OCN 750 - HW 4

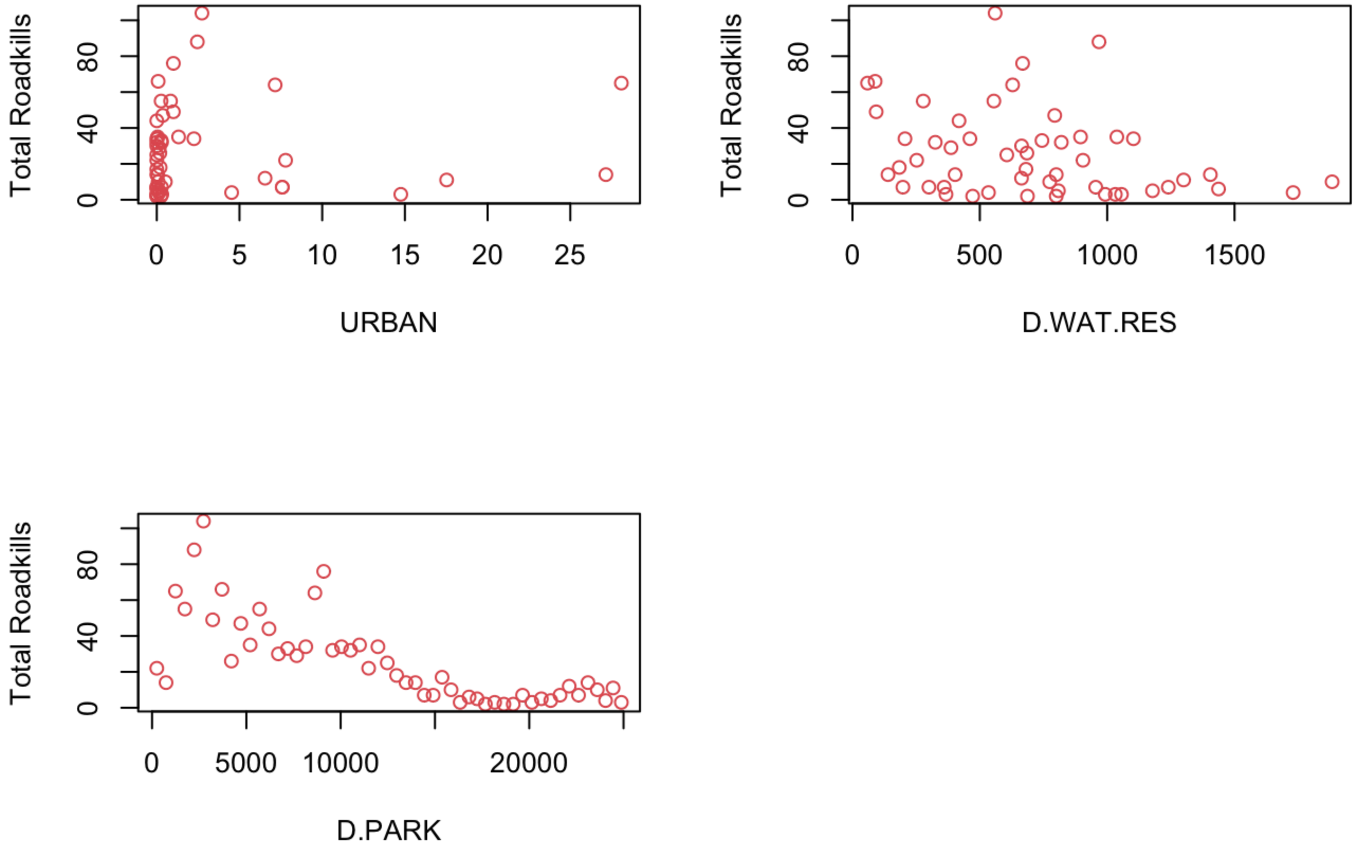
Maia Kapur

Thursday, September 17, 2015

```
#Set wd and load data table
#setwd("C:/Users/mkapur/Dropbox/2015 Fall/OCN 750")
roadkill = read.table("RoadKills.txt", header = TRUE, colClasses = "numeric")
```

Plot TOT.N vs. the other variables I mentioned. TOT.N, D.PARK, URBAN, D.WAT.RES. URBAN is highly skewed.

```
#set up graphing device
par(mfrow = c(2,2))

#create index of which columns we want to plot against TOT.N
colnums = c(13,18,20)

#a for-loop to plot each value quickly. URBAN is, indeed, quite skewed.
for (v in colnums){
plot(roadkill$TOT.N ~ roadkill[,v], xlab = names(roadkill[v]) , ylab = "Total Road
kills", col = "indianred3")
}
#dev.off()
```



Make a column where this predictors is square-root transformed,and plot the relationships with the new predictors.

```
#Create three new columns for the square-root transformations
roadkill[24] = NA

#designate their names and replace "V24" etc
newcols = c("urb_sqrt")
names(roadkill)[24] = newcols

#do it manually...
sqrt(roadkill$URBAN) -> roadkill[24]

head(roadkill)[24]
```

```
##   urb_sqrt
## 1 2.790520
## 2 5.210566
## 3 5.299623
## 4 0.911592
## 5 1.565886
## 6 1.652271
```

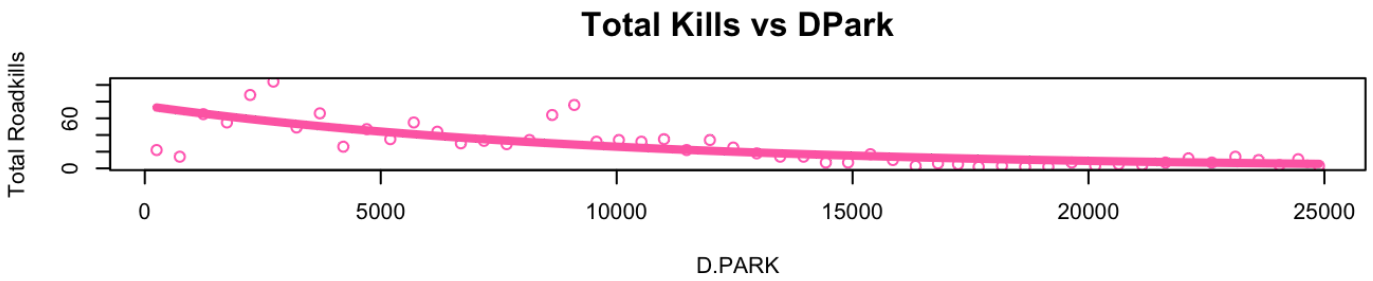
```
#a for-loop to plot the new transformed values against TOT.N.
dev.off()
```

