360 Final Project

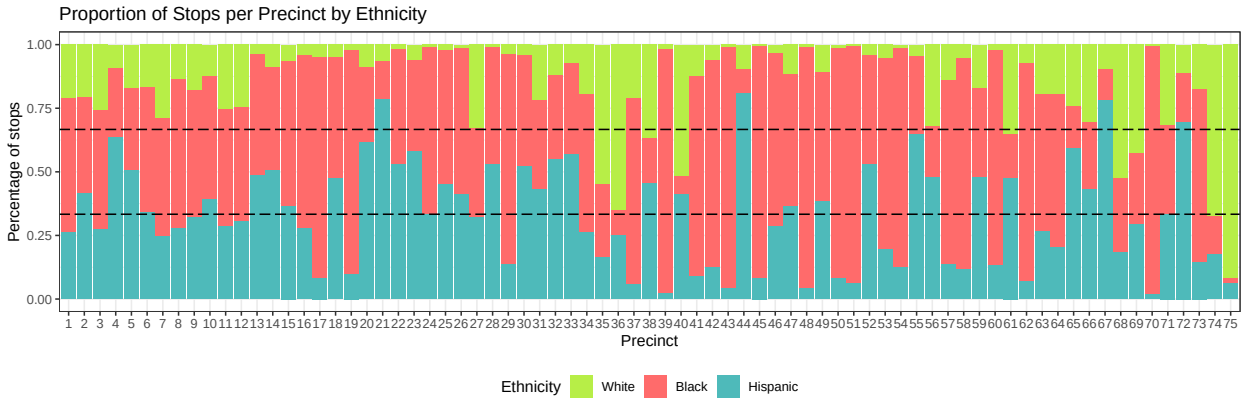
Chen Shi, Belle Xu

4/21/2021

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

Exploratory Data Analysis

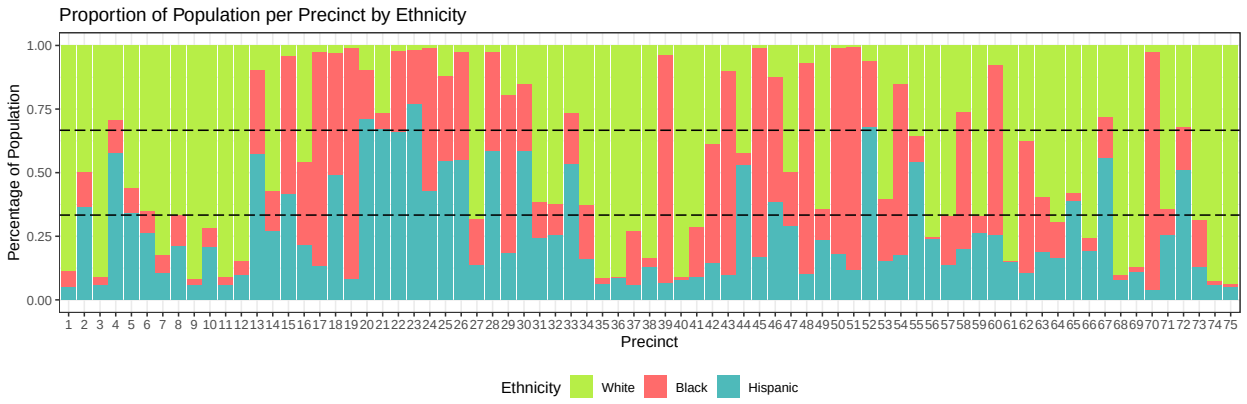
```
# percentage of stops for each ethnicity in each precinct
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct)) %>%
  group_by(eth, precinct) %>%
  mutate(stop_eth = sum(stops)) %>%
  ggplot(mapping = aes(x = precinct, y = stop_eth, fill = eth)) +
  geom_bar(position = "fill", stat = "identity") +
  geom_hline(yintercept = 1/3, linetype=5) +
  geom_hline(yintercept = 2/3, linetype=5) +
  labs(y = "Percentage of stops", x = "Precinct",
       title = "Proportion of Stops per Precinct by Ethnicity")+
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Ethnicity",
                      labels=c("White", "Black", "Hispanic"),
                      type = c("#b7ee47","#ff6b6b","#4ebaba"))
```



It seems that in most precincts, Blacks and Hispanics represent a much higher percentage of stops than white.

Is it because there are more blacks and Hispanics in these precincts so the police have a high possibility of stopping members from these two ethnic groups?

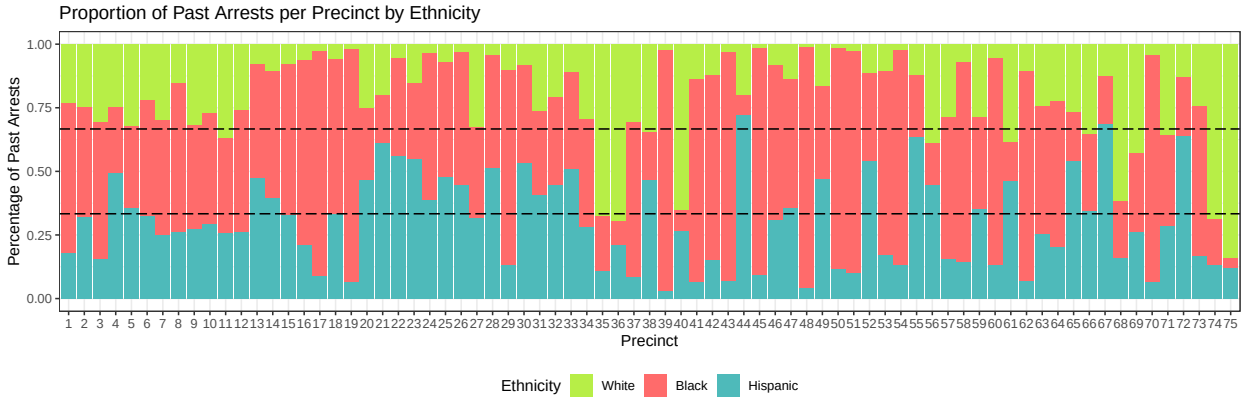
```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct)) %>%
  group_by(eth, precinct) %>%
  ggplot(mapping = aes(x = precinct, y = pop, fill = eth)) +
  geom_bar(position = "fill", stat = "identity") +
  geom_hline(yintercept = 1/3, linetype=5) +
  geom_hline(yintercept = 2/3, linetype=5) +
  labs(title = "Proportion of Population per Precinct by Ethnicity",
         y = "Percentage of Population", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Ethnicity",
                      labels=c("White", "Black", "Hispanic"),
                      type = c("#b7ee47","#ff6b6b","#4ebaba"))
```



In precincts where Blacks and Hispanics have larger population proportion, there is a corresponding higher proportion of stops for these two ethnic groups. However, in precincts where white accounts for a majority of the population, Blacks and Hispanics still account for high percentage of stops. (See precinct 1-12, 34-38, 40-42, 61-69).

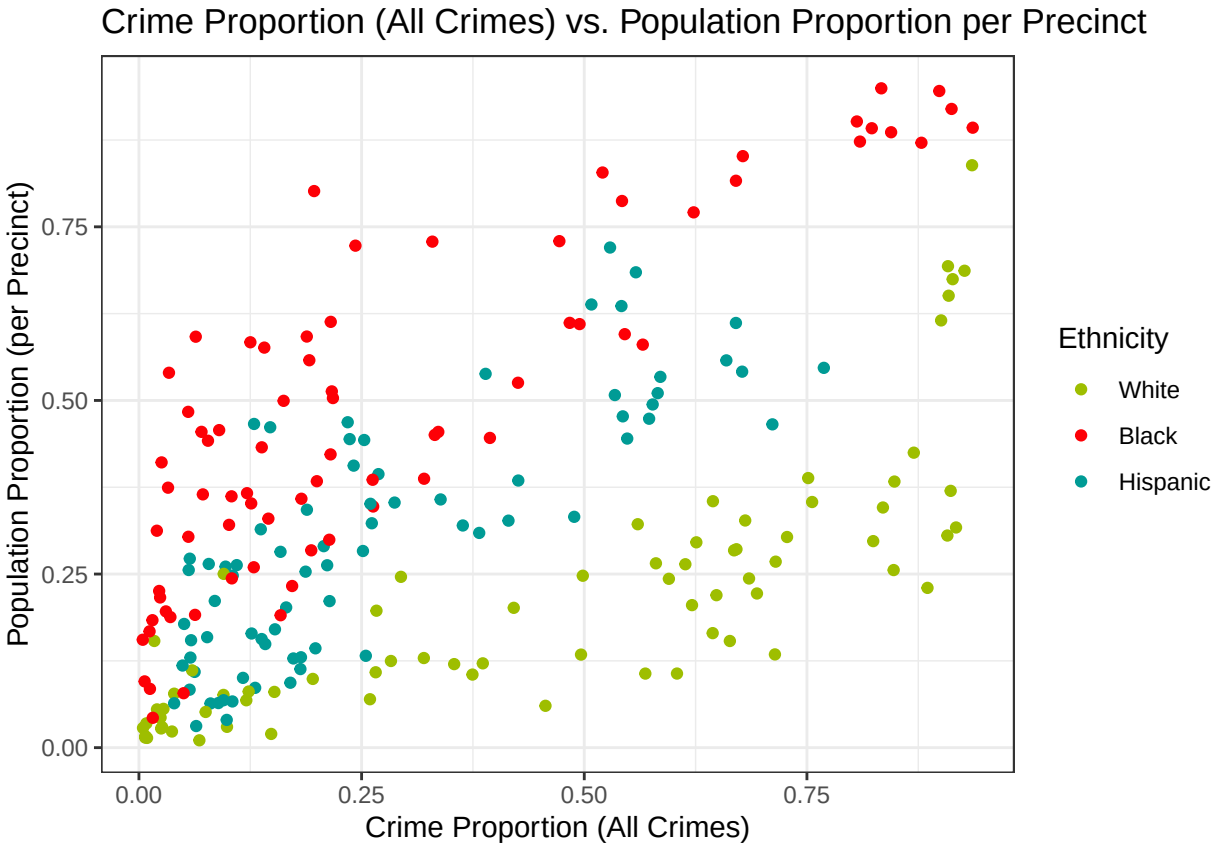
Is race an important factor for a police's decision-making process in stopping a pedestrian? How about the crime rate (past arrests of an ethnic group / total past arrests)?

