Bonus2

Nitin Gaonkar

Introduction: In this project, I have used the German credit data from the caret package, the go al is to predict the variable amount as the function of other variables using the OLS with lm () fu nction in R, the data is divided into training and holdout set 63.2:36.8 ratio. From the EDA, we c an see that there is no missing data in the data set. We will be using few of the EDA models to d etermine the predictor variables, we will be trying PCA, lasso and step wise models to determin e predictor variables.

The goal of this analysis is to investigate a different statistical resampling method, known as multiple train and test, or MUTATE (Chaturvedi 2014).

Below part of code loads the required libraries and loads the source data required for the analy sis, we are using the German credit data from caret.

```
install. packages("caret")
install. packages("lattice")
install. packages("ggplot2")
```

```
library(dplyr)
library(dummies)
library(caret)
```

```
data("GermanCredit")
attach(GermanCredit)
GC <- data.frame(GermanCredit)
```

```
nrow(GC)
```

```
## [1] 1000
```

```
dim(GC)
```

```
## [1] 1000   62
```

```
new_data<- dummy.data.frame(GC,names=c("Class"))
```

Below is the function to find the missing data

```
missing_func<- function(df){
  col <- NCOL(GC)
  rows<- nrow(GC)
  var<- c()
  missing_var<- nrow(GC)

  for (i in 1:col) {
    var[i] <- colnames(df[i])
    missing_var[i] <- sum(is.na(df[i]))

  }
  dframe <- data.frame(var, missing_var)
  dframe
}
```

There is no missing data in our dataset

```
missing_func(GC)
```

```
##                                     var missing_var
## 1                              Duration           0
## 2                                Amount           0
## 3            InstallmentRatePercentage           0
## 4                    ResidenceDuration           0
## 5                                   Age           0
## 6                NumberExistingCredits           0
## 7              NumberPeopleMaintenance           0
## 8                            Telephone           0
## 9                         ForeignWorker           0
## 10                                Class           0
## 11            CheckingAccountStatus.lt.0           0
## 12        CheckingAccountStatus.0.to.200           0
## 13          CheckingAccountStatus.gt.200           0
## 14             CheckingAccountStatus.none           0
## 15        CreditHistory.NoCredit.AllPaid           0
## 16        CreditHistory.ThisBank.AllPaid           0
## 17                CreditHistory.PaidDuly           0
## 18                   CreditHistory.Delay           0
## 19                CreditHistory.Critical           0
## 20                       Purpose.NewCar           0
## 21                      Purpose.UsedCar           0
## 22           Purpose.Furniture.Equipment           0
## 23              Purpose.Radio.Television           0
## 24            Purpose.DomesticAppliance           0
## 25                      Purpose.Repairs           0
## 26                    Purpose.Education           0
## 27                     Purpose.Vacation           0
## 28                   Purpose.Retraining           0
## 29                     Purpose.Business           0
## 30                        Purpose.Other           0
```

```
## 31             SavingsAccountBonds.lt.100           0
## 32          SavingsAccountBonds.100.to.500          0
## 33         SavingsAccountBonds.500.to.1000          0
## 34             SavingsAccountBonds.gt.1000          0
## 35             SavingsAccountBonds.Unknown          0
## 36                 EmploymentDuration.lt.1          0
## 37               EmploymentDuration.1.to.4          0
## 38               EmploymentDuration.4.to.7          0
## 39                 EmploymentDuration.gt.7          0
## 40           EmploymentDuration.Unemployed          0
## 41        Personal.Male.Divorced.Seperated          0
## 42               Personal.Female.NotSingle          0
## 43                    Personal.Male.Single          0
## 44           Personal.Male.Married.Widowed          0
## 45                  Personal.Female.Single          0
## 46             OtherDebtorsGuarantors.None          0
## 47      OtherDebtorsGuarantors.CoApplicant          0
## 48        OtherDebtorsGuarantors.Guarantor          0
## 49                     Property.RealEstate          0
## 50                      Property.Insurance          0
## 51                       Property.CarOther          0
## 52                        Property.Unknown          0
## 53              OtherInstallmentPlans.Bank          0
## 54            OtherInstallmentPlans.Stores          0
## 55              OtherInstallmentPlans.None          0
## 56                             Housing.Rent          0
## 57                              Housing.Own          0
## 58                          Housing.ForFree          0
## 59                 Job.UnemployedUnskilled          0
## 60                   Job.UnskilledResident          0
## 61                     Job.SkilledEmployee          0
## 62 Job.Management.SelfEmp.HighlyQualified          0
```

Below function is used to remove the columns containing only zeros in them

```
remove_zero_cols <- function(df) {
  rem_vec <- NULL
  for(i in 1:ncol(df)){
    this_sum <- summary(df[,i])
    zero_test <- length(which(this_sum == 0))
    if(zero_test == 6) {
      rem_vec[i] <- names(df)[i]
    }
  }
  features_to_remove <- rem_vec[!is.na(rem_vec)]
  rem_ind <- which(names(df) %in% features_to_remove)
```

```
  df <- df[,-rem_ind]
  return(df)
}

## running the above function to remove columns with only zeros###
new_data <- remove_zero_cols(GC)
```

Convert the columns into numeric for PCA

```
new_data<- dummy.data.frame(new_data,names=c("Class"))

####dividing data into test(holdout) and train based on the given requirement
####

set.seed(123)
indexes = sample(1:nrow(new_data), size=0.368*nrow(GC))
test = new_data[indexes,]
train=new_data[-indexes,]
dim(test)

## [1] 368  61

dim(train)

## [1] 632  61

indexes

##   [1] 288 788 409 881 937  46 525 887 548 453 948 449 670 566 102 993 243
##  [18]  42 323 996 872 679 627 972 640 691 530 579 282 143 935 875 669 770
##  [35]  24 462 732 209 307 223 138 398 397 353 146 133 961 445 254 816  44
##  [52] 420 758 116 531 196 121 711 844 957 626  90 361 258 763 949 757 990
##  [69] 741 410 702 585 660   1 441 204 351 987 325 103 225 614 384 723  95
##  [86] 959 901 951 809 160 119 594 312 596 291 170 708  85 422 461 540 300
## [103] 439 857 433 797 818 544 367 132 833 268  54 841 639 126 486 843 517
## [120] 357 571 977 271 193 324 862 135  80 124 601 539 775 929 976 452 880
## [137] 943 979 845 379 269 352   9 158 722 198 205  66 210 624 721 423 329
## [154] 963  94 330 483 183 375 184 902 297 545 314 298 446 618 185 344 221
## [171] 523 153 716 834 553 510 962 437 719 478 689 256 580 217 485 393 216
## [188] 460 742 964 954 261 895 501 756 376 328 974 123 459 191 769 480 411
## [205] 321 700 290 229 136 137 381 200 171 946  38 551 276 796 642 718 831
## [222] 749 567 534 983 824 370 434 900 706 477 969 417  45 789 304 152 635
## [239] 117 612 416 503 131 798 236 992 301 730 728 546 877 167 444 766 958
## [256] 586 885 754 350 644 686 652 498 701 790 424 247 255  15 368 636   5
## [273]  53 120 560 777 704 338 888 468 547  99 285 162 776 284  47 717  39
## [290] 967 212  72  51 623 533 576 692  73  70 726 550   7 893 509 898 335
## [307] 109   6 837 341 860 783 491 786 244 795 572 163 242 584 581 201 100
## [324] 864  71  23 674 966 228 615 414 192 493 557 852 804 464 426 427 647
```