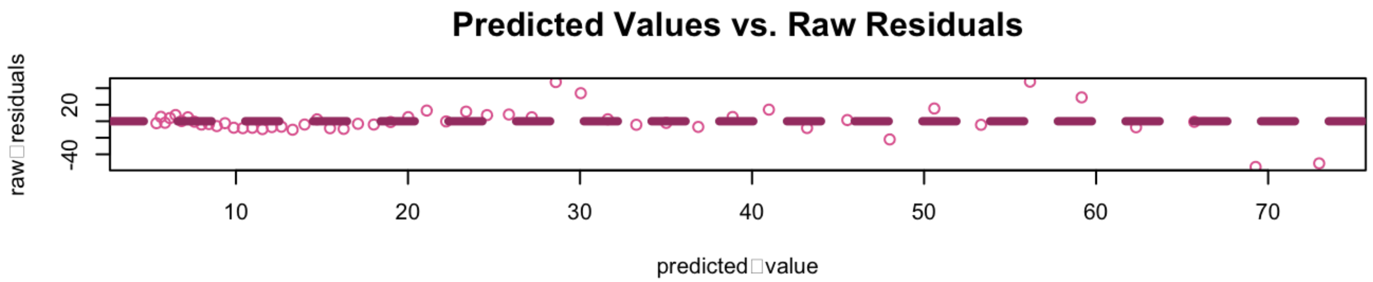
```
## null device 
##           1
```

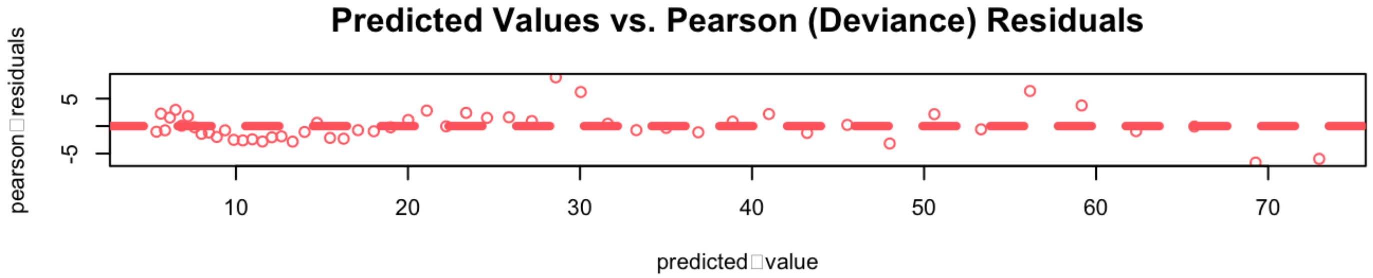
```
newnums = c(13,18,24)
par(mfrow = c(2,2))
for (k in newnums){
plot(roadkill$TOT.N ~ roadkill[,k], xlab = names(roadkill[k]) , ylab = "Total Road
kills", col = "hotpink4")
}
#dev.off()
```

Fit a Poisson GLM to test if total number of roadkills is predicted by D.PARK. Plot the raw relationship and plot the fitted curve on top of it. Plot residuals vs. the predictor: use both the raw (response) residuals and the deviance residuals.

```
require(stats)
#create your Poisson GLM using the square-root transformed values
#I've noticed that for the Effects package to work, you need to separate the data
from the field names.
#(No $ signs!)
poipark = glm(TOT.N ~ D.PARK, data = roadkill, family = poisson)

#Plot a curve using generated coefficients against the data
par(mfrow = c(3,1))
plot(roadkill$TOT.N ~roadkill$D.PARK, xlab = "D.PARK" , ylab = "Total Roadkills",
col = "hotpink1", main = "Total Kills vs DPark", cex.main = 1.5)
curve(exp(coef(poipark)[1]+coef(poipark)[2]*x),  col    = 'hotpink2',   lwd = 4, x
lab = "D.PARK", ylab = "Total Kills", add = TRUE)

#Plot of raw residuals against predicted values based on regression (these are ext
racted via fitted())
plot(residuals(poipark, type    = "response")   ~ fitted(poipark),  ylab    = "raw
residuals", xlab    = "predicted    value", col = "hotpink3", main = "Predicted Va
lues vs. Raw Residuals", cex.main = 1.5)
abline(h    = 0,    lty = 2,    lwd = 4,    col = 'hotpink4')

#Same as above, but with deviance (Pearson) residuals
plot(residuals(poipark,  type   = "pearson")    ~ fitted(poipark),  ylab    = "pea
rson  residuals", xlab    = "predicted    value", main = "Predicted Values vs. Pea
rson (Deviance) Residuals", cex.main = 1.5, col = "indianred1")
abline(h    = 0,    lty = 2,    lwd = 4,    col = "indianred2")
```







Are there any strong patterns? How do the raw and deviance residuals differ, and why?

```
#The deviance residuals are more constrained, though they don't seem to differ gre
atly from the pattern. There is a clear negative, potentially logarithmic relatoin
ship between the predictor and response variable, with variance appearing to incre
ase with D.Park. This would explain the tighter relationship between values and re
siduals at lower values of D.Park.
```

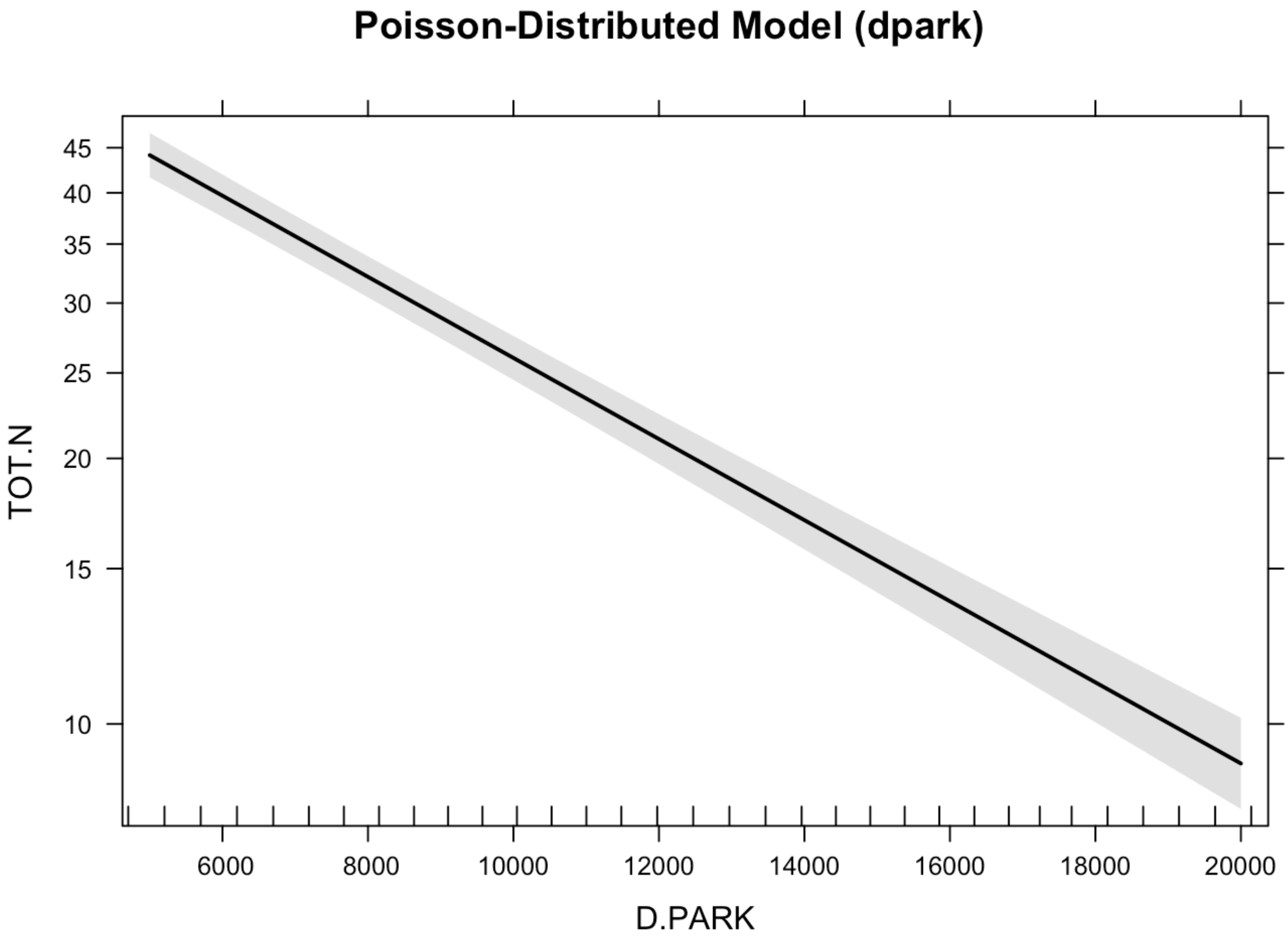
Do a likelihood ratio test to test for the significance of the predictor.

```
require(car)
require(effects)
Anova(poipark)
```

