

Poisson EDA example (W271 Unit 5)

Michael Winton

June 16, 2018

Introduction

- Introduce use of a Poisson regression to model a count response variable
- Consider situations in which the conditional distribution of the response variable follow a Poisson distribution
- Using a dataset on executives at large Canadian companies, study interlocking directorates (board members serving on each other's boards)

Dataset

This example works with the `Ornstein` dataset from the `car` library

```
# list the top 10 observations in the dataset
head(Ornstein, 10)
```

```
##   assets sector nation interlocks
## 1 147670   BNK   CAN         87
## 2 133000   BNK   CAN        107
## 3 113230   BNK   CAN         94
## 4  85418   BNK   CAN         48
## 5  75477   BNK   CAN         66
## 6  40742   FIN   CAN         69
## 7  40140   TRN   CAN         46
## 8  26866   BNK   CAN         16
## 9  24500   TRN   CAN         77
## 10 23700   MIN   US           6
```

```
# summary statistics about the dataset
summary(Ornstein)
```

```
##      assets      sector  nation  interlocks
## Min.   :    62  MIN   :54  CAN:117  Min.   :  0.00
## 1st Qu.:   519  MAN   :48  OTH: 18  1st Qu.:  3.00
## Median :  1397  AGR   :47  UK : 17  Median :  9.00
## Mean   :  5978  FIN   :22  US : 96  Mean   : 13.58
## 3rd Qu.:  4326  MER   :20           3rd Qu.: 18.00
## Max.   :147670  WOD   :19           Max.   :107.00
##                (Other):38
```

```
describe(Ornstein) #more verbose summary stats
```

```
## Ornstein
##
```

```
## 4 Variables      248 Observations
## -----
## assets
##      n missing distinct      Info      Mean      Gmd      .05      .10
##      248      0      240      1      5978      9114      257.8      326.0
##      .25      .50      .75      .90      .95
##      519.0     1397.0    4325.5   13103.1   20047.8
##
## 0 (111, 0.448), 2000 (55, 0.222), 4000 (25, 0.101), 6000 (13, 0.052), 8000
## (9, 0.036), 10000 (6, 0.024), 12000 (4, 0.016), 14000 (4, 0.016), 16000
## (4, 0.016), 18000 (4, 0.016), 20000 (1, 0.004), 22000 (1, 0.004), 24000
## (3, 0.012), 26000 (1, 0.004), 40000 (2, 0.008), 76000 (1, 0.004), 86000
## (1, 0.004), 114000 (1, 0.004), 132000 (1, 0.004), 148000 (1, 0.004)
## -----
## sector
##      n missing distinct
##      248      0      10
##
## Value      AGR  BNK  CON  FIN  HLD  MAN  MER  MIN  TRN  WOD
## Frequency    47   8   5   22   7   48   20   54   18   19
## Proportion 0.190 0.032 0.020 0.089 0.028 0.194 0.081 0.218 0.073 0.077
## -----
## nation
##      n missing distinct
##      248      0      4
##
## Value      CAN  OTH  UK  US
## Frequency   117   18   17  96
## Proportion 0.472 0.073 0.069 0.387
## -----
## interlocks
##      n missing distinct      Info      Mean      Gmd      .05      .10
##      248      0      50    0.997    13.58    15.22      0.0      0.0
##      .25      .50      .75      .90      .95
##      3.0      9.0     18.0     31.0     41.3
##
## lowest :    0    1    2    3    4, highest: 69 77 87 94 107
## -----
# show mean value of interlocks per nation/sector
round(with(Ornstein, tapply(interlocks, nation, mean)), 1)

## CAN  OTH  UK  US
## 19.6 14.2 8.7 7.0

round(with(Ornstein, tapply(interlocks, sector, mean)), 1)

## AGR  BNK  CON  FIN  HLD  MAN  MER  MIN  TRN  WOD
## 8.4 53.9 4.8 24.1 12.4 7.0 10.2 12.9 19.2 16.9
```

In the original study, the author performed an OLS regression with the response variable being `interlocks`, the number of interlocks maintained by each firm, on firm's assets (in millions of dollars), sector of operation, and nation of control.

However, as the variable `interlocks` is a count, a *Poisson* regression model may be more appropriate.

Let's take a look at the distribution of the `interlocks` variable. We will have to construct this graph in multiple steps.

