

360 Final Project

Group: Flora Shi, Belle Xu

21 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))

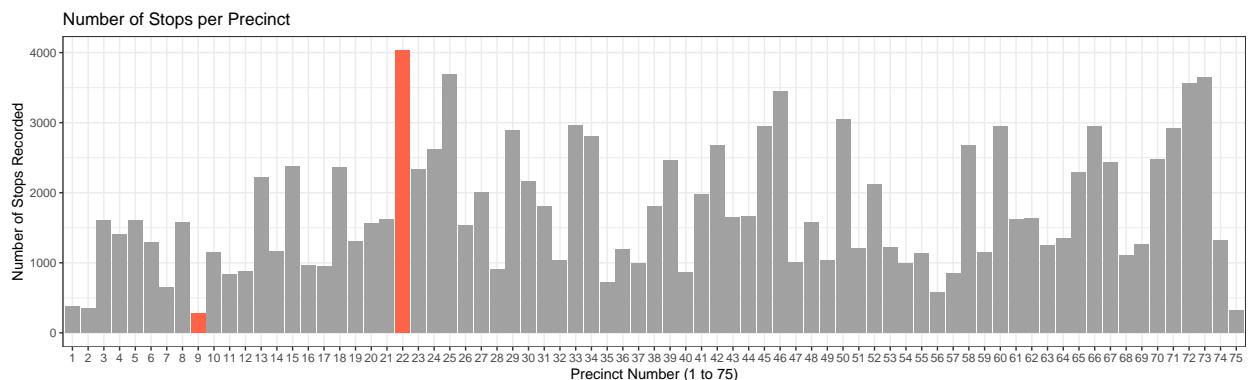
#crime rate for a certain crime for each eth in a certain precinct
stop <- stop %>% mutate(crime_rate = past.arrests/pop)
#compute population proportion with in each precinct
stop <- stop %>% group_by(precinct) %>%
  mutate(total_pop = sum(pop/4))
stop <- stop %>% mutate(pop_prop = pop/total_pop)

#make a by-precinct dataframe
stop <- stop %>%
  group_by(precinct) %>%
  mutate(stops_in_precinct = sum(stops))

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

by_precinct_df <- by_precinct_df %>%
  mutate(toHighlight = if_else(stops_by_precinct == min(stops_by_precinct),
                             | stops_by_precinct == max(stops_by_precinct),
                             "yes", "no"))
```

```
#number of stops by precinct
ggplot(data = by_precinct_df,
       mapping = aes(x = precinct, y = stops_by_precinct, fill = toHighlight)) +
  geom_col() +
  scale_fill_manual(values= c("yes"="tomato", "no"="#a1a1a1"), guide = FALSE) +
  labs(title = "Number of Stops per Precinct", x = "Precinct Number (1 to 75)",
       y = "Number of Stops Recorded")
```



```
#Need interpretation
# get the max and min stops with function and just type it
min(by_precinct_df$stops_by_precinct)
```

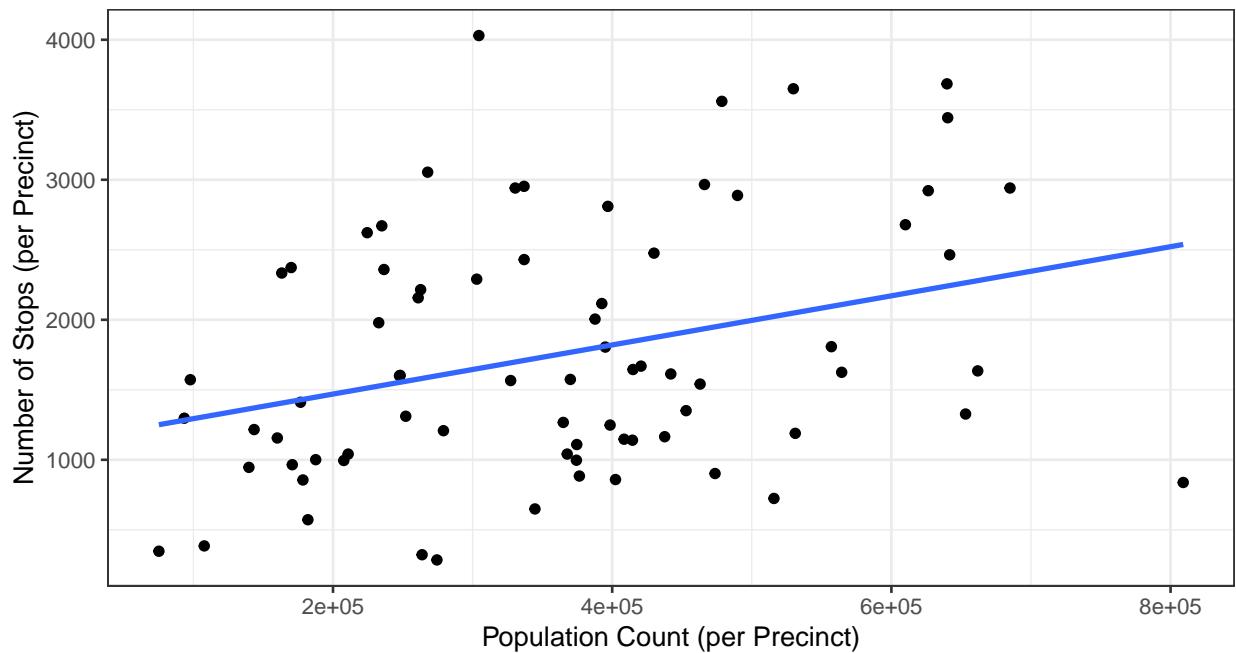
```
## [1] 285
```

```
max(by_precinct_df$stops_by_precinct)
```

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

