

Practical 6: Mixed effects models

Benjamin Rosenbaum

Table of contents

Random intercepts ANOVA	1
Random intercepts ANCOVA	11
Exercise	17
Without temperature	18
With temperature	21

We perform some classical linear modeling with a random grouping factor. Predictions etc can be computed with random effects, or fixed effects only. Model checks, model comparisons, hypotheses, etc are the same as with linear models.

```
rm(list=ls())
library("brms")
library("ggplot2")
library("performance")
library("ecostats")
library("cowplot")
try(dev.off())
```

Random intercepts ANOVA

From Warton, D. (2022) Eco-Stats: Data Analysis in Ecology

Counted number of invertebrate species in samples from different estuaries. Estuaries are either pristine or modified. Data are grouped, originate from 7 estuaries with multiple samples per estuary.

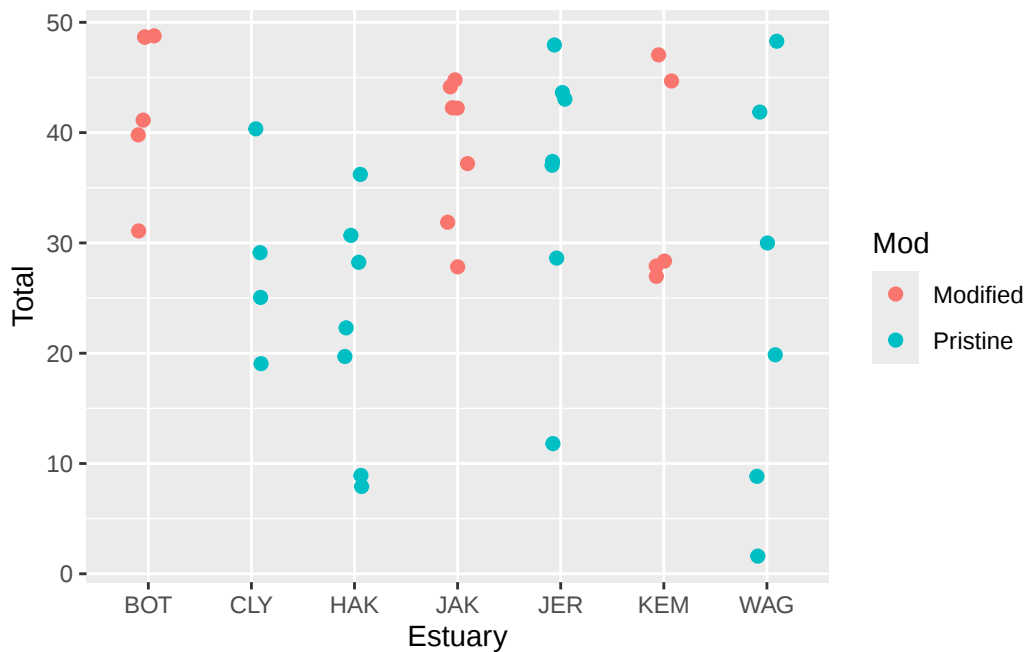
Question: Is there a difference in species richness between modified and pristine estuaries?

The random grouping factor `Estuary` is nested in `Mod`, because each `Estuary` is either modified or pristine. We could model this as `(1|Mod/Estuary)`, but since `Mod` only has two levels, of which we are interested in species difference, we model `Mod` as fixed effect with random effects for `Estuary`.

```
data("estuaries")
table(estuaries$Mod, estuaries$Estuary)
```

	BOT	CLY	HAK	JAK	JER	KEM	WAG
Modified	6	0	0	7	0	5	0
Pristine	0	4	7	0	7	0	6

```
ggplot(estuaries, aes(Estuary, Total, col=Mod)) + geom_jitter(width=0.1, size=2)
```



Remove NAs. `brms` automatically removes them in model fitting. But when different predictors have NAs for different observations, models that do not have the same predictors might end up with different datasets. In this case, model comparison would not be possible (requires identical datasets).

```
ID.complete = complete.cases(estuaries[, c("Total", "Mod", "Estuary", "Temperature")])
estuaries = estuaries[ID.complete, ]
```

Deterministic part: $\mu_i = b(\text{Mod}_i) + \delta(\text{Estuary}_i)$
 Stochastic part: $\text{Total}_i \sim \text{Normal}(\mu_i, \sigma)$
 Hierarchical part: $\delta_j \sim \text{Normal}(0, \sigma_{\text{Est}})$

$i = 1 \dots N$, $\text{Mod} = 1, 2$, $\text{Estuary} = 1 \dots 7$

```
default_prior(Total ~ Mod+(1|Estuary), data=estuaries)
```

	prior	class	coef	group	resp	dpar	nlpar	lb	ub	source
	(flat)	b								default
	(flat)	b	ModPristine							(vectorized)
student_t(3, 32, 14.8)		Intercept								default
student_t(3, 0, 14.8)		sd						0		default
student_t(3, 0, 14.8)		sd	Estuary					0		(vectorized)
student_t(3, 0, 14.8)		sd	Intercept	Estuary				0		(vectorized)
student_t(3, 0, 14.8)		sigma						0		default

We only need to specify a prior for the fixed effect (which is dummy-coded), the random effects part is already taken care of by default.

```
fit.est.1 = brm(Total ~ Mod+(1|Estuary),
  prior = prior(normal(0,10), class=b),
  data = estuaries)
```

Check convergence. There were divergent transitions, but just 2 which is no big deal.

Note that `summary()` and `plot()` do not display the actual random effects δ_j , only their sdev $\sigma_{\text{Estuary}} = \text{sd}(\text{Intercept})$

```
summary(fit.est.1, prior=TRUE)
```

Warning: There were 2 divergent transitions after warmup. Increasing adapt_delta above 0.8 may

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: Total ~ Mod + (1 | Estuary)
Data: estuaries (Number of observations: 41)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Priors:

```

b ~ normal(0, 10)
Intercept ~ student_t(3, 32, 14.8)
<lower=0> sd ~ student_t(3, 0, 14.8)
<lower=0> sigma ~ student_t(3, 0, 14.8)

```

Multilevel Hyperparameters:

~Estuary (Number of levels: 7)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	5.11	3.35	0.35	13.39	1.00	928	1367

Regression Coefficients:

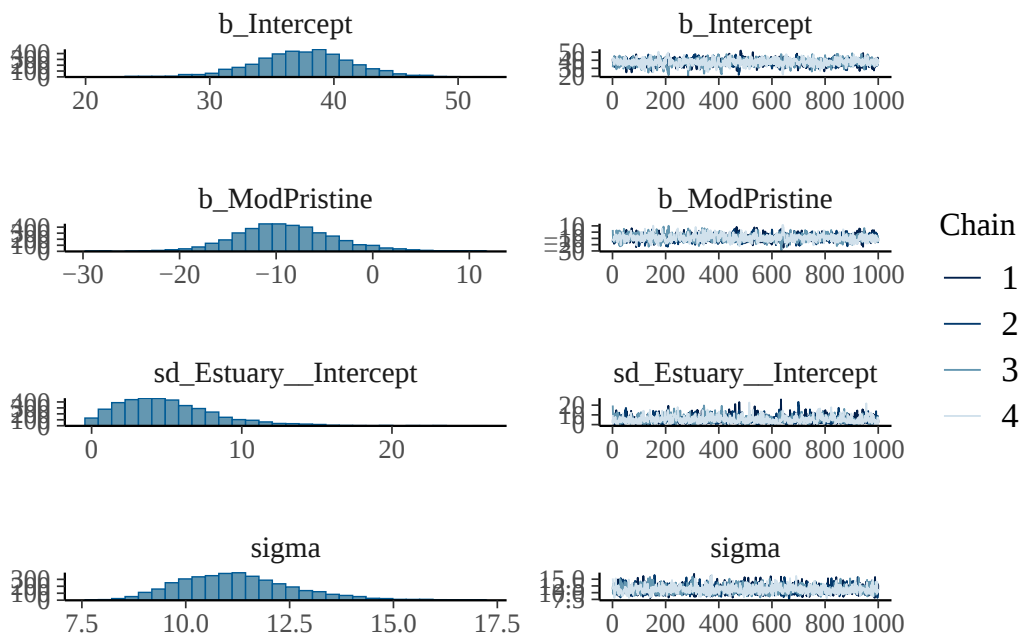
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	37.45	4.00	28.90	45.02	1.00	1485	1511
ModPristine	-8.96	5.11	-18.45	1.91	1.00	1475	1761

