Recognition of Human Activities using Smartphone sensors

SHARVIL SHETTY
AISHWARYA ARYA
ARVIND ELANGOVAN
PRITHVI BHAT
WILLIAM MEAD

DATE: 12/12/2017

Name	% Contribution
SHARVIL SHETTY	20%
AISHWARYA ARYA	20%
ARVIND ELANGOVAN	20%
PRITHVI BHAT	20%
WILLIAM MEAD	20%

Table of Contents

Executive Summary 4
Introduction 5
Project approach 6
- Description of new technique 7

Implementation details 8
- Initial data loading 8
- Implementing random forest on the dataset 9
- Selecting important predictors 9
- Creating datasets with 353 chosen columns 9
- Building a random forest model for top 353 predictors 10
- Implementing PCA 11
- Selecting principal components post PCA 12
- Implementing random forest with predictors from PCA 12
- LDA using 45 predictors determined by PCA 14
- LDA on predictors determined by randomforest$importance 14
- QDA using 45 predictors determined by PCA 15
- QDA on predictors determined by randomforest$importance 15
- Constructing a cross-validated KNN model 15
- SVM using 45 predictors determined by PCA 16
- SVM using 353 predictors determined by randomforest$importance 17
- Neural Networks implementation 18

Comparison 21
Conclusion 22
References 23
Appendix 24
- Code execution results: 24
- Markdown File for scripts 73
- Data Files and Execution scripts for easy execution by TA 73

Executive Summary

Activity tracking has exploded in health/fitness world. It has become a very popular and lucrative aspect of the market. Companies like Fitbit and Fitness Tracker Plus owe their existence to the ability to model human activities and predict a person's energy burning rate. A lot of other tech companies are getting in on the action like Samsung, Garmin, and Apple. In the year 2020 alone, it is expected that over 170 million fitness trackers/wearable devices are to be sold[1]. That is why it is very important that human activity models must be very accurate. This project outlines our process in the making of a human activity prediction model.

The objective was to create a model with greater than 90% classification rate for 12 activities from data collected by cell phone accelerometers.

1. Walking
2. Walking Upstairs
3. Walking Downstairs
4. Sitting
5. Standing
6. Laying
7. Stand to Sit
8. Sit to Stand
9. Sit to Lie
10. Lie to Sit
11. Stand to Lie
12. Lie to Stand

We used a few different techniques to create this model. We started off by reducing the number of predictors to 353 by eliminating the unimportant predictors and keeping the ones that were statistically important. From there we reduced the data through Principal Component Analysis (PCA). We then used predictors obtained by these two methods to create 2 models for each method (one using PCA predictors and the other using random forest predictors) viz. Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Neural Networks (neuralnet) models.

Neural Networks was chosen as the new method of creating the human activity model because it allows us to analyze the time-series better by using a network of loops. This allows the model to use information from previous passes, which acts as a memory. This capability of Neural Networks enables the model to better detect changes over time.

We chose the model with the best classification rate as our final human activity classifier. This model was Neural Networks with a correct classification rate (Accuracy) of 92%.

[1] https://www.statista.com/topics/1556/wearable-technology/

Introduction

Human activity monitoring and human activity recognition are intensively researched in recent years (Capela, *et al.,* 2015). The advancement in technology and designing of smart living have necessitated the need for information that can be used to assist Humans in day-to-day tasks. Human activity recognition (HAR) or monitoring are studied to enhance the preciseness of smart devices. The ease of availability of consumer devices with sensors such as audio and vision sensors, temperature and light sensors, GPS sensors, gyroscope and acceleration and direction sensors have generated vast amounts of data that can be used for delineating human activities and solving many issues related to safety, healthcare, surveillance and home automation (Hung, *et al.,* 2014). Most scientific researchers are trying to solve issues related to elderly people, sportspersons and patients (Sunny, *et al.,* 2015). The problem that is most commonly faced with such data sets is its vastness and channeling it so that it can be interpretable. Such activity recognition systems input direct sensor reading and estimates a human activity through machine learning.

In study below, accelerometer and gyroscope sensor data from smartphones were collected to model a system which recognizes the human activity, with an intention to determine signal features best suited for activity recognition using waist-worn smart phones. The study aims to establish a relationship between the sensor signals (from gyroscopes and accelerometers) to predict human activities (standing, sitting, lying, walking and climbing stairs). Numeric predictors are used to predict a categorical response.

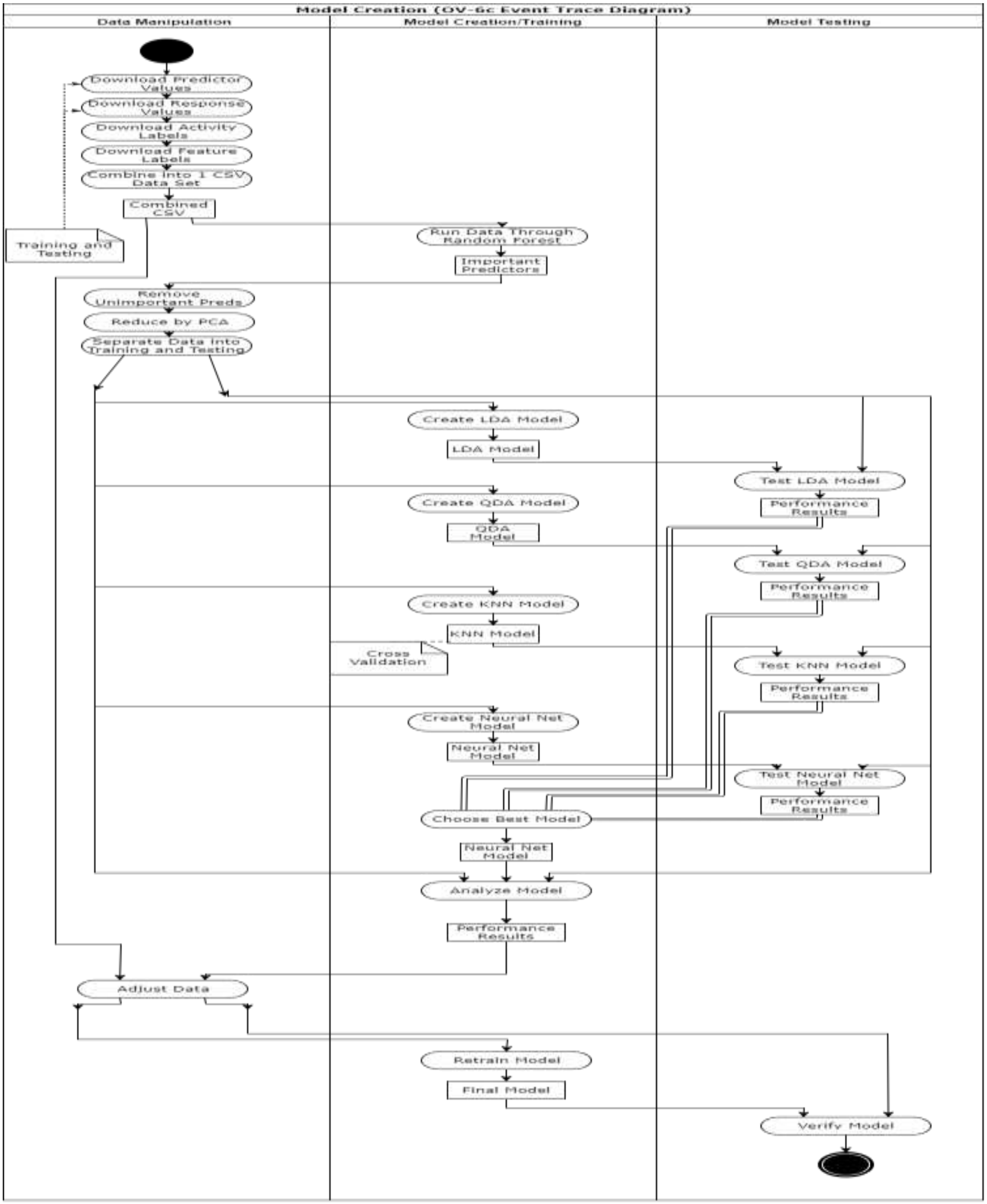
The dataset used is collected from a group of 30 volunteers aged between 19-48 years. The volunteers were made to wear smart phone devices on their waist while carrying out activities such as standing, sitting, lying, walking and going up or down stairs. Etc. 3-axial linear acceleration and 3-axial angular velocity were measured at a constant rate of 50Hz using the gyroscopes and accelerometers embedded in the smartphone.

The data used for training the model was generated using 70% of the volunteers and the testing data was generated using 30% of the volunteers. The raw data received was cleaned using filtering methods.

This dataset acquired from the UC Irvine's machine learning repository has 561 attributes and 10,299 rows of data. The range of the study involves the use of all classification methods from random forests to support vector machines. The project starts with choosing the important predicts after calculating the importance by random forests and reducing the dimension to be able to describe the variance through fewer number of components. LDA, QDA, KNN, Random forest and SVM will be performed to pick the best model and compared using misclassification error.

The new technique used in the study to encourage learning is Neural Networks (neuralnet). The use of loops using memory from previous passes will enable the model to detect changes over time and allows prediction of motion of user and motion related disorders.

Project approach



Text

Description of new technique

Methods like KNN, QDA and Decision Trees are not very effective when it comes to multivariate time series data as being generated by the sensors of the smartphone device. Traditional analyses of time series were mainly concerned with modeling the autocorrelation structure in a time series, and they require the data under study to be stationary. Trends in time series, most of the time, violate the condition of stationarity. Thus, the removal of the trend is often desirable in the time–series analysis and prediction. Compared to traditional approaches, neural networks have shown some promise in time series forecasting.

A Neural Network is basically a network of computing units linked by directed connections. Each computing unit performs some calculation and outputs a value that is then spread through all its outgoing connections as input into other units. Connections normally have weights that correspond to how strong two units are linked. Typically, the computation performed by a unit is separated into two stages: the aggregation and the activation functions. Applying the aggregation function commonly corresponds to calculating the sum of the inputs received by the unit through all its incoming connections. The resulting value is then fed into the activation function. It commonly varies in different network architectures. Most popular choices are the logistic sigmoid and the hyperbolic tangent functions. We have used the logistic sigmoid function to model the data in our case.

Neural Networks contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. 'Learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. When a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights.

Implementation details

Initial data loading

```
#loading training sets
training_x <- read.table("./X_train.txt", header = FALSE)
train_x_df = as.data.frame(training_x)

training_response <- read.table("./Y_train.txt")
train_response_df = as.data.frame(training_response)
names(train_response_df) = "response"
train_response_df$response = as.factor(train_response_df$response)

#removing non-transformed data
rm(training_x)
rm(training_response)

#creating dataset of x and y combined
train_df = cbind(train_x_df,train_response_df)

#rm(X_train,train)
#write.csv(hapt_train,"train.csv")

#loading testing data sets
testing_x <- read.table("./X_test.txt", header = FALSE)
test_x_df = as.data.frame(testing_x)

testing_response <- read.table("./Y_test.txt")
test_response_df = as.data.frame(testing_response)
names(test_response_df) = "response"
test_response_df$response = as.factor(test_response_df$response)



#removing non-transformed data
rm(testing_x,testing_response)
```

```
#configuring the output to be capable of prinitng all output lines
options(max.print = 12000)
```

Implementing random forest on the dataset

Random forest was implemented on the dataset to get the importance of the predictors to obtain the desired classification.

```
#implementing random forest on the data set
hapt_randomForest = randomForest(train_df$response ~ ., data=train_df, mtry=2
4 ,importance= TRUE)
hapt_randomForest$importance
```

Output for the importance values have been added to appendix.

Selecting important predictors

```
#We exported the above numbers we got from randomforest$importance into an ex
cel sheet which is being imported below
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
rm(Importance)

#We selected the Highest values for %IncMSE and set the bar at zero. Leaving
us with 353 predictors to create a random forest model

model_params = head(importance_sorted$Parameter,353)
predictors= paste(model_params, sep = "+")

model_expression = formula(paste("response~",paste(predictors,collapse = "+")
))
rm(predictors,model_params)
```

Creating datasets with 353 chosen columns

```
#Making training dataset with 353 important columns
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
model_params_353 = head(importance_sorted$Parameter,353)
predictors_353= paste(model_params_353, sep = "+")
X_train_353 = train_x_df[,predictors_353]
train_353 = cbind(X_train_353,response = train_response_df$response)
write.csv(X_train_353,"train_353.csv")

#Making testing dataset with 353 important columns
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
model_params_353_test = head(importance_sorted$Parameter,353)
predictors_353= paste(model_params_353_test, sep = "+")
X_test_353 = test_x_df[,predictors_353]
```

```r
write.csv(X_test_353,"test_353.csv")

#removing unused variables
rm(Importance,importance_sorted)
```

Building a random forest model for top 353 predictors

```r
#building a random forest model for top 353 predictors
hapt_randomForest_model_353 = randomForest(model_expression,data = train_df,
mtry=19, importance = TRUE)
summary(hapt_randomForest_model_353)

##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted        7767  factor numeric
## err.rate         6500  -none- numeric
## confusion         156  -none- numeric
## votes           93204  matrix numeric
## oob.times        7767  -none- numeric
## classes            12  -none- character
## importance       4942  -none- numeric
## importanceSD     4589  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                7767  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call

#Making predictions on Test data using above Random Forest model
yhat.rf_model_353 = predict(hapt_randomForest_model_353, newdata = test_x_df)

#Checking prediction power of the model
classification = yhat.rf_model_353 == test_response_df$response
classification_rate = sum(classification)/length(classification)
classification_rate

## [1] 0.9038583

table(test_response_df$response, yhat.rf_model_353)

##     yhat.rf_model_353
##        1   2   3   4   5   6   7   8   9  10  11  12
##   1  483   6   7   0   0   0   0   0   0   0   0   0
##   2   44 420   7   0   0   0   0   0   0   0   0   0
##   3   18  47 355   0   0   0   0   0   0   0   0   0
##   4    0   0   0 442  65   0   1   0   0   0   0   0
```

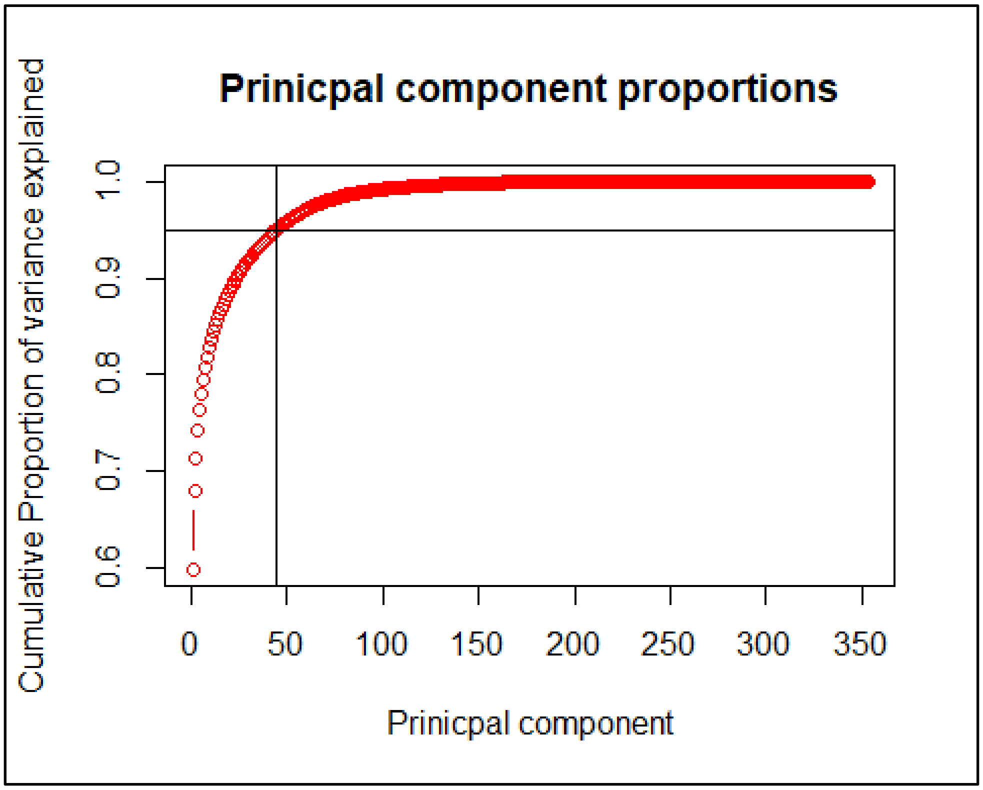
```
##    5    0    0    0   51  504    0    1    0    0    0    0    0
##    6    0    1    0    0    0  544    0    0    0    0    0    0
##    7    0    1    0    2    0    0   18    1    0    0    1    0
##    8    0    0    0    0    0    0    1    9    0    0    0    0
##    9    0    0    0    0    0    0    0    0   24    0    8    0
##   10    0    0    0    0    0    0    0    0    0   16    1    8
##   11    3    0    0    1    0    0    3    0   11    1   30    0
##   12    0    0    0    0    0    0    0    0    0   11    3   13
```

Implementing PCA

```
#Implementing PCA on the dataset with 353 predictors selected after randomfor
est$importance
#train_353 <- read_csv("./train_353.csv")
attach(X_train_353)
pr.out = prcomp(X_train_353, scale = TRUE)
#names(pr.out)
pc.var = pr.out$sdev^2
pc.pvar = pc.var/sum(pc.var)
options(max.print = 100000)
#pr.out$center
#pr.out$scale
#pr.out$rotation
dim(pr.out$x)
```

```
## [1] 7767  353
```

```
#Plotting
plot(cumsum(pc.pvar), xlab = "Prinicpal component", ylab = "Cumulative Propor
tion of variance explained", type ='b', main = "Prinicpal component proportio
ns", col="red")
abline(h=0.95)
abline(v=45)
```



Selecting principal components post PCA

As seen from the above plot, the first 45 principal components explain 95% of the variance in the response   and hence we selected PC1 to PC45 for subsequent models based on PCA.

```
#Selecting the PC1 to PC45 prinicpal compenents from original data set

train_45_pca = data.frame(response = train_df$response, pr.out$x)
train_45_pca = train_45_pca[,1:46]
train_45_pca$response = as.factor(train_45_pca$response)

#transforming test data for PCA
test.45_pca = predict(pr.out, newdata =X_test_353 )
test.45_pca = as.data.frame(test.45_pca)
test.45_pca = test.45_pca[,1:45]
```

Implementing random forest with predictors from PCA

```
hapt_pca_rf_model = randomForest(response ~ .,data = train_45_pca, mtry=7, im
portance = TRUE)
summary(hapt_pca_rf_model)

##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
```

```
## predicted         7767  factor numeric
## err.rate          6500  -none- numeric
## confusion          156  -none- numeric
## votes            93204  matrix numeric
## oob.times         7767  -none- numeric
## classes             12  -none- character
## importance         630  -none- numeric
## importanceSD       585  -none- numeric
## localImportance      0  -none- NULL
## proximity            0  -none- NULL
## ntree                1  -none- numeric
## mtry                 1  -none- numeric
## forest              14  -none- list
## y                 7767  factor numeric
## test                 0  -none- NULL
## inbag                0  -none- NULL
## terms                3  terms  call

#Making predictions on Test data using above Random Forest model
yhat.rf_pca = predict(hapt_pca_rf_model, newdata = test.45_pca)

#Checking prediction power of the model
classification = yhat.rf_pca == test_response_df$response
classification_rate = sum(classification)/length(classification)
classification_rate

## [1] 0.8621126

table(test_response_df$response, yhat.rf_pca)

##     yhat.rf_pca
##        1   2   3   4   5   6   7   8   9  10  11  12
##   1  459   2  35   0   0   0   0   0   0   0   0   0
##   2   33 427  11   0   0   0   0   0   0   0   0   0
##   3   50  49 321   0   0   0   0   0   0   0   0   0
##   4    0   0   0 391 117   0   0   0   0   0   0   0
##   5    0   1   0  49 506   0   0   0   0   0   0   0
##   6    0   0   0  24   0 521   0   0   0   0   0   0
##   7    0   6   0   2   1   0  13   0   0   0   1   0
##   8    0   0   0   1   0   0   1   8   0   0   0   0
##   9    0   0   0   0   0   0   0   0  23   0   9   0
##   10   0   1   0   0   0   0   0   0   0  16   0   8
##   11   2   6   0   2   0   0   1   0  12   0  26   0
##   12   0   0   0   0   0   0   1   1   0   9   1  15

##removing unused datasets with 561 predictors
rm(train_df,train_x_df)
```

LDA using 45 predictors determined by PCA

```
hapt_lda_45 = lda(response ~ ., data = train_45_pca)

pred_lda_45 = predict(hapt_lda_45, newdata =  test.45_pca)

#Confusion matrix for the model
table(pred_lda_45$class, test_response_df$response)

##
##        1   2   3   4   5   6   7   8   9  10  11  12
##    1 475  32  29   0   0   0   0   0   0   0   1   0
##    2   8 439  52   0   0   0   1   0   0   0   0   0
##    3  13   0 339   0   0   0   0   0   0   0   0   0
##    4   0   0   0 328  51   4   0   0   0   0   0   0
##    5   0   0   0 177 504   1   0   0   0   0   1   0
##    6   0   0   0   0   0 540   0   0   0   0   1   0
##    7   0   0   0   2   1   0  21   1   3   0  12   1
##    8   0   0   0   0   0   0   0   9   0   1   0   2
##    9   0   0   0   1   0   0   1   0  21   0  12   0
##    10  0   0   0   0   0   0   0   0   0  16   0   7
##    11  0   0   0   0   0   0   0   0   8   0  22   0
##    12  0   0   0   0   0   0   0   0   0   8   0  17

#Accuracy of the model
1- mean(pred_lda_45$class != test_response_df$response)

## [1] 0.8636939
```

LDA on predictors determined by randomforest$importance

```
##LDA using 353 predictors determined by randomforest$importance
hapt_lda_353 = lda(response ~ ., data = train_353)

## Warning in lda.default(x, grouping, ...): variables are collinear

pred_lda_353 = predict(hapt_lda_353, newdata =  test_x_df)

#Confusion matrix for the model
table(pred_lda_353$class, test_response_df$response)

##
##        1   2   3   4   5   6   7   8   9  10  11  12
##    1 492   9   1   0   0   0   0   0   0   0   2   0
##    2   4 462  17   0   0   0   1   0   0   0   1   0
##    3   0   0 402   0   0   0   0   0   0   0   0   0
##    4   0   0   0 390  29   0   1   0   0   0   2   0
```

```
##    5    0    0    0 115  527    0    0    0    0    0    0    0
##    6    0    0    0    0    0  545    0    0    0    0    1    0
##    7    0    0    0    3    0    0   21    0    0    1    8    1
##    8    0    0    0    0    0    0    0   10    0    0    0    0
##    9    0    0    0    0    0    0    0    0   25    0    8    0
##    10   0    0    0    0    0    0    0    0    0   13    0    3
##    11   0    0    0    0    0    0    0    0    7    0   27    1
##    12   0    0    0    0    0    0    0    0    0   11    0   22
```

```
#Accuracy of the model
1- mean(pred_lda_353$class != test_response_df$response)
```

```
## [1] 0.9285262
```

QDA using 45 predictors determined by PCA

```
hapt_qda_45 = qda(response ~ ., data = train_45_pca)
```

```
Error in qda.default(x, grouping, ...) : some group is too small for 'qda'
```

QDA on predictors determined by randomforest$importance

```
hapt_qda_353 = qda(response ~ ., data = train_353)
```

```
Error in qda.default(x, grouping, ...) : some group is too small for 'qda'
```

QDA could not be performed on the training data set as some classes like 10,11,12 did not have enough data points for QDA to train the model. This is one of the drawbacks for QDA as it needs a large amount of data to train.

Constructing a cross-validated KNN model

```
knn.final.50 = rep(0,50)
for(i in 1:10){
  set.seed(i)
knn.error.50 = rep(0,50)
for(j in 1:50){
  knn.pred = knn(X_train_353, X_test_353 , train_response_df$response, k=j)
  knn.error = mean(knn.pred != test_response_df$response)
  knn.error.50[j] = knn.error
}
knn.final.50 = knn.final.50 + knn.error.50
}
knn.final.error.50 = knn.final.50/10
knn.final.error.50
```

```
##   [1] 0.1423150 0.1441809 0.1260278 0.1276091 0.1254902 0.1259962
0.1227704
##   [8] 0.1237192 0.1206515 0.1216319 0.1223276 0.1222960 0.1225806
0.1237192
##  [15] 0.1254269 0.1257748 0.1242568 0.1247628 0.1248577 0.1253953
0.1245731
##  [22] 0.1259646 0.1261227 0.1291588 0.1298545 0.1319102 0.1329222
0.1303605
##  [29] 0.1305819 0.1314991 0.1315939 0.1323213 0.1326692 0.1318469
0.1311195
##  [36] 0.1327008 0.1325743 0.1332385 0.1321948 0.1324794 0.1331752
0.1332701
##  [43] 0.1336812 0.1342505 0.1352941 0.1349462 0.1357369 0.1353257
0.1350727
## [50] 0.1372233
```

Post Cross validation- We found K = 9 gives us very good classification results and increasing k value further keeps increasing the misclassification error.

SVM using 45 predictors determined by PCA

```
hapt_svm_45 = svm(response ~ ., data = train_45_pca)

pred_svm_45 = predict(hapt_svm_45, newdata =  test.45_pca)

#Confusion matrix for the model
table(pred_svm_45, test_response_df$response)

##
## pred_svm_45   1   2   3   4   5   6   7   8   9  10  11  12
##          1 457  52   9   0   0   0   1   0   0   0   1   0
##          2   4 411  32   1   0   0   2   0   0   1   0   0
##          3  35   8 379   1   1   2   2   1   6   3   6   9
##          4   0   0   0 397  56   0   1   0   1   0   2   0
##          5   0   0   0 106 499   1   1   0   0   0   0   0
##          6   0   0   0   2   0 542   0   0   0   0   2   0
##          7   0   0   0   0   0   0  16   0   0   0   2   1
##          8   0   0   0   0   0   0   0   9   0   0   0   0
##          9   0   0   0   0   0   0   0   0  21   0  10   0
##          10  0   0   0   0   0   0   0   0   0  17   0   8
##          11  0   0   0   1   0   0   0   0   4   0  26   0
##          12  0   0   0   0   0   0   0   0   0   4   0   9

#Accuracy of the model
1- mean(pred_svm_45 != test_response_df$response)

## [1] 0.8801392
```

SVM using 353 predictors determined by randomforest$importance

```r
hapt_svm_353 = svm(response ~ ., data = train_353)

pred_svm_353 = predict(hapt_svm_353, newdata =  test_x_df)

#Confusion matrix for the model
table(pred_svm_353, test_response_df$response)

##             
## pred_svm_353   1   2   3   4   5   6   7   8   9  10  11  12
##           1  481  25  10   0   0   0   0   0   0   0   2   0
##           2    8 444  37   0   0   0   1   0   0   0   1   0
##           3    7   1 373   0   0   0   0   1   1   0   0   1
##           4    0   0   0 401  47   0   2   0   0   0   2   0
##           5    0   0   0 105 509   0   0   0   0   0   0   0
##           6    0   0   0   0   0 544   0   0   0   0   1   0
##           7    0   1   0   2   0   0  20   0   0   1   0   1
##           8    0   0   0   0   0   0   0   9   0   0   0   0
##           9    0   0   0   0   0   0   0   0  27   0  11   0
##           10   0   0   0   0   0   0   0   0   0  18   0   8
##           11   0   0   0   0   0   1   0   0   4   0  32   1
##           12   0   0   0   0   0   0   0   0   0   6   0  16

#Accuracy of the model
1- mean(pred_svm_353 != test_response_df$response)

## [1] 0.9089184
```

Neural Networks implementation

```r
#Creating dataset for Neural Netowrk

#loading training sets
training_x_nn <- read_csv("./train_353.csv")

## Parsed with column specification:
## cols(
##   .default = col_double()
## )

## See spec(...) for full column specifications.

train_x_df_nn = as.data.frame(training_x_nn)

training_response_nn <- read.table("./Y_train.txt")
train_response_df_nn = as.data.frame(training_response_nn)
names(train_response_df_nn) = "response"
train_response_df_nn$response = as.factor(train_response_df_nn$response)

rm(training_x_nn,training_response_nn)

#loading testing data sets
testing_x_nn <- read_csv("./test_353.csv")

## Parsed with column specification:
## cols(
##   .default = col_double()
## )
## See spec(...) for full column specifications.

test_x_df_nn = as.data.frame(testing_x_nn)

testing_response_nn <- read.table("./Y_test.txt")
test_response_df_nn = as.data.frame(testing_response_nn)
names(test_response_df_nn) = "response"
test_response_df_nn$response = as.factor(test_response_df_nn$response)

#removing non-transformed data
rm(testing_x_nn,testing_response_nn)

#Scaling
maxs <- apply(train_x_df_nn, 2, max)
mins <- apply(train_x_df_nn, 2, min)


#Scaling training data
scaled_train_x <- as.data.frame(scale(train_x_df_nn, center = mins, scale = m
axs - mins))
```

```r
#Scaling testing data
scaled_test_x <- as.data.frame(scale(test_x_df_nn, center = mins, scale = max
s - mins))

#Binarizing the categorical output
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "1")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "2")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "3")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "4")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "5")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "6")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "7")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "8")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "9")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "10")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "11")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "12")
names(scaled_train_x)[354:365] = c('R1','R2','R3','R4','R5','R6','R7','R8','R
9','R10','R11','R12')


Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
rm(Importance)
```

```r
#We selected the Highest values for %IncMSE and set the bar at zero.
Leaving us with 353 predictors to create a random forest model

model_params_nn = head(importance_sorted$Parameter,353)
predictors_nn <-
as.formula(paste("R1+R2+R3+R4+R5+R6+R7+R8+R9+R10+R11+R12~",
paste(model_params_nn[!model_params_nn %in%
"R1+R2+R3+R4+R5+R6+R7+R8+R9+R10+R11+R12"], collapse = " + ")))
```

```r
#Creating neural network model

#Taking a recommended value of 2/3rd of number of predictors for the number

of neurons, hence 235.


nn = neuralnet(predictors_nn,data = scaled_train_x, hidden = c(235), act.fct
= "logistic", linear.output = F)

#Predicting using the neural network model

comp = compute(nn, scaled_test_x)
pred.weights = comp$net.result
```

```r
idx = apply(pred.weights, 1, which.max)
pred = c('R1','R2','R3','R4','R5','R6','R7','R8','R9','R10','R11','R12')[idx]


test_response_df_nn$response = as.factor(test_response_df_nn$response)

test_response_df_nn$response = revalue(test_response_df_nn$response, c("1" =
"R1","2" = "R2","3" = "R3","4" = "R4","5" = "R5","6" = "R6","7" = "R7","8" =
"R8","9" = "R9","10" = "R10","11" = "R11","12" = "R12"))


#Confusion matrix for the model
table(cbind(pred), test_response_df_nn$response)

##
##        R1  R2  R3  R4  R5  R6  R7  R8  R9 R10 R11 R12
##   R1  482  21  17   0   0   0   0   0   0   0   2   0
##   R10   0   0   0   0   0   0   0   0   1  18   1   7
##   R11   0   0   0   0   0   0   1   0   6   0  31   2
##   R12   0   0   0   0   0   0   0   0   0   7   0  17
##   R2    4 447  24   0   0   1   0   0   0   0   3   0
##   R3    9   1 379   0   0   0   0   0   0   0   0   0
##   R4    1   2   0 436  25   0   0   0   0   0   1   0
##   R5    0   0   0  68 528   0   2   0   0   0   0   0
##   R6    0   0   0   1   0 544   0   0   0   0   0   0
##   R7    0   0   0   3   3   0  19   0   0   0   1   0
##   R8    0   0   0   0   0   0   0  10   0   0   0   0
##   R9    0   0   0   0   0   0   1   0  25   0  10   1

#Accuracy of the model
1- mean(cbind(pred) != test_response_df_nn$response)

## [1] 0.9285262492
```

Comparison

Classification rate for training models creating using different methods.

