Titanic

Shrinath

Feb 29, 2018

Description: Exploratory analysis on Titanic misshap

missmap is use for checking the missing values in the train data.Roughly 20 percent of the Age data is missing.

```
df.train <- read.csv('C:/Users/patel/Documents/udemy/titanic_train.csv')
head(df.train)
```

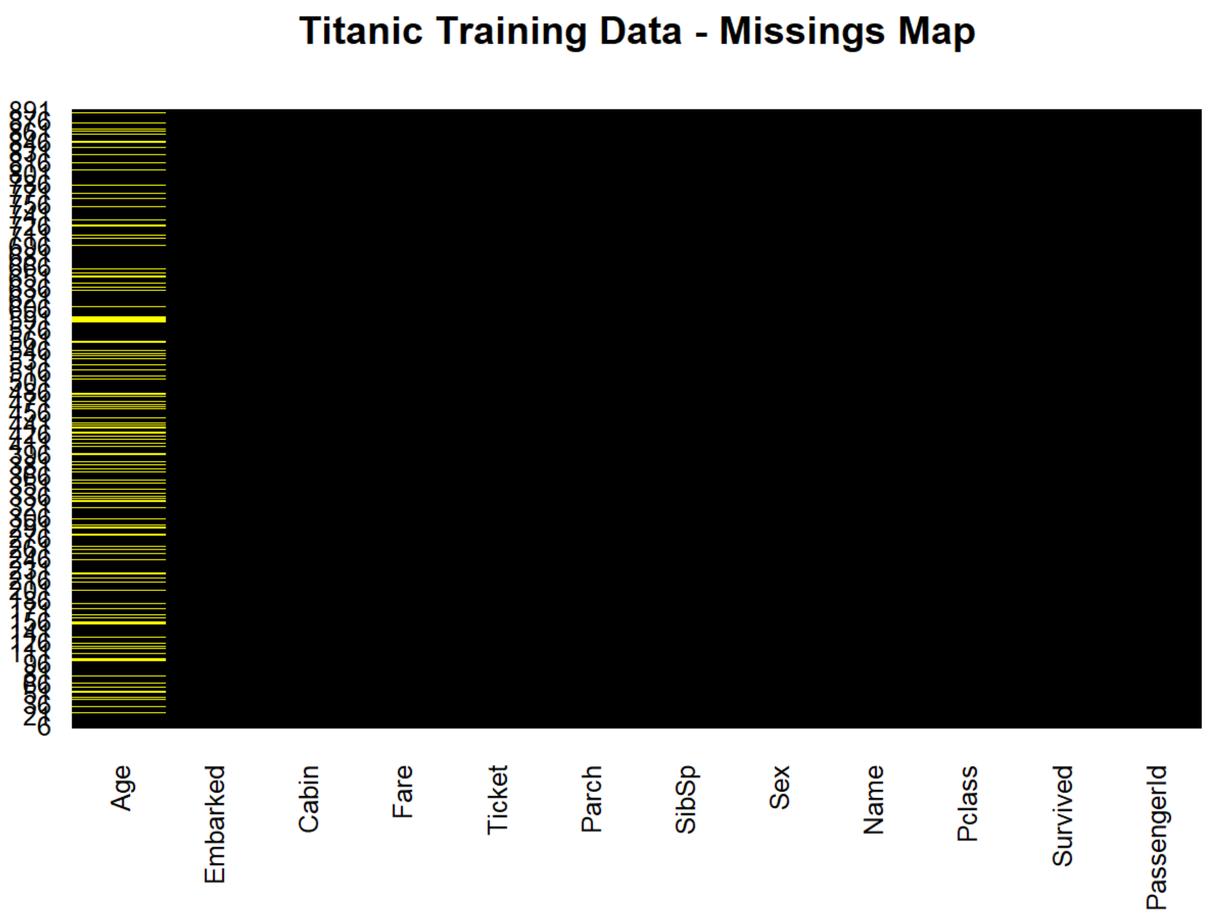
```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                  Name    Sex Age SibSp
## 1                             Braund, Mr. Owen Harris   male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                              Heikkinen, Miss. Laina female  26     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
## 5                            Allen, Mr. William Henry   male  35     0
## 6                                    Moran, Mr. James   male  NA     0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
## 3     0 STON/O2. 3101282  7.9250              S
## 4     0           113803 53.1000  C123        S
## 5     0           373450  8.0500              S
## 6     0           330877  8.4583              Q
```

