

# Residual Analysis + Generalized Linear Models

Lecture 04

Dr. Colin Rundel

# dukestm package

This is a companion package for the course where I will be putting useful functions for some of our common tasks in the course.

This is not an official or polished package but I've tried to include documentation for most functions. If you notice a problem open an issue or send me an email.

To install,

```
1 devtools::install_github("sta344-644-fa23/dukestm")
```

Updates will be made throughout the semester and I will attempt to remind you when something new is available, rerunning the above will get you the latest version.

# Example - epred\_draws\_fix()

```
1 tidybayes::epred_draws(b, newdata=d)
```

```
# A tibble: 400,000 × 7
```

```
# Groups:   x, y, .row [100]
```

	x	y	.row	.chain	.iteration	.draw
	<int>	<dbl>	<int>	<int>	<int>	<int>
1	1	-3.24	1	NA	NA	1
2	1	-3.24	1	NA	NA	2
3	1	-3.24	1	NA	NA	3
4	1	-3.24	1	NA	NA	4
5	1	-3.24	1	NA	NA	5
6	1	-3.24	1	NA	NA	6
7	1	-3.24	1	NA	NA	7
8	1	-3.24	1	NA	NA	8
9	1	-3.24	1	NA	NA	9
10	1	-3.24	1	NA	NA	10

```
# i 399,990 more rows
```

```
# i 1 more variable: .epred <dbl>
```

```
1 dukestm::epred_draws_fix(b, newdata=d)
```

```
# A tibble: 400,000 × 7
```

	x	y	.row	.chain	.iteration	.draw
	<int>	<dbl>	<int>	<int>	<int>	<int>
1	1	-3.24	1	1	1	1
2	1	-3.24	1	1	2	2
3	1	-3.24	1	1	3	3
4	1	-3.24	1	1	4	4
5	1	-3.24	1	1	5	5
6	1	-3.24	1	1	6	6
7	1	-3.24	1	1	7	7
8	1	-3.24	1	1	8	8
9	1	-3.24	1	1	9	9
10	1	-3.24	1	1	10	10

```
# i 399,990 more rows
```

```
# i 1 more variable: .epred <dbl>
```

# Where we left it - Empirical Coverage ( $\hat{y}$ )

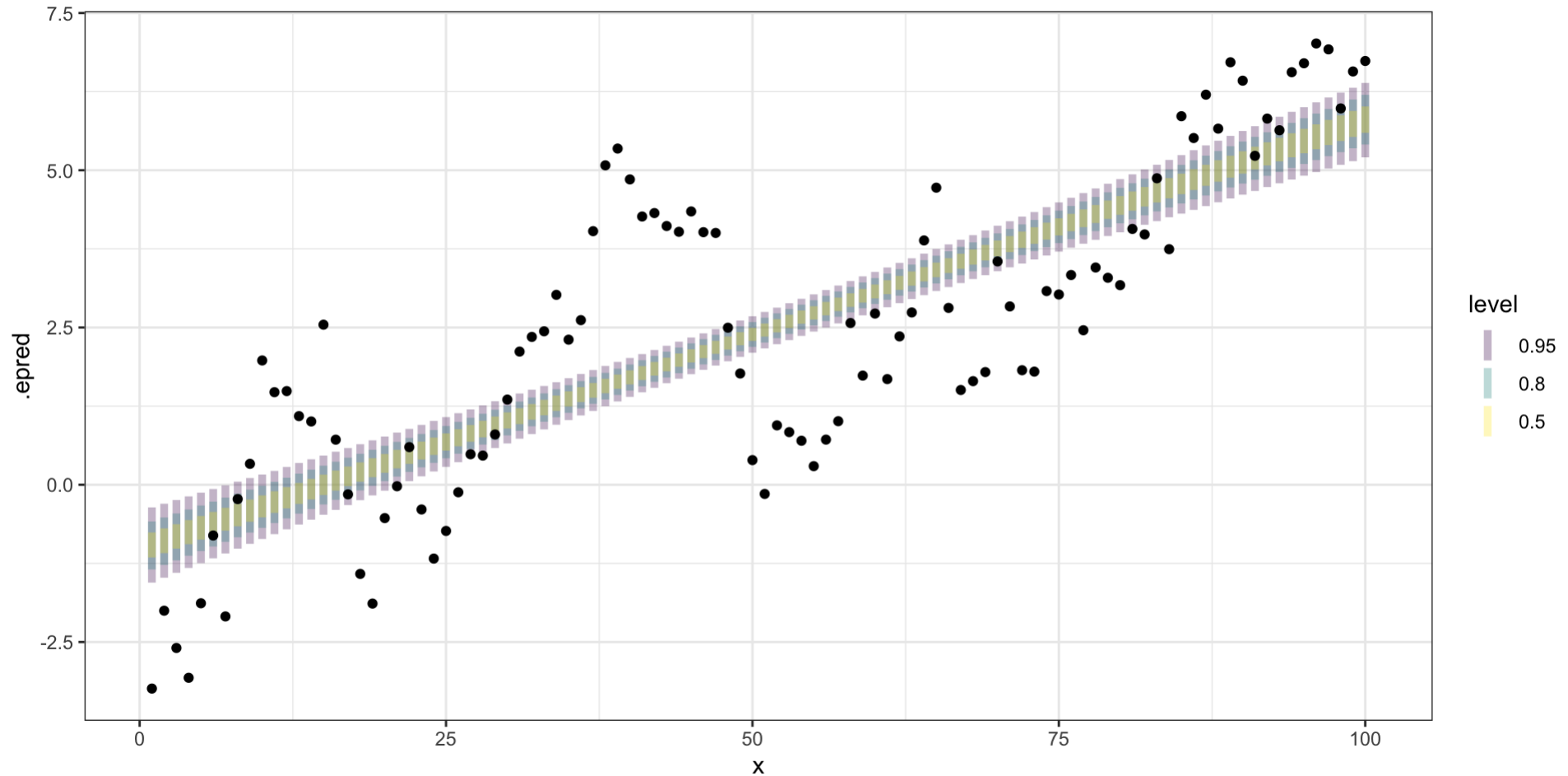
```
1 ( dukestm::epred_draws_fix(b, newdata=d) |>
2   group_by(x, y) |>
3   tidybayes::mean_hdi(
4     .epred, .width = c(0.5, 0.9, 0.95)
5   ) |>
6   mutate(contains = y >= .lower & y <= .upper) |>
7   group_by(prob = .width) |>
8   summarize(
9     emp_cov = sum(contains)/n()
10  )
11 )
```

# A tibble: 3 × 2

	prob	emp_cov
	<dbl>	<dbl>
1	0.5	0.02
2	0.9	0.11
3	0.95	0.14

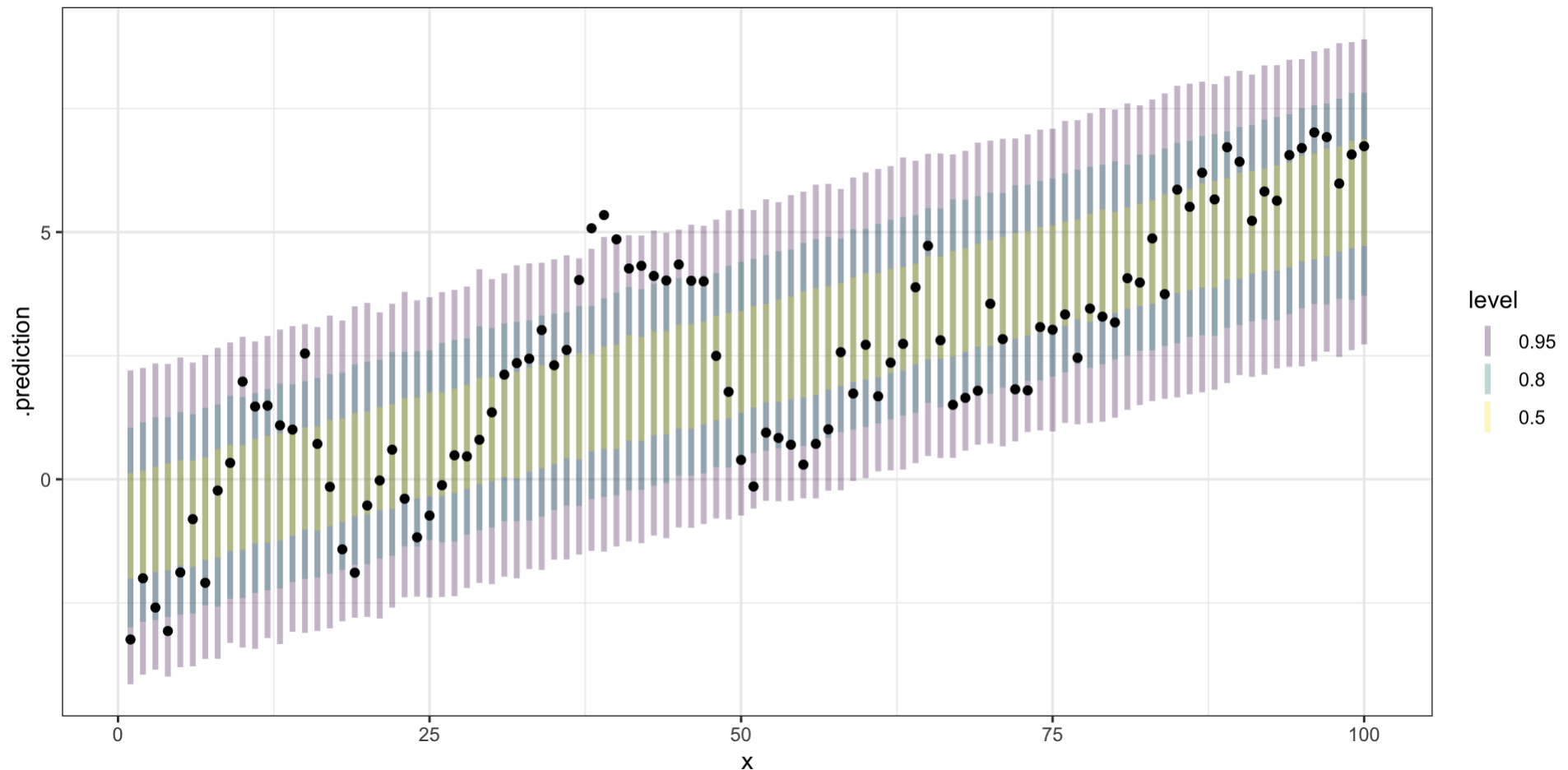
# What went wrong?

```
1 epred_draws_fix(b, newdata=d) |>  
2   ggplot(aes(x=x)) +  
3     ggdist::stat_interval(alpha=0.3, aes(y=.epred, group=x), linewidth=1.66) +  
4     geom_point(data=d, aes(y=y))
```



# The right predictions

```
1 predicted_draws_fix(b, newdata=d) |>  
2   ggplot(aes(x=x)) +  
3     ggdist::stat_interval(alpha=0.3, aes(y=.prediction, group=x), linewidth=1.25) +  
4     geom_point(data=d, aes(y=y))
```



# Empirical Coverage (y)

```
1 predicted_draws_fix(b, newdata=d) |>
2   group_by(x, y) |>
3   tidybayes::mean_hdi(
4     .prediction, .width = c(0.5, 0.8, 0.9, 0.95)
5   ) |>
6   mutate(contains = y >= .lower & y <= .upper) |>
7   group_by(prob = .width) |>
8   summarize(
9     emp_cov = sum(contains)/n()
10  )
```

# A tibble: 4 × 2

	prob	emp_cov
	<dbl>	<dbl>
1	0.5	0.42
2	0.8	0.81
3	0.9	0.95
4	0.95	0.97

# RMSE $-\hat{y}$ vs $y$



# CRPS

$\hat{y}$

```
1 epred_draws_fix(b, newdata=d) |>
2   group_by(.chain, x) |>
3     summarise(
4       crps = dukestm::calc_crps(.epred, obs=y)
5     ) |>
6   summarize(
7     mean(crps)
8   )
```