Methods	**Predictors obtained from PCA**	**Predictors selected from Randomforest$importance**
Random Forest	0.86211	0.9039
LDA	0.8636	0.9285
QDA	N.A.	N.A.
KNN (K=9)	N.A.	0.8793
SVM	0.8801	0.9089
Neural Networks	N.A.	0.9286

Random Forest gave the best results in almost all the problems we worked on previously, but a few other models worked better than Random Forests.

First, the unimportant predictors were removed by using the Importance function of Random Forests. The number of predictor dimensions was then reduced using PCA and lesser predictors which explained 95% of variance of the data were chosen. For all methods of classification, both PCA reduced dataset and the previous dataset (with variables reduced using Random Forest importance) were used. It was seen that, most of the times, the model built using the bigger dataset performed better than the model built using the PCA reduced dataset. We believe this is due the reduction in information caused due to reducing the variables, though they were less important.

LDA gave the second best model for this data only next to Neural Networks.

A model based on QDA could not be built because there weren't enough samples in few classes for QDA to predict its variance.

Cross-validation was done in KNN model to find out the best K value. Multiple instances of KNN model with the same K value but different seeds were run to do the cross validation. The best K value was determined to be 9.

Linear model of support vectors did not work well with the data. Support vector machines using Radial kernels worked reasonably better with an accuracy of 90%.

Neural Networks worked the best for the data with an accuracy of almost 93% on test data.

Conclusion

Based on the results from the predictions of various models, it was understood that sometimes even simpler models like LDA can model the data better and provide better predictions than complex models such as Random Forests and Support Vector Machines. So it is important to try all classification methods on the dataset before arriving at conclusions. As far as the current dataset is concerned, we believe that Neural Network performed better because it was a data collected from a time-series environment.

Neural Networks is a topic which is still open to a lot of research work. Building a Neural Network with many nodes took a lot of time and hence we experimented with round numbers of perceptrons in each hidden layer as suggested by research papers. The number of hidden layers was not increased beyond two since most published journals said that any complex function could be modeled and learned by just one hidden layer itself. We need to work more on Neural Networks to get a better understanding of the concept and intricacies and also learn the different sub-methods in Neural Networks.

References

1. Feature selection for Wearable Smartphone-Based Human Activity Recognition with Able bodied, Elderly, and Stroke Patients; Journal PLOS One; 2015
2. Activity recognition with sensors on mobile devices; Machine Learning and Cybernetics (ICMLC); 2014; published by IEEE
3. Applications and Challenges of Human Activity Recognition using Sensors in a Smart Environment
4. Qi, Min and Zhang, G. Peter. “Trend Time–Series Modeling and Forecasting With Neural Networks,” IEEE transactions on neural networks, vol. 19, No. 5, May 2008
5. Gamboa, John. “Deep Learning for Time-Series Analysis”; University of Kaiserslautern, Germany

Appendix

Code execution results:

```
#Library imports and installs
 install.packages("readr")
 install.packages("randomForest")
 install.packages("readxl")
 install.packages("MASS")
 install.packages("class")
 install.packages("boot")
 install.packages("e1071")
 install.packages("rnn")


library(readxl)

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

library(readr)

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

library(randomForest)

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

## randomForest 4.6-12

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

library(MASS)

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

library(class)

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

library(boot)

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

library("e1071")

## Warning: package 'e1071' was built under R version 3.4.3

library(rnn)

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

```r
#loading training sets
training_x <- read.table("./X_train.txt", header = FALSE)
train_x_df = as.data.frame(training_x)

training_response <- read.table("./Y_train.txt")
train_response_df = as.data.frame(training_response)
names(train_response_df) = "response"
train_response_df$response = as.factor(train_response_df$response)

#removing non-transformed data
rm(training_x)
rm(training_response)

#creating dataset of x and y combined
train_df = cbind(train_x_df,train_response_df)

#rm(X_train,train)
#write.csv(hapt_train,"train.csv")

#loading testing data sets
testing_x <- read.table("./X_test.txt", header = FALSE)
test_x_df = as.data.frame(testing_x)

testing_response <- read.table("./Y_test.txt")
test_response_df = as.data.frame(testing_response)
names(test_response_df) = "response"
test_response_df$response = as.factor(test_response_df$response)



#removing non-transformed data
rm(testing_x,testing_response)

#configuring the output to be capable of prinitng all output lines
options(max.print = 12000)

#implementing random forest on the data set
hapt_randomForest = randomForest(train_df$response ~ ., data=train_df, mtry=2
4 ,importance= TRUE)
hapt_randomForest$importance

##                 1            2             3            4            5
## V1    1.418518e-03 3.307621e-04  1.088547e-04 5.703781e-04 1.041494e-03
## V2    5.092000e-04 8.664096e-04  4.079115e-04 5.772087e-04 4.692152e-04
## V3    4.947387e-04 2.748149e-04  8.120916e-05 3.248429e-04 3.730975e-04
## V4    1.087546e-02 2.340247e-02  2.519003e-02 6.824089e-03 7.963964e-03
## V5    4.137416e-04 1.441745e-03  1.851271e-04 2.332144e-04 2.695836e-03
## V6    2.870329e-04 3.159517e-03  1.715074e-04 2.304948e-04 1.369809e-03
## V7    8.637672e-03 1.815725e-02  1.912399e-02 5.630149e-03 9.500001e-03
## V8    3.275748e-03 1.627852e-03 -1.593424e-04 4.731134e-04 3.785672e-03
## V9    3.860361e-04 2.496406e-03  8.379147e-05 1.311227e-04 8.933648e-04
```

```
## V10    9.340300e-03  3.982664e-02  6.004822e-02  2.278870e-03  2.584111e-03
## V11    1.451615e-04  1.352249e-03 -4.924758e-05  3.152898e-04  1.335887e-03
## V12    1.968744e-03  3.546239e-03  8.032149e-05  2.657338e-04  1.163588e-03
## V13    3.353687e-04  4.432771e-03  3.880958e-04  4.885073e-04  1.724920e-03
## V14    1.645826e-03  2.097103e-03  4.051870e-04  8.663609e-04  9.994150e-04
## V15    1.457694e-04  3.181657e-03  8.859521e-04  1.989959e-04  1.133169e-03
## V16    7.018195e-03  9.210533e-03  6.943290e-03  7.297995e-03  1.155271e-02
## V17    1.257111e-02  2.198163e-02  2.503737e-02  8.838920e-03  1.197058e-02
## V18    5.177969e-04  2.894695e-03  3.148702e-04  5.419919e-04  4.568659e-03
## V19    3.901202e-04  2.341551e-03  1.721116e-04  1.374439e-04  2.958019e-03
## V20    2.491018e-03  8.030319e-03  1.705494e-03  2.366289e-03  4.630749e-03
## V21    1.839721e-03  1.781907e-03  3.782809e-04  6.306916e-04  3.420078e-04
## V22    7.453780e-04  1.945313e-04  1.842938e-04  5.999779e-04  1.033070e-03
## V23    2.575052e-02  6.935574e-03  3.089480e-03  8.239193e-04  7.739481e-04
## V24    5.337224e-03  3.281795e-03  1.458061e-03  1.000130e-04  4.111956e-04
## V25    2.370918e-03  6.257415e-04  1.333148e-03  3.160952e-04  5.740311e-04
## V26    8.618502e-04  1.461771e-03  3.834857e-04  3.142725e-04  1.631607e-03
## V27    2.091117e-04  4.027816e-04  4.656784e-05  4.765715e-05  1.246293e-03
## V28   -5.707337e-06  1.062400e-04  2.904025e-05  1.038988e-04  5.840040e-05
## V29    4.377561e-05  1.769799e-04  1.023138e-03  1.201858e-04  7.653754e-05
## V30    4.872343e-05  3.300514e-04  5.448020e-05  4.836556e-04  4.537448e-04
## V31    1.498148e-04  1.302345e-04  2.076933e-05  4.164418e-04  5.058799e-04
## V32    2.617699e-05 -5.505976e-06  2.234375e-05  1.195402e-04  1.498802e-04
## V33    2.151182e-05  1.837519e-04  1.372459e-04  1.630509e-04  1.065924e-04
## V34    4.547442e-04  6.053394e-04  1.722877e-05  7.918424e-05  1.014278e-04
## V35    1.026072e-04  1.714210e-04  6.619726e-05  1.201386e-04  2.603049e-04
## V36   -4.452168e-07  8.718842e-05  2.807838e-05  7.325461e-06  5.484587e-05
## V37    9.270410e-05  5.490732e-04  3.722241e-05 -3.195554e-05  1.572807e-04
## V38    2.708694e-02  1.665392e-02  9.939112e-03  1.559086e-03  3.151744e-03
## V39    3.265006e-03  5.244054e-03  8.710263e-04  3.353811e-04  1.611256e-03
## V40    6.695789e-04  2.619030e-03  4.961790e-04  7.806695e-04  1.128227e-03
## V41    1.698546e-02  1.261013e-02  1.257410e-02  2.575146e-02  3.199695e-02
## V42    5.819236e-03  4.433877e-02  6.358841e-03  5.195773e-02  4.730826e-02
## V43    9.201376e-04  9.402474e-03  1.495046e-03  8.512982e-03  1.433293e-02
## V44    3.183775e-03  5.581974e-03  6.345587e-03  2.515001e-03  1.767972e-03
## V45    1.429346e-03  3.198196e-03  1.979043e-03  4.252590e-04  7.650422e-04
## V46    1.207510e-03  1.965824e-03  7.644458e-04  2.071580e-04  4.415013e-04
## V47    2.657670e-03  4.643650e-03  4.886504e-03  8.845052e-04  2.598660e-03
## V48    7.053838e-04  2.253212e-03  2.272411e-03  9.167518e-04  7.689213e-04
## V49    1.523936e-03  1.603987e-03  6.158715e-04  8.370457e-05  5.340773e-04
## V50    8.303447e-03  5.411623e-03  8.164527e-03  2.204234e-02  2.550277e-02
## V51    4.496280e-03  2.440671e-02  5.090443e-03  4.849477e-02  4.837083e-02
## V52    1.766946e-03  1.034797e-02  1.191518e-03  9.013100e-03  1.716986e-02
## V53    2.066319e-02  2.464986e-02  1.479959e-02  2.424717e-02  3.151832e-02
## V54    8.366793e-03  4.083709e-02  5.292383e-03  4.658815e-02  4.729043e-02
## V55    2.862751e-03  9.162511e-03  1.436904e-03  1.026946e-02  1.541118e-02
## V56    2.718033e-03  5.187289e-03  1.589964e-03  6.082473e-03  5.684345e-03
## V57    1.770574e-02  1.465102e-02  1.392988e-02  2.407934e-02  3.262498e-02
## V58    1.659440e-02  2.829681e-02  2.220369e-02  5.718904e-03  1.807912e-02
## V59    1.677843e-03  9.315048e-04  3.410395e-03  3.047840e-03  1.041879e-02
```

```
## V60   1.798272e-03  3.473216e-03  2.980578e-03  1.020003e-03  2.720602e-03
## V61   7.172524e-04  2.248493e-03  1.328419e-03  7.020857e-04  1.023054e-03
## V62   1.333927e-03  5.591249e-04  4.250686e-04  2.500918e-05  2.150958e-04
## V63   4.856005e-03  1.866570e-02  1.317580e-03  1.095113e-04  5.374566e-04
## V64   4.607166e-04  3.634700e-03  3.883048e-04  9.901534e-03  1.804644e-02
## V65   1.864594e-04  2.861544e-04  2.061614e-04  8.393487e-04  5.658102e-04
## V66   2.182754e-02  1.263314e-02  7.490225e-03  1.191138e-03  7.437772e-04
## V67   1.810723e-02  1.302140e-02  9.206892e-03  9.710817e-04  8.436029e-04
## V68   9.976290e-03  8.445054e-03  1.214305e-02  1.428068e-03  1.480829e-03
## V69   4.788442e-03  7.802809e-03  1.217596e-02  3.680400e-04  4.640091e-04
## V70   2.746791e-02  2.141388e-02  4.773024e-03  2.412936e-04  8.629352e-04
## V71   2.427925e-02  1.792143e-02  3.222589e-03  2.663735e-04  8.213491e-04
## V72   1.507159e-02  1.303215e-02  1.918296e-03  3.360547e-04  4.323002e-04
## V73   1.034793e-02  1.180832e-02  1.312136e-03  2.793531e-04  4.690193e-04
## V74   1.118073e-02  2.856430e-02  8.588051e-03  4.740705e-04  2.091145e-04
## V75   1.473302e-02  3.455462e-02  8.845779e-03  5.796730e-04  7.925617e-05
## V76   9.762037e-03  2.933669e-02  1.021160e-02  2.755781e-04  2.155269e-04
## V77   6.666584e-03  2.314589e-02  6.844848e-03  2.520432e-04  2.572325e-04
## V78  -3.153710e-06  4.501489e-04  3.848103e-04  3.440368e-03  2.411601e-03
## V79   1.521426e-04  1.043908e-03  4.220282e-04  7.157682e-04  9.972848e-04
## V80   1.740737e-04  1.356461e-04  2.555540e-04  8.112493e-04  8.326882e-04
## V81   1.153742e-04 -4.408130e-05  6.926701e-05  3.806771e-04  4.788284e-04
## V82   4.466563e-06  6.844818e-05 -2.122539e-05  3.094862e-04  1.344922e-03
## V83  -2.628061e-05  2.435188e-05 -5.644673e-06  4.167986e-04  1.237225e-03
## V84   1.022805e-02  1.116125e-02  1.683022e-02  7.484024e-03  6.639901e-03
## V85   9.301153e-03  6.218453e-03  4.946896e-03  3.885801e-03  2.489563e-03
## V86   1.723917e-03  9.074819e-04  2.574053e-03  1.631570e-04  2.251635e-03
## V87   1.258109e-02  1.454951e-02  1.849646e-02  9.662690e-03  9.664153e-03
## V88   1.491176e-02  8.540701e-03  9.058026e-03  6.587962e-03  7.719020e-03
## V89   6.392880e-03  4.253905e-03  7.679751e-03  4.577289e-03  6.762101e-03
## V90   1.210465e-03  2.958374e-03  5.430823e-02  4.948698e-04  3.323214e-04
## V91   2.849490e-03  2.518787e-04  2.161476e-03  1.395856e-04  4.451668e-04
## V92   1.870748e-03  2.820456e-03  9.793312e-03  5.485833e-04  1.177415e-03
## V93   1.739458e-03  2.675396e-03  2.051580e-03 -1.550718e-05  7.643851e-04
## V94   9.789448e-04  8.417686e-04  1.553740e-04  1.536573e-04  3.988051e-04
## V95   1.460357e-03  3.105261e-04  1.082471e-03  8.949769e-05  8.921367e-04
## V96   1.491728e-02  1.340089e-02  1.892751e-02  1.055285e-02  9.757507e-03
## V97   1.179303e-02  1.246539e-02  1.481219e-02  8.496265e-03  8.411878e-03
## V98   8.653180e-03  2.323107e-03  1.965272e-03  1.667878e-03  1.865476e-03
## V99   1.965641e-03  1.202844e-03  5.033495e-03  9.718650e-04  2.208915e-03
## V100  1.734788e-02  2.204233e-02  2.197394e-02  1.429004e-02  1.477861e-02
## V101  3.357390e-02  1.652714e-02  1.729108e-02  1.495181e-02  1.331966e-02
## V102  5.467106e-03  5.530542e-03  5.654516e-03  5.402690e-03  8.621699e-03
## V103  1.467466e-02  1.347727e-02  8.472173e-03  6.583017e-03  7.792723e-03
## V104  1.529779e-02  1.298272e-02  1.293358e-02  9.694056e-03  9.944381e-03
## V105  2.766808e-03  5.724962e-04  4.422552e-03  7.808807e-04  6.111870e-03
## V106  2.352353e-04  2.270175e-03  1.288195e-03  1.367468e-04  1.294525e-03
## V107 -7.708501e-06 -4.543812e-05  1.693840e-05  4.206015e-05  7.753791e-06
## V108 -1.703308e-05  3.467311e-04  3.686204e-04 -7.786261e-05  8.916886e-05
## V109  2.129758e-04  1.187915e-03  9.779470e-05  1.576413e-04  4.355148e-05
```

```
## V110  2.588130e-04  8.950019e-05  7.495469e-05  2.277201e-04  4.183888e-04
## V111  4.355346e-05 -2.031831e-05 -2.166641e-05  1.175908e-04  9.232188e-05
## V112  8.064674e-06  3.063161e-05  3.864025e-05  4.976016e-05  1.826864e-04
## V113  8.813312e-05  6.163916e-04  6.295779e-07  1.126994e-04  1.426875e-04
## V114  1.931063e-04  2.863220e-03  3.966693e-04  1.194029e-04  3.860517e-04
## V115  2.428628e-04  8.947467e-05  7.435211e-05  9.739478e-05  2.794362e-05
## V116  4.378485e-05  7.220810e-04  2.202750e-05  1.684719e-05  2.223385e-04
## V117  2.268285e-04  6.308938e-04  7.217693e-05  5.494339e-05  2.695673e-04
## V118  2.112098e-03  7.218464e-03  1.189542e-03  1.406266e-04  1.751470e-04
## V119  3.571589e-04  3.628045e-04  2.322770e-04  1.320173e-04  5.488191e-04
## V120  5.271508e-04  1.363821e-03  5.233117e-05  1.971798e-04  1.499828e-04
## V121  1.996777e-03  2.332864e-04  2.629998e-04  1.709165e-03  3.094966e-03
## V122  1.528480e-03  1.204385e-04  1.092249e-05  1.628221e-04  1.751109e-03
## V123  1.344222e-03  1.586380e-04  1.306571e-04  2.880217e-04  4.193621e-03
## V124  3.672054e-03  1.016906e-03  1.136065e-04  1.259406e-02  6.157551e-03
## V125  1.150172e-03  1.256127e-03  1.254819e-03  1.268330e-03  6.631716e-03
## V126  1.939156e-03  8.216710e-04  1.481000e-03  1.886868e-03  4.780303e-03
## V127  8.644999e-03  1.198954e-03  1.519917e-04  1.319807e-02  6.808641e-03
## V128  2.506415e-03  1.989247e-03  3.059287e-03  2.998045e-03  7.420748e-03
## V129  1.405055e-03  2.133894e-03  1.670336e-03  8.053973e-04  5.815596e-03
## V130  3.748862e-03  3.073328e-04  1.874634e-04  1.303171e-02  6.484492e-03
## V131  1.832603e-03  3.924319e-04  6.765648e-05  1.773880e-04  1.384193e-03
## V132  2.862606e-04  1.971849e-04  1.050282e-05  5.801416e-04  9.473655e-04
## V133  1.041563e-02  8.286813e-04  4.342709e-04  9.734400e-03  5.306081e-03
## V134  8.397945e-04  2.100248e-04  2.396509e-03  1.503624e-04  2.272369e-03
## V135  3.637813e-03  1.952149e-04  4.483556e-05  5.467474e-04  3.133708e-03
## V136  1.349811e-04  9.126497e-04  1.232685e-04  5.685455e-04  1.285027e-02
## V137  3.431867e-04  4.218392e-03  2.176242e-04  5.007652e-03  5.417458e-03
## V138  3.500087e-04  5.535349e-04  3.871664e-04  6.682981e-04  6.193591e-03
## V139  1.325606e-03  1.328231e-03  2.293068e-04  6.058701e-04  9.314187e-03
## V140  1.759610e-03  3.189147e-04  1.252890e-04  1.313472e-02  6.440912e-03
## V141  3.075937e-03  4.866908e-03  5.051410e-03  3.038180e-03  1.077869e-02
## V142  2.063873e-03  2.571515e-03  2.836447e-04  2.180841e-03  6.888371e-03
## V143  5.359951e-03 -2.512249e-06  2.505709e-04  5.267292e-03  2.468859e-03
## V144  1.862403e-04  2.125946e-04  2.117567e-03  6.464888e-04  9.615415e-04
## V145  3.927793e-03  3.929602e-04  4.288657e-04 -5.003314e-05  5.016686e-03
## V146  1.064069e-04  1.688328e-04  3.610148e-04  3.508304e-03  1.072779e-03
## V147  1.425037e-04  6.624016e-05  3.313973e-04  4.167817e-04  1.215886e-04
## V148  1.686281e-05  2.551626e-05  1.460378e-05  1.606606e-04  1.106326e-04
## V149  1.689134e-04  2.582921e-04  5.233314e-05  3.000146e-04  9.687939e-04
## V150  1.723775e-04  1.416390e-03  4.861076e-04  6.595592e-04  1.441865e-03
## V151  6.355331e-04  2.386551e-04 -2.536436e-05  9.192842e-05  1.213370e-03
## V152  3.586196e-05  6.567227e-05  1.633541e-05  2.724287e-05  3.348964e-04
## V153  1.450295e-04  2.712474e-04  7.743084e-04  1.279722e-04  2.072257e-05
## V154  3.626334e-04  2.704227e-04  9.860224e-05  1.288897e-04  5.091259e-04
## V155  2.909759e-04  6.288100e-04  1.643713e-04  8.988238e-05  7.588336e-04
## V156  2.408288e-04  3.415545e-04  2.215590e-05  1.668338e-04  3.594261e-04
## V157  1.299385e-03  1.566655e-04  2.503316e-04  1.891604e-04  1.056966e-04
## V158  3.027837e-03  5.864430e-04  1.342612e-03  5.420201e-04  1.067613e-03
## V159  2.150235e-03  6.778470e-04  1.097003e-03  1.915817e-04  4.762438e-04
```

```
## V160  1.726051e-03  5.795222e-03  1.413323e-02  8.927734e-04  1.244455e-03
## V161  3.984356e-05  2.272143e-06 -4.084301e-05  2.970138e-03  5.128081e-04
## V162  2.156455e-05 -9.307930e-05 -5.525003e-05  5.621194e-05  1.448700e-03
## V163 -1.353655e-05  1.789301e-05 -4.955590e-05  5.840241e-04  2.137581e-03
## V164  8.289485e-03  1.939099e-03  3.802853e-04  7.552270e-03  4.048252e-03
## V165  3.504317e-03  1.602790e-03  8.777278e-04  2.424342e-04  1.661786e-03
## V166  2.191756e-03  1.439159e-03  4.269620e-03  1.361607e-03  1.685151e-03
## V167  7.715597e-03  1.698893e-03  3.740641e-04  9.363764e-03  4.794750e-03
## V168  9.771957e-04  3.674458e-04  1.105554e-03  5.413041e-04  8.467743e-04
## V169  1.256593e-02  8.046061e-03  1.081415e-02  5.955015e-03  4.864840e-03
## V170  3.955541e-03  4.523900e-04  2.032039e-03  1.945862e-03  7.865129e-04
## V171  1.724691e-03  1.359030e-03  4.114473e-04  8.175460e-04  8.435274e-04
## V172  4.275556e-03  3.624158e-05  9.189650e-03  1.058250e-03  1.349372e-03
## V173  2.971298e-03  3.162608e-04  8.491156e-05  6.689770e-03  2.561226e-03
## V174  2.631147e-03  3.838382e-04  1.576370e-03  6.887399e-04  7.721476e-04
## V175  1.782063e-03  2.865476e-04  1.244493e-04  3.842548e-04  8.315588e-04
## V176  1.213144e-02  7.839080e-03  8.815031e-03  7.292221e-03  5.487928e-03
## V177  4.742299e-03  9.603450e-04 -2.616626e-05  1.123693e-02  4.323964e-03
## V178  3.877583e-03  3.854866e-04  2.757999e-04  9.812849e-04  9.901109e-04
## V179  4.475809e-03  2.953067e-03  4.331897e-03  2.271890e-03  2.330673e-03
## V180  5.364274e-03  3.501620e-03  2.814614e-03  1.504475e-02  6.242361e-03
## V181  3.046631e-03  2.365944e-03  6.905745e-03  3.401189e-03  2.517935e-03
## V182  2.459708e-02  2.014131e-02  2.502436e-02  1.663923e-02  1.153838e-02
## V183  3.394762e-04  1.324044e-03  1.129051e-03  7.534630e-03  1.807418e-03
## V184  1.181952e-04  1.571617e-05  7.810810e-04  8.431889e-04  9.019676e-04
## V185  2.748895e-03  2.985803e-03  7.538359e-03  1.701751e-03  2.018275e-03
## V186  2.102986e-04  2.103977e-04  2.087004e-04  4.469939e-03  2.481394e-03
## V187  1.832669e-05  6.660700e-05  3.207156e-05  1.817354e-04  4.082952e-04
## V188  6.045693e-05  9.126174e-05  5.531074e-05  1.725928e-03  1.834861e-03
## V189  2.406180e-06  1.219376e-04  5.367261e-06  2.050450e-04 -6.687321e-05
## V190  5.513727e-04  7.815847e-04  3.961996e-05  3.240700e-04  7.673278e-04
## V191  3.462113e-05  1.708408e-04  1.709170e-05  4.673396e-05  5.675938e-05
## V192  6.411564e-05  1.068485e-04  6.783406e-05  1.127394e-04  1.237617e-04
## V193  1.039602e-04  1.475238e-04  1.133262e-04  1.655890e-04  5.141907e-04
## V194  5.183585e-04  3.799377e-04  1.128152e-04  7.501380e-04  5.834548e-05
## V195  2.587355e-04  6.624108e-04  9.224662e-05  2.501367e-04  3.220359e-04
## V196  3.423698e-04  1.675539e-04 -3.135842e-05  1.949465e-04  1.649767e-04
## V197  2.837625e-04  1.219749e-03  1.001744e-04  1.322802e-04  2.110541e-04
## V198  7.735042e-04  3.837801e-04  5.695249e-04  9.885219e-04  1.668318e-03
## V199  8.411632e-04  1.053776e-03  6.983890e-04  1.429793e-04  4.501936e-04
## V200  3.205599e-04  2.483506e-04  1.736735e-04  4.003998e-04  1.135022e-03
## V201  3.132068e-03  6.058595e-03  3.735389e-03  3.685792e-03  1.176210e-02
## V202  1.593624e-02  8.824175e-03  6.269956e-02  1.101694e-04  1.060083e-03
## V203  1.269939e-02  4.575911e-03  4.260386e-02  5.199876e-04  1.947709e-03
## V204  1.944920e-03  8.581867e-03  1.415567e-02  1.711525e-03  5.483767e-04
## V205  1.659429e-04  6.118542e-04  2.651612e-05  1.502351e-04  4.554425e-04
## V206  3.715261e-03  7.678053e-03  4.826749e-03  4.705782e-03  1.061602e-02
## V207  3.849173e-03  4.546781e-03  4.036942e-03  1.072836e-03  4.934624e-03
## V208  3.422968e-03  9.347705e-04  1.220915e-03 -1.868039e-05  2.158628e-03
## V209  5.980613e-04  1.213238e-03  6.373744e-04  7.116474e-04  2.336581e-03
```

