Quizzes due May 9, 2021 07:01 +03

Question: Conditional Expectations

2/2 points (graded)

The `heights` dataset from the **dslabs** package (available from CRAN) contains self-reported heights (in inches) for male and female students from three Harvard Biostatistics classes:

```
# install.packages("dslabs")    if needed
library(dslabs)
data(heights)
head(heights)
```

```
##        sex height
## 1    Male     75
## 2    Male     70
## 3    Male     68
## 4    Male     74
## 5    Male     61
## 6 Female     65
```

For simplicity, round heights to the nearest inch:

```
heights$height <- round(heights$height)
```

Treat these data as data for the whole population.

Calculate the conditional probability that a person 67 inches tall is female.

0.1747573

✓ **Answer:** 0.1747573

0.1747573

Calculate the conditional probability that a person is female for the vector of heights `hts = 60:80`. Make a plot of this conditional probability versus `hts`. Suppose you predict female for any height for which the conditional probability of being female $\mathrm{E}\left(Y = \text{Female}|X = x\right)$ is > 0.5. What is the maximum height for which you predict a person is female?

64

✓ **Answer:** 64

64

Explanation

Part A:

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

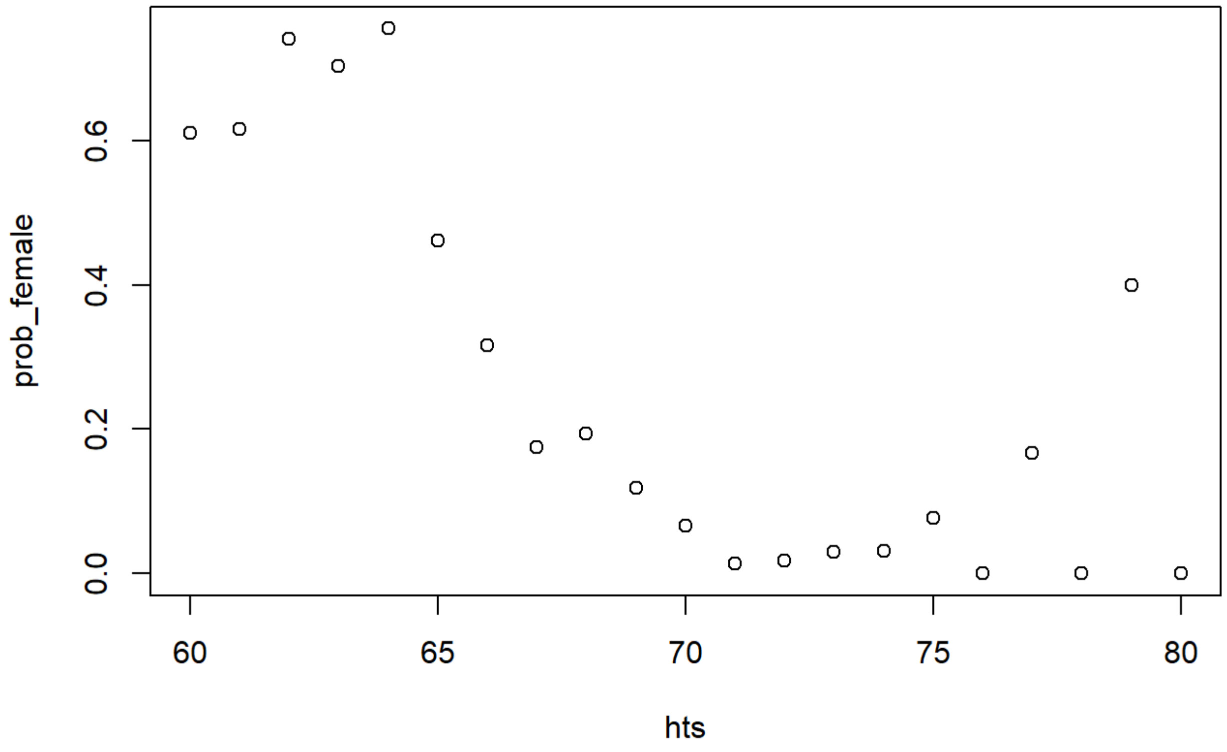
```
## -- Conflicts ------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
heights %>%
    filter(height == 67) %>%
    summarize(prob_female = mean(sex == "Female")) %>%
    pull(prob_female)
```

```
## [1] 0.1747573
```

