Lab 11 | BINF 6310 | Jon Lee | Spring 2020

Code

Lab #11:

Please e-mail code, graphs and answers to questions to afodor@uncc.edu (mailto:afodor@uncc.edu). Please put lab #11 in the subject line.

Please have lab submitted (whatever you have) before lab on Thursday, April 23rd.

This week's dataset is here: http://afodor.github.io/classes/stats2015/prePostPhylum.txt (http://afodor.github.io/classes/stats2015/prePostPhylum.txt) (This dataset is described, albeit from a different analysis pipeline, in these papers: http://www.sciencemag.org/content/sci/338/6103/120.full.html (http://www.sciencemag.org/content/sci/338/6103/120.full.html) and http://www.nature.com/ncomms/2014/140903/ncomms5724/full/ncomms5724.html (http://www.nature.com/ncomms/2014/140903/ncomms5724/full/ncomms5724.html)) Note that WT and IL10-/- animals are in different cages. So "Cage1_WT" is a different cage from "Cage1_10-/-".

1. Download the dataset. Perform PCA ordination. For example:

   inFileName <- paste("prePostPhylum.txt", sep ="")

   myT <-read.table(inFileName,header=TRUE,sep=") numCols <- ncol(myT) myColClasses <- c(rep("character",4), rep("numeric", numCols-4)) myT <- read.table(inFileName,header=TRUE,sep=",colClasses=myColClasses)

   myTData<-myT[,5:10]

   myPCOA <- princomp(myTData)

Hide

```
#load in data and perform PCA
pFile <- paste("prePostPhylum.txt", sep = "")

pTable <-read.table(pFile,header = TRUE, sep = "\t")
numCols <- ncol(pTable)
pClasses <- c(rep("character", 4), rep("numeric", numCols-4))
pTable <- read.table(pFile, header = TRUE, sep = "\t", colClasses = pClasses)

pData<- pTable[,5:10]

pPCOA <- princomp(pData)
```

2. Graph PCA1 vs. PCA2. Make three versions of the graph. One colored by genotype, one colored by cage and one colored by timepoint (pre-vs-post).

Hide

```
#index row locations for geneotypes in data
wtRowNum <- vector()
mutRowNum <- vector()

for(i in 1:length(pTable[,4]))
{
    if(pTable[i,4] == "WT")
    {
        wtRowNum <- c(wtRowNum, i)
    }
    if(pTable[i,4] == "10-/-")
    {
        mutRowNum <- c(mutRowNum, i)
    }
}
```

Hide

```
#index row locations for timpepoints in data
preRowNum <- vector()
postRowNum <- vector()

for(i in 1:length(pTable[,3]))
{
    if(pTable[i,3] == "PRE")
    {
        preRowNum <- c(preRowNum, i)
    }
    if(pTable[i,3] == "POST")
    {
        postRowNum <- c(postRowNum, i)
    }
}
```

Hide

```
#index row locations for cages in data
cage1WTRowNum <- vector()
cage2WTRowNum <- vector()
cage3WTRowNum <- vector()
cage4WTRowNum <- vector()
cage5WTRowNum <- vector()
cage6WTRowNum <- vector()

cage1MutRowNum <- vector()
cage2MutRowNum <- vector()
cage3MutRowNum <- vector()
cage4MutRowNum <- vector()
cage5MutRowNum <- vector()

for(i in 1:length(pTable[,2]))
{
    if(pTable[i,2] == "Cage1_WT")
    {
        cage1WTRowNum <- c(cage1WTRowNum, i)
    }
    if(pTable[i,2] == "Cage2_WT")
    {
        cage2WTRowNum <- c(cage2WTRowNum, i)
    }
    if(pTable[i,2] == "Cage3_WT")
    {
        cage3WTRowNum <- c(cage3WTRowNum, i)
    }
    if(pTable[i,2] == "Cage4_WT")
    {
        cage4WTRowNum <- c(cage4WTRowNum, i)
    }
    if(pTable[i,2] == "Cage5_WT")
    {
        cage5WTRowNum <- c(cage5WTRowNum, i)
    }
    if(pTable[i,2] == "Cage6_WT")
    {
        cage6WTRowNum <- c(cage6WTRowNum, i)
    }
    if(pTable[i,2] == "Cage1_10-/-")
    {
        cage1MutRowNum <- c(cage1MutRowNum, i)
    }
     if(pTable[i,2] == "Cage2_10-/-")
    {
        cage2MutRowNum <- c(cage2MutRowNum, i)
     }
     if(pTable[i,2] == "Cage3_10-/-")
    {
        cage3MutRowNum <- c(cage3MutRowNum, i)
     }
     if(pTable[i,2] == "Cage4_10-/-")
```

```
    {
        cage4MutRowNum <- c(cage4MutRowNum, i)
     }
     if(pTable[i,2] == "Cage5_10-/-")
    {
        cage5MutRowNum <- c(cage5MutRowNum, i)
    }
}
```