# A tibble: 4 × 2

	<code>.chain`mean(crps)`</code>
	<code>&lt;int&gt;</code> <code>&lt;dbl&gt;</code>
1	1      1.19
2	2      1.19
3	3      1.19
4	4      1.19

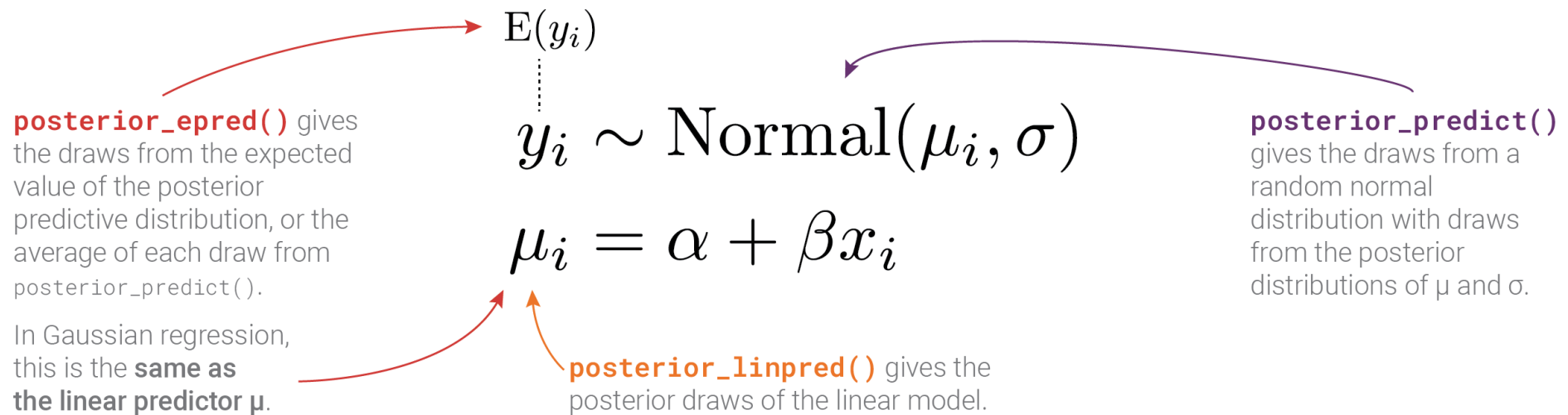
y

```
1 predicted_draws_fix(b, newdata=d) |>
2   group_by(.chain, x) |>
3     summarise(
4       crps = dukestm::calc_crps(.prediction,
5     ) |>
6   summarize(
7     mean(crps)
8   )
```

# A tibble: 4 × 2

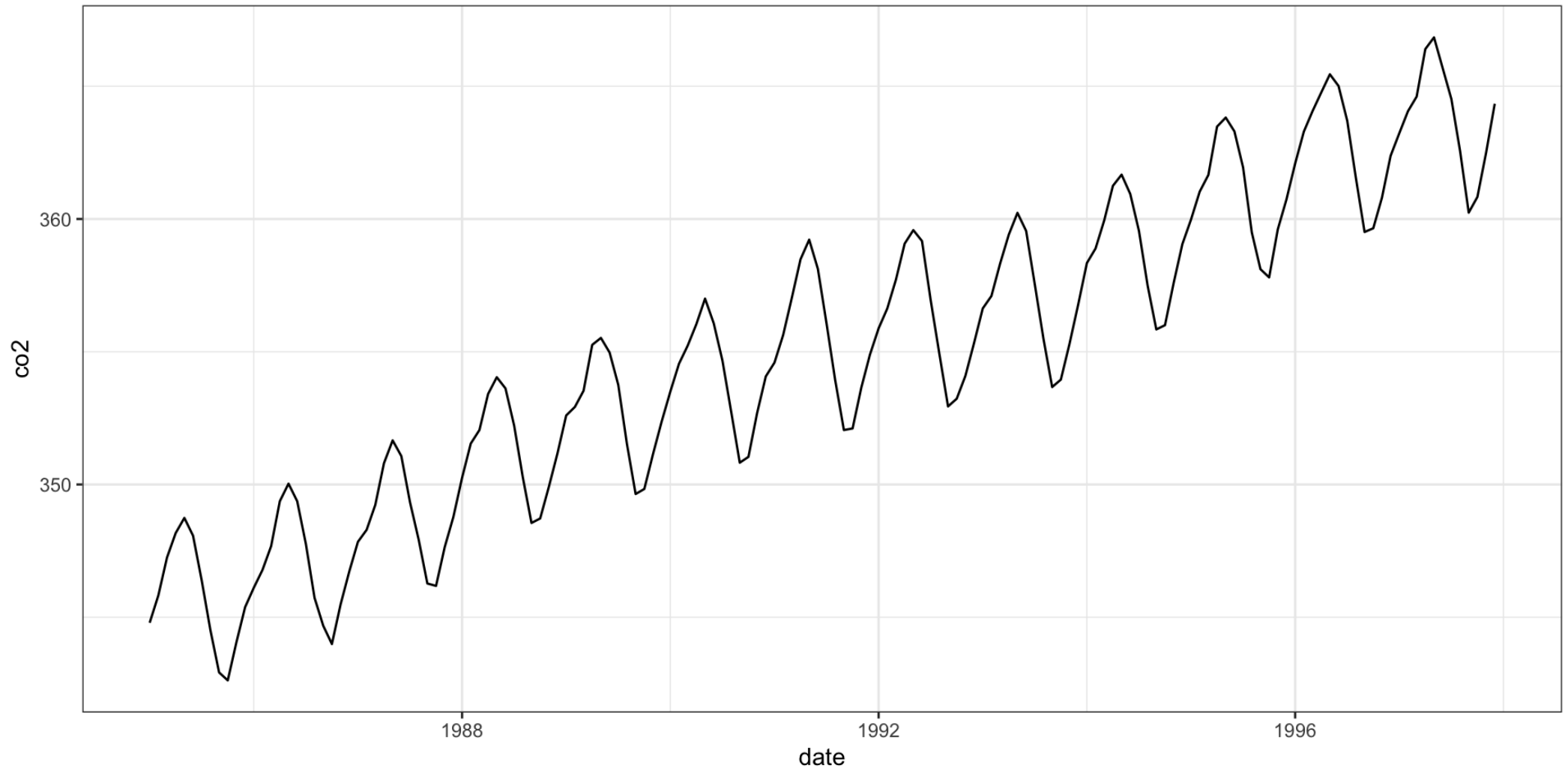
	<code>.chain`mean(crps)`</code>
	<code>&lt;int&gt;</code> <code>&lt;dbl&gt;</code>
1	1      0.885
2	2      0.878
3	3      0.882
4	4      0.884

# Posterior sampling functions



# Residual Analysis

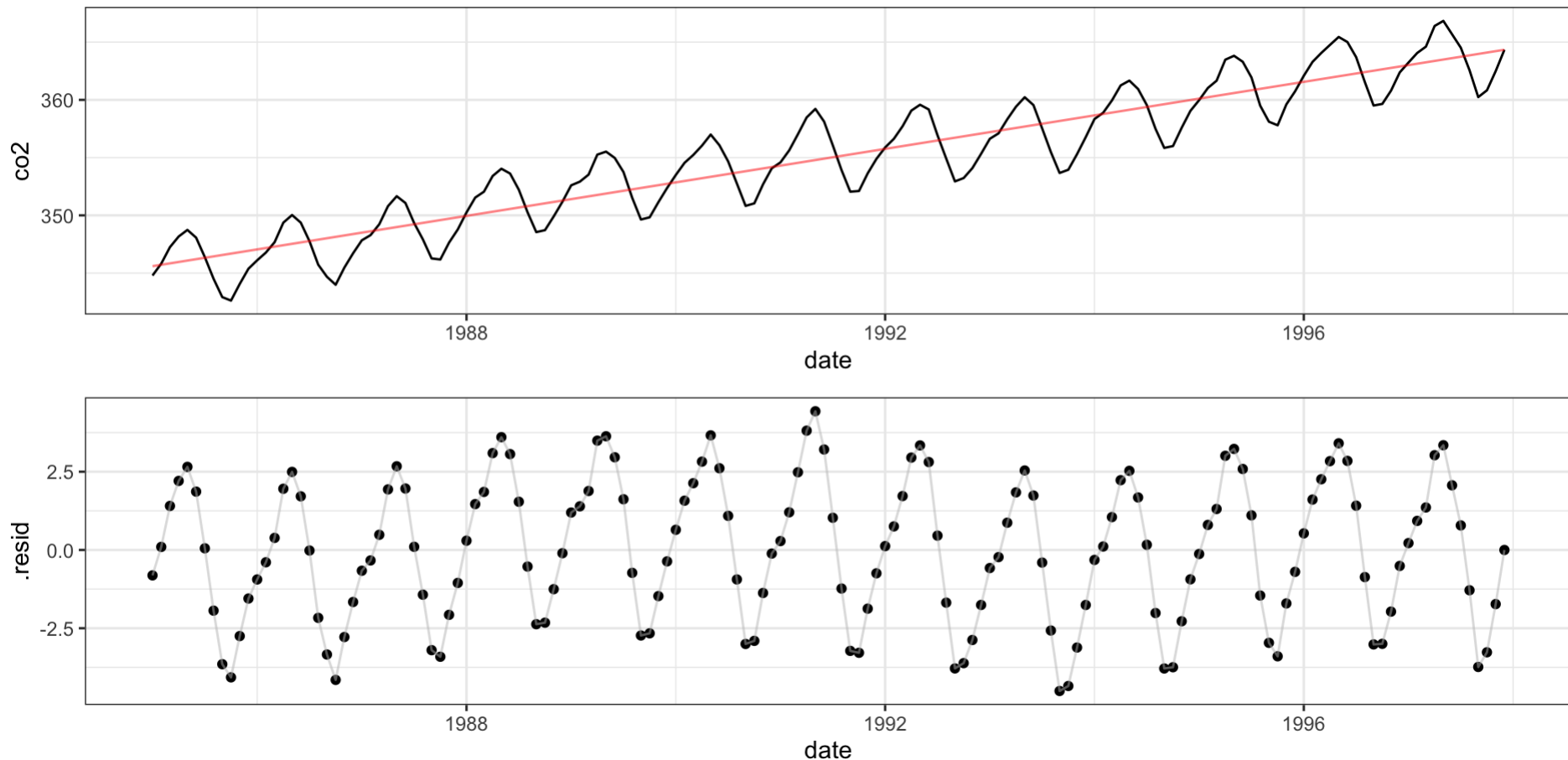
# Atmospheric CO<sub>2</sub> (ppm) from Mauna Loa



