

Econometrics

Rob Hayward

October 15, 2014

Asymptotic theory

As an overview of asymptotic theory. [Dave Giles](#) uses a monte carlo experiment to illustrate the behaviour of the OLS estimator.

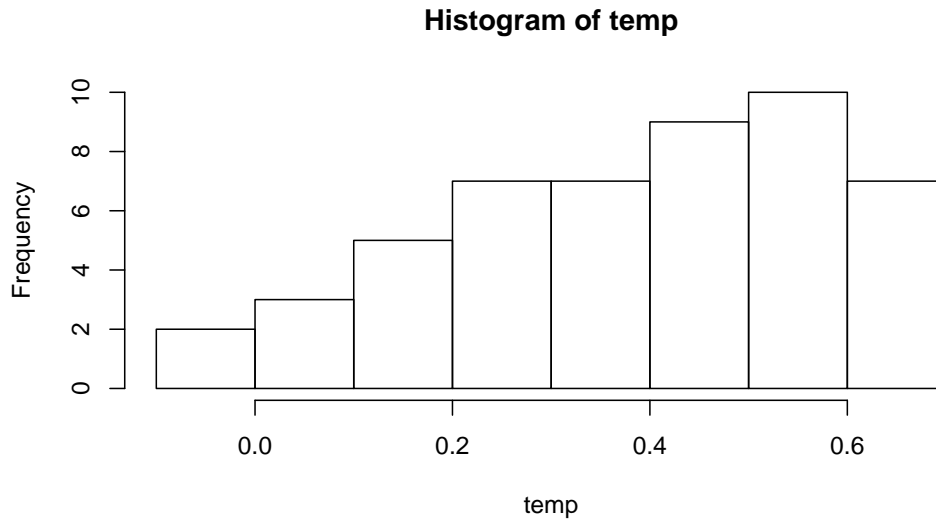
$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t, \quad t = 2, 3, \dots \quad y_1 = 0$$

ε is generated according to a *uniform* distribution on the interval $(-1, +1)$.

The attention here is on β_1 . As the estimator is a sample statistic (a function of random sample data) this distribution is the *sampling distribution*. The estimate may change as n changes. The form of the distribution also depends on the size of the sample.

```
Asymp <- function(n, reps){
  y <- rep(NA, n)
  y[1] <- 0
  temp <- rep(NA, reps)
  for(j in 1:reps){
    for(i in 2:n){
      e <- runif(n, -1, 1)
      y[i] <- 1 + 0.5*y[i-1] + e[i]
    }
  }
  ylag <- lag(y)
  da <- data.frame(y[2:n], y[1:n-1])
  colnames(da) <- c("y", "ylag")
  eq <- lm(y ~ ylag, data = da)
  temp[j] <- eq$coefficients[2]
}
hist(temp)
```

```
mean(temp)
}
Asymp(20, 50)
```



```
## [1] 0.3807
```

The estimate of the coefficient of the lagged dependent variable is biased downwards. There is also a negative skew to the estimate of the coefficient on the dependent variable. The estimator of β_1 is *consistent*. This means that as the sample gets bigger, the estimate moves towards the true value.

There is a **part two**. If x^* is the sample mean, this is an unbiased estimator of μ and the variance of x^* is σ^2/n . While the raw variance will disappear asymptotically, the scaled variance will not. The scaled statistic is $x^*n^{0.5}$. When focusing on this scaled statistic, notice that the variance is $(n^{1/2})^2 \text{var}(x^*) = \sigma^2$.

Part three.

It is possible to compare the OLS estimator with another estimator (say the Least Absolute Deviation (LAD)). If the scaled ('normalised') sampling distributions are compared to assess relative efficiency. These are $n^{0.5}(b_2 - \beta_2)$ and $N^{0.5}(b_2^* - \beta_2)$ respectively, where b_2 is the OLS estimator of β_2 and b_2^* is the LAD estimator of β_2 .

The comparison can be made with bias, variance and MSE. The estimators have different properties when the sample size is small but both are asymptotically unbiased and consistent. However, the variance of the

asymptotic distribution of the LAD estimator is smaller than the asymptotic distribution of the OLS estimator. The distribution of the estimators becomes normal after a sample size of about 5000.

ARDL models

This comes from [Dave Giles ARDL](#). Once again, thanks Dave.

The basic form of the model is

$$y_t = \beta_0 + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \alpha_0 x_t + \alpha_1 x_{t-1} + \cdots + \alpha_q x_{t-q} + \varepsilon_t \quad (1)$$

This is an ARDL(p, q) model. The model has an *autoregressive element*. As a result of the lagged values of the dependent variable, this model will yield *biased estimates* of the parameters; if the error term has *autocorrelation* the OLS estimates will be *inconsistent*.

[Frances and Oest \(2004\)](#) provide a historical overview of the *Koyck model*. There is also the *Almon distributed lag model*. [Dhrymes \(1971\)](#) provides an overview.

[Second part](#). Thanks Dave.

This is an implementation of the *Bounds Test*. This is a test to see if there is a long-run relationship. As usual, there are three types of data that may be encountered:

- stationary data that can be modelled in the levels with OLS
- non-stationary data (say $I(0)$ data) that can be modelled in first difference with OLS
- non-stationary data that are integrated and cointegrated. These data can give us a long-run relationship with OLS and a short-term correction with the *Error-correction model*

The ARDL/Bounds methodology of Persaran and Shinn (1999) and Persaran et al (2001) has a number of features that make it attractive.

- It can be used with a mixture of $I(0)$ and $I(1)$ data.
- It involves just a single equation set-up
- Different variables can be assigned different lag lengths.