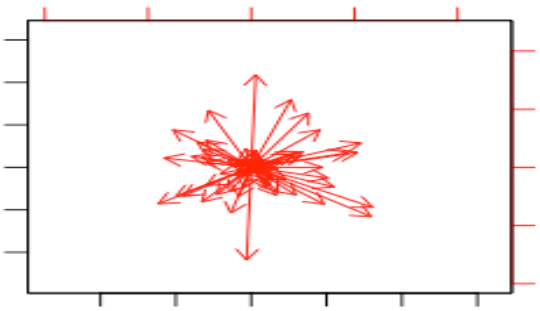
```
## [341] 274  79 347 148 320 984 932 952 150 406  89 628 334 106 402 753 431
## [358] 270 208 536  93 980 573 197 232 738 802  12
```

Running PCA on the data

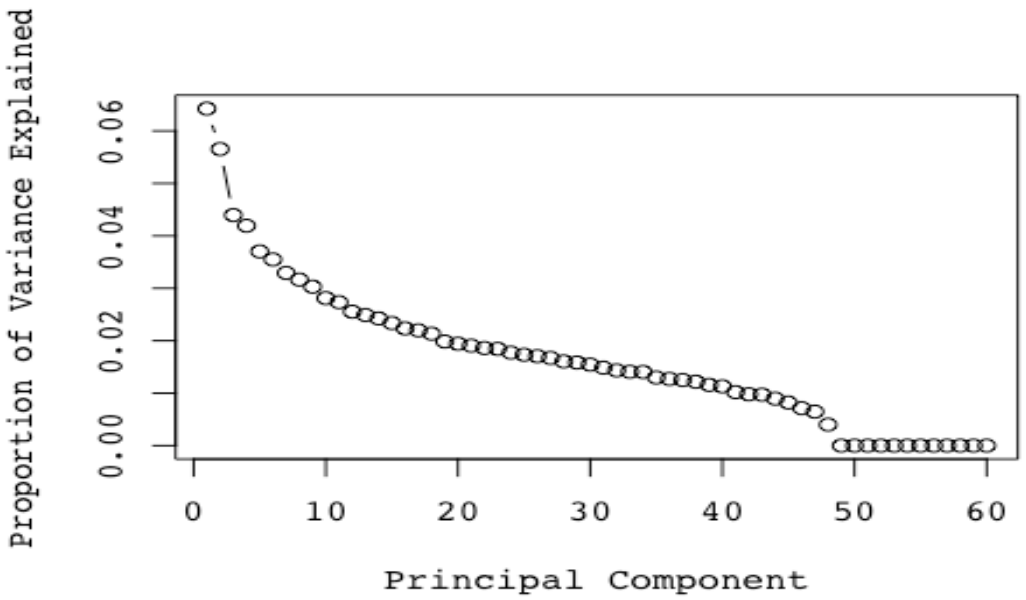
```
pca_train<- subset(train,select =-c(Amount))
prin_comp<- prcomp(pca_train,scale.=T)
names(prin_comp)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

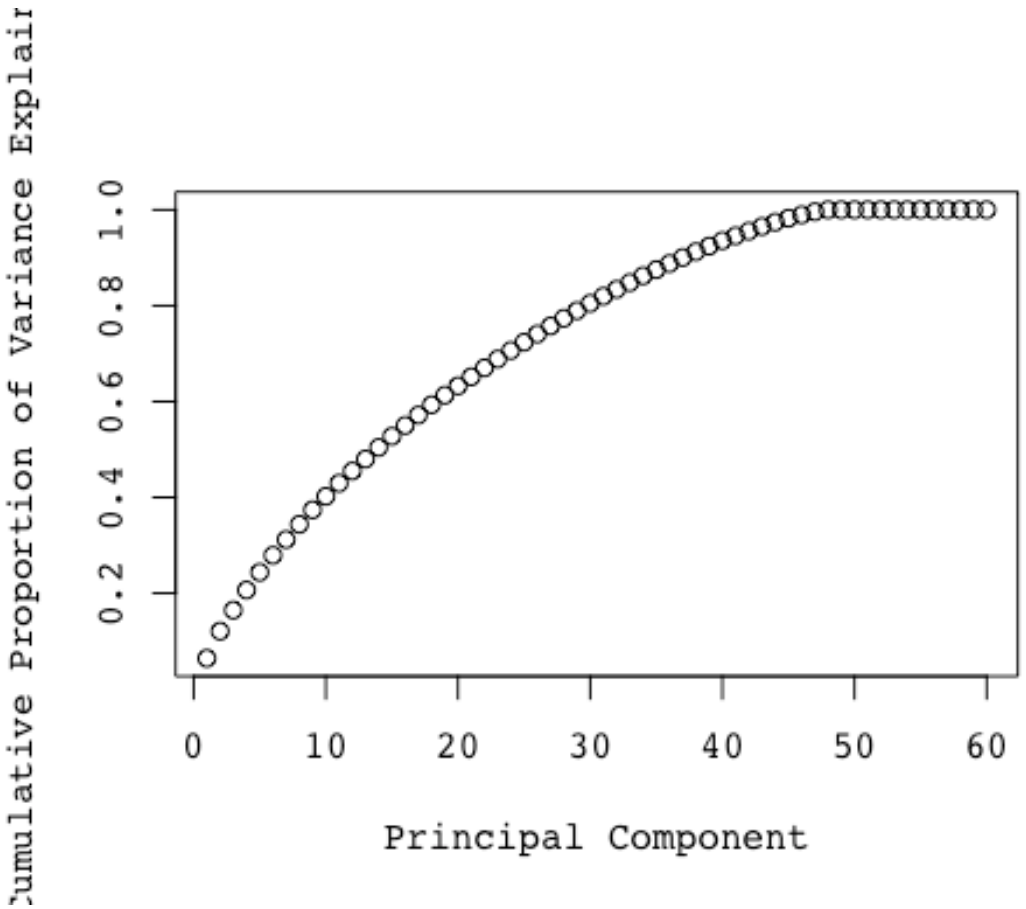
```
biplot(prin_comp,scale=0)
```



```
std_dev<- prin_comp$sdev
pr_var<- std_dev^2
prop_varex<-pr_var/sum(pr_var)
par(family="mono")
par(family="mono")
plot(prop_varex, xlab = "Principal Component",ylab = "Proportion of Variance
Explained",type = "b")
```




```r
plot(cumsum(prop_varex), xlab = "Principal Component",ylab = "Cumulative Prop
ortion of Variance Explained",type = "b")
```



```
train.data <- data.frame(Amount = train$Amount, prin_comp$x)
data<-prin_comp$x
```

50 PCA components almost explains more than 90 % of the variance

```
train.data <- train.data[,1:51]
model1<- lm(Amount~.,data=train.data)
summary(model1)  ##r2=0.5932

##
## Call:
## lm(formula = Amount ~ ., data = train.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6099.5 -1111.4  -140.8   731.3 10225.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.050e+03  1.881e+02  16.210  < 2e-16 ***
## PC1          8.491e+02  1.369e+02   6.201 1.07e-09 ***
## PC2          5.460e+02  4.434e+02   1.232   0.2186
## PC3          2.588e+02  3.693e+02   0.701   0.4836
## PC4          1.634e+03  9.455e+02   1.729   0.0844 .
## PC5         -1.378e+03  7.420e+02  -1.857   0.0638 .
## PC6          1.140e+03  4.795e+02   2.378   0.0177 *
## PC7         -3.161e+02  4.462e+02  -0.708   0.4790
## PC8          6.120e+02  3.447e+02   1.775   0.0764 .
## PC9          2.675e+02  3.753e+02   0.713   0.4763
## PC10        -4.409e+02  1.904e+02  -2.315   0.0209 *
## PC11         1.084e+03  4.837e+02   2.242   0.0254 *
## PC12        -8.886e+02  4.826e+02  -1.841   0.0661 .
## PC13         4.997e+02  2.973e+02   1.681   0.0933 .
## PC14         4.291e+02  9.583e+02   0.448   0.6545
## PC15         3.247e+02  1.114e+02   2.915   0.0037 **
## PC16        -1.045e+02  2.043e+02  -0.512   0.6091
## PC17         2.113e+02  7.438e+02   0.284   0.7764
## PC18        -4.121e+02  5.700e+02  -0.723   0.4700
## PC19         2.679e+02  2.338e+02   1.146   0.2523
## PC20        -3.462e+02  8.965e+02  -0.386   0.6995
## PC21        -1.643e+03  1.111e+03  -1.478   0.1400
## PC22        -1.393e+03  8.630e+02  -1.615   0.1069
## PC23        -5.239e+02  5.674e+02  -0.923   0.3562
## PC24         5.488e+02  4.229e+02   1.298   0.1949
```

