Capstone_II_Titanic

Berkeley Loveless

6/1/2019

Capstone Project II - who lives and who dies on the Titanic

Summary

The objective of the project is to predict whether a passenger on the Titanic survived or not based on criteria in the titanic data set included in the R Package "Titanic." This data set includes training set of 891 observations and a test set of 481 observations. The training set includes a variable indication whether an observation survived. This is not included in the test set and must be submitted to Kaggle to get the results of any predictions. The goal for this project is to predict who survived in the test data set with an accuracy greater than the baseline of predicting by sex only, and to land within the top twenty-five percent of submissions to the current Kaggle competition at the time of the submission.

The data are described in detail in the body of this document. In summary, the data describe passengers on the Titanic's tragic voyage. Variables include a passenger ID (PassengerId), the name of the passenger (Name), the ticket class (Pclass), sex (Sex), age (Age), ticket number (Ticket), passenger fare (Fare), cabin number (Cabin), port of embarkation (Embarked), and variables indicating the number of siblings or spouses (SibSp) and the number of parents or children (Parch) traveling with the person named, and whether the person named survived or not (Survived).

The procedure used to predict survival in the test set was to first examine the data in the training set to see if there were groupings that may appear as part of the basic understanding of attributes of people in the early twentieth century that may lead to better statistical analysis of the data. Second, a look at posterior density using the bayestestR package was used to choose the elements of the prediction model. Last, the carat package was used to choose a method of prediction. The prediction was formatted into a csv file according the specifications of the Kaggle competition "Titanic: Machine Learning from Disaster" and submitted to Kaggle for a score and ranking.

The bayesR package and method used for model component selection

I read an article on *R-blogger* about a relatively new package called "bayestestR" in the post *Describe and understand Bayesian models and posteriors using bayestestR. April 14, 2019. By R on easystats. https://www.r-bloggers.com/describe-and-understand-bayesian-models-and-posteriors-using-bayestestr/* I decided I would try to use the techniques in the article to look at posterior densities of the titanic data set in a very simplistic way. I would use the posterior densities not so much as analyzing the actual density or distribution of the component, bur rather just simply the rejection or acceptance of the null hypothesis that there is no significant difference between specified populations in determining survival. When I mention the null hypothesis in the following analysis, this is what I am talking about. Those elements that were accepted as elements consistent with the null hypothesis were rejected from the model. Those that were rejected as consistent with the null hypothesis were included in the model. Those undecided by the bayesR analysis were further analyzed and used primarily for clues in data modeling.

Results

Examination of the data led to the creation of two derived variables: "Title," the honorific contained in the passenger name, and "adultMale" indicating whether an observation was an adult male or not. In

summary, the Title was useful in deciding on model elements as it showed that although sex was the primary determinant of survival, males with the Title "Master," indicating infancy. This led to the creation of the adultMale variable that put adult males into a different category than women and children. The adultMale variable, along with passenger class (Pclass), embarkation port (Embarked), the number of siblings or spouses (SibSp), the number of parents or children (Parch) were used in the prediction model. The method "rf," a random forest algorithm was used as it gave the best results predicting survival in the training set. R 3.6.0 was used.

Outcome

A score of .79425 and a ranking of 2,214 out 11,322 was received from Kaggle. The baseline of prediction by sex only received a score of .76555. As the score showed an improvement over the sex only prediction and was within the top 20th percentile, the prediction met the goals of the project.

Process of prediction

Load the test and training sets

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.2     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------ 
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
if(!require(lda)) install.packages("lda", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: lda
```

```
if(!require(titanic)) install.packages("titanic", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: titanic
```

The R package "titanic" contains a pre-made training and test set

Load the test and training sets

```
data(titanic_test)
data(titanic_train)
```

Checking out the data

```
glimpse(titanic_train)
```

```
## Observations: 891
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3,...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bra...
## $ Sex         <chr> "male", "female", "female", "female", "male", "mal...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, ...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4,...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1,...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1138...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", ...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", ...
```

Besides variables that are well described by the name of the variable, there are a few variables that are less clear are described on the Kaggle website (https://www.kaggle.com/c/titanic/data):

- Pclass - the ticket class of the passenger and a proxy for socio-economic status (SES)
  - 1st = Upper
  - 2nd = Middle
  - 3rd = Lower
- SibSp - number of siblings / spouses aboard the Titanic
  - Sibling = brother, sister, stepbrother, stepsister
  - Spouse = husband, wife (mistresses and fiancés were ignored)
- Parch - number of parents / children aboard the Titanic
  - Parent = mother, father
  - Child = daughter, son, stepdaughter, stepson
    - Some children traveled only with a nanny, therefore parch=0 for them
- Embarked - Port of embarkation
  - C = Cherbourg
  - Q = Queenstown
  - S = Southampton

Some data attributes worth note include:

- Survived is a 0 or 1, with 1 indicating survival.
- Name is in a format of Last Name, a comma, honorific with a period and first name with some other things.
- Age is missing some entries
- Ticket has no real pattern
- Cabin is missing data but is not "NA." The missing data is indicated by two double quotes "".

Checking completeness by checking for nulls (na) and blanks ("")

```
missing <- colSums(titanic_train == "" | is.na.data.frame(titanic_train))
```

Getting the number of missing data in each column

```
missing
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Getting the percent of data missing in each column

```
missing/891
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.198653199
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
## 0.000000000 0.000000000 0.000000000 0.000000000 0.771043771 0.002244669
```

Conclusion:

With Cabin missing 77% of the data, I'm going to exclude it from analysis.

Embarked Data Issue

There are a small number of records missing the embarkation data. A proxy is needed if this variable is going to be used in prediction.

Getting general populations to see what should be done about the missing Embarked data

```
pops <- titanic_train %>% group_by(Sex = factor(Sex), Class = factor(Pclass),
                                   Embarked = factor(Embarked),
                                   SibSp = factor(if_else(SibSp>0,1,0))) %>%
                                    summarize(total_num = n())
```

Looking at percent of populations

Excluding Embarked to see if there are any major factors that would indicate the Embarkation point if missing.

```
SexClassE <- inner_join(filter(pops, Embarked != ""), pops %>%
                          filter(Embarked != "") %>%
                          group_by(Sex, Class, SibSp) %>%
                          summarize(tot = sum(total_num)),by = c("Sex", "Class", "SibSp"))
```

Seeing if there is a majority with each group

If there is a majority in each group, that could be used as a reasonable proxy for the missing information.

```
filter(SexClassE %>%
         group_by(Sex, Class, Embarked, SibSp) %>%
         summarise(Pct = sum(total_num/tot)),Pct>.5)
```

```
## # A tibble: 12 x 5
## # Groups:   Sex, Class, Embarked [6]
##    Sex    Class Embarked SibSp   Pct
##    <fct>  <fct> <fct>    <fct> <dbl>
##  1 female 1     S        0     0.511
##  2 female 1     S        1     0.533
##  3 female 2     S        0     0.909
##  4 female 2     S        1     0.844
##  5 female 3     S        0     0.506
##  6 female 3     S        1     0.746
##  7 male   1     S        0     0.636
##  8 male   1     S        1     0.676
##  9 male   2     S        0     0.921
## 10 male   2     S        1     0.844
## 11 male   3     S        0     0.767
## 12 male   3     S        1     0.753
```

Taking a closer look at 1st class females

Since 1st class females are so close, other information may be helpful.

```
filter(SexClassE %>%
         group_by(Sex, Class, Embarked, SibSp) %>%
         summarise(Pct = sum(total_num/tot)),
       Sex == "female", Class == 1)
```

```
## # A tibble: 5 x 5
## # Groups:   Sex, Class, Embarked [3]
##   Sex    Class Embarked SibSp    Pct
##   <fct>  <fct> <fct>    <fct>  <dbl>
## 1 female 1     C        0     0.489
## 2 female 1     C        1     0.444
## 3 female 1     Q        1     0.0222
## 4 female 1     S        0     0.511
## 5 female 1     S        1     0.533
```

What is the look of the missing data?

```
pops %>% filter(Embarked == "")
```

```
## # A tibble: 1 x 5
## # Groups:   Sex, Class, Embarked [1]
##   Sex    Class Embarked SibSp total_num
##   <fct>  <fct> <fct>    <fct>     <int>
## 1 female 1     ""       0             2
```

Unfortunately, it's the 51/48 S/C. . . one more quick look - Fares for 1st class females.

```
titanic_train %>% filter(Sex == "female", Pclass == 1) %>%
  group_by(Embarked) %>%
  summarize(avg_fare = mean(Fare))
```

```
## # A tibble: 4 x 2
##   Embarked avg_fare
##   <chr>       <dbl>
## 1 ""           80
## 2 C           116.
## 3 Q            90
## 4 S            99.0
```

Conclusion:

Although the 1st class female situation is close, there is a better than 50% chance that under any group of Sex and Pclass that they embarked from S, so it's pretty safe to assign any missing Embarked data to S. Additionally the fare indicates an S or Q embarkation and Q is very unlikely so S remains a good proxy.