```
#number of stops vs. precinct pop
ggplot(data = by_precinct_df,
       mapping = aes(x = pop_by_precinct, y = stops_by_precinct)) +
  geom_point() +
  labs(title = "Number of Stops vs. Population per Precinct",
       x = "Population Count (per Precinct)",
       y = "Number of Stops (per Precinct)") +
  geom_smooth(method = "lm", formula = y~x, se = FALSE)
```

Number of Stops vs. Population per Precinct

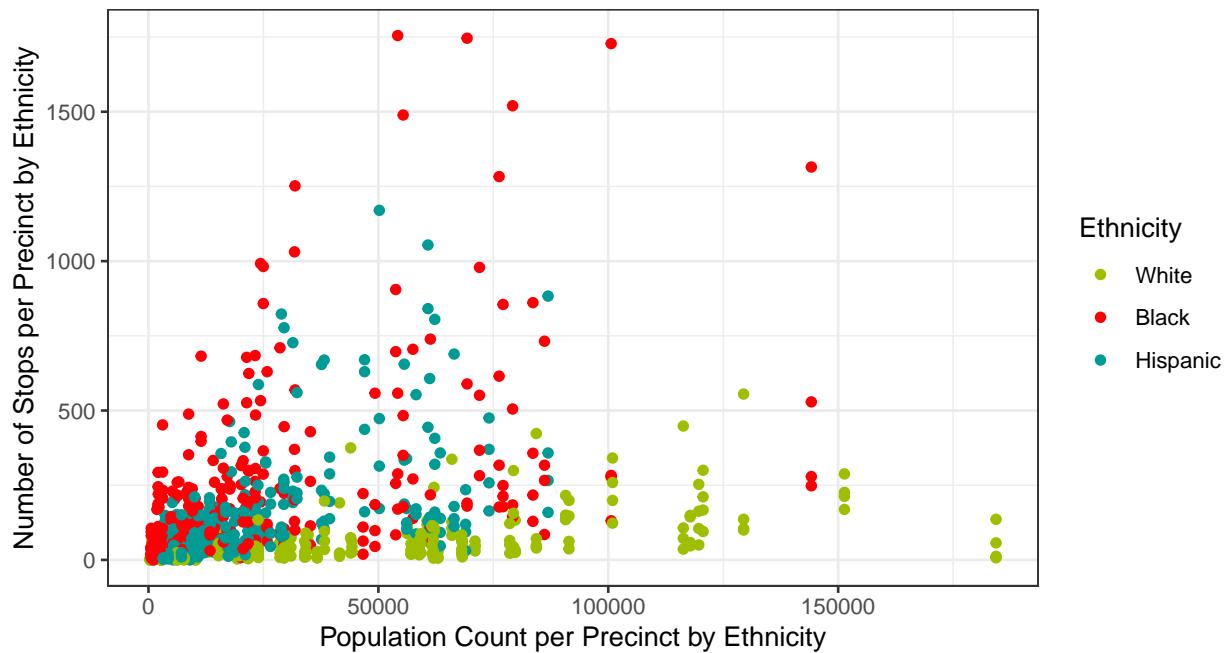


```
#no obvious trend between population in precinct vs. stops in a precinct
#maybe weakly positive
```

```
#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() +
  labs(title = "Number of Stops vs. Population per Precinct by Ethnicity",
       x = "Population Count per Precinct by Ethnicity",
       y = "Number of Stops per Precinct by Ethnicity",
       colour = "Ethnicity") +
  scale_color_manual(values = c("#9EBE00", "#FD0006", "#009B95"),
                     labels = c("White", "Black", "Hispanic"))
```

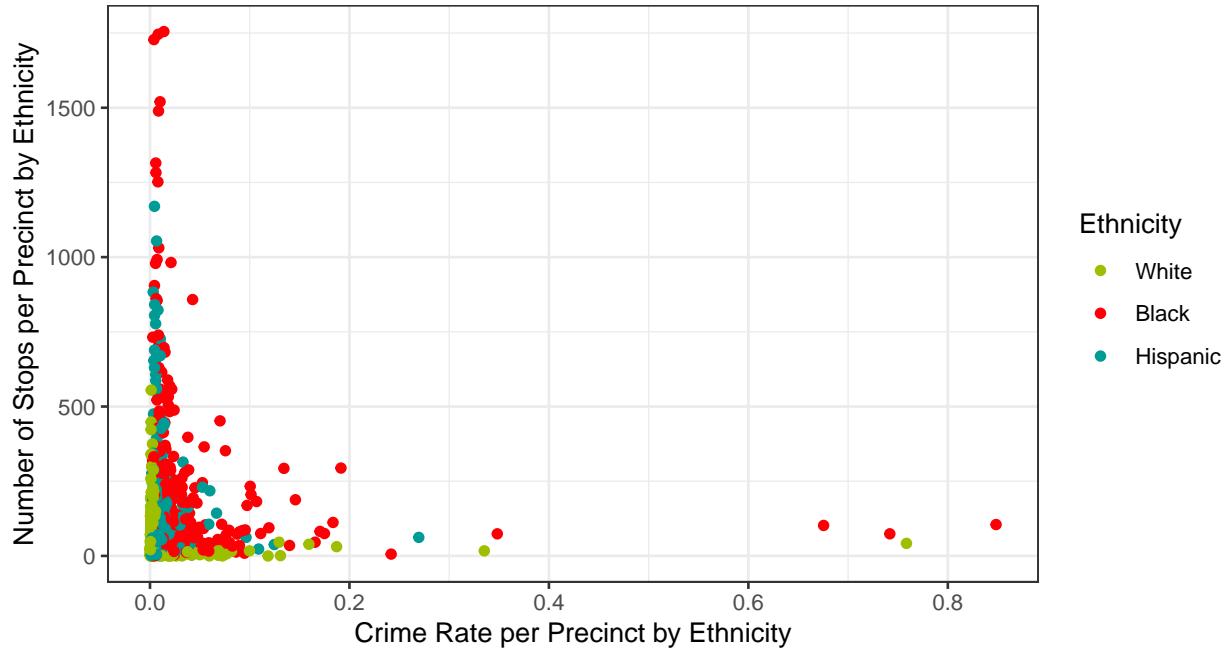
Number of Stops vs. Population per Precinct by Ethnicity