```
## V210  1.742274e-02  1.240281e-02  5.817770e-03  1.796638e-04  1.797083e-04
## V211  2.262340e-03  2.530765e-03  1.870274e-03  9.405704e-05  9.416324e-05
## V212  1.127461e-04  2.446464e-04  5.779538e-05  1.654185e-04  5.739882e-04
## V213 -1.465241e-05  1.270934e-03  8.190404e-05  6.129869e-05  3.483824e-04
## V214  3.654211e-03  6.030491e-03  2.935634e-03  3.020224e-03  3.710403e-03
## V215  1.493970e-02  9.448346e-03  6.147954e-02  9.687359e-05  1.406614e-04
## V216  1.174637e-02  3.932952e-03  4.289518e-02  2.444407e-04  1.826337e-03
## V217  2.970388e-03  6.302229e-03  1.544907e-02  7.833913e-04  1.549922e-03
## V218  1.094071e-03  1.073892e-03  1.399345e-04  3.925297e-04  1.770400e-03
## V219  4.086319e-03  5.591693e-03  3.082831e-03  3.339563e-03  1.280414e-02
## V220  2.367893e-03  4.805689e-03  3.416213e-03  2.807276e-03  9.694559e-03
## V221  4.495977e-03  1.200302e-03  6.094289e-04  1.473332e-04  1.707367e-03
## V222  4.899575e-04  1.178136e-03  1.963076e-03  1.091724e-03  2.600541e-03
## V223  1.898765e-02  1.170723e-02  5.849477e-03  3.965501e-06  1.766307e-04
## V224  3.204247e-03  2.845410e-03  1.934436e-03  1.644624e-04  3.381274e-04
## V225  1.216694e-04  2.233157e-04  2.278476e-05  2.101292e-07  3.439200e-04
## V226  7.647341e-05  1.047917e-03  5.933536e-05 -2.377375e-05  5.028424e-04
## V227  1.529305e-02  1.361362e-02  1.601191e-02  1.029261e-02  1.112028e-02
## V228  2.191518e-03  1.442878e-03  1.215974e-02  2.265241e-04  3.870084e-04
## V229  7.769699e-03  6.600064e-03  3.596398e-02  3.319027e-03  1.131781e-03
## V230  2.177871e-04  1.544207e-03  4.606670e-04  1.068470e-04  5.459008e-04
## V231  2.665697e-03  1.848777e-03  1.438193e-03 -1.067217e-04  1.123402e-03
## V232  1.709923e-02  1.671725e-02  1.933017e-02  1.285705e-02  1.423848e-02
## V233  1.018956e-02  7.880125e-03  1.635309e-02  6.010676e-03  6.264604e-03
## V234  1.062158e-02  8.936226e-03  2.914646e-02  5.097538e-03  5.560027e-03
## V235  1.418451e-02  1.333058e-02  2.427715e-02  9.866626e-03  1.017441e-02
## V236 -1.342672e-05  6.598150e-04  7.740037e-04  2.196060e-04  6.085086e-04
## V237 -7.440078e-05  2.843503e-03  1.236367e-04  2.683165e-05  4.026656e-04
## V238  4.007563e-05  4.588510e-05  1.494873e-05  5.494524e-05  6.647952e-06
## V239  6.033811e-05  2.183671e-05  9.301165e-05  5.481539e-05  6.122938e-05
## V240  1.679436e-04  8.662948e-04  2.715436e-05  6.450175e-04  6.500281e-03
## V241  8.159743e-04  1.715763e-04  2.184585e-04  4.190009e-04  5.366977e-04
## V242  7.336059e-04  2.640580e-04  2.357412e-04  8.988319e-04  5.328706e-04
## V243  1.519573e-04  2.972184e-04  6.223150e-05  5.738291e-04  1.027324e-03
## V244  2.046750e-04  4.391507e-04  6.065736e-05 -2.457255e-04  9.476713e-03
## V245  7.528926e-04  7.176333e-04  1.052428e-03  1.027428e-03  9.681230e-03
## V246  1.382686e-04  3.185694e-04  1.110608e-04  2.084610e-04  1.946708e-03
## V247  1.063073e-04  2.401574e-04  1.272002e-04  3.272668e-04  7.600687e-04
## V248  1.407060e-03 -6.988674e-05  4.677512e-04  7.739045e-05  6.725548e-04
## V249  1.346434e-04  4.094206e-04  6.162534e-04  2.211329e-04  2.678709e-04
## V250  2.038407e-04  5.038983e-05  3.004289e-04  2.346538e-04  3.266037e-04
## V251  1.499638e-04  1.446092e-04  7.738689e-05  1.813828e-04  2.697847e-04
## V252  2.041397e-04  5.623107e-05  7.812828e-05  7.051523e-05  2.358100e-04
## V253  4.542325e-03  1.893835e-03  2.528504e-03  2.090149e-03  2.126482e-03
## V254  3.226870e-03  7.422420e-04  1.579070e-03  4.113858e-05  9.395761e-04
## V255  2.678083e-03  5.567682e-04  1.222864e-03  8.115044e-04  5.460868e-04
## V256  4.303893e-04  1.886236e-04  8.752991e-04  1.025207e-03  5.244945e-04
## V257  2.318174e-04  7.170796e-04 -6.147266e-05  1.092201e-03  6.189943e-04
## V258  3.004131e-03  9.243004e-04  2.997095e-03  1.106745e-03  2.699562e-04
## V259  3.862485e-03  3.840507e-04  3.149953e-04  8.603878e-04  2.774679e-04
```

```
## V260  1.488022e-03  3.081424e-04  2.062189e-03  5.366586e-04  3.650537e-04
## V261  6.172275e-04  1.566041e-04  8.637207e-05  6.075060e-04  4.956320e-04
## V262  1.130571e-04  2.676581e-04  1.106211e-04  1.275654e-04  3.938749e-04
## V263  3.127880e-05  2.691763e-04  3.701458e-04  1.229468e-04  1.559579e-04
## V264  4.750289e-05  4.507979e-05  3.519963e-05  4.747951e-05  7.417158e-05
## V265  6.169769e-05  1.905156e-04  5.796743e-05  5.907803e-05  1.502539e-04
## V266  9.688758e-03  1.069840e-02  2.938324e-02  7.714421e-03  9.993853e-03
## V267  8.932734e-04  2.087670e-04  9.755841e-05  5.797464e-04  1.281659e-03
## V268  6.558548e-04  3.453356e-04  6.312861e-04  6.563274e-04  1.668834e-03
## V269  6.953254e-03  2.876732e-02  1.165025e-02  2.391672e-03  5.736554e-03
## V270  7.875950e-04  2.533336e-03  2.268886e-04  3.960068e-04  1.602269e-03
## V271  4.595548e-04  3.274436e-03  2.929584e-04  2.915153e-04  6.633628e-04
## V272  1.147356e-02  1.507589e-02  4.007928e-02  7.443325e-03  7.485646e-03
## V273  2.897299e-03  3.569495e-04  2.572126e-05  6.225374e-04  9.332953e-04
## V274  9.033003e-05  2.666682e-03  2.025558e-04  7.753206e-04  3.182076e-03
## V275  1.327006e-02  2.856573e-02  5.912426e-03  3.358075e-03  5.134018e-03
## V276  2.882321e-03  7.285814e-03  3.034277e-04  5.416359e-04  2.913613e-03
## V277  5.138087e-04  6.259542e-03  1.595097e-04  6.048018e-04  9.428857e-04
## V278  2.254092e-05  2.104173e-04  5.022727e-04  2.877464e-04  6.197001e-04
## V279  8.733616e-06  8.460313e-05  1.912104e-04  1.529554e-05  4.429493e-04
## V280  7.484406e-05  1.775645e-04  2.065476e-04  2.268852e-04  1.316643e-04
## V281  1.080253e-02  1.026597e-02  1.081678e-02  7.633880e-03  9.000376e-03
## V282  1.043528e-02  2.829840e-02  2.772419e-02  6.432367e-03  1.082991e-02
## V283  1.566697e-03  9.153359e-04  2.597144e-04  5.768114e-04  1.879685e-03
## V284  4.133744e-04  3.454941e-03  1.698494e-04  5.476082e-04  2.158579e-03
## V285  9.599584e-04  1.311770e-03  1.499476e-03  1.208174e-03  9.067507e-04
## V286  3.900587e-03  6.741607e-05 -2.101540e-06  3.566991e-04  1.946841e-04
## V287  2.138781e-03  2.529090e-04  5.809584e-03  1.172628e-04  1.341275e-03
## V288  5.953879e-03  6.765183e-03  7.644017e-03  5.192413e-03  5.364558e-03
## V289  2.534880e-03 -5.611265e-06  1.180017e-03  9.416280e-04  1.683833e-03
## V290  2.315997e-04  4.864398e-04  1.906010e-03  4.873593e-04  6.390369e-04
## V291  1.924217e-02  2.146793e-02  4.459544e-03  3.061186e-04  4.579235e-05
## V292  4.976785e-03  8.012622e-03  4.194474e-04  8.692899e-05  1.347065e-04
## V293  3.551648e-03  2.849058e-03  2.316663e-04  2.923776e-05  1.227327e-04
## V294  5.912314e-04  7.998814e-04  3.167273e-04  1.233388e-03  4.749314e-03
## V295  9.340942e-04  1.102539e-03  6.496792e-04  2.238287e-04  1.602093e-04
## V296  4.565609e-03  3.954095e-03  3.878231e-03  1.350022e-04  2.884367e-04
## V297  3.555558e-03  7.114995e-03  1.145136e-04  1.214702e-03  3.042924e-03
## V298  4.873157e-03  4.348227e-03  3.334380e-04  7.416850e-04  3.570183e-03
## V299  4.206013e-03  6.678201e-03  1.671263e-03  1.346446e-04  1.128496e-04
## V300  2.015416e-03  4.687279e-03  1.395210e-03  1.542929e-04  6.992589e-05
## V301  1.468817e-03  5.053093e-03  2.360876e-03  1.181146e-04  5.968286e-05
## V302  1.213320e-03  2.269365e-03  1.094221e-03  3.628002e-05  5.705005e-05
## V303  1.371444e-02  4.346061e-02  1.761612e-02  3.881226e-03  8.442337e-03
## V304  2.242998e-03  1.769310e-03  1.329712e-02  5.060649e-04  1.694746e-03
## V305  8.344242e-04  1.346649e-03  4.846910e-05  6.801948e-04  9.300903e-04
## V306  5.392292e-05  7.890638e-04  1.272607e-03  9.928058e-04  4.303402e-04
## V307  8.452185e-05  2.758686e-04  7.892445e-03  7.498679e-04  4.107778e-04
## V308  1.389914e-04  6.327785e-04  7.081872e-03  1.479578e-04  6.761365e-04
## V309  5.517090e-04  1.093309e-04  5.450551e-03  5.083418e-04  3.425760e-04
```

```
## V310  8.715617e-06  5.010708e-04  3.128843e-04  4.293190e-04  5.692529e-04
## V311  1.239162e-02  3.164880e-02  2.559503e-02  7.364400e-03  1.139303e-02
## V312  1.960589e-03  7.175325e-04  4.116855e-05  1.179827e-03  4.354294e-04
## V313  2.747309e-04 -2.147672e-05  7.412565e-03  3.765190e-04  2.710878e-04
## V314  3.533995e-05  7.608755e-04  8.358346e-04  7.863547e-04  2.578713e-04
## V315  1.610262e-02  2.680533e-02  2.901599e-02  9.542307e-03  1.346729e-02
## V316  1.724613e-04  3.059877e-04  1.616034e-03  3.685093e-04  2.830925e-04
## V317  4.009786e-03  6.210453e-03  1.067805e-04  4.428763e-04  1.814868e-03
## V318  5.210019e-03  4.294573e-04  2.291338e-04  1.821976e-04  5.698335e-04
## V319  4.953455e-03  1.797252e-03  7.766243e-04  4.267014e-04  5.378432e-04
## V320  1.572951e-04  4.649725e-05  1.382823e-04  2.275420e-05  6.209811e-04
## V321  3.946295e-05  2.738691e-04 -5.040249e-06  4.872729e-04  2.022869e-03
## V322  2.251714e-03  1.388745e-04 -5.762548e-05  1.272656e-04  1.482525e-03
## V323  4.932455e-04  1.999234e-04  1.071682e-03  1.882349e-04  5.482172e-04
## V324  6.987049e-04  4.084328e-05  5.106018e-04  5.703071e-06  1.802214e-03
## V325  2.670576e-04  9.900905e-04  1.512169e-04  8.098161e-04  1.390677e-03
## V326  5.684691e-03  9.706115e-04  2.734619e-04  4.862239e-04  7.421127e-04
## V327  4.120515e-04  6.460076e-04  2.204444e-05  2.096350e-04  4.046241e-04
## V328  3.057313e-04  8.362085e-05  3.510349e-05  2.872856e-04  7.728985e-04
## V329  3.483773e-03  1.802547e-03  9.684121e-05  4.625451e-04  1.106293e-03
## V330  1.026129e-03  1.705384e-04  5.520172e-04  2.621518e-04  6.389108e-04
## V331  3.339242e-04  1.014447e-02  3.009140e-04  4.322780e-04  1.432588e-03
## V332  1.220277e-03  3.908844e-04  3.730028e-04  2.757148e-04  1.498883e-03
## V333  3.864638e-03  1.369487e-03  1.298154e-02  2.049635e-04  9.242464e-04
## V334  1.383129e-03  1.403785e-04  3.398835e-03  1.722884e-04  1.003919e-03
## V335  4.055356e-04  2.917943e-04  1.572149e-03  1.452940e-04  5.081306e-04
## V336  1.345219e-03  1.346709e-04  2.226803e-03  2.434937e-04  2.676043e-04
## V337  7.002452e-05  4.523051e-06  6.774796e-03  1.054130e-03  1.308703e-03
## V338  2.311932e-04  1.863321e-04  5.508191e-06  5.804617e-04  5.455723e-04
## V339  3.250151e-04  6.019773e-03  1.500692e-04  2.664517e-04  1.020938e-03
## V340  4.485033e-03  1.211602e-03  1.960743e-02 -1.964015e-05  1.079472e-03
## V341  6.940233e-04  4.541654e-04  3.417394e-04  1.869723e-04  4.646433e-04
## V342  3.462124e-04  2.775802e-05  4.348877e-03  3.466681e-04  9.504649e-04
## V343  3.351526e-04  3.283923e-03  1.938540e-04  6.492361e-04  1.492041e-03
## V344  6.526414e-04  2.840279e-04  2.748050e-03  3.554766e-04  4.673310e-04
## V345  7.494222e-03  1.110847e-02  1.413753e-02  5.401678e-03  5.945248e-03
## V346  7.011388e-03  3.098958e-03  2.066040e-03  1.018310e-03  1.224506e-03
## V347  1.923455e-03  8.463232e-04  1.814533e-03  2.424214e-05  3.092390e-03
## V348  1.093740e-02  1.404003e-02  1.630087e-02  7.651775e-03  7.749488e-03
## V349  1.097300e-02  2.832537e-03  2.115035e-03  1.490731e-03  1.558345e-03
## V350  3.234348e-03  6.417826e-04  1.693634e-03  4.943574e-04  3.792818e-03
## V351  6.341957e-03  8.721347e-03  9.993785e-03  4.291728e-03  4.577959e-03
## V352  7.458643e-03  1.948615e-03  2.085535e-03  1.568085e-03  1.027142e-03
## V353  2.642144e-03  1.106768e-03  2.094619e-03  1.782010e-04  2.212893e-03
## V354  1.029439e-02  1.732527e-02  1.230744e-02  6.877298e-03  7.374334e-03
## V355  3.924256e-03  2.479413e-03 -2.268128e-05  1.552134e-04  1.518837e-04
## V356  3.448468e-03  2.475586e-03  2.107870e-03  1.120495e-03  2.796626e-03
## V357  8.491366e-05  3.951903e-04  9.472614e-05  9.638322e-04  2.298186e-04
## V358 -1.267404e-05  1.163329e-04  2.667935e-05  1.374870e-03  9.926366e-05
## V359  1.628053e-04  5.665704e-05  1.122685e-04  8.298316e-04  8.352996e-04
```

```
## V360  7.293461e-03  7.077739e-03  1.383386e-02  4.046737e-03  5.497030e-03
## V361  1.818063e-02  2.112686e-02  3.388068e-02  1.329498e-02  1.434609e-02
## V362  5.663773e-03  2.479742e-03  2.202966e-03  1.730418e-03  1.099728e-03
## V363  3.033184e-03  5.433435e-04  3.831170e-03  2.000266e-04  1.827656e-03
## V364  1.194610e-03  2.898395e-03  2.379623e-03  1.122786e-03  9.423741e-04
## V365  4.814272e-03  1.603388e-03  9.971215e-04  4.997122e-04  7.193740e-04
## V366  6.252546e-04  3.405251e-04  1.181105e-03  2.186683e-04  2.472115e-03
## V367  1.100366e-02  1.224853e-02  1.156327e-02  7.407644e-03  7.563610e-03
## V368  5.287174e-03  3.105296e-03  3.640036e-03  2.067633e-03  2.077837e-03
## V369  4.095822e-03  2.565600e-03  4.237497e-03  1.142415e-03  1.684676e-03
## V370  2.265689e-03  3.969388e-03  2.355116e-03  1.441619e-04  3.896160e-04
## V371  3.919512e-03  2.235168e-04  1.651038e-04  3.026243e-05  6.653965e-05
## V372  2.847401e-03  2.947743e-03  1.319274e-03  1.323846e-04  1.032180e-03
## V373  3.808999e-04  2.158119e-04  1.576355e-03  6.612554e-05  2.134516e-03
## V374  4.052779e-05  4.368516e-04  3.644388e-04  1.923162e-04  6.298003e-04
## V375  1.351569e-04  3.540215e-03  1.938432e-04  2.054189e-04  1.019185e-03
## V376  9.350634e-04  2.394889e-04  2.904010e-04  1.071012e-04  4.996678e-04
## V377  4.505192e-04  9.351405e-05  1.145626e-04  9.161753e-05  4.966092e-04
## V378  1.042236e-03  1.904367e-04  7.781067e-05  4.934411e-05  1.338107e-04
## V379  5.307369e-04  2.104893e-04  8.491330e-05  7.343042e-05  1.591541e-04
## V380  4.792352e-04  3.034003e-05  1.153790e-04  8.921537e-05  2.253184e-04
## V381  5.956315e-05  1.208626e-04  9.059478e-05  1.348323e-04  1.974212e-04
## V382  1.690437e-02  3.843075e-02  3.898581e-02  1.283135e-02  1.305268e-02
## V383  3.447543e-03  4.748544e-03  1.452830e-02  2.235146e-03  2.214638e-03
## V384  9.034010e-04  1.413569e-03  8.620475e-05  6.269037e-05  7.401311e-04
## V385  2.271249e-04  4.674334e-04  1.830743e-03  6.212365e-04  4.740254e-04
## V386  4.992435e-05  6.728700e-04  4.010544e-03  3.391567e-04  3.050991e-06
## V387  7.632278e-05  1.089649e-03  2.531683e-03  5.423392e-04  5.170997e-04
## V388  7.316276e-04  1.790786e-03  1.046999e-03  3.584623e-04  3.799627e-04
## V389  1.025136e-03  2.971272e-04  1.033214e-04  3.200750e-04  4.003330e-04
## V390  2.045026e-02  3.083516e-02  3.406463e-02  1.436398e-02  1.554548e-02
## V391  3.825690e-03  2.678866e-03  2.086431e-03  1.901921e-03  1.101115e-03
## V392  1.119311e-04  2.199788e-04  9.185376e-03  8.654745e-05  6.639615e-05
## V393  2.108038e-05  1.040772e-03  1.544460e-03  3.858877e-04  1.733325e-04
## V394  1.315710e-02  1.778278e-02  2.120115e-02  9.656581e-03  9.895317e-03
## V395  2.703014e-04  5.929903e-04  3.385093e-03  1.468351e-04  3.256020e-04
## V396  9.739284e-04  2.443420e-03  1.545299e-03  1.016783e-03  1.413906e-03
## V397  3.260661e-03  1.208356e-03  6.219104e-05  8.628547e-04  1.997243e-04
## V398  6.710652e-03  1.121583e-03  8.831360e-04  6.226229e-04  2.948724e-04
## V399  5.052376e-04  1.722231e-04  7.743741e-05  4.192369e-05  4.875172e-04
## V400  3.339281e-04  3.594514e-04  1.155656e-04  7.981575e-04  2.653326e-04
## V401  4.829846e-03 -4.454206e-05  1.161857e-03  6.324967e-04  5.406892e-04
## V402  2.875444e-03  9.743137e-05  4.837948e-05  1.440783e-04  4.096625e-04
## V403  2.414025e-04  2.798243e-04  3.843114e-04  1.087499e-03 -3.380450e-05
## V404  6.944746e-03  5.231483e-03  4.664068e-03  4.049158e-03  3.735466e-03
## V405  4.162067e-03  6.349333e-04  2.227947e-04  4.134212e-04  9.188284e-04
## V406  1.756790e-03  5.038350e-05  8.881730e-04  1.117142e-03  3.123718e-04
## V407  1.239352e-03  2.305883e-05  9.991029e-05  9.444379e-04  1.593215e-04
## V408  8.842876e-03  1.731080e-03  1.327285e-03  4.147873e-04  8.944705e-04
## V409  3.766223e-03  4.761608e-04  7.788061e-04  5.927037e-04  2.643660e-04
```

```
## V410  3.080067e-03  1.303281e-02  1.149689e-03  1.455670e-03  6.565520e-03
## V411  1.550463e-03  8.152997e-04  1.751277e-03  2.520296e-04  2.089385e-03
## V412  5.115257e-03  2.100825e-03  1.909086e-02  6.115200e-04  2.196876e-03
## V413  5.062956e-03  2.447206e-04  5.877965e-03  4.318146e-04  7.420340e-04
## V414  6.807028e-04  8.699106e-04  3.399925e-04  5.719485e-04  5.664953e-04
## V415  4.078244e-04 -2.694941e-05  1.311114e-03  1.432937e-04  2.896754e-03
## V416  5.269961e-04 -9.195107e-05  7.290100e-03  2.240454e-03  5.016593e-04
## V417  7.372410e-04 -4.650979e-06  3.451373e-04  1.201592e-04  1.391448e-03
## V418  4.315555e-04  2.067951e-03  3.619354e-04  1.309231e-05  4.516133e-03
## V419  4.648081e-03  2.320875e-03  2.448621e-02  8.178007e-05  1.826099e-03
## V420  1.016421e-03  8.291514e-04  1.607208e-03  1.434291e-04  2.148415e-03
## V421  6.078254e-04  5.404454e-05  6.237808e-03  8.656635e-04  3.329766e-04
## V422  1.917597e-03  1.497610e-03  4.021721e-03  1.263002e-03  3.358361e-03
## V423  1.780279e-03  5.625247e-04  5.522503e-03  1.265355e-04  1.450865e-03
## V424  1.186929e-03  4.425649e-04  1.153770e-04  1.102428e-02  5.199318e-03
## V425  1.313861e-03  4.023541e-04  1.535087e-04  3.838766e-04  2.022821e-03
## V426  1.255492e-03  3.362861e-04  4.607818e-05  1.043641e-03  1.168099e-03
## V427  6.138493e-03  8.824317e-04  2.314040e-04  1.359684e-02  6.577864e-03
## V428  4.650239e-04  5.454608e-04  3.261019e-04  7.220413e-04  4.312146e-03
## V429  2.121988e-03  1.240314e-03  1.126611e-03  8.857890e-04  3.721384e-03
## V430  2.020275e-03  8.577743e-04  3.835762e-04  1.790074e-02  6.661093e-03
## V431  1.395583e-03  1.556996e-03  1.094066e-03  1.454756e-03  3.826003e-03
## V432  2.471869e-04  2.742214e-04  5.033149e-04  1.948068e-03  3.435804e-03
## V433  6.572888e-03  8.354395e-04  6.831647e-05  1.114668e-02  7.240510e-03
## V434  4.737039e-04  1.414723e-03  6.854738e-04  5.927455e-04  7.003235e-03
## V435  4.117280e-03  1.985399e-03  4.420329e-04  4.578305e-04  6.206385e-03
## V436  6.709265e-05  2.972563e-04 -4.689893e-06  1.031486e-03  2.037425e-04
## V437  2.650091e-04  2.557127e-04  7.524837e-04  1.214292e-04  2.465528e-04
## V438  7.496361e-08  2.126916e-05  4.496131e-05  2.421881e-05  1.894412e-03
## V439  4.931173e-03  5.541431e-03  5.643442e-03  6.195121e-03  4.011434e-03
## V440  2.397510e-03  7.471987e-04  2.358683e-04  1.377126e-02  6.112954e-03
## V441  1.585207e-04  4.083138e-04  1.665170e-04  6.251556e-04  4.846937e-03
## V442  3.512673e-04  7.025296e-04  2.130860e-04  9.335013e-04  2.528675e-03
## V443  1.024935e-03  8.816432e-04 -7.136448e-05  3.932042e-03  5.566137e-04
## V444  1.450868e-03  1.295149e-04  4.474982e-04  1.314369e-04  1.806695e-03
## V445  1.528107e-04  7.020353e-05  3.634023e-03  6.364696e-04  1.257650e-03
## V446  6.310063e-04  3.913331e-04 -1.490469e-05  7.187268e-03  3.954934e-03
## V447  1.544125e-04  3.667920e-04  2.100595e-04  2.555025e-04  1.387960e-03
## V448  1.762773e-03  1.523585e-03  1.052933e-03  1.225261e-03  1.258235e-03
## V449  3.709110e-02  8.399150e-04  9.025190e-04  1.381873e-04  2.488130e-04
## V450  1.307843e-02  3.373167e-03  3.840641e-04  4.643601e-05  8.523799e-05
## V451  2.401648e-02  1.694891e-02  2.125636e-03  1.361299e-04 -9.339443e-06
## V452  1.701986e-03  6.193554e-04  1.433396e-04  8.093913e-03  4.939069e-03
## V453  9.697362e-04  6.170307e-04  9.674413e-04  1.434698e-04  2.713482e-04
## V454  1.012600e-03  6.513830e-04  1.982923e-03  3.413302e-04  1.615651e-04
## V455  5.057801e-04  1.105075e-04  5.078790e-05  5.260565e-04  9.632718e-04
## V456  6.149271e-04  9.909530e-05  1.871236e-05  3.770046e-04  4.634136e-04
## V457  1.484040e-03  6.196072e-04  4.349136e-03  3.078022e-04  8.964236e-04
## V458  1.689314e-03  7.721794e-04  5.165764e-03  1.856073e-04  6.712161e-04
## V459  2.161128e-03  1.868284e-03  3.285675e-03  1.413687e-04  1.791964e-04
```