It's not at all clear at this point whether the Embarked data is a type of determinant so that makes this assumption safer if it isn't. If it is, and the data are missing in the test set, a closer look may be warranted. However, at this point it is premature.

Filling in the null values so Embarked can used with the caret package.

```
titanic_train <- titanic_train %>%
  mutate(Embarked = if_else(Embarked == "","S",Embarked))
```

Checking out the data visually

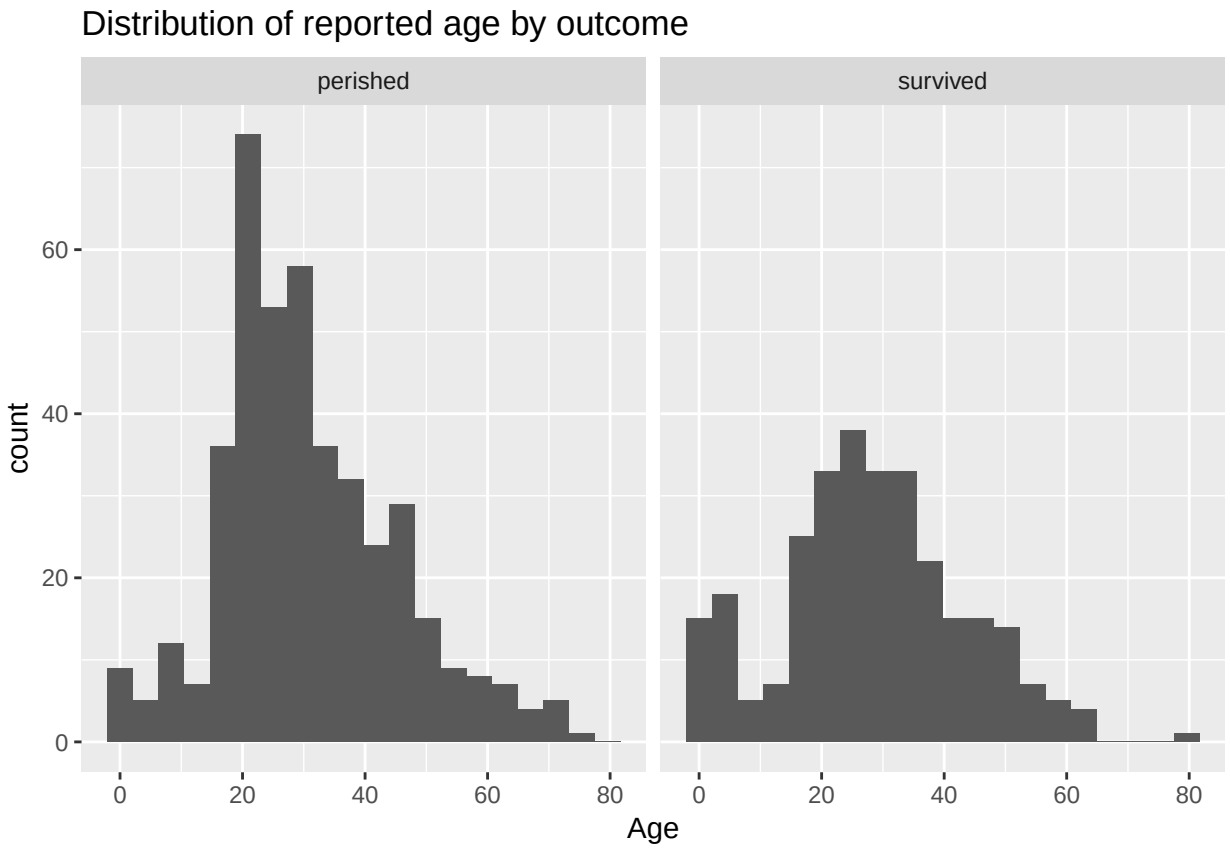
Creating labels to use for Survived and Pclass

Creating labels for "perished" and "survived" and passenger class to make charts easier to read.

```
survived_labels <- as_labeller(c('0' = "perished", '1' = "survived"))
class_labels <- as_labeller(c('1' = "1st class", '2' = "2nd class", '3' = "3rd class"))
```

Histogram of the age faceted by Survival

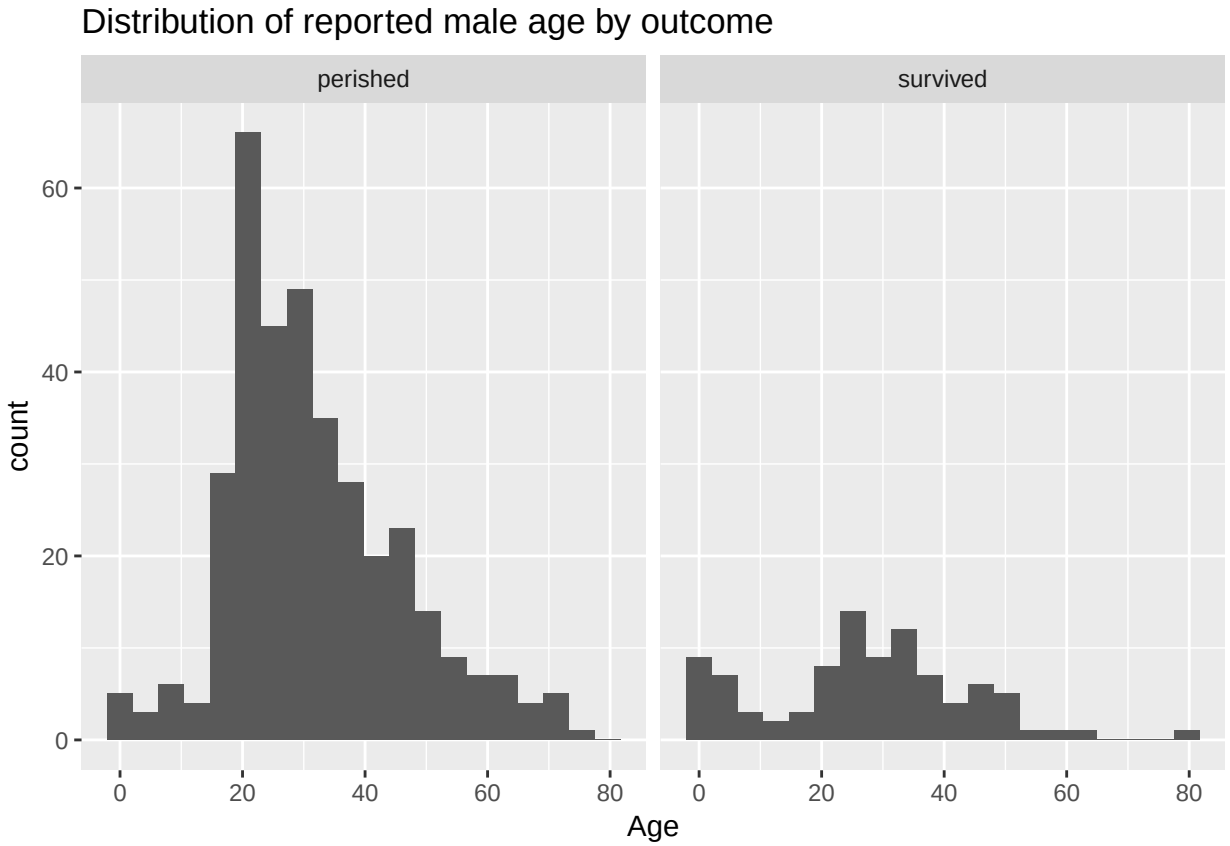
```
titanic_train %>% ggplot(aes(x = Age), rm.na = TRUE) +
  geom_histogram(bins = 20) +
  ggtitle("Distribution of reported age by outcome") +
  facet_grid(~Survived, labeller = survived_labels)
```



Since age was missing approximately twenty percent of the data, the first task is to look and see if it may be a determinate for survival. At this first look, there seems to be a larger number of children that survived compared to adults. Next step is to look at the same thing for males only.