```
#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
```

```
#num of stops vs crime rate
ggplot(data = stop, mapping = aes(x = crime_rate, y = stops, colour = eth)) +
  geom_point() +
  labs(title = "Number of Stops vs. Crime Rate in the Past Year per Precinct by Ethnicity",
       x = "Crime Rate per Precinct by Ethnicity",
       y = "Number of Stops per Precinct by Ethnicity",
       colour = "Ethnicity") +
  scale_color_manual(values = c("#9EBE00", "#FD0006", "#009B95"),
                     labels = c("White", "Black", "Hispanic"))
```

Number of Stops vs. Crime Rate in the Past Year per Precinct by Ethnicity



```
#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
```

```
#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)
#investigate mean and var of each stop
Violence <- c(mean(stop.1$stops), var(stop.1$stops))
Weapon <- c(mean(stop.2$stops), var(stop.2$stops))
Property <- c(mean(stop.3$stops), var(stop.3$stops))
Drug <- c(mean(stop.4$stops), var(stop.4$stops))
df <- data.frame(Violence, Weapon, Property, Drug)
row.names(df) <- c("Mean", "Variance")
df
```

	Violence	Weapon	Property	Drug
## Mean	142.5689	256.8356	117.9822	66.70222
## Variance	22489.0946	123810.0309	18919.3479	4978.23683

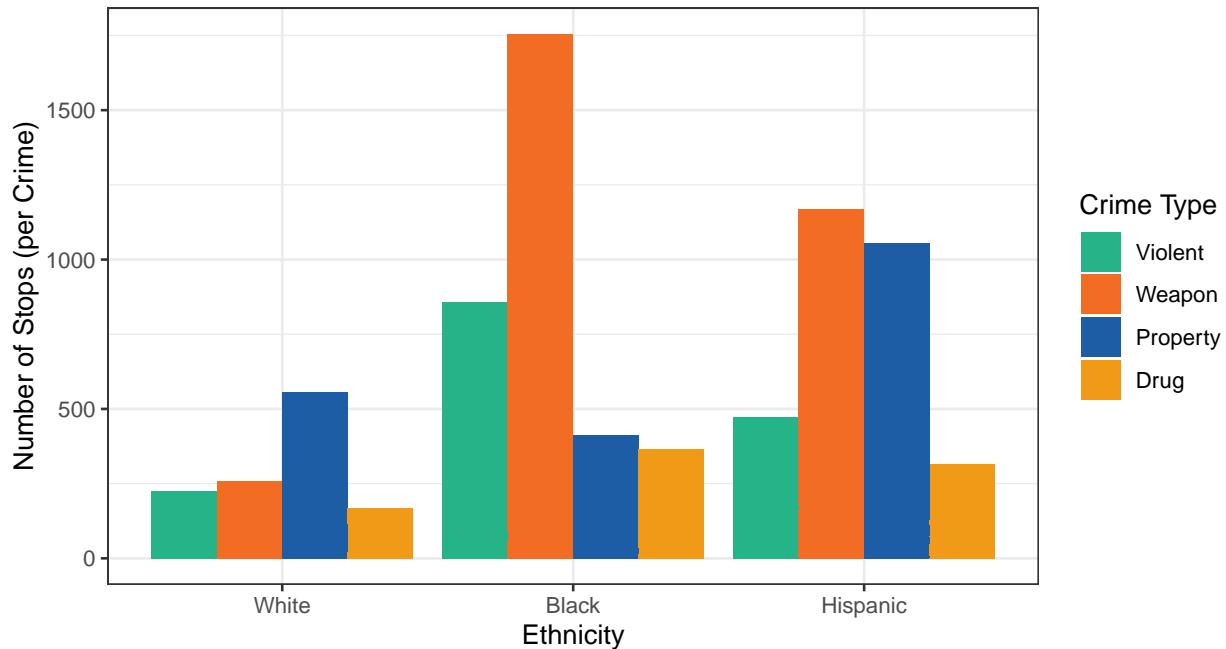
```
#number of stops per crime for each ethnicity
ggplot(data = stop, mapping = aes(x = eth, y = stops, fill = crime)) +
  geom_col(position = "dodge") +
  labs(title = "Number of Stops per Crime for each Ethnicity",
       x = "Ethnicity",
       y = "Number of Stops (per Crime)",
       fill = "Crime Type") +
  scale_x_discrete(labels=c("3" = "White", "1" = "Black",
```

```

    "2" = "Hispanic")) +
scale_fill_manual(values = c("#25B388", "#F26C24", "#1C5DA6", "#F09A18"),
                 labels = c("Violent", "Weapon",
                           "Property", "Drug"))

```

Number of Stops per Crime for each Ethnicity

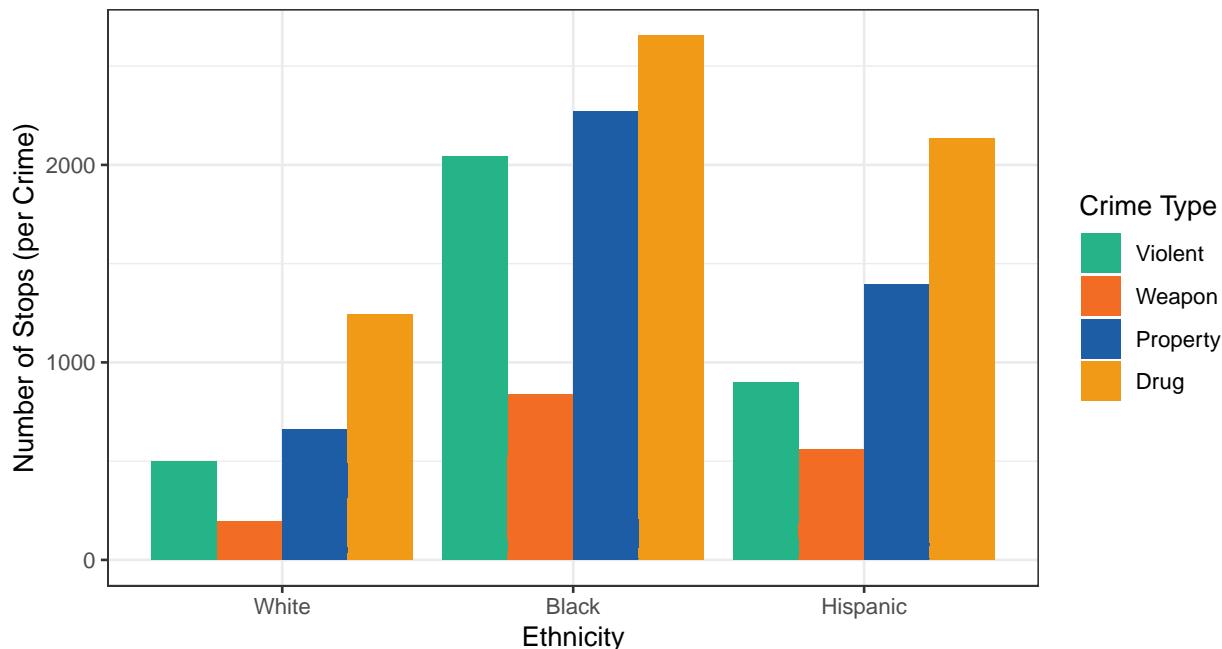