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	11.24	1.39	8.90	14.29	1.00	2437	2535

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).

```
plot(fit.est.1)
```



If needed, fixed and random effects can be extracted with these functions:

```
fixef(fit.est.1)
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	37.44954	4.003155	28.89980	45.022357
ModPristine	-8.96339	5.108531	-18.45213	1.912913

```
ranef(fit.est.1)
```

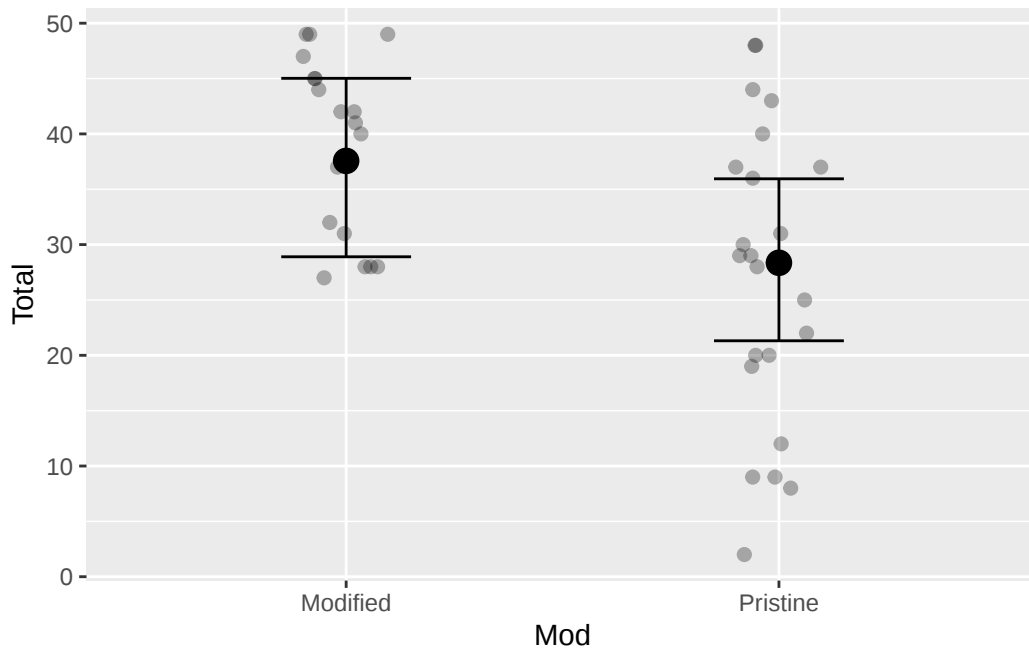
\$Estuary

, , Intercept

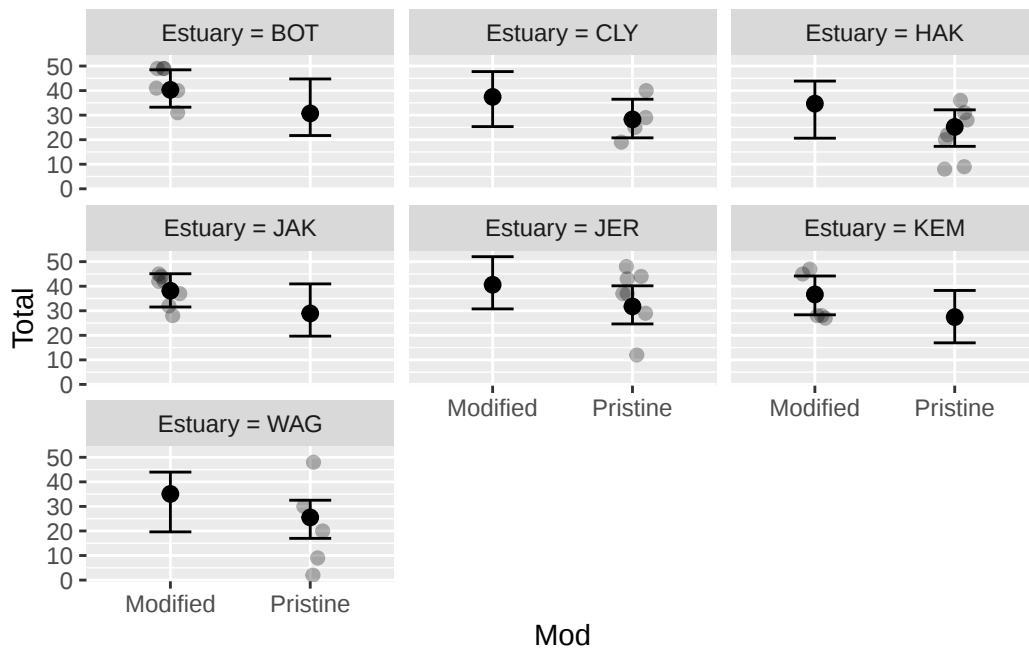
	Estimate	Est.Error	Q2.5	Q97.5
BOT	2.9199439	4.401671	-4.001824	13.455701
CLY	-0.1883263	4.088938	-8.995330	7.894171
HAK	-3.4409207	4.343956	-13.633125	3.606532
JAK	0.7751533	4.044399	-7.270259	9.503657
JER	3.4372259	4.163375	-3.426794	13.033421
KEM	-0.9766246	4.088095	-9.849672	7.344053
WAG	-3.1842804	4.438557	-13.908306	3.926569

Conditional effects plots of fixed effects only (default, “population-level”), but by specifying `re_formula=NULL` and `conditions`, we can plot on group-level, too. We even get predictions (on group-level) of what would happen if we applied the modification to the estuary instead of keeping it pristine.

```
plot(conditional_effects(fit.est.1),  
     points=T,  
     point_args=c(alpha=0.3, width=0.1))
```



```
plot(conditional_effects(fit.est.1,
                        re_formula = NULL,
                        conditions = make_conditions(fit.est.1, var=c("Estuary"))),
     points=T,
     point_args=c(alpha=0.3, width=0.1, size=2))
```



When computing predictions with `fitted` (deterministic part) or `predict` (deterministic & stochastic part), we can choose to include random effects (default), or fixed effects only with `re_formula=NA`.

In general, `re_formula=NULL` **enforces** random effects, `re_formula=NA` **omits** random effects.

```
fitted(fit.est.1) |> head()
```

	Estimate	Est.Error	Q2.5	Q97.5
[1,]	38.22469	3.459319	31.5402	45.08605
[2,]	38.22469	3.459319	31.5402	45.08605
[3,]	38.22469	3.459319	31.5402	45.08605
[4,]	38.22469	3.459319	31.5402	45.08605
[5,]	38.22469	3.459319	31.5402	45.08605
[6,]	38.22469	3.459319	31.5402	45.08605

```
fitted(fit.est.1, re_formula=NA) |> head()
```

	Estimate	Est.Error	Q2.5	Q97.5
[1,]	37.44954	4.003155	28.8998	45.02236
[2,]	37.44954	4.003155	28.8998	45.02236
[3,]	37.44954	4.003155	28.8998	45.02236
[4,]	37.44954	4.003155	28.8998	45.02236
[5,]	37.44954	4.003155	28.8998	45.02236
[6,]	37.44954	4.003155	28.8998	45.02236

Finally, we check the model fit. Again, we can choose to use predictions with random effects (default) and compute the **conditional R²**, while without random effects (`re_formula=NA`) we compute the **marginal R²**.

