

STAT 442, Fall 2008 Model Solutions for Assignment 3

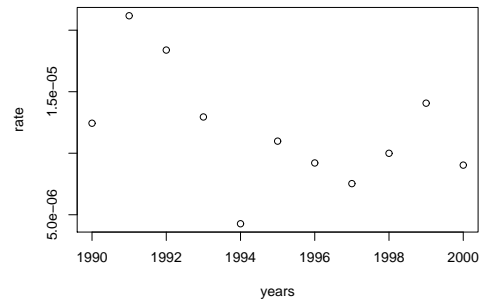
N.B. The following solutions should not be taken as the only "correct solution". In data analysis problems there is invariably more than one appropriate way to frame the analysis. If a number of different approaches lead to the same general conclusion we can be more confident in our findings.

Question 1

(a)

We read in the counts from the command line using `scan()` and then read the population figures from the provided CSV file. Overlooking the fact that the file actually has yearly totals for the state, I have calculated them by summing up the subtotals for the 14 counties.

```
> murders <- c(14, 24, 21, 15, 5, 13, 11, 9, 12, 17, 11)
> pops <- read.csv("vtPop.csv", skip = 2, row.names = 1)
> years <- 1990:2000
> totPop <- apply(pops[as.character(years), ], 1, sum)
> rate <- murders/totPop
> plot(years, rate)
```



1

The one notable feature of the graph is an apparent downward trend in the homicide rate.

(b)

```
> llfit <- glm(murders ~ offset(log(totPop)) + years, family = poisson)
> summary(llfit)
```

Call:

```
glm(formula = murders ~ offset(log(totPop)) + years, family = poisson)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.87088	-0.72958	-0.03086	0.68967	1.69458

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	105.18946	51.63454	2.037	0.0416 *
years	-0.05842	0.02589	-2.257	0.0240 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 22.656 on 10 degrees of freedom
Residual deviance: 17.520 on 9 degrees of freedom
AIC: 69.948

Number of Fisher Scoring iterations: 4

```
> coefTable(llfit, transf = exp)
```

	Estimate	Std. Error	Pr(> z)	2.5 %	97.5 %
(Intercept)	4.821700e+45	51.63454308	0.04163085	53.9259066	4.311248e+89
years	9.432513e-01	0.02588843	0.02402647	0.8965843	9.923473e-01

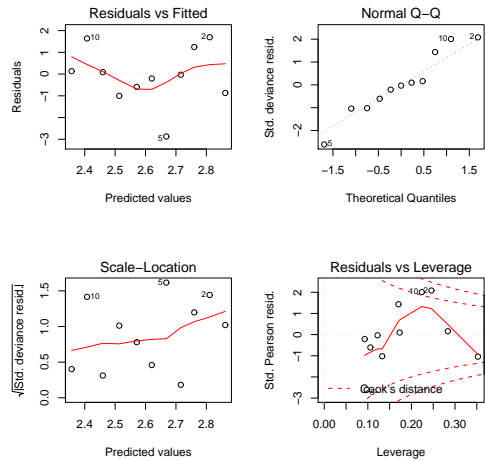
In the summary above with parameters in their logarithmic form, there is an indication of a roughly 6% (rounding $100 \times .05842$) decrease from year

2

to year in the rate, which is statistically significant ($P = .03$) Using my `coefTable()` function to provide transformed estimates and confidence intervals, we see that $e^{-.05842} \approx .94$, again indicating a 6% decrease from year to year.

Next we examine the fit.

```
> par(mfrow = c(2, 2))
> plot(llfit)
```



There is a weak indication of non-linearity, which seems mainly due to a single low value, that corresponds to year of 1994. That value appears as an outlier in the normal QQ plot, but because the counts are small the normal approximation to the Poisson distribution is not particularly relevant. However, the Poisson distribution tends to be skewed to the right, so an outlier in the left tail is still somewhat anomalous.

(c)

We can compute the vector of values required for the score statistic, z_1 , described by Simonoff, and apply the `t.test()` function to perform the test. The over-dispersion parameter estimate follows the form of a goodness-of-fit statistic divided by its' degrees of freedom.

