Descriptive Statistics (Categorical Data)

Sumanta Basu

Announcements

First “Exploration” will be posted on blackboard this Saturday (due next Saturday)

HW1 submission: pdf required (If you face issues with MikTex: knit to html or word, then convert to pdf)

Reading: Textbook Sections 1.2, 1.7

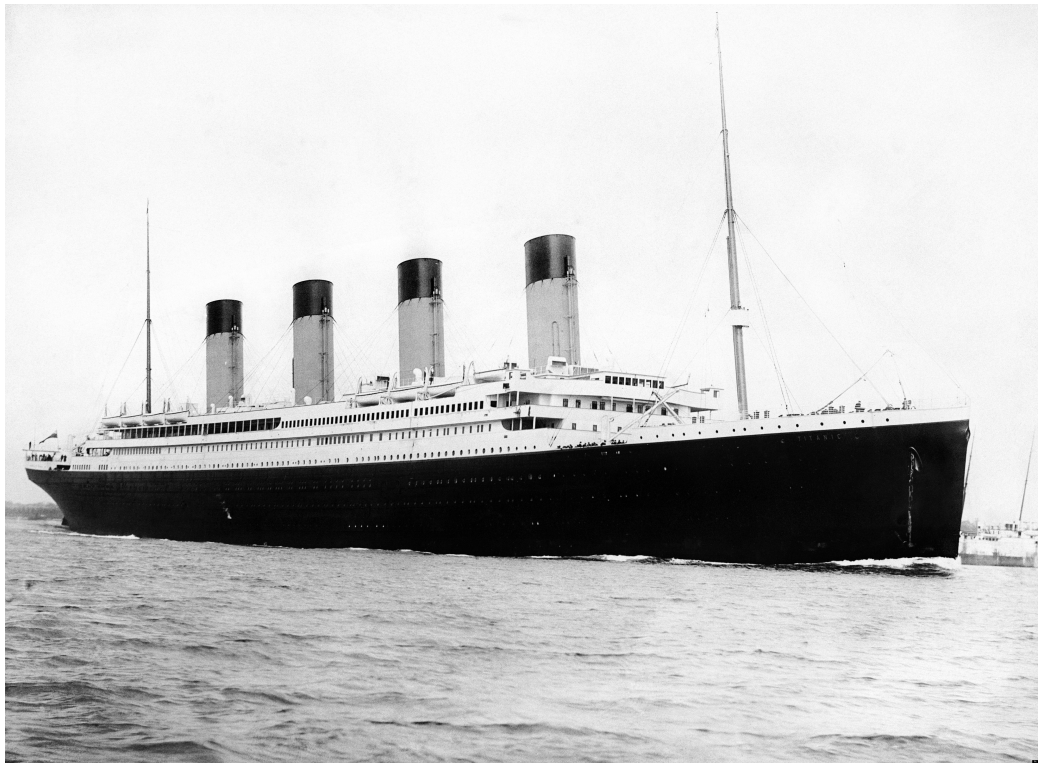
Recommended Exercise: 1.5, 1.7, 1.15, 1.37, 1.65, 1.66, 1.67, 1.68

Data Sets, Observations and Variables in R

The Titanic

From http://en.wikipedia.org/wiki/Titanic_(1997_film).

The Titanic (1911-1912)



From http://en.wikipedia.org/wiki/RMS_Titanic

The Titanic

Goal: describe survival patterns for the passengers

The data: - ~1300 passengers

- variables:
- age (in years)
- sex (male/female)
- class (first, second, third)
- survived (yes/no)

The raw data

Open `titanic.txt`.

Loading into R

```
dat <- read.table("titanic.txt", header=TRUE)
```

Type of variable?

Titanic data:

- age
- sex
- class
- survived

Data Sets in R

We will import the titanic data from the file into R

```
dat <- read.table("titanic.txt", header=TRUE)
head(dat, n=3)
```

```
##                                 Name PClass Age    Sex Survived
## 1        Allen, Miss Elisabeth Walton    1st  29 female        1
## 2         Allison, Miss Helen Loraine    1st   2 female        0
## 3 Allison, Mr Hudson Joshua Creighton    1st  30   male        0
```

Each row is an *observation*, each column is a *variable*

```
names(dat) # Names of variables in the data set
```

```
## [1] "Name"     "PClass"   "Age"      "Sex"      "Survived"
```

```
nrow(dat) # Number of observations in the data set
```

```
## [1] 1313
```

Basic Data Management

```
head(dat, n=2)
```

```
##                             Name PClass Age    Sex Survived
## 1 Allen, Miss Elisabeth Walton    1st  29 female        1
## 2  Allison, Miss Helen Loraine    1st   2 female        0
```

```
dat$Age[2] # Find the age of the second passenger
```

```
## [1] 2
```

```
dat[2,3] # Another way to reference her age
```

```
## [1] 2
```

```
dat$child <- (dat$Age < 5) # Add a new variable (children < 5 years)
head(dat, n=2)
```

```
##                             Name PClass Age    Sex Survived child
## 1 Allen, Miss Elisabeth Walton    1st  29 female        1 FALSE
## 2  Allison, Miss Helen Loraine    1st   2 female        0  TRUE
```

Basic Data Management

```
sum(dat$PClass == "1st") # count number of passengers in 1st class
```

```
## [1] 322
```

```
head(which(dat$Survived == 1)) # find which passengers survived (only show first 6)
```

```
## [1]  1  5  6  7  9 12
```

```
head(dat$Name[which(dat$Survived == 1)], n=3) # print their names (only show first 3)
```

```
## [1] Allen, Miss Elisabeth Walton  Allison, Master Hudson Trevor
## [3] Anderson, Mr Harry
## 1310 Levels: Abbing, Mr Anthony ... Zimmerman, Leo
```

```
survivors <- dat[which(dat$Survived == 1),] # create a data set of survivors
head(survivors, n=2) # see how it looks
```

```
##                             Name PClass   Age    Sex Survived child
## 1  Allen, Miss Elisabeth Walton    1st 29.00 female        1 FALSE
## 5 Allison, Master Hudson Trevor    1st  0.92   male        1  TRUE
```

Basic Data Management

Other useful functions to explore your data set

- `str`: describes structure of the data
- `summary`: summarizes the data

```
summary(dat, maxsum = 3) # Try `help(summary)` to find out more on maxsum
```

