The RMS Titanic was an ocean liner that struck an iceberg and sank 14/15 April 1912 on its maiden voyage to New York. More than 1,500 of the estimated 2,224 passengers and crew died in the accident making this one of the largest maritime disasters outside of war. The ship's passengers, of course, varied in age and sex, and included luxury travelers in first-class and poor immigrants in third-class. However, not all passengers were equally likely to survive the accident. We use real passenger data (a subset) to learn who were more likely to survive.

Using github or the class website, find and load the file `titanic.csv` into a data.frame `df`.

1. Describe `df`: number of observations, columns and the data type of those columns. The column meanings should be clear except perhaps for `Pclass`, which encodes the type of ticket: 1^{st} class, 2^{nd} class, 3^{rd} class (first class being the expensive ticket). `summarize` the data; record the numbers of `Survived`, male/female, 1/2/3 class, average `Age` and `Fare`.

2. Calculate the below probabilities from the corresponding whole number ratios. Here S=survived, G=gender, C=passenger class

   (a) $P(S \mid G ==$ "female"$)$ — i.e., the probability you survived given you are female. Do note that this is not the same as $P(S \;\&\; G ==$ "female"$)$

   (b) $P(S \mid G ==$ "female" $\&\; C == 1)$ (the probability first-class females survived).

   (c) $P(S \mid G ==$ "female" $\&\; C == 2)$

   (d) $P(S \mid G ==$ "female" $\&\; C == 3)$

   (e) $P(S \mid G ==$ "male"$)$

   (f) $P(S \mid G ==$ "male" $\&\; C == 1)$

   (g) $P(S \mid G ==$ "male" $\&\; C == 2)$

   (h) $P(S \mid G ==$ "male" $\&\; C == 3)$

   You can certainly calculate the above quantities by careful subsetting, using `sum` to count cases of logical variables and then forming the quotient, but there are much faster and easier methods to get answers (but a bit advanced, and not really part of this course). `xtabs` seeks to explain one variable by cross tabulating (breaking up) the cases according to the values in some other columns. This is expressed with a "formula" where the variable to be explained (the y to be explained) is expressed as a formal sum of the variables that will do the explaining (one or more xs). A formula then is of the form: `y~x` ("y tilde x"). The answer produced is the sum of the y in those cases (for a logical variable this is the count of `T`) for each of the x possibilities.

```
> xtabs(Survived~Sex+Pclass,data=df)
        Pclass
Sex       1  2  3
  female 91 70 72
  male   45 17 47

> xtabs(!Survived~Sex+Pclass,data=df)
        Pclass
Sex        1   2   3
  female   3   6  72
  male    77  91 296
```

   The first table says 91 first-class females survived and the second says 3 first-class females died. If there is no y in the formula (e.g., `~Sex+Pclass+Survived`) `xtabs` will count the occurance of all cases, in this case making a 3d contingency table (two 2d tables to be stacked).

A second method uses the data plyer package, loaded by `library(dplyr)`. This package adds lots of data wranging functionality, and I use it ALL the time, but it's not really part of class. A basic function is a pipe: `%>%` that takes the output of the previous command and sends it along to a following command. Just one nice aspect of this is if you put a data.frame into the pipe, you don't have to repeat that data.frame name when referring to columns. Here's what this looks like:

```
> library(dplyr)

> df %>% group_by(Sex,Pclass) %>%
summarize(mean(Survived),sum(Survived),length(Survived))
# A tibble: 6 x 5
# Groups:   Sex [2]
  Sex    Pclass `mean(Survived)` `sum(Survived)` `length(Survived)`
  <fct>   <int>            <dbl>           <int>               <int>
1 female      1            0.968              91                  94
2 female      2            0.921              70                  76
3 female      3            0.5                72                 144
4 male        1            0.369              45                 122
5 male        2            0.157              17                 108
6 male        3            0.137              47                 343
```

Here you see the often-used trick that the average of a logical variable is the proportion that is `T`, the sum of a logical is the count of `T`, and the length of a vector is the count of all the possibilities.

3. A basic property of probability is

$$P(A) = \sum_i P(A|B_i)P(B_i)$$

where the B_i disjointly cover all possibilities. Test this out for female survival (A) expressed in terms of three `Pclass` (B_i, $i = 1, 2, 3$). Write each probability in this expression in terms of an integer number of females (use the results above) in the numerator and denominator. See that the result must be true by simple algebra.

4. Bayes Theorem says:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Let A be that the passenger is 1^{st} class ($C = 1$), and B be that the passenger survived ($S =$`T`). Using Bayes Theorem, find the probability that a passenger was first-class, given they survived. As above write out all four probabilities in the above equation as integer ratios and show that the result must be true by simple algebra.

5. Events A and B are independent if

$$P(A \,\&\, B) = P(A)P(B)$$

If A=survived and B=($G ==$ "female"), show that the above does not hold so the two are not independent. As above express each probability as a ratio of integers.

6. A glance at the numbers should be enough to convince you that survival outcome was not independent of class, but sometimes a simple statistic can end all argument (or not[1]). Fisher

[1] I've observed both people being convinced by statistics they do not understand (successful bullying by stats), and people remaining unconvinced in spite of statistics they do understand (pigheadedness).

