Titanic

Lets take a look at the data:

```
train <- read.csv('c:/sas/r/train.csv')
test <- read.csv('c:/sas/r/test.csv')
names(train)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

```
str(train)
```

```
## '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 1
6 559 520 629 417 581 ...
##  $ 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 27
6 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 ...
```

```
summary(train)
```

```
##   PassengerId       Survived          Pclass     
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000  
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000  
##  Median :446.0   Median :0.0000   Median :3.000  
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309  
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000  
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000  
##                                                  
##                                     Name         Sex           Age       
##  Abbing, Mr. Anthony                  :  1   female:314   Min.   : 0.42  
##  Abbott, Mr. Rossmore Edward          :  1   male  :577   1st Qu.:20.12  
##  Abbott, Mrs. Stanton (Rosa Hunt)     :  1                Median :28.00  
##  Abelson, Mr. Samuel                  :  1                Mean   :29.70  
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00  
##  Adahl, Mr. Mauritz Nils Martin       :  1                Max.   :80.00  
##  (Other)                              :885                NA's   :177    
##      SibSp           Parch             Ticket         Fare       
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00  
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91  
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45  
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20  
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00  
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33  
##                                   (Other) :852                   
##          Cabin     Embarked
##             :687    :  2   
##  B96 B98    :  4   C:168   
##  C23 C25 C27:  4   Q: 77   
##  G6         :  4   S:644   
##  C22 C26    :  3           
##  D          :  3           
##  (Other)    :186           
```

We have considerable missingness on the age variable. To look at the bigger picture:

```
library(VIM)
```

```
## Warning: package 'VIM' was built under R version 3.4.3
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
## Warning: package 'data.table' was built under R version 3.4.3
```

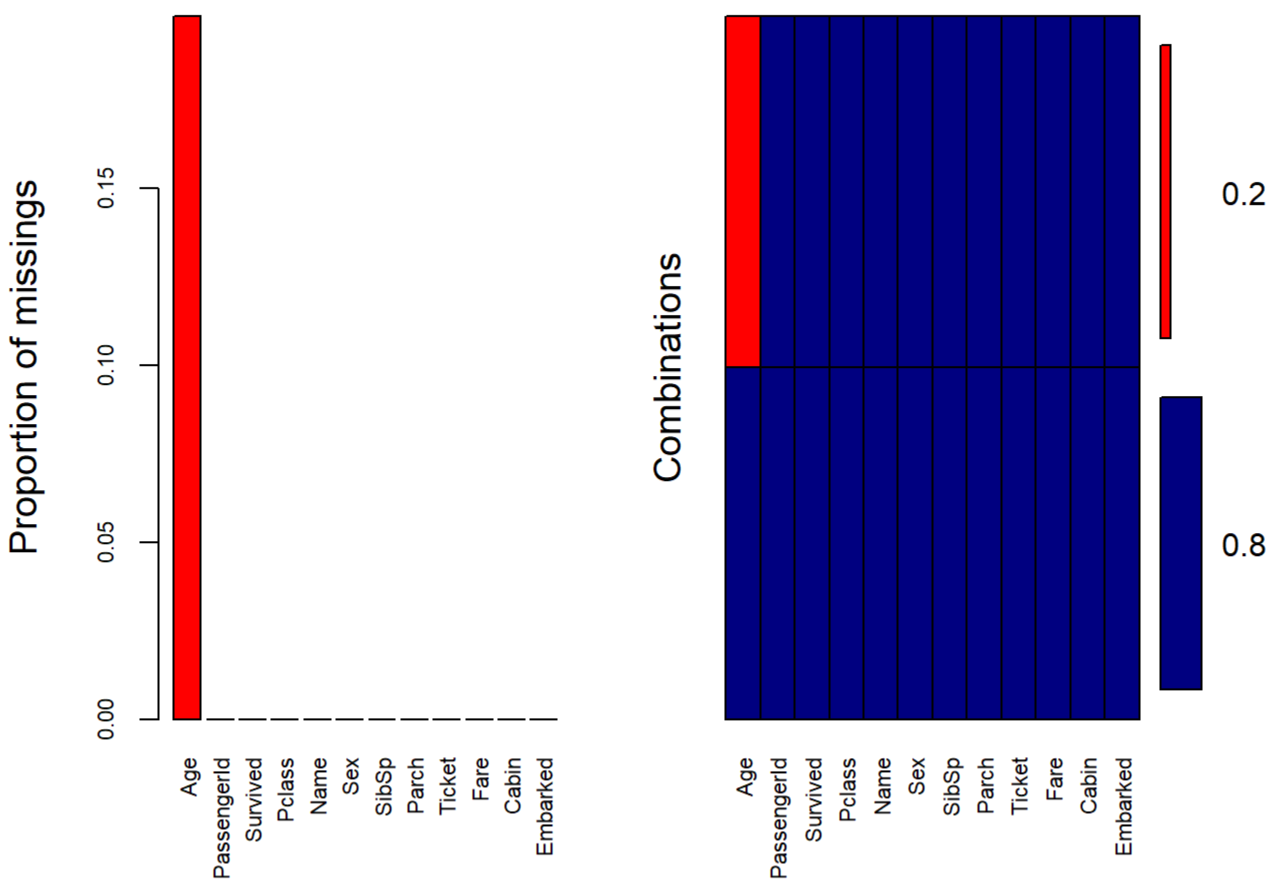
```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM
/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
mice_plot <- aggr(train, col=c('navyblue','red'),
                    numbers=TRUE, sortVars=TRUE,
                    labels=names(train), cex.axis=.7)
```



```
##
##  Variables sorted by number of missings:
##     Variable     Count
##          Age 0.1986532
##  PassengerId 0.0000000
##     Survived 0.0000000
##       Pclass 0.0000000
##         Name 0.0000000
##          Sex 0.0000000
##        SibSp 0.0000000
##        Parch 0.0000000
##       Ticket 0.0000000
##         Fare 0.0000000
##        Cabin 0.0000000
##     Embarked 0.0000000
```

It looks like my interpretation of the summary w as correct.

Because I don't know how to do sophisticated imputation w ith r, I am going to use Hmisc and replace NA w ith the median. How ever, I w ould prefer to use a GLM or some sort of bootstrapped imputation. If and w hen I learn to do that, the below w ith be helpful in determining predictors.

Let's examine w hat is most strongly correlated w ith Survival and Age.

```
library(vcd)
```

```
## Warning: package 'vcd' was built under R version 3.4.3
```

```
train.num <- subset(train, select = c(Survived, Pclass, Age, SibSp, Parch, Fare))
cor(train.num, use = "na.or.complete")
```

```
##             Survived      Pclass         Age       SibSp       Parch
## Survived  1.00000000 -0.35965268 -0.07722109 -0.01735836  0.09331701
## Pclass   -0.35965268  1.00000000 -0.36922602  0.06724737  0.02568307
## Age      -0.07722109 -0.36922602  1.00000000 -0.30824676 -0.18911926
## SibSp    -0.01735836  0.06724737 -0.30824676  1.00000000  0.38381986
## Parch     0.09331701  0.02568307 -0.18911926  0.38381986  1.00000000
## Fare      0.26818862 -0.55418247  0.09606669  0.13832879  0.20511888
##                 Fare
## Survived  0.26818862
## Pclass   -0.55418247
## Age       0.09606669
## SibSp     0.13832879
## Parch     0.20511888
## Fare      1.00000000
```

```
sxf <- table(train$Survived, train$Sex)
sxpc <- table(train$Survived, train$Pclass)
summary(sxf)
```

```
## Number of cases in table: 891
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 263.05, df = 1, p-value = 3.712e-59
```

```
assocstats(sxf)
```

```
##                      X^2 df P(> X^2)
## Likelihood Ratio 268.85  1        0
## Pearson          263.05  1        0
##
## Phi-Coefficient   : 0.543
## Contingency Coeff.: 0.477
## Cramer's V        : 0.543
```

```
summary(sxpc)
```

```
## Number of cases in table: 891
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 102.89, df = 2, p-value = 4.549e-23
```

```
assocstats(sxpc)
```

```
##                      X^2 df P(> X^2)
## Likelihood Ratio 103.55  2        0
## Pearson          102.89  2        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.322
## Cramer's V        : 0.34
```

Pclass, Parch, and SibSp are the most strongly correlated w ith age. We can also see that Pclass, Parch, Fare and Sex (phi = 0.54) are the most correlated w ith survival.

Let’s do a simple imputation.

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.4.3
```

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

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
train$Age <- with(train, impute(Age, median))
as.numeric(train$Age)
```