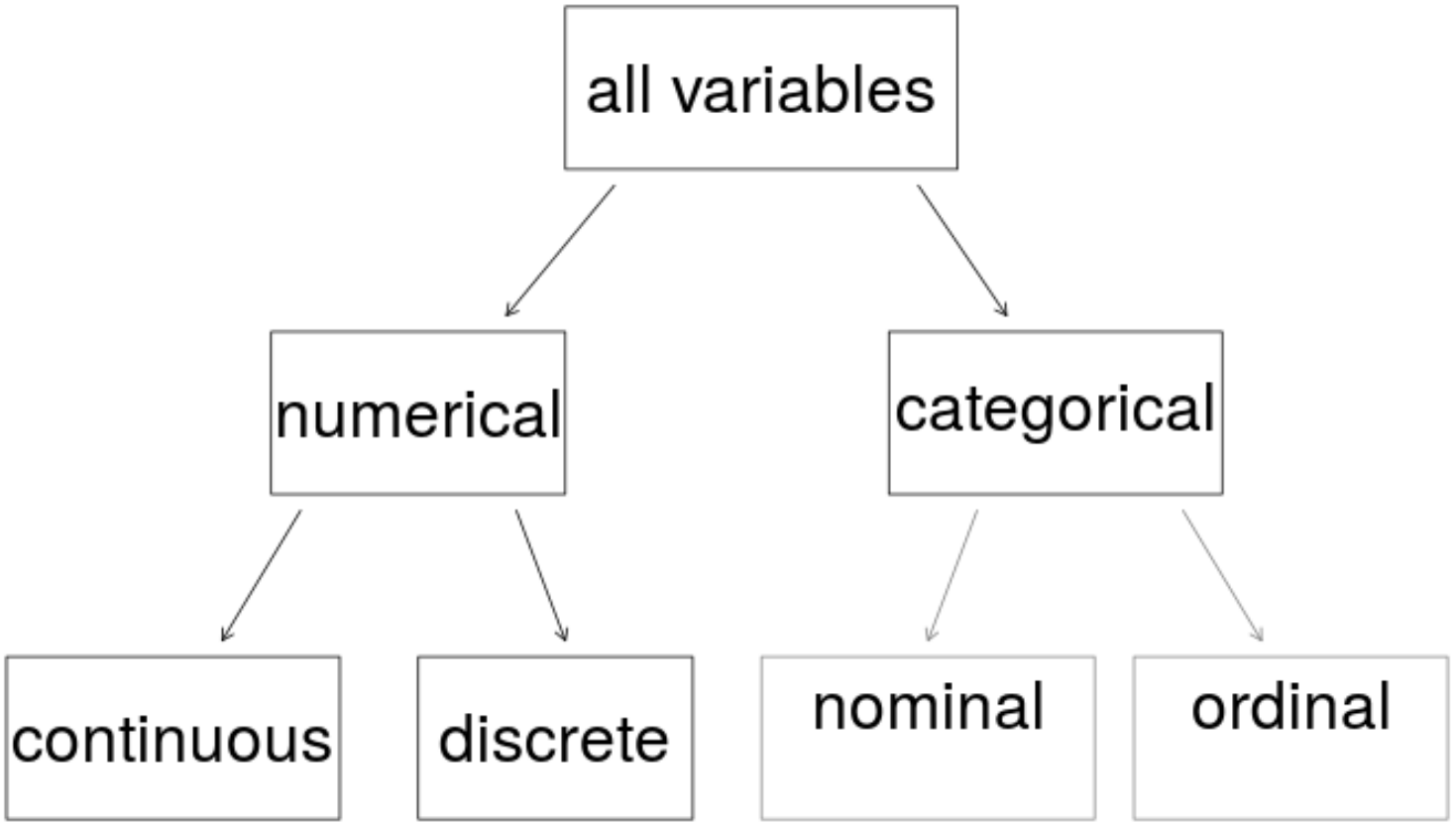
```
##                         Name      PClass         Age           Sex
##  Carlsson, Mr Frans Olof:   2   1st:322   Min.   : 0.17   female:462
##  Connolly, Miss Kate    :   2   2nd:280   1st Qu.:21.00   male  :851
##  (Other)                :1309   3rd:711   Median :28.00
##                                           Mean   :30.40
##                                           3rd Qu.:39.00
##                                           Max.   :71.00
##                                           NA's   :557
##     Survived        child
##  Min.   :0.0000   Mode :logical
##  1st Qu.:0.0000   FALSE:723
##  Median :0.0000   TRUE :33
##  Mean   :0.3427   NA's :557
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

Basic Data Management

> Write down a few simple questions that you want to answer using the titanic data. After the first lab, see if you can find some R commands to answer this. Feel free to check online (stackoverflow is great!) or post on piazza. Use `help(`<*function_name*>`)` as needed.

Types of Variables

Categorical and Numerical



Categorical Variables

Variables that only take a fixed (usually finite) number of possible values (`level`)

- Race, Gender, College Standing, Colors

Two subtypes:

- Ordinal (levels have natural ordering): College Standing (four levels Freshman, Sophomore, Junior, Senior)
- Nominal (no natural ordering among levels): Color, Race

> Use `factor` function in R to store and manage categorical variables

Categorical Variables in Titanic Data

```
str(dat)
```

```
## 'data.frame':    1313 obs. of  6 variables:
##  $ Name    : Factor w/ 1310 levels "Abbing, Mr Anthony",..: 22 25 26 27 24 31 45 46 50 54 ...
##  $ PClass  : Factor w/ 3 levels "1st","2nd","3rd": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Age     : num  29 2 30 25 0.92 47 63 39 58 71 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 1 1 2 1 2 2 1 2 1 2 ...
##  $ Survived: int  1 0 0 0 1 1 1 0 1 0 ...
##  $ child   : logi  FALSE TRUE FALSE FALSE TRUE FALSE ...
```

```
levels(dat$PClass)
```

```
## [1] "1st" "2nd" "3rd"
```

R uses type `factor` for categorical variables, `int` for integers, `num` for real numbers. Type `logical` stores Booleans (`TRUE` or `FALSE` – can be used as 1 and 0 within R)

Logical Variables in R

```
!TRUE # ! operator changes TRUE to FALSE and vice versa
```

```
## [1] FALSE
```

```
TRUE + TRUE
```

```
## [1] 2
```

```
# You can use them to create new variables
head(dat$child)
```

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

```
dat$adult <- (!dat$child)
head(dat$adult)
```

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

Logical Variables in R

… or count number of observations satisfying some conditions

```
temp <- (dat$Survived == 1) # create a logical variable of survivor status
head(temp) # see how it looks for first few passengers
```

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

```
sum(temp) # count number of `TRUE` among them
```

```
## [1] 450
```

```
sum(dat$Sex == 'female' & dat$Survived == 1)
```

```
## [1] 308
```

Changing variable types in R

```
levels(dat$Sex)
```

```
## [1] "female" "male"
```

```
levels(dat$Sex) = c("Female", "Male")
survivor_index = (dat$Survived==1)
head(survivor_index)
```

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

```
dat$Survived[survivor_index] = "Yes"
dat$Survived[!survivor_index] = "No"
dat$Survived = as.factor(dat$Survived)
head(dat, n=2)
```

```
##                             Name PClass Age    Sex Survived child adult
## 1 Allen, Miss Elisabeth Walton    1st  29 Female      Yes FALSE  TRUE
## 2  Allison, Miss Helen Loraine    1st   2 Female       No  TRUE FALSE
```

How to summarize categorical data?

> **frequency table** (for one categorical variable), **contingency tables** (two or more categorical variables)

- Stores counts of units in your data with a specified level
- Sometimes it helps to converts counts into proportions

R function: `table`

One categorical variable at a time