Looking at the same but for only men.

```
titanic_train %>% filter(Sex == 'male') %>%
  ggplot(aes(x = Age), rm.na = TRUE) +
  geom_histogram( bins = 20) +
  ggtitle("Distribution of reported male age by outcome") +
  facet_grid(~Survived, labeller = survived_labels)
```



On first glance, the data are stunning. I'm not ashamed to say I was personally choked up looking at this chart. The survival rate for men is very low. There is an indicator that male children survived at a better rate than adult males. Next step is to look at the same thing separated by class.

Adding Pclass

```
titanic_train %>% filter(Sex =='male') %>%
  ggplot(aes(x = Age), rm.na = TRUE) +
  geom_histogram(bins = 20) +
  ggtitle("Distribution of reported male age by class and outcome") +
  facet_grid(Survived ~ Pclass, labeller =
               labeller(Survived = survived_labels, Pclass = class_labels))
```

Again, the data are stunning. With the exception of the third class children, survival rates for males is abysmal. The number of third class deaths outpaced the other classes, but the population of third class men was the largest. Nearly every second class adult male died. However, the purpose of this histogram is primarily to look at the impact of age as a determinant. Although third class children died at a much higher rate than the other classes, they seem to have survived at a higher rate than men in general. Next step is to look at the rates of survival by age to eliminate the variance in populations.

Male survival rates by Age

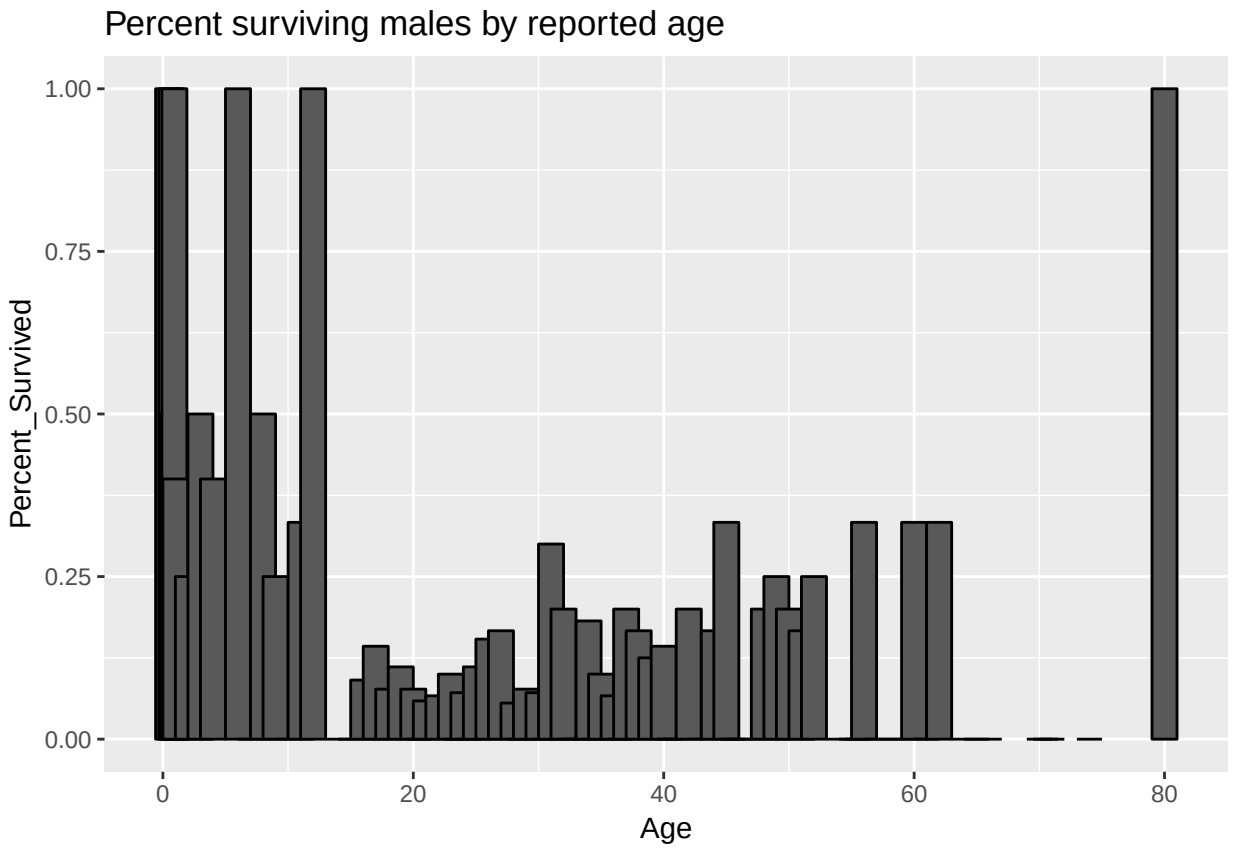
```
srates <- titanic_train %>% group_by(Sex, Age, Pclass, Survived) %>%
  summarize(Total=n())
srates %>% filter(Sex == 'male') %>% group_by(Age) %>%
  summarize(Percent_Survived = sum(Survived)/sum(Total))
```

```
## # A tibble: 83 x 2
##      Age Percent_Survived
##    <dbl>            <dbl>
##  1  0.42                1
##  2  0.67                1
##  3  0.83                0.5
##  4  0.92                1
##  5  1                   0.4
##  6  2                   0.25
##  7  3                   0.5
##  8  4                   0.4
##  9  6                   1
```

```
## 10  7                0
## # ... with 73 more rows
```

```
srates_percent <- srates %>% filter(Sex == 'male') %>%
  group_by(Age) %>%
  summarize(Percent_Survived = sum(Survived)/sum(Total),
            Total_at_Age = sum(Total))

ggplot(srates_percent, aes(x = Age, y = Percent_Survived)) +
  geom_col(width = 2, color = "black", na.rm = TRUE) +
  ggtitle("Percent surviving males by reported age")
```



The data indicate a very low survival rate for males between around 14 to 75. Male children have a higher rate of survival. The next step is to take a look at the honorific. Honorifics are part of what may be difficult for a machine algorithm to understand unless it has a reference. In this case, although female age cannot be determined by an honorific, male age can. The term "Master" is used to indicate infancy (not having reached the age of a "man") in males, at least it was at the turn of the 20th century. Title may also give other indicators such as military or religious duties. A military adult male may have been desired to captain a lifeboat so may have been more likely to survive. A member of the clergy may have had a lower survival rate as their duty may have been to remain to comfort the doomed. However, this is all conjecture at this point. Title is between first comma(,) and first dot (.). Regex gsub() will be used to extract the honorific and the result will be stored in the variable "Title."

```
titanic_train <- titanic_train %>%
  mutate(Title = gsub('(.*, )|(\\..*)','', titanic_train$Name))

table(titanic_train$Title)
```

```
##
```

```
##         Capt          Col          Don           Dr     Jonkheer
##            1            2            1            7            1
##         Lady        Major       Master         Miss         Mlle
##            1            2           40          182            2
##          Mme           Mr          Mrs           Ms          Rev
##            1          517          125            1            6
##          Sir the Countess
##            1            1
```

Looking at survival rates by Title

```
titanic_train %>% group_by(Title) %>% summarize(survived = sum(Survived),
                                                perished = (n()-survived),
                                                pct_survived = (survived/n()*100)) %>%
  knitr::kable()
```

Title	survived	perished	pct_survived
Capt	0	1	0.00000
Col	1	1	50.00000
Don	0	1	0.00000
Dr	3	4	42.85714
Jonkheer	0	1	0.00000
Lady	1	0	100.00000
Major	1	1	50.00000
Master	23	17	57.50000
Miss	127	55	69.78022
Mlle	2	0	100.00000
Mme	1	0	100.00000
Mr	81	436	15.66731
Mrs	99	26	79.20000
Ms	1	0	100.00000
Rev	0	6	0.00000
Sir	1	0	100.00000
the Countess	1	0	100.00000

These data show that adult males had a very low rate of survival in general. Married women had a better rate of survival than unmarried women, which included female children. Male children represented by the title "Master" survived at a rate significantly higher than adult males, but lower than females in general. Military officers fared better than average adult males at a rate of 50% for Col and Major. However, there numbers are so low it isn't worth looking at. Besides, to put together a list of all military titles for an unknown data set would be a large effort for a statistic that is so tenuous. Other observations include populations that were not large enough a solid statistical inference. However, they are interesting. All clergy (Rev) perished. Among those with foreign (non-English) honorifics (Don, Jonkheer, Mlle, Mme) all the men perished, while all the women survived. All royalty (Lady, Sir, the Countess) survived. Doctors survived at more than double the rate of the Title "Mr."