```
##   [1] 22.00 38.00 26.00 35.00 35.00 28.00 54.00  2.00 27.00 14.00  4.00
##  [12] 58.00 20.00 39.00 14.00 55.00  2.00 28.00 31.00 28.00 35.00 34.00
##  [23] 15.00 28.00  8.00 38.00 28.00 19.00 28.00 28.00 40.00 28.00 28.00
##  [34] 66.00 28.00 42.00 28.00 21.00 18.00 14.00 40.00 27.00 28.00  3.00
##  [45] 19.00 28.00 28.00 28.00 28.00 18.00  7.00 21.00 49.00 29.00 65.00
##  [56] 28.00 21.00 28.50  5.00 11.00 22.00 38.00 45.00  4.00 28.00 28.00
##  [67] 29.00 19.00 17.00 26.00 32.00 16.00 21.00 26.00 32.00 25.00 28.00
##  [78] 28.00  0.83 30.00 22.00 29.00 28.00 28.00 17.00 33.00 16.00 28.00
##  [89] 23.00 24.00 29.00 20.00 46.00 26.00 59.00 28.00 71.00 23.00 34.00
## [100] 34.00 28.00 28.00 21.00 33.00 37.00 28.00 21.00 28.00 38.00 28.00
## [111] 47.00 14.50 22.00 20.00 17.00 21.00 70.50 29.00 24.00  2.00 21.00
## [122] 28.00 32.50 32.50 54.00 12.00 28.00 24.00 28.00 45.00 33.00 20.00
## [133] 47.00 29.00 25.00 23.00 19.00 37.00 16.00 24.00 28.00 22.00 24.00
## [144] 19.00 18.00 19.00 27.00  9.00 36.50 42.00 51.00 22.00 55.50 40.50
## [155] 28.00 51.00 16.00 30.00 28.00 28.00 44.00 40.00 26.00 17.00  1.00
## [166]  9.00 28.00 45.00 28.00 28.00 61.00  4.00  1.00 21.00 56.00 18.00
## [177] 28.00 50.00 30.00 36.00 28.00 28.00  9.00  1.00  4.00 28.00 28.00
## [188] 45.00 40.00 36.00 32.00 19.00 19.00  3.00 44.00 58.00 28.00 42.00
## [199] 28.00 24.00 28.00 28.00 34.00 45.50 18.00  2.00 32.00 26.00 16.00
## [210] 40.00 24.00 35.00 22.00 30.00 28.00 31.00 27.00 42.00 32.00 30.00
## [221] 16.00 27.00 51.00 28.00 38.00 22.00 19.00 20.50 18.00 28.00 35.00
## [232] 29.00 59.00  5.00 24.00 28.00 44.00  8.00 19.00 33.00 28.00 28.00
## [243] 29.00 22.00 30.00 44.00 25.00 24.00 37.00 54.00 28.00 29.00 62.00
## [254] 30.00 41.00 29.00 28.00 30.00 35.00 50.00 28.00  3.00 52.00 40.00
## [265] 28.00 36.00 16.00 25.00 58.00 35.00 28.00 25.00 41.00 37.00 28.00
## [276] 63.00 45.00 28.00  7.00 35.00 65.00 28.00 16.00 19.00 28.00 33.00
## [287] 30.00 22.00 42.00 22.00 26.00 19.00 36.00 24.00 24.00 28.00 23.50
## [298]  2.00 28.00 50.00 28.00 28.00 19.00 28.00 28.00  0.92 28.00 17.00
## [309] 30.00 30.00 24.00 18.00 26.00 28.00 43.00 26.00 24.00 54.00 31.00
## [320] 40.00 22.00 27.00 30.00 22.00 28.00 36.00 61.00 36.00 31.00 16.00
## [331] 28.00 45.50 38.00 16.00 28.00 28.00 29.00 41.00 45.00 45.00  2.00
## [342] 24.00 28.00 25.00 36.00 24.00 40.00 28.00  3.00 42.00 23.00 28.00
## [353] 15.00 25.00 28.00 28.00 22.00 38.00 28.00 28.00 40.00 29.00 45.00
## [364] 35.00 28.00 30.00 60.00 28.00 28.00 24.00 25.00 18.00 19.00 22.00
## [375]  3.00 28.00 22.00 27.00 20.00 19.00 42.00  1.00 32.00 35.00 28.00
## [386] 18.00  1.00 36.00 28.00 17.00 36.00 21.00 28.00 23.00 24.00 22.00
## [397] 31.00 46.00 23.00 28.00 39.00 26.00 21.00 28.00 20.00 34.00 51.00
## [408]  3.00 21.00 28.00 28.00 28.00 33.00 28.00 44.00 28.00 34.00 18.00
## [419] 30.00 10.00 28.00 21.00 29.00 28.00 18.00 28.00 28.00 19.00 28.00
## [430] 32.00 28.00 28.00 42.00 17.00 50.00 14.00 21.00 24.00 64.00 31.00
## [441] 45.00 20.00 25.00 28.00 28.00  4.00 13.00 34.00  5.00 52.00 36.00
## [452] 28.00 30.00 49.00 28.00 29.00 65.00 28.00 50.00 28.00 48.00 34.00
## [463] 47.00 48.00 28.00 38.00 28.00 56.00 28.00  0.75 28.00 38.00 33.00
## [474] 23.00 22.00 28.00 34.00 29.00 22.00  2.00  9.00 28.00 50.00 63.00
## [485] 25.00 28.00 35.00 58.00 30.00  9.00 28.00 21.00 55.00 71.00 21.00
## [496] 28.00 54.00 28.00 25.00 24.00 17.00 21.00 28.00 37.00 16.00 18.00
## [507] 33.00 28.00 28.00 26.00 29.00 28.00 36.00 54.00 24.00 47.00 34.00
## [518] 28.00 36.00 32.00 30.00 22.00 28.00 44.00 28.00 40.50 50.00 28.00
## [529] 39.00 23.00  2.00 28.00 17.00 28.00 30.00  7.00 45.00 30.00 28.00
## [540] 22.00 36.00  9.00 11.00 32.00 50.00 64.00 19.00 28.00 33.00  8.00
```

```
## [551] 17.00 27.00 28.00 22.00 22.00 62.00 48.00 28.00 39.00 36.00 28.00
## [562] 40.00 28.00 28.00 28.00 24.00 19.00 29.00 28.00 32.00 62.00 53.00
## [573] 36.00 28.00 16.00 19.00 34.00 39.00 28.00 32.00 25.00 39.00 54.00
## [584] 36.00 28.00 18.00 47.00 60.00 22.00 28.00 35.00 52.00 47.00 28.00
## [595] 37.00 36.00 28.00 49.00 28.00 49.00 24.00 28.00 28.00 44.00 35.00
## [606] 36.00 30.00 27.00 22.00 40.00 39.00 28.00 28.00 28.00 35.00 24.00
## [617] 34.00 26.00  4.00 26.00 27.00 42.00 20.00 21.00 21.00 61.00 57.00
## [628] 21.00 26.00 28.00 80.00 51.00 32.00 28.00  9.00 28.00 32.00 31.00
## [639] 41.00 28.00 20.00 24.00  2.00 28.00  0.75 48.00 19.00 56.00 28.00
## [650] 23.00 28.00 18.00 21.00 28.00 18.00 24.00 28.00 32.00 23.00 58.00
## [661] 50.00 40.00 47.00 36.00 20.00 32.00 25.00 28.00 43.00 28.00 40.00
## [672] 31.00 70.00 31.00 28.00 18.00 24.50 18.00 43.00 36.00 28.00 27.00
## [683] 20.00 14.00 60.00 25.00 14.00 19.00 18.00 15.00 31.00  4.00 28.00
## [694] 25.00 60.00 52.00 44.00 28.00 49.00 42.00 18.00 35.00 18.00 25.00
## [705] 26.00 39.00 45.00 42.00 22.00 28.00 24.00 28.00 48.00 29.00 52.00
## [716] 19.00 38.00 27.00 28.00 33.00  6.00 17.00 34.00 50.00 27.00 20.00
## [727] 30.00 28.00 25.00 25.00 29.00 11.00 28.00 23.00 23.00 28.50 48.00
## [738] 35.00 28.00 28.00 28.00 36.00 21.00 24.00 31.00 70.00 16.00 30.00
## [749] 19.00 31.00  4.00  6.00 33.00 23.00 48.00  0.67 28.00 18.00 34.00
## [760] 33.00 28.00 41.00 20.00 36.00 16.00 51.00 28.00 30.50 28.00 32.00
## [771] 24.00 48.00 57.00 28.00 54.00 18.00 28.00  5.00 28.00 43.00 13.00
## [782] 17.00 29.00 28.00 25.00 25.00 18.00  8.00  1.00 46.00 28.00 16.00
## [793] 28.00 28.00 25.00 39.00 49.00 31.00 30.00 30.00 34.00 31.00 11.00
## [804]  0.42 27.00 31.00 39.00 18.00 39.00 33.00 26.00 39.00 35.00  6.00
## [815] 30.50 28.00 23.00 31.00 43.00 10.00 52.00 27.00 38.00 27.00  2.00
## [826] 28.00 28.00  1.00 28.00 62.00 15.00  0.83 28.00 23.00 18.00 39.00
## [837] 21.00 28.00 32.00 28.00 20.00 16.00 30.00 34.50 17.00 42.00 28.00
## [848] 35.00 28.00 28.00  4.00 74.00  9.00 16.00 44.00 18.00 45.00 51.00
## [859] 24.00 28.00 41.00 21.00 48.00 28.00 24.00 42.00 27.00 31.00 28.00
## [870]  4.00 26.00 47.00 33.00 47.00 28.00 15.00 20.00 19.00 28.00 56.00
## [881] 25.00 33.00 22.00 28.00 25.00 39.00 27.00 19.00 28.00 26.00 32.00
```

```
summary(train$Age)
```

```
##
##  177 values imputed to 28
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   28.00   29.36   35.00   80.00
```

Below is my best attempt to visualize things

```
library('dplyr')
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library('ggplot2')
library('ggthemes')
```