```
table(dat$Sex)
```

```
##
## Female   Male
##    462    851
```

```
table(dat$Survived)
```

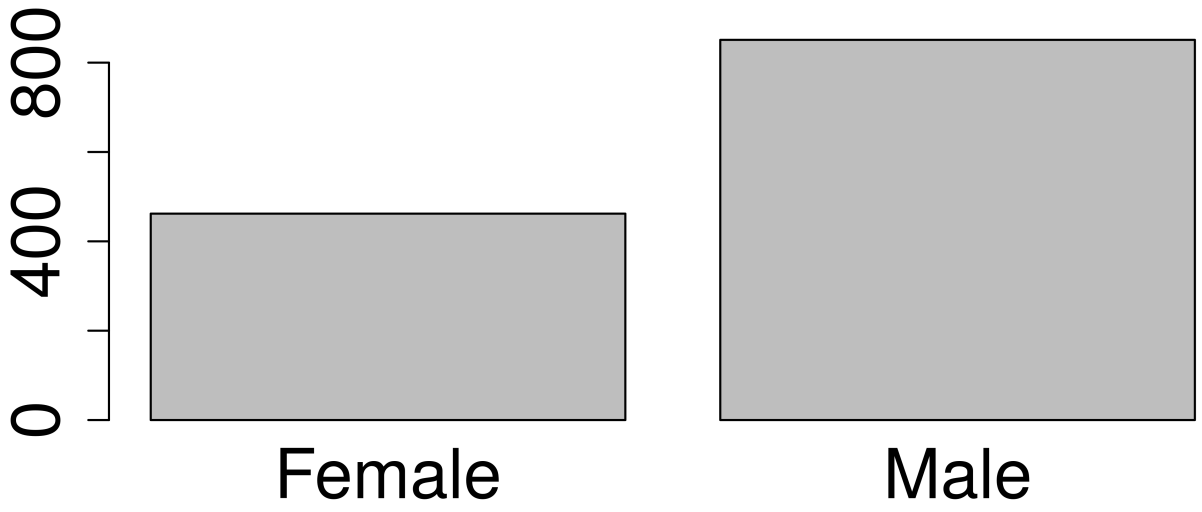
```
##
##  No Yes
## 863 450
```

```
table(dat$PClass)
```

```
##
## 1st 2nd 3rd
## 322 280 711
```

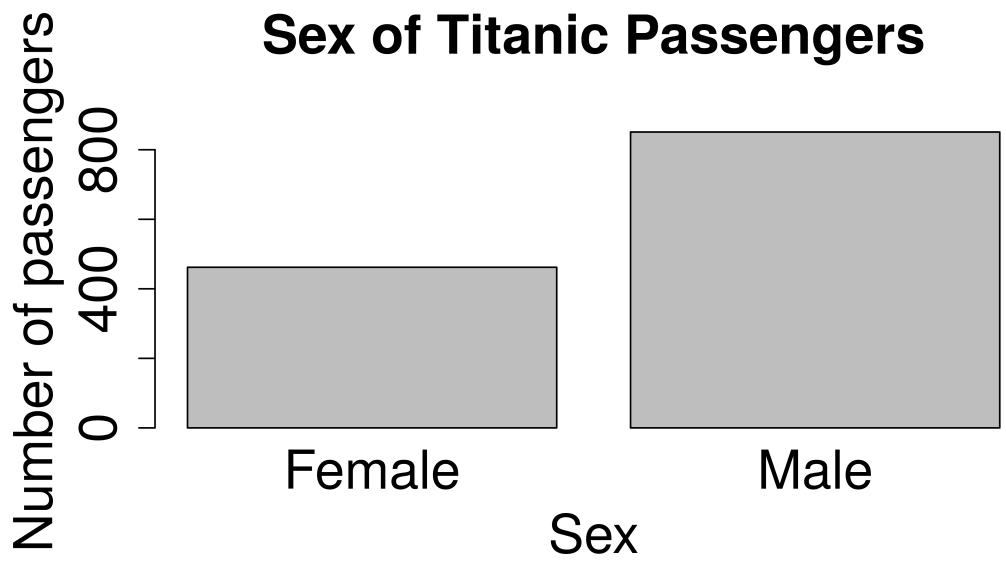
Graphical display of a categorical variable

```
counts = table(dat$Sex)
barplot(counts)
```



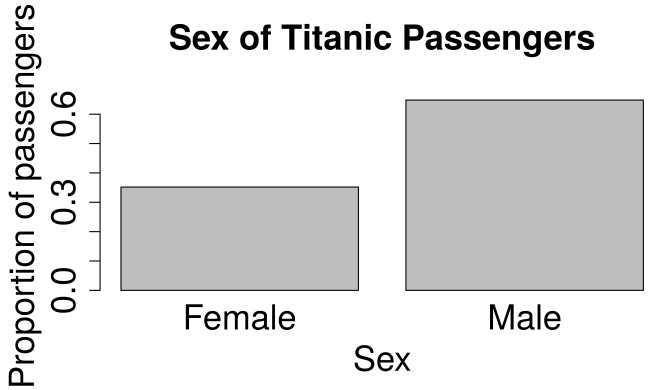
Graphical display of a categorical variable

```
barplot(counts, main="Sex of Titanic Passengers",
  ylab="Number of passengers", xlab="Sex")
```



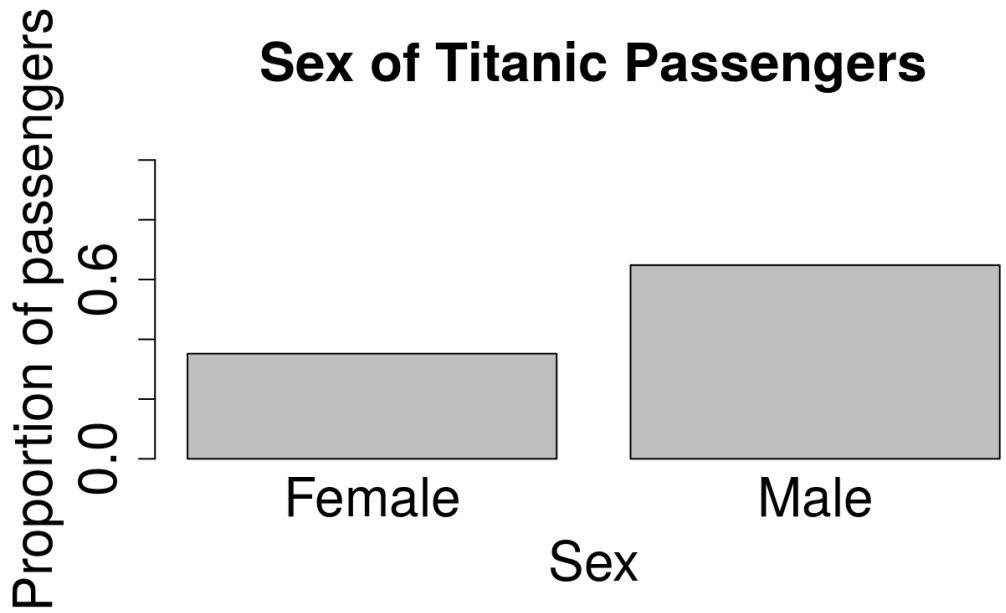
Graphical display of a categorical variable