```
## PC25         4.896e+02  2.469e+02   1.983   0.0478 *
## PC26         5.787e+02  1.208e+03   0.479   0.6321
## PC27        -3.383e+01  3.782e+02  -0.089   0.9287
## PC28         1.199e+03  6.646e+02   1.804   0.0718 .
## PC29        -2.914e+02  2.252e+02  -1.294   0.1962
## PC30         1.463e+02  4.402e+02   0.332   0.7397
## PC31        -6.759e+01  1.074e+02  -0.629   0.5293
## PC32        -5.139e+02  4.285e+02  -1.199   0.2309
## PC33        -3.199e+02  3.051e+02  -1.049   0.2948
## PC34        -2.890e+03  1.473e+03  -1.962   0.0503 .
## PC35         1.109e+03  9.404e+02   1.179   0.2388
## PC36        -1.925e+03  1.039e+03  -1.852   0.0645 .
## PC37         5.484e+02  3.825e+02   1.434   0.1521
## PC38         9.855e+02  6.292e+02   1.566   0.1178
## PC39         1.231e+03  1.062e+03   1.159   0.2471
## PC40         8.550e+02  1.029e+03   0.831   0.4066
## PC41         1.175e+03  1.221e+03   0.962   0.3365
## PC42        -9.731e+02  6.356e+02  -1.531   0.1263
## PC43        -2.881e+03  1.790e+03  -1.610   0.1080
## PC44         7.390e+02  7.104e+02   1.040   0.2986
## PC45        -6.682e+02  4.679e+02  -1.428   0.1538
## PC46        -3.765e+01  3.104e+02  -0.121   0.9035
## PC47        -8.322e+02  4.691e+02  -1.774   0.0766 .
## PC48         4.461e+02  4.062e+02   1.098   0.2727
## PC49        -4.863e+18  2.308e+18  -2.107   0.0355 *
## PC50        -8.665e+17  2.222e+18  -0.390   0.6967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1903 on 581 degrees of freedom
## Multiple R-squared:  0.6254, Adjusted R-squared:  0.5932
## F-statistic:  19.4 on 50 and 581 DF,  p-value: < 2.2e-16

######## running model on the holdout data##########
pca_test<- subset(test,select =-c(Amount))
test.data <- predict(prin_comp, newdata = pca_test)
test.data <- as.data.frame(test.data)
test.data <- test.data[,1:51]
linear.prediction <- predict(model1, test.data)
linear.prediction

cor(test$Amount,linear.prediction)^2  ## 0.5517428

## [1] 0.5517428

sqrt(mean((test$Amount - linear.prediction)^2)) ## 1759.787

## [1] 1759.787
```

stepwise backward and forward selection

```
model_full<- lm (train$Amount~.,data=train)
summary(model_full)
```

```
##
## Call:
## lm(formula = train$Amount ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5839.2 -1109.1  -162.5   724.2 10372.1
##
## Coefficients: (12 not defined because of singularities)
##                                        Estimate Std. Error t value
## (Intercept)                            7950.266   1220.627   6.513
## Duration                                122.746      7.522  16.319
## InstallmentRatePercentage              -859.749     73.014 -11.775
## ResidenceDuration                       -17.282     80.588  -0.214
## Age                                       1.687      8.051   0.210
## NumberExistingCredits                   -43.894    176.003  -0.249
## NumberPeopleMaintenance                 -287.405    247.240  -1.162
## Telephone                               -626.987    180.863  -3.467
## ForeignWorker                           -352.787    467.009  -0.755
## ClassBad                                 370.511    197.770   1.873
## ClassGood                                     NA         NA      NA
## CheckingAccountStatus.lt.0              -100.257    212.707  -0.471
## CheckingAccountStatus.0.to.200           392.800    206.599   1.901
## CheckingAccountStatus.gt.200            -568.244    365.831  -1.553
## CheckingAccountStatus.none                    NA         NA      NA
## CreditHistory.NoCredit.AllPaid           996.360    467.778   2.130
## CreditHistory.ThisBank.AllPaid          -346.379    425.490  -0.814
## CreditHistory.PaidDuly                  -488.823    231.423  -2.112
## CreditHistory.Delay                      -83.670    304.370  -0.275
## CreditHistory.Critical                        NA         NA      NA
## Purpose.NewCar                         -1624.710    743.961  -2.184
## Purpose.UsedCar                        -1002.732    761.621  -1.317
## Purpose.Furniture.Equipment            -1813.581    749.853  -2.419
## Purpose.Radio.Television               -1888.797    748.036  -2.525
## Purpose.DomesticAppliance              -2680.818   1083.602  -2.474
## Purpose.Repairs                        -1402.406    930.932  -1.506
## Purpose.Education                      -1959.197    814.810  -2.404
## Purpose.Retraining                      -455.249   1592.464  -0.286
## Purpose.Business                       -2023.377    763.230  -2.651
## Purpose.Other                                 NA         NA      NA
## SavingsAccountBonds.lt.100              -678.649    219.731  -3.089
## SavingsAccountBonds.100.to.500          -943.936    307.951  -3.065
## SavingsAccountBonds.500.to.1000         -690.498    373.462  -1.849
```

```
## SavingsAccountBonds.gt.1000              -700.842    407.474  -1.720
## SavingsAccountBonds.Unknown                    NA         NA      NA
## EmploymentDuration.lt.1                   104.487    397.002   0.263
## EmploymentDuration.1.to.4                 -80.105    380.102  -0.211
## EmploymentDuration.4.to.7                 444.672    410.492   1.083
## EmploymentDuration.gt.7                  -420.833    381.351  -1.104
## EmploymentDuration.Unemployed                  NA         NA      NA
## Personal.Male.Divorced.Seperated          945.188    466.874   2.025
## Personal.Female.NotSingle                 267.124    293.021   0.912
## Personal.Male.Single                      822.792    291.510   2.823
## Personal.Male.Married.Widowed                  NA         NA      NA
## OtherDebtorsGuarantors.None                52.754    356.249   0.148
## OtherDebtorsGuarantors.CoApplicant        624.914    535.598   1.167
## OtherDebtorsGuarantors.Guarantor               NA         NA      NA
## Property.RealEstate                      -998.211    380.102  -2.626
## Property.Insurance                       -840.686    378.858  -2.219
## Property.CarOther                        -871.686    364.339  -2.393
## Property.Unknown                               NA         NA      NA
## OtherInstallmentPlans.Bank               -265.000    240.669  -1.101
## OtherInstallmentPlans.Stores             -365.618    375.503  -0.974
## OtherInstallmentPlans.None                     NA         NA      NA
## Housing.Rent                              212.614    434.634   0.489
## Housing.Own                               233.244    409.442   0.570
## Housing.ForFree                                NA         NA      NA
## Job.UnemployedUnskilled                 -1882.120    567.779  -3.315
## Job.UnskilledResident                   -1282.490    321.313  -3.991
## Job.SkilledEmployee                     -1262.014    251.502  -5.018
## Job.Management.SelfEmp.HighlyQualified         NA         NA      NA
##                                         Pr(>|t|)
## (Intercept)                             1.59e-10 ***
## Duration                                 < 2e-16 ***
## InstallmentRatePercentage                < 2e-16 ***
## ResidenceDuration                       0.830267
## Age                                     0.834122
## NumberExistingCredits                   0.803146
## NumberPeopleMaintenance                 0.245526
## Telephone                               0.000566 ***
## ForeignWorker                           0.450303
## ClassBad                                0.061507 .
## ClassGood                                     NA
## CheckingAccountStatus.lt.0              0.637574
## CheckingAccountStatus.0.to.200          0.057760 .
## CheckingAccountStatus.gt.200            0.120895
## CheckingAccountStatus.none                    NA
## CreditHistory.NoCredit.AllPaid          0.033591 *
## CreditHistory.ThisBank.AllPaid          0.415937
## CreditHistory.PaidDuly                  0.035090 *
## CreditHistory.Delay                     0.783495 NA
```

Stepwise backward method:

