An Illustrative Look

Marshall McQuillen

Guiding question

Which passengers will survive the sinking of the Titanic?

Secondary Questions

1. What characteristics separate those who survived from those who died?
2. What charactersistics make someone more likely to survive?
3. How do different characteristics of passengers vary with one another?

```
# install.packages("tidyverse")
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(dplyr))
```

Data Overview

Looking at the training data froma bird-eye view, there are 891 observations representing passengers and 12 variables. First and foremost, we can see in the Cabin and Embarked columns that the first entry is an empty string, indicating that our data is probably not perfectly clean (no surprises there). Checking to see where any Null's might be, it becomes clear that there are in fact no nulls. This illustrates the difference between a

Test 1	Test 2	Test 3
1	2	3

```
str(training)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
# view null count
nulls <- lapply(training[,1:12], is.null)
as.data.frame(lapply(nulls, sum))
```

```
##   PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin
## 1           0        0      0    0   0   0     0     0      0    0     0
##   Embarked
```

```
## 1        0
```

```r
# view NA count
na_count <- lapply(training[,1:12], is.na)
as.data.frame(lapply(na_count, sum))
```

```
##   PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin
## 1           0        0      0    0   0 177     0     0      0    0     0
##   Embarked
## 1        0
```

Data Cleaning

All the NA's in the data set are in the Age column. The column is close to 20% NA's so building a model on that variable won't be the best idea.

```r
#Set classes

training$PassengerId <- as.factor(training$PassengerId)
training$Survived <- as.factor(training$Survived)
training$Pclass <- as.factor(training$Pclass)
training$SibSp <- as.factor(training$SibSp)
training$Parch <- as.factor(training$Parch)




#Percent of Age attribute that is NA
paste(round(sum(is.na(training$Age))/length(training$Age)*100, digits = 2), "%", sep = "")
```

```
## [1] "19.87%"
```

2 Bayesian Survival

2.2 Does Money Sink or Swim?

Illustrating Bayes Theorem with Survival Rates and Socio-Economic Status

By creating a table with the Pclass and Survived variables, I can get a good sense of the number of passengers that lived and died, based on their Socio-Economic Status (SES). Simple summation and division returns the probabilites of a passenger living given their respective SES.

```r
#Probability of living by socio-economic status
pclass_table <- with(training, table(Survived, Pclass))
upper_class <- pclass_table[2,1]/sum(pclass_table[,1])*100
middle_class <- pclass_table[2,2]/sum(pclass_table[,2])*100
lower_class <- pclass_table[2,3]/sum(pclass_table[,3])*100

pclass_table
```

```
##         Pclass
## Survived   1   2   3
##        0  80  97 372
##        1 136  87 119
```

```r
#Probability of living given Upper Class
paste(round(upper_class, digits = 2), "%", sep = "")
```

```
## [1] "62.96%"
```

```
#Probability of living given Middle Class
paste(round(middle_class, digits = 2), "%", sep = "")
```