```
## V460  1.463406e-03  2.089145e-03  2.429722e-03  1.971641e-04  2.371195e-04
## V461  1.573724e-02  8.090946e-05  4.596330e-05  1.297556e-02  7.529275e-03
## V462  5.586103e-03  2.221010e-03  1.862677e-04  1.143689e-02  1.748955e-03
## V463  1.410650e-03  4.859464e-04  2.302620e-05  2.041681e-03  1.709316e-04
## V464  1.610141e-03  4.472453e-04 -6.789333e-05  1.422088e-03  8.368331e-04
## V465  3.603755e-04  2.832702e-04 -3.175112e-05  2.248770e-03  1.123756e-03
## V466  2.289968e-03  6.284084e-05  1.549777e-06  2.912530e-03  1.020875e-03
## V467  6.696638e-04  4.192710e-05 -9.444608e-05  4.191808e-03  7.161844e-04
## V468  5.321715e-04  9.600556e-04 -5.835910e-06  1.844439e-03  2.112851e-04
## V469  2.715630e-03  1.695574e-03  3.567390e-04  1.583048e-02  7.568322e-03
## V470  1.646309e-03  6.003992e-04 -5.342839e-05  2.135572e-03  5.912162e-04
## V471  7.818705e-04  2.216173e-04 -7.782742e-05  2.677850e-03  9.714675e-04
## V472  2.547585e-04  9.814169e-05 -2.292144e-05  4.281784e-03  7.873622e-04
## V473  8.176140e-04  5.710214e-04  4.972255e-04  1.496279e-02  6.815797e-03
## V474  1.340154e-03  9.857502e-05  1.274810e-05  2.730379e-03  1.157429e-03
## V475  1.675204e-04  8.194884e-04  5.075789e-04  6.698076e-04  6.142084e-03
## V476  2.163031e-03  7.356767e-04  3.822284e-03  1.815599e-04  7.862013e-04
## V477  4.541629e-04  5.852870e-04  1.464582e-03  3.319268e-04  1.907587e-03
## V478  8.400446e-05  4.812369e-05  2.971776e-04  4.498544e-04  1.147452e-03
## V479  1.341857e-04  9.733381e-04  2.904346e-04  8.883221e-04  2.718171e-04
## V480  1.328419e-03  5.380729e-04  2.251027e-04  1.995830e-04  6.494500e-04
## V481  2.245865e-05  4.643249e-06  2.535134e-04  3.439129e-04  1.152253e-03
## V482  1.784882e-05  3.821710e-04  4.173939e-03 -4.741635e-05  9.842940e-04
## V483  2.371993e-04  1.225680e-03  5.649798e-04  1.098280e-03  3.529558e-03
## V484  1.956712e-03  6.000264e-04  5.664555e-05  5.621521e-04  1.563098e-03
## V485  9.050347e-04  2.372780e-03  1.222843e-04  6.755353e-04  7.170203e-04
## V486  3.200438e-04  5.360868e-04  3.310670e-05  5.427326e-05  1.092268e-03
## V487  2.013286e-04  2.701549e-04  3.523839e-04  9.062244e-04  4.266454e-03
## V488  1.501429e-03  6.674490e-04  3.889532e-04  1.061596e-03  1.167724e-03
## V489  1.147691e-03  1.171431e-03  7.658075e-04  1.591374e-03  4.343863e-03
## V490  1.659598e-04  1.881794e-04  3.210879e-03  9.316693e-04  1.374101e-04
## V491  1.470148e-03  5.356161e-04  2.031855e-03  8.165126e-04  1.521359e-03
## V492  7.514820e-05  1.149124e-04  1.578798e-03  8.348229e-04  1.812144e-03
## V493  1.719667e-04  2.668010e-04 -3.846315e-05  5.759095e-04  2.190303e-03
## V494  4.448634e-04  1.108947e-04  4.031485e-05  3.853439e-04  1.399649e-03
## V495  7.343997e-04  2.808515e-04  1.252948e-06  3.768903e-04  1.203210e-03
## V496  1.288866e-05  1.475139e-04  8.990844e-05  7.242502e-04  1.818287e-03
## V497  1.017468e-03  8.202208e-04  8.774837e-05  1.314212e-03  2.522522e-03
## V498  2.364438e-03  1.443919e-03  4.363837e-03  2.076974e-03  2.236977e-03
## V499  8.992470e-05  3.044527e-04 -8.346961e-05  9.152941e-04  1.319619e-03
## V500  9.541905e-04  2.292068e-04 -3.384427e-05  1.509832e-04  1.706078e-03
## V501  1.170014e-03  1.292239e-03  8.127609e-04  9.690894e-04  2.425155e-03
## V502  1.344336e-05  3.062502e-04  1.329970e-03  7.525881e-04  1.424351e-03
## V503  3.114170e-03  4.619405e-03  3.200907e-02  7.494677e-05  4.487988e-04
## V504  2.549010e-02  1.549112e-02  6.684721e-02  1.215579e-04  3.345966e-04
## V505  1.514244e-02  5.774544e-03  5.263940e-02  7.270406e-05  1.596569e-03
## V506  8.529337e-03  1.307659e-02  7.788312e-03  1.430032e-04  1.165426e-03
## V507  1.427434e-04  5.188594e-05  3.743627e-04  9.804178e-07  3.598753e-04
## V508  3.780796e-03  5.024565e-03  3.095835e-02  3.043980e-05  6.166320e-04
## V509  1.307626e-02  9.018517e-03  7.287718e-02  1.617167e-04  7.038875e-04
```

```
## V510  1.097886e-03  2.146983e-03  7.285783e-03  5.918456e-05  3.333411e-04
## V511  4.374729e-04  1.038123e-03  5.517625e-03  1.346226e-04  8.833914e-04
## V512  1.442039e-03  1.788921e-02  6.012435e-03  1.304622e-04  1.386845e-04
## V513  8.934079e-03  1.710098e-03  3.076210e-03  1.241216e-04  5.898474e-05
## V514  2.083603e-03  1.354843e-03  4.153234e-04  9.487731e-05 -2.224924e-05
## V515  1.394904e-03  1.006724e-03  2.952062e-04  1.376558e-04  7.092753e-05
## V516  6.686593e-04  1.125456e-03  6.823403e-03  5.626019e-05  1.464929e-03
## V517  7.618729e-03  1.786561e-03  2.675125e-02  1.010036e-04  9.933705e-04
## V518  5.315450e-04  1.754218e-03  2.996518e-03  2.821548e-04  7.787602e-04
## V519  1.569243e-02  1.708410e-03  1.125146e-02  2.523953e-05  8.100776e-05
## V520  7.178761e-05  4.232115e-04  7.738136e-04  2.085949e-04  3.628719e-04
## V521  4.241281e-04  7.149237e-04  4.741890e-03  9.255505e-05  1.843075e-04
## V522  1.110065e-03  9.914117e-04  1.284387e-02  2.340995e-05  3.737469e-04
## V523  2.009005e-03  3.087361e-03  2.327322e-03  2.480459e-03  1.137013e-03
## V524  2.941614e-04  1.411685e-03  1.961324e-04  1.081625e-04  3.618015e-04
## V525  2.141006e-03  5.388005e-03  1.925692e-03  6.457476e-05  9.341735e-05
## V526  4.834896e-05  1.231949e-04  6.288313e-04  6.139638e-05  3.505783e-04
## V527  3.779374e-03  5.460964e-04  6.085820e-04  9.235857e-05  1.223247e-04
## V528  2.543766e-03  3.784455e-04  5.134861e-04  1.729944e-04  7.952657e-05
## V529  1.383524e-04  1.033300e-04  7.236175e-05  2.405720e-04  1.096516e-03
## V530  2.647431e-04  2.576396e-04  4.744975e-04  5.336066e-04  4.318525e-04
## V531  8.777293e-05  2.330877e-04  6.484357e-04  4.938337e-04  5.638299e-04
## V532  2.362235e-04  8.583266e-04 -4.771909e-05  3.237770e-04  1.230725e-03
## V533  1.352390e-04  1.722553e-04  3.810437e-05  2.144744e-04  8.735896e-04
## V534  8.727794e-04  5.232303e-05  7.617262e-05  1.197354e-03  5.149063e-04
## V535  5.977755e-05  4.717239e-04  9.125219e-05  2.658129e-04  1.532073e-03
## V536  5.880336e-05  2.663610e-04  7.234505e-05  7.518582e-04  1.183013e-03
## V537  1.410629e-03  5.651762e-04  2.854201e-04  5.482527e-04  7.272974e-04
## V538  8.804722e-03  1.899350e-04  5.044165e-05  4.554885e-05  1.456566e-04
## V539  9.725697e-03  1.904307e-03  4.732387e-04  7.328734e-05  3.701526e-04
## V540  1.637695e-04  3.322477e-04  1.996905e-04  3.178854e-04  2.880425e-04
## V541  1.800679e-04  1.484229e-04  1.113037e-04  1.694872e-04  1.051717e-04
## V542  8.403499e-05  3.239212e-04  4.099795e-04  3.880390e-04  5.242576e-04
## V543  2.898067e-03  1.341217e-03  1.876007e-03  9.047910e-04  6.786390e-04
## V544  7.176469e-04  1.861431e-04  1.217568e-03  2.963712e-04  7.782176e-04
## V545  3.183429e-03  1.382508e-03  9.947247e-04  4.415196e-04  8.407691e-04
## V546  5.081434e-05  2.205116e-04  2.070782e-04  7.144878e-04  1.938075e-04
## V547  1.357263e-03  2.881547e-04  1.747181e-03  7.826284e-04  4.508243e-04
## V548  1.501293e-03  1.300089e-04  2.773272e-03  3.084004e-04  4.136553e-04
## V549  8.852671e-04  3.215288e-04  9.490037e-04  1.859086e-04  2.818007e-04
## V550  1.419537e-03  5.442744e-05  2.025264e-03  2.641530e-04  2.275189e-04
## V551  3.147390e-03  1.273988e-04  3.179602e-04 -1.221085e-07  1.018119e-04
## V552  5.815863e-05  4.735890e-04  3.399542e-05  1.111380e-06  4.591216e-04
## V553  2.196109e-03  4.518840e-05  5.038632e-04 -2.732758e-05  3.650416e-04
## V554  2.562927e-03  1.803283e-04  5.114873e-04  1.584626e-05  4.622108e-04
## V555 -1.776754e-05 -3.350197e-05  1.234423e-05  1.267773e-04  3.569251e-04
## V556 -3.106341e-05 -4.026012e-07 -2.301067e-05  2.222417e-04  1.474087e-04
## V557  2.383538e-04  5.762079e-05  4.846677e-05  3.907769e-03  1.935662e-04
## V558 -1.211394e-06 -5.662706e-06 -5.028606e-05  1.558218e-03  9.227248e-05
## V559  1.270922e-02  1.743615e-02  1.347767e-02  2.654129e-02  3.219741e-02
```

```
## V560  6.551585e-03  4.272627e-02  5.148931e-03  4.626045e-02  4.565201e-02
## V561  3.096383e-03  7.911331e-03  1.107381e-03  1.093075e-02  1.855232e-02
##                  6             7             8             9            10
## V1    6.530948e-04 -2.536890e-03 -1.362626e-03  2.186686e-02  3.366526e-02
## V2    3.415908e-05  9.279550e-04  2.043616e-02  2.809068e-02  1.280416e-02
## V3    9.171154e-05  2.384804e-03  4.791342e-03  2.572458e-03  3.539729e-03
## V4    4.963631e-03  6.120934e-03  5.975758e-03  4.070827e-03  3.299302e-03
## V5    1.679617e-04  3.089908e-03  2.247619e-03  5.696184e-04 -1.788212e-05
## V6    3.916339e-04  6.878487e-05  8.333333e-04  2.953619e-05 -2.033494e-03
## V7    4.627805e-03  4.372483e-03  5.719986e-03  2.320953e-03  3.251721e-03
## V8    4.481123e-04  9.923768e-04  3.819048e-03  9.348316e-04  4.704786e-04
## V9    3.342292e-05 -2.005013e-04  1.923077e-04  1.910446e-04 -1.186286e-03
## V10   2.339382e-02  7.192640e-03  9.537124e-03  2.219270e-02  2.427039e-02
## V11   5.246105e-04  9.668431e-03  8.474359e-03  2.698253e-02  4.635275e-03
## V12   2.499355e-04  8.311474e-04  1.959596e-03  4.227090e-03  2.202367e-04
## V13   6.696822e-05  1.554785e-03  2.520901e-03  1.255990e-02  5.542070e-03
## V14   4.460648e-04  1.249573e-03  1.474246e-02  5.746806e-03  5.305388e-03
## V15   5.321678e-04  4.220355e-04  3.515152e-04  7.197271e-04 -2.674542e-04
## V16   3.615259e-04  1.668782e-03  9.024603e-03  7.167042e-03  5.678757e-03
## V17   9.569807e-03  4.461165e-03  4.001587e-03  5.391235e-03  1.376167e-02
## V18   7.580625e-04  3.287247e-03  4.355384e-03  3.291656e-03  1.296292e-03
## V19   5.368226e-04 -1.790175e-04  6.944444e-04  9.198616e-05 -3.047200e-04
## V20   2.371619e-03  5.288101e-04  1.701948e-03  2.002564e-03  2.590948e-03
## V21   1.818410e-04  1.789964e-03 -1.292929e-04  1.378250e-03  2.674825e-04
## V22   2.444514e-04 -3.069054e-05  1.337302e-03  1.481406e-03 -5.662393e-04
## V23   5.214262e-04  1.034536e-03  1.458275e-03  5.976781e-03  2.510932e-02
## V24   3.502761e-04 -5.674523e-04  7.353053e-03  1.355008e-02  2.866218e-03
## V25   2.661793e-05  1.977716e-04  1.806227e-03  7.272522e-03  2.720380e-03
## V26   1.440920e-04  9.837141e-04  1.515629e-03  6.794151e-04  5.361472e-04
## V27  -7.754851e-06  1.250000e-04  3.636364e-04  4.882772e-04 -1.717405e-04
## V28   3.507899e-05  8.690476e-05  1.666667e-04  5.297691e-03  1.792989e-04
## V29   2.707271e-05  3.335422e-04  5.229437e-04  8.641166e-04  1.105401e-04
## V30   2.206856e-04 -1.710526e-04  7.763126e-04  9.836460e-05 -1.551875e-05
## V31   1.204004e-04  1.250000e-04 -6.818182e-05 -9.491819e-05 -1.167184e-04
## V32   4.174835e-06 -5.019608e-04 -4.722222e-04  3.014880e-04 -1.520580e-04
## V33   1.854543e-05 -3.356643e-04  0.000000e+00  9.459903e-04  6.052568e-04
## V34   1.695823e-05  6.969298e-04 -3.666667e-04  1.238590e-05 -1.113322e-03
## V35   4.779656e-05  2.444444e-04  0.000000e+00 -7.926414e-05 -1.001693e-03
## V36   3.782938e-06  4.326923e-04  6.285714e-04  1.154367e-03 -9.865994e-04
## V37  -1.581176e-05  6.535948e-06 -4.500000e-04  4.096644e-04 -2.989130e-04
## V38   7.861876e-04  4.500390e-03  2.494322e-03  8.860329e-03  5.731257e-04
## V39   6.960729e-05  1.792488e-03  3.666667e-04  1.226064e-04  7.386160e-04
## V40   3.206040e-03  4.061651e-03  5.280902e-03  5.914626e-03  1.763858e-03
## V41   1.482957e-01  1.076459e-02  1.065863e-02  1.072349e-02  1.302153e-02
## V42   2.434612e-02  1.660955e-02  1.024890e-02  1.826187e-02  2.048828e-02
## V43   1.269395e-02  2.343675e-03  3.713997e-03  9.276404e-04  4.855012e-03
## V44   8.600104e-04  5.213194e-03  2.623088e-03  6.550395e-03  1.095178e-02
## V45   1.726463e-04  5.330363e-03  1.440142e-02  1.164615e-02  2.748300e-03
## V46   8.563387e-05  1.426835e-03  2.696082e-03 -3.911621e-05  4.041062e-04
## V47   6.377386e-04  6.908708e-03  1.800000e-03  8.515486e-03  1.024450e-02
```

```
## V48   6.049293e-04  1.760661e-03  7.311339e-03  7.793701e-03  3.874753e-03
## V49   1.140662e-04  1.209213e-03  2.013869e-03  3.199304e-03 -2.397023e-05
## V50   1.040573e-01  4.892674e-03  8.538095e-03  5.904961e-03  2.571093e-03
## V51   2.322253e-02  8.018149e-03  2.009504e-02  1.947441e-02  1.813932e-02
## V52   1.283075e-02  2.663658e-03  2.484848e-04  2.409073e-04  4.186175e-03
## V53   1.319914e-01  1.051701e-02  1.138020e-02  1.072744e-02  4.341115e-02
## V54   2.242750e-02  1.639732e-02  1.383030e-02  2.723556e-02  9.230899e-03
## V55   1.002433e-02  2.177247e-03  2.578644e-03  2.468596e-03  5.639101e-03
## V56   2.061845e-03  2.971622e-05  1.461977e-03 -1.013911e-03  2.477656e-03
## V57   1.481576e-01  8.434874e-03  1.153554e-02  9.551331e-03  7.255137e-03
## V58   2.816053e-02  2.486168e-03  2.372150e-03  2.090608e-03  1.125185e-02
## V59   1.513301e-02  2.458750e-03 -2.515873e-04  6.544452e-04  2.089125e-03
## V60   4.555696e-04  8.976626e-03  1.534199e-03  4.626524e-03  7.471979e-03
## V61   4.176860e-04  1.529146e-03  1.495079e-02  6.504300e-03  1.792922e-03
## V62   8.244100e-05  1.296104e-03  1.337229e-03  8.897645e-04  1.444444e-04
## V63   1.789898e-04  4.631104e-03  4.216667e-03  5.454806e-03  6.700213e-03
## V64   1.905744e-03  3.181297e-03  2.019695e-02  2.033684e-02  1.134975e-02
## V65   3.253331e-04 -8.230159e-04 -6.401099e-04  1.111085e-03  8.253402e-05
## V66   3.226001e-04  3.643555e-03  6.592075e-03  1.747146e-02  1.010353e-02
## V67   5.428069e-04  4.719908e-03  4.405456e-03  1.332094e-02  9.501411e-03
## V68   4.514383e-04  2.542739e-03  8.821789e-04  8.686289e-03  8.273728e-03
## V69   1.663100e-04  3.155336e-04  1.820707e-03  9.263350e-03  3.794636e-03
## V70   2.786955e-04  1.344975e-02  1.279605e-02  1.184514e-02  4.865933e-03
## V71   3.404119e-04  8.001252e-03  4.242857e-03  8.175222e-03  5.831219e-03
## V72   1.683661e-04  6.846709e-03  4.809374e-03  7.085131e-03  2.780186e-03
## V73   7.409126e-05  3.401459e-03  3.811927e-03  5.368274e-03  1.143781e-03
## V74   4.526640e-04  1.552405e-02  1.391926e-02  1.005828e-02  4.926853e-03
## V75   2.849791e-04  2.135475e-02  1.391508e-02  1.316545e-02  7.974326e-03
## V76   2.846107e-04  1.534464e-02  6.437107e-03  8.707607e-03  3.461455e-03
## V77   4.085686e-05  1.609473e-02  6.231879e-03  6.516002e-03  3.478290e-03
## V78   4.190487e-04  6.407607e-03  3.708009e-03  2.011046e-02  1.957585e-04
## V79   1.940753e-04  1.361298e-03  1.166667e-04  9.464930e-04  2.063456e-03
## V80   1.074586e-03  9.953907e-04  1.866667e-03 -2.815700e-04  1.662082e-04
## V81   9.318769e-05 -3.694444e-04  6.515152e-04  1.201712e-04  1.680800e-04
## V82   2.423037e-04  5.275846e-03  8.722222e-04 -4.064228e-04 -1.172517e-03
## V83   2.344165e-05  1.852430e-03  3.000000e-04  2.815208e-04 -9.631649e-05
## V84   7.256948e-03  2.093080e-03  3.765873e-03  3.753636e-03  1.844324e-03
## V85   3.760023e-03 -4.794872e-04 -5.914141e-04  1.156865e-03  6.507712e-04
## V86   6.522844e-05  1.058362e-03  9.079365e-04  8.359370e-04 -3.343541e-04
## V87   9.537696e-03  4.310817e-04  2.811833e-03  2.923858e-03  2.001036e-03
## V88   4.969232e-03  2.299254e-03  2.393218e-04  2.310516e-03  3.555294e-03
## V89   1.439552e-03  1.356098e-03 -2.785714e-04  1.447095e-03 -4.879490e-04
## V90   1.922800e-05  9.123260e-04  5.023810e-04  3.051392e-03  2.127071e-03
## V91   9.964514e-05  4.464286e-04  5.467033e-04 -3.902480e-04 -5.068861e-04
## V92   1.050209e-03  6.792929e-05  4.040404e-05  3.374470e-04  1.585584e-03
## V93   3.779334e-05  4.133675e-04  2.857143e-05  1.138835e-03  4.542436e-04
## V94  -2.333123e-05  2.176471e-04  2.500000e-04 -5.561812e-05  2.966667e-04
## V95  -1.174657e-07  1.469298e-04  3.333333e-04  3.357854e-04 -7.608696e-05
## V96   9.769568e-03  1.340321e-03  3.296753e-03  5.218683e-03  5.637070e-03
## V97   9.548970e-03 -1.953166e-03  1.505556e-03  2.171455e-03  2.928282e-03
```

```
## V98    1.637116e-03  9.511612e-04  1.002381e-03  6.951966e-04 -8.944932e-05
## V99    5.431176e-04  4.096605e-04  6.833333e-04 -4.009961e-04 -1.029813e-04
## V100   1.310022e-02  6.356413e-04  4.037707e-03  4.949262e-03  4.089007e-03
## V101   1.393968e-02  2.210113e-03  2.566545e-03  6.626711e-03  3.820826e-03
## V102   1.963240e-03  1.812083e-03  1.069697e-03  1.695456e-03  2.046995e-04
## V103   6.835960e-03  1.714154e-03  4.576984e-03  1.091069e-03  1.205667e-03
## V104   7.228130e-03  4.870428e-03  2.463492e-03  1.913356e-03  2.846666e-03
## V105   3.386408e-04  3.881763e-04  6.944444e-04  1.750323e-03  5.915826e-04
## V106   4.362230e-05  3.827846e-04 -2.222222e-05 -1.952544e-04  0.000000e+00
## V107  -3.824092e-06 -5.860806e-05 -4.000000e-04 -3.838423e-04 -2.797619e-04
## V108   8.246167e-06 -2.500000e-04  0.000000e+00  3.521330e-04  1.348230e-04
## V109  -7.728973e-06  1.829940e-04 -2.222222e-05  6.951170e-04  1.914321e-03
## V110   4.232874e-05 -2.222222e-04  2.777778e-05  1.266053e-05 -7.476375e-04
## V111   2.224531e-05 -9.523810e-05  0.000000e+00 -2.159294e-04 -5.193359e-04
## V112  -7.933315e-07  1.031746e-04  6.944444e-04  7.524699e-04 -1.808696e-04
## V113   3.509811e-06  2.202381e-04 -4.000000e-04  1.350467e-03  4.286843e-04
## V114   4.178346e-06  4.126264e-04 -4.285714e-04 -4.740828e-04 -1.014922e-03
## V115   2.235909e-05 -1.285714e-04  2.000000e-04  5.927765e-04  3.834025e-04
## V116   4.605606e-05  5.582638e-04  1.396825e-03 -1.125926e-04 -3.323161e-04
## V117   6.829624e-05 -4.886364e-04  6.666667e-04  4.760532e-04  4.449681e-04
## V118   1.826906e-05  2.046499e-04  1.150794e-03  4.573221e-04 -5.483176e-04
## V119  -3.339738e-06  2.805556e-04  1.055556e-03 -1.802688e-04  2.583851e-04
## V120   5.406184e-05 -3.444444e-04  7.252747e-04 -1.843190e-05 -7.575758e-06
## V121   8.094850e-05  7.343750e-04  1.396898e-03  4.992872e-03  2.169688e-03
## V122   7.426963e-05  7.006880e-04  7.380952e-04 -4.856310e-04 -2.731037e-04
## V123   3.822883e-04  3.695703e-03  3.260101e-03  4.418327e-03  3.653923e-03
## V124   1.407913e-04  1.738688e-03  7.191292e-03  5.352707e-03  1.978485e-03
## V125   6.878936e-03  6.914787e-04  1.414114e-03  5.901432e-05  6.549362e-04
## V126   7.069800e-03  4.525394e-03  3.326479e-03  2.427625e-03  1.872785e-03
## V127   2.304884e-03  1.712013e-03  7.436508e-03  2.971502e-03  8.573474e-04
## V128   1.444914e-02  1.346503e-03  3.294444e-03 -1.242102e-04  5.190394e-03
## V129   8.753775e-03  3.655421e-03  2.089377e-03  2.259605e-03  5.108449e-03
## V130   1.354028e-04  2.330151e-03  1.631025e-03  2.222471e-03  1.404313e-03
## V131   3.566173e-04  1.083279e-03  8.787879e-04 -1.155280e-04  9.807409e-04
## V132   9.281214e-04 -1.565699e-04  1.147840e-02 -4.599146e-04  3.046899e-03
## V133   2.149979e-04  3.822554e-03  1.625275e-03  5.930261e-03  1.632951e-04
## V134   2.509759e-04 -2.509804e-04  3.333333e-04  1.836844e-04  7.132948e-04
## V135   4.509916e-04  4.872451e-03  3.610462e-03  4.248987e-03  5.688308e-04
## V136   2.140026e-03  2.349895e-03  1.896825e-03  7.478194e-04  9.643624e-04
## V137   1.629604e-04  3.642931e-03  6.291414e-03  2.941708e-03  1.809542e-03
## V138   2.849613e-03  2.154762e-04  2.500000e-04  1.865830e-03  6.120449e-04
## V139   3.912453e-04  3.212313e-03  2.652381e-03  1.753948e-03  1.518779e-03
## V140   4.552403e-04  2.340862e-03  2.949206e-03  2.969730e-03  6.335084e-04
## V141   2.036715e-02  1.146908e-03  5.760073e-03  2.224160e-03  3.264707e-03
## V142   9.138628e-03  5.516673e-03  2.810317e-03  4.078969e-03  3.255491e-03
## V143   1.626303e-04  7.894273e-03  3.186652e-03  7.276616e-04  1.495003e-03
## V144   2.177961e-04  1.129033e-03  1.666667e-04  1.938172e-04 -1.871429e-04
## V145   2.743320e-04  8.522765e-04  1.126984e-04  6.954521e-04 -1.349648e-04
## V146   1.727679e-04  9.451104e-04  7.151515e-04  5.751903e-04  1.784848e-04
## V147   3.386103e-06 -1.857143e-04  8.316239e-04 -3.724310e-04 -8.088403e-05
```

```
## V148 -2.347143e-05  0.000000e+00  3.818182e-04  1.405178e-03  1.785496e-03
## V149 -6.554800e-06  2.062436e-04  2.857143e-04  1.834315e-03  1.100619e-03
## V150  1.074354e-04 -1.111111e-04  4.895105e-04  1.019436e-04 -1.798358e-04
## V151  3.757120e-05 -1.350251e-04  2.000000e-04  2.561664e-04 -8.574412e-04
## V152  1.612528e-05 -1.637427e-05  4.000000e-04  8.220991e-04  8.937845e-04
## V153  1.169113e-05  4.895769e-04 -1.666667e-04  4.992962e-04  7.608696e-05
## V154  1.859146e-04  3.452381e-04  1.035714e-03  6.887149e-06 -5.555993e-04
## V155  3.804155e-04  2.696078e-04  0.000000e+00  3.180952e-04 -8.584795e-05
## V156  1.322855e-04 -3.380952e-04 -2.777778e-04  8.218858e-04  1.177793e-03
## V157  1.534223e-05 -3.239963e-04  2.651515e-04  4.602705e-04  2.430083e-03
## V158  9.821260e-05  1.223191e-03  3.246753e-04  7.003001e-05  5.951984e-04
## V159  3.014661e-05  1.879085e-04  2.427778e-03  3.504715e-04 -2.287582e-04
## V160  3.790509e-05  3.931622e-03  1.691270e-03 -5.445662e-05 -1.232278e-04
## V161 -7.716321e-06  3.255013e-04  9.829060e-05  9.289957e-05 -1.209693e-04
## V162  5.789080e-04  3.046757e-05  2.722222e-04 -1.210726e-04 -4.015043e-04
## V163  1.797417e-03  2.428368e-04  1.329915e-03  6.350375e-04  1.451082e-04
## V164  5.409794e-04  2.589667e-03  9.412698e-04  9.764187e-04  2.214221e-03
## V165  1.047313e-03  4.205128e-04 -1.538462e-04  6.608162e-04 -6.344241e-04
## V166  1.131281e-03 -8.450418e-05  1.111111e-04  6.930920e-04  1.160829e-03
## V167  3.213423e-04  5.328144e-04  8.436508e-04  1.159557e-03  4.661388e-04
## V168  3.038611e-04  9.712406e-04  2.142857e-04  3.533487e-04  8.586910e-04
## V169  6.237560e-03  2.120699e-03  2.640781e-03  3.520106e-04  6.149250e-04
## V170  9.153763e-04  4.210328e-05  3.333333e-04  1.441495e-03  4.762271e-04
## V171  6.753034e-05  4.666667e-04  7.888889e-04  5.065249e-04  3.891049e-04
## V172  1.163712e-03  5.583524e-04 -5.000000e-04  1.099811e-04 -3.623401e-05
## V173  6.166198e-05 -4.991094e-05 -3.294372e-04  1.369268e-03  6.211306e-04
## V174  4.801893e-05  2.489680e-05 -4.668110e-04  4.430169e-04  1.456689e-03
## V175  7.924507e-05  2.285714e-04  5.095960e-04  5.242220e-04  4.111629e-04
## V176  6.835636e-03 -5.160612e-04  2.155556e-03  8.472277e-04  1.051460e-03
## V177  4.045874e-04  1.179949e-03  1.713492e-03  1.160721e-03  7.013729e-04
## V178  7.815421e-05  1.869048e-04 -5.555556e-05  3.764192e-04  6.668489e-04
## V179  6.680464e-04  1.336078e-03  1.725397e-03  5.555331e-04 -6.309942e-05
## V180  1.976276e-03  1.619082e-03  1.410317e-03  1.632250e-03  2.937792e-03
## V181  3.724053e-03  2.419066e-05  2.333333e-04  8.133661e-04  1.135766e-03
## V182  1.731953e-02  4.384430e-03  8.257931e-03 -1.358245e-03 -5.013000e-04
## V183  1.319444e-04 -2.743992e-05  2.473810e-03  3.985339e-04  4.963626e-04
## V184  4.677168e-04  3.554258e-04  3.484848e-04  1.293640e-03  5.993210e-04
## V185  7.848070e-03  5.614035e-04  2.788095e-03  1.826348e-04  8.617252e-04
## V186  1.275562e-04 -1.544241e-04  9.777778e-04 -4.907531e-04 -9.978022e-05
## V187 -4.134236e-06  9.523810e-05 -2.222222e-04  5.295802e-05 -4.032723e-04
## V188  5.644954e-05 -2.568340e-04  6.666667e-04  3.042737e-04  4.996673e-04
## V189  4.041135e-06  2.005013e-05 -4.038462e-04  4.424779e-04  7.765364e-04
## V190  1.849048e-04  2.219298e-04  6.785714e-04  4.723817e-04 -2.722110e-04
## V191  5.313868e-05  9.934042e-05  6.349206e-05 -2.121092e-04  2.499222e-04
## V192  2.660509e-05 -1.152966e-04  0.000000e+00  5.956648e-04 -2.962185e-04
## V193 -1.523422e-05 -2.426471e-04  3.095238e-04 -2.120051e-04  8.000000e-05
## V194  6.489762e-05 -1.529670e-04  4.500000e-04  4.691833e-04 -1.047253e-04
## V195  3.664209e-05  5.382775e-06  0.000000e+00  1.516869e-04  4.467024e-04
## V196  8.084032e-05  1.111111e-05 -1.818182e-05  3.409585e-04  8.387250e-04
## V197  3.363926e-06  1.008454e-04 -4.040404e-04  4.533177e-04  2.379601e-03
```