Hide

```
#separate out data into first 2 principal components
PCA1 <- pPCOA$scores[,1]
PCA2 <- pPCOA$scores[,2]

#data separation vectors for PCA1 into genotype, timepoint, and cage
wtPCA1 <- vector()
mutPCA1 <- vector()

prePCA1 <- vector()
postPCA1 <- vector()

cg1WTPCA1 <- vector()
cg2WTPCA1 <- vector()
cg3WTPCA1 <- vector()
cg4WTPCA1 <- vector()
cg5WTPCA1 <- vector()
cg6WTPCA1 <- vector()
cg1MutPCA1 <- vector()
cg2MutPCA1 <- vector()
cg3MutPCA1 <- vector()
cg4MutPCA1 <- vector()
cg5MutPCA1 <- vector()

#data separation vectors for PCA2 into genotype, timepoint, and cage
wtPCA2 <- vector()
mutPCA2 <- vector()

prePCA2 <- vector()
postPCA2 <- vector()

cg1WTPCA2 <- vector()
cg2WTPCA2 <- vector()
cg3WTPCA2 <- vector()
cg4WTPCA2 <- vector()
cg5WTPCA2 <- vector()
cg6WTPCA2 <- vector()
cg1MutPCA2 <- vector()
cg2MutPCA2 <- vector()
cg3MutPCA2 <- vector()
cg4MutPCA2 <- vector()
cg5MutPCA2 <- vector()
```

Hide

```
#data separation for PCA1 and PCA2
for(i in 1:length(wtRowNum))
{
    wtPCA1 <- c(wtPCA1, PCA1[wtRowNum[i]])
    wtPCA2 <- c(wtPCA2, PCA2[wtRowNum[i]])
}
for(i in 1:length(mutRowNum))
{
    mutPCA1 <- c(mutPCA1, PCA1[mutRowNum[i]])
    mutPCA2 <- c(mutPCA2, PCA2[mutRowNum[i]])
}
for(i in 1:length(preRowNum))
{
    prePCA1 <- c(prePCA1, PCA1[preRowNum[i]])
    prePCA2 <- c(prePCA2, PCA2[preRowNum[i]])
}
for(i in 1:length(postRowNum))
{
   postPCA1 <- c(postPCA1, PCA1[postRowNum[i]])
   postPCA2 <- c(postPCA2, PCA2[postRowNum[i]])
}

for(i in 1:length(cage1WTRowNum))
{
    cg1WTPCA1 <- c(cg1WTPCA1, PCA1[cage1WTRowNum[i]])
    cg1WTPCA2 <- c(cg1WTPCA2, PCA2[cage1WTRowNum[i]])
}
for(i in 1:length(cage2WTRowNum))
{
    cg2WTPCA1 <- c(cg2WTPCA1, PCA1[cage2WTRowNum[i]])
    cg2WTPCA2 <- c(cg2WTPCA2, PCA2[cage2WTRowNum[i]])
}
for(i in 1:length(cage3WTRowNum))
{
    cg3WTPCA1 <- c(cg3WTPCA1, PCA1[cage3WTRowNum[i]])
    cg3WTPCA2 <- c(cg3WTPCA2, PCA2[cage3WTRowNum[i]])
}
for(i in 1:length(cage4WTRowNum))
{
    cg4WTPCA1 <- c(cg4WTPCA1, PCA1[cage4WTRowNum[i]])
    cg4WTPCA2 <- c(cg4WTPCA2, PCA2[cage4WTRowNum[i]])
}
for(i in 1:length(cage5WTRowNum))
{
    cg5WTPCA1 <- c(cg5WTPCA1, PCA1[cage5WTRowNum[i]])
    cg5WTPCA2 <- c(cg5WTPCA2, PCA2[cage5WTRowNum[i]])
}
for(i in 1:length(cage6WTRowNum))
{
    cg6WTPCA1 <- c(cg6WTPCA1, PCA1[cage6WTRowNum[i]])
    cg6WTPCA2 <- c(cg6WTPCA2, PCA2[cage6WTRowNum[i]])
}
for(i in 1:length(cage1MutRowNum))
```

