✖

Quiz 2

5 questions

<div style="border:1px solid #ccc; display:inline-block; padding:10px; text-align:center;">

1

point

</div>

1.

Load the Alzheimer's disease data using the commands:

```
1   library(AppliedPredictiveModeling)
2   data(AlzheimerDisease)
```

Which of the following commands will create non-overlapping training and test sets with about 50% of the observations assigned to each?

○
```
1   adData = data.frame(predictors)
2   trainIndex = createDataPartition(diagnosis,p=0.5,list=FALSE)
3   training = adData[trainIndex,]
4   testing = adData[-trainIndex,]
```

○
```
1   adData = data.frame(diagnosis,predictors)
2   trainIndex = createDataPartition(diagnosis, p = 0.50)
3   training = adData[trainIndex,]
4   testing = adData[-trainIndex,]
```

○
```
1   adData = data.frame(diagnosis,predictors)
2   trainIndex = createDataPartition(diagnosis, p = 0.50,list=FALSE)
3   training = adData[trainIndex,]
4   testing = adData[-trainIndex,]
```

○
```
1   adData = data.frame(diagnosis,predictors)
2   train = createDataPartition(diagnosis, p = 0.50,list=FALSE)
3   test = createDataPartition(diagnosis, p = 0.50,list=FALSE)
4   
```

<div style="border:1px solid #ccc; display:inline-block; padding:10px; text-align:center;">

1

point

</div>

2.

Load the cement data using the commands:

```
1   library(AppliedPredictiveModeling)
2   data(concrete)
3   library(caret)
4   set.seed(1000)
5   inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
6   training = mixtures[ inTrain,]
7   testing = mixtures[-inTrain,]
```

Make a plot of the outcome (CompressiveStrength) versus the index of the samples. Color by each of the variables in the data set (you may find the cut2() function in the Hmisc package useful for turning continuous covariates into factors). What do you notice in these plots?

○   There is a non-random pattern in the plot of the outcome versus index that is perfectly explained by the FlyAsh variable so there may be a variable missing.

○   There is a non-random pattern in the plot of the outcome versus index that is perfectly explained by the Age variable so there may be a variable missing.

○   There is a non-random pattern in the plot of the outcome versus index that does not appear to be perfectly explained by any predictor suggesting a variable may be missing.

○   There is a non-random pattern in the plot of the outcome versus index.

> 1
> point

3.

Load the cement data using the commands:

```
1   library(AppliedPredictiveModeling)
2   data(concrete)
3   library(caret)
4   set.seed(1000)
5   inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
6   training = mixtures[ inTrain,]
7   testing = mixtures[-inTrain,]
```

Make a histogram and confirm the SuperPlasticizer variable is skewed. Normally you might use the log transform to try to make the data more symmetric. Why would that be a poor choice for this variable?

○

The SuperPlasticizer data include negative values so the log transform can not be performed.

○ The log transform does not reduce the skewness of the non-zero values of SuperPlasticizer

○ The log transform is not a monotone transformation of the data.

○ There are a large number of values that are the same and even if you took the log(SuperPlasticizer + 1) they would still all be identical so the distribution would not be symmetric.

1
point

4.

Load the Alzheimer's disease data using the commands:

```
1   library(caret)
2   library(AppliedPredictiveModeling)
3   set.seed(3433)data(AlzheimerDisease)
4   adData = data.frame(diagnosis,predictors)
5   inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]training = adData[
      inTrain,]
6   testing = adData[-inTrain,]
```

Find all the predictor variables in the training set that begin with IL. Perform principal components on these variables with the preProcess() function from the caret package. Calculate the number of principal components needed to capture 80% of the variance. How many are there?

○ 9

○ 10

○ 7

○ 11

1
point

5.

Load the Alzheimer's disease data using the commands:

```
1   library(caret)
2   library(AppliedPredictiveModeling)
3   set.seed(3433)data(AlzheimerDisease)
4   adData = data.frame(diagnosis,predictors)
5   inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]training = adData[
      inTrain,]
6   testing = adData[-inTrain,]
```

Create a training data set consisting of only the predictors with variable names beginning with IL and the diagnosis. Build two predictive models, one using the predictors as they are and one using PCA with principal components explaining 80% of the variance in the predictors. Use method="glm" in the train function.

What is the accuracy of each method in the test set? Which is more accurate?

○   Non-PCA Accuracy: 0.72

      PCA Accuracy: 0.71

○   Non-PCA Accuracy: 0.65

      PCA Accuracy: 0.72

○   Non-PCA Accuracy: 0.72

      PCA Accuracy: 0.65

○   Non-PCA Accuracy: 0.75

      PCA Accuracy: 0.71

Submit Quiz