360 Final Project

Group: Flora Shi, Belle Xu

18 April, 2021

```
stop <- read.table("stop-and-frisk.dat", header = TRUE)
```

Exploratory Data Analysis

Exploratory data analysis should support project goals and help guide specification of model.

```
#make categorical variables factor
stop <- stop %>%
  mutate(precinct = factor(precinct)) %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2"))) %>%
  mutate(crime = factor(crime))
```

```
#investigate mean and var of stops
stop <- stop %>%
  group_by(precinct) %>%
  mutate(stops_in_precinct = sum(stops))

stops_by_precinct <- stop %>% distinct(stops_in_precinct) %>%
  pull(stops_in_precinct)

mean(stops_by_precinct)
```

```
## [1] 1752.267
```

```
var(stops_by_precinct) # might need to do NB regression...
```

```
## [1] 808148.5
```

```
sd(stops_by_precinct)
```

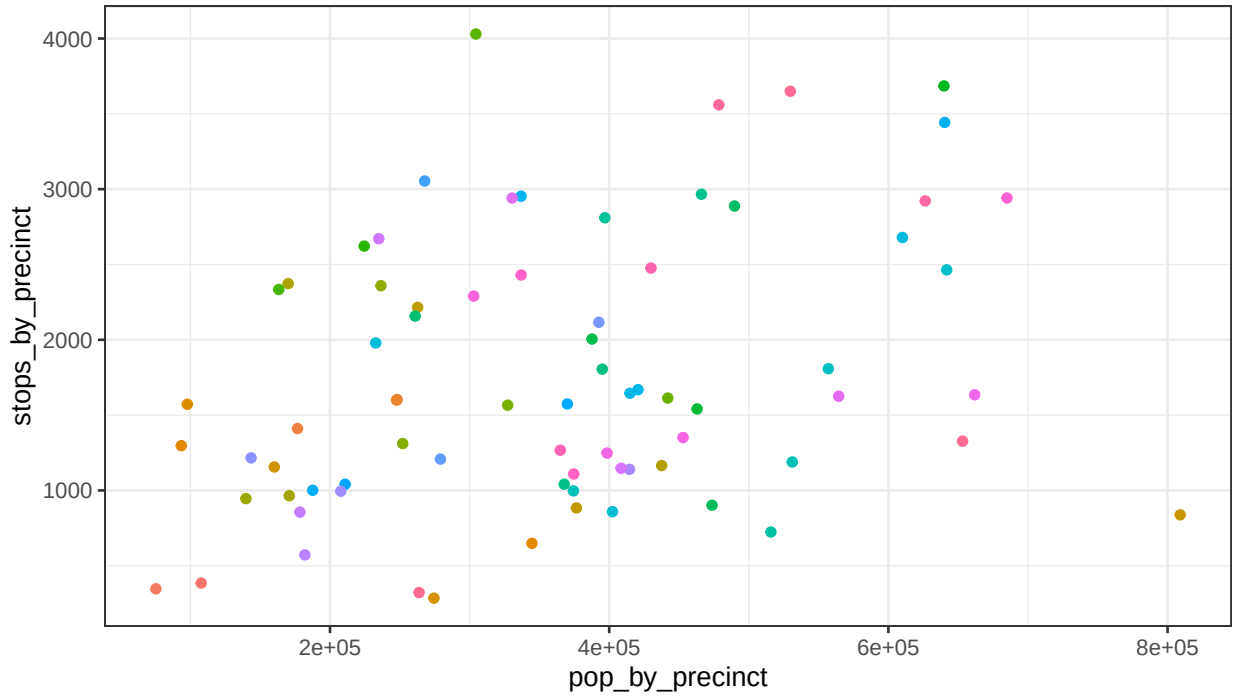
```
## [1] 898.9708
```

```
max(stops_by_precinct)
```

```
## [1] 4030
```

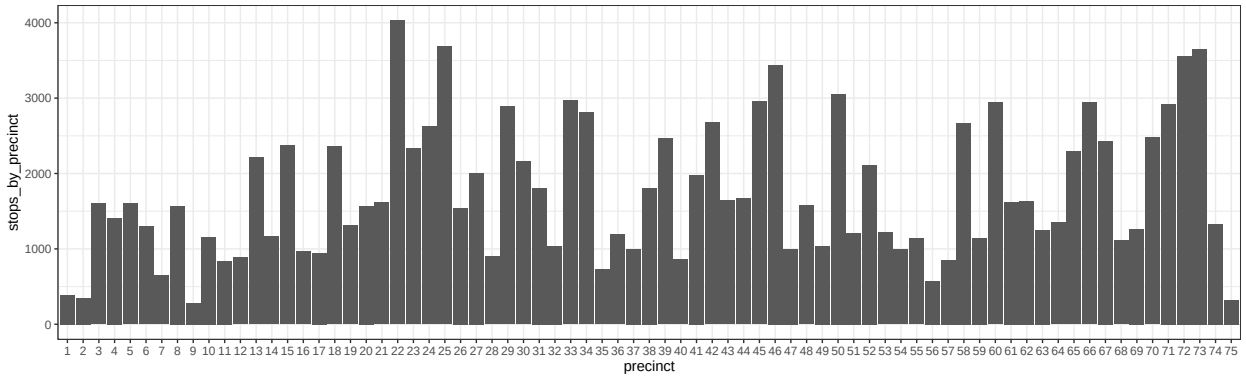
```
precinct <- seq(from = 1, to = 75, by = 1)
pop_by_precinct<- stop %>% group_by(precinct) %>%
  mutate(pop_in_precinct = sum(pop)) %>%
  distinct(pop_in_precinct) %>%
  pull(pop_in_precinct)
by_precinct_df <- data.frame(pop_by_precinct = pop_by_precinct,
                             stops_by_precinct = stops_by_precinct,
                             precinct = as.factor(precinct))

#number of stops vs. precinct pop
ggplot(data = by_precinct_df,
       mapping = aes(x = pop_by_precinct, y =stops_by_precinct,
                     colour = precinct)) + geom_point()+
  theme(legend.position = "none")
```




```
#no obvious trend between population in precinct vs. stops in a precinct
```