```
counts = table(dat$Sex)
totnum = sum(counts)
barplot(counts / totnum,
    main = "Sex of Titanic Passengers",
    ylab = "Proportion of passengers",
    xlab = "Sex")
```



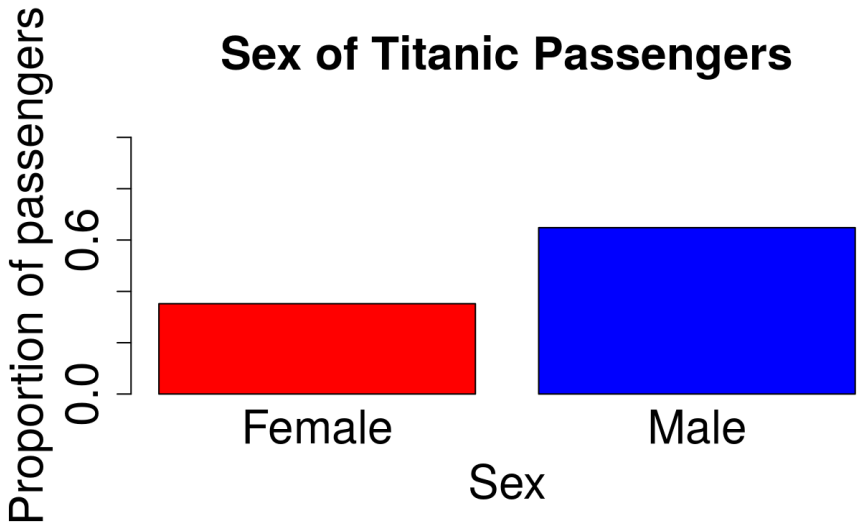
Graphical display of a categorical variable

```
barplot(counts / totnum,
 main="Sex of Titanic Passengers",
 ylab="Proportion of passengers",
 xlab="Sex", ylim=c(0,1))
```



Should we add color here?

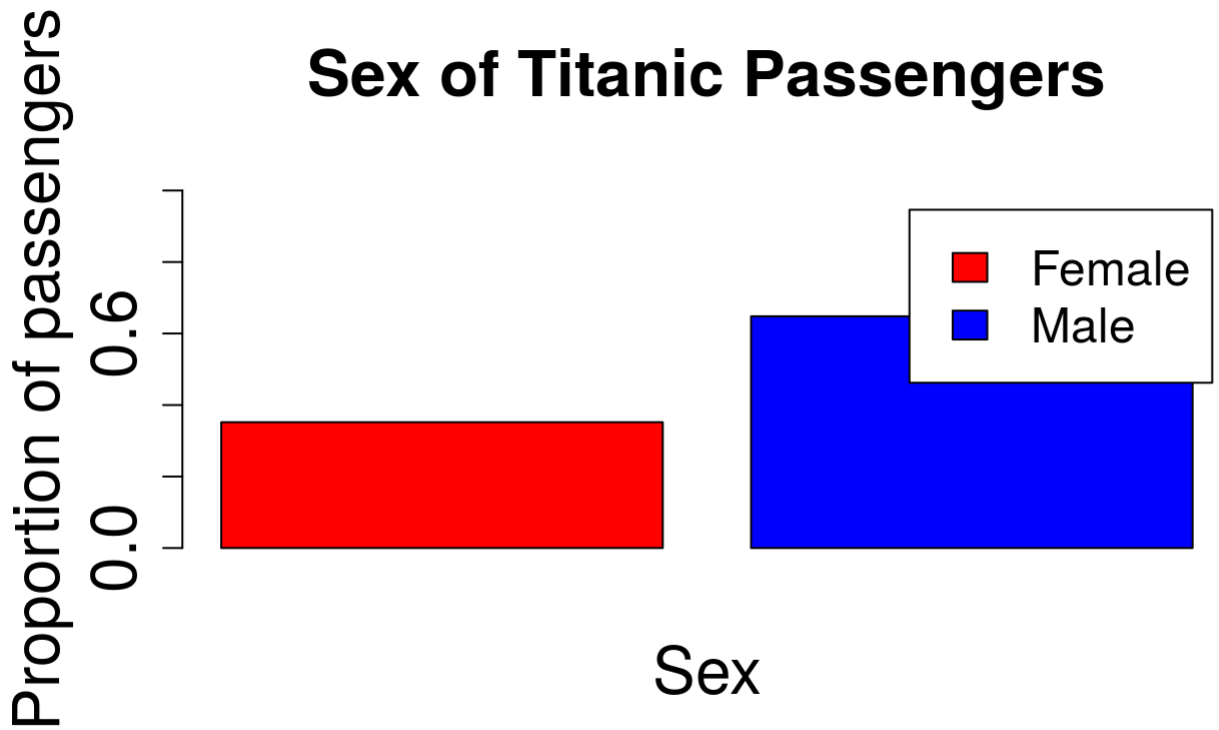
```
barplot(counts / totnum,
  main="Sex of Titanic Passengers",
  ylab="Proportion of passengers",
  xlab="Sex", ylim=c(0,1),
  col=c("red", "blue"))
```



And a legend?

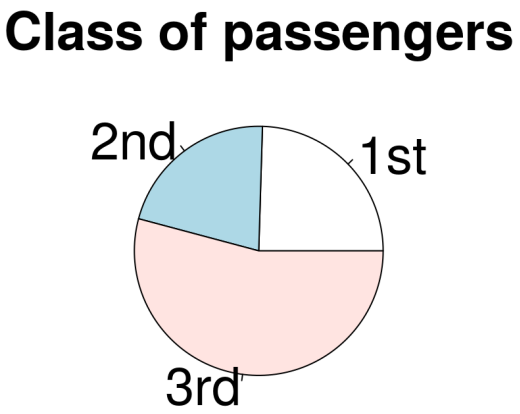
```
par(mar=2+c(5.1,4.1,4.1,2.1),
  cex.axis=2, cex.main=2, cex.names=2, cex.lab=2)
barplot(counts / totnum, main="Sex of Titanic Passengers",
  ylab="Proportion of passengers", xlab="Sex", ylim=c(0,1),
  args.legend=list(cex=1.5),legend.text=c("Female","Male"),
  col=c("red", "blue"))
```

How about this?



Which is better?