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct)) %>%
  group_by(eth, precinct) %>%
  mutate(crime_eth = sum(past.arrests)) %>%
  ggplot(mapping = aes(x = precinct, y = crime_eth, fill = eth)) +
  geom_bar(position = "fill", stat = "identity") +
  geom_hline(yintercept = 1/3, linetype=5) +
  geom_hline(yintercept = 2/3, linetype=5) +
  labs(title = "Proportion of Past Arrests per Precinct by Ethnicity",
       y = "Percentage of Past Arrests", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Ethnicity",
                      labels=c("White", "Black", "Hispanic"),
                      type = c("#b7ee47","#ff6b6b","#4ebaba"))
```



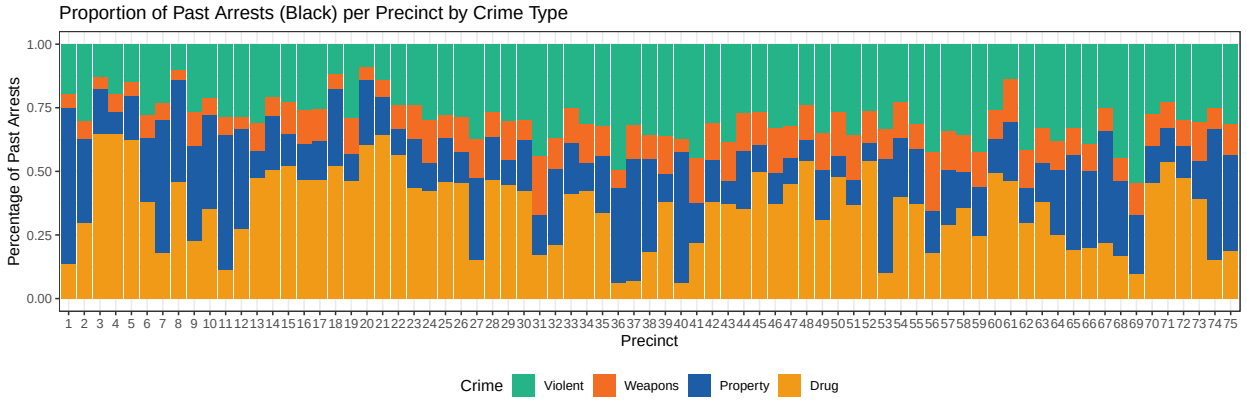
This plot looks similar to the first plot (for the proportion of stops). It seems that in most precincts, if there is a high percentage of past arrests for an ethnic group, there is also a high percentage of stops for that particular ethnic group. Is it possible that `past.arrests` or crime rate of an ethnic group is an important factor in determining whether the police stop members of that ethnic group? (The police might use `past.arrests` as an factor in decision-making process).

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  group_by(precinct) %>%
  mutate(total_pop = sum(pop) / 4,
         pop_prop = pop / total_pop,
         total_arrest = sum(past.arrests),
         crime_prop = past.arrests / total_arrest) %>%
  group_by(eth, precinct) %>%
  mutate(eth_crime = sum(crime_prop)) %>%
  filter(crime == 1) %>%
  ggplot(mapping = aes(x = pop_prop, y = eth_crime, color = eth)) +
  geom_point() + theme_bw()+
  labs(title = "Crime Proportion (All Crimes) vs. Population Proportion per Precinct",
       y = "Population Proportion (per Precinct)", x = "Crime Proportion (All Crimes)") +
  scale_colour_discrete(name = "Ethnicity",
                        labels=c("White", "Black", "Hispanic"),
                        type = c("#9EBE00","#FD0006","#009B95"))
```

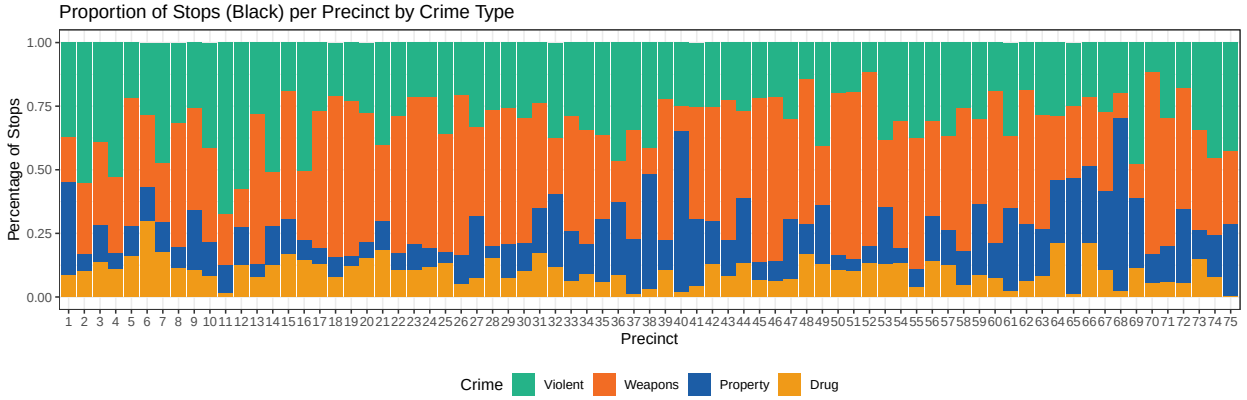


Is crime type an important factor? For example, are white people more likely to be stopped for drug crime if they are arrested in the past most for drug crime?

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "1") %>%
  ggplot(mapping = aes(x = precinct, y = past.arrests, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
  labs(title = "Proportion of Past Arrests (Black) per Precinct by Crime Type",
       y = "Percentage of Past Arrests", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```



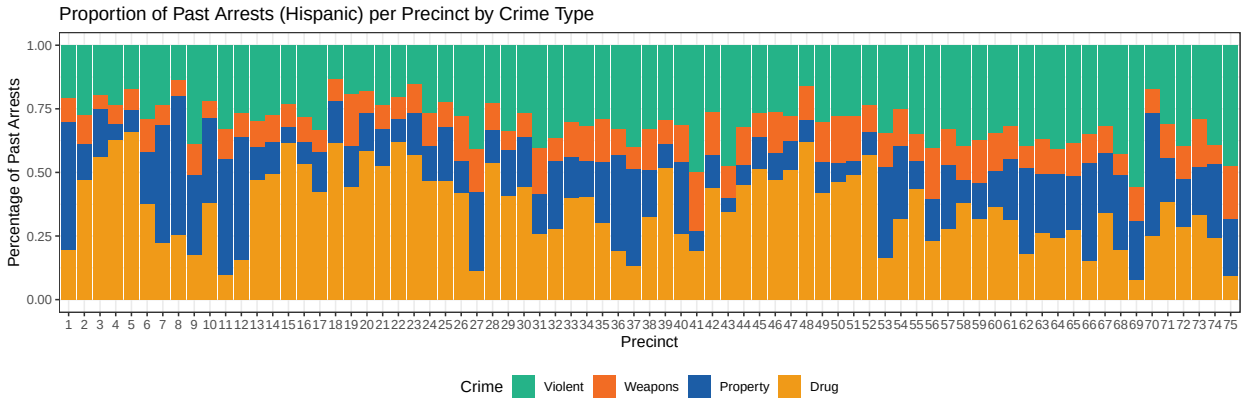

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "1") %>%
  ggplot(mapping = aes(x = precinct, y = stops, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
  labs(title = "Proportion of Stops (Black) per Precinct by Crime Type",
       y = "Percentage of Stops", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```



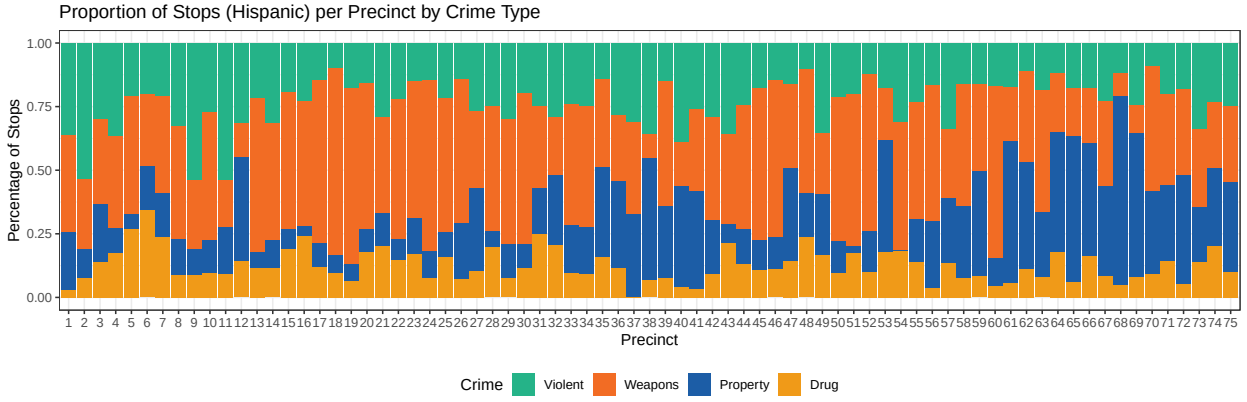
Previously, it seems that police is likely to use crime rate as a factor in determining whether to stop members of an ethnic group. However, after we control for the crime type, though black people are most arrested for drug crime, they are stopped most for weapon or violent crime.

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "2") %>%
  ggplot(mapping = aes(x = precinct, y = past.arrests, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
```

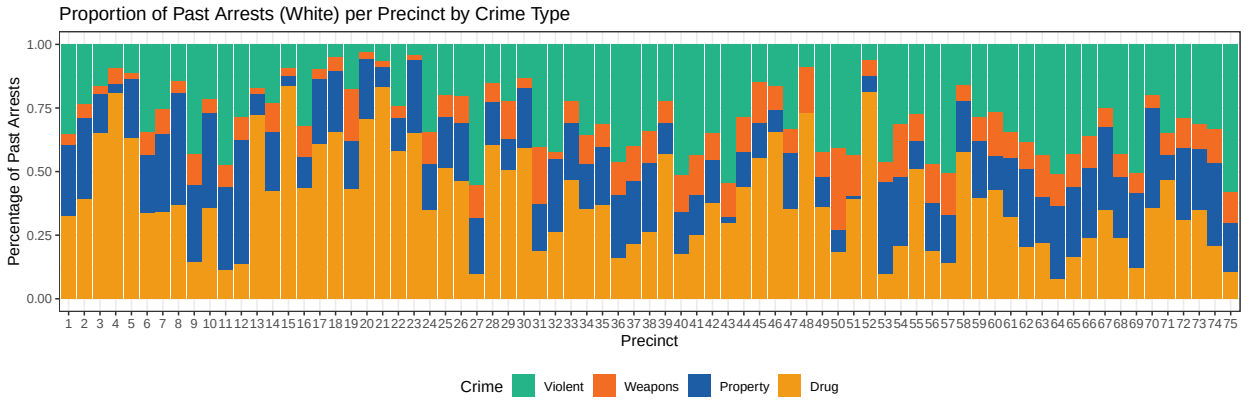
```
  labs(title = "Proportion of Past Arrests (Hispanic) per Precinct by Crime Type",
       y = "Percentage of Past Arrests", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```