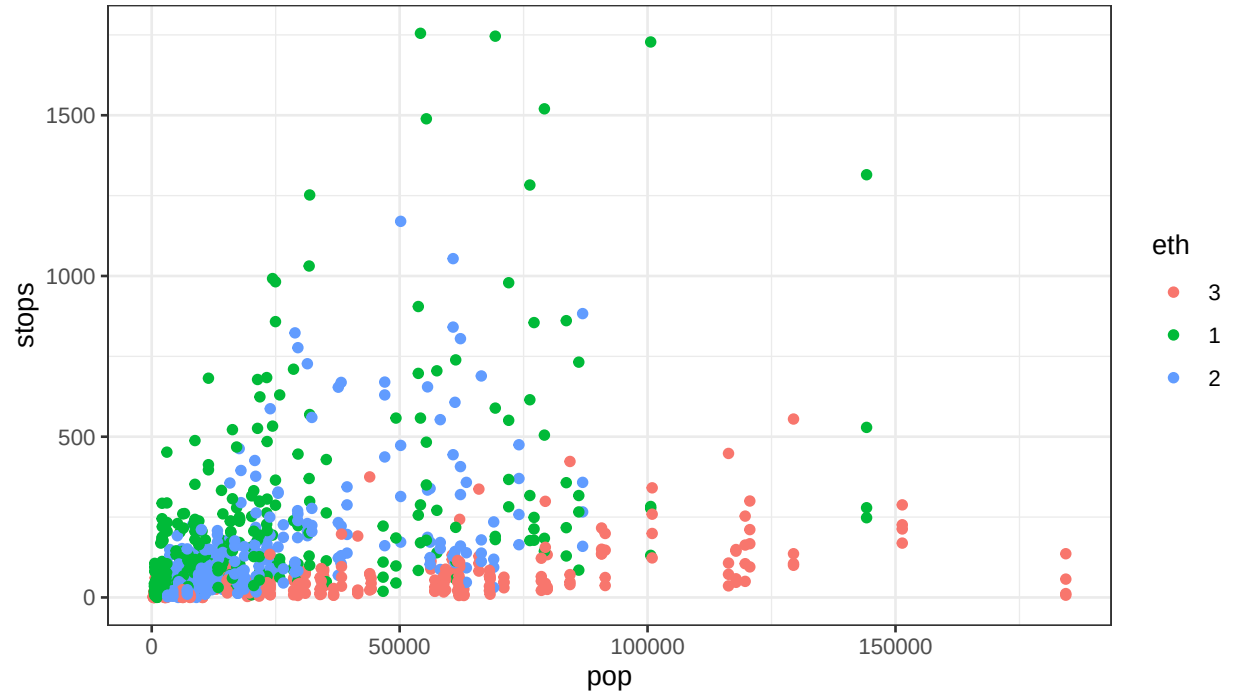
```
#number of stops by precinct
ggplot(data = by_precinct_df,
       mapping = aes(x = precinct, y =stops_by_precinct)) +
  geom_col()
```




```
#Need interpretation
```

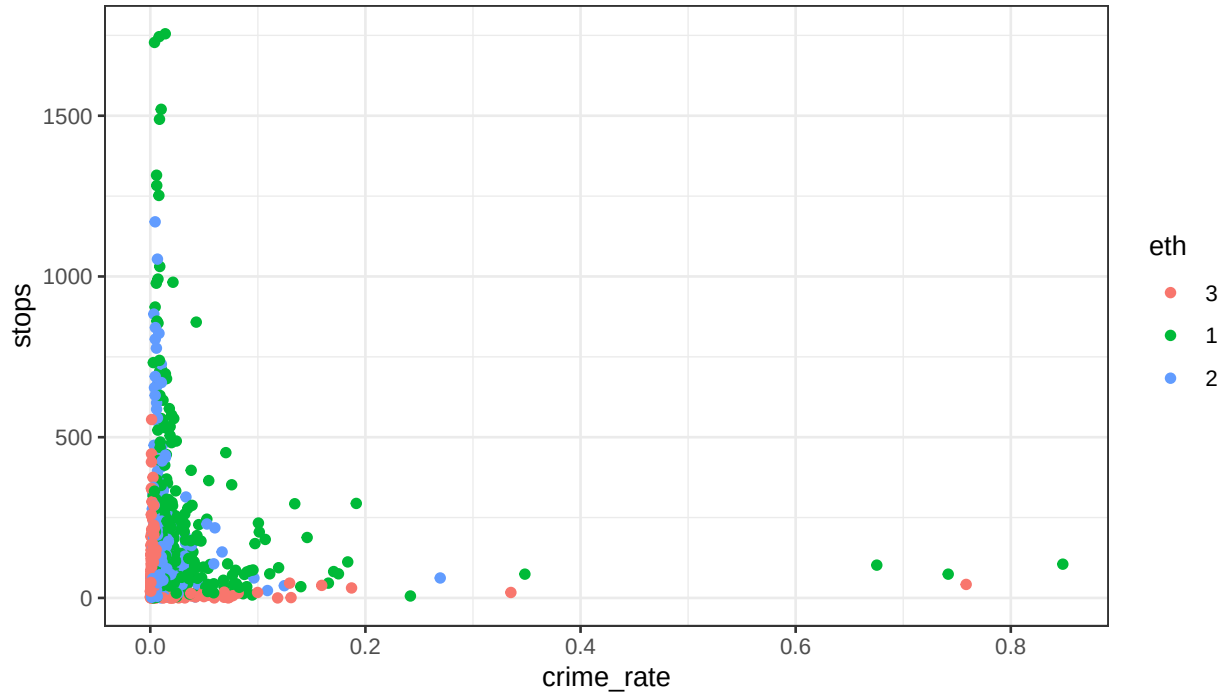
```
#number of stops for each ethnicity vs. population for each ethnicity per precinct

ggplot(data = stop, mapping = aes(x = pop, y = stops, colour = eth)) +geom_point()
```




```
#There does seem to be some trend because we can see that none of the points for
# the white population is in high number of stops despite having very high population
```

```
#crime rate for a certain crime for each eth
stop <- stop %>% mutate(crime_rate = past.arrests/pop)
#num of stops vs crime rate
ggplot(data = stop, mapping = aes(x = crime_rate, y = stops, colour = eth)) +geom_point()
```




```
#crime rate is generally low with a few exceptions; however black and hispanic stop count are still
# a lot higher in low crime rate regions
```