```
model.aic.backward <- step(model_full, direction = "backward", trace = 1)
```

```
-631.804    266.618  -2.370 0.018118
## Purpose.Furniture.Equipment         -850.244    289.058  -2.941 0.003393
## Purpose.Radio.Television            -903.521    264.821  -3.412 0.000689
## Purpose.DomesticAppliance          -1706.405    818.533  -2.085 0.037517
## Purpose.Education                   -934.532    395.898  -2.361 0.018567
## Purpose.Business                   -1085.877    322.060  -3.372 0.000795
## SavingsAccountBonds.lt.100          -684.623    212.301  -3.225 0.001329
## SavingsAccountBonds.100.to.500      -932.889    300.706  -3.102 0.002010
## SavingsAccountBonds.500.to.1000     -743.534    365.258  -2.036 0.042225
## SavingsAccountBonds.gt.1000         -731.423    398.918  -1.834 0.067220
## EmploymentDuration.4.to.7            469.317    223.601   2.099 0.036242
## EmploymentDuration.gt.7             -395.045    196.308  -2.012 0.044626
## Personal.Male.Divorced.Seperated     735.460    390.895   1.881 0.060390
## Personal.Male.Single                 589.344    172.602   3.414 0.000682
## OtherDebtorsGuarantors.CoApplicant   764.137    405.387   1.885 0.059918
## Property.RealEstate                 -811.130    266.463  -3.044 0.002436
## Property.Insurance                  -622.204    266.046  -2.339 0.019677
## Property.CarOther                   -680.890    243.400  -2.797 0.005316
## Job.UnemployedUnskilled            -1843.059    526.274  -3.502 0.000496
## Job.UnskilledResident              -1341.534    301.221  -4.454 1.01e-05
## Job.SkilledEmployee                -1338.919    233.366  -5.737 1.52e-08
##
## (Intercept)                        ***
## Duration                           ***
## InstallmentRatePercentage          ***
## NumberPeopleMaintenance
## Telephone                          ***
## ClassBad
## CheckingAccountStatus.0.to.200     *
## CheckingAccountStatus.gt.200
## CreditHistory.NoCredit.AllPaid     *
## CreditHistory.PaidDuly             *
## Purpose.NewCar                     *
## Purpose.Furniture.Equipment        **
## Purpose.Radio.Television           ***
## Purpose.DomesticAppliance          *
## Purpose.Education                  *
## Purpose.Business                   ***
## SavingsAccountBonds.lt.100         **
## SavingsAccountBonds.100.to.500     **
## SavingsAccountBonds.500.to.1000    *
## SavingsAccountBonds.gt.1000        .
## EmploymentDuration.4.to.7          *
## EmploymentDuration.gt.7            *
```

```
## Personal.Male.Divorced.Seperated   .
## Personal.Male.Single               ***
## OtherDebtorsGuarantors.CoApplicant .
## Property.RealEstate                **
## Property.Insurance                 *
## Property.CarOther                  **
## Job.UnemployedUnskilled            ***
## Job.UnskilledResident              ***
## Job.SkilledEmployee                ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1891 on 601 degrees of freedom
## Multiple R-squared:  0.6172, Adjusted R-squared:  0.5981
## F-statistic:  32.3 on 30 and 601 DF,  p-value: < 2.2e-16
```

Stepwise forward method:

```
model.aic.forward <- step(model_full, direction = "forward", trace = 1)
```

```
## Step:  AIC=9567.11
## train$Amount ~ Duration + InstallmentRatePercentage + NumberPeopleMaintena
nce +
##     Telephone + ClassBad + CheckingAccountStatus.0.to.200 + CheckingAccoun
tStatus.gt.200 +
##     CreditHistory.NoCredit.AllPaid + CreditHistory.PaidDuly +
##     Purpose.NewCar + Purpose.Furniture.Equipment + Purpose.Radio.Televisio
n +
##     Purpose.DomesticAppliance + Purpose.Education + Purpose.Business +
##     SavingsAccountBonds.lt.100 + SavingsAccountBonds.100.to.500 +
##     SavingsAccountBonds.500.to.1000 + SavingsAccountBonds.gt.1000 +
##     EmploymentDuration.4.to.7 + EmploymentDuration.gt.7 + Personal.Male.Di
vorced.Seperated +
##     Personal.Male.Single + OtherDebtorsGuarantors.CoApplicant +
##     Property.RealEstate + Property.Insurance + Property.CarOther +
##     Job.UnemployedUnskilled + Job.UnskilledResident + Job.SkilledEmployee
##
##                                    Df  Sum of Sq        RSS    AIC
## <none>                                             2149742260 9567.1
## + OtherInstallmentPlans.None        1    6470948 2143271312 9567.2
## - NumberPeopleMaintenance           1    7244761 2156987021 9567.2
## - ClassBad                          1    8229655 2157971915 9567.5
```

```
## - Job.SkilledEmployee                  1  117745566 2267487826 9598.8
## - InstallmentRatePercentage            1  538410432 2688152692 9706.4
## - Duration                             1 1024973705 3174715965 9811.5
```

Stepwise in both direction:

```
summary(model.aic.both)
```

```
##
## Call:
## lm(formula = train$Amount ~ Duration + InstallmentRatePercentage +
##     NumberPeopleMaintenance + Telephone + ClassBad + CheckingAccountStatus
.0.to.200 +
##     CheckingAccountStatus.gt.200 + CreditHistory.NoCredit.AllPaid +
##     CreditHistory.PaidDuly + Purpose.NewCar + Purpose.Furniture.Equipment
+
##     Purpose.Radio.Television + Purpose.DomesticAppliance + Purpose.Educati
on +
##     Purpose.Business + SavingsAccountBonds.lt.100 + SavingsAccountBonds.10
0.to.500 +
##     SavingsAccountBonds.500.to.1000 + SavingsAccountBonds.gt.1000 +
##     EmploymentDuration.4.to.7 + EmploymentDuration.gt.7 + Personal.Male.Di
vorced.Seperated +
##     Personal.Male.Single + OtherDebtorsGuarantors.CoApplicant +
##     Property.RealEstate + Property.Insurance + Property.CarOther +
##     Job.UnemployedUnskilled + Job.UnskilledResident + Job.SkilledEmployee,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5712.5 -1083.3  -142.9   722.1 10375.5
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      6859.063    574.104  11.947  < 2e-16
## Duration                          123.328      7.286  16.928  < 2e-16
## InstallmentRatePercentage        -870.616     70.962 -12.269  < 2e-16
## NumberPeopleMaintenance          -339.217    238.354  -1.423 0.155206
## Telephone                        -614.903    175.402  -3.506 0.000489
## ClassBad                          274.948    181.265   1.517 0.129837
## CheckingAccountStatus.0.to.200    411.354    180.265   2.282 0.022842
## CheckingAccountStatus.gt.200     -528.517    346.652  -1.525 0.127877
## CreditHistory.NoCredit.AllPaid   1078.196    437.161   2.466 0.013927
## CreditHistory.PaidDuly           -388.261    162.043  -2.396 0.016878
## Purpose.NewCar                   -631.804    266.618  -2.370 0.018118
## Purpose.Furniture.Equipment      -850.244    289.058  -2.941 0.003393
## Purpose.Radio.Television         -903.521    264.821  -3.412 0.000689
## Purpose.DomesticAppliance       -1706.405    818.533  -2.085 0.037517
```