```
## Warning: package 'ggthemes' was built under R version 3.4.3
```

```
summary(train$Sex)
```

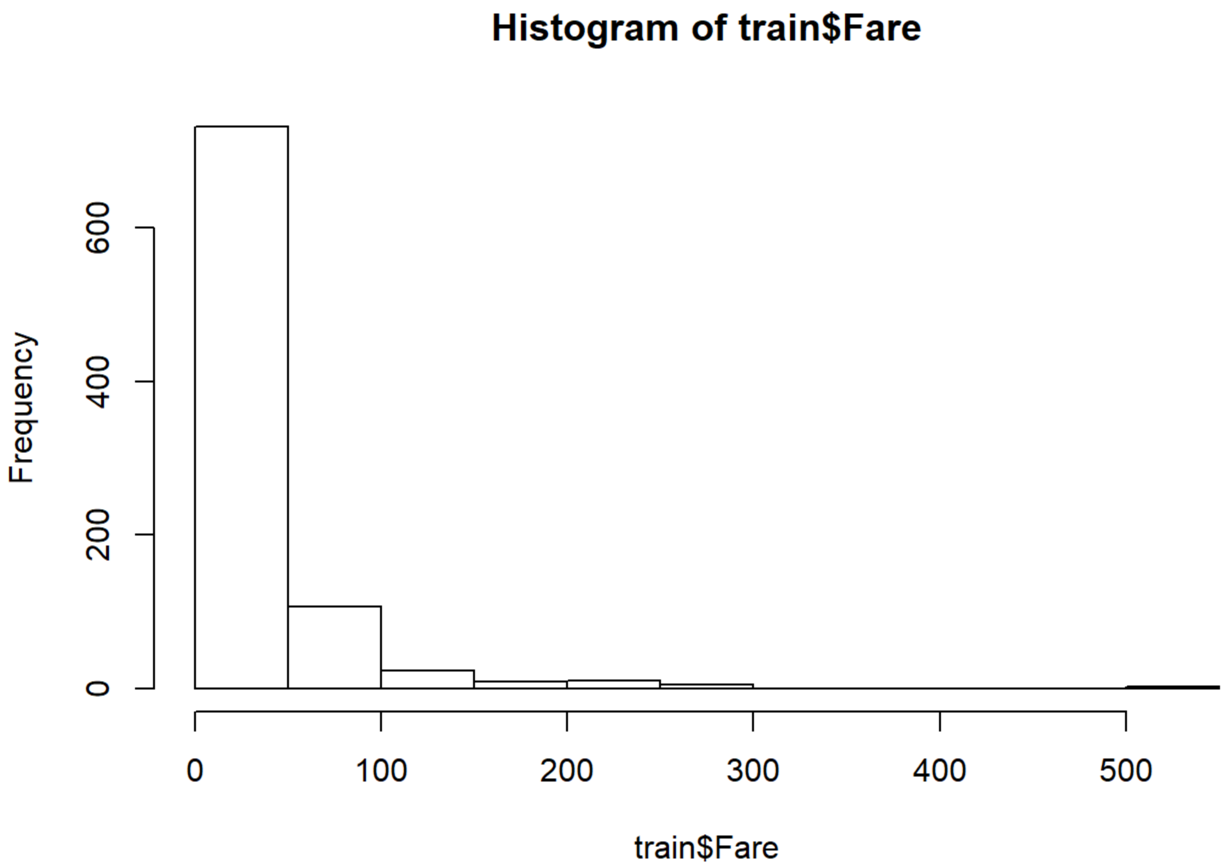
```
## female   male
##    314    577
```

```
summary(train$Fare)
```

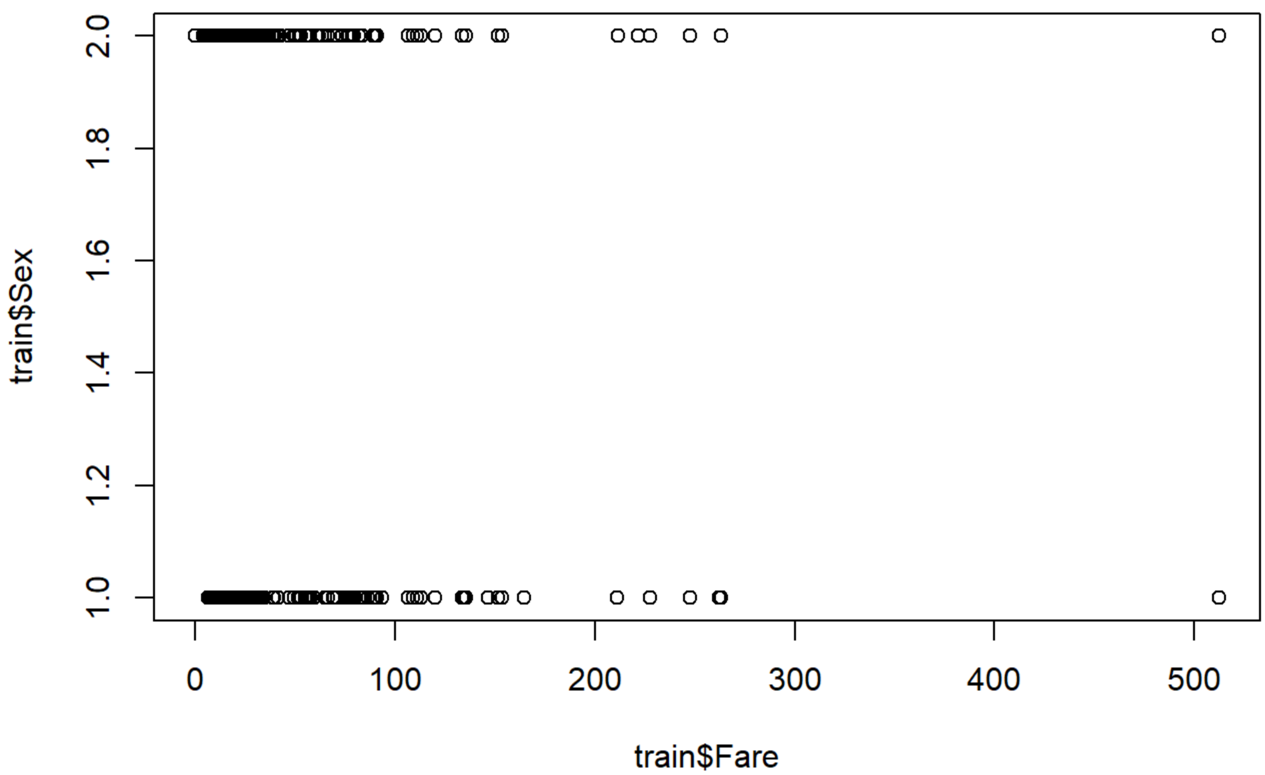
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.91   14.45   32.20   31.00  512.33
```

```
hist(train$Age)
```