In the end, titles with significant populations are Master, Miss, Mr and Mrs. Sex seems to be the overwhelming determinant. Finding the others is the remaining task to create the model. Before that, the actual percent of survival by Sex alone will be calculated for a baseline.

Looking at survival rates by Sex for baseline

```
titanic_train %>% group_by(Sex) %>% summarize(survived = sum(Survived),
                                              perished = (n()-survived),
                                              pct_survived = (survived/n()*100)) %>%
  knitr::kable()
```

Sex	survived	perished	pct_survived
female	233	81	74.20382
male	109	468	18.89081

Checking out a baysian model

A number of techniques have been used by Kaggle competitors on the project "Titanic: Machine Learning from Disaster." I personally wanted to try something different. I read an article on *R-blogger* about a relatively new package called "bayestestR" in the post *Describe and understand Bayesian models and posteriors using bayestestR. April 14, 2019. By R on easystats. https://www.r-bloggers.com/describe-and-understand-bayesian-models-and-posteriors-using-bayestestr/* I decided I would try to use the techniques in the article to look at posterior densities of the titanic data set. My process is to look start with a large number of elements in a model and first eliminate elements that do not reject the null hypothesis, and then quiet the noise by iteratively eliminating large determinants. In the end I hope to have the right elements in a model to achieve the project goals.

Starting Title, Age, Sex, Pclass, Fare, Embarked, SibSp, and Parch

Starting with a model that excludes only Name and Ticket because they seemed too unique for predictive purposes, and Cabin because it was missing too much data.

```
model <- stan_glm(Survived ~ Title + Age + Sex + Pclass + Fare + Embarked + SibSp + Parch,
                  refresh = 0, data = titanic_train)
equivalence_test(model)
```

```
## Possible multicollinearity between TitleMr and TitleCol (r = 0.7), TitleMaster and TitleDr (r = 0.8)
```

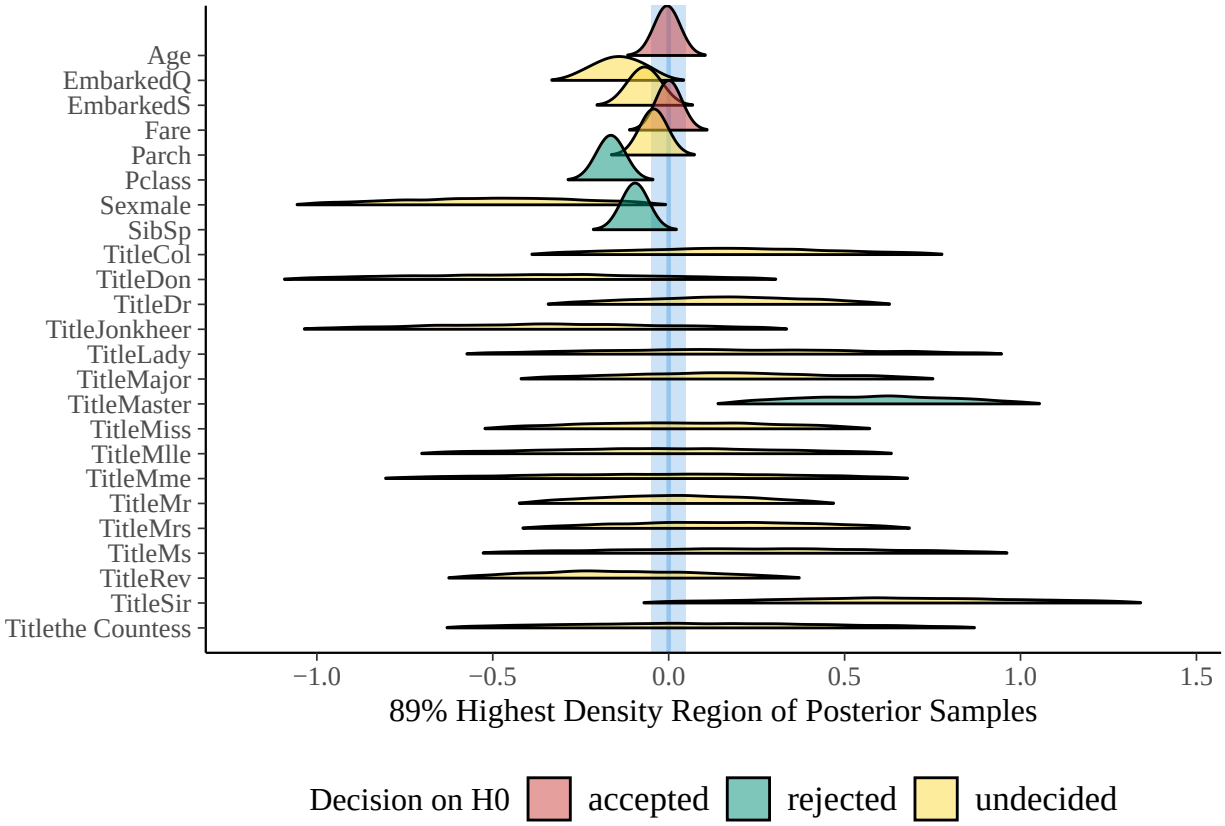
```
## # Test for Practical Equivalence
##
##   ROPE: [-0.05 0.05]
##
##          Parameter        H0 inside ROPE       89% HDI
##        (Intercept)  rejected      0.00 % [ 0.75  1.82]
##           TitleCol undecided     10.98 % [-0.37  0.75]
##           TitleDon undecided      6.63 % [-1.07  0.29]
##            TitleDr undecided     11.99 % [-0.31  0.59]
##      TitleJonkheer undecided      7.05 % [-1.03  0.31]
##          TitleLady undecided      9.41 % [-0.56  0.93]
##         TitleMajor undecided     10.92 % [-0.40  0.73]
##        TitleMaster  rejected      0.00 % [ 0.17  1.03]
##          TitleMiss undecided     12.38 % [-0.50  0.55]
##          TitleMlle undecided     10.61 % [-0.68  0.61]
```

```
##            TitleMme undecided      8.93 % [-0.80  0.66]
##             TitleMr undecided     16.74 % [-0.39  0.43]
##            TitleMrs undecided     12.50 % [-0.39  0.66]
##             TitleMs undecided      7.83 % [-0.51  0.95]
##            TitleRev undecided     12.58 % [-0.59  0.35]
##            TitleSir undecided      3.59 % [-0.05  1.33]
##  Titlethe Countess undecided      9.83 % [-0.61  0.85]
##                 Age  accepted    100.00 % [-0.01 -0.00]
##             Sexmale undecided      0.34 % [-1.03 -0.04]
##              Pclass  rejected      0.00 % [-0.20 -0.13]
##                Fare  accepted    100.00 % [-0.00  0.00]
##           EmbarkedQ undecided      7.16 % [-0.28 -0.02]
##           EmbarkedS undecided     28.14 % [-0.13 -0.01]
##               SibSp  rejected      0.00 % [-0.12 -0.07]
##               Parch undecided     65.40 % [-0.07 -0.01]
```

```
plot(equivalence_test(model))
```

```
## Possible multicollinearity between TitleMr and TitleCol (r = 0.7), TitleMaster and TitleDr (r = 0.8)
```

```
## Picking joint bandwidth of 0.037
```



This plot shows that Age and Fare are not rejected from the null hypothesis, and are therefore not determinants of Survival.