```
## Purpose.Education                    -934.532    395.898  -2.361 0.018567
## Purpose.Business                    -1085.877    322.060  -3.372 0.000795
## SavingsAccountBonds.lt.100           -684.623    212.301  -3.225 0.001329
## SavingsAccountBonds.100.to.500       -932.889    300.706  -3.102 0.002010
## SavingsAccountBonds.500.to.1000      -743.534    365.258  -2.036 0.042225
## SavingsAccountBonds.gt.1000          -731.423    398.918  -1.834 0.067220
## EmploymentDuration.4.to.7             469.317    223.601   2.099 0.036242
## EmploymentDuration.gt.7              -395.045    196.308  -2.012 0.044626
## Personal.Male.Divorced.Seperated      735.460    390.895   1.881 0.060390
## Personal.Male.Single                  589.344    172.602   3.414 0.000682
## OtherDebtorsGuarantors.CoApplicant    764.137    405.387   1.885 0.059918
## Property.RealEstate                  -811.130    266.463  -3.044 0.002436
## Property.Insurance                   -622.204    266.046  -2.339 0.019677
## Property.CarOther                    -680.890    243.400  -2.797 0.005316
## Job.UnemployedUnskilled             -1843.059    526.274  -3.502 0.000496
## Job.UnskilledResident               -1341.534    301.221  -4.454 1.01e-05
## Job.SkilledEmployee                 -1338.919    233.366  -5.737 1.52e-08
##
## (Intercept)                         ***
## Duration                            ***
## InstallmentRatePercentage           ***
## NumberPeopleMaintenance
## Telephone                           ***
## ClassBad
## CheckingAccountStatus.0.to.200      *
## CheckingAccountStatus.gt.200
## CreditHistory.NoCredit.AllPaid      *
## CreditHistory.PaidDuly              *
## Purpose.NewCar                      *
## Purpose.Furniture.Equipment         **
## Purpose.Radio.Television            ***
## Purpose.DomesticAppliance           *
## Purpose.Education                   *
## Purpose.Business                    ***
## SavingsAccountBonds.lt.100          **
## SavingsAccountBonds.100.to.500      **
## SavingsAccountBonds.500.to.1000     *
## SavingsAccountBonds.gt.1000         .
## EmploymentDuration.4.to.7           *
## EmploymentDuration.gt.7             *
## Personal.Male.Divorced.Seperated    .
## Personal.Male.Single                ***
## OtherDebtorsGuarantors.CoApplicant  .
## Property.RealEstate                 **
## Property.Insurance                  *
## Property.CarOther                   **
## Job.UnemployedUnskilled             ***
## Job.UnskilledResident               ***
## Job.SkilledEmployee                 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1891 on 601 degrees of freedom
## Multiple R-squared:  0.6172, Adjusted R-squared:  0.5981
## F-statistic:  32.3 on 30 and 601 DF,  p-value: < 2.2e-16
```

Using lasso regression to identify the best predictors

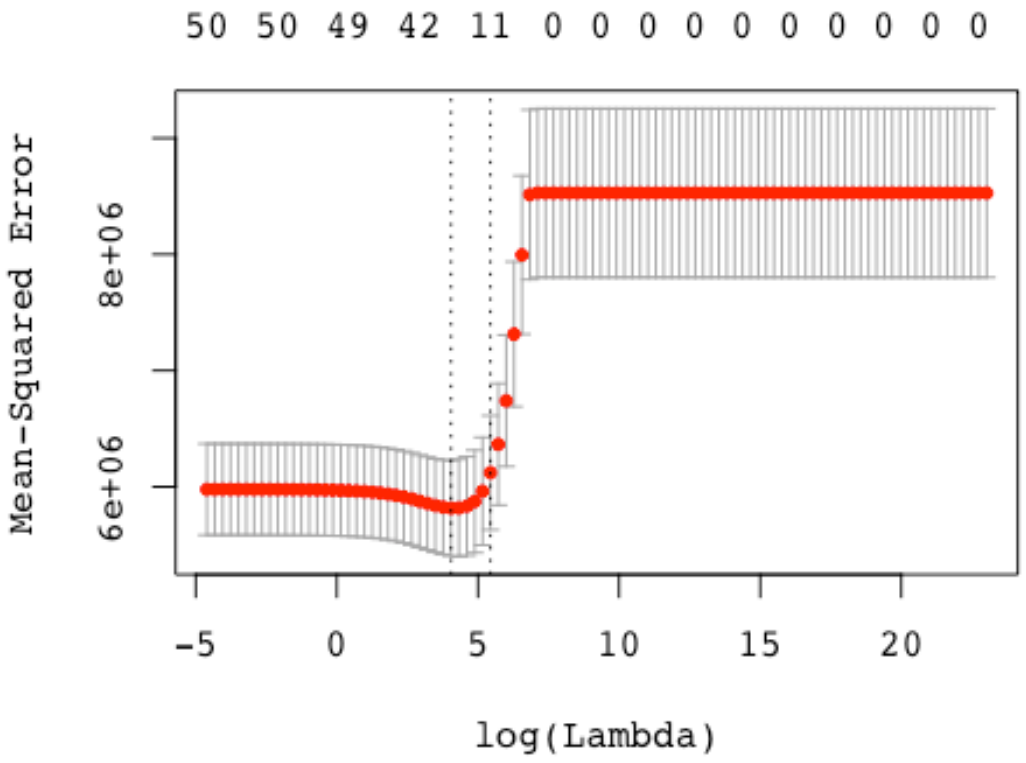
```
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-5
```

```
grid <- 10 ^ seq(from = 10, to = -2, length = 100)
x <- model.matrix(Amount~., data = GC)[, -2]
y <- GC$Amount

set.seed(123)
train_lasso <- sample(nrow(x), size =  round(0.632 * nrow(x)))
y.test <- y[-train_lasso]

set.seed(1)
lasso.cv <- cv.glmnet(x[train_lasso, ], y[train_lasso], alpha = 1, lambda = g
rid)
plot(lasso.cv)
```




```
bestlam <- lasso.cv$lambda.min
bestlam

## [1] 57.22368

lasso.cv$nzero

##  s0  s1  s2  s3  s4  s5  s6  s7  s8  s9 s10 s11 s12 s13 s14 s15 s16 s17
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## s18 s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## s36 s37 s38 s39 s40 s41 s42 s43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## s54 s55 s56 s57 s58 s59 s60 s61 s62 s63 s64 s65 s66 s67 s68 s69 s70 s71
##   0   0   0   0   1   4   7   8   9  11  15  18  25  30  33  36  37  41
## s72 s73 s74 s75 s76 s77 s78 s79 s80 s81 s82 s83 s84 s85 s86 s87 s88 s89
##  42  44  45  45  48  48  49  49  49  49  50  50  50  50  50  50  50  50
## s90 s91 s92 s93 s94 s95 s96 s97 s98 s99
##  50  50  50  50  50  50  50  50  50  50

lasso.cv$glmnet.fit

##
## Call:  glmnet(x = x[train_lasso, ], y = y[train_lasso], lambda = grid,
```

```
alpha = 1)


lasso.pred <- predict(object = lasso.cv, s = bestlam, newx = x[-train_lasso,
])
mean((y.test - lasso.pred)^2) # MSE

