

Logistic Regression and Residual Analysis

Lecture 05

Dr. Colin Rundel

Last time

Last time

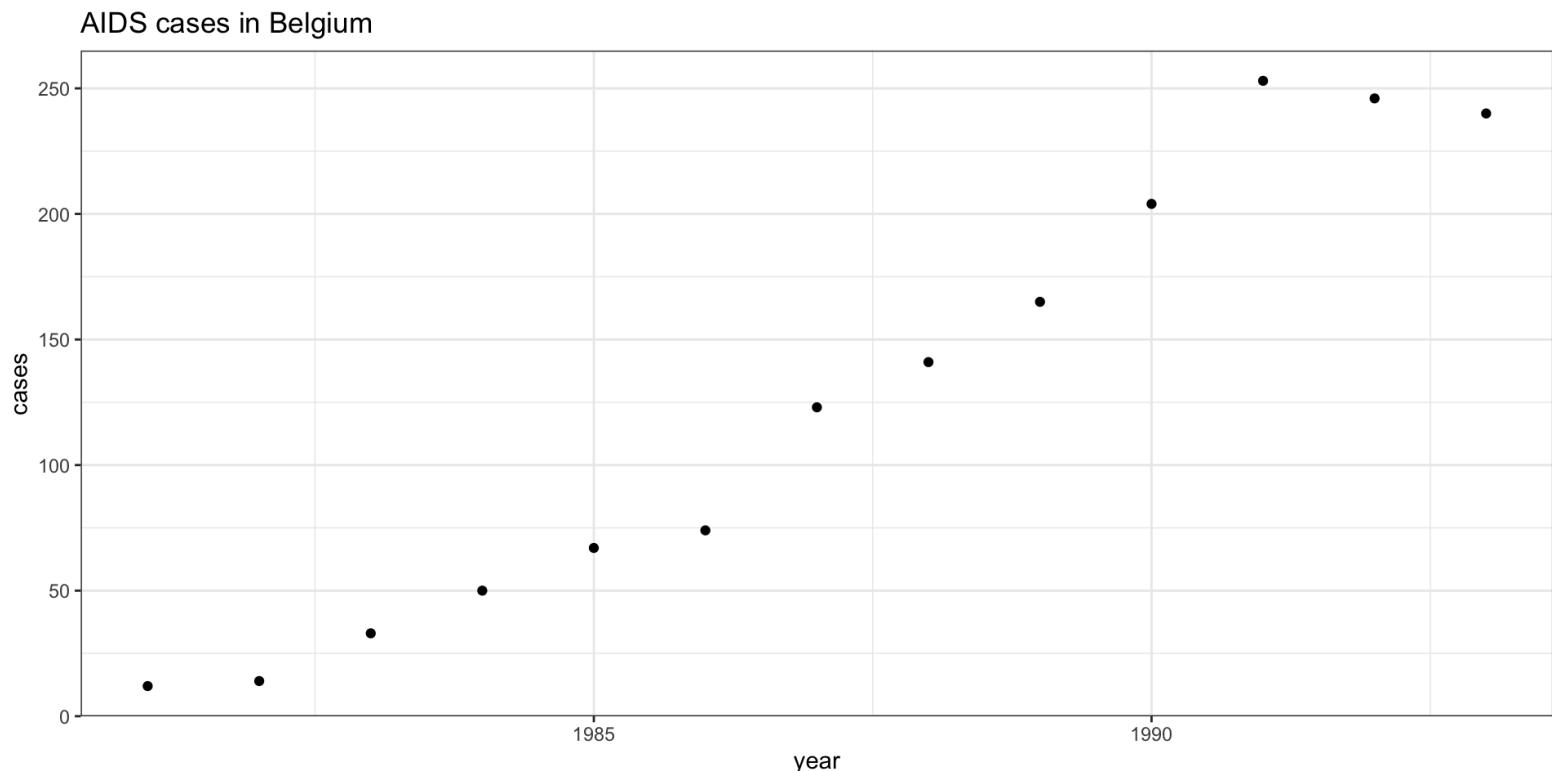
```
1 aids
```

```
# 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
```



Model Fit

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

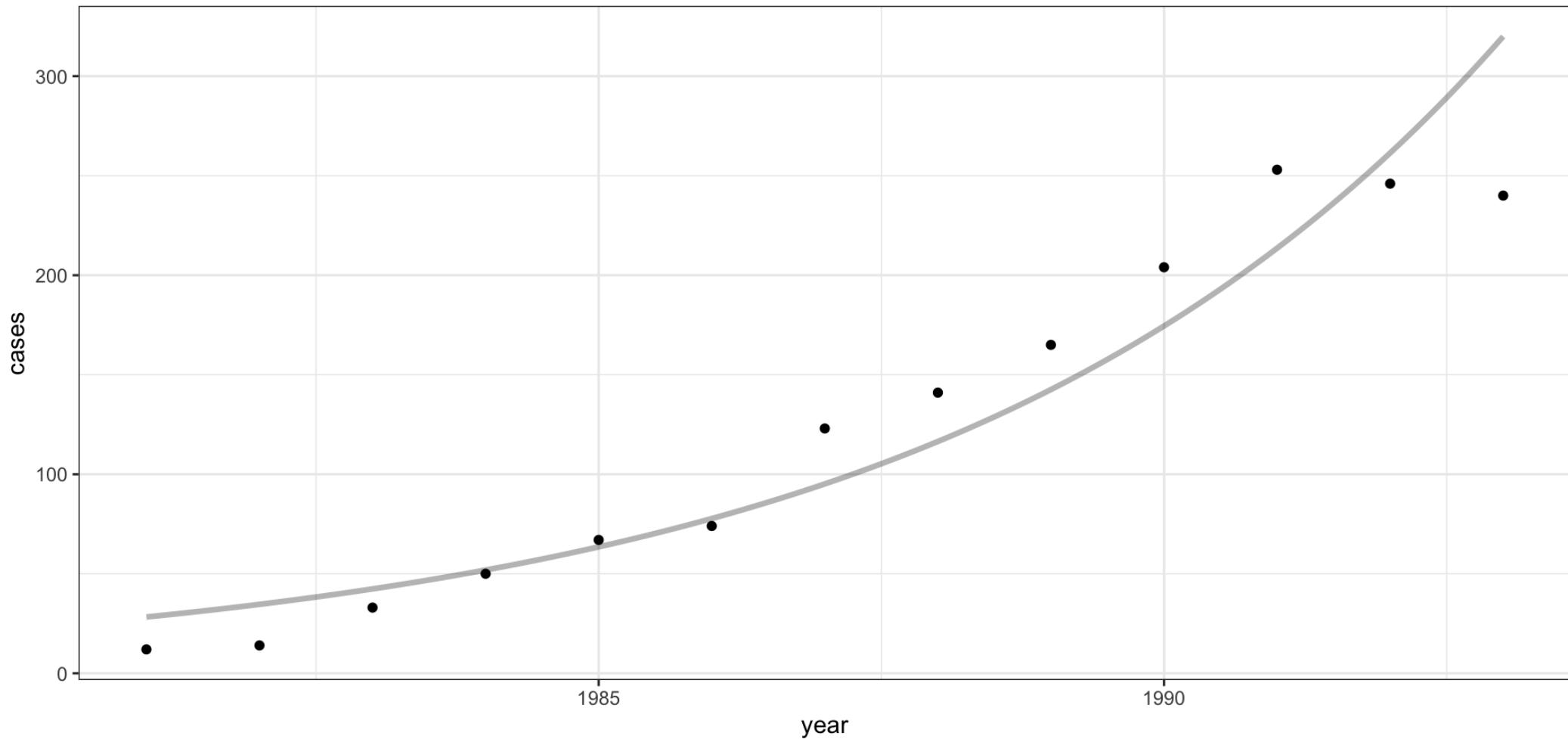
```
# A tibble: 121 × 2
```

```
  year .fitted
```

```
  <dbl>    <dbl>
```

```
1 1981      28.3
2 1981.     28.9
3 1981.     29.5
4 1981.     30.1
5 1981.     30.7
6 1982.     31.3
7 1982.     31.9
8 1982.     32.6
9 1982.     33.3
```

AIDS cases in Belgium



Bayesian Poisson Regression Model

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

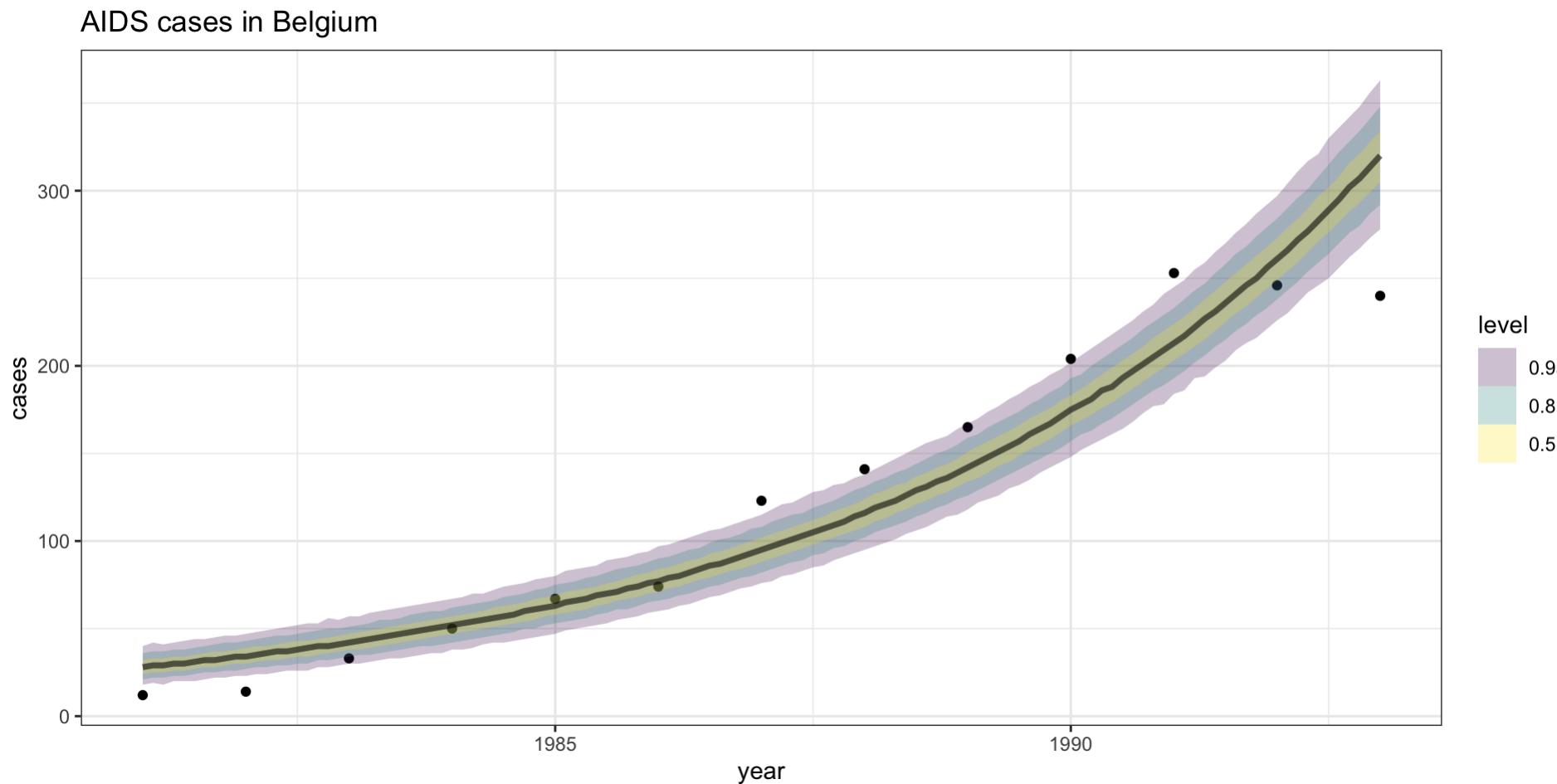
```
Family: poisson  
Links: mu = log  
Formula: cases ~ year  
Data: aids (Number of observations: 13)  
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;  
       total post-warmup draws = 4000
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-397.24	15.85	-427.98	-366.73	1.00	1356	1912
year	0.20	0.01	0.19	0.22	1.00	1357	1912

Draws were sampled using `sample(hmc)`. 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 fit



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  17.6
2      2  26.5  17.8
3      3  26.2  17.4
4      4  26.4  17.7
```

