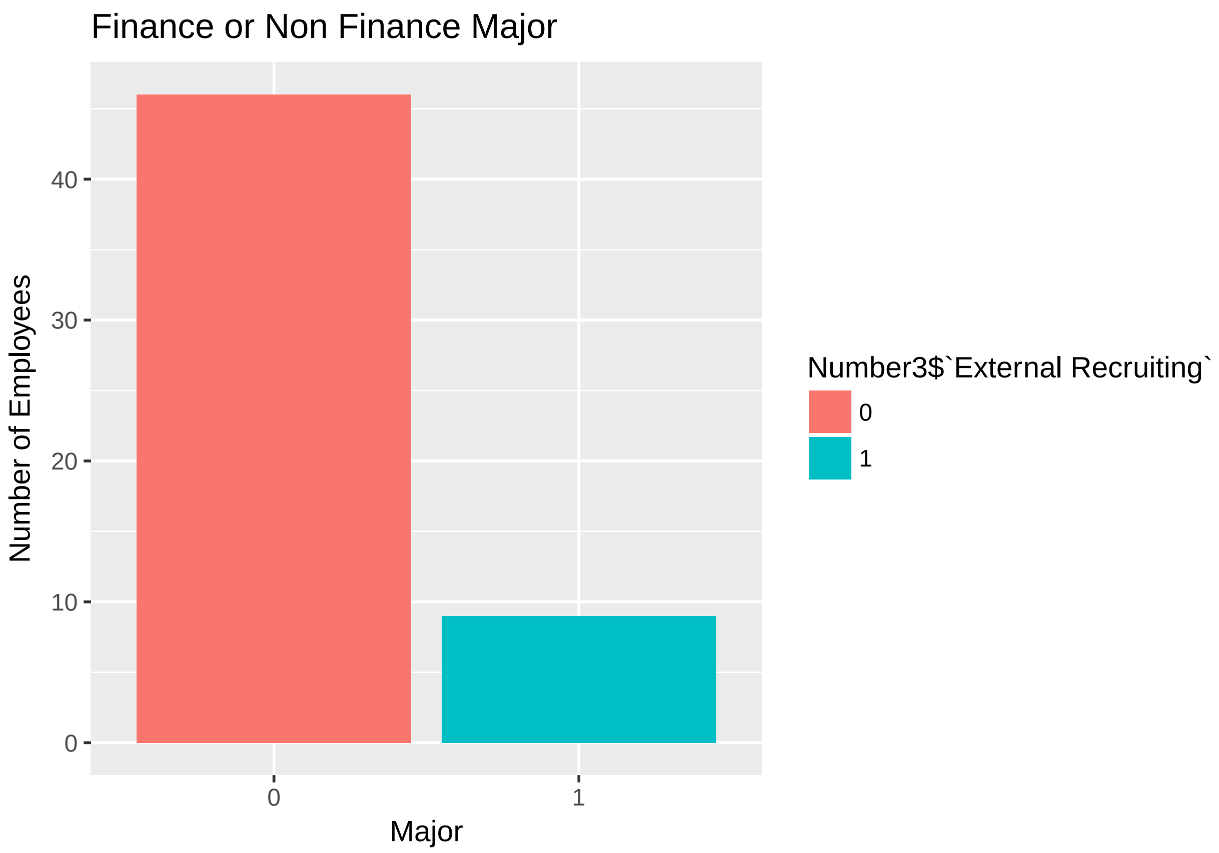
```r
ggplot(data = Number1, aes(x = Number1$`External Recruiting`)) +
  geom_bar(aes(fill = Number1$`External Recruiting`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



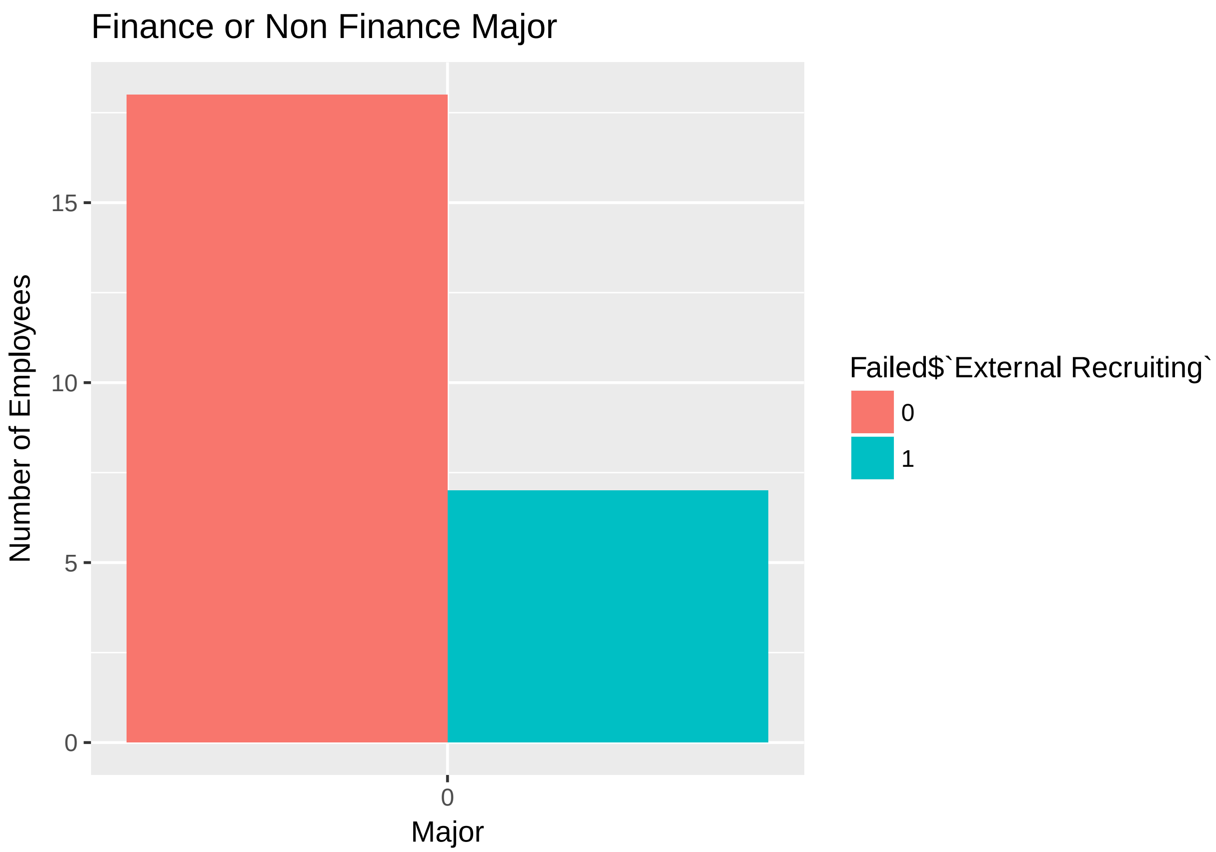
```
ggplot(data = Number2, aes(x = Number2$`External Recruiting`)) +
  geom_bar(aes(fill = Number2$`External Recruiting`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



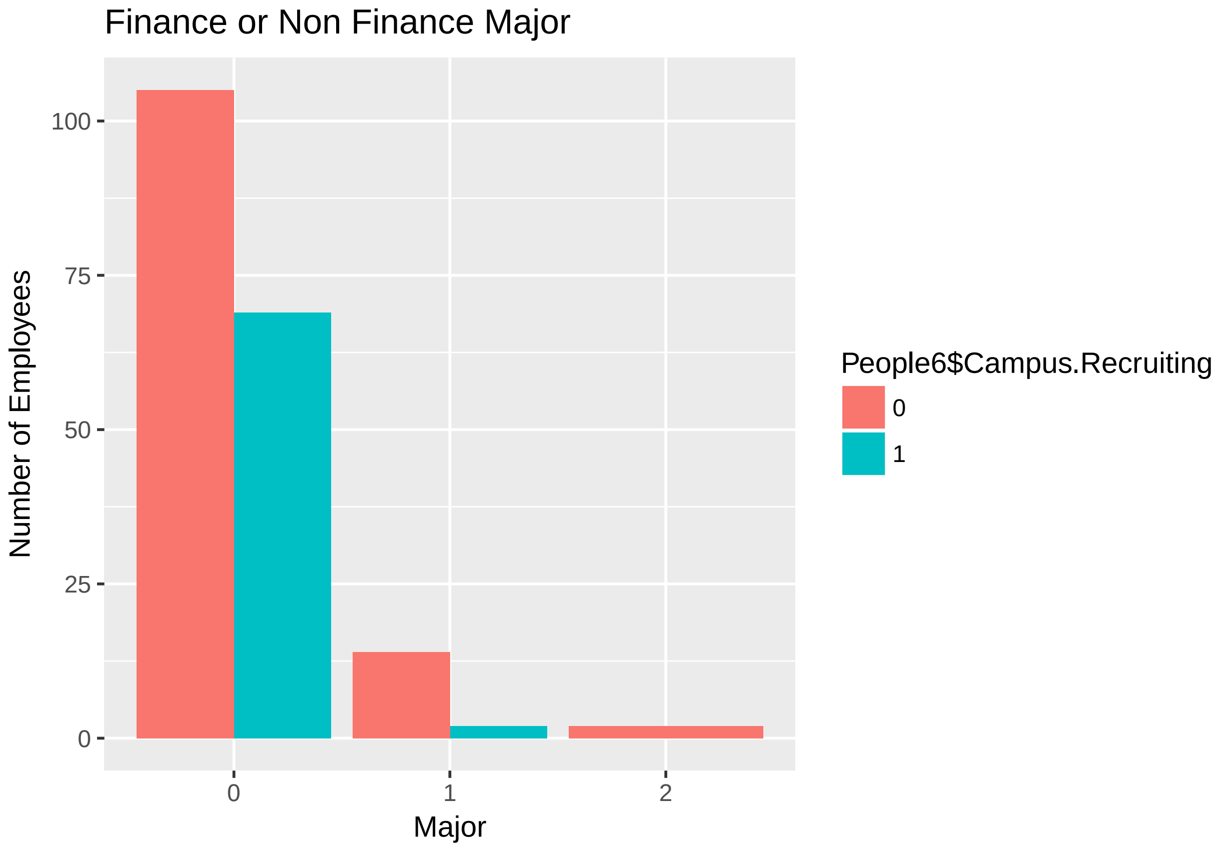
```r
ggplot(data = Number3, aes(x = Number3$`External Recruiting`)) +
  geom_bar(aes(fill = Number3$`External Recruiting`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



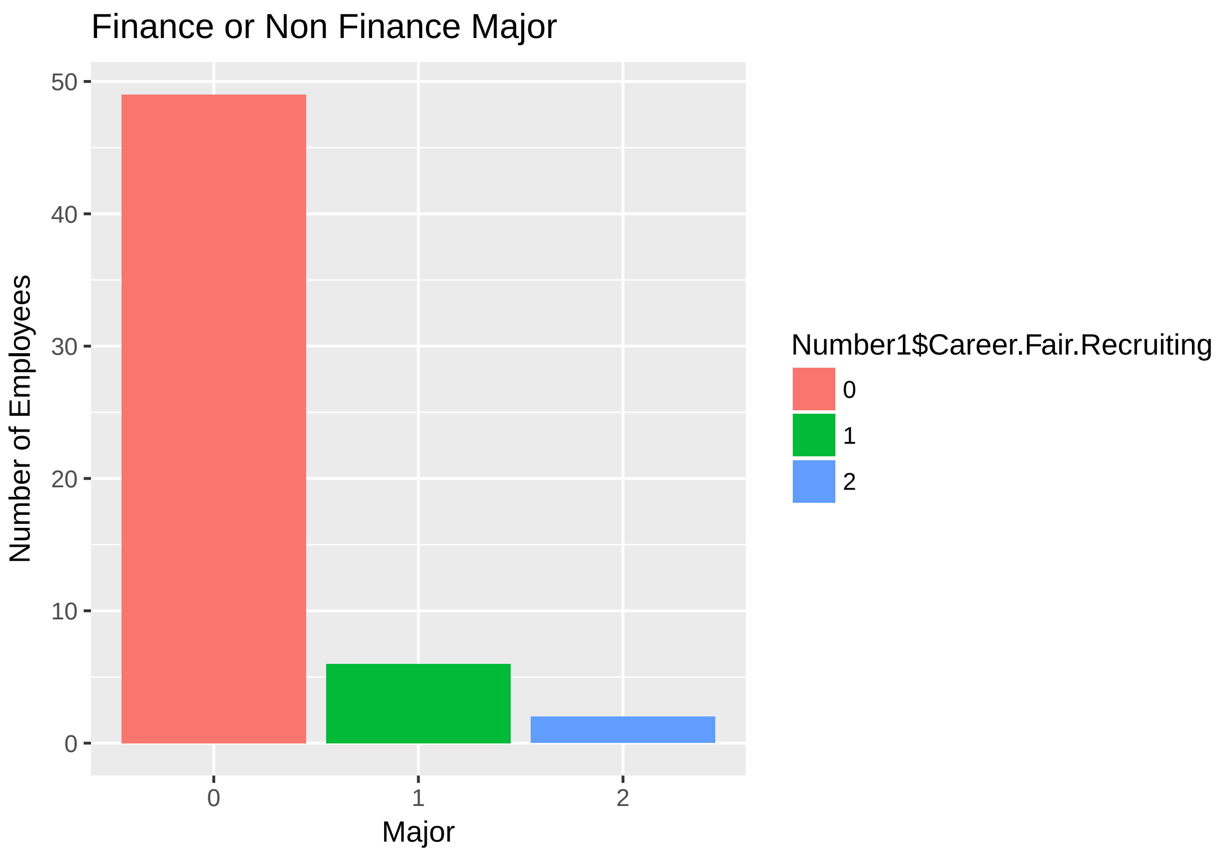
```r
ggplot(data = Failed, aes(x = Failed$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Failed$`External Recruiting`), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



