FISH 552
Homework 4 Solutions

Question 1

a) Set a random seed to start the state of R's random number generator

```
set.seed(23424)
```

b) Generate 1000 random variables from a Cauchy distribution with a location parameter 5 and scale parameter 1. Store these values in `randomCauchy`. What are the minimum and maximum values? What is the mean value?

```
randomCauchy <- rcauchy(1000)
range(randomCauchy)
mean(randomCauchy)
```

The Cauchy distribution is an unusual distribution. In fact, it's theoretical mean does not exist! When taking the range, you should have found the minimum and maximum values are quite large relative to the mean of this distribution. In theory, this distribution has an infinite variance. Crazy.

c) Create a sequence of numbers from 0.005 to 0.995 in increments of 0.025. Call this vector `x`.

```
x <- seq(from = 0.005, to = 0.995, by = 0.025)
```

d) Evaluate the density of the points in c) assuming a Beta distribution with shape parameters equal to 5 and 2 respectively. Name this vector `betaDensity`.

```
betaDensity <- dbeta(x, shape1 = 5, shape2 = 2)
```

e) Renormalize the vector `betaDensity` by dividing every observation in the vector by the sum of all the observations. Call this new vector `betaDensityNormalized`. The sum of all the observations should now be 1. Check this.

```
betaDensityNormalized <- betaDensity / sum(betaDensity)
sum(betaDensityNormalized)
```

f) Create a sequence of integers from 12 to 90 in increments of 2. Call this vector `lengths`.

```
lengths <- seq(from = 12, to = 90, by = 2)
```

g) Create a random sample of 1000 lengths from the vector `lengths` with weighted probabilities for each length given by the vector `betaDensityNormalized`. Call this vector `lengthsSample`.

```
lengthsSample <- sample(x = lengths, size = 1000,
    prob = betaDensityNormalized, replace = TRUE)
```

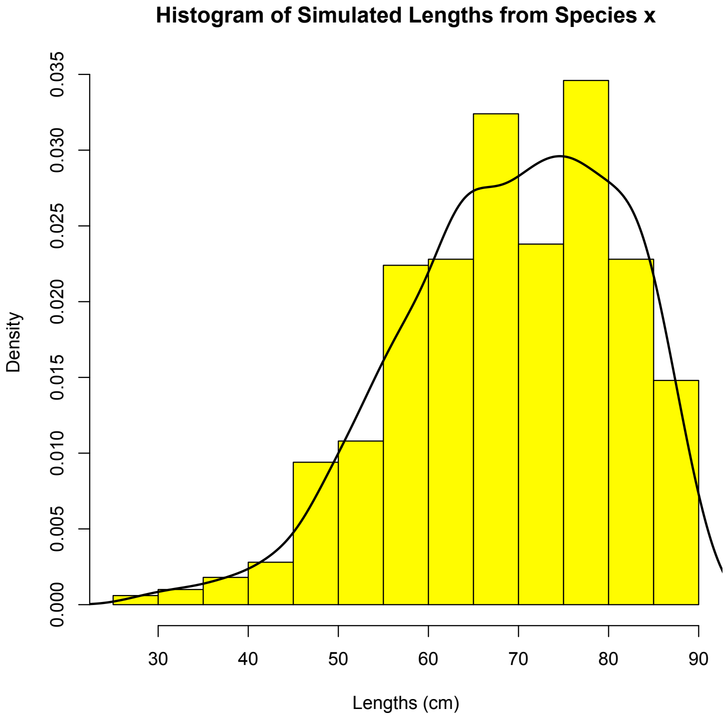
The overall idea here and in questions c) – g) was to show one way we can simulate fake data with the `sample` command. We had to specify `replace = TRUE` because we were sampling

1000 lengths from only 40 and of course there may be many *species* of length *x*. Assuming a population is "large", this is a reasonable assumption since we'll never know every possible length that could have given us our sample.

h) Create a histogram from the vector `lengthsSample`. *Add a density curve. Change the default plot to make your histogram "pretty" (This is subjective so you can add a title, color, . . . or anything that you feel this plot needs).*

```
hist(lengthsSample, freq = FALSE,
  xlab="Lengths (cm)",
  main="Histogram of Simulated Lengths from Species x",
  col="yellow", border="black")
lines(density(lengthsSample), lwd = 2)
```

Your plot should have shown a strong negative skew if you generated the probabilities correctly. Note that setting `freq = FALSE` was necessary to plot the histogram and the density curve on the same scale.



Question 2

For this question we'll be using the `iris` data in R.

a) Compute a t-test to test whether versicolor and virginica irises have unequal mean sepal widths. Decide if the observations should be paired, the variances treated as equal, etc. Carry out this test at the α (Type 1 error rate) = 0.01 significance level.

```
var.test(iris$Sepal.Width[iris$Species == "versicolor"],
  iris$Sepal.Width[iris$Species == "virginica"])
```

This test indicates we can assume equal variances.

```
t.test(iris$Sepal.Width[iris$Species == "versicolor"],
  iris$Sepal.Width[iris$Species == "virginica"],
  conf.level = 0.99, var.equal = TRUE)
```

Note is that the confidence level for a one-sided statistical test is defined by 1 – α. We need to be careful interpreting significance levels, the direction of the test, and then how we tell R that information.

b) Check the normality assumption of the test in a). Comment in your code with your analysis.

```
qqnorm(iris$Sepal.Width[iris$Species == "versicolor"])
qqline(iris$Sepal.Width[iris$Species == "versicolor"])
qqnorm(iris$Sepal.Width[iris$Species == "virginica"])
qqline(iris$Sepal.Width[iris$Species == "virginica"])
```

The data looked pretty normal. For comparison, look at the QQ plots for some randomly generated data from different distributions (see the hands-on exercise from lecture 10).

A few people also used statistical tests to evaluate normality. The Shapiro-Wilk test gave a high p-value for both species, so we would not reject the null hypothesis that the underlying distributions were normal. For the Kolmogorov-Smirnov we need to be careful about not just comparing the sample data to the standard normal distribution with mean 0 and standard deviation 1.

```
> mean(versicolorSepalWidth)
[1] 2.77
> sd(versicolorSepalWidth)
[1] 0.3137983
```

We also get a large p-value and fail to reject the null hypothesis that the distributions are the same.

```
ks.test(versicolorSepalWidth, pnorm,
        mean(versicolorSepalWidth), sd(versicolorSepalWidth))
```

c) Repeat the comparison of means, but this time without assuming normality.

```
wilcox.test(iris$Sepal.Width[iris$Species == "versicolor"],
  iris$Sepal.Width[iris$Species == "virginica"],
  conf.level = 0.99)
```

d) Create a plot of the `iris` data that color codes by species and includes multiple plots (using utilizes `layout` or `par(mfrow)`. Which plots to combine are up to you.

Nice plots!