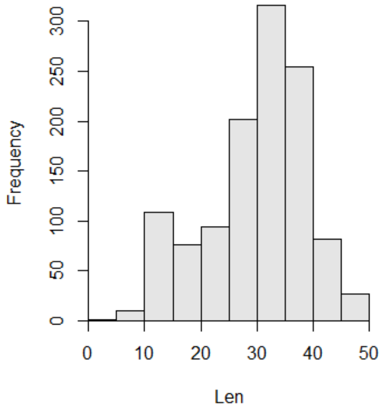
Plotting in R – an Introduction using data from the FM Database
3/8/2016
Zach Lawson, Dan Oele, Gretchen Hansen

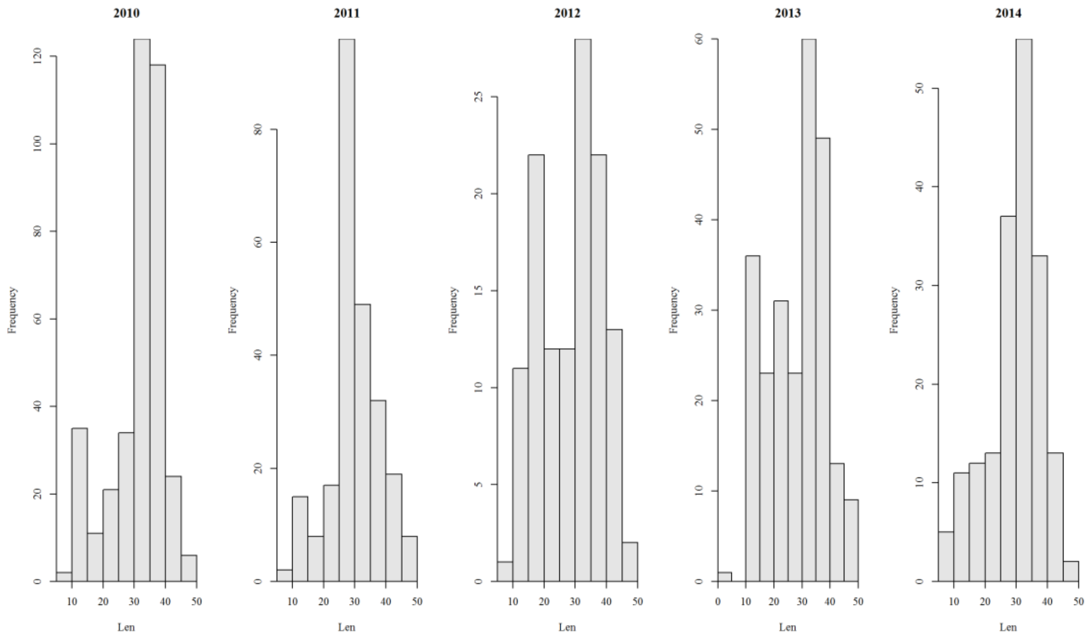
AGENDA

- Section 1: Plotting in base R
- Section 2: Plotting using ggplot
- Section 3: Making your plots look pretty in ggplot
- Troubleshooting tips and tricks
  - GGplot website http://ggplot2.org/
  - R graph catalog http://shiny.stat.ubc.ca/r-graph-catalog/
  - Cookbook for R http://www.cookbook-r.com/
  - Stack overflow http://stackoverflow.com/
  - Google!
- Putting your knowledge to use: assignments or your own data

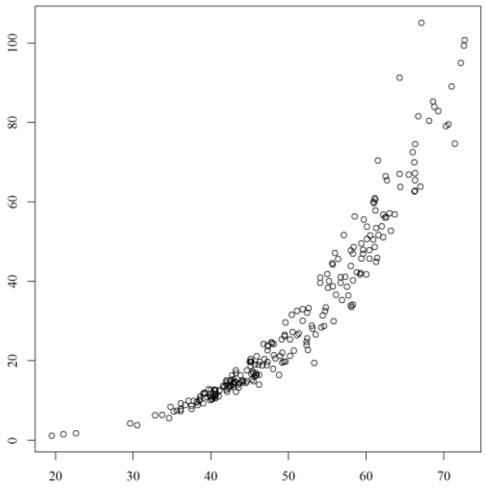
```
# R workshop - plotting in R
#FM statewide, March 8, 2016
##############################################################################
rm(list = ls()) #removes objects from workspace (starts over)
graphics.off() #turns off all graphics
########################### ###########################
# Load packages
library(fishWiDNR)   # for read.FMDB()
library(plyr)       # for summarySE()
library(dplyr)      # for filter(), select(), mutate(), group_by(), summarize()
require(FSA)        # for Summarize(), hist(), expandCounts(), filterD()
library(lubridate)   # for month()
library(ggplot2)    # for ggplot() and facet wrapping
library(gridExtra)   # for tableGrot()
#######################################################################
#prep data
# Load the Sawyer County FMDB data
setwd("C:/Users/oeled/Documents/R_workshop/2016")
d <- read.FMDB("SAWYER_fish_raw_data_012915.csv",expandCounts=TRUE)

# Remove columns with data that will not be used from here on
d <- select(d,Species1,Waterbody.Name,Gear,Survey.Year,
Mon,Len,Weight.Pounds,Age..observed.annuli.,Gender)

##########################################################
# 1 Plotting in Base R
##########################################################
# Subset data for clean musky data
MUE <- filterD(d,Species1=="Muskellunge",Len !="NA")                    # Subset data file for all MUE
########################################################## Histograms
# 1a single plot of single variable
windows()
hist(~Len,data=MUE)                  # Plot all MUE length data
```



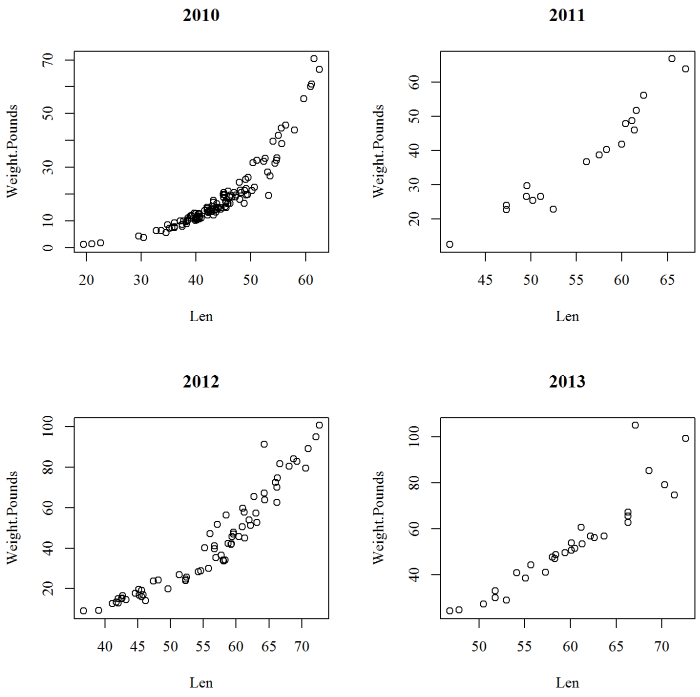
```
# 1b plotting multiple factors (LF x Year)
windows()
par(mfrow=c(1,5))       # Plotting matrix with 2 rows and 2 columns

MUE10 <- filterD(d,Species1=="Muskellunge",Survey.Year=="2010")       # Subset 2010 MUE length data
MUE11 <- filterD(d,Species1=="Muskellunge",Survey.Year=="2011")       #  Subset 2011 MUE length data
MUE12 <- filterD(d,Species1=="Muskellunge",Survey.Year=="2012")       #  Subset 2012 MUE length data
MUE13 <- filterD(d,Species1=="Muskellunge",Survey.Year=="2013")       #  Subset 2013 MUE length data
MUE14 <- filterD(d,Species1=="Muskellunge",Survey.Year=="2014")       #  Subset 2014 MUE length data

hist(~Len,data=MUE10, main="2010")    # Plot 2010 MUE length data
hist(~Len,data=MUE11, main="2011")    # Plot 2011 MUE length data
hist(~Len,data=MUE12, main="2012")    # Plot 2012 MUE length data
hist(~Len,data=MUE13, main="2013")    # Plot 2013 MUE length data
hist(~Len,data=MUE14, main="2014")    # Plot 2014 MUE length data
```