# Where to start?

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

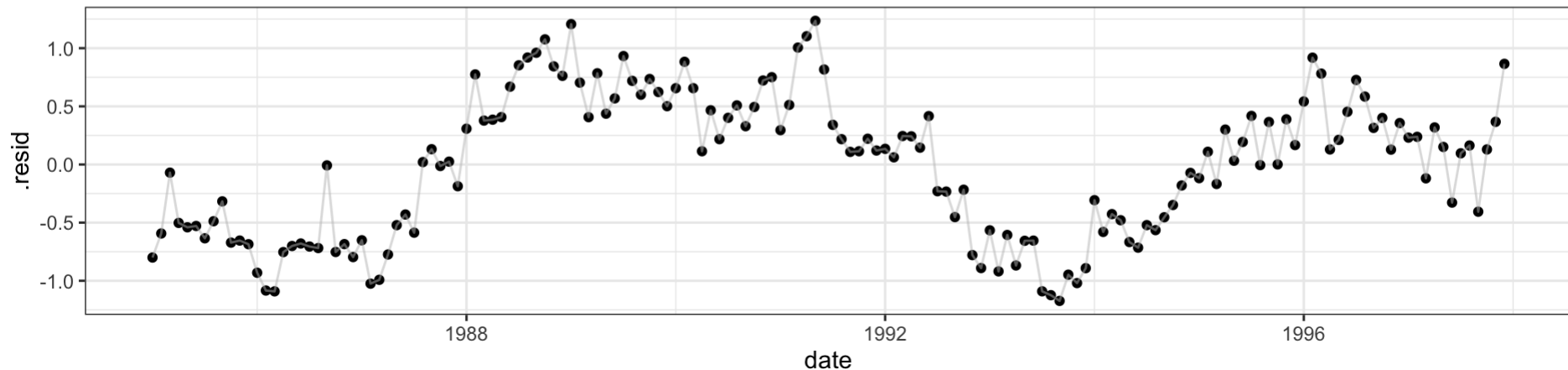
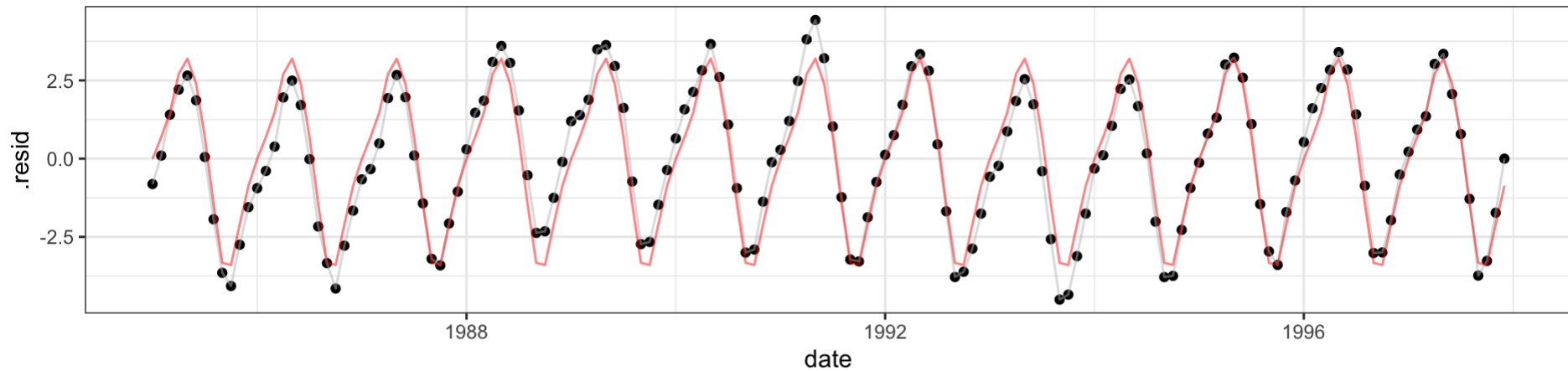
```
1 1 = lm(co2~date, data=mauna_loa)
```



# and then?

Well there is some periodicity lets add the month (as a factor) ...

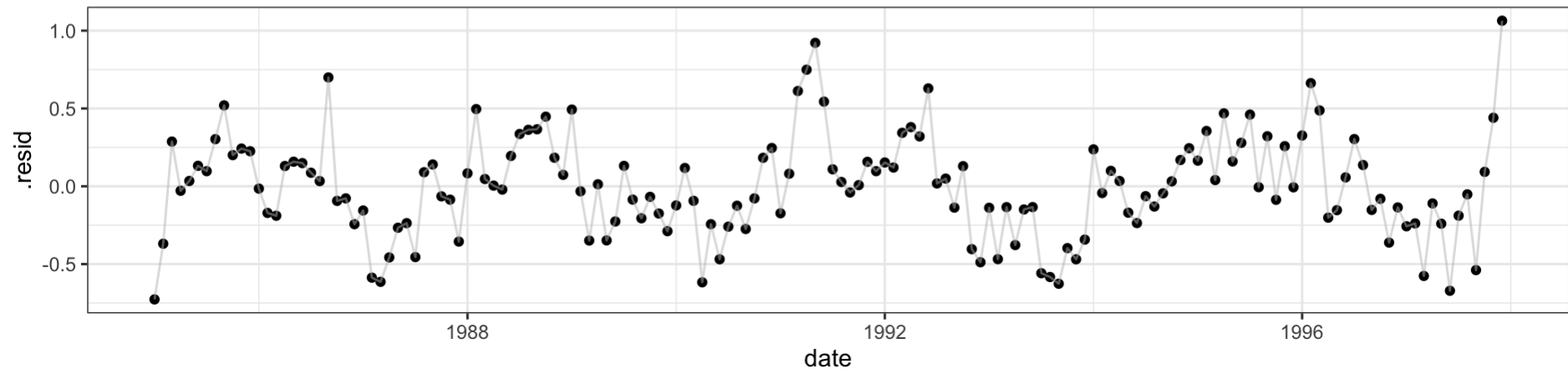
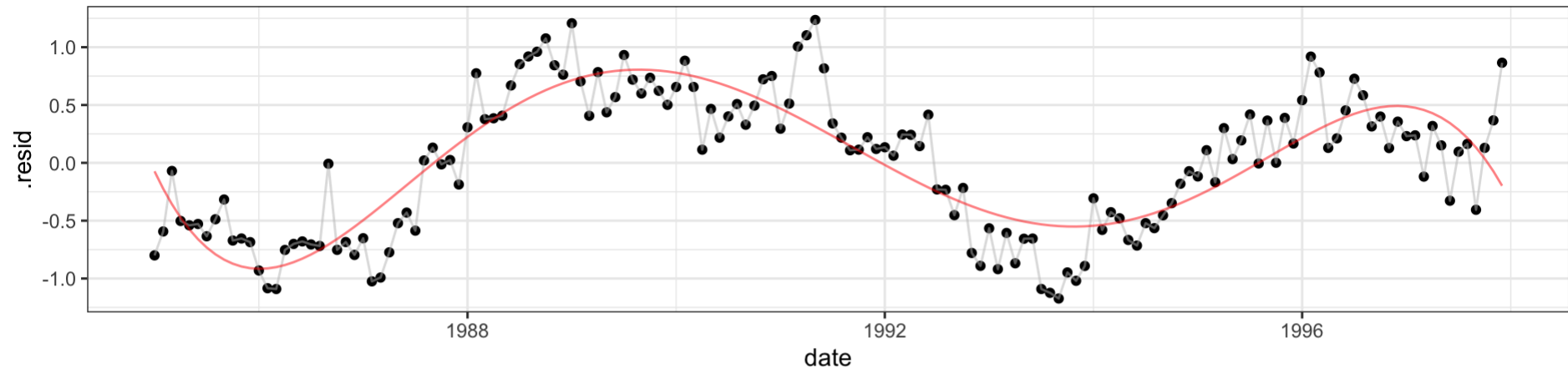
```
1 ls = lm(.resid~month, data=mauna_loa_1)
```



# and then and then?

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

```
1 lsy = lm(.resid~poly(date,5), data=mauna_loa_ls)
```





# Putting it all together ...

```
1 l_comb = lm(co2~date + month + poly(date,5), data=mauna_loa)
2 summary(l_comb)
```

Call:

```
lm(formula = co2 ~ date + month + poly(date, 5), data = mauna_loa)
```

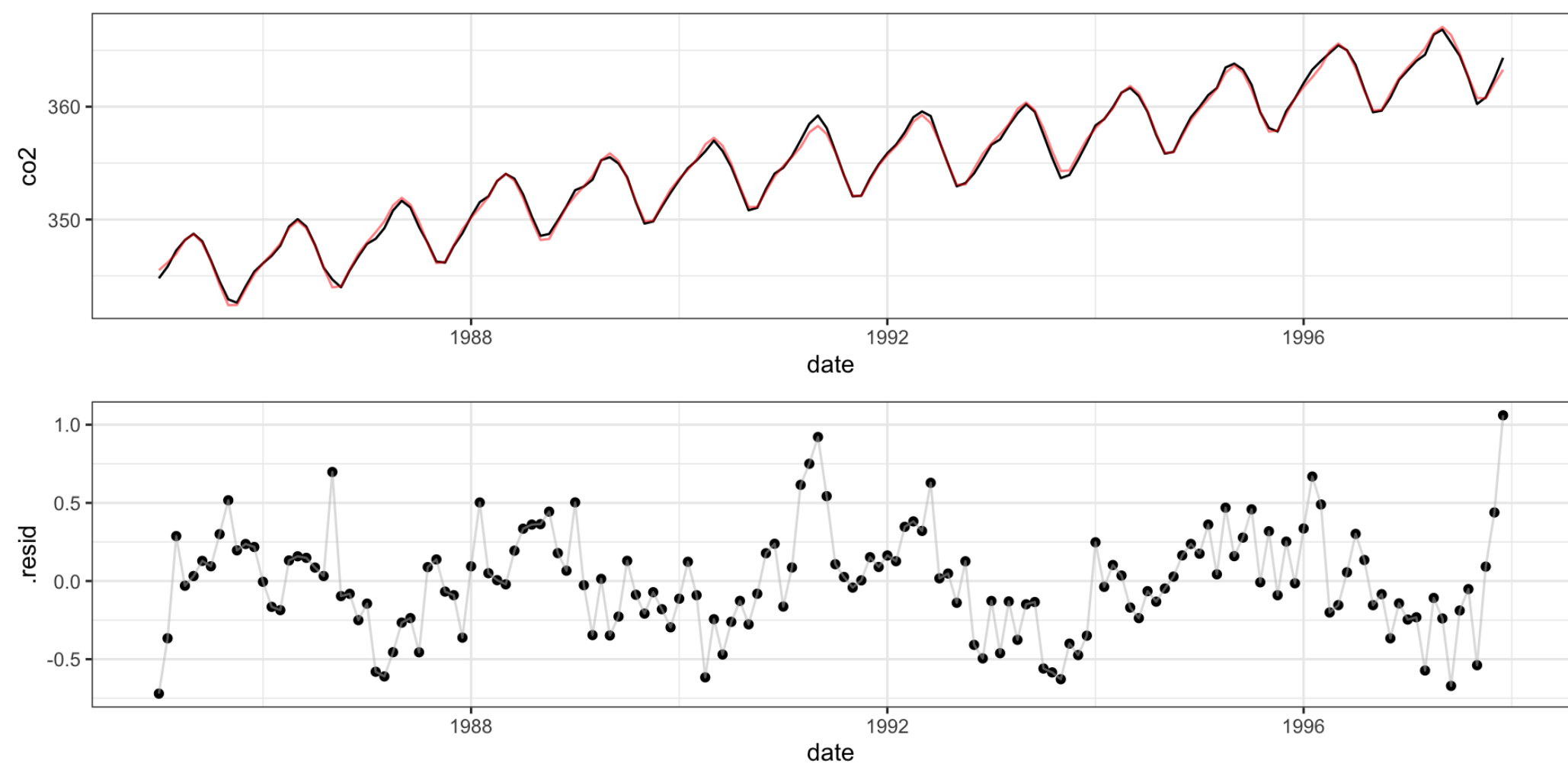
Residuals:

	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	***

# Combined fit + Residuals



# Model performance

Model	rmse
co2 ~ date	2.248
co2 ~ month	5.566
co2 ~ date+month	0.594
co2 ~ poly(date,5)	2.171
co2 ~ month+poly(date,5)	0.323
co2 ~ date+month+poly(date,5)	0.323

# Generalized Linear Models

# Background

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  $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ ,
3. and a link function  $g$  such that  $g(E(\boldsymbol{Y}|\boldsymbol{X})) = \boldsymbol{\eta}$  (or  $E(\boldsymbol{Y}|\boldsymbol{X}) = g^{-1}(\boldsymbol{\eta})$ ).