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(df.train, main="Titanic Training Data - Missings Map",
        col=c("yellow", "black"), legend=FALSE)
```

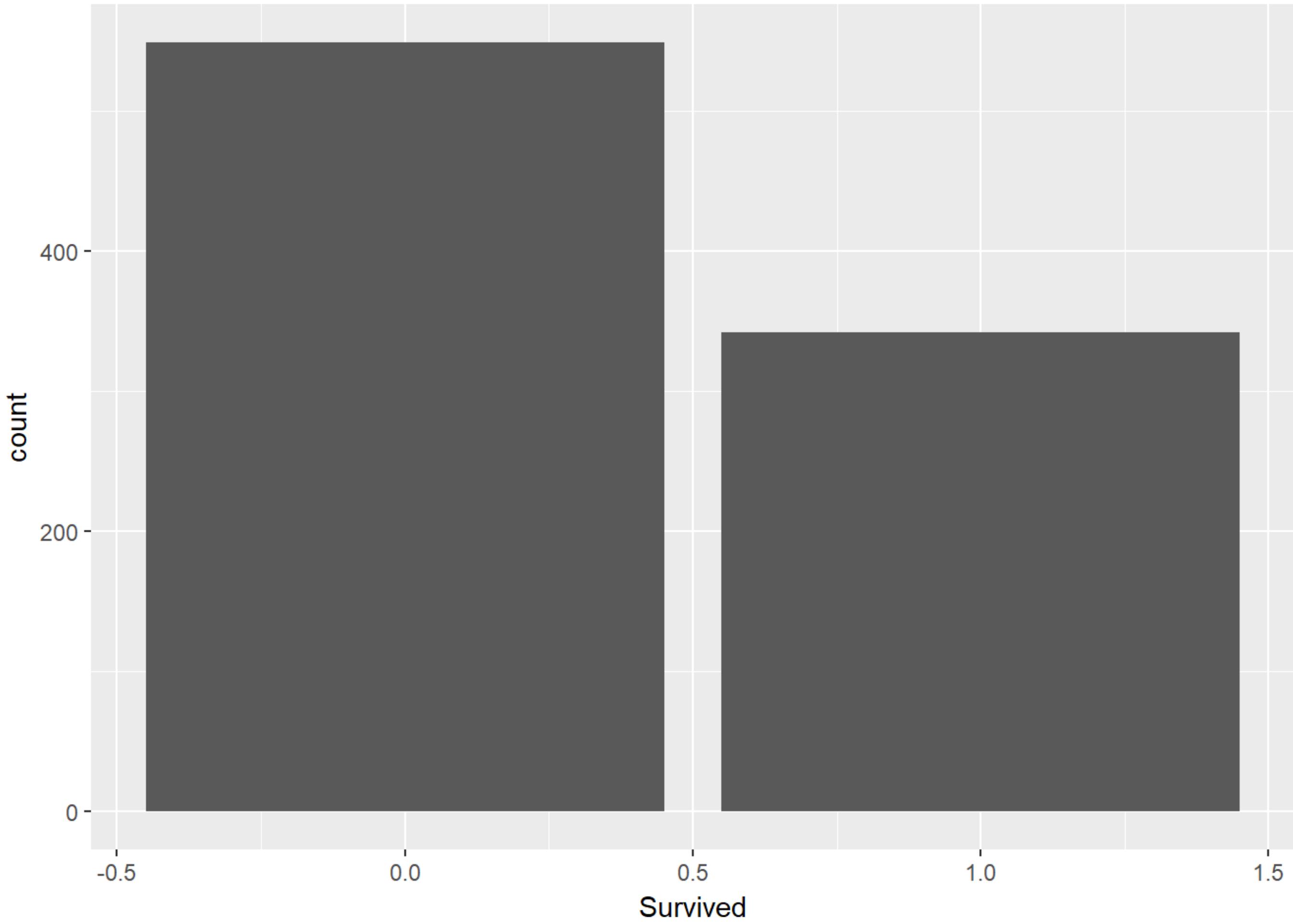


Data Visualization with ggplot2

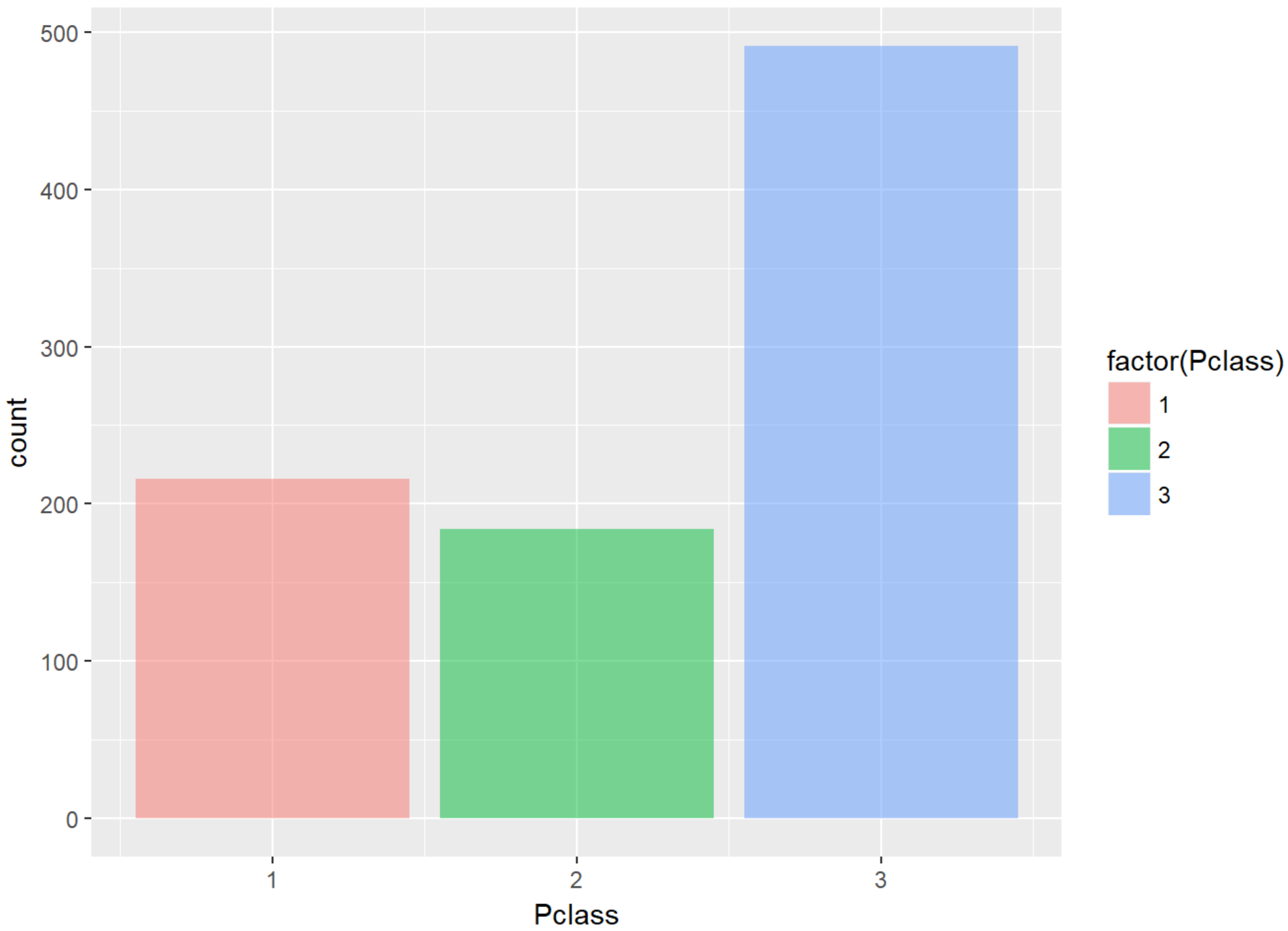
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