Part B:

```
hts = 60:80
prob_female = sapply(hts, function(x){
    heights %>%
        filter(height == x) %>%
        summarize(prob_female = mean(sex == "Female")) %>%
        pull(prob_female)
})
plot(hts, prob_female)
```



```
ind = max(which(prob_female > 0.5))
hts[ind]
```

```
## [1] 64
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Assignment Setup

The `leukemiasEset` contains 60 sets of bone marrow gene expression data from patients with one of the 4 main types of leukemia (ALL, AML, CLL, CML) as well as control patients without leukemia (NoL).

Install and load the `leukemiasEset` data from the **leukemiasEset** Bioconductor package:

```
# BiocManager::install("leukemiasEset")    # install if nee
library(leukemiasEset)<

## Loading required package: Biobase

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel'
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEv
##     clusterExport, clusterMap, parApply, parCapply, parL
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbin
##     dirname, do.call, duplicated, eval, evalq, Filter, F
##     grepl, intersect, is.unsorted, lapply, Map, mapply,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Positi
##     rbind, Reduce, rownames, sapply, setdiff, sort, tabl
##     union, unique, unsplit, which, which.max, which.min

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pl

data(leukemiasEset)
```

These data are stored in a container called an *ExpressionSet*. In future courses, we will learn how to work with *ExpressionSets* directly, but for now we can extract gene expression data as a matrix `dat` (features are rows, columns are samples):

```
dat = exprs(leukemiasEset)
```

We can also create a vector noting which type of leukemia is present in each sample:

```
leuk = leukemiasEset$LeukemiaType
```

Question 1

3/3 points (graded)

A. How many features are present in `dat` ?

20172

✓ **Answer:** 20172

20172

B. How many samples are present in `dat` ?

60

✓ **Answer:** 60

60

C. How many samples are from patients with AML?

12

✓ **Answer:** 12

12

Explanation

Part A:

```
nrow(dat)
```

```
## [1] 20172
```

Part B:

```
ncol(dat)
```

```
## [1] 60
```

Part C:

```
sum(leuk == "AML")
```

```
## [1] 12
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 2

1/1 point (graded)

Make an MDS plot of `dat` and color the points by `leuk`.

Which of the following are TRUE?

Select ALL that apply.