Removing Age and Fare

```
model <- stan_glm(Survived ~ Title + Sex + Pclass + Embarked + SibSp + Parch,
                  refresh = 0, data = titanic_train)
equivalence_test(model)
```

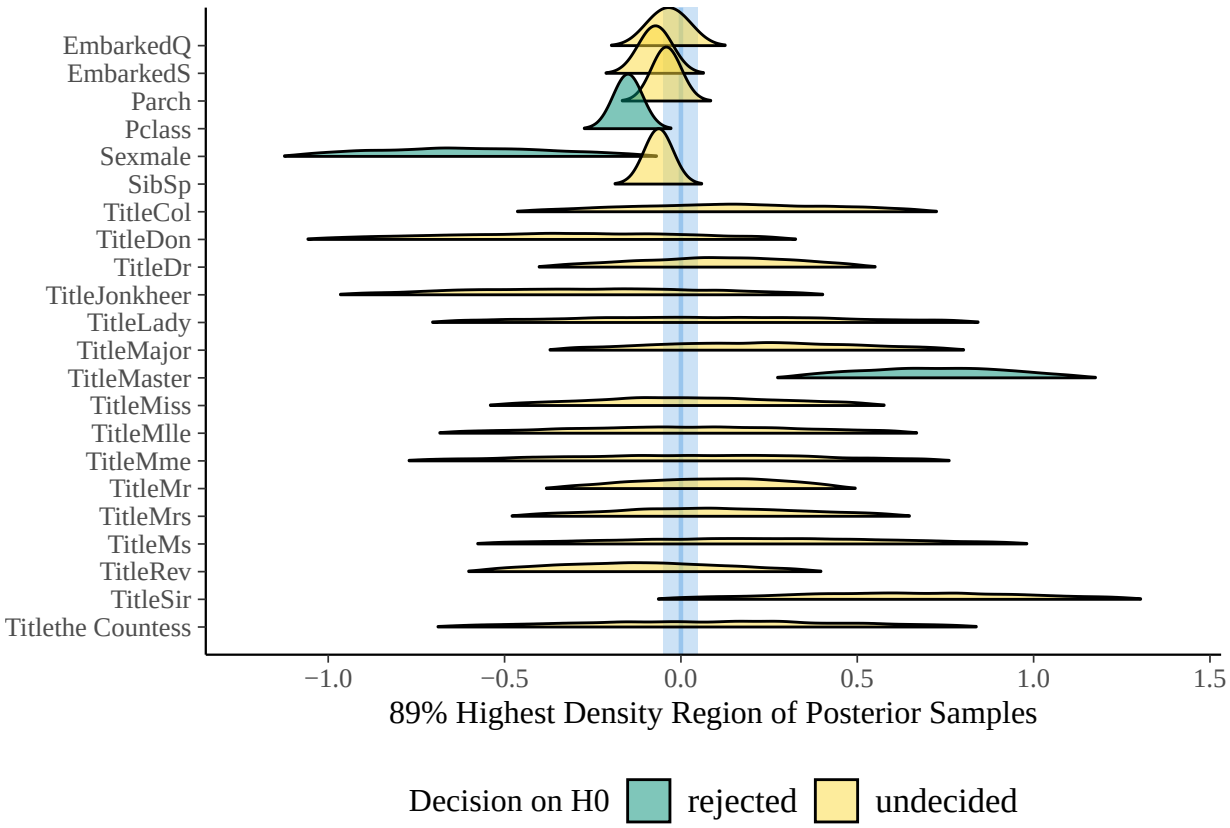
```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.81), TitleMr and TitleDr (r = 0.84)
```

```
## # Test for Practical Equivalence
##
##   ROPE: [-0.05 0.05]
##
##          Parameter        H0 inside ROPE       89% HDI
##        (Intercept)  rejected      0.00 % [ 0.61  1.67]
##           TitleCol undecided     10.90 % [-0.44  0.69]
##           TitleDon undecided      8.12 % [-1.04  0.30]
##            TitleDr undecided     13.84 % [-0.37  0.51]
##      TitleJonkheer undecided      8.59 % [-0.95  0.38]
##          TitleLady undecided      8.87 % [-0.68  0.82]
##         TitleMajor undecided     10.78 % [-0.33  0.77]
##        TitleMaster  rejected      0.00 % [ 0.31  1.14]
##          TitleMiss undecided     13.09 % [-0.51  0.55]
##          TitleMlle undecided     10.25 % [-0.66  0.64]
##           TitleMme undecided      9.10 % [-0.75  0.73]
##            TitleMr undecided     16.06 % [-0.35  0.45]
##           TitleMrs undecided     12.92 % [-0.45  0.62]
##            TitleMs undecided      7.55 % [-0.55  0.96]
##           TitleRev undecided     13.09 % [-0.57  0.36]
##           TitleSir undecided      3.00 % [-0.04  1.28]
##  Titlethe Countess undecided      8.99 % [-0.68  0.82]
##            Sexmale  rejected      0.00 % [-1.10 -0.10]
##             Pclass  rejected      0.00 % [-0.17 -0.12]
##          EmbarkedQ undecided     60.57 % [-0.12  0.05]
##          EmbarkedS undecided     20.78 % [-0.12 -0.02]
##              SibSp undecided      8.57 % [-0.08 -0.04]
##              Parch undecided     67.76 % [-0.07 -0.01]
```

```
plot(equivalence_test(model))
```

```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.81), TitleMr and TitleDr (r = 0.84)
```

```
## Picking joint bandwidth of 0.0399
```



Pclass, Sex-male, and Title-master are rejected in the null hypothesis. Temporarily removing Embarked to get a closer look at other determinants.

Removing Embarked

```
model <- stan_glm(Survived ~ Title + Sex + Pclass + SibSp + Parch,
                  refresh = 0, data = titanic_train)
equivalence_test(model)
```

