# Data Exploration, Visualization, and Feature Engineering using R

*Yuhui Zhang, and Raja Iqbal*

## Titanic tragedy data

---

### Reading RAW training data

- Download the data set "Titanic_train.csv" from https://raw.githubusercontent.com/datasciencedojo/datasets/master/Titanic_train.csv
- Set working directory of R to the directory of the file using setwd()

```
titanic = read.csv('Titanic_train.csv')
```

---

### Look at the first few rows

What would be some good features to consider here?

```
options(width = 110)
head(titanic)
```

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

---

### What is the data type of each column?

```
sapply(titanic,class)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch
##   "integer"   "integer"   "integer"    "factor"    "factor"   "numeric"   "integer"   "integer"
##        Fare       Cabin    Embarked
##   "numeric"    "factor"    "factor"
```

---

## Converting class label to a factor

```
titanic$Survived = factor(titanic$Survived, labels=c("died", "survived"))
titanic$Embarked = factor(titanic$Embarked, labels=c("unkown", "Cherbourg", "Queenstown", "Southampton"))
sapply(titanic,class)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch
##   "integer"    "factor"   "integer"    "factor"    "factor"   "numeric"   "integer"   "integer"
##        Fare       Cabin    Embarked
##   "numeric"    "factor"    "factor"
```

```
str(titanic$Survived)
```

```
##  Factor w/ 2 levels "died","survived": 1 2 2 2 1 1 1 1 2 2 ...
```

```
str(titanic$Sex)
```

```
##  Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
```

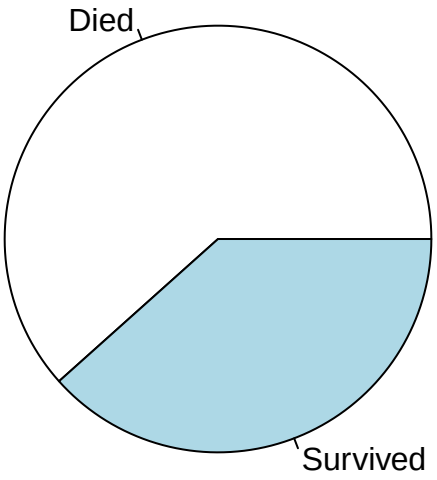
---

## Class distribution - PIE Charts

```
survivedTable = table(titanic$Survived)
survivedTable
```

```
##
##     died survived
##      549      342
```

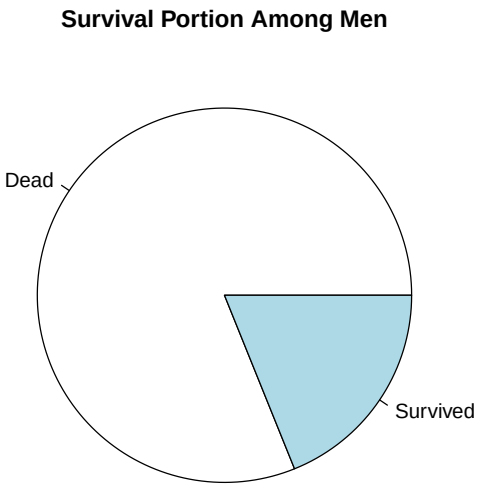
```
par(mar = c(0, 0, 0, 0), oma = c(0, 0, 0, 0))
pie(survivedTable,labels=c("Died","Survived"))
```

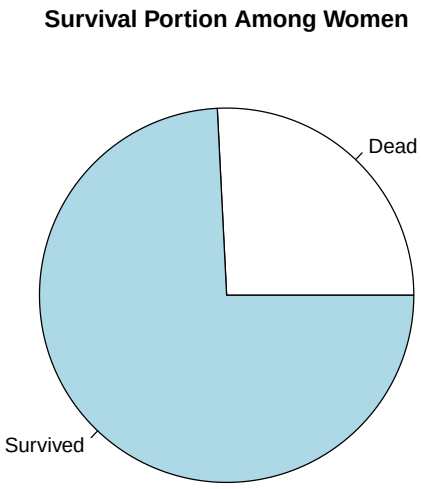



---

## Is Sex a good predictor?

```
male = titanic[titanic$Sex=="male",]
female = titanic[titanic$Sex=="female",]
par(mfrow = c(1, 2), mar = c(0, 0, 2, 0), oma = c(0, 1, 0, 1))
pie(table(male$Survived),labels=c("Dead","Survived"),  main="Survival Portion Among Men")
pie(table(female$Survived),labels=c("Dead","Survived"), main="Survival Portion Among Women")
```




---

## Is Age a good predictor?

```
Age <- titanic$Age; summary(Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

How about summary segmented by **survival**