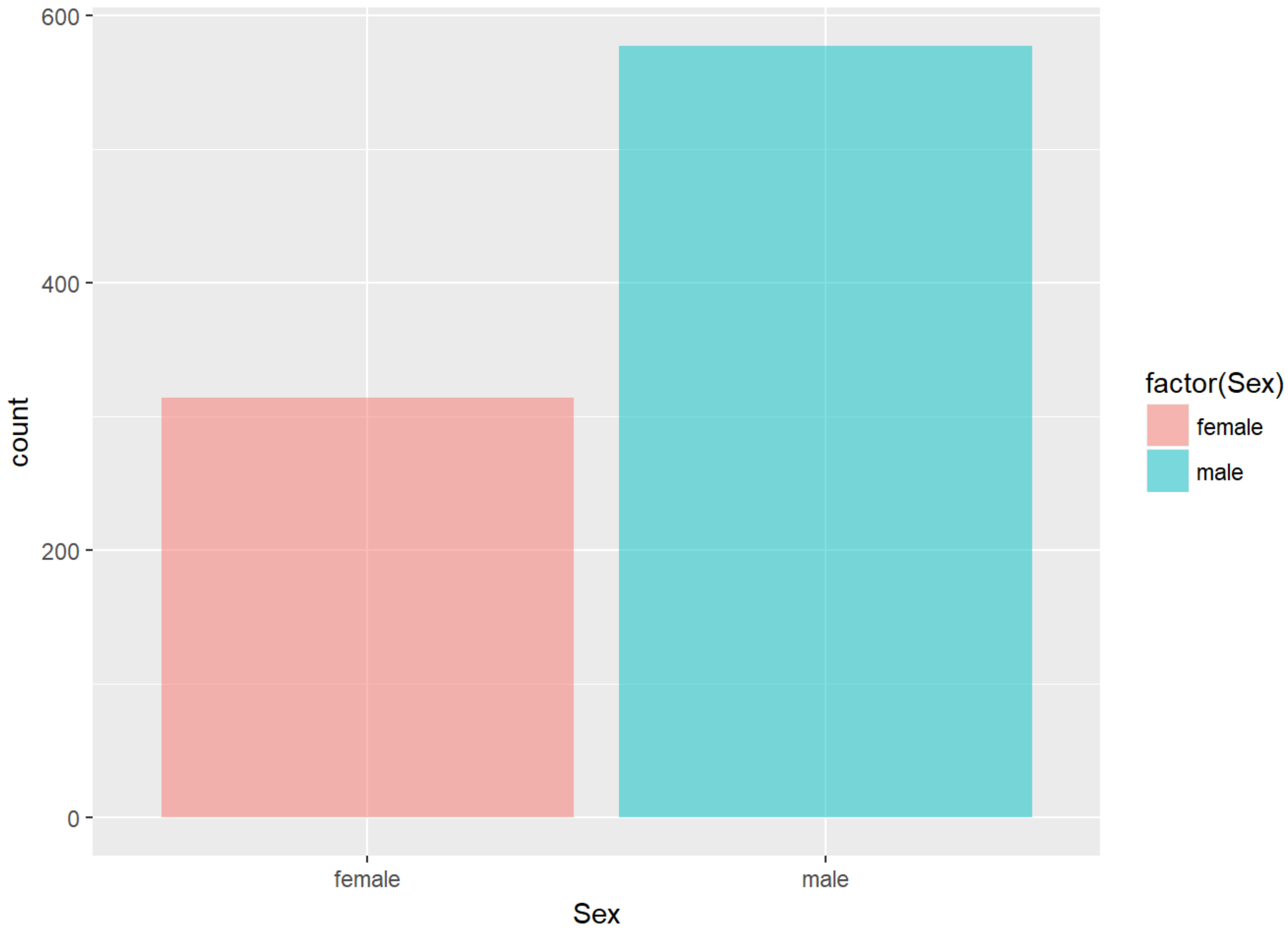
```
ggplot(df.train,aes(Survived)) + geom_bar()
```



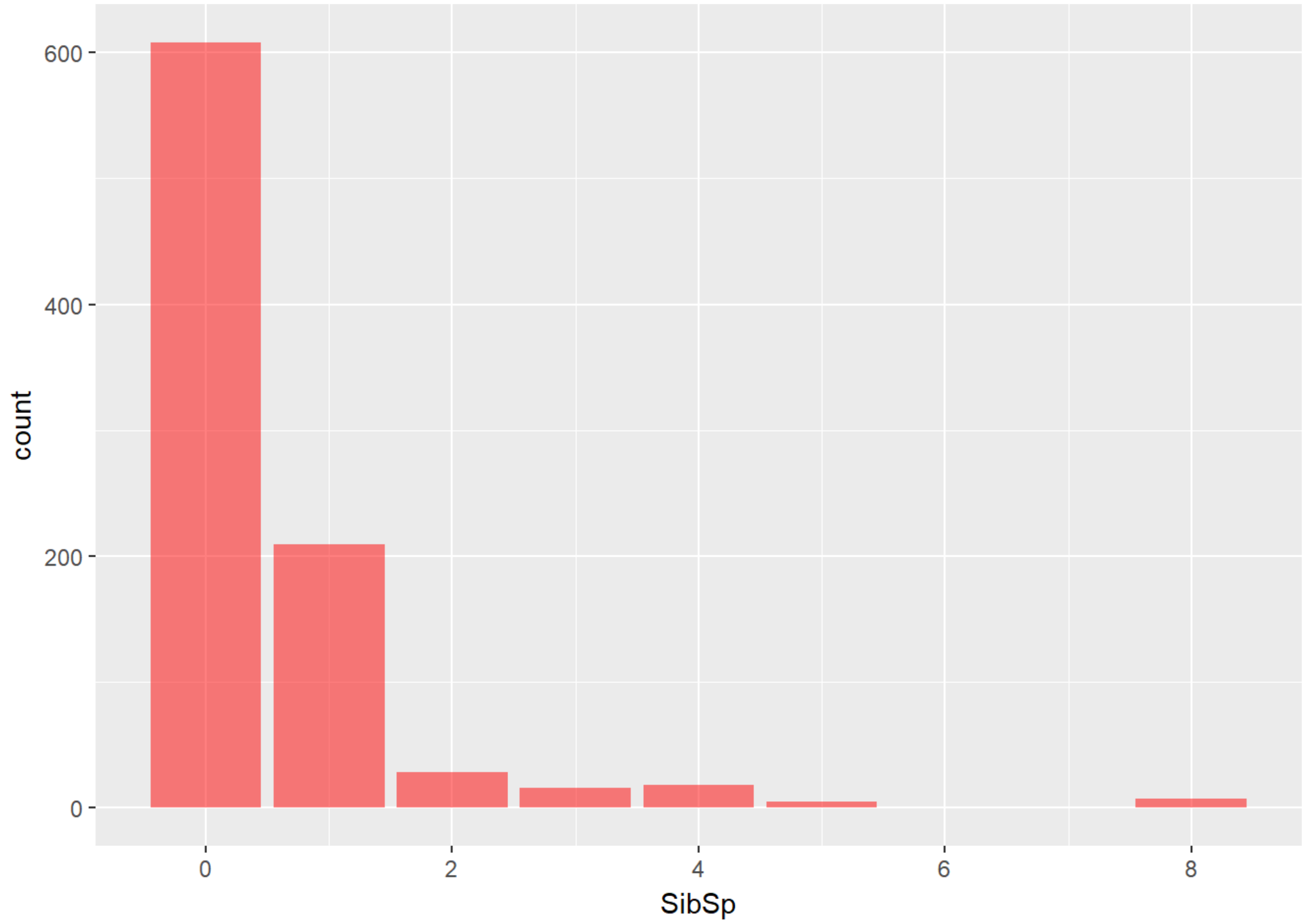