```
bayes_R2(fit.est.1) # conditional (default)
```

	Estimate	Est.Error	Q2.5	Q97.5
R2	0.2695999	0.104506	0.06138227	0.4586746

```
bayes_R2(fit.est.1, re_formula=NULL) # conditional (the same)
```

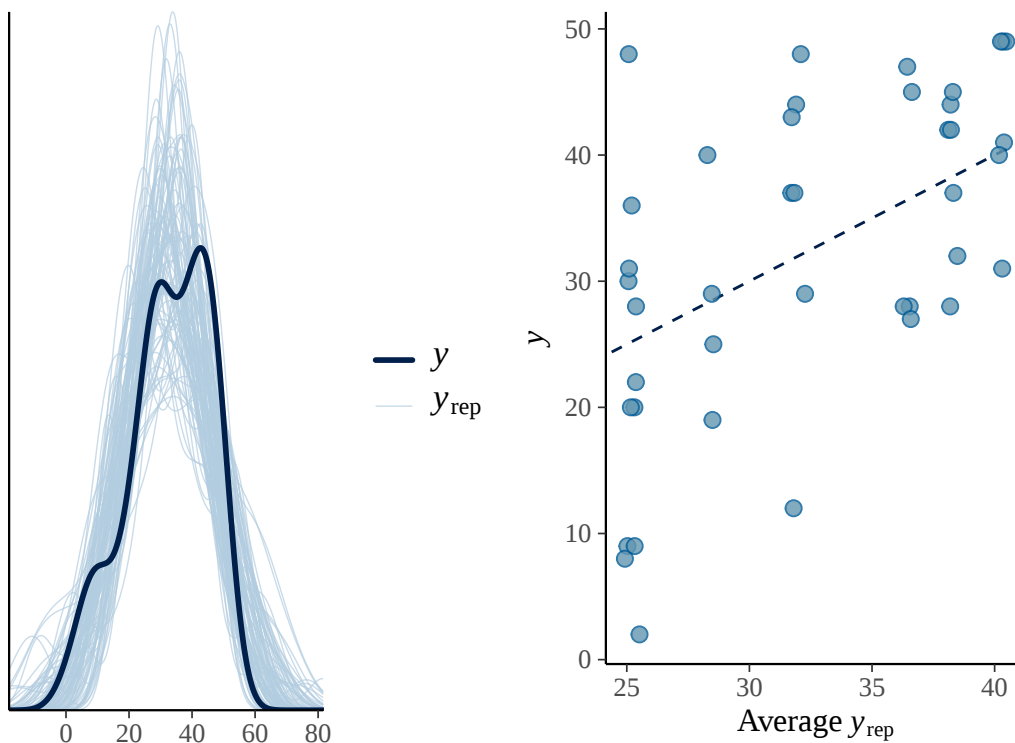
	Estimate	Est.Error	Q2.5	Q97.5
R2	0.2695999	0.104506	0.06138227	0.4586746

```
bayes_R2(fit.est.1, re_formula=NA) # marginal
```

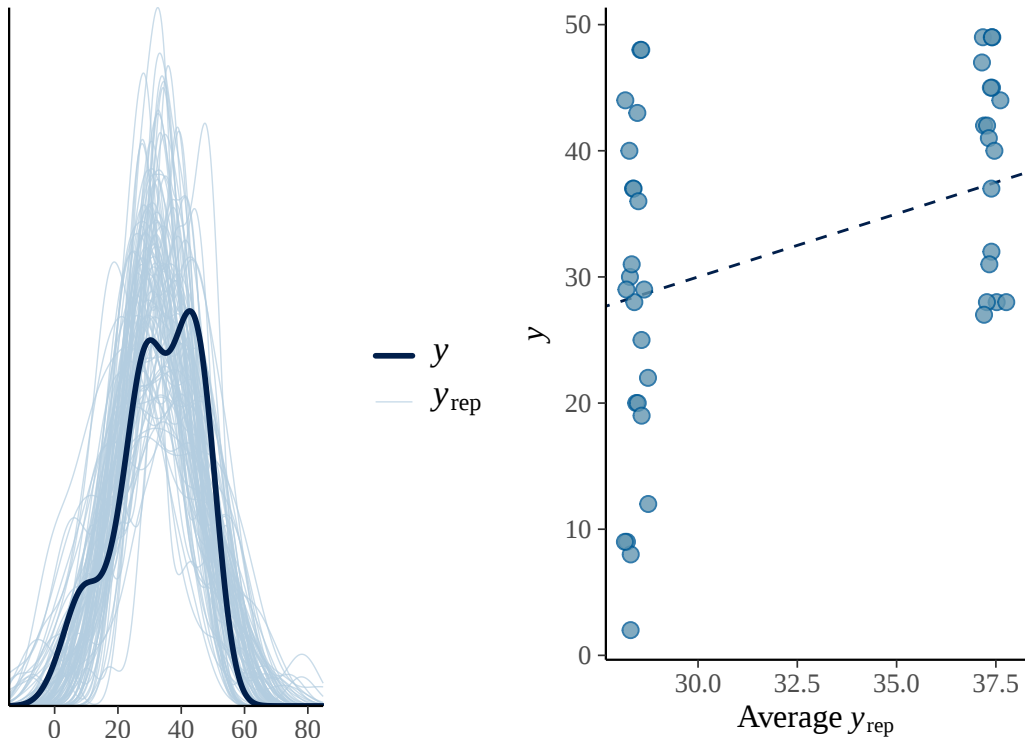
	Estimate	Est.Error	Q2.5	Q97.5
R2	0.1543518	0.1097426	0.0007859223	0.3845651

Model checks use random effects per default, but can also be done with fixed effects only if required.

```
p1 = pp_check(fit.est.1, ndraws=100)
p2 = pp_check(fit.est.1, type="scatter_avg")
plot_grid(p1,p2)
```



```
p1 = pp_check(fit.est.1, ndraws=100, re_formula=NA)
p2 = pp_check(fit.est.1, type="scatter_avg", re_formula=NA)
plot_grid(p1,p2)
```

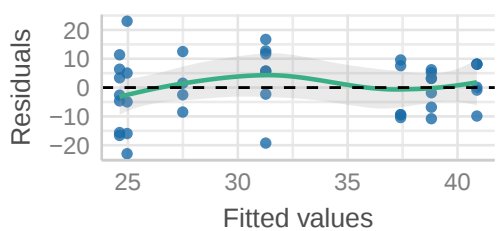


Since this is a linear model, we can assess model assumptions with `check_model()`

```
check_model(fit.est.1, check=c("linearity","homogeneity","qq","normality"))
```

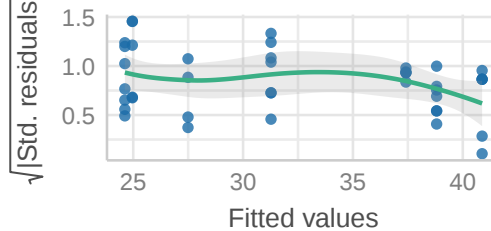
Linearity

Reference line should be flat and horizontal



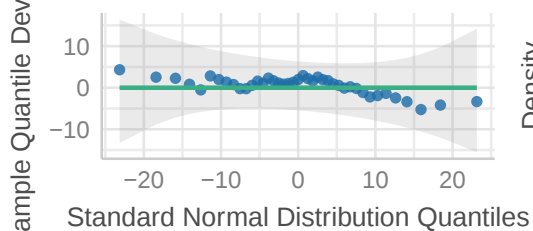
Homogeneity of Variance

Reference line should be flat and horizontal



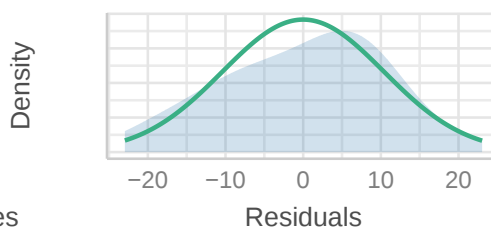
Normality of Residuals

Points should fall along the line



Normality of Residuals

Distribution should be close to the normal distribution

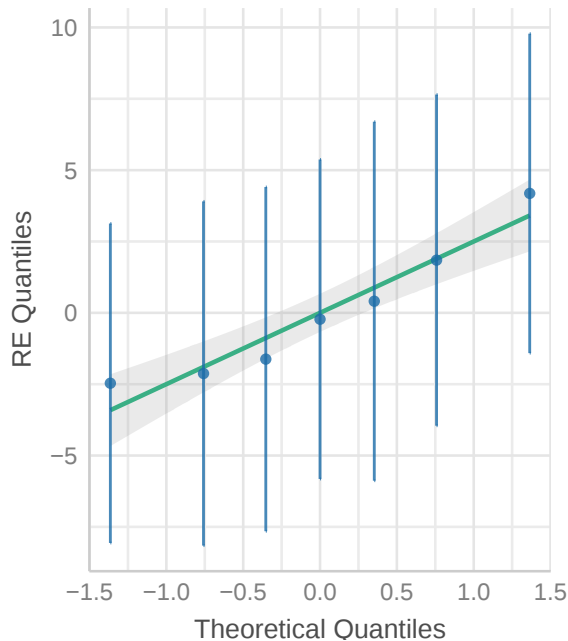


It also provides an additional check for normality of random effects

```
check_model(fit.est.1, check=c("reqq"))
```

Normality of Random Effects (Estuary)

Dots should be plotted along the line



The summary table already tells us that the difference between pristine and modified estuaries `ModPristine` is -8.96, 95% CI [-18.45,1.91]. Or we use the `hypothesis` function. Note that here we get 90% CIs. But `Post.Prob` is the quantity of interest here

```
hypothesis(fit.est.1, "ModPristine<0")
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(ModPristine) < 0	-8.96	5.11	-16.83	-0.15	20.28	0.95	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.