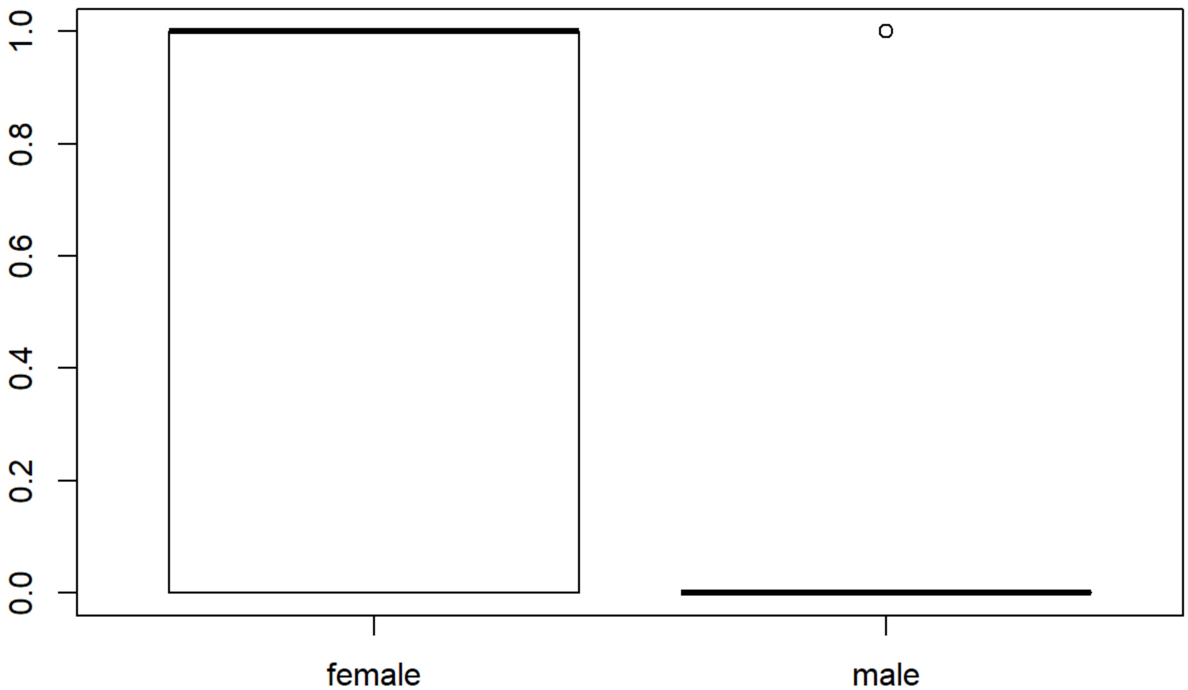

```
hist(train$Fare)
```



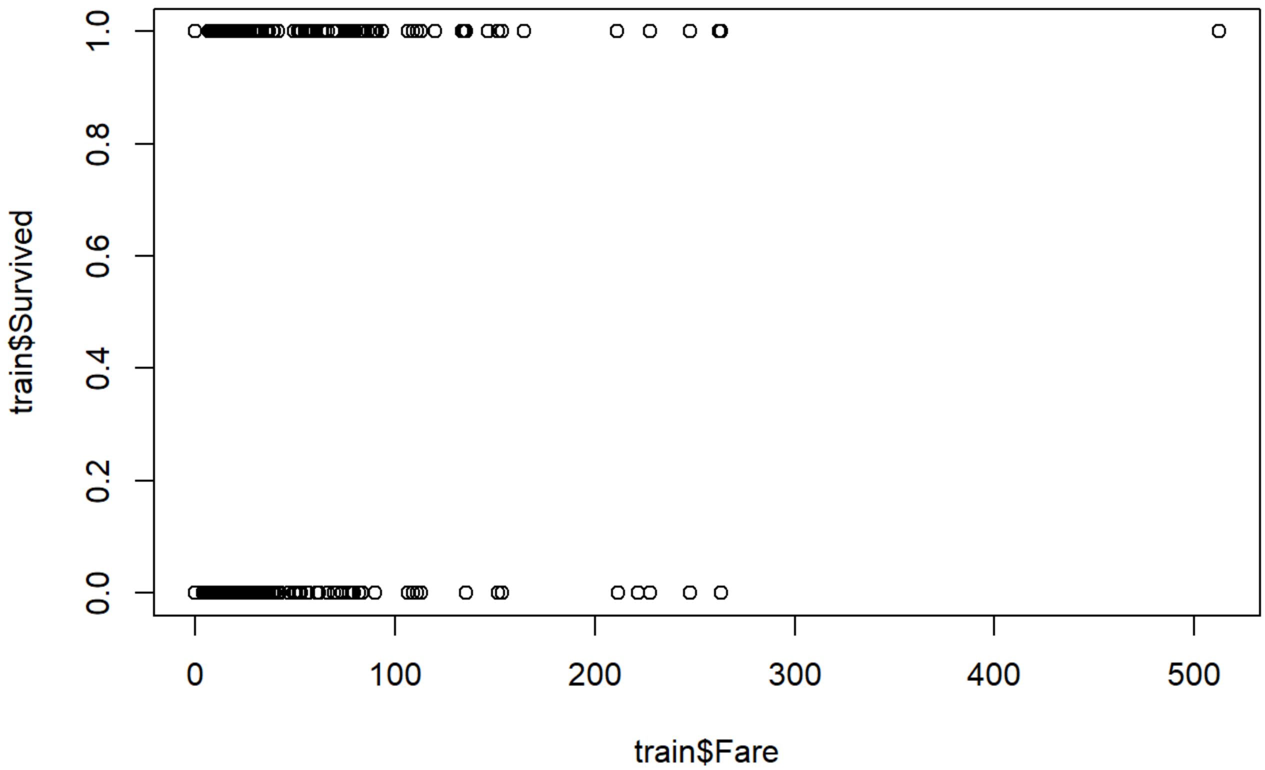

```
plot(train$Fare, train$Sex)
```



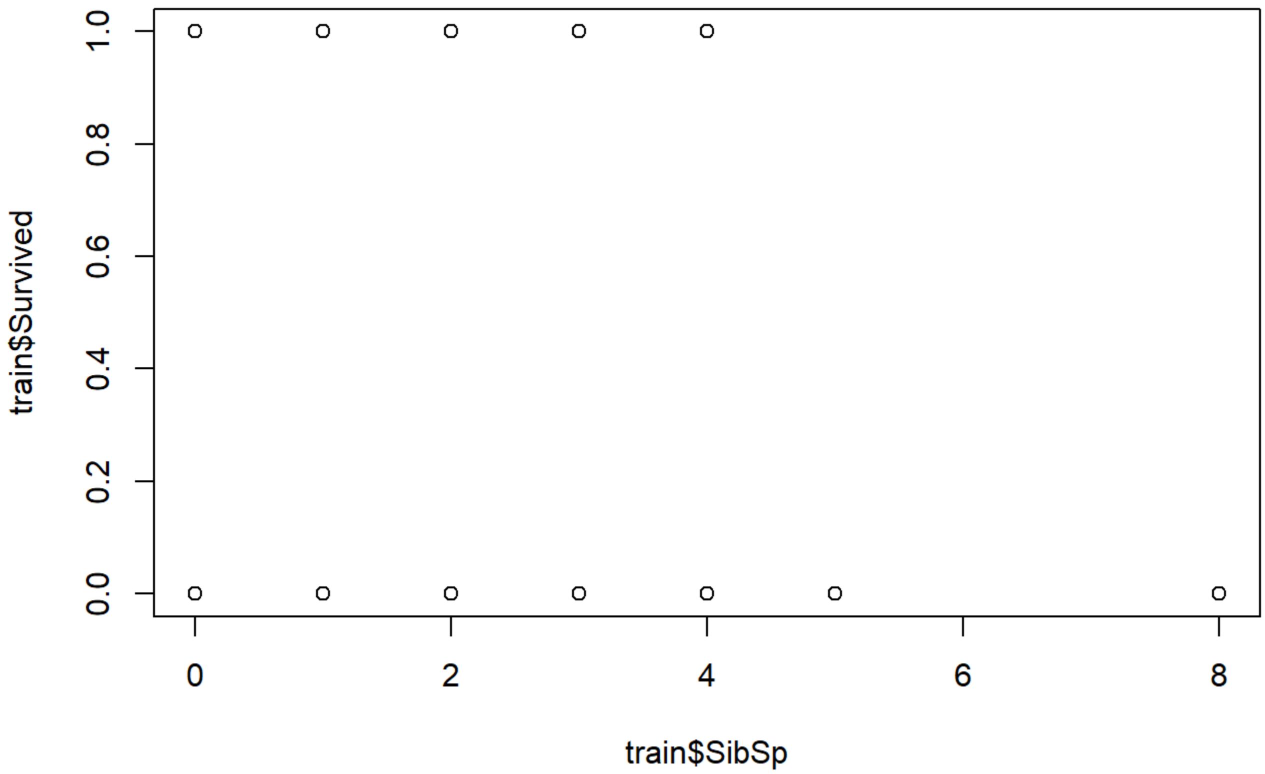

```
plot(train$Sex, train$Survived)
```



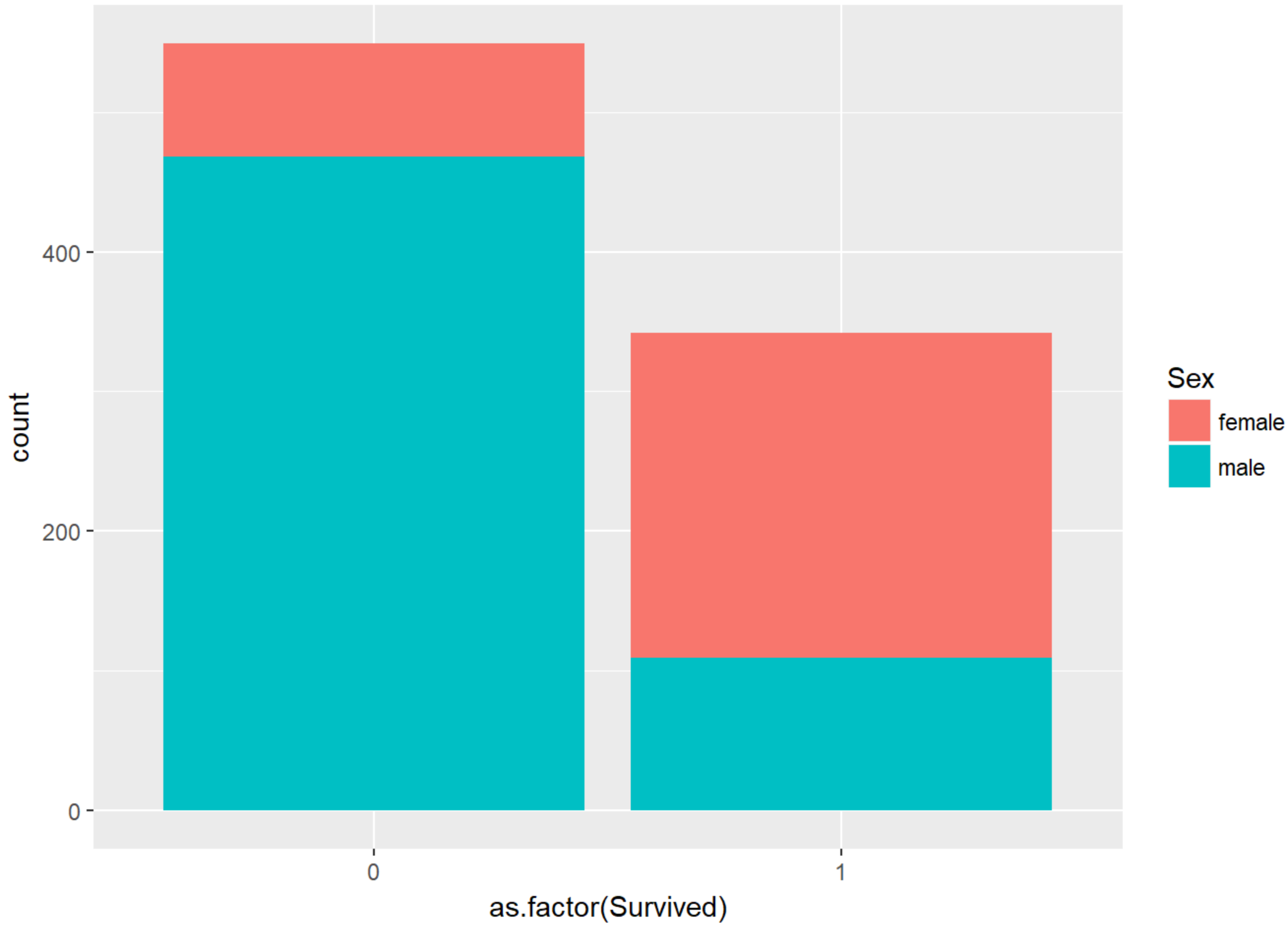

```
plot(train$Fare, train$Survived)
```