The road map is as follows

1. Make sure that none of the variables are $I(2)$
2. formulate an *unrestricted* ECM
3. Determine the appropriate lag structure
4. Make sure that the errors are serial independent
5. Make sure that the model is *dynamically stable*
6. Perform a *Bounds Test* to see if there is evidence of a long-run relationship
7. If the answer to the previous question is "yes", estimate a long-run model as well as a *restricted* ECM.
8. Use these results to estimate the long run relationship and the short run relationship

Step one

Use the ADF and KPSS tests for $I(2)$

Step two

Formulate the model

$$\Delta y_t = \beta_0 + \sum \beta_i \Delta y_{t-i} + \sum \gamma_j \Delta x_{1,t-j} + \sum \delta_k \Delta x_{2,t-k} + \theta_0 y_{t-1} + \theta_1 x_{1,t-1} + \theta_2 x_{2,t-1} + \varepsilon_t \quad (2)$$

This is like an unrestricted ECM.

Step three

The appropriate lag lengths for $p1$, $q1$ and $q2$ need to be selected. Zero length lags may not be required. This is usually carried out with *Information Criteria*. Dave uses a combination of SIC and significance of coefficients.

Step four

A key assumption is that the errors must be serially independent. This requirement can also influence the selection of lag length. Use the LM test to test the null that there is serial independence against the alternative that they are AR or MA.

Step five

The model must be tested for *dynamic stability*. There is more [here](#). This essentially means that the auto-regressive coefficients must lie within the unit circle and so there is no *unit root*. The roots of the *characteristic equation* must lie outside the unit circle.

Step six

Now perform the *F-test* of the hypothesis $H0 : \theta_0 = \theta_1 = \theta_2 = 0$; against the alternative that $H0$ is not true. As in the conventional *cointegration test*, this is a test of *absence* of a long-run relationship. The distribution of the F-statistic is non-standard. However, Pesaran et al have *bounds* on the critical values for the *asymptotic* distribution of the F-test for different number of variables that range from the case of $I(0)$ to $I(1)$. If the F statistic falls below the lower bound, conclude that the variables are $I(0)$ so there is no cointegration; if the test exceeds the upper bound, there is cointegration.

As a cross-check, test $H0 : \theta_0 = 0$ against the alternative $H1 : \theta_0 < 0$

Step seven

If the bounds tests suggestst that there is cointegration, the relationship can be estimated.

$$y_t = \alpha_0 + \alpha_1 x_{1,t} + \alpha_2 x_{2,t} + \varepsilon_t \quad (3)$$

and the ECM

$$\Delta y_t = \beta_0 + \sum \beta_i \Delta y_{t-i} + \sum \gamma_j \Delta x_{1,t-j} + \sum \delta_k \Delta x_{2,t-k} + \psi z_{t-1} + \varepsilon \quad (4)$$

where $z_{t-1} = y_{t-1} - \alpha_0 - \alpha_1 x_{1,t} - \alpha_2 x_{2,t}$

Step Eight

Extract the long-run effects from the unrestricted ECM. From Equation 1, note that in the long-run $\Delta y_t = \Delta x_{1,t} = \Delta x_{2,t} = 0$ and therefore, the long-run coefficients for X_1 and X_2 are $-(\theta_1/\theta_0)$ and $-(\theta_2/\theta_0)$ respectively.

Example

Complete this after the Granger Causality.

Dynamic Stability

An AR(p) process of the form,

$$y_t = \gamma_1 y_{t-1} + \gamma_2 y_{t-2} \dots \gamma_p y_{t-p} + \varepsilon \quad (5)$$

Will be dynamically stable if the roots of the *characteristic equation*

$$1 - \gamma_1 z - \gamma_2 z^2 \dots \gamma_p z^p = 0 \quad (6)$$

lie *strictly outside* the unit circle

or, if the characteristic equation is defined as

$$z^p - \gamma_1 z^{p-1} - z^{p-2} \dots \gamma_p = 0 \quad (7)$$

lie *strictly inside* the unit circle.

Therefore, with a AR(1) model, $p = 1$ and the characteristic equation is

$$1 - \gamma_1 z = 0 \quad (8)$$

Solving for z , $z = 1/\gamma_1$, so the stationarity condition is that $|1/\gamma_1| > 1$ or $|\gamma_1| < 1$. With an AR(2),

$$1 - \gamma_1 z - \gamma_2 z^2 = 0 \quad (9)$$

lie strictly *outside* the unit circle, or

$$z^2 - \gamma_1 z - \gamma_2 = 0 \quad (10)$$

must be strictly *inside* the unit circle.

Grange Causality

Granger Causality. I need to go through this.

1 GLM

This comes from the Coursera Regression Models course. The video files are in the Teaching-Econometrics-Coursera folders. Some initial thoughts.

Think about the confidence intervals round the predicted values. This is outlined in week 2 and predicted intervals. I may need an update.

Interactions

Using the regular lineal model

$$Hu_i = b_0 + b_1Y_i + e_i \quad (11)$$

Relationship between hunger and year can be estimated for male and female. This can be two separate models. This produces two different models. Alternatively, it is possible to estimate one model with different intercepts for each.

$$Hu_i = b_0 + b_11(Sex_i = "Male") + b_2Y_i + e_i \quad (12)$$

Intercept is b_0 for females and $b_0 + b_1$ for males.

It is also possible to create a model where there are two different slopes for each model.

$$Hu_i = b_0 + b_11(Sex_i = "Male") + b_2Y_i + b_31(Sex_i = "Male") \times Y_i + e_i \quad (13)$$

Now the intercept for males is $b_0 + b_1$ while it is b_0 for females; the slope is $b_2Y_i + b_3Y_i$ for males and b_2 for females.

