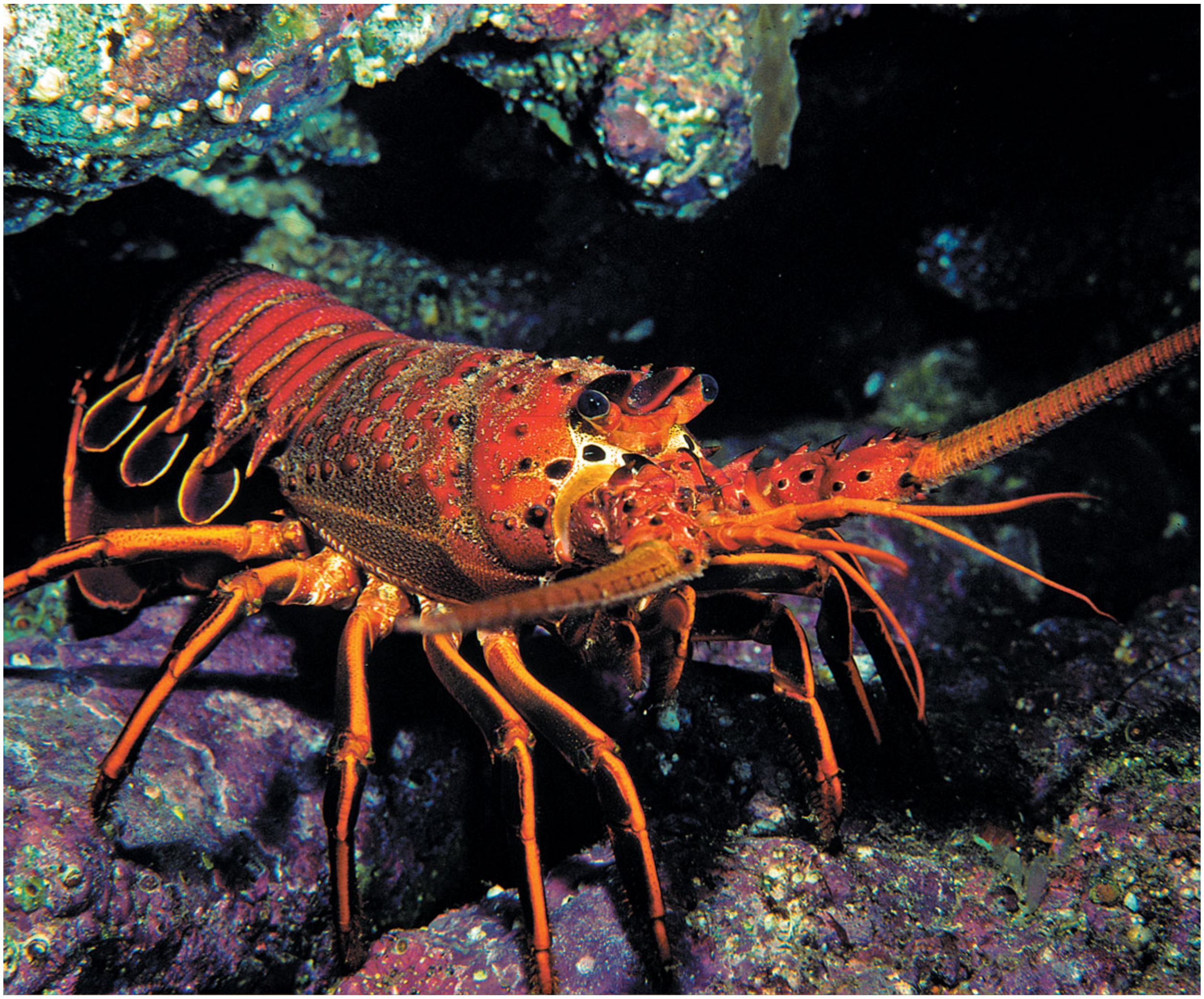
Assignment 1

California Spiny Lobster (*Panulirus Interruptus*): Assessing the Impact of Marine Protected Areas (MPAs) at 5 Reef Sites in Santa Barbara County

Stephan Kadonoff

1/8/26



Assignment Instructions:

- Working with partners to troubleshoot code and concepts is encouraged! If you work with a partner, please list their name next to yours at the top of your assignment so Annie and I can easily see who collaborated.

- All written responses must be written independently (**in your own words**).
- Please follow the question prompts carefully and include only the information each question asks in your submitted responses.
- Submit both your knitted document and the associated `RMarkdown` or `Quarto` file.
- Your knitted presentation should meet the quality you'd submit to research colleagues or feel confident sharing publicly. Refer to the rubric for details about presentation standards.