```
summary(titanic[titanic$Survived=="died",]$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00   21.00   28.00   30.63   39.00   74.00     125
```

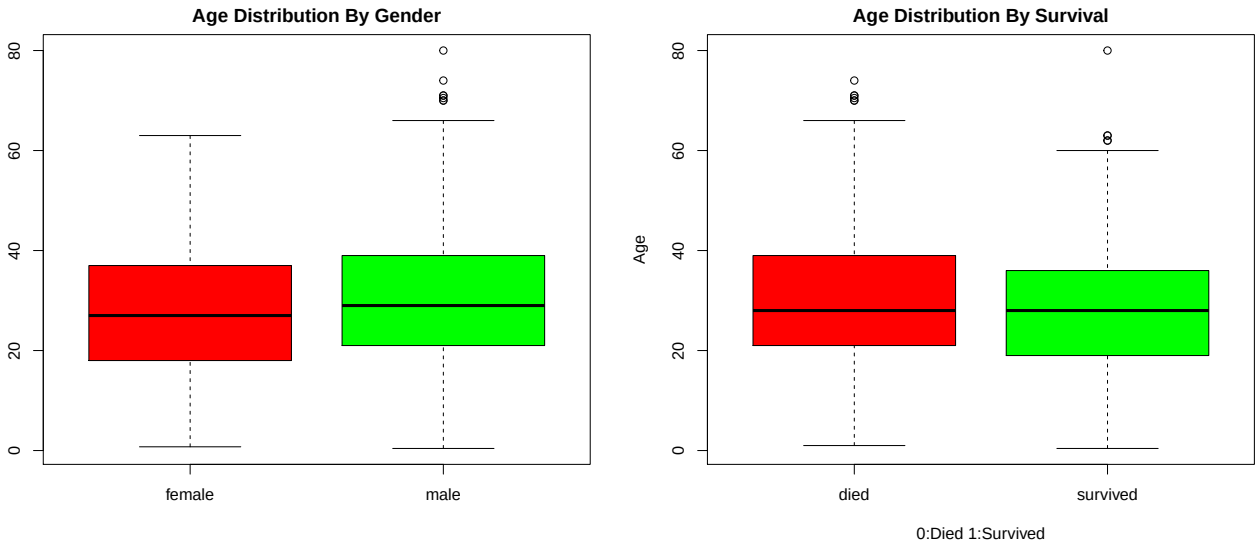
```
summary(titanic[titanic$Survived=="survived",]$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   19.00   28.00   28.34   36.00   80.00      52
```

---

## Age distribution by Survival and Sex

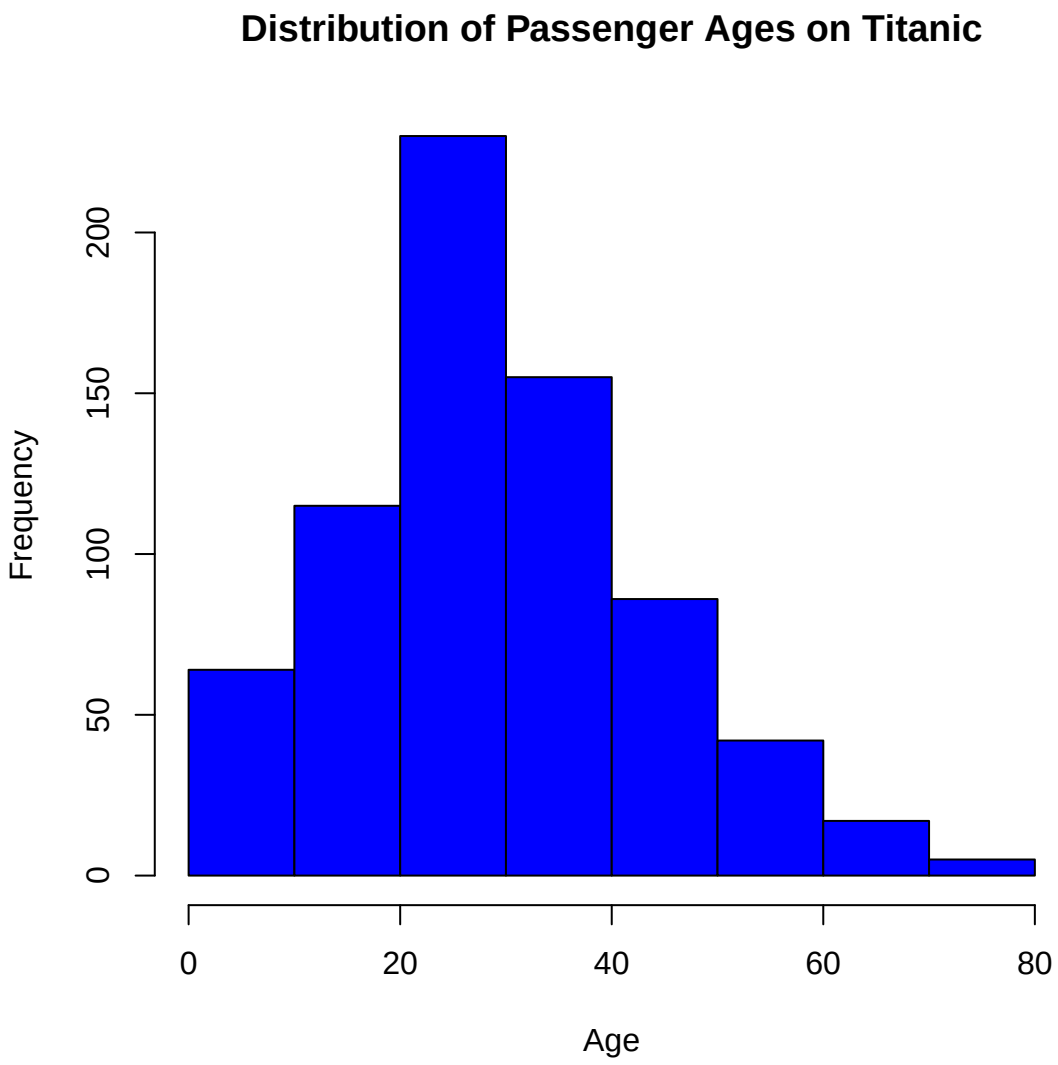
```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 2), oma = c(1, 1, 1, 1))