Analysis of the Response Variable

```
# Frequency distribution of the interlocks
tab <- xtabs(~interlocks, data=Ornstein)
str(tab)      # this view is less useful for a table

## 'xtabs' int [1:50(1d)] 28 19 14 11 8 14 11 6 12 7 ...
## - attr(*, "dimnames")=List of 1
## ..$ interlocks: chr [1:50] "0" "1" "2" "3" ...
## - attr(*, "call")= language xtabs(formula = ~interlocks, data = Ornstein)

class(tab)    # check class (so we know what methods and attributes are available)

## [1] "xtabs" "table"

names(tab)    # variable names

## [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
## [12] "11" "12" "13" "14" "15" "16" "17" "18" "19" "20" "21"
## [23] "22" "23" "25" "27" "28" "29" "30" "31" "32" "33" "34"
## [34] "35" "36" "39" "40" "42" "43" "44" "46" "48" "51" "55"
## [45] "66" "69" "77" "87" "94" "107"

nrow(tab)

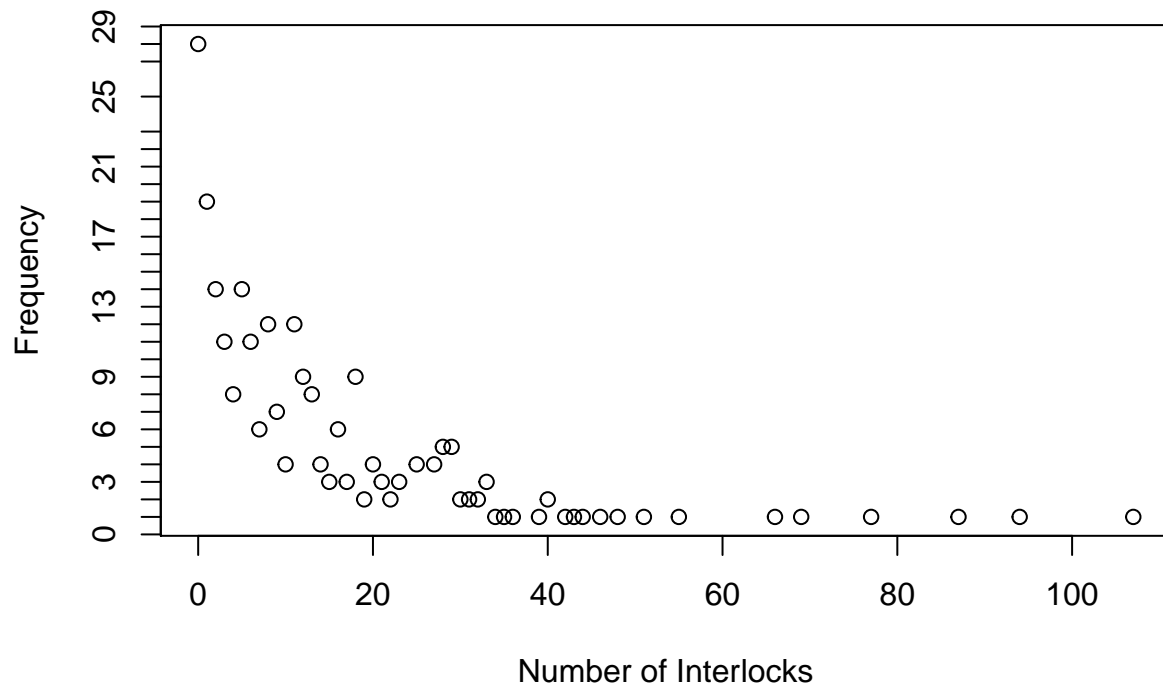
## [1] 50

tab

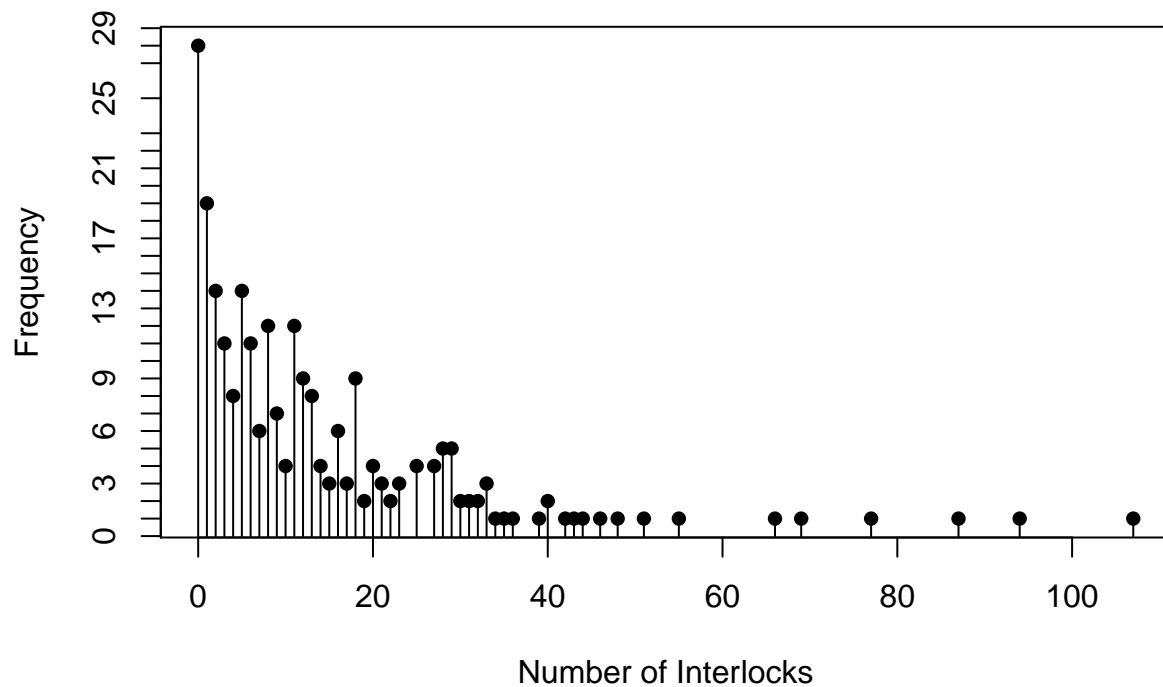
## interlocks
##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
## 28 19 14 11  8 14 11  6 12  7  4 12  9  8  4  3  6  3
## 18 19 20 21 22 23 25 27 28 29 30 31 32 33 34 35 36 39
##  9  2  4  3  2  3  4  4  5  5  2  2  2  3  1  1  1  1
## 40 42 43 44 46 48 51 55 66 69 77 87 94 107
##  2  1  1  1  1  1  1  1  1  1  1  1  1  1

# Record the distinct values of the interlocks
x <- as.numeric(names(tab)) # xtabs stored these as char variables

# Scatter plot of data
plot(x, tab, xlab='Number of Interlocks', ylab='Frequency')
```