Assignment submission (YOUR NAME): Stephan Kadonoff

```r
#install.packages("estimatr")
#install.packages("performance")
#install.packages("jtools")
#install.packages("interactions")

library(tidyverse)
library(here)
library(janitor)
library(estimatr)
library(performance)
library(jtools)
library(gt)
library(gtsummary)
library(interactions)
```

DATA SOURCE:

> Reed D. 2019. SBC LTER: Reef: Abundance, size and fishing effort for California Spiny Lobster (Panulirus interruptus), ongoing since 2012. Environmental Data Initiative. Data accessed 11/17/2019.

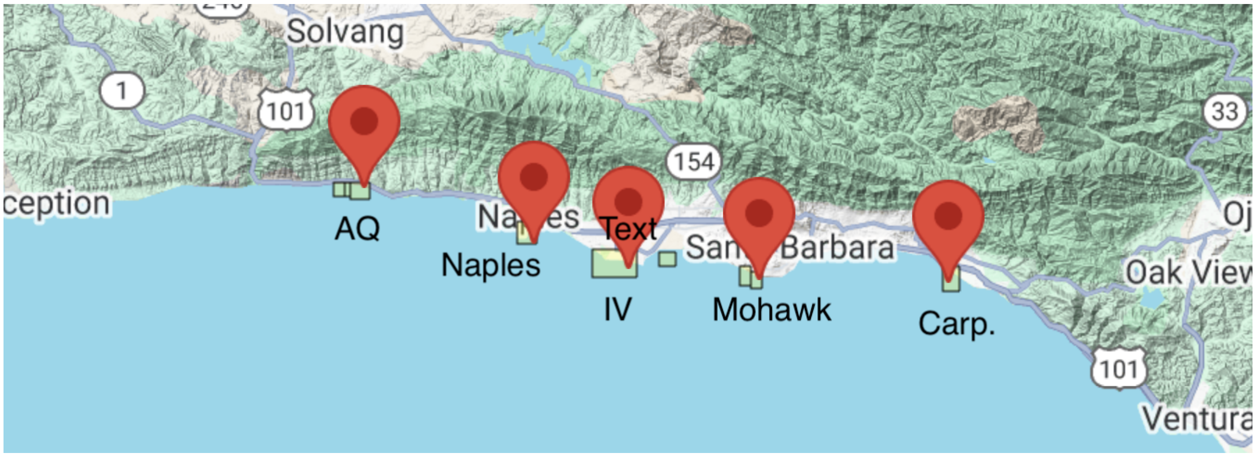
Introduction

You're about to dive into some deep data collected from five reef sites in Santa Barbara County, all about the abundance of California spiny lobsters! Data was gathered by divers annually from 2012 to 2018 across Naples, Mohawk, Isla Vista, Carpinteria, and Arroyo Quemado reefs.

Why lobsters? Well, this sample provides an opportunity to evaluate the impact of Marine Protected Areas (MPAs) established on January 1, 2012 (Reed, 2019). Of these five reefs, Naples, and Isla Vista are MPAs, while the other three are not protected (non-MPAs). Comparing lobster health between these protected and non-protected areas gives us the chance to study how commercial and recreational fishing might impact these ecosystems.

We will consider the MPA sites the `treatment` group and use regression methods to explore whether protecting these reefs really makes a difference compared to non-MPA sites (our control group). In this assignment, we'll think deeply about which causal inference assumptions hold up under the research design and identify where they fall short.

Let's break it down step by step and see what the data reveals!



Step 1: Anticipating potential sources of selection bias **a.** Do the control sites (Arroyo Quemado, Carpenteria, and Mohawk) provide a strong counterfactual for our treatment sites (Naples, Isla Vista)? Write a paragraph making a case for why this comparison is ceteris paribus or whether selection bias is likely (be specific!).

I think that while there are certain parts of these selections that help keep all else equal (identical climate, food source, pollution levels, predators, etc) but I would think a better study would be one where we are pulling from locations a bit further apart. Spatially, there is very little difference between the two MPA locations and the other AOIs, so really they are almost one giant area of study. I would think pulling from Santa Barbara, Morro Bay, and Oxnard areas might given a better picture.