boxplot(titanic$Age~titanic$Sex, main="Age Distribution By Gender",col=c("red","green"))
boxplot(titanic$Age~titanic$Survived, main="Age Distribution By Survival",col=c("red","green"),
        xlab="0:Died 1:Survived",ylab="Age")
```




---

## Histogram of Age

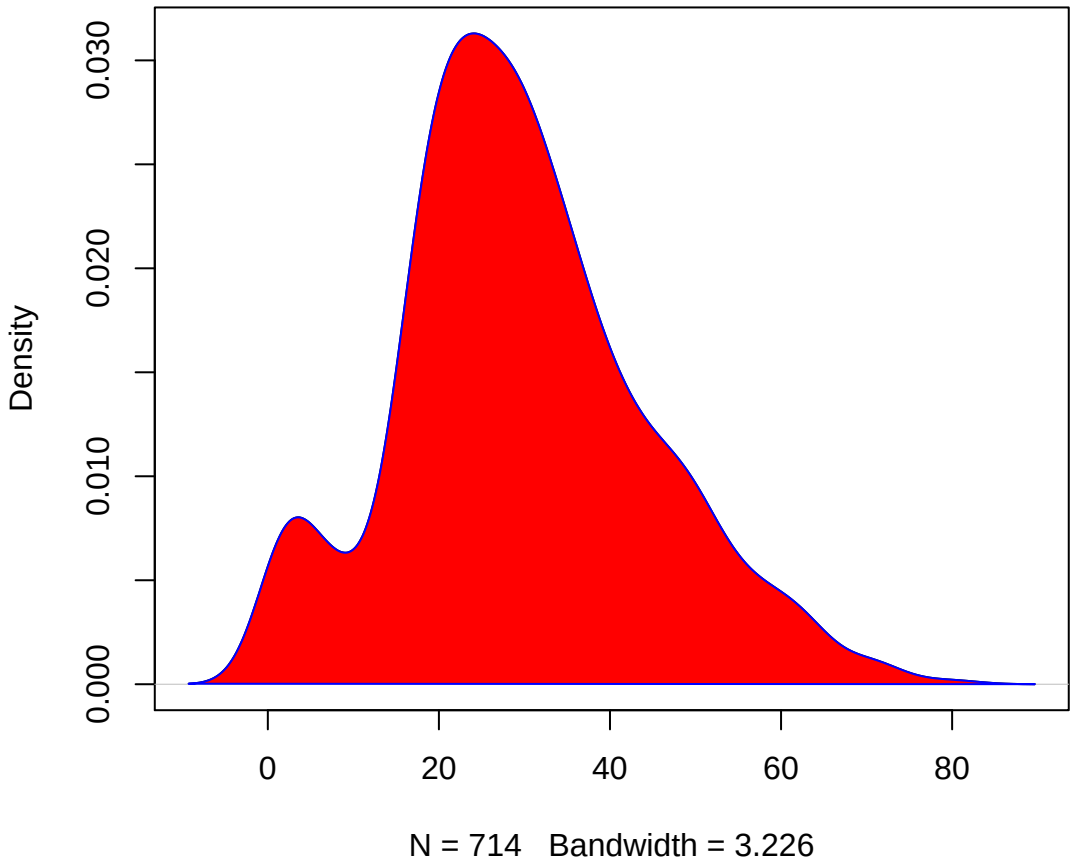
```
hist(Age, col="blue", xlab="Age", ylab="Frequency",
     main = "Distribution of Passenger Ages on Titanic")
```



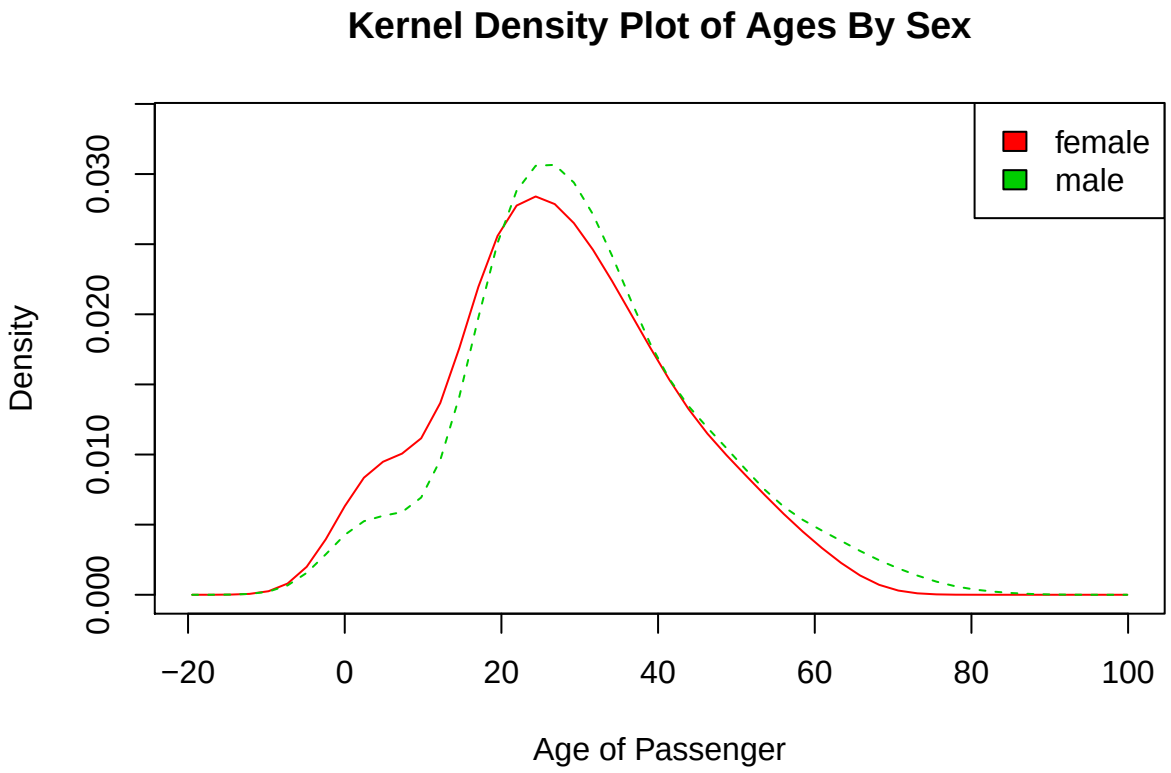

---

### Kernel density plot of age