Modeling

Rstan GLM

```
#compute population proportion with in each precinct
stop <- stop %>% group_by(precinct) %>%
  mutate(total_pop = sum(pop/4))
stop <- stop %>% mutate(pop_prop = pop/total_pop)
#get datasets for different crime types
stop.1 <- stop %>% filter(crime == 1)
stop.2 <- stop %>% filter(crime == 2)
stop.3 <- stop %>% filter(crime == 3)
stop.4 <- stop %>% filter(crime == 4)
```

```
#glm for crime 1; done for testing only. don't repeat the code for other crimes yet.
#assuming negative binom sampling model

stan.glm.1 <- stan_glm(data = stop.1,
                   formula = stops ~ crime_rate+ pop_prop+ eth+ crime_rate * eth+ pop_prop*eth + crime_
                   family = neg_binomial_2,
                   seed = 360,
                   prior = cauchy(),
                   prior_intercept = cauchy(),
                   refresh = 0)
```

```
## Warning: There were 3 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
#no r-hat warning = converged. no ess warning = good enough ess. but what to do with divergent transiti
```

```
# first glm but with standard normal prior
stan.glm.2 <- update(stan.glm.1, prior = normal(0,1), prior_intercept = normal(0,1))
# first glm but with poisson family
stan.glm.1.pois <- update(stan.glm.1, family = poisson)
```

```
## Warning: There were 30 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

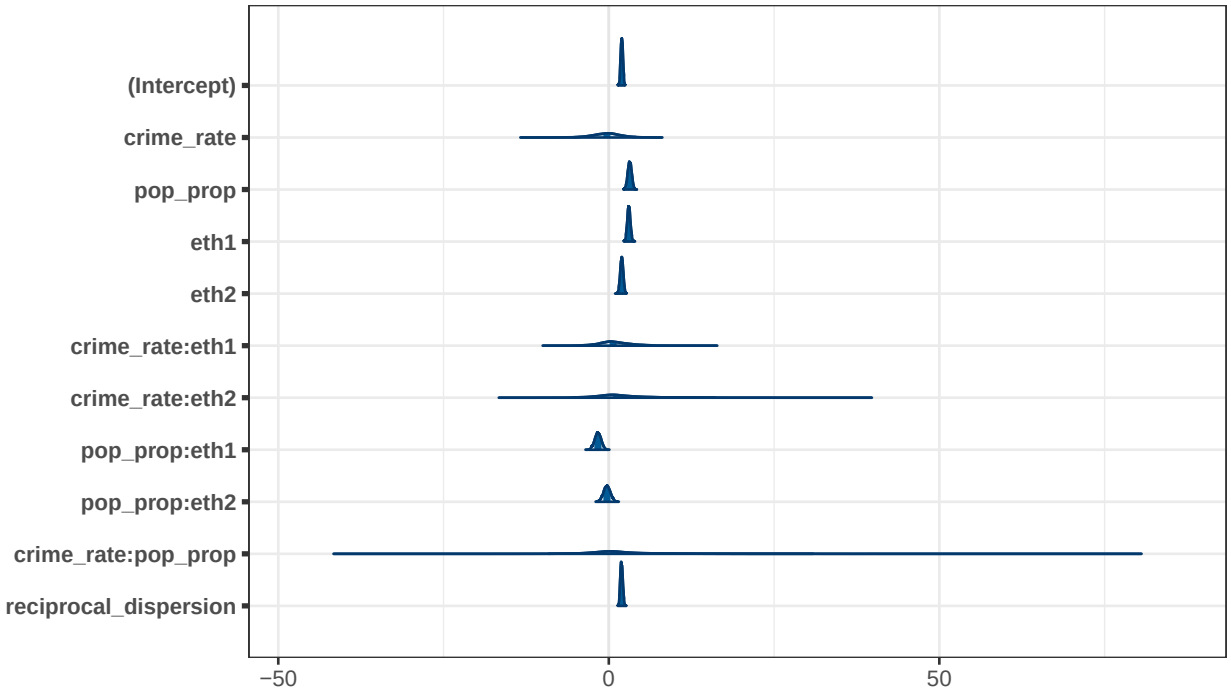
```
# first glm but with standard normal prior & poisson family
stan.glm.2.pois <- update(stan.glm.2, family = poisson )
```

```
#look at coefficients
summary(stan.glm.1)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       neg_binomial_2 [log]
##  formula:      stops ~ crime_rate + pop_prop + eth + crime_rate * eth + pop_prop *
##     eth + crime_rate * pop_prop
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 225
##  predictors:   10
##
## Estimates:
##                          mean   sd   10%   50%   90%
## (Intercept)             2.0    0.2  1.7   2.0   2.2
## crime_rate             -0.4    2.3 -3.1  -0.3   2.4
## pop_prop                3.2    0.3  2.8   3.2   3.5
## eth1                    3.0    0.2  2.7   3.0   3.3
## eth2                    2.0    0.2  1.7   2.0   2.2
## crime_rate:eth1         0.9    2.5 -1.9   0.7   4.0
## crime_rate:eth2         2.3    5.2 -2.5   1.2   8.8
## pop_prop:eth1          -1.7    0.5 -2.3  -1.7  -1.0
## pop_prop:eth2          -0.3    0.5 -1.0  -0.3   0.4
## crime_rate:pop_prop     3.0    9.6 -3.8   0.8  12.3
## reciprocal_dispersion   1.9    0.2  1.7   1.9   2.2
##
```

```
## Fit Diagnostics:
##            mean   sd    10%   50%   90%
## mean_PPD 150.2   14.2 132.7 149.0 168.8
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                          mcse Rhat n_eff
## (Intercept)              0.0  1.0  2409
## crime_rate               0.0  1.0  2476
## pop_prop                 0.0  1.0  2528
## eth1                     0.0  1.0  2123
## eth2                     0.0  1.0  2138
## crime_rate:eth1          0.0  1.0  2562
## crime_rate:eth2          0.1  1.0  2914
## pop_prop:eth1            0.0  1.0  2178
## pop_prop:eth2            0.0  1.0  2621
## crime_rate:pop_prop      0.2  1.0  2895
## reciprocal_dispersion 0.0  1.0  3951
## mean_PPD                 0.2  1.0  4252
## log-posterior            0.0  1.0  1885
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
#posterior distributions
mcmc_areas(as.matrix(stan.glm.1), prob = 0.95, prob_outer = 1)
```