Step 2: Read & wrangle data **a.** Read in the raw data from the "data" folder named `spiny_abundance_sb_18.csv`. Name the data.frame `rawdata`

b. Use the function `clean_names()` from the `janitor` package

```
# HINT: check for coding of missing values (`na = "-99999"`)

rawdata <- read_csv(here("data", "spiny_abundance_sb_18.csv"))
rawdata <- rawdata %>%
    mutate(SIZE_MM = na_if(SIZE_MM, -99999)) %>%
    clean_names()
```

c. Create a new `df` named `tidyata`. Using the variable `site` (reef location) create a new variable `reef` as a `factor` and add the following labels in the order listed (i.e., re-order the `levels`):

`"Arroyo Quemado", "Carpenteria", "Mohawk", "Isla Vista",  "Naples"`

```
# tidy up the raw data
tidydata <- rawdata %>%
    mutate(reef = factor(site,
                         levels = c("AQUE", "CARP", "MOHK", "IVEE", "NAPL"),
                         labels = c("Arroyo Quemado", "Carpenteria", "Mohawk", "Isla Vista", "Naples"))
```

Create new `df` named `spiny_counts`

d. Create a new variable `counts` to allow for an analysis of lobster counts where the unit-level of observation is the total number of observed lobsters per `site`, `year` and `transect`.

- Create a variable `mean_size` from the variable `size_mm`

- NOTE: The variable `counts` should have values which are integers (whole numbers).
- Make sure to account for missing cases (`na`)!

e. Create a new variable `mpa` with levels `MPA` and `non_MPA`. For our regression analysis create a numerical variable `treat` where MPA sites are coded `1` and non_MPA sites are coded `0`

```
#HINT(d): Use `group_by()` & `summarize()` to provide the total number of lobsters observed at each sit

#HINT(e): Use `case_when()` to create the 3 new variable columns

spiny_counts <- tidydata %>%
    group_by(site, year, transect) %>%
    summarize(counts = sum(count, na.rm = TRUE),
              mean_size = mean(size_mm, na.rm = TRUE)) %>%
    mutate(mpa = case_when(site %in% c("IVEE", "NAPL") ~ "MPA",
                           site %in% c("AQUE", "CARP", "MOHK") ~ "not_MPA",
                           TRUE ~ NA_character_),
           mpa = factor(mpa, levels = c("not_MPA", "MPA")),
           treat = case_when(mpa == "MPA" ~ 1,
                             mpa == "not_MPA" ~ 0,
                             TRUE ~ NA_integer_))
```

NOTE: This step is crucial to the analysis. Check with a friend or come to TA/instructor office hours to make sure the counts are coded correctly!

Step 3: Explore & visualize data **a.** Take a look at the data! Get familiar with the data in each `df` format (`tidydata`, `spiny_counts`)

b. We will focus on the variables `count`, `year`, `site`, and `treat`(`mpa`) to model lobster abundance. Create the following 4 plots using a different method each time from the 6 options provided. Add a layer (`geom`) to each of the plots including informative descriptive statistics (you choose; e.g., mean, median, SD, quartiles, range). Make sure each plot dimension is clearly labeled (e.g., axes, groups).

- Density plot
- Ridge plot
- Jitter plot
- Violin plot
- Histogram
- Beeswarm

Create plots displaying the distribution of lobster **counts**:

