```
require(xtable)

## Loading required package:  xtable

rubyRBT <- read.csv("~/Documents/Stat505/data files/Ruby-RBT.csv", head = TRUE)
# summary(rubyRBT[,-8])
```

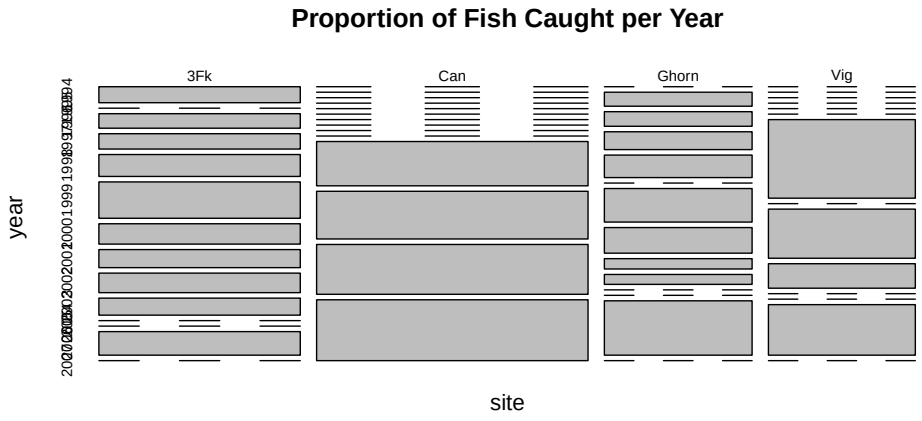
Homework 1 STAT 505 Fall 2013 Leslie Gains-Germain

The csv file contains records on 7,439 Rainbow trout caught by FWP personnel in the Ruby river at four locations (Canyon, Greenhorn, ThreeForks and Vigilante) from 1994 to 2007. We need to tabulate counts of the fish in a table which will tell us: year, site, and their capture status: captured in first pass, captured in second pass and unmarked, or captured in second pass and marked. (These values get used to estimate population size using mark-recapture modeling.

1. Show plots to describe the data:

   (a) A mosaicplot or stacked barchart to show the relative proportion of fish caught per year within each site. Explain what you see.

```
mosaicplot(site ~ year, data = rubyRBT, main = "Proportion of Fish Caught per Year")
```



In the Three Forks area, we can see that about the same proportion of fish were caught each year, except for the years when no fish were caught (1995,2004,2005, and 2007), and 1999 when a larger proportion of fish were caught.

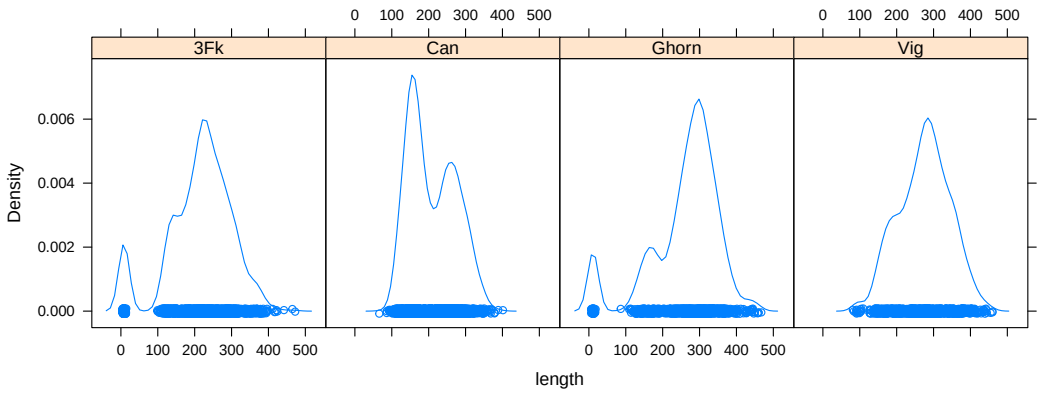
In Canyon, we can see that fish were only collected in 2004 – 2007, and about the same proportion of fish were caught in each of these years.

In Greenhorn, we see more variation in the amount of fish caught each year. The greatest proportion of fish caught in Greenhorn were caught in 2006.For those years that fish were caught, the smallest proportion of fish were caught in 2002 and 2003. There were no fish caught in 2007, 2005, 2004, 1999, and 1994.