```
round(coef(stan.glm.1), 3)
```

```
##         (Intercept)          crime_rate            pop_prop                eth1
```

```
##               1.967              -0.287               3.161               3.028
##                eth2     crime_rate:eth1     crime_rate:eth2       pop_prop:eth1
##               1.957               0.650               1.183              -1.651
##       pop_prop:eth2 crime_rate:pop_prop
##              -0.283               0.837
```

```
round(posterior_interval(stan.glm.1, prob = 0.95), 3)
```

```
##                          2.5%  97.5%
## (Intercept)             1.641  2.293
## crime_rate             -5.258  4.099
## pop_prop                2.592  3.739
## eth1                    2.586  3.494
## eth2                    1.509  2.397
## crime_rate:eth1        -3.796  6.271
## crime_rate:eth2        -5.287 16.060
## pop_prop:eth1          -2.618 -0.725
## pop_prop:eth2          -1.296  0.743
## crime_rate:pop_prop    -9.121 30.896
## reciprocal_dispersion   1.591  2.300
```

```
#posterior predictive check
loo1 <- loo(stan.glm.1, save_psis = TRUE)
loo2 <- loo(stan.glm.2, save_psis = TRUE)
loo1.pois <- loo(stan.glm.1.pois, save_psis = TRUE)
```

```
## Warning: Found 37 observations with a pareto_k > 0.7. With this many problematic observations we rec
```

```
loo2.pois <- loo(stan.glm.2.pois, save_psis = TRUE)
```

```
## Warning: Found 31 observations with a pareto_k > 0.7. With this many problematic observations we rec
```

```
rstanarm::loo_compare(loo1, loo2, loo1.pois, loo2.pois) #to interpret, need to wait for lab12 answers..
```

```
##                 elpd_diff se_diff
## stan.glm.1          0.0       0.0
## stan.glm.2         -0.9       1.6
## stan.glm.2.pois -5625.0     735.0
## stan.glm.1.pois -5630.1     728.4
```

```
#making sure that bayesian ver of AIC (pareto k) is good
loo1
```

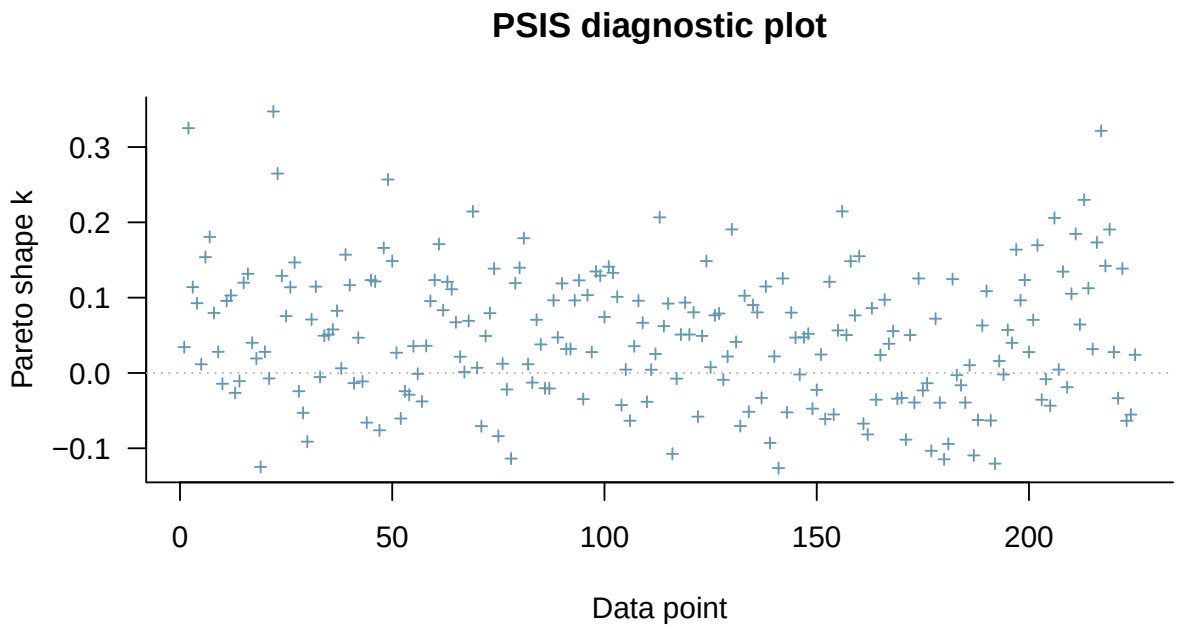
```
##
## Computed from 4000 by 225 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1229.3 19.5
## p_loo         7.4  0.8
## looic      2458.5 39.0
```