Model performance - emp coverage

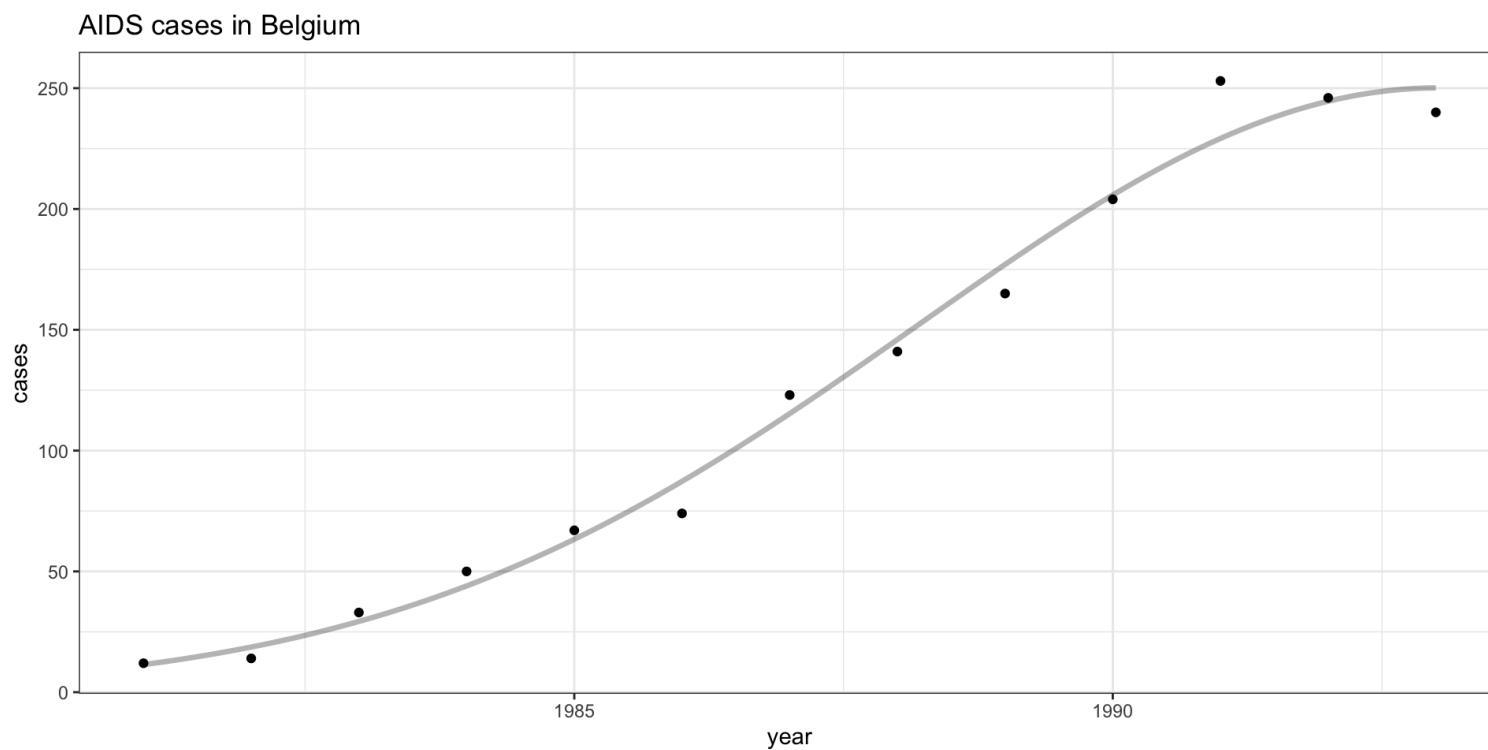
```
1 predicted_draws_fix(g_bayes, newdata = aids) |>
2   group_by(.row, cases) |>
3   ggdist::mean_hdi(
4     .prediction, .width = c(0.5,0.9,0.95)
5   ) |>
6   mutate(contains = cases >= .lower & cases <= .upper) %>%
7   group_by(.width) |>
8   summarize(
9     emp_cov = sum(contains)/n()
10    )
```

```
# A tibble: 3 × 2
  .width emp_cov
  <dbl>    <dbl>
1 0.5      0.231
2 0.9      0.385
3 0.95     0.462
```

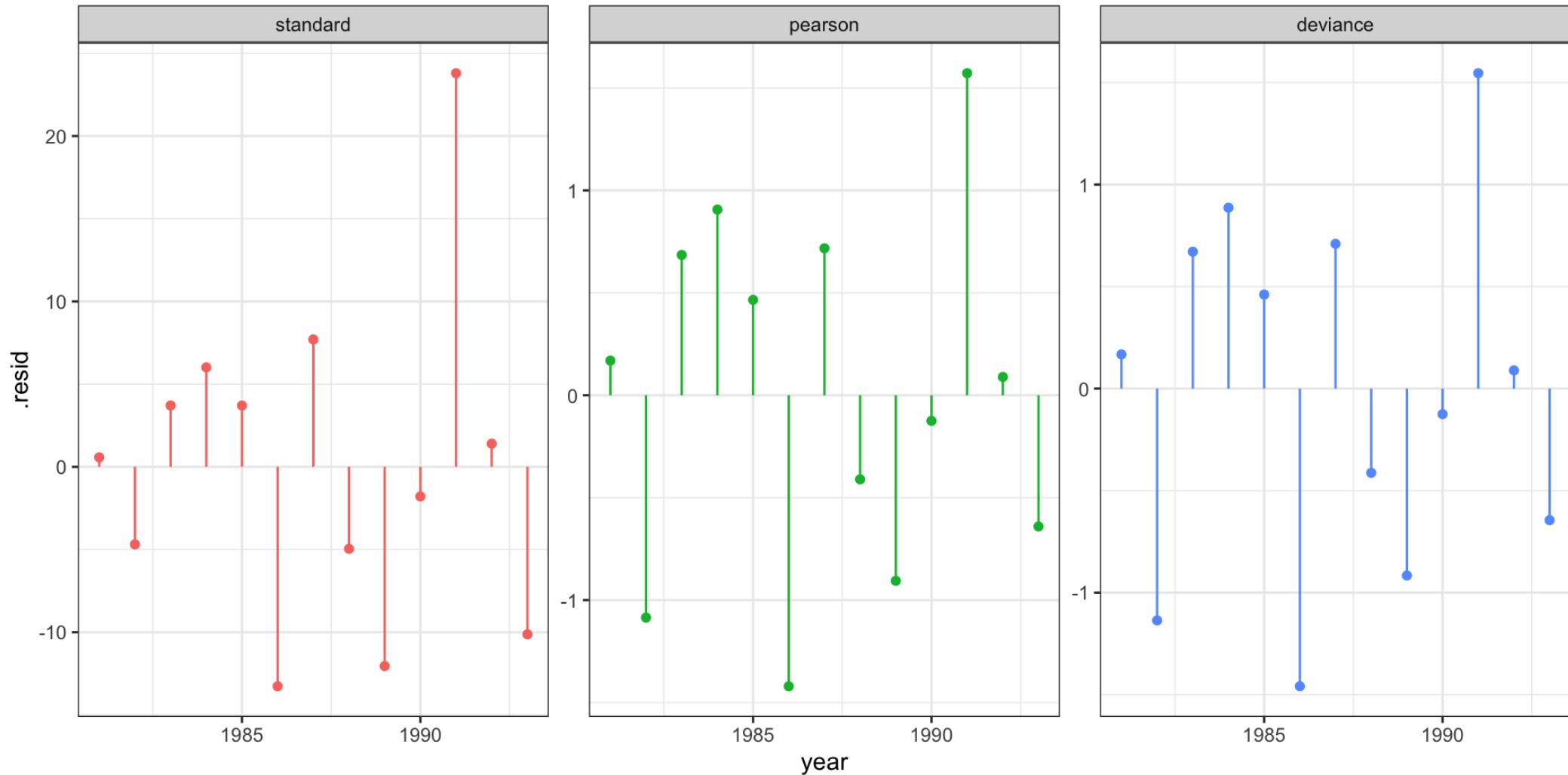
Updating the model

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



Bayesian quadratic fit

```
1 g2_bayes = brms::brm(  
2   cases~year+I(year^2), data=aids, family=poisson,  
3   refresh=0, backend = "cmdstanr"  
4 ) )
```

Family: poisson

Links: mu = log

Formula: cases ~ year + I(year^2)

Data: aids (Number of observations: 13)

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

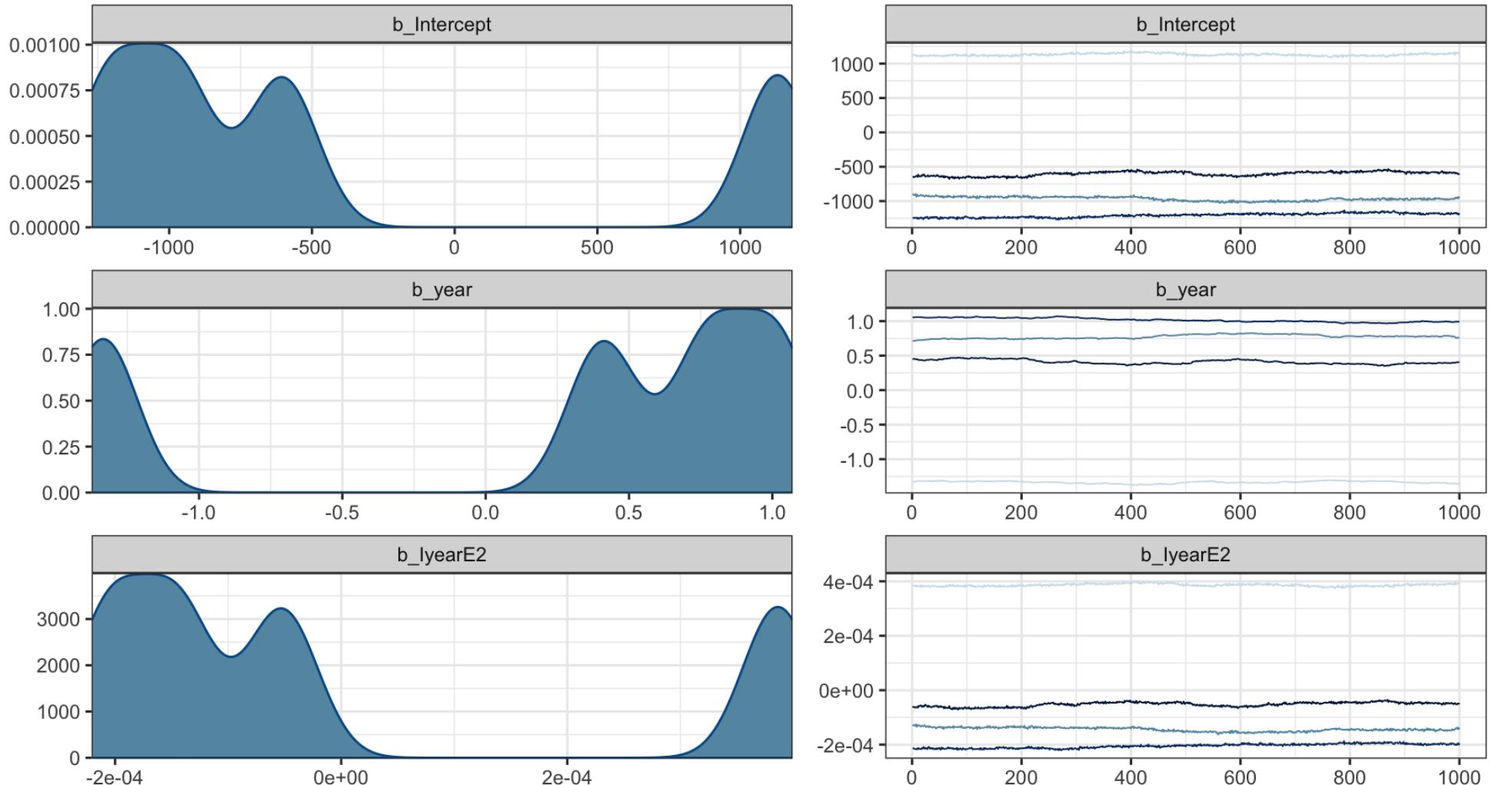
Population-Level Effects:

	Estimate	Est.Error	l-95%	CI	u-95%	CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-410.98	916.52	-1246.31	1156.17	3.45		4	11	
year	0.22	0.92	-1.36	1.06	3.48		4	11	
IyearE2	-0.00	0.00	-0.00	0.00	3.43		4	11	

Draws were sampled using sample(hmc). 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).