'*': For one-sided hypotheses, the posterior probability exceeds 95%;

for two-sided hypotheses, the value tested against lies outside the 95%-CI.

Posterior probabilities of point hypotheses assume equal prior probabilities.

If we had fitted the model **without random intercepts** (complete pooling), the model would be overly confident in the Modified-Pristine difference (uncertainty is smaller here):

```
fit.est.compl = brm(Total ~ Mod,
  prior = prior(normal(0,10), class=b),
  data = estuaries)
```

```
fixef(fit.est.compl)
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	38.27533	2.653983	33.15225	43.339139
ModPristine	-10.46960	3.453143	-17.11447	-3.830119

```
fixef(fit.est.1)
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	37.44954	4.003155	28.89980	45.022357
ModPristine	-8.96339	5.108531	-18.45213	1.912913

Random intercepts ANCOVA

Now we add the continuous predictor Temperature.

Question: Is there still a difference between modified and pristine estuaries when controlling for temperature?

Deterministic part: $\mu_i = b(\text{Mod}_i) + c \cdot \text{Temp} + \delta(\text{Estuary}_i)$

Stochastic part: $\text{Total}_i \sim \text{Normal}(\mu_i, \sigma)$

Hierarchical part: $\delta_j \sim \text{Normal}(0, \sigma_{Est})$

$i = 1 \dots N$, $\text{Mod} = 1, 2$, $\text{Estuary} = 1 \dots 7$

```
default_prior(Total ~ Mod+scale(Temperature)+(1|Estuary),
  data = estuaries)
```

	prior	class	coef	group	resp	dpar	nlpar	lb	ub	source
	(flat)	b								default
	(flat)	b	ModPristine							(vectorized)
	(flat)	b	scaleTemperature							(vectorized)
student_t(3, 32, 14.8)		Intercept								default
student_t(3, 0, 14.8)		sd						0		default
student_t(3, 0, 14.8)		sd	Estuary					0		(vectorized)
student_t(3, 0, 14.8)		sd	Intercept	Estuary				0		(vectorized)
student_t(3, 0, 14.8)		sigma						0		default

We only need to specify a prior for the fixed effects, the effect of Modified and slope in Temperature (scaled). We choose a very weak prior for both.

```
fit.est.2 = brm(Total ~ Mod+scale(Temperature)+(1|Estuary),
  prior = prior(normal(0,10), class=b) +
  data = estuaries)
```

Check convergence.

```
summary(fit.est.2)
```

Warning: There were 4 divergent transitions after warmup. Increasing adapt_delta above 0.8 may

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: Total ~ Mod + scale(Temperature) + (1 | Estuary)
Data: estuaries (Number of observations: 41)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
total post-warmup draws = 4000
```

Multilevel Hyperparameters:

~Estuary (Number of levels: 7)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	3.35	2.79	0.15	10.27	1.00	931	1010

Regression Coefficients:

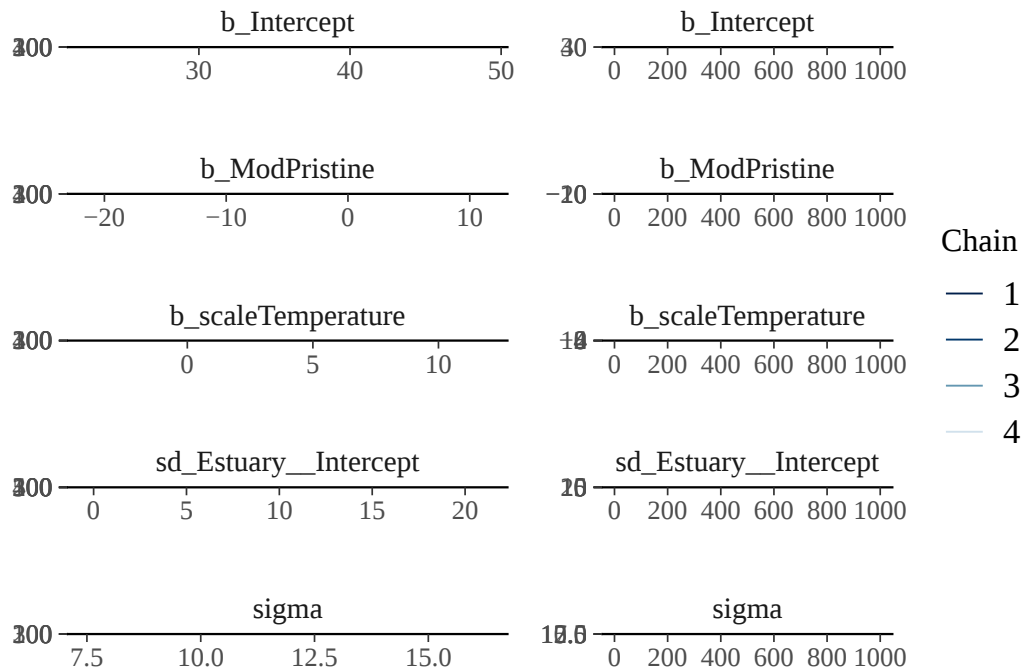
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	37.04	3.27	30.08	43.12	1.00	1527	1099
ModPristine	-8.21	4.24	-15.73	0.95	1.00	1442	892
scaleTemperature	3.79	2.12	-0.69	7.85	1.00	2209	1576

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	11.00	1.28	8.88	13.80	1.00	3408	2981

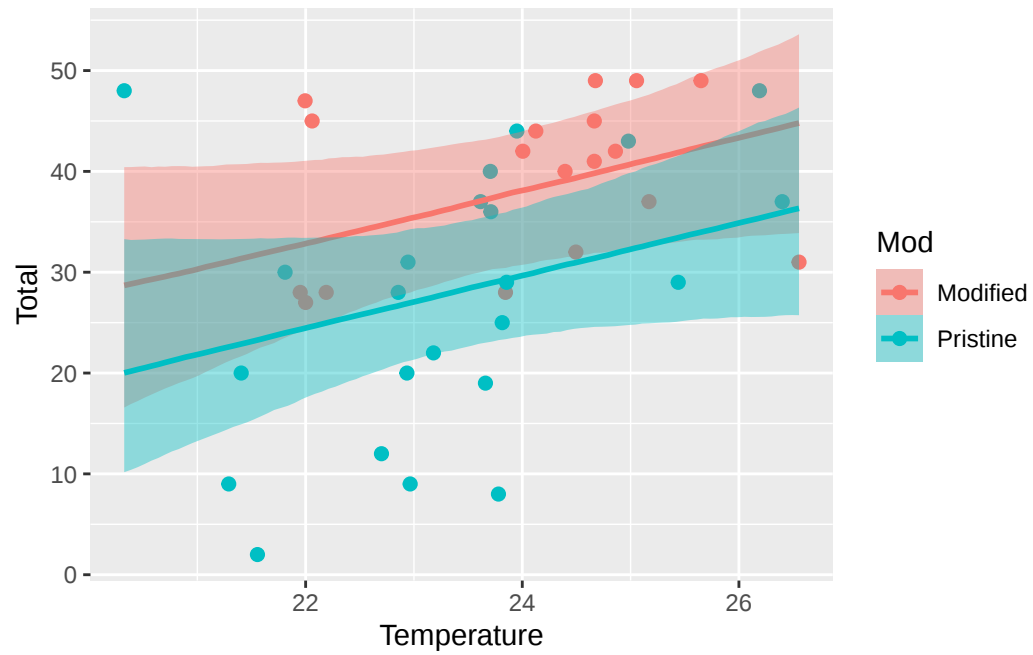
Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

