A Master's Crash Course to R

Maria Kamenetsky

Friday, October 30, 2015

Data Cleaning

- *Read in Data*

```
#Import from CSV/tab-delimited
deer <- read.csv("deer.csv")
```

- *Read Out Data*

```
write.table(df,"df.csv", sep=",")
#This creates CSV, but in excel you need to move over the column names by one cell and then drop first
```

- *Gather Data/Create Observation Unit*
  - dmu_long <- gather(dmu, year, plotSize, X1983:X2011)
    - This will rectangularize the dataframe to make the obs. units filled with missing where necessary, but it will be panel-shaped then
- *Substring*

```
#Syntax
substr(x, start, stop)

#Example
##this will substring _starting_ at the second element; R is indexed at 1
dmulong\$Decade <- substring(dmulong\$year,2)
```

- *Merge*

```
#Syntax
merge(x, y, by = intersect(names(x), names(y)),
      by.x = by, by.y = by, all = FALSE, all.x = all, all.y = all,
      sort = TRUE, suffixes = c(".x",".y"),
      incomparables = NULL, ...)

#Example 1 - Merge two dataframes; you can specify keep using/master/both by working with all/all.x/all
df <- merge(df, dmu_long, allby=c("Decade","DMU"))
```

- *Upper/Lower Case*

```
df\$county <- toupper(df\$county)
#Same thing but use *tolower*
```

- *Subsetting Data by Vars and Obs*

```r
#1) Subset DataFrame by Decade and Create New DF
deer83 <-subset(deeryearIR, Decade=="1983")

#2) Same subset based on "or"
pre <- subset(df, (Decade=="1983" | Decade=="1996"))

#3) Subset so you don't have NA's in whole DF
df <- subset(df, !is.na(df$State))

#4) Subset so you don't have NA's based on NA's in a single var
df <- subset(df, subset= !is.na(df$Model))

#5) Drop specific observations (and create new dataframe)
df <- df[!(df$zip=="30038" & df$statelong=="Georgia"),]

#6) Subset - take only vars that you want
zipdf <- subset(df, select=c(zip,statelong,Households, Income,latitude,longitude,Population.2010,
                             zipSoftware, zipModelA:zipModelU, zipsales:zipStateGov,White:Latino))

#7) Subset so you don't have Inf
zip <- subset(zipdf, zipdf$salesdensity!=Inf)

#8) Take out obs that are '0'
models <- models[which(models$salesModel!=0),]
```

- *Changing Values of Specific Obs*

```r
#Example 1: Drop Where a var ==Inf
df$prop[df$prop=="Inf"] <- 0

#Example 2: Change two values of vars based on another var
df$palNew<-NULL
df$palNew[df$Palatability=="1" | df$Palatability=="2"] <-"low"
df$palNew[df$Palatability=="3" | df$Palatability=="4"] <-"high"

#Example 3: Replace value based on two criterion
df$DeerDensity[df$DMU=="48" & df$Decade=="1983"] <- deer83[[2]]

#Example 4: Replacing with value
df$countFortune[df$Classification=="Fortune 500"] <-1
```

- *Dealing with Duplicates*

```r
#Example 1: Create subset with unduplicated data based on specific columns
sub <- df[!duplicated(df[c(1:2,8)]),]

#Example 2: Create subset with unduplicated data based on specific var names
BA <-df[!duplicated(df[c("Decade","DMU","Species","BAgroup","county","SmallAgg","MedAgg","LargeAgg", "S

#Example 3: ~duplicates drop, force in Stata
zipdf <- unique(zipdf) #keeps only unique; may need to create subset so this works the way you want to;
```

- *Aggregate Data by a Var*

```
#Example 1: Aggregate by Decade on a Subset, take the mean
deeryearIR <- aggregate(DeerDensity~Decade, data=subset(df, Owner=="IR"), FUN=mean)

#Example 2: Aggregate Based on Two Key Vars (~by group in Stata)
subAgg <- aggregate(sub$deerNum, by=list(sub$Decade, sub$county), FUN=sum)

#Example 3: Aggregate Based on Three Key Vars
aggSmall <- aggregate(df$Small, by=list(df$Decade, df$DMU, df$Species, df$BAgroup), FUN=sum)

#Merge Aggregated Back Into DF Based on Key Var
df <- merge(df, aggSmall, by.x=c("Decade", "DMU","Species","BAgroup"), by.y=c("Group.1","Group.2","Grou
df$SmallAgg <-df$x #rename aggregated var in new dataframe from x
df$x <-NULL #get rid of old var

#Merge on two different var names and keep all from master
df <- merge(df, statenames, by.x="state", by.y="Abbreviation", all.x=TRUE)
```

- *tapply/lapply/sapply*

```
#By county (county=long and unique), Sum the area by county
areaCounty <- tapply(df$Area, df$county, FUN=sum)

#Merge this summed var back into DF
df$areaCounty <- merge(df, areaCounty, allby="county")

#Find means and sd by group
tapply(dfnew$trans, dfnew$trt, FUN=mean)
tapply(dfnew$trans, dfnew$trt, FUN=sd)
```

- *Dates in R*

```
df$test <- as.Date(df$Date, "%m/%d/%y")
#For 12/01/1990 format
```

- *Regular Expressions*

```
df$year <- as.factor(sub(".*/.*/","",df$Date))
#This takes somethinfr from form "12/01/1990" and takes everything after the second "/"
# * is a wildcard for numbers(?)
```

- *Binning a variable into equal groups and specifying levels*

```
#1) Binning
df$deerGroup <- cut(df$DeerDensity,3)
#Be careful not to have too many groups - overspecification

#2) Specify levels
zip$densityBlack <- factor(zip$densityBlack, levels=c("lo","med","hi"))
## should do this with all factor variables because once you run the regression, it's going to take a le

#3) Relevel levels
df <- within(df, Population <- relevel(Population, ref="Before"))
```

- *Reshape*

```
#Example 1: Reshape Long to Wide
##This uses 'reshape'
zipNew <- reshape(zip,
                  varying=c("zipFortune","zipGov","zipNonProf","zipPersonal","zipSmallBus","zipStateGov
                  v.names="salesClassification",
                  timevar="Classification",
                  times=c("zipFortune","zipGov","zipNonProf","zipPersonal","zipSmallBus","zipStateGov")
                  direction="long")

#Example 2: Reshape Long to Wide
##This uses 'reshape2'
models<- melt(zipNew, id.vars=c("zip","statelong","Classification"),
               measure.vars = c("zipModelA","zipModelB","zipModelC","zipModelD","zipModelE","zipModelF"
                                "zipModelG","zipModelH","zipModelI","zipModelJ",
                                "zipModelK","zipModelL","zipModelM","zipModelN","zipModelO","zipModelP"
                                "zipModelQ", "zipModelR","zipModelS","zipModelT","zipModelU"),
               variable.name="Model",
               value.name="salesModel")
```

Exploratory Tricks (not included in other sections)

- *Plot regression fit*

```
plot(SmallAgg/allTrees ~ deerGroup*Palatability + Decade + BAgroup*Species + DMU, data= BA)
#this will create all plots on the different vars specified in regression; helps see what's going on
```

- *Sort/Re-order*

```
#Example 1: from smallest to largest
sales <- sales[order(sales$ZIP),]

#Example 2: From largest to smallest
sales <- sales[order(sales$ZIP, decreasing=TRUE),]
```

- *Looking at Quantiles/Modifying Quantiles*

```
quantile(df$sales)

upper.limit <- quantile(zip$zipsales)[4] + 1.5*IQR(zip$zipsales)
lower.limit <- quantile(zip$zipsales)[2] - 1.5*IQR(zip$zipsales)
```

- *Table of Top X*

```
#Find Top 20 (by salesdensityState in this case) and print also state, region, and regulation
StateSalesperCapita <- round(statepop$salesdensityState[1:20],2)
StateName <- statepop$statelong[1:20]
Region <- statepop$region[1:20]
Regulation <- statepop$reg[1:20]
printme <- cbind.data.frame(StateName, StateSalesperCapita, Region, Reg)
xtable(printme)
```