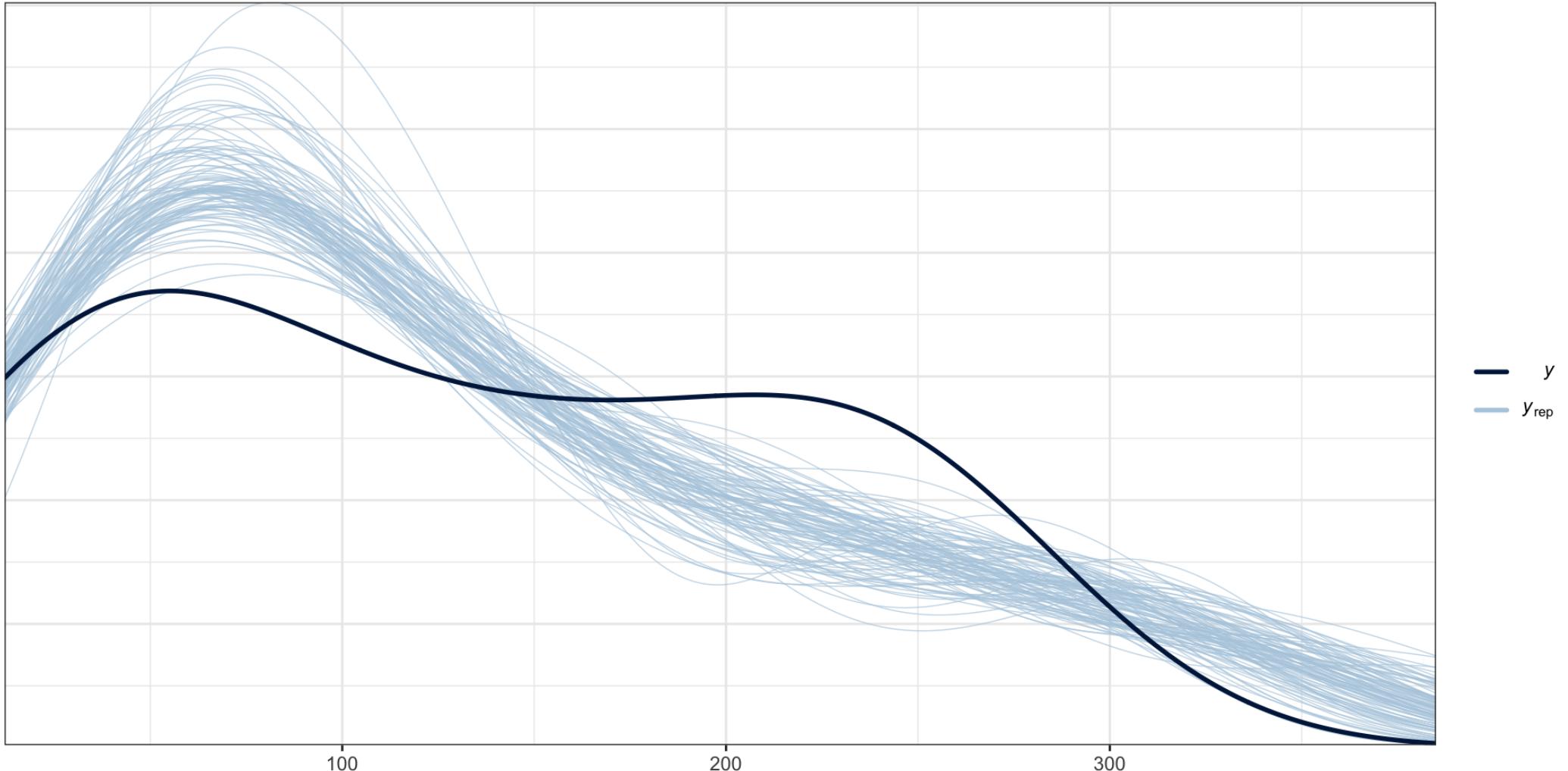
Diagnostics

```
1 plot(g2_bayes)
```



PP Checks

```
1 brms::pp_check(g2_bayes, ndraws=100)
```



Bayesian quadratic fit (fixed)

```
1 g2_bayes = brms::brm(  
2   cases~I(year-min(year))+I((year-min(year))^2), data=aids, family=poisson,  
3   refresh=0, backend = "cmdstanr"  
4 ) )
```

Family: poisson

Links: mu = log

Formula: cases ~ I(year - min(year)) + I((year - min(year))^2)

Data: aids (Number of observations: 13)

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

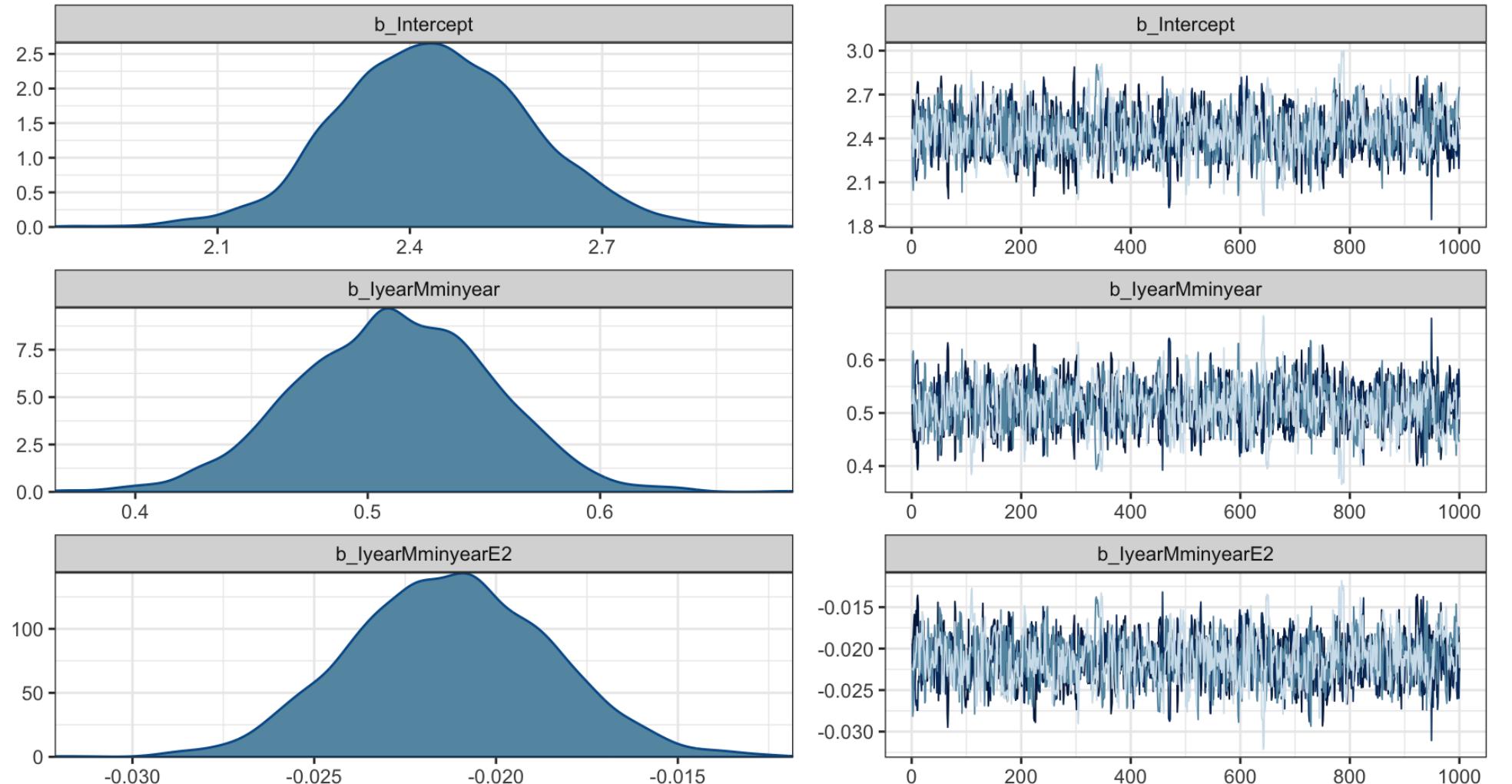
Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	2.44	0.15	2.15	2.74	1.00	1082	1331
IyearMminyear	0.51	0.04	0.43	0.59	1.00	1098	1229
IyearMminyearE2	-0.02	0.00	-0.03	-0.02	1.00	1170	1247

Draws were sampled using sample(hmc). 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).

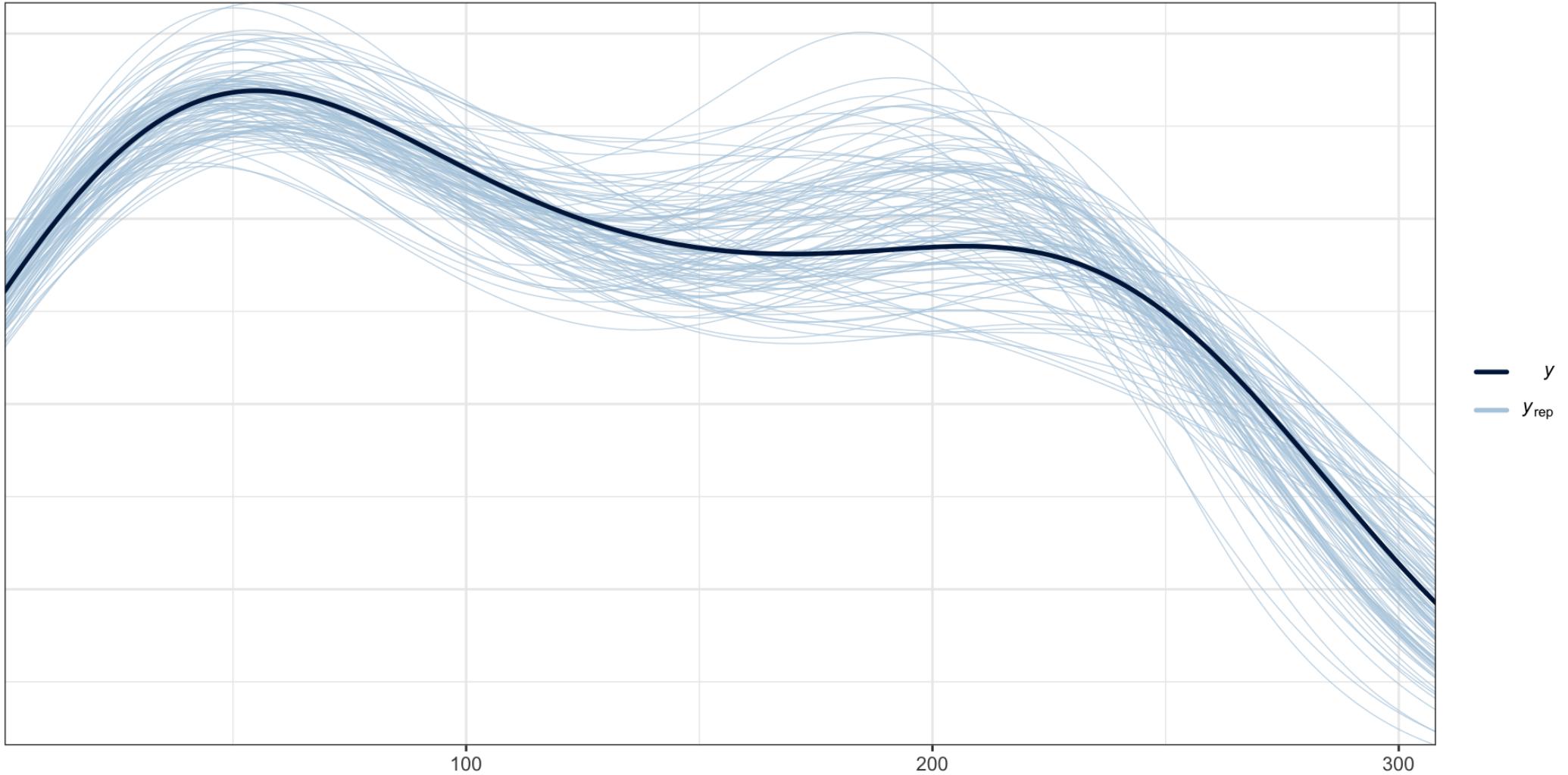
Diagnostics

```
1 plot(g2_bayes)
```



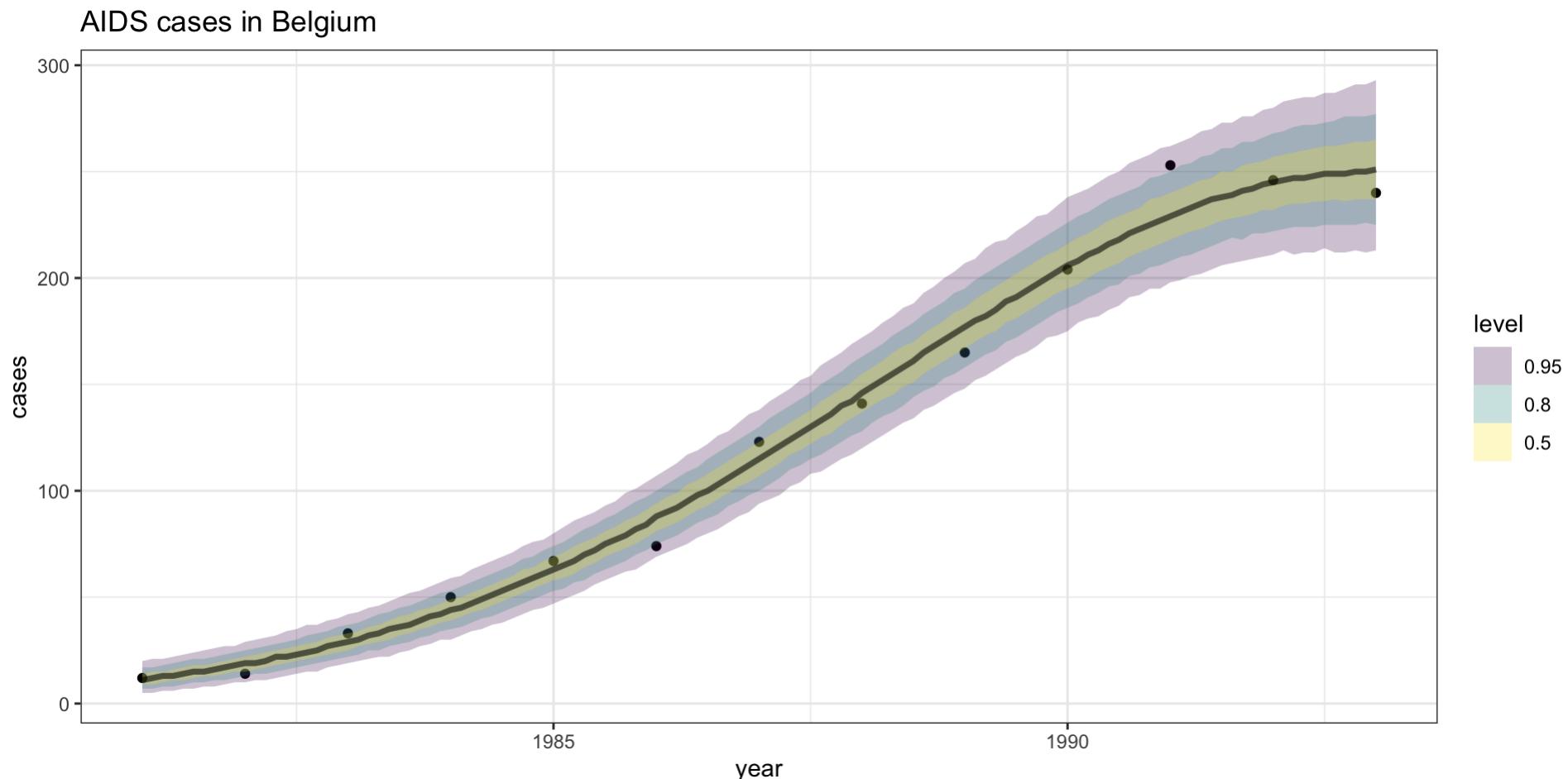
PP Checks

```
1 brms:::pp_check(g2_bayes, ndraws=100)
```