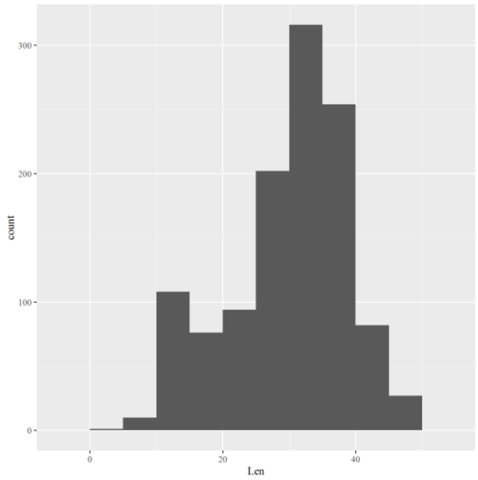
```
###################################################### Scatterplot
# Subset sturgeon data
sturgeon_weights<- data.frame(filterD(d,Weight.Pounds > 0, Survey.Year !="2014",Weight.Pounds !=
"NA",Species1=="Lake Sturgeon"))

# 1c
windows()
plot(Weight.Pounds ~ Len,data=sturgeon_weights)         # Plot all sturgeon length-weight data
```



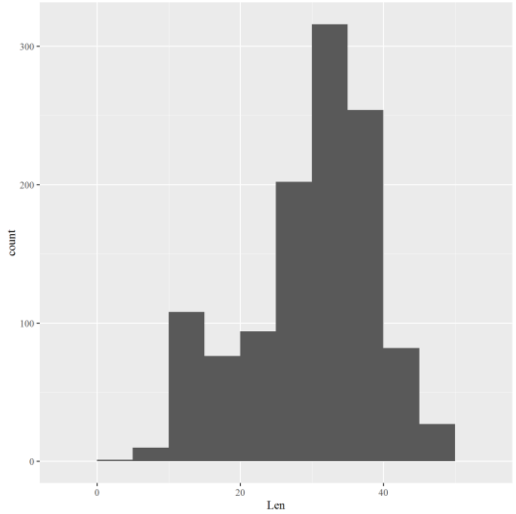
```
# 1d
windows()
par(mfrow=c(2,2))                                                              # Plotting matrix with 2
rows and 2 columns
sturgeon_lw_10 <- filterD(sturgeon_weights,Survey.Year=="2010")      # Subset for 2010 L-W data
sturgeon_lw_11 <- filterD(sturgeon_weights,Survey.Year=="2011")      # Subset for 2011 L-W data
sturgeon_lw_12 <- filterD(sturgeon_weights,Survey.Year=="2012")      # Subset for 2012 L-W data
sturgeon_lw_13 <- filterD(sturgeon_weights,Survey.Year=="2013")      # Subset for 2013 L-W data
plot(Weight.Pounds ~ Len,data=sturgeon_lw_10,main="2010")            # Plot 2010 L-W data
plot(Weight.Pounds ~ Len,data=sturgeon_lw_11,main="2011")            # Plot 2011 L-W data
plot(Weight.Pounds ~ Len,data=sturgeon_lw_12,main="2012")            # Plot 2012 L-W data
plot(Weight.Pounds ~ Len,data=sturgeon_lw_13,main="2013")            # Plot 2013 L-W data
```



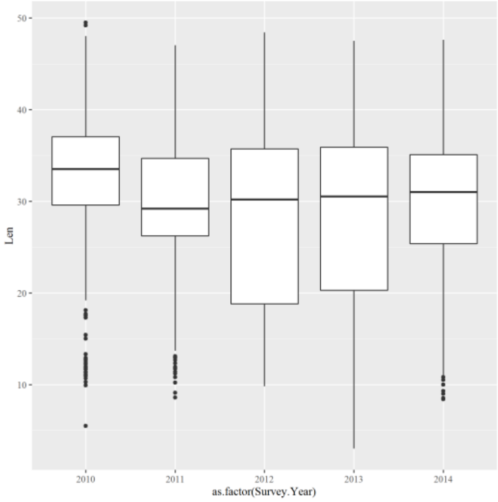
```
##################################################
#Section 2 – plotting in ggplot()
# Simple object plotting GGplot() - similar to BaseR just a different approach (and more efficient for multiple factors)
###################################################
MUE <- filterD(d,Species1=="Muskellunge",Len !="NA") # subset datafram for muskellunge, exclude those values in Len that are NA's
################################################### Histograms
#2a plotting L-F histogram
ggplot(data=MUE, aes(Len)) +                # set up basic ggplot structure
  geom_histogram(binwidth = 5)              # tell ggplot what kind of plot to make and how many bins
```



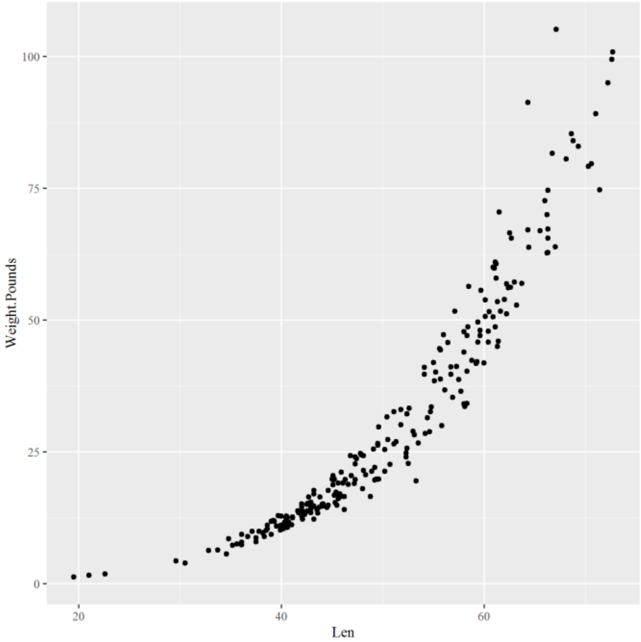
```
#2b plotting L-F histogram 'object'

 # set up basic ggplot structure and object based plotting
MUE_histogram<-ggplot(data=MUE, aes(Len)) +
  geom_histogram(binwidth = 5)
MUE_histogram      # call the name of the ggplot to see it
```