- *Correlation Matrix*

```
myvars <- c("zipsales","Households","Income","Population.2010","White",
              "Black","NativeAmerican","Asian","PacificIslander","Other","Latino","enacted_law_count","
cordata <- zip[myvars]
xtable(cor(cordata))
#Correlation matrix can only have numeric values (obvi)
```

- *Winsorizing*

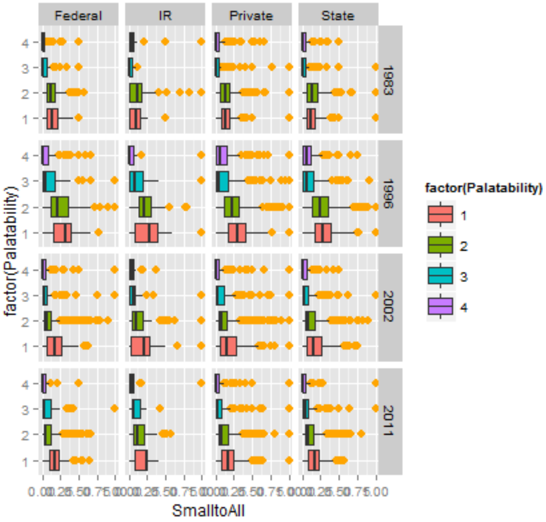
```
library(robustHD)
zip$zipsalesWin <- round(winsorize(zip$zipsales))
stateState$salesdensityStateW <- round(winsorize(stateState$salesdensityState))
```

Graphs!

```
#1) Regular boxplots
q <- ggplot(df, aes(factor(Palatability), SmalltoAll))
q + geom_boxplot(outlier.colour="orange", outlier.size=3, aes(fill=factor(Palatability))) +  coord_flip

#2) Boxplots with jittering
r <- ggplot(data=sub, aes(Concentration,DevTimeHA))
r + geom_boxplot(outlier.colour="orange", outlier.size=6, aes(fill=Concentration)) +
    geom_jitter() + coord_flip() + facet_grid(Population~.)
```

- *Faceted Boxplots*

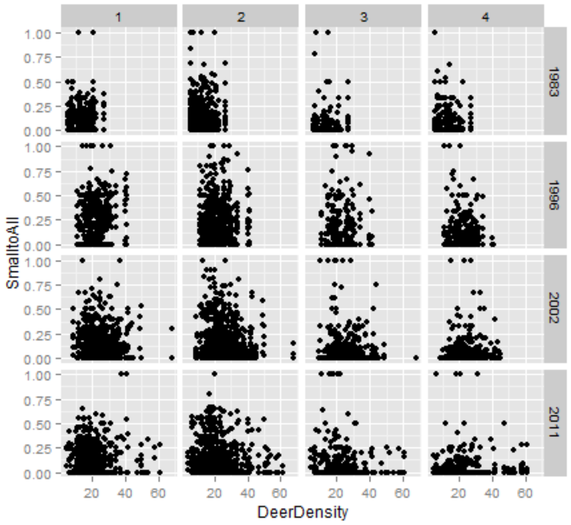
Figure 1: Faceted Boxplots



- *GeomPoint*

```
q <- ggplot(BA, aes(SmalltoAll, DeerDensity))
q + geom_point() +  coord_flip() + facet_grid(Decade ~ Palatability) #+  stat_smooth(method=lm, fullran
```

Figure 2: GeomPoint



- *Histogram*

```
#1) histogram on log scale
ggplot(df, aes(LtoSM)) + geom_histogram() + scale_x_log10()

#2) Facetted Histogram
ggplot(savesNewer, aes(Saves, fill = reg)) + geom_histogram(binwidth=1) + facet_grid(region ~ reg + Sof


#3) Bates Histogram
p <- qplot(mathkind, data=classroom, geom="histogram",
           xlab="Mathematics Score in Kindergarten",main="Kindergarten Mathematics Score",
           binwidth = 10)
p + geom_histogram(binwidth=10)
```

- *qplot*

```
qplot(Black,zipsales,  data=zip)
```

- *statsmooth*

```
p <- ggplot(zip, aes(White, zipsales))
p +  scale_y_continuous(limits=c(0,15)) +stat_smooth()
#smoothing method (function) to use, eg. lm, glm, gam, loess, rlm. For datasets with n < 1000 default i
```

Figure 3: Histogram on Log Scale

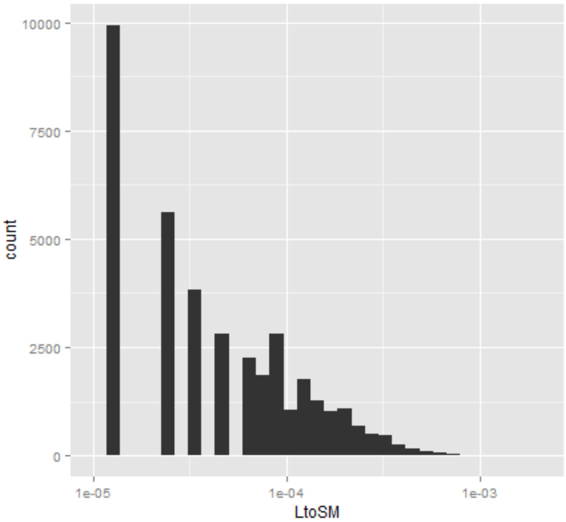
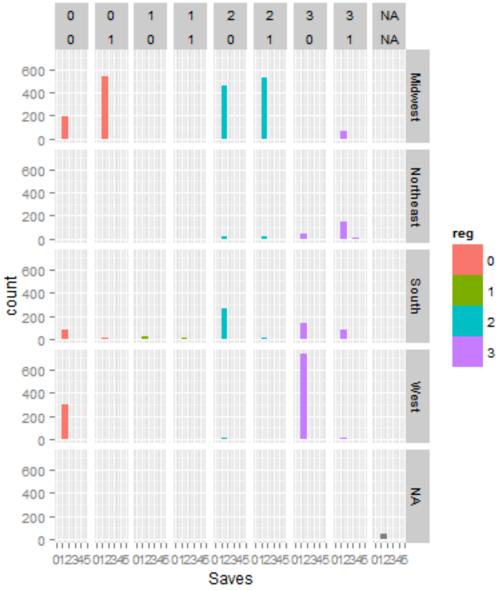


Figure 4: Facetted Histogram



- *bar chart*

```
#Classic Barchart
qplot(factor(Model), data=zip, geom="bar", fill=factor(incomeGroup))

#Barchart with x-axis labels on 45 degree angle
q <-qplot(factor(Model), data=zip, geom="bar", fill=factor(reg))
q + theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Facetted Bar Charts
ggplot(zip, aes(Model, fill=factor(reg))) + geom_bar() + facet_grid(region ~ Classification) + coord_fl
```