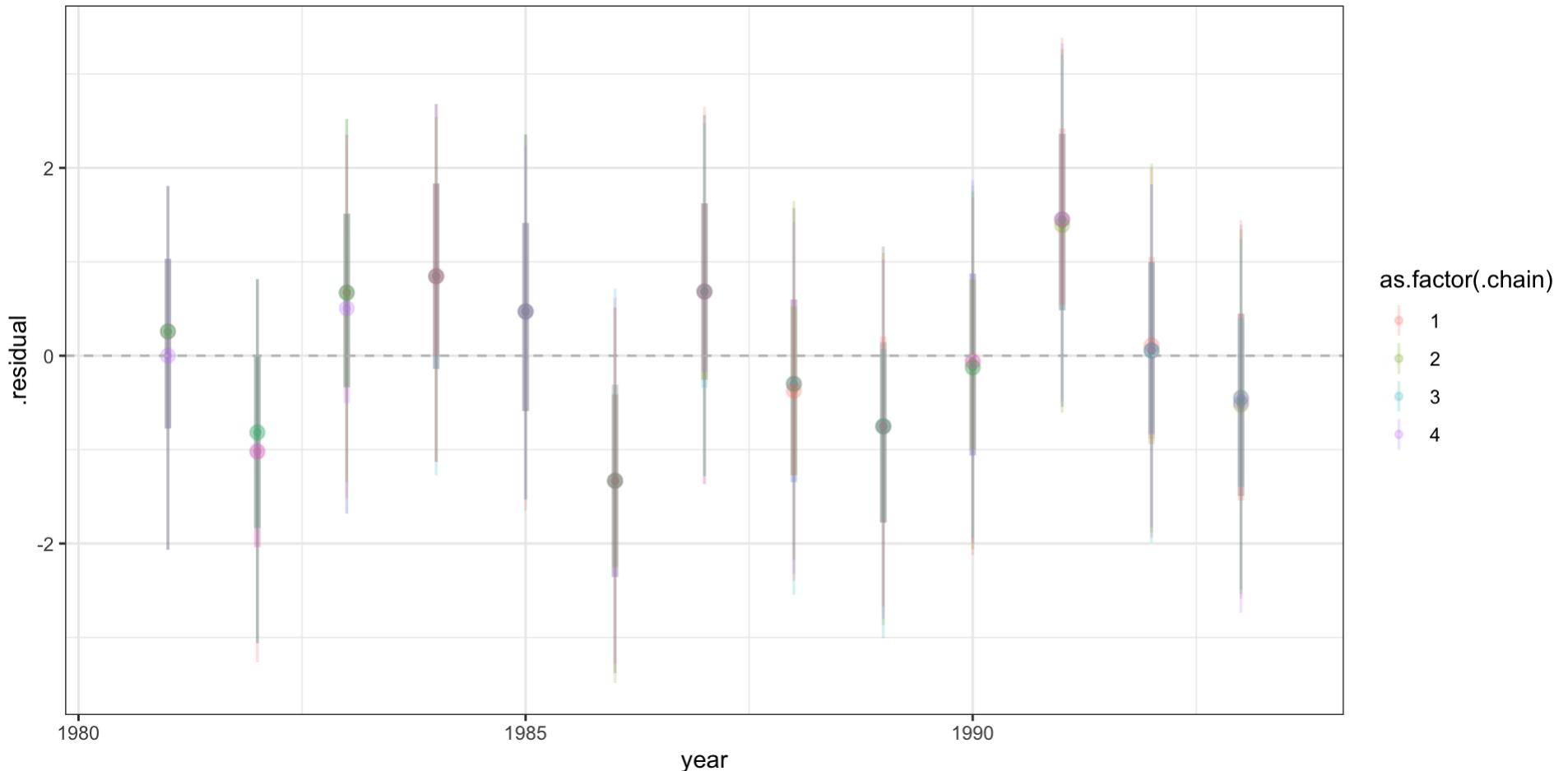
Model fit - Y CI

```
1 aids_base +  
2   tidybayes::stat_lineribbon(  
3     data = g2_bayes_pred, aes(y=.prediction), alpha=0.25  
4   )
```



Residuals (pearson)

```
1 residual_draws_fix(g2_bayes, newdata = aids, type = "pearson") |>
2   ggplot(aes(y = .residual, x = year, color=as.factor(.chain), group=.chain)) +
3     tidybayes::stat_pointinterval(alpha=0.2) +
4     geom_hline(yintercept = 0, color='grey', linetype=2)
```



Model performance - rmse, crps

```
1 predicted_draws_fix(g2_bayes, newdata = aids) |>
2   group_by(.chain, .row) |>
3   summarize(
4     rmse = yardstick::rmse_vec(cases, .prediction),
5     crps = 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  14.4  5.25
2      2  14.2  5.08
3      3  14.2  5.17
4      4  14.4  5.12
```

Model performance - emp coverage

```
1 predicted_draws_fix(g2_bayes, newdata = aids) |>
2   group_by(.row, cases) |>
3   tidybayes::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     0.462
2 0.9     1
3 0.95    0.929
```

Logistic regression

Logistic regression as a GLM

This is another case of a generalized linear model, specifically where the outcome is 0-1 data (i.e. Bernoulli draws),

$$Y_i \sim \text{Bern}(p_i)$$
$$\text{logit } E(Y_i | X_{i \cdot}) = \text{logit}(p_i) = X_{i \cdot} \beta$$

$$E(Y_i) = p_i$$
$$\text{Var}(Y_i) = p_i(1 - p_i)$$

$$\text{logit}(p_i) = \log \frac{p_i}{1 - p_i}$$
$$\text{logit}^{-1}(x) = \frac{\exp(x)}{1 + \exp(x)} = \frac{1}{1 + \exp(-x)}$$

Background

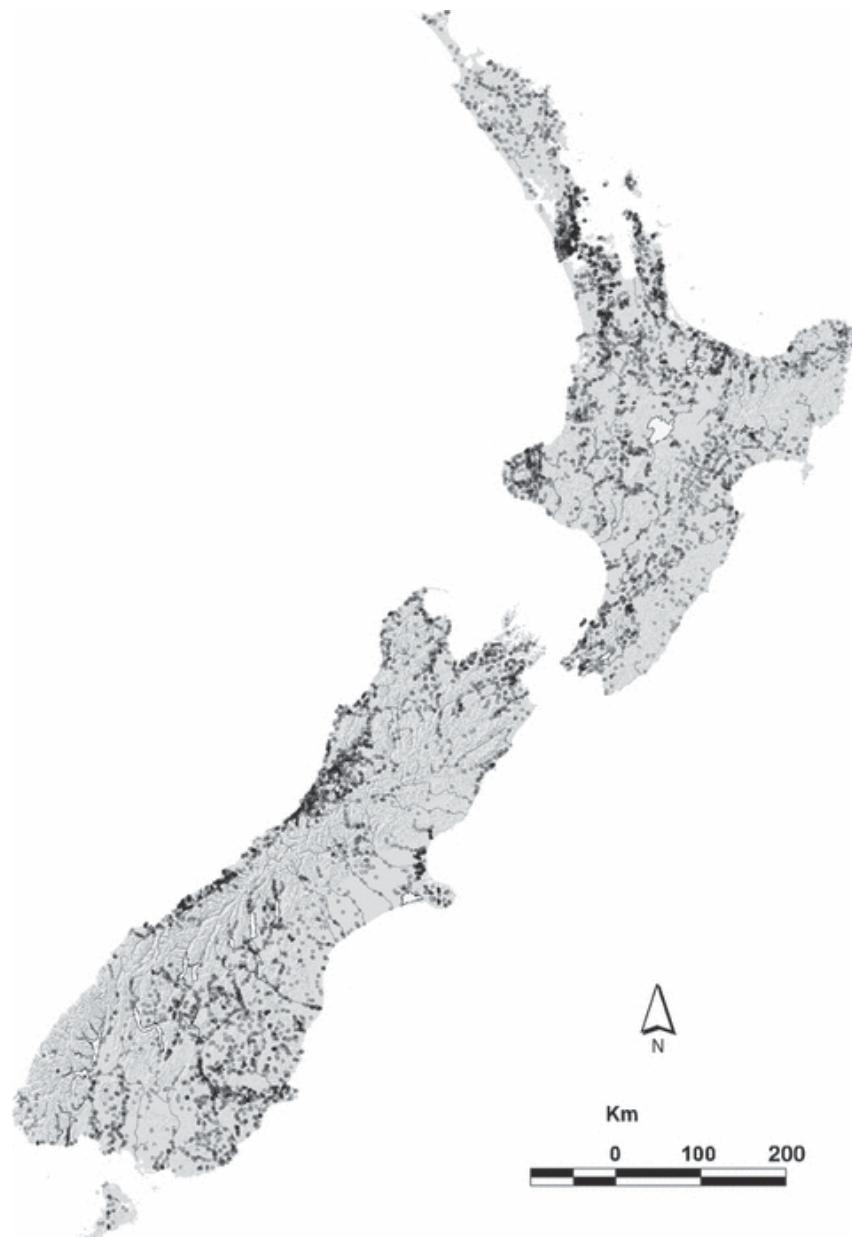
Next we'll be looking at data on the presence and absence of the short-finned eel (*Anguilla australis*) at a number of sites in New Zealand.

These data come from

- Leathwick, J. R., Elith, J., Chadderton, W. L., Rowe, D. and Hastie, T. (2008), Dispersal, disturbance and the contrasting biogeographies of New Zealand's diadromous and non-diadromous fish species. *Journal of Biogeography*, 35: 1481–1497.



Species Distribution



Sta 344/644 - Fall 2023

Codebook:

- presence - presence (**1**) or absence (**0**) of Anguilla australis at the sampling location
- SegSumT - Summer air temperature (degrees C)
- DSDist - Distance to coast (km)
- DSMaxSlope - Maximum downstream slope (degrees)
- USRainDays - days per month with rain greater than 25 mm
- USSlope - average slope in the upstream catchment (degrees)
- USNative - area with indigenous forest (proportion)
- DSDam - Presence of known downstream obstructions, mostly dams
- Method - fishing method (**electric**, **net**, **spot**, **trap**, or **mixture**)
- LocSed - weighted average of proportional cover of bed sediment
 - 1. mud
 - 2. sand
 - 3. fine gravel
 - 4. coarse gravel
 - 5. cobble
 - 6. boulder
 - 7. bedrock

Data

```
1 ( anguilla = readRDS("data/anguilla.rds") )
```

	presence	SegSumT	DSDist	DSMaxSlope	USRainDays	USSlope	USNative	DSDam
1	0	16.0	50.2000	0.57	2.470	9.8	0.81	0
2	1	18.7	132.5300	1.15	1.153	8.3	0.34	0
3	0	18.3	107.4400	0.57	0.847	0.4	0.00	0
4	0	16.7	166.8200	1.72	0.210	0.4	0.22	1
5	1	17.2	3.9500	1.15	1.980	21.9	0.96	0
6	0	15.1	11.1700	1.72	3.300	25.7	1.00	0
7	0	12.7	42.3700	2.86	0.430	9.6	0.09	0
10	1	18.2	94.3960	3.43	0.847	20.5	0.92	0
11	0	14.9	45.7370	2.29	2.249	3.9	0.38	0
12	1	18.3	91.8920	1.72	0.861	6.7	0.58	1
13	1	17.1	6.8000	0.52	0.620	0.7	0.00	0
14	0	13.4	190.3900	3.43	0.770	20.1	0.99	0
15	0	13.1	224.4400	6.84	0.290	9.8	0.98	0
17	0	17.0	1.3800	9.09	1.980	10.5	1.00	0
18	1	18.5	1.8685	1.72	0.847	11.8	0.30	0
19	0	15.7	13.8800	0.57	3.300	2.7	0.29	0
20	0	16.6	28.0200	1.15	0.860	19.0	0.57	0

See <https://github.com/sta344-644-fa23/sta344-644-fa23.github.io/tree/main/static/slides/data> for all
Sta 344/644 - Fall 2023

Test / train split

```
1 set.seed(20220908)
2 part = rsample::initial_split(anguilla, prop = 3/4)
3
4 anguilla_train = rsample::training(part)
5 anguilla_test  = rsample::testing(part)
```