For the interaction term, using $*$ will automatically include the terms on their own, using $:$ will only include the interaction in the model, excluding the component parts.

This can be extended to continuous variables. For example,

$$Hu_i = b_0 + b_1Inc_i + b_2Y_i + b_3Inc_i \times Y_i + e_i \quad (14)$$

Now b_3 is the interaction or the rate of change of hunger for a change in income.

There are a number of exercises that are carried out. The code for the examples is [here](#). There are a huge variety of relationships and also the questions, once this has been done, of causation.

Residuals

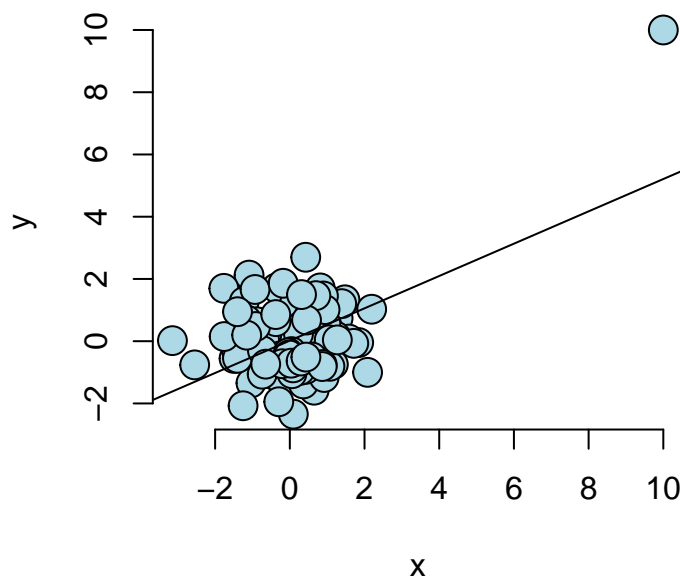
Outliers have a large potential to affect the results. There are a number of influence measures. The list includes

- `rstandard`: standardised residuals or residuals divided by the standard deviation.
- `rstudent`: *i*th residual is deleted.

- `dffits` : measures the influence of residuals on prediction.
- `cooks.distance`: measures the effect on coefficient when residual is deleted.
- `resid`: returns the ordinary residuals.
- `resid(fit)/(1 - hatvalues(fit))` where `fit`

is the linear model fit. This returns the PRESS residuals.

```
n <- 100
x <- c(10, rnorm(n))
y <- c(10, c(rnorm(n)))
plot(x, y, frame = FALSE, cex = 2, pch = 21, bg = "lightblue",
     col = "black")
abline(lm(y ~ x))
```



The estimated coefficients will change dramatically if points are taken out.

Model selection

Difference in assessing models and assessing predictability. Model is a lense through which to look at the data. Different models for prediction, for studying mechanisms and looking at causal effects.

General rules

- Omitting variables can result in bias in the coefficients unless their regressors are uncorrelated with the omitted variables. Therefore, randomising trials will attempt to ensure that there is no correlation.
- Including unnecessary variables will increase the standard error of the regression variables.

variance inflation is much worse when the extra variables that are added to the model are highly related to x_1 . There are some simulations in the slides that show this. The *variance inflation factor* will show how much the variance of estimate will increase by adding other variables. The function to use is `vif` as in `vif(fit)`; `fit <- lm(x ~ y)`. Nested models can be tested with ANOVA.

GLM

Limitations of linear models

- Do not work well for discrete or strictly-positive data
- Transformations may be difficult to interpret
- Transformations like logs are not always applicable.

Generalized Linear Models introduced in a 1972 paper by Nelder and Wedderburn. There are three components:

- An exponential family model for the response
- A systematic component via the linear predictor
- A link function that connects the means of the response to the linear predictor

The GLM model is

$$Y_i \sim N(\mu_i, \sigma^2) \quad (15)$$

The Gaussian is an exponential distribution.

- The *linear predictor* is $\eta_i = \sum_{k=1}^p X_{ik}\beta_k$.
- The *link function* is $g(\mu) = \mu$ so that, for linear models, $\eta_i = \mu_i$

This yields the *likelihood model* as an additive Gaussian linear model.

$$Y_i = \sum_{k=1}^p X_{ik}\beta_k + \varepsilon_i \quad (16)$$

Logistic model

Assume that $Y_i \sim \text{Bernoulli}(\mu_i)$ so that $E[Y_i] = \mu_i$ and where $0 \leq \mu_i \leq 1$.

The linear predictor is $\eta_i = \sum_{k=1}^p X_{ik}\beta_k$

The link function $g(\mu) = \eta = \log\left(\frac{\mu}{1-\mu}\right)$. This is the log-odds function.¹

The logit can be inverted so that $\mu_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$ and $1 - \mu_i = \frac{1}{1+\exp(\eta_i)}$.

Therefore, the likelihood is

$$\prod_{i=1}^n \mu_i^{y_i} (1 - \mu_i)^{1-y_i} = \exp\left(\sum_{i=1}^n y_i \eta_i\right) \prod_{i=1}^n (1 + \exp(\eta_i))^{-1} \quad (17)$$

Poisson regression

Assume that $Y_i \sim \text{Poisson}(\mu_i)$ so that $E[Y_i] = \mu_i$ and $\mu_i > 0$.