```
pie(nclass,
  main="Class of passengers")
```

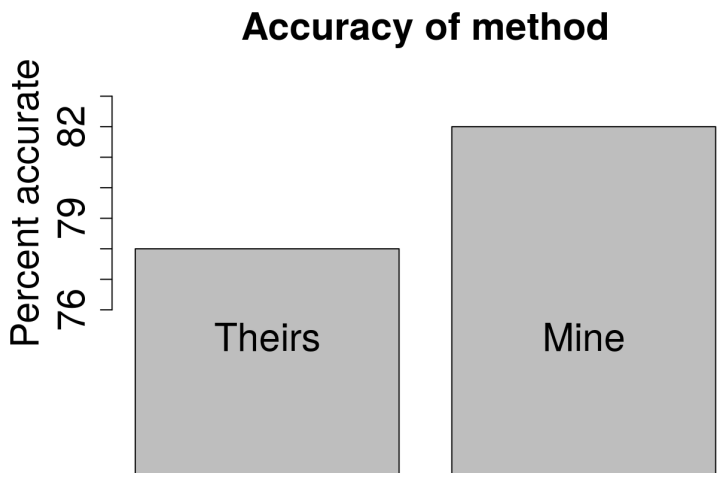


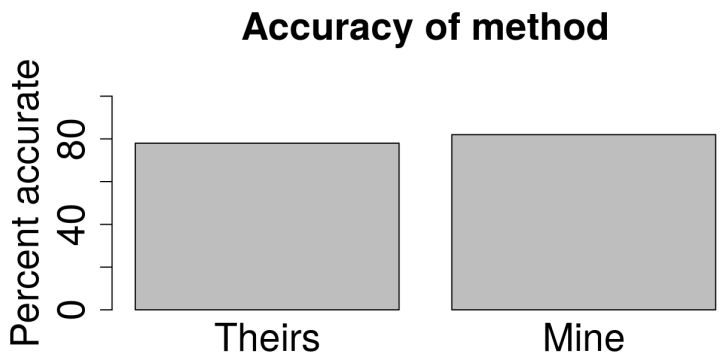

```
barplot(nclass/sum(nclass),
  main="Class of passengers")
```

Pie charts

- similarly sized wedges hard to compare
- angle of wedge affects perceived size [Maclean & Stacey 1971]

A note about bar plots





Moral: Remember that bar's length/area has meaning.

Summarize relationships between two categorical variables

Contingency table

- row proportions, column proportions

Customary to put explanatory variable in rows, response variable in columns

Check if proportions are similar across rows

Relationship b/w categorical variables

"Marginal" information: one-at-a-time

```
table(dat$Sex)
```

```
##
## Female   Male
##    462    851
```

```
table(dat$Survived)
```

```
##
##  No Yes
## 863 450
```

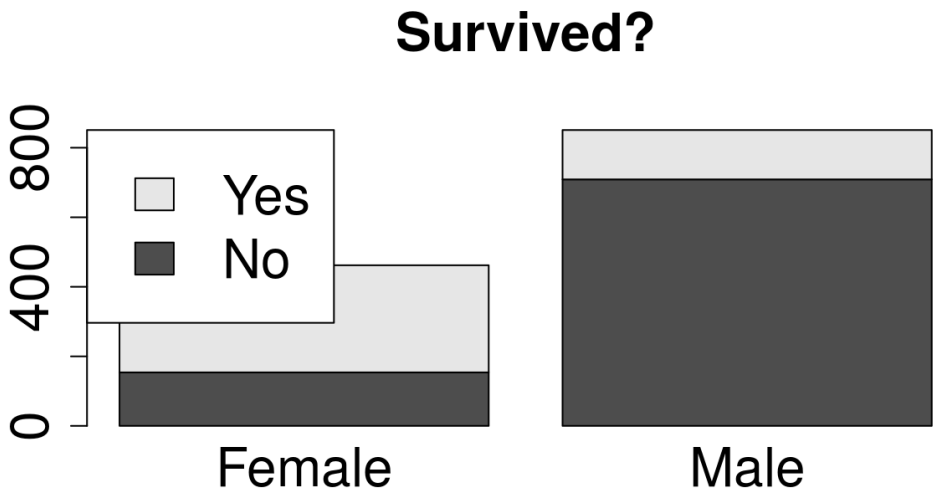
"Joint" information:

```
table(dat$Sex,dat$Survived)
```

```
##
##           No Yes
##   Female 154 308
##   Male   709 142
```

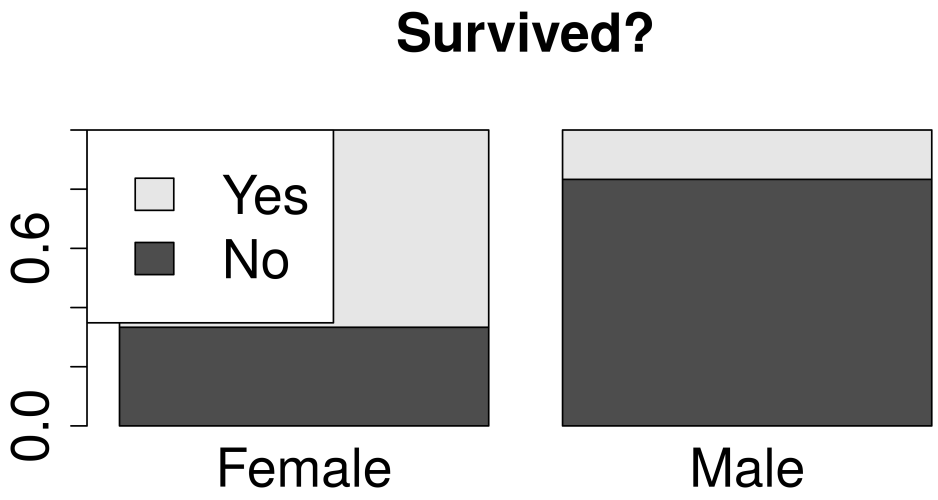
Graphical display of multiple categorical variables