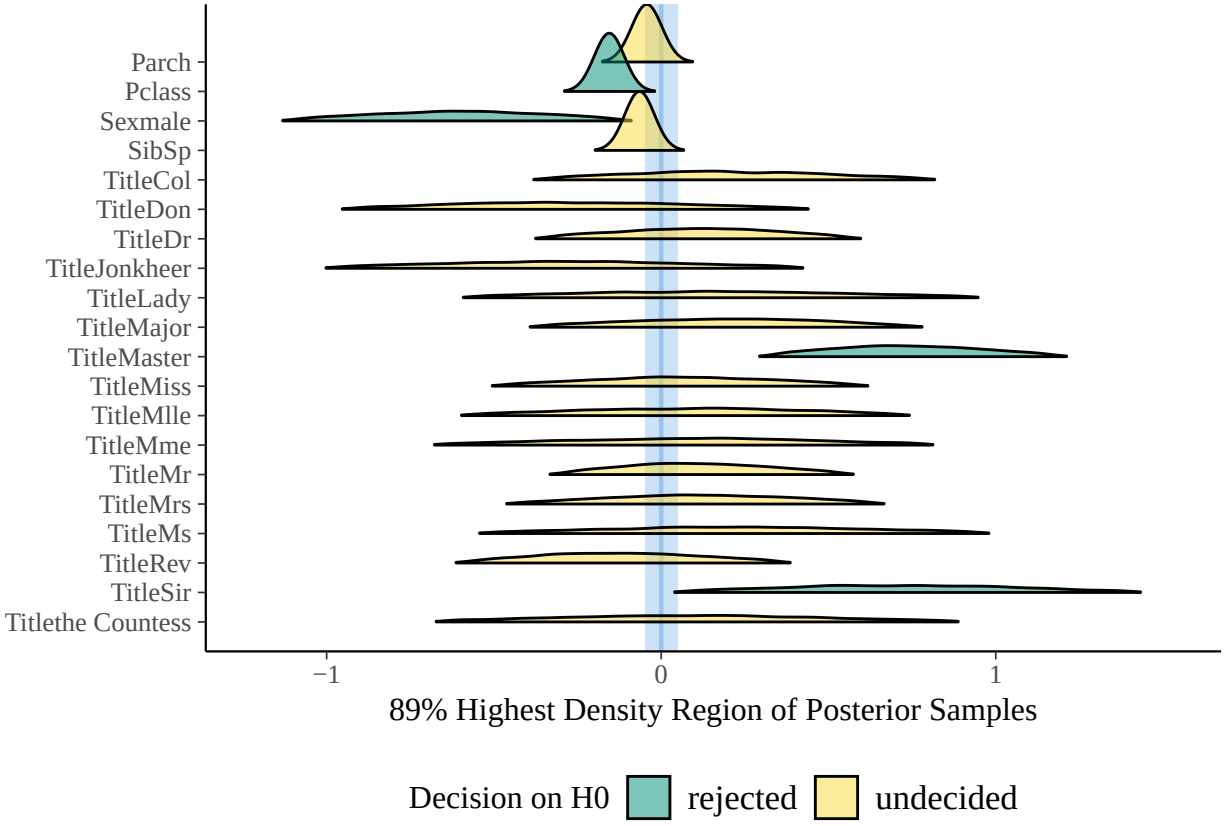
```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.83), TitleMr and TitleDr (r = 0.86)
## # Test for Practical Equivalence
##
##   ROPE: [-0.05 0.05]
##
##          Parameter        H0 inside ROPE       89% HDI
##        (Intercept)  rejected      0.00 % [ 0.51  1.58]
##           TitleCol undecided     10.92 % [-0.35  0.78]
##           TitleDon undecided      8.79 % [-0.93  0.42]
##            TitleDr undecided     13.98 % [-0.33  0.55]
##      TitleJonkheer undecided      7.83 % [-0.98  0.39]
##          TitleLady undecided      7.95 % [-0.56  0.93]
##         TitleMajor undecided     10.47 % [-0.36  0.75]
##        TitleMaster  rejected      0.00 % [ 0.34  1.17]
##          TitleMiss undecided     13.56 % [-0.47  0.59]
##          TitleMlle undecided      9.46 % [-0.57  0.71]
##           TitleMme undecided      9.41 % [-0.66  0.78]
```

```
##            TitleMr undecided     16.46 % [-0.29  0.53]
##           TitleMrs undecided     12.33 % [-0.43  0.63]
##            TitleMs undecided      8.79 % [-0.52  0.95]
##           TitleRev undecided     13.37 % [-0.58  0.35]
##           TitleSir  rejected      0.00 % [ 0.07  1.41]
##  Titlethe Countess undecided      9.30 % [-0.65  0.87]
##            Sexmale  rejected      0.00 % [-1.09 -0.12]
##             Pclass  rejected      0.00 % [-0.18 -0.13]
##              SibSp undecided      5.70 % [-0.09 -0.05]
##              Parch undecided     63.18 % [-0.07 -0.01]
```

```
plot(equivalence_test(model))
```

```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.83), TitleMr and TitleDr (r = 0.86)
```

```
## Picking joint bandwidth of 0.0432
```



Removing Sex

Beginning to remove determinate items beginning with Sex.

```
model <- stan_glm(Survived ~ Title + Pclass,
                  refresh = 0, data = titanic_train)
equivalence_test(model)
```

```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.83), TitleMiss and TitleDr (r = 0.8
```

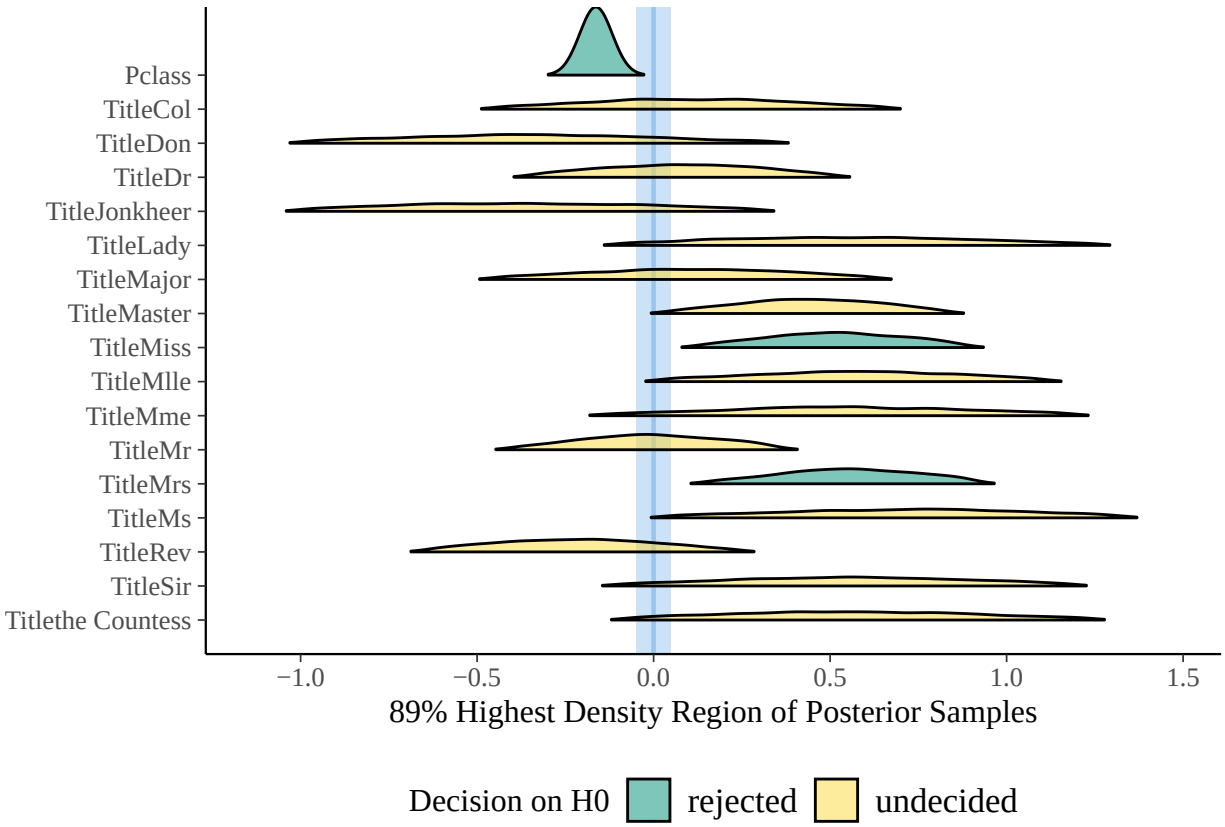
```
## # Test for Practical Equivalence
##
```

```
##   ROPE: [-0.05 0.05]
##
##          Parameter        H0 inside ROPE       89% HDI
##        (Intercept)  rejected      0.00 % [ 0.18  0.94]
##           TitleCol undecided     12.16 % [-0.45  0.66]
##           TitleDon undecided      7.47 % [-1.00  0.35]
##            TitleDr undecided     14.52 % [-0.35  0.52]
##      TitleJonkheer undecided      7.95 % [-1.01  0.31]
##          TitleLady undecided      4.27 % [-0.11  1.26]
##         TitleMajor undecided     12.64 % [-0.45  0.64]
##        TitleMaster undecided      0.65 % [ 0.04  0.83]
##          TitleMiss  rejected      0.00 % [ 0.12  0.89]
##          TitleMlle undecided      1.74 % [ 0.01  1.12]
##           TitleMme undecided      4.47 % [-0.16  1.20]
##            TitleMr undecided     18.79 % [-0.40  0.36]
##           TitleMrs  rejected      0.00 % [ 0.15  0.92]
##            TitleMs undecided      0.84 % [ 0.03  1.34]
##           TitleRev undecided     11.12 % [-0.65  0.24]
##           TitleSir undecided      4.38 % [-0.12  1.20]
##  Titlethe Countess undecided      4.24 % [-0.08  1.25]
##             Pclass  rejected      0.00 % [-0.19 -0.14]
```

```
plot(equivalence_test(model))
```

```
## Possible multicollinearity between TitleMaster and TitleDr (r = 0.83), TitleMiss and TitleDr (r = 0.8
```

```
## Picking joint bandwidth of 0.0442
```



As Title-Miss and Title-Mrs are determinants, I will apply another grouping according to knowledge of the

culture of the turn of the 20th century.