Figure 5: Bates Histogram

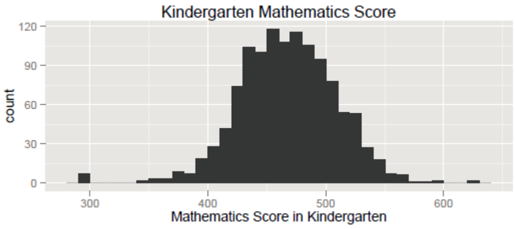


Figure 6: qplot

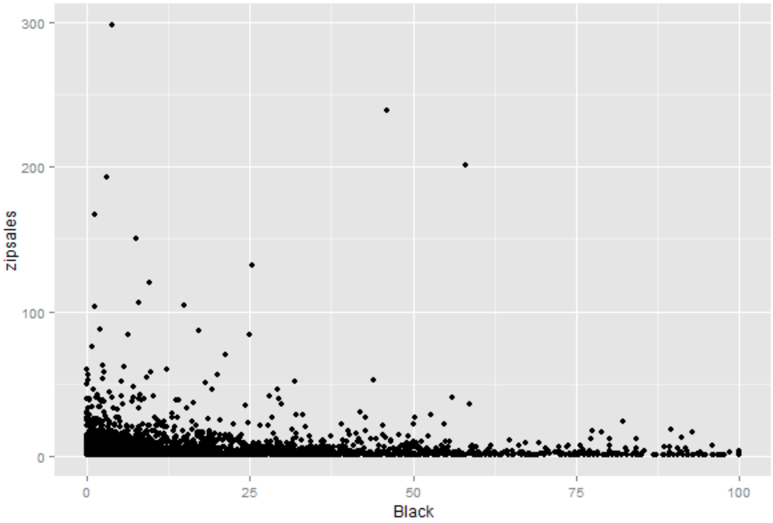


Figure 7: stat smooth

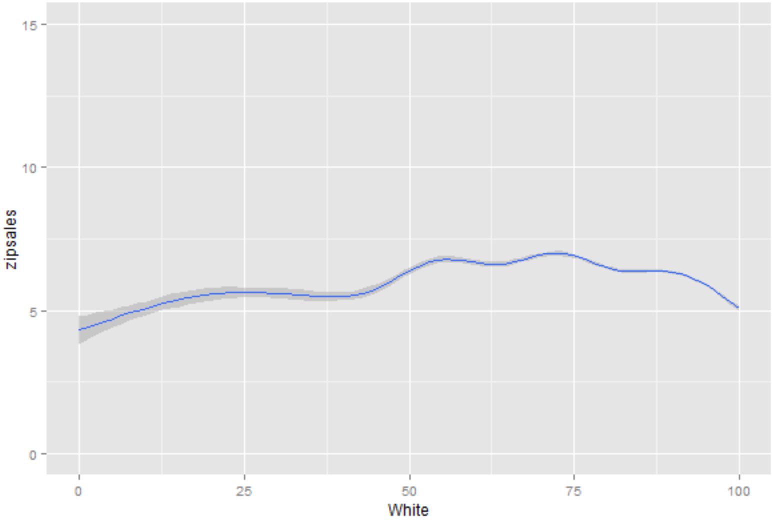
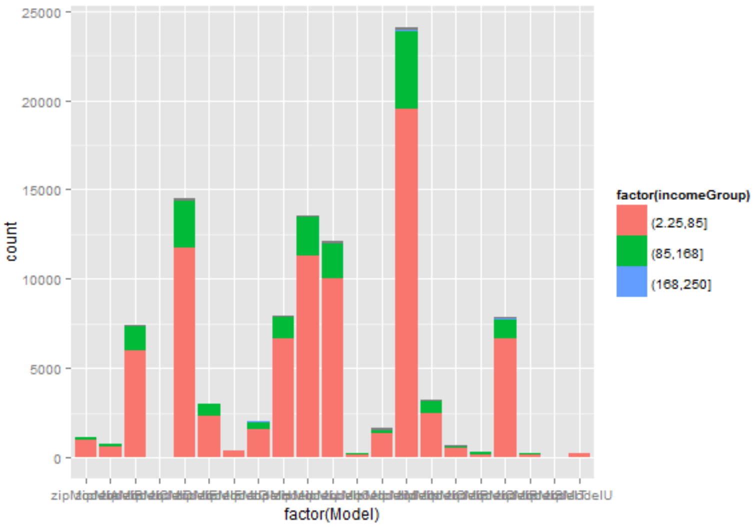


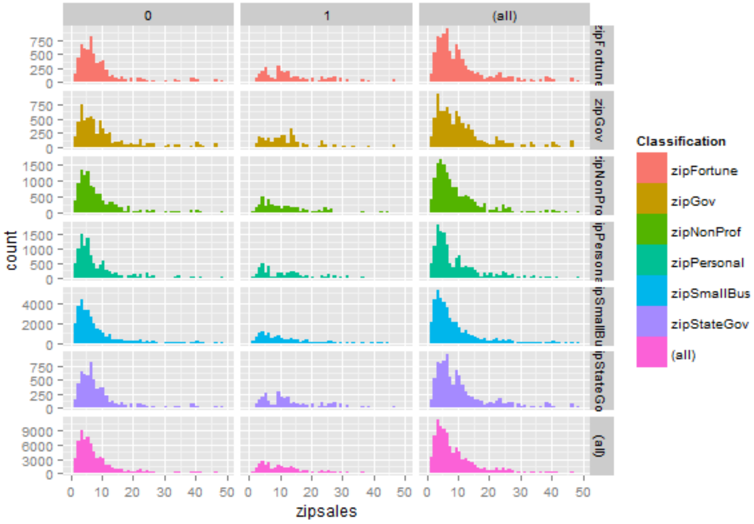
Figure 8: Bar Chart



- *Histograms*

```
#Facetted Histograms
ggplot(zip, aes(Model, fill=factor(reg))) + geom_bar() + facet_grid(region ~ Classification) + coord_fl
```

Figure 9: Facetted Histogram



- *Density Plots*

```
#1) Facetted Density Plots
ggplot(zip, aes(salesdensity, fill = Classification)) +  geom_density() +   scale_x_continuous(limits=c

#2) Density Plots with 2 Factors - color
p <- qplot(mathkind, data=classroom, geom="density",
        linetype=sex, color=sex,xlab="Mathematics Score in Kindergarten")
p + geom_density(aes(x=mathkind)) + labs(title="'Mathkind' by 'Sex'")
```

Figure 10: Facetted Density Plots

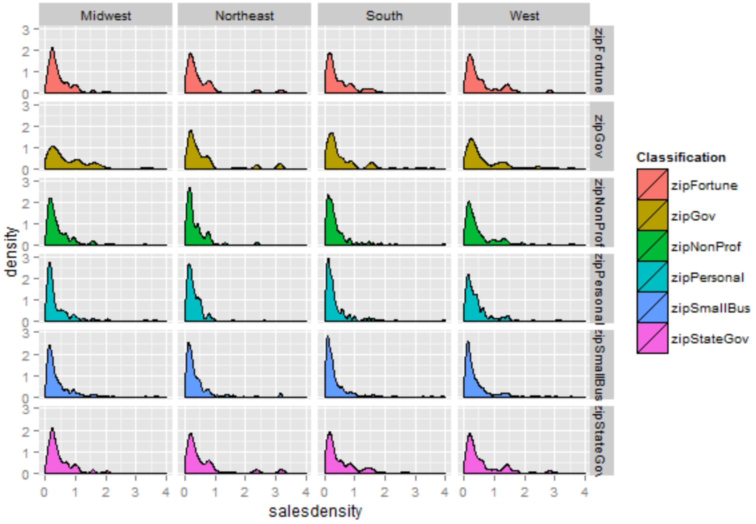
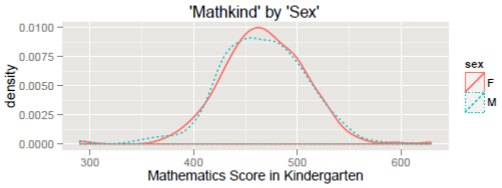


Figure 11: Density Plot - 2 Colors, 2 Factors



- *ScatterPlots*

```
#Scatter plot with point-wise confidence intervals
p <- qplot(mathgain, mathkind, data=classroom, geom=c("smooth"),
        method="loess", xlab="Mathematics Gain for Students in Sample",
        ylab="Mathematics Score in Kindergart.",
        main="Scatterplot of 'Mathgain' vs. 'Mathkind'")
p + geom_point() + stat_smooth(se=TRUE) +
  geom_smooth(method="loess") +
  labs(title="'Mathgain' vs. 'Mathkind'")

#ScatterPlot with Regression Line (and no confidence bands)
p <- ggplot(classroom, aes(x=mathgain,y=mathkind),
         geom_smooth(method="gam", formula = mathkind~mathgain),
            stat_smooth(se=FALSE),
            xlab="Mathematics Gain for Students in Sample",
            ylab="Mathematics Score in Kindergarten")
p + geom_point() + stat_smooth(se=FALSE) +
  labs(title="Scatterplot of 'Mathgain' vs. 'Mathkind'") +
  geom_smooth(method= "gam")

#Log Scatter Plot with GLM Smoother
v <- ggplot(ufc, aes(x=log(Height), y = log(Dbh))) + geom_point()
v + stat_smooth(se=TRUE) + geom_smooth(method="glm", formula = log(Dbh)~log(Height))
```