```
barplot(table(dat$Survived,dat$Sex), main="Survived?",
  legend.text=c("No","Yes"))
```

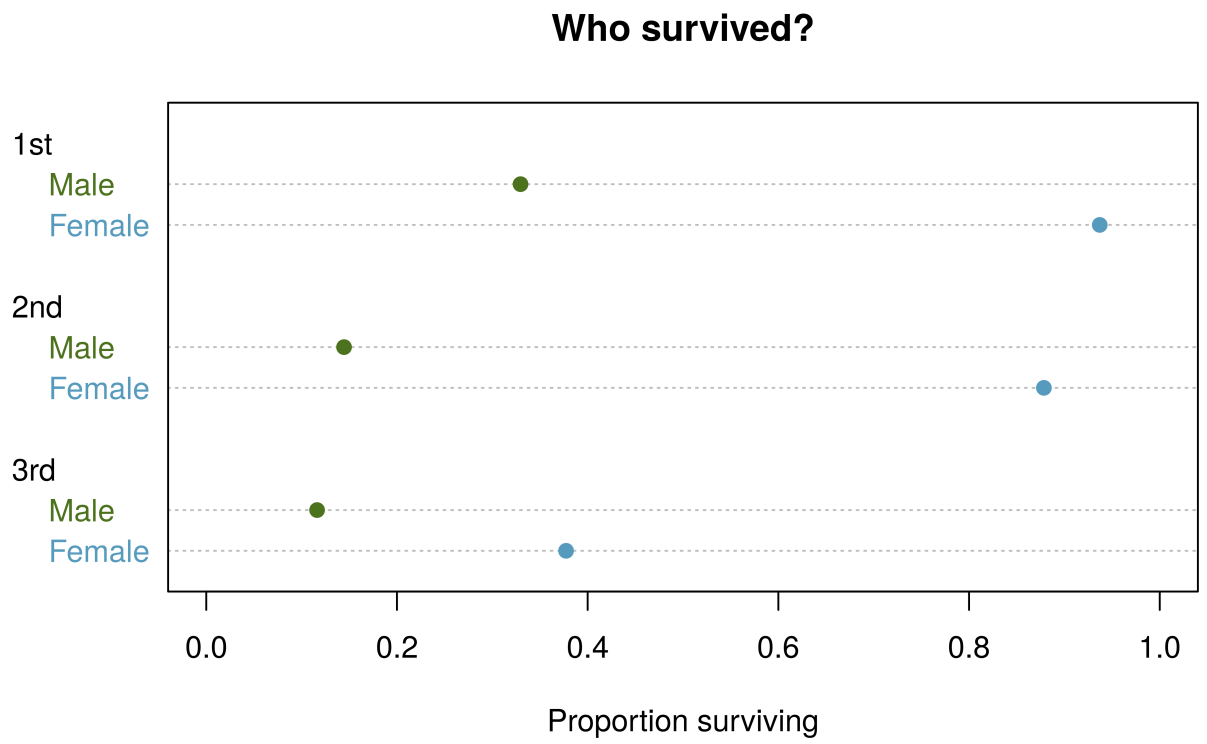


Graphical display of multiple categorical variables

```
pt=prop.table(table(dat$Survived,dat$Sex),2)
barplot(pt, main="Survived?", legend.text=c("No","Yes"))
```

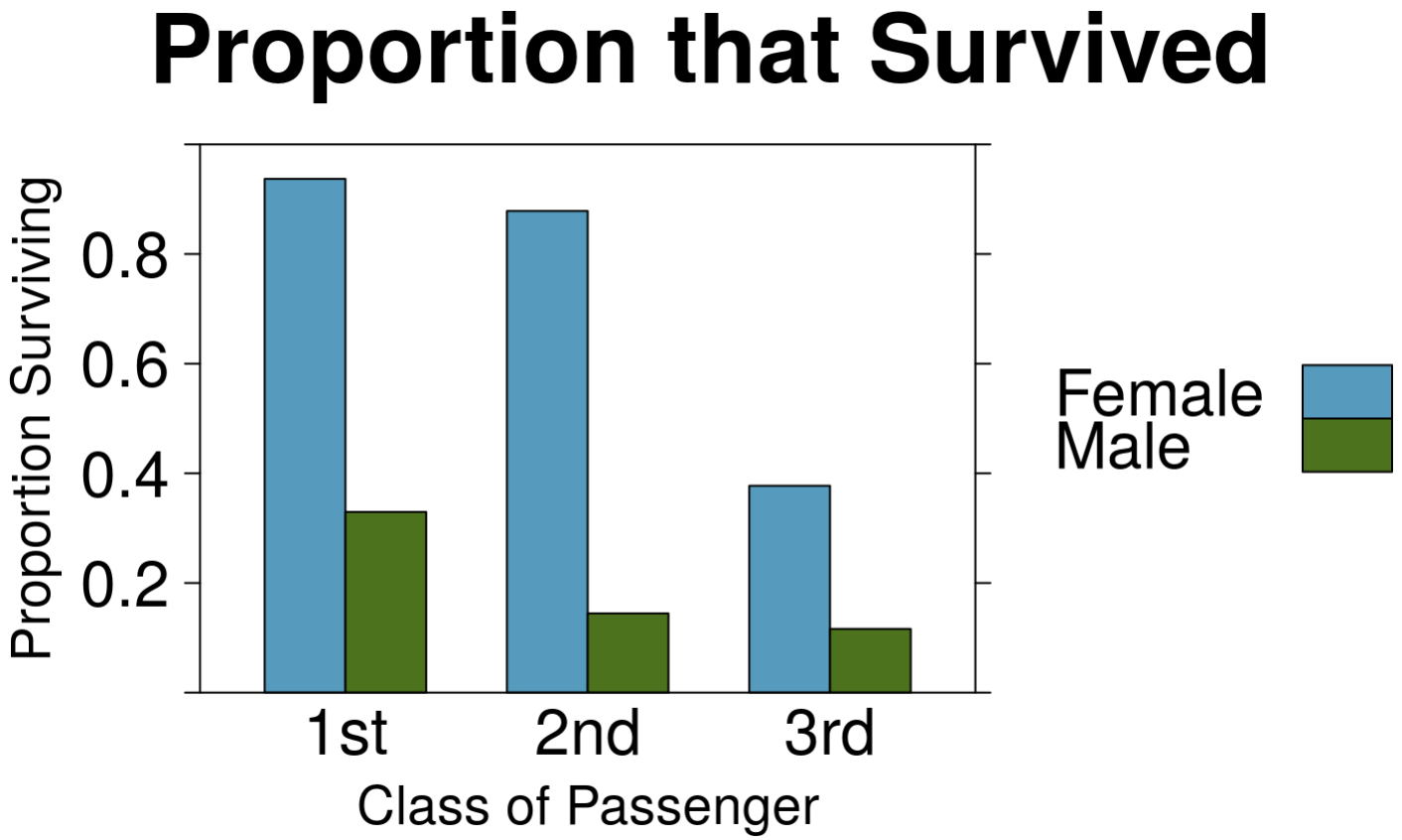


Dot Plots



See appendix for the code

Side-by-side barplot



See appendix for the code

Appendix: More on Descriptive Statistics

Be careful when working with data!

It’s like defensive driving…

Image from
http://www.nyautoschool.com/

Titanic data

```
summary(dat)
```

-
```
> summary(dat)
                          Name      PClass         Age           Sex         Survived
 Carlsson, Mr Frans Olof     :   2   1st:322   Min.   : 0.17   female:462   Min.   :0.0000
 Connolly, Miss Kate         :   2   2nd:280   1st Qu.:21.00   male  :851   1st Qu.:0.0000
 Kelly, Mr James             :   2   3rd:711   Median :28.00                Median :0.0000
 Abbing, Mr Anthony          :   1             Mean   :30.40                Mean   :0.3427
 Abbott, Master Eugene Joseph:   1             3rd Qu.:39.00                3rd Qu.:1.0000
 Abbott, Mr Rossmore Edward  :   1             Max.   :71.00                Max.   :1.0000
 (Other)                     :1304             NA's   :557
```

Titanic data

```
summary(dat)
```

- Reveals three duplicated people!

