Regression Diagnostics

An excellent review of regression diagnostics is provided in John Fox's aptly named *Overview of Regression Diagnostics*.

Dr. Fox's ***car*** package provides advanced utilities for regression modeling.

0.1 Alcohol and Tobacco Data

This example is for exposition only. We will ignore the fact that this may not be a great way of modeling the this particular set of data!

```
alctob <- data.frame( cbind(
Alcohol = c(6.47, 6.13, 6.19, 4.89, 5.63, 4.52,
            5.89, 4.79, 5.27, 6.08, 4.02),
Tobacco = c(4.03, 3.76, 3.77, 3.34, 3.47, 2.92,
            3.20, 2.71, 3.53, 4.51, 4.56)),
row.names = c("North", "Yorkshire", "Northeast",
"East Midlands", "West Midlands", "East Anglia",
"Southeast", "Southwest", "Wales",
"Scotland", "N. Ireland"))

# Assume that we are fitting a multiple linear regression
# on the MTCARS data
library(car)
fit <- lm(mpg~disp+hp+wt+drat, data=mtcars)
```

```
alctobwo <- subset(alctob,rownames(alctob)!="N. Ireland")
#without North Ireland

plot(alctob$Tobacco, alctob$Alcohol,
main="Weekly Household Spending on Alcohol vs. Tobacco",
xlab="Tobacco Spending (GBP)",
ylab="Alcohol Spending (GBP)",
pch=16,col="red",cex=1.5,font.lab=2)
#note N. Ireland in the bottom-right
```

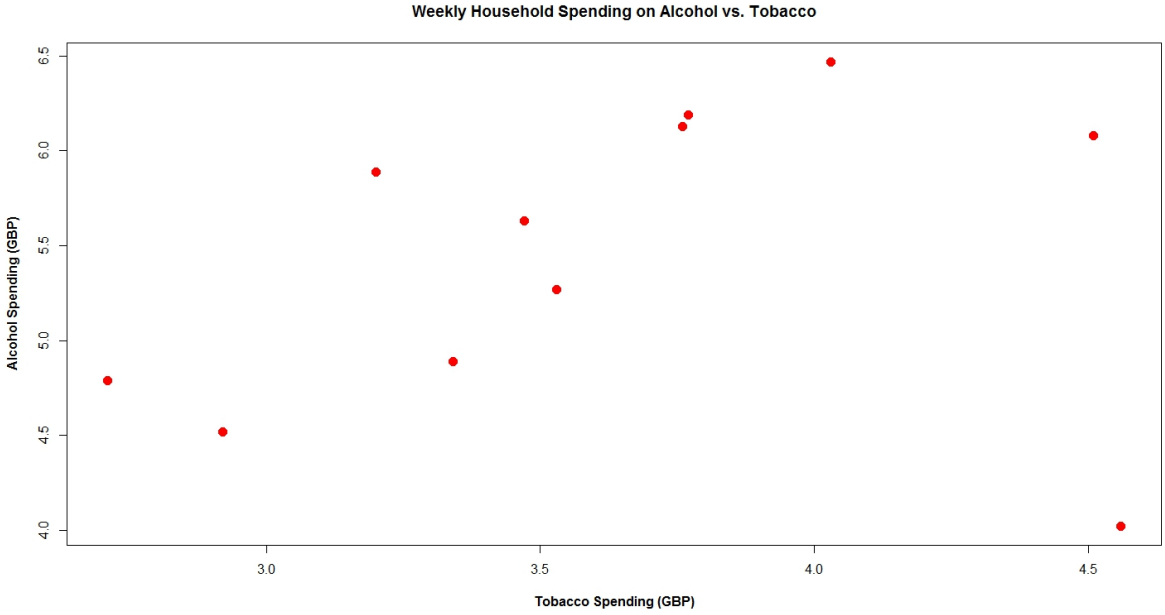


Figure 1:

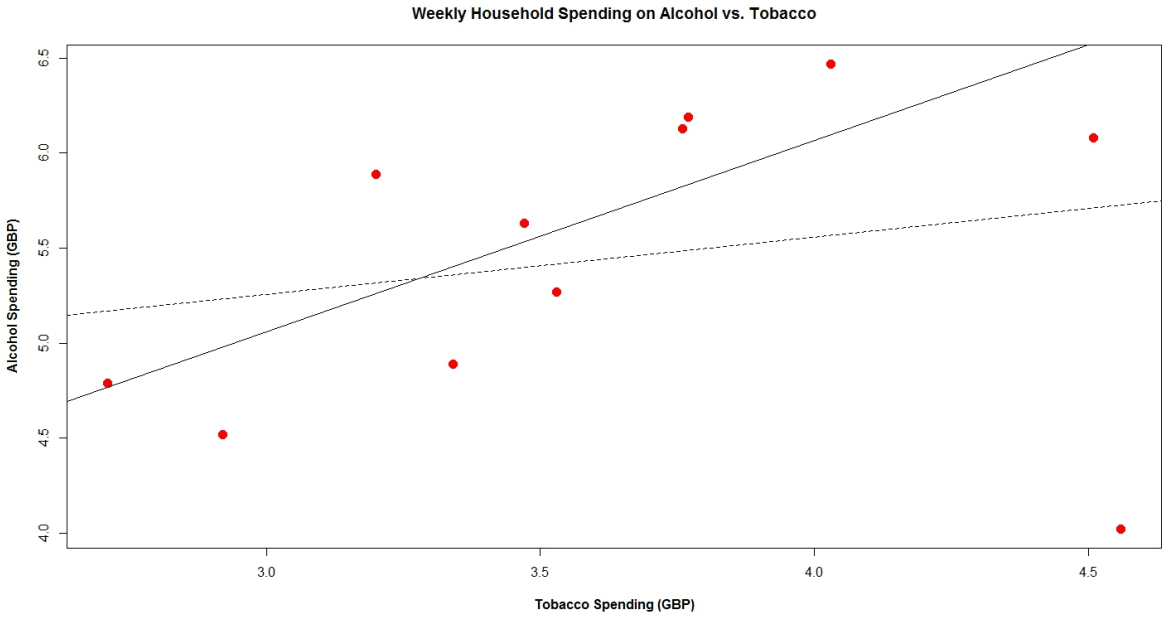


Figure 2:

All Observations

```
> summary(fit1)

Call:
lm(formula = Alcohol ~ Tobacco, data = alctob)

Residuals:
Min      1Q  Median      3Q     Max
-1.7080 -0.4245  0.2311  0.6081  0.9020

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.3512     1.6067   2.708   0.0241 *
Tobacco       0.3019     0.4388   0.688   0.5087
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.8196 on 9 degrees of freedom
Multiple R-squared:  0.04998,   Adjusted R-squared:  -0.05557
F-statistic: 0.4735 on 1 and 9 DF,  p-value: 0.5087
```

Outlier Removed

```
> summary(fit2)

Call:
lm(formula = Alcohol ~ Tobacco, data = alctobwo)

Residuals:
Min       1Q   Median       3Q      Max
-0.51092 -0.42434  0.06056  0.34406  0.62991

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.0412     1.0014   2.038  0.07586 .
Tobacco       1.0059     0.2813   3.576  0.00723 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.446 on 8 degrees of freedom
Multiple R-squared:  0.6151,    Adjusted R-squared:  0.567
```

```
F-statistic: 12.78 on 1 and 8 DF,  p-value: 0.007234
```

Outliers

The conservative outlier test that we talked about in class uses the Bonferonni inequality to calculate the p-values we associate with the Student's-t test.

In R, we can use the `outlierTest` command to perform this test on our model. Remember that when we test for influence, we are testing the effect of an observation on model coefficients.

Therefore we need to give the outlierTest command a linear model as its input.

```
> # Assessing Outliers
> # syntax: outlierTest(fit)
> # Bonferonni p-value for most extreme obs
>
> outlierTest(fit1)
            rstudent  unadjusted p-value Bonferonni p
N. Ireland -4.732091           0.0014789     0.016268
>
```

We can also use R to calculate Cook's distance. Remember that generally we label any observation with Cook's distance greater than 1 as influential.