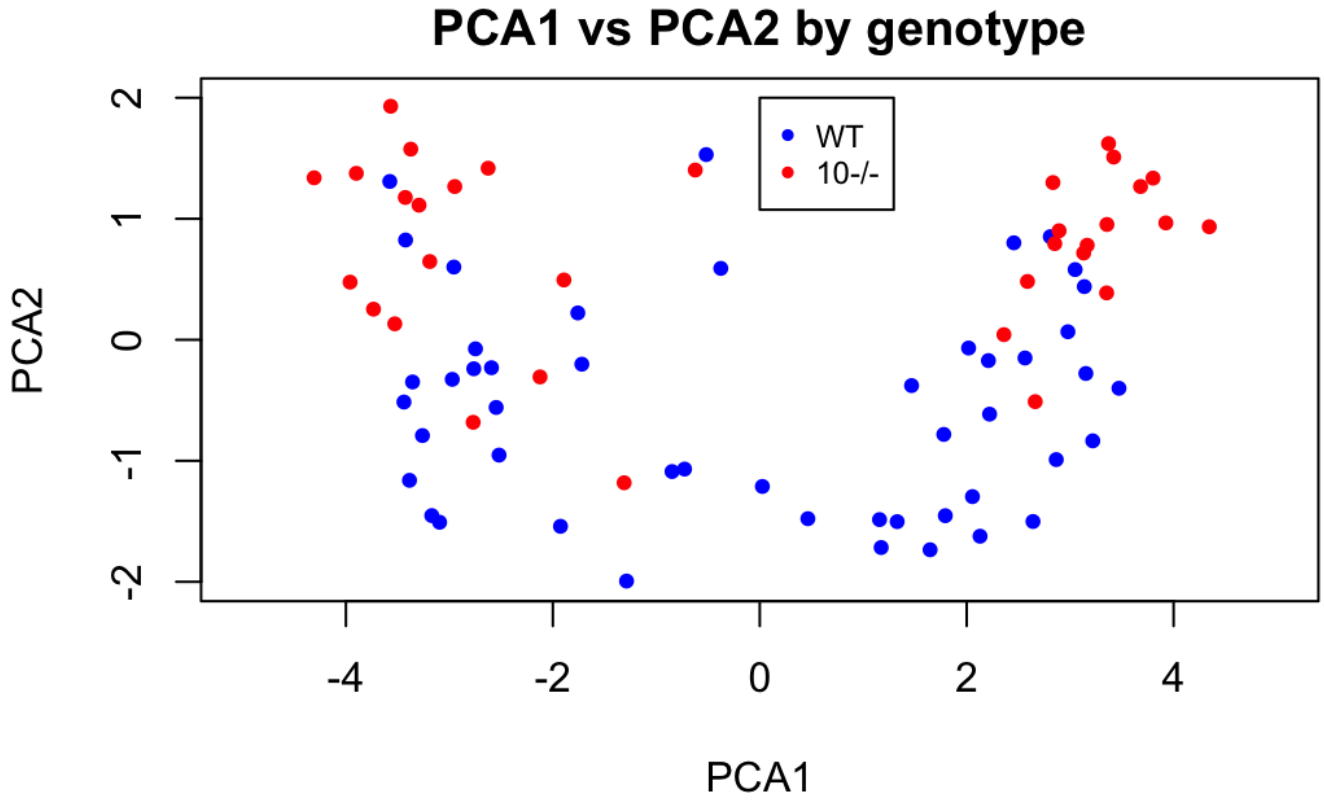
```
{
    cg1MutPCA1 <- c(cg1MutPCA1, PCA1[cage1MutRowNum[i]])
    cg1MutPCA2 <- c(cg1MutPCA2, PCA2[cage1MutRowNum[i]])
}
for(i in 1:length(cage2MutRowNum))
{
    cg2MutPCA1 <- c(cg2MutPCA1, PCA1[cage2MutRowNum[i]])
    cg2MutPCA2 <- c(cg2MutPCA2, PCA2[cage2MutRowNum[i]])
}
for(i in 1:length(cage3MutRowNum))
{
    cg3MutPCA1 <- c(cg3MutPCA1, PCA1[cage3MutRowNum[i]])
    cg3MutPCA2 <- c(cg3MutPCA2, PCA2[cage3MutRowNum[i]])
}
for(i in 1:length(cage4MutRowNum))
{
    cg4MutPCA1 <- c(cg4MutPCA1, PCA1[cage4MutRowNum[i]])
    cg4MutPCA2 <- c(cg4MutPCA2, PCA2[cage4MutRowNum[i]])
}
for(i in 1:length(cage5MutRowNum))
{
    cg5MutPCA1 <- c(cg5MutPCA1, PCA1[cage5MutRowNum[i]])
    cg5MutPCA2 <- c(cg5MutPCA2, PCA2[cage5MutRowNum[i]])
}
```