In the Vigilante area, fish were only caught in 2006, 2003, 2002, and 2000. The greatest proportion of fish were caught in 2000.

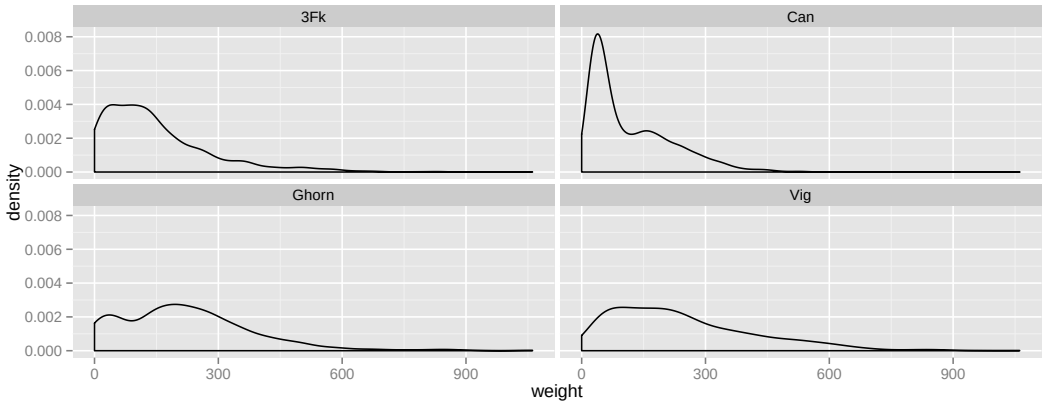
(b) Plots to compare **distributions** of length separated by site. (Use lattice or ggplot so this is one plot with multiple panels). Do the same for weight, or if you prefer, log(weight). Explain what you see.

```
require(lattice)
densityplot(~length | site, data = rubyRBT)
```



It looks like the Greenhorn and Vigilante areas had the longest fish on average, with the greatest proportion of fish being around 300 mm in length. Greenhorn, Vigilante, and Three Forks were all similar in that they all had a small bump between 100 and 200 mm and then a larger peak above 200 mm. In the Canyon, we see the larger peak at $\approx$ 150 mm and the smaller peak at $\approx$ 280 mm. It's possible that we are looking at two distinct groups (male vs. female, spawning vs. non-spawning, spring vs. fall or other). Three Forks and Greenhorn also have a strange peak below 50 mm. I am guessing that these data points were recorded in the wrong units, because I can't imagine that they were able to collect fish that small.

```
require(ggplot2)
qplot(x = weight, facets = ~site, data = rubyRBT, geom = "density")
```

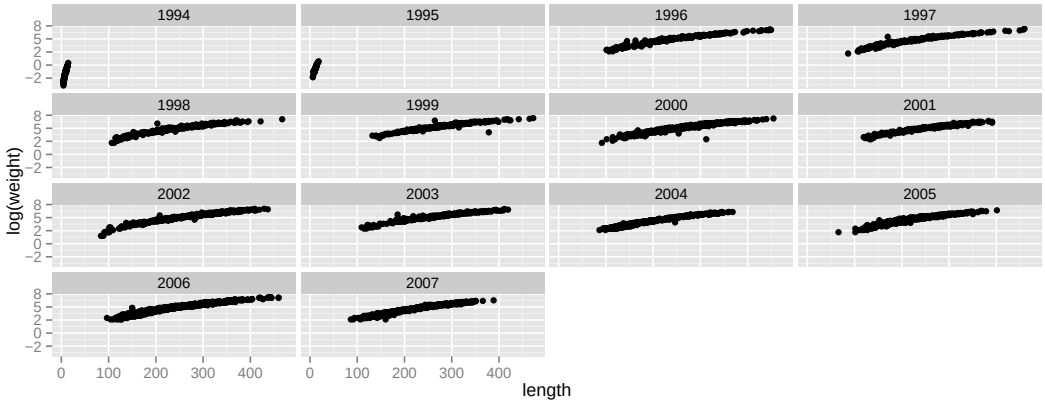


Three Forks and Canyon had the lightest fish, with most of the fish collected weighing between 100 and 300 grams. Canyon had the smallest range of fish weights. The fish collected in Greenhorn and Vigilante had the largest range of weights, with fish weighing anywhere from almost nothing to 600 grams. The

fish were distributed more evenly across the range of weights in Greenhorn and Vigilante than they were in the Canyon.