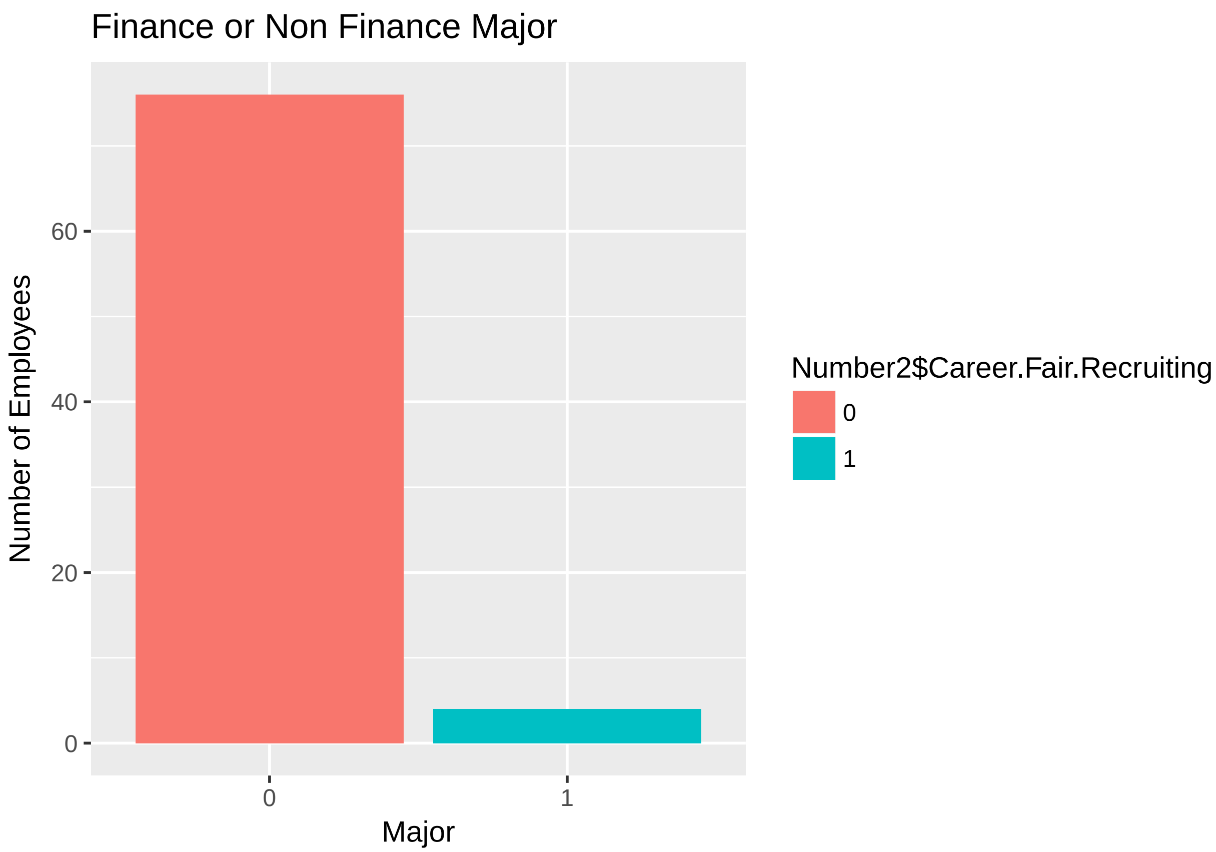
```
ggplot(data = People6, aes(x = People6$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = People6$Campus.Recruiting), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



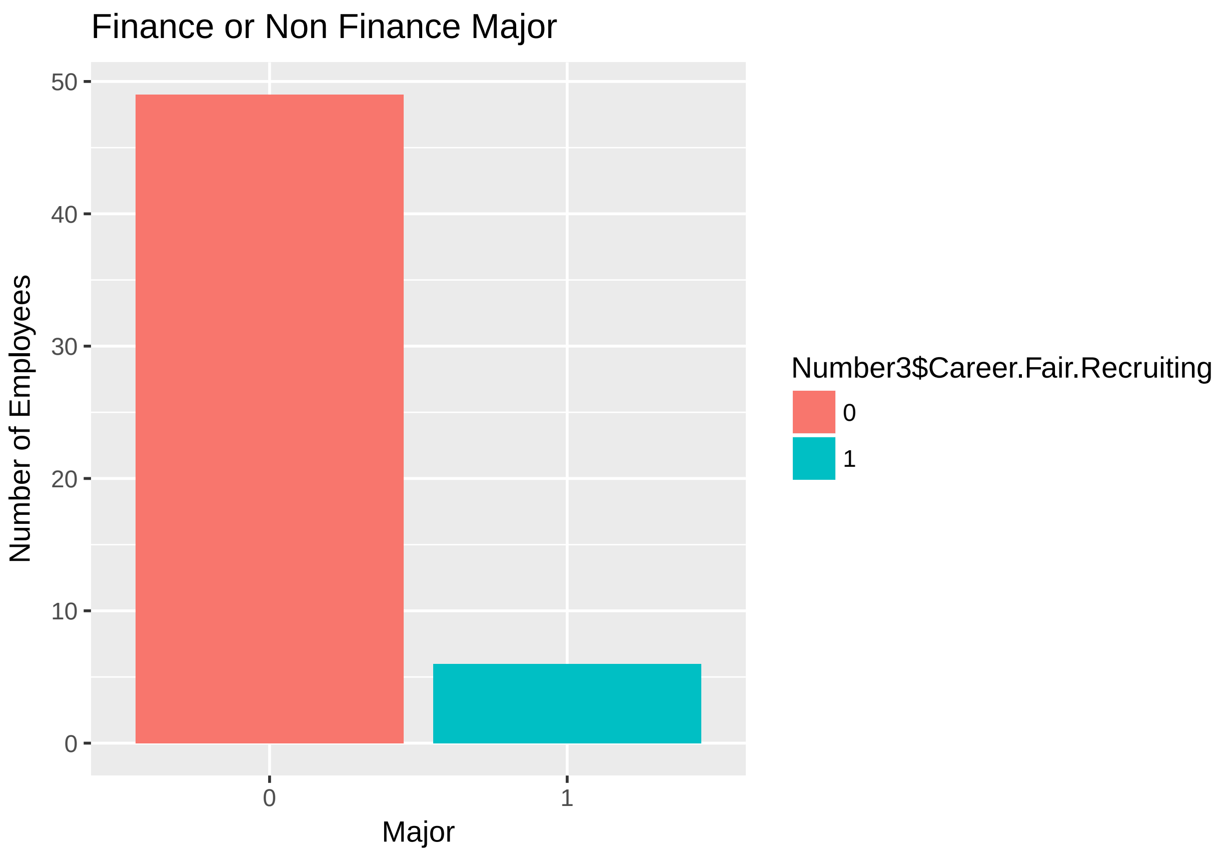
```r
ggplot(data = Number1, aes(x = Number1$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number1$Career.Fair.Recruiting), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



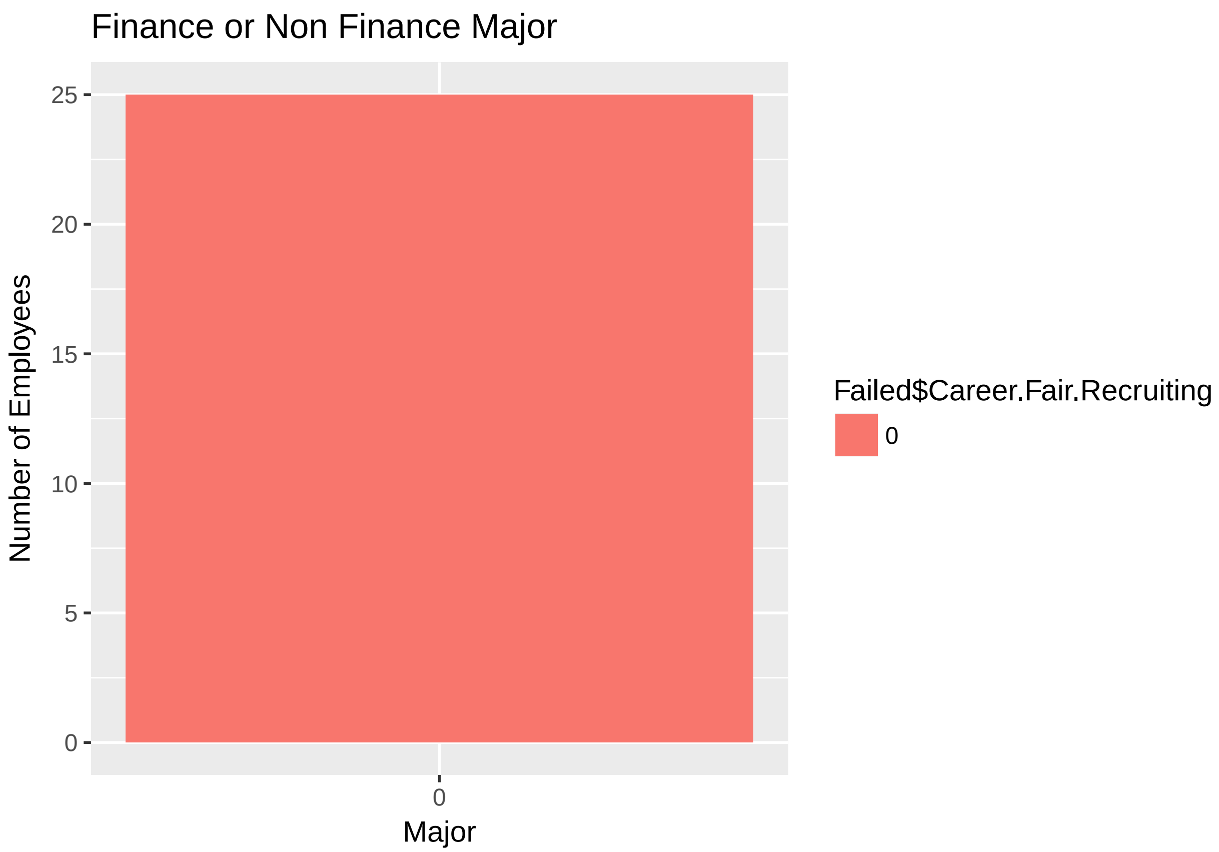
```r
ggplot(data = Number2, aes(x = Number2$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number2$Career.Fair.Recruiting), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



```
ggplot(data = Number3, aes(x = Number3$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Number3$Career.Fair.Recruiting), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



```