## [1] 4632443

sqrt(mean((y.test - lasso.pred)^2)) # residual standard error

## [1] 2152.311

cor(y.test, lasso.pred)^2 # r2holdout

##             1
## [1,] 0.344821
```

Below predictors were selected after all the EDA models:

```
model_lm<- lm(formula = train$Amount ~ Duration + InstallmentRatePercentage +
Telephone + Job.Management.SelfEmp.HighlyQualified
            +     Personal.Male.Single + CreditHistory.NoCredit.AllPaid ,data
= train)

summary(model_lm)

##
## Call:
## lm(formula = train$Amount ~ Duration + InstallmentRatePercentage +
##     Telephone + Job.Management.SelfEmp.HighlyQualified + Personal.Male.Sin
gle +
##     CreditHistory.NoCredit.AllPaid, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5857.2 -1052.8  -212.2   689.5 10982.5
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                           3040.067    290.056  10.481
## Duration                               136.758      6.908  19.797
## InstallmentRatePercentage             -914.132     70.370 -12.990
## Telephone                             -729.660    176.255  -4.140
## Job.Management.SelfEmp.HighlyQualified 1632.679    233.170   7.002
```

```
## Personal.Male.Single                     626.005    159.750   3.919
## CreditHistory.NoCredit.AllPaid          1244.691    421.823   2.951
##                                        Pr(>|t|)
## (Intercept)                             < 2e-16 ***
## Duration                                < 2e-16 ***
## InstallmentRatePercentage               < 2e-16 ***
## Telephone                              3.95e-05 ***
## Job.Management.SelfEmp.HighlyQualified 6.53e-12 ***
## Personal.Male.Single                   9.89e-05 ***
## CreditHistory.NoCredit.AllPaid          0.00329 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1967 on 625 degrees of freedom
## Multiple R-squared:  0.5693, Adjusted R-squared:  0.5652
## F-statistic: 137.7 on 6 and 625 DF,  p-value: < 2.2e-16
```

Mutate Function:

Below code has the mutate function, I have divided the data into 63.31 and 36.8 f or train and test, the code below runs the mutate and stores the coefficient and o ther important statistics from each 1000 randomly sampled iterations.

```
intercept <- c()
Duration <- c()
InstallmentRatePercentage <- c()
Telephone<-c()
Job.Management.SelfEmp.HighlyQualified <- c()
Personal.Male.Single <- c()
CreditHistory.NoCredit.AllPaid <- c()
rsquared.train <- c()
rsquared.holdout <- c()

# create 1000 separate OLS fits using different 63.32/36.8 training/validatio
n sets.
for (i in 1:1000) {

  # create reproducible, randomly sampled training vs validation data.
  set.seed(i + 1)
  indices <- sample(nrow(GC), size = round(0.632 * nrow(GC)))
  data.train <- GC[indices, ]
  data.holdout <- GC[-indices, ]

  # linear regression model fit with training data. Predictors chosen from pr
ior EDA models.
  model_lm<- lm(formula = Amount ~ Duration + InstallmentRatePercentage + Tel
ephone + Job.Management.SelfEmp.HighlyQualified
                +     Personal.Male.Single + CreditHistory.NoCredit.AllPaid ,
data = data.train)


  # capture fitted coefficient values.
  coeff <- coefficients(model_lm)
  intercept[i] <- coeff[1]
  Duration[i] <- coeff[2]
  InstallmentRatePercentage[i] <- coeff[3]
  Telephone[i] <- coeff[4]
  Job.Management.SelfEmp.HighlyQualified[i] <- coeff[5]
  Personal.Male.Single[i]<-coeff[6]
  CreditHistory.NoCredit.AllPaid<-coeff[7]


  # capture r-squared from training
  rsquared.train[i] <- summary(model_lm)$r.squared
```

```
  # predict model_lm on unseen records, store predictions.
  predicted.amount <- predict(model_lm,newdata=data.holdout)
  # calculate the r-squared on holdout data.
  # square the correlation between actual vs predicted amount.
  rsquared.holdout[i] <- cor(data.holdout$Amount, predicted.amount)^2

}
```

Get all the output in the dataframe

```
results_mutate <- data.frame(model_number = rep(1:1000), rsquared.train, rsqu
ared.holdout,intercept,
                             + Duration, InstallmentRatePercentage,Telephone,J
ob.Management.SelfEmp.HighlyQualified,
                             + Personal.Male.Single, CreditHistory.NoCredit.Al
lPaid)
```

```
## Warning in data.frame(model_number = rep(1:1000), rsquared.train,
## rsquared.holdout, : row names were found from a short variable and have
## been discarded
```

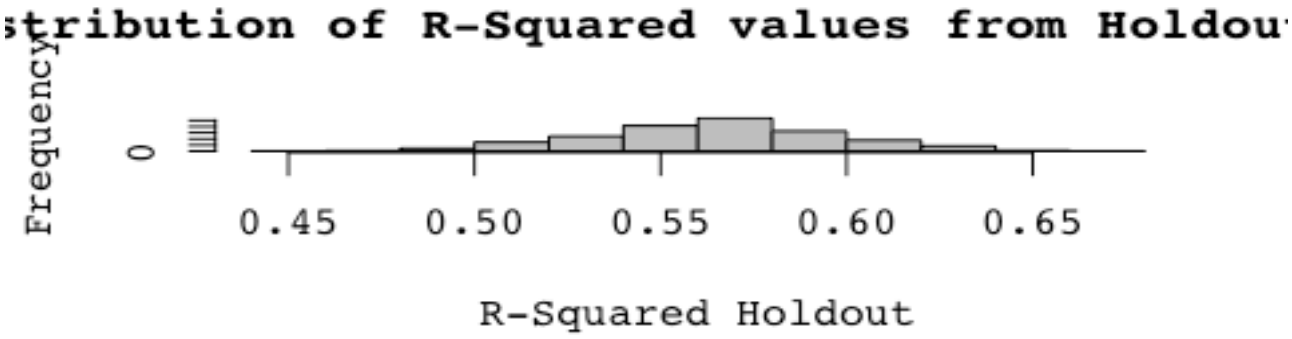
```
stats_mutate <- results_mutate %>%
summarise(intercept.mean.coef = mean(intercept),
            intercept.sd.coef = sd(intercept),
            Duration.mean.coef = mean(Duration),
            Duration.sd.coef = sd(Duration),
            InstallmentRatePercentage.mean.coef = mean(InstallmentRatePercent
age),
            InstallmentRatePercentage.sd.coef = sd(InstallmentRatePercentage)
,
            Telephone.mean.coef = mean(Telephone),
            Telephone.sd.coef = sd(Telephone),
            Job.Management.SelfEmp.HighlyQualified.mean.coef = mean(Job.Manag
ement.SelfEmp.HighlyQualified),
            Job.Management.SelfEmp.HighlyQualified.sd.coef = sd(Job.Managemen
t.SelfEmp.HighlyQualified),
            Personal.Male.Single.mean.coef=mean(Personal.Male.Single),
            Personal.Male.Single.sd.coef=sd(Personal.Male.Single),
            CreditHistory.NoCredit.AllPaid.mean.coef=mean(CreditHistory.NoCre
dit.AllPaid),
            CreditHistory.NoCredit.AllPaid.mean.coef=sd(CreditHistory.NoCredi
t.AllPaid),
            rsquared.train.mean = mean(rsquared.train),
            rsquared.train.sd = sd(rsquared.train),
            rsquared.holdout.mean = mean(rsquared.holdout),
            rsquared.holdout.sd = sd(rsquared.holdout),
            rsquared.difference.mean = mean(rsquared.holdout.mean - rsquared.
train.mean)
  )
```

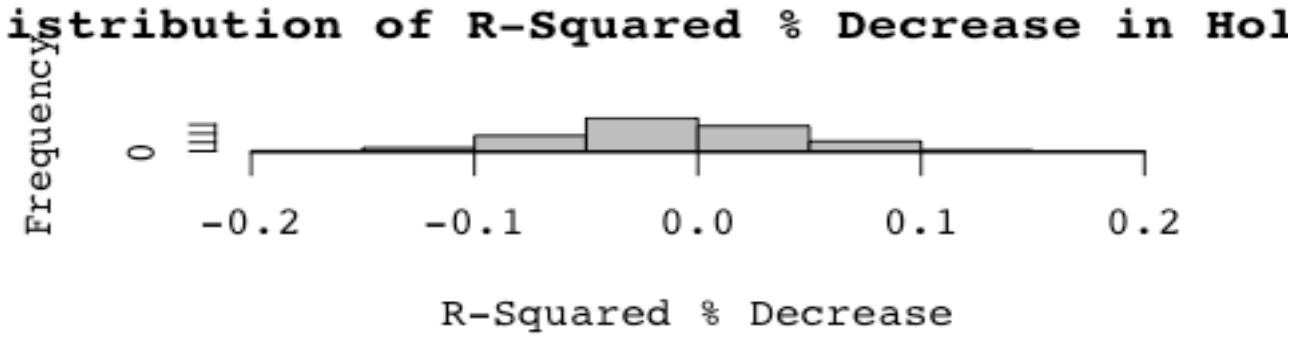
```
#########