```

#number of past arrests per crime for each ethnicity
ggplot(data = stop, mapping = aes(x = eth, y = past.arrests, fill = crime)) +
  geom_col(position = "dodge") +
  labs(title = "Number of Past Arrests per Crime for each Ethnicity",
       x = "Ethnicity",
       y = "Number of Stops (per Crime)",
       fill = "Crime Type") +
  scale_x_discrete(labels=c("3" = "White", "1" = "Black",
                           "2" = "Hispanic")) +
  scale_fill_manual(values = c("#25B388", "#F26C24", "#1C5DA6", "#F09A18"),
                 labels = c("Violent", "Weapon",
                           "Property", "Drug"))

```

Number of Past Arrests per Crime for each Ethnicity



```
#from the two plots above, the stop reasons and past arrests seem to have no relation
```

Modeling

Rstan GLM

```
#clean datasets and add interaction
stop.1<- stop.1 %>% ungroup() %>%
  dplyr::select(-crime, -stops_in_precinct,
                -total_pop, -precinct) %>%
  mutate(eth = as.numeric(eth)) %>%
  mutate(int_pop_pastarrests = pop*past.arrests,
         int_pop_eth = pop*eth,
         int_pastarrests_eth = past.arrests*eth,
         int_pastarrests_popprop = past.arrests*pop_prop,
         int_crimerate_eth = crime_rate*eth, #1
         int_popprop_eth = pop_prop*eth, #2
         int_crimerate_popprop = crime_rate*pop_prop) %>%
  mutate(eth = factor(eth))
```

```
stop.1.model1 <- stop.1 %>%
  dplyr::select()
```

```
#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_rate:pop_prop,
                        family = neg_binomial_2,
                        seed = 360,
                        prior = cauchy(0, 2.5),
                        prior_intercept = cauchy(0, 2.5),
                        refresh = 0,
                        diagnostic_file = file.path(tempdir(), "glm.csv"))
```

Violent Crimes

```
## 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. apparently divergent transitions can b

#checking if adding/changing stuff will make the model better
stan.glm.2 <- stan_glm(data = stop.1,
                        formula = stops ~ past.arrests+ pop_prop+ eth+
                        past.arrests:eth+ pop_prop:eth + past.arrests:pop_prop,
                        family = neg_binomial_2, #how do I justify not having a prior?
                        seed = 360,
                        refresh = 0,
                        diagnostic_file = file.path(tempdir(), "glm.csv"))
stan.glm.3 <- update(stan.glm.2, formula = stops ~ past.arrests+ pop+ eth+
                        past.arrests:eth+ pop:eth + past.arrests:pop,
                        diagnostic_file = file.path(tempdir(), "glm.csv"))
```

```
#bayes factor backward selection
removeCovariate <- function(df, remove) {
  #remove one covariate from model1
  df.temp <- df %>% dplyr::select(-remove)
  #specify new model which omits the removed variable
  model <- stan_glm(data = df.temp,
                     formula = stops ~ ., #how do we add interaction?
                     family = neg_binomial_2, #how do I justify not having a prior?
                     seed = 360,
                     refresh = 0,
                     diagnostic_file = file.path(tempdir(), "glm.csv"))
  return(model)
}
bfSelection <- function() {

}
```

```
set.seed(360)
bridge1 <- bridge_sampler(stan.glm.1)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
## Iteration: 11
## Iteration: 12
## Iteration: 13
## Iteration: 14
## Iteration: 15
## Iteration: 16
## Iteration: 17
## Iteration: 18
## Iteration: 19
## Iteration: 20
## Iteration: 21
## Iteration: 22
## Iteration: 23
## Iteration: 24
## Iteration: 25
## Iteration: 26
## Iteration: 27
## Iteration: 28
## Iteration: 29
## Iteration: 30
## Iteration: 31
## Iteration: 32
## Iteration: 33
## Iteration: 34
## Iteration: 35
## Iteration: 36
## Iteration: 37
## Iteration: 38
## Iteration: 39
## Iteration: 40
## Iteration: 41
## Iteration: 42
## Iteration: 43
## Iteration: 44
## Iteration: 45
## Iteration: 46
## Iteration: 47
## Iteration: 48
## Iteration: 49
## Iteration: 50
```

```
## Iteration: 51
## Iteration: 52
## Iteration: 53
## Iteration: 54
## Iteration: 55
## Iteration: 56
## Iteration: 57
## Iteration: 58
## Iteration: 59
## Iteration: 60
## Iteration: 61
## Iteration: 62
## Iteration: 63
## Iteration: 64
## Iteration: 65
## Iteration: 66
## Iteration: 67
## Iteration: 68
## Iteration: 69
## Iteration: 70
## Iteration: 71
## Iteration: 72
## Iteration: 73
## Iteration: 74
## Iteration: 75
## Iteration: 76
## Iteration: 77
## Iteration: 78
## Iteration: 79
## Iteration: 80
## Iteration: 81
## Iteration: 82
## Iteration: 83
## Iteration: 84
## Iteration: 85
## Iteration: 86
## Iteration: 87
## Iteration: 88
## Iteration: 89
## Iteration: 90
## Iteration: 91
## Iteration: 92
## Iteration: 93
## Iteration: 94
## Iteration: 95
## Iteration: 96
## Iteration: 97
## Iteration: 98
## Iteration: 99
## Iteration: 100
## Iteration: 101
## Iteration: 102
## Iteration: 103
## Iteration: 104
```

