Econometrics HW4

Timothy Schwieg

```
library(sandwich)
EngelData <- read.table( "Engel.dat" )

summary( EngelData )
```

```
##        V1               V2
##  Min.   : 377.1   Min.   : 242.3
##  1st Qu.: 638.9   1st Qu.: 429.7
##  Median : 884.0   Median : 582.5
##  Mean   : 982.5   Mean   : 624.2
##  3rd Qu.:1164.0   3rd Qu.: 743.9
##  Max.   :4957.8   Max.   :2032.7
```

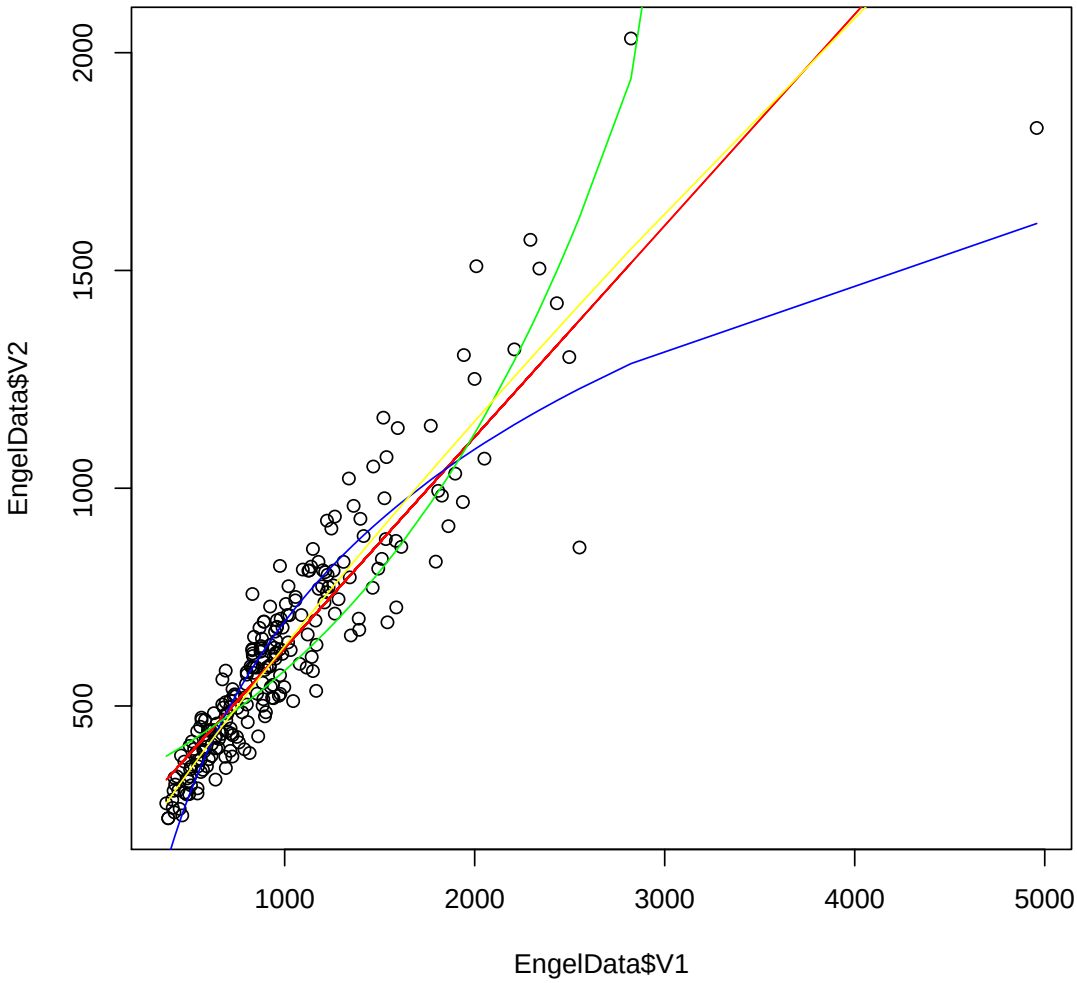
```
firstReg <- lm( EngelData$V1 ~ EngelData$V2 )
summary( firstReg )
```

```
##
## Call:
## lm(formula = EngelData$V1 ~ EngelData$V2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -570.58 -106.12  -21.57   72.12 1916.37
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -85.73632   34.58206  -2.479   0.0139 *
## EngelData$V2   1.71146    0.05068  33.772   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 214.3 on 233 degrees of freedom
## Multiple R-squared:  0.8304,Adjusted R-squared:  0.8296
## F-statistic:  1141 on 1 and 233 DF,  p-value: < 2.2e-16
```

```
#This is the homoskedastic errors
vcov( firstReg )
```

```