coef.means <- rbind(stats_mutate[, c(1,3,5,7,9)])
coef.sd <- rbind(stats_mutate[, c(2,4,6,8,10)])

reshape.coef.stats <- data.frame(coefficient = c("intercept", "Duration",
                                                 "Job.Management.SelfEmp.High
lyQualified",
                                                 "InstallmentRatePercentage",
                                                 "Purpose.UsedCar"),
                                 coef.means = as.numeric(coef.means), coef.sd
= as.numeric(coef.sd))
reshape.coef.stats

##                               coefficient coef.means    coef.sd
## 1                               intercept  2673.6651 189.292595
## 2                                Duration   136.9176   5.065651
## 3 Job.Management.SelfEmp.HighlyQualified  -840.6597  47.178241
## 4              InstallmentRatePercentage  -569.9471 100.445355
## 5                         Purpose.UsedCar  1617.8857 206.038946
```

Plots

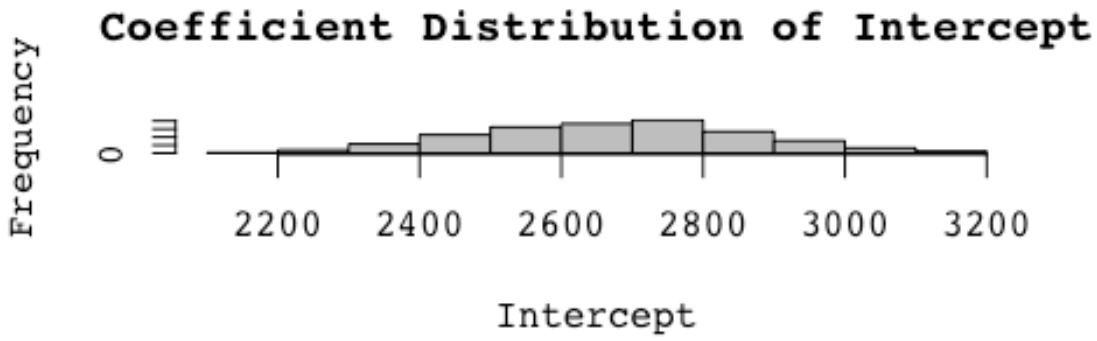
Below are the distribution of the rsquared values of the holdout and other co
effcients:

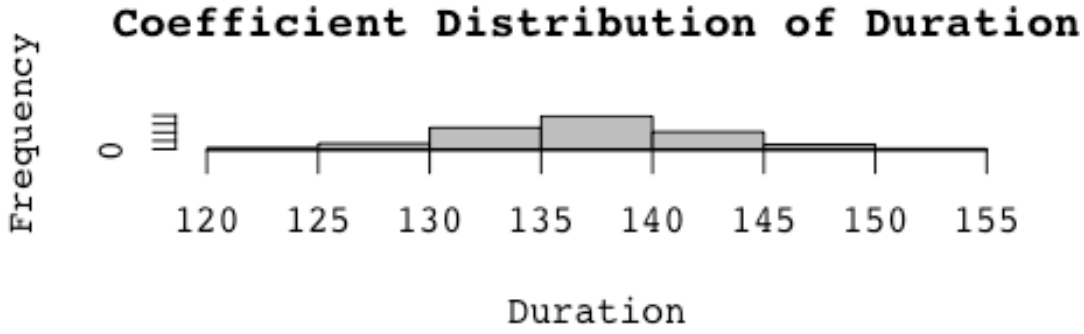
```
par(mfrow = c(2,1))
# distribution of test Rsquared.
hist(results_mutate$rsquared.holdout, xlab = "R-Squared Holdout", col = "grey
",
     main = "Distribution of R-Squared values from Holdout Data")
abline(v = results_mutate$rsquared.holdout.mean, lty = "dashed", lwd = "3", c
ol = "blue")



hist(results_mutate$rsquared.holdout - results_mutate$rsquared.train, xlab =
"R-Squared % Decrease",
     col = "grey", main = "Distribution of R-Squared % Decrease in Holdout")
abline(v = results_mutate$rsquared.difference.mean, lty = "dashed", lwd = "3"
, col = "blue")
```

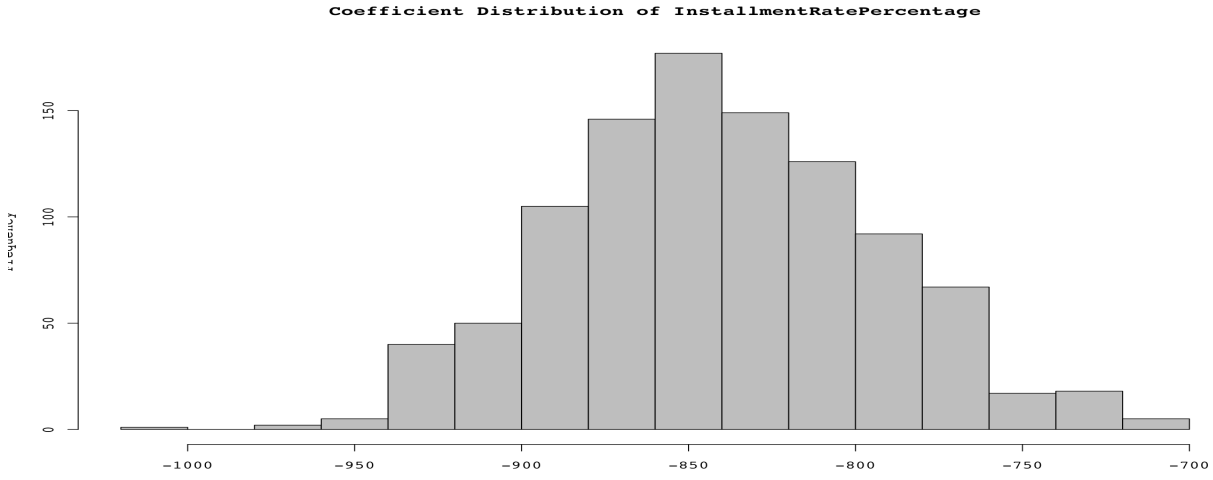


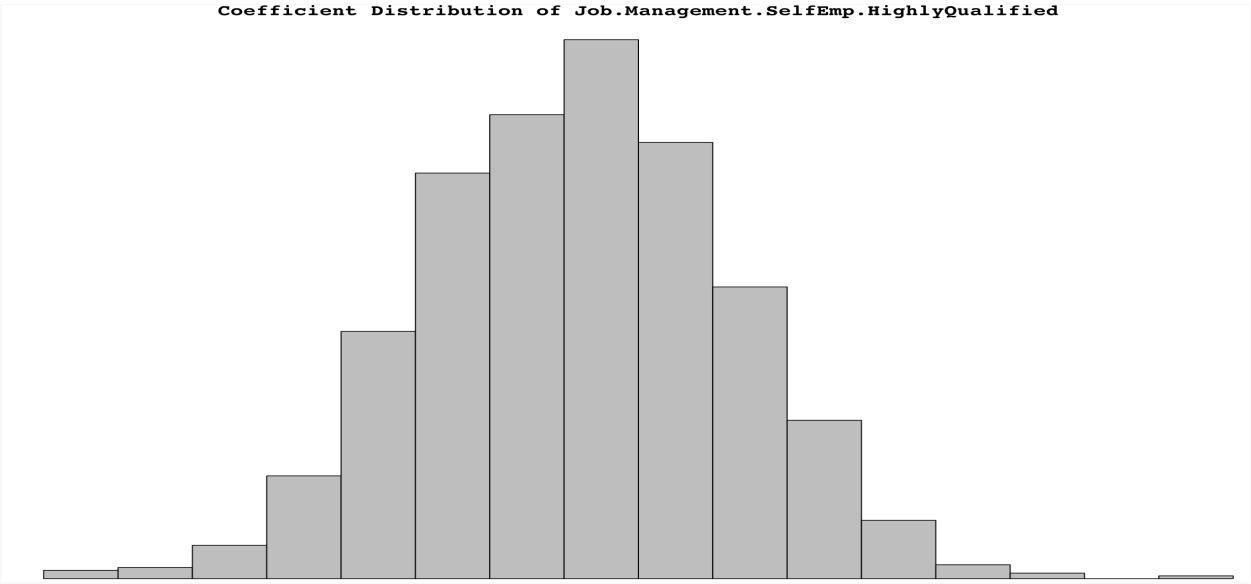