```
## V198  4.760048e-05  7.252430e-04 -1.746032e-05  2.587368e-04 -1.963884e-05
## V199 -1.858704e-05  1.052632e-04  1.165079e-03  5.898787e-04 -2.282947e-04
## V200  4.199634e-05 -1.926471e-04  1.334055e-03 -3.464614e-05  9.179413e-04
## V201  6.125224e-04  9.277993e-04  8.202453e-03  2.681330e-03  1.857201e-03
## V202  1.245521e-04  5.259254e-03  1.389091e-02  5.243152e-03  5.931526e-03
## V203  2.987356e-05  3.502244e-03  1.118133e-02  4.851770e-03  4.293718e-03
## V204  7.146812e-04  1.836776e-03  5.372294e-04  1.782522e-03  1.013167e-03
## V205  6.996746e-06  2.173464e-04  8.047619e-04 -3.897450e-04  5.362510e-04
## V206  1.406062e-04  3.767246e-03  2.612843e-03  7.116742e-03  2.169029e-03
## V207  7.525871e-04  1.222432e-03  1.660317e-03  3.407568e-03  2.621680e-03
## V208  4.852527e-04  5.274634e-04  6.177417e-03  3.760130e-03  1.859887e-03
## V209  8.795491e-04  1.547619e-04  7.777778e-04  1.759964e-03  2.993609e-05
## V210  2.694149e-05  2.832316e-03  1.660317e-03  3.128556e-03 -1.345526e-04
## V211  1.098416e-05 -5.524998e-05 -4.849206e-04  3.631126e-03  5.892943e-04
## V212  7.451677e-06  3.714932e-04  0.000000e+00  1.907813e-03 -1.673677e-04
## V213 -1.181864e-05 -4.891813e-04  2.214286e-03  2.853944e-03  2.089928e-04
## V214  3.177376e-04  1.454777e-03  5.324359e-03  4.772300e-03  2.194340e-03
## V215  4.148148e-04  2.846003e-03  1.415014e-02  8.564720e-03  1.038246e-02
## V216  3.631675e-04  2.228394e-03  8.907576e-03  8.860861e-03  4.767008e-03
## V217  2.316644e-04  1.043280e-03 -1.047619e-04  1.758489e-03 -3.643294e-04
## V218  5.024593e-05  5.715848e-04  2.025541e-03 -8.616494e-04  1.649630e-03
## V219  2.815413e-04  1.955302e-03  5.477434e-03  3.417389e-03  1.818000e-03
## V220  9.349980e-04  1.934201e-03  6.910245e-03  3.657563e-03  2.453947e-03
## V221  3.392007e-05  7.933420e-04  8.720996e-03  2.292693e-03  3.107105e-03
## V222  1.480120e-03  1.847286e-03  2.065224e-03  1.304350e-03  1.677103e-03
## V223  1.952061e-04  4.000951e-03  1.469048e-03  3.257790e-03  1.636901e-03
## V224  3.411853e-05  7.544064e-04 -5.809524e-04  3.513965e-03  1.067372e-03
## V225 -1.922207e-05 -3.714286e-04 -2.000000e-04  2.508083e-03  6.097144e-04
## V226  3.720870e-06 -2.052632e-04  1.533333e-03  2.317376e-03  1.465839e-04
## V227  1.120122e-02  3.155927e-04  5.469192e-03  4.477640e-03  3.242773e-03
## V228 -1.507331e-05  9.536326e-04  8.214286e-04  1.411744e-03  1.253397e-03
## V229  3.246834e-03  1.638271e-03  2.000000e-03  1.380238e-03  7.793885e-04
## V230  3.786273e-06  3.398693e-04  5.333333e-04 -6.956945e-05  5.084118e-04
## V231  4.630645e-03  2.544872e-04  3.354701e-04 -4.032046e-05  3.354073e-04
## V232  1.321809e-02 -3.549063e-04  3.049423e-03  3.820831e-03  8.909247e-04
## V233  6.511917e-03  1.931061e-03  7.238817e-04  1.802121e-03  9.781971e-04
## V234  1.656172e-03  3.799232e-03  2.722294e-03  3.392789e-03  9.241091e-04
## V235  5.706118e-03  8.793601e-04 -6.365079e-04  3.127505e-03  9.982187e-04
## V236  2.278898e-05 -4.103937e-05  1.007937e-03  8.182161e-04  8.757783e-05
## V237  1.108972e-04  1.919414e-04  9.095238e-04 -7.441013e-05 -3.298264e-04
## V238 -1.133372e-05 -5.107042e-04 -2.000000e-04  1.934866e-04 -1.635115e-04
## V239  7.952924e-06 -1.250000e-04  0.000000e+00 -1.681717e-04  2.502142e-04
## V240  1.661220e-03  1.689740e-03  2.100000e-03  2.696242e-04  5.960244e-04
## V241  5.746181e-04  1.877289e-05  1.181818e-03 -2.754531e-04  9.092125e-04
## V242  6.638739e-04  1.212255e-03  4.444444e-04  1.164419e-04  2.847826e-04
## V243  2.305486e-05  3.131599e-04  4.444444e-04 -1.491749e-04 -5.172078e-04
## V244 -7.437376e-06 -6.350859e-04  3.209013e-04  1.616370e-04  7.279394e-04
## V245  1.211679e-03  2.351167e-03  8.571429e-05  3.653497e-04 -1.683007e-04
## V246  1.598717e-03  7.248461e-04  4.261905e-04  3.392622e-04 -1.889575e-05
## V247  6.061069e-04  2.718254e-04  1.818182e-04  6.386114e-04  1.750697e-04
```

```
## V248  2.672455e-05  2.183011e-03  1.739105e-03  9.991471e-05  8.695652e-05
## V249  3.503320e-05 -1.052632e-04  1.666667e-04  4.836267e-04  4.567009e-05
## V250  2.101617e-05 -9.287926e-05 -8.333333e-05  6.794931e-04  5.564646e-04
## V251  7.541209e-06  1.052632e-04  4.857143e-04 -7.632230e-05  7.462963e-04
## V252 -7.098508e-06  2.604997e-05  3.571429e-05 -1.878323e-04  1.222636e-03
## V253  9.624045e-04  8.091563e-04  6.000000e-04  1.791306e-04 -6.405724e-04
## V254  5.781682e-05 -4.294962e-04 -1.944444e-04 -1.095083e-04  2.296129e-04
## V255  2.806324e-05  6.445618e-04  2.523810e-04  6.412140e-04  6.730536e-04
## V256  8.039451e-05  9.150327e-06 -1.239316e-04  2.736742e-04  1.577560e-04
## V257  3.499362e-03  3.963356e-04  4.761905e-04  7.150274e-05 -2.318684e-05
## V258  8.665124e-04  1.422642e-03  7.215007e-06  7.137219e-04  3.846443e-05
## V259  1.466832e-04  6.430409e-04 -1.818182e-05  6.507162e-04  7.724379e-04
## V260  2.285865e-05  3.969298e-04  2.857143e-04  8.474174e-04  6.142481e-06
## V261  3.470461e-05  1.199762e-03  1.476190e-03  4.180553e-04  5.289258e-04
## V262  3.890928e-05  1.000000e-04 -1.818182e-04 -1.184872e-04  5.217827e-04
## V263  4.167870e-05 -1.818182e-04  6.944444e-04  5.648749e-05 -1.050697e-04
## V264  1.127486e-05  3.133700e-04  4.444444e-04  3.070493e-05  3.956010e-04
## V265 -1.819378e-05  1.111111e-04 -3.333333e-04  2.496099e-04 -8.695652e-05
## V266  5.939680e-03  1.042963e-03  3.864164e-03  1.206197e-03  2.254809e-03
## V267  4.158349e-04  1.011485e-03  2.084199e-03  8.016055e-04 -2.666850e-04
## V268  3.036160e-04  6.484127e-04  1.094444e-03 -8.426677e-05  1.728982e-04
## V269  2.649787e-03  5.128160e-03  3.281025e-03  3.224028e-03  3.238117e-03
## V270  3.146549e-04  1.150897e-03  1.578644e-03  8.972234e-04 -1.377926e-05
## V271  1.507242e-04  2.967532e-04  4.871795e-04  3.523283e-04 -2.920746e-04
## V272  5.341741e-03  2.877603e-03  1.452381e-03  3.213631e-03  3.946831e-03
## V273  9.216537e-05  1.556667e-03  3.151515e-04  3.675946e-04 -9.573935e-05
## V274  8.132962e-05  2.535842e-04 -1.031746e-04 -4.645996e-05  9.889076e-05
## V275  1.904659e-03  4.332253e-03  5.667216e-03  1.768766e-03  3.506714e-03
## V276  5.667058e-04  1.590100e-03  2.471429e-03  1.515383e-03  6.664950e-04
## V277  3.223561e-04  2.777778e-05  4.000000e-04  5.279694e-04 -6.783217e-05
## V278  1.106989e-05  4.506286e-04  3.636364e-04 -1.300325e-04 -5.811556e-04
## V279 -1.620431e-05  1.819380e-03  6.666667e-04 -3.121767e-04 -9.487819e-05
## V280  6.442472e-05  3.039904e-04 -1.269841e-04  1.263447e-03 -1.214969e-04
## V281  8.160779e-03  1.191484e-03 -2.226190e-03  2.482783e-03  1.340577e-03
## V282  3.274534e-03  6.485403e-03  7.562698e-03  2.614543e-03  4.914823e-03
## V283  4.017063e-04  1.147287e-03  1.574603e-03  1.325517e-03  9.741126e-04
## V284  5.352705e-04 -7.891559e-05  0.000000e+00  1.132601e-03 -9.784811e-04
## V285  7.993298e-04 -2.619048e-04 -5.555556e-06  1.617382e-04  6.217096e-04
## V286  8.814490e-05  4.785215e-04  2.967033e-04  3.058201e-04  5.338787e-04
## V287  2.664600e-03  1.264922e-03  1.666667e-04  8.035229e-05  2.464803e-04
## V288  4.769388e-03  1.826485e-03  1.997691e-03 -5.699961e-04  6.135723e-04
## V289  1.175946e-03 -1.052632e-04 -4.500000e-04  2.690665e-04 -4.714666e-04
## V290  9.111210e-04  5.564664e-04  1.818182e-04 -2.871795e-04  2.376110e-04
## V291  4.006148e-04  5.911541e-03  9.482479e-03  1.276622e-02  1.121224e-02
## V292 -2.205453e-05  3.182704e-03  3.804762e-03  1.914126e-03  1.325978e-04
## V293  1.137972e-05  2.604656e-03  1.413636e-03  2.288268e-03 -9.386473e-05
## V294  3.309468e-04  4.394213e-03  5.180952e-03  1.271604e-03  3.089908e-03
## V295  1.923654e-04  2.937845e-04  1.698254e-02  4.236839e-04  1.376853e-03
## V296  6.613148e-07  2.835377e-03  6.484127e-03  3.255906e-04  1.069438e-04
## V297  6.508808e-04 -1.693206e-03  3.093529e-03  1.504055e-03  2.730003e-03
```

```
## V298  3.427333e-04 -1.481255e-04  8.078644e-04  7.259338e-04  3.430332e-03
## V299  3.777880e-05  2.655670e-03  4.978449e-03  1.828847e-03  2.356687e-03
## V300  1.090196e-04  1.420744e-03  2.963370e-03  1.965096e-03  1.358979e-03
## V301  4.392129e-05  2.531550e-04  3.246753e-04  1.159834e-03  7.692308e-05
## V302 -1.485873e-05 -5.523563e-04  9.895105e-04 -2.778670e-04  3.074879e-04
## V303  3.076049e-03  6.103219e-03  5.231097e-03  3.547459e-03  3.007112e-03
## V304  1.266158e-04  1.631852e-03 -5.389610e-05  1.549720e-03  9.185367e-04
## V305  3.829416e-05  2.778656e-04 -5.555556e-04 -7.858068e-04  1.150247e-03
## V306  8.837916e-04  4.355263e-04  3.888889e-04  2.279381e-04  1.379014e-03
## V307  7.550445e-05  9.972558e-04  8.136364e-04  4.606999e-04  3.400602e-04
## V308  4.592398e-05  2.644703e-04  6.897547e-04  6.920192e-04  1.079271e-03
## V309  3.755051e-05  1.238447e-03  7.000000e-04  9.282330e-05  2.426121e-04
## V310  6.789264e-05 -5.521368e-04  2.777778e-05  9.496807e-05  4.039262e-04
## V311  3.717974e-03  4.229060e-03  8.150622e-03  1.256486e-03  5.251612e-03
## V312  1.273235e-05  3.974450e-04 -7.070707e-05  2.235703e-04  9.807947e-04
## V313  1.381760e-03 -2.777778e-06  8.984127e-04  7.380441e-05  9.849937e-04
## V314  1.966834e-04  5.119858e-04  1.189899e-03 -5.873964e-04  1.823246e-04
## V315  4.955780e-03  5.666464e-03  5.388167e-03  3.925905e-03  5.465883e-03
## V316  5.381508e-04  8.637149e-04  1.150000e-03 -3.214830e-05  4.548748e-04
## V317  3.378226e-04  1.468186e-03  2.518998e-03  2.021484e-03  5.590077e-04
## V318  7.982084e-06  6.103175e-04  7.571429e-04 -2.358754e-04  9.585367e-04
## V319  7.730236e-04  2.626114e-03  2.272222e-03  1.500811e-03  7.509431e-04
## V320  4.130744e-04  1.006689e-03  2.228571e-03  1.180419e-03  2.241300e-04
## V321  3.794931e-05  9.737179e-04  3.595238e-04  1.770069e-04 -2.679555e-04
## V322  1.483720e-03  1.060024e-03  3.532540e-03  1.484299e-04  5.536510e-05
## V323  7.981913e-06  1.231530e-03  8.743590e-04 -3.892631e-05 -2.026399e-04
## V324  2.289594e-05  3.258482e-03  2.779221e-04  6.346513e-04 -9.961125e-04
## V325  2.170987e-04  1.495594e-03  1.783333e-03  1.470146e-04  6.268794e-04
## V326  6.406642e-04  1.614893e-03  9.079365e-04  6.836193e-04  9.131203e-04
## V327  1.300585e-04  2.213290e-03  1.421501e-03  1.004003e-03  4.976795e-04
## V328  4.106762e-04  1.315688e-03  3.131369e-04  1.254301e-05 -1.695979e-04
## V329  3.157758e-04  2.404148e-03  1.712121e-03  5.852831e-04  1.683057e-04
## V330  4.282256e-04  1.589681e-03  3.630159e-03 -1.428976e-04  2.496218e-04
## V331  1.803852e-04  7.218385e-04  1.086508e-03  5.441046e-04 -1.304133e-03
## V332  3.125628e-04  8.307149e-05  6.000000e-04  1.118522e-04  9.185312e-04
## V333  1.620173e-03  1.168817e-03 -4.857143e-04 -2.910166e-05 -7.990323e-04
## V334 -1.239684e-05  6.964545e-04  6.444444e-04 -1.278769e-04  3.090909e-05
## V335  1.784018e-05  1.764706e-05 -2.175325e-04  1.358029e-04 -2.635726e-04
## V336 -7.656292e-06  5.210084e-04  8.038462e-04 -1.398601e-05  3.838762e-04
## V337  5.598492e-04  2.490196e-04  3.642857e-04  5.200149e-04 -5.984067e-04
## V338  6.684855e-05  0.000000e+00 -2.857143e-04  4.901297e-04 -2.376235e-04
## V339  5.302720e-04  1.047512e-03  1.111111e-03  3.153519e-04 -1.435872e-03
## V340  5.230422e-04  9.127226e-04  0.000000e+00 -3.496805e-04 -7.650683e-05
## V341  4.518265e-05 -4.121795e-04  6.857143e-04  5.279785e-04 -2.188877e-04
## V342  8.646728e-06  2.285714e-04  1.666667e-04  3.412707e-04 -4.183256e-07
## V343  4.324258e-04 -1.493931e-04  1.818182e-04 -1.785657e-04 -9.445883e-04
## V344 -1.950773e-05  1.250000e-04  3.269841e-04  4.766340e-04 -6.080268e-05
## V345  6.181254e-03  2.282211e-03  1.173810e-03  2.349653e-03  1.071228e-03
## V346  1.547112e-03  3.677232e-05  2.833333e-04  1.284026e-03  5.315868e-04
## V347  1.383007e-03  6.715630e-04  2.500000e-04  2.407860e-06  6.447525e-04
```

```
## V348  8.697958e-03 -3.502381e-05  8.446886e-04  3.161809e-03  2.589081e-03
## V349  1.188266e-04  1.267112e-03 -7.936508e-05  7.224174e-04  1.173044e-03
## V350  6.908450e-05  5.930559e-04  3.888889e-04 -5.333333e-04 -4.812253e-04
## V351  5.251666e-03  1.657904e-03  5.961039e-04  2.699353e-03  3.808251e-03
## V352  1.165647e-03  9.271836e-04 -1.666667e-04  1.064543e-03 -5.713967e-06
## V353  1.953677e-05  4.826680e-04 -2.222222e-04  8.219446e-04  3.043303e-04
## V354  9.844515e-03  2.036384e-03  1.142857e-04  3.617598e-03  3.244943e-03
## V355  4.370575e-04  9.863339e-04  1.202381e-03  9.436098e-04  3.070446e-04
## V356  8.562491e-04  1.081735e-03  2.196032e-03  7.043987e-04  1.849832e-04
## V357  4.151124e-05 -7.424242e-05 -4.722222e-04 -1.086102e-04  6.574315e-05
## V358 -1.561512e-05  5.327770e-04 -2.000000e-04  2.653596e-04  2.146839e-04
## V359  2.727562e-03  3.594771e-04  0.000000e+00  1.252942e-03  6.155844e-04
## V360  4.695511e-03  2.039919e-03  9.095238e-04  2.355444e-03  2.147439e-03
## V361  1.467634e-02  3.399492e-03  2.505556e-03  3.848815e-03  3.110432e-03
## V362  1.158972e-03  4.257374e-04 -6.160895e-04  4.102234e-04  6.333967e-04
## V363  4.657344e-05  4.631470e-04  1.142857e-04  6.355683e-04  5.512842e-04
## V364  9.418660e-04  2.328219e-04 -7.650794e-04  2.979264e-03 -1.284114e-04
## V365  1.176508e-05  1.461664e-03  0.000000e+00  4.299497e-04  4.766087e-04
## V366  1.199362e-04  6.993007e-04  1.111111e-04  1.835897e-04  2.568182e-04
## V367  8.970865e-03  1.039127e-03  1.316667e-03  1.460269e-03  1.300541e-03
## V368  2.935298e-03  1.044423e-03  1.471429e-03  7.404313e-05  7.473330e-04
## V369  2.038154e-03  1.271544e-04  3.333333e-04  4.955002e-04  6.648786e-04
## V370  3.976052e-06  3.351991e-04 -3.834776e-04  8.385155e-04  1.343951e-03
## V371  1.042667e-05 -4.407895e-04  1.100000e-03 -7.074939e-04  3.401189e-04
## V372  2.761232e-05  9.095651e-04 -2.857143e-04  3.551487e-04  1.391304e-05
## V373  4.806011e-05  1.381117e-03  2.214646e-03 -2.850794e-04  9.989648e-05
## V374  2.710088e-05  3.209778e-04  1.688889e-03  3.036331e-04 -1.111111e-04
## V375 -3.163665e-06 -1.147252e-03  2.095960e-03 -3.933601e-04 -1.680288e-05
## V376  1.473134e-05 -1.176471e-04 -6.666667e-04  6.569892e-05 -4.183396e-04
## V377 -1.150981e-05 -3.914950e-05  1.142857e-04  8.661258e-05  1.233482e-04
## V378  3.443757e-05  5.906879e-05 -2.833333e-04 -1.678322e-04  1.405797e-04
## V379  1.913597e-05  2.289683e-04  0.000000e+00  3.002350e-04  3.041393e-04
## V380 -3.456508e-07  3.035114e-05  0.000000e+00  2.397228e-05 -4.566654e-04
## V381 -1.415485e-05 -2.580176e-04  1.270563e-03  4.175450e-05 -1.200397e-04
## V382  1.211685e-02  8.945514e-03  1.052648e-02  5.962479e-03  2.977045e-03
## V383  1.855734e-03  4.988742e-04  3.792063e-03  1.316722e-03  1.695521e-03
## V384  7.186994e-05  8.390977e-04  0.000000e+00  4.265693e-04  5.111481e-04
## V385  9.025498e-05  2.028177e-04  2.000000e-04  1.139602e-03  1.484193e-04
## V386  3.785710e-05 -3.562753e-04 -2.777778e-05 -9.003344e-05 -4.543558e-04
## V387  2.896592e-05  7.955004e-04  1.210317e-03  6.813939e-04  2.908396e-04
## V388  1.908779e-05  1.094394e-03  7.500000e-04 -4.049527e-04  7.927820e-04
## V389  7.710152e-06  1.078853e-03  1.047619e-03 -7.893288e-05  7.251047e-05
## V390  1.552813e-02  2.388913e-03  2.632756e-03  7.785092e-03  5.690594e-03
## V391  1.809628e-03 -1.641667e-03 -4.166667e-04 -1.594990e-04  1.377860e-03
## V392  4.175454e-05 -6.880788e-04 -2.222222e-04  5.856593e-04  3.131313e-04
## V393  1.627715e-05 -3.707582e-05  5.714286e-04  5.097094e-04  3.272129e-04
## V394  1.071186e-02  7.756069e-04  7.199134e-04  4.330152e-03  5.273064e-03
## V395  7.275671e-05  3.753469e-04  1.015152e-03  1.147672e-03  7.472765e-04
## V396  9.301071e-04  3.766082e-04  1.048291e-03  5.344065e-04 -3.218864e-04
## V397  1.484695e-05  1.946387e-04 -1.555556e-04  3.066423e-04  1.419048e-04
```

```
## V398  9.666257e-04  3.851295e-04 -1.984127e-05  8.196594e-04 -2.554231e-04
## V399  4.130058e-03  3.140580e-04  7.792208e-05  2.085889e-03  1.324819e-04
## V400 -1.176962e-05  4.322464e-04 -3.523810e-04  9.604040e-04  8.011469e-04
## V401  6.537826e-04 -1.568627e-05 -1.628205e-04  3.198819e-05  6.551176e-04
## V402  6.103696e-04 -5.347594e-05  2.833333e-04  4.536370e-05  1.517894e-03
## V403  7.618562e-06 -2.564480e-04  2.222222e-04  5.733267e-04  2.315955e-04
## V404  3.404158e-03 -4.486690e-04  2.366522e-04  8.884965e-04  3.758259e-05
## V405  5.896978e-03  1.236297e-03  4.273504e-05  5.954907e-04  1.492796e-03
## V406 -1.581151e-05 -3.230676e-04 -2.222222e-04  3.868639e-04  2.505848e-04
## V407  8.164513e-05  3.111111e-04  0.000000e+00  3.428571e-04  1.847360e-03
## V408  1.131229e-03  2.446312e-04  4.000000e-04 -2.361923e-04  2.548951e-04
## V409  3.806733e-03 -2.837011e-04  6.083694e-04  8.302395e-04 -1.322557e-04
## V410  1.483475e-03  2.123552e-03 -2.285714e-04  7.979230e-04 -4.575163e-06
## V411  1.518090e-03  2.971313e-04 -1.818182e-04 -7.526932e-06  3.316924e-04
## V412  5.531726e-04  1.691846e-03 -5.707071e-04  6.169448e-04  2.845285e-04
## V413  3.937793e-05  2.986275e-04 -6.349206e-05  7.745832e-04 -3.786518e-04
## V414  2.747493e-03  8.296630e-04  1.333333e-03  8.418172e-04 -7.075874e-05
## V415  8.450555e-04  2.780159e-04  5.000000e-04 -2.941029e-04  3.028656e-04
## V416  1.593544e-05 -9.210526e-06 -5.555556e-05  1.157325e-03  3.484352e-04
## V417  3.248024e-03 -2.750000e-04  2.857143e-04  6.519456e-05  2.619048e-04
## V418  1.127107e-03 -7.979798e-05 -5.000000e-04  1.534749e-04 -2.240896e-05
## V419  9.600577e-04  2.045513e-03  6.841991e-04  9.826584e-04  3.609963e-04
## V420  4.630757e-04 -1.261610e-05  1.349206e-04  7.759042e-04  1.352941e-04
## V421  2.717891e-05  1.111111e-04 -2.500000e-04  1.951239e-03  9.045156e-04
## V422  1.701226e-03  1.591168e-04  1.609524e-03  8.282299e-04  2.461557e-04
## V423  8.188690e-04  8.141495e-04  5.000000e-04  1.230416e-03 -7.304014e-04
## V424  4.076783e-04  6.678673e-04  3.180159e-03  1.277469e-03  7.238522e-04
## V425  1.355835e-03 -1.561111e-04  5.000000e-04 -1.179622e-04 -1.432079e-04
## V426  2.102341e-03  9.073296e-04  1.079365e-04  4.761001e-04  1.370734e-03
## V427  7.857797e-04  1.334245e-03  3.435015e-03  2.274178e-03  1.587494e-03
## V428  1.260952e-02  6.083333e-04  6.262626e-04  3.740399e-04  2.176614e-03
## V429  5.170090e-03  3.556269e-03  4.905556e-03  1.075856e-03  3.099767e-03
## V430  4.449280e-04  2.288773e-03  5.436286e-03  3.022004e-03  8.139751e-04
## V431  2.263554e-03  1.375794e-03  1.003247e-03  9.881839e-04 -6.229056e-04
## V432  2.733608e-03  1.721911e-03  7.373737e-04  9.688065e-04  9.322855e-04
## V433  2.679949e-04  7.486219e-04  4.019913e-03  4.006752e-03  6.940632e-04
## V434  1.214704e-02  1.335799e-03  2.353175e-03  1.142291e-03  3.261946e-03
## V435  6.751180e-03  6.437463e-03  3.975469e-03  1.874656e-03  3.029742e-03
## V436 -1.155913e-05 -5.382820e-04 -2.000000e-04  2.290100e-04  2.844091e-04
## V437  1.060310e-04  1.052632e-04  2.500000e-04 -5.175662e-07  1.254652e-04
## V438 -7.600579e-06  2.222222e-04  0.000000e+00  5.402525e-04  1.130690e-03
## V439  5.701540e-03  6.688392e-04  1.321429e-03  1.474545e-03  8.642869e-04
## V440  5.064960e-04  1.787470e-03  2.871856e-03  3.929616e-03  2.655027e-03
## V441  1.169964e-02  0.000000e+00  7.936508e-04  7.558220e-04  5.155110e-04
## V442  9.399295e-03  2.781720e-03  3.232828e-03  1.847705e-03  8.350667e-04
## V443  2.791368e-04  2.000000e-04  4.681818e-04  1.536938e-04  2.246441e-04
## V444  4.968288e-05 -1.967601e-04  6.047619e-04 -1.735127e-04  7.467197e-04
## V445  3.950373e-04  3.438596e-04  7.000000e-04 -3.142066e-04  1.531784e-03
## V446  4.115533e-04  8.259702e-04  3.079365e-04  1.469739e-03  6.553294e-04
## V447  1.112176e-04 -2.991453e-04  0.000000e+00  5.058788e-04  6.995819e-05
```