# 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).

$$Y_i \sim \text{Poisson}(\lambda_i)$$
$$\log E(Y_i | \mathbf{X}_i) = \log \lambda_i = \mathbf{X}_i \cdot \underset{1 \times p \quad p \times 1}{\boldsymbol{\beta}}$$

# Example - AIDS in Belgium

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

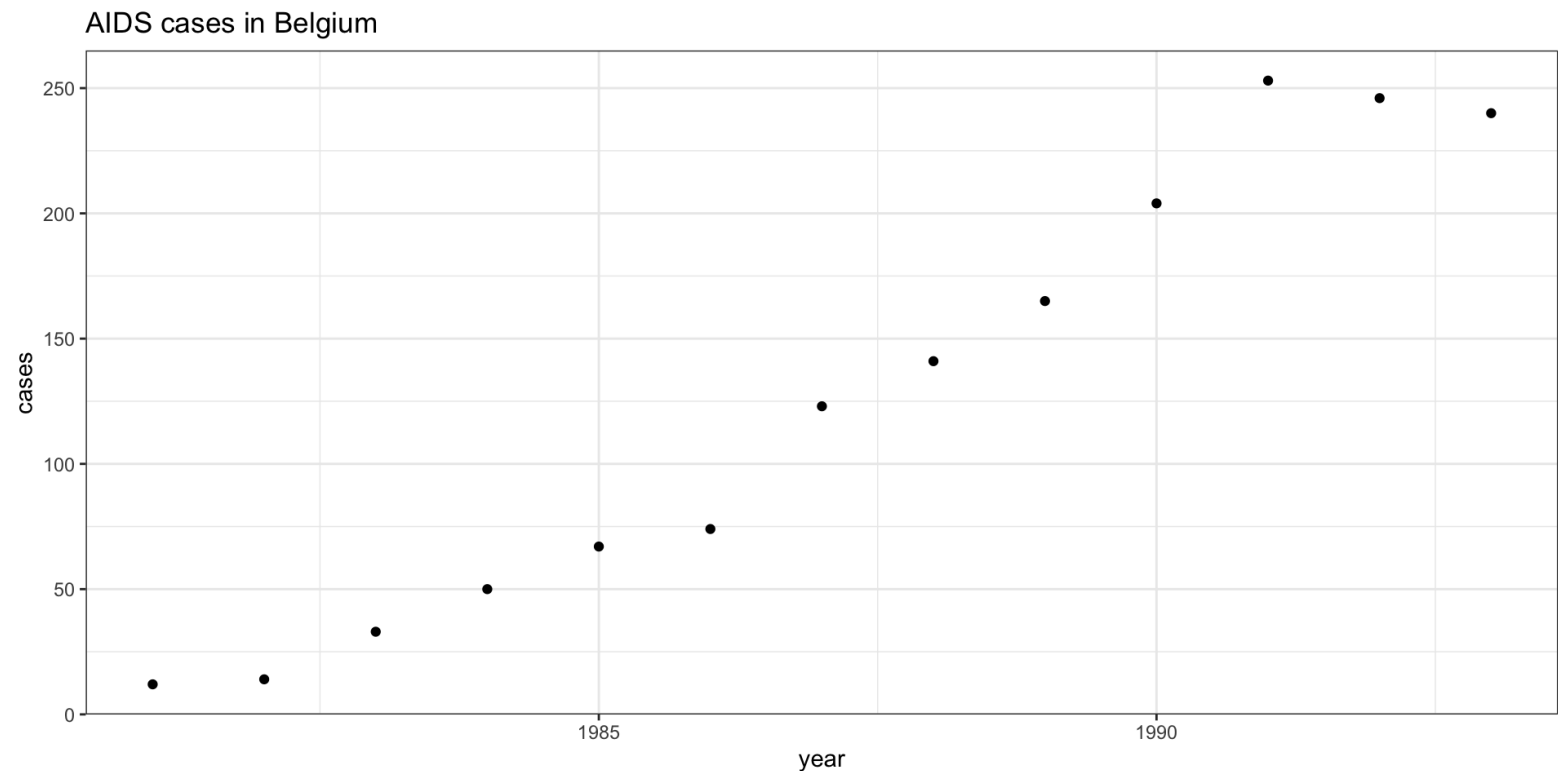
```
1 aids
```

```
# A tibble: 13 × 2
```

```
  year cases
```

```
<int> <int>
```

1	1981	12
2	1982	14
3	1983	33
4	1984	50
5	1985	67
6	1986	74
7	1987	123
8	1988	141
9	1989	165
10	1990	204
11	1991	253
12	1992	246
13	1993	240



# Frequentist glm fit

```
1 ( g = glm(cases~year, data=aids, family=poisson) )
```

Call: `glm(formula = cases ~ year, family = poisson, data = aids)`

Coefficients:

(Intercept)	year
-397.0594	0.2021

Degrees of Freedom: 12 Total (i.e. Null); 11 Residual

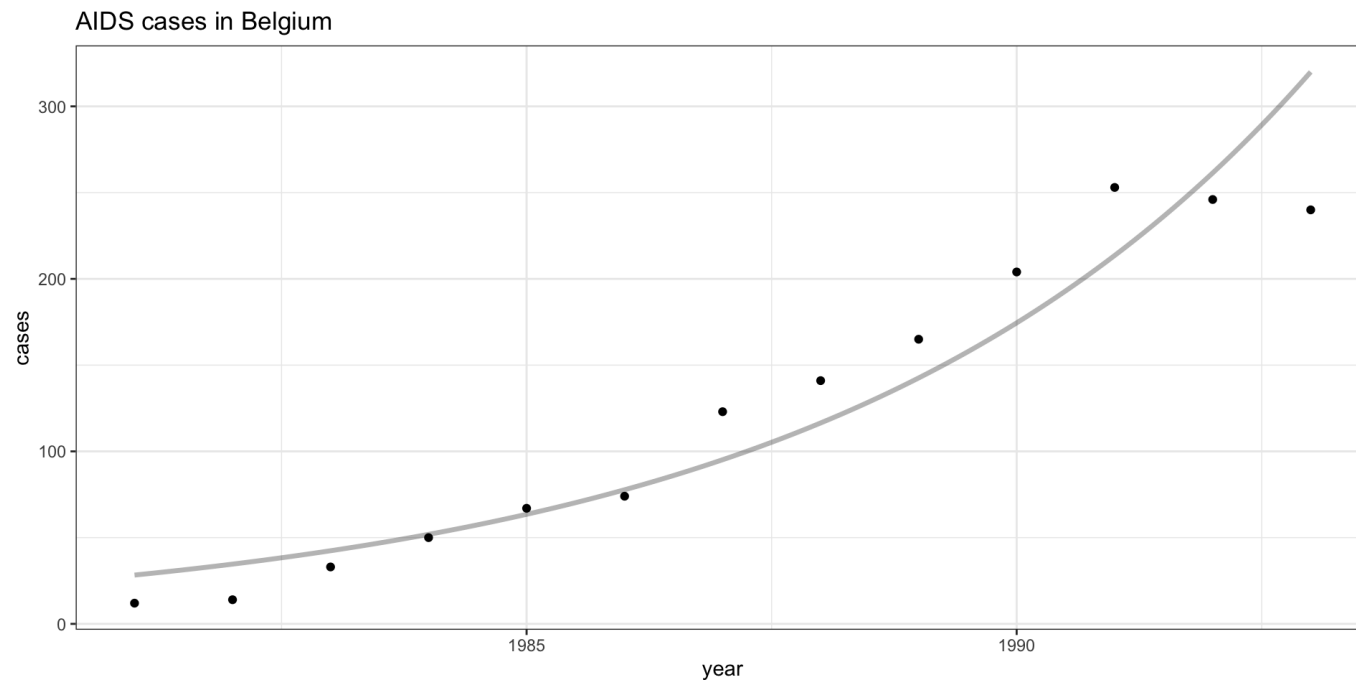
Null Deviance: 872.2

Residual Deviance: 80.69 AIC: 166.4



# Model Fit

```
1 g_pred = broom::augment(  
2   g, type.predict = "response",  
3   newdata = tibble(year=seq(1981,1993,by=0.1))  
4 )  
5  
6 aids_base +  
7   geom_line(data=g_pred, aes(y=.fitted), size=1.2, alpha=0.3)
```

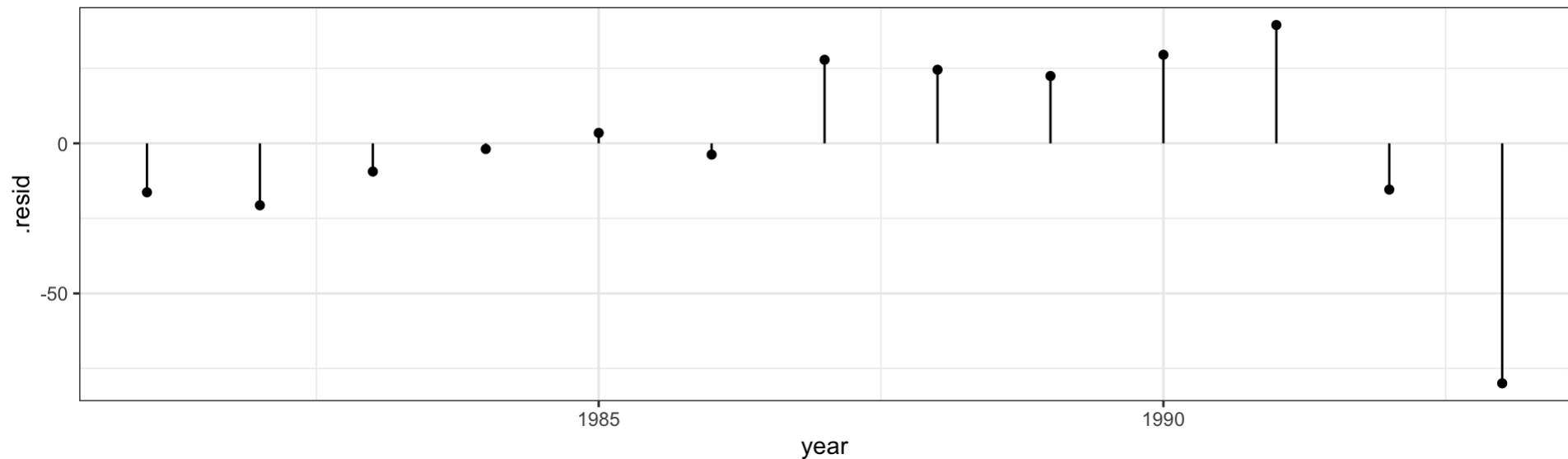


# Residuals?

The naive approach is to use standard residuals,

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

```
1 g_pred_std = broom::augment(  
2   g, type.predict = "response"  
3 ) |>  
4   mutate(.resid = cases - .fitted)
```

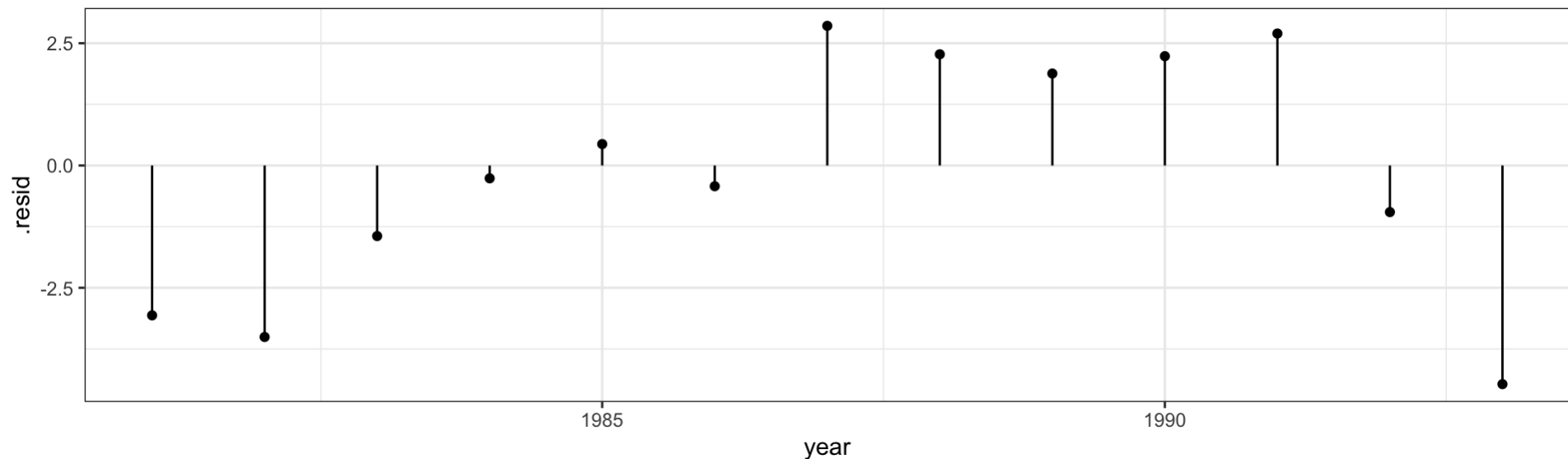


# Accounting for variability

Pearson residuals:

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

```
1 g_pred_pearson = broom::augment(  
2   g, type.predict = "response", type.residuals = "pearson"  
3 )
```



# Deviance

Deviance is a way of measuring the difference between a GLM's fit and the fit of a perfect model (i.e. where  $\theta_{\text{best}} = E(Y_i|X) = Y_i$ ).