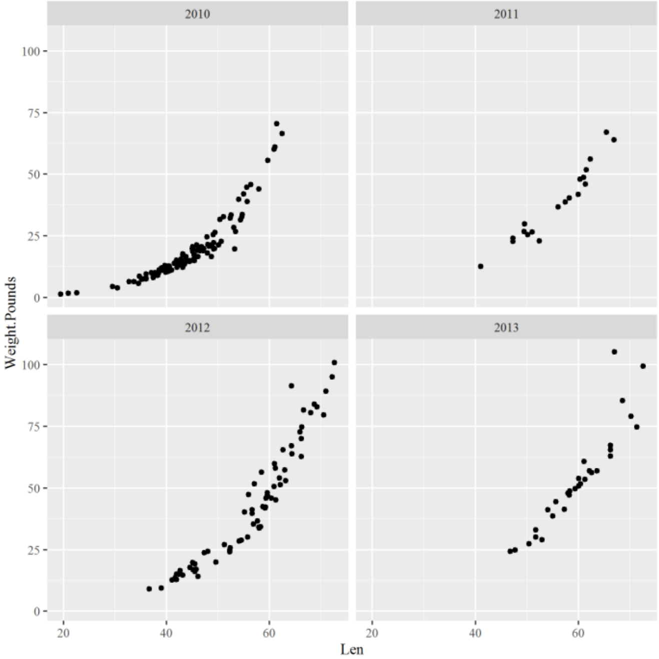
```
#################################################### Boxplots
#2c Plotting length distributions using boxplots instead of histogram
MUE_Length_boxplot<-ggplot(data=MUE, aes(as.factor(Survey.Year),Len)) +
  geom_boxplot()
MUE_Length_boxplot
```



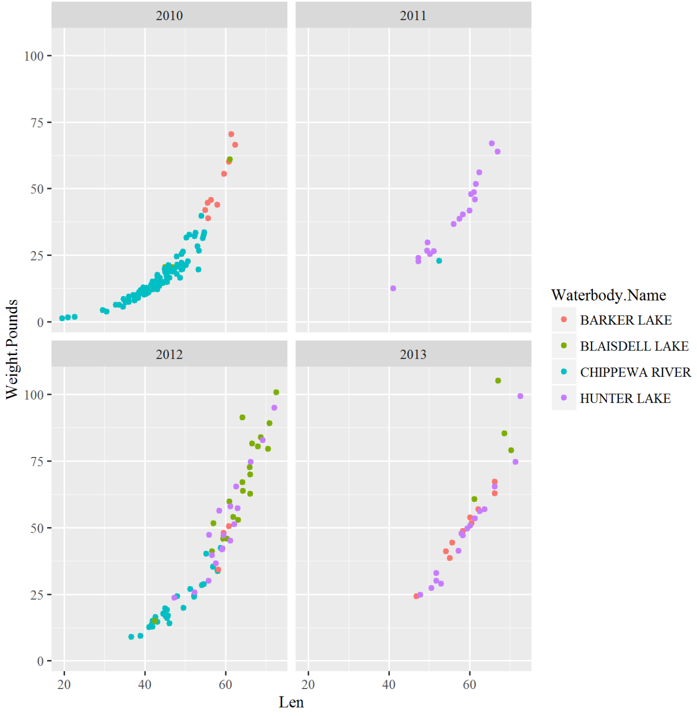
```
#################################################### Scatterplots
# Subset data to get clean lake sturgeon L-W data
sturgeon_weights<- data.frame(filterD(d,Weight.Pounds > 0, Survey.Year !="2014",Weight.Pounds !=
"NA",Species1=="Lake Sturgeon"))

#2d Simple X-Y plotting 2 factor (Length, Weight)
windows()
L_vs_W<- ggplot(data=sturgeon_weights, aes(x=Len,y=Weight.Pounds))+
  geom_point()
L_vs_W
```



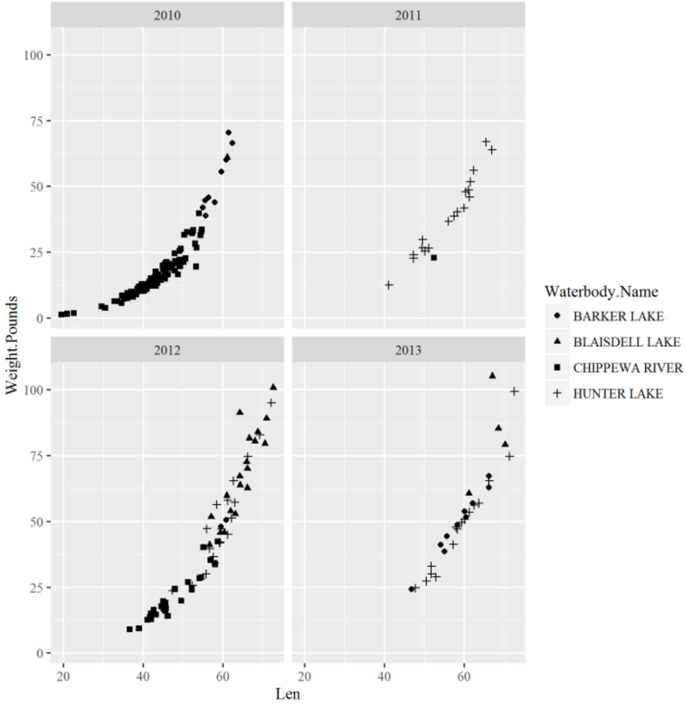
```
#2e X-Y plotting, 3 factor (Length, Weight, Year)
windows()
L_vs_W_year<- ggplot(data=sturgeon_weights, aes(x=Len,y=Weight.Pounds)) +
  geom_point()+
  facet_wrap( ~ Survey.Year)
L_vs_W_year
```



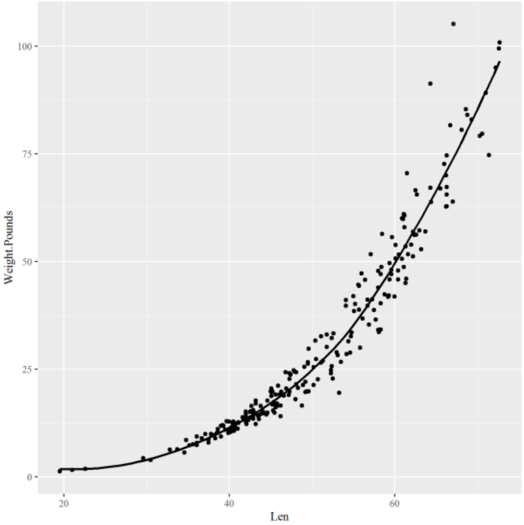
```
#2f X-Y plotting, 4 factor (Length x Weight x Year x Waterbody) (done using 'colour')
windows()
L_vs_W_year_waterbody<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,colour=Waterbody.Name)) +
  geom_point()+
  facet_wrap(~Survey.Year)
L_vs_W_year_waterbody
```