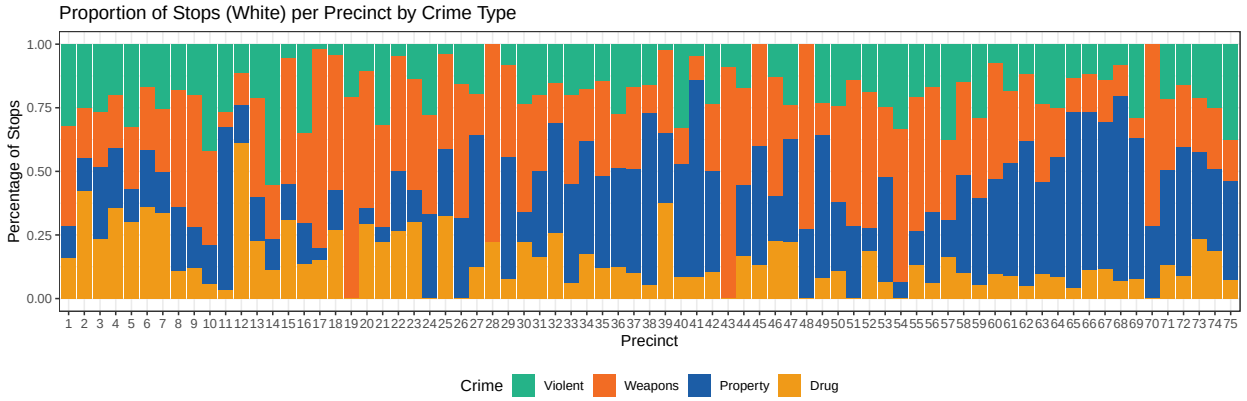

```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "2") %>%
  ggplot(mapping = aes(x = precinct, y = stops, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
  labs(title = "Proportion of Stops (Hispanic) per Precinct by Crime Type",
       y = "Percentage of Stops", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```



```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "3") %>%
  ggplot(mapping = aes(x = precinct, y = past.arrests, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
  labs(title = "Proportion of Past Arrests (White) per Precinct by Crime Type",
       y = "Percentage of Past Arrests", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```




```
stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(eth == "3") %>%
  ggplot(mapping = aes(x = precinct, y = stops, fill = crime)) +
  geom_bar(position = "fill", stat = "identity") +
  labs(title = "Proportion of Stops (White) per Precinct by Crime Type",
       y = "Percentage of Stops", x = "Precinct") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_discrete(name = "Crime",
                      labels=c("Violent", "Weapons", "Property", "Drug"),
                      type = c("#25B388","#F26C24","#1C5DA6", "#F09A18"))
```



A similar trend can be see in the graphs for white and hispanic ethnic groups. In each precinct, the crime type which people have high past arrest proportion in does not seem to be the crime type which people have high stop proportion in.

What happened? What determines the number of stops of an ethnic group in a certain precinct for a certain crime type? We control for crime type and build 3 models for each crime type. We hope to look at the effects of `past.arrests`, population distribution, race, and their possible interactions.

```
#create datasets for each crime
violent_crime <- stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(crime == "1")  %>%
  group_by(precinct) %>%
  mutate(total_pop = sum(pop),
         pop_prop = pop / total_pop,
         total_arrest = sum(past.arrests),
         crime_prop = past.arrests / total_arrest) %>%
  ungroup(precinct) %>%
  select(-crime, -total_pop, -total_arrest, -precinct)

weapon_crime <- stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(crime == "2")  %>%
  group_by(precinct) %>%
  mutate(total_pop = sum(pop),
         pop_prop = pop / total_pop,
         total_arrest = sum(past.arrests),
         crime_prop = past.arrests / total_arrest) %>%
  ungroup(precinct) %>%
  select(-crime, -total_pop, -total_arrest, -precinct)

property_crime <- stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(crime == "3")  %>%
  group_by(precinct) %>%
```

```
  mutate(total_pop = sum(pop),
         pop_prop = pop / total_pop,
         total_arrest = sum(past.arrests),
         crime_prop = past.arrests / total_arrest) %>%
  ungroup(precinct) %>%
  select(-crime, -total_pop, -total_arrest, -precinct)

drug_crime <- stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
         precinct = factor(precinct),
         crime = factor(crime)) %>%
  filter(crime == "4")  %>%
  group_by(precinct) %>%
  mutate(total_pop = sum(pop),
         pop_prop = pop / total_pop,
         total_arrest = sum(past.arrests),
         crime_prop = past.arrests / total_arrest) %>%
  ungroup(precinct) %>%
  select(-crime, -total_pop, -total_arrest, -precinct)
```

```
#investigate mean and var of each stop
violent <- c(mean(violent_crime$stops), var(violent_crime$stops))
Weapon <- c(mean(weapon_crime$stops), var(weapon_crime$stops))
Property <- c(mean(property_crime$stops), var(property_crime$stops))
Drug <- c(mean(drug_crime$stops), var(drug_crime$stops))
df <- data.frame(violent, Weapon, Property, Drug)
row.names(df) <- c("Mean", "Variance")
df
```

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

In the mean and variances seen above, we can clearly see that the variance values for the number of stops by each crime type are all way larger than the mean number of stops for each crime type. This motivates us to use the Negative Binomial regression model.

Modeling

Justification for choosing covariates:

1. ethnicity: obvious because we want to look at the effect of race in the decision-making process of police and want to investigate whether there is potential bias against blacks or Hispanics
2. pop_prop/crime_pop: in EDA, we observe the relationship between population distribution, past arrests, and number of stops after we control for precinct. (cite a previous paper that talks about how precinct characteristics affect the number of stops). So in our model, we hope to include precinct characteristics as covariates. We summarize these characteristics using pop_prop and crime_prop, which reflect the population distribution and crime distribution of each ethnicity within each precinct.
3. ethnicity:pop_prop: from EDA, it seems that the relationship between number of stops and pop_prop changes when the ethnicity changes.

4. ethnicity:crime_prop: for different ethnic groups, whether the effect of crime_prop on number of stops will change. From EDA, black people have high drug crime proportion, but they got high violent crime stops.

5. crime_prop:pop_prop: from EDA, we observe that even in precincts where majority of the population is white, black and hispanics still account for a high percentage of arrests.

For each crime type, we are going to fit 2 models initially, where each β will be the coefficient for which we will derive a posterior distribution for. All models will be fit with a negative binomial sampling model as motivated by the EDA; and all models will be fit with a Cauchy prior with a location (or center) parameter of 0 and a scale of 2.5. This prior is proposed by the Gelman paper.

Model 1:

$$log(mean(stops)) = log(past.arrests) + \beta_0 + \beta_1 * eth + \beta_2 * pop\_prop + \beta_3 * (eth : pop\_prop)$$

Model 2:

$$log(mean(stops)) = \beta_0 + \beta_1 * eth + \beta_2 * pop\_prop + \beta_3 * crime\_prop + \beta_4 * (eth : crime\_prop) + \beta_5 * (eth : pop\_prop) + \beta_6 * (pop\_pro$$

From the above 2 models, we will use both the looIC and Bayes' Factor diagnostic to determine which base model has the most predictive power. The Bayes' Factor diagnostic allows us to test under which model the observed data are more probable by comparing the marginal likelihood of the two models. A Bayes' Factor greater than 1 can be interpreted as that we have enough evidence to claim against the null (which is the hypothesis that some model $\mathcal{M}_1$ is better than $\mathcal{M}_2$)(Wetzels et al. 2011). However, this measure has some ambiguity as the two models proposed above do not have the same amount of covariates, therefore they do not have the same amount of coefficients; one can also further argue that, for model 1, we are in fact modeling for $log(mean(stops)/past.arrests)$ as opposed to the response variable $log(mean(stops))$ in model 2. Therefore, we choose to adjust for this ambiguity by using looIC as a second measure.

The looIC diagnostic is a model comparison measure for the two models' predictive accuracy. When comparing two fitted models, we can estimate the difference in their expected predictive accuracy by the difference in elpd_loo or elpd_waic (or multiplied by -2, if desired, to be on the deviance scale). When that difference, elpd_diff, is positive then the expected predictive accuracy for the second model is higher. A negative elpd_diff favors the first model.

After selecting the optimal base model, we will compare it with its Poisson sampling model variant to ensure that our current base model indeed captures the overdispersion better than the Poisson sampling model. We will not be examining potential models where certain main effects or interactions gets dropped from the base model we chose earlier. This is because all main effects and interactions have shown to have obvious interactions and relationships through our EDA; even if their interaction may not be statistically significant, we still need to keep these covariates in order to adjust for the coefficients of other covariates present in the model.

Violent Crimes

```
model.1 <- stan_glm(data = violent_crime,
                    formula = stops ~ eth * pop_prop,
                    family = neg_binomial_2(link = "log"),
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm1.csv"))
```

```
model.2 <- stan_glm(data = violent_crime,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = neg_binomial_2(link = "log"),
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    seed = 360,
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm3.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.2))
```

```
##         elpd_diff se_diff
## model.1   0.0       0.0
## model.2 -18.0      11.7
```

```
bayesfactor_models(model.1, model.2, denominator = model.1)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.
```

```
## Computation of Bayes factors: estimating marginal likelihood, please wait...
```

```
## Bayes Factors for Model Comparison
##
##     Model                                                          BF
## [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop < 0.001
##
## * Against Denominator: [1] eth * pop_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

Clearly model.1 yielded the best result above as we see that the elpd_diff is negative for model.2, and that the Bayes' Factor has little evidence to suggest that model.2 are indeed better than model.1 in terms of posterior predictive performance. The small se_diff compiled with our small sample size of 225 may indicate that the performance difference between those three models may not differ too drastically; however, we can still reasonably claim that model.1 is better.

```
model.3 <- stan_glm(data = violent_crime,
                    formula = stops ~ eth * pop_prop,
                    family = poisson(link = "log"),
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm4.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.3))
```

```
## Warning: Found 13 observations with a pareto_k > 0.7. With this many problematic observations we rec
```

```
##         elpd_diff se_diff
## model.1     0.0       0.0
## model.3 -3318.0     488.4
```

```
bayesfactor_models(model.1, model.3, denominator = model.1)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.

## Computation of Bayes factors: estimating marginal likelihood, please wait...

## Bayes Factors for Model Comparison
##
##     Model                BF
## [2] eth * pop_prop < 0.001
##
## * Against Denominator: [1] eth * pop_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

Note that our base model model 1 contains a formula in which all covariates were deemed significant in the EDA, therefore no further comparison needs to be made.

```
violent_model <- model.1
```

Weapon Crimes

```
model.1 <- stan_glm(data = weapon_crime,
                    formula = stops ~ eth * pop_prop,
                    family = neg_binomial_2(link = "log"),
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm1.csv"))
```

```
model.2 <- stan_glm(data =  weapon_crime,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = neg_binomial_2(link = "log"),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm3.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.2))
```

```
##         elpd_diff se_diff
## model.1   0.0       0.0
## model.2 -24.1      12.7
```

```
bayesfactor_models(model.1, model.2,denominator = model.1)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.

## Computation of Bayes factors: estimating marginal likelihood, please wait...

## Bayes Factors for Model Comparison
##
##     Model                                                        BF
## [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop < 0.001
##
## * Against Denominator: [1] eth * pop_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

Clearly model.1 yielded the best result above as we see that the elpd_diff is negative for model.2, and that the Bayes' Factor has little evidence to suggest that model.2 are indeed better than model.1 in terms of posterior predictive performance. The small se_diff compiled with our small sample size of 225 may indicate that the performance difference between those three models may not differ too drastically; however, we can still reasonably claim that model.1 is better.