##              (Intercept) EngelData$V2
## (Intercept)  1195.918530 -1.602933677
## EngelData$V2   -1.602934  0.002568186

#This is the heteroskadistic one
vcovHC( firstReg, type = "HC1")

##              (Intercept) EngelData$V2
## (Intercept)   6553.91854 -11.67148273
## EngelData$V2   -11.67148   0.02107756
```

```
NoLogReg <- glm( EngelData$V2 ~ EngelData$V1, family="gaussian" )

LogXOnly <- glm( EngelData$V2 ~ I( log( EngelData$V1 ) ), family="gaussian" )

LogYOnly <- glm( I(log(EngelData$V2)) ~ EngelData$V1, family="gaussian" )

logBoth <-  glm( I( log( EngelData$V2)) ~ I( log( EngelData$V1 ) ), family="gaussian" )

plot( EngelData$V1, EngelData$V2 )
xweight <- EngelData$V1
yweight <- (predict(NoLogReg, type="response"))
lines(xweight, yweight, col="red")

#LogXOnly
xweight <- sort(EngelData$V1)
yweight <- sort(predict(LogXOnly, type="response"))
lines(xweight, yweight, col="blue")

#LogYOnly
xweight <- sort(EngelData$V1)
yweight <- exp( sort(predict(LogYOnly, type="response")) )
lines(xweight, yweight, col="green")

#logBoth
xweight <- sort(EngelData$V1)
yweight <- exp(sort(predict(logBoth, type="response")))
lines(xweight, yweight, col="yellow")
```