```
ggplot(df.train,aes(Pclass)) + geom_bar(aes(fill=factor(Pclass)),alpha=0.5)
```



```
ggplot(df.train,aes(Sex)) + geom_bar(aes(fill=factor(Sex)),alpha=0.5)
```

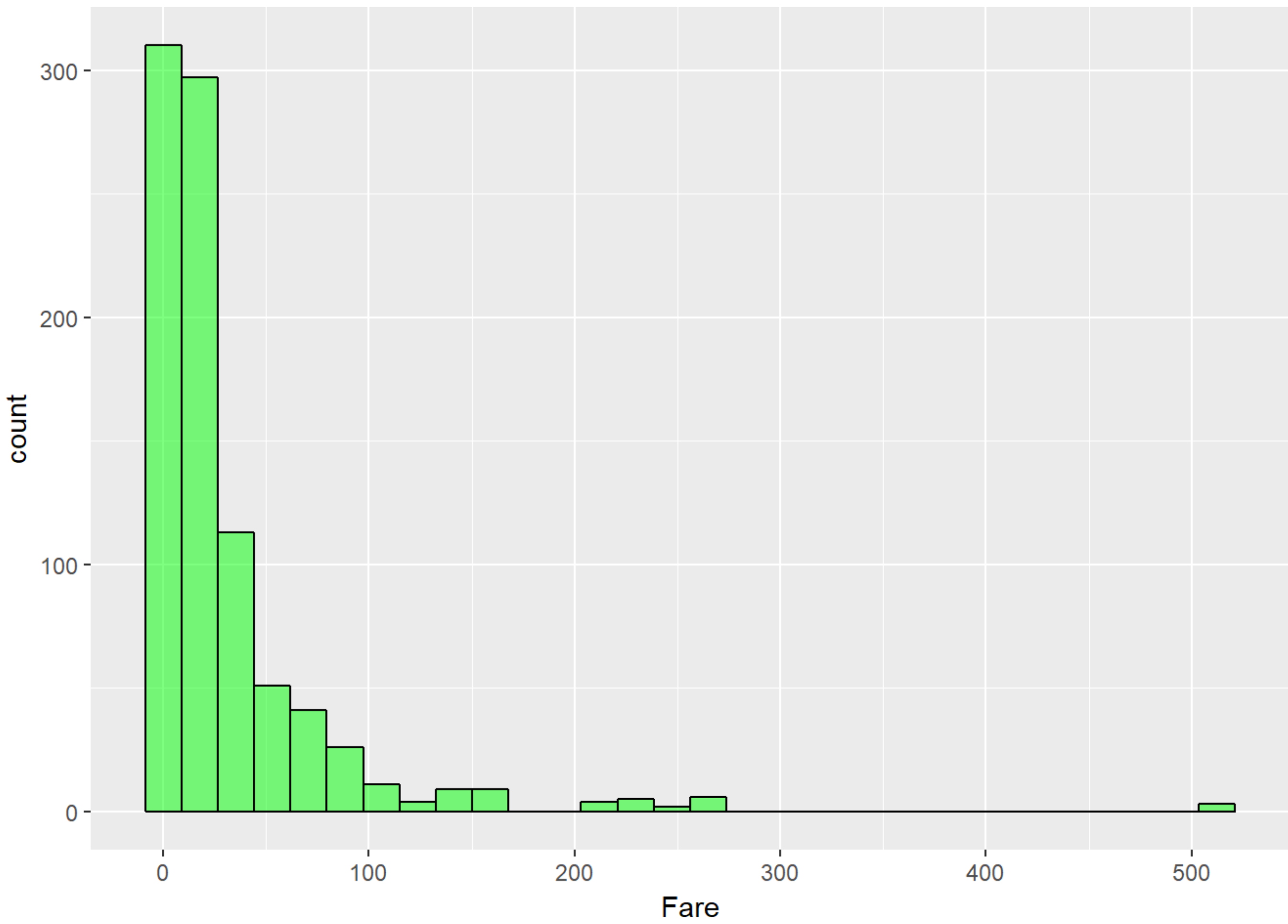