```
plot(fit.est.2)
```

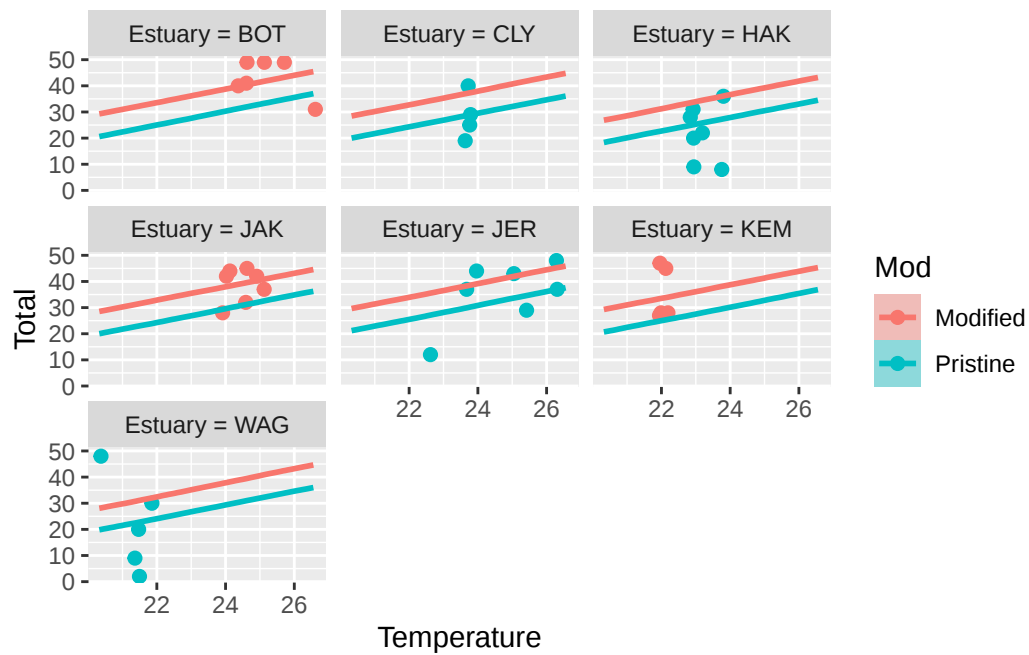


Conditional effects plots of fixed effects only (default, “population-level”), but by specifying `re_formula` and `conditions`, we can plot on group-level, too

```
plot(conditional_effects(fit.est.2, effect="Temperature:Mod"),
     points=T)
```

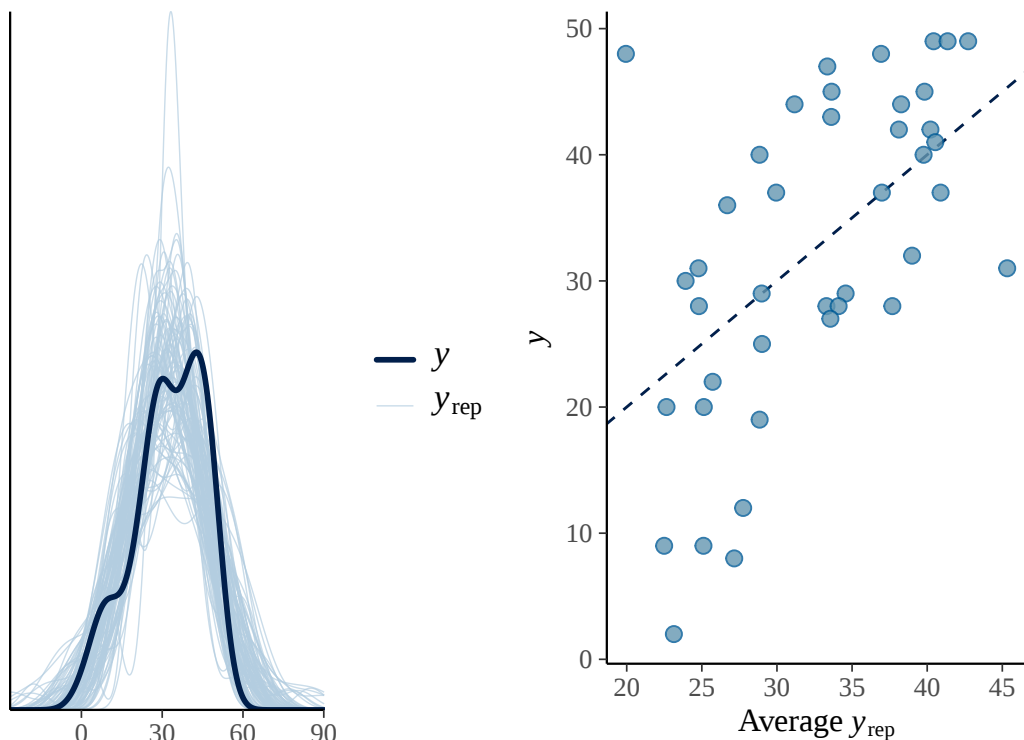


```
plot(conditional_effects(fit.est.2, effect="Temperature:Mod",
  re_formula = NULL,
  conditions = make_conditions(fit.est.2, var=c("Estuary")),
  prob=0),
  points=T,
  point_args=c(alpha=1, width=0.1, size=2))
```



Finally, we check the model fit

```
p1 = pp_check(fit.est.2, ndraws=100)
p2 = pp_check(fit.est.2, type="scatter_avg")
plot_grid(p1,p2)
```



There is only very weak support for the temperature model. Both models come to similar conclusions regarding Mod-Pristine difference in species richness.

```
L00(fit.est.1, fit.est.2)
```

```

      elpd_diff se_diff
fit.est.2  0.0      0.0
fit.est.1 -0.5      2.3

```

```
hypothesis(fit.est.1, "ModPristine<0")
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(ModPristine) < 0	-8.96	5.11	-16.83	-0.15	20.28	0.95	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.

'*': For one-sided hypotheses, the posterior probability exceeds 95%;

for two-sided hypotheses, the value tested against lies outside the 95%-CI.

Posterior probabilities of point hypotheses assume equal prior probabilities.

```
hypothesis(fit.est.2, "ModPristine<0")
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(ModPristine) < 0	-8.21	4.24	-14.75	-1	28.2	0.97	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.

'*': For one-sided hypotheses, the posterior probability exceeds 95%;

for two-sided hypotheses, the value tested against lies outside the 95%-CI.

Posterior probabilities of point hypotheses assume equal prior probabilities.

Exercise

Also from Warton, D. (2022) Eco-Stats: Data Analysis in Ecology

Test the additional effect of outer vs inner (Zone) in this new dataset. Does the effect still hold when accounting for temperature?

```
data("estuaryZone")
table(estuaryZone$Estuary, estuaryZone$Mod, estuaryZone$Zone)
```

, , = Inner

	Modified	Pristine
BOT	6	0
CLY	0	4
HAK	0	7
JAK	7	0
JER	0	7
KEM	5	0
WAG	0	6

, , = Outer

	Modified	Pristine
BOT	6	0
CLY	0	5
HAK	0	7

JAK	4	0
JER	0	5
KEM	6	0
WAG	0	2

```
ID.complete = complete.cases(estuaryZone[, c("Total", "Mod", "Estuary", "Temperature", "Zone")])
estuaryZone = estuaryZone[ID.complete, ]
```

Without temperature

```
default_prior(Total ~ Mod+Zone+(1|Estuary), data=estuaryZone)
```

	prior	class	coef	group	resp	dpar	nlpar	lb	ub	source
	(flat)	b								default
	(flat)	b	ModPristine							(vectorized)
	(flat)	b	ZoneOuter							(vectorized)
student_t(3, 37, 14.1)		Intercept								default
student_t(3, 0, 14.1)		sd						0		default
student_t(3, 0, 14.1)		sd		Estuary				0		(vectorized)
student_t(3, 0, 14.1)		sd	Intercept	Estuary				0		(vectorized)
student_t(3, 0, 14.1)		sigma						0		default

```
fit.zone.1 = brm(Total ~ Mod+Zone+(1|Estuary),
  prior = prior(normal(0,10), class=b),
  data = estuaries)
```

Check convergence