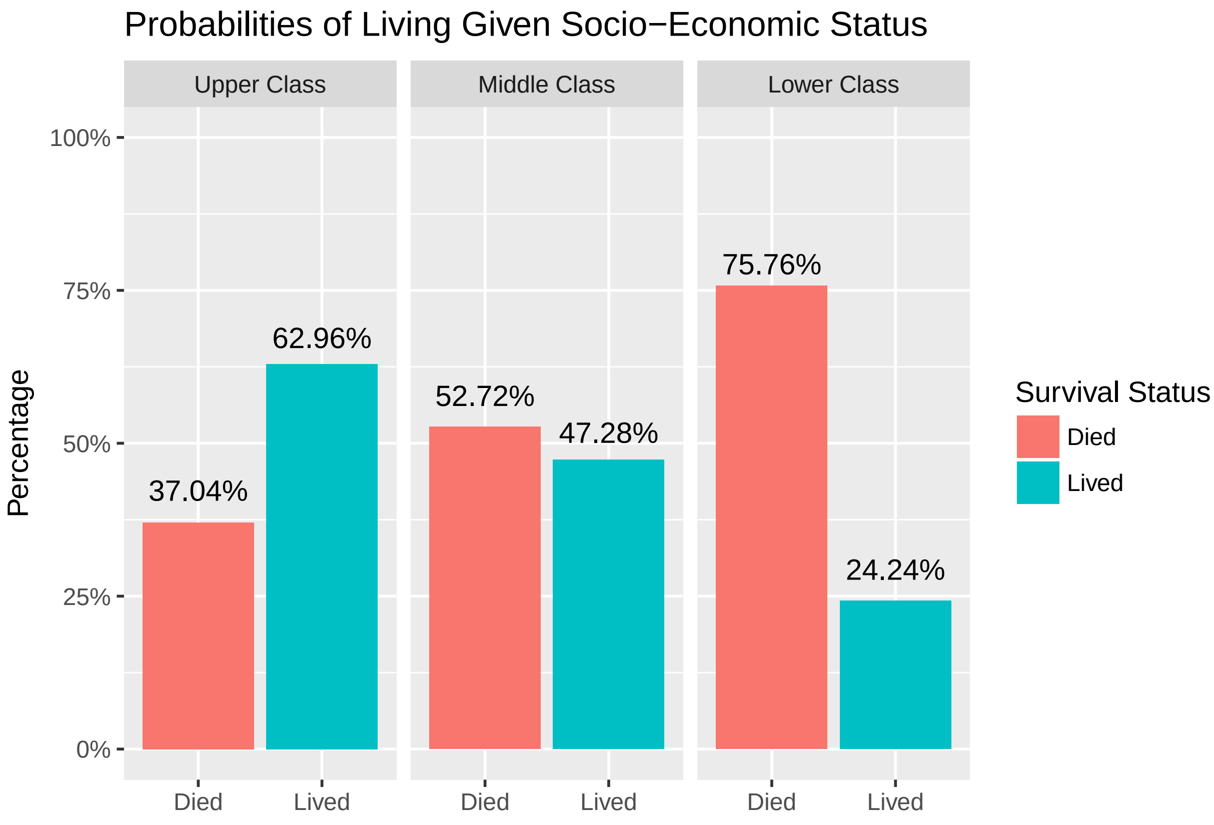
```
## [1] "47.28%"
```

```
#Probability of living given Lower Class
paste(round(lower_class, digits = 2), "%", sep = "")
```

```
## [1] "24.24%"
```

The same information can be displayed visually as follows.

```
g <- ggplot(training, aes(y=Survived,  x=factor(Survived,
                                                labels=c("Died","Lived"))))
g <- g + geom_bar(aes(y=..prop.., group=Pclass,
                      fill=factor(..x.., labels=c("Died","Lived"))))
g <- g + facet_grid(~factor(Pclass,
                            labels=c("Upper Class", "Middle Class",
                                     "Lower Class")))
g <- g + scale_y_continuous(labels = scales::percent)
g <- g + scale_fill_discrete(name="Survival Status")
g <- g + labs(x="", y = "Percentage",
              title = "Probabilities of Living Given Socio-Economic Status")
g <- g + geom_text(
      aes(label = paste(round((..count../c(216,216,184,184,491,491)), 4)*100, "%", sep = ""), y = ..prop
      vjust = c(17.25,10,12.75,14.5,6.5,21))
g
```



For a simple proof of Bayes Theorem, defined as...

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

...I can set **P(A) = the probability of a passenger belonging to a defined SES (X)** and **P(B) = the probability of a passenger living**. I can now rewrite the previously defined Theorm using my definitions as:

$$P(\text{ ”}X\ class\ citizen\text{” } | \text{ ”}Lived\text{” }) \;=\; \frac{P(\text{ ”}Lived\text{” } | \text{ ”}X\ class\ citizen\text{” })\ P(\text{ ”}X\ class\ citizen\text{” })}{(\text{ ”}Lived\text{” })}$$

```
total <- nrow(training)
total_died <- nrow(subset(training, Survived == 0))
total_lived <- nrow(subset(training, Survived == 1))

#Probability of Living = P(B)
prob_lived <- total_lived/(total_died + total_lived)

#Probability of being Upper, Middle or Lower class = P(A)
upper_prob <- as.numeric(table(training$Pclass)[1])/total
middle_prob <- as.numeric(table(training$Pclass)[2])/total
lower_prob <- as.numeric(table(training$Pclass)[3])/total
```

Now that I have found both **P("X class citizen")** (objects upper_prob, middle_prob and lower_prob) and **P("Lived")** (object prob_lived), and I have **P("Lived"|"X class ctiizen")** (objects upper_class, middle_class and lower_class), I can solve for **P("X class citizen"|"Lived")**...