c

We can see that if we take the limit as λ tends towards zero, $\frac{Y_n^\lambda - 1}{\lambda}$ tends to $\log(Y_n)$, so letting λ tend towards zero gives us the log in the Ys and letting $\lambda = 1$ gives us the standard form. The same can be done for ψ to obtain $\log(x_n)$ enabling us to nest all of these specifications in one model. We

can see this because:

$$\frac{\alpha^{\beta}-1}{\beta} = \frac{\exp \beta \log \alpha - 1}{\beta} =$$
$$\frac{(1+\beta \log \alpha + \frac{\beta^2}{2} \log \alpha^2 + ... - 1}{\beta}$$
$$\log \alpha + \frac{\beta}{2} \log \alpha^2 + ...$$

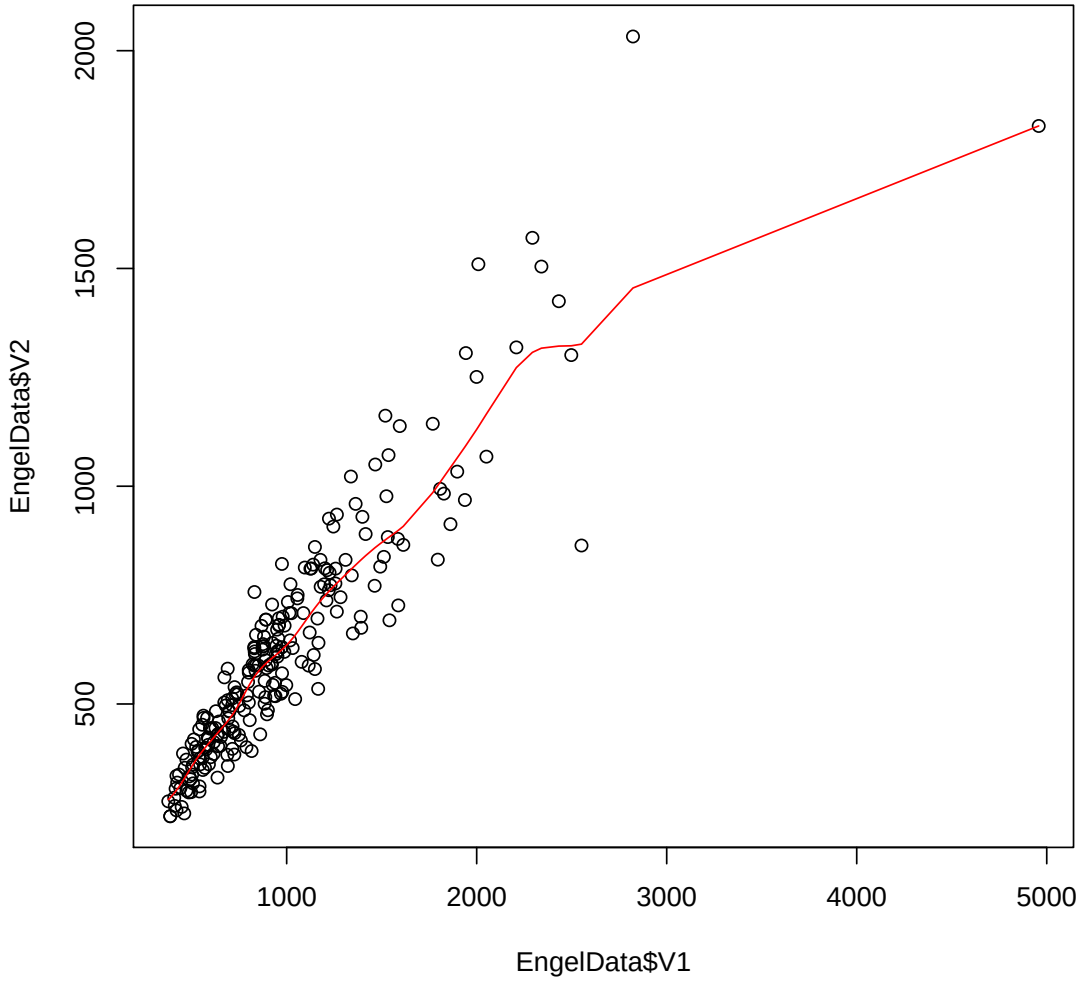
Taking the limit as β tends towards zero, we see that this simplifies to $\log \alpha$.

To test the different specifications from part b, we would take the liklihood with nothing imposed upon λ, ψ and calculate the liklihood under the estimated models, twice the difference between those two would be distributed $\chi^2(1)$ and from this we could test the hypothesis that those models are acceptable cases of the Box-Cox transformation.