```
## V448  3.682800e-04  6.526611e-05  0.000000e+00 -2.234848e-04  4.069846e-04
## V449  8.041075e-06 -1.472025e-05  3.571429e-05  1.231840e-03  2.781902e-04
## V450  3.738318e-06  8.746443e-04  2.857143e-04  7.160374e-04  4.113675e-04
## V451  4.125135e-05  6.367522e-03  2.011183e-03  5.569559e-03  4.485373e-03
## V452  5.364478e-05  1.966942e-03  3.300794e-03  1.888082e-03  3.586585e-04
## V453  1.904852e-04 -1.865503e-04  3.353247e-03  4.768111e-05  1.624768e-03
## V454  1.921819e-04  8.979725e-04  3.174747e-03  1.082535e-03 -4.551821e-06
## V455  3.772449e-05  1.000000e-04  5.206349e-04  5.271110e-04 -8.840689e-04
## V456  4.925583e-05 -1.176471e-04 -4.040404e-05 -1.354692e-04 -5.856536e-05
## V457  6.958961e-05  2.556137e-04  1.066667e-03  2.259944e-04  5.756917e-04
## V458  1.107935e-05  4.149416e-04  4.523810e-04  1.066376e-03  2.386797e-03
## V459  1.463510e-04  3.637269e-04  1.446970e-03  2.504468e-03  4.552986e-03
## V460  9.122384e-05  3.773717e-04  1.265873e-03  1.518396e-03  8.229651e-04
## V461  2.465522e-04  1.815147e-03  4.116883e-03  3.446046e-03  1.014559e-03
## V462  3.057404e-05  1.234552e-03  5.589466e-04  3.752904e-04  1.105332e-03
## V463  6.246064e-04  2.961722e-04  2.857143e-04  7.766402e-04  1.273624e-03
## V464  4.502425e-05  1.454402e-05  4.222222e-04 -6.704336e-05  6.054843e-04
## V465  2.637246e-04  5.883041e-04  6.136364e-04  1.138508e-03  4.216194e-06
## V466  1.279542e-04  8.388889e-04  1.057143e-03  2.278518e-03 -7.683710e-04
## V467  1.058670e-05  7.935579e-04 -1.777778e-04  5.342099e-04  3.233453e-04
## V468  4.139225e-05  7.351835e-05  5.404040e-04  3.909637e-04  5.368141e-04
## V469  4.341910e-04  1.435772e-03  3.106061e-03  2.622617e-03  2.493793e-03
## V470  7.320410e-05  7.042918e-04  1.428571e-04  1.207727e-03  9.134598e-04
## V471  6.142249e-05  3.329353e-04 -3.555556e-04  4.131043e-04  3.240828e-04
## V472  2.743636e-05  1.351748e-03 -9.896104e-04  1.033673e-03 -4.412088e-04
## V473  1.025724e-03  1.884178e-03  4.391919e-03  1.526566e-03  1.559281e-03
## V474  1.066257e-05  8.525253e-04  2.135714e-03  6.336705e-04  1.143619e-03
## V475  1.089565e-02  1.225774e-03  6.500000e-04  1.054956e-03  3.257981e-03
## V476  2.090706e-03  1.897343e-04  2.857143e-04  1.771360e-05 -2.476379e-04
## V477 -1.554856e-05  1.250000e-04  4.000000e-04  6.221694e-04  9.610954e-04
## V478  1.273865e-03 -1.187913e-04 -8.190476e-04  5.079659e-04  5.163373e-04
## V479  9.621240e-05  3.371738e-04  8.333333e-05  1.341986e-04  1.208125e-05
## V480  6.223438e-05 -1.960759e-04  2.857143e-04  2.415587e-04  1.254329e-04
## V481  6.145552e-04 -9.047619e-05  0.000000e+00  4.650694e-04  1.821691e-04
## V482  2.883876e-04 -2.444444e-04 -2.857143e-04 -7.692308e-05 -7.297717e-05
## V483  1.223406e-02  1.457172e-04  6.000000e-04  1.191867e-03  4.921226e-03
## V484  2.405317e-04  2.241830e-04 -4.444444e-04  4.720238e-04 -5.889606e-04
## V485  8.590192e-04  0.000000e+00  8.650794e-04  1.574219e-04  2.270893e-04
## V486  4.831259e-04 -2.113527e-05  0.000000e+00 -3.702475e-04 -6.168896e-04
## V487  6.081192e-03  4.942704e-04  8.001443e-04  1.127714e-04 -1.370920e-04
## V488  2.307837e-04  2.797619e-04  2.222222e-04  3.211491e-04  8.521778e-04
## V489  5.191310e-03  2.974831e-03  7.776335e-03  3.814059e-03  4.076469e-03
## V490  3.163546e-05  2.023597e-04 -4.040404e-04  1.823519e-04  1.608703e-04
## V491  4.745798e-04  7.861111e-04  5.151515e-04 -1.364146e-04  1.434809e-04
## V492  1.801872e-04  3.741360e-04  1.414141e-04  1.413999e-04  5.922596e-04
## V493  6.605059e-04 -2.292929e-04  3.214286e-04  1.874942e-04  6.865117e-04
## V494  7.745401e-05  3.416667e-04 -2.000000e-04 -3.379342e-04  3.970499e-04
## V495  7.277541e-06 -1.215368e-04 -1.111111e-04  2.908360e-04  4.060559e-04
## V496  7.046401e-05  4.539177e-04  1.818182e-04 -1.302968e-04  6.592212e-04
## V497  6.351324e-03  5.046319e-03  6.655123e-03  3.426982e-03  2.528567e-03
```

```
## V498  1.185548e-03  4.997076e-04  5.707071e-04  4.979550e-05  3.585237e-04
## V499  4.741832e-04  2.111111e-04  9.993506e-04  1.859989e-04  2.814010e-04
## V500  1.385154e-04  1.840251e-04  2.000000e-04 -3.869048e-05 -7.209638e-05
## V501  8.717647e-03  4.414023e-03  2.326407e-03  2.287109e-03  3.089765e-03
## V502  1.340983e-04  9.034842e-05  3.386580e-03  8.652248e-05  1.002934e-03
## V503  4.470391e-04 -8.169308e-04  8.508658e-04  5.265841e-04 -1.608306e-03
## V504  6.494311e-05  7.098846e-03  1.677698e-02  8.316500e-03  6.536676e-03
## V505  1.213800e-04  1.297375e-03  5.152453e-03  6.273347e-04  4.331356e-03
## V506  2.268474e-04  3.221534e-03  7.119192e-03  5.036638e-03  5.081778e-03
## V507  7.465011e-06  3.104878e-04 -1.000866e-03  7.180825e-04 -2.577955e-03
## V508  4.686599e-04 -3.553386e-04  2.777778e-05  2.269055e-04 -1.372860e-03
## V509  2.116722e-04  2.073408e-03  1.169293e-02  4.964537e-03  7.262399e-03
## V510  7.740375e-04  1.069441e-03 -1.777778e-04  3.731458e-04  9.443657e-04
## V511  7.744911e-05  5.761905e-04  7.142857e-04  6.881873e-04  2.333826e-03
## V512  9.927312e-05  1.128629e-02  1.163254e-02  9.044260e-03  6.043396e-03
## V513  6.156054e-05  2.957718e-03  7.109618e-03  2.106853e-03  3.869839e-03
## V514  2.203048e-05 -9.036415e-04  7.882540e-03  1.283336e-03  5.102690e-03
## V515  1.477955e-05  4.745430e-04  6.352758e-03  7.940742e-04  2.395199e-03
## V516  2.188089e-05  7.113264e-04  3.538961e-04  7.794231e-04  1.906667e-04
## V517  1.180328e-04  1.213607e-03  1.666667e-04  2.210842e-03  5.239011e-04
## V518  8.979890e-04  9.879371e-04  5.500000e-04  1.097049e-03  6.963803e-05
## V519  7.267649e-05  1.420379e-03  1.003175e-03  1.894100e-03  2.507525e-04
## V520  1.476033e-03  4.228419e-04  0.000000e+00  5.809524e-05  9.814553e-04
## V521  3.200830e-04  6.767343e-04  7.142857e-04  1.302527e-03  1.075323e-03
## V522  3.265141e-05  1.643603e-03  8.316739e-04  1.187974e-03  9.079785e-04
## V523  2.074338e-03  1.872294e-05 -6.444444e-04  4.559407e-04  4.725469e-04
## V524 -3.734265e-06  4.761905e-06  1.027778e-03  4.907163e-04 -4.273194e-04
## V525  3.939962e-06  2.109185e-03  4.642857e-04  1.580989e-03  9.454664e-04
## V526  2.256086e-05 -9.523810e-06  1.048485e-03  3.684274e-04  2.325748e-04
## V527 -7.671880e-06  0.000000e+00 -1.818182e-04 -1.355153e-04 -5.187663e-04
## V528 -3.968127e-06  1.176471e-04  3.190476e-04  5.144587e-05  1.871276e-04
## V529  3.701465e-04  1.166667e-04  4.857143e-04  6.138976e-05 -2.349389e-04
## V530  4.844382e-04 -4.166667e-05  8.222222e-04 -4.510026e-04 -2.382335e-05
## V531  7.702797e-04  6.526611e-05  6.722222e-04  3.409091e-05  3.073468e-05
## V532  5.890059e-04  4.093137e-04  6.857143e-04  3.083369e-04  9.297577e-04
## V533  2.287577e-04  2.673797e-05  4.500000e-04 -1.898272e-04  4.259259e-05
## V534  4.695016e-04  3.600251e-04  2.666667e-04  2.954902e-04  7.079089e-04
## V535  1.243914e-03  5.206349e-04 -8.333333e-05  2.853627e-04  7.470996e-04
## V536  2.743178e-05  1.782578e-04  6.444444e-04  5.602779e-04 -1.563917e-04
## V537  2.396191e-05 -1.523974e-04  5.758741e-04  2.355136e-04  3.646004e-04
## V538  1.914913e-05  8.717949e-05  4.040404e-04  2.195767e-04 -3.621946e-04
## V539  2.649239e-05  1.011532e-03  1.821429e-03  3.794729e-04  6.712296e-04
## V540  2.295533e-05  6.658890e-04  2.666667e-04 -2.332852e-05  1.071571e-03
## V541  3.406587e-05  9.696889e-04  3.484848e-04 -1.421348e-06  1.103700e-03
## V542  7.420549e-07  2.111472e-04  4.833333e-04 -1.947557e-04 -9.713348e-05
## V543  3.697615e-05  7.133181e-04 -3.333333e-04  7.383331e-04  9.706328e-04
## V544  4.618283e-04  6.634921e-04  1.538462e-04  9.324504e-04  9.075992e-04
## V545  3.920262e-05  4.637129e-04  3.888889e-04  3.729548e-04 -2.144502e-04
## V546  1.077930e-04  1.333333e-04  3.000000e-04  3.253273e-04 -4.629590e-04
## V547  3.158135e-05 -2.756892e-04  2.500000e-04  5.412506e-04  9.302549e-05
```

```
## V548  1.541603e-05 -1.300310e-04  2.000000e-04 -7.908775e-05  3.503162e-04
## V549  1.294456e-04 -2.825397e-04  2.500000e-04  4.415309e-04  3.186991e-04
## V550  3.898635e-06  4.000000e-04  9.000000e-04 -1.033710e-04 -2.571429e-04
## V551  1.859121e-05  3.163919e-04 -2.857143e-04  7.246831e-04  5.035197e-04
## V552 -6.837011e-06  7.503608e-05  2.222222e-04  3.112079e-06  2.079785e-04
## V553  2.736900e-06  2.222222e-04  0.000000e+00 -7.809991e-05  0.000000e+00
## V554  1.866728e-05  1.176471e-04  0.000000e+00  1.042424e-04 -2.770563e-04
## V555  1.108780e-05  1.013422e-03  1.720779e-04  1.018584e-03  8.905154e-04
## V556  6.215042e-05  2.092916e-03  2.310440e-03  6.263075e-04 -2.571874e-04
## V557  3.122536e-05  1.115050e-03  2.179437e-03  2.167994e-03  1.439615e-03
## V558  7.704493e-05  2.485701e-04  1.211977e-03  2.117346e-03 -6.977224e-04
## V559  1.382701e-01  8.276592e-03  5.901782e-03  9.573469e-03  9.979659e-03
## V560  2.301808e-02  1.621695e-02  1.006999e-02  1.755239e-02  1.974567e-02
## V561  1.082910e-02  1.120361e-03  4.149278e-03  1.856741e-03  5.814850e-03
##                 11            12 MeanDecreaseAccuracy MeanDecreaseGini
## V1    1.395595e-02  2.272420e-02         1.447438e-03        9.2899508
## V2    2.248789e-02  9.615867e-03         1.197762e-03        7.7139452
## V3    8.930154e-03  1.986551e-03         4.621207e-04        3.6451313
## V4    1.590571e-04  2.019437e-03         1.173695e-02       47.0060419
## V5    7.546132e-04  1.258084e-03         8.939583e-04        3.0524853
## V6    1.146788e-03  1.814635e-03         8.716398e-04        2.8086215
## V7    1.106035e-03  1.616324e-03         9.933148e-03       33.8082322
## V8    1.800790e-04 -5.419145e-04         1.585464e-03        3.3399600
## V9    4.811799e-04  1.678927e-03         6.196792e-04        1.7869338
## V10   1.463451e-02  2.118118e-02         2.039259e-02       54.7553865
## V11   2.247372e-02  3.883453e-03         1.254183e-03        5.4978377
## V12   6.098763e-03  4.307775e-04         1.236434e-03        2.5208353
## V13   4.460915e-03  2.438347e-03         1.369969e-03        4.2776527
## V14   1.819628e-03  8.244504e-03         1.255108e-03        3.9922175
## V15   1.100904e-03  6.025697e-03         9.871527e-04        2.4368605
## V16   1.006640e-02  5.165531e-03         6.952321e-03       21.2709231
## V17   7.239000e-04  1.335178e-02         1.388095e-02       43.9806726
## V18   9.472764e-03  3.680168e-03         1.799061e-03        6.7452401
## V19   2.753728e-03  2.102079e-03         1.118176e-03        3.0753409
## V20   1.546027e-03  3.234340e-03         3.451382e-03        9.6744343
## V21   5.889879e-04 -3.179532e-04         8.163075e-04        2.8289968
## V22   6.034150e-04  4.600000e-04         5.294631e-04        1.7414266
## V23   2.630503e-03  2.563043e-02         6.278580e-03       13.6973252
## V24   1.248420e-02  1.291854e-03         1.940007e-03        4.3574677
## V25   1.088745e-02  6.220639e-04         1.013016e-03        3.4246929
## V26   5.246709e-05  1.279720e-04         7.826771e-04        3.1713871
## V27  -8.748404e-05  4.908213e-04         3.394129e-04        1.3712341
## V28  -1.349816e-04 -5.064370e-04         1.021525e-04        1.6359415
## V29   4.376491e-04  6.656010e-05         2.173998e-04        2.1004826
## V30   3.861463e-04  5.194805e-06         2.712551e-04        1.5681045
## V31  -5.128205e-05  3.861102e-04         2.354785e-04        1.1157382
## V32  -8.158093e-05  1.070521e-03         5.945214e-05        1.3561490
## V33  -1.462386e-04 -3.036749e-04         1.048781e-04        1.6172463
## V34   6.578745e-04  1.681334e-03         2.077516e-04        1.3541183
## V35  -2.322767e-06  2.863012e-03         1.373284e-04        1.2110967
```

```
## V36  -6.112034e-04  1.910788e-03          4.191297e-05          1.2148254
## V37   6.060606e-05  2.094805e-04          1.196834e-04          1.4318349
## V38   1.125904e-03  2.893957e-04          8.917158e-03         33.3083286
## V39  -4.511984e-04 -2.081106e-05          1.725026e-03          7.0845404
## V40   2.916264e-03 -6.713105e-05          1.595647e-03         16.1622775
## V41   6.525128e-03  2.161077e-02          4.361464e-02        189.7995998
## V42  -8.616210e-04  1.451426e-02          3.021831e-02        170.8395325
## V43   4.243571e-04 -3.656899e-04          8.048605e-03         53.9602043
## V44   6.455031e-03  6.471628e-03          3.282979e-03         13.5978382
## V45   1.692923e-02  1.332680e-03          1.575544e-03         11.1152451
## V46   5.479540e-03  2.060611e-03          7.896703e-04          4.6924106
## V47   3.065276e-03  6.870098e-03          2.701669e-03         12.3362346
## V48   9.031879e-03  7.460609e-05          1.338437e-03          9.1940469
## V49   5.539057e-03  3.281211e-03          8.133933e-04          5.9179846
## V50   4.467459e-03  4.624195e-03          3.068416e-02        143.7143917
## V51   7.145127e-03  1.753834e-02          2.654159e-02        155.6321946
## V52   8.874073e-04  1.133927e-03          8.879415e-03         59.3846208
## V53   6.340375e-03  3.667739e-02          4.329484e-02        172.9033416
## V54   1.394401e-02  8.197637e-03          2.879512e-02        161.5173085
## V55   2.744695e-03 -6.942159e-04          8.380274e-03         60.1685028
## V56   5.367301e-04  6.146317e-04          3.793415e-03         25.1962287
## V57   8.487589e-03  2.282530e-02          4.407645e-02        177.6137233
## V58   3.532450e-03  1.191325e-02          1.899266e-02         94.7206692
## V59   3.637563e-04 -1.069487e-03          6.024308e-03         39.2086665
## V60   1.330862e-03  1.246540e-03          2.061102e-03          8.0778560
## V61   7.983287e-03  1.381277e-03          1.198109e-03          6.3632612
## V62   3.527848e-03  1.578254e-04          4.607257e-04          3.3764140
## V63   9.859501e-03  5.703474e-03          3.952579e-03         20.2186198
## V64   7.218696e-03  9.820091e-03          6.429787e-03         50.6341982
## V65  -3.758981e-04  2.161861e-04          4.001665e-04          3.2270534
## V66   7.094256e-03  7.109005e-03          6.923881e-03         38.2167016
## V67   5.515146e-03  6.101177e-03          6.566551e-03         36.5332626
## V68   2.058800e-03  1.230808e-04          5.071267e-03         24.5230010
## V69   1.615269e-03  3.820292e-03          3.747801e-03         18.4708208
## V70   1.242702e-02  6.159956e-03          8.578417e-03         48.8110546
## V71   9.009827e-03  3.152921e-03          7.276936e-03         40.9531972
## V72   5.715100e-03  3.021805e-03          4.821322e-03         29.5290280
## V73   8.809106e-04  6.432643e-04          3.696902e-03         23.8454267
## V74   1.508617e-02  6.316829e-03          7.467747e-03         46.4006808
## V75   2.120214e-02  9.162205e-03          9.058268e-03         62.3002667
## V76   1.434954e-02  5.291360e-03          7.436772e-03         42.4040387
## V77   1.087616e-02  4.835656e-03          5.570922e-03         35.2623479
## V78  -2.654965e-03  1.586687e-03          1.414142e-03         19.0382477
## V79   3.154789e-04 -7.559024e-04          5.845204e-04          5.5559548
## V80   6.889544e-06  1.750988e-04          5.713244e-04         12.3893284
## V81   6.782976e-04  1.377931e-03          2.106830e-04          1.6119260
## V82  -3.373985e-04  2.202472e-03          3.840137e-04          1.7546412
## V83  -8.269095e-05  6.998333e-04          3.197654e-04          1.4233529
## V84   5.027470e-03  8.973294e-04          9.201587e-03         29.5387474
## V85   1.987502e-03  1.094634e-03          4.783370e-03         15.8534809
```

```
## V86   7.050664e-04 -3.168182e-04          1.198405e-03         3.2783258
## V87   4.751235e-03  2.779771e-03          1.158133e-02        33.8945734
## V88   5.524914e-03 -1.247823e-03          8.192841e-03        25.6103935
## V89   3.462208e-03  1.346308e-03          4.894510e-03        14.6494120
## V90   3.259818e-03  1.552880e-03          7.781542e-03        20.3489011
## V91   4.198064e-04  6.266051e-04          8.942738e-04         1.9528046
## V92   3.221755e-04 -8.105219e-04          2.430523e-03         4.1750631
## V93   1.268004e-03  2.355334e-04          1.084423e-03         2.6184378
## V94   3.019701e-04 -2.194652e-04          3.914556e-04         1.7014038
## V95   4.514002e-04  7.407407e-05          5.997131e-04         1.6023094
## V96   7.191269e-03  5.264408e-03          1.215893e-02        36.2877546
## V97   3.743193e-03  3.153643e-04          1.022103e-02        31.4678881
## V98   3.348971e-03  3.707886e-05          2.906305e-03         7.9810082
## V99   1.371940e-03  9.294904e-04          1.804472e-03         5.3702465
## V100  7.713690e-03  3.230681e-03          1.626143e-02        45.5830685
## V101  1.093449e-02  1.488471e-03          1.747867e-02        45.7772548
## V102  2.803897e-03 -4.233289e-05          5.247001e-03        14.8171183
## V103  1.722752e-03  1.558515e-03          9.081817e-03        25.6712451
## V104  7.297948e-03 -5.246114e-04          1.074157e-02        32.5719207
## V105 -4.058776e-04 -7.895714e-04          2.403015e-03         6.2683235
## V106 -2.653524e-04 -1.036232e-04          7.811507e-04         1.7723742
## V107 -4.077674e-04  6.465484e-04         -5.020233e-06         1.1710169
## V108  8.938777e-06 -1.025641e-04          9.867777e-05         1.3258158
## V109 -1.563224e-04 -1.649397e-03          2.480424e-04         1.9569961
## V110 -1.265366e-04 -4.139683e-04          1.706349e-04         1.1935480
## V111 -3.257498e-05  2.430181e-04          3.647180e-05         1.3268124
## V112 -2.718726e-04 -2.408027e-04          5.465931e-05         1.4230292
## V113 -3.094406e-04 -8.695652e-05          1.574950e-04         1.6336315
## V114  5.406098e-04  7.960945e-04          5.674678e-04         1.8381585
## V115  3.127462e-04  6.622507e-04          1.035929e-04         1.3905055
## V116  2.974957e-05  1.583057e-04          1.653985e-04         1.4363871
## V117  2.913214e-04  8.684150e-06          2.147317e-04         1.8837301
## V118  8.736429e-04 -2.434178e-04          1.551339e-03         7.3058691
## V119 -5.305192e-04  1.052632e-04          2.566311e-04         2.6873128
## V120  1.537612e-04  9.671291e-05          3.512582e-04         2.3536155
## V121  3.619535e-03  1.773427e-03          1.384693e-03         6.5903560
## V122  6.732709e-04  2.380952e-04          6.311352e-04         2.4919273
## V123 -1.481623e-04  3.544892e-03          1.263597e-03         6.2598422
## V124  7.823094e-03  1.101487e-03          4.167208e-03        14.0581151
## V125  1.614364e-03  6.662251e-04          3.226836e-03         8.0828122
## V126  8.602561e-04  2.033944e-03          3.205229e-03         7.0769234
## V127  4.060956e-03  1.200404e-03          5.554683e-03        14.8987304
## V128  2.476289e-03  5.128360e-03          5.678956e-03        13.5678195
## V129  8.158492e-04  1.356186e-03          3.643362e-03         6.7176066
## V130  1.037606e-03  3.284692e-04          4.103820e-03        18.3076325
## V131  9.578213e-04 -3.053433e-04          7.310056e-04         2.2167875
## V132  3.338104e-04  3.160055e-03          5.918990e-04         3.3949370
## V133  2.539783e-03  8.243989e-04          4.553603e-03        17.5953430
## V134  1.741272e-04  1.885360e-04          9.598234e-04         2.1310312
## V135  4.785767e-04  2.928909e-04          1.439421e-03         4.7217886
```

```
## V136  6.299586e-03  1.999588e-03          3.126499e-03         4.2742862
## V137  1.257782e-02  1.429443e-03          2.791002e-03         9.3050465
## V138 -4.498020e-05  3.253263e-04          1.960080e-03         4.9304643
## V139  7.368338e-04  1.343563e-03          2.359437e-03         5.9379297
## V140  2.255124e-03  1.056440e-03          3.875036e-03        17.7526554
## V141  4.064991e-03  1.528017e-03          8.065579e-03        18.4495287
## V142 -2.097847e-04  7.357656e-05          4.087284e-03         8.6700134
## V143  3.047747e-03 -8.506437e-04          2.337959e-03        12.6514700
## V144  6.559512e-04 -2.802445e-05          6.653464e-04         2.7185954
## V145 -2.094330e-04  3.193968e-03          1.721066e-03         3.8238202
## V146  2.525528e-04  3.810893e-04          9.183779e-04         6.0657206
## V147  6.221891e-04 -4.376471e-04          1.683883e-04         1.1887831
## V148  3.210237e-05 -2.267478e-03          6.100931e-05         1.3792590
## V149  3.254463e-04 -1.034330e-03          3.181922e-04         2.6943867
## V150  2.467708e-04  2.380952e-05          6.783851e-04         3.0167951
## V151  1.165541e-04  8.479186e-04          3.782698e-04         1.6713069
## V152 -5.042932e-04 -3.524910e-04          9.286605e-05         1.3101362
## V153  3.008451e-04 -2.731481e-04          1.951225e-04         1.7643616
## V154  1.018182e-04  5.954248e-04          2.596507e-04         1.0989871
## V155  6.033725e-04  1.110653e-03          3.924432e-04         1.2988551
## V156 -9.307359e-05 -3.724357e-04          2.169386e-04         1.4752788
## V157  4.023063e-05 -1.103492e-03          3.273409e-04         2.6100503
## V158  2.736274e-04  7.905057e-05          1.045099e-03         5.6704155
## V159 -1.729948e-04 -2.067520e-04          7.019155e-04         3.4907884
## V160  9.750453e-04  3.219843e-04          3.303917e-03        11.3854607
## V161  8.886216e-04  6.222277e-04          6.075968e-04         2.5203061
## V162  8.182566e-04 -1.210474e-04          3.693912e-04         1.4464616
## V163 -4.881056e-04  5.854130e-04          8.159206e-04         2.2315422
## V164  7.285636e-04  8.982200e-04          3.755828e-03        13.3709850
## V165  5.854917e-05  7.814107e-04          1.452632e-03         2.5424631
## V166  1.449028e-03 -4.454621e-05          1.847660e-03         4.7985702
## V167  3.866764e-04 -1.538462e-04          4.010384e-03        11.8501000
## V168  5.743221e-04 -3.081232e-04          6.630457e-04         2.3740236
## V169  3.222558e-03 -6.005925e-04          7.545630e-03        21.4059462
## V170 -1.113139e-04  2.147186e-04          1.595377e-03         3.1180938
## V171  7.025645e-04  1.010310e-03          8.535410e-04         2.2944882
## V172  5.086542e-04 -4.182931e-05          2.467818e-03         2.4260462
## V173  1.738772e-03  6.949146e-04          2.157299e-03         7.2801934
## V174  8.371095e-04 -6.654215e-04          9.535061e-04         2.2326438
## V175  2.277584e-04 -3.144894e-04          5.728864e-04         1.8793499
## V176  3.612367e-03 -2.293749e-06          7.659086e-03        21.7172818
## V177  1.208719e-03 -6.638645e-05          3.656045e-03        11.3724472
## V178 -1.962283e-05  3.342082e-05          1.052449e-03         2.7610513
## V179  1.346563e-03  7.082467e-04          2.620822e-03         8.2054723
## V180  1.213407e-03  1.224009e-03          5.786322e-03        20.9932590
## V181  9.633512e-04  1.923748e-03          3.437058e-03         8.9124101
## V182  9.775027e-03 -3.170108e-03          1.804370e-02        49.8992742
## V183  2.519900e-04  3.129305e-04          2.013419e-03         5.0702348
## V184 -6.694985e-05 -3.987934e-04          5.260734e-04         2.0681873
## V185  2.790651e-04  2.160292e-04          3.890115e-03         7.6099504
```

```
## V186  2.340840e-04  6.340341e-05         1.309786e-03        8.6454371
## V187  1.138095e-04 -4.253828e-04         1.159977e-04        1.8666164
## V188 -4.991652e-05 -5.451833e-04         6.682538e-04        4.6267793
## V189  5.846932e-04 -1.676379e-05         5.514882e-05        1.4597921
## V190 -6.855586e-04  3.848739e-05         4.231845e-04        1.8463750
## V191  2.583851e-04  1.872911e-04         6.300939e-05        1.3851117
## V192  4.813118e-04 -1.297101e-04         8.755011e-05        1.3456118
## V193  1.165637e-04 -1.693122e-04         1.685044e-04        1.9736810
## V194  3.646346e-04 -2.544952e-04         3.031587e-04        1.3255591
## V195  6.050149e-04 -1.002506e-05         2.628066e-04        1.2884586
## V196  4.477316e-04 -1.240195e-04         1.609773e-04        1.6912014
## V197 -6.030951e-05 -1.825585e-04         3.098579e-04        2.5766177
## V198  2.847003e-04  4.011905e-04         7.340160e-04        5.3780833
## V199 -2.527357e-04 -3.155280e-04         4.748794e-04        3.6898688
## V200  3.263784e-04 -1.022383e-04         4.013060e-04        3.2752866
## V201  9.831571e-03  3.194388e-03         4.862551e-03       13.0537034
## V202  1.144892e-02  7.211234e-03         1.225295e-02       50.9482078
## V203  9.640805e-03  4.969130e-03         8.759800e-03       37.0479504
## V204  4.776665e-03  3.881245e-03         3.974528e-03       10.6122374
## V205  5.244796e-03  9.981685e-06         2.843242e-04        1.8690637
## V206  5.808773e-03  3.410869e-03         5.231089e-03       14.9822263
## V207  8.707299e-03  6.958579e-03         3.177393e-03       10.8089581
## V208  3.095076e-03  4.424571e-03         1.445518e-03        4.9326414
## V209  6.841681e-04  4.557358e-04         1.095653e-03        3.8785679
## V210  5.395515e-04  3.701500e-03         5.335076e-03       28.8627437
## V211  1.050929e-03  2.540314e-03         1.050152e-03        8.6198392
## V212 -4.051860e-04 -5.602631e-05         2.097276e-04        1.5150648
## V213 -3.708443e-04 -2.631497e-04         2.837501e-04        1.7220016
## V214  8.486038e-03  4.912718e-03         3.236004e-03       10.4091960
## V215  1.072562e-02  7.344322e-03         1.194194e-02       49.6872066
## V216  1.094451e-02  5.689980e-03         8.613431e-03       37.0596514
## V217  5.551869e-03  4.671076e-03         3.851270e-03       19.1521639
## V218  3.252397e-03  4.828917e-04         7.903266e-04        2.1605854
## V219  8.465941e-03  2.961587e-03         4.974357e-03       12.9892806
## V220  1.044405e-02  8.400864e-03         4.140071e-03       11.6244386
## V221  3.701841e-03  1.796410e-03         1.424330e-03        4.5387795
## V222 -2.045153e-04 -2.319158e-04         1.460677e-03        3.2002645
## V223  2.207600e-03  3.553341e-03         5.569670e-03       29.7604762
## V224  1.097238e-03  5.743258e-04         1.300879e-03       12.1791347
## V225  2.886249e-04 -6.975116e-04         1.349981e-04        1.5015049
## V226 -1.897584e-04 -3.044085e-04         2.721471e-04        1.5023261
## V227  7.005674e-03 -9.806083e-05         1.229764e-02       35.4837013
## V228  1.986980e-03  1.584380e-03         2.267740e-03        6.9283507
## V229  2.122403e-03  1.858012e-03         8.108088e-03       17.5341200
## V230 -3.609645e-06  9.209223e-04         4.373287e-04        1.5880692
## V231  8.740938e-04  5.578945e-04         1.923227e-03        1.5117779
## V232  9.785163e-03  3.321964e-03         1.480242e-02       42.0615307
## V233  4.378572e-03  2.789018e-03         8.198819e-03       21.8741519
## V234  3.530120e-03  2.122739e-03         8.888204e-03       24.8010925
## V235  4.880899e-03  2.983552e-03         1.185916e-02       33.0720969
```