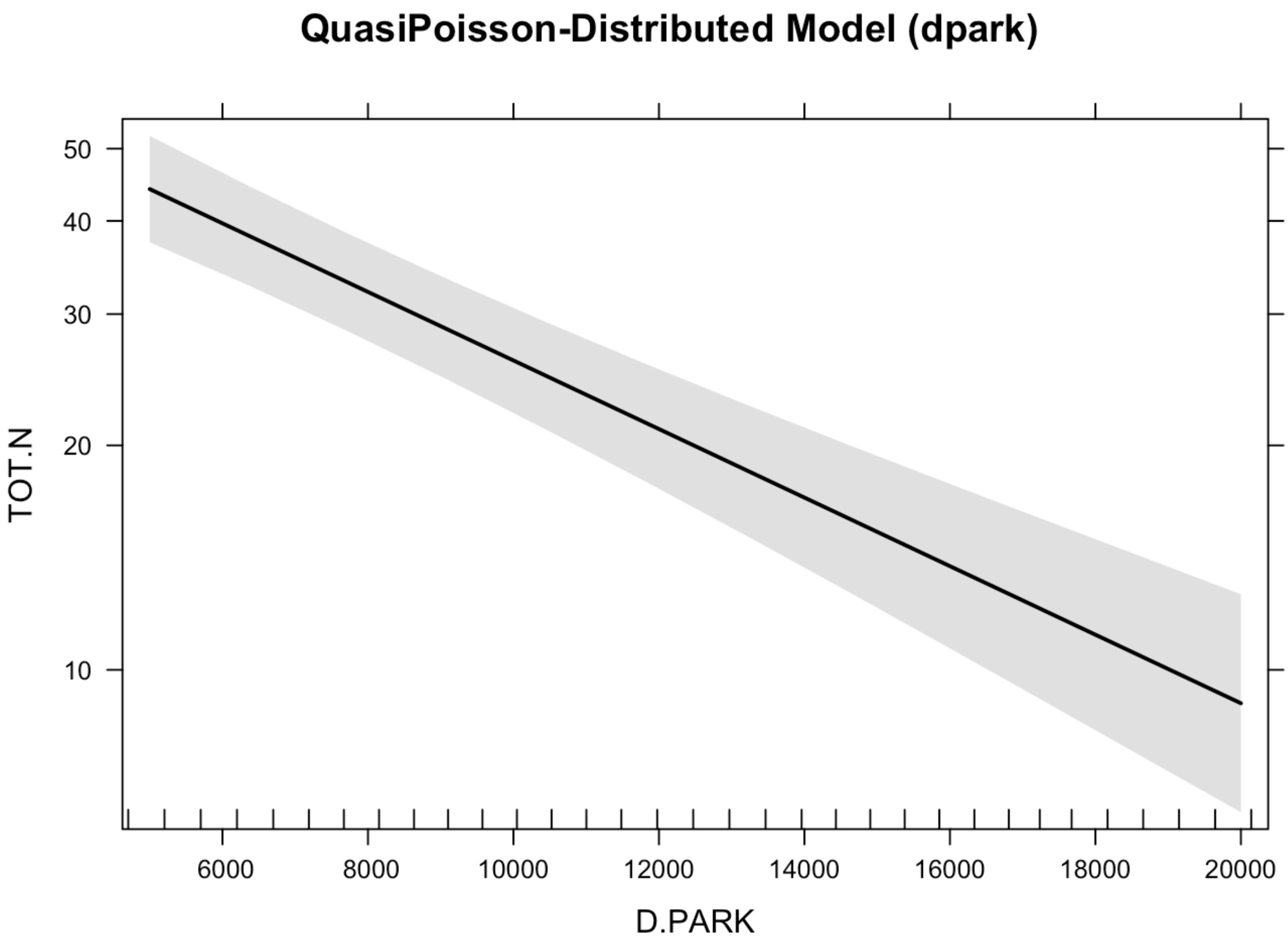
```
## Analysis of Deviance Table (Type II tests)
##
## Response: TOT.N
##         LR Chisq Df Pr(>Chisq)
## D.PARK    680.55  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fit the same model with quasipoisson, and with a negative binomial distribution.

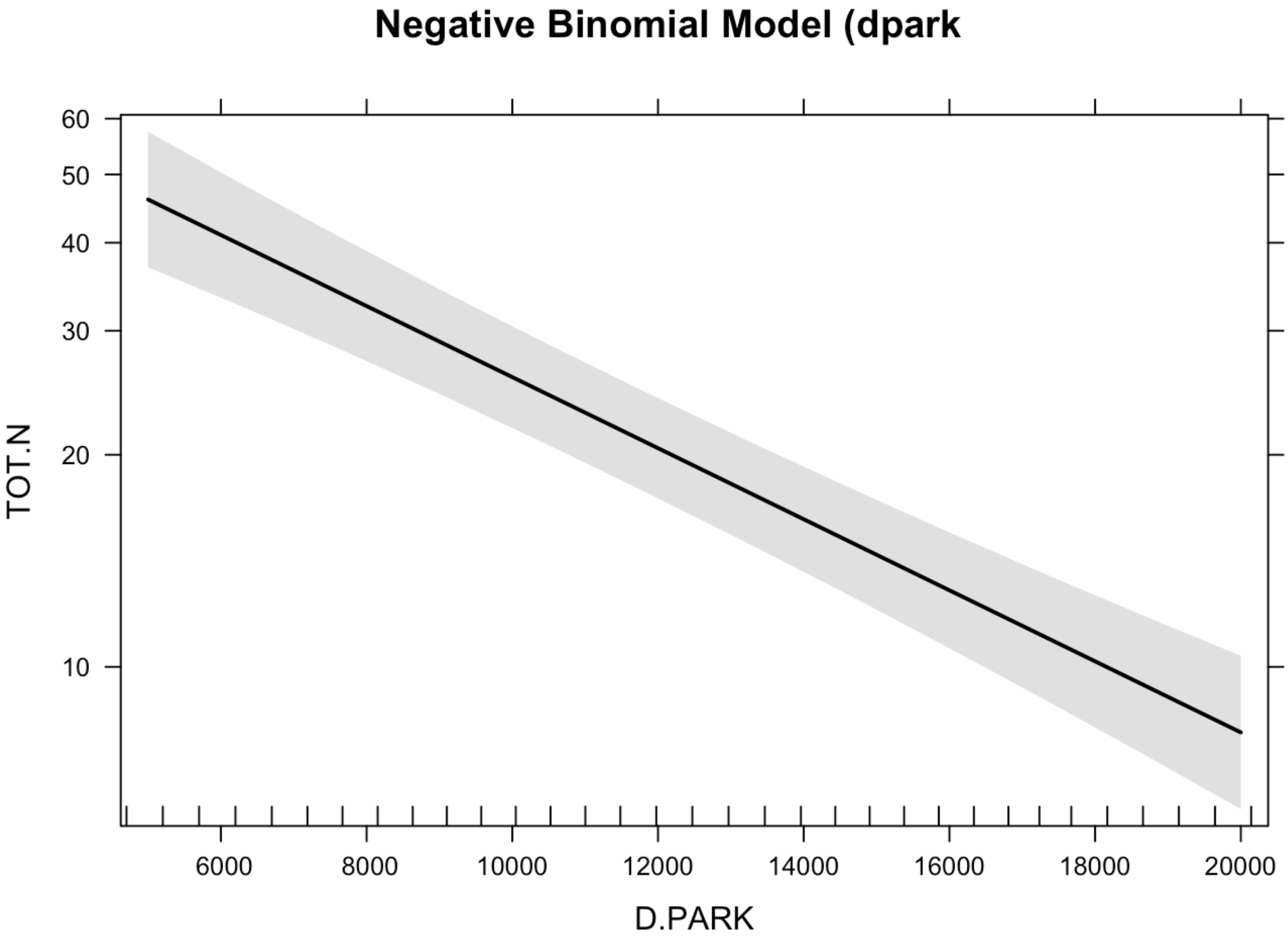
```
#create your Quasi - Poisson GLM using the square-root transformed values
qpoipark = glm(TOT.N ~ D.PARK, data = roadkill, family = quasipoisson)