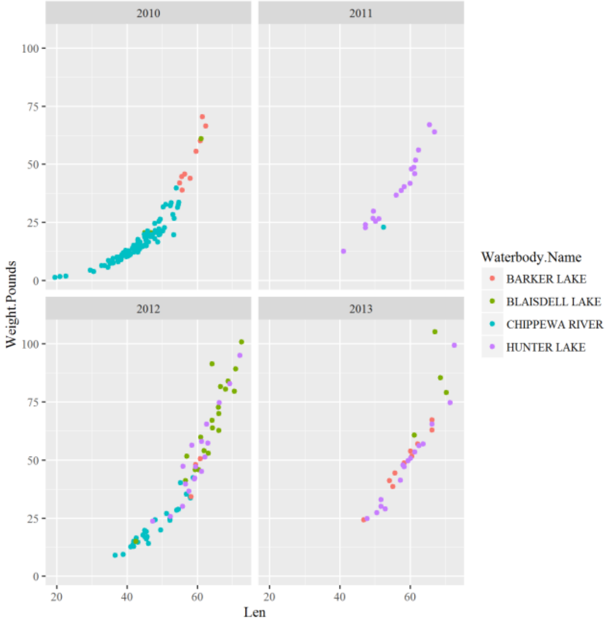
```
#2g X-Y plotting, 4 factor (Length x Weight x Waterbody x Year) (done using 'shape')
windows()
L_vs_W_year_waterbody2<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,shape=Waterbody.Name)) +
  geom_point()+
  facet_wrap(~Survey.Year)
L_vs_W_year_waterbody2
```



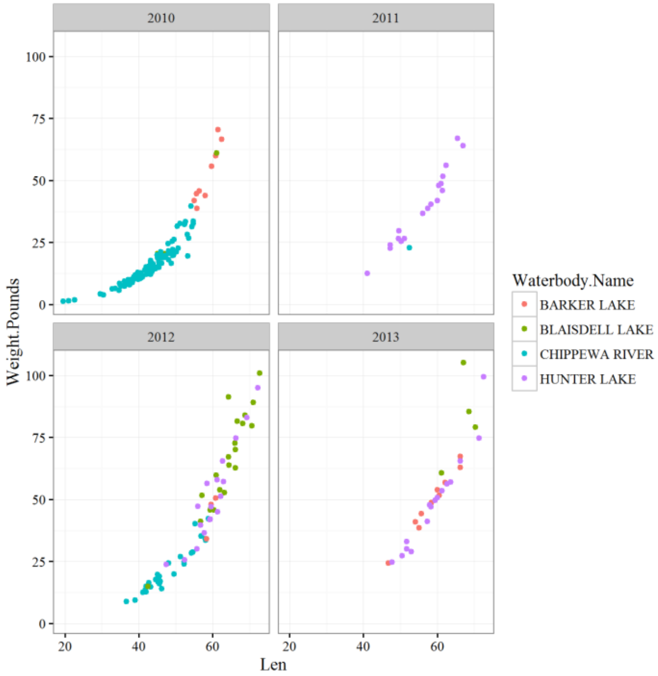
```
################################################### Scatterplot + regression line
#2h (similar to 2c) Simple X-Y plotting 2 factor (Length, Weight) with regression line
windows()
sturg_LW<-ggplot(data=sturgeon_weights, aes(x=Len,y=Weight.Pounds))+
  geom_point()+
  stat_smooth(method = "loess",se=F,fullrange=F,color='black')
sturg_LW
```



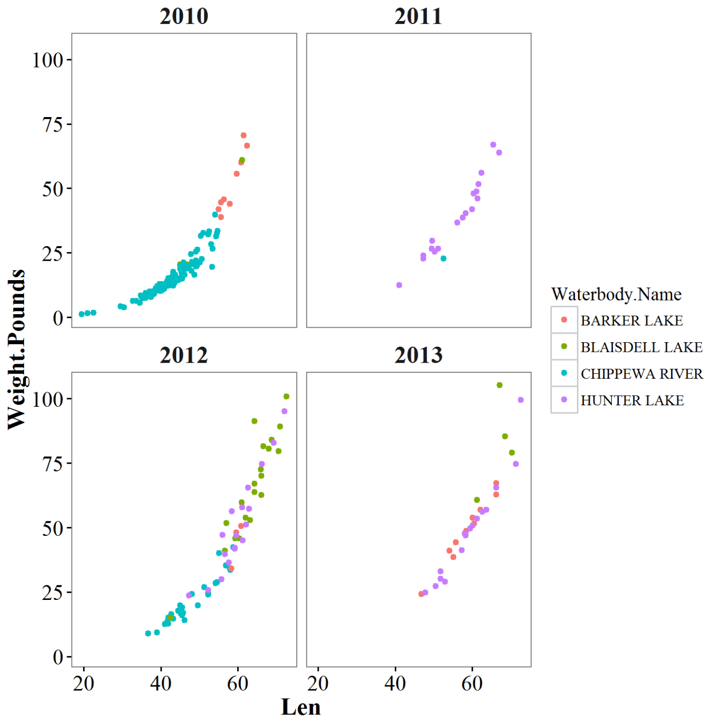
```
#########################################################################
#3. Aesthetics in ggplot - making your plots pretty
##########################################################################
# Remember 2f: X-Y plotting, 4 factor (Length x Weight x Year x Waterbody) (done using 'colour')
windows()
L_vs_W_year_waterbody<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,colour=Waterbody.Name)) +  geom_point()+  facet_wrap(~Survey.Year)
L_vs_W_year_waterbody
```



```
############### basic appearance ####################
#3A change "theme" using built in commands
windows()
L_vs_W_year_waterbody<-L_vs_W_year_waterbody +   theme_bw()
print(L_vs_W_year_waterbody)
```

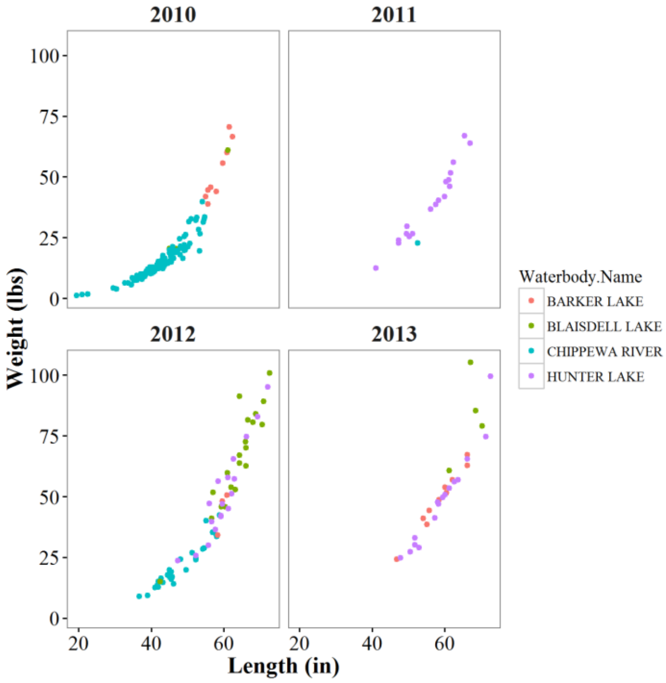


```
#3B change "theme" elements manually
windows()
L_vs_W_year_waterbody<-L_vs_W_year_waterbody +
  theme(axis.title=element_text(size=16, face="bold"), #works on both axes; change axes independently
#using axis.title.x and axis.title.y
      axis.text=element_text(size=14), #works on both axes; change axes independently using axis.text.x
#and axis.text.y
      strip.text.x=element_text(size=16, face="bold"), #changes text of facet box titles
      panel.grid.major = element_line(colour=NA), #removes major gridlines
      panel.grid.minor = element_line(colour = NA), #removes minor gridlines
      panel.background = element_rect(colour = NA), #removes fill color from plot background
      strip.background=element_blank() ) #removes fill color from facet box titles
print(L_vs_W_year_waterbody)
```