```
d = density(na.omit(Age)) # density(Age) won't work, need to omit all NAs
plot(d, main = "kernel density of Ages of Titanic Passengers")
polygon(d, col="red", border="blue")
```

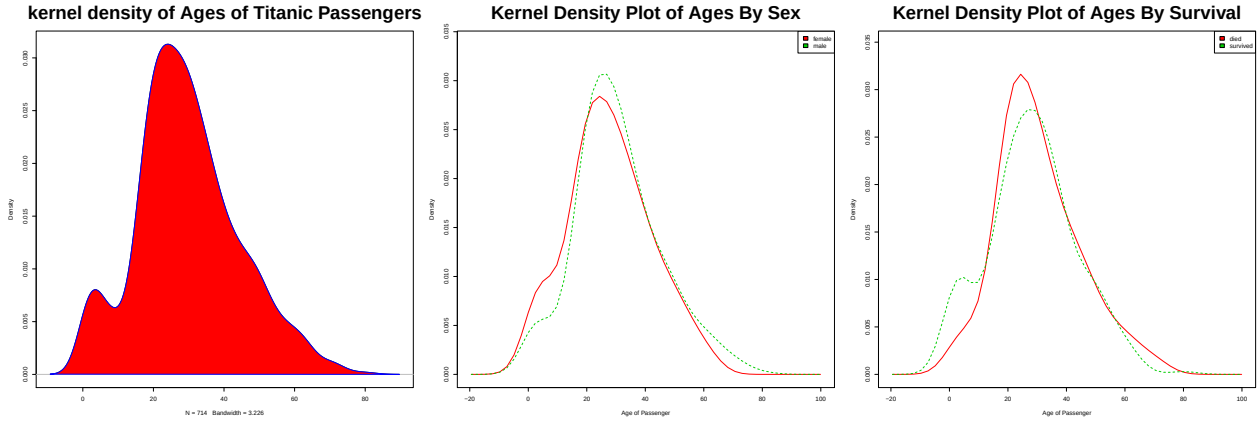

kernel density of Ages of Titanic Passengers
Density
0.000
0.010
0.020
0.030
0
20
40
60
80
N = 714 Bandwidth = 3.226

Comparison of density plots of Age with different Sex




---

## Did Age have an impact on survival?




---

## Create categorical groupings: Adult vs Child

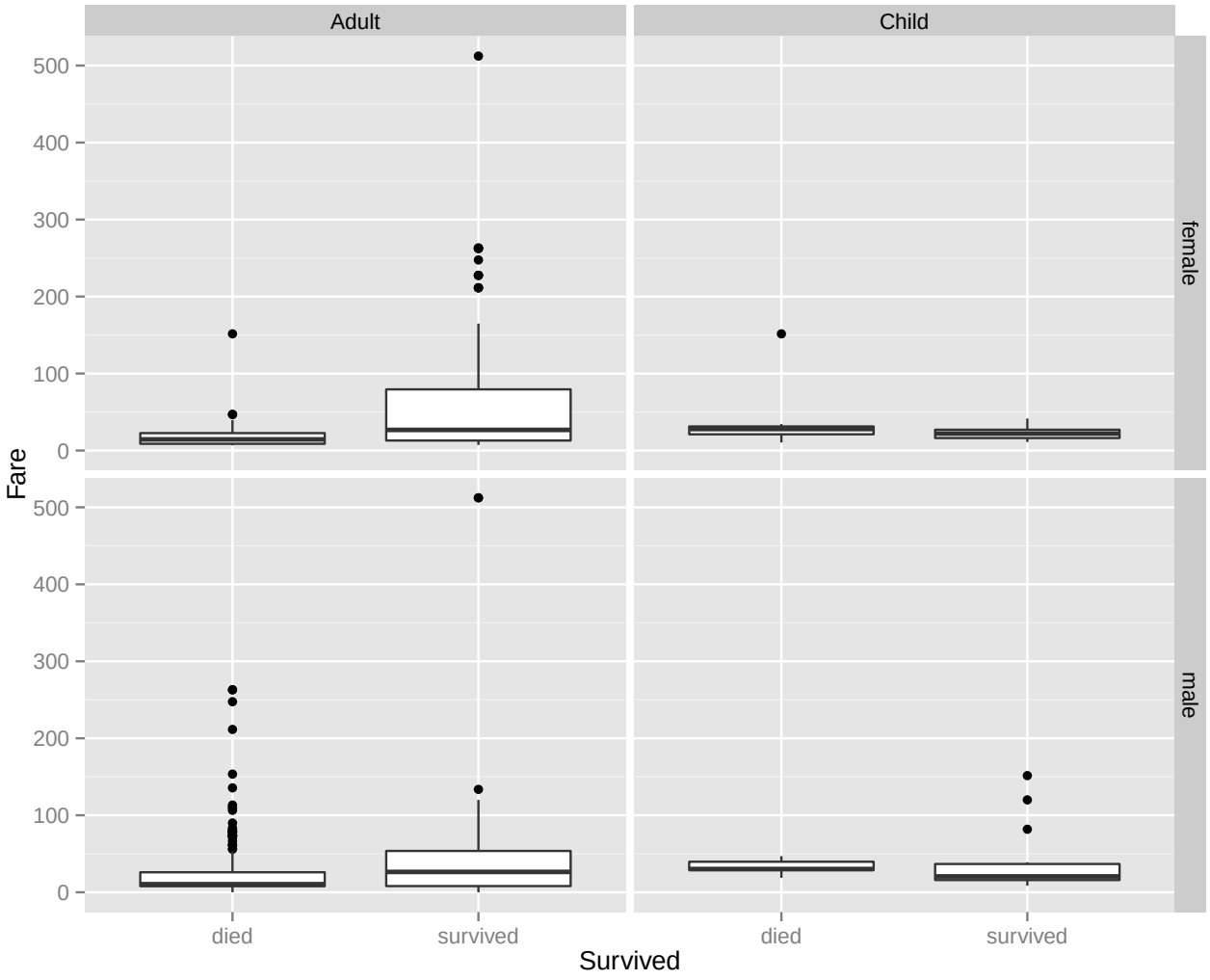
An example of **feature engineering**!