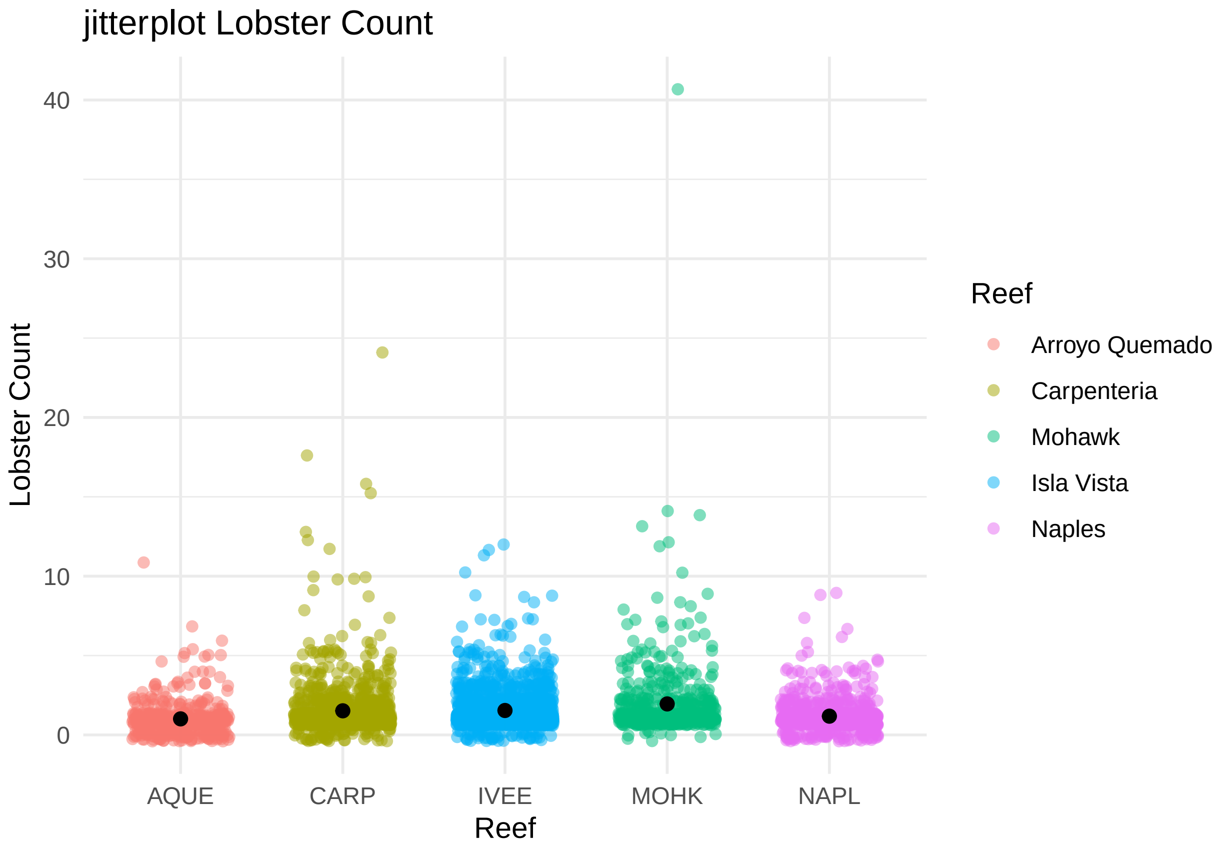
1) grouped by reef site

2) grouped by MPA status
3) grouped by year

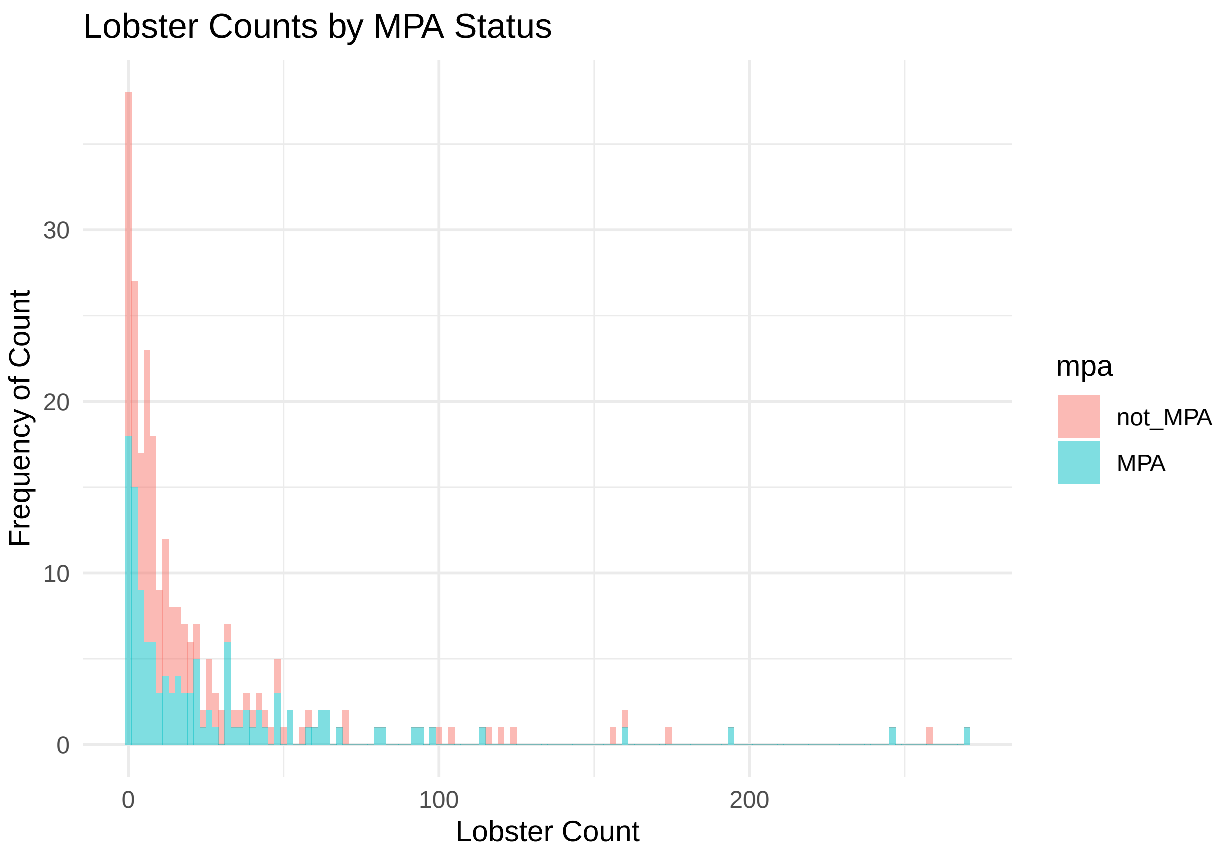
Create a plot of lobster **size** :

4) You choose the grouping variable(s)!