Exact test and/or χ^2 test reduce a contingency table to a p value for independence. (These tests have different supports and do not typically produce the same p value, but the p values should tell the same story.) χ^2 is not appropriate if there are small counts-in-cell. Let's make survival contingency tables separately for females and males, and record the resulting p value.

```
> fout=df %>% filter(Sex=="female") %>% xtabs(~Survived+Pclass,data=.)
> fout
        Pclass
Survived  1  2  3
   FALSE  3  6 72
   TRUE  91 70 72

> mout=df %>% filter(Sex=="male") %>% xtabs(~Survived+Pclass,data=.)
> mout
        Pclass
Survived   1   2   3
   FALSE  77  91 296
   TRUE   45  17  47

> a=fisher.test(fout)
> b=chisq.test(mout)
```

Report the p-values. There is not much additional a Fisher test *can* report (there is no simple odds ratio for this 2×3 table), but χ^2 test can report the 'expected' counts-in-cell under the assumed independence. Examine the structure (`str`) of the χ^2 result and look at the expected contingency table. Where do you see big differences between actual and expected?

7. Age is a continuous[2] (rather than categorical) variable so we can't proceed as above. One option is to make it categorical by breaking up the range of possible ages into bins say: child(0): `Age` $\in [0, 9)$, tween(1): `Age` $\in [9, 13)$, teen(2): `Age` $\in [13, 20)$, twentyish(3): `Age` $\in [20, 30)$, ... Once we have a categorical variable we can make a table of counts in each bin. Unlike the above example, bins are usually the same width, so a uniform age distribution would put the same number of counts in each bin. A plot of count-in-bin vs. bin-center is called a histogram (`hist`, RCook p. 248). (Typically many bins will be used so a graphical display of results is nicer than tabular results, but you can access the numerical data e.g., from `$counts`, `$mids`, etc.) R will automatically select bins (you can change it if you want: `breaks`) and plot bars with height giving count-in-bin with the command `hist`. (By default counts-in-bin is displayed; if you want proportion-in-bin try `freq=F`.)

   Now we're going to want to be comparing age distributions so we are required to make plots with identical settings so they can be nicely superimposed. Since there are more males than females, proportion rather than count will most often be of interest. (The proportion-in-bin divided by bin-size—called density—will integrate to 1; i.e., the are under a histogram is fixed at 1.) To get density we must turn off the default count display by setting: `freq=F`. If we aim to superimpose the plots we'll want the colors to be somewhat transparent; this is called alpha blending. The fourth argument in `rgb` is the opaqueness, α; the first three are the amounts of red, green, and blue. We'll use blue for boys and red for girls.

```
> pM=hist(df$Age[df$Sex=="male"],xlim=c(0,80),breaks=seq(0,80,8),freq=F)
> pF=hist(df$Age[df$Sex=="female"],xlim=c(0,80),breaks=seq(0,80,8),freq=F)
> plot(pM, col=rgb(0,0,1,1/4), freq=F)
> plot(pF,col=rgb(1,0,0,1/4),freq=F,add=T)
```

   (a) Using the above code, describe (words!) how the age distributions differ by sex.

[2] In this dataset almost all the ages are integers, so `Age` seems discrete. We ignore this.

(b) Now restrict the male/female data to just the first-class passengers... what changes?

(c) Now compare the age distribution of males in first-class to males in third-class

(d) Now compare survivors of all sexes to the dead.

(e) Now compare the *counts* of all males and male survivors. The difference in the the bars counts the dead. You can judge by eye the proportion that survived in each age bin.

(f) Now break this male survive/all histogram data down by `Pclass` and describe the results

i. in just first class: age histogram of counts of all males and male survivors.

ii. in just second class: age histogram of counts of all males and male survivors.

iii. in just third class: age histogram of counts of all males and male survivors.

(g) Now compare the counts of all females and female survivors.

(h) (FYI) smoothed histograms a.k.a. density plots are easy in R: `plot(density(df$Age))` and multiple density plots can be combined: `lines(density(df$Age[df$Sex=="female"]))` (RCook p. 250).

8. Recall that the t-test (`t.test`) is used to look for differences in the mean of datasets. Does the mean age of first-class women differ significantly from that of first-class men? Does the mean age of first-class men differ significantly from that of third-class men?

9. Boxplots[3] (RCook p. 246) are produced if `plot` is given a factor as the x value. In this dataset we essentially have no factors so we must either convert say `Pclass` or `Survived` to a factor using `as.factor`, or we can directly plot a boxplot using the `boxplot` command. Note that `boxplot` uses the formula format discussed above to denote the x and y values to be plotted. So, for example,

```
> boxplot(df$Fare~df$Pclass)
```

displays how fares vary with passenger class. Notice that the bottom whisker seems to touch zero for each class; additionally note that, particularly for third-class, things are getting scrunched together. To spread out wide ranging values, log scales are suggested, but with zero values of `Fare` log transformations are impossible (NaN). Create a data.frame `df2` in which the zero-fare rows have been deleted. Now:

```
> boxplot(log(df2$Fare)~df2$Pclass)
```

Plotting the $\log(y)$ solves the scrunched problem, but it obscures the actual values of the data. To label the log-scale with the actual data values use the parameter `log` as below.

```
> boxplot(df2$Fare~df2$Pclass,log="Y")
```

Drilling down a bit deeper note that the minimum (non-zero) fare in first-class was an outlier (and hence separately displayed) for the log transformation, whereas it was just the minimum (and not exceptional) for the normally labeled plot. We leave this bit of arcanea behind.

Find and report who paid that low price for first-class, and did (s)he survive? I'd suggest the R command `which.min` which reports where (the integer index) the minimum is found. (`min` finds the minimum value, but I'm not requesting that.)

[3] see also: `http://www.physics.csbsju.edu/stats/display.distribution.html`