```
ggplot(df.train,aes(SibSp)) + geom_bar(fill='red',alpha=0.5)
```




```
ggplot(df.train,aes(Fare)) + geom_histogram(fill='green',color='black',alpha=0.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



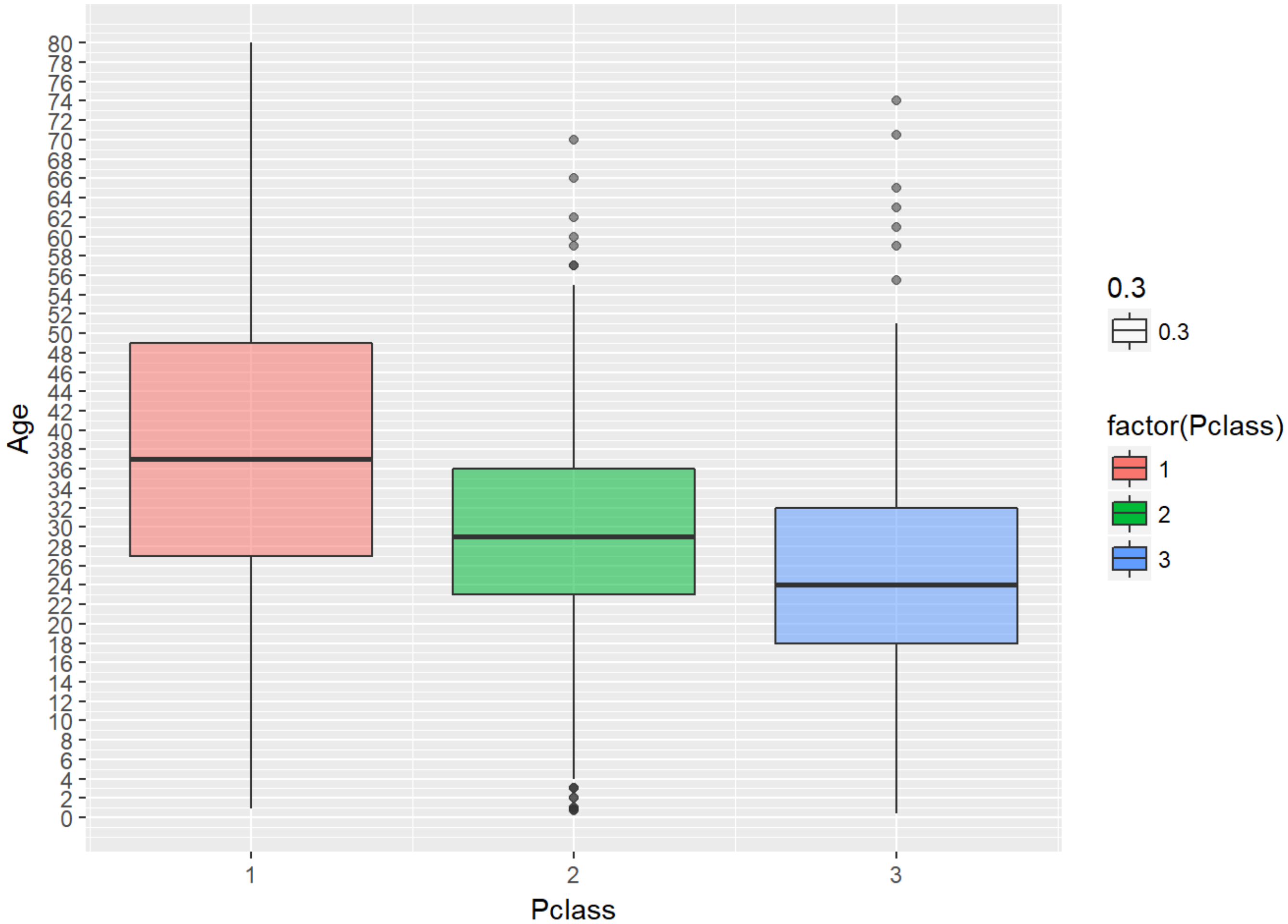
Data Cleaning

We want to fill in missing age data instead of just dropping the missing age data rows.

One way to do this is by filling in the mean age of all the passengers (imputation).