```
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

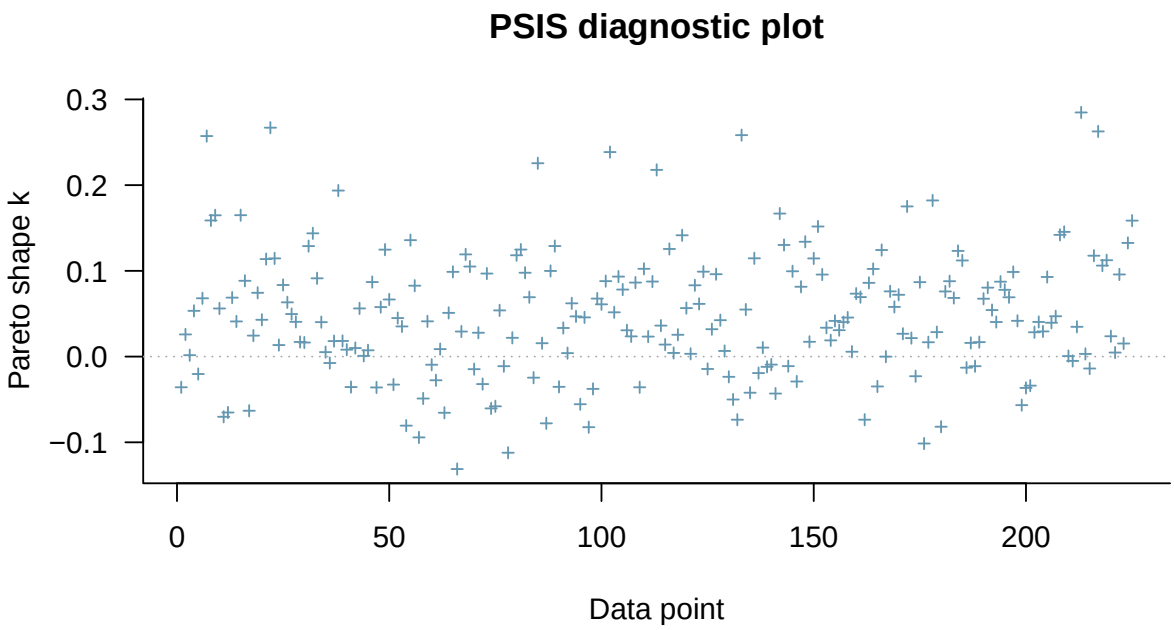
```
loo2
```

```
##
## Computed from 4000 by 225 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1230.2 19.4
## p_loo         6.4  0.7
## looic      2460.4 38.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

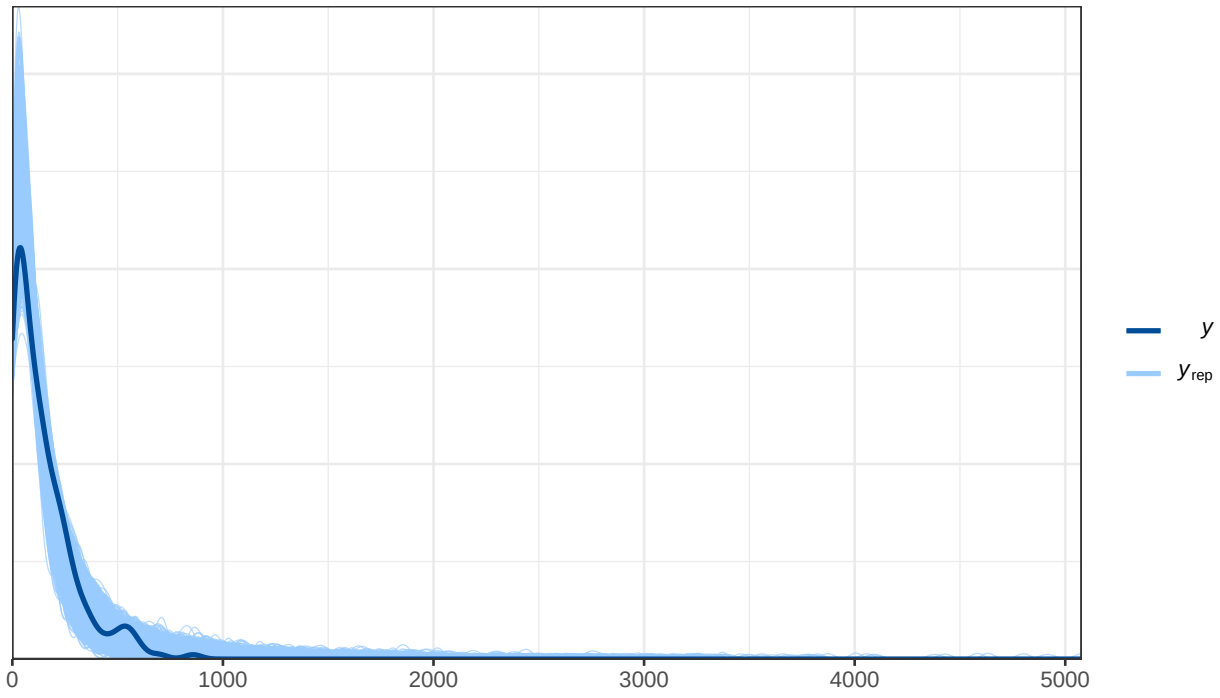
```
#check for outliers (if there are outliers, then post pred will be sensative to 1 observation)
plot(loo1, label_points = TRUE)
```



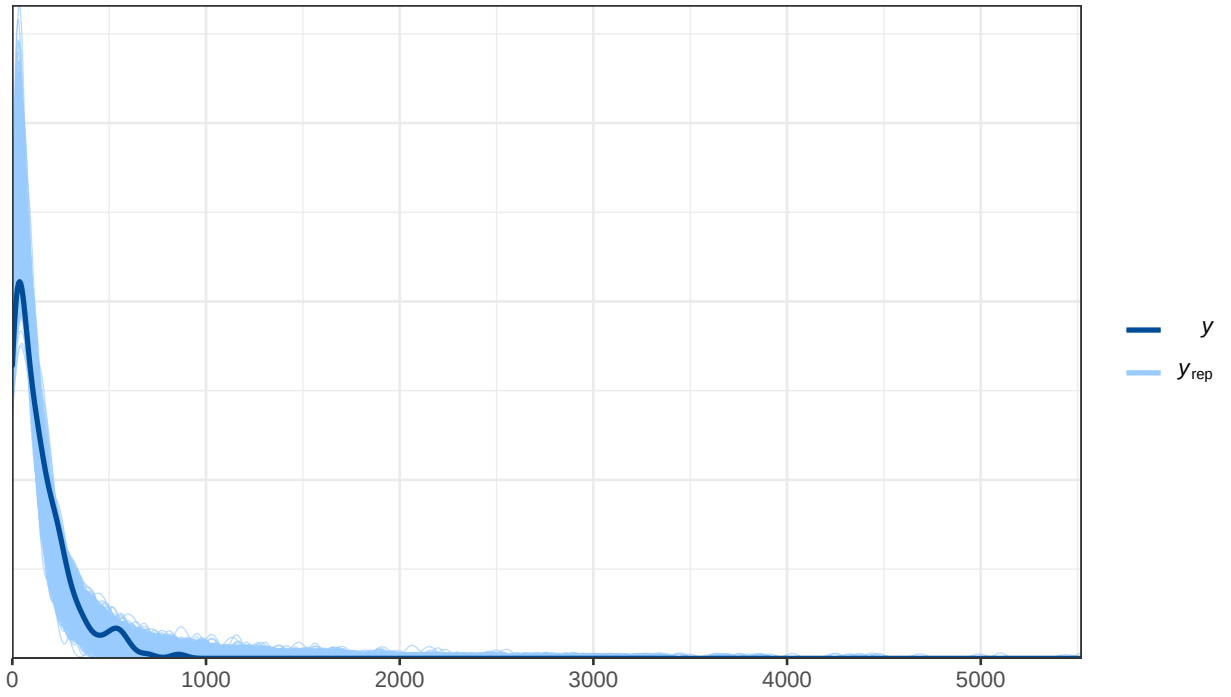