The linear predictor is $\eta_i = \sum_{k=1}^p X_{ik}\beta_k$

Link function $g(\mu) = \eta = \log(\mu)$. Therefore,

$$\mu_i = e^{\eta_i}$$

Therefore, the likelihood function is,

$$\prod_{i=1}^n (y_i!)^{-1} \mu_i^{y_i} e^{-\mu_i} \propto \exp\left(\sum_{i=1}^n y_i \eta_i - \sum_{i=1}^n \mu_i\right) \quad (18)$$

These notes are copied directly from the slides.

In each case, the only way in which the likelihood depends on the data is through

$$\sum_{i=1}^n y_i \eta_i = \sum_{i=1}^n y_i \sum_{k=1}^p X_{ik}\beta_k = \sum_{k=1}^p \beta_k \sum_{i=1}^n X_{ik} y_i$$

¹ $\frac{1}{1-\mu}$. The probability divided by one minus the probability is the odds. Therefore the logit is the log of the odds.

Thus if we don't need the full data, only $\sum_{i=1}^n X_{ik}y_i$. This simplification is a consequence of choosing so-called 'canonical' link functions. All models achieve their maximum at the root of the so called normal equations

$$0 = \sum_{i=1}^n \frac{(Y_i - \mu_i)}{Var(Y_i)} W_i$$

where W_i are the derivative of the inverse of the link function.

Notes about variances. For Bernoulli and Poisson cases, there are direct relationships between the mean and the variance that can be relaxed a little by introducing the variable ϕ .

$$0 = \sum_{i=1}^n \frac{(Y_i - \mu_i)}{Var(Y_i)} W_i$$

* For the linear model $Var(Y_i) = \sigma^2$ is constant. * For Bernoulli case $Var(Y_i) = \mu_i(1 - \mu_i)$ * For the Poisson case $Var(Y_i) = \mu_i$. * In the latter cases, it is often relevant to have a more flexible variance model, even if it doesn't correspond to an actual likelihood

$$0 = \sum_{i=1}^n \frac{(Y_i - \mu_i)}{\phi \mu_i (1 - \mu_i)} W_i \quad \text{and} \quad 0 = \sum_{i=1}^n \frac{(Y_i - \mu_i)}{\phi \mu_i} W_i$$

These are called 'quasi-likelihood' normal equations when using the ϕ function. In R it is possible to use the quasi-poisson or quasi-binomial families to get this flexibility.

The equations have to be solved iteratively. The predicted linear predictor responses can be obtained as $\hat{\eta} = \sum_{k=1}^p X_b \beta_k$ and the predicted mean response as $\hat{\mu} = g^{-1}(\hat{\eta})$.

Coefficients are interpreted as

$$g(E[Y|X_k = x_k + 1, X_{\sim k} = x_{\sim k}]) - g(E[Y|X_k = x_k, X_{\sim k} = x_{\sim k}]) = \beta_k$$

This is the change in the link function of the expected response for unit of change in X_k holding other regressors constant. Uses variation on the Newon/Raphson algorithm. Asypotics are used for inference.

GLM binary data

Two possible outcomes mean that there is binary or Bernoulli data.

```
load("../Data/ravensData.rda")
head(ravensData)
```

```
##      ravenWinNum ravenWin ravenScore opponentScore
## 1             1         W          24             9
## 2             1         W          38            35
## 3             1         W          28            13
## 4             1         W          34            31
## 5             1         W          44            13
## 6             0         L          23            24
```

The results can be built up gradually.

- The binary outcome is win or lose

$$RW_i$$

- Probability of winning

$$\Pr(RW_i | RS_i, b_0, b_1)$$

- Odds of winning

$$\frac{\Pr(RW_i | RS_i, b_0, b_1)}{1 - \Pr(RW_i | RS_i, b_0, b_1)}$$

- Log odds

$$\log \left(\frac{\Pr(RW_i | RS_i, b_0, b_1)}{1 - \Pr(RW_i | RS_i, b_0, b_1)} \right)$$

Interpreting the results. From

$$\log \left(\frac{\Pr(RW_i | RS_i, b_0, b_1)}{1 - \Pr(RW_i | RS_i, b_0, b_1)} \right)$$

b_0 is the log odds of winning if there are no point scored. b_1 is the log odds of a win for each point scored (leaving all other variables constant). $\exp(b_1)$ is the odds ratio of a win for each point scored (holding all else constant).

Odds

- Imagine that you are playing a game where you flip a coin with success probability p . - If it comes up heads, you win X . If it comes up tails, you lose Y . - What should we set X and Y for the game to be fair?