```
## Multi dimensional comparison
Child <- titanic$Age # Isolating age.
## Now we need to create categories: NA = Unknown, 1 = Child, 2 = Adult
## Every age below 13 (exclusive) is classified into age group 1
Child[Child<13] <- 1
## Every child 13 or above is classified into age group 2
Child[Child>=13] <- 2
```

```
# Use labels instead of 0's and 1's
Child[Child==1] <- "Child"
Child[Child==2] <- "Adult"
# Appends the new column to the titanic dataset
titanic_with_child_column <- cbind(titanic, Child)
# Removes rows where age is NA
titanic_with_child_column <- titanic_with_child_column[!is.na(titanic_with_child_column$Child),]
```

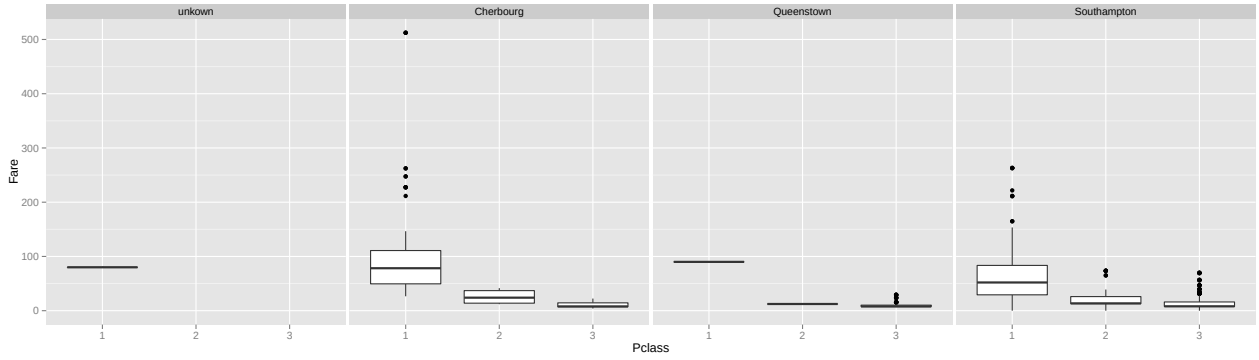
---

### Fare matters?




***

### How about fare, ship class, port embarkation?




---

# Diamond data

---

### Overview of the diamond data

```
data(diamonds) # loading diamonds data set
head(diamonds, 16) # first few rows of diamond data set
```

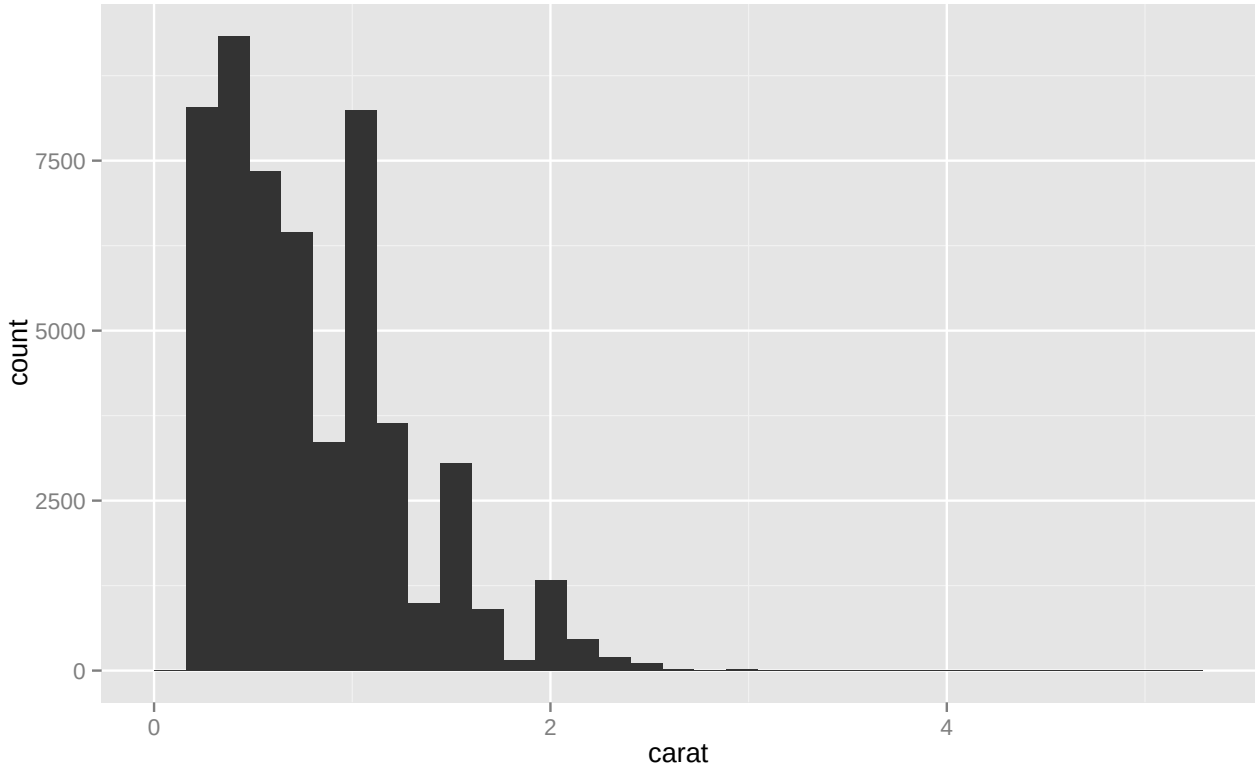
```
##    carat       cut color clarity depth table price    x    y    z
## 1   0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2   0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3   0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4   0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5   0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6   0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
## 7   0.24 Very Good     I    VVS1  62.3    57   336 3.95 3.98 2.47
## 8   0.26 Very Good     H     SI1  61.9    55   337 4.07 4.11 2.53
## 9   0.22      Fair     E     VS2  65.1    61   337 3.87 3.78 2.49
## 10  0.23 Very Good     H     VS1  59.4    61   338 4.00 4.05 2.39
## 11  0.30      Good     J     SI1  64.0    55   339 4.25 4.28 2.73
## 12  0.23     Ideal     J     VS1  62.8    56   340 3.93 3.90 2.46
## 13  0.22   Premium     F     SI1  60.4    61   342 3.88 3.84 2.33
## 14  0.31     Ideal     J     SI2  62.2    54   344 4.35 4.37 2.71
## 15  0.20   Premium     E     SI2  60.2    62   345 3.79 3.75 2.27
## 16  0.32   Premium     E      I1  60.9    58   345 4.38 4.42 2.68
```

---

### Histogram of carat