Figure 12: Scatter Plot with COnfidence Bands

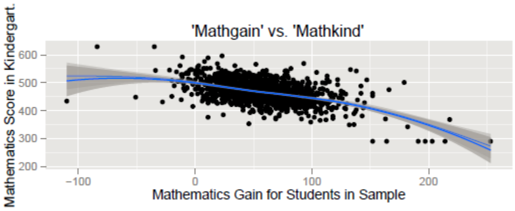


Figure 13: Scatter Plot with Regression Line

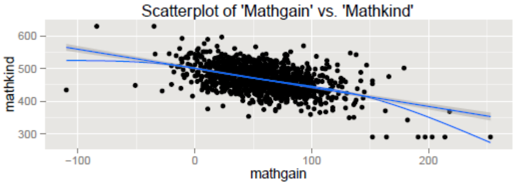
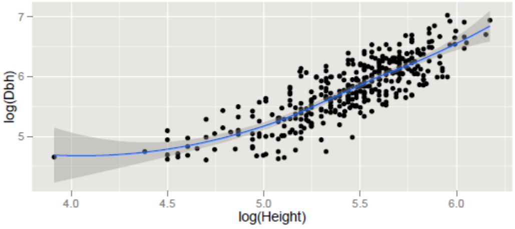


Figure 14: Log Scatter Plot with GLM Smoother



- *Dot Plot*

```
#Example 1
eleven <- subset(classroom, schoolid == "11")
p <- ggplot(eleven, aes(x=classid,y=mathgain),
            xlab="Classroom Identifier ",
            ylab="Gain in Mathematics Score (from Kindergarten to First Grade")
p + geom_point() + labs(title="'Mathgain' by 'Classid' for SchoolID #11")
```

```
#Example 2
ggplot(ufc, aes(x=Tree, y = Dbh)) + geom_point() + facet_wrap(~ Species)
```

Figure 15: Dot Plot

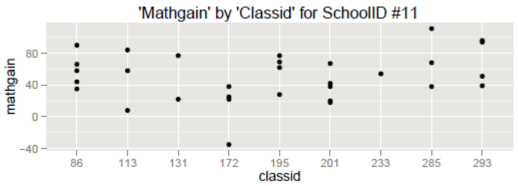
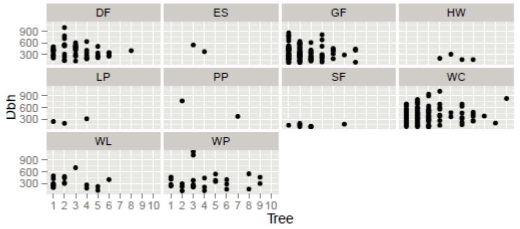


Figure 16: Dot Plot Facets



GGMAP

- *Getting the Map*

```
map <- get_map(location='united states', zoom=4, maptype='toner')
ggmap(map) + geom_point(aes(x=longitude, y= latitude, size=(zipsales)), data=zip, alpha=0.5)
#I supplied the long/lat here from my dataset
```

- *Plotting points*

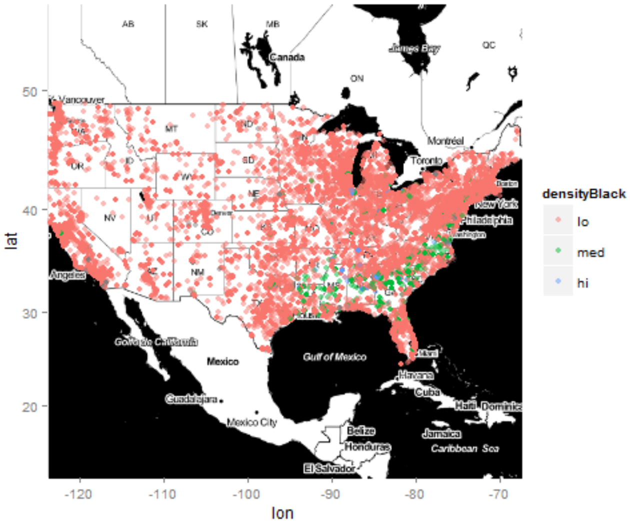
```
#1) Basic Map
ggmap(map) + geom_point(aes(x=longitude, y= latitude, color=densityBlack, fill=densityBlack), data=zip,

#2) Facetted Map (with colours specified)
ggmap(map) + geom_point(aes(x=longitude, y= latitude,  colour=zipsales), size = 3, data=zip, alpha=0.5)
```

- *Keep Unit Aspect Ratio*

```
+ theme(aspect.ratio = 1)
```

- *Change Color Gradient and Point Size*

Figure 17: Sample Map



```r
#1) Change color gradient Scale and specify which will be low
+ scale_colour_gradient(limits=c(3,4), low="red")

#2) Change with color brewer
+ scale_colour_brewer(type="div",palette="Set1")

#3) Point size (size=3)
ggmap(map) + geom_point(aes(x=longitude, y= latitude, color=Saves), data=savesNewer, colour="red", size
```

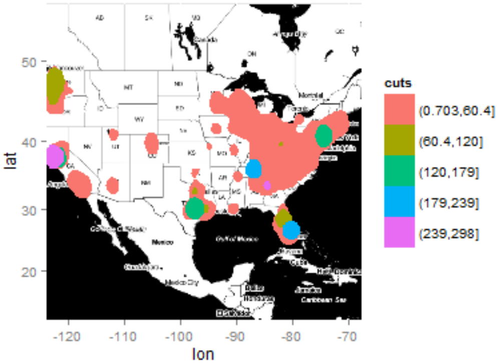
- *density map*

```r
zip$cuts <- cut(zip$zipsales, breaks=5)
ggmap(map) + stat_density2d(aes(x=longitude, y= latitude, fill=cuts),
                            data=zip,h=3 ,geom="polygon")
```

Maps with Spatial Polygons

```r
statepop$name <- tolower(statepop$statelong)
statepop <- statepop[order(statepop$name),] # reorder alpha
statepop <- subset(statepop, (name!="hawaii" & name!="alaska"))
us <- map("state", plot=FALSE, fill=TRUE)
us.ids <- sapply(strsplit(us$names,":"), function(x) x[1])
us.sp <- map2SpatialPolygons(us, us.ids, CRS("+proj=longlat + datum=wgs84"))
#stateNew <- state[c(1:20,20,20,23,23,25:34,34,34,34,38,38,38,41:53,53,53,56,56,56,56,56,61:63),]
IDs <- match(statepop$name, row.names(us.sp))
us.spNew <- us.sp[IDs]
row.names(statepop) <- row.names(us.spNew) #these need to match
us.spdf <- SpatialPolygonsDataFrame(us.spNew, data=as.data.frame(statepop)) # rownames of state need to
```

Figure 18: Density Map



```
#Plot
spplot(us.spdf, "reg", main="Regulation Type By State")

#Exclude Obs and Plot
exclude <- us.spdf[!(us.spdf$stateSmallBus==1888 | us.spdf$stateSmallBus==1734),]
sp5 <-spplot(exclude, "stateSmallBus",cuts=99, main="Sales to Small Businesses excl. 2 Outliers")
```

- *Plotting Multiple ggplots into a Grid*

```
library(grid)
library(automap)
vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)

plot1 <- ggplot(zip, aes(physician_required,zipsales)) + geom_boxplot(outlier.colour="orange", outlier.s
plot2 <- ggplot(zip, aes(training_required,zipsales)) + geom_boxplot(outlier.colour="orange", outlier.si
plot3 <- ggplot(zip, aes(registration_required,zipsales)) + geom_boxplot(outlier.colour="orange", outli

grid.newpage()
pushViewport(viewport(layout = grid.layout(2,2)))
print(plot1, vp=vplayout(1,1))
print(plot2, vp=vplayout(1,2))
print(plot3, vp=vplayout(2,1))
```

ggplot2 - General Tips and Tricks

- *Suppress Plotting NA's*

```
#outlier.shape=NA
r + geom_boxplot(outlier.shape=NA, aes(fill=factor(physician_required))) +  coord_flip() + scale_y_conti
```

- *Setting Limits on X/Y Coords in ggplot*

```
#for y (or x) limits
+ scale_y_continuous(limits=c(0,50))
```