```
plot(loo2, label_points = TRUE)
```



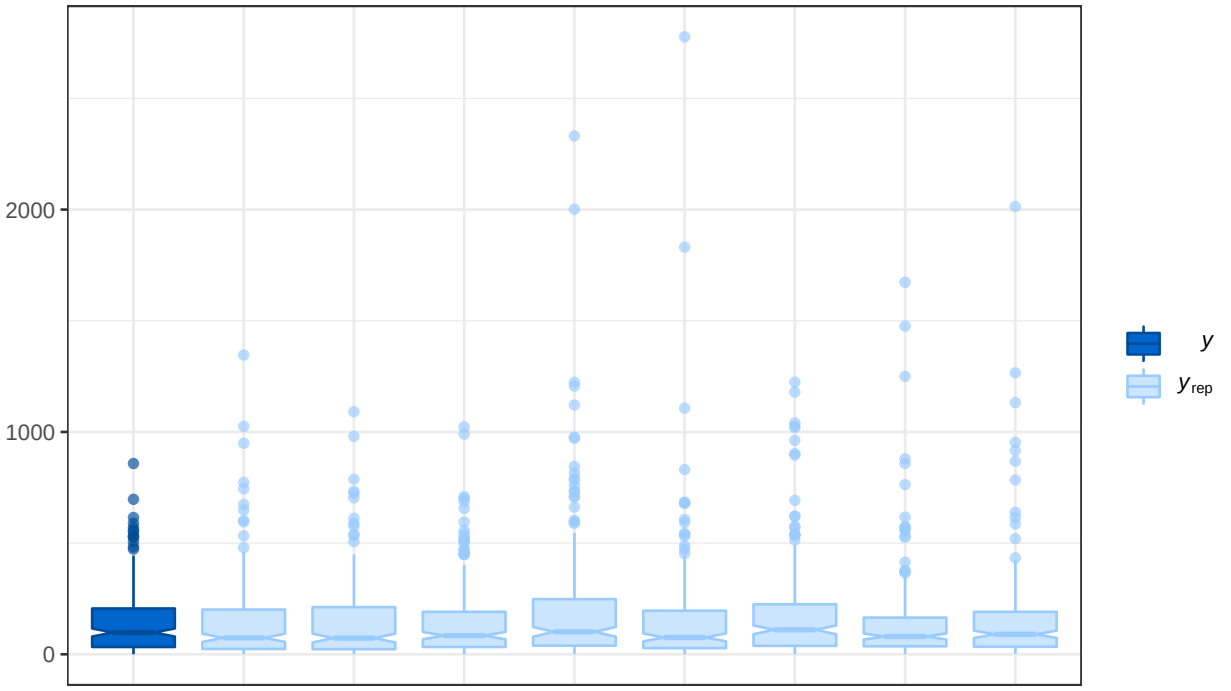

```
#individual model diagnostics - stan.glm.1
y.postpred <- posterior_predict(stan.glm.1)
color_scheme_set("brightblue")
ppc_dens_overlay(stop.1$stops, y.postpred)
```



```
#individual model diagnostics - stan.glm.2
y.postpred <- posterior_predict(stan.glm.2)
color_scheme_set("brightblue")
ppc_dens_overlay(stop.1$stops, y.postpred)
```

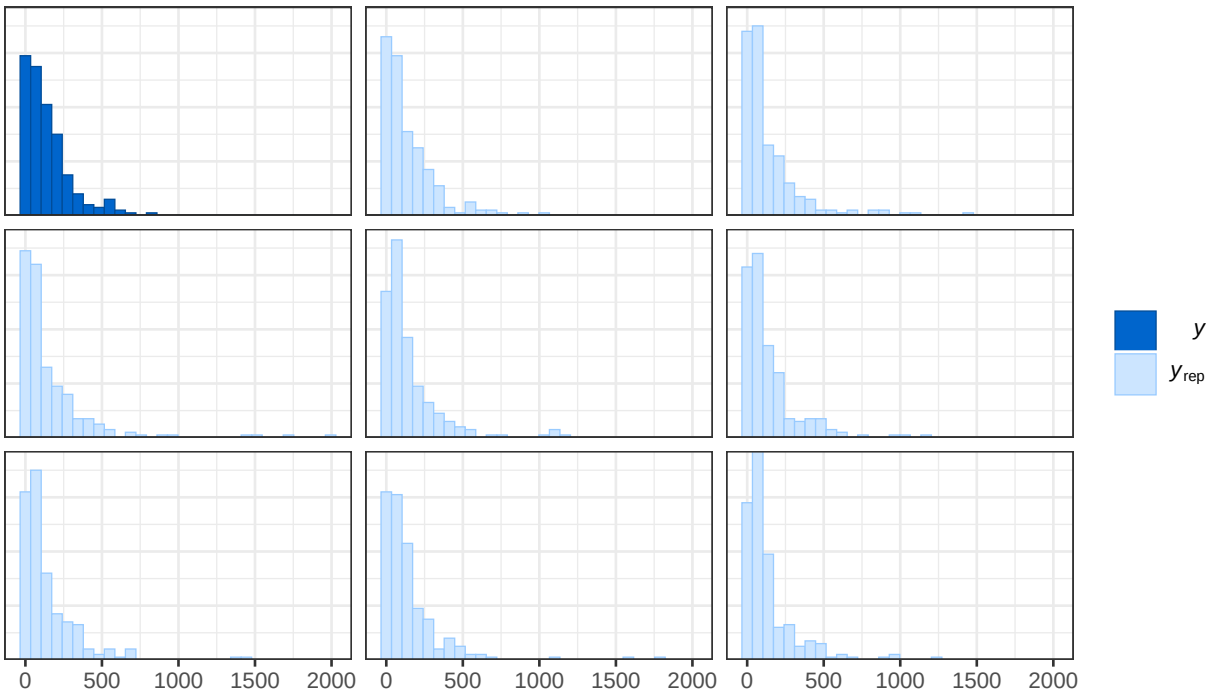