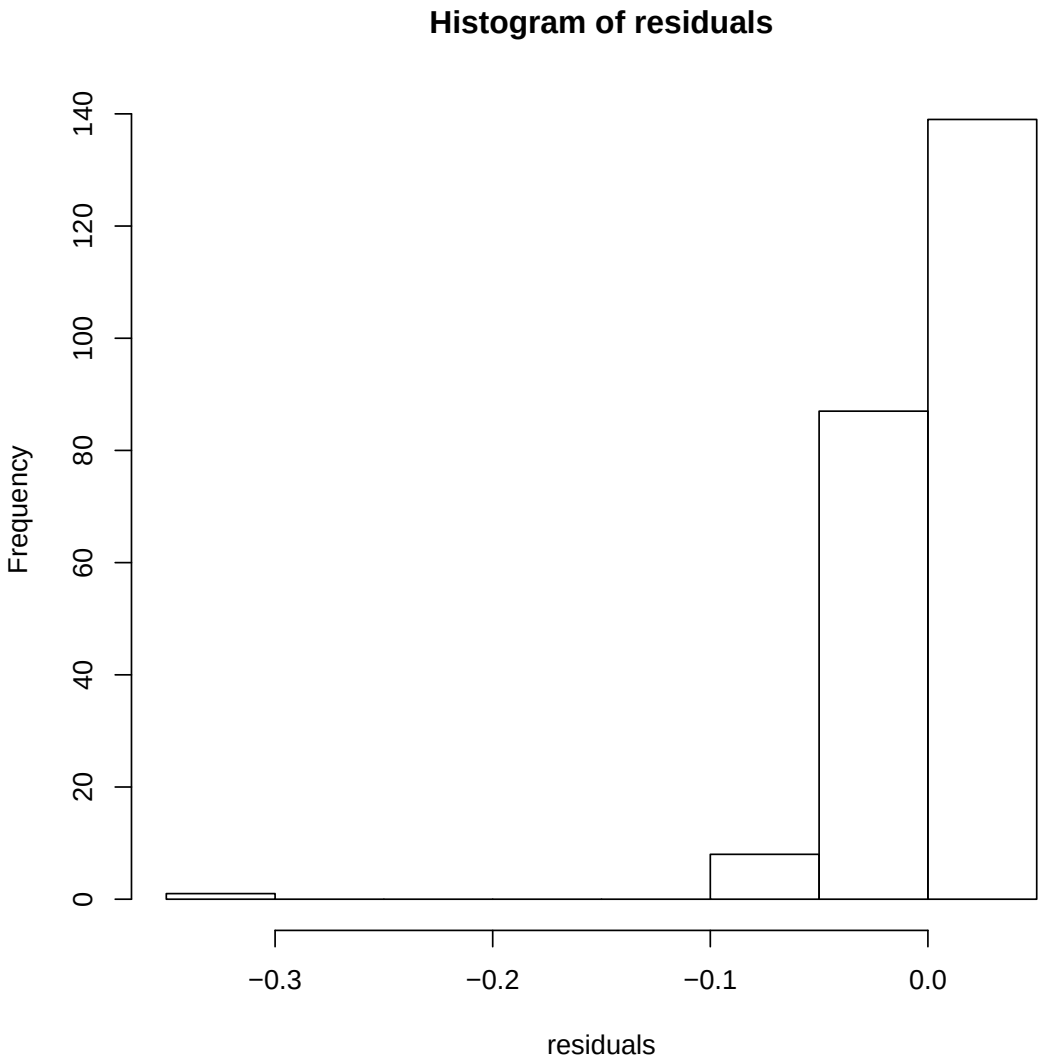
This introduces a problem, that we are merely testing if under the assumption that the Box-Cox Transformation captures the data, then these function forms are appropiate, not if they are appropiate given only the data. We have imposed a function form upon the data that affects our hypothesis testing. One possible way to get around this problem is to use non-parametrics and a kernal smoothed density function.

d

```
plot( EngelData$V1, EngelData$V2 )
#Using our previous bandwidth estimates gave us some really bad stuff, so im sticking to .25
SmoothBoy <- ksmooth( log(EngelData$V1), log(EngelData$V2),
                      bandwidth=.25, kernel="normal", x.points = log(EngelData$V1) )
xweight <- exp( SmoothBoy$x )
yweight <- exp( SmoothBoy$y )
lines(xweight, yweight, col="red")
```




```
residuals <- (SmoothBoy$y - logBoth$coefficients[1] -
              logBoth$coefficients[2]*SmoothBoy$x)

summary( residuals )

##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.3172000 -0.0050960  0.0027230  0.0002791  0.0128900  0.0344900

hist( residuals )
```



If we believe that the Non-parametric estimate is equal to the truth plus some error term, and under the null hypothesis we believe that $\log y_n = \hat{\alpha} + \hat{\beta} + U_n$ then our residuals must be equal to the sum of these two error terms. Given that both error terms are normal, we can simply apply a test of normality to the residuals.

```
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
```

```
## W = 0.62808, p-value < 2.2e-16
```

Based on this p-value we find that the non-parametric regression rejects the null hypothesis that the model is a valid fit.