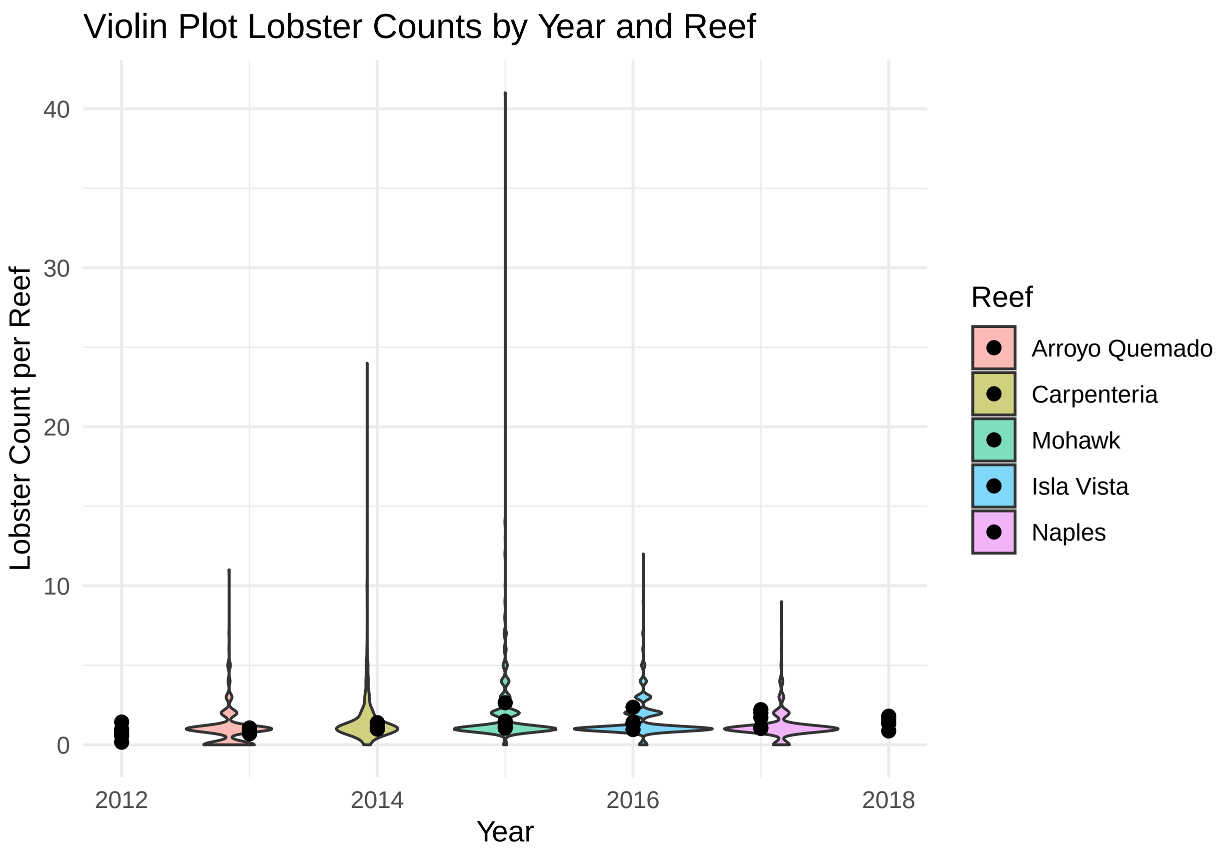
```
# plot 1: jitter counts grouped by reef site
tidydata %>%
    ggplot(aes(x = site, y = count, color = reef)) +
    geom_jitter(width = 0.3, alpha = 0.5) +
    stat_summary(fun = mean, geom = "point", size = 2, color = "black") +
    labs(title = "jitterplot Lobster Count",
         x = "Reef",
         y = "Lobster Count",
         color = "Reef") +
    theme_minimal()
```