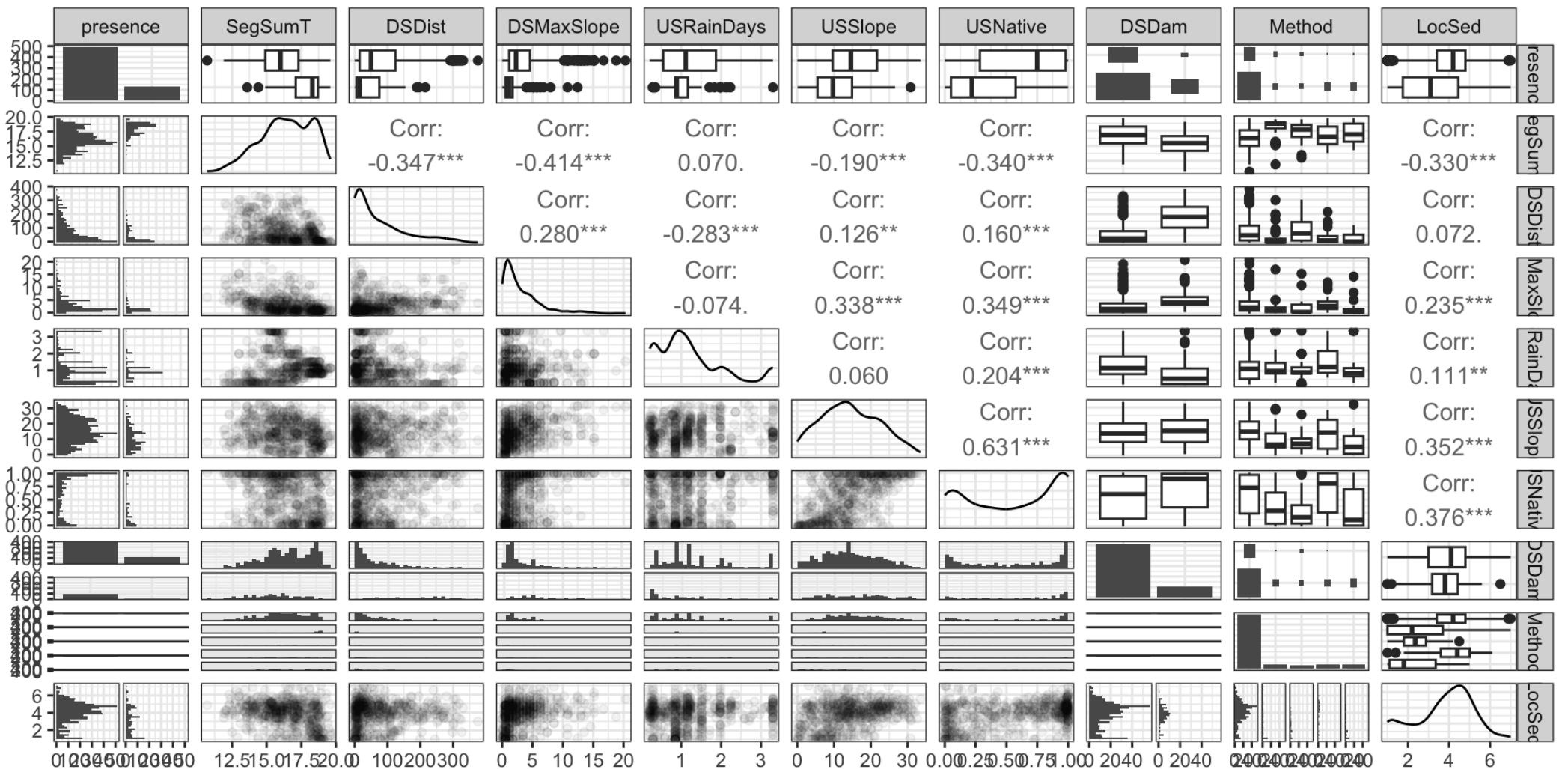
```
1 dim(anguilla_train)
```

```
[1] 618 10
```

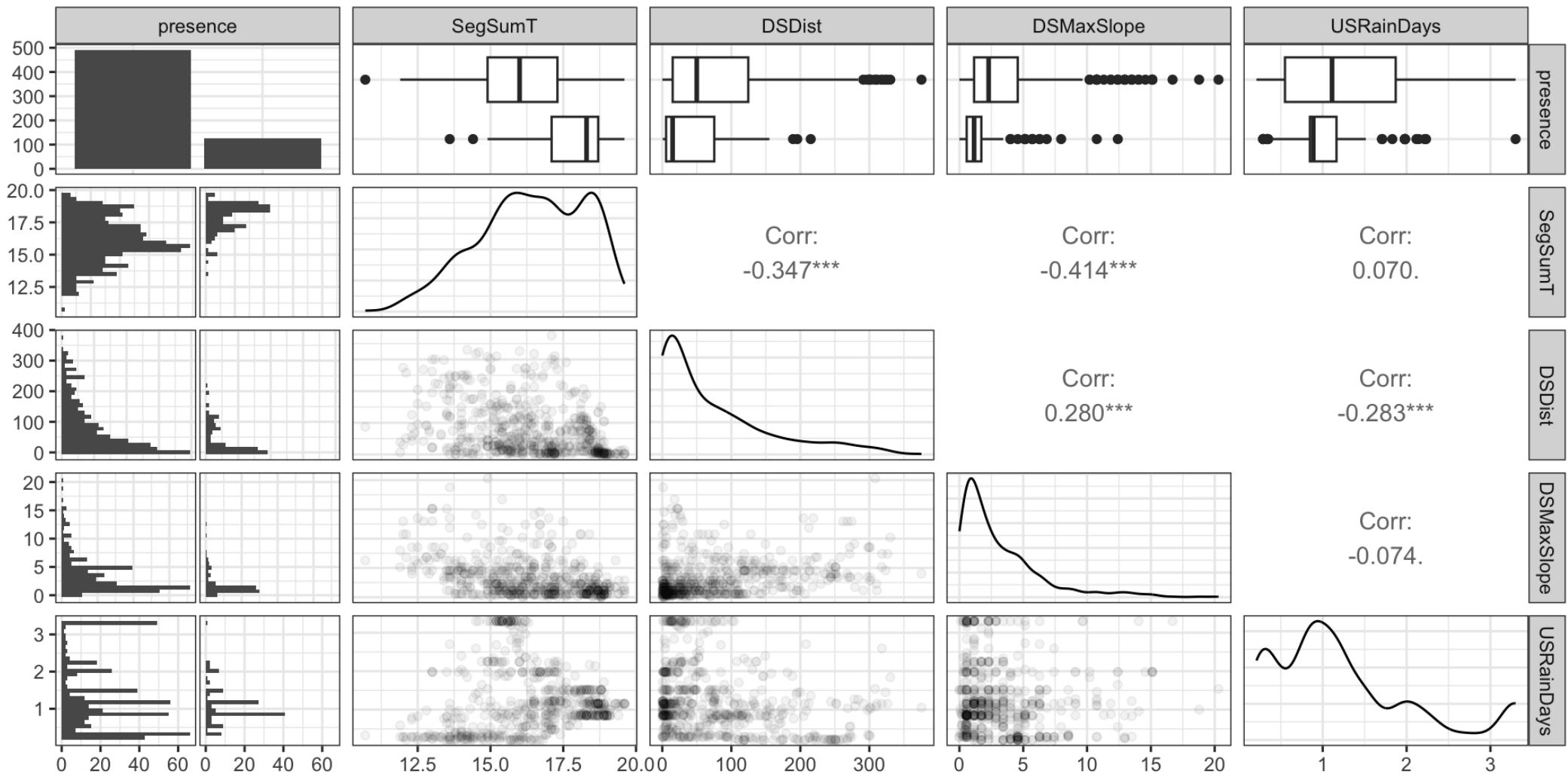
```
1 dim(anguilla_test)
```

```
[1] 206 10
```

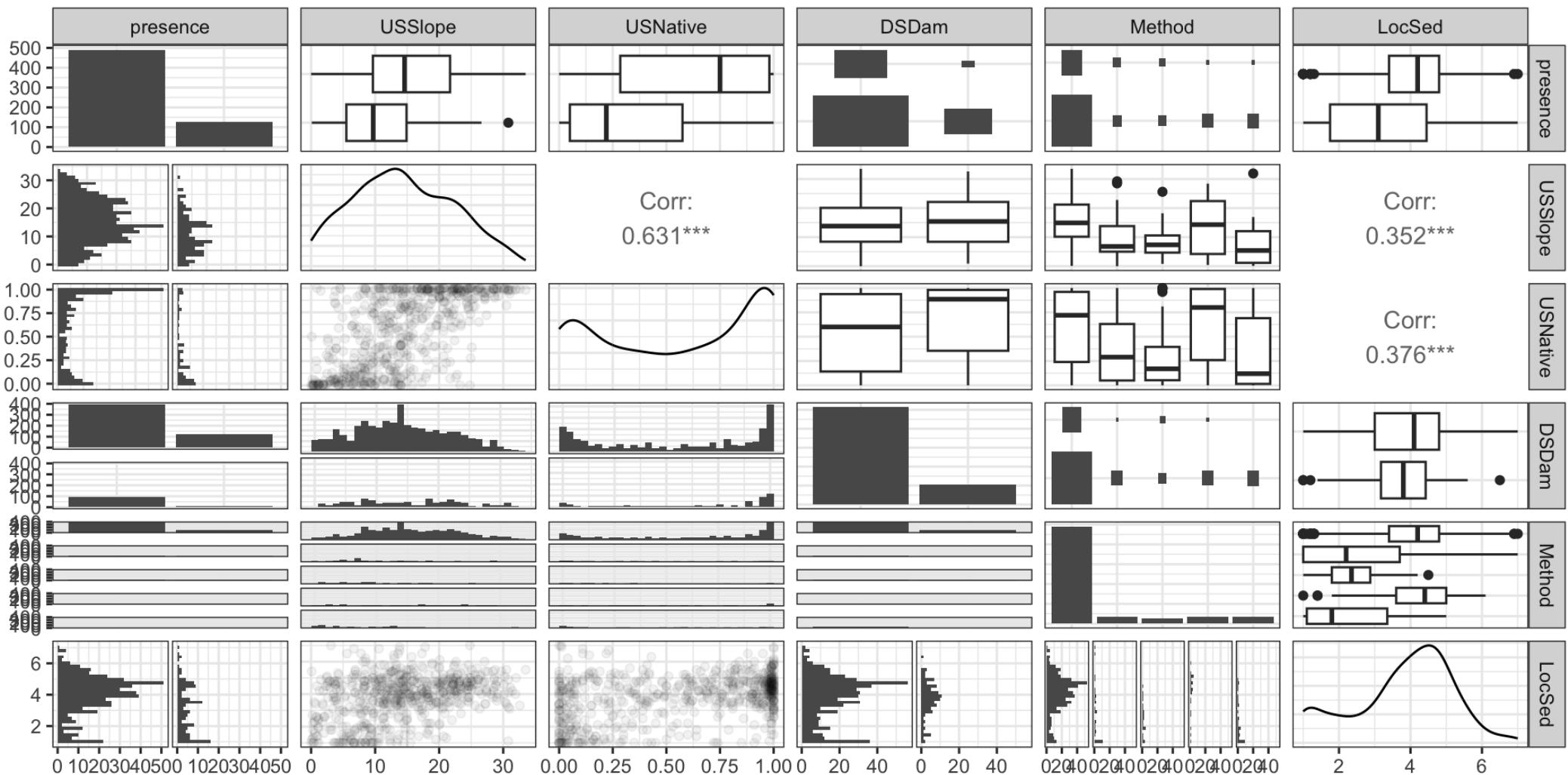
EDA



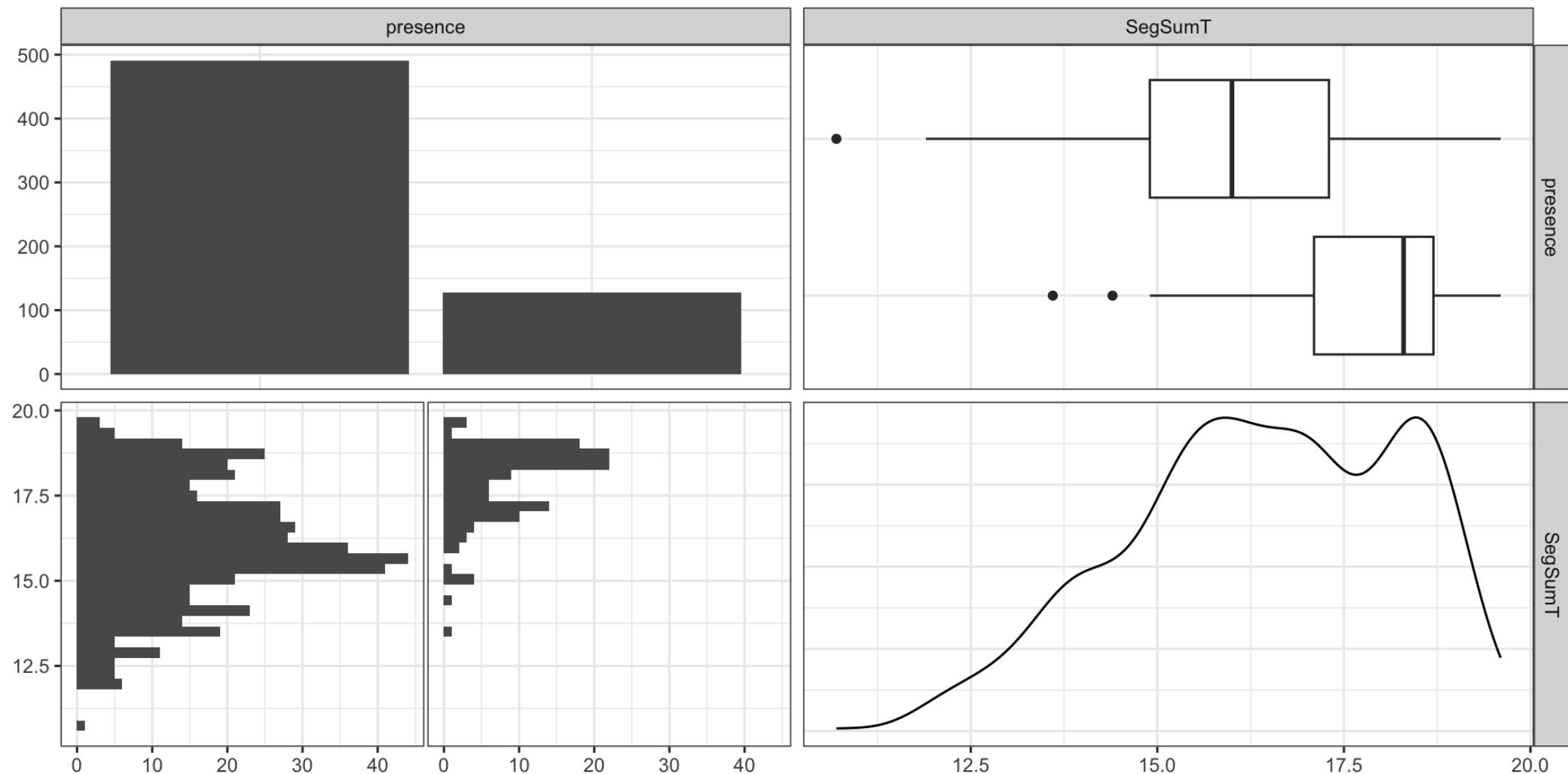
EDA (part 1)