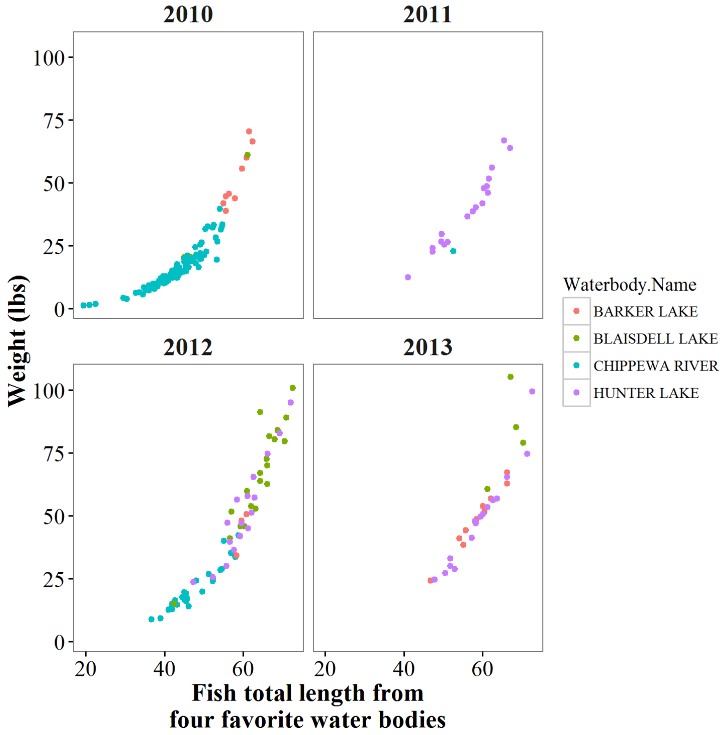


#it is possible to change any aspect of a plot in ggplot
#list of plot aspects you can change using theme: http://docs.ggplot2.org/current/theme.html

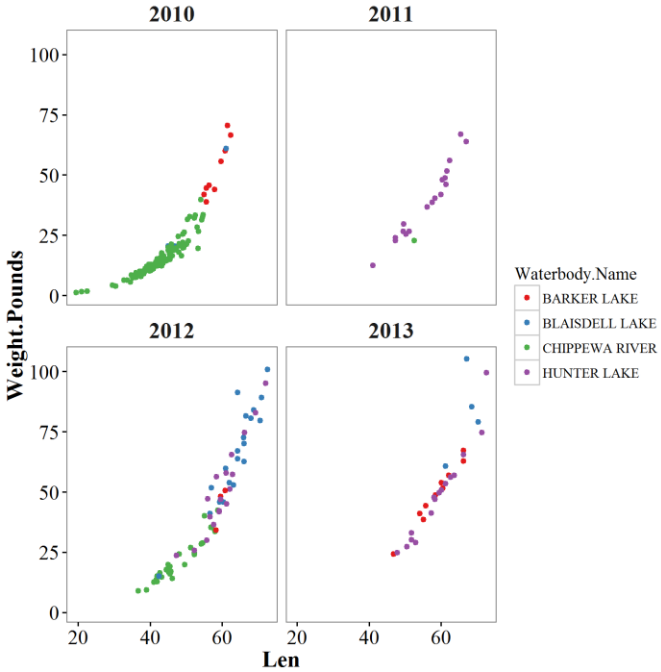
############### Changing or adding text ###################
#3C Changing axis labels
windows()
L_vs_W_year_waterbody6<- L_vs_W_year_waterbody+ xlab("Length (in)")+ylab("Weight (lbs)")
print(L_vs_W_year_waterbody)



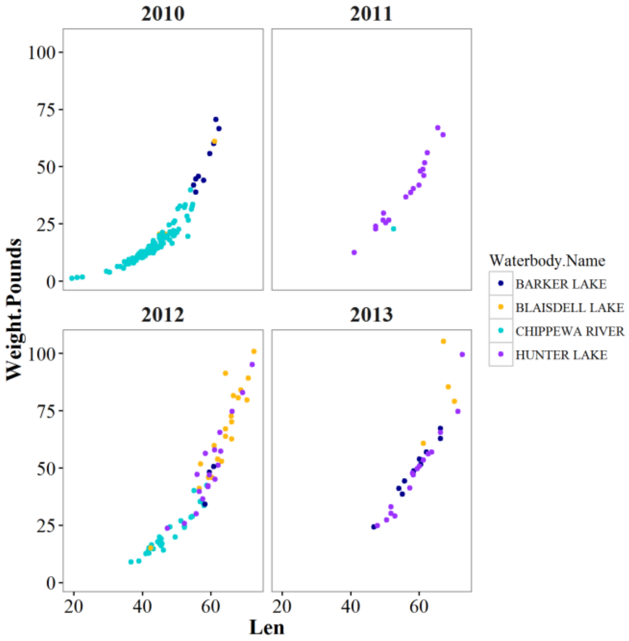
```
#3C.2 changing axis labels - wrapping text onto two lines
windows()
L_vs_W_year_waterbody.test<- L_vs_W_year_waterbody+
  xlab("Fish total length from\nfour favorite water bodies")+ylab("Weight (lbs)")
print(L_vs_W_year_waterbody.test)
```