"Women and children first"" - setting up an adultMale factor

I'm going to fit the titles into categories - Men and all others including boys. The idea behind this is the culture of the time. Men were expected to sacrifice for women and children. If this actually played out on the Titanic, it should show as a simple binary factor. The logic is if male and title does not equal Master, then adultMale, otherwise not adultMale.

```
titanic_train <- titanic_train %>%
  mutate(adultMale = ifelse(Sex == "male" & Title != "Master",1,0))
```

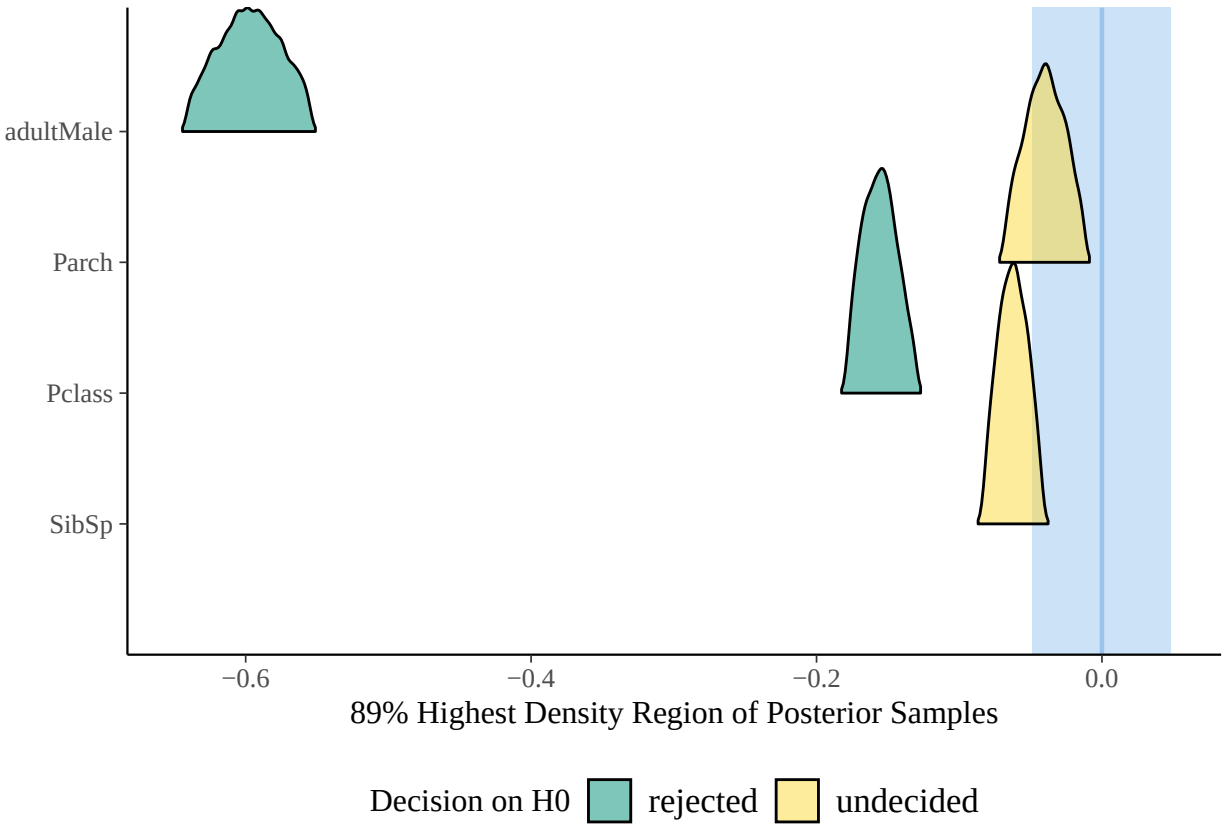
Loading the adultMale factor into the model

```
glm_model <- stan_glm(Survived ~ adultMale + Pclass + SibSp + Parch, refresh = 0,
                      data = titanic_train)
equivalence_test(glm_model)
```

```
## # Test for Practical Equivalence
##
##   ROPE: [-0.05 0.05]
##
##    Parameter        H0 inside ROPE       89% HDI
##  (Intercept)  rejected      0.00 % [ 1.09  1.21]
##    adultMale  rejected      0.00 % [-0.64 -0.55]
##       Pclass  rejected      0.00 % [-0.18 -0.13]
##        SibSp undecided      9.13 % [-0.08 -0.04]
##        Parch undecided     71.69 % [-0.07 -0.01]
```

```
plot(equivalence_test(glm_model))
```

```
## Picking joint bandwidth of 0.00245
```



Adding back Embarked

Embarked seemed to be possible determinants but I removed them to look closer at other determinants earlier. I am adding them back now.

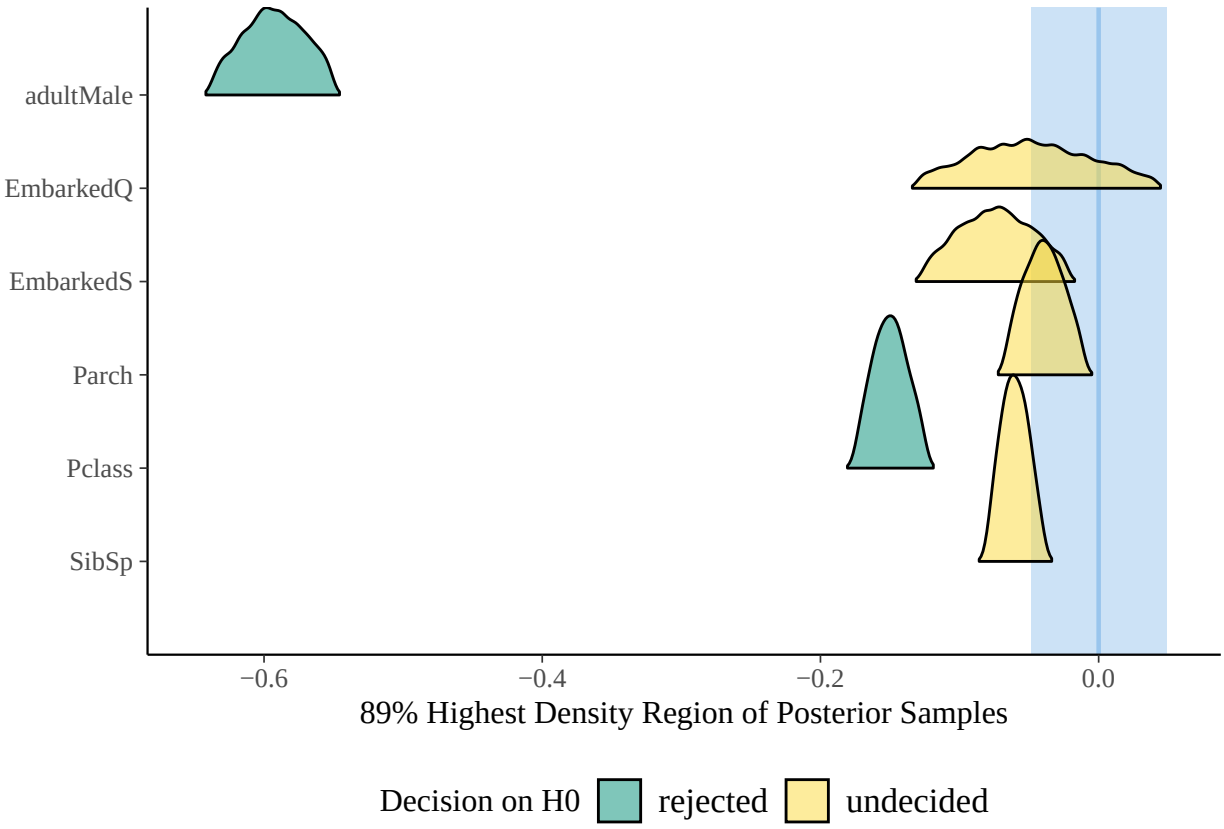
```
glm_model <- stan_glm(Survived ~ adultMale + Pclass + SibSp + Parch + Embarked, refresh = 0,
                      data = titanic_train)
equivalence_test(glm_model)
```

```
## # Test for Practical Equivalence
##
##   ROPE: [-0.05 0.05]
##
##    Parameter        H0 inside ROPE       89% HDI
##  (Intercept)  rejected      0.00 % [ 1.12  1.26]
##    adultMale  rejected      0.00 % [-0.64 -0.55]
##       Pclass  rejected      0.00 % [-0.17 -0.12]
##        SibSp undecided     12.75 % [-0.08 -0.04]
##        Parch undecided     72.51 % [-0.07 -0.01]
##    EmbarkedQ undecided     49.28 % [-0.13  0.04]
##    EmbarkedS undecided     19.04 % [-0.13 -0.02]
```

```
plot(equivalence_test(glm_model))
```

```
## Picking joint bandwidth of 0.0036
```

```
## Warning: Removed 2634 rows containing non-finite values
## (stat_density_ridges).
```