```
model.3 <- stan_glm(data =  weapon_crime,
                    formula = stops ~ eth * pop_prop,
                    family = poisson(link = "log"),
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm4.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.3))
```

```
## Warning: Found 26 observations with a pareto_k > 0.7. With this many problematic observations we rec

##         elpd_diff se_diff
## model.1     0.0       0.0
## model.3 -6396.9    1011.0
```

```
bayesfactor_models(model.1, model.3, denominator = model.1)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.

## Computation of Bayes factors: estimating marginal likelihood, please wait...

## Bayes Factors for Model Comparison
##
##     Model               BF
## [2] eth * pop_prop < 0.001
##
## * Against Denominator: [1] eth * pop_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

Note that our base model model 1 contains a formula in which all covariates were deemed significant in the EDA, therefore no further comparison needs to be made.

```
weapon_model <- model.1
```

Property Crimes

```
df <- property_crime %>%
  mutate(past.arrests = if_else(past.arrests!= 0, past.arrests, as.integer(1)))
```

```
model.1 <- stan_glm(data = df,
                    formula = stops ~ eth * pop_prop,
                    family = neg_binomial_2(link = "log"),
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                    diagnostic_file = file.path(tempdir(), "glm1.csv"))
```

```
model.2 <- stan_glm(data = df,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = neg_binomial_2(link = "log"),
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    seed = 360,
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm3.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.2))
```

```
##         elpd_diff se_diff
## model.2   0.0       0.0
## model.1 -47.2      10.5
```

```
bayesfactor_models(model.1, model.2, denominator = model.2)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.

## Computation of Bayes factors: estimating marginal likelihood, please wait...

## Bayes Factors for Model Comparison
##
##     Model               BF
## [1] eth * pop_prop < 0.001
##
## * Against Denominator: [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

```
model.3 <- stan_glm(data = property_crime,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = poisson(link = "log"),
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    seed = 360,
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm4.csv"))
```

```
rstanarm::loo_compare(loo(model.2), loo(model.3))
```

```
## Warning: Found 45 observations with a pareto_k > 0.7. With this many problematic observations we rec
```

```
##         elpd_diff se_diff
## model.2     0.0       0.0
## model.3 -6832.6     753.9
```

```
bayesfactor_models(model.2, model.3, denominator = model.2)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.
```

```
## Computation of Bayes factors: estimating marginal likelihood, please wait...
```

```
## Bayes Factors for Model Comparison
##
##     Model                                                          BF
## [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop < 0.001
##
## * Against Denominator: [1] eth * (crime_prop + pop_prop) + pop_prop * crime_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

```
property_model <- model.2
```

In the model above, we chose to modify the `past.arrests` value from 0 to 1 for one of the observations recording property crimes. We chose to make this modification because our proposed model 1 would not allow us to have 0 entries for `past.arrests`. We did not omit this entry because we still need to account for the variability in population proportion and crime proportion in that precinct for all ethnic groups. If we did not make the adjustment from 0 to 1, we would be ruling out the possibility that people described in that particular observation has ever been arrested for a property crime. Therefore, it is safe to assume that at least 1 person in this particular group has been arrested for a property crime in the past.

Drug Crimes

```
model.1 <- stan_glm(data = drug_crime,
                    formula = stops ~ eth * pop_prop,
                    family = neg_binomial_2(link = "log"),
```

```
                    offset = log(past.arrests),
                    seed = 360,
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm1.csv"))
```

```
model.2 <- stan_glm(data = drug_crime,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = neg_binomial_2(link = "log"),
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    seed = 360,
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm3.csv"))
```

```
rstanarm::loo_compare(loo(model.1), loo(model.2))
```

```
##         elpd_diff se_diff
## model.2   0.0       0.0
## model.1 -23.2      12.9
```

```
bayesfactor_models(model.1, model.2, denominator = model.2)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.

## Computation of Bayes factors: estimating marginal likelihood, please wait...

## Bayes Factors for Model Comparison
##
##     Model                BF
## [1] eth * pop_prop < 0.001
##
## * Against Denominator: [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

Clearly model.2 yielded the best result above as we see that the elpd_diff is negative for model.1, and that the Bayes' Factor has little evidence to suggest that model.1 are indeed better than model.2 in terms of posterior predictive performance. The small se_diff compiled with our small sample size of 225 may indicate that the performance difference between those three models may not differ too drastically; however, we can still reasonably claim that model.2 is better.

```
model.3 <- stan_glm(data = drug_crime,
                    formula = stops ~ eth * (crime_prop + pop_prop) +
                      pop_prop * crime_prop,
                    family = poisson(link = "log"),
                    prior = cauchy(0, 2.5),
                    prior_intercept = cauchy(0, 2.5),
                    seed = 360,
                    refresh = 0,
                   diagnostic_file = file.path(tempdir(), "glm4.csv"))
```

```
rstanarm::loo_compare(loo(model.2), loo(model.3))
```

```
## Warning: Found 21 observations with a pareto_k > 0.7. With this many problematic observations we rec
```

```
##         elpd_diff se_diff
## model.2     0.0       0.0
## model.3 -2624.5     340.2
```

```
bayesfactor_models(model.2, model.3, denominator = model.2)
```

```
## Warning: Bayes factors might not be precise.
## For precise Bayes factors, it is recommended sampling at least 40,000 posterior samples.
```

```
## Computation of Bayes factors: estimating marginal likelihood, please wait...
```

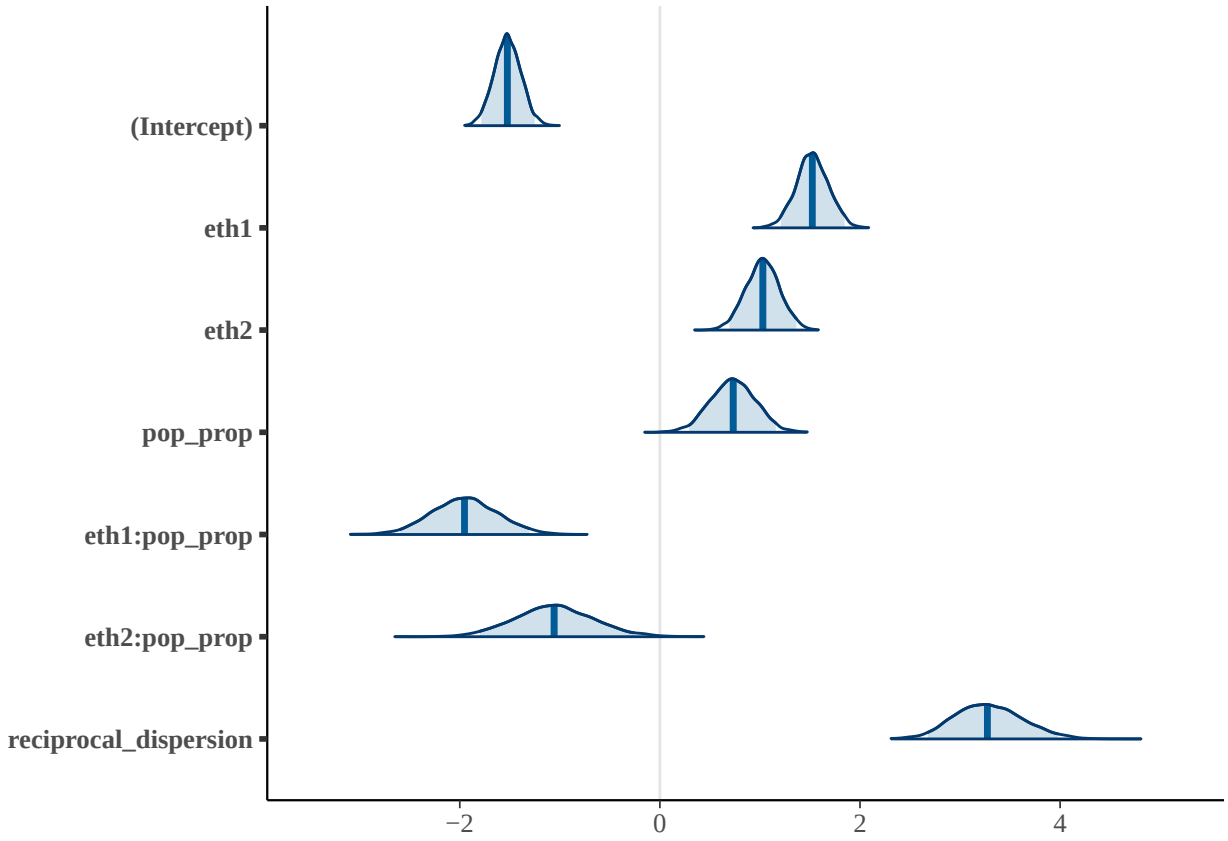
```
## Bayes Factors for Model Comparison
##
##     Model                                                          BF
## [2] eth * (crime_prop + pop_prop) + pop_prop * crime_prop < 0.001
##
## * Against Denominator: [1] eth * (crime_prop + pop_prop) + pop_prop * crime_prop
## *   Bayes Factor Type: marginal likelihoods (bridgesampling)
```

```
drug_model <- model.2
```

Interpretations & Conclusion

Violent

```
mcmc_areas(as.matrix(violent_model), prob = 0.95, prob_outer = 1)
```




```
round(coef(violent_model), 3)
```

```
##   (Intercept)          eth1          eth2      pop_prop eth1:pop_prop
##        -1.524         1.523         1.029         0.733        -1.952
## eth2:pop_prop
##        -1.057
```