$$E[\text{earnings}] = Xp - Y(1 - p) = 0$$

- Implies

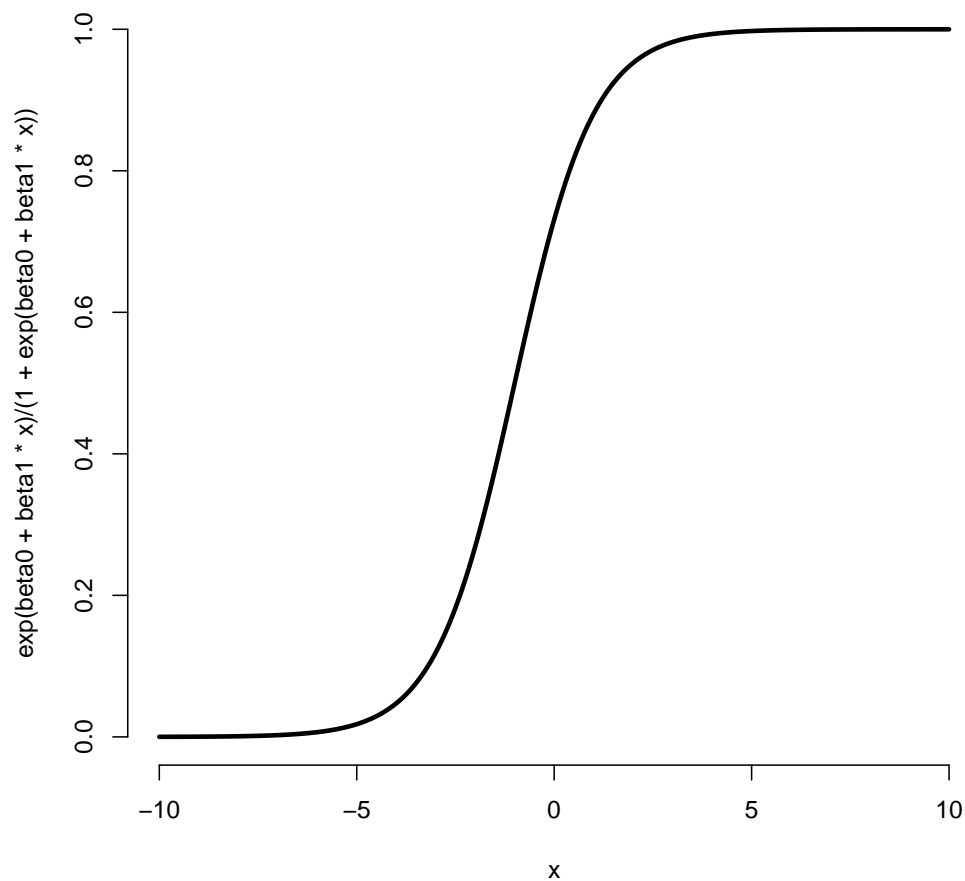
$$\frac{Y}{X} = \frac{p}{1 - p}$$

- The odds can be said as "How much should you be willing to pay for a p probability of winning a dollar?" - (If $p > 0.5$ you have to pay more if you lose than you get if you win.) - (If $p < 0.5$ you have to pay less if you lose than you get if you win.)

For the logistic regression,

$$p_x = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

```
x <- seq(-10, 10, length = 1000)
beta1 <- 1
beta0 <- 1
plot(x, exp(beta0 + beta1 * x) / (1 + exp(beta0 + beta1 * x)),
     type = "l", lwd = 3, frame = FALSE)
```



As the β_1 shifts value, the curve will shift from being downward sloping (negative) to being upwards sloping (positive). There is a straight line at zero. β_0 will shift the whole line left or right.

It is determining whether the data are zero or one.

```
logRegRavens <- glm(ravensData$ravenWinNum ~ ravensData$ravenScore, family="binomial")
summary(logRegRavens)

##
## Call:
## glm(formula = ravensData$ravenWinNum ~ ravensData$ravenScore,
##      family = "binomial")
##
## Deviance Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -1.758 -1.100   0.530   0.806   1.495
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.6800     1.5541  -1.08    0.28
## ravensData$ravenScore  0.1066     0.0667   1.60    0.11
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 24.435  on 19  degrees of freedom
## Residual deviance: 20.895  on 18  degrees of freedom
## AIC: 24.89
##
## Number of Fisher Scoring iterations: 5
```

Given the following results from the logistic regression with Raven data intercept coefficient -1.6 and slope 0.1006, this is saying that there is a log odds of -1.6 that they win when they have no points. $e^{-1.6}$ would give the straight odds. $e^{0.1066}$ gives the increase in the odds for each point scored.

For small numbers, $e^x \approx 1 + x$. The probability of winning is $\frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$

Confidence intervals and Anova can be carried out. This will test whether adding variables reduces the variance by a significant amount.

```
exp(logRegRavens$coeff)

##              (Intercept) ravensData$ravenScore
##              0.1864              1.1125

exp(confint(logRegRavens))

## Waiting for profiling to be done...

##              2.5 % 97.5 %
## (Intercept)    0.005675  3.106
## ravensData$ravenScore 0.996230  1.303
```

```
anova(logRegRavens, test="Chisq")

## Analysis of Deviance Table
```

```
##
## Model: binomial, link: logit
##
## Response: ravensData$ravenWinNum
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL              19      24.4
## ravensData$ravenScore 1      3.54      18      20.9      0.06 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residual deviance falls from 24.4 to 20.9 as the Raven score is added. This is 3.54 increase, something that the Chi-squared distribution says would be seen only 6% of the time if it were actually having no effect.

1.0.1 Poisson Regression

These are data that take the form of count. It could be the number of calls received by a call centre or rate, such as the percentage of children passing a test. The rate is a count per unit of time. In these cases, linear regression with a transformation is an option. The *Poisson* distribution can also be used to model approximately binominal data with a small p and a large n or to model contingency tables.

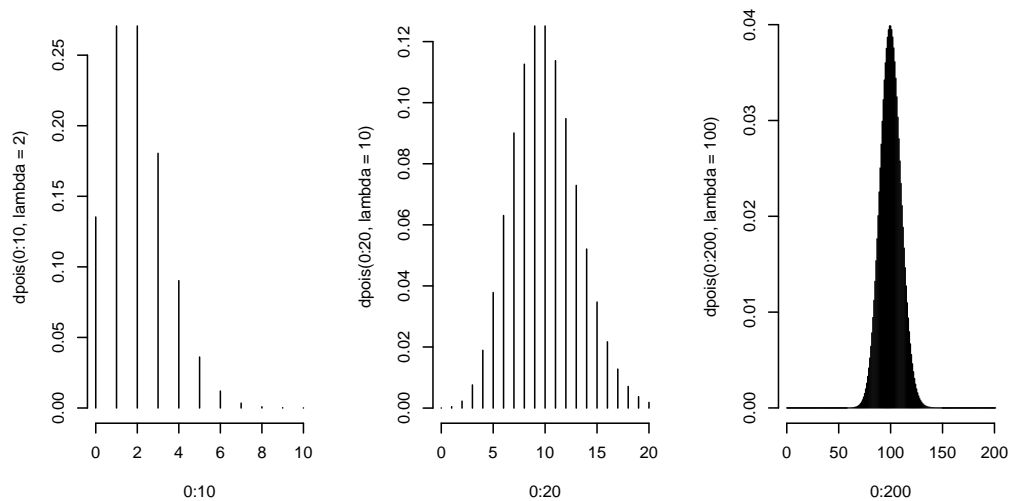
$$X \sim \text{Poisson}(t\lambda) \quad (19)$$

if

$$P(X = x) = \frac{t\lambda^x e^{-t\lambda}}{x!} \quad (20)$$

For $x = 1, 0 \dots$ The mean of the Poisson is $E[X] = t\lambda$. Therefore, the $E[X/t] = \lambda$. Therefore, λ is the expected counts per unit of time. The variance of the Poisson is $t\lambda$. The Poisson tends to normal as the $t\lambda$ gets large.