(c) Relationship between log(weight) and length separated by year. Explain what you see. Which years look odd? Make a guess about why earlier years were measured differently and use R to fix the problem.

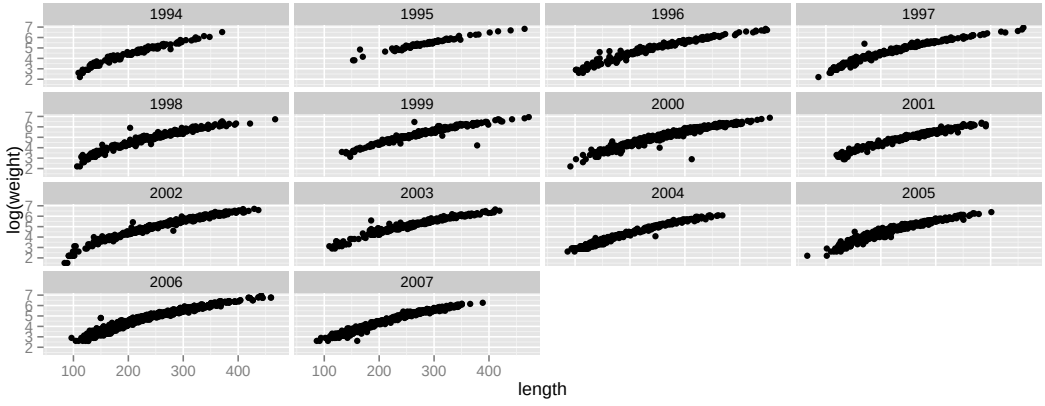
```
require(ggplot2)
qplot(x = length, y = log(weight), facets = ~year, data = rubyRBT, geom = "point")
```



It looks like length and weight vary directly for each year. The range of lengths and weights seems to generally stay the same from 1996 to 2007. In 1994 and 1995, I'm guessing that they recorded the data in different units. Instead of using mm for length and g for weight, they probably used inches for length and pounds for weight. This can be fixed by rescaling the data.

```
rubyRBT[rubyRBT$year == 1994, 3] <- rubyRBT[rubyRBT$year == 1994, 3] * 25.4  #change length, 1
rubyRBT[rubyRBT$year == 1994, 4] <- rubyRBT[rubyRBT$year == 1994, 4] * 454.5  #change weight,
rubyRBT[rubyRBT$year == 1995, 3] <- rubyRBT[rubyRBT$year == 1995, 3] * 25.4  #change length, 9
rubyRBT[rubyRBT$year == 1995, 4] <- rubyRBT[rubyRBT$year == 1995, 4] * 454.5  #change weight,
```

```
qplot(x = length, y = log(weight), facets = ~year, data = rubyRBT, geom = "point")
```



2. Make a new column which tells capture status: first pass, 2nd marked, or 2nd unmarked. We know which of the second pass fish had also been captured on first pass by their mark, so the mark identifies fish caught in both passes.¡br¿ This requires

answering two questions, one about pass number, another about mark, so use two ifelse statements, one nested in the other.

```
rubyRBT$capture <- ifelse(rubyRBT$trip == 1, "1", ifelse(rubyRBT$trip == 2 & rubyRBT$mark ==
    1, "2M", "2U"))
```

3. Make a table of capture status by site. Use xtable [in package xtable] to make it print nicely.

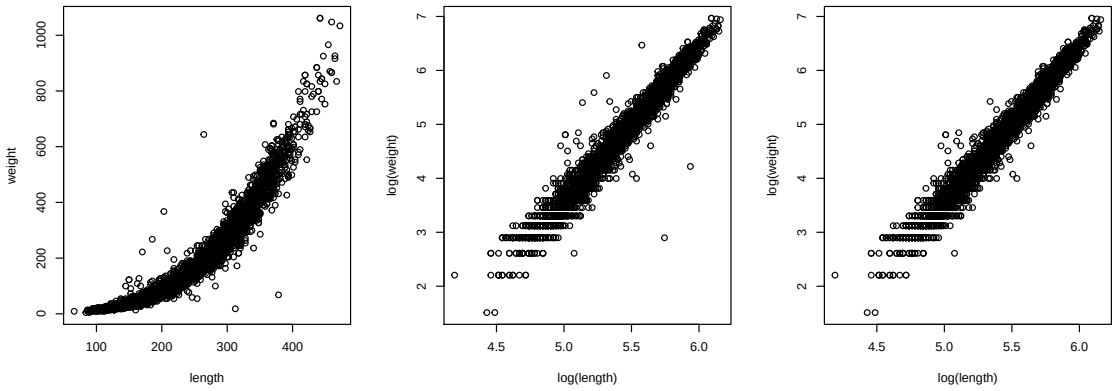
```
require(xtable)
xtable(with(rubyRBT, table(capture, site)))
```