```
round(posterior_interval(violent_model, prob = 0.95), 3)
```

```
##                          2.5%  97.5%
## (Intercept)            -1.785 -1.251
## eth1                    1.209  1.846
## eth2                    0.695  1.364
## pop_prop                0.292  1.159
## eth1:pop_prop          -2.596 -1.303
## eth2:pop_prop          -1.800 -0.258
## reciprocal_dispersion  2.678  3.977
```

As we can see in both the posterior distribution graphs as well as the estimated coefficients and 95% credible intervals above, the coefficients for covariates `eth`, `pop_prop`, as well as the interaction between `eth` and `pop_prop` are all estimated to be non-zero. This indicates that ethnic group, the population proportion of an ethnic group in a particular precinct, as well as the interaction between those two variables are significant in predicting the log of the mean number of stops for violent crimes for a particular ethnic group in a precinct after controlling the number of arrests for violent crimes in the past year in this precinct for a particular ethnic group.

From the `stan_glm()` manual page, we know that the coefficient estimates yielded above represents the posterior median values of each coefficient. Therefore, we can interpret the above coefficients as follows:

- We have an intercept estimate of -1.561. This means that, for the white ethnic group with 0 population proportion in a precinct (i.e. there are no people of white descent in this precinct), the mean number of stops for violent crimes in this precinct after controlling for the number of arrests for violent crimes in the past year for the white ethnic group is expected to be 0.21.
- The `eth1` estimate of 1.568 means that: for people of black ethnic group, the mean number of stops for violent crimes in a precinct after controlling for the number of arrests for violent crimes in the past year for the black ethnic group is expected to be 4.797 times the mean number of stops for violent crimes in the same precinct for the white ethnic group, holding their corresponding population proportion constant. This indicates blacks have about 156.8% higher chance of being stopped compared to whites.
- The `eth2` estimate of 1.071 means that: for people of hispanic ethnic group, the mean number of stops for violent crimes in a precinct after controlling for the number of arrests for violent crimes in the past year for the hispanic ethnic group is expected to be 2.918 times the mean number of stops for violent crimes in the same precinct for the white ethnic group, holding their corresponding population proportion constant. This indicates hispanics have about 107.1% higher chance of being stopped compared to whites.

Note that in the results for the `eth` variable above, we could see that both people of black or hispanic descent have higher chance of being stopped than whites after controlling for the number of past arrests if we held their corresponding population proportion constant.

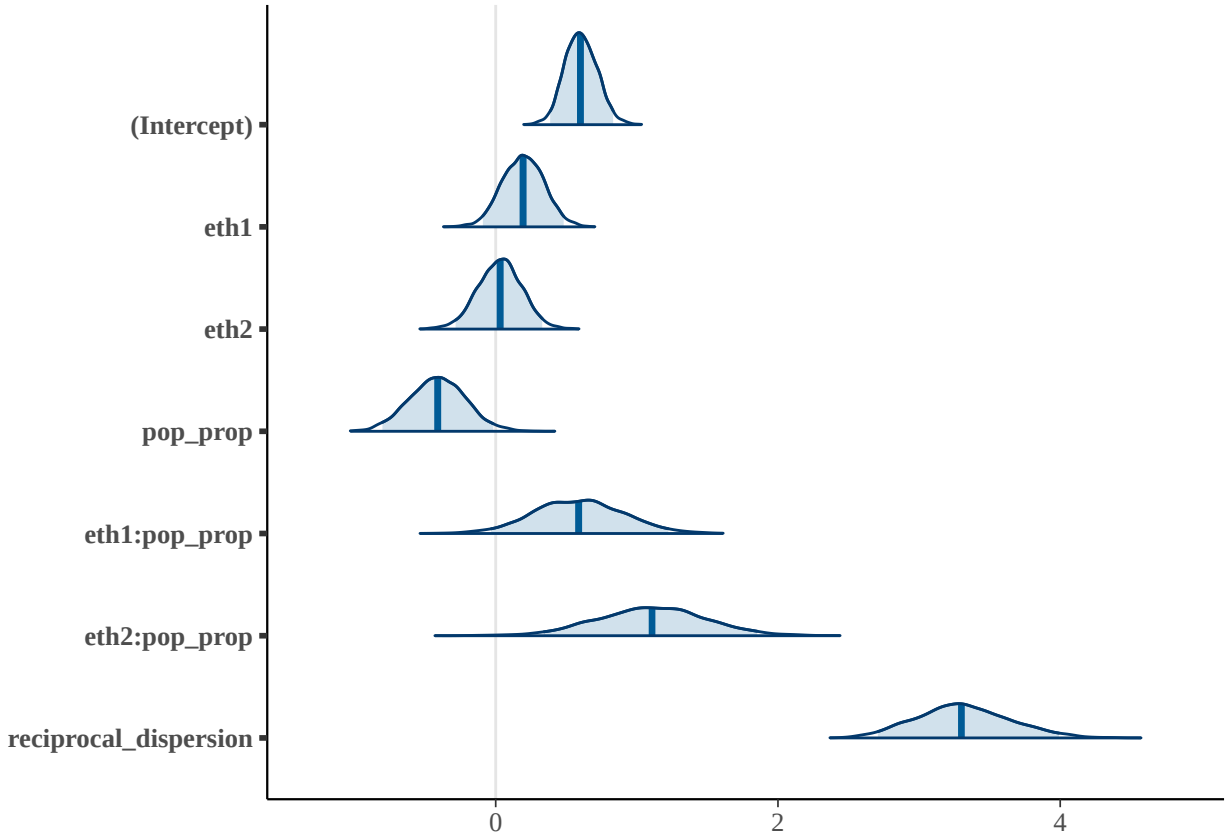
- The `pop_prop` estimate of 0.795 means that: For each additional unit(?????) in the population proportion of whites in a precinct, the mean number of stops for violent crimes for whites is expected to multiply by a factor of 1.406, holding ethnic group constant. This positive correlation between the mean number of stops and ethnic group population proportion implies the relationship we have explored in the EDA, which is that if there were more people or high proportion of a particular ethnic group in a precinct, then people of this ethnic group will have a higher chance of being stopped.
- The `eth1:pop_prop` estimate of -2.050 means that: For blacks, the previously interpreted effect of population proportion on the mean number of stops in a precinct for violent crimes will decrease by a factor of `round(exp(-2.050),3)` compared to the white ethnic group.
- The `eth2:pop_prop` estimate of -1.164 means that: For Hispanics, the previously interpreted effect of population proportion on the mean number of stops in a precinct for violent crimes will decrease by a factor of `round(exp(-2.050),3)` compared to the white ethnic group.

In both of the interpretations above, we can see that the effect of population proportion for blacks or hispanics significantly decreases compared to whites. This would imply that, adjusting for the corresponding past number of arrests, population proportion play a much smaller role in determining whether a person will get stopped given the person is black or Hispanic.

From the interpretations above, we can see that, compared to whites, blacks and Hispanics have a higher chance of being stopped; and that population proportion plays a smaller role in determining whether they will get stopped. This significance could indicate potential bias in the stopping policy as a person is more likely to be stopped if they are of black or hispanic descents than if they are of white descent. Furthermore, we have also seen that the models which saw past number of arrests or crime proportion (by ethnic group in a precinc) as covariates did not have as strong of a posterior predictive performance compared to our current model. This could also indicate that, the number of past arrests or crime proportion has little effect in predicting the mean number of stops for violent crimes in a precinct for an ethnic group. Combining the previous three results, we can see a further emphasis placed on the role of ethnicity in a person's chances of being stopped over the population proportion or the past crime counts and proportions.

Weapon

```
mcmc_areas(as.matrix(weapon_model), prob = 0.95, prob_outer = 1)
```



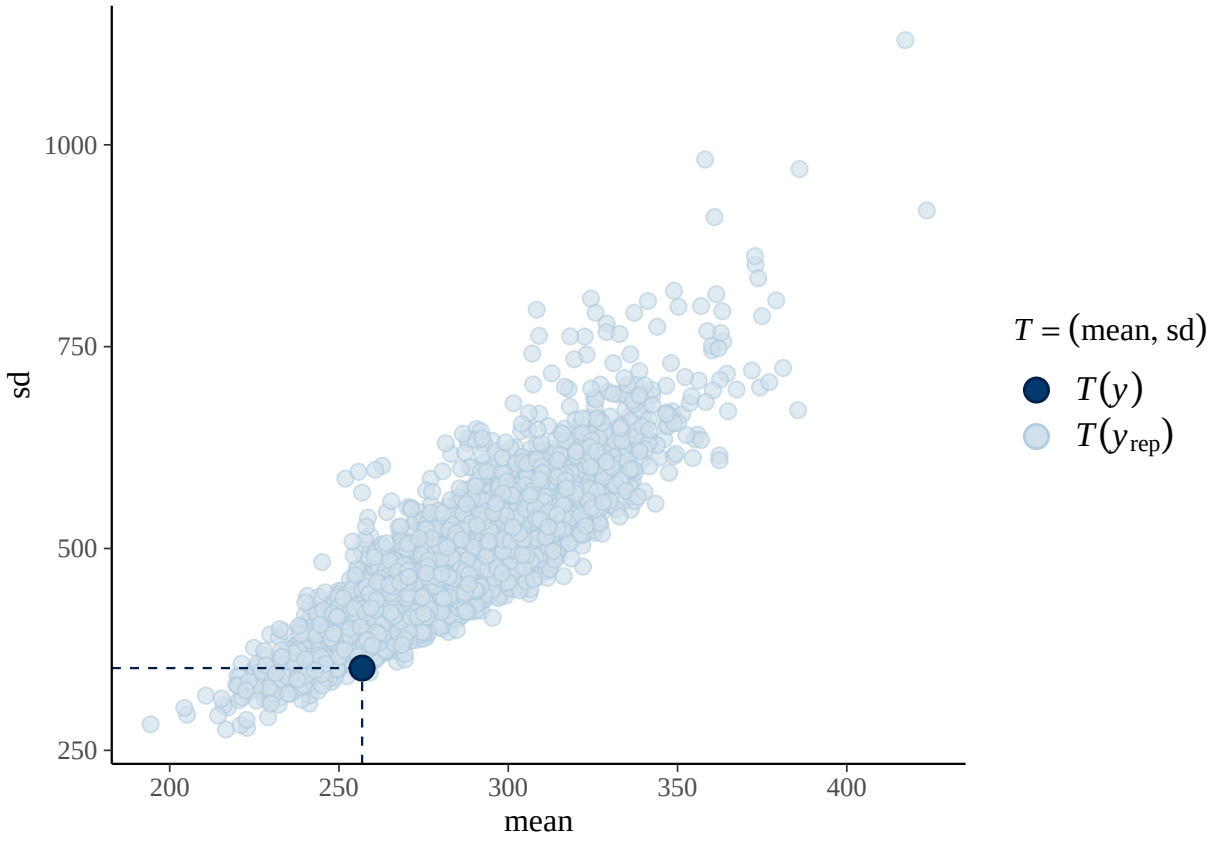