#Create the negative binomial model
require(MASS)
negbpark = glm.nb(TOT.N ~ D.PARK, data = roadkill)

#Plot them all together
par(mfrow = c(3,1))
#This looks very different from the example in class.
plot(allEffects(poipark), main = "Poisson-Distributed Model (dpark)", col = "darkseagreen1")
```



```
plot(allEffects(qpoipark), main = "QuasiPoisson-Distributed Model (dpark)", col = "darlseagreen2")
```



```
plot(allEffects(negbpark), main = "Negative Binomial Model (dpark", col = "darksea
green3")
```



Do appropriate hypothesis tests on these models.

```
#A bootstrapping may be more approriate, but here we do a  Likelhihood Ratio Test
(X2 distributed).
#A t-test would assume normal error (analogous to Wald test).
#The LRT is quickly summarized by Anova()

#Here I run the test and pin it to a new data frame. All three indicate significan
ce.
LRT = cbind(Anova(poipark)[1],
Anova(qpoipark)[1],
Anova(negbpark)[1])
names(LRT)[1:3] = c("poisson", "qpoisson", "negbinom")
LRT
```

```
##          poisson qpoisson negbinom
## D.PARK 680.5468 89.19183  100.703
```

How big is overdispersion based on the quasipoisson?

```
#dispersion  parameter  function - this way you can quickly quantify Phi based on
whatever glm you used.
overdis = function(model)   {
        sum(residuals(model,    type    = "pearson")^2)/(length(model$y) - lengt
h(model$coefficients))
}
overdis(qpoipark) #Phi is quite high
```

```
## [1] 7.629755
```

What is the theta parameter for the negative binomial?

```
theta = overdis(negbpark)
theta #much smaller
```

```
## [1] 1.011433
```

Does overdispersion affect the conclusions you would draw from this analysis?

```
#Our Phi value was pretty high even for the QPoisson.
#Looks like the negative binomial model successfully accounts for the overdispersi
on, as the Theta value is well under 1.5.
```

Now fit a Poisson model that adds in the other two predictors, URBAN (squareroot transformed) and D.WAT.RES.