☑ CLL samples tend to have higher values of `mds[,2]` than ALL

□ CLL samples tend to have higher values of `mds[,1]` than ALL

☑ The samples with the highest values of `mds[,1]` are all CML

□ The samples with the lowest values of `mds[,2]` are all NoL

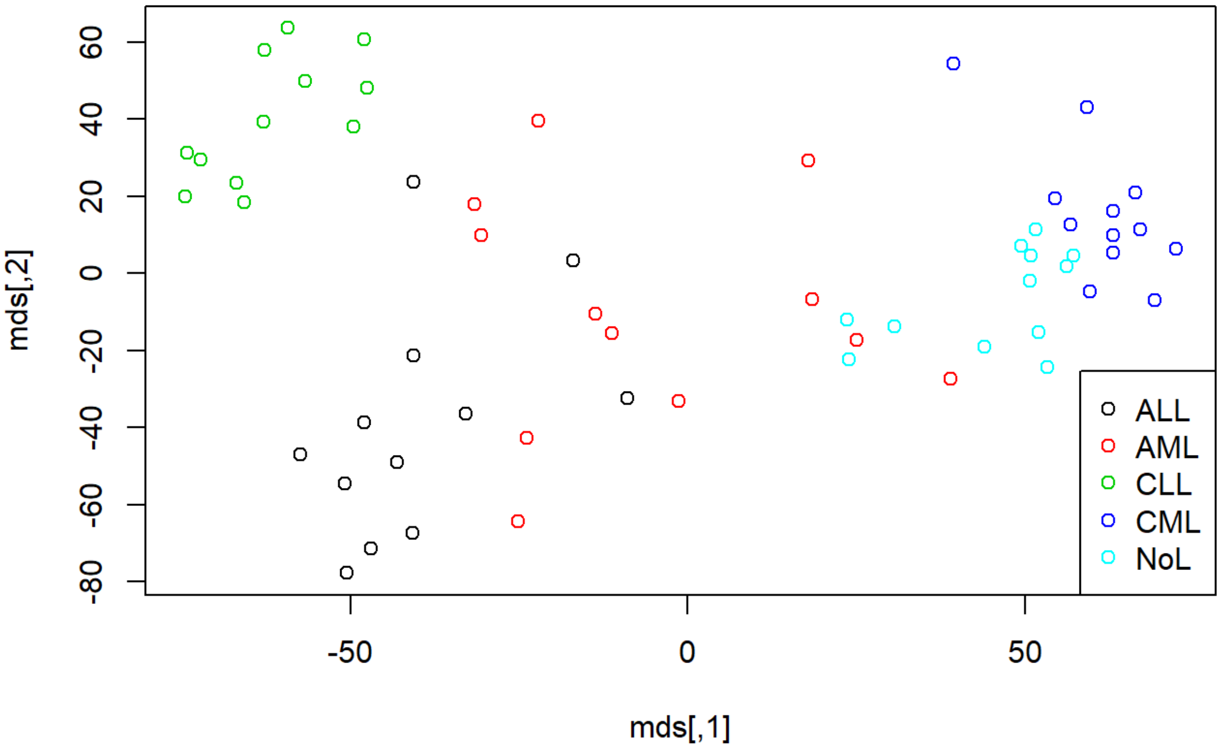
□ All five of the leukemia types form clear, non-overlapping clusters.

☑ At a glance, CML samples are more similar to NoL samples than to other leukemias.

✓

Explanation

```
mds = cmdscale(dist(t(dat)))
plot(mds, col=leuk)
legend("bottomright", levels(leuk), col=seq_along(leuk), pc
```



Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 3

1/1 point (graded)
Run hierarchical clustering on this data with the `hclust()` function with default parameters to cluster the columns. Create a dendrogram and use the leukemia type `leuk` as labels.

Suppose you want to cut the tree so that there are 5 clusters. Which of these heights would be the best cutoff?

