STA 214 Homework 3

Due: Friday, February 3, 12:00pm (noon) on Canvas.

Instructions: There are three parts to this assignment. Part I is practice with logistic regression, Part II is practice with for loops, and Part III is a short (extra credit) problem on debugging in R.

Getting started: Begin by downloading the HW3 template from the course website:

`https://sta214-s23.github.io/homework/hw_03_template.Rmd`

Save this template file to your computer, then open it in RStudio. As you complete the assignment, you will write down your answers to all questions in the R Markdown file, and include all R code in code chunks. *If a question requires code, you will not receive credit if no code is provided.* Refer to the R Markdown instructions on the course website (`https://sta214-s23.github.io/resources/rmarkdown_instructions/`) if you have issues getting started.

Submission: When you have completed the assignment, knit your homework to HTML and submit on Canvas.

Data

The RMS Titanic was a huge, luxury passenger liner designed and built in the early 20th century. Despite the fact that the ship was believed to be unsinkable, during her maiden voyage on April 15, 1912, the Titanic collided with an iceberg and sank. Of all the passengers and crew, less than half survived. Part of the reason why so few people survived has been attributed to the fact that the Titanic did not carry enough lifeboats for its passengers and crew. This meant that there was competition for space in the boats, and not everyone was able to make it aboard. Communication errors, stress and shock...there were a great many factors that contributed to this tragedy.

The loss of life during the Titanic tragedy was enormous, but there were survivors. Was it random chance that these particular people survived? Or were there some specific characteristics of these people that led to their positions in the life boats? Let's investigate.

We have observations on 12 different variables, some categorical and some numeric:

- `Passenger`: A unique ID number for each passenger.
- `Survived`: An indicator for whether the passenger survived (1) or perished (0) during the disaster.
- `Pclass`: Indicator for the class of the ticket held by this passengers; 1 = 1st class, 2 = 2nd class, 3 = 3rd class.
- `Name`: The name of the passenger.
- `Sex`: Binary indicator for the sex of the passenger.
- `Age`: Age of the passenger in years; Age is fractional if the passenger was less than 1 year old.

- `SibSp`: number of siblings/spouses the passenger had aboard the Titanic. Here, siblings are defined as brother, sister, stepbrother, and stepsister. Spouses are defined as husband and wife.
- `Parch`: number of parents/children the passenger had aboard the Titanic. Here, parent is defined as mother/father and child is defined as daughter,son, stepdaughter or stepson. NOTE: Some children traveled only with a nanny, therefore parch=0 for them. There were no parents aboard for these children.
- `Ticket`: The unique ticket number for each passenger.
- `Fare`: How much the ticket cost in US dollars.
- `Cabin`: The cabin number assigned to each passenger. Some cabins hold more than one passenger.
- `Embarked`: Port where the passenger boarded the ship; C = Cherbourg, Q = Queenstown, S = Southampton

Goal: Our goal is to predict the probability that a passenger survives the Titanic disaster.

Loading the data

The `titanic` data can be loaded into R with the following command:

```
titanic <- read.csv("https://sta214-s23.github.io/homework/Titanic.csv")
```

Here `read.csv` is a function that imports data from a CSV file. We can pass `read.csv` either a local path on our computer, or a URL – in this case, we use the URL where the data is stored online. We have called the data `titanic` in R.

Copy the command to load the data into the setup chunk of your R Markdown file, and run it.

1 Hypothesis testing

In the first part of this assignment, we will fit logistic regression models and test hypotheses about the Titanic data. We are interested in testing whether there is a relationship between Age and the probability of survival for passengers on the Titanic. For now, we will ignore the other variables, and fit a model with only Age as the explanatory variable:

$$Survived_i \sim Bernoulli(\pi_i)$$
$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 Age_i$$

1. We want to formally test whether there is a relationship between Age and Survival using this model. Write down null and alternative hypotheses, in terms of one or more model parameters, for this test.

2. The null hypothesis corresponds to a reduced model for the data, and the alternative hypothesis corresponds to a full model. Write down the reduced and full models corresponding to your hypotheses in question 1.

3. Now let's fit the models. Remember to deal with missing data before you fit the models!

   (a) Fit the reduced model on the titanic data, and report the equation of the fitted model.

   (b) What is the deviance for the reduced model? (This will be the *residual deviance* in R)

   (c) Fit the full model on the titanic data, and report the equation of the fitted model.

   (d) What is the deviance of the full model? (the residual deviance in R)

4. You will notice that both the full and reduced models also have a *null deviance* reported in R. The **null deviance** is the deviance of the *intercept-only* model for the data. Explain why in this case, the null deviance is the same as the residual deviance of the reduced model (they won't always be the same).