```
poi.urwapa = glm(TOT.N ~ urb_sqrt + D.WAT.RES + D.PARK, data = roadkill, family =
"poisson")
summary(poi.urwapa)
```

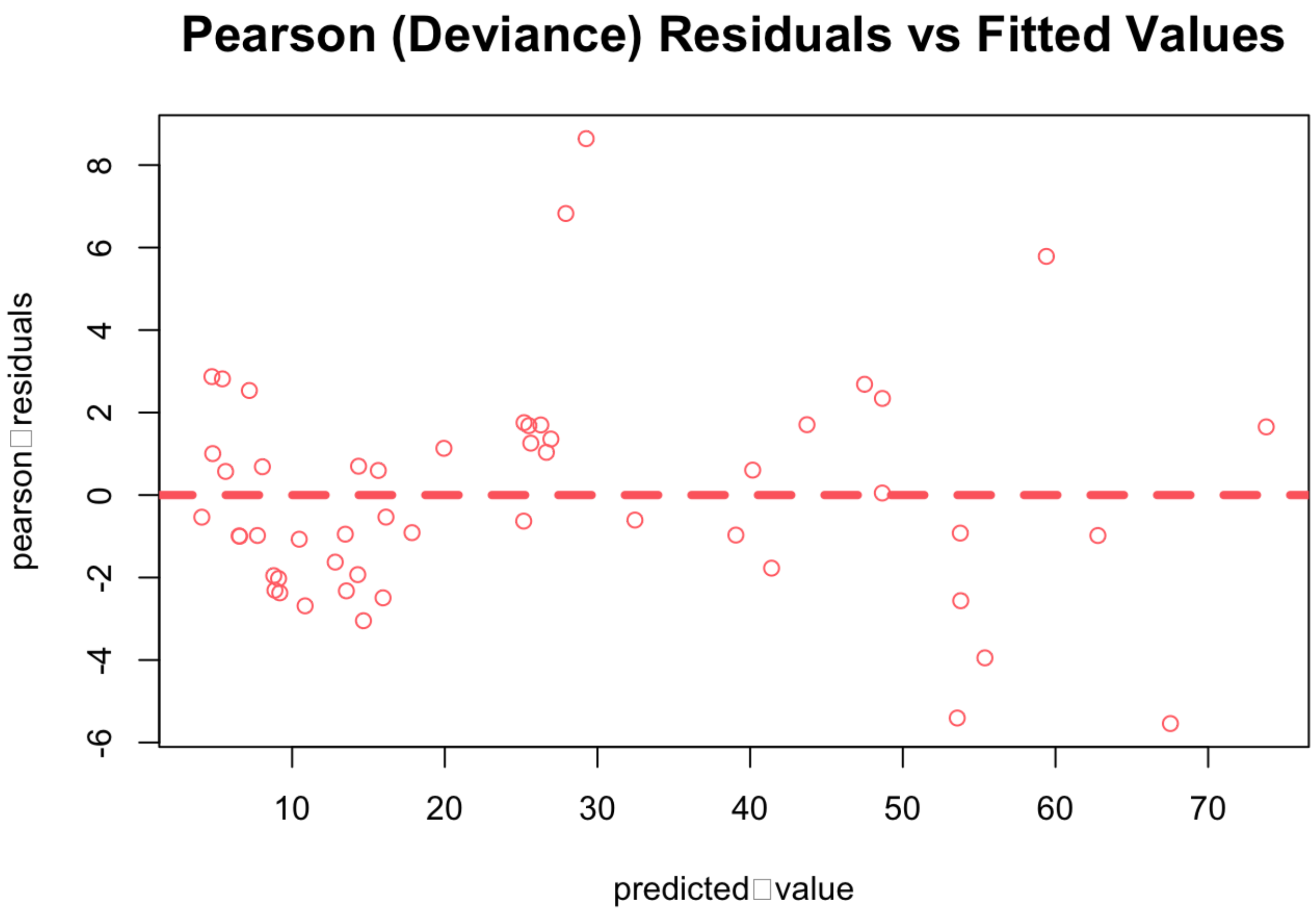
```
##
## Call:
## glm(formula = TOT.N ~ urb_sqrt + D.WAT.RES + D.PARK, family = "poisson",
##     data = roadkill)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.4582  -1.9345  -0.5904   1.3783   7.1839
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.298e+00  7.196e-02  59.734  < 2e-16 ***
## urb_sqrt    -5.345e-02  2.145e-02  -2.491   0.0127 *
## D.WAT.RES    3.729e-04  9.053e-05   4.119 3.81e-05 ***
## D.PARK      -1.228e-04  5.499e-06 -22.334  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1071.4  on 51  degrees of freedom
## Residual deviance:  361.7  on 48  degrees of freedom
## AIC: 609.09
##
## Number of Fisher Scoring iterations: 5
```

What is effect size of the 3 different predictors, i.e. how much does # roadkills change as these predictors vary? How do residuals vs. fitted values and residuals vs. predictors look (you can just use deviance or pearson residuals, as they are more appropriate for GLMs)? Do appropriate (marginal) likelihood ratio tests for each of the three predictors.

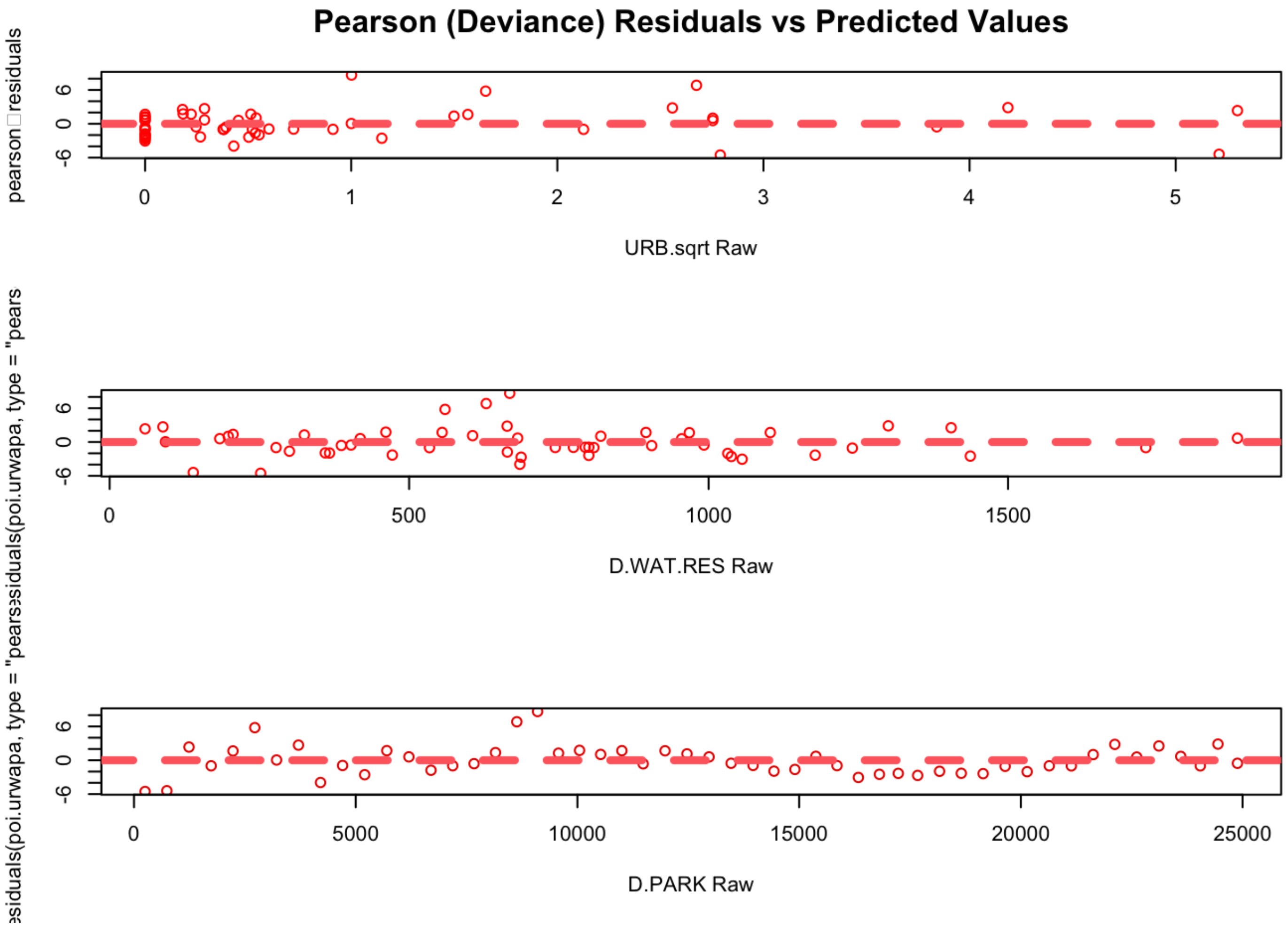
```
#The effect of each of these predictors is given by the ANOVA.
Anova(poi.urwapa) #it appears that Dpark is still the highest influence upon the m
odel.
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: TOT.N
##           LR Chisq Df Pr(>Chisq)
## urb_sqrt      6.43  1    0.01125 *
## D.WAT.RES    16.78  1  4.208e-05 ***
## D.PARK      595.27  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Plot of fitted values vs. deviance (Pearson) residuals
plot(residuals(poi.urwapa,  type    = "pearson")    ~ fitted(poi.urwapa),   ylab
= "pearson  residuals", xlab    = "predicted    value", main = "Pearson (Deviance)
Residuals vs Fitted Values", cex.main = 1.5, col = "indianred1")
abline(h    = 0,    lty = 2,    lwd = 4,    col = "indianred2")
```