EDA (part 2)



EDA (part 3)



Basic Model

Model

```
1 g = glm(presence~SegSumT, family=binomial, data=anguilla_train)
2 summary(g)
```

Call:

```
glm(formula = presence ~ SegSumT, family = binomial, data = anguilla_train)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-15.02201	1.53770	-9.769	<2e-16 ***
SegSumT	0.80047	0.08726	9.173	<2e-16 ***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 627.81 on 617 degrees of freedom
Residual deviance: 501.93 on 616 degrees of freedom
AIC: 505.93

Number of Fisher Scoring iterations: 5

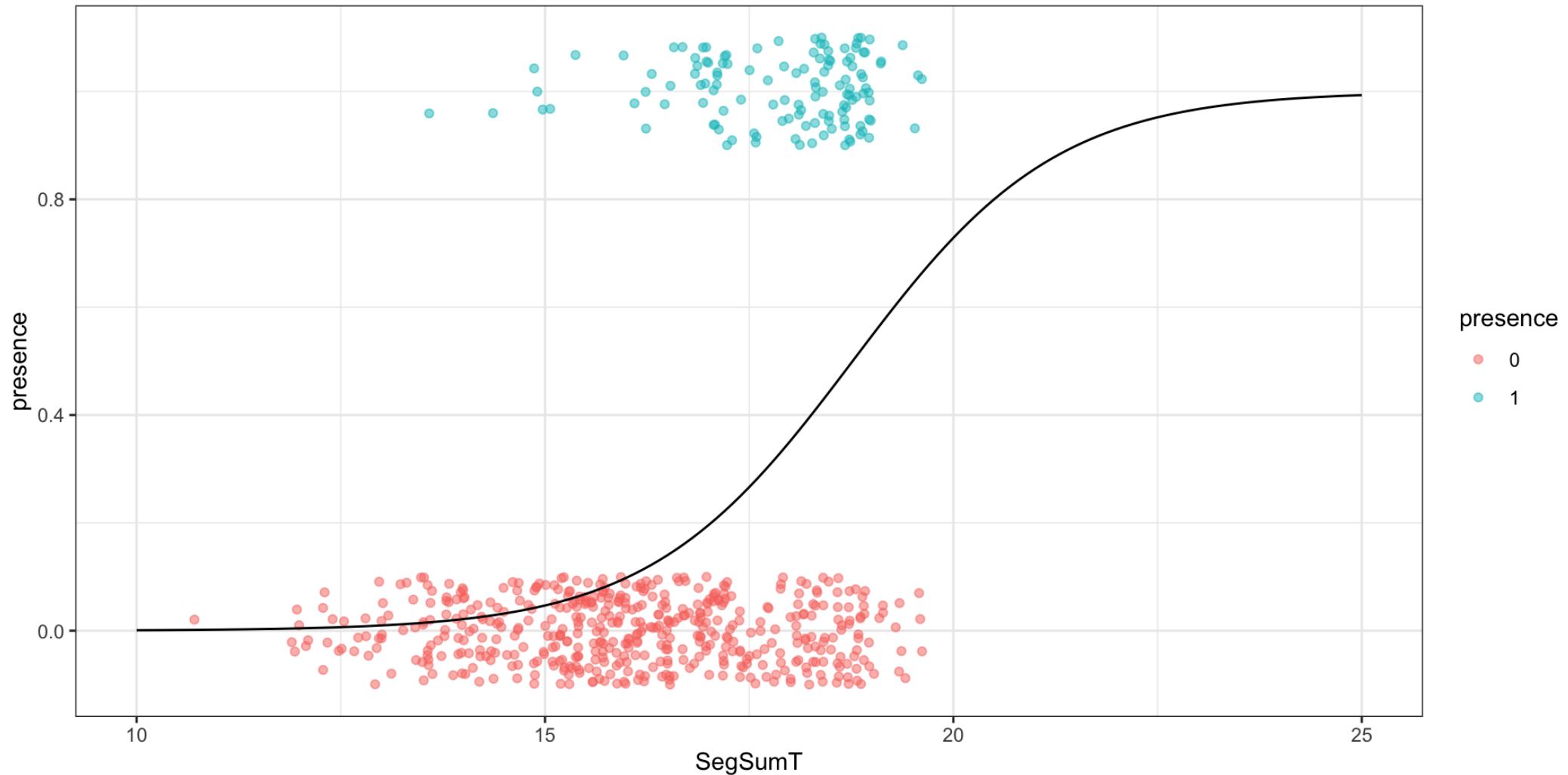
Fit

```
1 ( g_pred = broom::augment(g, type.predict = "response") )
```

```
# A tibble: 618 × 8
```

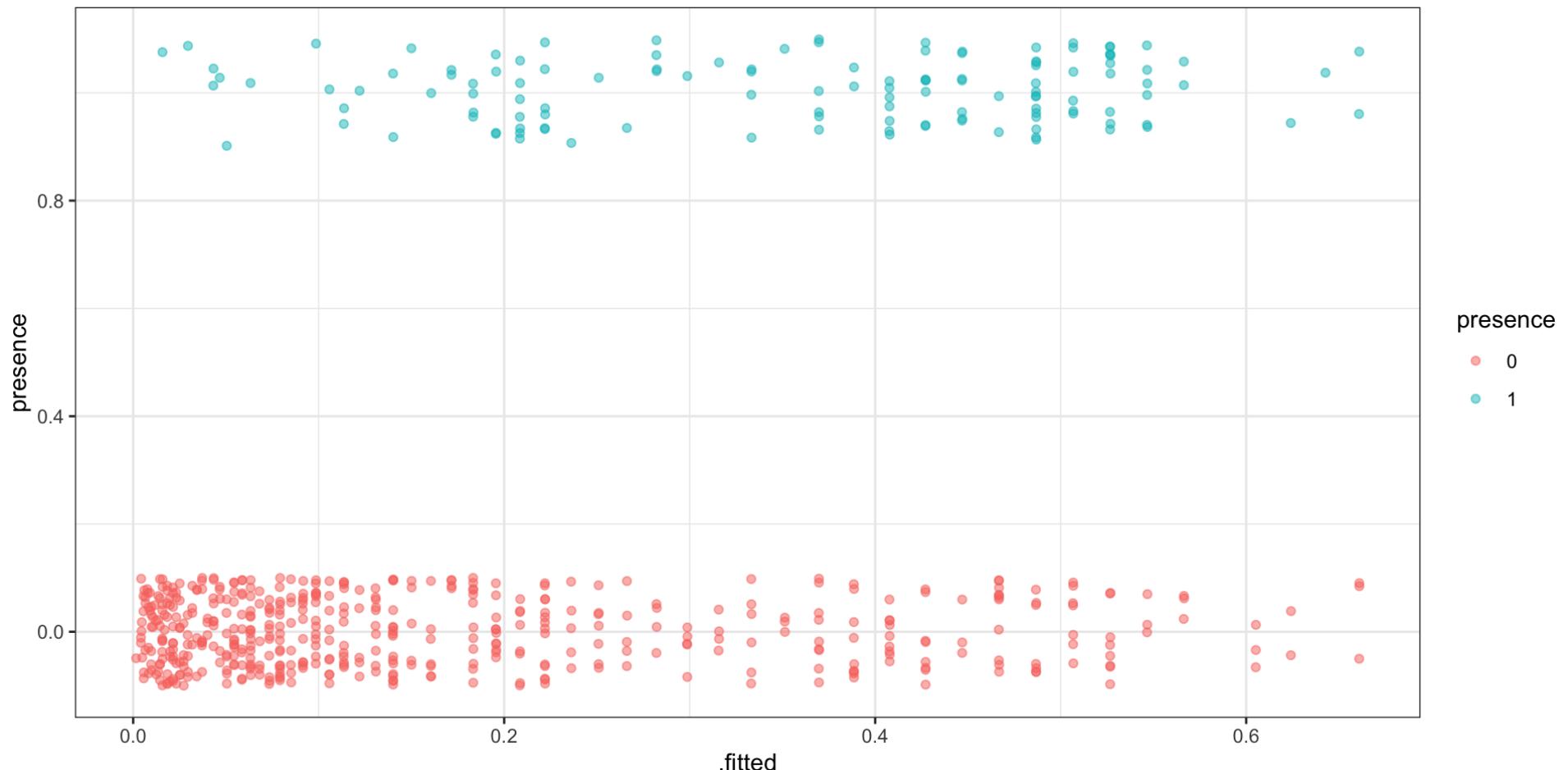
	presence	SegSumT	.fitted	.resid	.hat	.sigma	.cooksd	.std.resid
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	16.4	0.131	-0.529	0.00260	0.903	0.000197	-0.530
2	1	17.1	0.209	1.77	0.00232	0.901	0.00443	1.77
3	0	14	0.0216	-0.209	0.00231	0.903	0.0000256	-0.209
4	0	18.2	0.389	-0.992	0.00364	0.903	0.00117	-0.994
5	0	15.6	0.0735	-0.391	0.00286	0.903	0.000114	-0.391
6	0	18.3	0.408	-1.02	0.00395	0.902	0.00137	-1.03
7	0	18.5	0.447	-1.09	0.00466	0.902	0.00190	-1.09
8	0	16.2	0.114	-0.491	0.00270	0.903	0.000174	-0.492
9	0	18	0.351	-0.930	0.00313	0.903	0.000853	-0.932
10	1	17.3	0.236	1.70	0.00233	0.901	0.00379	1.70
# i 608 more rows								

Visually



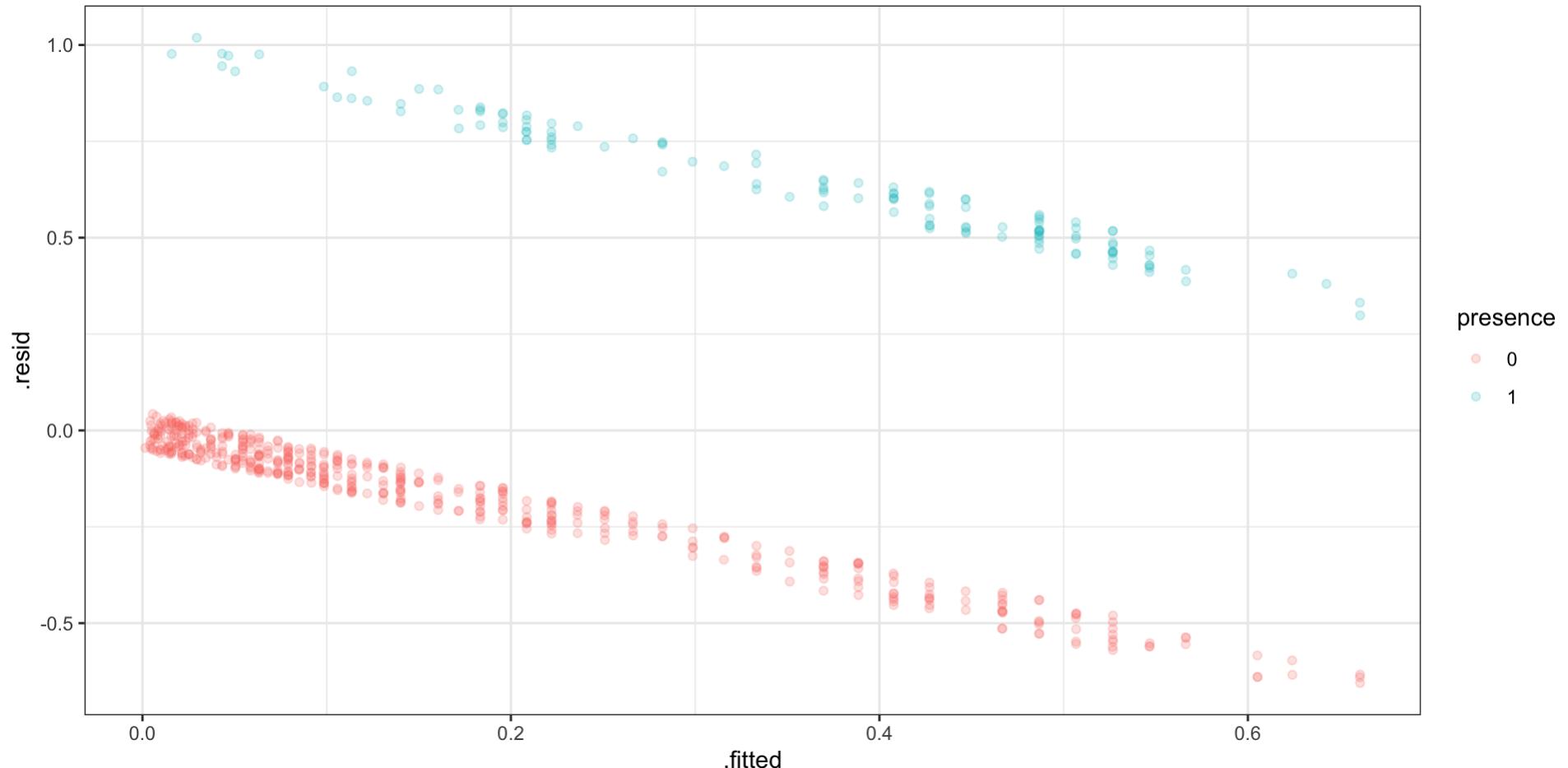
Separation