```
pp_check(stan.glm.1, plotfun = "boxplot")
```

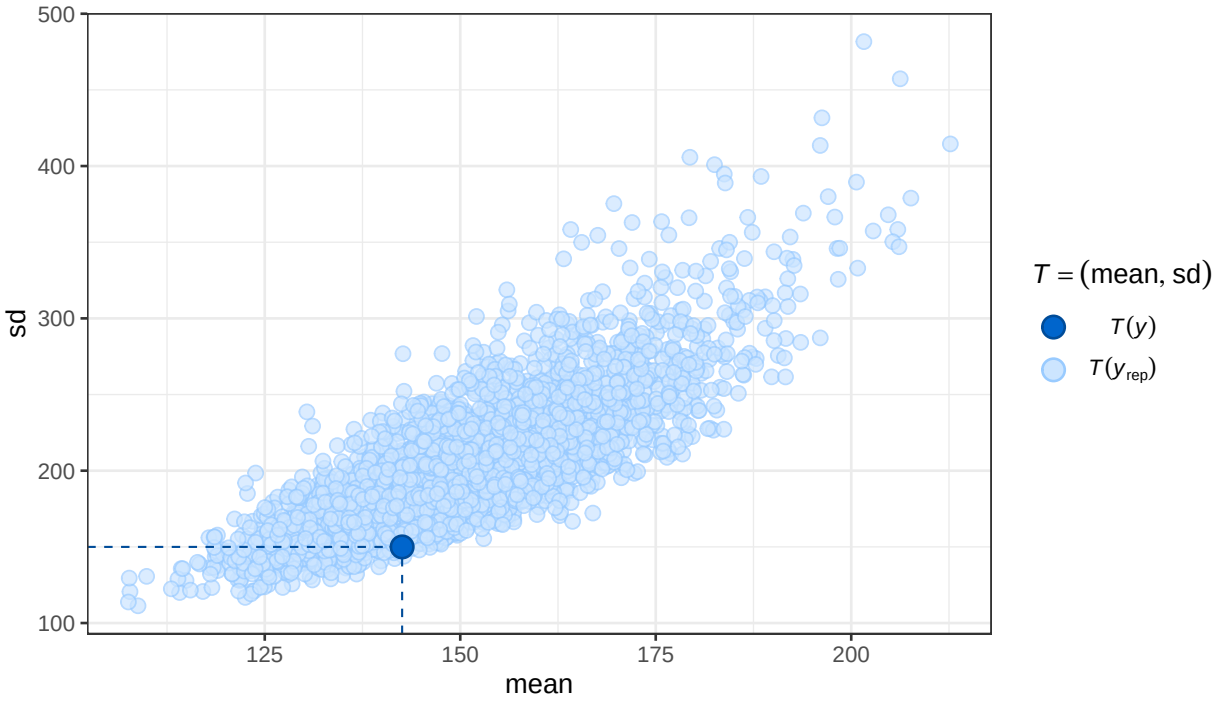