○ 100

○ 125

● 150

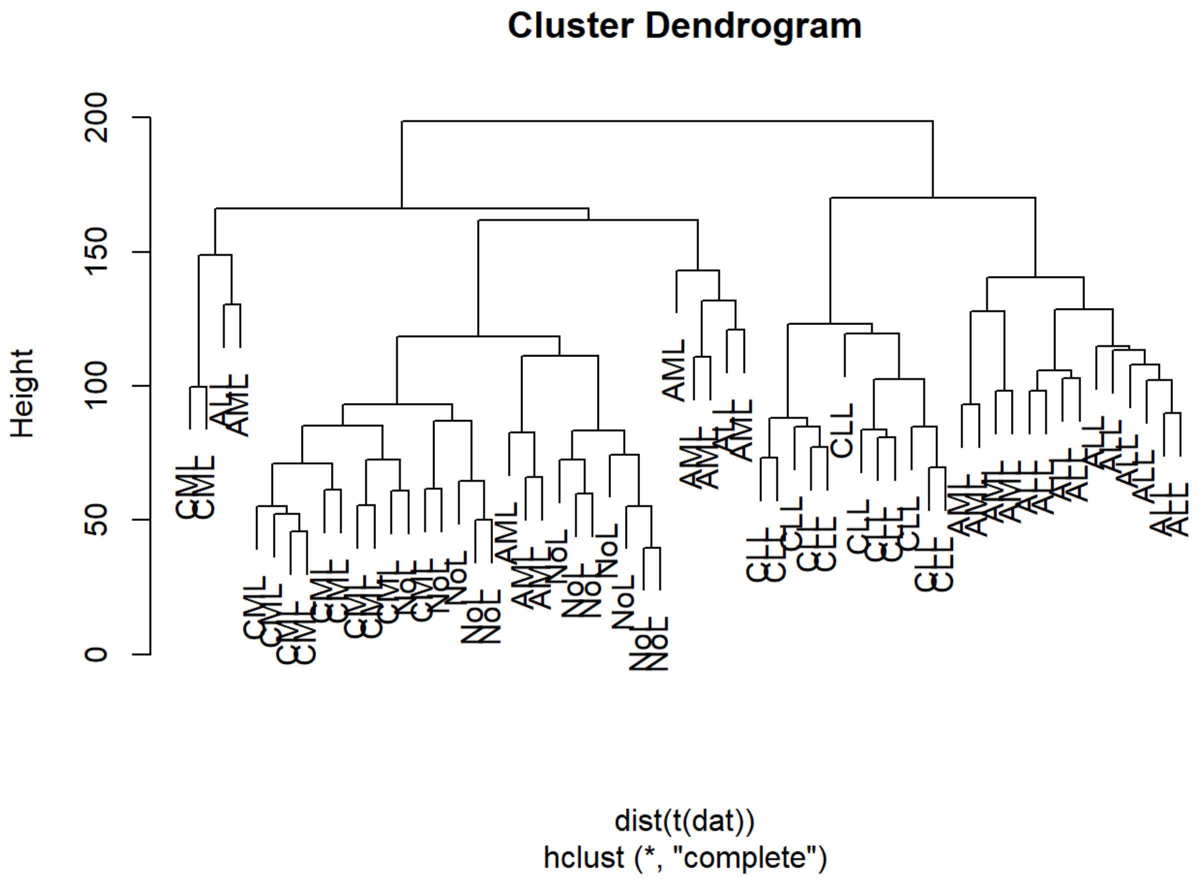
○ 175

○ 200

✓

Explanation

```
hc = hclust( dist( t(dat)))
plot(hc, labels = leuk)
```



```
table(cutree(hc, h=150))
```

```
##
##   1   2   3   4   5
##   5   4  14  25  12
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 4

3/3 points (graded)

Using the cutoff height that generates 5 clusters in the previous problem, one cluster contains exactly 12 samples that are all from the same leukemia type.

Which two leukemia types have all samples of that type in a unique cluster?

Check two.

□ ALL

□ AML

☑ CLL

☑ CML

□ NoL

✓

Which leukemia type is sometimes clustered with CML?

Check one.

○ AML

○ NoL

● CLL

○ CML

○ ALL

✓

Which leukemia type forms this cluster?

Check one.

○ ALL

○ NoL

○ CML

○ AML

● CLL

✓

Explanation

```
set.seed(4)
result=kmeans(t(dat),5)
table(result$cluster, leuk)
```

```
##     leuk
##       ALL AML CLL CML NoL
##    1    0   3   0   0   8
##    2   10   2   0   0   0
##    3    0   0   0  12   4
##    4    0   0  12   0   0
##    5    2   7   0   0   0
```

The correct answer for part 3 is CLL. All other clusters are mixes of different leukemia types.

Submit

You have used 1 of 5 attempts

ℹ Answers are displayed within the problem

Question 5

1/1 point (graded)

Pick the 25 genes with the highest across sample variance using the `rowMads()` function from **matrixStats**:

```
library(matrixStats)