P("Upper class citizen" | "Lived") = upper_class X upper_prob

prop_lived

P("Middle class citizen" | "Lived") = middle_class X middle_prob

prop_lived

P("Lower class citizen" | "Lived") = lower_class X lower_prob

prop_lived

```
#Probability of being an Upper class citizen given a passenger lived
prob_upper_given_lived <- (upper_class*upper_prob)/prob_lived

#Probability of being a Middle class citizen given a passenger lived
prob_middle_given_lived <- (middle_class*middle_prob)/prob_lived

#Probability of being a Lower class ctizen given a passenger lived
prob_lower_given_lived <- (lower_class*lower_prob)/prob_lived
```

The "shorthand" way of finding these probabilites can be accomplished by dividing the the number of X class passengers that lived by the total number of passengers that lived using the pclass_table.

```
#"Shorthand" for calculating probability of being a X class citizen using pclass_table
upper_class_by_total <- pclass_table[2,1]/total_lived*100
middle_class_by_total <- pclass_table[2,2]/total_lived*100
lower_class_by_total <- pclass_table[2,3]/total_lived*100
```

```
#Showing that the probabilities using Bayes Theorem and pclass_table are equal
all.equal(upper_class_by_total, prob_upper_given_lived)
```

```
## [1] TRUE
```

```
all.equal(middle_class_by_total, prob_middle_given_lived)
```

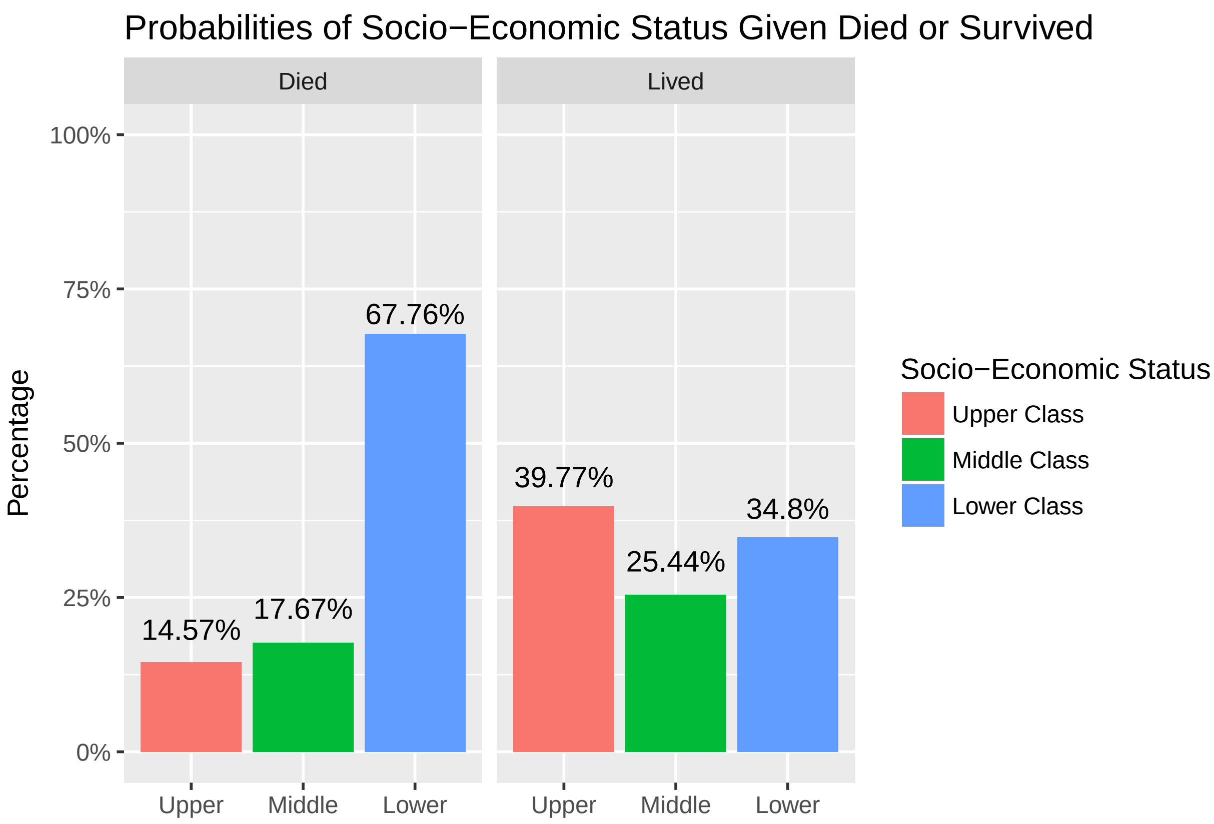
```
## [1] TRUE
```

```
all.equal(lower_class_by_total, prob_lower_given_lived)
```

```
## [1] TRUE
```