```
pp_check(stan.glm.1, plotfun = "hist")
```

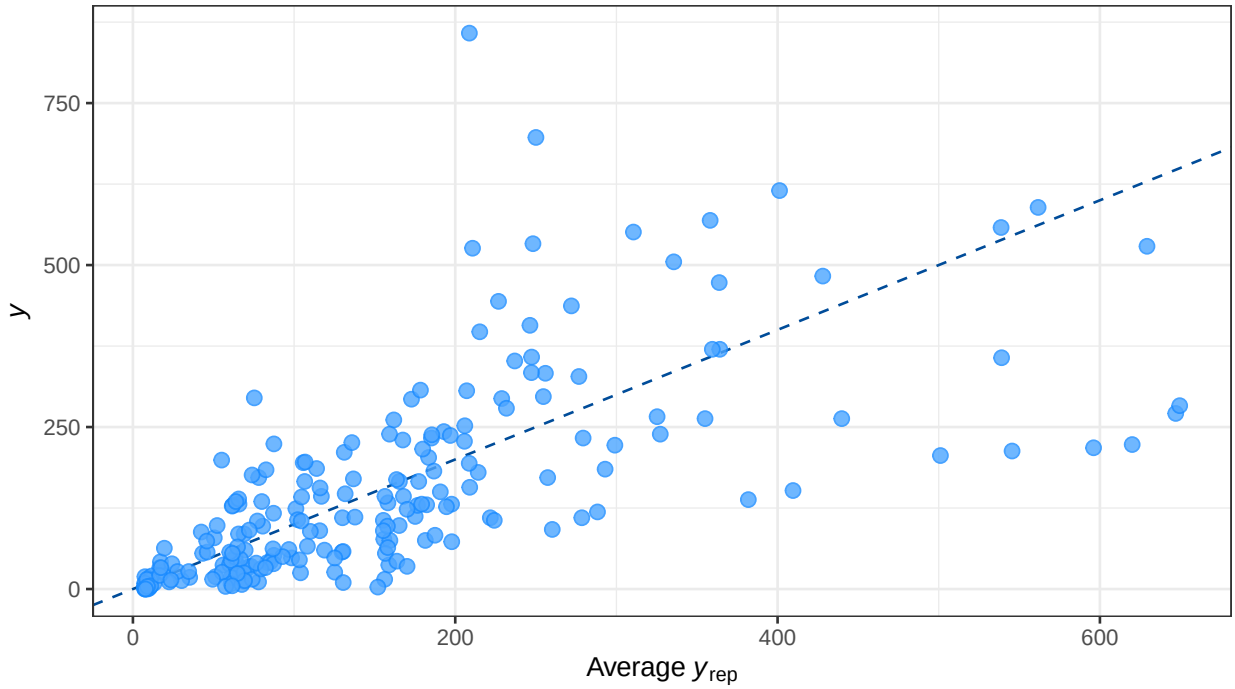
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



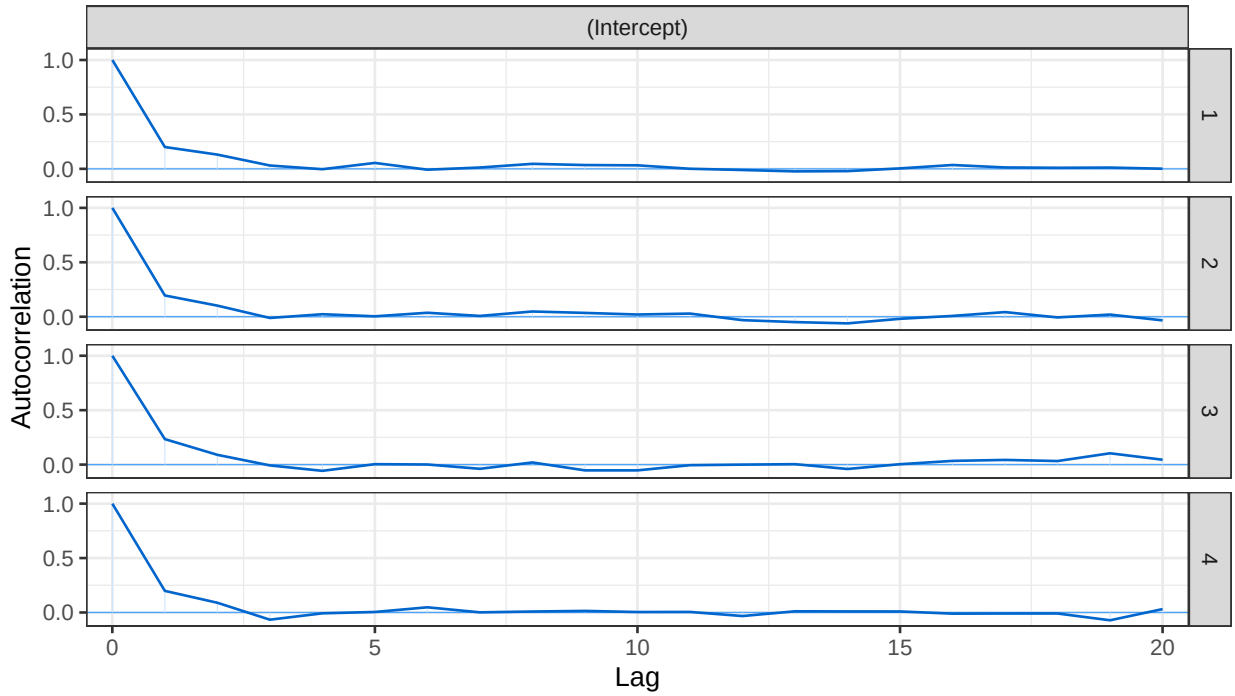

```
pp_check(stan.glm.1, plotfun = "stat_2d", stat = c("mean", "sd"))  # Scatterplot of two test statistics
```



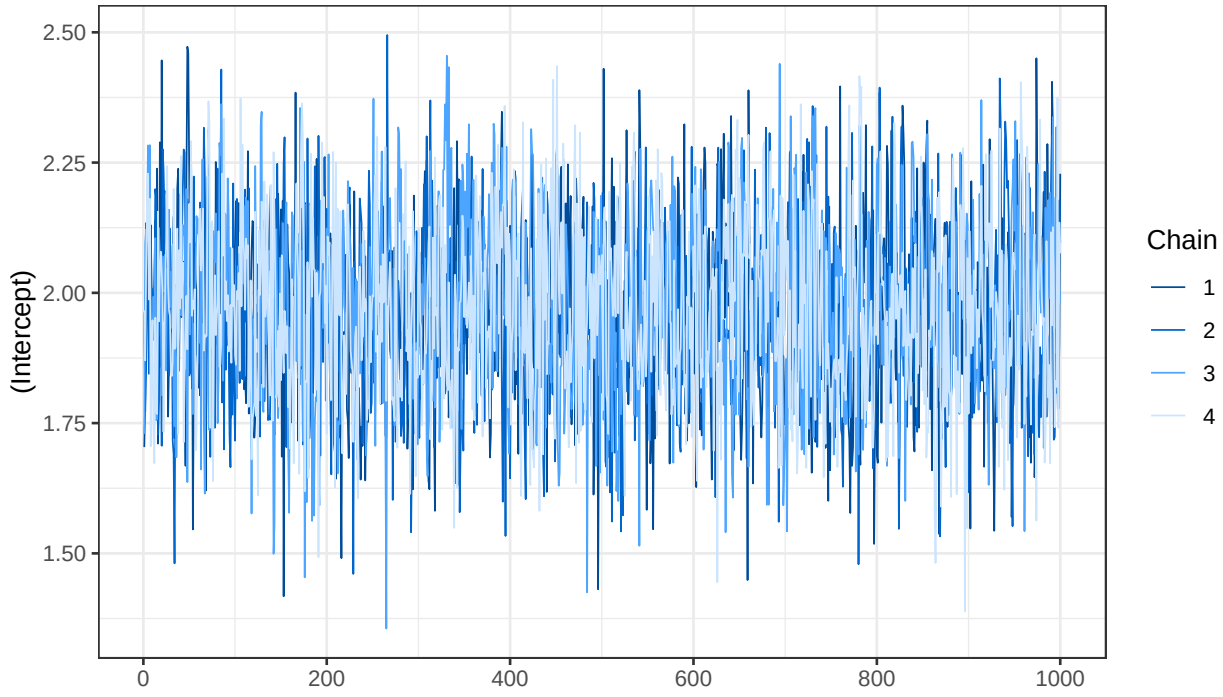
```
pp_check(stan.glm.1, plotfun = "scatter_avg") # Scatterplot of y vs. average yrep
```




```
plot(stan.glm.1, "acf", pars = "(Intercept)")# autocorrelation by chain
```



```
plot(stan.glm.1, "trace", pars = "(Intercept)") #traceplot. how to separate by chain?
```




```
pp_check(stan.glm.2, plotfun = "stat_2d", stat = c("mean", "sd"))
```