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

## The following object is masked from 'package:dplyr':
##
##     count

sds =rowMads(dat)
ind = order(sds,decreasing=TRUE)[1:25]
```

Use `heatmap.2()` from **gplots** to make a heatmap showing the `leuk` type with column colors as well as column labels, and scaling the rows. (In the future, we will learn how to convert gene IDs, like "ENSG000...", into gene names.)

Which of the following statements are TRUE about the heatmap?
Select ALL that apply.

☑ Over 20 of the genes with the highest across sample variance are upregulated in CML and NoL and downregulated in other leukemias.

☑ The bottom 2 genes in the plot tend to be upregulated in ALL and CLL and downregulated in AML and CML.

☐ All of the CLL samples cluster together and are directly adjacent to each other in the heatmap.

☐ All of the AML samples cluster together and are directly adjacent to each other in the heatmap.

☑ Based on these 25 genes, the type of leukemia with the closest expression pattern to normal (NoL) bone marrow is CML.

✓

Explanation

CLL samples are mixed in with ALL and AML, so the third and fourth choices above are false.

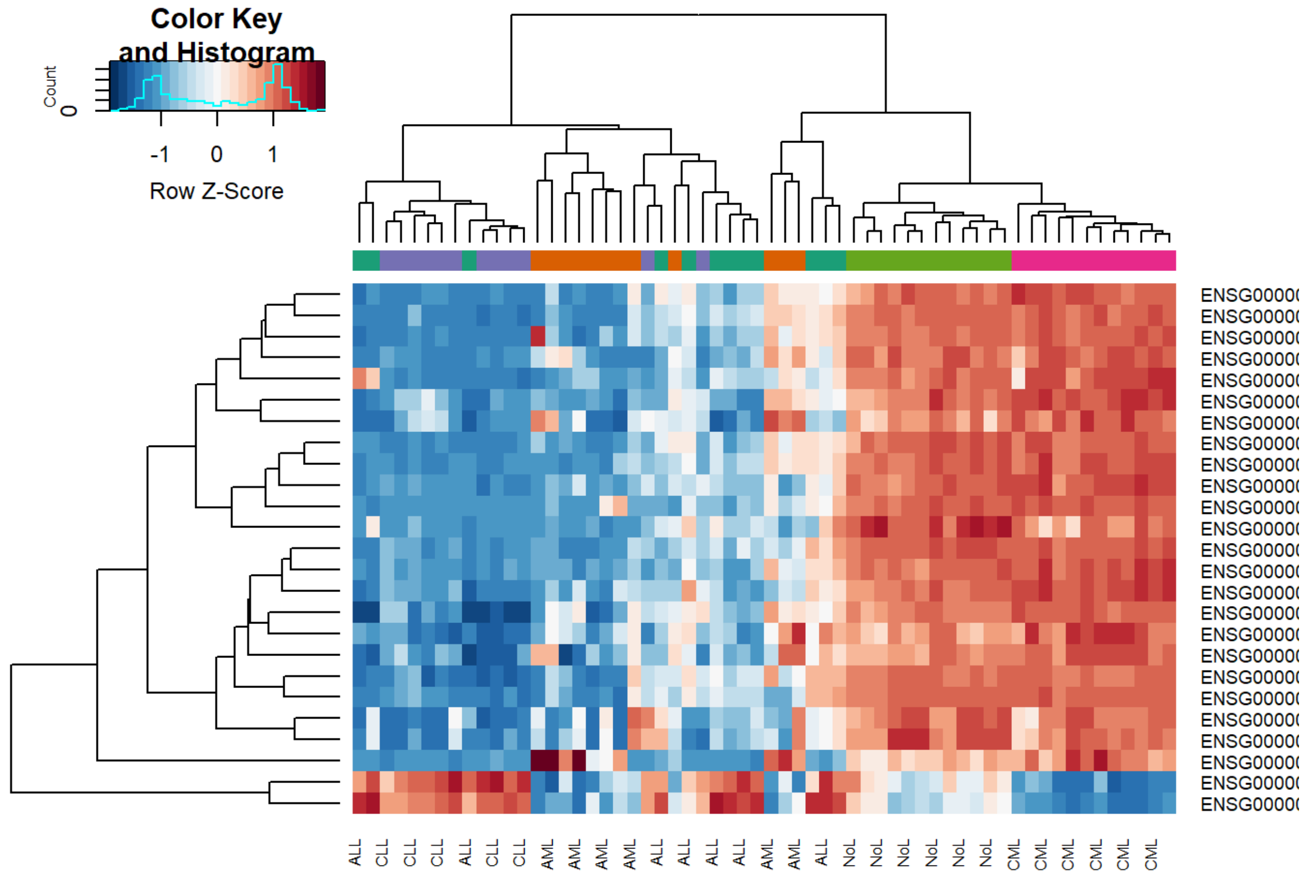
```
library(RColorBrewer)
cols = colorRampPalette(rev(brewer.pal(11,"RdBu")))(25)
gcol=brewer.pal(5,"Dark2")
gcol=gcol[as.numeric(leuk)]