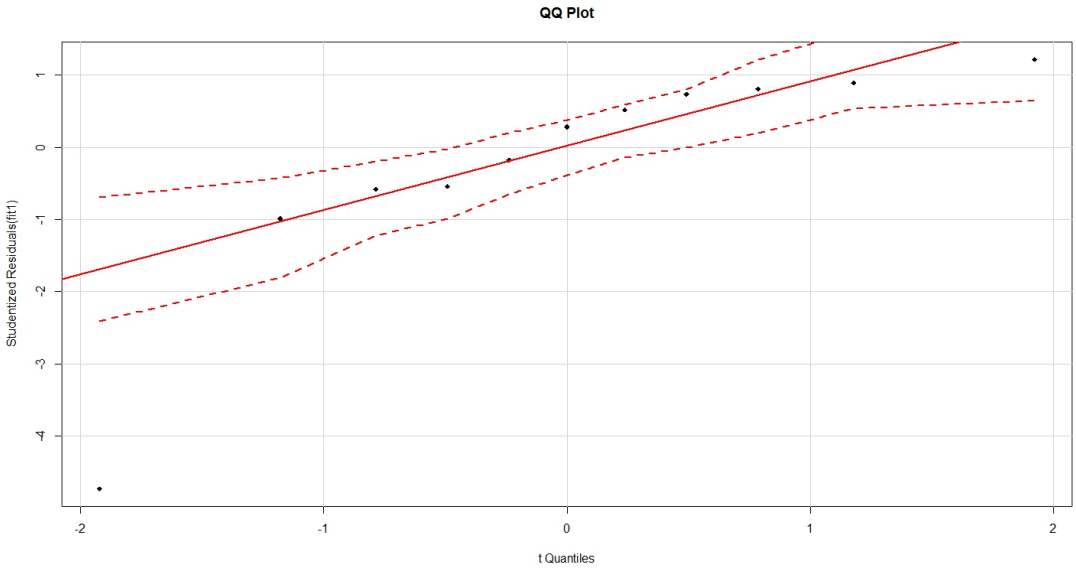
```
> cooks.distance(fit1)
North           Yorkshire       Northeast     East Midlands
0.114101051     0.036517838     0.043728951   0.023600304
West Midlands   East Anglia     Southeast     Southwest
0.004740759     0.147326647     0.046646563   0.077488350
Wales           Scotland        N. Ireland
0.001821694     0.068921892     1.747233521
```

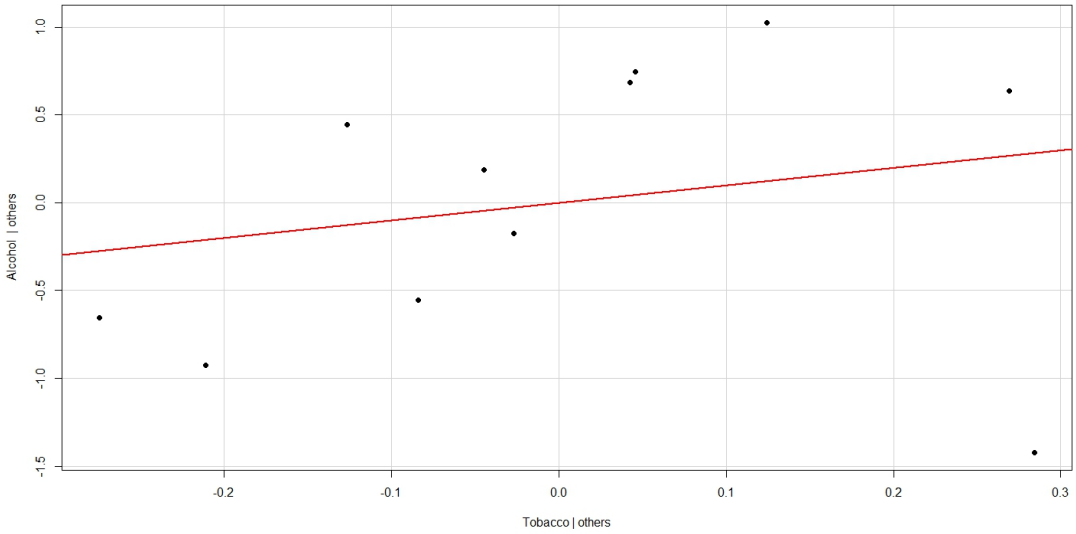
Finally, one of the easier ways to evaluate our residuals and look for for influential points is through plots.

```
qqPlot(fit1, main="QQ Plot",pch=18, lim=c(-3,2)) #qq plot for studentized resid
leveragePlots(fit1) # leverage plots
```

In R, the `plot` command takes a special form when you pass it an `lm` object (see ?plot.lm for all of the details).

