

## Lecture 3

### Residual Analysis + Generalized Linear Models

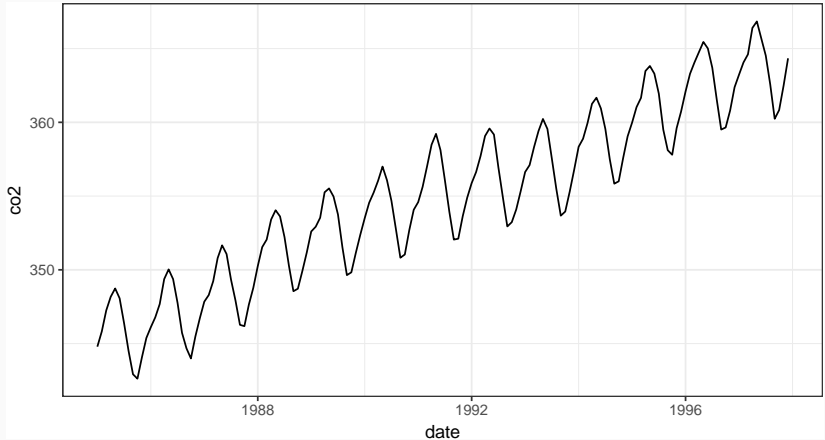
---

Colin Rundel

1/23/2018

## Residual Analysis

# Atmospheric CO<sub>2</sub> (ppm) from Mauna Loa

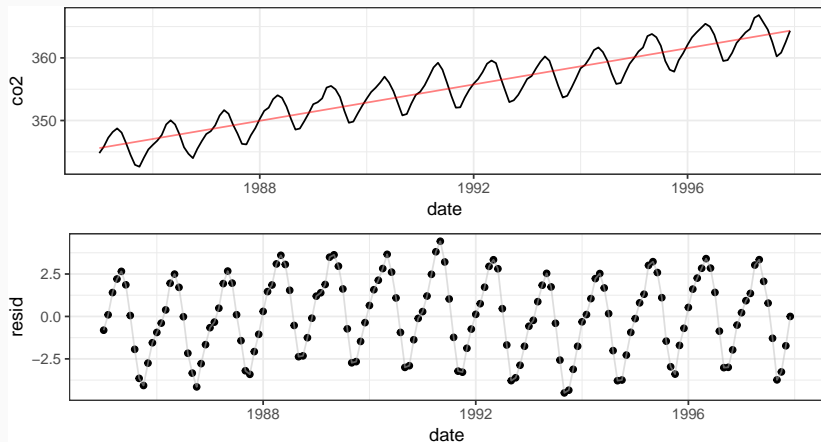


## Where to start?

Well, it looks like stuff is going up on average ...

## Where to start?

Well, it looks like stuff is going up on average ...

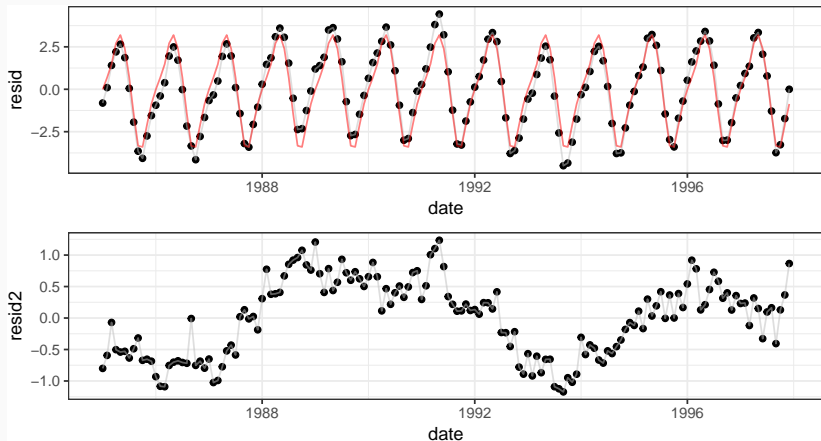


and then?

Well there is some periodicity lets add the month ...

and then?

Well there is some periodicity lets add the month ...