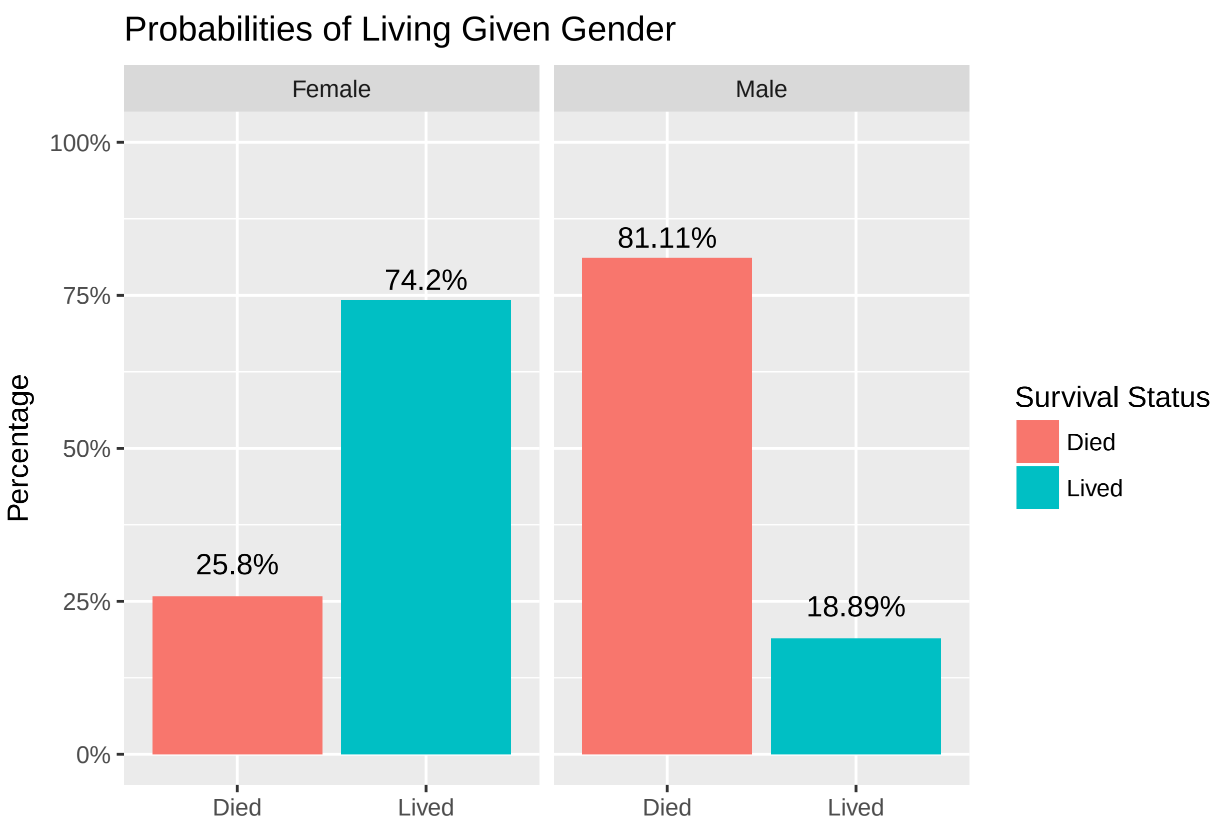
A graphic illustrating said results.

```
f <- ggplot(training, aes(y = Pclass,
                          x = factor(Pclass,labels=c("Upper","Middle","Lower"))))
f <- f + geom_bar(aes(y=..prop.., group=Survived,
                      fill = factor(..x.., labels=c("Upper Class","Middle Class","Lower Class"))))
f <- f + facet_grid(~factor(Survived, labels = c("Died","Lived")))
f <- f + scale_y_continuous(labels = scales::percent)
f <- f + scale_fill_discrete(name = "Socio-Economic Status")
f <- f + labs(x = "", y = "Percentage", title = "Probabilities of Socio-Economic Status Given Died or Su
f <- f + geom_text(
      aes(label = paste(round((..count../c(549,549,549,342,342,342)), 4)*100, "%", sep = ""), y = ..pro
      vjust = c(24,23,9,16.75,20.75,18.25))
f
```