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

$$\begin{aligned} D &= 2 \log \left( \frac{\ell(\theta_{\text{best}}|Y)}{\ell(\hat{\theta}|Y)} \right) \\ &= 2 \left( \ell(\theta_{\text{best}}|Y) - \ell(\hat{\theta}|Y) \right) \end{aligned}$$

# Derivation - Normal

# Derivation - Poisson

# glm output

```
1 summary(g)
```

Call:

```
glm(formula = cases ~ year, family = poisson, data = aids)
```

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

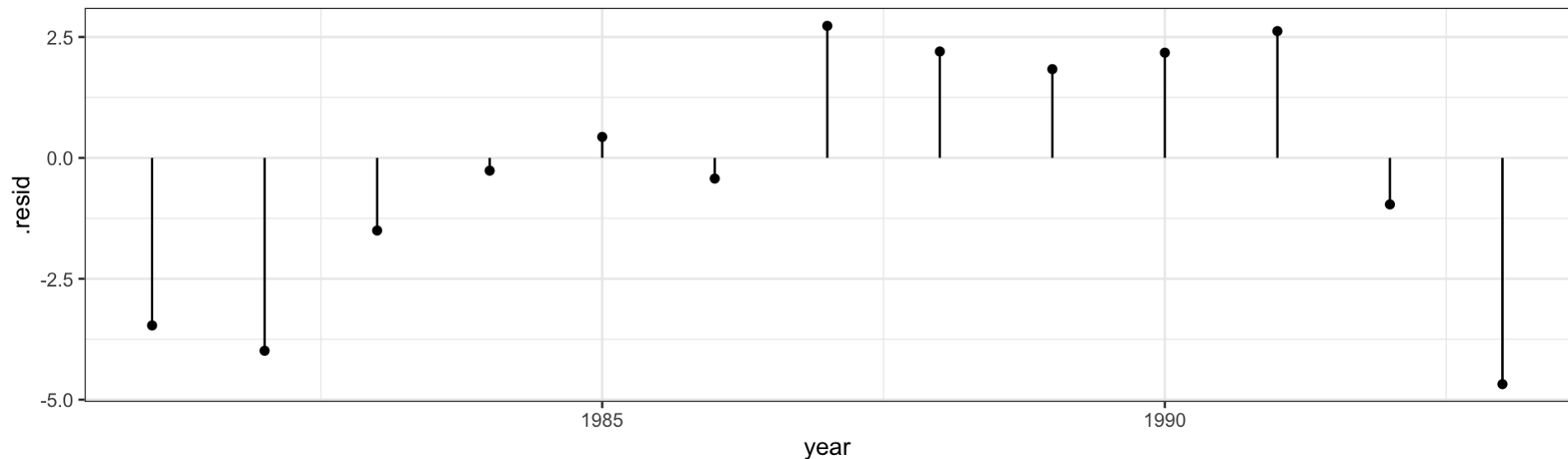
# Deviance residuals

We can therefore think of deviance as  $D = \sum_{i=1}^n d_i^2$  where  $d_i$  is a generalized residual.

In the Poisson case we have,

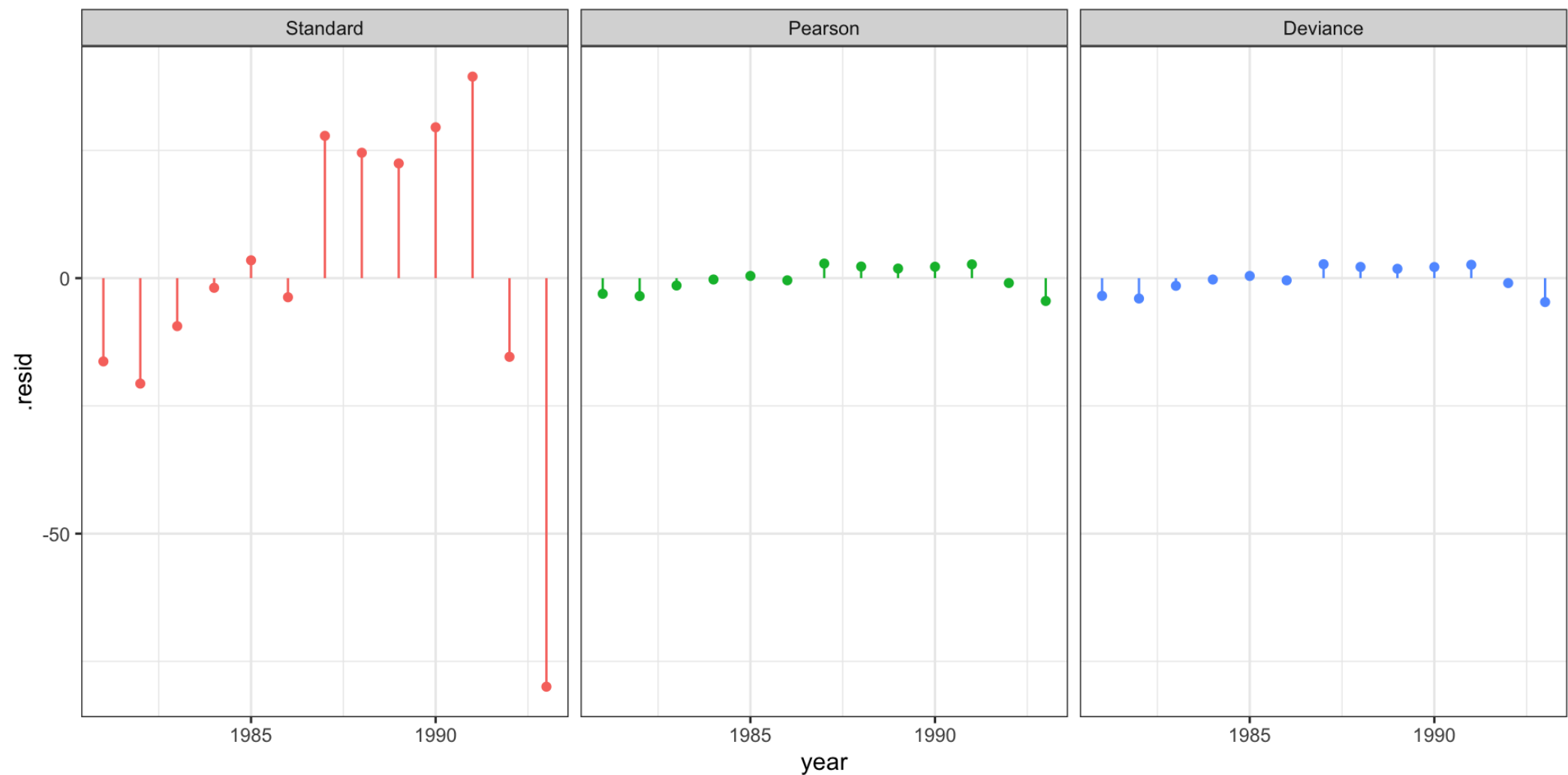
$$d_i = \text{sign}(y_i - \lambda_i) \sqrt{2(y_i \log(y_i/\hat{\lambda}_i) - (y_i - \hat{\lambda}_i))}$$

```
1 g_pred_dev = broom::augment(  
2   g, type.predict = "response", type.residuals = "deviance"  
3 )
```