```
summary(fit.zone.1)
```

Warning: There were 5 divergent transitions after warmup. Increasing adapt_delta above 0.8 may

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: Total ~ Mod + Zone + (1 | Estuary)
Data: estuaryZone (Number of observations: 76)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Multilevel Hyperparameters:

~Estuary (Number of levels: 7)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	2.40	1.92	0.09	7.43	1.00	1293	1393

Regression Coefficients:

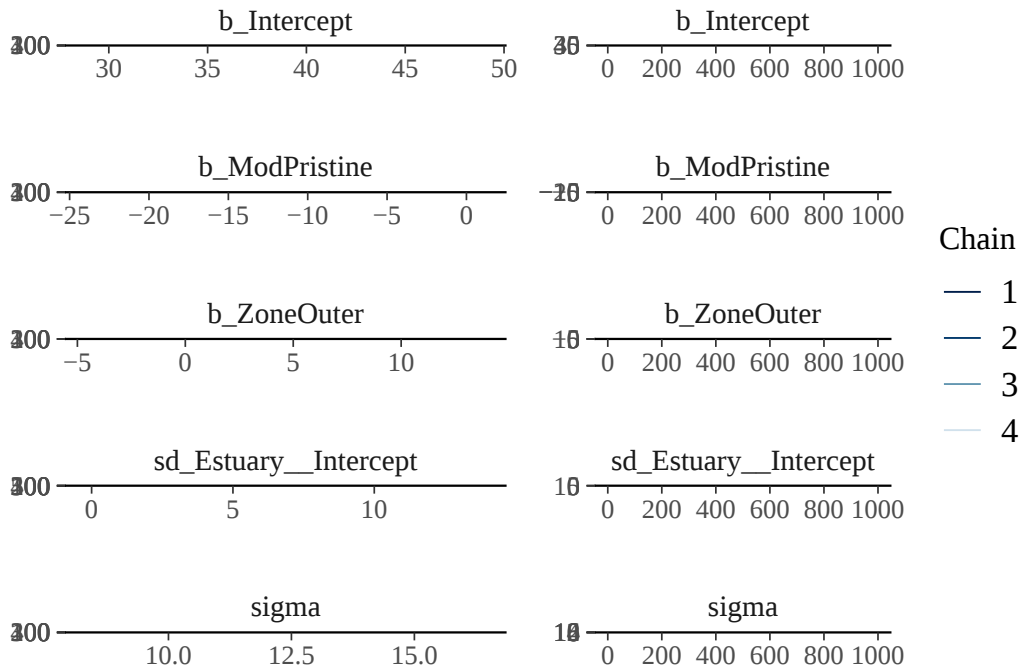
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	39.47	2.76	33.52	44.67	1.00	1723	1390
ModPristine	-12.47	3.28	-18.63	-5.51	1.00	1512	1237
ZoneOuter	5.15	2.50	0.21	10.11	1.00	3563	2572

Further Distributional Parameters:

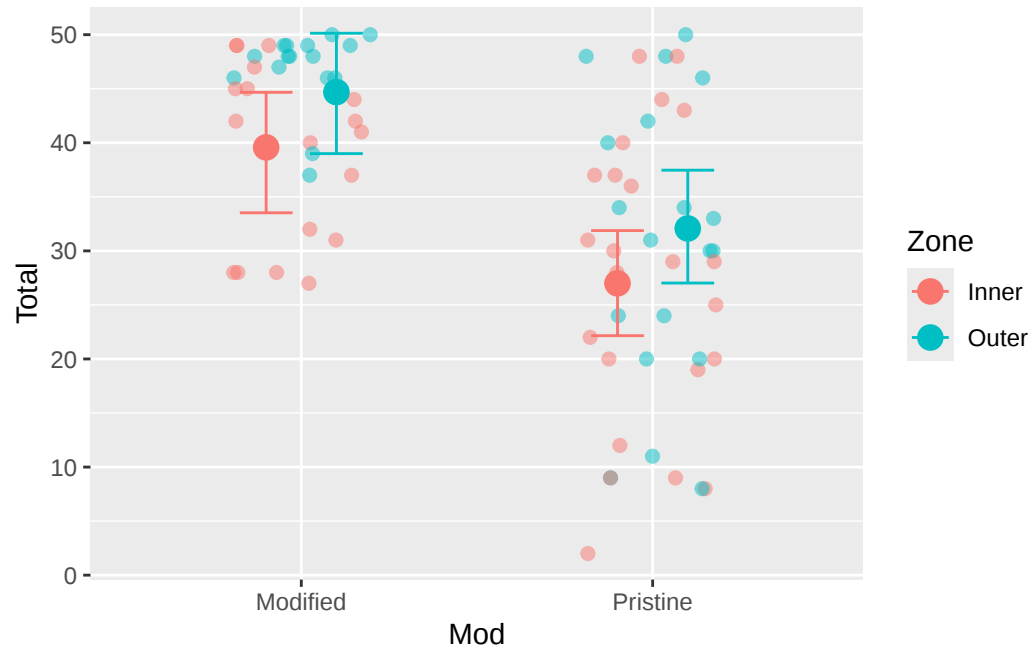
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	10.88	0.93	9.25	12.83	1.00	3217	2467

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).

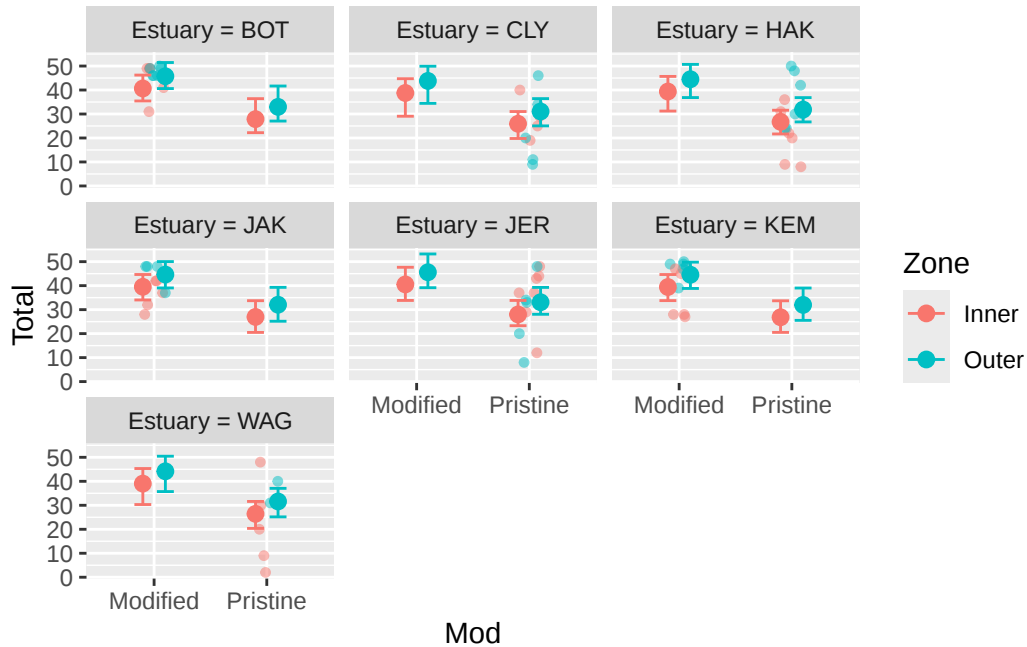
```
plot(fit.zone.1)
```



```
plot(conditional_effects(fit.zone.1, effect="Mod:Zone"),
     points=T,
     point_args=c(alpha=0.5, width=0.2, size=2))
```



```
plot(conditional_effects(fit.zone.1, effect="Mod:Zone",
                        re_formula = NULL,
                        conditions = make_conditions(fit.zone.1, var=c("Estuary"))),
     points=T,
     point_args=c(alpha=0.5, width=0.1))
```



From the summary table we already have the information that there are on average +5.15 species more in the outer zone, but you can also check the hypothesis function.

```
hypothesis(fit.zone.1, c("ZoneOuter>0",
                          "ModPristine<0"))
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(ZoneOuter) > 0	5.15	2.50	1.09	9.17	47.78	0.98	*
2	(ModPristine) < 0	-12.47	3.28	-17.69	-6.89	799.00	1.00	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.

'*': For one-sided hypotheses, the posterior probability exceeds 95%;

for two-sided hypotheses, the value tested against lies outside the 95%-CI.

Posterior probabilities of point hypotheses assume equal prior probabilities.