```
## Iteration: 105
## Iteration: 106
## Iteration: 107
## Iteration: 108
## Iteration: 109
## Iteration: 110
## Iteration: 111
## Iteration: 112
## Iteration: 113
## Iteration: 114
## Iteration: 115
## Iteration: 116
## Iteration: 117
## Iteration: 118
## Iteration: 119
## Iteration: 120
## Iteration: 121
## Iteration: 122
## Iteration: 123
## Iteration: 124
## Iteration: 125
## Iteration: 126
## Iteration: 127
## Iteration: 128
## Iteration: 129
## Iteration: 130
## Iteration: 131
## Iteration: 132
## Iteration: 133
## Iteration: 134
## Iteration: 135
## Iteration: 136
## Iteration: 137
## Iteration: 138
## Iteration: 139
## Iteration: 140
## Iteration: 141
## Iteration: 142
## Iteration: 143
## Iteration: 144
## Iteration: 145
## Iteration: 146
## Iteration: 147
## Iteration: 148
## Iteration: 149
## Iteration: 150
## Iteration: 151
## Iteration: 152
## Iteration: 153
## Iteration: 154
## Iteration: 155
## Iteration: 156
## Iteration: 157
## Iteration: 158
```

```
## Iteration: 159
## Iteration: 160
## Iteration: 161
## Iteration: 162
## Iteration: 163
## Iteration: 164
## Iteration: 165
## Iteration: 166
## Iteration: 167
## Iteration: 168
## Iteration: 169
## Iteration: 170
## Iteration: 171
## Iteration: 172
## Iteration: 173
## Iteration: 174
## Iteration: 175
## Iteration: 176
## Iteration: 177
## Iteration: 178
## Iteration: 179
## Iteration: 180
## Iteration: 181
## Iteration: 182
## Iteration: 183
## Iteration: 184
## Iteration: 185
## Iteration: 186
## Iteration: 187
## Iteration: 188
## Iteration: 189
## Iteration: 190
## Iteration: 191
## Iteration: 192
## Iteration: 193
## Iteration: 194
## Iteration: 195
## Iteration: 196
## Iteration: 197
## Iteration: 198

bridge2 <- bridge_sampler(stan.glm.2)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
## Iteration: 11
```

```

## Iteration: 12
## Iteration: 13
## Iteration: 14
## Iteration: 15
## Iteration: 16
## Iteration: 17
## Iteration: 18
## Iteration: 19
## Iteration: 20
## Iteration: 21
## Iteration: 22
## Iteration: 23
## Iteration: 24
## Iteration: 25
## Iteration: 26
## Iteration: 27
## Iteration: 28
## Iteration: 29

bridge3 <- bridge_sampler(stan.glm.3)

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5

a <- bf(bridge2, bridge1)
b <- bf(bridge2, bridge3)
c <- bf(bridge3, bridge2)
a$bf

## [1] 6.554613e+48

b$bf

## [1] 814.2705

c$bf

## [1] 0.001228093

#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 +

```

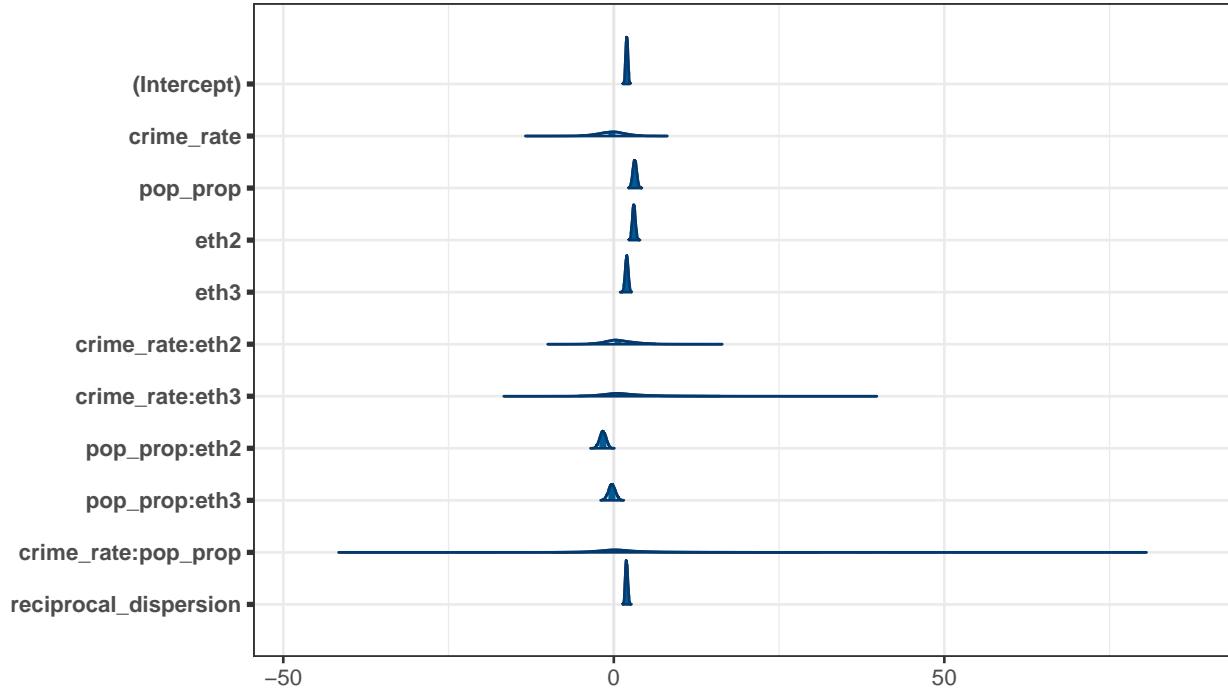
```