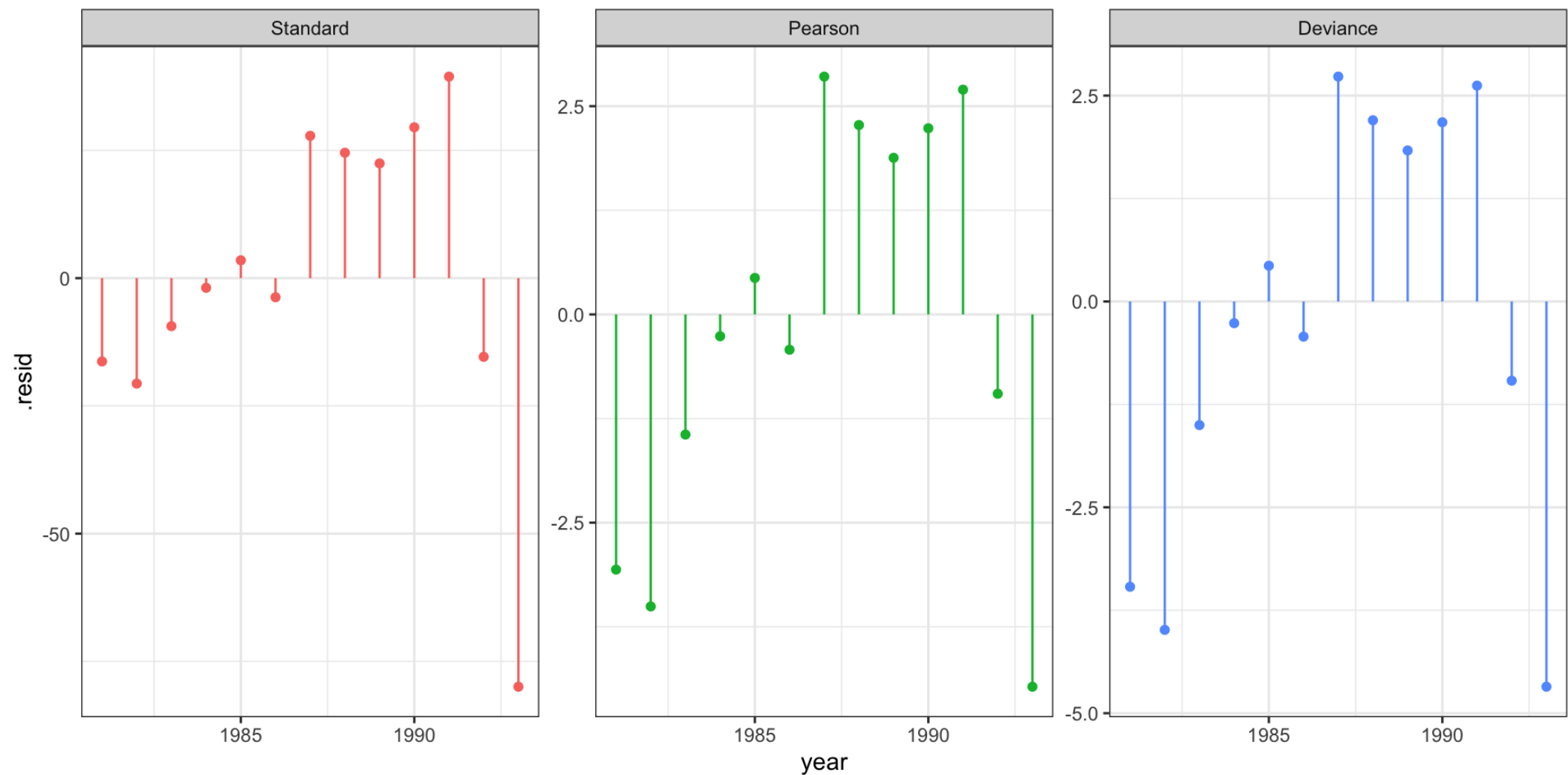




# Comparing Residuals



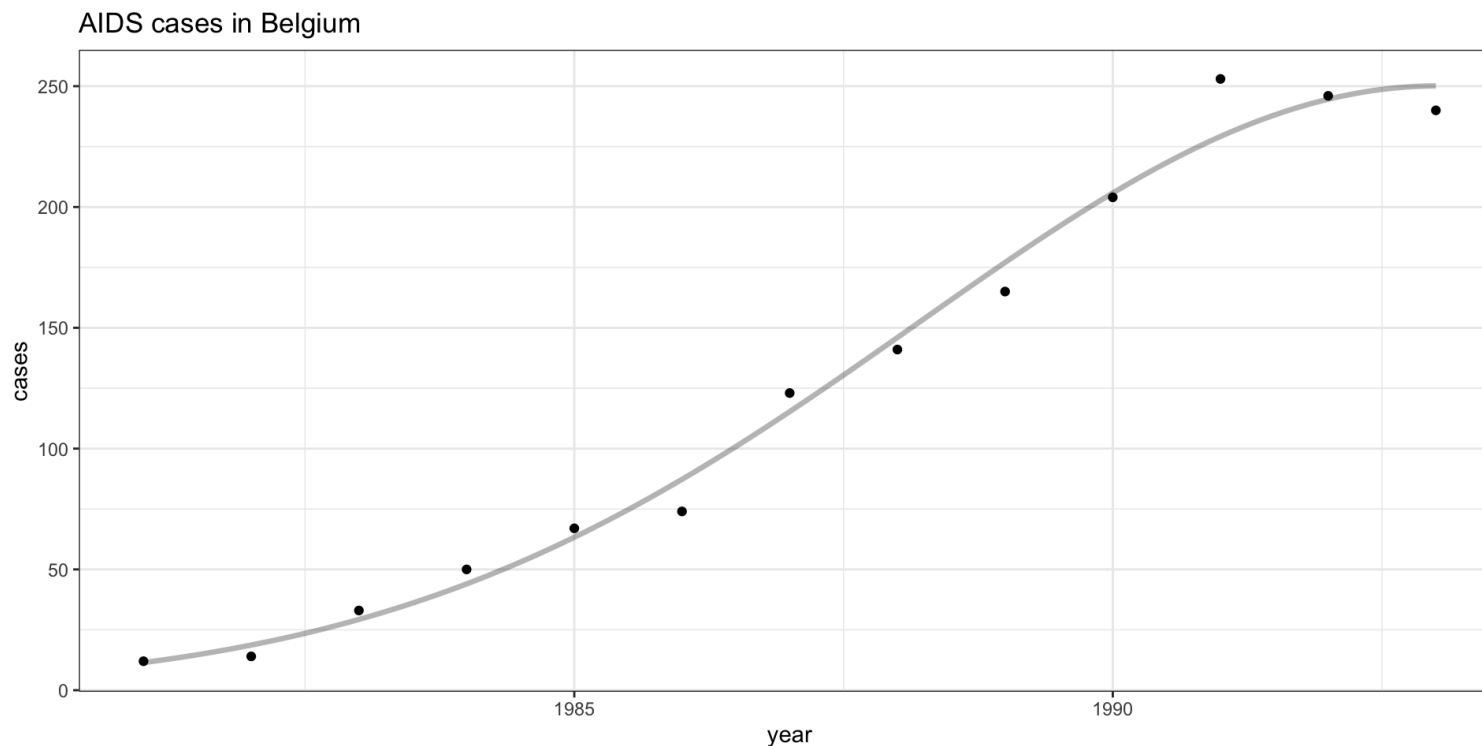
# Comparing Residuals - scale free



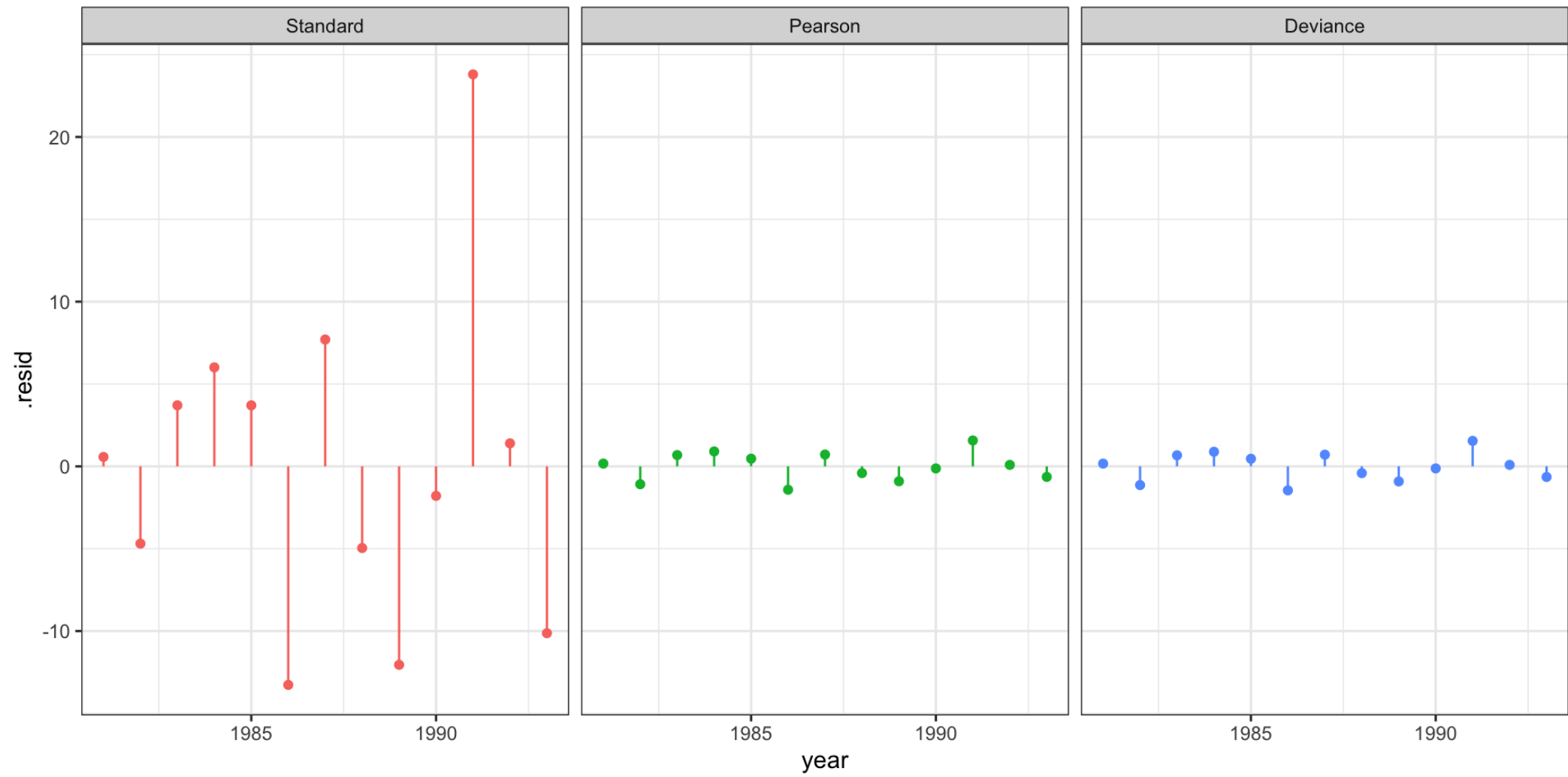
# Updating the model

# Quadratic fit

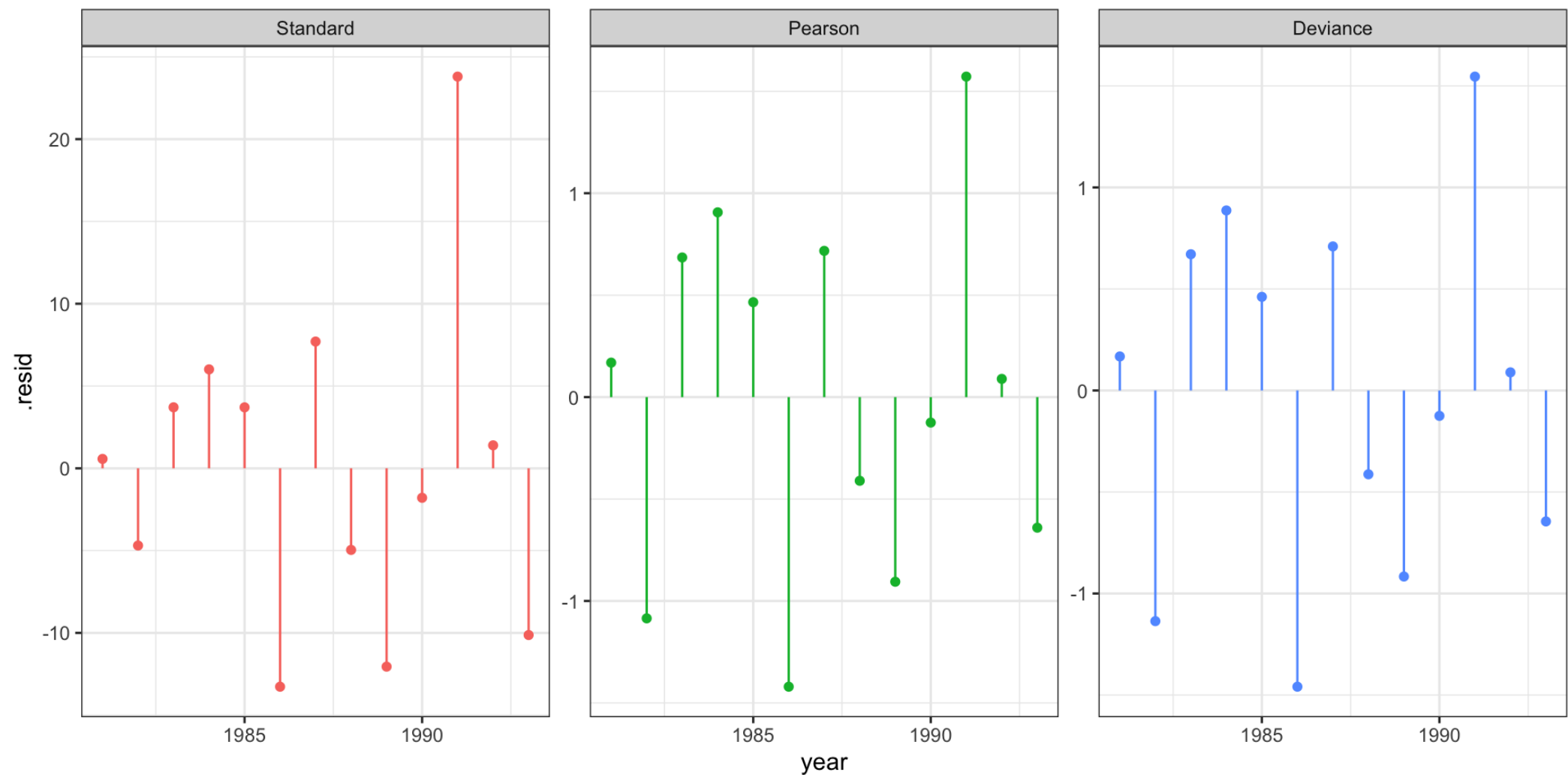
```
1 g2 = glm(cases~year+I(year^2), data=aids, family=poisson)
2
3 g2_pred = broom::augment(
4   g2, type.predict = "response",
5   newdata=tibble(year=seq(1981,1993,by=0.1))
6 )
```



# Quadratic fit - residuals



# Quadratic fit - residuals (scale free)



# Bayesian Model

# Bayesian Poisson Regression Model

```
1 ( g_bayes = brms::brm(  
2   cases~year, data=aids, family=poisson,  
3   silent=2, refresh=0  
4 ) )
```

Family: poisson

Links: mu = log

Formula: cases ~ year

Data: aids (Number of observations: 13)

Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;  
total post-warmup draws = 4000

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-396.93	15.47	-427.20	-366.60	1.00	1502	2091
year	0.20	0.01	0.19	0.22	1.00	1504	2091

Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).