library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
heatmap.2(dat[ind,],
          col=cols,
          trace="none",
          scale="row",
          labCol=leuk,
          ColSideColors=gcol)
```



Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 6

2/2 points (graded)

Suppose you want to design an algorithm that can predict whether a sample from the leukemia dataset is normal ("NoL") versus any type of leukemia. Start by creating a vector `leukTF` that is `TRUE` when a sample is normal and `FALSE` when a sample is leukemia:

```
leukTF = leuk == "NoL"
```

Load the **caret** library and set the seed to 2. Use `createFolds()` on `leuk2` to create 5 folds for cross-validation. Save the indices for these folds as `idx`.

Before running any machine learning algorithms on these folds, it is best to ensure that each fold contains both normal and leukemia samples. Count the number of normal samples in each fold.

A: How many folds have at least 1 normal sample?

5

5

✓ **Answer:** 5

B: How many folds have exactly 3 normal samples?

2

2

✓ **Answer:** 2

Explanation

Part A:

```
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(2)
idx = createFolds(leukTF, k=5)

normal_counts = sapply(1:length(idx), function(x){
    fold_ind = idx[[x]]
    sum(leukTF[fold_ind]==TRUE)
})

sum(normal_counts > 0)
```

```
## [1] 5
```

Part B:

```
sum(normal_counts == 3)
```

```
## [1] 2
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 7

1/1 point (graded)

We are going to consider a smaller set of predictors by filtering genes using t-tests. Specifically, we will perform a t-test and select the m genes with the smallest p-values.

Let $m = 3$. Leave out the first fold, `idx[[1]]`, and perform `rowttests()` from the **genefilter** library on the remaining samples. Find the row numbers of the 3 genes with the lowest p-values and save these as `gene_ind`.

Which of these rows does ***not*** represent one of the three genes with the lowest p-values when omitting the first fold, stored in `gene_ind`?

● 3637

○ 14613

○ 16993

○ 17033

✓

Explanation

```
library(genefilter)

##
## Attaching package: 'genefilter'

## The following objects are masked from 'package:matrixSta
##
##     rowSds, rowVars

## The following object is masked from 'package:readr':
##
##     spec

## set m = number of genes
m = 3

# define fold and find top m genes in fold
fold_ind = idx[[1]]
pvals = rowttests(dat[,-fold_ind],factor(leukTF[-fold_ind])
gene_ind = order(pvals)[1:m]
gene_ind
```

```
## [1] 16993 14613 17033
```

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Question 8

1/1 point (graded)

Separate `dat` into a test set consisting of samples in the first fold and a training set consisting of samples in all other folds. Keep only genes from `gene_ind` in these sets. (Your test set should be an 11×3 matrix and your training set should be a 49×3 matrix.)

Train a kNN model and generate predictions for the test set using the `knn` function from the **class** library and `k=5`.

How many errors does this model make on the test set for the first fold?

1

✓ **Answer:** 1

1

Explanation

```
library(class)

# use gene_ind and fold_ind to define training and test sets and training c
train_set = t(dat[gene_ind, -fold_ind])
test_set = t(dat[gene_ind, fold_ind])
train_classes = leukTF[-fold_ind]

# set k=number of nearest neighbors
k = 5

# run knn
pred = knn(train_set, test_set, train_classes, k)

# count the number of errors
sum(pred!=leukTF[fold_ind])
```

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

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 9

3/3 points (graded)

Repeat the steps from questions 8 and 9 above for each of the 5 folds.

A. What is the total number of errors across all 5 folds?

4 ✓

4

B. What proportion of the 60 samples are classified incorrectly by this model?

0.06666667 ✓

0.06666667

C. Accuracy is defined as 1 minus the error rate. What is the accuracy of this kNN model?

0.9333333 ✓

0.9333333

Submit

You have used 2 of 5 attempts

✓ Correct (3/3 points)