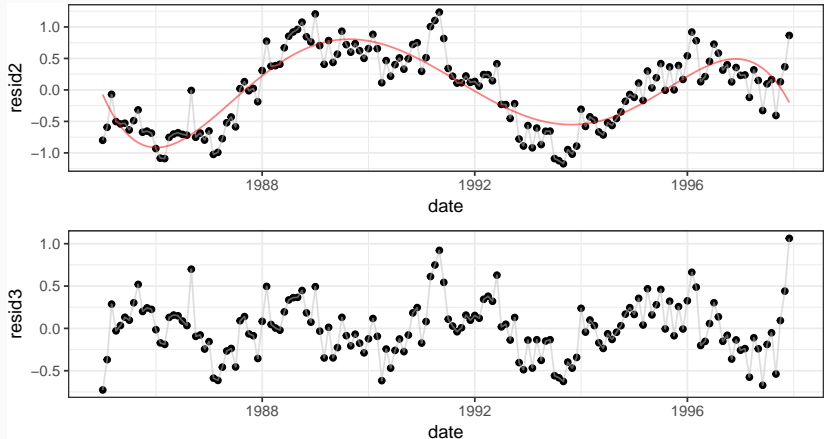
## and then and then?

There is still some long term trend in the data, maybe a fancy polynomial can help ...



## and then and then?

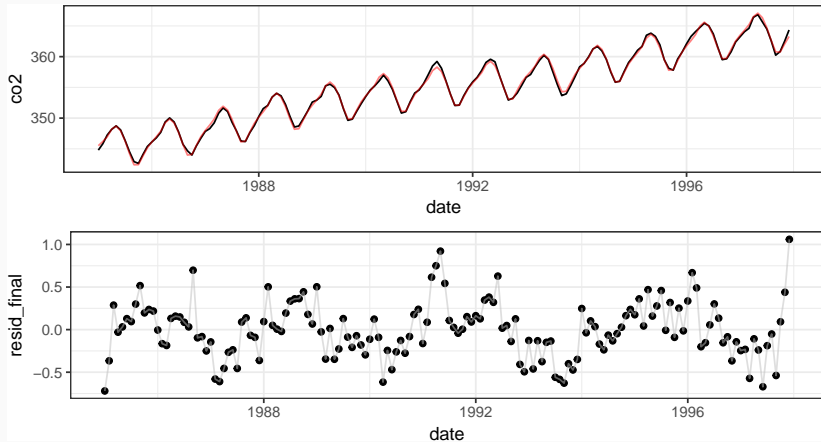
There is still some long term trend in the data, maybe a fancy polynomial can help ...



## Putting it all together ...

```
l_final = lm(co2-date + month + poly(date,5), data=co2_df)
summary(l_final)
##
## Call:
## lm(formula = co2 ~ date + month + poly(date, 5), data = co2_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72022 -0.19169 -0.00638  0.17565  1.06026
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.587e+03  1.460e+01 -177.174 < 2e-16 ***
## date         1.479e+00  7.334e-03  201.649 < 2e-16 ***
## monthAug    -4.155e+00  1.346e-01 -30.880 < 2e-16 ***
## monthDec    -3.566e+00  1.350e-01 -26.404 < 2e-16 ***
## monthFeb    -2.022e+00  1.345e-01 -15.041 < 2e-16 ***
## monthJan    -2.729e+00  1.345e-01 -20.286 < 2e-16 ***
## monthJul    -2.018e+00  1.345e-01 -15.003 < 2e-16 ***
## monthJun    -3.136e-01  1.345e-01  -2.332 0.021117 *
## monthMar    -1.233e+00  1.344e-01  -9.175 5.54e-16 ***
## monthMay     4.881e-01  1.344e-01   3.631 0.000396 ***
## monthNov    -4.799e+00  1.349e-01 -35.577 < 2e-16 ***
## monthOct    -6.102e+00  1.348e-01 -45.282 < 2e-16 ***
## monthSep    -6.036e+00  1.346e-01 -44.832 < 2e-16 ***
## poly(date, 5)1      NA           NA           NA           NA
## poly(date, 5)2    -1.920e+00  3.427e-01  -5.602 1.09e-07 ***
## poly(date, 5)3    3.920e+00  3.451e-01  11.358 < 2e-16 ***
## poly(date, 5)4    8.946e-01  3.428e-01   2.609 0.010062 *
## poly(date, 5)5   -4.340e+00  3.462e-01 -12.535 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3427 on 139 degrees of freedom
## Multiple R-squared:  0.997, Adjusted R-squared:  0.9966
## F-statistic: 2872 on 16 and 139 DF, p-value: < 2.2e-16
```

# Final fit + Residualss



# Generalized Linear Models

A generalized linear model has three key components:

1. a probability distribution (from the exponential family) that describes your response variable
2. a linear predictor  $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ ,
3. and a link function  $g$  such that  $g(E(\mathbf{Y}|\mathbf{X})) = \boldsymbol{\eta}$ .