```
#Plot of sqrt-transformed predictor values vs. deviance (Pearson) residuals for th
ree variables
par(mfrow = c(3,1))

#raw urban values
plot(residuals(poi.urwapa,  type  = "pearson")  ~ roadkill$urb_sqrt,    ylab    =
"pearson  residuals", xlab    = "URB.sqrt Raw", main = "Pearson (Deviance) Residua
ls vs Predicted Values", cex.main = 1.5, col = "firebrick1")
abline(h    = 0,    lty = 2,    lwd = 4,    col = "indianred2")

#raw dwat values
plot(residuals(poi.urwapa,  type  = "pearson") ~ roadkill$D.WAT.RES, xlab = "D.WA
T.RES Raw", col = "firebrick2")
abline(h  = 0,  lty = 2,    lwd = 4,    col = "indianred2")

#raw dpark values
plot(residuals(poi.urwapa,  type  = "pearson") ~ roadkill$D.PARK, xlab = "D.PARK R
aw", col = "firebrick3")
abline(h  = 0,  lty = 2,    lwd = 4,    col = "indianred2")
```

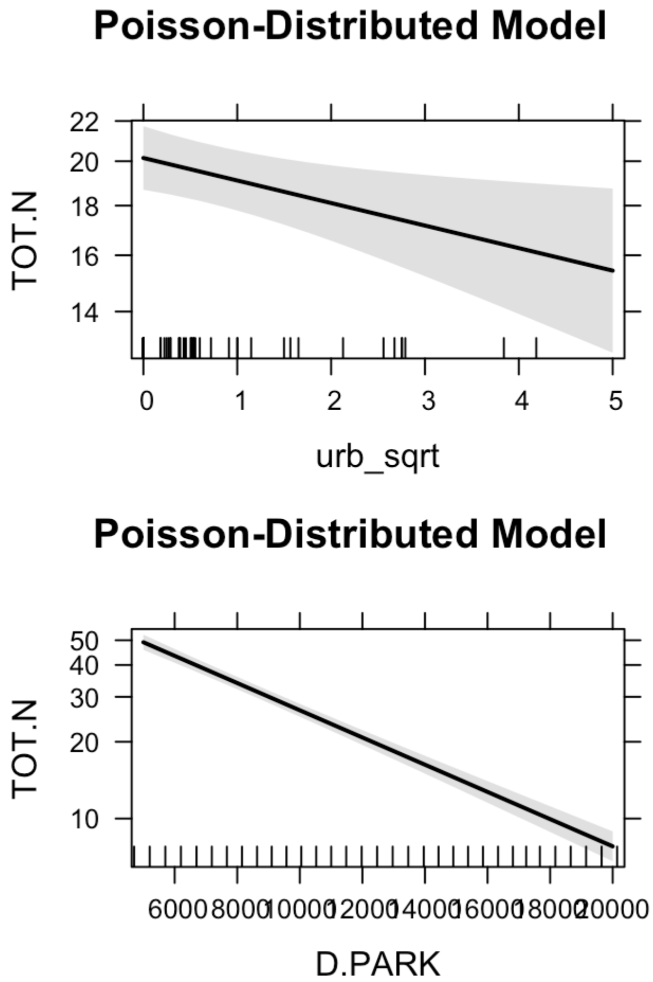


```
dev.off()
```

```
## null device
##           1
```

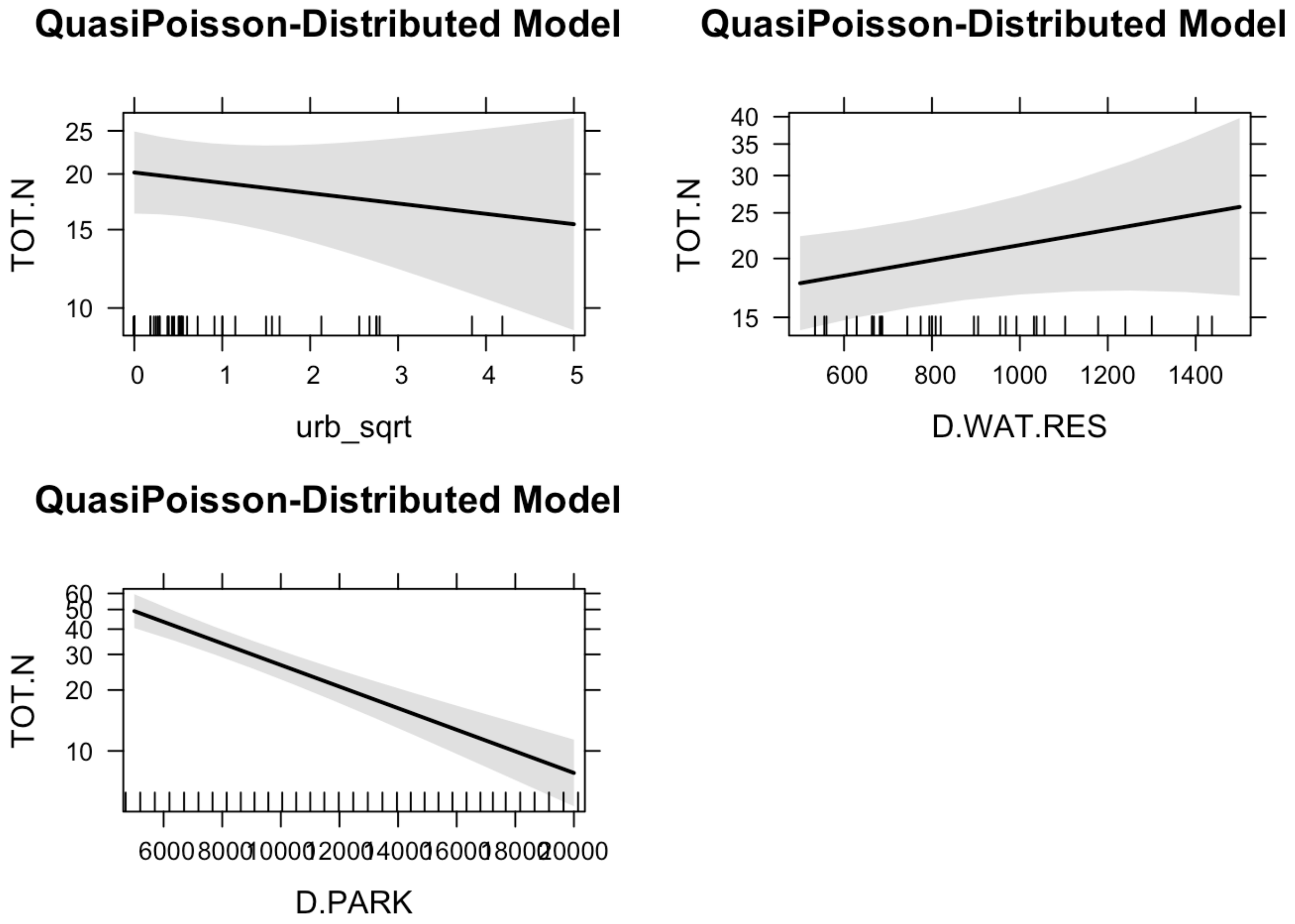
Plot the fitted effects using the ‘effects’ package. How do you interpret these results?