```
library(ggplot2)
ggplot(data=diamonds) + geom_histogram(aes(x=carat))
```

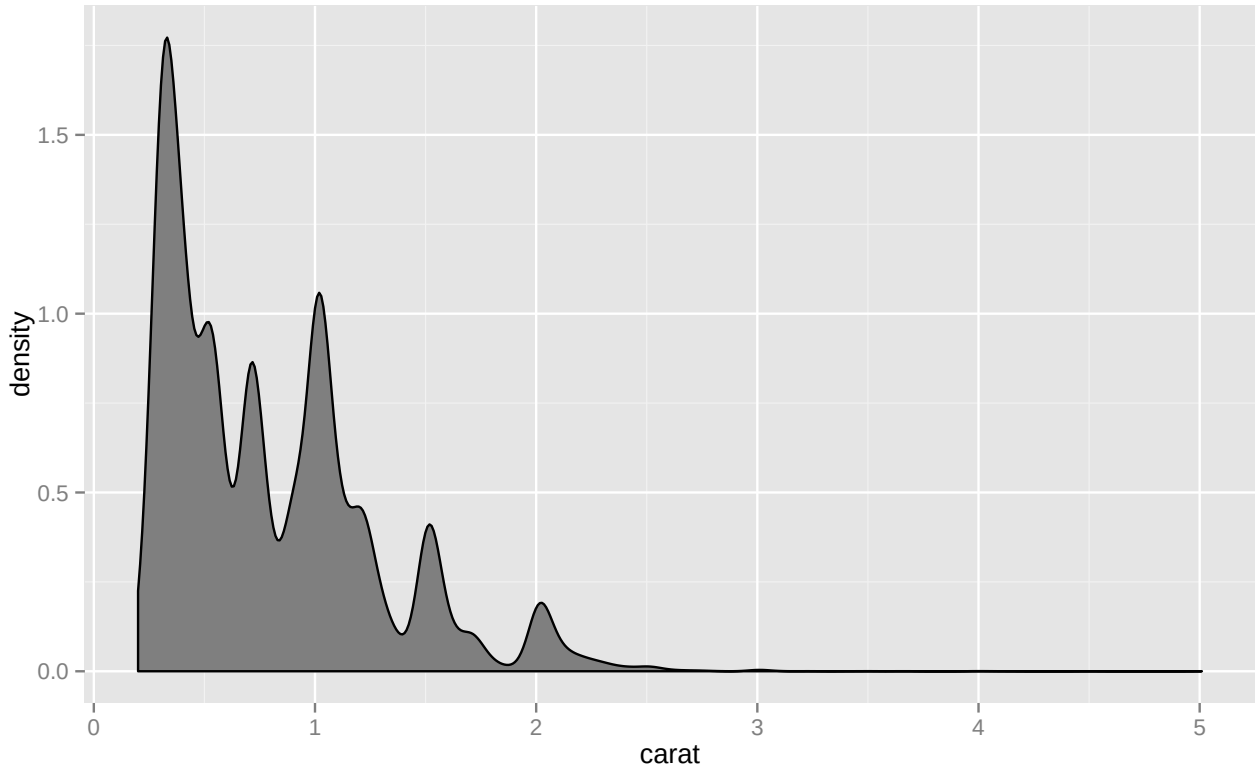
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




---

### Density plot of carat

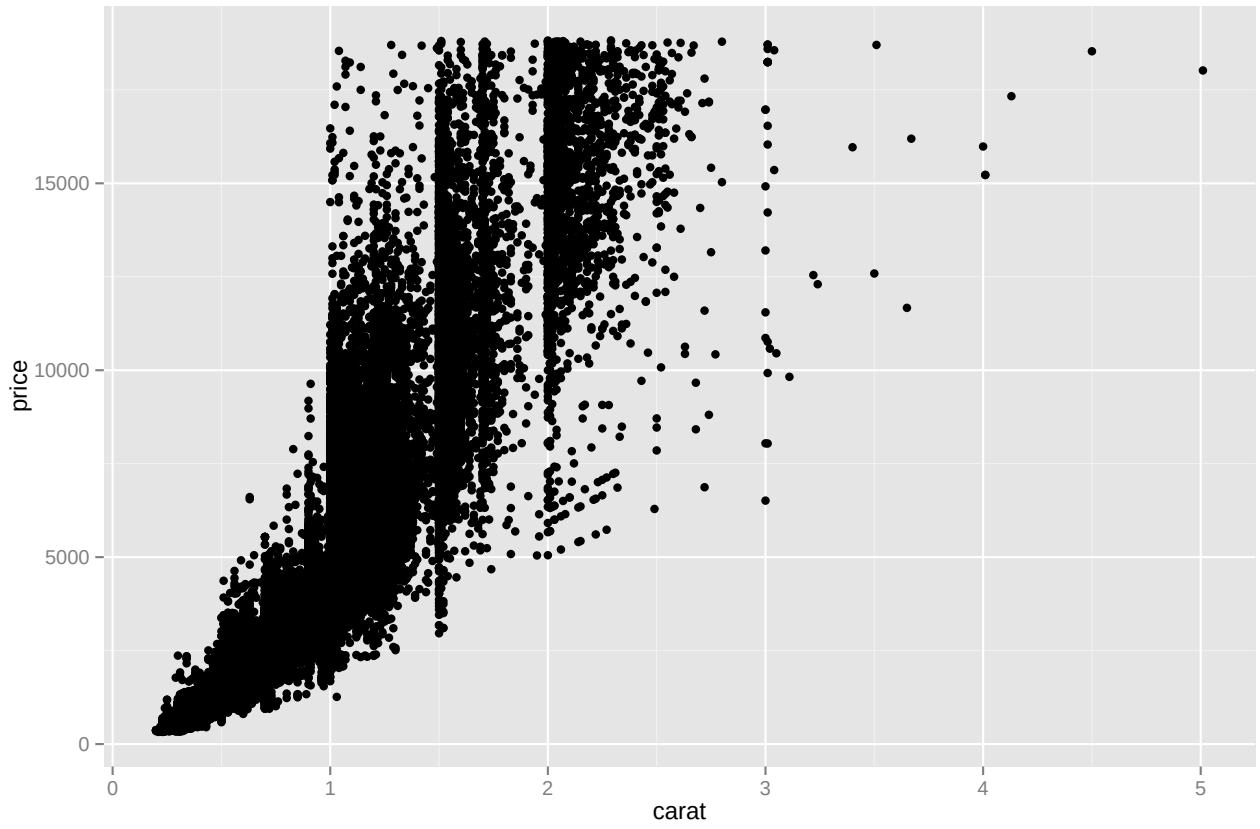
```
ggplot(data=diamonds) +
geom_density(aes(x=carat),fill="gray50")
```




---

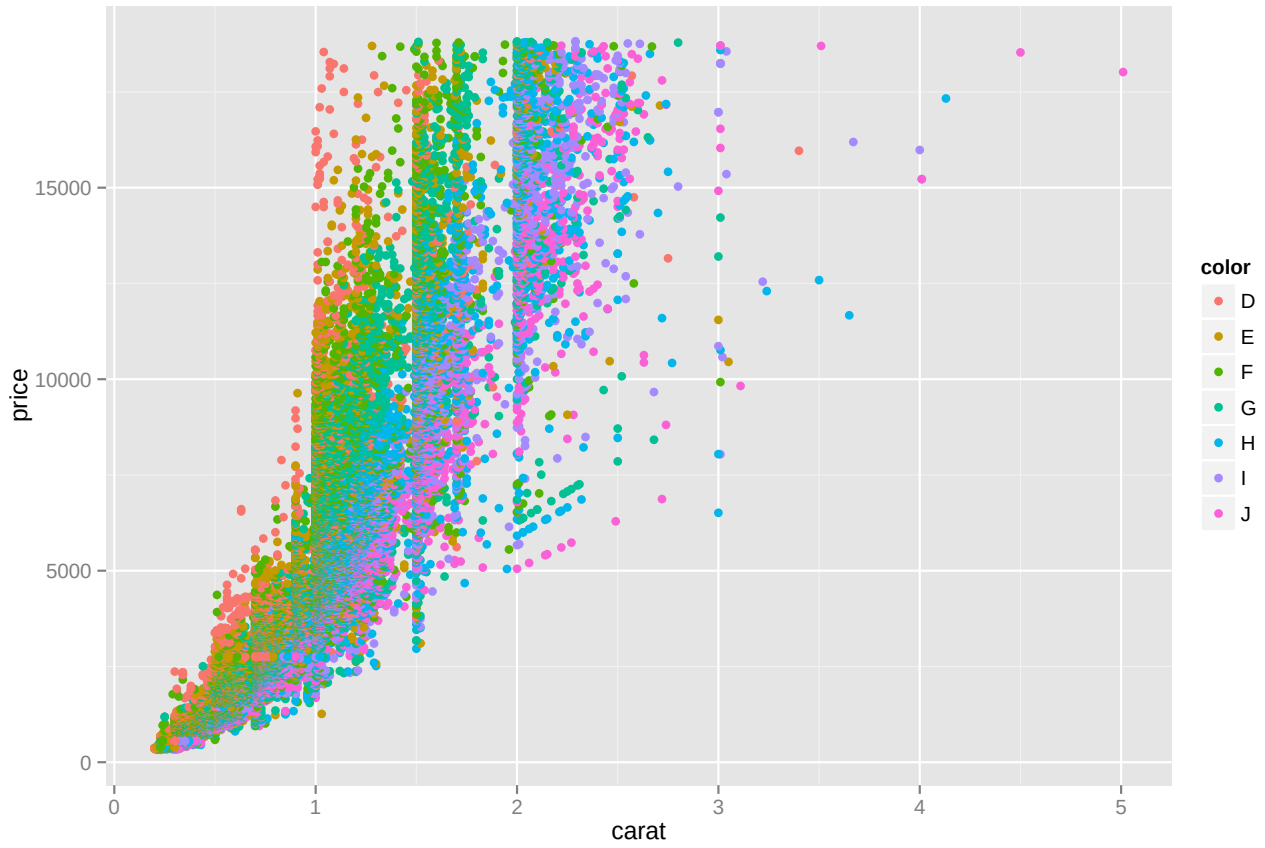
### Scatter plots (carat vs. price)