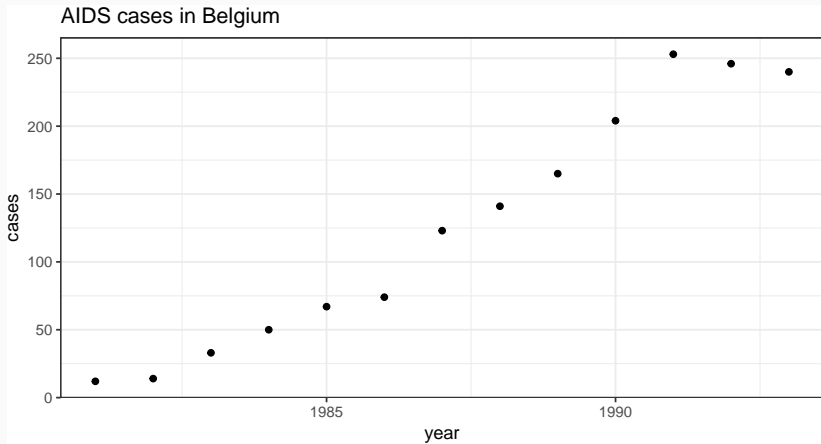
## Poisson Regression

This is a special case of a generalized linear model for count data where we assume the outcome variable follows a poisson distribution (mean = variance).

$$Y_i \sim \text{Poisson}(\lambda_i)$$
$$\log E(Y_i | \mathbf{X}_i) = \log \lambda_i = \mathbf{X}_i \cdot \boldsymbol{\beta}$$

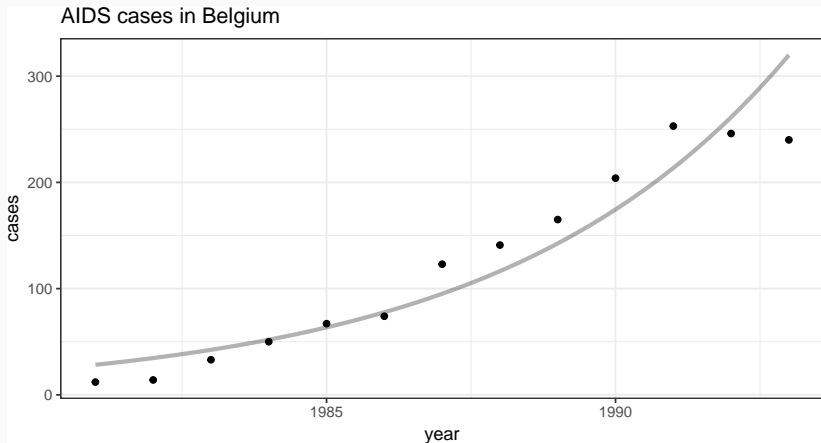
## Example - AIDS in Belgium

These data represent the total number of new AIDS cases reported in Belgium during the early stages of the epidemic.



## Frequentist glm fit

```
g = glm(cases~year, data=aids, family=poisson)
pred = data_frame(year=seq(1981,1993,by=0.1)) %>%
  mutate(cases = predict(g, newdata=., type = "response"))
```





## Residuals?

The naive approach is to use standard residuals,

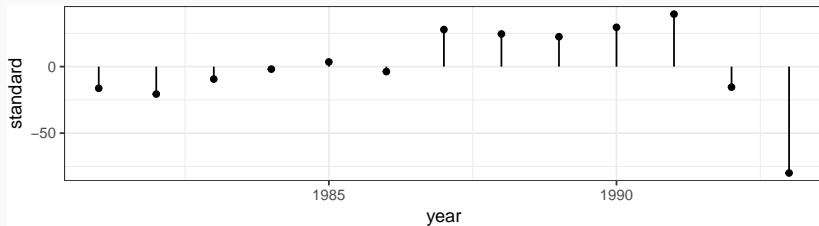
$$r_i = Y_i - E(Y_i|X) = Y_i - \hat{\lambda}_i$$

# Residuals?

The naive approach is to use standard residuals,

$$r_i = Y_i - E(Y_i|X) = Y_i - \hat{\lambda}_i$$

```
aids = aids %>%  
  mutate(pred = predict(g, newdata=., type = "response")) %>%  
  mutate(standard = cases - pred)  
  
ggplot(aids, aes(x=year, y=standard)) +  
  geom_point() + geom_segment(aes(xend=year, yend=0))
```



Pearson residuals:

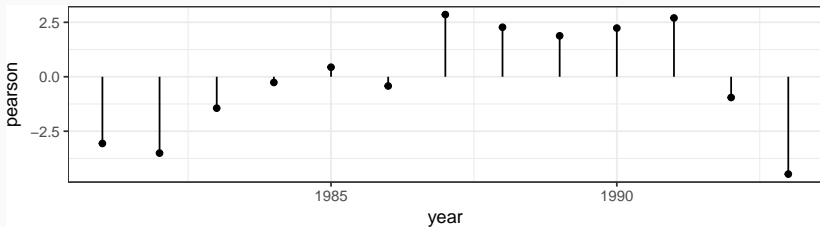
$$r_i = \frac{Y_i - E(Y_i|X)}{\sqrt{\text{Var}(Y_i|X)}} = \frac{Y_i - \hat{\lambda}_i}{\sqrt{\hat{\lambda}_i}}$$

## Accounting for variability

Pearson residuals:

$$r_i = \frac{Y_i - E(Y_i|X)}{\sqrt{\text{Var}(Y_i|X)}} = \frac{Y_i - \hat{\lambda}_i}{\sqrt{\hat{\lambda}_i}}$$

```
aids = aids %>%  
  mutate(pearson = (cases - pred)/sqrt(pred))  
  
ggplot(aids, aes(x=year, y=pearson)) +  
  geom_point() + geom_segment(aes(xend=year, yend=0))
```



Deviance is a way of measuring the difference between your glm's fit and the fit of a perfect model (where  $E(\hat{Y}_i|X) = Y_i$ ).

It is defined as twice the log of the ratio between the likelihood of a perfect model and the likelihood of the given model,

$$\begin{aligned} D &= 2 \log(\mathcal{L}(\theta_{best}|Y) / \mathcal{L}(\hat{\theta}|Y)) \\ &= 2(l(\theta_{best}|Y) - l(\hat{\theta}|Y)) \end{aligned}$$





```
summary(g)
##
## Call:
## glm(formula = cases ~ year, family = poisson, data = aids)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6784  -1.5013  -0.2636   2.1760   2.7306
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.971e+02  1.546e+01  -25.68  <2e-16 ***
## year         2.021e-01  7.771e-03   26.01  <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: 872.206  on 12  degrees of freedom
## Residual deviance:  80.686  on 11  degrees of freedom
## AIC: 166.37
##
## Number of Fisher Scoring iterations: 4
```



We can therefore think of deviance as  $D = \sum_{i=1}^n d_i^2$  where  $d_i$  is a generalized residual. So in the Poisson case we can define,

$$d_i = \text{sign}(y_i - \lambda_i) \sqrt{2(y_i \log(y_i/\hat{\lambda}_i) - (y_i - \hat{\lambda}_i))}$$

## Deviance residuals

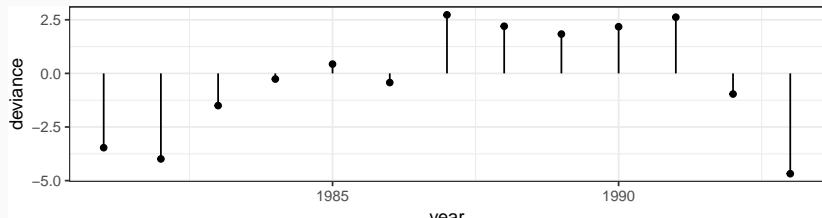
We can therefore think of deviance as  $D = \sum_{i=1}^n d_i^2$  where  $d_i$  is a generalized residual. So in the Poisson case we can define,

$$d_i = \text{sign}(y_i - \lambda_i) \sqrt{2(y_i \log(y_i/\hat{\lambda}_i) - (y_i - \hat{\lambda}_i))}$$

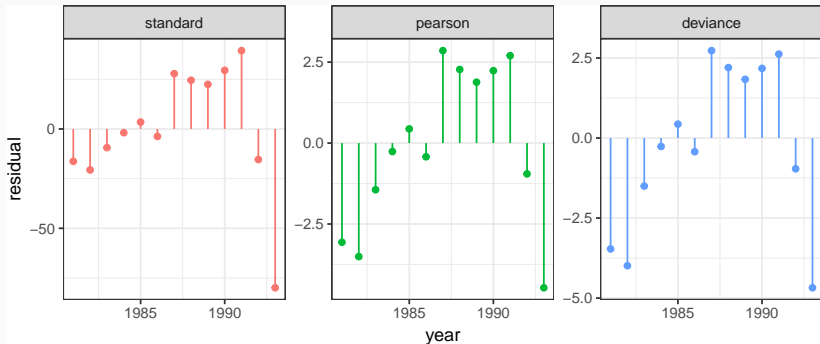
```
dev_resid = function(obs,pred)
  sign(obs-pred) * sqrt(2*(obs*log(obs/pred)-(obs-pred)))

aids = aids %>%
  mutate(deviance = dev_resid(cases, pred))

ggplot(aids, aes(x=year, y=deviance)) +
  geom_point() + geom_segment(aes(xend=year, yend=0))
```