ggplot(data = Failed, aes(x = Failed$Career.Fair.Recruiting)) +
  geom_bar(aes(fill = Failed$Career.Fair.Recruiting), position = "dodge")+
  ggtitle("Finance or Non Finance Major")+
  labs(x = "Major", y = "Number of Employees")
```



```
summary(People6$Campus.Recruiting)
```

```
##   0   1
## 121  71
```

```
is.na(People6$Campus.Recruiting)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [122] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [144] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [155] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [166] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [177] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [188] FALSE FALSE FALSE FALSE FALSE
```

```
dplyr::filter(People6, People6$Career.Fair.Recruiting == 2)
```

```
##   Years.in.Fin Age Test.Score Number.Tries Learning Aptitude Motivation
## 1            0  22         79            1                 4          9
## 2            0  22         84            1                 5          2
##   Gender Train.RatingN Finance.Math Previously.Employed Internal.Referral
## 1      0             3            1                   0                 0
## 2      1             2            1                   0                 0
##   Campus.Recruiting External Recruiting Career.Fair.Recruiting Hiring Site
## 1                 0                   1                      2           2
## 2                 0                   0                      2           2
##   The.Recruiter College.GPA School.Type Citizen People7$Training.Practice
## 1             4         3.1           9       1                  74.21221
## 2             2         3.3           9       1                  86.96416
##   Total.Cost Total.Recruiting.Cost