- *Dealing with NA's in DF to Plot*

```
#Solution - create sub data frame without NA's

#Example: use ggplot(na.omit(subset))
subzip <- zip[c("zipsales","densityBlack", "incomeGroup")]
b <- ggplot(na.omit(subzip), aes(factor(densityBlack), zipsales))
b + geom_boxplot(aes(fill=factor(densityBlack))) + scale_y_continuous(limits=c(0,100)) + coord_flip() +
```

Models, Assumptions, and Diagnositcs

OLS/Linear Models

- *Assumptions of the Linear Model*
  - 1) Linearity assumption
  - 2) Errors are iid normal
  - 3) Predictors are assumed to be independent of each other (no collinearity)
- *Interpretation*
  - 1) Standard OLS
    - For a one unit increase in x, we expect to see a [coef on x] increase in response
  - 2) OLS with Some log-transformed coefficients
    - $log(write) + \beta_0 + \beta_1 + error$
    - The intercept of 3.89 is the log of geometric mean of write when female = 0, i.e., for males. Therefore, the exponentiated value of it is the geometric mean for the male group: exp(3.892) = 49.01. What can we say about the coefficient for female? In the log scale, it is the difference in the expected geometric means of the log of write between the female students and male students. In the original scale of the variable write, it is the ratio of the geometric mean of write for female students over the geometric mean of write for male students, exp(.1032614) = 54.34383/49.01222 = 1.11. In terms of percent change, we can say that switching from male students to female students, we expect to see about 11% increase in the geometric mean of writing scores.
    - The exponentiated coefficient $\exp(\beta_1)$ for female is the ratio of the expected geometric mean for the female students group over the expected geometric mean for the male students group, when read and math are held at some fixed value. Of course, the expected geometric means for the male and female students group will be different for different values of read and math. However, their ratio is a constant: $\exp(\beta_1)$. In our example, $\exp(\beta_1) = \exp(.114718) = 1.12$.
    - http://www.ats.ucla.edu/stat/mult_pkg/faq/general/log_transformed_regression.htm
- R Code

```
#Example 1: Reguler OLS
lmod <- lm(prop ~ DeerDensity*Palatability*Species + BA + Decade + Owner, data=df)

#Example 2: Transformation
lm <- lm(sqrt(SmalltoAll) ~ Decade+Species*BAgroup+deerGroup*palNew, data=BA)
```

OLS/Linear Models - Diagnostics

- *Residual Plots*
  - In residual plots, want to look for now pattern - pattern indicates heterogeneity
  - Need to also look for leverge points/ influential points/ outliers
  - *Cook's Distance*
    - measures the difference between the regression coefficients obtained from the full data and the regression coefficients obtained by deleting the ith observaton; the difference between the fitted values obtained from full data and fitted values obtained by deleting the ith observation.
    - a large Cook's value (C_i) indicates the point is influential

```
plot(model1)
```

GLM (general)

GLM-Binomial

- **Interpretation**
- **R Code**

```
#Example 1: Regular Binomial (0/1 or Success/Failure)
mod1 <- glm(SmalltoAll ~ deerGroup + Decade + BAgroup*Species*Shade + Palatability + DMU, family="binomi

#Example 2: Quasibinomial
mod11 <- glm(SmalltoAll ~ deerGroup*Palatability + Decade + BAgroup*Species + DMU, family="quasibinomial

#Example 3: Weighted Quasibinomial
mod12 <- glm(sqrt(SmalltoAll) ~ DeerDensity*Palatability + Decade + BAgroup*Species + DMU, family="quasi

#Example 4: Different Parameterization of Binomial + Backward Selection
model1<-glm(cbind(esoph$ncases,esoph$ncontrols)~ esoph$agegp*esoph$alcgp*esoph$tobgp,family=binomial(lin
step(model1,direction="backward")
```

- *Sample Plot for Binomial GLM*

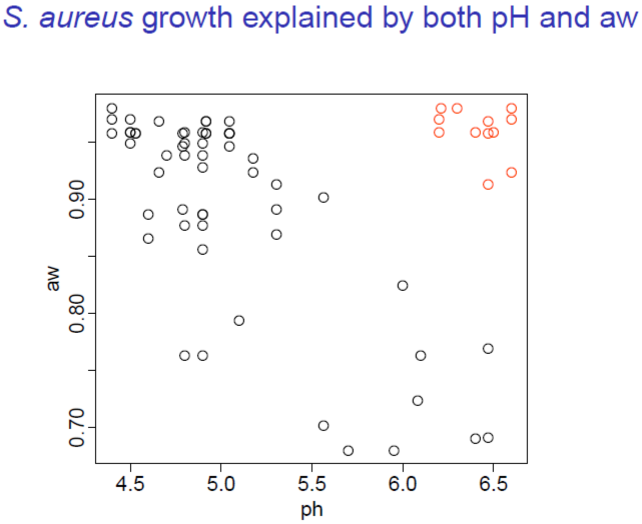
```
fit.ph = glm(growth ~ ph, family=binomial, data=rte)
# summary(fit.ph)
# ph 6.38 2.55 2.502 0.0123 *
# Residual deviance: 20.226 on 66 degrees of freedom
plot(growth ~ ph, data=rte)
mypH = seq(4,7,by=.05)
lines(mypH, predict(fit.ph, type="response", list(ph=mypH)))
```

- *Trouble-Shooting Binomial GLM*

```
fit.awph = glm(growth ~ aw+ph, family=binomial, data=rte)
#1: In glm.fit(x=X, y=Y, weights=weights, start=start, etastart=algorithm did not converge
#2: In glm.fit(x=X, y=Y, weights=weights, start=start, etastart=fitted probabilities numerically 0 or 1
###This will also give you p-values ==1 in regression summary

#1) Plot
growthcolor = rep(NA, 68)
growthcolor[rte$growth==0] = "black"
growthcolor[rte$growth==1] = "orangered"
plot(aw~ph, data=rte, col=growthcolor)
#In this case, can use bias-reduction algorithm
```

Interpretation

Figure 19: Troubleshooting Binomial GLM



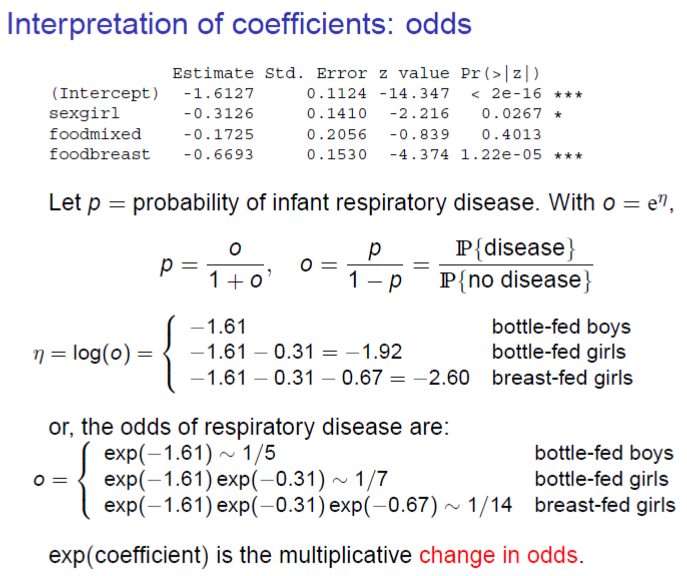
GLM-Logistics

- *Interpretation*
  - *Prediction*:

```
#In Binomial logistic reg
predict(fit.raw, type="link")
predict(fit.raw, type="response")
```

```
* Plot Logit Curve
```

```
layout(matrix(1:2,2,1))
my.etas = seq(-8,8, by=.01)
my.prob = 1/(1+exp(-my.etas))
```



Interpretation of coefficients: odds

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.6127     0.1124 -14.347  < 2e-16 ***
sexgirl      -0.3126     0.1410  -2.216   0.0267 *
foodmixed    -0.1725     0.2056  -0.839   0.4013
foodbreast   -0.6693     0.1530  -4.374 1.22e-05 ***
```

Let p = probability of infant respiratory disease. With $o = e^{\eta}$,

$$p = \frac{o}{1+o}, \quad o = \frac{p}{1-p} = \frac{\mathbb{P}\{\text{disease}\}}{\mathbb{P}\{\text{no disease}\}}$$

$$\eta = \log(o) = \begin{cases} -1.61 & \text{bottle-fed boys} \\ -1.61 - 0.31 = -1.92 & \text{bottle-fed girls} \\ -1.61 - 0.31 - 0.67 = -2.60 & \text{breast-fed girls} \end{cases}$$

or, the odds of respiratory disease are:

$$o = \begin{cases} \exp(-1.61) \sim 1/5 & \text{bottle-fed boys} \\ \exp(-1.61)\exp(-0.31) \sim 1/7 & \text{bottle-fed girls} \\ \exp(-1.61)\exp(-0.31)\exp(-0.67) \sim 1/14 & \text{breast-fed girls} \end{cases}$$

exp(coefficient) is the multiplicative change in odds.