```
ggplot(diamonds, aes(x=carat,y=price)) + geom_point()
```



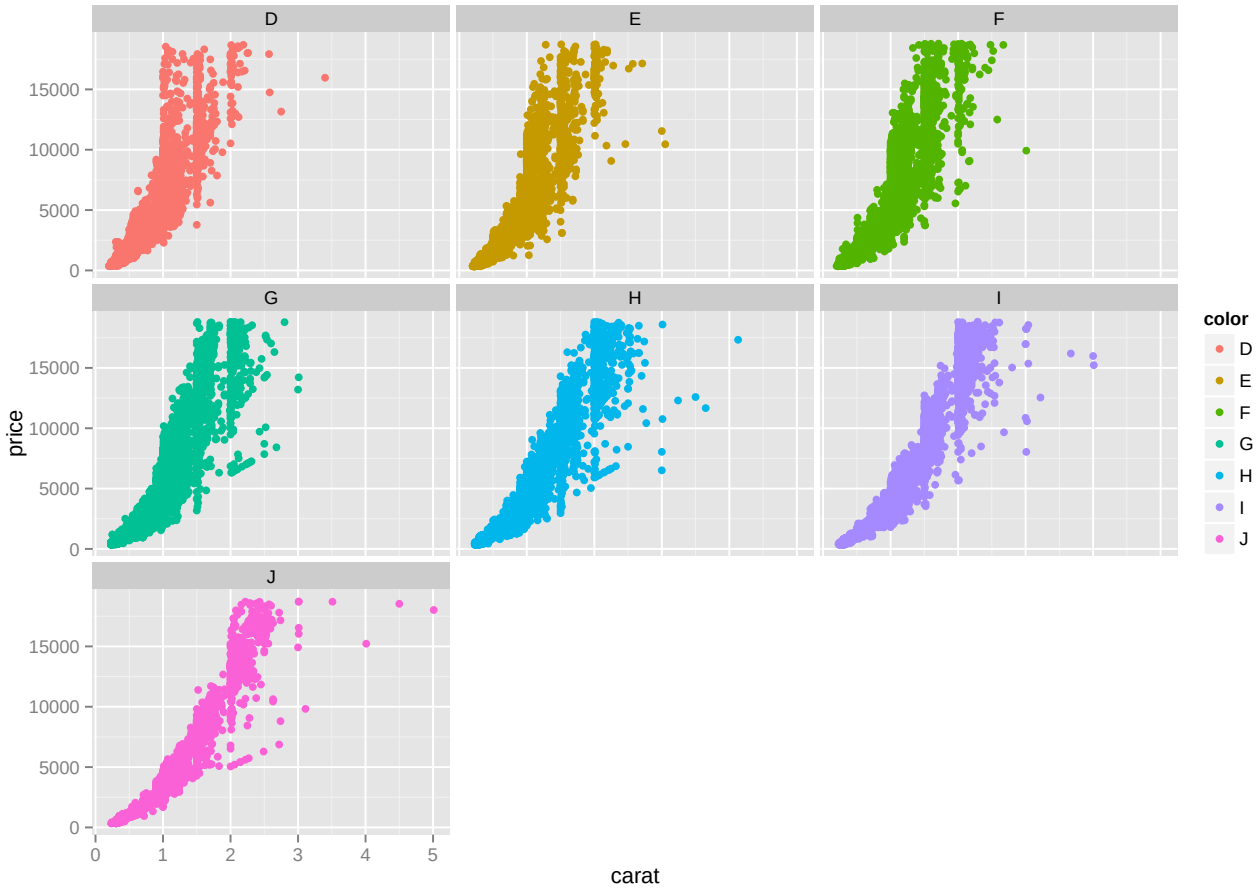

## Carat with colors