```
round(coef(weapon_model), 3)
```

```
##   (Intercept)          eth1          eth2      pop_prop eth1:pop_prop
##         0.600         0.194         0.031        -0.411         0.588
## eth2:pop_prop
##         1.108
```

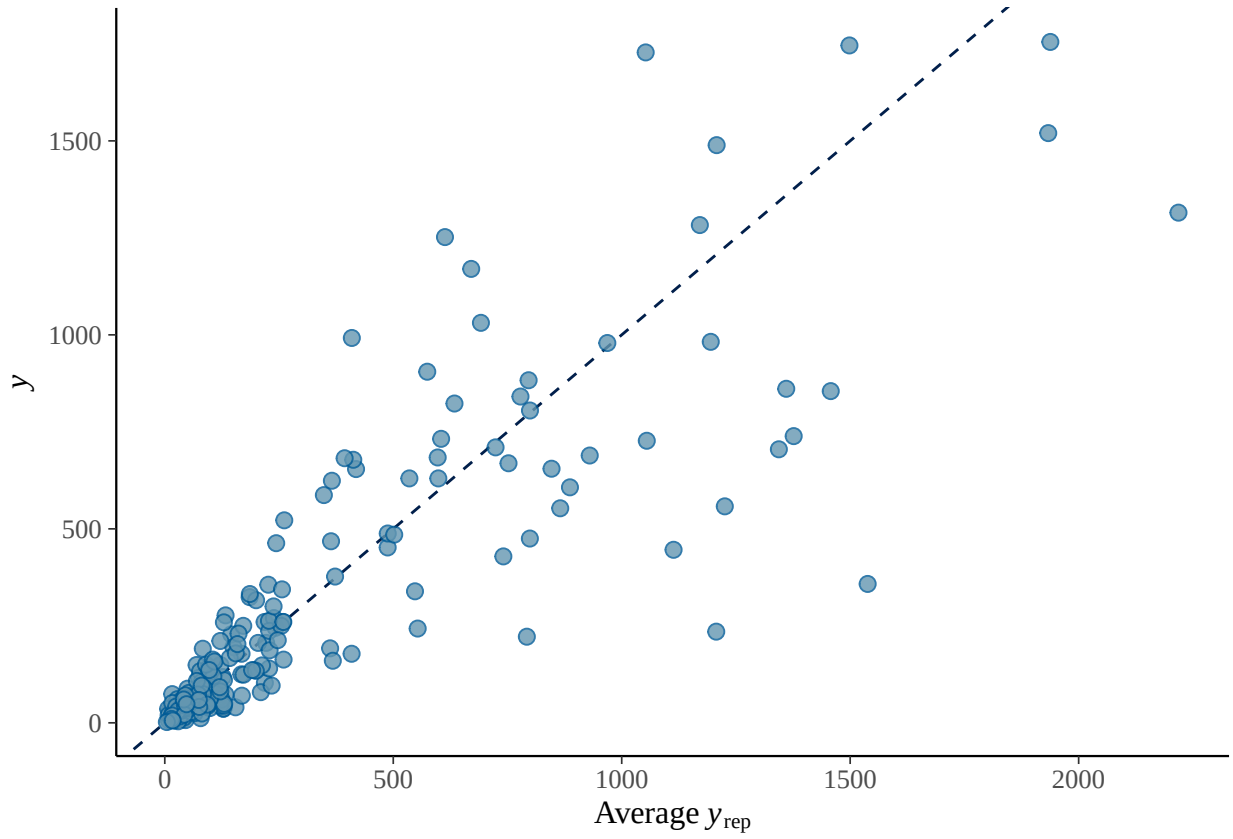
```
round(posterior_interval(weapon_model, prob = 0.95), 3)
```

```
##                           2.5%  97.5%
## (Intercept)              0.386  0.832
## eth1                    -0.091  0.482
## eth2                    -0.284  0.330
## pop_prop                -0.803 -0.017
## eth1:pop_prop           -0.032  1.223
## eth2:pop_prop            0.341  1.852
## reciprocal_dispersion    2.708  3.991
```

```
pp_check(weapon_model, plotfun = "stat_2d", stat = c("mean", "sd"))
```

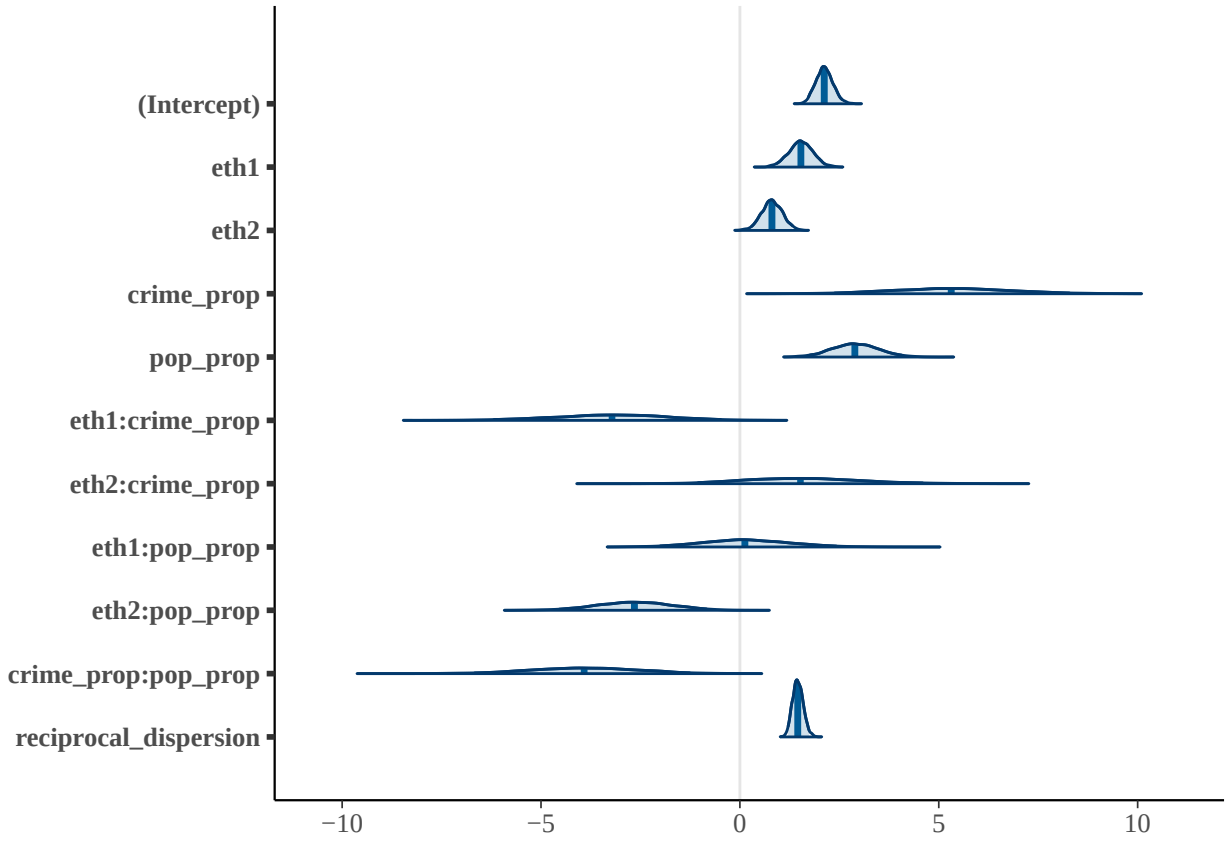



```
pp_check(weapon_model, plotfun = "scatter_avg")
```



Property

```
mcmc_areas(as.matrix(property_model), prob = 0.95, prob_outer = 1)
```



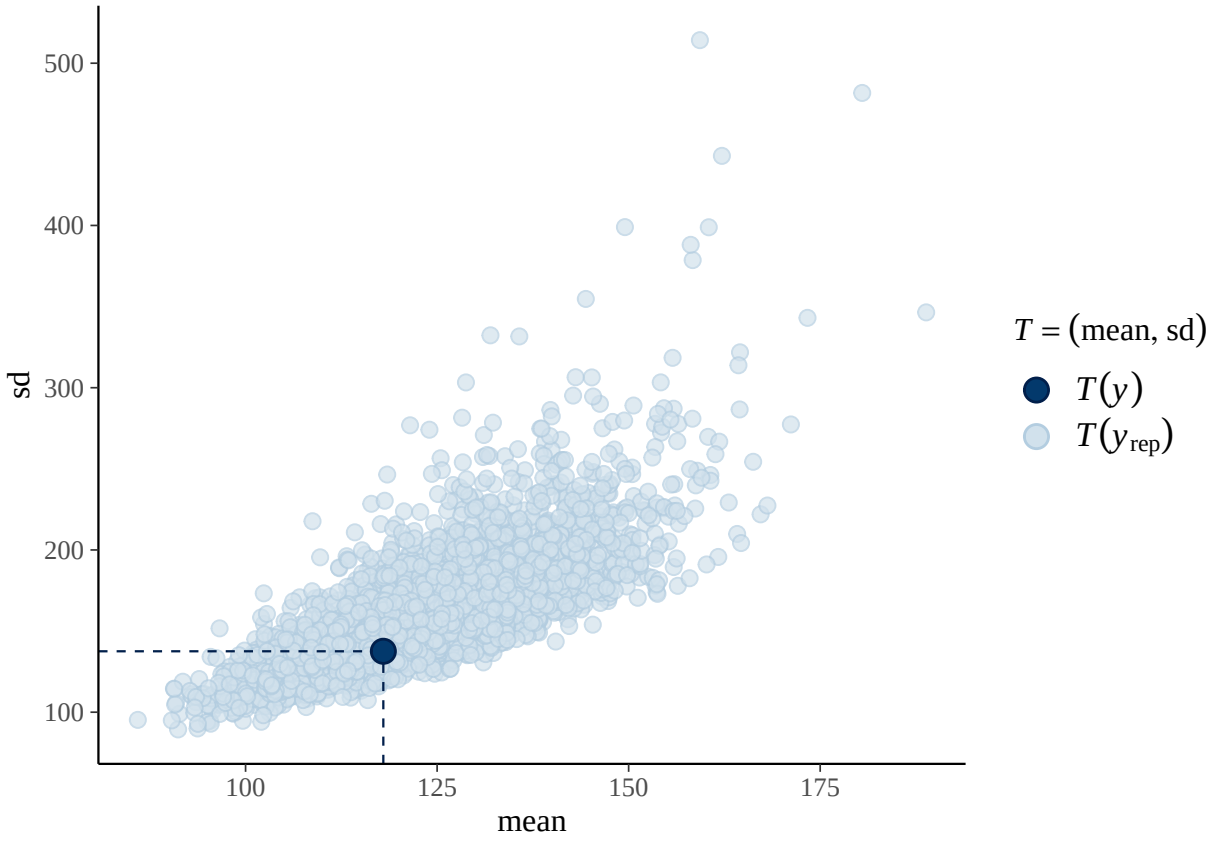