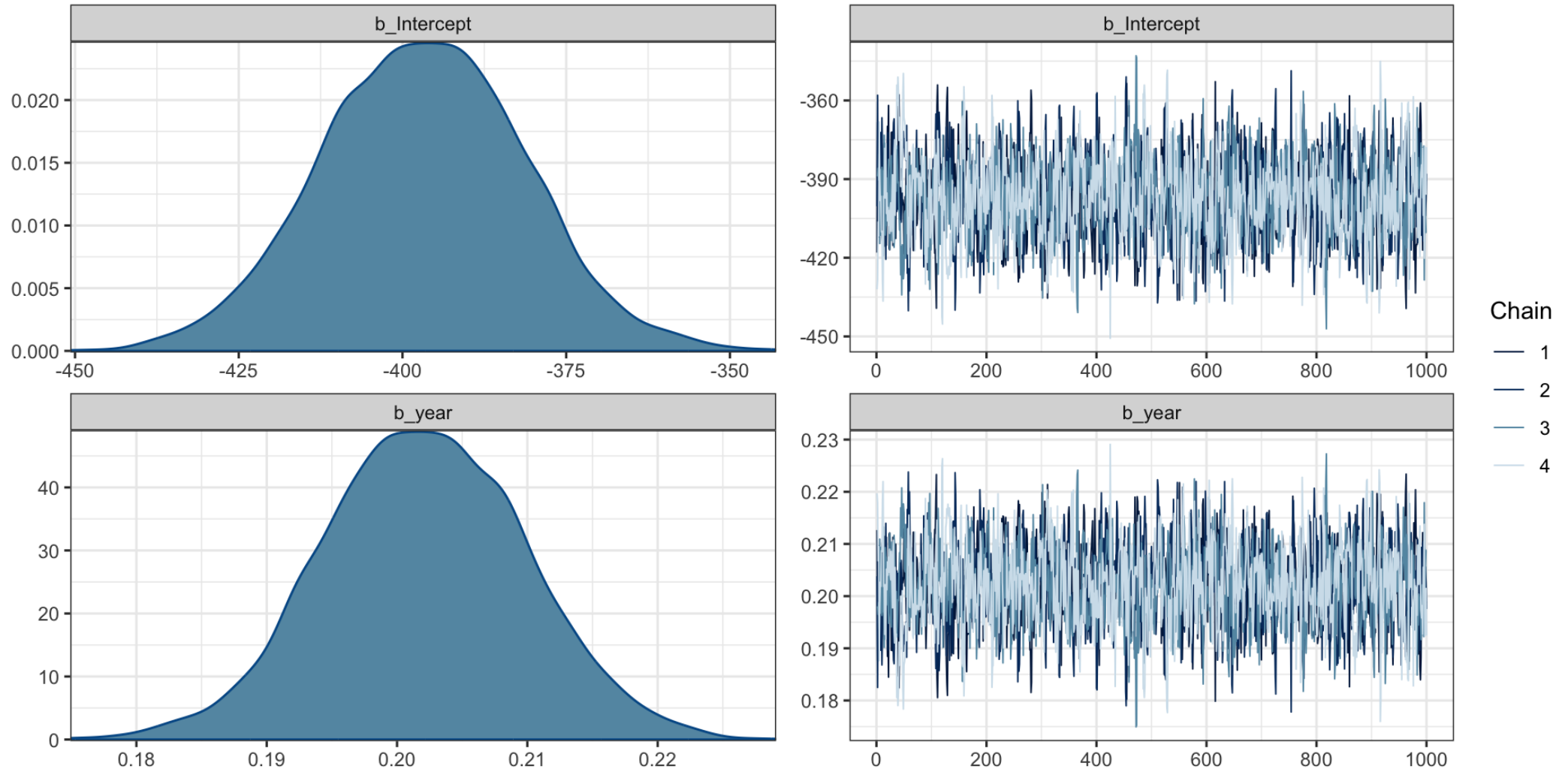
# Model priors

```
1 brms::prior_summary(g_bayes)
```

	prior	class	coef	group	resp	dpar	nlpar	lb	ub	source
	(flat)		b							default
	(flat)		b	year						(vectorized)
student_t(3, 4.8, 2.5)			Intercept							default

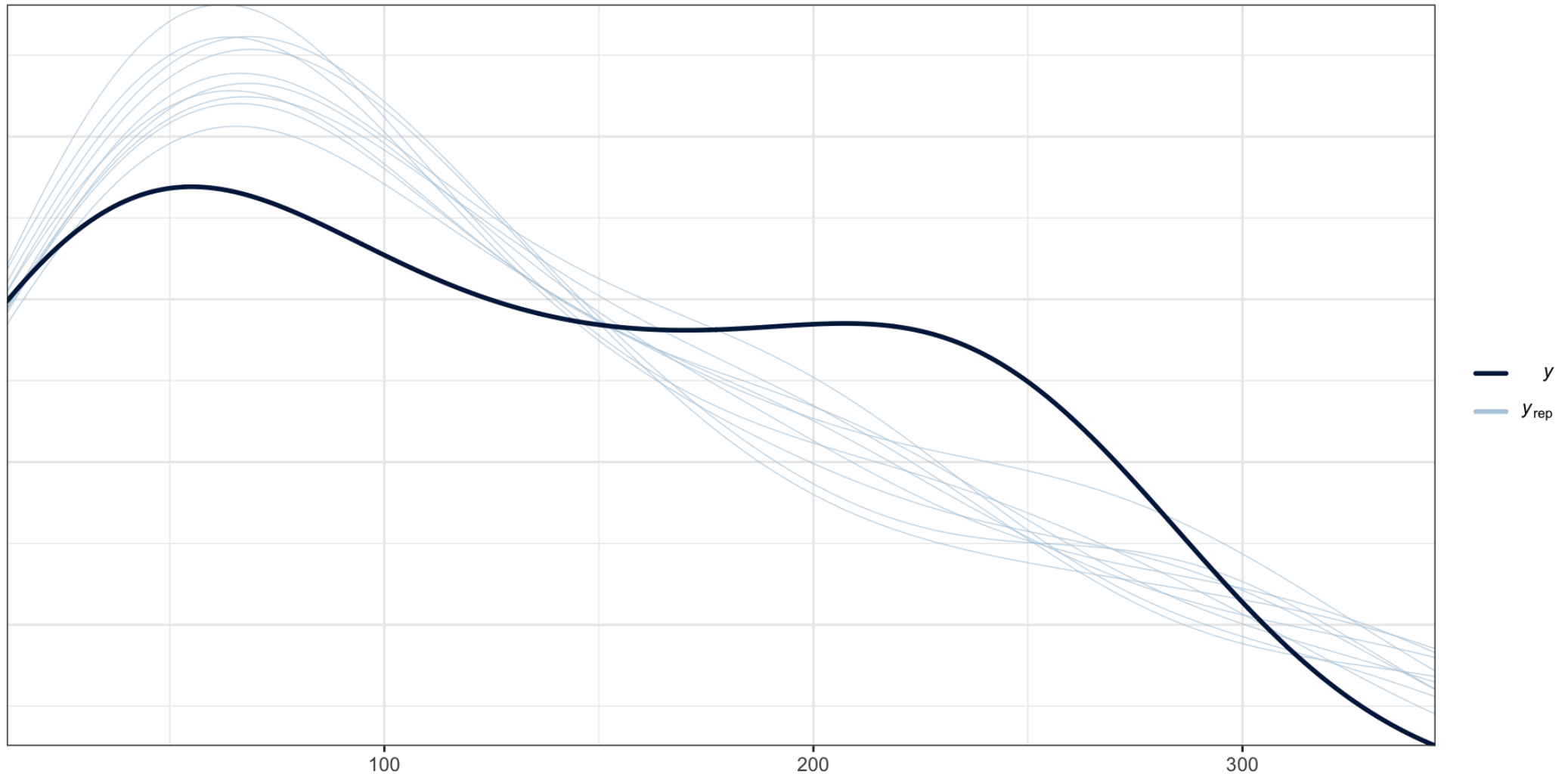
# MCMC Diagnostics

```
1 plot(g_bayes)
```



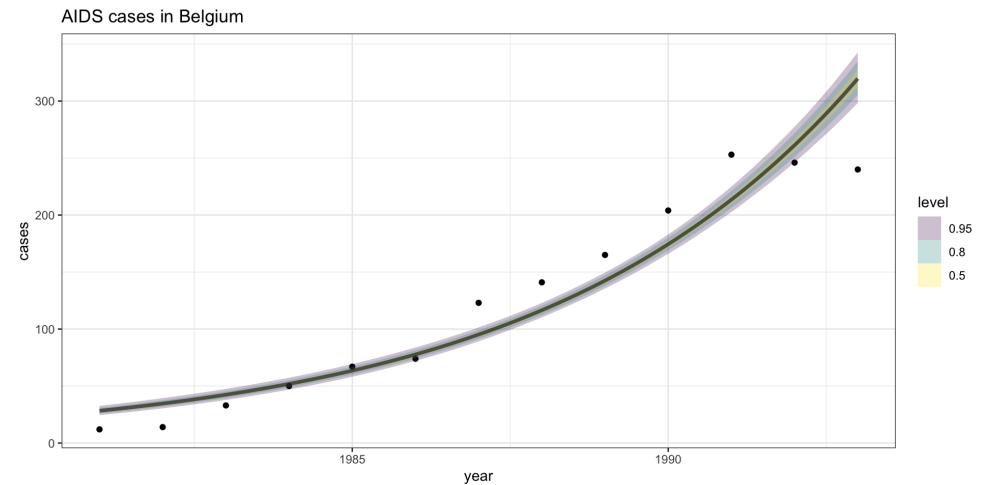
# Posterior Predictive Check

```
1 brms::pp_check(g_bayes)
```