	3Fk	Can	Ghorn	Vig
1	1194	1411	735	781
2M	321	345	235	317
2U	437	875	462	326

4. Give a biological or a geometry-based argument suggesting that the effect of length on weight is best viewed by taking logs on each. Plot log(weight) as a function of log(length). Comment on the relationship. Discuss: does it make sense to remove any outliers? If so, remove up to 1% of the data, and give a justification for removing those points.

   Biologically, we would expect the distributions of lengths and weights to be skewed to the right. The largest number of fish are small, but small fish have a lot of predators! Not all of the small fish live to grow bigger, so each higher length and weight class has a smaller amount of fish. But, there are probably at least a few fish in a population that avoid predators and grow really big. Thus, the distribution of lengths and weights is likely to be skewed right. To make our data more symmetrical it is better to work with log(weight) and log(length). Also, log(weight) and log(length) show a linear relationship.

```
par(mfrow = c(1, 3))
with(rubyRBT, plot(length, weight))
with(rubyRBT, plot(log(length), log(weight)))
# identify(x=(rubyRBT£length), y=(rubyRBT£weight))
ruby.adj3 <- rubyRBT[-c(478, 6828, 2064, 6259, 6554, 6833), ]
with(ruby.adj3, plot(log(length), log(weight)))
```



Let's first look at the four lone points lying above the main body of values. Row 6833 describes a fish that is 264.16 mm long (thats about 6.7 inches) but weighs 643.97 grams (thats about 1.4 lbs). That seems like a pretty heavy six inch fish, so there might have been an error when the data on that fish was recorded. Now, let's turn our attention to the two lone points lying below the majority of the data. Row 6828 describes a fish that is 37.8 cm long (thats about 14.9 inches) and weighs 68.0 grams (thats .15 lbs). That would be a really long fish with a really small weight. That doesn't seem physically possible, so it is therefore likely that this fish's weight or length was recorded incorrectly. Therefore, I removed the two outliers lying below the majority of the data, and I removed the four outliers lying above the majority of the data.

5. Fit a linear model for log(weight) on log(length). Does the intercept depend on site? Does the slope? Fit a model with main effects and appropriate interactions. Interpret each coefficient estimate. Explain exactly what effect each is measuring. What have we learned about the relationships?

```
rubyregress <- lm(log(weight) ~ site * log(length), data = ruby.adj3)
require(xtable)
xtable(summary(rubyregress), digits = 3)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-10.887	0.069	-157.770	0.000
siteCan	0.364	0.087	4.172	0.000
siteGhorn	0.308	0.113	2.712	0.007
siteVig	-0.375	0.112	-3.357	0.001
log(length)	2.893	0.013	226.790	0.000
siteCan:log(length)	-0.074	0.016	-4.565	0.000
siteGhorn:log(length)	-0.065	0.021	-3.151	0.002
siteVig:log(length)	0.064	0.020	3.138	0.002

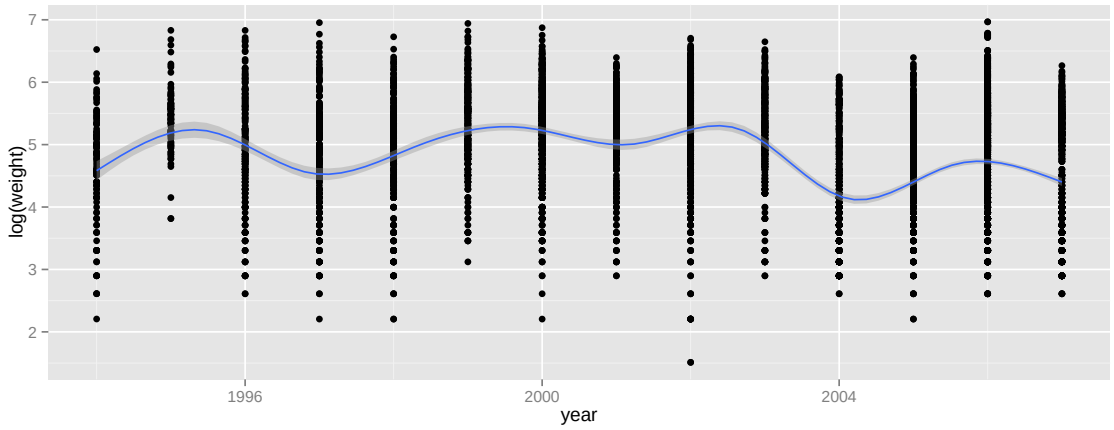
Yes, the intercept does depend on site. For Three Forks, the intercept is -11.035, but for Canyon and Greenhorn the intercept is a little higher, $-10.887+0.364 = -10.523$

and $-10.887 + .308 = -10.579$ respectively. For Vigilante, the intercept is a little lower, $-10.887 - 0.375 = -11.262$. The slope also depends on site. For the Three Forks area, the slope is 2.893. Canyon and Greenhorn have smaller slopes, 2.819 and 2.828 respectively. Vigilante has a slightly larger slope of 2.957. The intercept measures the intercept of the regression model for that site, and the slope measures the increase in log(weight) for a one unit increase in log(length). From this table, we have learned that the slope and intercept of our model are affected by site.

6. Consider adding year to the model. Note: you need to think about how to add it. It's numeric, so just `+ year` will fit a regression coefficient to year. Alternatively, verb—+ factor(year)— fits an adjustment for each year after 1994. Which makes more sense to you? Does it improve the fit?