```
round(coef(property_model), 3)
```

```
##         (Intercept)                eth1                eth2         crime_prop
##               2.121               1.534               0.806               5.312
##            pop_prop     eth1:crime_prop     eth2:crime_prop       eth1:pop_prop
##               2.890              -3.213               1.524               0.124
##       eth2:pop_prop crime_prop:pop_prop
##              -2.652              -3.918
```

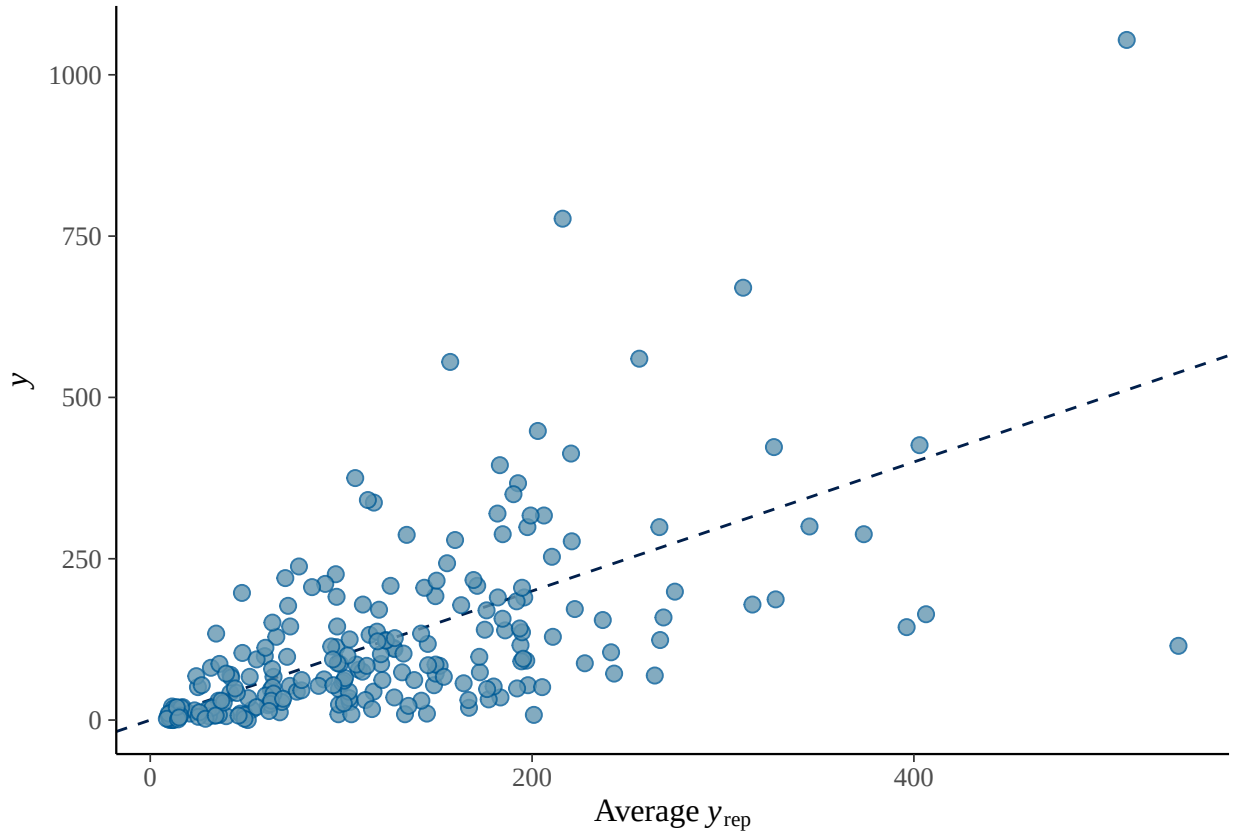
```
round(posterior_interval(property_model, prob = 0.95), 3)
```

```
##                           2.5%  97.5%
## (Intercept)              1.727  2.564
## eth1                     0.924  2.142
## eth2                     0.290  1.319
## crime_prop               2.386  8.281
## pop_prop                 1.759  4.065
## eth1:crime_prop         -6.182 -0.486
## eth2:crime_prop         -1.045  4.592
## eth1:pop_prop           -2.002  2.421
## eth2:pop_prop           -4.543 -0.844
## crime_prop:pop_prop     -6.666 -1.265
## reciprocal_dispersion    1.214  1.739
```

```
pp_check(property_model, plotfun = "stat_2d", stat = c("mean", "sd"))
```

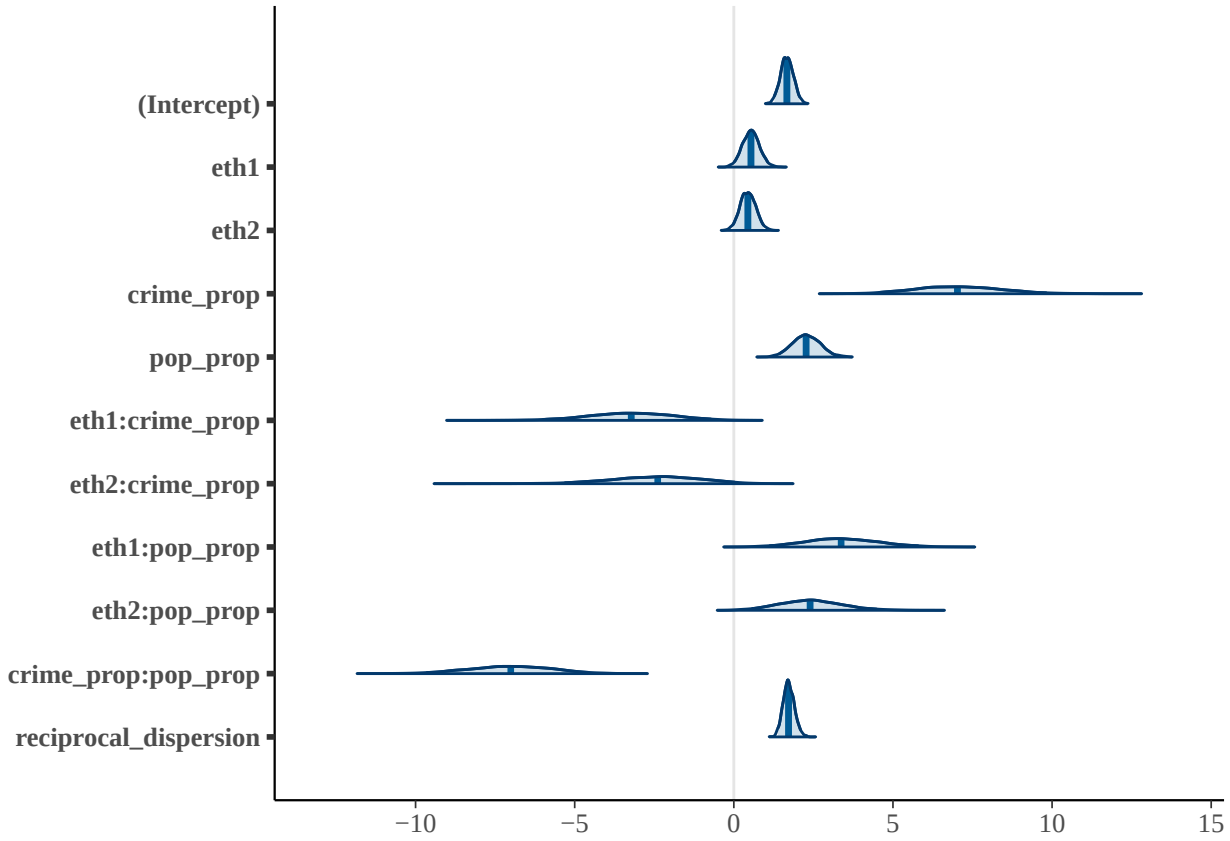



```
pp_check(property_model, plotfun = "scatter_avg")
```



Drug

```
mcmc_areas(as.matrix(drug_model), prob = 0.95, prob_outer = 1)
```



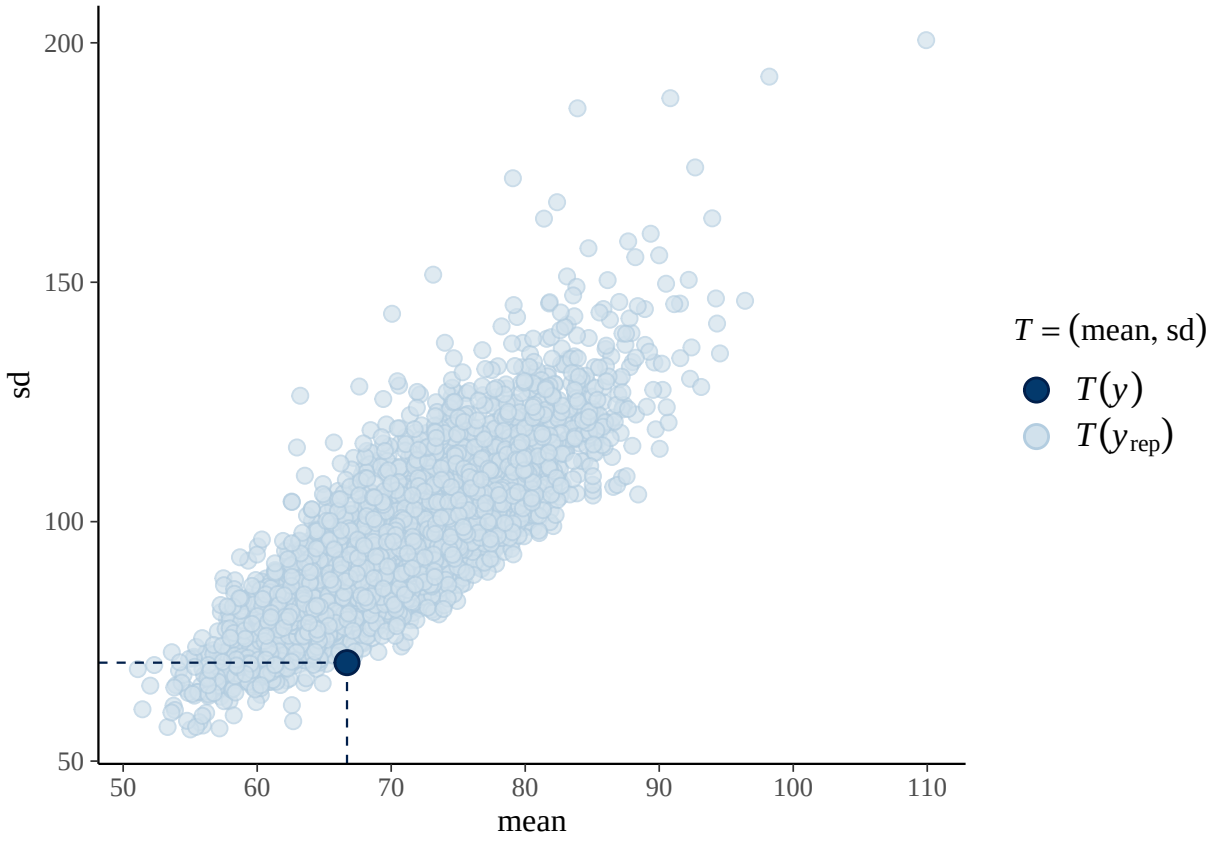