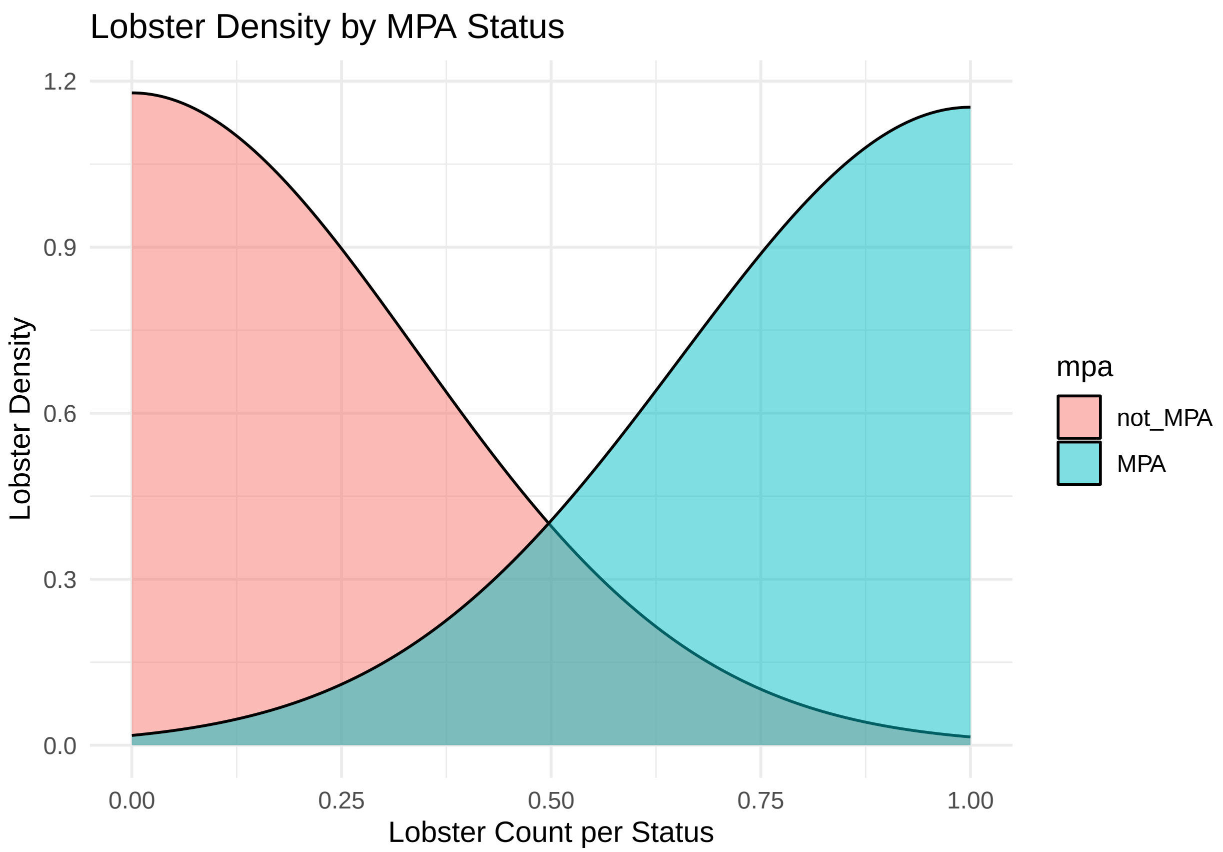
```
# plot 2: histogram counts grouped by MPA status

spiny_counts %>%
    ggplot(aes(x = counts, fill= mpa)) +
    geom_histogram(binwidth = 2, alpha = 0.5) +
    labs(title = "Lobster Counts by MPA Status",
         x = "Lobster Count",
         y = "Frequency of Count") +
    theme_minimal()
```



```r
# plot 3: Violin Plot count grouped by year
tidydata %>% #factor(year) and stat_summary(mean)
    ggplot(aes(x = year, y = count, fill = reef)) +
    geom_violin(trim = TRUE, alpha = 0.5) +
    stat_summary(fun = mean, geom = "point", size = 2, color = "black") +
    labs(title = "Violin Plot Lobster Counts by Year and Reef",
         x = "Year",
         y = "Lobster Count per Reef",
         fill = "Reef") +
    theme_minimal()
```



```
# plot 4: Density Plot mean size
spiny_counts %>%
    ggplot(aes(x = treat, color = mean_size, fill = mpa)) +
    geom_density(alpha = 0.5) +
    labs(title = "Lobster Density by MPA Status",
         x = "Lobster Count per Status",
         y = "Lobster Density") +
    theme_minimal()
```



c. Compare means of the outcome by treatment group. Using the `tbl_summary()` function from the package `gt_summary`

```
# USE: gt_summary::tbl_summary()
spiny_counts %>%
    tbl_summary(by = treat,
                statistic = list(counts ~ "{mean} ({sd})"),
                digits = counts ~ 2,
                label = list(counts ~ "Mean Lobster Count"))
```

Step 4: OLS regression- building intuition **a.** Start with a simple OLS estimator of lobster counts regressed on treatment. Use the function `summ()` from the `jtools` package to print the OLS output

b. Interpret the intercept & predictor coefficients *in your own words.* Use full sentences and write your interpretation of the regression results to be as clear as possible to a non-academic audience.

