The "Data Science" Specialization Learn More ×

Feedback — Quiz 2

Help Center

You submitted this quiz on **Sun 11 Oct 2015 11:38 AM CEST**. You got a score of **12.00** out of **12.00**.

For this quiz we will be using several R packages. R package versions change over time, the right answers have been checked using the following versions of the packages.

AppliedPredictiveModeling: v1.1.6
caret: v6.0.47

If you aren't using these versions of the packages, your answers may not exactly match the right answer, but hopefully should be close.

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(diagnosis,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 = 0.50) 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(1000)
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 is not a monotone transformation of the data.	
◯ 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	
Total	3.00 / 3.00

Question 3

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 80% of the variance. How many are there?

Your Answer		Score	Explanation
(•) 7	✔	3.00	
( ) 12			
( ) 9			
( ) 11			
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,]
```

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: 0.91 PCA Accuracy: 0.93			
( ) Non-PCA Accuracy: 0.72 PCA Accuracy: 0.71			
( ) Non-PCA Accuracy: 0.75 PCA Accuracy: 0.71			
Total		3.00 / 3.00	