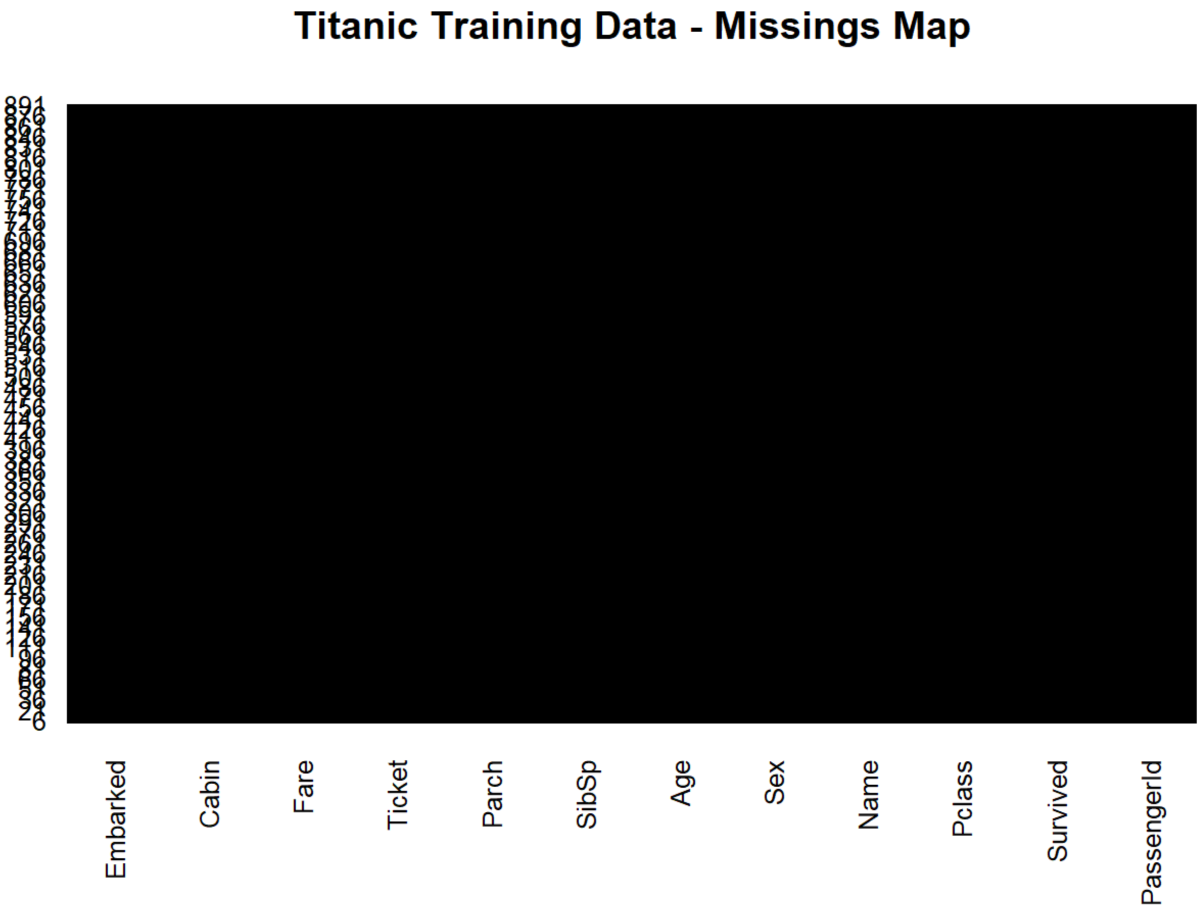
```
ggplot(df.train,aes(Pclass,Age)) + geom_boxplot(aes(group=Pclass,fill=factor(Pclass
),alpha=0.3)) + scale_y_continuous(breaks = seq(min(0), max(80), by = 2))
```

```
## Warning: Removed 177 rows containing non-finite values (stat_boxplot).
```



We can see the wealthier passengers in the higher classes tend to be older, which makes sense. # We'll use these average age values to impute based on Pclass for Age

```r
impute_age <- function(age,class){
  out <- age
  for (i in 1:length(age)){

    if (is.na(age[i])){

      if (class[i] == 1){
        out[i] <- 37

      }else if (class[i] == 2){
        out[i] <- 29

      }else{
        out[i] <- 24
      }
    }else{
      out[i]<-age[i]
    }
  }
  return(out)
}


fixed.ages <- impute_age(df.train$Age,df.train$Pclass)
df.train$Age <- fixed.ages




missmap(df.train, main="Titanic Training Data - Missings Map",
        col=c("yellow", "black"), legend=FALSE)
```



#Building a Logistic Regression Model #Let's begin by doing a final "clean-up" of our data by removing the features we won't be using and making sure that the features are of the correct data type.

```
str(df.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 24 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 ...
```

```
head(df.train,3)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
##                                                Name    Sex Age SibSp
## 1                           Braund, Mr. Owen Harris   male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                            Heikkinen, Miss. Laina female  26     0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
## 3     0 STON/O2. 3101282  7.9250              S
```

selected the relevant columns for training

```
library(dplyr)
```

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

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

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

```
df.train <- df.train[,c(2,3,5,6,7,8,10,12)]
head(df.train,3)
```

```
##   Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 1        0      3   male  22     1     0  7.2500        S
## 2        1      1 female  38     1     0 71.2833        C
## 3        1      3 female  26     0     0  7.9250        S
```

Now let’s set factor columns

```
str(df.train)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 24 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 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
df.train$Survived <- factor(df.train$Survived)
df.train$Pclass <- factor(df.train$Pclass)
df.train$Parch <- factor(df.train$Parch)
df.train$SibSp <- factor(df.train$SibSp)
```