Using the same code as above, with a few minor adjustments I can make similar graphs with other qualitative variables, such as the sex of the passenger.

```r
#Sex vs. Survived
a <- ggplot(training, aes(y=Survived,  x=factor(Survived,
                                          labels=c("Died","Lived"))))
a <- a + geom_bar(aes(y=..prop.., group=Sex,
                  fill=factor(..x.., labels=c("Died","Lived"))))
a <- a + facet_grid(~factor(Sex,
                        labels=c("Female", "Male")))
a <- a + scale_y_continuous(labels = scales::percent)
a <- a + scale_fill_discrete(name="Survival Status")
a <- a + labs(x="", y = "Percentage",
              title = "Probabilities of Living Given Gender")
a <- a + geom_text(
      aes(label = paste(round((..count../c(314,314,577,577)), 4)*100, "%", sep = ""), y = ..prop..),  s
      vjust = c(20.5,7,5,22.5))
a
```



3 Cabin Classification

It seems logical that looking at *where* each passenger was when the Titanic started sinking could provide some insight as to why some lived and others did not. The “Sinking” section on the Titanic Wikipedia Page states that the iceberg was struck at 11:40 pm. Considering the time of night, combined with the likely cold air temperature, I think it is safe to say that most passengers were inside, if not in their rooms sleeping.

Finding out where each passenger was will be a two fold process:

1. Subsetting on the Deck they were on, noted by the letter in the Cabin column.
2. Subsetting where on that deck they were, noted by the number in the Cabin column.

An important note is that the vast majority of the passengers did not have an entry in the Cabin column. (There aren't any NA's, the entries are not even filled with spaces, they are simply "nothing"). In order to subset these observations, I used the ouput from a "nothing" observation in the logical statement.