```
1 ggplot(g_pred, aes(x=.fitted, y=presence, color=as.factor(presence))) +  
2   geom_jitter(height=0.1, alpha=0.5) +  
3   labs(color="presence")
```



Standard Residuals

```
1 g_std = broom::augment(g, type.predict = "response") |>  
2   mutate(.resid = presence - .fitted)
```



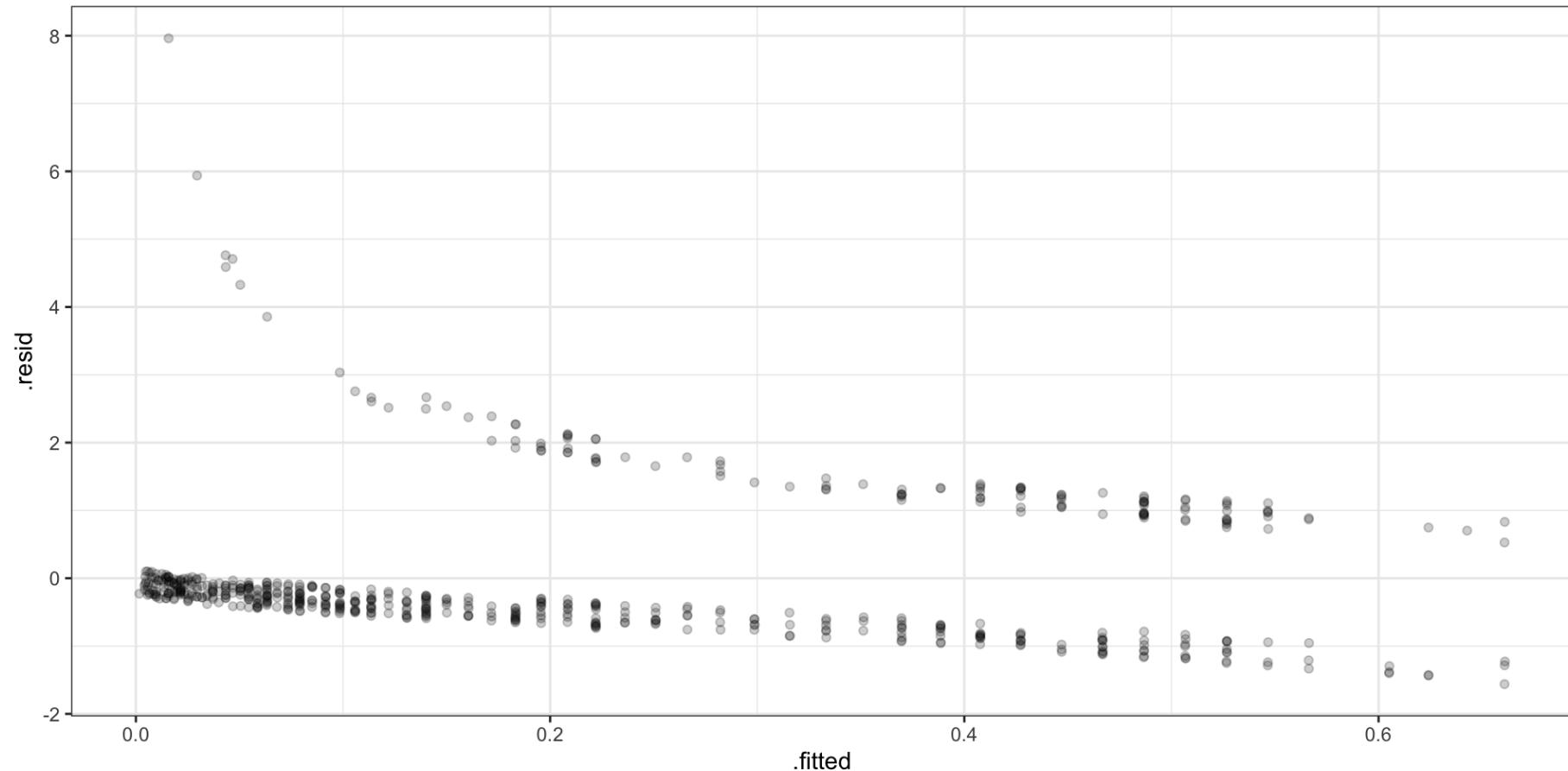
Binned Residuals

```
1 bin_width = 0.05
2 g_std_bin = g_std |>
3   mutate(bin = .fitted - (.fitted %% bin_width) + bin_width/2) |>
4   group_by(bin) |>
5   summarize(.resid_bin = mean(presence - .fitted))
```

Pearson Residuals

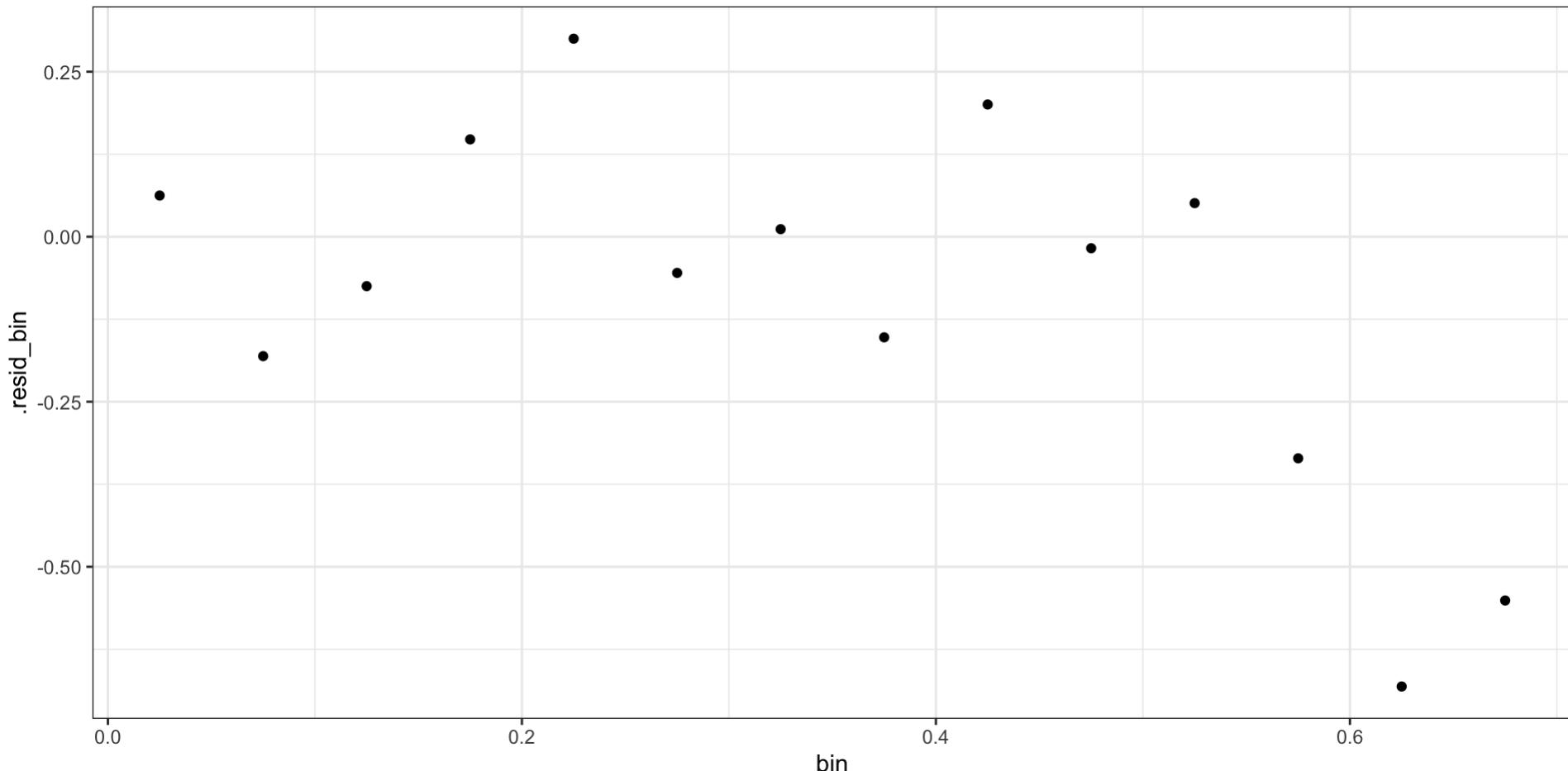
$$r_i = \frac{Y_i - E(Y_i)}{\sqrt{\text{Var}(Y_i)}} = \frac{Y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}}$$

```
1 g_pearson = broom::augment(  
2   g, type.predict="response",  
3   type.residuals="pearson"  
4 )
```



Binned Pearson Residuals

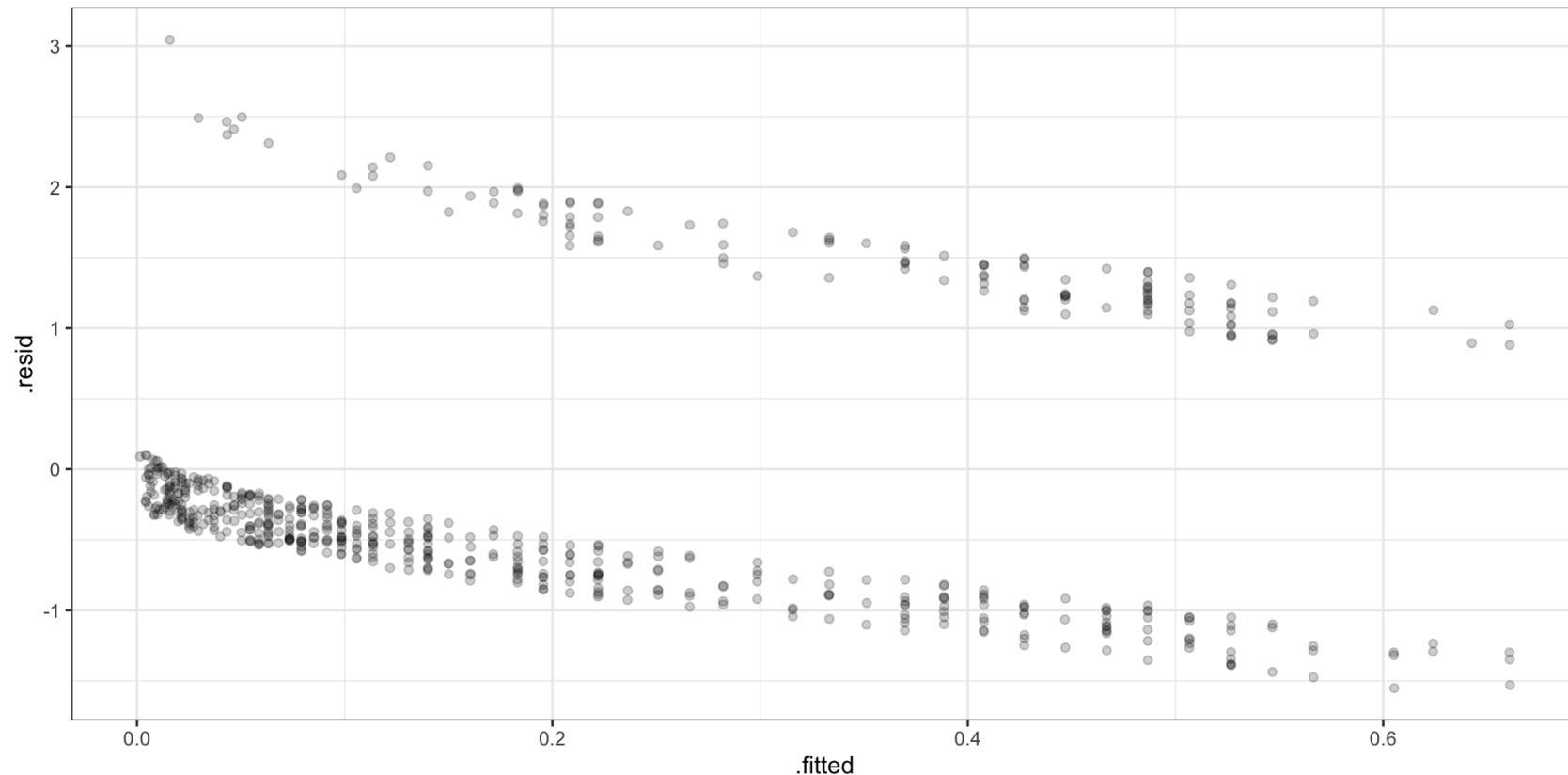
```
1 g_pearson_bin = g_pearson |>
2   mutate(bin = .fitted - (.fitted %% bin_width) + bin_width/2) |>
3   group_by(bin) |>
4   summarize(.resid_bin = mean(.resid))
```



Deviance Residuals

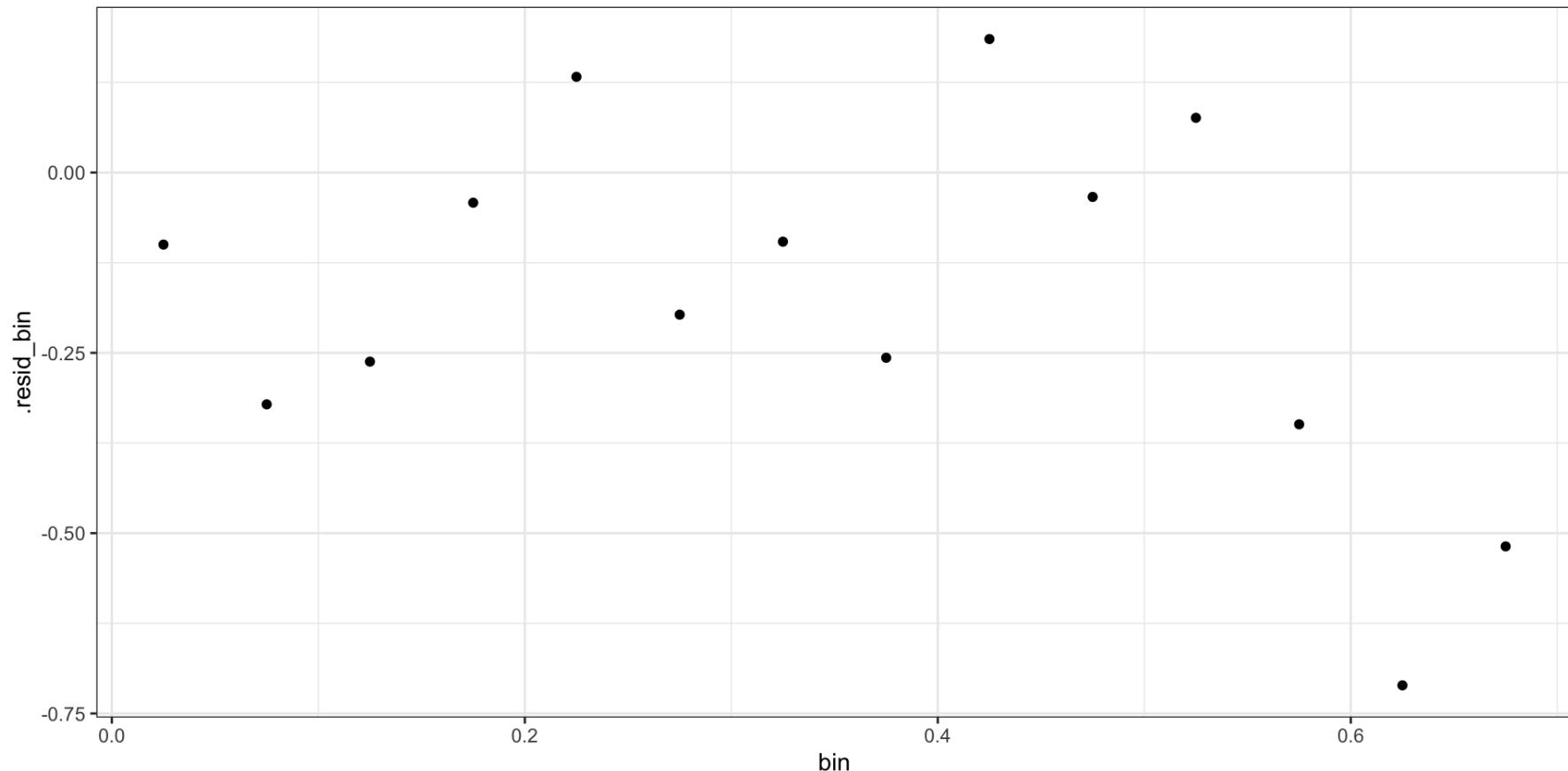
$$d_i = \text{sign}(Y_i - \hat{p}_i) \times \sqrt{-2 \left(Y_i \log \hat{p}_i + (1 - Y_i) \log(1 - \hat{p}_i) \right)}$$