Here we want to focus on three of the plots available through the command.


QQ Plot
Studentized Residuals(fit1)
t Quantiles
1
0
-1
-2
-3
-4
-2
-1
0
1
2


Alcohol | others
Tobacco | others
1.0
0.5
0.0
-0.5
-1.0
-1.5
-0.2
-0.1
0.0
0.1
0.2
0.3

- The first one displays the residuals vs. the fitted values we use this to evlauate the mean, variance and correlation of residuals. If our assumptions of constant variance and uncorrelated residuals are violated we may be able to correct this with a variance-stabilizing transformation.
- The second plot helps us check the normality of the residuals. If the residuals are indeed normal, they should fall along the dashed line. Remember that the normality assumption for our errors allows us to determine the standard errors of our coefficients and predictions.
- The final plot will display our residuals vs. their leverage. The dashed red lines are level curves that denote a particular value of Cook's distance. We will pay attention to points lying beyond the distance of 1. Notice that when we have data with row labels, the points will be labeled with their names. Otherwise, the row number will be shown.

```
plot(lmwith, which=c(1,2,5))
```

```
Influential Observations
# Influential Observations
# added variable plots
avPlots(fit1)


# Cook's D plot
# identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(mtcars)-length(fit1$coefficients)-2))
plot(fit, which=4, cook.levels=cutoff)
```

```
# Influence Plot
influencePlot(fit1,id.method="identify",
main="Influence Plot",
col="red"
sub="Circle size is proportial to Cook's Distance" )
```

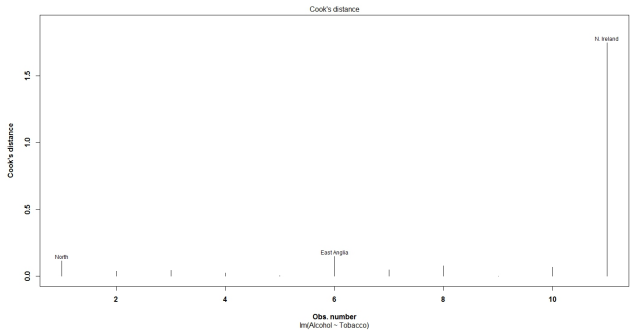


Figure 3:

Non-normality

```
# Normality of Residuals
# qq plot for studentized resid
qqPlot(fit1, main="QQ Plot")


# distribution of studentized residuals
library(MASS)
sresid <- studres(fit)
hist(sresid, freq=FALSE,
main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```

Non-constant Error Variance

```
# Evaluate homoscedasticity
# non-constant error variance test
ncvTest(fit)
# plot studentized residuals vs. fitted values
spreadLevelPlot(fit)
spread vs. levels click to view
```

```
Multi-collinearity
# Evaluate Collinearity
vif(fit) # variance inflation factors
sqrt(vif(fit)) > 2 # problem?
```

0.2 Nonlinearity

```
# Evaluate Nonlinearity
# component + residual plot
crPlots(fit)
# Ceres plots
ceresPlots(fit)
component plus residual plot Ceres plots click to view
Non-independence of Errors
# Test for Autocorrelated Errors
durbinWatsonTest(fit)
```

0.3 Additional Diagnostic Help

The gvlma( ) function in the gvlma package, performs a global validation of linear model assumptions as well separate evaluations of skewness, kurtosis, and heteroscedasticity.

```
# Global test of model assumptions
library(gvlma)
gvmodel <- gvlma(fit)
summary(gvmodel)
```

0.4 Going Further

If you would like to delve deeper into regression diagnostics, two books written by John Fox can help: Applied regression analysis and generalized linear models (2nd ed) and An R and S-Plus companion to applied regression.