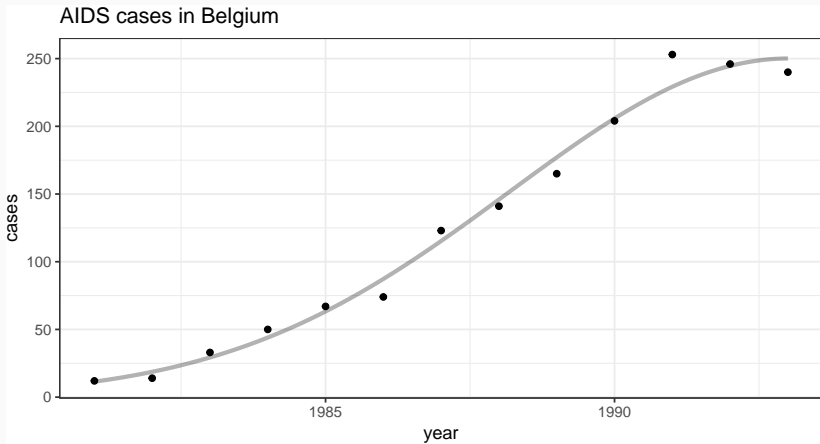
# Comparing Residuals



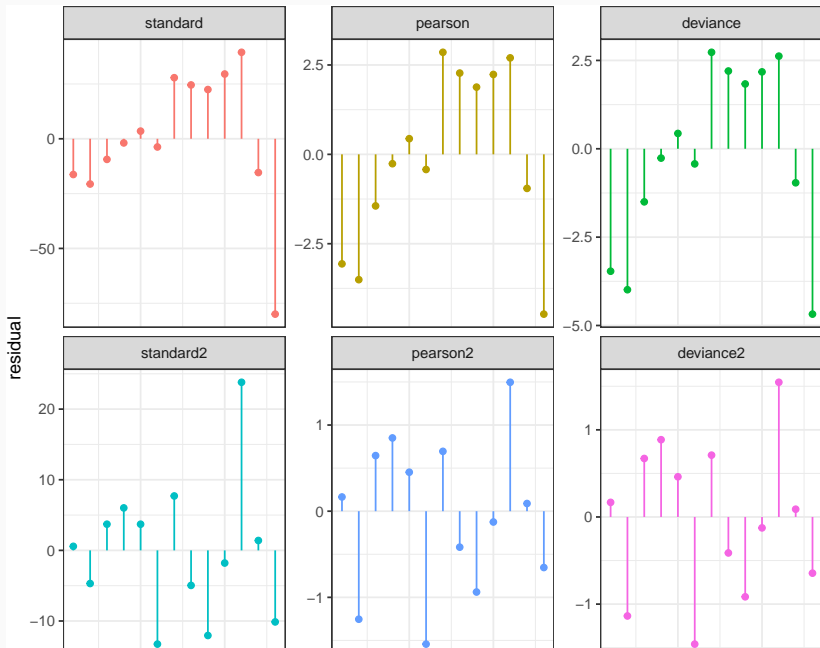
## Updating the model

# Quadratic fit

```
g2 = glm(cases~year+I(year^2), data=aids, family=poisson)
pred2 = data_frame(year=seq(1981,1993,by=0.1)) %>%
  mutate(cases = predict(g2, newdata=., type = "response"))
```