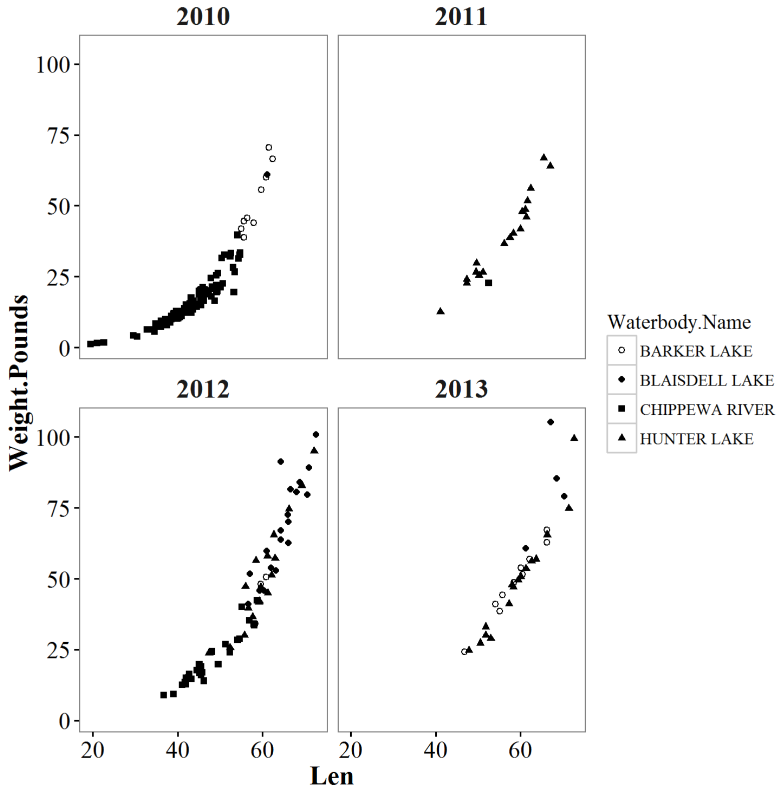
```
#see extra "bonus" code for adding symbols, subscripts, etc to axis labels

############################## Colors and shapes #########################################

#3D changing colors using built in functions
#color brewer for discrete values
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody +
scale_colour_brewer(palette="Set1")  #colour_brewer is one of many built in colour functions. "set1" is
#one of many palettes.
print(L_vs_W_year_waterbody)
```



```
#see http://docs.ggplot2.org/current/scale_brewer.html
#and https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/colorPaletteCheatsheet.pdf
# and http://colorbrewer2.org/ for ideas, tips, and color palettes.

#3E changing colors manually
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody +
  scale_colour_manual(values=c("blue4", "darkgoldenrod1", "darkturquoise", "purple1")) #could be any
#colours
print(L_vs_W_year_waterbody)
```

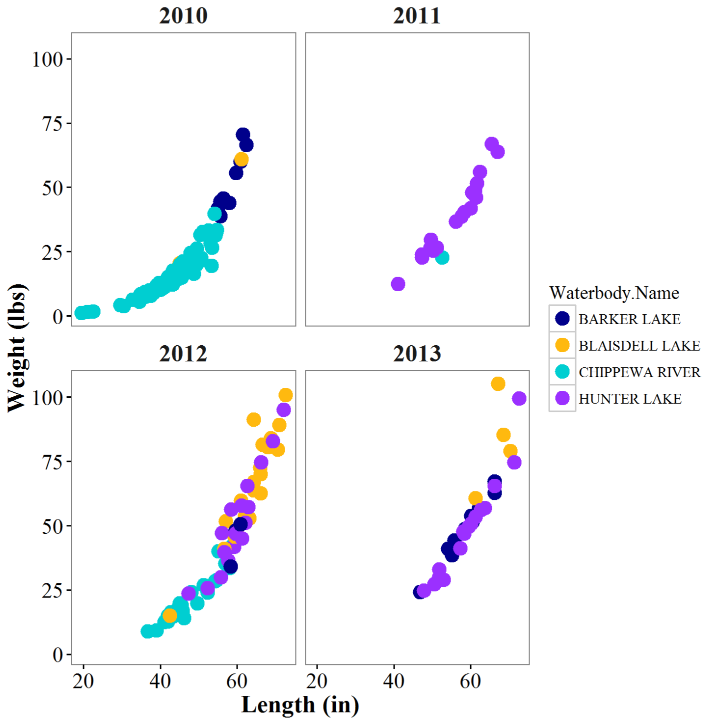


```
#3F changing shapes - using shape instead of colour for waterbody name, as in
#2g X-Y plotting, 4 factor (Length x Weight x Waterbody x Year) (done using 'shape')
#note that we also have to change other aesthetics to make it pretty

windows()
L_vs_W_year_waterbody2<- L_vs_W_year_waterbody2+  scale_shape_manual(values=c(1,16,15,17))+
  theme_bw()+
  theme(axis.title=element_text(size=16, face="bold"),
        axis.text=element_text(size=14),
        strip.text.x=element_text(size=16, face="bold"),
        panel.grid.major = element_line(colour=NA),
        panel.grid.minor = element_line(colour = NA),
        panel.background = element_rect(colour = NA),
        strip.background=element_blank() )
print(L_vs_W_year_waterbody2)
```

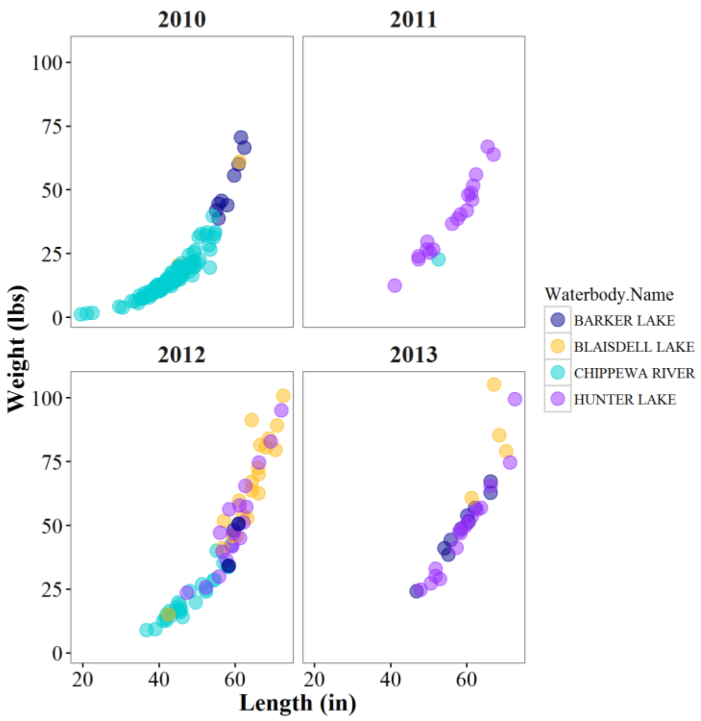


```
#3G changing size of points
#THIS WON'T WORK (AT LEAST NOT THE WAY WE INTEND)
#this will add another set of points on top of existing set
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+geom_point(size=4)
print(L_vs_W_year_waterbody)
```