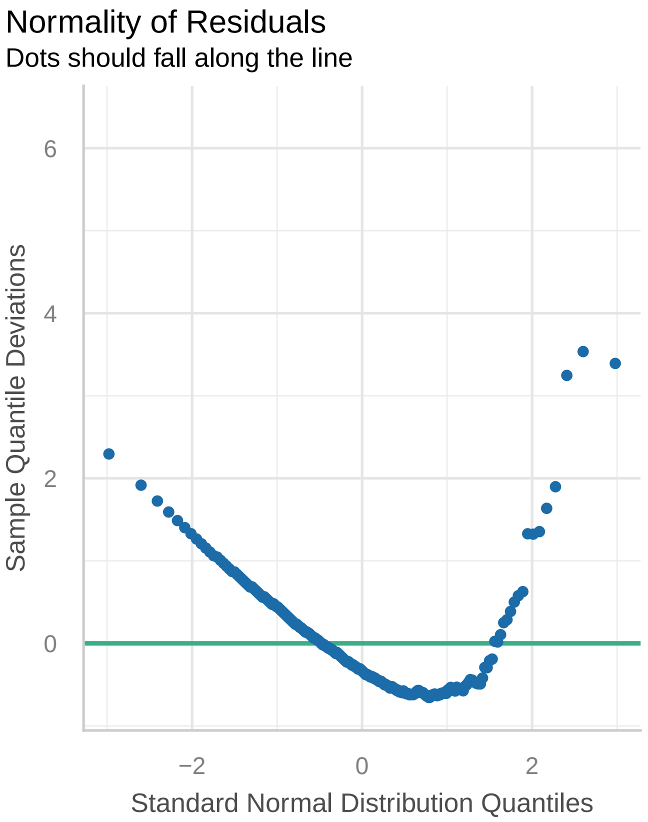
The intercept, or B0, is the average number of lobsters we would expect to count in non-MPA zone. Meaning we would see 13 lobsters, on average. With the treatment, meaning MPA zone, we would expect to see an average of 5 more lobsters than without an MPA.

```
# NOTE: We will not evaluate/interpret model fit in this assignment (e.g., R-square)


m1_ols <- lm(counts ~ treat, spiny_counts)

summ(m1_ols, model.fit = FALSE)
```

c. Check the model assumptions using the `check_model` function from the `performance` package

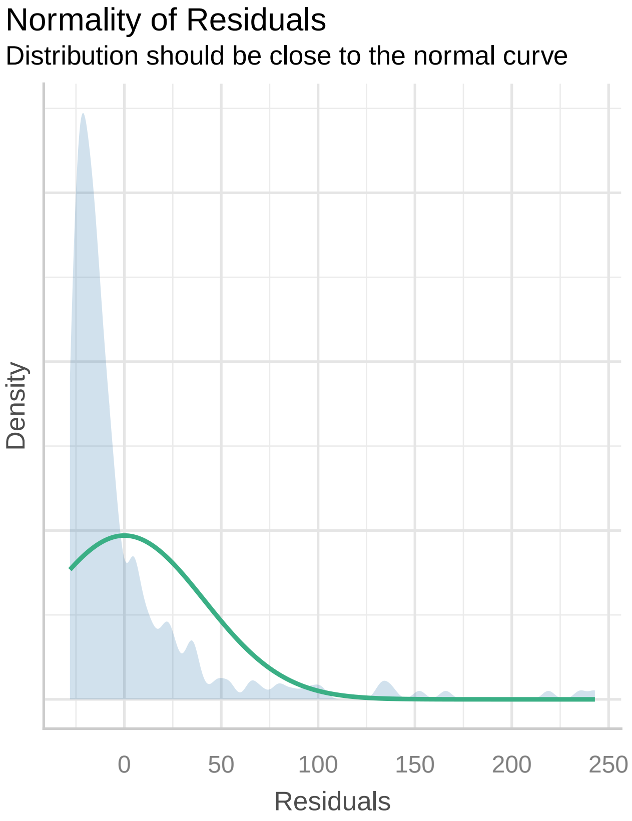
d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

```
check_model(m1_ols,  check = "qq" )
```



