Bayesplot :: CHEAT SHEET

```
library("bayesplot")
library("rstanarm")
library("ggplot2")
library("rstan")
library("dplyr")
```

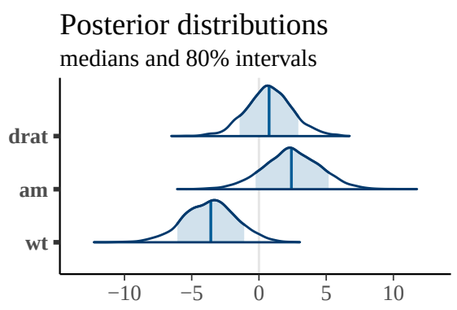
rstanarm

To showcase `bayesplot`, we'll fit linear regression using `rstanarm::stan_glm` and use this model throughout.

```
model <- stan_glmer(mpg ~ .,
                    data=mtcars, chains=4)
posterior <- as.matrix(model)
```

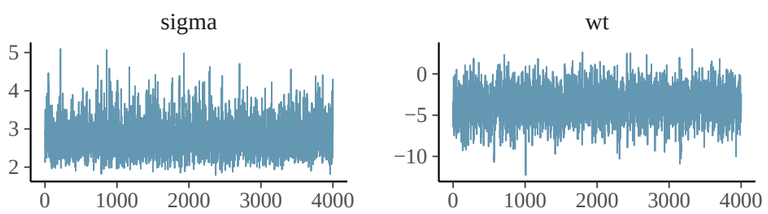
Chances are good you're most interested in the posterior distributions for select parameters.

```
plot_title <- ggtitle("Posterior distributions",
                      "medians and 80% intervals")
mcmc_areas(posterior,
           pars = c("drat", "am", "wt"),
           prob = 0.8) + plot_title
```



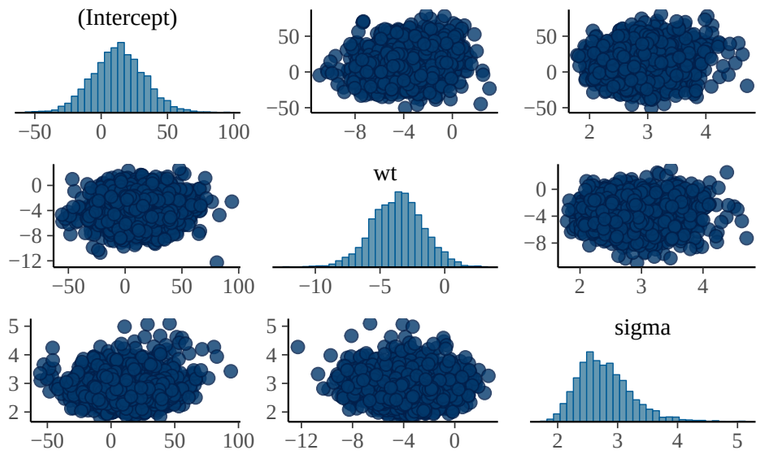
Diagnosing convergence with traceplots is simple.

```
mcmc_trace(posterior, pars=c("sigma", "wt"))
```



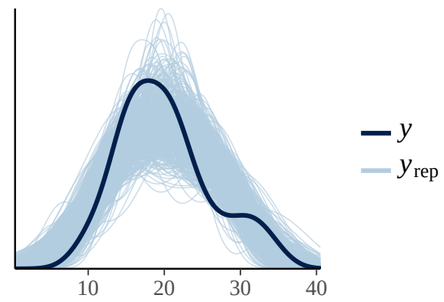
The pairs plot is helpful in determining if you have any highly correlated parameters. Note that different from above, we'll now extract each of the four chains posterior draws separately using `as.array`.

```
posterior_chains <- as.array(model)
pairs <- posterior_chains %>%
    mcmc_pairs(pars = c("(Intercept)", "wt", "sigma"))
```

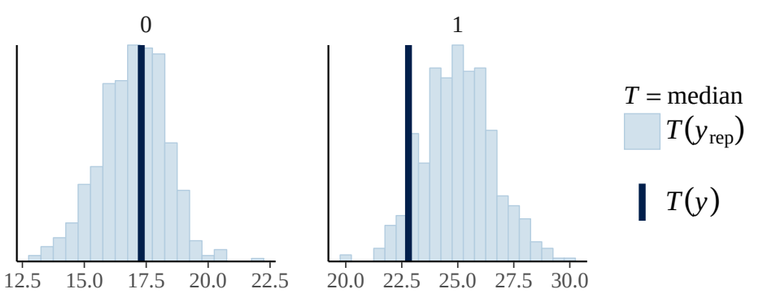


Check how well the model covers your data with draws from the posterior predictive density.

```
ppd <- posterior_predict(model, draws=500)
ppd %>%
    ppc_dens_overlay(y = model$y, yrep = .)
```



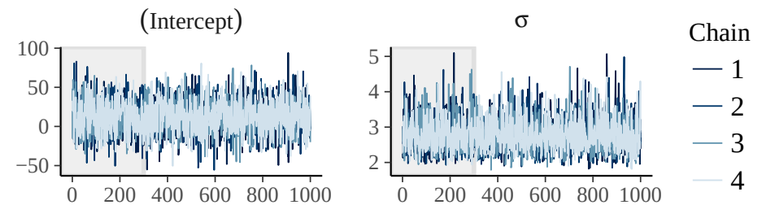

```
ppd %>%
    ppc_stat_grouped(y = mtcars$mpg,  group = mtcars$am,
                     stat = "median", binwidth=0.5)
```



RStan

Of course, `bayesplot` plays nicely with `RStan`, too. We'll just pretend like we fit a model with `rstan::sampling` by extracting the `stanfit` object from out `stanreg` model. This allows you to visualize individual chains.

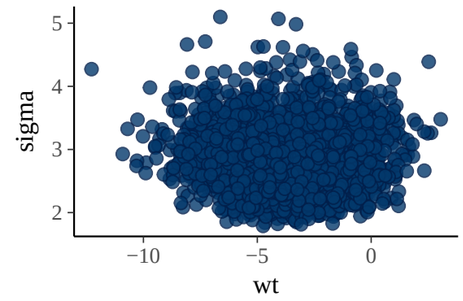
```
fargs <- list(ncol = 2, labeller = label_parsed)
pars <- c("(Intercept)", "sigma")
mcmc_trace(posterior_chains, pars = pars,
           n_warmup = 300, facet_args = fargs)
```



Diagnostics

Bayesplot makes it easy to check diagnostics specific to the NUTS sampling method that `rstanarm` uses by default.

```
mcmc_scatter(posterior, pars = c("wt", "sigma"),
  np = nuts_params(model$stanfit))
```




```
np <- nuts_params(model$stanfit)
mcmc_nuts_energy(np, binwidth=1) +
    ggtitle("NUTS Energy Diagnostic")
```