# Quadratic fit - residuals



# Bayesian Model

## Bayesian Poisson Regression Model

```
poisson_model =  
"model{  
  # Likelihood  
  for (i in 1:length(Y)) {  
    Y[i] ~ dpois(lambda[i])  
    log(lambda[i]) <- beta[1] + beta[2]*X[i]  
  
    # In-sample prediction  
    Y_hat[i] ~ dpois(lambda[i])  
  }  
  
  # Prior for beta  
  for(j in 1:2){  
    beta[j] ~ dnorm(0,1/100)  
  }  
}"
```



```
n_burn=1000; n_iter=5000

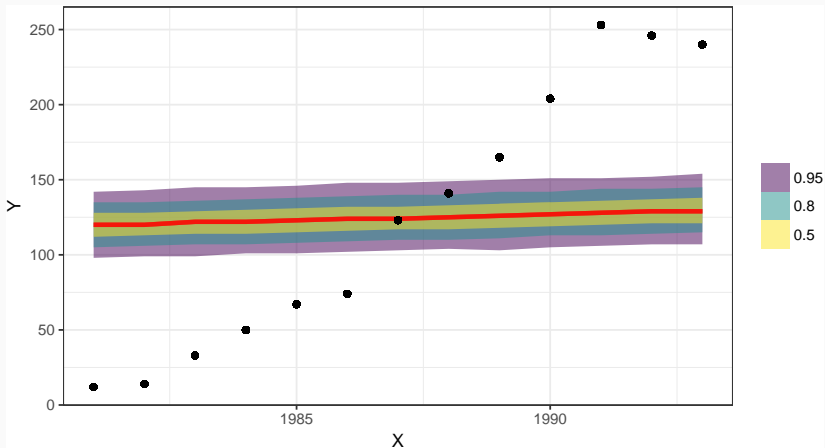
m = rjags::jags.model(
  textConnection(poisson_model), quiet = TRUE,
  data = list(Y=aids$cases, X=aids$year)
)

update(m, n.iter=1000, progress.bar="none")

samp = rjags::coda.samples(
  m, variable.names=c("beta", "lambda", "Y_hat", "Y", "X"),
  n.iter=5000, progress.bar="none"
)
```

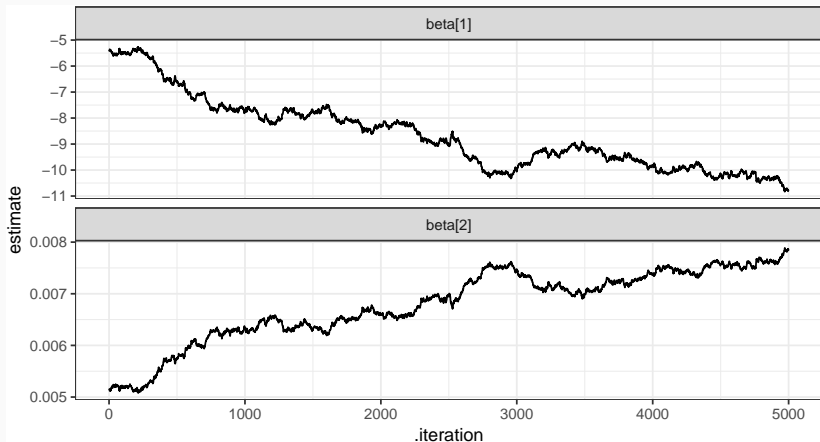
## Model Fit?

```
tidybayes::spread_samples(samp, Y_hat[i], X[i],Y[i]) %>%  
  ungroup() %>%  
  ggplot(aes(x=X,y=Y)) +  
    tidybayes::stat_lineribbon(aes(y=Y_hat), alpha=0.5) +  
    geom_point()
```



# MCMC Diagnostics

```
tidybayes::gather_samples(samp, beta[i]) %>%  
  mutate(param = paste0(term, "[", i, "]")) %>%  
  ggplot(aes(x=.iteration, y=estimate)) +  
    geom_line() +  
    facet_wrap(~param, ncol=1, scale="free_y")
```

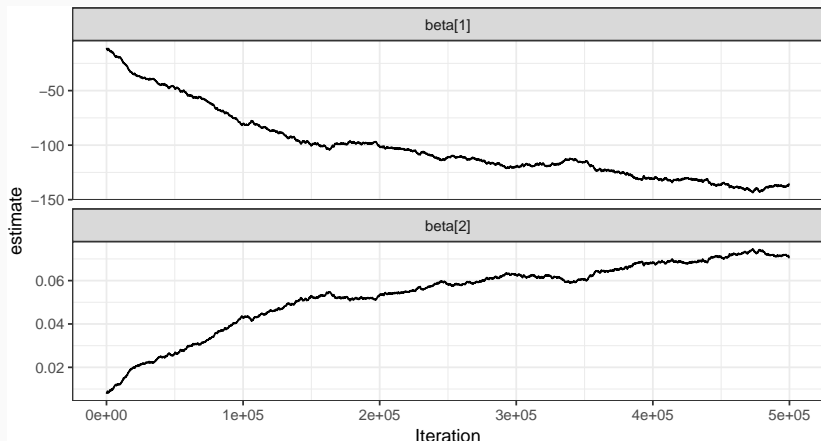


## Now what?

Maybe more iterations will fix everything ...

