Homework 8

For this homework you will create an R Markdown document that outputs to a PDF and you will upload both the .Rmd and .pdf files to wolfware. Be sure to put your name in the title of the document.

The purpose of this homework is to get practice with topics covered in the improving R programs section.

Topic: Bootstrapping is a useful technique for investigating the properties of statistical quantities. For instance, we know a single point estimate provides no idea of variability. Often we can derive an estimate of variability and report it along with the estimate - usually we use the **standard error** of the estimator.

Standard errors are really just the standard deviation of the sampling distribution of our estimator. For instance, we know the CLT implies that the distribution of a sample average (point estimator for the population mean) from a 'good' sample can be well described by a normal or bell curve. The standard deviation of this normal distribution is $\sigma/\sqrt{n}$ where σ is the population standard deviation. This is usually estimated by $s/\sqrt{n}$ where s is the sample standard deviation of the data and is referred to as the (estimated) standard error of the sample mean.

In more difficult cases it isn't clear what to use as the standard error without a lot of work searching references or deriving things yourself. For instance, suppose we fit a quadratic regression model. Let Y be the response and x be the predictor then our model looks like

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + E_i$$

We may want to determine where the max or min of the fitted curve occurs. This happens at $max = -\beta_1/(2\beta_2)$. A reasonable estimate of this is to plug in the least squares estimates for each β (found using `lm`, see below). Finding the standard error of the estimate is not straightforward! Insert the bootstrap.

As a standard error is just the standard deviation of the sampling distribution of our quantity of interest, we can use simulation to determine that spread! (Nonparametric) Bootstrap idea:

1. Resample n pairs of (y_i, x_i) from your data with replacement (where n is the sample size you have).
2. Fit your quadratic relationship and obtain an estimate of the max.
3. Save that estimate (it is called a bootstrap estimate).
4. Repeat the above process $B = 5000$ times (or some reasonable sized number).
5. Use the B values as a proxy for the sampling distribution (some properties should mirror the actual sampling distribution). Find the standard deviation of the B values of the max estimate and use this as an approximate standard error.

As a concrete example of this (example based on a consulting project I did years ago), suppose a biological engineer is looking at the effects of concentration (predictor variable) of a media on some aspects of a plant. The response variables investigated were

- Total lignin
- Glucose

- Xylose
- Arabinose

We might fit a quadratic model for each of these responses using the concentration as the predictor, find the estimated maximums for each along with corresponding standard errors for use in the design of a future experiment. The data are avaible with the homework assignment.

To do: Write a markdown file that outputs a pdf (upload both) that does the following:

- Read in the `concentration.csv` data set. Use only observations for species M.giganteus and S.ravennae.
- Use a for loop to implement the bootstrap (use the `sample` function from base R or `sample_n` from dplyr for resampling) for fitting a quadratic model using `concentration` as the predictor and `Total_lignin` as the response (code for a quadratic model fit is `lm(y~x+I(x^2), data = data_set)` - you'll need to extract the coefficients from the returned object to get your estimate). Report an estimate of the maximum with a corresponding standard error.
- Redo the bootstrap analysis for the `Total_lignin` response but make use of the `replicate` function instead of a for loop. Hint: To do this, I created a function called `bootFun` that essentially did everything within one iteration of a for loop. `bootFun` took in only the data set the predictor, and the response to use (both variable names in quotes).
  Report an estimate of the maximum with a corresponding standard error.
- Create a wrapper function for `replicate` that will return the standard deviation of the bootstrapped estimates. Hint: I created a function called `seBootFun` that takes in `resp`, `pred`, `B`, and `data` and returns the standard deviation of the bootstrapped estimates. Apply this function using `Glucose` as the response
- Use parallel computing to send each of the four bootstrap standard error computations (one for each response) to a different core (if you only have a dual core, use two cores). Report estimates of each maximum (no need to find these with parallel computing) with corresponding standard errors.
- The output of my code from above is given here for your reference. Note that some of these standard errors are pretty large and quite variable! - this is a tough quantity to estimate with small sample sizes!

```
## concentration
##      42.06076
```

```
## [1] 1.386729
```

```
## concentration
##      42.06076
```

```
## [1] 1.352775
```

```
## concentration
##      31.99959
```

```
## [1] 17.59023
```

	Max	SE
Lignin	42.06	1.34
Glucose	32.00	4.02
Xylose	43.38	395.54
Arabinose	32.83	120.69