```
par(mfrow = c(1, 3))
plot(0 : 10, dpois(0 : 10, lambda = 2), type = "h", frame = FALSE)
plot(0 : 20, dpois(0 : 20, lambda = 10), type = "h", frame = FALSE)
plot(0 : 200, dpois(0 : 200, lambda = 100), type = "h", frame = FALSE)
```

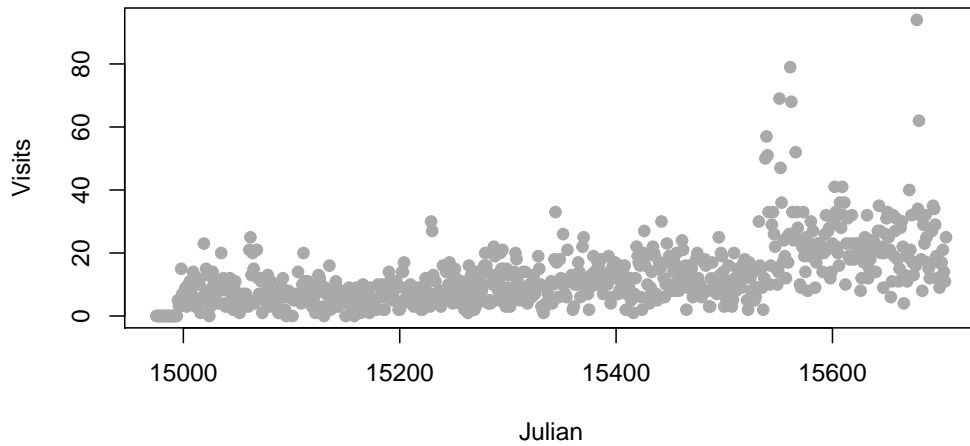



Example,
This example will work on hits to a web site per day.

```
load("../Data/gaData.rda")
head(gaData)

##           date visits simplystats
## 1 2011-01-01      0            0
## 2 2011-01-02      0            0
## 3 2011-01-03      0            0
## 4 2011-01-04      0            0
## 5 2011-01-05      0            0
## 6 2011-01-06      0            0

gaData$julian <- julian(gaData$date)
plot(gaData$julian, gaData$visits, pch=19,
     col="darkgrey", xlab="Julian", ylab="Visits")
```



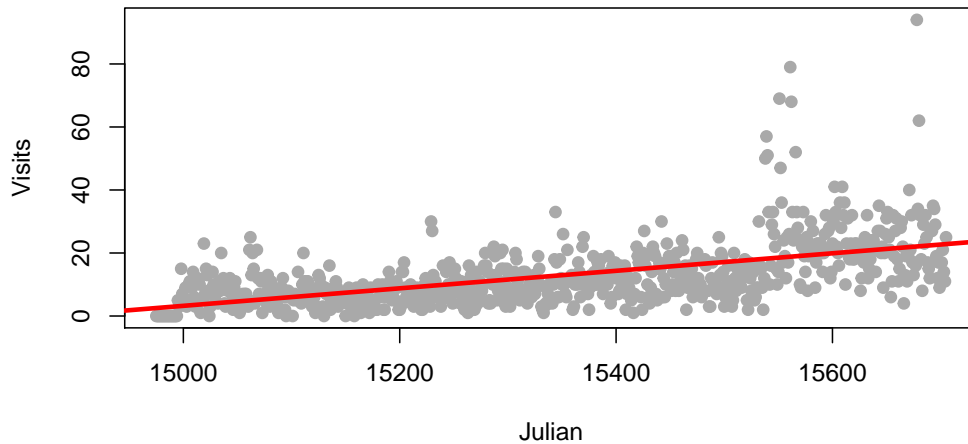
It is possible to model this as a linear regression. For example,

$$NH_i = b_0 + b_1 JD_i + e_i \quad (21)$$

where,

- NH_i is the number of hits to the web site
- JD_i is the day of the year (Julien)
- b_0 is the number of hits on Julien day zero
- b_1 is the increase in the number of hits per day.
- e_I is the variation due to everything that is not measured.

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
lm1 <- lm(gaData$visits ~ gaData$julian)
abline(lm1,col="red",lwd=3)
```



This whole exercise can be completed by logging the hits that are achieved (leaving aside for now that the log of the zero initial days is not possible). However, this is a very special transformation that is being carried out. If that is the case, $e^{E[\log(Y)]}$ is the geometric mean of Y . With no covariates, this is estimated as $e^{\frac{1}{n} \sum_{t=1}^n \log(y_t)}$. This is the same as $(\prod_{t=1}^n y_i)^{1/n}$. The exponential coefficients estimate things about geometric means.