After subsetting, summing the number of rows in each subset, *which should equal 891, the total number of observations*, returns 894. A little searching led to finding the duplicates, show below.

```
#Split data on Cabin Letter
a_class <- training[grep("A", training$Cabin),]
b_class <- training[grep("B", training$Cabin),]
c_class <- training[grep("C", training$Cabin),]
d_class <- training[grep("D", training$Cabin),]
e_class <- training[grep("E", training$Cabin),]
f_class <- training[grep("F", training$Cabin),]
g_class <- training[grep("G", training$Cabin),]

#the "nothing" class
blank_class <- subset(training, Cabin == training[1,11])

sum(nrow(a_class) + nrow(b_class) + nrow(c_class) + nrow(d_class) +
          nrow(e_class) + nrow(f_class) + nrow(g_class) + nrow(blank_class))
```

```
## [1] 894
```

```
#Duplicate Cabin Values
duplicates <- training[c(76,129,700,716),]
duplicates
```

```
##     PassengerId Survived Pclass                                   Name
## 76           76        0      3                   Moen, Mr. Sigurd Hansen
## 129         129        1      3                        Peter, Miss. Anna
## 700         700        0      3   Humblen, Mr. Adolf Mathias Nicolai Olsen
## 716         716        0      3 Soholt, Mr. Peter Andreas Lauritz Andersen
##        Sex Age SibSp Parch Ticket    Fare Cabin Embarked
## 76    male  25     0     0 348123  7.6500 F G73        S
## 129 female  NA     1     1   2668 22.3583 F E69        C
## 700   male  42     0     0 348121  7.6500 F G63        S
## 716   male  19     0     0 348124  7.6500 F G73        S
```

```
#look at passenger id 534
training[grep("Peter", training$Name),]
```

```
##     PassengerId Survived Pclass
## 129         129        1      3
## 356         356        0      3
## 398         398        0      2
## 407         407        0      3
## 477         477        0      2
## 534         534        1      3
## 681         681        0      3
## 716         716        0      3
## 727         727        1      2
## 844         844        0      3
## 858         858        1      1
## 861         861        0      3
##                                     Name    Sex  Age SibSp Parch
## 129                        Peter, Miss. Anna female   NA     1     1
```

```
## 356                 Vanden Steen, Mr. Leo Peter   male 28.0     0     0
## 398                     McKane, Mr. Peter David   male 46.0     0     0
## 407            Widegren, Mr. Carl/Charles Peter   male 51.0     0     0
## 477                     Renouf, Mr. Peter Henry   male 34.0     1     0
## 534      Peter, Mrs. Catherine (Catherine Rizk) female   NA     0     2
## 681                         Peters, Miss. Katie female   NA     0     0
## 716  Soholt, Mr. Peter Andreas Lauritz Andersen   male 19.0     0     0
## 727 Renouf, Mrs. Peter Henry (Lillian Jefferys) female 30.0     3     0
## 844                  Lemberopolous, Mr. Peter L   male 34.5     0     0
## 858                       Daly, Mr. Peter Denis   male 51.0     0     0
## 861                     Hansen, Mr. Claus Peter   male 41.0     2     0
##     Ticket    Fare Cabin Embarked
## 129   2668 22.3583 F E69        C
## 356 345783  9.5000              S
## 398  28403 26.0000              S
## 407 347064  7.7500              S
## 477  31027 21.0000              S
## 534   2668 22.3583              C
## 681 330935  8.1375              Q
## 716 348124  7.6500 F G73        S
## 727  31027 21.0000              S
## 844   2683  6.4375              C
## 858 113055 26.5500   E17        S
## 861 350026 14.1083              S
```

To decide which subset to assign these observations too, looking at the Embarked and Ticket columns for those observations in the g_class subset, I can see that everyone in this cabin class embarked from Southampton and had similar ticket

```r
table(a_class$Survived)[2]/sum(table(a_class$Survived))
```

```
##         1
## 0.4666667
```

```r
table(b_class$Survived)[2]/sum(table(b_class$Survived))
```

```
##         1
## 0.7446809
```

```r
table(c_class$Survived)[2]/sum(table(c_class$Survived))
```

```
##         1
## 0.5932203
```

```r
table(d_class$Survived)[2]/sum(table(d_class$Survived))
```

```
##         1
## 0.7575758
```

```r
table(e_class$Survived)[2]/sum(table(e_class$Survived))
```

```
##         1
## 0.7575758
```

```r
table(f_class$Survived)[2]/sum(table(f_class$Survived))
```

```
##         1
## 0.6153846
```

```r
table(g_class$Survived)[2]/sum(table(g_class$Survived))
```

```
##         1
## 0.2857143
```