```
par(mfrow = c(1,2))
#This looks very different from the example in class.
plot(allEffects(poi.urwapa), main = "Poisson-Distributed Model", col = "darkseagre
en1")
```

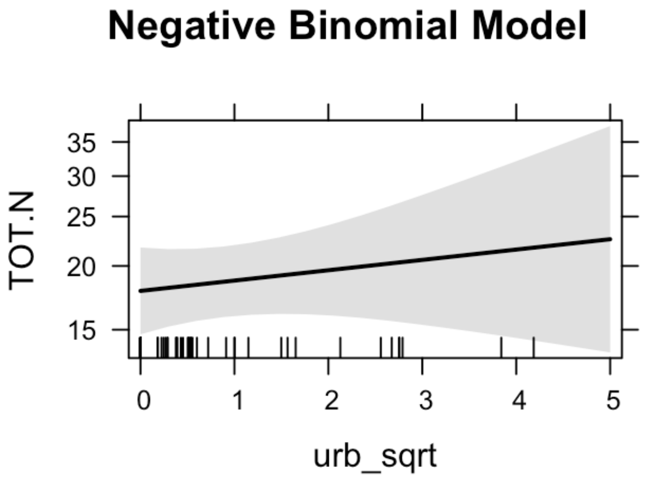


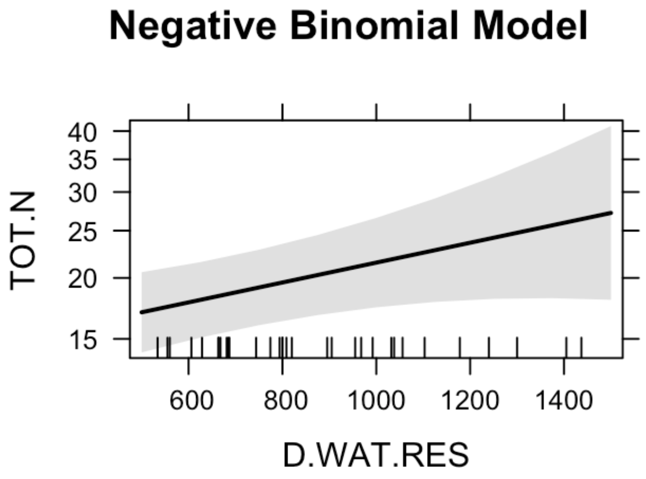
Now fit the same model with quasi-Poisson and negative binomial. Plot the fittedeffects using the ‘effects’ package. How similar are the parameter estimates between Poisson, quasi-Poisson, and negative binomial model?

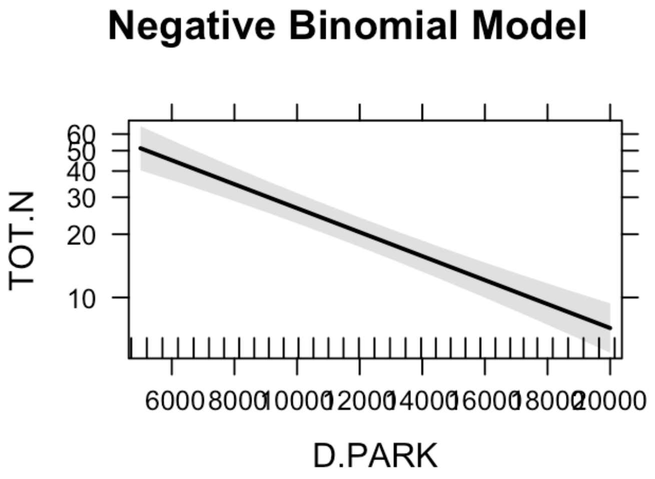
```
#create your Quasi - Poisson GLM using urban square-root transformed values
qpoi.urwapa = glm(TOT.N ~ urb_sqrt + D.WAT.RES + D.PARK, data = roadkill, family =
quasipoisson)

#Create the negative binomial model using all transformed values
require(MASS)
negb.urwapa = glm.nb(TOT.N ~ urb_sqrt  + D.WAT.RES + D.PARK, data = roadkill)

par(mfrow = c(4,3))
#plot for quasi-poisson using "all effects"
plot(allEffects(qpoi.urwapa), main = "QuasiPoisson-Distributed Model", col = "darl
seagreen2")
```



```
#same as above for negative bionamial
plot(allEffects(negb.urwapa), main = "Negative Binomial Model", col = "darkseagree
n3")
```


Negative Binomial Model


Negative Binomial Model


Negative Binomial Model

```
#dev.off()

#The parameter estimates appear to vary greatly amongst model families as well as
variables.
```

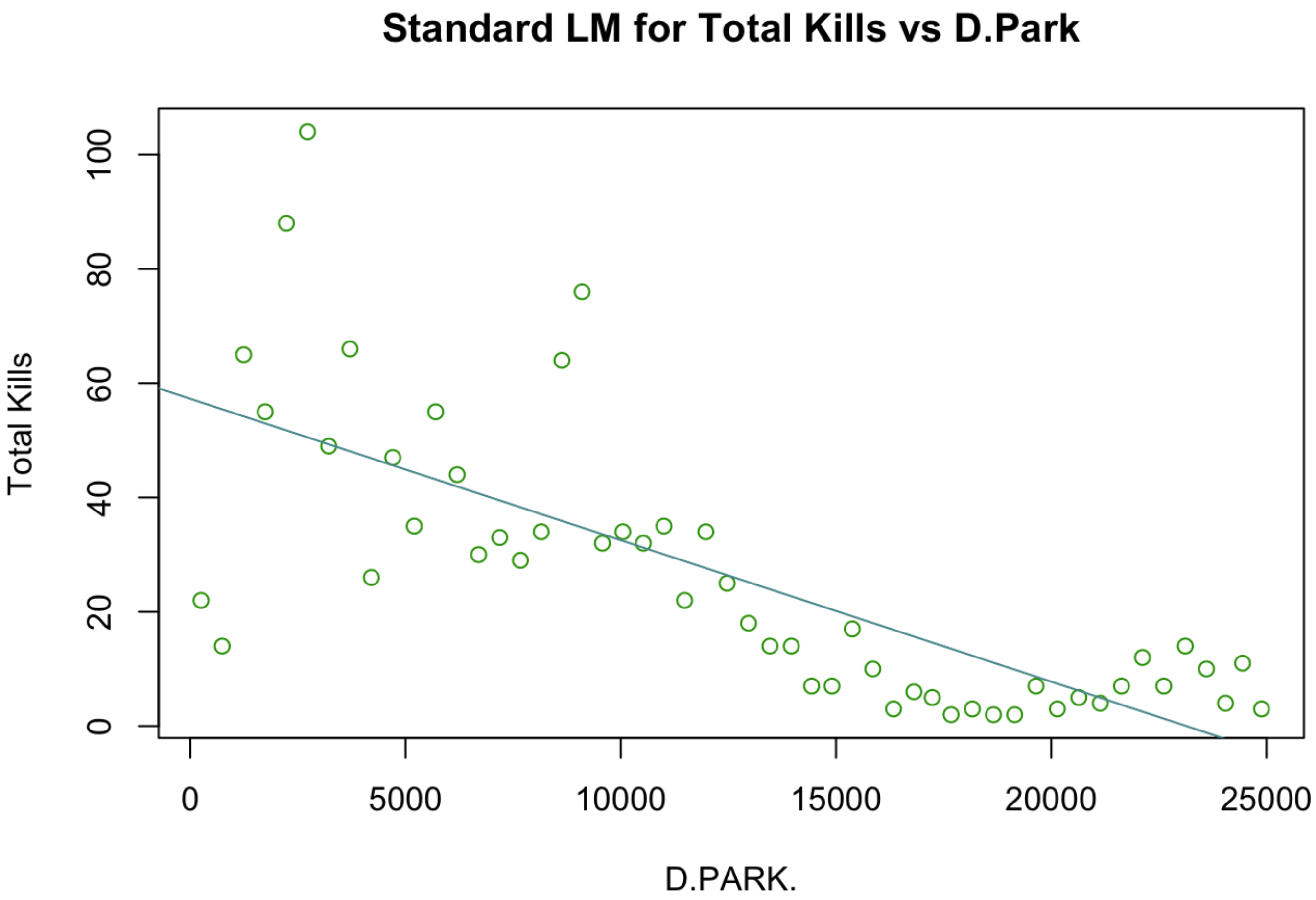
How similar are hypothesis test results for the three models? Why do you think these results would differ from what you found for #3?