- e^{β_0} is the geometric mean of the hits on day zero.
- e^{β_1} is the geometric mean of the increase in hits per day.

There's a problem with logs with you have zero counts, adding a constant works, though this changes the interpretation as it is not the increase in the hits per day plus one.

```
round(exp(coef(lm(I(log(gaData$visits + 1)) ~ gaData$julian))), 5)

##      (Intercept) gaData$julian
##           0.000           1.002
```

This is not telling us that there is a 2% increase in web hits per day.

The difference between Linear and Poisson regression is mainly in the interpretation of the coefficients.

Linear vs. Poisson regression

The linear regression

$$NH_i = b_0 + b_1 JD_i + e_i$$

implied an expected value relationship.

$$E[NH_i|JD_i, b_0, b_1] = b_0 + b_1 JD_i$$

The *Poisson/log-linear* model will model the log of the expected value.

$$\log(E[NH_i|JD_i, b_0, b_1]) = b_0 + b_1 JD_i$$

Otherwise, this can be converted so that you get the expected value of the

$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0 + b_1 JD_i)$$

From this equation, b_1 may be interpreted as the relative increase in the hits per day holding everything else constant.

$$e^{(E[NH_i|JD_i=J_1]) - (E[NH_i|JD_i=j])}$$

This is different from the log of the count data because that would be $E[\log(NH_i)]$

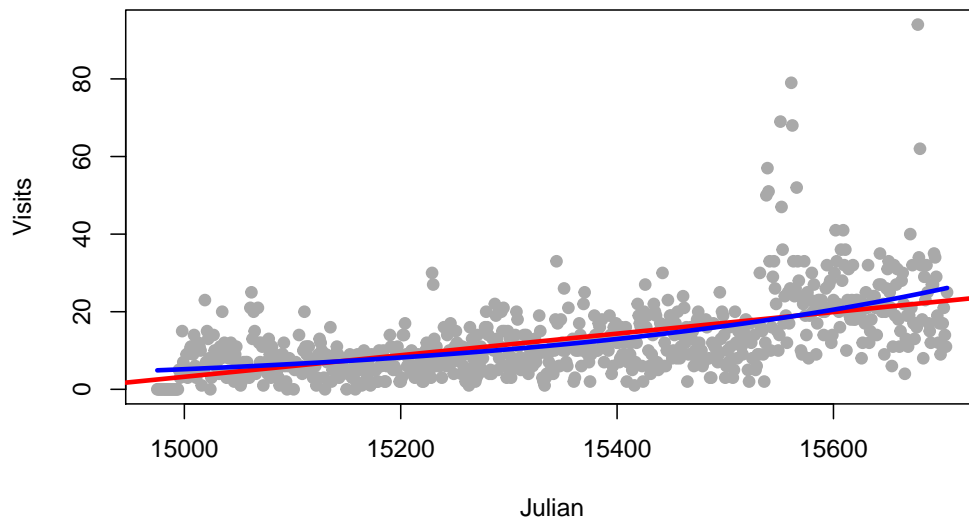
Multiplicative differences

$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0 + b_1 JD_i)$$

$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0) \exp(b_1 JD_i)$$

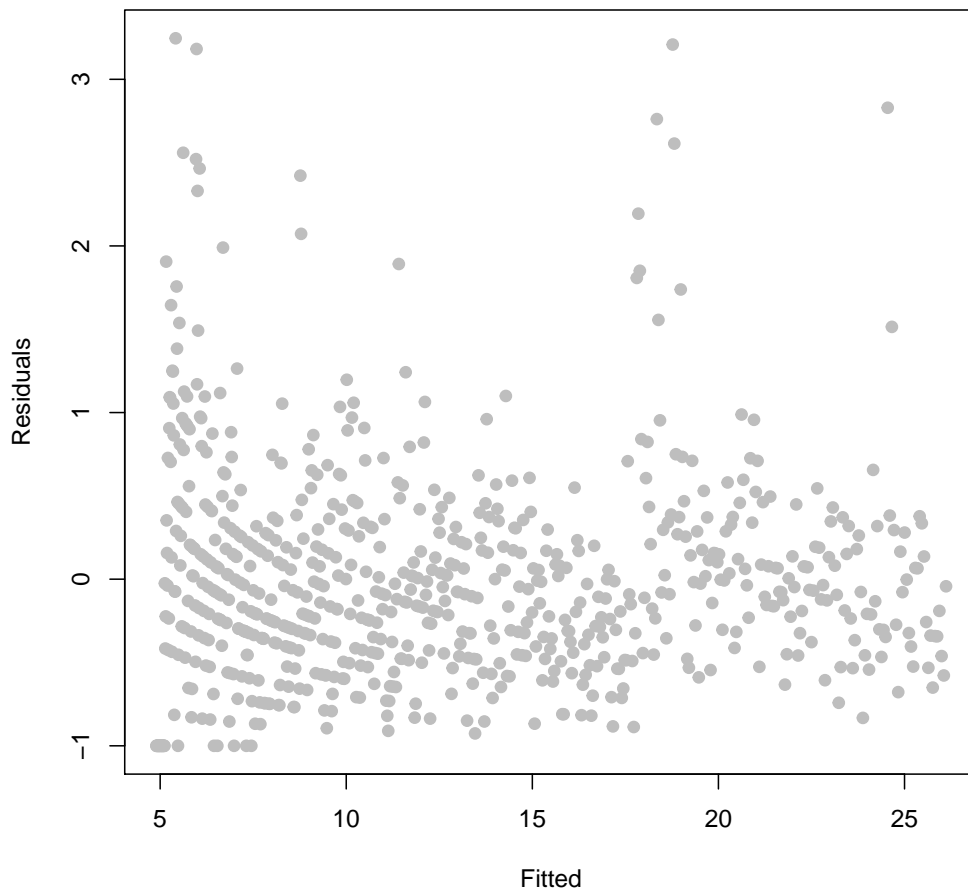
If JD_i is increased by one unit, $E[NH_i|JD_i, b_0, b_1]$ is multiplied by $\exp(b_1)$

```
plot(gaData$julian, gaData$visits, pch=19, col="darkgrey", xlab="Julian", ylab="Visits")
glm1 <- glm(gaData$visits ~ gaData$julian, family="poisson")
abline(lm1, col="red", lwd=3); lines(gaData$julian, glm1$fitted, col="blue", lwd=3)
```