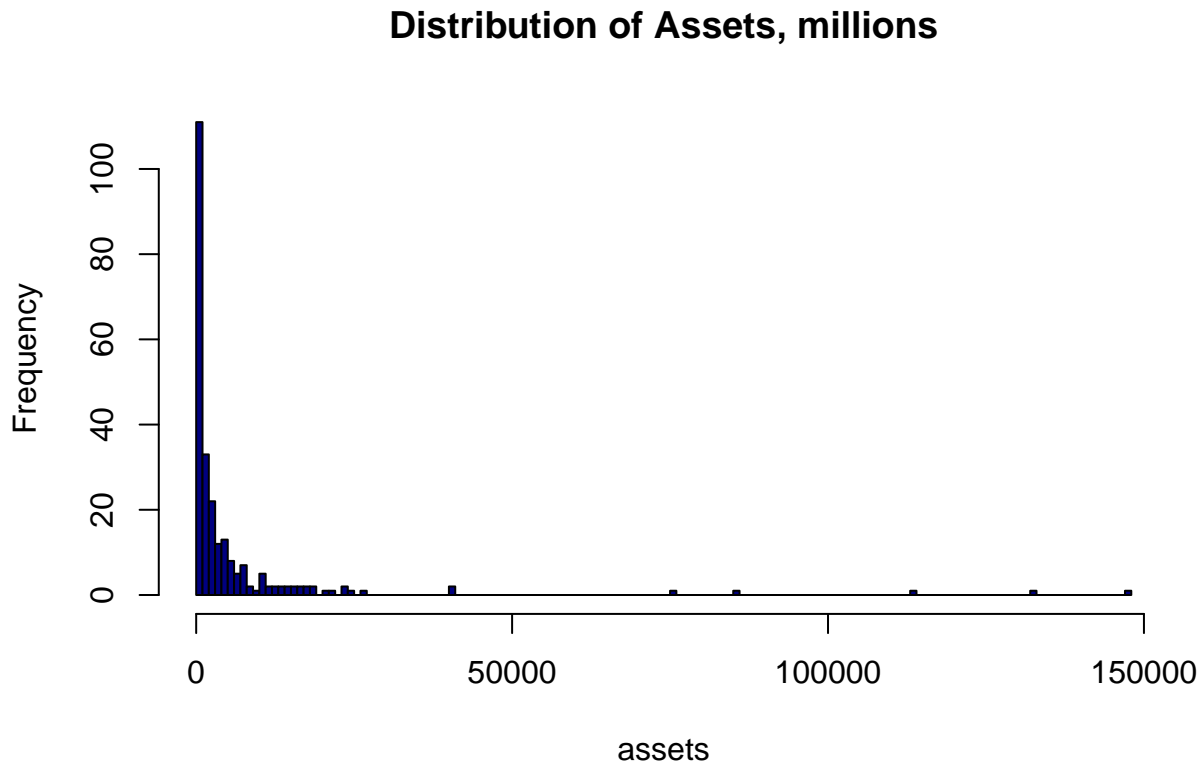


```
# Bar plot of frequencies
plot(x, tab, type='h', xlab='Number of Interlocks', ylab='Frequency')
points(x, tab, pch=16) # decorate top of bars with a point
```



Quick Analysis of Exploratory Variables

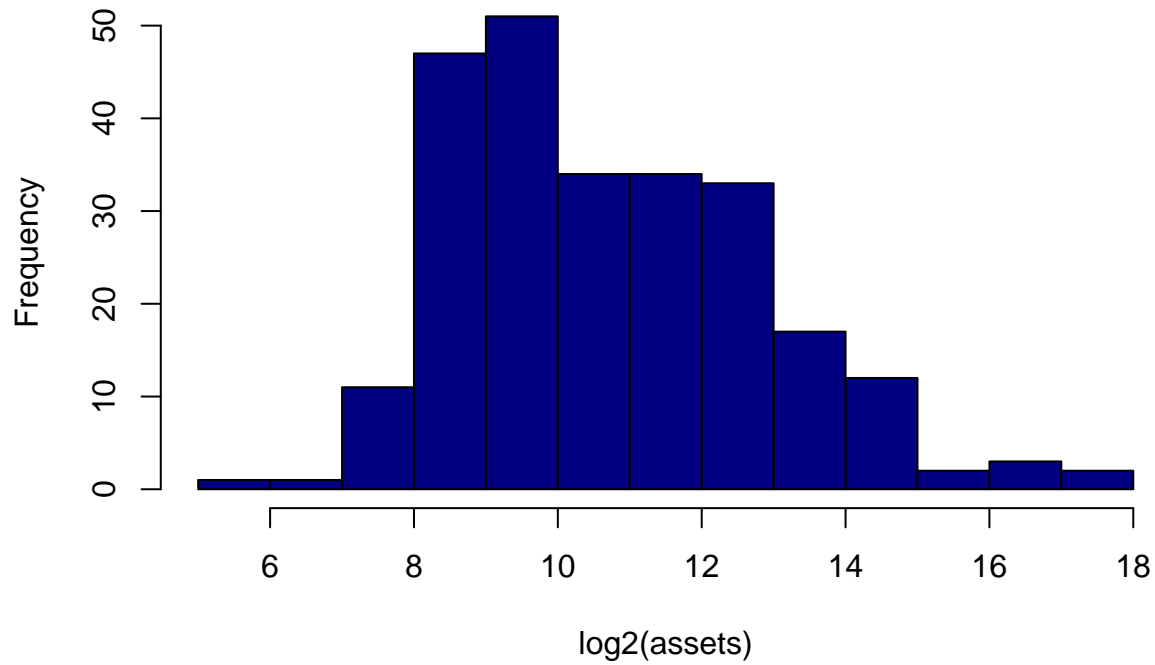
```
# Histogram (untransformed)
with(Ornstein, hist(assets, breaks='FD', col='navy', main='Distribution of Assets, millions'))
```