Although being an adult male is by far the dominant determinant, Pclass is also rejected from the null hypothesis, and Embarked, Parch and SibSp so some small promise. This will be the final model.

Fitting and testing

Setting a baseline at Sex as the only determinant

```
fit_glm <- glm(Survived ~ Sex, data = titanic_train)
p_hat_glm <- predict(fit_glm, titanic_train)
y_hat_glm <- factor(ifelse(p_hat_glm > 0.5, 1, 0))

x_result <-
  tibble(model = "Sex Only",
         method = "glm",
         accuracy = confusionMatrix(data = y_hat_glm,
                                    reference = factor(titanic_train$Survived))$overall["Accuracy"])
```

Fitting the model

```
fit_glm <- glm(Survived ~ adultMale + Pclass + SibSp + Parch + Embarked, data = titanic_train)
p_hat_glm <- predict(fit_glm, titanic_train)
y_hat_glm <- factor(ifelse(p_hat_glm > 0.5, 1, 0))
x_result <-
```

```
  add_row(x_result,
          model = "final",
          method = "glm",
          accuracy = confusionMatrix(data = y_hat_glm,
                                     reference = factor(titanic_train$Survived))$overall["Accuracy"])
```

Using caret to find the best method

A handful of methods usable in the carat package were picked as appropriate for classification.

```
methods_lst <- c("rf", "parRF", "ranger", "bayesglm", "blackboost", "svmPoly")
```

Creating a function to run through the different methods

```
itterate_methods <- function(x) {
  set.seed(50)
  train_x <- train(factor(Survived) ~ adultMale + Pclass + SibSp + Parch + Embarked,
                   method = methods_lst[x],  data = titanic_train)
    p_hat_x <- predict(train_x, titanic_train)
}
```

Loop to run through all the methods

```
for(i in 1:length(methods_lst)) {
    p_hat_x <- itterate_methods(i)
    x_result <-
      add_row(x_result,
              model = "final",
              method = methods_lst[i],
              accuracy =
                confusionMatrix(data = p_hat_x,
                                reference =
                                  factor(titanic_train$Survived))$overall["Accuracy"])
  }
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
## Registered S3 method overwritten by 'coin':
##   method   from
##   print.ci bayestestR
```

Looking at the results

```
x_result %>% knitr::kable()
```

model	method	accuracy
Sex Only	glm	0.7867565
final	glm	0.8327722
final	rf	0.8428732

model	method	accuracy
final	parRF	0.8428732
final	ranger	0.8305275
final	bayesglm	0.8271605
final	blackboost	0.8249158
final	svmPoly	0.8305275

Conclusion

The best I could get was .842, and I got this with the "rf" and the "ParRF" method. I'm simply choosing one and it will be the "rf" method.

Going for a solution

Data duties for the test set

1. Adding the title

```
titanic_test <- titanic_test %>%
  mutate(Title = gsub('(.*, )|(\\..*)','', titanic_test$Name))
```

Ensuring the code worked

```
table(titanic_test$Title)
```

```
##
##    Col   Dona     Dr Master   Miss     Mr    Mrs     Ms    Rev
##      2      1      1     21     78    240     72      1      2
```

2. Adding the adultMale determiner

```
titanic_test <- titanic_test %>%
  mutate(adultMale = ifelse(Sex == "male" & Title != "Master",1,0))
```

Look for missing data in other determinants

```
table(titanic_test$Embarked)
```

```
##
##   C   Q   S
## 102  46 270
```

```
table(titanic_test$Pclass)
```

```
##
##   1   2   3
## 107  93 218
```

```
table(titanic_test$SibSp)
```

```
##
##   0   1   2   3   4   5   8
## 283 110  14   4   4   1   2
```

```
table(titanic_test$Parch)
```

```
##
##   0   1   2   3   4   5   6   9
## 324  52  33   3   2   1   1   2
```

Wow, nothing is missing. Good to go!

Fitting the model

Using the decided on "rf" model

```
train_rf <- train(factor(Survived) ~ adultMale + Pclass + SibSp + Parch + Embarked,
                  method = "rf", data = titanic_train)
p_hat_rf <- predict(train_rf, titanic_test)
solution <- data.frame(Survived = p_hat_rf, PassengerId = titanic_test$PassengerId)

write.csv(solution, file = 'rf_model_sol.csv', row.names = F)
```

Kaggle returned a score of 0.79425 and a rank of 2,214 out of 11,322.
The base Sex Only base case returned a score of 0.76555.

```
improvement_over_base <- 1-(.76555/.79425)
ranking_in_top_percent <- 2214/11322
improvement_over_base
```

```
## [1] 0.03613472
```

```
ranking_in_top_percent
```

```
## [1] 0.1955485
```

Conclusion

At the beginning of the capstone projects, I realized I just didn't have the knowledge or experience in R to be able to successfully complete the projects. I took the next few months to take online classes in R, R markdown and statistics to have the tools to tackle these projects. I was able to use some elementary knowledge of Bayesian statistics and elements of R including functions, looping, data frames, faceted charting and other techniques I was unable to do when I began this course of study. I also was able to represent part of the human portion of the tragic disaster in the sinking of the Titanic through data. And, I took a stab at predicting survival by using data and a random forest method to achieve a better than average result.

The primary difficulty in this project was with such a large determinant of adult male death rates, it was difficult to find significant other factors for prediction. It was a small data set. So many interesting items

showed up, but the populations were way too small to make anything of them. The data sets were roughly two thirds training and one third test. The test set was divergent enough from the training set to drop the 84% predictive rate on the training set to under 80% on the test set. Without the specific results, it is hard to know where the predictions failed.

Using a Bayesian model to determine the components of the predictive model was interesting, but not dramatically successful. Alternate models would have been interesting. However, learning about Bayesian statistics during the project was a personal benefit that I found enlightening and enjoyable, and an interest of further study.

The data definitely showed some things that speak to the bravery of the people of that time, and to their dedication to their values. It is clear that the people of the Titanic followed the value of preserving women and children at the cost of men. Seeing the charts showing the number and percent of men that perished was heartbreaking. It caused me to imagine the young men lost to their families, the surviving women and children without them. The bravery of those men including the clergy, who suffered 100% loss in the training set. I assume they stayed behind to comfort the doomed.

I don't believe it inappropriate to dwell on the human tragedy in a machine learning project. One should never forget there are most likely people behind the numbers they analyze. In my own business, I analyze programs for low income customers. It is easy to manage these programs by production numbers and penetration rates, but there are people behind these numbers. In addition to a numbing effect to true human tragedy, analysis can cause unintended bias. For example, the highest density of poverty in my county coincides with ethnic minorities. If one ignored the impact of this correlation, one could end up doing something offensive and wrong like using Latino names to determine who would be offered low-income programs. In the case of the Titanic, the data involved people living over 100 years ago. There is no use in trying to predict anything in the current year based on these data. However, the emotional impact of what the data tell us should not be dismissed, as we must always remember in our analysis: people are behind these numbers, and they deserve to be treated as humans with all the dignity and sensitivity we all deserve.