```
## V236  2.179686e-04  1.936079e-04           3.555449e-04         1.4121363
## V237 -1.468406e-04  3.771880e-05           4.935913e-04         1.8251647
## V238 -1.765724e-04  2.260870e-04           2.153259e-05         1.2977304
## V239  7.397182e-06 -6.452830e-04           4.130140e-05         1.2564509
## V240  6.025278e-03  5.931906e-04           1.852135e-03         3.5987707
## V241  2.028250e-04  8.254705e-04           4.716265e-04         2.0838867
## V242  5.662767e-04  1.521655e-03           5.875386e-04         1.9499652
## V243  9.464331e-04  3.296296e-04           3.724128e-04         1.9158038
## V244  3.693912e-03 -6.211378e-04           1.838712e-03         2.5200792
## V245  4.702892e-03 -2.882449e-04           2.569845e-03         5.6866837
## V246  4.221071e-03  5.818253e-04           8.389211e-04         2.6708444
## V247 -1.676748e-04  2.351047e-04           3.824287e-04         1.8387952
## V248  1.074543e-03  2.213275e-04           4.394452e-04         1.9233233
## V249  3.073662e-06 -2.328237e-04           2.506098e-04         1.6389401
## V250  4.533190e-04  1.071429e-04           1.948120e-04         1.3872164
## V251  6.711702e-04 -8.514577e-04           1.419586e-04         1.5347061
## V252  6.885723e-04 -5.535714e-04           1.165219e-04         1.5580580
## V253  1.986388e-03  4.765228e-04           2.243980e-03         7.6250013
## V254  1.248680e-03  7.738029e-04           1.035789e-03         2.6161365
## V255  6.680877e-04 -2.249720e-04           9.110723e-04         3.2424593
## V256  2.205043e-04  1.427635e-04           4.947773e-04         1.5519972
## V257  2.265882e-04  5.329450e-04           1.073196e-03         1.5427487
## V258  8.470869e-04  3.193559e-04           1.390550e-03         3.2906963
## V259 -1.406390e-04 -1.380201e-04           9.304717e-04         2.6496484
## V260  2.370480e-05 -5.099157e-04           7.026522e-04         2.4240721
## V261  7.786045e-04  7.677583e-04           3.606268e-04         1.9176112
## V262  9.557945e-06 -2.324362e-04           1.681782e-04         1.8707409
## V263  4.192569e-04  7.205327e-04           1.547840e-04         1.7175368
## V264  3.052148e-04 -1.687523e-04           4.904618e-05         1.5593466
## V265  4.080001e-04 -2.397504e-04           8.198649e-05         1.4488761
## V266  2.638916e-03  3.265677e-03           1.105741e-02        34.2633014
## V267  1.317980e-04 -5.137073e-05           6.004463e-04         2.2396223
## V268  7.996590e-04  7.561786e-04           7.227473e-04         2.1372472
## V269  8.982596e-04  3.587155e-03           8.625199e-03        24.8145450
## V270  1.390951e-03 -9.421403e-06           9.460494e-04         3.6412309
## V271  7.007821e-04  5.258290e-04           7.774992e-04         3.4152688
## V272  3.108374e-03  3.458979e-03           1.270111e-02        48.7433892
## V273  1.389752e-03  2.789526e-04           8.409891e-04         2.3538711
## V274  7.066284e-04  1.191287e-03           1.169409e-03         2.5738757
## V275  4.536098e-04  3.807938e-03           8.777979e-03        29.2659019
## V276  1.647618e-03 -4.929418e-05           2.275411e-03         6.2486647
## V277  1.569999e-03  6.325490e-04           1.333788e-03         2.9782378
## V278 -1.527755e-04  5.508840e-04           2.607638e-04         1.3225287
## V279  4.699139e-04  2.967422e-04           1.333491e-04         1.4943594
## V280 -4.314241e-05  2.651852e-04           1.505070e-04         1.1103432
## V281  4.710438e-03  3.782766e-03           8.988812e-03        27.8262627
## V282  1.901429e-03  4.094257e-03           1.291881e-02        44.0122285
## V283  6.327565e-04  6.968601e-04           9.668980e-04         3.0714428
## V284  6.348598e-04  1.511271e-03           1.156302e-03         2.8773306
## V285  9.604239e-05  6.987097e-04           1.042672e-03         4.0502904
```

```
## V286  5.254330e-04 -6.603403e-04          7.505464e-04          0.9865915
## V287  2.692861e-04  1.829602e-04          1.863837e-03          2.2997648
## V288  3.875583e-03  1.023799e-03          5.615647e-03         16.7633802
## V289  1.756468e-03  4.928628e-04          1.245226e-03          3.8652215
## V290  1.054526e-03 -5.903179e-04          7.183318e-04          1.7203873
## V291  1.319065e-02  7.939970e-03          7.179876e-03         11.7731222
## V292  3.267878e-03  1.360032e-03          2.080803e-03          4.2292728
## V293  1.335982e-03  1.308290e-03          1.076777e-03          3.0466448
## V294 -3.527968e-04  2.584878e-03          1.469591e-03          8.0923390
## V295  8.778576e-04 -7.706869e-04          5.601718e-04          5.2443343
## V296  2.444117e-03  2.661821e-03          1.926922e-03         11.4929160
## V297  8.381881e-04  1.449319e-03          2.478763e-03         10.9030866
## V298  4.923289e-04  2.231099e-03          2.306101e-03         11.6235601
## V299  3.971003e-03  1.413073e-03          1.970692e-03          8.0386903
## V300  1.776805e-03  1.384910e-03          1.281077e-03          6.4869617
## V301  2.151714e-03  9.142874e-04          1.314955e-03          3.3986985
## V302  1.875602e-03 -4.406667e-04          6.728634e-04          2.5443307
## V303  1.912189e-03  3.428426e-03          1.331013e-02         34.1853952
## V304  2.325638e-03  1.074886e-03          2.785949e-03          4.9227342
## V305  9.241144e-04  5.517451e-05          6.268288e-04          2.1414876
## V306 -5.481263e-05  2.938911e-04          7.115731e-04          1.5848774
## V307  8.461322e-04  4.757115e-04          1.297570e-03          1.6543460
## V308  3.434523e-04  6.228649e-06          1.187541e-03          2.0172808
## V309  1.581243e-03  8.221212e-04          1.006447e-03          1.3861299
## V310  6.015137e-05  1.273376e-03          3.095622e-04          1.4276467
## V311  2.053243e-03  2.937431e-03          1.368780e-02         46.5450290
## V312  2.404598e-04  5.178266e-04          7.017279e-04          2.1001454
## V313  7.378193e-04  3.057481e-04          1.355076e-03          2.0325919
## V314  1.000711e-03  1.050209e-03          4.541079e-04          1.4102526
## V315  4.309441e-03  3.719897e-03          1.503391e-02         59.5252833
## V316  5.140547e-04 -1.667702e-04          5.088195e-04          1.7246239
## V317  2.425556e-03  9.462231e-04          2.038377e-03          7.9095838
## V318  9.594592e-04  3.657700e-04          1.078606e-03          2.6283510
## V319  8.579103e-04  9.769693e-04          1.487765e-03          4.1263031
## V320  1.676485e-03 -1.224430e-04          2.862002e-04          1.8059487
## V321  1.031636e-03  9.637561e-04          5.292538e-04          1.4007061
## V322  9.238153e-04  1.904432e-03          9.683604e-04          1.7087158
## V323  8.277017e-04  1.841522e-03          4.085765e-04          1.5984176
## V324 -1.642440e-04  3.043167e-03          5.567333e-04          1.4312799
## V325  4.617712e-05 -3.622193e-04          6.432744e-04          3.7858899
## V326  1.219744e-03  1.088458e-03          1.434020e-03          4.5970103
## V327  1.005321e-03  1.597413e-03          3.440985e-04          1.5605396
## V328  5.076023e-04  2.307151e-03          3.567317e-04          1.2094869
## V329  9.329709e-04  0.000000e+00          1.189012e-03          2.8218377
## V330  4.280614e-04  6.772487e-04          5.272449e-04          1.5340995
## V331  1.537405e-03  1.779283e-03          1.908263e-03          4.0850340
## V332  1.155941e-04  4.246961e-05          6.762037e-04          2.3079529
## V333  7.940839e-04  1.099789e-03          2.950523e-03          5.3017023
## V334  9.544980e-04 -1.249413e-04          9.046398e-04          2.0651171
## V335  2.216420e-04  4.736415e-04          4.292722e-04          1.1784797
```

```
## V336  2.467525e-04 -2.222222e-05          6.132772e-04         1.2912725
## V337  6.428571e-05  1.884717e-04          1.403478e-03         1.4166949
## V338 -4.299211e-04 -3.471844e-04          2.690428e-04         1.1325785
## V339  7.955551e-04  2.491258e-03          1.268953e-03         4.7174256
## V340  1.448439e-03  5.956375e-04          3.701902e-03         6.1860938
## V341  1.290221e-03  6.269024e-04          3.630118e-04         1.5358139
## V342  1.067354e-05 -1.914139e-04          8.535147e-04         1.1489547
## V343  1.124360e-03  2.218202e-03          1.005586e-03         2.8859321
## V344  4.079511e-04  7.455348e-04          6.506762e-04         2.0783907
## V345  5.094563e-03  1.688270e-03          7.746031e-03        23.8464197
## V346  1.776407e-03  5.831926e-04          2.505414e-03         6.6291209
## V347  5.332529e-04  3.014103e-04          1.488829e-03         3.1922881
## V348  5.726137e-03  1.918313e-03          1.015919e-02        31.0088474
## V349  1.463421e-03 -3.743133e-04          2.994840e-03         9.2766527
## V350  1.789069e-03  8.278867e-06          1.620055e-03         2.6879764
## V351  2.438911e-03  1.326091e-03          6.060791e-03        18.1488078
## V352  9.228719e-04  9.553808e-04          2.390784e-03         7.0565547
## V353 -6.376263e-05  1.715848e-04          1.282929e-03         3.0110405
## V354  3.747857e-03  1.571948e-03          9.965397e-03        26.4967083
## V355  4.408800e-04  2.814010e-04          1.143943e-03         3.8825605
## V356  1.955638e-03  5.382857e-04          2.058468e-03         7.1602759
## V357  1.000240e-04  4.260073e-04          2.890678e-04         1.0173012
## V358 -6.837637e-04  5.291005e-05          2.650722e-04         1.0997310
## V359 -1.186613e-04 -3.556277e-04          8.453466e-04         1.1988856
## V360  1.363869e-03  1.125371e-03          6.519272e-03        18.8038933
## V361  6.416239e-03  3.769253e-03          1.776703e-02        49.8215158
## V362  1.448765e-03 -2.414876e-04          2.252358e-03         6.9363509
## V363  4.175370e-05  4.123598e-04          1.420222e-03         2.4237028
## V364  9.977849e-04 -5.648459e-04          1.455719e-03         4.3063736
## V365  2.119664e-03 -1.388380e-04          1.349858e-03         3.7779022
## V366  8.195499e-04 -5.214597e-04          8.271921e-04         2.4417964
## V367  3.053220e-03  5.174697e-04          9.198001e-03        25.3230599
## V368  1.985151e-03 -3.023729e-04          3.032536e-03         8.4658983
## V369  1.376956e-03 -3.542189e-06          2.435608e-03         7.3565828
## V370  1.110101e-03  4.503123e-04          1.343463e-03         8.9870228
## V371 -1.782082e-04 -1.905797e-04          6.805325e-04         1.5586497
## V372 -2.402859e-05  2.411761e-05          1.246632e-03         8.6213124
## V373  3.447195e-05 -1.831502e-05          7.134516e-04         1.7866572
## V374 -4.211142e-04 -1.605546e-04          2.689048e-04         1.7017324
## V375  5.113859e-04  5.456978e-04          7.593145e-04         2.5333915
## V376  1.633890e-04 -2.311828e-04          3.220817e-04         1.6613385
## V377  1.980518e-04  1.438211e-04          2.067005e-04         1.4689344
## V378  2.271898e-04  0.000000e+00          2.411898e-04         2.0905043
## V379  6.204536e-04 -5.847953e-06          1.808453e-04         1.5503158
## V380  2.882552e-04  8.994709e-05          1.523799e-04         1.7311154
## V381  1.243258e-04 -1.954244e-04          9.401307e-05         1.4224214
## V382  7.445518e-03  4.622470e-03          1.996441e-02        57.1691202
## V383  6.123028e-03  1.974693e-04          4.251447e-03        10.0211235
## V384  8.676294e-04 -3.185859e-04          5.260477e-04         1.9841915
## V385  2.915628e-04  1.250000e-04          5.496831e-04         1.9279983
```

```
## V386  6.852012e-04  3.174603e-04          6.516478e-04         1.4690333
## V387  1.642941e-03  1.506901e-04          7.076589e-04         1.7162352
## V388  1.163399e-03 -2.702478e-04          6.654020e-04         1.3875647
## V389  1.367707e-03  3.202390e-04          3.750726e-04         0.9887755
## V390  3.709172e-03  3.319893e-03          2.007491e-02        59.5453000
## V391  1.711088e-04 -5.712832e-04          2.076462e-03         6.7094386
## V392  5.656584e-04  2.666667e-04          1.257046e-03         2.1314695
## V393  3.960528e-04 -4.907407e-05          4.619176e-04         1.0987772
## V394  4.884659e-03  2.867559e-03          1.274828e-02        35.7275232
## V395  7.328890e-04  5.290966e-04          6.916268e-04         2.2146446
## V396  8.923857e-04 -5.488889e-04          1.298799e-03         5.2419805
## V397  1.755764e-03  4.634521e-04          8.989572e-04         2.4445584
## V398  1.213294e-03  5.181355e-04          1.678730e-03         3.1041252
## V399  1.960422e-03  6.251068e-05          1.008838e-03         2.1256572
## V400  6.507913e-04  3.284334e-04          3.243187e-04         1.3573385
## V401  1.911284e-03 -3.151227e-04          1.243691e-03         3.7174474
## V402  2.996525e-04 -3.004891e-04          6.935155e-04         1.1992741
## V403  5.754777e-05  5.311594e-04          3.171189e-04         1.1688964
## V404  1.498711e-03  6.693460e-04          4.387709e-03        11.3909820
## V405  1.747302e-03  9.071280e-04          2.130057e-03         3.5754615
## V406  9.160250e-04 -4.949806e-04          6.384357e-04         1.3324478
## V407  6.000485e-05  3.693468e-04          4.329193e-04         1.1462581
## V408  1.395229e-03  1.895993e-04          2.264715e-03         4.8850814
## V409  2.336532e-03 -7.405567e-04          1.615602e-03         1.8050581
## V410  4.390991e-04 -7.660554e-06          4.181425e-03         6.4769897
## V411 -1.363968e-04 -7.637681e-05          1.271743e-03         2.6548203
## V412  1.113493e-03  7.046232e-04          4.147501e-03         9.4443711
## V413  1.056853e-03 -3.322298e-04          1.792724e-03         3.1268887
## V414  1.865847e-05  4.638133e-04          1.011777e-03         1.4861213
## V415  1.166667e-04 -2.138596e-04          9.431307e-04         1.5324839
## V416  3.578066e-04 -1.045607e-04          1.470637e-03         1.3130746
## V417  5.890234e-04 -2.280702e-04          1.032978e-03         1.1657812
## V418  2.471205e-04  1.168350e-04          1.446768e-03         2.8855826
## V419  1.747209e-03  2.912458e-05          4.734151e-03         6.3921686
## V420  8.746550e-05  0.000000e+00          9.927145e-04         1.7361072
## V421  2.068116e-04 -5.856961e-04          1.133512e-03         1.6480518
## V422  9.404467e-04  2.836318e-04          2.163986e-03         5.4923553
## V423  1.605374e-04  9.392356e-04          1.536993e-03         3.1417255
## V424  1.730320e-03  8.505720e-04          3.166463e-03        15.0328790
## V425  1.037788e-03  8.578828e-04          9.886011e-04         3.3015347
## V426  3.750000e-04  1.317503e-03          1.070475e-03         3.2766283
## V427  3.438883e-03  8.629883e-04          4.819088e-03        17.2404006
## V428  3.227123e-04  1.445544e-03          3.438767e-03         5.1213892
## V429  8.366086e-04 -1.716638e-04          2.498342e-03         5.3270855
## V430  3.424321e-03  5.197729e-04          4.884932e-03        17.1130213
## V431  6.551574e-04  5.681529e-05          1.926186e-03         6.3256960
## V432 -9.471505e-05 -1.069565e-04          1.626915e-03         4.4682817
## V433  5.179880e-03  6.929735e-04          4.514611e-03        15.5085276
## V434 -4.697298e-05  1.847920e-03          4.018973e-03         5.8527506
## V435 -7.855443e-05  2.440562e-03          3.521046e-03         8.0710955
```

```
## V436  2.381597e-03 -5.342010e-04          2.787429e-04         1.5579185
## V437  2.954347e-04  2.526316e-05          2.613650e-04         1.0504251
## V438  2.156002e-04 -1.118408e-03          3.700725e-04         1.9502458
## V439  4.248797e-03  1.813083e-03          5.168910e-03        16.6015099
## V440  7.870568e-03  8.216122e-04          4.173292e-03        13.7784427
## V441  6.839507e-04  8.255777e-04          3.274689e-03         4.3173201
## V442 -1.161534e-04  1.339506e-03          2.558165e-03         5.8541493
## V443  4.070230e-04 -2.673664e-04          1.094711e-03         3.3997108
## V444  1.980909e-04 -6.603314e-05          6.734272e-04         2.0203700
## V445  7.206429e-04 -7.459056e-04          9.207514e-04         2.1373627
## V446  9.663346e-04  2.739130e-04          2.175943e-03         9.2888987
## V447 -1.310112e-04  9.811700e-04          4.259926e-04         2.9387446
## V448  1.217704e-03 -2.974298e-04          1.123734e-03         3.8955252
## V449  7.193196e-04  6.894754e-04          6.128736e-03        17.1802900
## V450  9.099713e-04  2.891354e-04          2.620293e-03         6.4633937
## V451  7.989715e-03  3.332528e-03          6.671887e-03        38.1852487
## V452  1.714836e-04  1.450137e-03          2.697728e-03        20.7850229
## V453  4.444444e-05 -5.704617e-04          4.908568e-04         2.5622899
## V454 -1.381611e-04  2.856522e-04          6.466590e-04         2.3965331
## V455  4.035727e-04  1.528495e-03          3.889048e-04         3.8831517
## V456  1.446848e-04  1.129786e-03          2.738088e-04         2.6414187
## V457  2.980747e-04  5.053193e-04          1.122704e-03         4.2949656
## V458  4.754063e-04 -2.129493e-04          1.219153e-03         4.3057570
## V459 -3.722972e-04 -8.973923e-05          1.151705e-03         7.1738732
## V460 -9.685535e-04  5.641113e-04          9.442323e-04         5.8032730
## V461  5.385371e-03  8.700089e-04          6.192583e-03        15.6816392
## V462  1.267295e-03  3.109533e-04          3.488239e-03         9.5616000
## V463  2.217508e-04 -7.537519e-04          8.002969e-04         2.8448603
## V464  1.193005e-03  1.703428e-04          7.223755e-04         2.3445589
## V465  3.812305e-04 -7.460272e-04          7.332394e-04         3.1475015
## V466  1.325018e-03  5.209203e-04          1.107334e-03         3.7302129
## V467  8.408143e-04  9.090909e-05          9.437528e-04         2.6223570
## V468  1.164276e-03  4.711485e-05          5.904427e-04         2.1434452
## V469  3.820783e-03  1.271154e-03          4.926687e-03        17.1882118
## V470  3.172478e-04  1.148207e-03          8.541310e-04         3.6746208
## V471  7.381516e-04  8.522577e-04          7.932389e-04         3.2810748
## V472  8.947918e-04  3.859430e-04          9.325142e-04         2.5472664
## V473  5.187107e-03  1.417004e-03          4.337323e-03        16.0292377
## V474  1.614099e-03  4.775079e-04          9.380846e-04         2.9448767
## V475  1.293849e-03  1.571431e-03          3.493238e-03         5.3390526
## V476  5.954185e-04  6.942480e-04          1.492962e-03         2.1046203
## V477 -2.849003e-05  9.523810e-06          7.614522e-04         2.2522718
## V478  1.692399e-04  1.231288e-04          5.855008e-04         2.6010332
## V479  1.876282e-04  7.871854e-04          4.139058e-04         1.5743748
## V480  1.050420e-04  4.972877e-04          4.758976e-04         1.5756499
## V481  4.211765e-04 -4.547758e-05          4.292245e-04         1.1082428
## V482  4.473304e-05  1.163191e-04          7.947249e-04         1.3276599
## V483  5.745150e-04  2.019496e-03          3.391878e-03         5.7941535
## V484  3.811780e-04  5.516400e-04          8.353178e-04         2.5402507
## V485  5.090370e-04 -1.971272e-04          8.845567e-04         1.7005828
```

```
## V486  8.108018e-04  6.891064e-04          4.314966e-04         1.1879272
## V487  9.720995e-04  2.955892e-03          2.188204e-03         5.4125278
## V488  0.000000e+00 -5.183945e-04          8.261449e-04         2.4622687
## V489  3.237814e-04  1.245290e-03          2.581747e-03         8.0612324
## V490  5.107793e-05  2.990629e-05          6.506653e-04         1.9338956
## V491  1.037215e-03  9.446227e-04          1.086008e-03         2.6544553
## V492  3.476581e-04 -7.847086e-05          7.401867e-04         2.8756436
## V493 -3.121637e-04 -4.001272e-05          6.754189e-04         2.2562868
## V494  6.001204e-04  9.665992e-05          4.367013e-04         1.9378047
## V495  7.143817e-04 -7.987845e-04          4.419542e-04         1.9656758
## V496 -1.406162e-04 -1.511387e-04          5.095547e-04         1.9040405
## V497  2.723716e-04  1.408505e-03          2.244726e-03         5.3254057
## V498 -5.872250e-05 -2.002849e-04          2.105861e-03         5.8198237
## V499  2.328976e-04  0.000000e+00          5.382846e-04         1.8381938
## V500 -9.500844e-05 -7.411318e-04          5.246311e-04         1.7795059
## V501  6.666223e-04  8.404282e-04          2.745242e-03         5.8852157
## V502 -2.138628e-04 -4.378821e-04          6.270262e-04         2.1203068
## V503  1.346160e-03  4.045391e-03          5.389218e-03        27.3234543
## V504  1.483516e-02  1.115995e-02          1.517022e-02        53.0804966
## V505  4.159194e-03  6.039486e-03          1.035755e-02        45.1685738
## V506  7.619330e-03  5.378294e-03          4.668751e-03        15.5459484
## V507 -2.541536e-04  3.986953e-03          1.593997e-04         1.4649680
## V508  5.644340e-04  4.269329e-03          5.487220e-03        28.7593062
## V509  9.234705e-03  5.786542e-03          1.301121e-02        46.0662290
## V510  9.294632e-04  1.083019e-03          1.631779e-03         7.8634729
## V511  4.202092e-04  5.547950e-04          1.136316e-03         6.0085546
## V512  1.411892e-02  5.591655e-03          3.956916e-03        10.8721450
## V513  5.687390e-03  1.779650e-03          2.244143e-03        12.9306083
## V514  7.733032e-03  2.453046e-03          7.512673e-04         4.2891874
## V515  2.741350e-03  3.130513e-04          5.121750e-04         2.6952267
## V516  6.006044e-04 -1.717495e-04          1.446465e-03         3.3494769
## V517  1.887902e-03  7.244179e-04          5.104804e-03        10.4391676
## V518  1.176266e-03  5.061570e-04          1.081050e-03         4.2245231
## V519  6.059740e-04  8.799172e-04          4.222209e-03         8.9390994
## V520  1.048382e-04  7.040537e-04          5.562681e-04         1.0350233
## V521  7.110736e-04  7.058033e-04          8.960804e-04         2.5338810
## V522  2.217840e-03  7.722438e-04          2.073549e-03         6.0258774
## V523  1.114026e-03  5.068052e-04          2.045399e-03         6.0550578
## V524  7.476725e-04 -6.343803e-05          3.626870e-04         1.0956878
## V525  5.702212e-04  1.040904e-04          1.399767e-03         2.8442929
## V526  1.206081e-04  1.946852e-03          2.066324e-04         1.5486222
## V527 -9.952637e-06 -1.619734e-04          7.783219e-04         2.4920907
## V528 -6.621290e-05 -4.757829e-04          5.599891e-04         2.2563309
## V529  2.901991e-04  2.994987e-04          3.596864e-04         1.6592996
## V530  5.982273e-04  6.209401e-04          3.917169e-04         1.6856237
## V531  3.408582e-04  2.351962e-04          4.652735e-04         1.8977788
## V532  7.688918e-04  3.925758e-04          5.630377e-04         2.1697427
## V533 -3.876457e-05 -3.970094e-04          2.828391e-04         1.2095396
## V534  6.294918e-05  2.597118e-04          5.520282e-04         1.9595763
## V535  6.220843e-04  1.010241e-03          6.654423e-04         1.8099861
```

```
## V536  2.021825e-04 -1.312665e-04         4.135146e-04        1.6329314
## V537  6.106382e-04 -1.935698e-04         5.872167e-04        1.2945619
## V538  3.419801e-04  8.093356e-07         1.473715e-03        5.2415646
## V539  9.899701e-04 -2.223570e-05         1.963971e-03        6.2748415
## V540  4.313465e-04  1.792555e-04         2.275691e-04        2.5770783
## V541  1.038127e-03 -3.202917e-04         1.425679e-04        2.2731576
## V542 -2.385762e-05  5.878788e-04         2.760003e-04        1.5589157
## V543  8.501375e-04  3.778259e-04         1.193698e-03        4.0904319
## V544 -1.609890e-04 -6.964984e-04         5.842416e-04        1.8828231
## V545  3.850740e-04  4.761353e-04         1.082164e-03        4.0819258
## V546  5.362523e-05  7.503608e-05         2.398860e-04        1.1208420
## V547  4.705184e-04  1.182436e-03         7.127079e-04        2.0243470
## V548  2.825773e-05  1.608706e-04         7.305204e-04        1.8751740
## V549 -9.888230e-06  2.323814e-04         4.162480e-04        1.3165148
## V550  6.598551e-04  9.090909e-05         5.768385e-04        0.9421238
## V551  5.715140e-04 -3.005602e-04         5.906188e-04        1.7807847
## V552 -3.626494e-04 -1.098590e-04         1.599194e-04        1.6438250
## V553  1.548387e-05 -2.704762e-04         4.825519e-04        2.8730406
## V554 -2.975580e-04  3.131313e-04         5.784862e-04        2.4216387
## V555  7.969159e-04  1.824021e-03         1.230225e-04        0.9304246
## V556  2.607355e-03  6.714505e-03         1.722591e-04        2.0529795
## V557  6.867515e-04  1.899914e-03         8.093665e-04        4.0745271
## V558  5.226025e-04 -4.516668e-04         3.058760e-04        3.5584844
## V559  5.872068e-03  1.582343e-02         4.197339e-02      187.6945036
## V560  1.972511e-03  1.200842e-02         2.837343e-02      157.2762535
## V561  3.358984e-04 -1.026611e-03         9.013751e-03       67.0842928

#We exported the above numbers we got from randomforest$importance into an ex
cel sheet which is being imported below
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
rm(Importance)

#We selected the Highest values for %IncMSE and set the bar at zero. Leaving
us with 353 predictors to create a random forest model

model_params = head(importance_sorted$Parameter,353)
predictors= paste(model_params, sep = "+")

model_expression = formula(paste("response~",paste(predictors,collapse = "+")
))
rm(predictors,model_params)