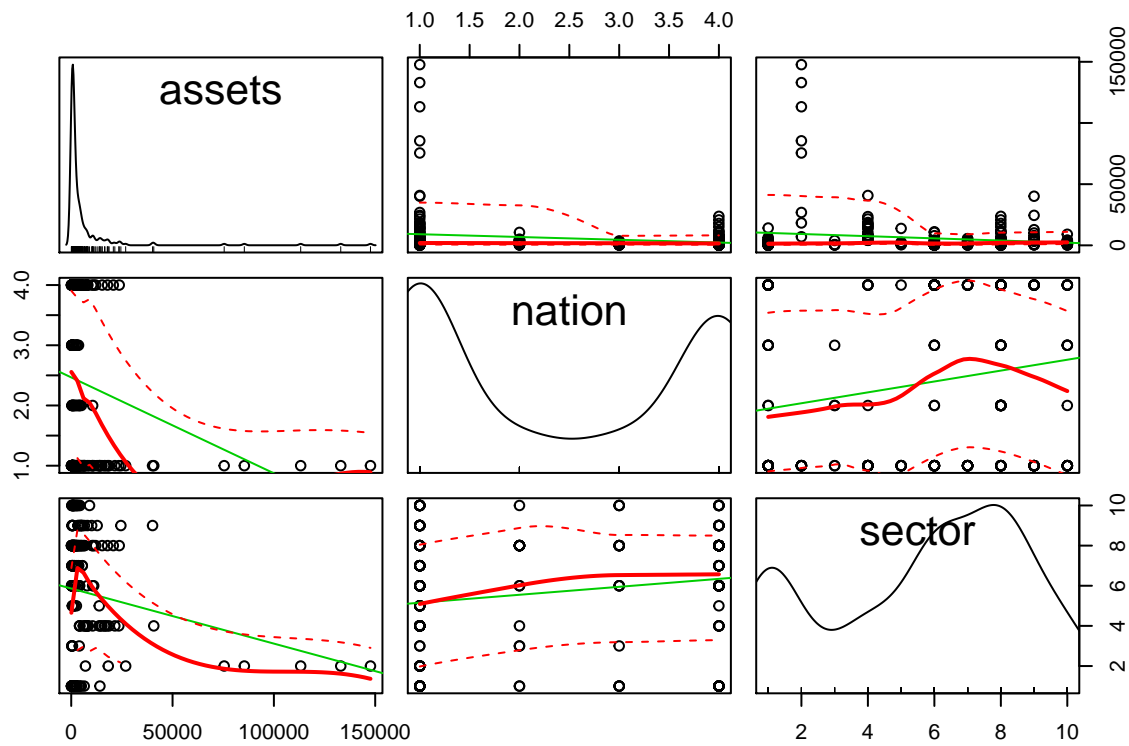
Shape suggests a log transform may help the distribution be more approximately normal.

```
# Histogram (log2 transformed)
with(Ornstein, hist(log2(assets), breaks='FD', col='navy', main='Distribution of Log_2(Assets)'))
```

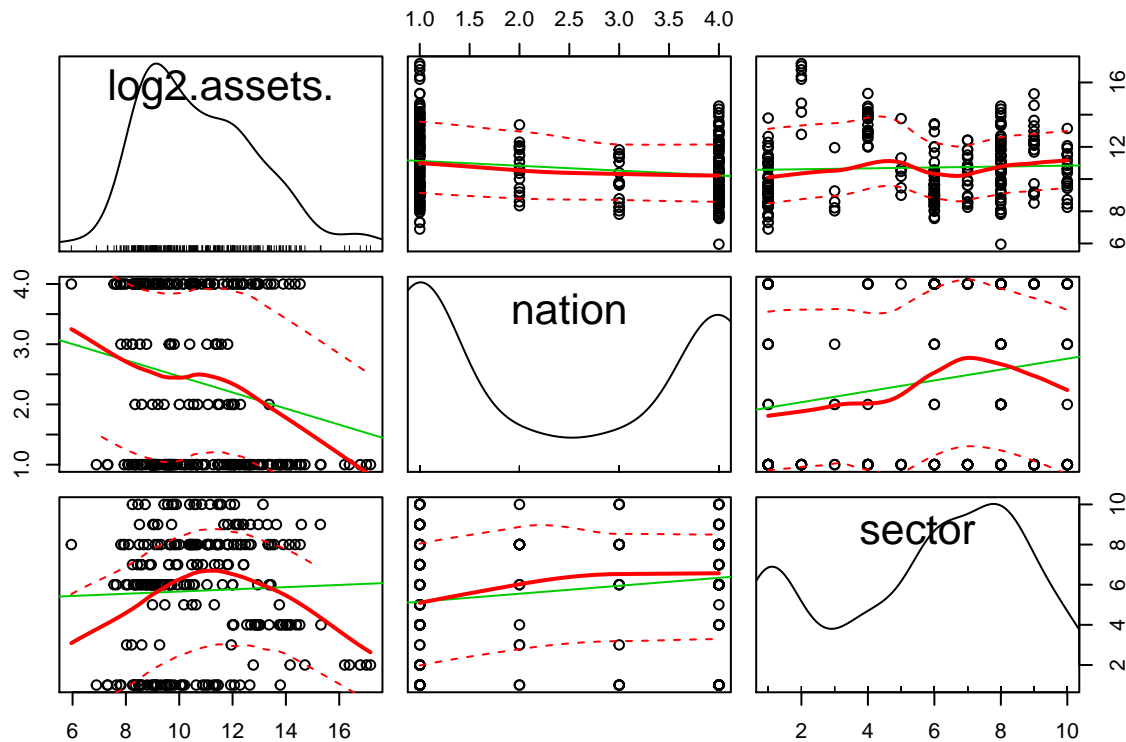
Distribution of Log₂(Assets), millions



```
# scatterplot matrices to look at variable correlations
scatterplotMatrix(~assets + nation + sector, data=Ornstein)
```

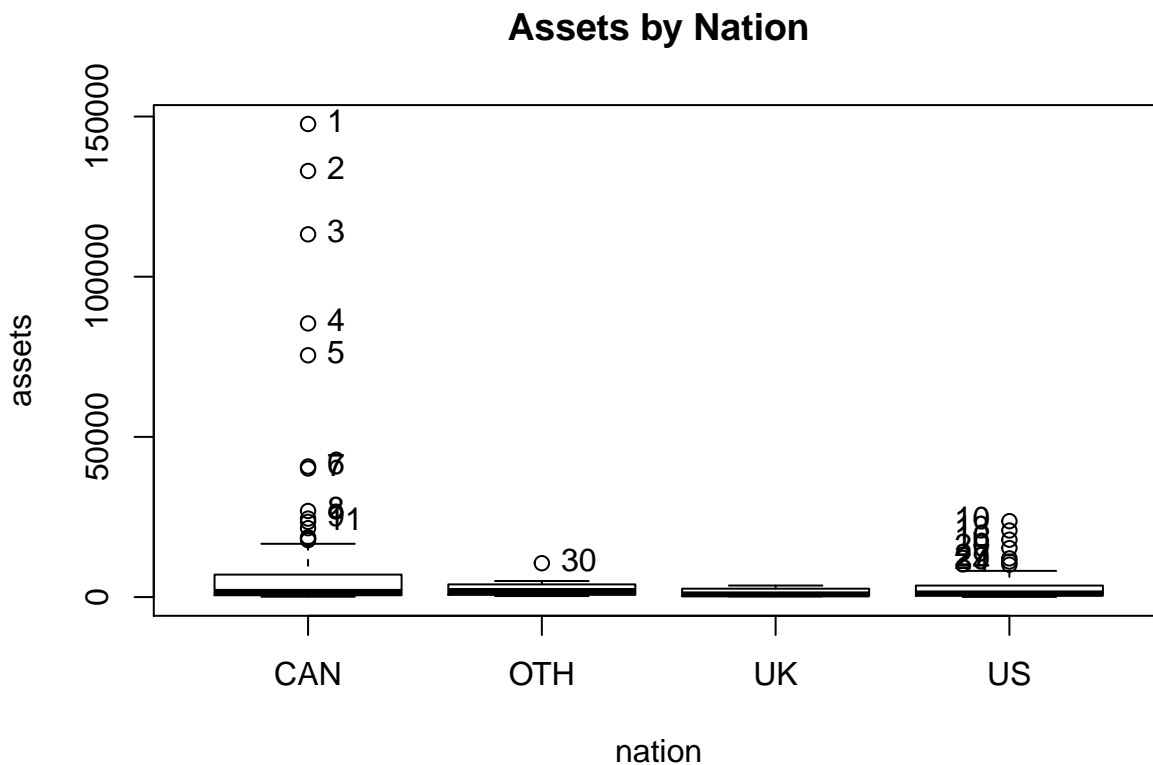


```
scatterplotMatrix(~log2(assets) + nation + sector, data=Ornstein)
```