Train the Model

```
log.model <- glm(formula=Survived ~ . , family = binomial(link='logit'),data = df.t
rain)
summary(log.model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = df.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8158  -0.6134  -0.4138   0.5808   2.4896
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.845e+01  1.660e+03   0.011 0.991134
## Pclass2     -1.079e+00  3.092e-01  -3.490 0.000484 ***
## Pclass3     -2.191e+00  3.161e-01  -6.930 4.20e-12 ***
## Sexmale     -2.677e+00  2.040e-01 -13.123  < 2e-16 ***
## Age         -3.971e-02  8.758e-03  -4.534 5.79e-06 ***
## SibSp1       8.135e-02  2.245e-01   0.362 0.717133
## SibSp2      -2.897e-01  5.368e-01  -0.540 0.589361
## SibSp3      -2.241e+00  7.202e-01  -3.111 0.001862 **
## SibSp4      -1.675e+00  7.620e-01  -2.198 0.027954 *
## SibSp5      -1.595e+01  9.588e+02  -0.017 0.986731
## SibSp8      -1.607e+01  7.578e+02  -0.021 0.983077
## Parch1       3.741e-01  2.895e-01   1.292 0.196213
## Parch2       3.862e-02  3.824e-01   0.101 0.919560
## Parch3       3.655e-01  1.056e+00   0.346 0.729318
## Parch4      -1.586e+01  1.055e+03  -0.015 0.988007
## Parch5      -1.152e+00  1.172e+00  -0.983 0.325771
## Parch6      -1.643e+01  2.400e+03  -0.007 0.994536
## Fare         2.109e-03  2.490e-03   0.847 0.397036
## EmbarkedC   -1.458e+01  1.660e+03  -0.009 0.992995
## EmbarkedQ   -1.456e+01  1.660e+03  -0.009 0.993001
## EmbarkedS   -1.486e+01  1.660e+03  -0.009 0.992857
## ---
## 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:  763.41  on 870  degrees of freedom
## AIC: 805.41
##
## Number of Fisher Scoring iterations: 15
```

Interpretation We can see clearly that Sex,Age, and Class are the most significant features. Which makes sense given the women and children first policy. The null deviance shows how well the response is predicted by the model with nothing but an intercept. The residual deviance shows how well the response is predicted by the model when the predictors are included.

We can also use the residual deviance to test whether the null hypothesis is true (i.e. Logistic regression model provides an adequate fit for the data). This is possible because the deviance is given by the chi-squared value at a certain degrees of freedom. # In order to test for significance, we can find out associated p-values using the below formula in R:

p-value = 1 - pchisq(deviance, degrees of freedom) Using the above values of residual deviance and DF, a p-value showing that there is a significant evidence to support the null hypothesis.

```r
1 - pchisq(763.41, 870)
```

```
## [1] 0.9959952
```

```r
library(caTools)
set.seed(101)

split = sample.split(df.train$Survived, SplitRatio = 0.70)

final.train = subset(df.train, split == TRUE)
final.test = subset(df.train, split == FALSE)
final.log.model <- glm(formula=Survived ~ . , family = binomial(link='logit'),data
= final.train)
summary(final.log.model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = final.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8288  -0.5607  -0.4096   0.6174   2.4898
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.777e+01  2.400e+03   0.007 0.994091
## Pclass2     -1.230e+00  3.814e-01  -3.225 0.001261 **
## Pclass3     -2.160e+00  3.841e-01  -5.624 1.87e-08 ***
## Sexmale     -2.660e+00  2.467e-01 -10.782  < 2e-16 ***
## Age         -3.831e-02  1.034e-02  -3.705 0.000212 ***
## SibSp1      -2.114e-02  2.755e-01  -0.077 0.938836
## SibSp2      -4.000e-01  6.463e-01  -0.619 0.536028
## SibSp3      -2.324e+00  8.994e-01  -2.584 0.009765 **
## SibSp4      -1.196e+00  8.302e-01  -1.440 0.149839
## SibSp5      -1.603e+01  9.592e+02  -0.017 0.986666
## SibSp8      -1.633e+01  1.004e+03  -0.016 0.987019
## Parch1       7.290e-01  3.545e-01   2.056 0.039771 *
## Parch2       1.406e-01  4.504e-01   0.312 0.754892
## Parch3       7.919e-01  1.229e+00   0.645 0.519226
## Parch4      -1.498e+01  1.552e+03  -0.010 0.992300
## Parch5      -9.772e-03  1.378e+00  -0.007 0.994343
## Parch6      -1.635e+01  2.400e+03  -0.007 0.994563
## Fare         3.128e-03  3.091e-03   1.012 0.311605
## EmbarkedC   -1.398e+01  2.400e+03  -0.006 0.995353
## EmbarkedQ   -1.387e+01  2.400e+03  -0.006 0.995386
## EmbarkedS   -1.431e+01  2.400e+03  -0.006 0.995243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 829.60  on 622  degrees of freedom
## Residual deviance: 530.63  on 602  degrees of freedom
## AIC: 572.63
##
## Number of Fisher Scoring iterations: 15
```

```
fitted.probabilities <- predict(final.log.model,newdata=final.test,type='response')
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != final.test$Survived)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.798507462686567"
```

Looks like we were able to achieve around 80% accuracy

```
table(final.test$Survived, fitted.probabilities > 0.5)
```

```
##
##       FALSE TRUE
##    0    140   25
##    1     29   74
```