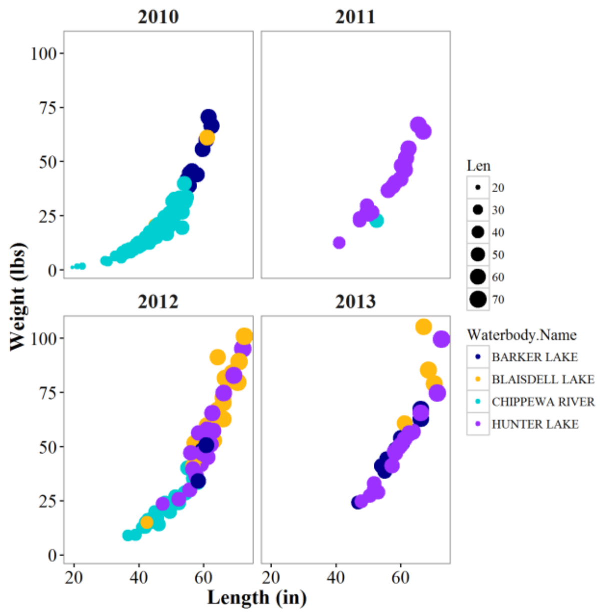
```
#Instead, reset object with full set of code
windows()
L_vs_W_year_waterbody<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,colour=Waterbody.Name)) +
  geom_point(size=4)+
  facet_wrap(~Survey.Year)+
  scale_colour_manual(values=c("blue4", "darkgoldenrod1", "darkturquoise", "purple1"))+
  xlab("Length (in)")+ylab("Weight (lbs)")+
  theme_bw()+
  theme(axis.title=element_text(size=16, face="bold"),
        axis.text=element_text(size=14),
        strip.text.x=element_text(size=16, face="bold"),
        panel.grid.major = element_line(colour=NA),
        panel.grid.minor = element_line(colour = NA),
        panel.background = element_rect(colour = NA),
        strip.background=element_blank() )
print(L_vs_W_year_waterbody)
```



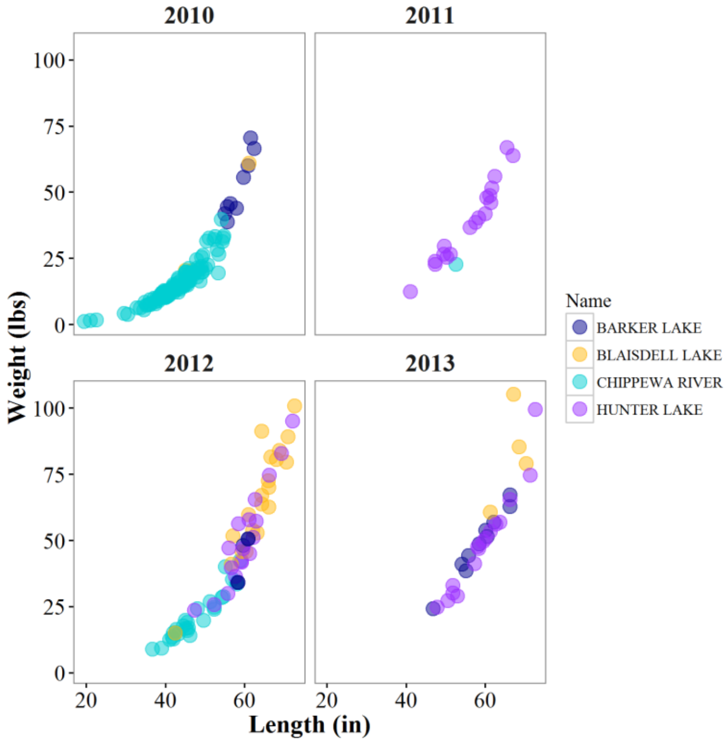
```
#3H make points see-through so you can better see overlapping points
windows()
L_vs_W_year_waterbody<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,colour=Waterbody.Name)) +
  geom_point(size=4, alpha=.5)+ #alpha changes transparency of points, ranges from 0-1
  facet_wrap(~Survey.Year)+
  scale_colour_manual(values=c("blue4", "darkgoldenrod1", "darkturquoise", "purple1"))+
  xlab("Length (in)")+ylab("Weight (lbs)")+
  theme_bw()+
 theme(axis.title=element_text(size=16, face="bold"),
       axis.text=element_text(size=14),
       strip.text.x=element_text(size=16, face="bold"),
       panel.grid.major = element_line(colour=NA),
       panel.grid.minor = element_line(colour = NA),
       panel.background = element_rect(colour = NA),
       strip.background=element_blank() )
print(L_vs_W_year_waterbody)
```



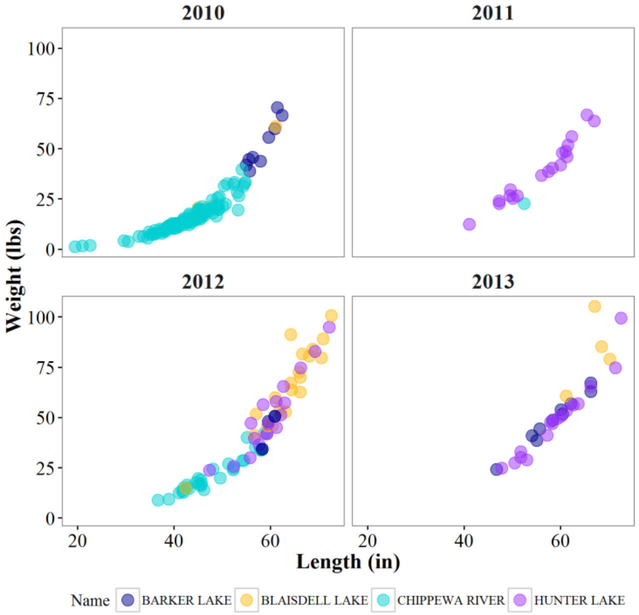
```
#3I change size as function of variable
windows()
L_vs_W_year_waterbody_test2<- ggplot(data=sturgeon_weights,
aes(x=Len,y=Weight.Pounds,colour=Waterbody.Name)) +
 geom_point(aes(size=Len))+ #size of point is proportional to length
 facet_wrap(~Survey.Year)+
 scale_colour_manual(values=c("blue4", "darkgoldenrod1", "darkturquoise", "purple1"))+
 xlab("Length (in)")+ylab("Weight (lbs)")+
 theme_bw()+
 theme(axis.title=element_text(size=16, face="bold"),
       axis.text=element_text(size=14),
       strip.text.x=element_text(size=16, face="bold"),
       panel.grid.major = element_line(colour=NA),
       panel.grid.minor = element_line(colour = NA),
       panel.background = element_rect(colour = NA),
       strip.background=element_blank() )
print(L_vs_W_year_waterbody_test2)
```



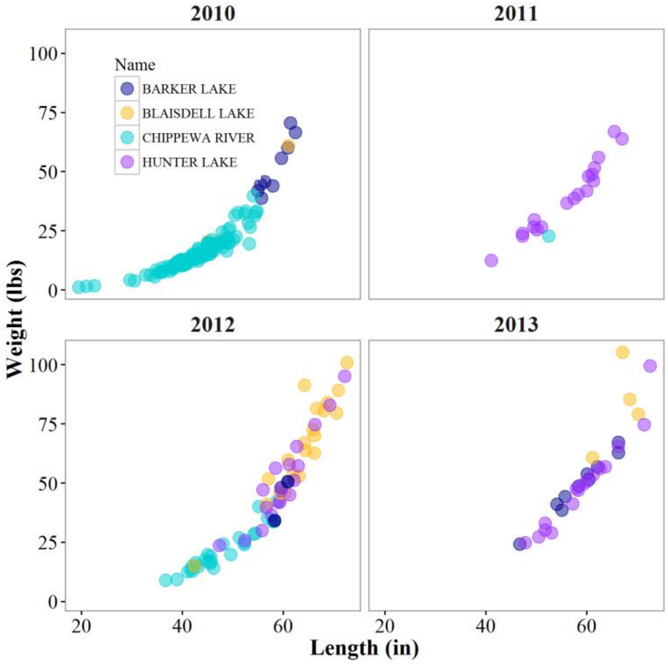
```
####################### Changing the legend ##############################
#3J Changing legend title
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+
  scale_colour_manual(name= "Name",values=c("blue4", "darkgoldenrod1", "darkturquoise",
"purple1")) #set legend title within color scale
print(L_vs_W_year_waterbody)
```