Hide

```
#PCA1 vs PCA2 (graph number 1: genotype)
plot(wtPCA1, wtPCA2, col = "blue", pch = 20, main = "PCA1 vs PCA2 by genotype", xlab =
"PCA1", ylab = "PCA2", xlim = c(-5, 5), ylim = c(-2, 2))
points(mutPCA1, mutPCA2, col = "red", pch = 20)
```

Hide

```
legend(0, 2, col = c("blue", "red"), pch = c(20,20), legend = c("WT", "10-/-"), cex = 0.
75)
```

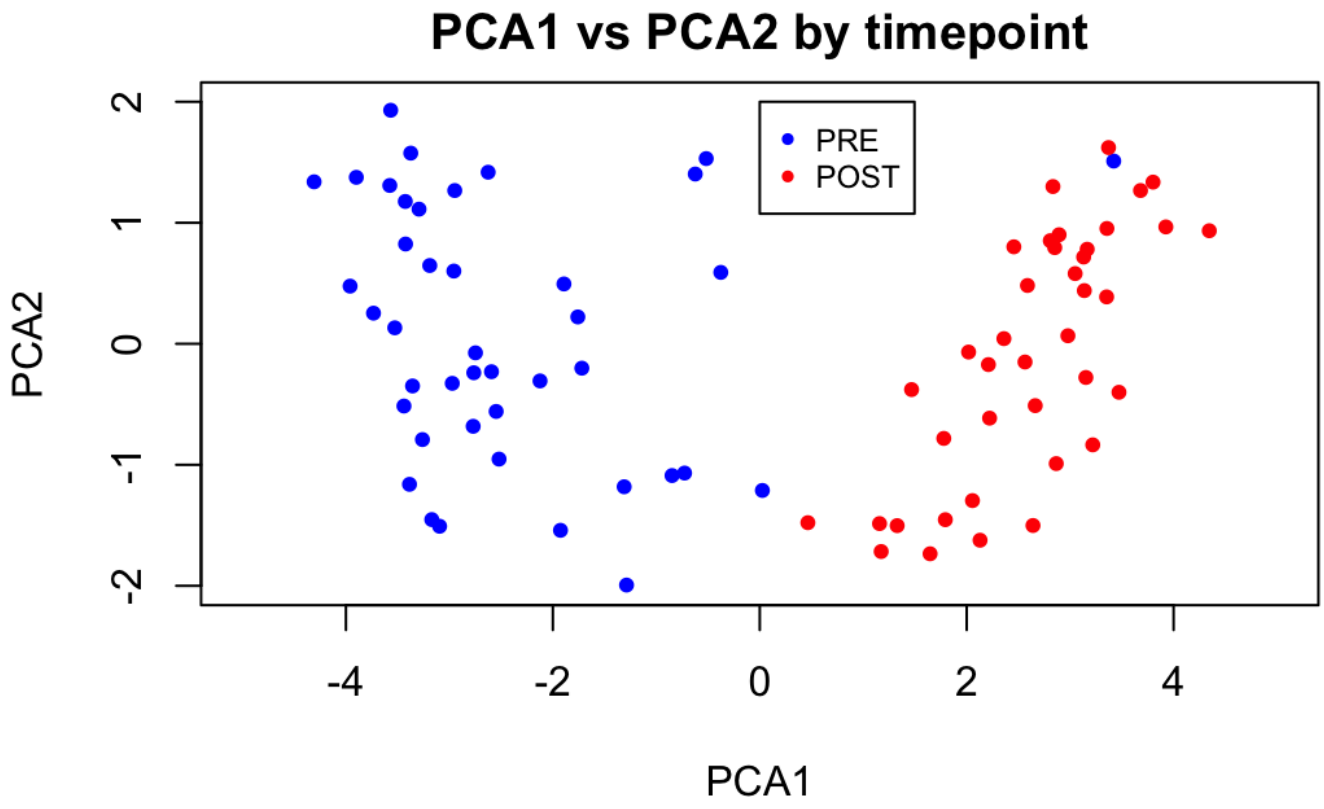


Hide

```
#PCA1 vs PCA2 (graph number 2: timepoint)
plot(prePCA1, prePCA2, col = "blue", pch = 20, main = "PCA1 vs PCA2 by timepoint", xlab
 = "PCA1", ylab = "PCA2", xlim = c(-5, 5), ylim = c(-2, 2))
points(postPCA1, postPCA2, col = "red", pch = 20)
```

Hide

```
legend(0,2, col = c("blue", "red"), pch = c(20,20), legend = c("PRE", "POST"), cex = 0.7
5)
```



Hide

```
#PCA1 vs PCA2 (graph number 3: cage)
plot(cg1WTPCA1, cg1WTPCA2, col = "blue", pch = 20, main = "PCA1 vs PCA2 by timepoint", x
lab = "PCA1", ylab = "PCA2", xlim = c(-5, 5), ylim = c(-2, 2))
points(cg2WTPCA1, cg2WTPCA2, col = "red", pch = 20)
```

Hide

```
points(cg3WTPCA1, cg3WTPCA2, col = "green", pch = 20)
points(cg4WTPCA1, cg4WTPCA2, col = "orange", pch = 20)
```

Hide

```
points(cg5WTPCA1, cg5WTPCA2, col = "purple", pch = 20)
points(cg6WTPCA1, cg6WTPCA2, col = "yellow", pch = 20)
```