#Making training dataset with 353 important columns
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
model_params_353 = head(importance_sorted$Parameter,353)
predictors_353= paste(model_params_353, sep = "+")
X_train_353 = train_x_df[,predictors_353]
```

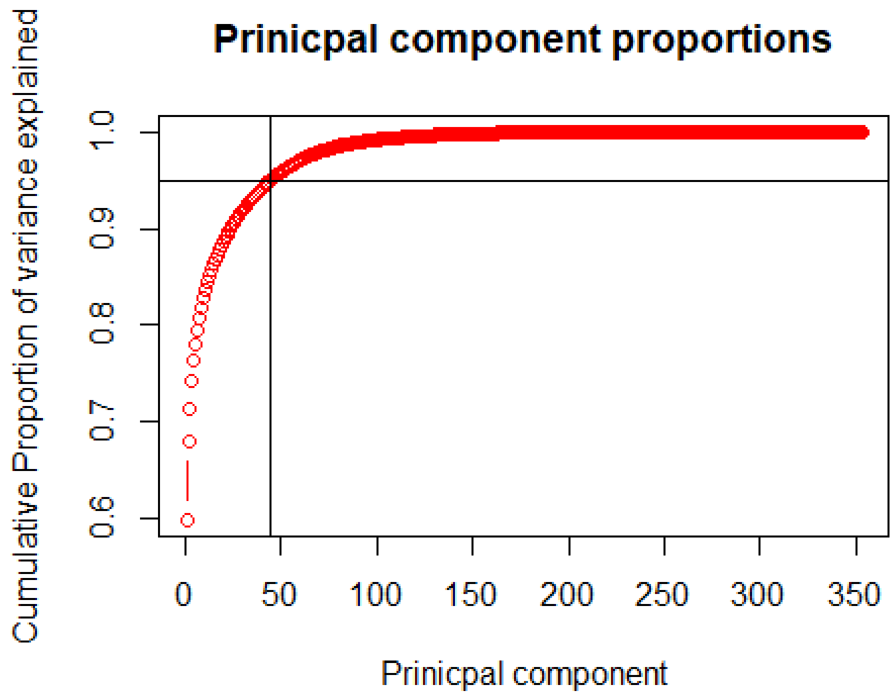
```r
train_353 = cbind(X_train_353,response = train_response_df$response)
write.csv(X_train_353,"train_353.csv")

#Making testing dataset with 353 important columns
Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
model_params_353_test = head(importance_sorted$Parameter,353)
predictors_353= paste(model_params_353_test, sep = "+")
X_test_353 = test_x_df[,predictors_353]
write.csv(X_test_353,"test_353.csv")

#removing unused variables
rm(Importance,importance_sorted)

#building a random forest model for top 353 predictors
hapt_randomForest_model_353 = randomForest(model_expression,data = train_df,
mtry=19, importance = TRUE)
summary(hapt_randomForest_model_353)

##                   Length Class  Mode
## call                  5  -none- call
## type                  1  -none- character
## predicted          7767  factor numeric
## err.rate           6500  -none- numeric
## confusion           156  -none- numeric
## votes             93204  matrix numeric
## oob.times          7767  -none- numeric
## classes              12  -none- character
## importance         4942  -none- numeric
## importanceSD       4589  -none- numeric
## localImportance       0  -none- NULL
## proximity             0  -none- NULL
## ntree                 1  -none- numeric
## mtry                  1  -none- numeric
## forest               14  -none- list
## y                  7767  factor numeric
## test                  0  -none- NULL
## inbag                 0  -none- NULL
## terms                 3  terms  call

#Making predictions on Test data using above Random Forest model
yhat.rf_model_353 = predict(hapt_randomForest_model_353, newdata = test_x_df)

#Checking prediction power of the model
classification = yhat.rf_model_353 == test_response_df$response
classification_rate = sum(classification)/length(classification)
classification_rate

## [1] 0.9038583

table(test_response_df$response, yhat.rf_model_353)
```

```
##     yhat.rf_model_353
##         1   2   3   4   5   6   7   8   9  10  11  12
##    1  483   6   7   0   0   0   0   0   0   0   0   0
##    2   44 420   7   0   0   0   0   0   0   0   0   0
##    3   18  47 355   0   0   0   0   0   0   0   0   0
##    4    0   0   0 442  65   0   1   0   0   0   0   0
##    5    0   0   0  51 504   0   1   0   0   0   0   0
##    6    0   1   0   0   0 544   0   0   0   0   0   0
##    7    0   1   0   2   0   0  18   1   0   0   1   0
##    8    0   0   0   0   0   0   1   9   0   0   0   0
##    9    0   0   0   0   0   0   0   0  24   0   8   0
##    10   0   0   0   0   0   0   0   0   0  16   1   8
##    11   3   0   0   1   0   0   3   0  11   1  30   0
##    12   0   0   0   0   0   0   0   0   0  11   3  13
```

```
#Implementing PCA on the dataset with 353 predictors selected after randomfor
est$importance
#train_353 <- read_csv("./train_353.csv")
attach(X_train_353)
pr.out = prcomp(X_train_353, scale = TRUE)
#names(pr.out)
pc.var = pr.out$sdev^2
pc.pvar = pc.var/sum(pc.var)
options(max.print = 100000)
#pr.out$center
#pr.out$scale
#pr.out$rotation
dim(pr.out$x)
```

```
## [1] 7767  353
```

```
#Plotting
plot(cumsum(pc.pvar), xlab = "Prinicpal component", ylab = "Cumulative Propor
tion of variance explained", type ='b', main = "Prinicpal component proportio
ns", col="red")
abline(h=0.95)
abline(v=45)
```




```
#first 45 principal components explain 95% of the variance in the response
#Selecting the PC1 to PC45 prinicpal compenents from original data set

train_45_pca = data.frame(response = train_df$response, pr.out$x)
train_45_pca = train_45_pca[,1:46]
train_45_pca$response = as.factor(train_45_pca$response)

#transforming test data for PCA
test.45_pca = predict(pr.out, newdata =X_test_353 )
test.45_pca = as.data.frame(test.45_pca)
test.45_pca = test.45_pca[,1:45]

##Implementing random forest with predictors from PCA

hapt_pca_rf_model = randomForest(response ~ .,data = train_45_pca, mtry=7, im
portance = TRUE)
summary(hapt_pca_rf_model)

##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted        7767  factor numeric
## err.rate         6500  -none- numeric
## confusion         156  -none- numeric
## votes           93204  matrix numeric
## oob.times        7767  -none- numeric
## classes            12  -none- character
```

```
## importance        630  -none- numeric
## importanceSD      585  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                7767  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call

#Making predictions on Test data using above Random Forest model
yhat.rf_pca = predict(hapt_pca_rf_model, newdata = test.45_pca)

#Checking prediction power of the model
classification = yhat.rf_pca == test_response_df$response
classification_rate = sum(classification)/length(classification)
classification_rate

## [1] 0.8621126

table(test_response_df$response, yhat.rf_pca)

##     yhat.rf_pca
##        1   2   3   4   5   6   7   8   9  10  11  12
##   1  459   2  35   0   0   0   0   0   0   0   0   0
##   2   33 427  11   0   0   0   0   0   0   0   0   0
##   3   50  49 321   0   0   0   0   0   0   0   0   0
##   4    0   0   0 391 117   0   0   0   0   0   0   0
##   5    0   1   0  49 506   0   0   0   0   0   0   0
##   6    0   0   0  24   0 521   0   0   0   0   0   0
##   7    0   6   0   2   1   0  13   0   0   0   1   0
##   8    0   0   0   1   0   0   1   8   0   0   0   0
##   9    0   0   0   0   0   0   0   0  23   0   9   0
##   10   0   1   0   0   0   0   0   0   0  16   0   8
##   11   2   6   0   2   0   0   1   0  12   0  26   0
##   12   0   0   0   0   0   0   1   1   0   9   1  15

##removing unused datasets with 561 predictors
rm(train_df,train_x_df)

##LDA using 45 predictors determined by PCA

hapt_lda_45 = lda(response ~ ., data = train_45_pca)

pred_lda_45 = predict(hapt_lda_45, newdata =  test.45_pca)

#Confusion matrix for the model
table(pred_lda_45$class, test_response_df$response)
```

```
##
##        1   2   3   4   5   6   7   8   9  10  11  12
##   1  475  32  29   0   0   0   0   0   0   0   1   0
##   2    8 439  52   0   0   0   1   0   0   0   0   0
##   3   13   0 339   0   0   0   0   0   0   0   0   0
##   4    0   0   0 328  51   4   0   0   0   0   0   0
##   5    0   0   0 177 504   1   0   0   0   0   1   0
##   6    0   0   0   0   0 540   0   0   0   0   1   0
##   7    0   0   0   2   1   0  21   1   3   0  12   1
##   8    0   0   0   0   0   0   0   9   0   1   0   2
##   9    0   0   0   1   0   0   1   0  21   0  12   0
##   10   0   0   0   0   0   0   0   0   0  16   0   7
##   11   0   0   0   0   0   0   0   0   8   0  22   0
##   12   0   0   0   0   0   0   0   0   0   8   0  17
```

```
#Accuracy of the model
1- mean(pred_lda_45$class != test_response_df$response)
```

```
## [1] 0.8636939
```

```
##LDA using 353 predictors determined by randomforest$importance
hapt_lda_353 = lda(response ~ ., data = train_353)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
pred_lda_353 = predict(hapt_lda_353, newdata =  test_x_df)

#Confusion matrix for the model
table(pred_lda_353$class, test_response_df$response)
```

```
##
##        1   2   3   4   5   6   7   8   9  10  11  12
##   1  492   9   1   0   0   0   0   0   0   0   2   0
##   2    4 462  17   0   0   0   1   0   0   0   1   0
##   3    0   0 402   0   0   0   0   0   0   0   0   0
##   4    0   0   0 390  29   0   1   0   0   0   2   0
##   5    0   0   0 115 527   0   0   0   0   0   0   0
##   6    0   0   0   0   0 545   0   0   0   0   1   0
##   7    0   0   0   3   0   0  21   0   0   1   8   1
##   8    0   0   0   0   0   0   0  10   0   0   0   0
##   9    0   0   0   0   0   0   0   0  25   0   8   0
##   10   0   0   0   0   0   0   0   0   0  13   0   3
##   11   0   0   0   0   0   0   0   0   7   0  27   1
##   12   0   0   0   0   0   0   0   0   0  11   0  22
```

```
#Accuracy of the model
1- mean(pred_lda_353$class != test_response_df$response)
```

```
## [1] 0.9285262
```

```
##QDA using 45 predictors determined by PCA
#hapt_qda_45 = qda(response ~ ., data = train_45_pca)
```

```r
#hapt_qda_353 = qda(response ~ ., data = train_353)

#QDA could not be performed on the training data set as some classes like 10,
11,12 did not have enough data points for QDA to train the model

##K = 10

##KNN using 45 predictors determined by PCA
library(class)
train_45_pca_forKNN = train_45_pca[2:ncol(train_45_pca)]


pred_knn_45 = knn(train_45_pca_forKNN, test.45_pca, train_45_pca$response, k=
10)

#Confusion matrix for the model
table(pred_knn_45, test_response_df$response)

##
## pred_knn_45   1   2   3   4   5   6   7   8   9  10  11  12
##          1  468  45  53   0   0   0   0   0   0   0   0   0
##          2    8 418  56   2   1   2   5   0   0   0   5   0
##          3   20   7 311   0   0   0   0   0   0   0   1   0
##          4    0   0   0 353  45  14   2   0   0   0   2   0
##          5    0   0   0 153 510  11   0   0   0   0   0   0
##          6    0   0   0   0   0 518   0   0   0   0   1   0
##          7    0   1   0   0   0   0  15   1   0   1   1   1
##          8    0   0   0   0   0   0   0   9   0   0   0   1
##          9    0   0   0   0   0   0   0   0  25   0  21   0
##          10   0   0   0   0   0   0   0   0   0  15   0  13
##          11   0   0   0   0   0   0   1   0   7   0  18   1
##          12   0   0   0   0   0   0   0   0   0   9   0  11

#Accuracy of the model
1- mean(pred_knn_45 != test_response_df$response)

## [1] 0.8447185

##KNN using 353 predictors determined by randomforest$importance
pred_knn_353 = knn(X_train_353, X_test_353 , train_response_df$response, k=10
)

#Confusion matrix for the model
table(pred_knn_353, test_response_df$response)

##
## pred_knn_353   1   2   3   4   5   6   7   8   9  10  11  12
##            1  478  38  27   0   0   0   0   0   0   0   1   0
##            2    5 429  51   3   0   1   3   0   0   0   5   0
```

```
##           3   13    4 342    0    0    0    0    0    0    0    0    0
##           4    0    0    0 362   39    0    2    0    0    0    1    0
##           5    0    0    0 143 517    0    0    0    0    0    0    0
##           6    0    0    0    0    0 544    0    0    0    0    1    0
##           7    0    0    0    0    0    0   17    1    0    0    3    0
##           8    0    0    0    0    0    0    0    9    0    0    0    0
##           9    0    0    0    0    0    0    1    0   29    1   20    2
##           10   0    0    0    0    0    0    0    0    0   19    0    7
##           11   0    0    0    0    0    0    0    0    3    0   18    2
##           12   0    0    0    0    0    0    0    0    0    5    0   16
```

```
#Accuracy of the model
1- mean(pred_knn_353 != test_response_df$response)
```

```
## [1] 0.8791904
```

```
## K = 20
##KNN using 45 predictors determined by PCA
pred_knn_45 = knn(train_45_pca_forKNN, test.45_pca, train_45_pca$response, k=
20)

#Confusion matrix for the model
table(pred_knn_45, test_response_df$response)
```

```
##
## pred_knn_45   1   2   3   4   5   6   7   8   9  10  11  12
##           1  475  47  60   0   0   0   0   0   0   0   0   0
##           2    2 417  54   2   0   2   6   0   0   1   6   0
##           3   19   7 306   0   0   0   0   0   0   0   1   0
##           4    0   0   0 335  29  15   2   1   0   0   2   0
##           5    0   0   0 171 527  11   0   0   0   0   0   0
##           6    0   0   0   0   0 517   0   0   0   0   1   0
##           7    0   0   0   0   0   0  14   2   0   0   0   1
##           8    0   0   0   0   0   0   0   7   0   0   0   1
##           9    0   0   0   0   0   0   0   0  23   0  21   0
##           10   0   0   0   0   0   0   0   0   0  18   0  11
##           11   0   0   0   0   0   0   1   0   9   0  18   2
##           12   0   0   0   0   0   0   0   0   0   6   0  12
```

```
#Accuracy of the model
1- mean(pred_knn_45 != test_response_df$response)
```

```
## [1] 0.844086
```

```
##KNN using 353 predictors determined by randomforest$importance
pred_knn_353 = knn(X_train_353, X_test_353 , train_response_df$response, k=20
)

#Confusion matrix for the model
table(pred_knn_353, test_response_df$response)
```

```
##
## pred_knn_353   1   2   3   4   5   6   7   8   9  10  11  12
##           1  480  46  30   0   0   0   1   0   0   0   2   1
##           2    3 422  52   3   1   1   2   0   0   0   6   0
##           3   13   3 338   0   0   0   0   0   0   0   0   0
##           4    0   0   0 353  35   0   2   0   0   0   1   0
##           5    0   0   0 152 520   0   0   0   0   0   0   0
##           6    0   0   0   0   0 544   0   0   0   0   0   0
##           7    0   0   0   0   0   0  17   2   0   0   1   0
##           8    0   0   0   0   0   0   0   8   0   0   0   0
##           9    0   0   0   0   0   0   1   0  28   1  19   1
##           10   0   0   0   0   0   0   0   0   0  12   0   7
##           11   0   0   0   0   0   0   0   0   4   0  20   2
##           12   0   0   0   0   0   0   0   0   0  12   0  16

#Accuracy of the model
1- mean(pred_knn_353 != test_response_df$response)

## [1] 0.8722328

## K = 50
##KNN using 45 predictors determined by PCA
pred_knn_45 = knn(train_45_pca_forKNN, test.45_pca, train_45_pca$response, k=
50)

#Confusion matrix for the model
table(pred_knn_45, test_response_df$response)

##
## pred_knn_45   1   2   3   4   5   6   7   8   9  10  11  12
##          1  484  43  69   0   0   0   0   0   0   0   0   0
##          2    0 423  61   2   3   2  11   0   1   1  10   1
##          3   12   5 290   0   0   0   0   0   0   0   1   0
##          4    0   0   0 311  34  16   3   1   0   0   1   0
##          5    0   0   0 195 519  11   0   0   0   0   1   0
##          6    0   0   0   0   0 516   0   0   0   0   1   0
##          7    0   0   0   0   0   0   8   4   0   0   1   0
##          8    0   0   0   0   0   0   0   5   0   0   0   1
##          9    0   0   0   0   0   0   0   0  26   0  20   1
##          10   0   0   0   0   0   0   0   0   0  16   0   8
##          11   0   0   0   0   0   0   1   0   5   0  14   1
##          12   0   0   0   0   0   0   0   0   0   8   0  15

#Accuracy of the model
1- mean(pred_knn_45 != test_response_df$response)

## [1] 0.8308033

##KNN using 353 predictors determined by randomforest$importance
pred_knn_353 = knn(X_train_353, X_test_353 , train_response_df$response, k=50
)
```

```r
#Confusion matrix for the model
table(pred_knn_353, test_response_df$response)

##
## pred_knn_353   1   2   3   4   5   6   7   8   9  10  11  12
##           1  488  41  34   0   0   0   1   0   0   0   2   1
##           2    1 428  59   4   1   1  10   1   0   0  11   0
##           3    7   2 327   0   0   0   0   0   0   0   0   0
##           4    0   0   0 331  34   0   2   1   0   0   0   0
##           5    0   0   0 173 521   0   0   0   0   0   1   0
##           6    0   0   0   0   0 544   0   0   0   0   0   0
##           7    0   0   0   0   0   0  10   4   0   0   0   0
##           8    0   0   0   0   0   0   0   4   0   0   0   0
##           9    0   0   0   0   0   0   0   0  28   1  21   1
##           10   0   0   0   0   0   0   0   0   0  17   0  12
##           11   0   0   0   0   0   0   0   0   4   0  14   2
##           12   0   0   0   0   0   0   0   0   0   7   0  11

#Accuracy of the model
1- mean(pred_knn_353 != test_response_df$response)

## [1] 0.8611638

## K = 100
##KNN using 45 predictors determined by PCA
pred_knn_45 = knn(train_45_pca_forKNN, test.45_pca, train_45_pca$response, k=
100)

#Confusion matrix for the model
table(pred_knn_45, test_response_df$response)

##
## pred_knn_45   1   2   3   4   5   6   7   8   9  10  11  12
##          1  485  32  74   0   1   0   0   0   0   0   0   0
##          2    0 435  61   3   2   1  16   1   2   1  19   2
##          3   11   4 285   0   0   0   0   0   0   0   0   0
##          4    0   0   0 294  30  16   3   1   0   0   1   0
##          5    0   0   0 211 523  19   0   0   0   0   1   0
##          6    0   0   0   0   0 509   0   0   0   1   0   1
##          7    0   0   0   0   0   0   2   7   1   0   0   0
##          8    0   0   0   0   0   0   0   1   0   0   0   0
##          9    0   0   0   0   0   0   1   0  26   0  23   0
##          10   0   0   0   0   0   0   0   0   0  22   0  19
##          11   0   0   0   0   0   0   1   0   3   0   5   2
##          12   0   0   0   0   0   0   0   0   0   1   0   3

#Accuracy of the model
1- mean(pred_knn_45 != test_response_df$response)

## [1] 0.8191018
```

```
##KNN using 353 predictors determined by randomforest$importance
pred_knn_353 = knn(X_train_353, X_test_353 , train_response_df$response, k=10
0)

#Confusion matrix for the model
table(pred_knn_353, test_response_df$response)

##
## pred_knn_353   1   2   3   4   5   6   7   8   9  10  11  12
##           1  492  42  52   0   0   0   1   0   0   0   2   0
##           2    0 427  60   3   1   1  16   1   2   1  18   2
##           3    4   2 308   0   0   0   0   0   0   0   0   0
##           4    0   0   0 328  42   0   3   1   0   0   0   0
##           5    0   0   0 177 513   0   0   0   0   0   1   0
##           6    0   0   0   0   0 544   0   0   0   0   0   0
##           7    0   0   0   0   0   0   2   5   0   0   0   0
##           8    0   0   0   0   0   0   0   1   0   0   0   0
##           9    0   0   0   0   0   0   1   1  26   0  18   2
##           10   0   0   0   0   0   0   0   0   0  22   0  18
##           11   0   0   0   0   0   0   0   1   4   0  10   2
##           12   0   0   0   0   0   0   0   0   0   2   0   3

#Accuracy of the model
1- mean(pred_knn_353 != test_response_df$response)

## [1] 0.8462998

knn.final.50 = rep(0,50)
for(i in 1:10){
  set.seed(i)
knn.error.50 = rep(0,50)
for(j in 1:50){
  knn.pred = knn(X_train_353, X_test_353 , train_response_df$response, k=j)
  knn.error = mean(knn.pred != test_response_df$response)
  knn.error.50[j] = knn.error
}
knn.final.50 = knn.final.50 + knn.error.50
}
knn.final.error.50 = knn.final.50/10
knn.final.error.50

##  [1] 0.1423150 0.1441809 0.1260278 0.1276091 0.1254902 0.1259962 0.1227704
##  [8] 0.1237192 0.1206515 0.1216319 0.1223276 0.1222960 0.1225806 0.1237192
## [15] 0.1254269 0.1257748 0.1242568 0.1247628 0.1248577 0.1253953 0.1245731
## [22] 0.1259646 0.1261227 0.1291588 0.1298545 0.1319102 0.1329222 0.1303605
## [29] 0.1305819 0.1314991 0.1315939 0.1323213 0.1326692 0.1318469 0.1311195
## [36] 0.1327008 0.1325743 0.1332385 0.1321948 0.1324794 0.1331752 0.1332701
## [43] 0.1336812 0.1342505 0.1352941 0.1349462 0.1357369 0.1353257 0.1350727
## [50] 0.1372233

#K = 9 gives 0.1206831
```

```
##SVM using 45 predictors determined by PCA

hapt_svm_45 = svm(response ~ ., data = train_45_pca)

pred_svm_45 = predict(hapt_svm_45, newdata =  test.45_pca)

#Confusion matrix for the model
table(pred_svm_45, test_response_df$response)

## 
## pred_svm_45   1   2   3   4   5   6   7   8   9  10  11  12
##          1  457  52   9   0   0   0   1   0   0   0   1   0
##          2    4 411  32   1   0   0   2   0   0   1   0   0
##          3   35   8 379   1   1   2   2   1   6   3   6   9
##          4    0   0   0 397  56   0   1   0   1   0   2   0
##          5    0   0   0 106 499   1   1   0   0   0   0   0
##          6    0   0   0   2   0 542   0   0   0   0   2   0
##          7    0   0   0   0   0   0  16   0   0   0   2   1
##          8    0   0   0   0   0   0   0   9   0   0   0   0
##          9    0   0   0   0   0   0   0   0  21   0  10   0
##          10   0   0   0   0   0   0   0   0   0  17   0   8
##          11   0   0   0   1   0   0   0   0   4   0  26   0
##          12   0   0   0   0   0   0   0   0   0   4   0   9

#Accuracy of the model
1- mean(pred_svm_45 != test_response_df$response)

## [1] 0.8801392

##SVM using 353 predictors determined by randomforest$importance
hapt_svm_353 = svm(response ~ ., data = train_353)

pred_svm_353 = predict(hapt_svm_353, newdata =  test_x_df)

#Confusion matrix for the model
table(pred_svm_353, test_response_df$response)

## 
## pred_svm_353   1   2   3   4   5   6   7   8   9  10  11  12
##           1  481  25  10   0   0   0   0   0   0   0   2   0
##           2    8 444  37   0   0   0   1   0   0   0   1   0
##           3    7   1 373   0   0   0   0   1   1   0   0   1
##           4    0   0   0 401  47   0   2   0   0   0   2   0
##           5    0   0   0 105 509   0   0   0   0   0   0   0
##           6    0   0   0   0   0 544   0   0   0   0   1   0
##           7    0   1   0   2   0   0  20   0   0   1   0   1
##           8    0   0   0   0   0   0   0   9   0   0   0   0
##           9    0   0   0   0   0   0   0   0  27   0  11   0
##           10   0   0   0   0   0   0   0   0   0  18   0   8
##           11   0   0   0   0   0   1   0   0   4   0  32   1
##           12   0   0   0   0   0   0   0   0   0   6   0  16
```

```r
#Accuracy of the model
1- mean(pred_svm_353 != test_response_df$response)

## [1] 0.9089184



#Creating dataset for Neural Network

#loading training sets
training_x_nn <- read_csv("./train_353.csv")

## Parsed with column specification:
## cols(
##   .default = col_double()
## )

## See spec(...) for full column specifications.

train_x_df_nn = as.data.frame(training_x_nn)

training_response_nn <- read.table("./Y_train.txt")
train_response_df_nn = as.data.frame(training_response_nn)
names(train_response_df_nn) = "response"
train_response_df_nn$response = as.factor(train_response_df_nn$response)

rm(training_x_nn,training_response_nn)

#loading testing data sets
testing_x_nn <- read_csv("./test_353.csv")

## Parsed with column specification:
## cols(
##   .default = col_double()
## )
## See spec(...) for full column specifications.

test_x_df_nn = as.data.frame(testing_x_nn)

testing_response_nn <- read.table("./Y_test.txt")
test_response_df_nn = as.data.frame(testing_response_nn)
names(test_response_df_nn) = "response"
test_response_df_nn$response = as.factor(test_response_df_nn$response)

#removing non-transformed data
rm(testing_x_nn,testing_response_nn)

#Scaling
maxs <- apply(train_x_df_nn, 2, max)
mins <- apply(train_x_df_nn, 2, min)
```

```r
#Scaling training data
scaled_train_x <- as.data.frame(scale(train_x_df_nn, center = mins, scale = m
axs - mins))

#Scaling testing data
scaled_test_x <- as.data.frame(scale(test_x_df_nn, center = mins, scale = max
s - mins))

#Binarizing the categorical output
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "1")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "2")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "3")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "4")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "5")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "6")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "7")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "8")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "9")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "10")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "11")
scaled_train_x = cbind(scaled_train_x,train_response_df_nn$response == "12")
names(scaled_train_x)[354:365] = c('R1','R2','R3','R4','R5','R6','R7','R8','R
9','R10','R11','R12')


Importance <- read_excel("./Importance.xlsx")
importance_sorted = Importance[order(-Importance$`%IncMSE`),]
rm(Importance)
```

```r
#We selected the Highest values for %IncMSE and set the bar at zero.
Leaving us with 353 predictors to create a random forest model

model_params_nn = head(importance_sorted$Parameter,353)
predictors_nn <-
as.formula(paste("R1+R2+R3+R4+R5+R6+R7+R8+R9+R10+R11+R12~",
paste(model_params_nn[!model_params_nn %in%
"R1+R2+R3+R4+R5+R6+R7+R8+R9+R10+R11+R12"], collapse = " + ")))
```

```r
#Creating neural network model

#Taking a recommended value of 2/3rd of number of predictors for the number

of neurons, hence 235.


nn = neuralnet(predictors_nn,data = scaled_train_x, hidden = c(235), act.fct
= "logistic", linear.output = F)
```

```
#Predicting using the neural network model

comp = compute(nn, scaled_test_x)
pred.weights = comp$net.result
idx = apply(pred.weights, 1, which.max)
pred = c('R1','R2','R3','R4','R5','R6','R7','R8','R9','R10','R11','R12')[idx]


test_response_df_nn$response = as.factor(test_response_df_nn$response)

test_response_df_nn$response = revalue(test_response_df_nn$response, c("1" =
"R1","2" = "R2","3" = "R3","4" = "R4","5" = "R5","6" = "R6","7" = "R7","8" =
"R8","9" = "R9","10" = "R10","11" = "R11","12" = "R12"))


#Confusion matrix for the model
table(cbind(pred), test_response_df_nn$response)

##
##        R1  R2  R3  R4  R5  R6  R7  R8  R9 R10 R11 R12
##   R1  482  21  17   0   0   0   0   0   0   0   2   0
##   R10   0   0   0   0   0   0   0   0   1  18   1   7
##   R11   0   0   0   0   0   0   1   0   6   0  31   2
##   R12   0   0   0   0   0   0   0   0   0   7   0  17
##   R2    4 447  24   0   0   1   0   0   0   0   3   0
##   R3    9   1 379   0   0   0   0   0   0   0   0   0
##   R4    1   2   0 436  25   0   0   0   0   0   1   0
##   R5    0   0   0  68 528   0   2   0   0   0   0   0
##   R6    0   0   0   1   0 544   0   0   0   0   0   0
##   R7    0   0   0   3   3   0  19   0   0   0   1   0
##   R8    0   0   0   0   0   0   0  10   0   0   0   0
##   R9    0   0   0   0   0   0   1   0  25   0  10   1

#Accuracy of the model
1- mean(cbind(pred) != test_response_df_nn$response)

## [1] 0.9285262492
```

Markdown File for scripts

Project-Final.Rmd

Data Files and Execution scripts for easy execution by TA

Group 6- Submission.zip