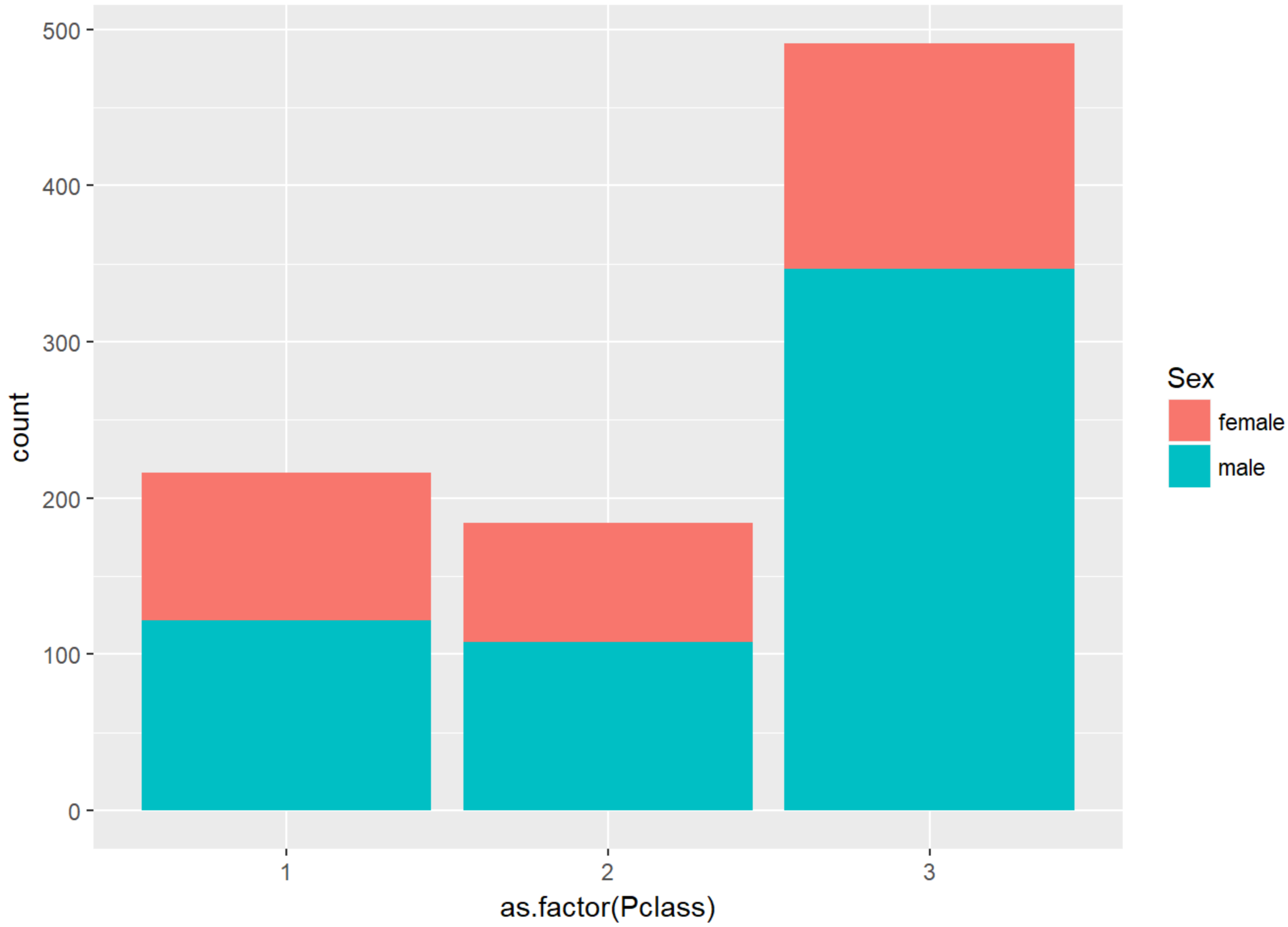

```
plot(train$SibSp, train$Survived)
```




```
ggplot(train, aes(as.factor(Survived), fill=Sex)) + geom_bar()
```

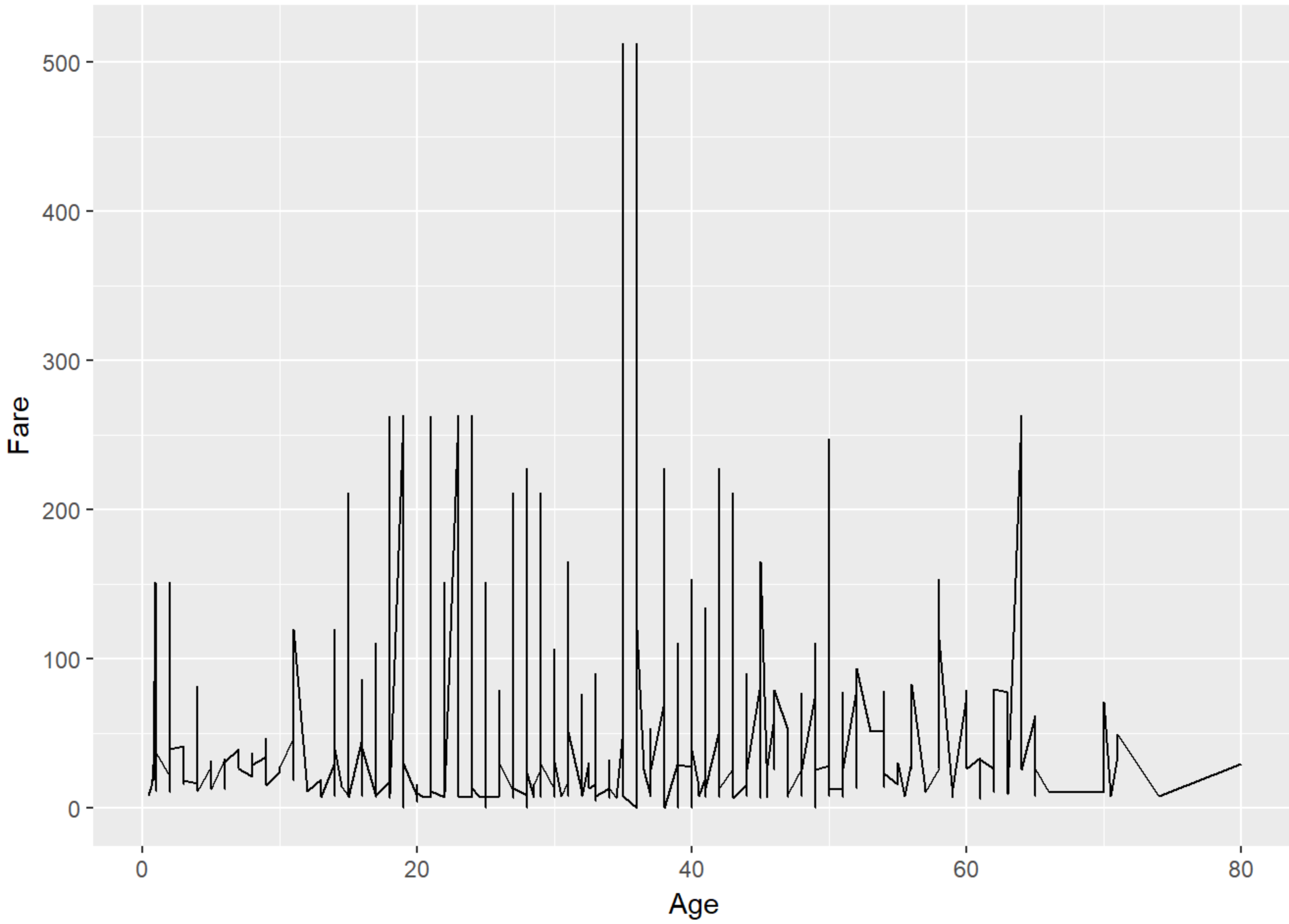



```
ggplot(train, aes(as.factor(Pclass), fill=Sex)) + geom_bar()
```