```
1 g_deviance = broom::augment(  
2   g, type.predict = "response",  
3   type.residuals = "deviance"  
4 )
```



Binned Deviance Residuals

```
1 g_deviance_bin = g_deviance |>
2   mutate(bin = .fitted - (.fitted %% bin_width) + bin_width/2) |>
3   group_by(bin) |>
4   summarize(.resid_bin = mean(.resid))
```



Checking Deviance

```
1 g
```

```
Call: glm(formula = presence ~ SegSumT, family = binomial, data = anguilla_train)
```

Coefficients:

(Intercept)	SegSumT
-15.0220	0.8005

Degrees of Freedom: 617 Total (i.e. Null); 616 Residual

Null Deviance: 627.8

Residual Deviance: 501.9 AIC: 505.9

```
1 summarize(g_deviance, sum(.resid^2))
```

```
# A tibble: 1 × 1
```

```
`sum(.resid^2)`  
  <dbl>  
1      502.
```

Full Model

Model

```
1 f = glm(presence~, family=binomial, data=anguilla_train)
2 summary(f)
```

Call:

```
glm(formula = presence ~ ., family = binomial, data = anguilla_train)
```

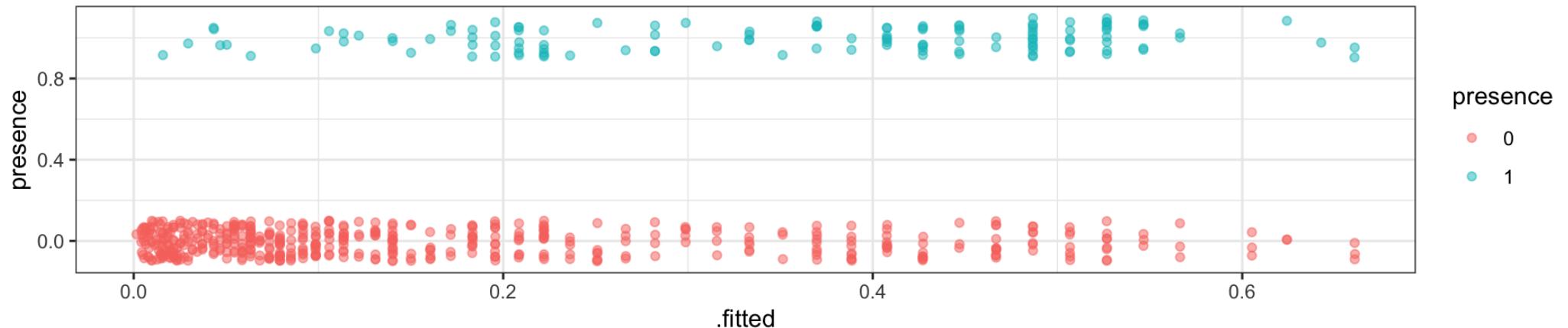
Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-9.352885	1.761202	-5.311	1.09e-07 ***
SegSumT	0.654186	0.096921	6.750	1.48e-11 ***
DSDist	-0.004837	0.002302	-2.102	0.03559 *
DSMaxSlope	-0.030776	0.061995	-0.496	0.61959
USRainDays	-0.710920	0.225814	-3.148	0.00164 **
USSlope	-0.069814	0.025443	-2.744	0.00607 **
USNative	-0.456598	0.455261	-1.003	0.31589
DSDam	-1.095360	0.516960	-2.119	0.03410 *
Methodmixture	-0.430351	0.475411	-0.905	0.36535
Methodnet	-0.066214	0.559162	-0.118	0.90574
Methodspo	-1.583905	0.701902	-2.257	0.02403 *
Methodtrap	-2.958398	0.688146	-4.299	1.72e-05 ***
LocSed	-0.140495	0.096849	-1.451	0.14688

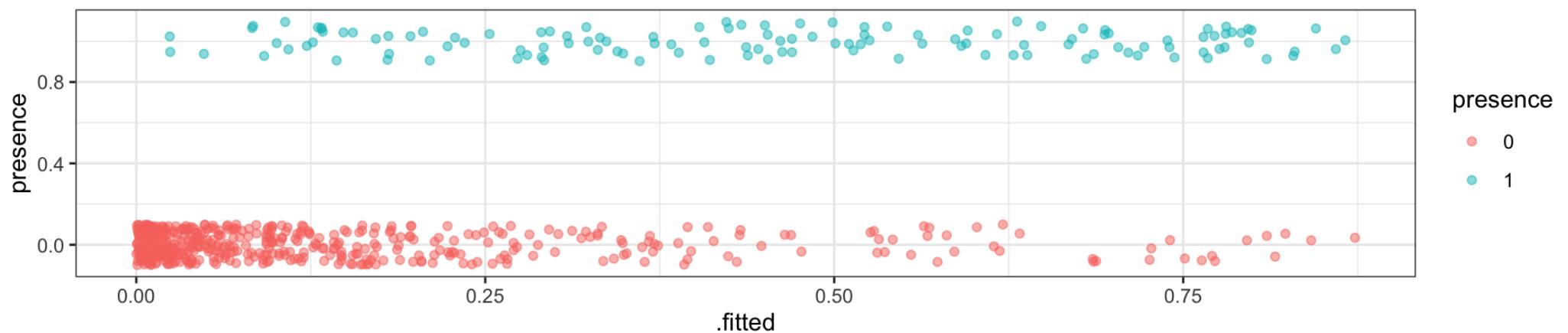
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Separation

SegSumT Model

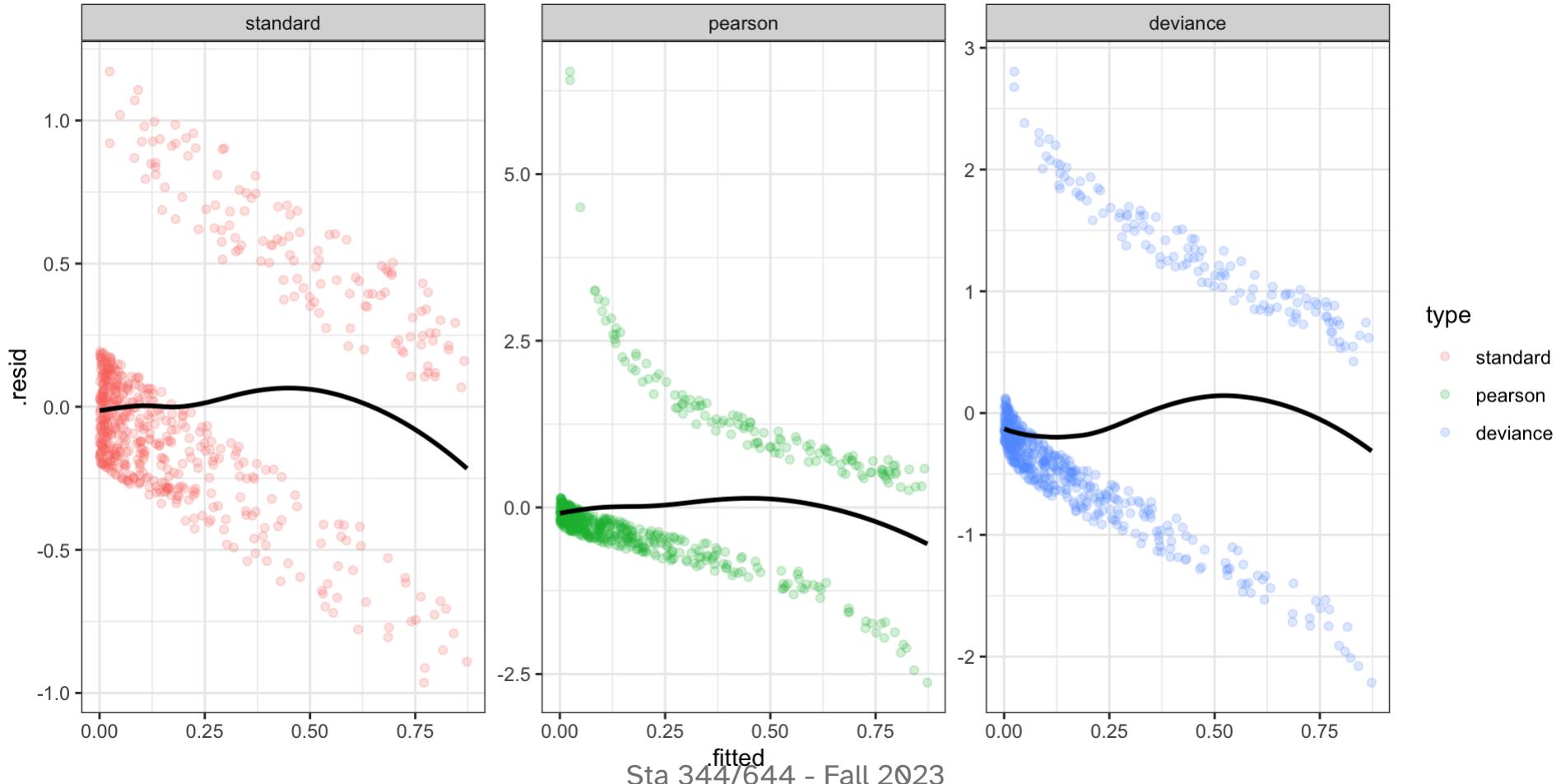


Full Model



Residuals vs fitted

```
1 f_resid |>
2   ggplot(aes(x=.fitted, y=.resid, color=type)) +
3   geom_jitter(height=0.2, alpha=0.2) +
4   facet_wrap(~type, ncol=3, scale="free_y") +
5   geom_smooth(se = FALSE, color="black")
```



Residuals (binned) vs fitted

```
1 f_resid_bin |>
2   mutate(type = as_factor(type)) |>
3   ggplot(aes(x=bin, y=.resid_bin, color=type)) +
4   geom_point() +
5   facet_wrap(~type, ncol=3, scales = "free_y")
```