```
> cat("Overdispersion estimate:", round(1/9 * sum((murders - fitted(llfit))^2/fitted(llfit) + 2), fill = TRUE)
```

Overdispersion estimate: 1.81

```
> z1Vec <- ((murders - fitted(llfit))^2 - murders)/(fitted(llfit))
> t.test(z1Vec)
```

One Sample t-test

```
data: z1Vec
t = 0.776, df = 10, p-value = 0.4557
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.8908214 1.8428610
sample estimates:
mean of x
0.4760198
```

The estimated overdispersion is 1.81, which taken at face value is "practically" significant, since it would lead to nearly doubling the standard errors. However, the score test yields $P = .46$, so one can't be sure. It is prudent to examine the potential impact of an adjustment for over-dispersion, as below:

```
> llfitQ <- glm(murders ~ offset(log(totPop)) + years, family = quasipoisson)
> summary(llfitQ)
```

```
Call:
glm(formula = murders ~ offset(log(totPop)) + years, family = quasipoisson)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.87088 -0.72958 -0.03086 0.68967 1.69458
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	105.18946	69.40641	1.516	0.164
years	-0.05842	0.03480	-1.679	0.127

(Dispersion parameter for quasipoisson family taken to be 1.806835)

Null deviance: 22.656 on 10 degrees of freedom
Residual deviance: 17.520 on 9 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4

We see that the estimates do not change but the trend is no longer significant if we allow for over-dispersion.

(d)

The conflicting results from the two analysis would lead me to observe that there is an indication of a decline, with some uncertainty as to whether it is "real". It may simply be an artefact of chance fluctuations.

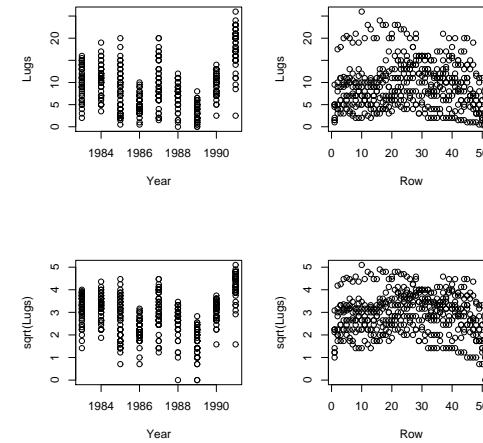
2.

(a)

Plotting the data as directed yields:

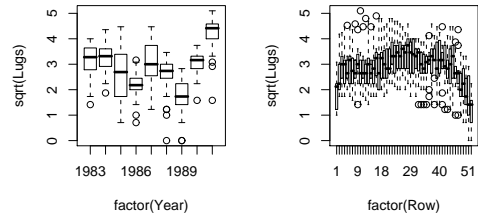
```
> par(mfrow = c(2, 2))
> df <- read.csv("vineyard.csv")
> attach(df)
> par(mfrow = c(2, 2))
> plot(Lugs ~ Year)
> plot(Lugs ~ Row)
> plot(sqrt(Lugs) ~ Year)
> plot(sqrt(Lugs) ~ Row)
```

5



```
> par(mfrow = c(1, 2))
> plot(sqrt(Lugs) ~ factor(Year))
> plot(sqrt(Lugs) ~ factor(Row))
```

6

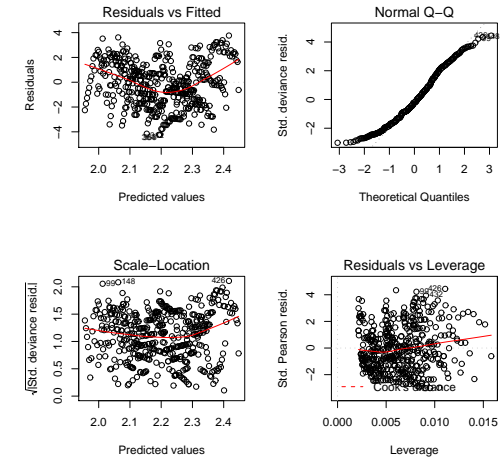


The differences in the plot of the "raw" and square root transformed data are not huge, though the square root transformation has "dragged down" the larger values, reduced apparent skewness and made the curvature in the relationship between the response and Row more apparent. The boxplots make the up and down trends from year to year easier to see and re-inforce the impression that there is a decreasing trend from row 40 onwards.