Total.Training.Cost
## 1       6500                  5000                1500
## 2       2500                  1000                1500
```

```
dplyr::filter(People4, People6$Career.Fair.Recruiting == 2)
```

```
##   Employee.ID                  School Accredited. Years.in.Fin.Svcs
## 1          26 St. Joseph's University         Yes                 0
## 2          74 St. Joseph's University         Yes                 0
##   College.GPA Finance...Math.major Previously.employed.at.Vanguard Gender
## 1         3.1                    Y                               N      M
## 2         3.3                    Y                               N      F
##   Age US.Citizen. Internal.Referral Hiring.Site Recruiter Career.Fair
## 1  22           Y                 N          PA        OK           n
## 2  22           Y                 N          PA        DT           n
##   Campus..Recruiting External.Recruiter FINRA.Series.7.test.score
## 1                  N                  Y                        79
## 2                  N                  N                        84
##   Num..of.tries.before.passing.Series.7 Training.Rating Learning.Aptitude
## 1                                     1            High                 7
## 2                                     1          Medium                 8
##   motivation Training.practice.test.score GenderN AccreditedN FinanceMathN
## 1          9                     74.21221       0           1            1
## 2          3                     86.96416       1           1            1
##   Previously.employedN US.CitizenN Internal.ReferralN Career.FairN
## 1                    0           1                  0            2
## 2                    0           1                  0            2
##   External.RecruiterN Training.RatingN SchoolN Hiring.SiteN RecruiterN
## 1                   1                2       9            2          4
## 2                   0                1       9            2          2
##   Campus.RecruitingN External.Cost Career.FairCost Campus.Recruiting.Cost
## 1                  0          5000               0                      0
## 2                  0             0               0                      0
##   TotalRecruitingCost TotalTrainingCost TotalCost
## 1                5000              1500      6500
## 2                1000              1500      2500
```

```
correlate(People6$`Learning Aptitude`, People6$College.GPA)
```

```
##
## CORRELATIONS
## ============
## - correlation type:  pearson
## - correlations shown only when both variables are numeric
##
##       y.var
## x.var 0.133
```

]