With temperature

```
default_prior(Total ~ Mod+Zone+scale(Temperature)+(1|Estuary),
              data = estuaryZone)
```

	prior	class	coef	group	resp	dpar	nlpar	lb	ub	source
	(flat)	b								default
	(flat)	b	ModPristine							(vectorized)
	(flat)	b	scaleTemperature							(vectorized)
	(flat)	b	ZoneOuter							(vectorized)
student_t(3, 37, 14.1)		Intercept								default
student_t(3, 0, 14.1)		sd						0		default
student_t(3, 0, 14.1)		sd		Estuary				0		(vectorized)
student_t(3, 0, 14.1)		sd	Intercept	Estuary				0		(vectorized)
student_t(3, 0, 14.1)		sigma						0		default

Here we got 19 divergent transitions while fitting, so it's time to increase MCMC accuracy with `adapt_delta=0.9`. Then, everything is fine:

```
fit.zone.2 = brm(Total ~ Mod+Zone+scale(Temperature)+(1|Estuary),
  prior = prior(normal(0,10), class=b) +
  prior(normal(0,10), class=b, coef=scaleTemperature),
  data = estuaryZone,
  control = list(adapt_delta=0.9) )
```

Check convergence

```
summary(fit.zone.2)
```

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: Total ~ Mod + Zone + scale(Temperature) + (1 | Estuary)
Data: estuaryZone (Number of observations: 76)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Multilevel Hyperparameters:

~Estuary (Number of levels: 7)

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	2.09	1.81	0.08	6.53	1.00	1134	1724

Regression Coefficients:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	36.04	2.99	29.98	41.96	1.00	2631	2438
ModPristine	-10.76	3.16	-16.90	-4.38	1.00	2692	1986
ZoneOuter	10.61	3.43	3.66	17.27	1.00	2983	2300

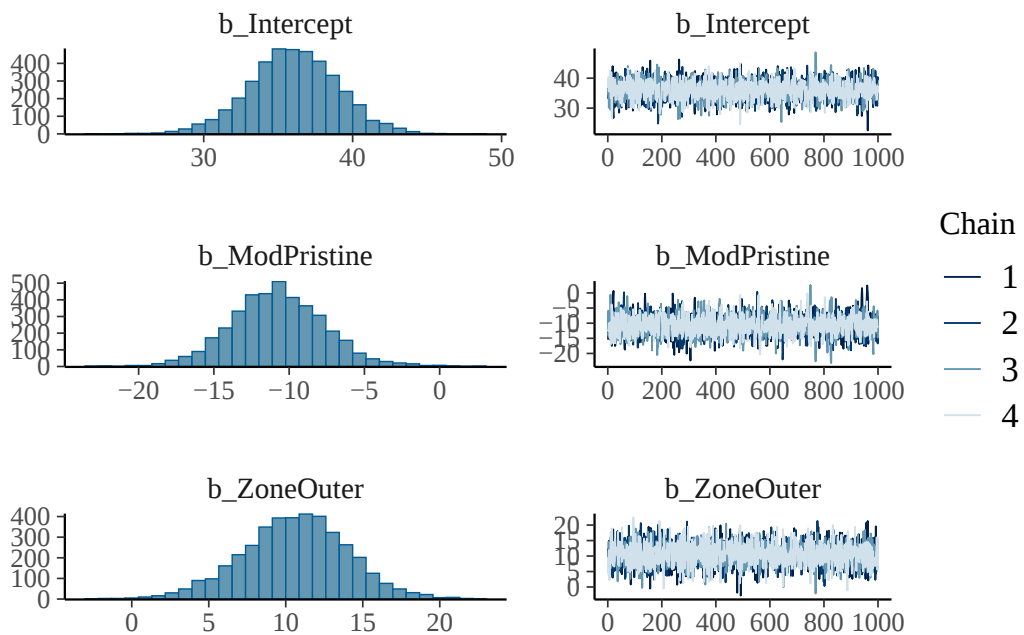
scaleTemperature	4.23	1.84	0.58	7.81	1.00	2878	2733
------------------	------	------	------	------	------	------	------

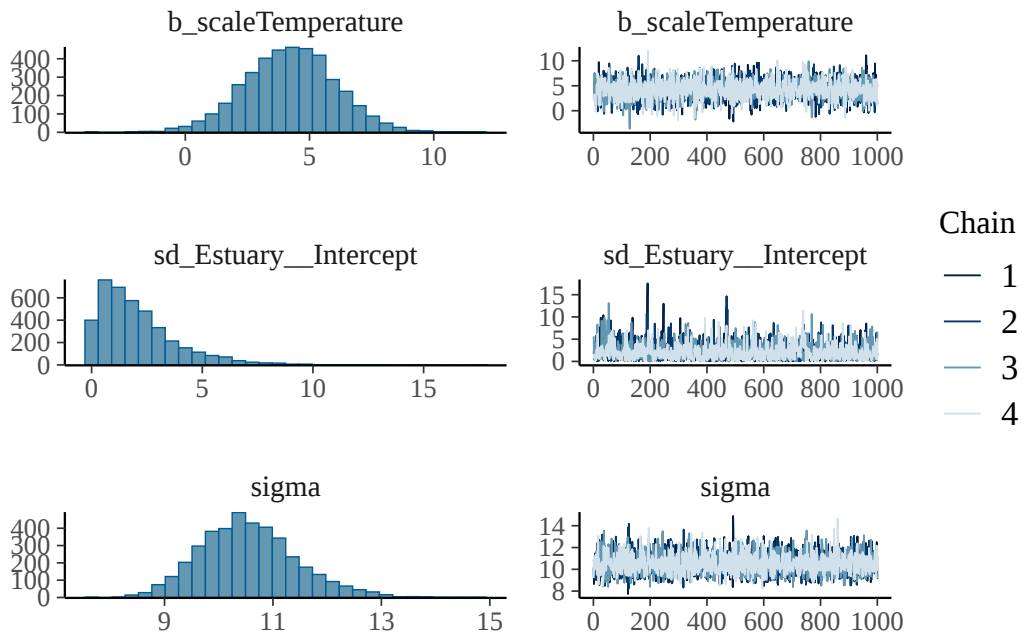
Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	10.55	0.89	8.96	12.50	1.00	4920	3028

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).