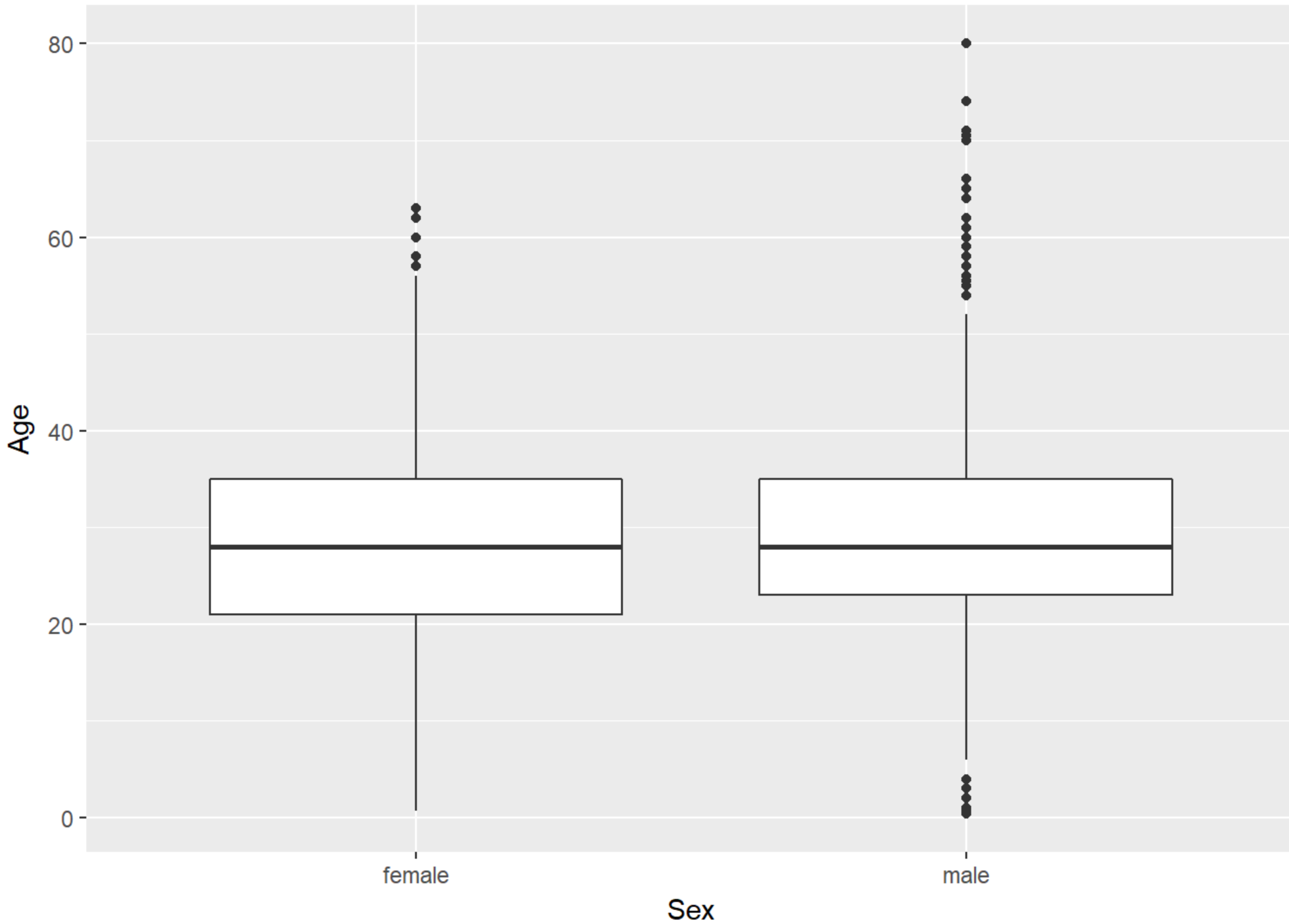

```
ggplot(train, aes(Age, Fare)) + geom_line()
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting
to continuous.
```



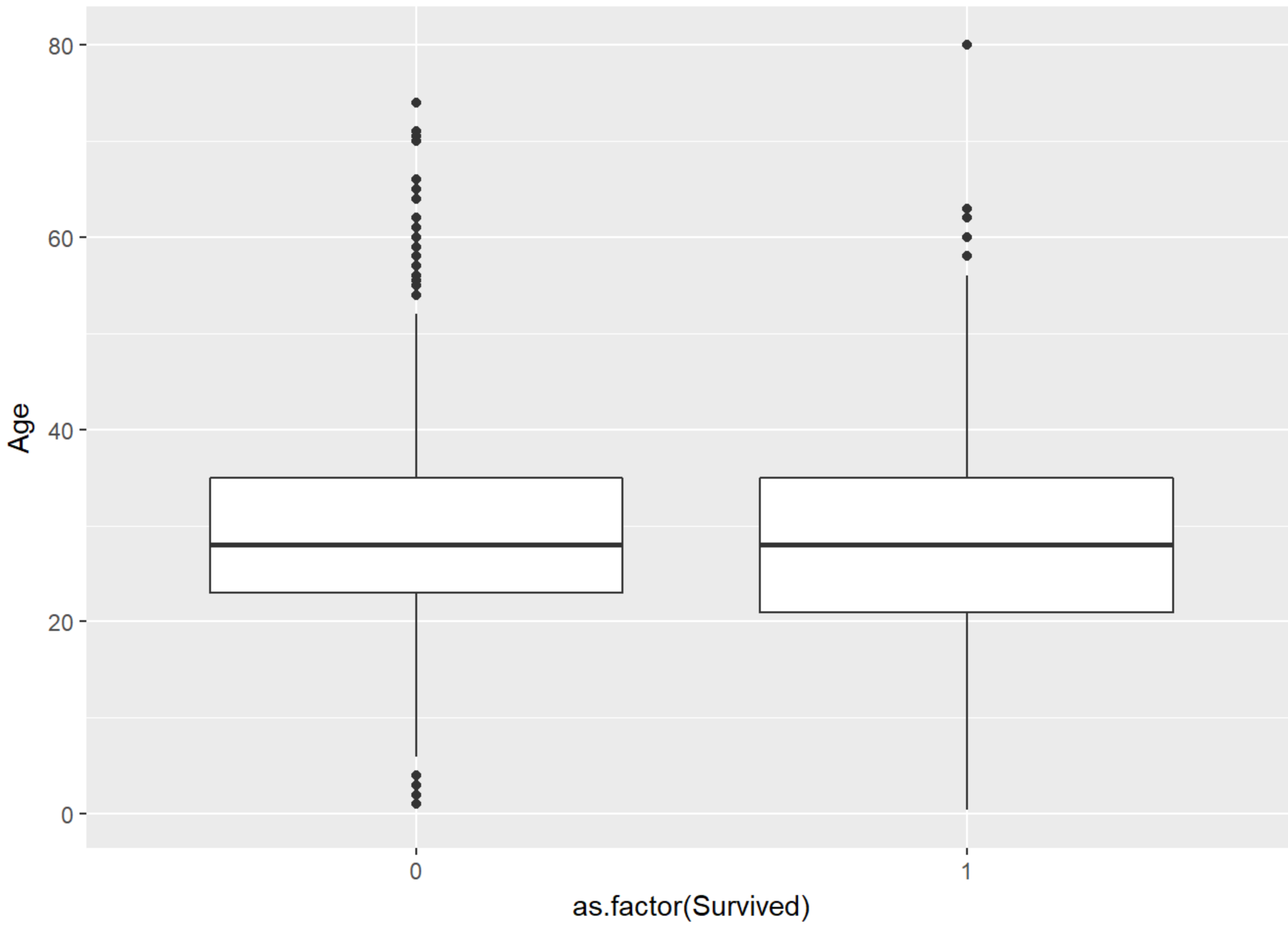

```
ggplot(train, aes(Sex, Age)) + geom_boxplot()
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting
to continuous.
```

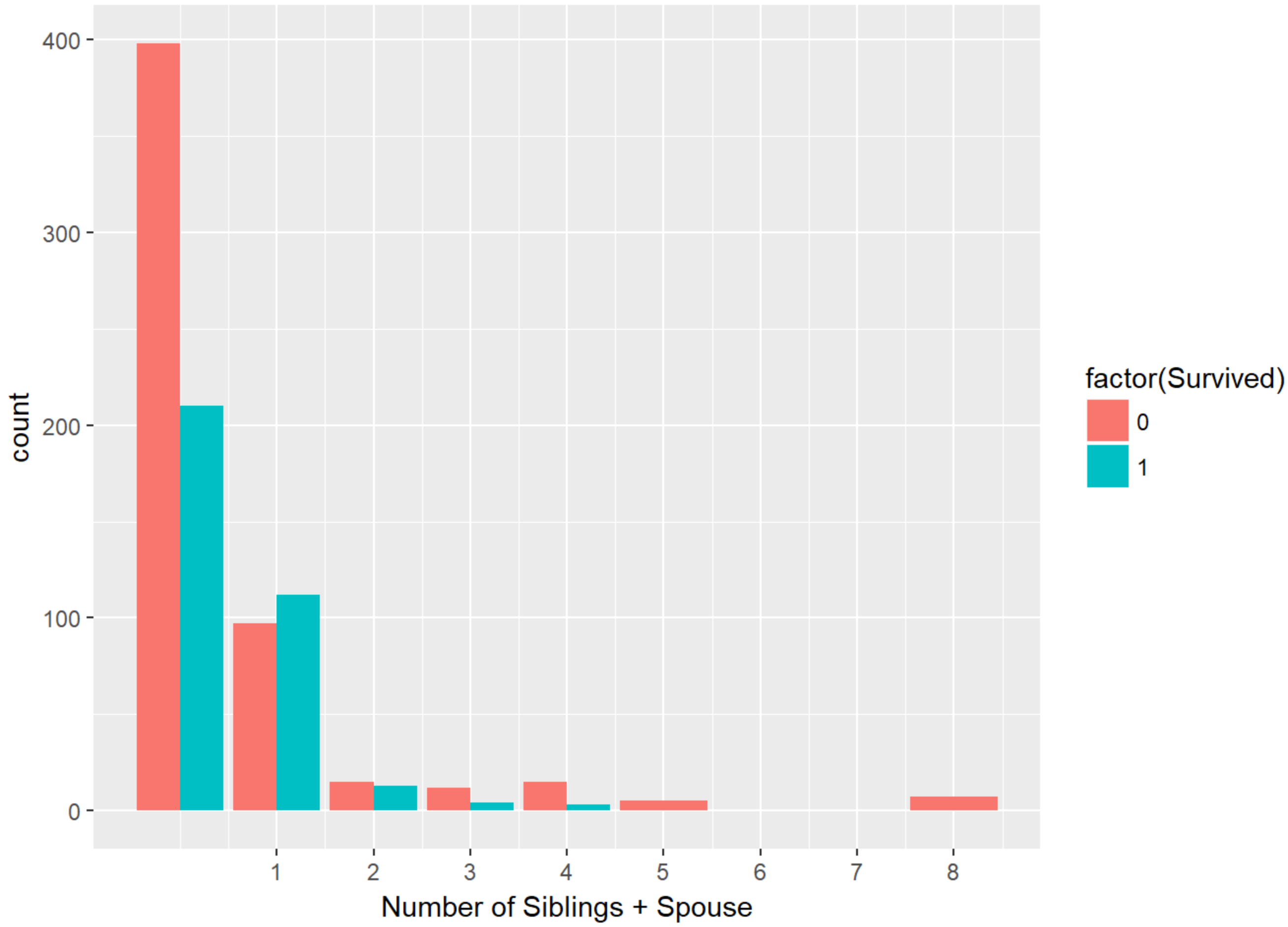



```
ggplot(train, aes(as.factor(Survived), Age)) + geom_boxplot()
```