Now what?

Maybe more iterations will fix everything ...



What went wrong?

## What went wrong?

```
summary(g)
##
## Call:
## glm(formula = cases ~ year, family = poisson, data = aids)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6784  -1.5013  -0.2636   2.1760   2.7306
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.971e+02  1.546e+01  -25.68  <2e-16 ***
## year         2.021e-01  7.771e-03   26.01  <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: 872.206  on 12  degrees of freedom
## Residual deviance:  80.686  on 11  degrees of freedom
## AIC: 166.37
##
## Number of Fisher Scoring iterations: 4
```

## A simple fix

```
summary(glm(cases~I(year-1981), data=aids, family=poisson))
##
## Call:
## glm(formula = cases ~ I(year - 1981), family = poisson, data = aids)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6784  -1.5013  -0.2636   2.1760   2.7306
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.342711   0.070920  47.13  <2e-16 ***
## I(year - 1981) 0.202121   0.007771  26.01  <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: 872.206  on 12  degrees of freedom
## Residual deviance:  80.686  on 11  degrees of freedom
## AIC: 166.37
##
## Number of Fisher Scoring iterations: 4
```

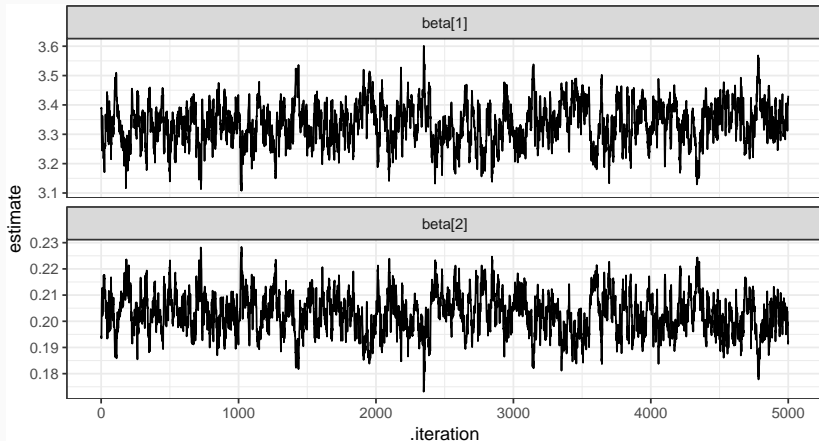


## Revising the jags model

```
poisson_model2 =  
"model{  
  # Likelihood  
  for (i in 1:length(Y)) {  
    Y[i] ~ dpois(lambda[i])  
    log(lambda[i]) <- beta[1] + beta[2]*(X[i] - 1981)  
  
    Y_hat[i] ~ dpois(lambda[i])  
  }  
  
  # Prior for beta  
  for (j in 1:2) {  
    beta[j] ~ dnorm(0,1/100)  
  }  
}"
```

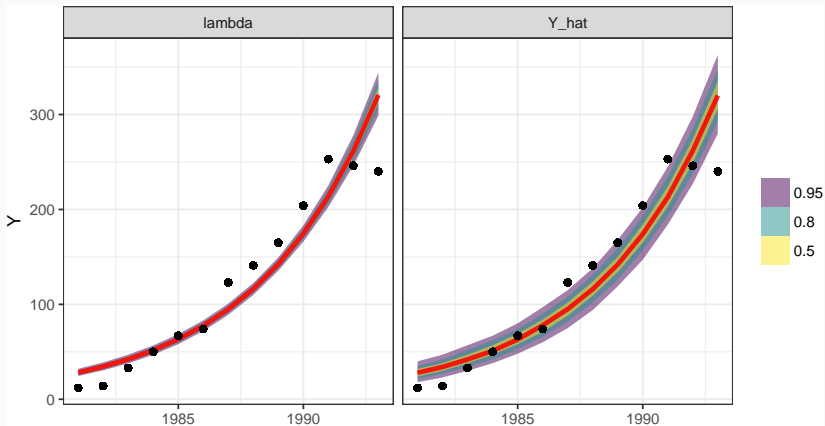
# MCMC Diagnostics

```
tidybayes::gather_samples(samp2, beta[i]) %>%  
  mutate(param = paste0(term, "[", i, "]")) %>%  
  ggplot(aes(x=.iteration, y=estimate)) +  
    geom_line() +  
    facet_wrap(~param, ncol=1, scale="free_y")
```