```
#This table shows the results from the Anova function for all three variables acro
ss all three models.
#As suggested by the narrowness of the CI for the above plots, D.Park is the only
significant predictor for Tot.n
LRT = cbind(Anova(poi.urwapa)[1],
Anova(qpoi.urwapa)[1],
Anova(negb.urwapa)[1])
names(LRT)[1:3] = c("poisson", "qpoisson", "negbinom")
LRT
```

```
##              poisson   qpoisson   negbinom
## urb_sqrt    6.425626  0.8525959  0.5360025
## D.WAT.RES  16.775319  2.2258637  3.7738560
## D.PARK    595.271539 78.9846863 92.8860084
```

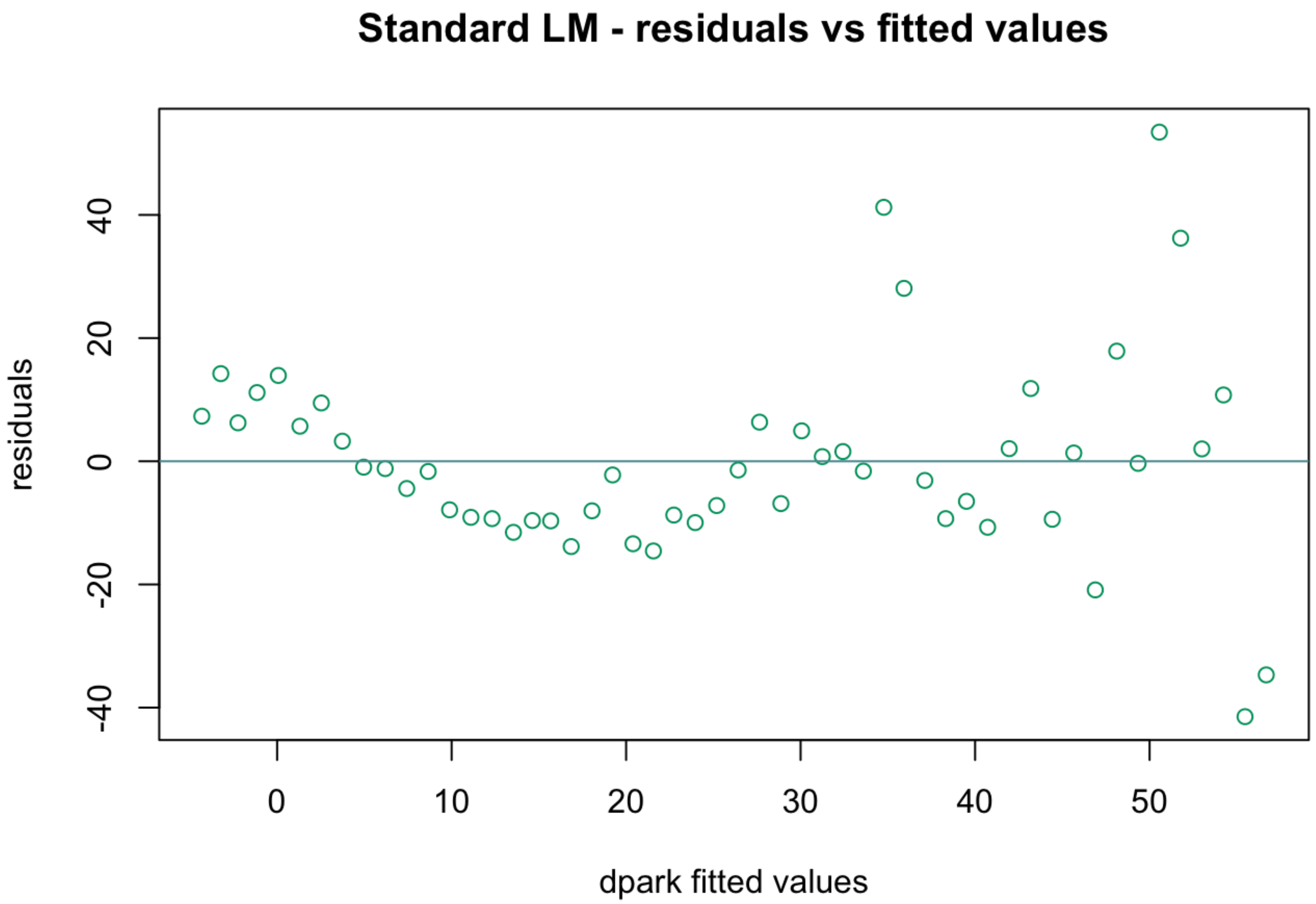
Let’s go back to a model with just TOT.N vs D.PARK. Fit a standard linear regression for this relationship, i.e. with normally distributed error. Plot the data and the fitted relationship.

```
#a standard linear model of the two variables
totpark = lm(TOT.N ~ D.PARK, data = roadkill)

plot(TOT.N ~ D.PARK, data = roadkill, col = "chartreuse4", main = "Standard LM for
Total Kills vs D.Park", xlab = "D.PARK.", ylab = "Total Kills")
abline(coef= coefficients(totpark), col = "cadetblue4")
```



Plot residuals vs. fitted values. Based on these plots, do you think this model is a good alternative for this data? Explain why you think yes/no.

```
plot(residuals(totpark,  type   = "response")   ~ fitted(totpark), xlab = "dpark f
itted values", ylab = "residuals", col = "seagreen4", main = "Standard LM - residu
als vs fitted values")
abline(h=0, col = "cadetblue4")
```



```
#While the general LM seems to fit alright for higher levels of DPark, the residuals have a "funnel" shape which indicates the variance is not normally distributed. It is likely a bad choice to select this model type while the other tests don't retain an assumption of normality.
```