Hide

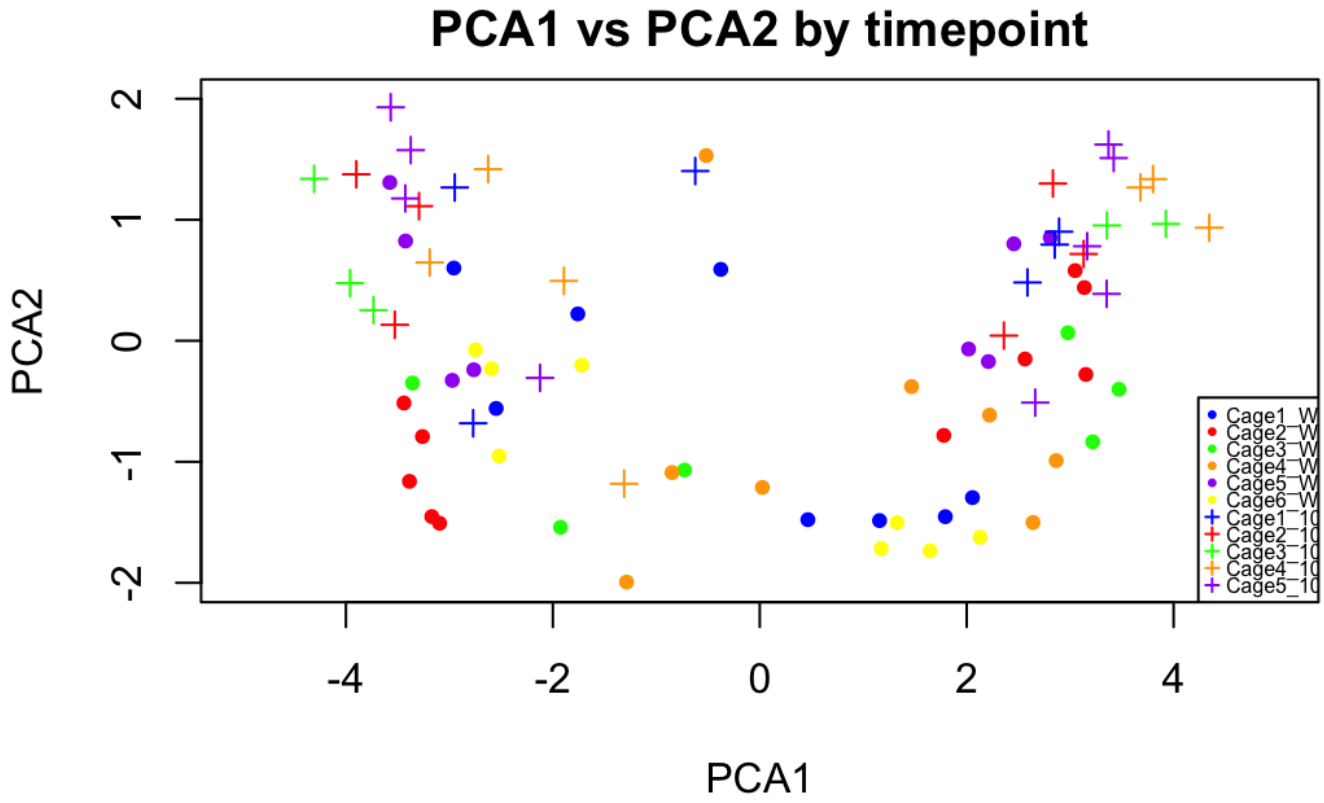
```
points(cg1MutPCA1, cg1MutPCA2, col = "blue", pch = 3)
points(cg2MutPCA1, cg2MutPCA2, col = "red", pch = 3)
```

Hide

```
points(cg3MutPCA1, cg3MutPCA2, col = "green", pch = 3)
points(cg4MutPCA1, cg4MutPCA2, col = "orange", pch = 3)
```

Hide

```
points(cg5MutPCA1, cg5MutPCA2, col = "purple", pch = 3)
legend("bottomright", col = c("blue", "red", "green", "orange", "purple", "yellow", "blu
e", "red", "green", "orange", "purple"), pch = c(rep(20, 6), rep(3, 5)), legend = c("Cag
e1_WT", "Cage2_WT", "Cage3_WT", "Cage4_WT", "Cage5_WT", "Cage6_WT", "Cage1_10-/-", "Cage
2_10-/-", "Cage3_10-/-", "Cage4_10-/-", "Cage5_10-/-"), cex = 0.5)
```



3. Fill in the following table for p-values testing the null hypothesis for PCA 1 and 2. For cage, use a one way-ANOVA. For genotype and timepoint ("pre" vs "post") use a t-test.