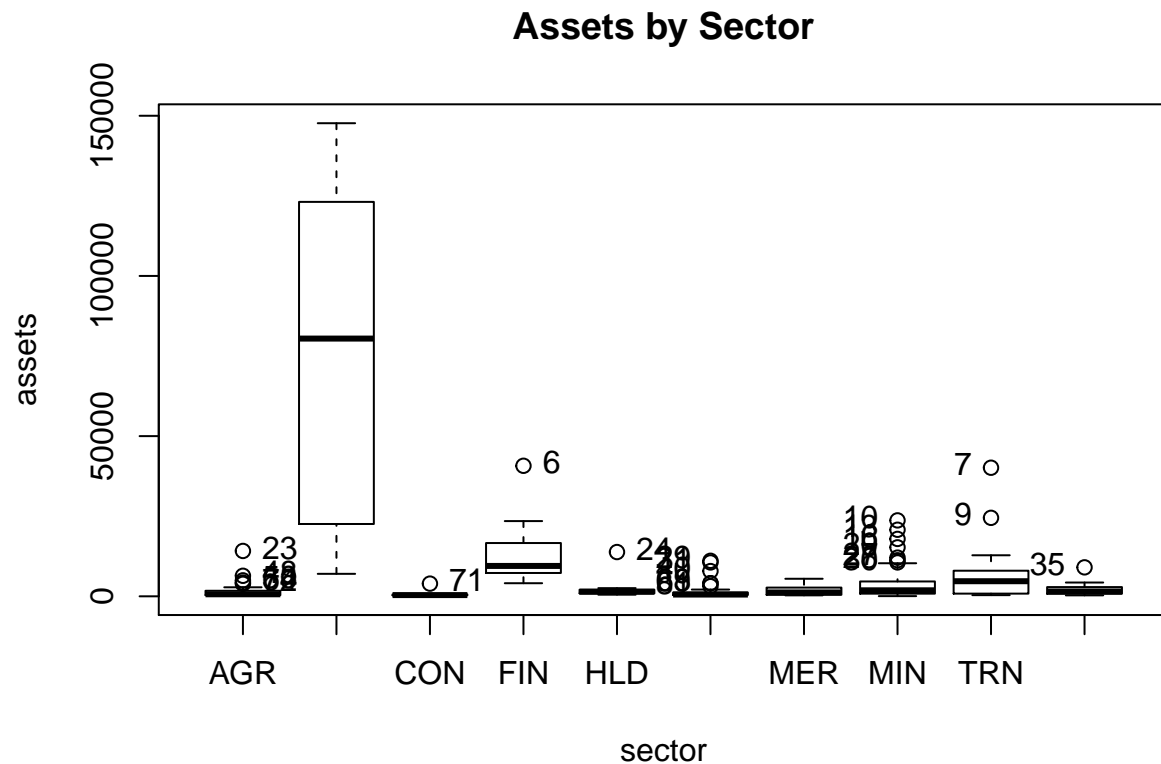
Note: this is NOT a good approach. Scatterplot matrices don't make a lot of sense in this case since explanatory variables are categorical. SCM doesn't display the features very well. Instead, use boxplots:

```
Boxplot(assets ~ nation, data=Ornstein, main='Assets by Nation')
```



```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "11" "30" "10" "13" "16"
## [15] "20" "27" "29" "34"
```

```
Boxplot(assets ~ sector, data=Ornstein, main='Assets by Sector')
```



```
## [1] "23" "48" "58" "64" "72" "71" "6" "24" "29" "31" "40" "68" "69" "81"
## [15] "10" "13" "16" "20" "27" "28" "30" "7" "9" "35"
```

Observations: Because `assets` is extremely right-skewed, we see the extended tail of points for CAN. Banking sector has the largest assets, and also extremely wide distribution. Since data is extremely right-skewed, we may want to look at modified plots: 95th percentile assets; excluding banking.

```
summary(Ornstein$assets)
```