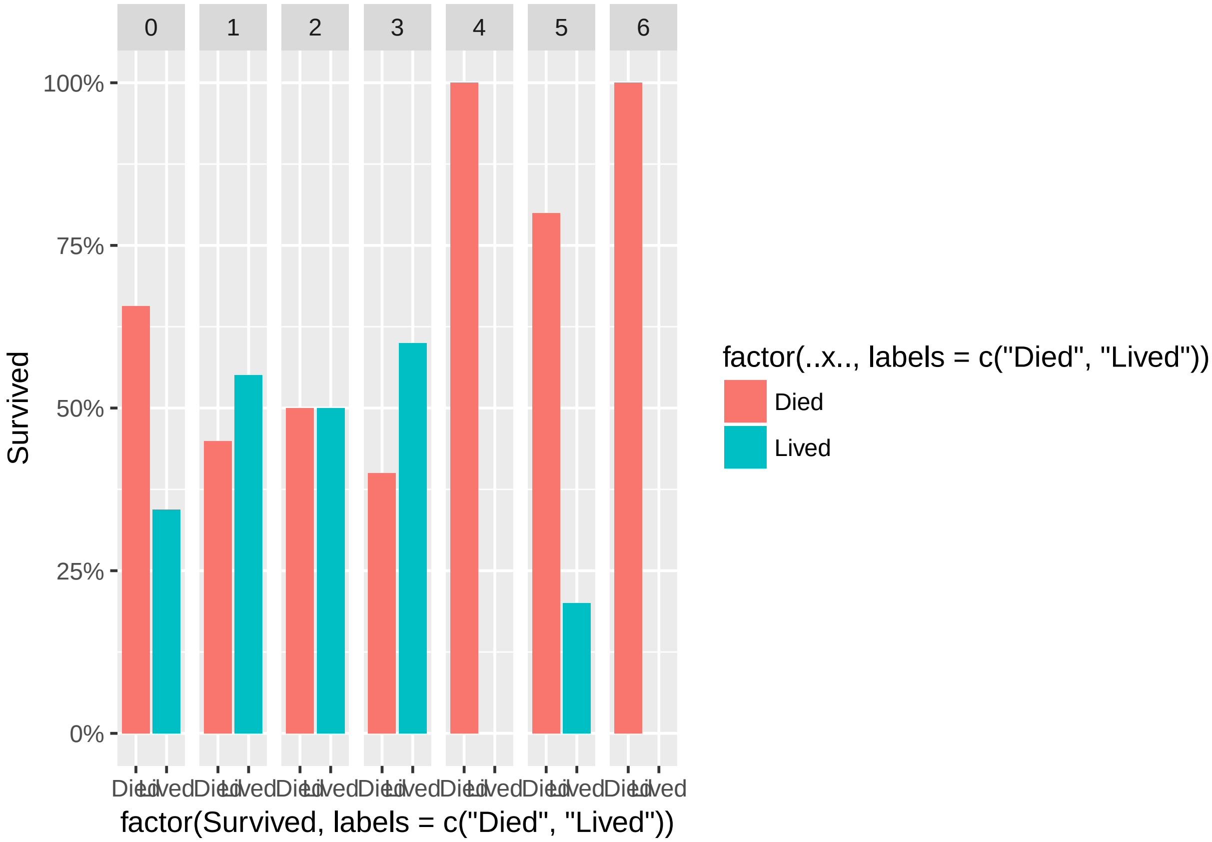
```r
table(blank_class$Survived)[2]/sum(table(blank_class$Survived))
```

```
##         1
## 0.2998544
```