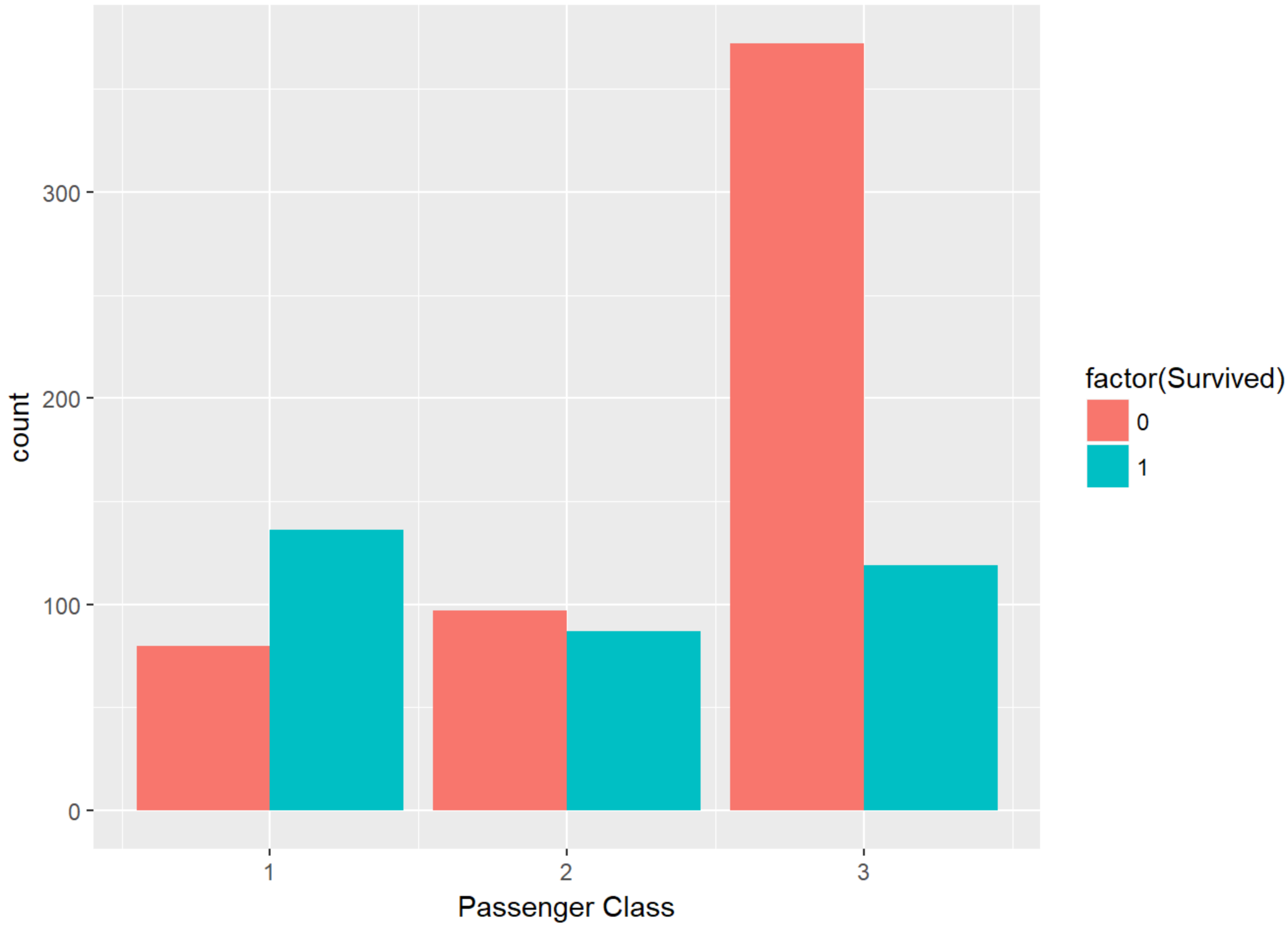
```
## Don't know how to automatically pick scale for object of type impute. Defaulting
to continuous.
```



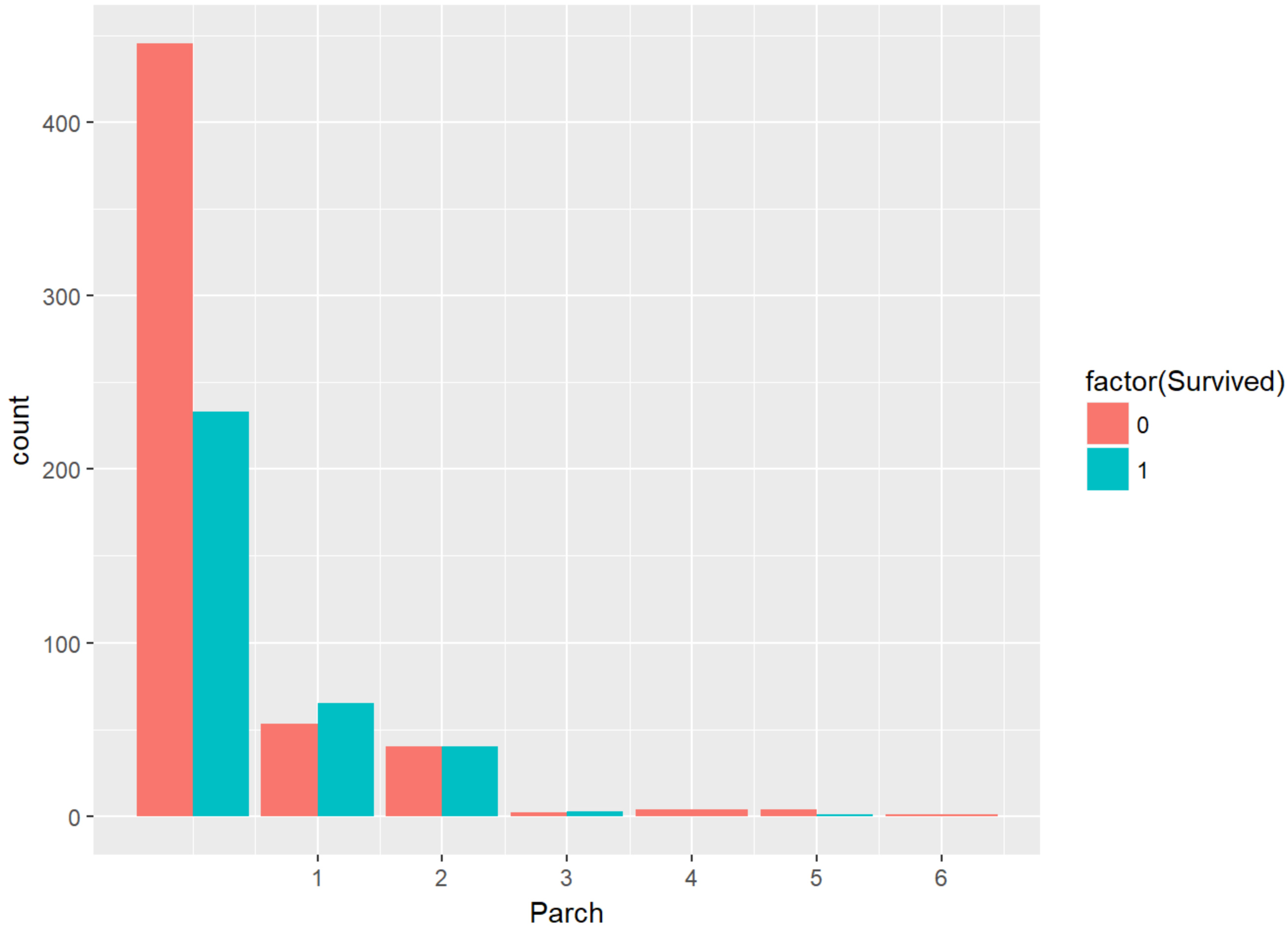

```
#the code for the below ggplots is a template taken from an r website.
(ggplot(train[1:891,], aes(x = SibSp, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Number of Siblings + Spouse'))
```




```
(ggplot(train[1:891,], aes(x = Pclass, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Passenger Class'))
```




```
(ggplot(train[1:891,], aes(x = Parch, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Parch'))
```



1st and 2nd class passengers fared much better than 3rd. Females disproportionately survived despite there being more males in 1st and 2nd class. there doesn't appear to be an age by gender difference.

