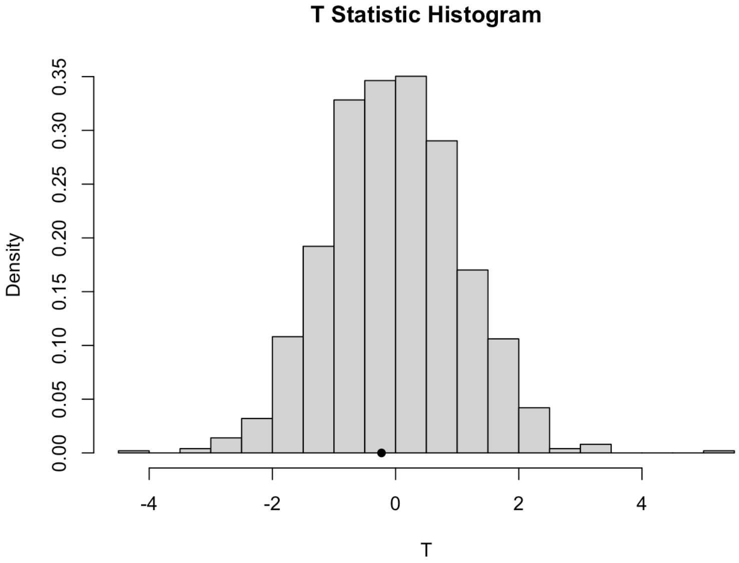
Question 10.1

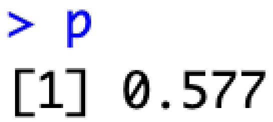
Purpose: Refer to Example 10.1 and Figure 10.1. Suppose that we want to test H0 : F = G, where F is the distribution of weight for the casein feed group and G is the distribution of weight for the sunflower feed group of the chickwts data. A test can be based on the two-sample Kolmogorov-Smirnov statistic as shown in Example 10.1. Display a histogram of the permutation replicates of the Kolmogorov-Smirnov two-sample test statistic for this test. Is the test significant at α= 0.10?

R code:

```
attach(chickwts)
x <- sort(weight[feed == "casein"])
y <- sort(weight[feed == "sunflower"])
detach(chickwts)
R <- 999 #number of replicates
z <- c(x, y) #pooled sample
K <- 1:length(z)
reps <- numeric(R) #storage for replicates
t0 <- t.test(x, y)$statistic
for (i in 1:R) {
   k <- sample(K, size = length(x), replace = FALSE) #generate indices k for the first sample
   x1 <- z[k]
   y1 <- z[-k] #complement of x1
   reps[i] <- t.test(x1, y1)$statistic
}
p <- mean(c(t0, reps) >= t0)
hist(reps, main = "T Statistic Histogram", freq = FALSE, xlab = "T", breaks = "scott")
points(t0, 0, cex = 1, pch = 16)
```

Results:





```
> p
[1] 0.577
```

Approach and Conclusion:

- The original samples of chick weights were placed in variables x and y
- The test statistic T was found based on the original samples, this t stat assumes the same normal distribution from both samples
- The samples were combined together and shuffled randomly for 999 different permutations. They were compared and their T statistics were collected
- The randomly shuffled permutation T statistics were compared to the T statistic from the original sample
- The amount of T’s that were greater or equal to the original T were found to be 0.577, this is our p value
- Since the p value is so high, we fail to reject that F = G at the alpha = 0.1 significance level.
- The histogram shows the point where the original test statistic fell, as well as the permutation results

Question 10.3

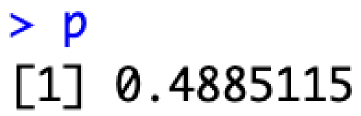
Purpose: Implement the two-sample Cramér-von Mises test for equal distributions as a permutation test using (10.14). Apply the test to the data in Examples 10.1 and 10.2.

$$W^2 = \frac{U}{nm(n+m)} - \frac{4mn-1}{6(m+n)}.$$

R code:

```
library(twosamples)
R=1000
K=24
D = numeric(R)
z <- c(x, y)
D0 <- cvm_stat(x, y)
for (i in 1:R) {
   #generate indices k for the first sample
   k <- sample(1:K, 12, replace = FALSE)
   x1 <- z[k]
   y1 <- z[-k]      #complement of x1
   D[i] <- cvm_stat(x1, y1)
}
p <- mean(c(D0, D) >= D0)
p
```

Results:



```
> p
[1] 0.4885115
```

Approach and Conclusion:

- The package twosamples was loaded in to utilize the cvm_stat function to generate the Cramér-von Mises statistic W^2
- The original samples of chick weights were still placed in variables x and y from the last question
- The test statistic W^2, which gives an idea of the distance between the cdfs of two distributions, was found based on the original samples
- The samples were combined together and shuffled randomly for 1000 different permutations. They were compared and their W^2 statistics were collected
- The randomly shuffled permutation W^2 statistics were compared to the W^2 statistic from the original sample
- The amount of W^2 that were greater than or equal to the original W^2 were found to be 0.4885, this is our p value
- Since the W^2 statistic was so high, we fail to reject that the original samples come from different distributions

Question 10.4

Purpose: An rth Nearest Neighbors test statistic for equal distributions: Write a function (for the statistic argument of the boot function) to compute the test statistic Tn,r (10.6). The function syntax should be Tnr(z, ix, sizes, nn) with the data matrix z as its first argument, and an index vector ix as the second argument. The vector of sample sizes and the number of nearest neighbors nn should be the third and fourth arguments. (See the ann function in package yaImpute and Example 10.6.)