5. To test our hypotheses from question 1, we will compare the deviance of the full and reduced models. Calculate the drop-in-deviance G: the deviance of the reduced model − the deviance of the full model.

6. If our observed G is unusual under H_0, this is evidence against the null hypothesis. This means that we need the distribution of G under the null hypothesis H_0. In class, we used the following code to simulate the null distribution of G for the grad admissions data:

```
nsim <- 500
null_statistics <- rep(0, nsim)
for(i in 1:nsim){
  x <- grad_app$gre
  p <- exp(-0.77 + 0*x)/(1 + exp(-0.77 + 0*x))
  y <- rbinom(length(x), 1, p)
  m1 <- glm(y ~ x, family = binomial)

  null_statistics[i] <- m1$null.deviance -
    m1$deviance
}
hist(null_statistics)
```

(a) Adapt the code from class to simulate the null distribution of G for the titanic data. Run the corrected code to show the histogram of G in the simulations.

(b) How unusual is our observed test statistic from question 5, compared to our simulated null distribution from 6(a)?

(c) Finally, let's calculate an approximate p-value. Our p-value is the probability of "our data or more extreme", if H_0 is true. So, we can approximate a p-value using the simulated null distribution. Our approximate p-value is the fraction of simulations in which G was greater than or equal to the observed value in the data. Fill in the following code, replacing `...` with your drop-in-deviance from question 5:

```
mean(null_statistics >= ...)
```

How unusual is our observed data, if there is no relationship between Age and Survival?

2 Practice with for loops

The purpose of this section is to give you some more practice working with for loops and sequences, which are useful tools for efficiently repeating a process many times. Here is an example for loop that calculates x^2 for a sequence of numbers $x = 0, 0.1, 0.2, ..., 0.9, 1$:

```
x <- seq(0, 1, 0.1)
x_squared <- rep(0, length(x))
for(i in 1:length(x)){
  x_squared[i] <- x[i]^2
}
x_squared
```

All `for` loops consist of three parts:

- the *output* is the values we want to calculate with the for loop (in this case, `x_squared`)
- the *sequence* is the sequence that we iterate over. In this case our sequence is `1:length(x)`, i.e. the values 1, 2, 3, ..., `length(x)`. For each value in the sequence, we perform a calculation in the for loop.
- the *body* is what we do at each iteration of the loop (i.e., what goes between the curly braces `{ }`). In this case, we are calculating x^2 at each iteration

Below are some short practice questions to help you get more comfortable creating `for` loops.

7. Modify the loop above so that instead of calculating x^2, we calculate x^3
8. Modify the loop above so that instead of considering $x = 0, 0.1, 0.2, ..., 0.9, 1$ (i.e. the numbers between 0 and 1, in increments of 0.1), we consider $x = 0, 0.05, 0.10, 0.15, ..., 1.95, 2$ (the numbers between 0 and 2, in increments of 0.05).
9. Suppose we want to simulate a single draw from a normal distribution with mean 0 and standard deviation 1. The code in R is

   ```
   rnorm(n=1, mean=0, sd=1)
   ```

(a) Fill in the following `for` loop to create a sample of 1000 draws from a normal distribution and plot a histogram of the results:

```
n <- ...
normal_sample <- rep(0, n)
for(i in 1:n){
  normal_sample[i] <- ...
}
hist(normal_sample)
```

(b) Does the histogram look like a normal distribution?

10. Now let's simulate from a different distribution.

(a) Modify your loop from the previous question to simulate 2000 samples from a χ^2_1 distribution (a χ^2 distribution with 1 degree of freedom) and plot a histogram of the results. The code in R for a single draw from a χ^2_1 distribution is

```
rchisq(n=1, df=1)
```

(b) How does the χ^2_1 distribution compare to your simulated test statistics in Part 1?

3 Extra credit: debugging practice

This part is separate from parts I and II. The purpose of this section is to practice debugging the errors that we sometimes encounter in R and RStudio. This part is *optional*, and a correct submission will earn a small number of extra credit points on the assignment.

11. Your friend is trying to calculate a likelihood $L(\pi_0)$ for different values of π_0 between 0 and 1. Their likelihood function is $L(\pi_0) = \pi_0^2(1-\pi_0)^3$. They are using the following code to calculate the likelihood for $\pi_0 = 0, 0.1, 0.2, ..., 0.9, 1$.

```
pi0 <- seq(0, 1, 0.1)
likelihood <- rep(0, 12)
for(i in 1:12){
  likelihood[i] <- pi0[i]^2 * (1 - pi0[i])^3
}
likelihood
```

However, when they run the code in R, they get `NA` for the final likelihood!

(a) Why does your friend get `NA` for the final likelihood? (Hint: think about the indexes of `pi0` and `likelihood`)

(b) Fix your friend's code so that the `NA` value no longer appears, and the likelihoods are all correctly calculated.