(b)

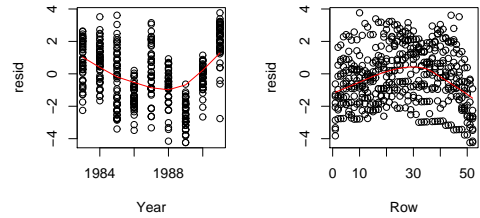
We begin by examining the default plots produced by the generic `plot()` function.

```
> llfit <- glm(Lugs ~ Year + Row, family = poisson)
> par(mfrow = c(2, 2))
> plot(llfit)
```



Because we have two explanatory variables it is informative to plot residuals against each of them. I've added a lowess smoother using the default values that are coded into `plot.lm()`, which is the function that actually gets called when `plot()` is applied to objects produced by `lm()` or `glm()`.

```
> par(mfrow = c(1, 2))
> resid <- residuals(llfit)
> plot(Year, resid)
> lines(lowess(Year, resid, f = 2/3, iter = 3), col = "red")
> plot(Row, resid)
> lines(lowess(Row, resid, f = 2/3, iter = 3), col = "red")
```

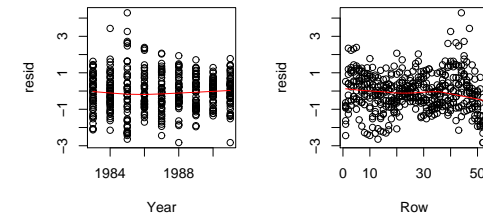


There is clear indication of non-linearity in both Year and Row.

(c)

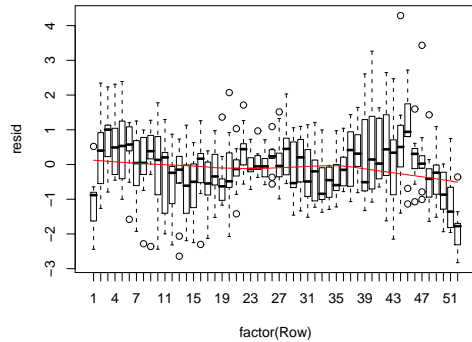
To allow for non-linearity in the Row variable we'll use polynomial functions. The pattern in Year will be accommodated by entering it as a factor, following the directions in the assignment. The pattern in Row in the smooth in the residual plot looks roughly quadratic. As a start we consider a quadratic model, and then look at the residuals plotted against the Year and Row

```
> llfitp2 <- glm(Lugs ~ factor(Year) + poly(Row, 2), family = poisson)
> resid <- residuals(llfitp2)
> par(mfrow = c(1, 2))
> plot(Year, resid)
> lines(lowess(Year, resid, f = 2/3, iter = 3), col = "red")
> plot(Row, resid)
> lines(lowess(Row, resid, f = 2/3, iter = 3), col = "red")
```



No pattern of non-linearity is evident in the plot for Year, since the taking Year as a categorical variable ensures that the residuals are centered around 0 for each year. The smoother in the residual plot looks pretty linear, but the downward trend evident of the points at the end of the plot suggests that a higher order polynomial is required to accurately reflect the pattern. Using boxplots (see below) confirms this suspicion.

```
> plot(resid ~ factor(Year))
> lines(lowess(Row, resid, f = 2/3, iter = 3), col = "red")
```



Rather than proceeding by trial and error, I fit a sequence of higher order polynomials, and then use the C_p criterion (equivalent to AIC) to decide on the appropriate degree.

```
> llfitp3 <- glm(Lugs ~ factor(Year) + poly(Row, 3), family = poisson)
> llfitp4 <- glm(Lugs ~ factor(Year) + poly(Row, 4), family = poisson)
> llfitp5 <- glm(Lugs ~ factor(Year) + poly(Row, 5), family = poisson)
> llfitp6 <- glm(Lugs ~ factor(Year) + poly(Row, 6), family = poisson)
> llfitp7 <- glm(Lugs ~ factor(Year) + poly(Row, 7), family = poisson)
> llfitp8 <- glm(Lugs ~ factor(Year) + poly(Row, 8), family = poisson)
> llfitp9 <- glm(Lugs ~ factor(Year) + poly(Row, 9), family = poisson)
> llfitp10 <- glm(Lugs ~ factor(Year) + poly(Row, 10), family = poisson)
> llfitp11 <- glm(Lugs ~ factor(Year) + poly(Row, 11), family = poisson)
> llfitp12 <- glm(Lugs ~ factor(Year) + poly(Row, 12), family = poisson)
> anova(llfitp2, llfitp3, llfitp4, llfitp5, llfitp6, llfitp7, llfitp8,
+       llfitp9, llfitp10, llfitp11, llfitp12, test = "Cp")
```