```r
plot(my.etas, my.prob, type="l", bty="n",
xlab="linear predictor: log-odds eta",
ylab="probability of 'success'")
abline(h=0); abline(h=1);
lines(c(-10,0),c(.5,.5), lty=2)
lines(c(0,0),c(0,.5), lty=2)
my.conc = seq(0,2.5,by=.05)
my.etas = -6.469 + 5.567 * my.conc
my.prob = 1/(1+exp(-my.etas))
plot(my.conc, my.prob, type="l", bty="n", adj=1,
xlab="", ylab="prob. no movement")
mtext("concentration", side=1, line=0.4)
mtext("eta", side=1, line=2.4)
mtext("-6.5\n(intercept)",side=1,at=0, line=4)
mtext("-0.9\n(-6.5+5.6)",side=1,at=1, line=4)
conc.5 = (0-(-6.469))/5.567
mtext("0",side=1,at=conc.5, line=3)
mtext("4.7\n(-6.5+2*5.6)",side=1,at=2, line=4)
lines(c(-1,conc.5),c(.5,.5), lty=2)
lines(c(conc.5,conc.5),c(0,.5), lty=2)
```

```
* Plot Probability of Response vs. Factor
```

```r
#Example 1 - Plot Binomial Response 0/1
plot(movement ~ conc, data=dat)
plot(movement ~ as.factor(conc), data=dat)
plot(nomove ~ conc, data=dat)
plot(jitter(nomove) ~ conc, data=dat)
plot(jitter(nomove,amount=.02) ~ conc, data=dat)
myconc = seq(0.8,2.5,by=.05)
lines(myconc, predict(fit.raw, type="response",
list(conc = myconc)))
```

```
#Example 2 - Plot Proportional Response (between 0 and 1)
plot(prop ~ conc, data=dat2)
lines(myconc, predict(fit.raw, type="response",
list(conc=myconc))


#Both are logistic regressions
```

GLM-Poisson

- R Code

```
m1 <- glm(Large ~ Decade + Species + DeerDensity + BA, data=df, family=poisson(link="log") )
```

http://www.ats.ucla.edu/stat/r/dae/poissonreg.htm

- *Interpretation*
    - *Get Robust Standard Errors and Calculate P-Val*

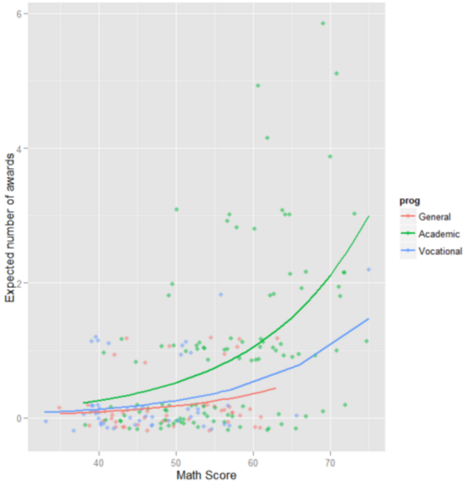
```
#use package 'sandwich'
cov.m1 <- vcovHC(m1, type="HC0")
std.err <- sqrt(diag(cov.m1))
r.est <- cbind(Estimate= coef(m1), "Robust SE" = std.err,
"Pr(>|z|)" = 2 * pnorm(abs(coef(m1)/std.err), lower.tail=FALSE),
LL = coef(m1) - 1.96 * std.err,
UL = coef(m1) + 1.96 * std.err)

r.est
```

```
* _Poisson Plot_
```

```
## calculate and store predicted values
p$phat <- predict(m1, type="response")

## order by program and then by math
p <- p[with(p, order(prog, math)), ]

## create the plot
ggplot(p, aes(x = math, y = phat, colour = prog)) +
  geom_point(aes(y = num_awards), alpha=.5, position=position_jitter(h=.2)) +
  geom_line(size = 1) +
  labs(x = "Math Score", y = "Expected number of awards")
```

```
## calculate and store predicted values
p$phat <- predict(m1, type="response")

## order by program and then by math
p <- p[with(p, order(prog, math)), ]
```

Figure 20: Poisson Plot



```r
## create the plot
ggplot(p, aes(x = math, y = phat, colour = prog)) +
  geom_point(aes(y = num_awards), alpha=.5, position=position_jitter(h=.2)) +
  geom_line(size = 1) +
  labs(x = "Math Score", y = "Expected number of awards")
```

GLM-Negative Binomial

- *Interpretation*
- R Code

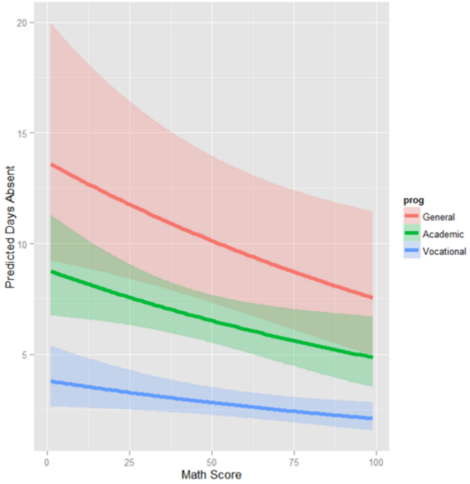
```r
m3 <- glm.nb(zipsales ~ Classification +Income*Population.2010 + training_required + registration_requi
```

- *Negative Binomial Plot*

```r
#1) look at predicted counts for each value of prog while holding math at its mean
newdata1 <- data.frame(math = mean(dat$math),
prog = factor(1:3, levels = 1:3, labels = levels(dat$prog)))
newdata1$phat <- predict(m1, newdata1, type = "response")

#2) obtain the mean predicted number of events for values of math across its entire range for each leve
newdata2 <- data.frame(
  math = rep(seq(from = min(dat$math), to = max(dat$math), length.out = 100), 3),
  prog = factor(rep(1:3, each = 100), levels = 1:3, labels =
  levels(dat$prog)))

newdata2 <- cbind(newdata2, predict(m1, newdata2, type = "link", se.fit=TRUE))
newdata2 <- within(newdata2, {
  DaysAbsent <- exp(fit)
  LL <- exp(fit - 1.96 * se.fit)
  UL <- exp(fit + 1.96 * se.fit)
})
```

```
ggplot(newdata2, aes(math, DaysAbsent)) +
  geom_ribbon(aes(ymin = LL, ymax = UL, fill = prog), alpha = .25) +
  geom_line(aes(colour = prog), size = 2) +
  labs(x = "Math Score", y = "Predicted Days Absent")
```

Figure 21: Poisson Plot



The graph shows the expected count across the range of math scores, for each type of program along with 95 percent confidence intervals. Note that the lines are not straight because this is a log linear model, and what is plotted are the expected values, not the log of the expected values.

Zero-Inflated Models

- *Zero-Inflated Poisson*
  - R Code

```
m3 <- zeroinfl(Large~ Decade + DeerDensity +BA*Species, data=df, dist="poisson", link="log", maxit=1000
#here, I also specify max number of iterations after had error with optim not converging
```

GLM - Diagnostics (general)

- Chi-squared Diagnostics

```
#Example 1: Regular GLM
pchisq(2 * (logLik(m2) - logLik(mNull)), df = 15, lower.tail = FALSE)
###
#If this is significant,that means there is a significant difference between the two models and implies