Notice that the GLM model has a small curve.
It appears that there are some issues with the residuals.

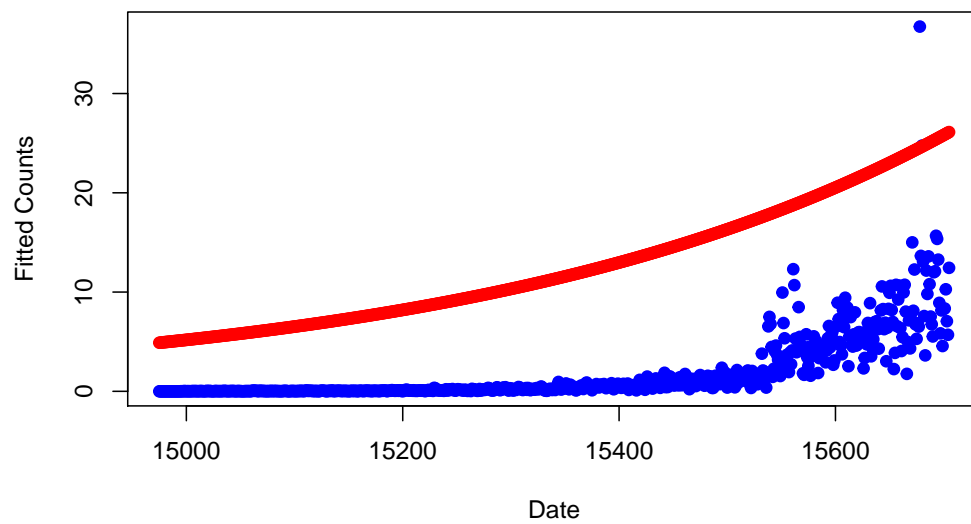
```
plot(glm1$fitted,glm1$residuals,pch=19,col="grey",ylab="Residuals",xlab="Fitted"
```



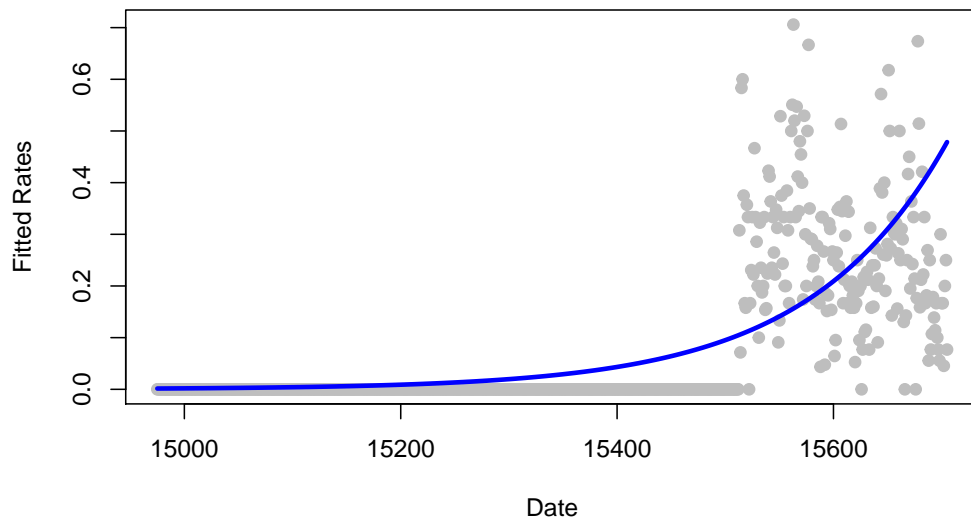
One way to deal with that would be to use the *quasi-poisson* family argument. An alternative would be to try to get *robust standard errors*.

Fitting rates in R Rate models. To create a model of the rate, the `offset` argument can be used. This will define the variable that is to form the base of the rate. It must be logged. See the slides for more details.

```
glm2 <- glm(gaData$simplystats ~ julian(gaData$date), offset=log(visits+1),
            family="poisson", data=gaData)
plot(julian(gaData$date), glm2$fitted, col="blue", pch=19, xlab="Date", ylab="Fitted")
points(julian(gaData$date), glm1$fitted, col="red", pch=19)
```



```
## Fitting rates in R
lm2 <- glm(gaData$simplystats ~ julian(gaData$date), offset=log(visits+1),
           family="poisson", data=gaData)
plot(julian(gaData$date), gaData$simplystats/(gaData$visits+1), col="grey", xlab="Date",
     ylab="Fitted Rates", pch=19)
lines(julian(gaData$date), glm2$fitted/(gaData$visits+1), col="blue", lwd=3)
```



ZIP model when there is inflation of the zero value that must be addressed.