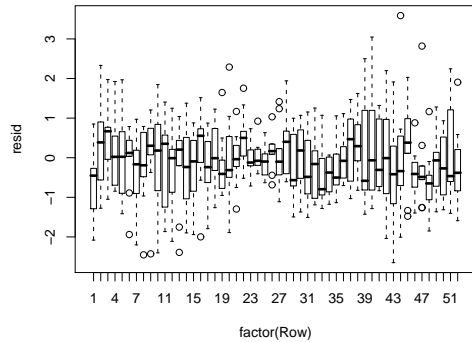
Analysis of Deviance Table

Model 1: Lugs ~ factor(Year) + poly(Row, 2)

Model	Resid. Df	Resid. Dev	Df	Deviance	Cp
Model 2: Lugs ~ factor(Year) + poly(Row, 3)	457	507.82			529.82
Model 3: Lugs ~ factor(Year) + poly(Row, 4)	456	486.11	1	21.71	510.11
Model 4: Lugs ~ factor(Year) + poly(Row, 5)	455	466.23	1	19.88	492.23
Model 5: Lugs ~ factor(Year) + poly(Row, 6)	454	459.98	1	6.25	487.98
Model 6: Lugs ~ factor(Year) + poly(Row, 7)	453	422.58	1	37.40	452.58
Model 7: Lugs ~ factor(Year) + poly(Row, 8)	452	415.73	1	6.85	447.73
Model 8: Lugs ~ factor(Year) + poly(Row, 9)	451	415.71	1	0.02	449.71
Model 9: Lugs ~ factor(Year) + poly(Row, 10)	450	415.61	1	0.10	451.61
Model 10: Lugs ~ factor(Year) + poly(Row, 11)	449	411.84	1	3.77	449.84
Model 11: Lugs ~ factor(Year) + poly(Row, 12)	448	411.27	1	0.57	451.27
	447	411.01	1	0.26	453.01

The minimum value of C_p occurs for the 6th degree polynomial. The corresponding boxplots of residuals versus Row indicates that the fit is good.

```
> resid <- residuals(llfitp6)
> plot(resid ~ factor(Row))
```



(d)

We begin by adding the interaction between the Year and Row terms from our previous model, and testing the significance of the interaction.

```
> llfitp6x <- glm(Lugs ~ factor(Year) * poly(Row, 6), family = poisson)
> anova(llfitp6x, test = "Chi")
```

Analysis of Deviance Table

Model: poisson, link: log

Response: Lugs

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	P(> Chi)
NULL			467	1485.76	

13

factor(Year)	8	780.57	459	705.19	3.165e-163
poly(Row, 6)	6	282.61	453	422.58	4.335e-58
factor(Year):poly(Row, 6)	48	280.49	405	142.09	1.366e-34

Just to be safe, we might want to check this by considering the possibility of overdispersion.

```
> quasifitp6x <- glm(Lugs ~ factor(Year) * poly(Row, 6), family = quasipoisson)
> anova(quasifitp6x, test = "F")
```

Analysis of Deviance Table

Model: quasipoisson, link: log

Response: Lugs

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	F	Pr(>F)
NULL			467	1485.76		
factor(Year)	8	780.57	459	705.19	281.596	< 2.2e-16
poly(Row, 6)	6	282.61	453	422.58	135.939	< 2.2e-16
factor(Year):poly(Row, 6)	48	280.49	405	142.09	16.865	< 2.2e-16

NULL

factor(Year) ***

poly(Row, 6) ***

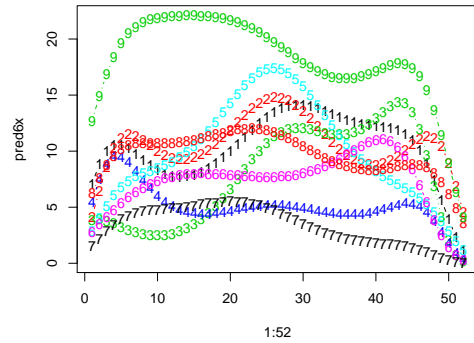
factor(Year):poly(Row, 6) ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Either way, there are indications of interaction. To examine the nature of the interaction, we follow directions and apply `matplot()`.

```
> pred6x <- matrix(fitted(llfitp6x), nrow = 52)
> matplot(1:52, pred6x, type = "b")
```

14



We see that the rows from about 44 on seem to do consistently poorly compared with the majority of rows. However, we see that the pattern in the 9th year (1991) may explain the interaction, since the Rows 5 to 20 were the highest producers in that year, while in other years they were amongst the lowest producers.