```
duplicatedNames = names(which(table(dat$Name) > 1))
as.matrix(duplicatedNames)
```

```
##      [,1]
## [1,] "Carlsson, Mr Frans Olof"
## [2,] "Connolly, Miss Kate"
## [3,] "Kelly, Mr James"
```

Titanic data

```
dat[dat$Name==duplicatedNames[1],]
```

```
##                         Name PClass Age  Sex Survived child adult SurvBin
## 45  Carlsson, Mr Frans Olof    1st  33 Male       No FALSE  TRUE   FALSE
## 708 Carlsson, Mr Frans Olof    3rd  33 Male       No FALSE  TRUE   FALSE
```

- According to historyofthetitanic.org, he was in 1st class.

Titanic data

```
dat[dat$Name==duplicatedNames[2],]
```

```
##                     Name PClass Age    Sex Survived child adult SurvBin
## 729 Connolly, Miss Kate    3rd  30 Female       No FALSE  TRUE   FALSE
## 730 Connolly, Miss Kate    3rd  22 Female      Yes FALSE  TRUE    TRUE
```

- According to historyofthetitanic.org, there were actually two people!

Titanic data

```
dat[dat$Name==duplicatedNames[3],]
```

```
##                 Name PClass Age  Sex Survived child adult SurvBin
## 922 Kelly, Mr James    3rd  44 Male       No FALSE  TRUE   FALSE
## 923 Kelly, Mr James    3rd  42 Male       No FALSE  TRUE   FALSE
```

- According to historyofthetitanic.org, there were actually two passengers.
- The one listed as 44 was actually 19.
- The 42 year old might have been 41.
- There was also a crew member with this name!

> "He gave his age as 44 when signing on to the Titanic… His age is given in the 1911 census as 41."

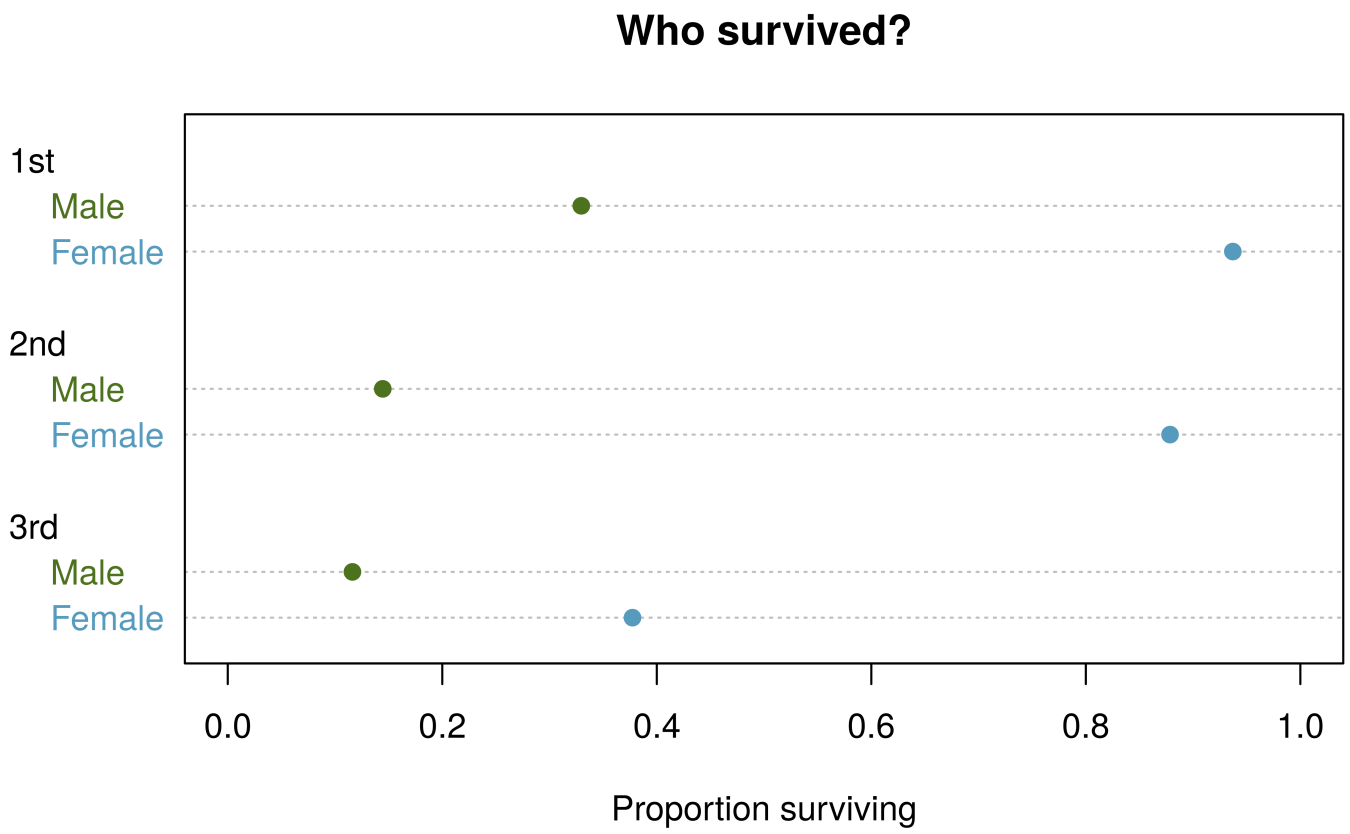
Moral

Look at your data in many ways to make sure you aren't missing something.

Another good function:

```
library(Hmisc)
describe(dat)
```

Dot Plots


Who survived?
1st
Male
Female
2nd
Male
Female
3rd
Male
Female
0.0
0.2
0.4
0.6
0.8
1.0
Proportion surviving

Code for that…

1. `aggregate` creates smaller "summary" data frame:

```
dat$SurvBin=dat$Survived=="Yes" # make it 1 or 0
prop=aggregate(SurvBin ~ PClass + Sex, FUN=mean, data=dat)
prop
```