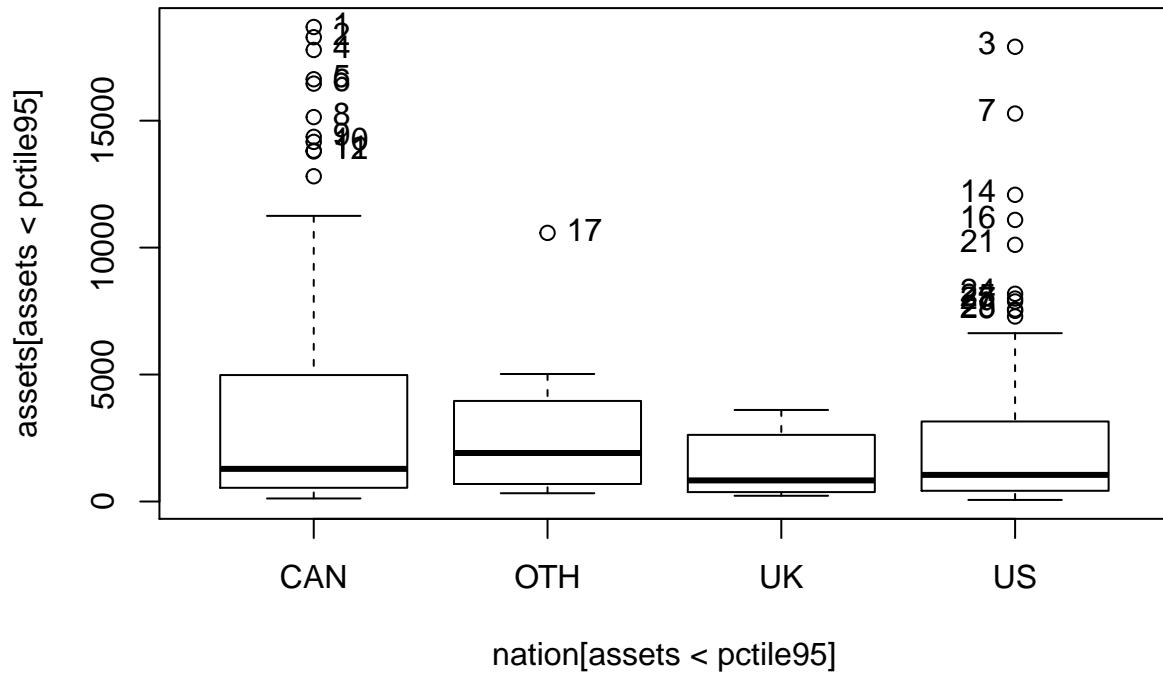
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      62     519     1397    5978    4326   147670
```

```
(pctile95 <- quantile(Ornstein$assets, 0.95))
```

```
##      95%
## 20047.8
```

```
Boxplot(assets[assets<pctile95] ~ nation[assets<pctile95], data=Ornstein, main='Assets by Nation')
```

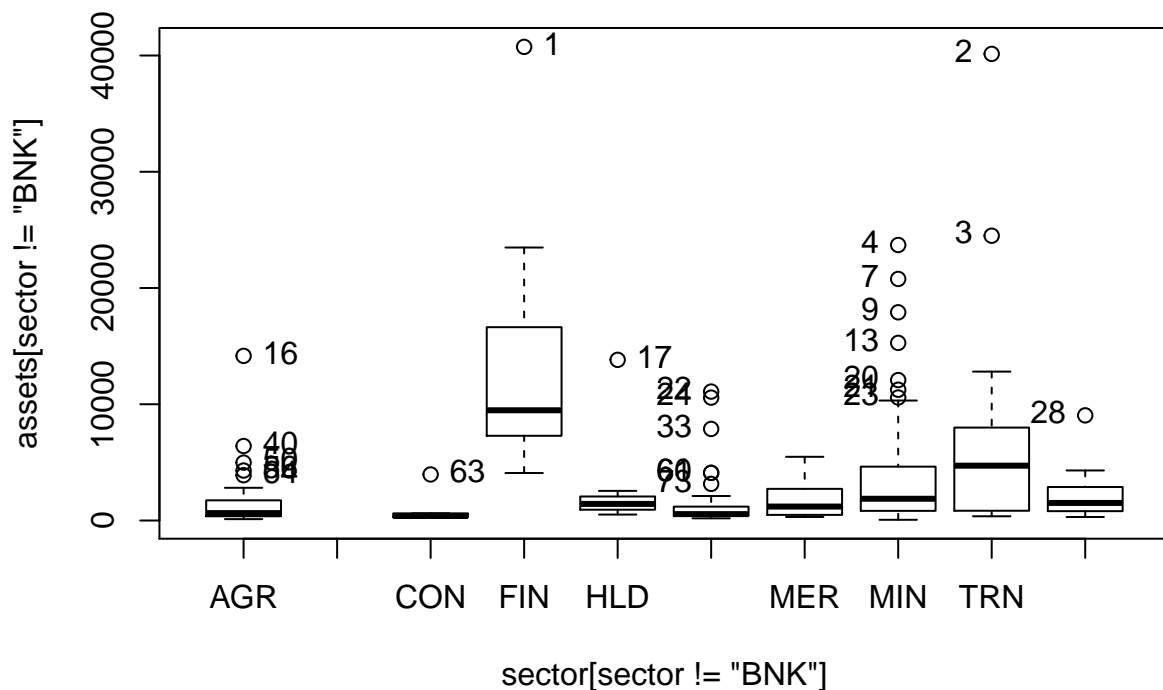

Assets by Nation (Assets < ~\$20B)



```
## [1] "1" "2" "4" "5" "6" "8" "9" "10" "11" "12" "17" "3" "7" "14"
## [15] "16" "21" "24" "25" "27" "28" "29"
```

```
Boxplot(assets[sector!='BNK'] ~ sector[sector!='BNK'], data=Ornstein, main='Assets by Sector (Excluding Banking)')
```

Assets by Sector (Excluding Banking)



```
## [1] "16" "40" "50" "56" "64" "63" "1" "17" "22" "24" "33" "60" "61" "73"
## [15] "4" "7" "9" "13" "20" "21" "23" "2" "3" "28"
```

Poisson Regression

Now, build the regression model. Note that we're using the log-transformation of `assets`.

```
poisson_fit <- glm(interlocks ~ log2(assets) + nation + sector, family=poisson(link=log), data=
summary(poisson_fit)
```