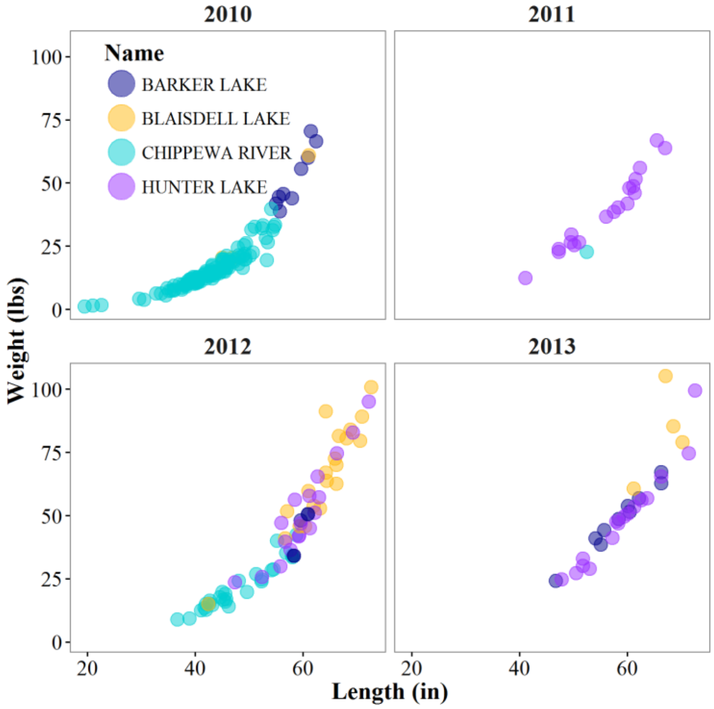
```
#3K Changing legend position
#using directional commands
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+
  theme(legend.position="bottom")
print(L_vs_W_year_waterbody)
```



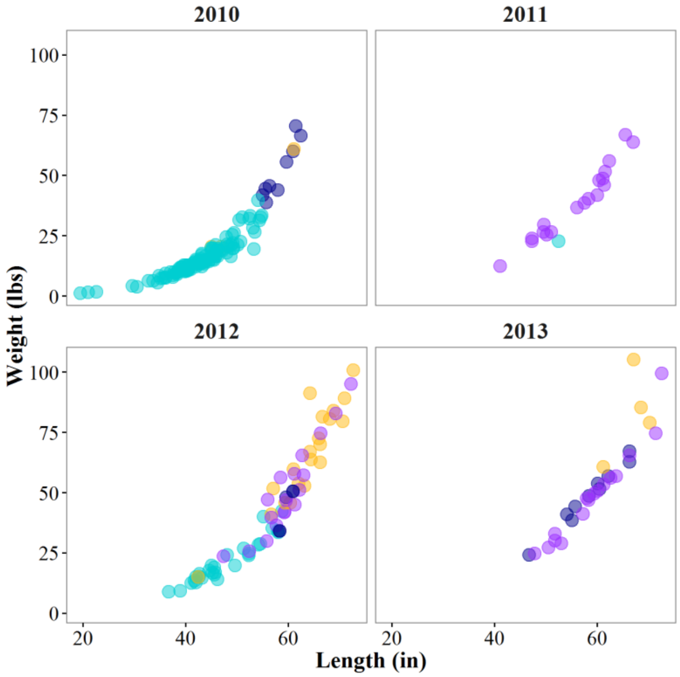
```
#using coordinates
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+
  theme(legend.position=c(.2,.85)) #coordinates are x,y, and indicate proportion of total length of axis
print(L_vs_W_year_waterbody)
```



```
#3L changing appearance of legend - change legend font, remove boxes around dots, remove legend
background, and
#make legend symbols bigger
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+
  guides(colour = guide_legend(override.aes = list(size=8))) + #this sets the size of legend points
#independently
  theme(legend.title=element_text(size=16, face="bold"),
        legend.text=element_text(size=12),
        legend.key = element_blank(), #this removes boxes around dots in legend
        legend.background=element_blank() #this removes white fill from inside legend rectangle
  )
print(L_vs_W_year_waterbody)
```



```
#3M turning legend off
windows()
L_vs_W_year_waterbody<- L_vs_W_year_waterbody+
  theme(legend.position="none")
print(L_vs_W_year_waterbody)
```



```
######################Saving your figure#################################################
#3N save as png
ggsave('C:/Users/hanseg/Documents/R workshop/FM statewide March
2016/sturgeon_length_weights.png',
    height=4, width=4, units='in', dpi=300)

#3O save as Tiff, compress file size
ggsave('C:/Users/hanseg/Documents/R workshop/FM statewide March
2016/sturgeon_length_weights.tiff',
    height=4, width=4, units='in', dpi=300, compression="lzw")

#3p save as Tiff, change figure dimensions - this is too small for font size and point size!
ggsave('C:/Users/hanseg/Documents/R workshop/FM statewide March
2016/sturgeon_length_weights_too_small.tiff',
    height=2, width=2, units='in', dpi=300, compression="lzw")
```

ASSIGNMENTS (or make whatever plot you want, using your own data)

#Assignment 1:

Using BaseR, make a 2x2 panel figure plotting Length Vs. Weight for all muskies sampled in Sawyer County for 2010-2014 (hint: look to 1b for subsets and 1c for plotting code).

#Assignment 2:

For 2014 data, use GGplot to plot length distributions for all panfish species sampled (hint, look to 2b for similar plots, but keep in mind you have to use facet_wrap()).

#Assignment 3:

Now, for the same subset of data, plot the same length distributions using geom_boxplot() (hint, don't use facet_wrap() to plot by factor, use aes() in the ggplot() line).

#Assignment 4:

For length-weight scatterplots in 2e of the script, log transform both length and weight data and add a linear trendline (hint: use log() function to transform, and use stat_smooth() to add line similar to 2h only 'method' will be 'lm' instead of 'loess').

#Assignment 5:

Using the figure created in #4 above, remove major gridlines, minor gridlines, re-label the axes, and create a title (hint: 3b and 3c).

#Assignment 6:

Using the plot from figure 5 (or from 2e), change the color of each datapoint to represent the waterbody that the individual was sampled from. Now re-label the legend and change the color and size of the datapoints to your liking (hint: use 2e as a base figure and 3e and 3g for customizing datapoint aesthetics, and 3L for changing the legend).

#Assignment 7:

Now, print your figure as a PNG file to your desktop (hint: 3n)