```r
# distribution of coefficient

hist(results_mutate$intercept, col = "grey", xlab = "Intercept",
     main = paste("Coefficient Distribution of", "Intercept"))
abline(v = results_mutate$intercept.mean.coef, lty = "dashed", lwd = "3", col
= "blue")

# distribution of duration predictor coef.
hist(results_mutate$X.Duration, col = "grey", main = paste("Coefficient Distr
ibution of", "Duration"),
     xlab = "Duration")
abline(v = results_mutate$X.Duration.mean.coef, lty = "dashed", lwd = "3", co
l = "blue")
```


Coefficient Distribution of Intercept
Frequency
0
2200
2400
2600
2800
3000
3200
Intercept


Coefficient Distribution of Duration
Frequency
0
120
125
130
135
140
145
150
155
Duration


Coefficient Distribution of InstallmentRatePercentage
Frequency
0
50
100
150
−1000
−950
−900
−850
−800
−750
−700




```
##installment rate percentage

hist(results_mutate$InstallmentRatePercentage, col = "grey",
     main = paste("Coefficient Distribution of", "InstallmentRatePercentage")
,
     xlab = "InstallmentRatePercentage")
abline(v = results_mutate$InstallmentRatePercentage.mean.coef, lty = "dashed"
, lwd = "3",
       col = "blue")


# distribution of Job.Management.SelfEmp.HighlyQualified predictor coef.

graphics.off()
par("mar")

## [1] 5.1 4.1 4.1 2.1

par(mar=c(1,1,1,1))
hist(results_mutate$Job.Management.SelfEmp.HighlyQualified, col = "grey",
     main = paste("Coefficient Distribution of", "Job.Management.SelfEmp.High
lyQualified"),
     xlab = "Job.Management.SelfEmp.HighlyQualified")
abline(v = results_mutate$Job.Management.SelfEmp.HighlyQualified.mean.coef,
       lty = "dashed", lwd = "3", col = "blue")

## distribution of telephone coeff
hist(results_mutate$Telephone, col = "grey",
     main = paste("Coefficient Distribution of", "Purpose.UsedCar"), xlab = "
```

```
Purpose.UsedCar")
abline(v = results_mutate$Telephone, lty = "dashed", lwd = "3",
       col = "blue")

#### distribution of Personal.Male.Single

hist(results_mutate$X.Personal.Male.Single, col = "grey",
     main = paste("Coefficient Distribution of", "Purpose.UsedCar"), xlab = "
Purpose.UsedCar")


#### distribution of CreditHistory.NoCredit.AllPaid
graphics.off()
par("mar")

## [1] 5.1 4.1 4.1 2.1

par(mar=c(1,1,1,1))

hist(results_mutate$CreditHistory.NoCredit.AllPaid, col = "grey",
     main = paste("Coefficient Distribution of", "Purpose.UsedCar"), xlab = "
Purpose.UsedCar")
abline(v = results_mutate$CreditHistory.NoCredit.AllPaid, lty = "dashed", lwd
= "3",
       col = "blue")
```

Compare these re-sampling results to a single model built on entire sample

```
full_model_lmfit <- lm(Amount ~ Duration + InstallmentRatePercentage + Telephone + Job.Management.SelfEmp.HighlyQualified
                 +     Personal.Male.Single + CreditHistory.NoCredit.AllPaid, data = GC)
summary(full_model_lmfit)

##
## Call:
## lm(formula = Amount ~ Duration + InstallmentRatePercentage +
##     Telephone + Job.Management.SelfEmp.HighlyQualified + Personal.Male.Single +
##     CreditHistory.NoCredit.AllPaid, data = GC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5324.6 -1042.6  -179.0   693.9 11349.8
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                           2678.013    217.248  12.327
## Duration                               136.842      5.024  27.237
## InstallmentRatePercentage             -840.835     52.991 -15.867
## Telephone                             -572.161    130.451  -4.386
## Job.Management.SelfEmp.HighlyQualified 1613.843    179.806   8.975
## Personal.Male.Single                   547.007    119.615   4.573
## CreditHistory.NoCredit.AllPaid         875.777    301.835   2.902
##                                       Pr(>|t|)
## (Intercept)                            < 2e-16 ***
## Duration                               < 2e-16 ***
## InstallmentRatePercentage              < 2e-16 ***
## Telephone                             1.28e-05 ***
## Job.Management.SelfEmp.HighlyQualified  < 2e-16 ***
## Personal.Male.Single                  5.41e-06 ***
## CreditHistory.NoCredit.AllPaid          0.0038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1852 on 993 degrees of freedom
## Multiple R-squared:  0.5719, Adjusted R-squared:  0.5694
## F-statistic: 221.1 on 6 and 993 DF,  p-value: < 2.2e-16
```

```r
# names of different objects saved with the fitted model.
names(full_model_lmfit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```r
# Names of the objects within the function
names(summary(model_lm))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
# model coefficients  ## full model r squared value
full_model_lmfit.coef <- data.frame(coefficients(full_model_lmfit))
full_model_lmfit.coef
```

```
##                                       coefficients.full_model_lmfit.
## (Intercept)                                                 2678.0130
## Duration                                                     136.8423
## InstallmentRatePercentage                                   -840.8349
## Telephone                                                   -572.1608
## Job.Management.SelfEmp.HighlyQualified                      1613.8429
## Personal.Male.Single                                         547.0069
## CreditHistory.NoCredit.AllPaid                               875.7771
```

```r
round(summary(full_model_lmfit)$r.squared, 3)
```

```
## [1] 0.572
```