```
##
## Call:
## glm(formula = interlocks ~ log2(assets) + nation + sector, family = poisson(link = log),
##      data = Ornstein)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6.7111  -2.3159  -0.4595   1.2824   6.2849
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.83938    0.13664  -6.143 8.09e-10 ***
## log2(assets)   0.31292    0.01177  26.585 < 2e-16 ***
## nationOTH     -0.10699    0.07438  -1.438 0.150301
## nationUK      -0.38722    0.08951  -4.326 1.52e-05 ***
## nationUS      -0.77239    0.04963 -15.562 < 2e-16 ***
## sectorBNK     -0.16651    0.09575  -1.739 0.082036 .
## sectorCON     -0.48928    0.21320  -2.295 0.021736 *
## sectorFIN     -0.11161    0.07571  -1.474 0.140457
## sectorHLD     -0.01491    0.11924  -0.125 0.900508
## sectorMAN      0.12187    0.07614   1.600 0.109489
## sectorMER      0.06157    0.08670   0.710 0.477601
## sectorMIN      0.24985    0.06888   3.627 0.000286 ***
## sectorTRN      0.15181    0.07893   1.923 0.054453 .
## sectorWOD      0.49825    0.07560   6.590 4.39e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 3737.0  on 247  degrees of freedom
## Residual deviance: 1547.1  on 234  degrees of freedom
## AIC: 2473.1
##
## Number of Fisher Scoring iterations: 5
```

Observations: We can look at the summary to see which are the (omitted) base cases. In this case, it's `nation=CAN` and `sector=AGR`. Also, remember that the coefficients are for the *linear predictor*,

ie. for the $\log(\text{counts})$. We have to exponentiate to get effect on the counts.

We can also pull out residual deviance from the model and calculate a “goodness of fit” p-value. The deviance and degrees of freedom come from the fit. The null hypothesis H_0 is that the data are consistent with the specified distribution; H_a is that they are not.

```
resid_dev <- with(poisson_fit, cbind(resid_deviance = deviance, df = df.residual,
                                     p = pchisq(deviance, df.residual, lower.tail = FALSE)))
round(resid_dev, 4)
```

```
##      resid_deviance df p
## [1,]      1547.083 234 0
```

Because the goodness of fit χ^2 test is statistically significant, we reject the null hypothesis and conclude the model does not fit the data particularly well.

Let’s look at the frequency table.

```
# Create frequency table
with(Ornstein, table(nation, sector))
```

```
##      sector
## nation AGR BNK CON FIN HLD MAN MER MIN TRN WOD
##   CAN  27   8   2  17   6  14  13   9  11  10
##   OTH   2   0   2   1   0   2   0  10   0   1
##   UK    4   0   1   0   0   3   0   6   0   3
##   US   14   0   0   4   1  29   7  29   7   5
```

Observations: Our reference category (CAN + AGR) has a substantial number of observations. This is good; if our base case had a small number, we might be better off changing the base case.

** Analysis of Deviance

```
Anova(poisson_fit)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: interlocks
##              LR Chisq Df Pr(>Chisq)
## log2(assets)   731.21  1 < 2.2e-16 ***
## nation         276.04  3 < 2.2e-16 ***
## sector         102.71  9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observations: Capital Anova does a Type II analysis. All predictors have high significance.

Interpretation of Coefficients

Coefficients of the model are interpreted as effects on the log-count scale (ie. the scale of the linear predictor). Exponentiating the coefficients produces the multiplicative effects on the count scale.

```
exp(coef(poisson_fit))
```

```
## (Intercept) log2(assets) nationOTH nationUK nationUS
## 0.4319777 1.3674101 0.8985317 0.6789410 0.4619077
## sectorBNK sectorCON sectorFIN sectorHLD sectorMAN
## 0.8466116 0.6130649 0.8943943 0.9852027 1.1296019
## sectorMER sectorMIN sectorTRN sectorWOD
## 1.0635096 1.2838282 1.1639375 1.6458464
```

Observations: a corporation that is twice the size (based on assets) of another one has an estimated 36.7% higher number of interlocks, holding all other factors constant.

A US firm on average maintains only 46.2% as many interlocks as a Canadian firm, holding all other factors constant.

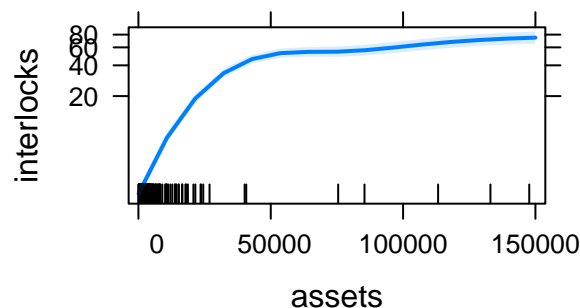
Visualize the Effects of Changes in Explanatory Variables

Effects plots show how estimated responses change with predictor values. (These are not diagnostic plots.)

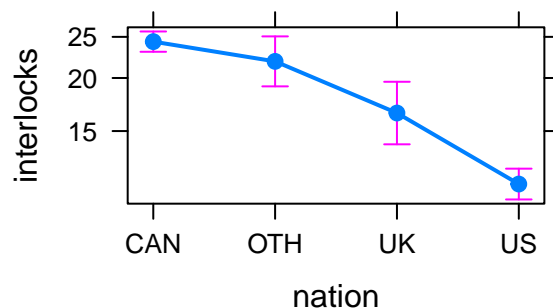
One variable at a time is changed, while others are held constant at “typical” values. (For continuous variables, at their mean; for factor variables, sets proportional distribution to match that observed in data.)

```
plot(allEffects(poisson_fit, default.levels=50))
```