```
round(coef(drug_model), 3)
```

```
##         (Intercept)                eth1                eth2          crime_prop
##               1.667               0.538               0.442               7.026
##            pop_prop     eth1:crime_prop     eth2:crime_prop       eth1:pop_prop
##               2.272              -3.221              -2.390               3.370
##       eth2:pop_prop crime_prop:pop_prop
##               2.396              -7.010
```

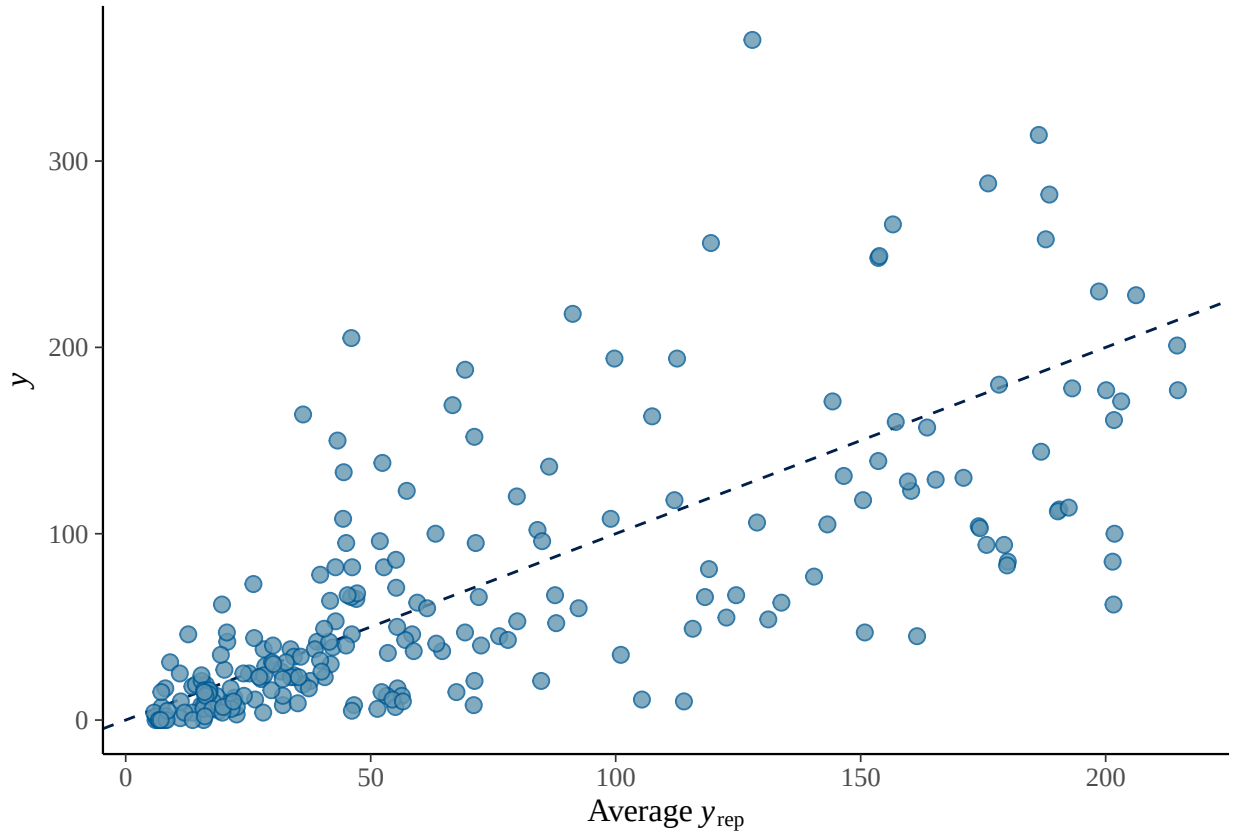
```
round(posterior_interval(drug_model, prob = 0.95), 3)
```

```
##                           2.5%  97.5%
## (Intercept)              1.276  2.070
## eth1                     0.022  1.067
## eth2                    -0.030  0.934
## crime_prop               4.604  9.811
## pop_prop                 1.435  3.171
## eth1:crime_prop         -5.917 -0.727
## eth2:crime_prop         -5.300  0.023
## eth1:pop_prop            1.135  5.786
## eth2:pop_prop            0.584  4.480
## crime_prop:pop_prop     -9.639 -4.572
## reciprocal_dispersion    1.390  2.098
```

```
pp_check(drug_model, plotfun = "stat_2d", stat = c("mean", "sd"))
```



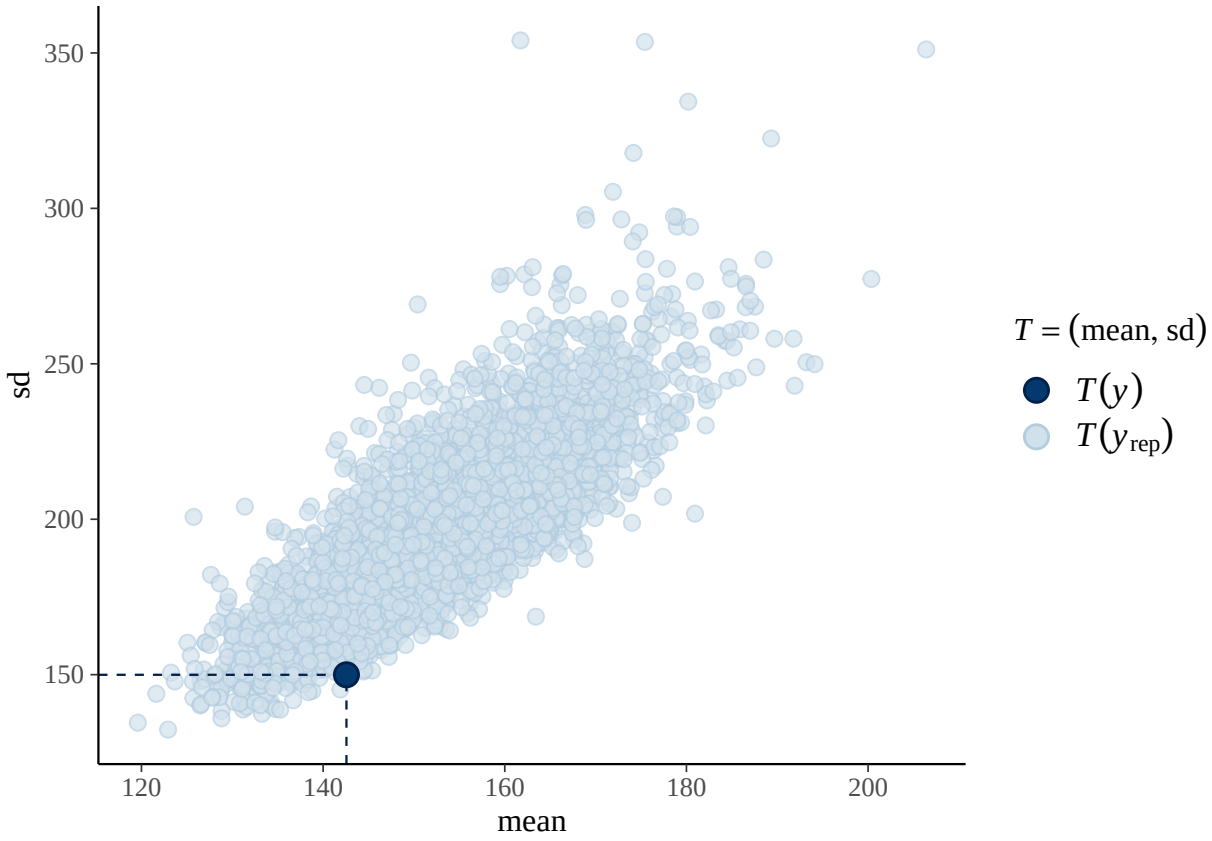

```
pp_check(drug_model, plotfun = "scatter_avg")
```



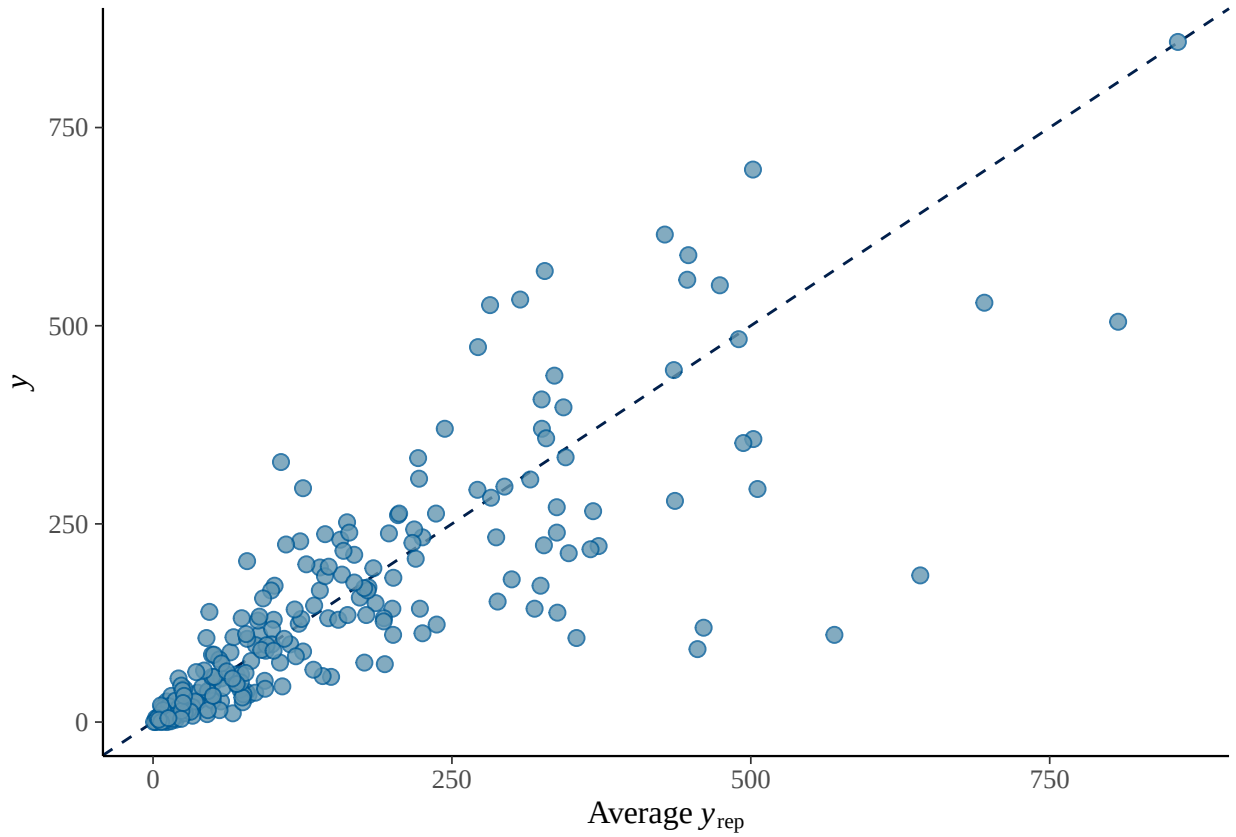
Posterior Diagnostics & Limitations

Violent

```
pp_check(violent_model, 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(violent_model, plotfun = "scatter_avg") # Scatterplot of y vs. average yrep (our model is good
```




```
#more checks that doesn't need much interpretation to be added later on
#(i.e. acf, traceplot, density plot)
```

As seen in the first scatterplot above, the mean and standard deviation of our posterior predictive samples are plotted against our original data. It seems that our model would overpredict both the mean and standard deviation, though the cluster still captures the observed data. Our model can be further examined in the second scatterplot, which plots the mean posterior predictive samples against our original data. A perfect model would have its points clustered around the diagonal line in the graph. The points in this plot are clustered to the diagonal line in lower values, yet becomes more and more dispersed in higher values. This indicates that our model is sufficient in predicting the mean number of stops for violent crimes in a ethnic group after adjusting for the number of past arrests for this ethnic group for violent crimes given that the number of stops here is small. However, for larger number of crimes, further information and data needs to be collected in order to hone our model further to account for the overdispersion.