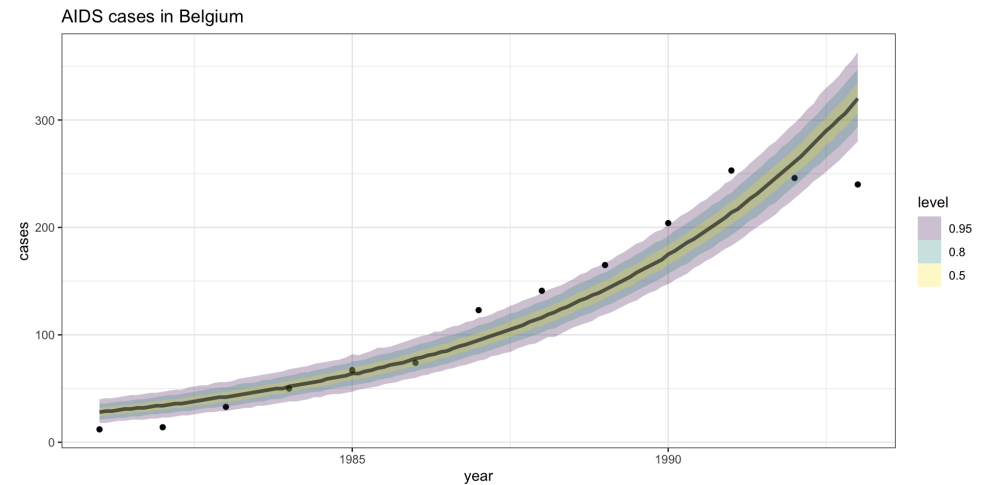
# Model fit - $\lambda$ CI

```
1 aids_base +  
2   ggdist::stat_lineribbon(  
3     data = tidybayes::epred_draws(  
4       g_bayes,  
5       newdata = tibble(year=seq(1981,1993,by=  
6     )),  
7     aes(y=.epred),  
8     alpha=0.25  
9   )
```



# Model fit - Y CI

```
1 aids_base +  
2   ggdist::stat_lineribbon(  
3     data = tidybayes::predicted_draws(  
4       g_bayes,  
5       newdata = tibble(year=seq(1981,1993,by=  
6     )),  
7     aes(y=.prediction),  
8     alpha=0.25  
9   )
```



# Residuals

```
1 ( g_bayes_resid = residual_draws_fix(  
2   g_bayes, newdata = aids  
3 ) )
```

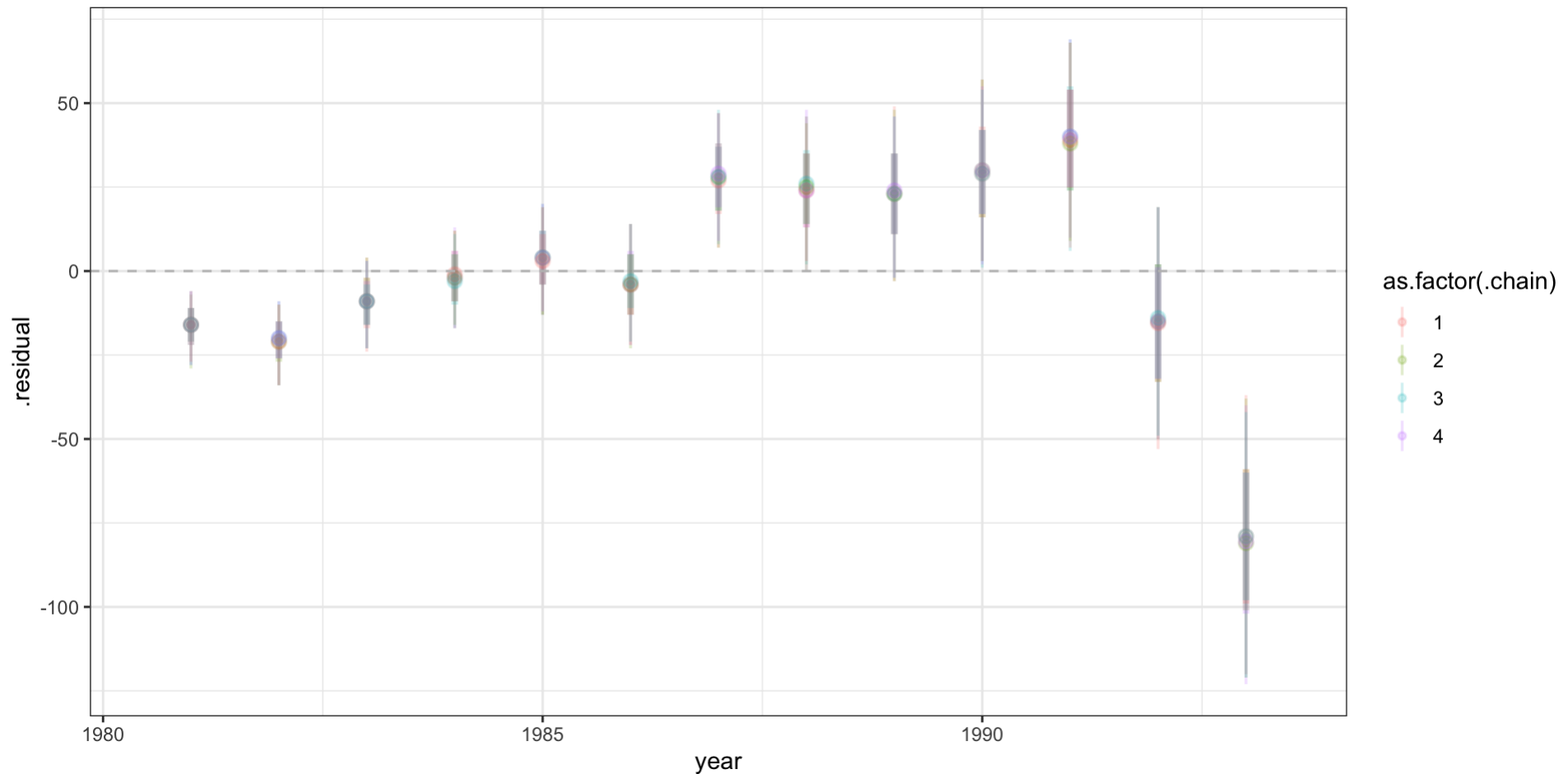
```
# A tibble: 52,000 × 7
```

	year	cases	.row	.chain	.iteration	.draw	.residual
	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	1981	12	1	1	1	1	-18
2	1981	12	1	1	2	2	-21
3	1981	12	1	1	3	3	-19
4	1981	12	1	1	4	4	-12
5	1981	12	1	1	5	5	-17
6	1981	12	1	1	6	6	-11
7	1981	12	1	1	7	7	-13
8	1981	12	1	1	8	8	-21
9	1981	12	1	1	9	9	-18
10	1981	12	1	1	10	10	-5

```
# i 51,990 more rows
```

# Residual plot

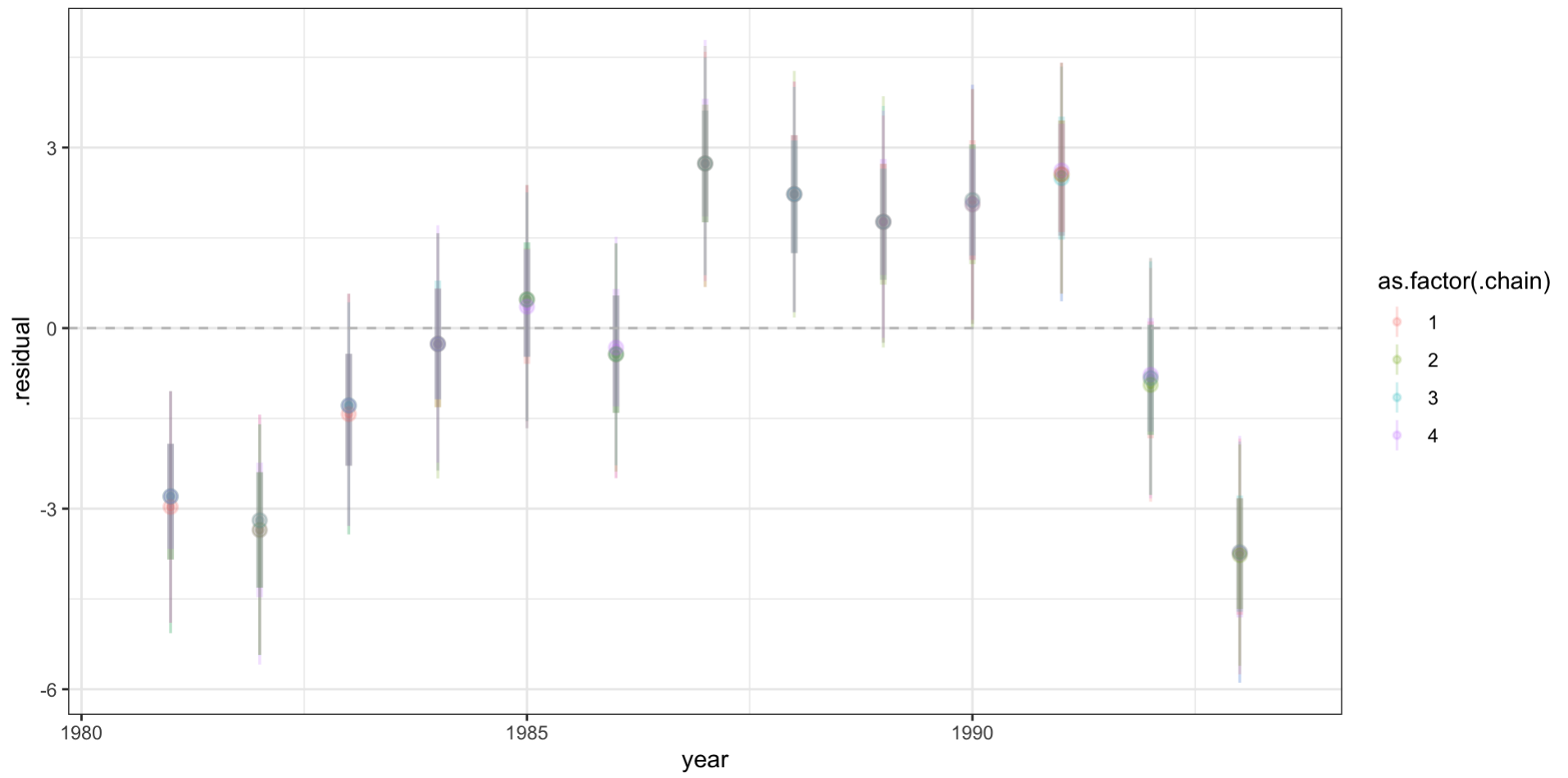
```
1 g_bayes_resid |>
2   ggplot(aes(y = .residual, x = year, color=as.factor(.chain), group=.chain)) +
3     ggdist::stat_pointinterval(alpha=0.2) +
4     geom_hline(yintercept = 0, color='grey', linetype=2)
```



# Standardized residuals?

```
1 residual_draws_fix(  
2   g_bayes, newdata = aids, type = "pearson"  
3 ) |>  
4   ggplot(aes(y = .residual, x = year, color=as.factor(.chain), group=.chain)) +  
5     ggdist::stat_pointinterval(alpha=0.2) +  
6     geom_hline(yintercept = 0, color='grey', linetype=2)
```





# Model performance - rmse, crps

```
1 predicted_draws_fix(g_bayes, newdata = aids) |>
2   group_by(.chain, .row) |>
3   summarize(
4     rmse = yardstick::rmse_vec(cases, .prediction),
5     crps = dukestm::calc_crps(.prediction, cases)
6   ) |>
7   group_by(.chain) |>
8   summarize(
9     rmse = mean(rmse),
10    crps = mean(crps)
11  )
```

# A tibble: 4 × 3

	.chain	rmse	crps
	<int>	<dbl>	<dbl>
1	1	26.3	17.7
2	2	26.2	17.6
3	3	26.3	17.8
4	4	26.3	17.7

# Model performance - emp coverage

```
1 predicted_draws_fix(g_bayes, newdata = aids) |>
2   group_by(.row, cases) |>
3   ggdist::mean_hdi(
4     .prediction, .width = c(0.5, 0.9, 0.95)
5   ) |>
6   mutate(contains = cases >= .lower & cases <= .upper) %>%
7   group_by(.width) |>
8   summarize(
9     emp_cov = sum(contains)/n()
10  )
```

# A tibble: 3 × 2

	.width	emp_cov
	<dbl>	<dbl>
1	0.5	0.231
2	0.9	0.385
3	0.95	0.462