R code:

```
library(boot)
library(yaImpute)
NN.idx <- function(x, tree.type="kd", k=NROW(x)){
   x <- as.matrix(x)
   k <- min(c(k+1, NROW(x)))
   NN <- yaImpute::ann(ref=x, target=x, tree.type="kd", k=k, verbose=FALSE)
   idx <- NN$knnIndexDist[,1:k]
   nn.idx <- idx[,-1]   #first NN is in column 2
   row.names(nn.idx) <- idx[,1]   #give row names, without this line, it is fine
   nn.idx
}
Tnr <- function(z, ix=1:NROW(z), sizes, nn)
{
   z <- as.matrix(z)
   n1 <- sizes[1]
   n2 <- sizes[2]
   n <- n1 + n2
   z <- as.matrix(z[ix, ])
   nn.idx <- NN.idx(z, k=nn)
   block1 <- nn.idx[1:n1, ]
   block2 <- nn.idx[(n1+1):n, ]
   i1 <- sum(block1 < n1 + .5)
   i2 <- sum(block2 > n1 + .5)
   return((i1 + i2) / (nn * n))
}
```

Approach and Conclusion:

- To calculate the rth Nearest Neighbors test statistic for equal distributions, you must:
    - Pass in multidimensional data containing 2 or more samples
    - Calculate the euclidean distance that the points are from each other
    - Order the distances from least to greatest
    - Replace the distances with the index of the point that corresponds to that distance, these are the points that are the nearest neighbors
    - Filter through all of the neighbors up to the rth nearest neighbor, if the neighbors belonged to the same group of the original point it was compared to, add a 1, if it belonged to a different group, don't add anything
    - Divide this summation by that by the total number of neighbors (samples * nearest neighbors)
    - This result is basically the proportion of nearest neighbors that fell within the same group

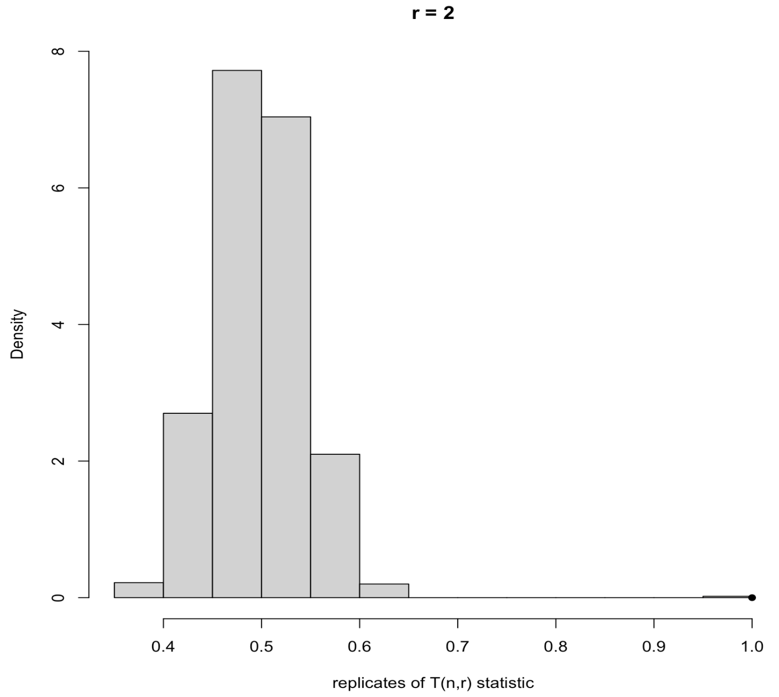
Question 10.5

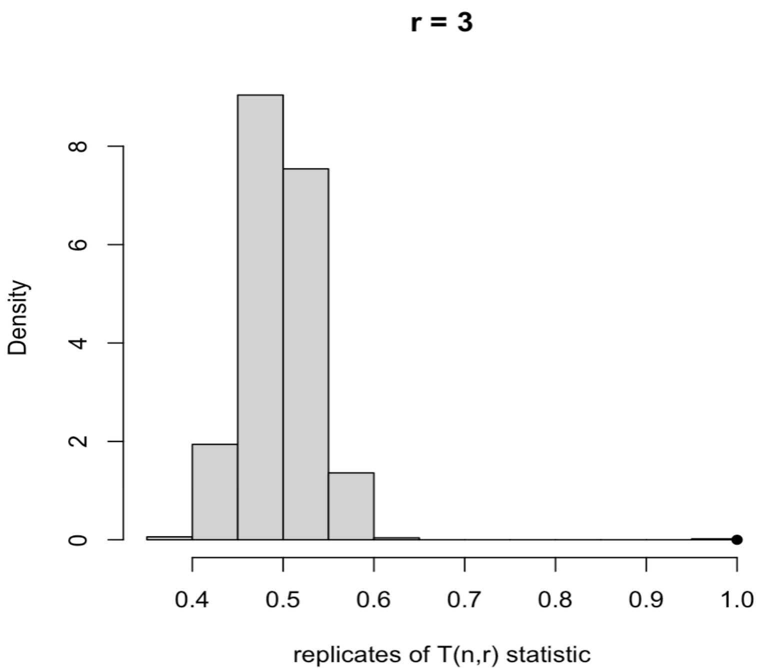
Purpose: The iris data is a four-dimensional distribution with measurements on three species of iris flowers. Using your function Tnr of Exercise 10.10.4 and the boot function, apply your nearest neighbors statistic (r= 2) to test H0 : F= G, where F is the distribution of the iris setosa species, and G is the distribution of the iris virginica species. Repeat the test with r= 3 and r= 4.