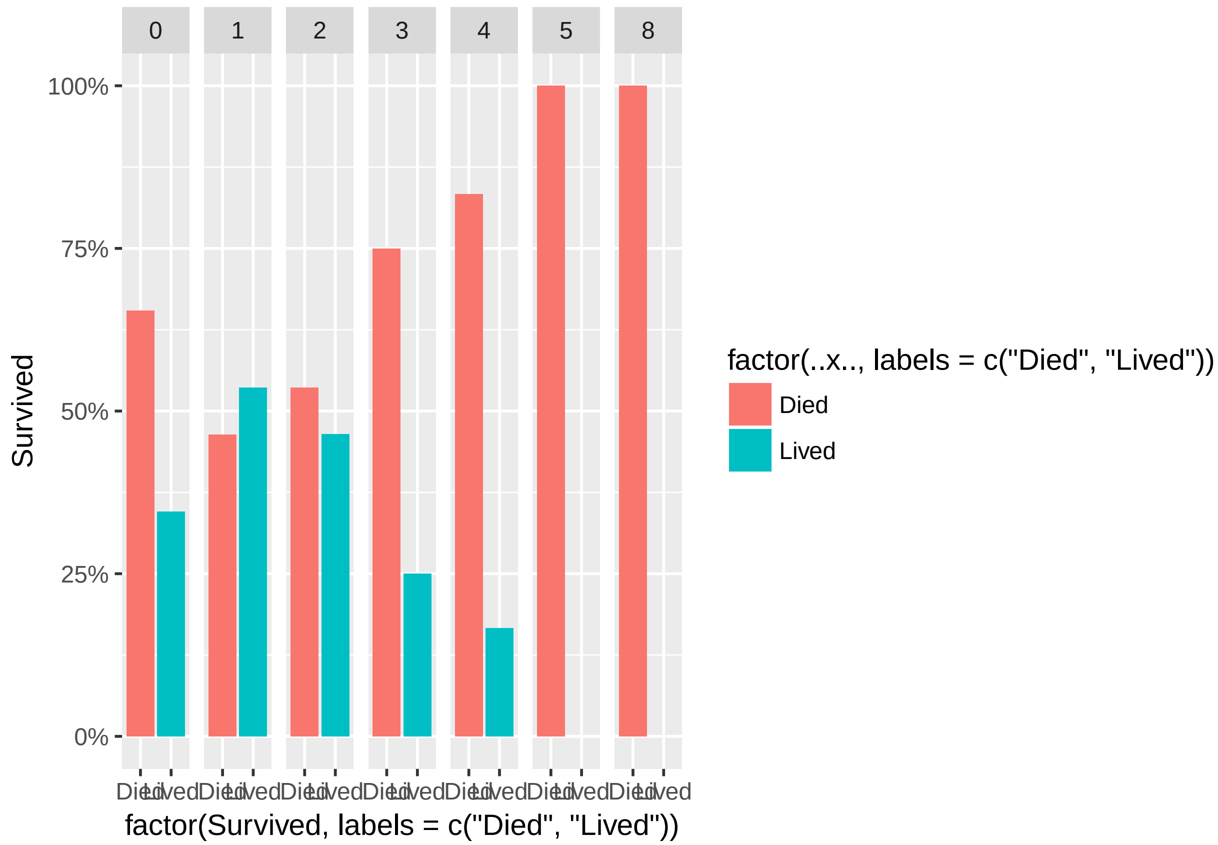
```r
#Split data on Cabin Room Number
tCabin_number = grep("[0-9]{2,}", training$Cabin)
test <- mutate(training, test_column = c(strsplit(as.character(training$Cabin), " ")))

#Parch and Sibsp Analysis
table(training$Parch, training$Survived)
```

```
##
##       0   1
##   0 445 233
##   1  53  65
##   2  40  40
##   3   2   3
##   4   4   0
##   5   4   1
##   6   1   0
```

```r
h <- ggplot(data = training,
            aes(y = Survived,
                x = factor(Survived, labels = c("Died","Lived"))))
h <- h + geom_bar(aes(y = ..prop.., group = Parch,
                      fill = factor(..x.., labels = c("Died","Lived"))))
h <- h + facet_grid(~Parch)
h <- h + scale_y_continuous(labels = scales::percent)
h <- h + scale_fill_discrete()
h
```




```
d <- ggplot(data = training,
            aes(y = Survived,
                x = factor(Survived, labels = c("Died","Lived"))))
d <- d + geom_bar(aes(y = ..prop.., group = SibSp,
                  fill = factor(..x.., labels = c("Died","Lived"))))
d <- d + facet_grid(~SibSp)
d <- d + scale_y_continuous(labels = scales::percent)
d <- d + scale_fill_discrete()
d
```


0
1
2
3
4
5
8
100%
75%
50%
25%
0%
Survived
Died
Lived
factor(Survived, labels = c("Died", "Lived"))
factor(..x.., labels = c("Died", "Lived"))
Died
Lived