#Example 2: Quasi-models
sum(residuals(mod11, type="pearson")^2)
deviance(mod11)
1-pchisq(deviance(mod11), df.residual(mod11))
###Need to do this with quasi-models because they don't have a likelihood
```

```
#Example 3: Checking One Model within itself
pchisq(deviance(model6), + df.residual(model6),lower=FALSE)
##"best" model can also be determined by lowest p.value (via above method) and lowest AIC

#Example 4: Evidence for lack of fit
summary(fit)
#... Null deviance: 26.37529 on 5 degrees of freedom
#Residual deviance: 0.72192 on 2 degrees of freedom
pchisq(0.72192, df=2, lower.tail=F)
[1] 0.6970069
### no sign of lack of fit (this works because n_i is big)
###When n_i are really small, chi-squared approximation is really bad

#Example 5: Chi-squared for nested models
##Mod_reduced - Mod_full ~ Chi-squared (with d degrees of freedom (difference between the models))
summary(fit1)
#... glm(formula = RunoffEvent ~ Precipfamily = binomial, data = runoff)
#... Residual deviance: 148.13 on 229 degrees of freedom
summary(fit2)
#... glm(formula = RunoffEvent ~ Precip + MaxIntensity10,family = binomial, data = runoff)
#... Residual deviance: 116.11 on 228 degrees of freedom
pchisq(148.13-116.11, df=229-228, lower.tail=F)
#[1] 1.525e-08
anova(fit1, fit2, test="Chisq")
#Analysis of Deviance Table
#Model 1: RunoffEvent ~ Precip
#Model 2: RunoffEvent ~ Precip + MaxIntensity10
#Resid. Df Resid. Dev Df Deviance P(>|Chi|)
#1 229 148.129
#2 228 116.106 1 32.023 1.524e-08
```

- *Drop1/Add1*

```
drop1(fit.awph, test="Chisq")

add1(lrfit, ~.^2,test="Chisq")
```

- ANOVA with Chi-Squared

```
anova(mod11, mod10, test="Chisq")
```

- Looking at Residuals

```
##Diagnositcs from Venables and Ripley
rs <- resid(m2a, type="deviance")
plot(predict(m2a), rs, xlab="Linear predictors", ylab = "Deviance residuals")
abline(h=0, lty=2)
qqnorm(rs, ylab="Deviance residuals")
qqline(rs)
```

- Confidence Intervals

```
#Regular Confidence Intervals
est <- cbind(Estimate = coef(model), confint(model))
xtable(est)


#CI with quasi-models
confint.default(mod11)
####need to use Standard errors for confint because used quasibinomial and we dont have a log-likelihoo

#Confidence Intervals with Logit GLM
##Need to exponentiate
summary(fit)

#Estimate Std. Error z value Pr(>|z|)
#sexgirl -0.3126 0.1410 -2.216 0.0267 *
#foodmixed -0.1725 0.2056 -0.839 0.4013
#foodbreast -0.6693 0.1530 -4.374 1.22e-05 ***

# CI for breastfeeding effect:
c(-0.6693 - 2*0.1530, -0.6693 + 2*0.1530)
#[1] -0.9753 -0.3633

# CI for change in odds due to breastfeeding:
exp(c(-0.6693 - 2*0.1530, -0.6693 + 2*0.1530))
#[1] 0.3770792 0.6953778
```

- *Halfnorm*

```
halfnorm(residuals(mod11))
#These should look normal-ish-ish
```

- *Check for Overdispersion*

```
sigma2_2 <- sum(residuals(m2, type="pearson")^2/m2$df.res)
```

- *Diagnostic Plots for Mixed Models GLM*

```
#1) Plots from sjPlot Package
#From package(sjPlot)
plot(m2)
sjp.glmer(m2, type="fe.cor")
sjp.glmer(m2, type="re.qq") #this looks good!
sjp.glmer(m2, type="ri.pc", facet.grid=TRUE)
VarCorr(m2)

#By default, this function plots odds ratios (exponentiated coefficients) with confidence intervalls of


#2) Plotting residuals vs link/response
plot(residuals(m4) ~ predict(m4, type="response"))
plot(residuals(m4) ~ predict(m4, type="link"))
#If we see a linear relationship suggesting that no further transformation is needed
```

- *Getting Standard Errors from Mixed Model GLM*

```
#Example 1
m3_a <- glmer(Hatched ~ Concentration*Population + (1|Clutch)  +Sex, data=df, family=binomial) #model -
se <- sqrt(diag(vcov(m3_a)))
tab <- cbind(Est=fixef(m3_a), LL=fixef(m3_a) - 1.96*se, UL=fixef(m3_a)+1.96*se)
exp(tab)

#Example 2 (sigma_e^2)
lm_anova <- lm(weight~bull, data=df)
resid_error <-sqrt(deviance(lm_anova)/df.residual(lm_anova))
#1-alpha confidence interval for sigma_e (error variance)
alpha <- 0.1
sigma_e_CI <- print(c(sse/qchisq((1-alpha/2),20), sse/qchisq((alpha/2), 20)))

#Example 2 (sigma_a^2)
ssb <-(var(df$weight)*(24-1)) - sse
rho <-1-sqrt(1-alpha)
c<- print(1/6*(ssb/qchisq(1-rho/2,3) - sse/qchisq(1-alpha/2,20)))
d <- print(1/6*(ssb/qchisq(rho/2,3) - sse/qchisq(rho/2, 20)))

sigma_a_CI <- print(c(c,d))
###Reference:
#lm_fix <-lm(weight~1, data=df)
#lm_random <- lmer(weight~1+(1|bull), data=df, REML=FALSE) #after chi-squared test between the two mode
```

- *Estimate Variance Components of Mixed MOdel*

```
#Store ANOVA results
anova1 <- anova(reg1)

#Extract the mean square errors from the ANOVA table
ms_a <-anova1$"Mean Sq"[1]
ms_b <-anova1$"Mean Sq"[2]
ms_ab <-anova1$"Mean Sq"[3]
ms_error <-anova1$"Mean Sq"[4]

#Set a,b, n parameters
a <- length(levels(chol$patient)) #levels of patient factor variable
b <- length(levels(chol$run)) #levels of run variable
n <- 2 #number of runs per patient

#Define variance components
sigma2 <- ms_error
sigma2_ab <- (ms_ab - ms_error)/n
sigma2_a <- (ms_a - ms_ab)/(n*b)
sigma2_b <- (ms_b - ms_ab)/(n*a)

#Call the newly-defined variance components
sigma2
sigma2_ab
sigma2_a
sigma2_b
```

```r
#Run F-tests to test for significance
F_a <- ms_a/ms_ab #test main effect of A
1-pf(F_a, a-1, (a-1)*(b-1))

F_b <- ms_b/ms_ab #test main effect of B
1-pf(F_b, b-1, (a-1)*(b-1))
```

Multinomial Regression

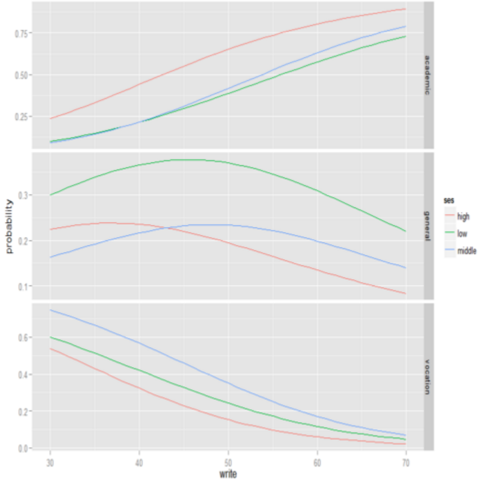
- Multinomial logistic regression is used to model nominal outcome variables, in which the log odds of the outcomes are modeled as a linear combination of the predictor variables.
- Faraway pg. 98
- http://www.ats.ucla.edu/stat/r/dae/mlogit.htm
- Contingency Tables/Multinomial - Faraway pg. 72
- *R Code*

```r
ml$prog2 <- relevel(ml$prog, ref = "academic")
test <- multinom(prog2 ~ ses + write, data = ml)
z <- summary(test)$coefficients/summary(test)$standard.errors