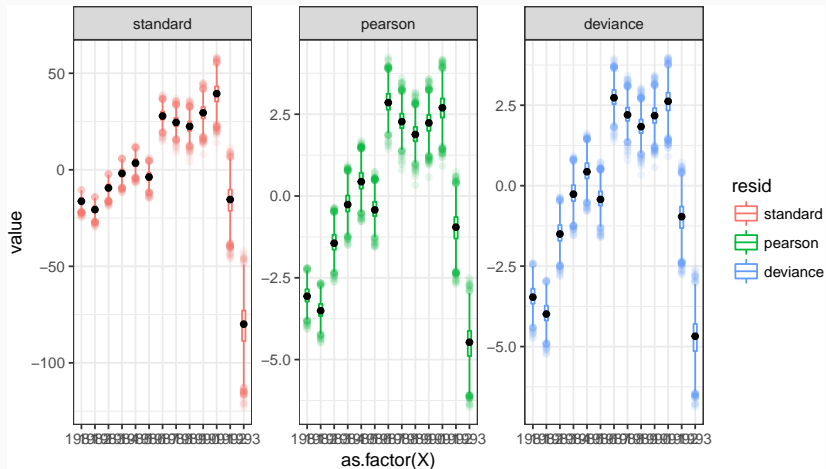


# Model Fit

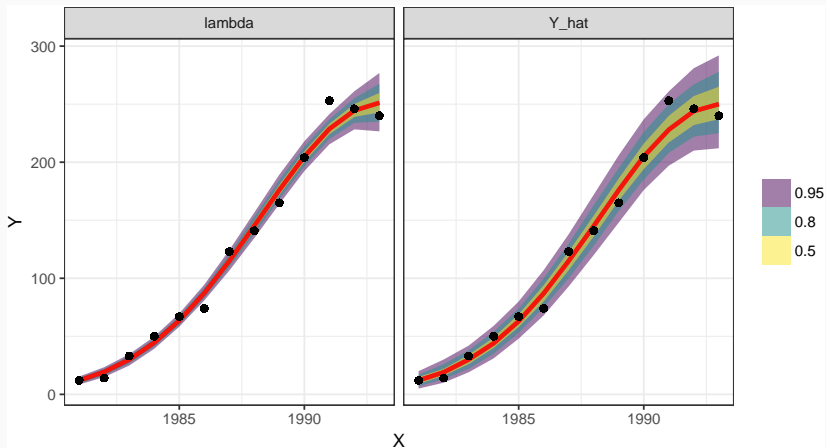
```
tidybayes::spread_samples(samp2, Y_hat[i], lambda[i], X[i], Y[i]) %>%  
  ungroup() %>%  
  tidyr::gather(param, value, Y_hat, lambda) %>%  
  ggplot(aes(x=X, y=Y)) +  
    tidybayes::stat_lineribbon(aes(y=value), alpha=0.5) +  
    geom_point() +  
    facet_wrap(~param)
```



# Residual Plots

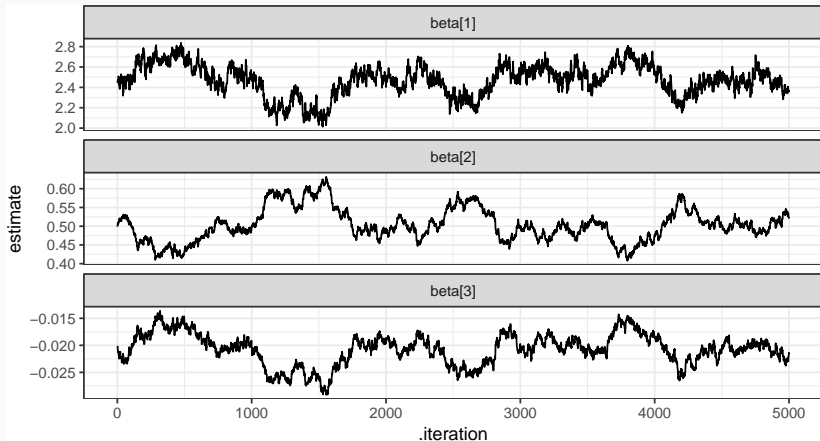


# Quadratic Fit



# MCMC Diagnostics

```
tidybayes::gather_samples(samp3, beta[i]) %>%  
  mutate(param = paste0(term,"[",i,"]")) %>%  
  ggplot(aes(x=.iteration, y=estimate)) +  
    geom_line() +  
    facet_wrap(~param, ncol=1, scale="free_y")
```



# Residual Plots

