BIOS 4120 Lab 3

January 30 - 31, 2018

Objectives

In today's lab we will:

1. Discuss the relationship between hypothesis testing and confidence intervals
2. Use RStudio to create tables and barcharts from a dataset
3. Walk through an example of calculating a weighted average

Hypothesis Testing and Confidence Intervals

There is very close relationship between confidence intervals and hypothesis testing. All values within a constructed 95% interval are considered plausible values for the parameter that we are estimating. Values outside the interval are rejected as unlikely and improbable.

If the value of the parameter specified by the null hypothesis (for instance 0) is contained within the 95% interval, then the null hypothesis cannot be rejected at the 0.05 level. If the value specified by the null hypothesis is not in the interval, then the null hypothesis can be rejected at the 0.05 level. Likewise, for a 99% confidence interval, values outside the interval are rejected at the 0.01 level.

Constructing Tables

First, let's read in the 'titanic' dataset and compute some summary statistics.

```
titanic <- read.delim("http://myweb.uiowa.edu/pbreheny/data/titanic.txt")
summary(titanic)
```

```
##   Class         Sex          Age           Survived
##  1st :325   Female: 470   Adult:2092   Died    :1490
##  2nd :285   Male  :1731   Child: 109   Survived: 711
##  3rd :706
##  Crew:885
```

By default, when the summary() function encounters categorical data, it produces a table for that column, as evidenced above, when it created 4 separate tables. We can replicate that using the table() function.

```
table(titanic$Class)
```

```
##
##  1st  2nd  3rd Crew
##  325  285  706  885
```

But the table function is more versatile than that. For example, we can create 2x2 tables:
(The with() function lets us use the column names as variables, instead of writing out titanic$ every time.)

```
with(titanic, table(Class,Survived))
```

```
##       Survived
## Class  Died Survived
##   1st   122      203
##   2nd   167      118
##   3rd   528      178
```

```
##   Crew  673      212
```

If we give the function more than two variables, it creates multiple tables (one for each level):

```
with(titanic, table(Class,Survived,Sex))
```

```
## , , Sex = Female
##
##       Survived
## Class  Died Survived
##   1st     4      141
##   2nd    13       93
##   3rd   106       90
##   Crew    3       20
##
## , , Sex = Male
##
##       Survived
## Class  Died Survived
##   1st   118       62
##   2nd   154       25
##   3rd   422       88
##   Crew  670      192
```

I'd recommend keeping the number of variables down to 2 or 3, as more than that begins to get a bit cluttered and confusing.

If we save a table, we can use brackets to access individual numbers [row,column]:

```
table1 <- with(titanic, table(Class,Survived))
print(table1)
```

```
##       Survived
## Class  Died Survived
##   1st   122      203
##   2nd   167      118
##   3rd   528      178
##   Crew  673      212
```

```
table1[3,2]
```

```
## [1] 178
```

```
# The 3 indicates the third row and the 2 indicates the second column,
# so this is the number of 3rd class passengers who survived.
```

We can also use prop.table() to get the proportions for each cell of a table:

```
prop.table(table1, 1) # Gives proportions for each class
```

```
##       Survived
## Class       Died  Survived
##   1st  0.3753846 0.6246154
##   2nd  0.5859649 0.4140351
##   3rd  0.7478754 0.2521246
##   Crew 0.7604520 0.2395480
```

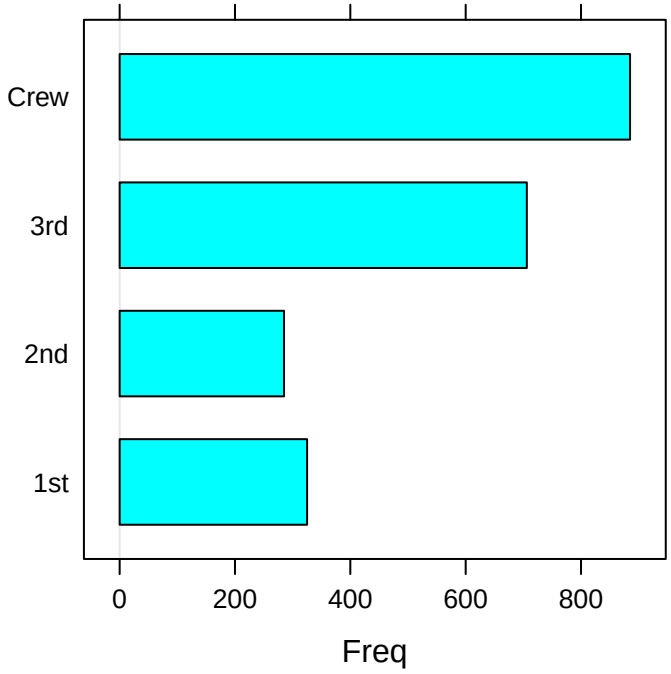
Creating a Bar Chart

If you have data in which the data is catagorical (like we see in the 'titanic' dataset), you will want to use a bar chart to display information. In order to do this you must first install the lattice package and use require(lattice) to load the package.