Some things that stick out a bit: having one or tw o parents or children seems to improve chances of survival; same for siblings and spouse, how ever large families seem to get penalized. How ever, using the box plot things aren't so clear regarding age and survival.

```
t.test(train$Age~train$Survived)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  train$Age by train$Survived
## t = 1.8966, df = 671.15, p-value = 0.05831
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06126264  3.53486344
## sample estimates:
## mean in group 0 mean in group 1 
##        30.02823        28.29143
```

```r
train$child <- 0
train$child[train$Age < 8] <- 1
test$child <- 0
test$child[test$Age < 8] <- 1
as.factor(test$child)
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [246] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [281] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [316] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [351] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [386] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```r
as.factor(train$child)
```

```
##   [1] 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0
## [176] 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [246] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
## [281] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
## [316] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
## [351] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0
## [386] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [421] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
## [456] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
## [491] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [526] 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [561] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [596] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [631] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [666] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## [701] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [736] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [771] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
## [806] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
## [841] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [876] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```
summary(train$child)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.00000 0.00000 0.05612 0.00000 1.00000
```

```
cxs <- table(train$Survived, train$child)
assocstats(cxs)
```

```
##                      X^2 df   P(> X^2)
## Likelihood Ratio 19.030  1 1.2866e-05
## Pearson          19.646  1 9.3216e-06
## 
## Phi-Coefficient   : 0.148 
## Contingency Coeff.: 0.147 
## Cramer's V        : 0.148
```

```
summary(glm(Survived ~ child, data = train, family = binomial))
```

```
##
## Call:
## glm(formula = Survived ~ child, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5096  -0.9551  -0.9551   1.4174   1.4174
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.54842    0.07157  -7.662 1.83e-14 ***
## child        1.30219    0.31150   4.180 2.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance: 1167.6  on 889  degrees of freedom
## AIC: 1171.6
##
## Number of Fisher Scoring iterations: 4
```

```
exp(1.30219)
```

```
## [1] 3.677341
```

While there is no overall difference in survival by mean age, there is a small but significant association w ith being a young child (>8) and surviving; from the logistic model, young children are 3.68 times as likely to survive that those older.

Lets make a few more tables to quantify things

```
table(train$Survived)
```

```
##
##   0   1
## 549 342
```

```
table(train$Survived, train$Pclass)
```

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

```
table(train$Embarked, train$Survived)
```

```
##
##         0   1
##         0   2
##     C  75  93
##     Q  47  30
##     S 427 217
```

```
prop.table(table(train$Pclass, train$Survived))
```

```
##
##              0          1
##     1 0.08978676 0.15263749
##     2 0.10886644 0.09764310
##     3 0.41750842 0.13355780
```

```
prop.table(table(train$SibSp, train$Survived))
```

```
##
##               0           1
##     0 0.446689113 0.235690236
##     1 0.108866442 0.125701459
##     2 0.016835017 0.014590348
##     3 0.013468013 0.004489338
##     4 0.016835017 0.003367003
##     5 0.005611672 0.000000000
##     8 0.007856341 0.000000000
```

```
with(train, aggregate(Survived ~ Pclass + Sex, data=train, FUN=sum))
```

```
##   Pclass    Sex Survived
## 1      1 female       91
## 2      2 female       70
## 3      3 female       72
## 4      1   male       45
## 5      2   male       17
## 6      3   male       47
```

```
wilcox.test(train$Fare~train$Survived)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train$Fare by train$Survived
## W = 57806, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
logtest <- glm(train$Survived~train$Embarked, family = binomial)
summary(logtest)
```

```
##
## Call:
## glm(formula = train$Survived ~ train$Embarked, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2700  -0.9065  -0.9065   1.3730   1.4750
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)        13.57     378.59   0.036    0.971
## train$EmbarkedC   -13.35     378.59  -0.035    0.972
## train$EmbarkedQ   -14.02     378.59  -0.037    0.970
## train$EmbarkedS   -14.24     378.59  -0.038    0.970
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance: 1157.0  on 887  degrees of freedom
## AIC: 1165
##
## Number of Fisher Scoring iterations: 12
```

Unsurprisingly, the t-test validates the above graphics regarding fares: those w ho survived spent significantly more–because of the distribution, a non-parametric test w as preferred. Embarked does not seem statistically related to survival.

With w hat w e know, lets fit a model. The sample is not very large and w e have (w hat I consider) to be a number of predictors. I think this scenario favors a less flexible option given my current bag of tools.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.3
```

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

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
set.seed(56741)
logfit.train1 <- glm(Survived ~ Sex + Fare + Age + child + Pclass + SibSp + Parch,
data = train, family = binomial)
summary(logfit.train1)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Fare + Age + child + Pclass +
##     SibSp + Parch, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1762  -0.5925  -0.4282   0.5786   2.5002
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.221864   0.554568   7.613 2.68e-14 ***
## Sexmale     -2.876170   0.205353 -14.006  < 2e-16 ***
## Fare         0.003993   0.002510   1.591 0.111572
## Age         -0.021126   0.008701  -2.428 0.015182 *
## child        2.167599   0.518276   4.182 2.89e-05 ***
## Pclass      -0.998850   0.142072  -7.031 2.06e-12 ***
## SibSp       -0.465480   0.122710  -3.793 0.000149 ***
## Parch       -0.225160   0.124315  -1.811 0.070110 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  769.93  on 883  degrees of freedom
## AIC: 785.93
##
## Number of Fisher Scoring iterations: 5
```

```
predtrain <- predict(logfit.train1, type = 'response')
predtrain <- ifelse(predtrain > 0.5,1,0)
confusionMatrix(data=predtrain, reference=train$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 490 101
##          1  59 241
##
##                Accuracy : 0.8204
##                  95% CI : (0.7936, 0.8451)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6114
##  Mcnemar's Test P-Value : 0.00119
##
##             Sensitivity : 0.8925
##             Specificity : 0.7047
##          Pos Pred Value : 0.8291
##          Neg Pred Value : 0.8033
##              Prevalence : 0.6162
##          Detection Rate : 0.5499
##    Detection Prevalence : 0.6633
##       Balanced Accuracy : 0.7986
##
##        'Positive' Class : 0
##
```

```
log.predictions = predict(logfit.train1, test, type = 'response')
log.predictions <- ifelse(log.predictions > 0.5,1,0)
log.predictions[is.na(log.predictions)] <- 0
output <- data.frame(PassengerID = test$PassengerId, Survived = log.predictions)
table(output$Survived)
```

```
##
##   0   1
## 290 128
```

```
write.csv("C:/sas/r/TPred.csv" , x = output, row.names = FALSE)
```

Lets try LDA

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.4.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda.tfit <- with(train, lda(Survived ~ Sex + Fare + Age + child + Pclass + SibSp +
Parch, data = train))
ldat <- table(predict(lda.tfit)$class, train$Survived)
confusionMatrix(data=ldat, reference=train$Survived)
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 482 100
##   1  67 242
##
##                Accuracy : 0.8126
##                  95% CI : (0.7854, 0.8377)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5964
##  Mcnemar's Test P-Value : 0.01328
##
##             Sensitivity : 0.8780
##             Specificity : 0.7076
##          Pos Pred Value : 0.8282
##          Neg Pred Value : 0.7832
##              Prevalence : 0.6162
##          Detection Rate : 0.5410
##    Detection Prevalence : 0.6532
##       Balanced Accuracy : 0.7928
##
##        'Positive' Class : 0
##
```

And QDA

```
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 3.4.3
```

```
qda.fit <- with(train, qda(Survived ~ Sex + Fare + Age + child + Pclass + SibSp + P
arch, data = train))
qdadat <- table(predict(qda.fit)$class, train$Survived)
confusionMatrix(data=qdadat, reference=train$Survived)
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 481 106
##   1  68 236
##
##                Accuracy : 0.8047
##                  95% CI : (0.7771, 0.8303)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5783
##  Mcnemar's Test P-Value : 0.005032
##
##             Sensitivity : 0.8761
##             Specificity : 0.6901
##          Pos Pred Value : 0.8194
##          Neg Pred Value : 0.7763
##              Prevalence : 0.6162
##          Detection Rate : 0.5398
##    Detection Prevalence : 0.6588
##       Balanced Accuracy : 0.7831
##
##        'Positive' Class : 0
##
```

Of these three models, the logistic regression seems to be the best performer having a training error rate of just under 18%.

Examining w hat people have done online, tw o things jump out at me. First is that flexible models seem to be preferred (one example using Random Forest had a training error rate of 10%, though that of course doesn’t guarantee the same test error rate), and that more time is spent on cleaning data and creating “proxy” variables. A number of submissions spent considerable time cleaning and recoding variables based on passenger title: w hat better indicator of class could their for a ship departing the United Kingdom in 1912? As my explorations indicate, w ealth, proxy’s thereof, sex and age are the most strongly associated indicators of survival. This is very creative and had I not looked at examples I w ould have never thought to do it.

That said, most examples spent very little time visualizing the data, examining its correlation structure, etc. Also surprisingly, rarely w ere statistical tests used to drive model building. Additionally, most w orked on a combined data set, only splitting into train and test at the last moment.

My script is very clunky: I am still very new to r.