```
require(ggplot2)
qplot(year, log(weight), data = na.omit(ruby.adj3), geom = c("point", "smooth"))


## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with
formula:  y ~ s(x, bs = "cs").  Use 'method = x' to change the smoothing method.
```



Looking at this qplot, we can see that there is not a clear linear relationship between year and log(weight). I am going to treat year as a categorical variable.

```
rubyregress2 <- lm(log(weight) ~ site * log(length) + factor(year), data = ruby.adj3)
# require(xtable) xtable(summary(rubyregress2), digits=3)
```

Lets look at the anova tables for each of our models to see if adding year improves the fit.

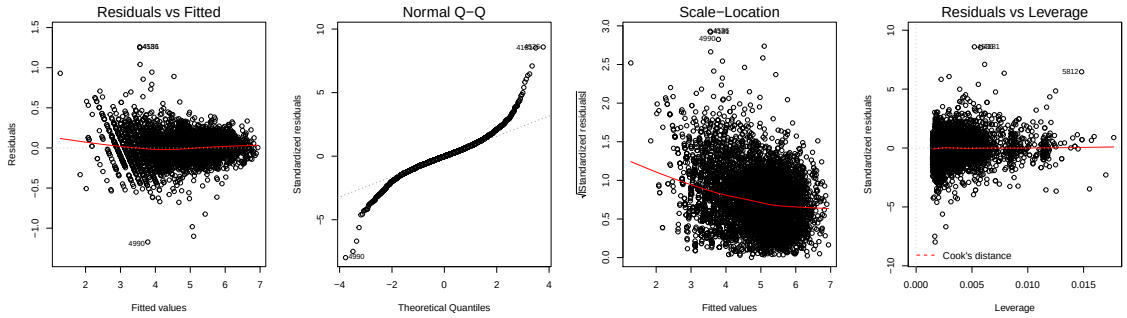
```
require(xtable)
xtable(anova(rubyregress, rubyregress2))
```

We can see that the residual sum of squares is smaller once year is added to the model, and therefore the p-value is small indicating that the fit is improved when year is added.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	6207	139.95				
2	6194	133.96	13	5.98	21.27	0.0000

7. Plot the usual four diagnostic plots and comment on what you see.

```
par(mfrow = c(1, 4))
plot(rubyregress2)
```



Looking at these diagnostic plots, it seems that residuals might have a small problem with variance and normality. Looking at the residuals vs. fitted values plot, we seem to have linearity because the residuals are centered around the fitted line. However, the residuals seem to have smaller variance for higher fitted values, and the normal Q-Q plot shows a larger gap in the tails than we expect. I only removed 6 outliers, so we might see normality and more constant variance if I went back and removed some additional outliers.

R Code

```
mosaicplot(site ~ year, data = rubyRBT, main = "Proportion of Fish Caught per Year")
```

```
require(lattice)
densityplot(~length | site, data = rubyRBT)
```