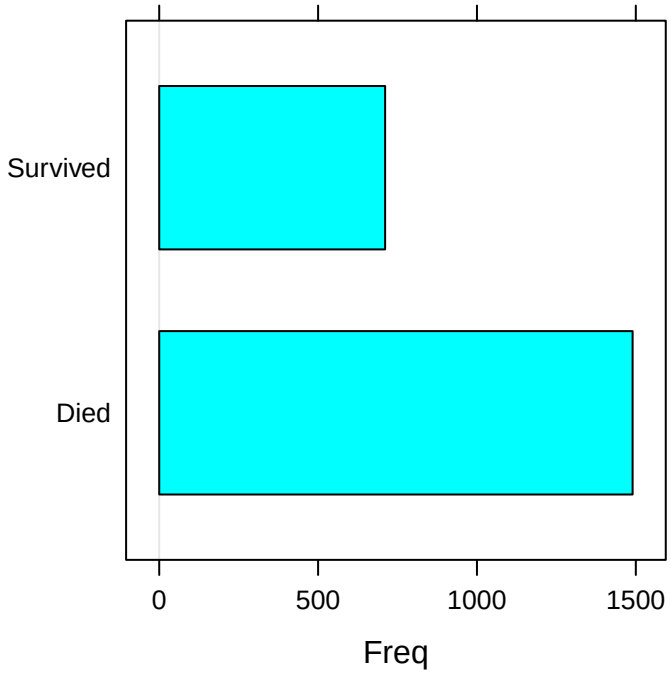
```
require(lattice)
```

Creating a bar chart is pretty simple. You use the barchart() function and the data that you are interested in to create graphics in a few different ways.

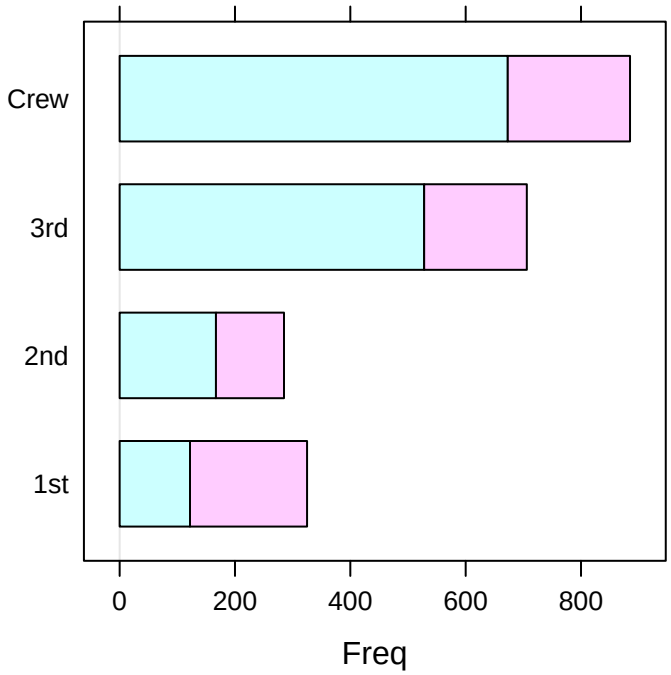
```
table2 <- table(titanic$Class)
barchart(table2)
```



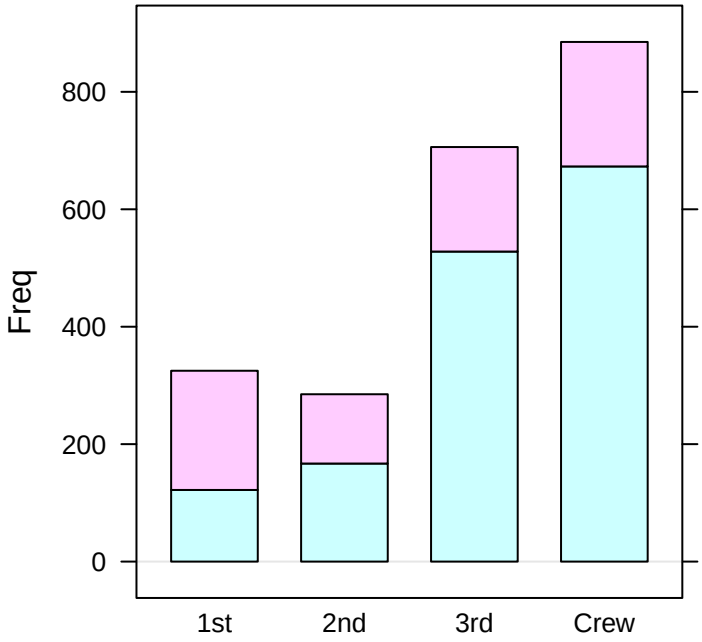

```
barchart(table(titanic$Survived))
```




```
barchart(table(titanic$Class, titanic$Survived)) # Indicates survival within each class
```