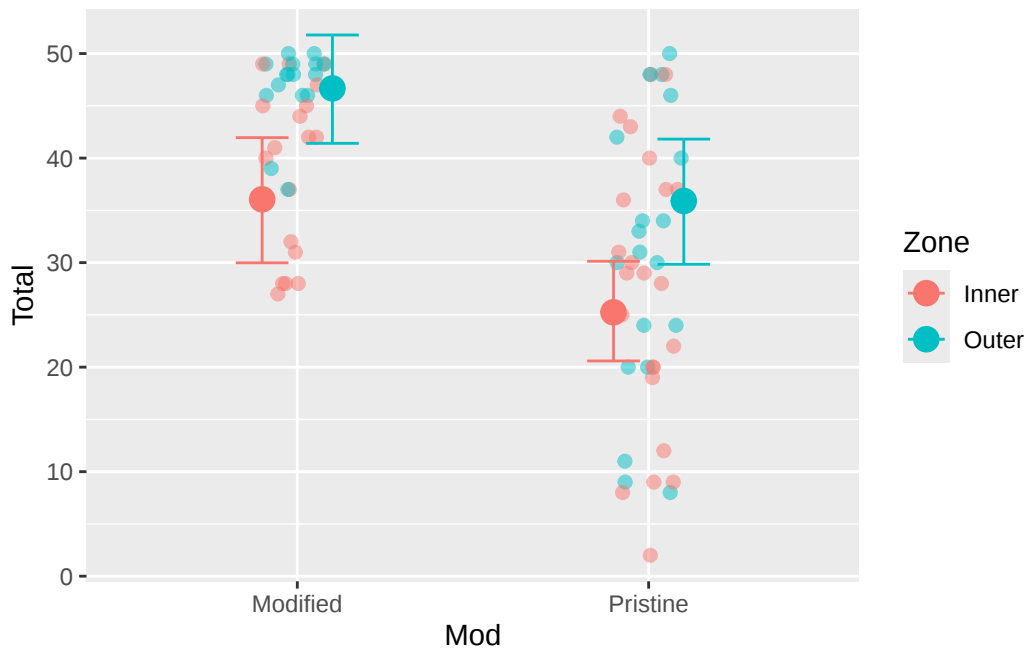
```
plot(fit.zone.2)
```



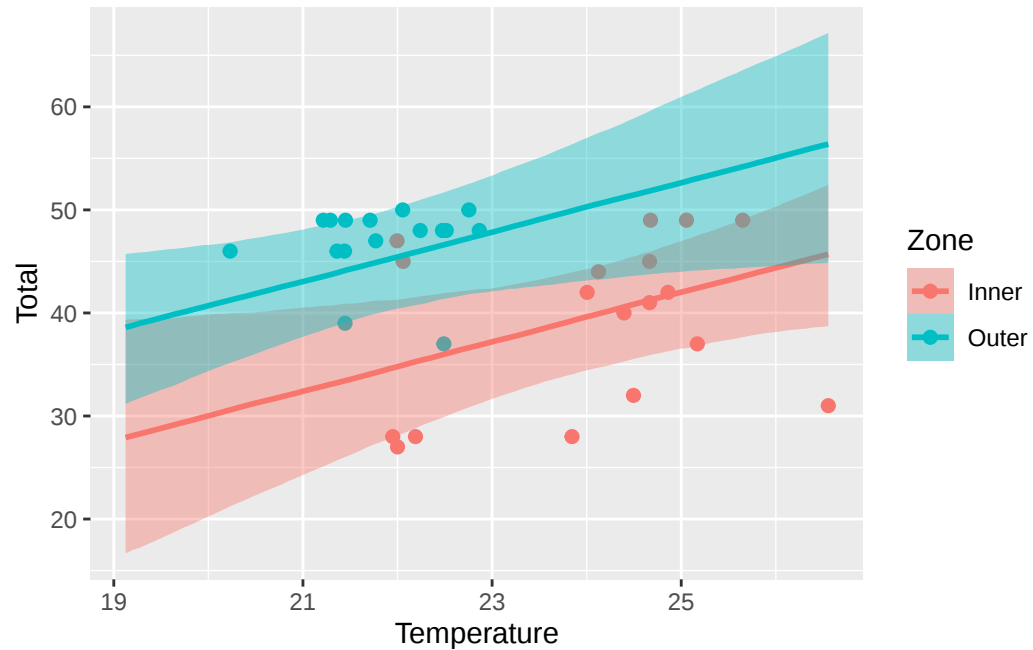


With 3 predictors, we can plot against 2 while the 3rd is at its reference level

```
plot(conditional_effects(fit.zone.2,
                        effect="Mod:Zone"), # Temperature=mean
     points=T,
     point_args=c(alpha=0.5, width=0.1))
```

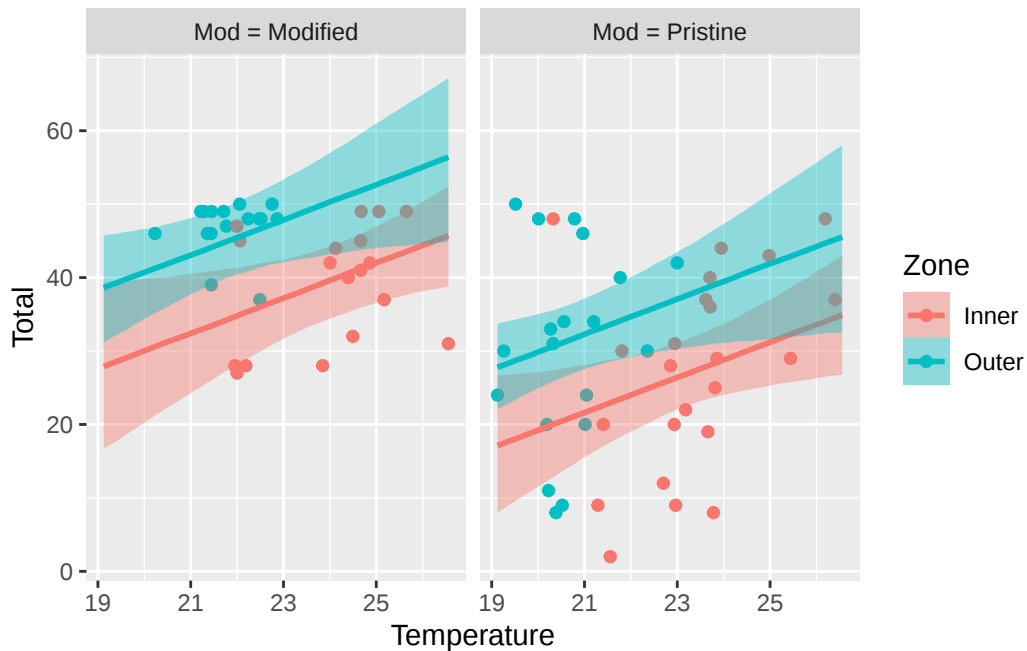


```
plot(conditional_effects(fit.zone.2,
                        effect="Temperature:Zone"), # Mod=modified
     points=T)
```



Or we can plot against all 3

```
plot(conditional_effects(fit.zone.2,
                        effect="Temperature:Zone",
                        conditions = make_conditions(fit.zone.2, var=c("Mod"))),
     points=T)
```



There is only very weak support for including temperature regarding predictive accuracy, but temperature effect is positive

```
LOO(fit.zone.1, fit.zone.2)
```

	elpd_diff	se_diff
fit.zone.2	0.0	0.0
fit.zone.1	-2.1	3.0

The direct effect of Zone is now 2x stronger when controlling for temperature! There are ~10 species more in the outer zone than in an inner zone of the same temperature. How can that happen??

```
fixef(fit.zone.1)
```

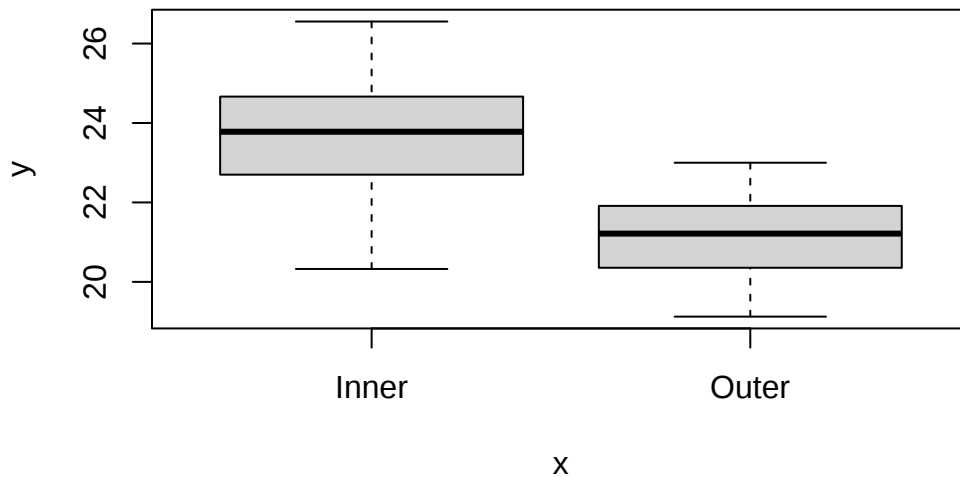
	Estimate	Est.Error	Q2.5	Q97.5
Intercept	39.470272	2.757911	33.5211542	44.668288
ModPristine	-12.473993	3.275475	-18.6265579	-5.514716
ZoneOuter	5.151376	2.499280	0.2089617	10.108564

```
fixef(fit.zone.2)
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	36.04123	2.986821	29.9818478	41.958176
ModPristine	-10.75982	3.156734	-16.8988015	-4.380493
ZoneOuter	10.60718	3.433391	3.6634936	17.270785
scaleTemperature	4.23131	1.838677	0.5823119	7.806587

However, inner zones are usually warmer, so this difference is a rather pointless prediction!

```
plot(estuaryZone$Zone, estuaryZone$Temperature)
```



If interested in a meaningful difference, compare predictions between inner & outer zone at their typical temperatures.

```
mus = fitted(fit.zone.2,
             newdata = data.frame(Zone=c("Inner", "Outer"),
                                   Temperature=c(23.5, 21.5),
                                   Mod=c("Modified", "Modified")),
             re_formula = NA, # only fixed effects prediction
             summary=FALSE
)
mus = data.frame(mus)
names(mus)=c("Inner","Outer")
hypothesis(mus, "Inner<Outer")
```

Hypothesis Tests for class :

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(Inner)-(Outer) < 0	-5.85	2.41	-9.78	-1.88	94.24	0.99	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
 '*': For one-sided hypotheses, the posterior probability exceeds 95%;
 for two-sided hypotheses, the value tested against lies outside the 95%-CI.
 Posterior probabilities of point hypotheses assume equal prior probabilities.

This is much closer to the model without temperature, which actually would be sufficient here.

```
hypothesis(fit.zone.1, c("ZoneOuter>0",
                        "ModPristine<0"))
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(ZoneOuter) > 0	5.15	2.50	1.09	9.17	47.78	0.98	*
2	(ModPristine) < 0	-12.47	3.28	-17.69	-6.89	799.00	1.00	*

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
 '*': For one-sided hypotheses, the posterior probability exceeds 95%;
 for two-sided hypotheses, the value tested against lies outside the 95%-CI.
 Posterior probabilities of point hypotheses assume equal prior probabilities.

Be careful if predictors are correlated. Parameters only describe **direct** effect, meaning other predictors are held constant. Just looking at effect size, you might compare predictions which are very unlikely to occur in the data (Inner and Outer zone with same temperature).