Feedback — Quiz 2

You submitted this quiz on **Sun 25 Jan 2015 12:51 AM PST**. You got a score of **15.00** out of **15.00**. However, you will not get credit for it, since it was submitted past the deadline.

Question 1

Load the Alzheimer's disease data using the commands:

```
library(AppliedPredictiveModeling)
library(caret)
data(AlzheimerDisease)
```

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

Your Answer		Score	Explanation
○ adData = data.frame(diagnosis,predictors) trainIndex = createDataPartition(diagnosis,p=0.5,list=FALSE) training = adData[trainIndex,] testing = adData[trainIndex,]			
○ adData = data.frame(predictors) trainIndex = createDataPartition(diagnosis,p=0.5,list=FALSE) training = adData[trainIndex,] testing = adData[-trainIndex,]			
◉ adData = data.frame(diagnosis,predictors) trainIndex = createDataPartition(diagnosis, p = 0.50,list=FALSE) training = adData[trainIndex,] testing = adData[-trainIndex,]	✔	3.00	
○ adData = data.frame(diagnosis,predictors) trainIndex = createDataPartition(diagnosis,p=0.5,list=FALSE) training = adData[trainIndex,] testing = adData[trainIndex,]			

Total   3.00 /
                                                           3.00

Question 2

Load the cement data using the commands:

```
library(AppliedPredictiveModeling)
data(concrete)
library(caret)
set.seed(975)
inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
training = mixtures[ inTrain,]
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?

Your Answer	Score	Explanation
○ The outcome variable is highly correlated with FlyAsh.		
○ The data show a step like pattern that is perfectly explained by the Age variable so there may be a variable missing.		
○ There is a step-like pattern in the plot of outcome versus index in the training set.		
⦿ There is a step-like pattern in the plot of outcome versus index in the training set that isn't explained by any of the predictor variables so there may be a variable missing.	✔ 3.00	
Total	3.00 / 3.00	

Question 3

Load the cement data using the commands:

```
library(AppliedPredictiveModeling)
data(concrete)
library(caret)
```

```
set.seed(975)
inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
training = mixtures[ inTrain,]
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?

Your Answer		Score	Explanation
⊙ There are values of zero so when you take the log() transform those values will be -Inf.	✔	3.00	
○ The log transform does not reduce the skewness of the non-zero values of SuperPlasticizer			
○ The log transform produces negative values which can not be used by some classifiers.			
○ The log transform is not a monotone transformation of the data.			
Total		3.00 / 3.00	

Question 4

Load the Alzheimer's disease data using the commands:

```
library(caret)
library(AppliedPredictiveModeling)
set.seed(3433)
data(AlzheimerDisease)
adData = data.frame(diagnosis,predictors)
inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training = adData[ inTrain,]
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 90% of the variance. How many are there?

Your Answer	Score	Explanation
○ 7		

○ 9 ✔ 3.00

○ 10

○ 5

Total 3.00 / 3.00

Question 5

Load the Alzheimer's disease data using the commands:

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

Your Answer		Score	Explanation
○ Non-PCA Accuracy: 0.65 PCA Accuracy: 0.72	✔	3.00	
○ Non-PCA Accuracy: 0.72 PCA Accuracy: 0.65			
○ Non-PCA Accuracy: 0.75 PCA Accuracy: 0.71			
○ Non-PCA Accuracy: 0.74 PCA Accuracy: 0.74			
Total		3.00 / 3.00	