##      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
## eth2        3.0  0.2  2.7  3.0  3.3
## eth3        2.0  0.2  1.7  2.0  2.2
## crime_rate:eth2 0.9  2.5 -1.9  0.7  4.0
## crime_rate:eth3 2.3  5.2 -2.5  1.2  8.8
## pop_prop:eth2 -1.7  0.5 -2.3 -1.7 -1.0
## pop_prop:eth3 -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
## eth2       0.0  1.0 2123
## eth3       0.0  1.0 2138
## crime_rate:eth2 0.0  1.0 2562
## crime_rate:eth3 0.1  1.0 2914
## pop_prop:eth2 0.0  1.0 2178
## pop_prop:eth3 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       eth2
##            1.967        -0.287        3.161      3.028
##             eth3      crime_rate:eth2  crime_rate:eth3  pop_prop:eth2
##            1.957         0.650         1.183      -1.651
##   pop_prop:eth3 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
## eth2	2.586	3.494
## eth3	1.509	2.397
## crime_rate:eth2	-3.796	6.271
## crime_rate:eth3	-5.287	16.060
## pop_prop:eth2	-2.618	-0.725
## pop_prop:eth3	-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)
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argument
```

```

loo3 <- loo(stan.glm.3, save_psis = TRUE)

## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argument
## rstanarm::loo_compare(loo1, loo2, loo3) #loo2 is the best.

##          elpd_diff se_diff
## stan.glm.2    0.0      0.0
## stan.glm.3 -13.2      5.0
## stan.glm.1 -31.7     7.4

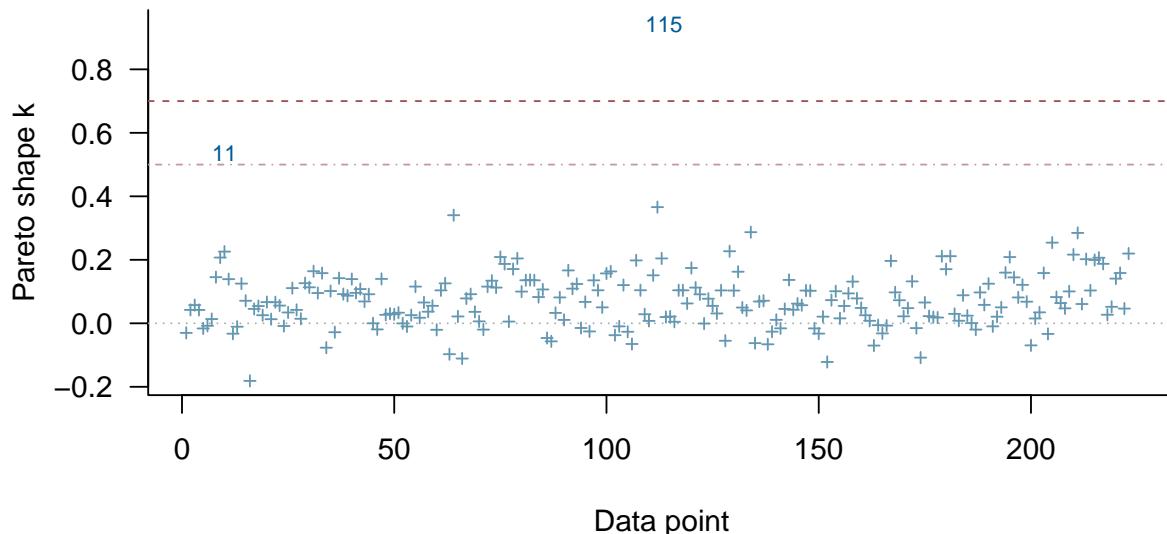
#making sure that bayesian ver of AIC (pareto k) is good
loo1