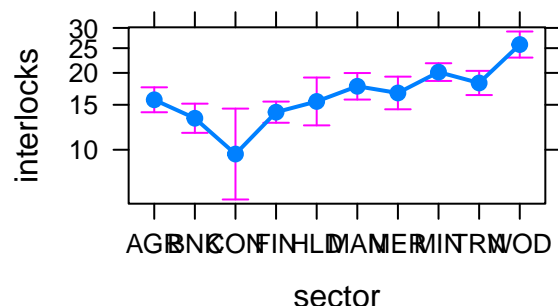
assets effect plot



nation effect plot



sector effect plot

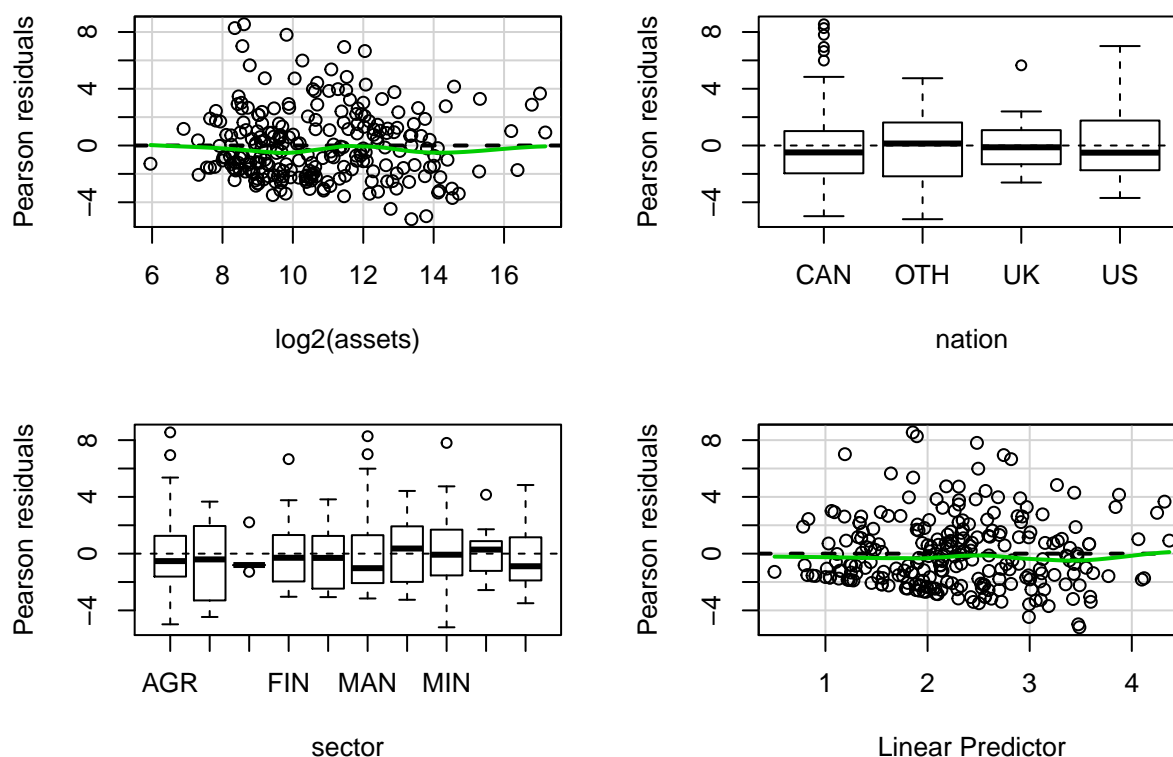


The y-axis is on the scale of the linear predictor (in this case, `log`). The bands around the means are 95% confidence intervals, calculated using standard errors from the model. From the y-axis range of data, we can get a sense of strength of each predictor's effect; we can also get a sense of level of uncertainty in the estimates.

Model Diagnostics

The diagnostic plots for a Poisson model are different than for linear regression. These plots use Pearson residuals. There is one plot per explanatory variable, as well as one for the linear predictor. We want to see that there are *no systematic patterns* between residuals and the explanatory variables. We should look for nonlinear trends (esp. curved splines), trends in variation across the graph, and outlier points. Box plots should show roughly similar centers and spreads.

```
residualPlots(poisson_fit, layout=c(2,2))
```



```
##          Test stat Pr(>|t|)
## log2(assets)    15.745      0
## nation          NA      NA
## sector          NA      NA
```

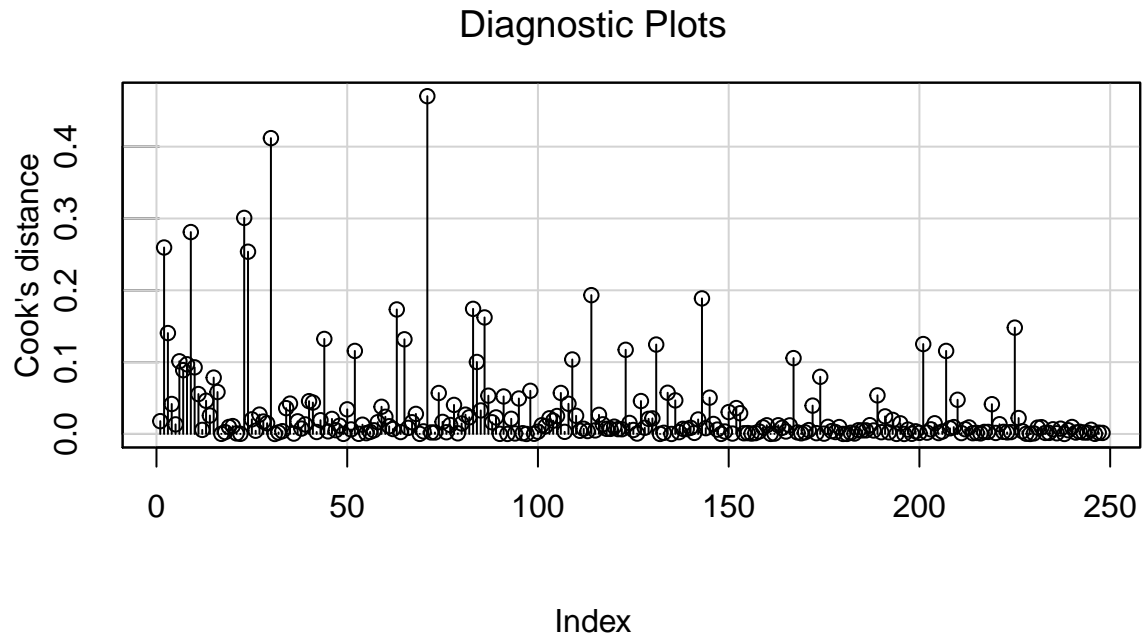
Observations: not much pattern observed here. The *lack of fit test* results are also displayed (only meaningful for continuous variables, otherwise NA). According to the R help text, this is a curvature test. For plots against a term in the model formula, say X , the test displayed is the t-test for $I(X^2)$ in the fit of an updated model: $\sim . + I(X^2)$.

The null hypothesis of this test is H_0 : the model fits the data well (ie. the coefficient for a curvature term $\beta_{I(X^2)} = 0$), vs. H_a : the coefficient for a curvature term is non-zero. Here, the *p-value* of ~ 0

indicates a lack of fit, even though it's not as evident from visual inspection.

We can also look for influential points:

```
influenceIndexPlot(poisson_fit, vars = "Cook")
```



```
influenceIndexPlot(poisson_fit, vars = "hat")
```