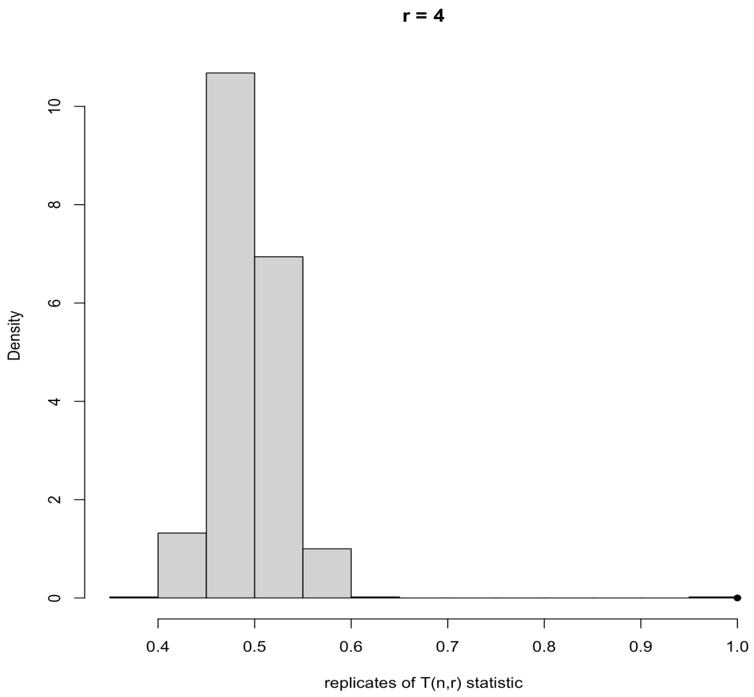
R code:

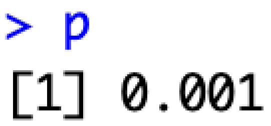
```
data(iris)
x <- iris[iris$Species == 'setosa',]
y <- iris[iris$Species == 'virginica',]
z <- as.matrix(rbind(x[, 1:4], y[, 1:4]))
r = 4
N <- c(nrow(x), nrow(y))
boot.obj <- boot(data = z, statistic = Tnr, sim = "permutation", R = 999, sizes = N, nn = r)
tb <- c(boot.obj$t, boot.obj$t0)
p <- mean(tb >= boot.obj$t0)
p
hist(tb, freq=FALSE, main="r = 4", xlab="replicates of T(n,r) statistic")
points(boot.obj$t0, 0, cex=1, pch=16)
```

Results:




r = 3
Density
0
2
4
6
8
0.4
0.5
0.6
0.7
0.8
0.9
1.0
replicates of T(n,r) statistic


r = 4
Density
0
2
4
6
8
10
0.4
0.5
0.6
0.7
0.8
0.9
1.0
replicates of T(n,r) statistic



```
> p
[1] 0.001
```

Approach and Conclusion:

- Setosa and virginica iris were placed in variables x and y
- The rth Nearest Neighbors test statistic was calculated for setosa and virginica iris
- The boot function was used to combine the setosa and virginica iris together randomly for 999 different permutations. The rth Nearest Neighbors test statistic was collected for each permutation.
- The randomly shuffled permutations of the Nearest Neighbor statistic were compared to the Nearest Neighbor statistics from the original setosa and virginica iris
- The proportion of nearest neighbors that fell within their flower group was 100% for the original setosa and virginica iris. When the groups were created via random permutations, this percentage fell significantly.
- The more nearest neighbors that were included in the calculations, the histogram distributions moved towards the lower proportions falling within the same group.
- The p-value stayed the same regardless of the amount of nearest neighbors that were included, this is because there were not any NN statistics that were greater than the original setosa and virginica iris nearest neighbor statistic.
- We reject that these flowers come from the same distribution, as the p value is so low (p=0.001).
- The histogram shows the point where the original test statistic fell, as well as the results for the random permutations of the nearest neighbor statistic.
- If a proportion of 0.5 was found for the nearest neighbor statistic, that means that the chance the sample fell within the same group is the same as random chance. (only true if only 2 groups were included in the comparison.)