## 
## 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.

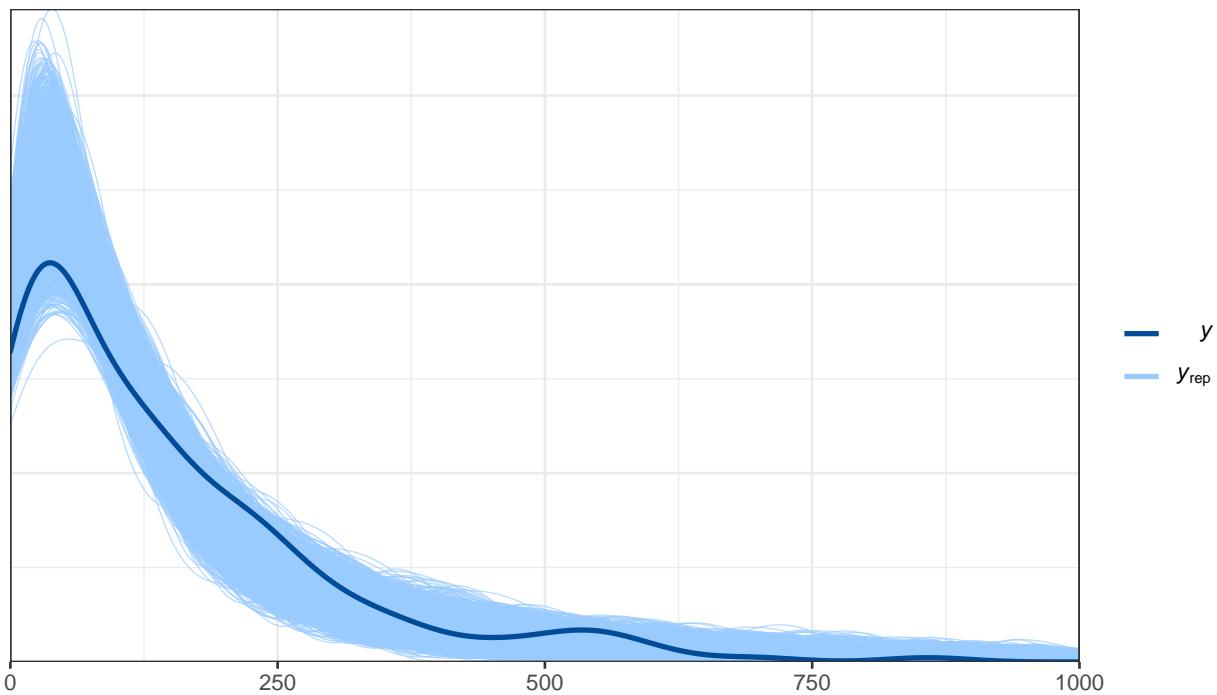
#check for outliers (if there are outliers, then post pred will be sensitive to 1 observation)
plot(loo2, label_points = TRUE)

```

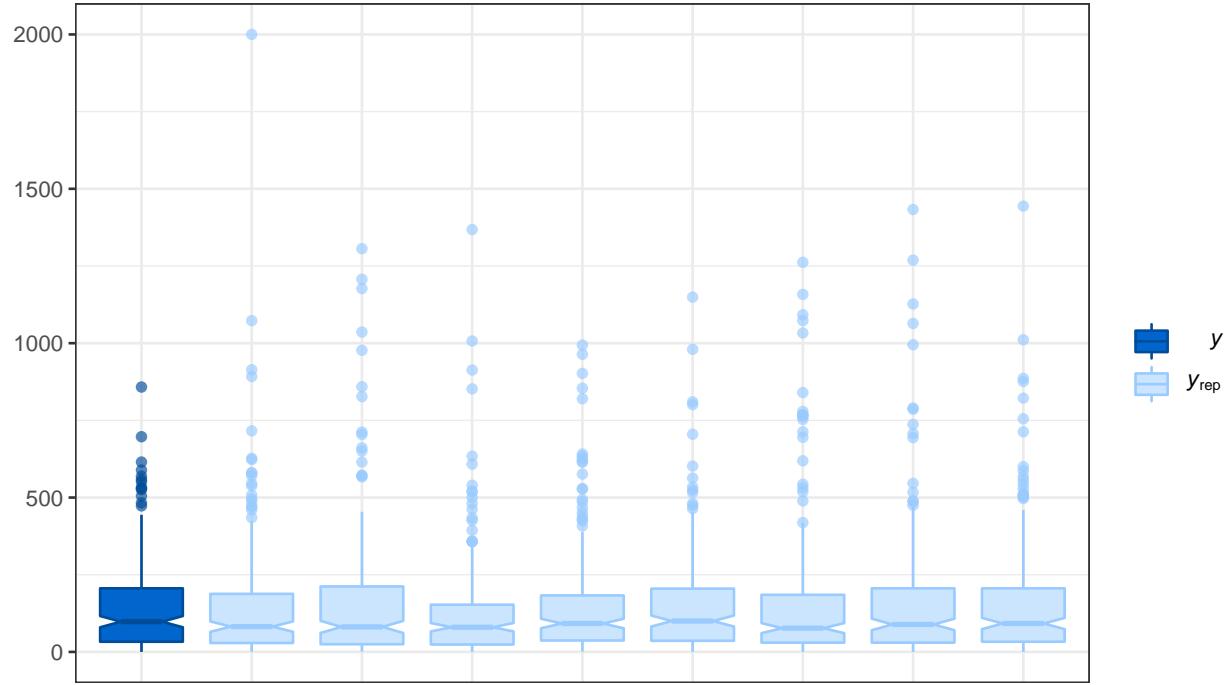
PSIS diagnostic plot



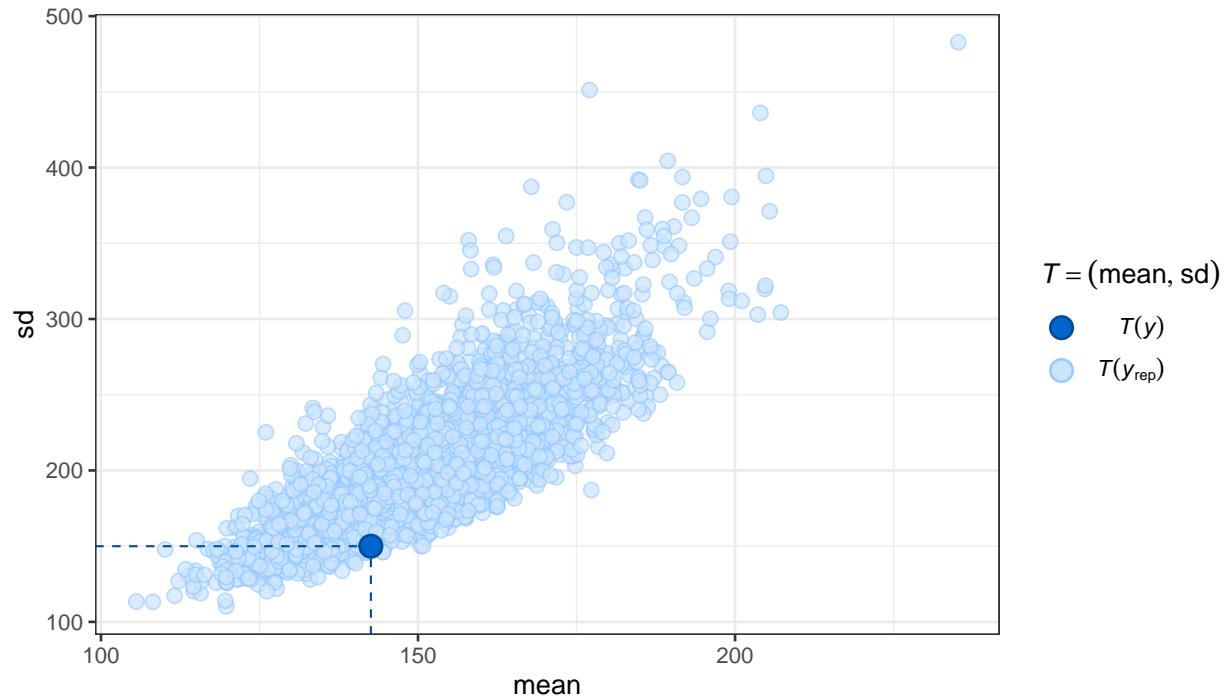
```
#individual model diagnostics - stan.glm.1
y.postpred <- posterior_predict(stan.glm.1)
color_scheme_set("brightblue")
ppc_dens_overlay(stop.1$stops, y.postpred) + xlim(0, 1000) #xlim() truncates so that we focus on the pa
## Warning: Removed 8640 rows containing non-finite values (stat_density).
```