```
g = ggplot(diamonds, aes(x=carat, y=price)) # saving first layer as variable
g + geom_point(aes(color=color)) # rendering first layer and adding another layer
```



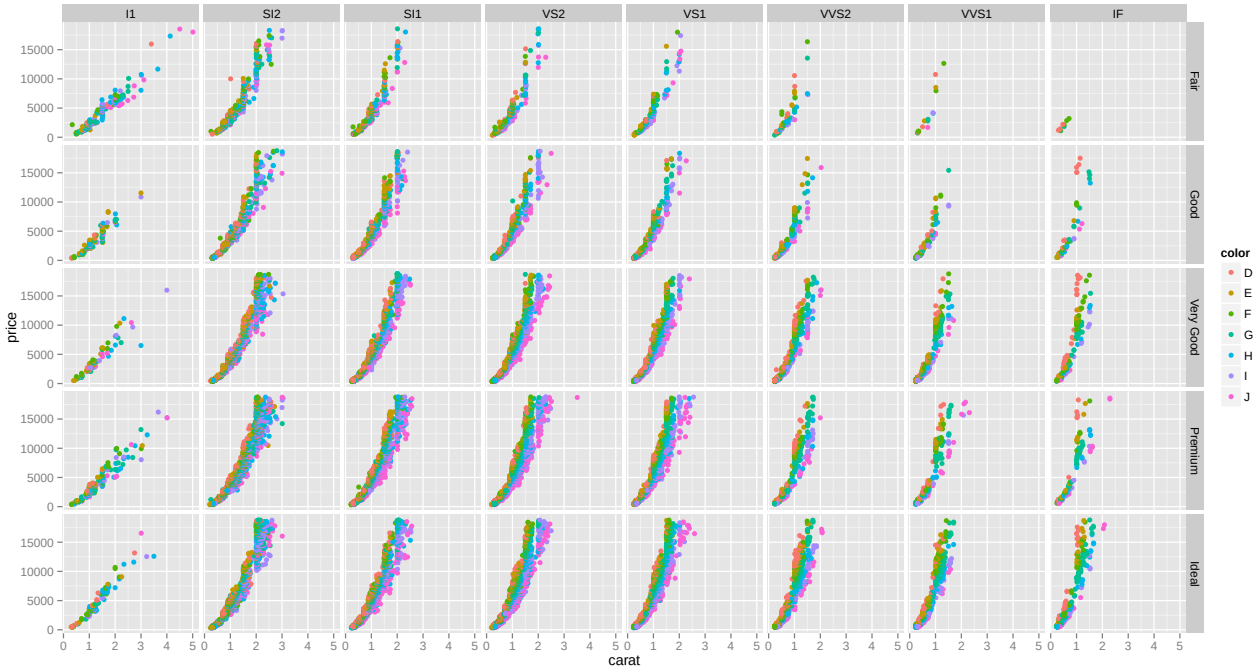

## Carat with colors (more details)

```
g + geom_point(aes(color=color)) + facet_wrap(~color)
```




---

## Let's consider cut and clarity



---

### Your trun!

What is your knowledge of diamond's price after exploring this data?