Hide

```
#genotype t-test p-values for PCA1
t.test(wtPCA1, mutPCA1)$p.value
```

```
[1] 0.929701
```

Hide

```
#genotype t-test p-values for PCA2
t.test(wtPCA2, mutPCA2)$p.value
```

```
[1] 1.274344e-10
```

Hide

```
#timepoint t-test p-values for PCA1
t.test(prePCA1, postPCA1)$p.value
```

```
[1] 2.519974e-29
```

Hide

```
#timepoint t-test p-values for PCA2
t.test(prePCA2, postPCA2)$p.value
```

```
[1] 0.4268188
```

Hide

```
#cage one way ANOVA p-values for PCA1
cagePCA1Data <- c(cg1WTPCA1, cg2WTPCA1, cg3WTPCA1, cg4WTPCA1, cg5WTPCA1, cg6WTPCA1, cg1M
utPCA1, cg2MutPCA1, cg3MutPCA1, cg4MutPCA1, cg5MutPCA1)
cage1Data <- c(rep("Cage1_WT", length(cg1WTPCA1)), rep("Cage2_WT", length(cg2WTPCA1)), r
ep("Cage3_WT", length(cg3WTPCA1)), rep("Cage4_WT", length(cg4WTPCA1)), rep("Cage5_WT", l
ength(cg5WTPCA1)), rep("Cage6_WT", length(cg6WTPCA1)), rep("Cage1_10-/-", length(cg1MutP
CA1)), rep("Cage2_10-/-", length(cg2MutPCA1)), rep("Cage3_10-/-", length(cg3MutPCA1)), r
ep("Cage4_10-/-", length(cg4MutPCA1)), rep("Cage45_10/-", length(cg5MutPCA1)))
cagePCA1LM <- lm(cagePCA1Data ~ cageData, x = TRUE)
anova(cagePCA1LM)$"Pr(>F)"[1]
```

```
[1] 0.9920581
```

Hide

```
#cage one way ANOVA p-values for PCA1
cagePCA2Data <- c(cg1WTPCA2, cg2WTPCA2, cg3WTPCA2, cg4WTPCA2, cg5WTPCA2, cg6WTPCA2, cg1M
utPCA2, cg2MutPCA2, cg3MutPCA2, cg4MutPCA2, cg5MutPCA2)
cage2Data <- c(rep("Cage1_WT", length(cg1WTPCA2)), rep("Cage2_WT", length(cg2WTPCA2)), r
ep("Cage3_WT", length(cg3WTPCA2)), rep("Cage4_WT", length(cg4WTPCA2)), rep("Cage5_WT", l
ength(cg5WTPCA2)), rep("Cage6_WT", length(cg6WTPCA2)), rep("Cage1_10-/-", length(cg1MutP
CA2)), rep("Cage2_10-/-", length(cg2MutPCA2)), rep("Cage3_10-/-", length(cg3MutPCA2)), r
ep("Cage4_10-/-", length(cg4MutPCA2)), rep("Cage45_10/-", length(cg5MutPCA2)))
cagePCA2LM <- lm(cagePCA2Data ~ cage2Data, x = TRUE)
anova(cagePCA2LM)$"Pr(>F)"[1]
```

```
[1] 1.629589e-07
```

```
                          PCA1           PCA2
Cage                  0.9920581      1.629589e-07
Genotype              0.929701       1.274344e-10
Time (pre vs. post)   2.519974e-29   0.4268188
```

Which variable seems to be most associated with the first PCA axis? Which variable is most associated with the second PCA axis? Does cage seem to be having an effect on these data?

Answer:

The variable that seems most associated with the first PCA component is the timepoint, and for the second componenet is the genotype. The caging does seem to have an affect on the data for the second component but not for the first component.