```
##   PClass    Sex   SurvBin
## 1    1st Female 0.9370629
## 2    2nd Female 0.8785047
## 3    3rd Female 0.3773585
## 4    1st   Male 0.3296089
## 5    2nd   Male 0.1445087
## 6    3rd   Male 0.1162325
```

Code for that…

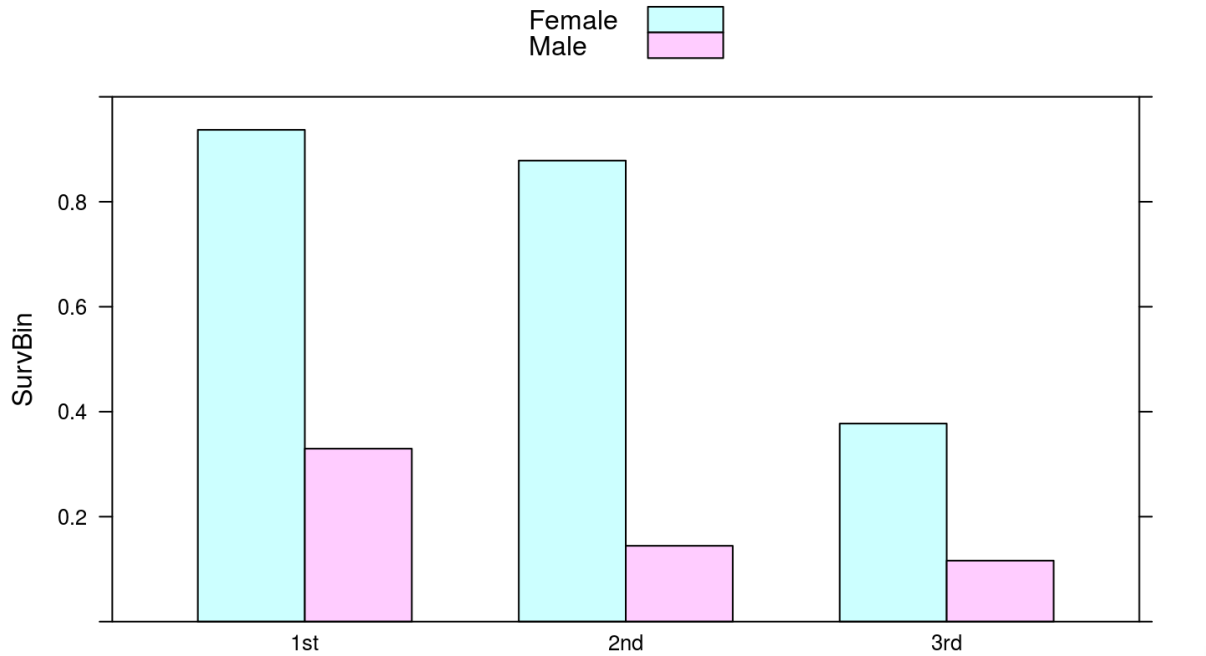
1. Plot this: (using book's color scheme!)

```
twocolors = c("#569BBD", "#4C721D")
dotchart(prop$SurvBin,
        groups = prop$PClass,
        labels = prop$Sex,
        xlim=c(0,1),
        main="Who survived?",
        xlab="Proportion surviving",
        col=twocolors[c(1,1,1,2,2,2)],
        pch=19,         # shape
        cex.lab=1.5,    # size of label
        cex.main=1.5,   # and title
        cex=1.7)        # and dots
```

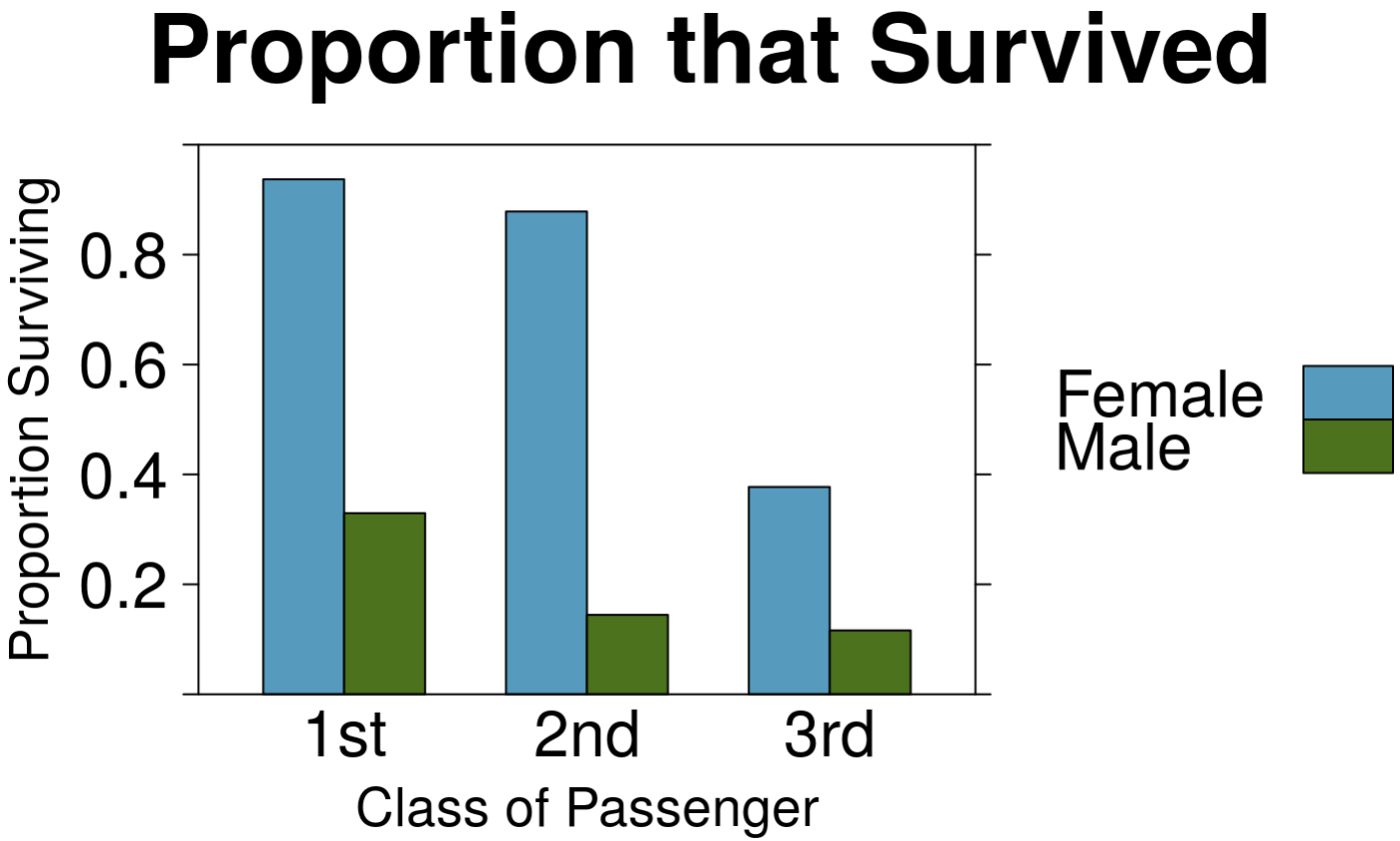
Alternate approach

The library `lattice` is great for plotting data:

```
library(lattice)
barchart(SurvBin~PClass,groups=Sex, ylim=c(0,1), data=prop, auto.key=TRUE)
```



We can make it look nicer…



Though the code is more involved…

```
barchart(SurvBin ~ PClass, groups = Sex, dat = prop,
         ylim = c(0, 1),
         pch = 19, cex = 2,
         xlab = list(label = "Class of Passenger", fontsize = 20),
         ylab = list(label = "Proportion Surviving", fontsize = 20),
         main = list(label = "Proportion that Survived", fontsize = 30),
         par.settings = list(superpose.polygon = list(col = twocolors)),
         scales = list(cex = 2),
         auto.key = list(space = "right", cex = 2))
```