# 2-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2

## extract the coefficients from the model and exponentiate
exp(coef(test))

#Predict
head(pp <- fitted(test))
```

- *Multinomial Plot*

```r
## melt data set to long for ggplot2
lpp <- melt(pp.write, id.vars = c("ses", "write"), value.name = "probability")
head(lpp)  # view first few rows
##   ses write variable probability
## 1 low    30 academic     0.09844
## 2 low    31 academic     0.10717
## 3 low    32 academic     0.11650
## 4 low    33 academic     0.12646
## 5 low    34 academic     0.13705
## 6 low    35 academic     0.14828

## plot predicted probabilities across write values for each level of ses
## facetted by program type
ggplot(lpp, aes(x = write, y = probability, colour = ses)) + geom_line() + facet_grid(variable ~
    ., scales = "free")
```

Figure 22: Poisson Plot



Multinomial Diagnositcs

Mixed Models

- R Code

```
#from package lme4

#Example 1: Regular MM with 1 RE
gmod <- lmer(LtoAll ~ Decade + DeerDensity +Palatability*BA*Species + Owner + (1|county), data = df)
#Where county is a random effect

#Example 2: Nested RE
m6 <- lmer(SmalltoAll~Decade + deerGroup +BA*Species + palNew +(1|county:DMU:Owner), data=BA)
#where Owner is nested in DMU which is nested in county
```

Mixed Models - Diagnostics

- *Compare Mixed Model to Linear Model*

```
lm_fix <-lm(weight~1, data=df)
lm_random <- lmer(weight~1+(1|bull), data=df, REML=FALSE)
print(l_lm_fix <- logLik(lm_fix))
print(l_lm_random <- logLik(lm_random))
-2*(l_lm_fix - l_lm_random)
1- pchisq(1.196221, df=1)
1-pchisq(0.95, df=1)
#p-val is 0.27 - not enough evidence to reject the null
```

Mixed Effects Logistic Regression

- R Code

```
m5<- glmer(Large ~ Decade + Species*DeerDensity +Palatability*BA + (1|Plot),data=df,family=poisson )

#need packages
library("pscl")
library("boot")
```

Testing

- *T test*

```
#Women testing positive
##Do women who test positive have higher diastolic blood pressures and is the diastolic blood pressure
x <- pima1$diastolic[pima1$test==1]
y <- pima1$diastolic[pima1$test==0]

t.test(x,y, alternative="greater")

##we reject the null hypothesis in favor of the alternative (where the null is that is that the differen
```

- *Pairwise T test*

```
# In ANOVA, it tells you say a factor is significant, but you don't know which level sof the factor are

#1) Pairwise T-test
##Calculate pairwise comparisons between group levels with corrections for multiple testing

pairwise.t.test(x, g, p.adjust.method = p.adjust.methods,
                pool.sd = !paired, paired = FALSE,
                alternative = c("two.sided", "less", "greater"),
                ...)
#Example
pairwise.t.test(df$DevTimeHA, df$Concentration, p.adj="none")
#p.adjust allows for Bonferroni (more conservative); Benjamini & Hocherg ("hochberg")
## Bonferonni: designed to give strong control of the family-wise error rate.

##Hochberg: control the false discovery rate, the expected proportion of false discoveries amongst the
```

- *Test for Equal Means within Group*

```
# H0: mu1 = mu2 = m3 =...= mu11
# HA: at least one mean is not equal to the others
summary(aov(trans~as.factor(trt),data=dfnew))
pairwise.t.test(dfnew$trans, as.factor(dfnew$trt), p.adj="none")
```

- *Tukey*

```
#1) For one factor
a1 <- aov(write ~ ses)
TukeyHSD(a1)
```

```
#2) For two (or more factors)
a2 <- aov(write ~ ses + female)
TukeyHSD(a2, "ses")
```

- *Excluding Outliers from Model*

```
model7 <- glm(cbind(esoph$ncases,esoph$ncontrols)~ esoph$agegp+unclass(esoph$alcgp)+unclass(esoph$tobgp
#exclude obs # 67 & 13
```

- *Effect of Moving Up One Category (from GLM)*

```
#What is the predicted effect of moving one category higher in alcohol consumption?
model4<-glm(cbind(esoph$ncases,esoph$ncontrols)~ esoph$agegp+unclass(esoph$alcgp)+unclass(esoph$tobgp),
model4$coefficients

x0 <-c(1,0,0,0,0,0,0,0)
eta0 <- sum(x0*coef(model4))

x1 <-c(1,0,0,0,0,0,1,0)
eta1 <- sum(x1*coef(model4))

effect <-eta1-eta0
effect

exp((model4$coefficients[7]))
#The predicted effect of moving one category up in alcohol consumption will be 0.6530835 on the raw sca

#95% CI
exp(confint(model4))[7,]
```

Latex Code

```
\begin{figure}[ht!]
    \caption{Regulation Type by State}
    \centering
        \includegraphics[scale=0.7]
{reg_state_map_min.png}
\\[1.5pt]
\begin{footnotesize}
Footnote here
\end{footnotesize}
\end{figure}
```

Neat Random Tricks

- *Calculating Estimates and Standard Errors With Coefficients*

```
sum(mod11$coefficients[1:8]) - sum(mod11$coefficients[9:79], na.rm=TRUE) + sum(mod11$coefficients[80:10

sqrt(vcov(mod11)[1:8]))
```

- *Using the 'Effects' Package*

```
test <- effect("deerGroup*Palatability", mod11)
exp(summary(test)) #exponentiates effects
#library(effect)
```

- *Using texreg Package for Publication-Quality Model Output Comparisons*

```
screenreg(list(mod11Fed, mod11IR, mod11Priv, mod11Stat)) #this makes output pretty on the screen
texreg(list(mod11Fed, mod11IR, mod11Priv, mod11Stat)) # this makes output pretty in latex
```

- *For Loop to check rownumber=childid*

```
#Before removing the "childid" variable, we check that it is indeed the same as the the row number.

classroom$row <-rownames(classroom)
classroom$row <- as.factor(classroom$row)

for (i in 1:length(classroom$row)){
  if (classroom$row[i]!=classroom$childid[i]) {
    print(classroom$row)
  }
}
```

- *Make Dataframe by hand*

```
ratio <-c(97,83,85,64,52,48,96,87,84,72,56,58,92,78,78,63,44,49,95,81,79,74,50,53)
nitro <-factor(rep(c(1,1,2,3,4,4), times=4, levels=c(1,2,3,4)))
irrig <-factor(rep(c("N","Y","N","N","N","Y"), times=4, levels=c("Y","N")))
grass <-cbind.data.frame(ratio,nitro,irrig)
```

- *Other Tables*

```
xtabs(disease/total ~ sex+food, babyfood)

or xtable
or
table
```

Sweet R Packages

- **{tidyr} = some data wrangling things; haven't really used - has some SQL style things**
- **{ggplot2} = literally the best**

- **{gridExtra} = need this if you want to create panels with ggplot that aren't automated (where you can just facet-grid/wrap)**
- **{faraway} = needed for the *halfnorm* function**
- **{ggmap} = mapping with ggmpa**
- **{mapproj} = needed to get maps using spatial polygons**
- **{stats} = used for function *aggregate***
- **{texreg} = used for fancy tex output**
- **reshape = this one makes more sense and is stata-like**
- **reshape2 = this one contains 'melt'**
- **xtable**

Notes

- Don't use Tukey ** Consider Bonferroni/Scheffe
- In Random Effects, if σ^2 is close to 0, then something went wrong; not enough degrees of freedom; or consider using fewer random effects
- logistic and multinomial regression
- GLM won't run if you have multicollinearity
  - multicollinearity usually occurs with continuous vars
  - Ex: salt content and water amount - multicollinearity
  - Solution: pick one var

Investigate

- marginaleffects <- negbinmfx(zipsales ~ Classification + Income + Population.2010 + region + White + Black + Asian + Latino + physicianrequired +trainingrequired + registration_required, data=zip) xtable(marginaleffects$mfxest) xtable(anova(m2a))