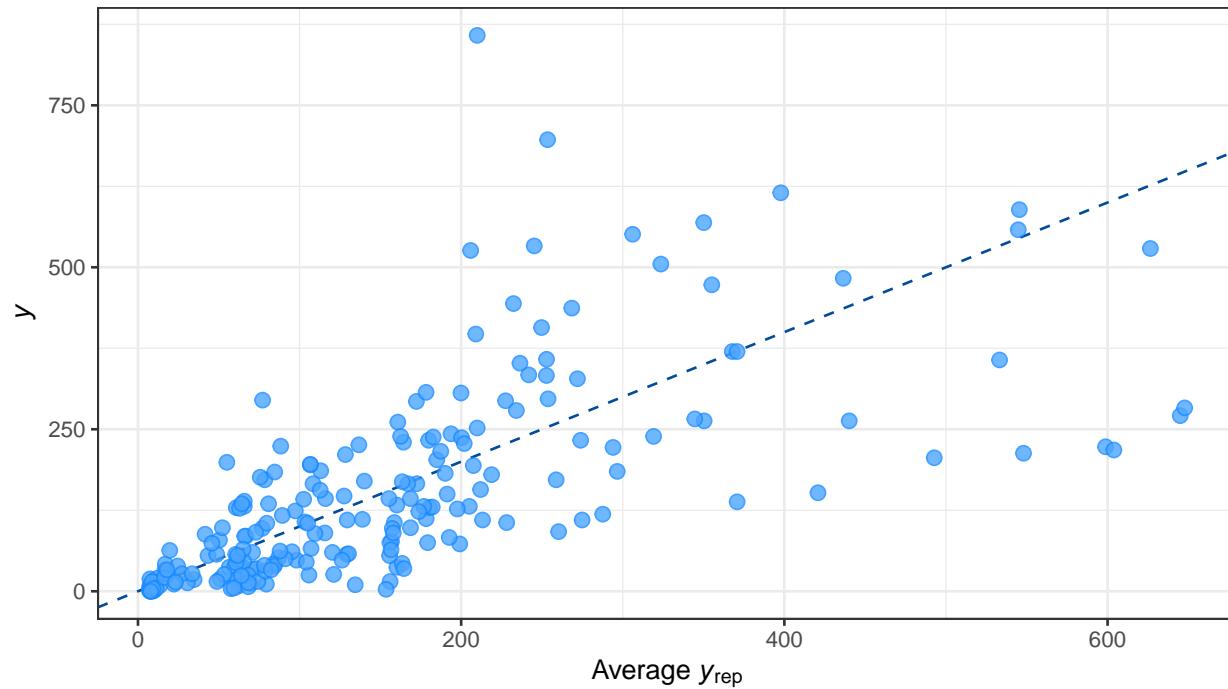
```
pp_check(stan.glm.1, plotfun = "boxplot") #check median/range
```



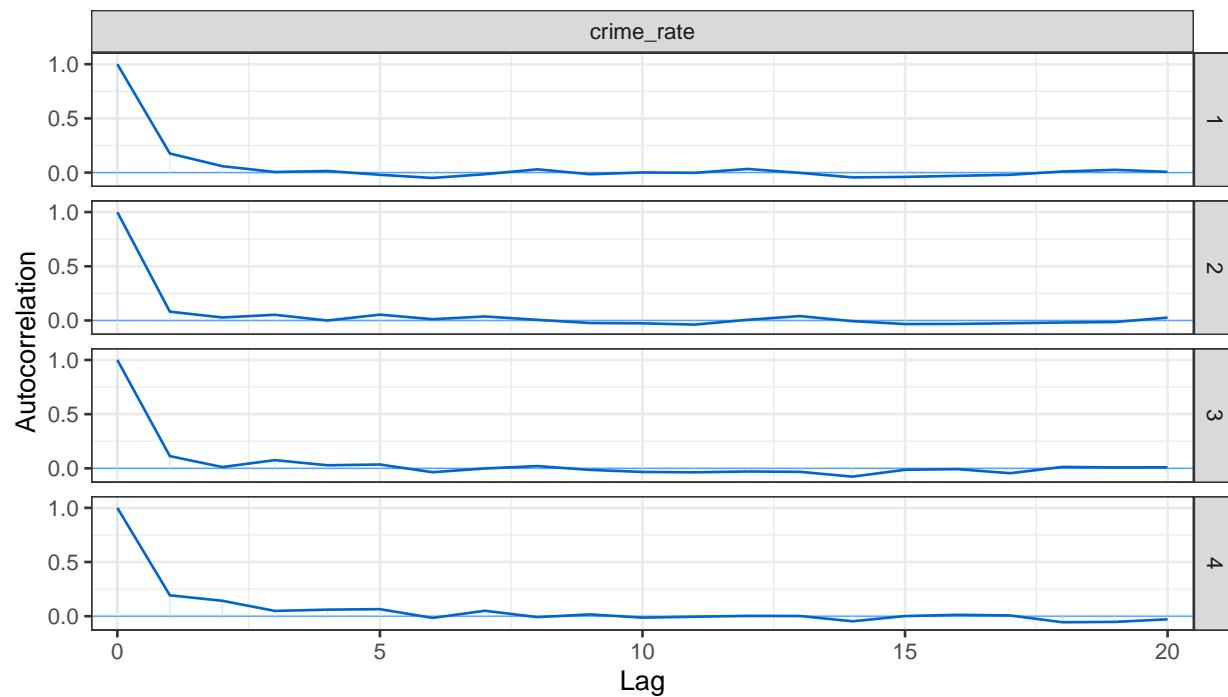
```
pp_check(stan.glm.1, plotfun = "stat_2d", stat = c("mean", "sd"))
```



```
# Scatterplot of two test statistics (capture the mean somewhat but
# sd is kind of bad. at least it's not high sd)
pp_check(stan.glm.1, plotfun = "scatter_avg") # Scatterplot of y vs. average yrep (our model is good for
```



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



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