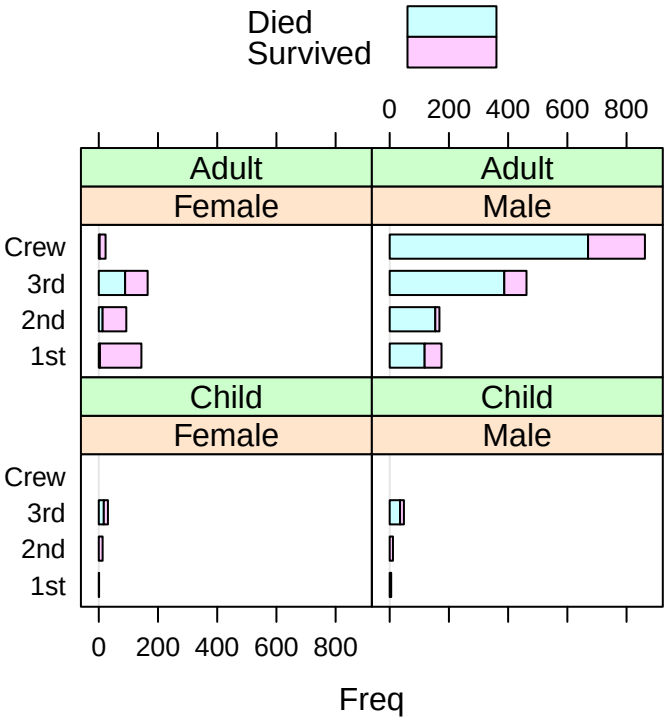



```
barchart(table(titanic$Class, titanic$Survived), horizontal = FALSE) # Vertical Bars
```



There are also more complex options for bar charts where you can visualize many variables such as:

```
table3 <- with(titanic, table(Class, Sex, Age, Survived))
barchart(table3,auto.key=TRUE)
```



Now try adding the argument "scales = 'free' " to the above function. *How did the bar chart change?*

Weighted Averages

Weighted averages can be tricky, so here's an example:
Let's investigate Titanic survival rates based on class.
(For the sake of practice, we will do this by hand and then R.)

```
, , Sex = Female

      Survived
Class  Survived Total
  1st       141   145
  2nd        93   106
  3rd        90   196
  Crew       20    23

, , Sex = Male

      Survived
Class  Survived Total
  1st        62   180
  2nd        25   179
  3rd        88   510
  Crew      192   862
```

Part a

From the tables above, calculate the overall percentages of survival for each class.

Part b

Now, create a table listing the percentage of passengers in each class who survived, broken down by sex.

Part c

Finally, construct a weighted average of the percentage of passengers in each class who survived, controlling for the effect of sex (i.e., report one number for each class).

Do any of these results surprise you? What changed in Part a when compared to Part c? What conclusions can we draw from this?

Answers

```
## Part a
```

```
##       1st       2nd       3rd      Crew
## 0.6246154 0.4140351 0.2521246 0.2395480
```

```
## Part b
```

```
##          Women       Men
## 1st  0.9724138 0.3444444
## 2nd  0.8773585 0.1396648
## 3rd  0.4591837 0.1725490
## Crew 0.8695652 0.2227378
```

```
## Part c
```

```
## 1st :  0.479
```

```
## 2nd :  0.297
```

```
## 3rd :  0.234
```

```
## Crew:  0.361
```

Weighted Averages in R

We can also use R, to solve these problems. There is no simple function that allows you to calculate the weighted average. Below are a few of ways to do this. Note that the first method could introduce mistakes since you are inputting values individually and also could be lengthy process depending on the structure of the dataset. Typically, we would prefer to use the second or third methods which are more efficient and have less opportunity for error.

```
overallSex <- with(titanic,prop.table(table(Sex)))

#First Method
firstclass <- c(141, 62)/ c(145, 180) # From table provided

(first <- weighted.mean(firstclass, overallSex))
```

```
## [1] 0.4785406
```

```
#Second Method
classtable <- with(titanic,table(Sex,Class,Survived))
classes <- prop.table(classtable, 1:2)[,,2]

(class1 <- weighted.mean(classes[,1], overallSex))
```

```
## [1] 0.4785406
```

```
#Third Method (all classes)
apply(classes, 2, weighted.mean, w=overallSex)
```

```
##       1st       2nd       3rd      Crew
## 0.4785406 0.2971914 0.2337568 0.3608609
```

Practice

Now let's say that we want to investigate the difference in survival by sex for the Titanic data set. Construct a weighted average of the percentage of passengers for each sex who survived, controlling for the effect of class.

Practice Answers

```
##    Female      Male
## 0.7541256 0.2138535
```