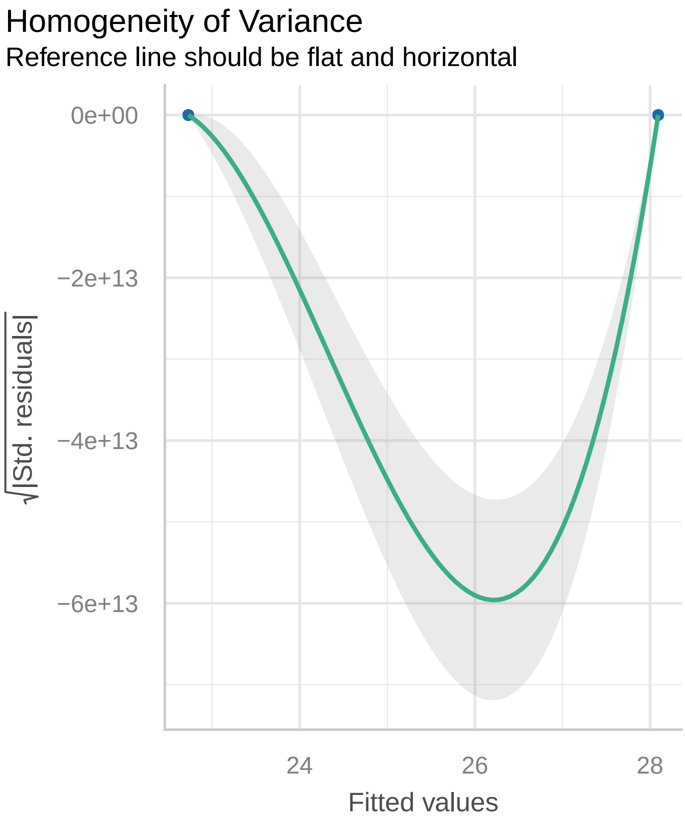
Based on the U shape of the dot distributions, I would say that the residuals are not normally distributed.

```
check_model(m1_ols, check = "normality")
```



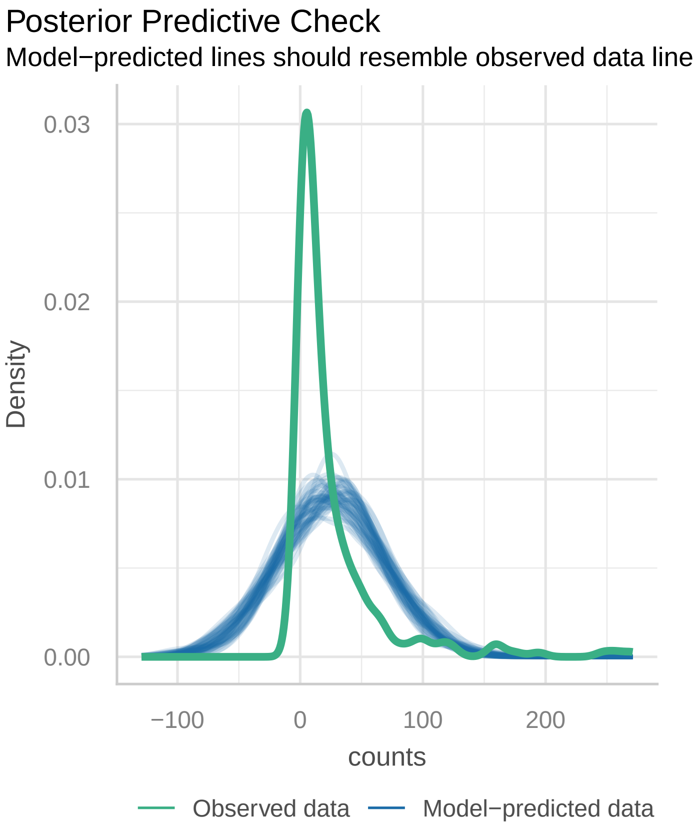
Since the residuals are distributed in a left-skewed manner, we cannot say that this model is a good fit.

```
check_model(m1_ols, check = "homogeneity")
```



Since the reference line is curving along the fitted values axis, we can say this is not a good fit.

```r
check_model(m1_ols, check = "pp_check")
```



For this model, the observed data has a lot higher density mean and a narrower standard deviation, so the model isn't the best fit.

Step 5: Fitting GLMs **a.** Estimate a Poisson regression model using the `glm()` function

b. Interpret the predictor coefficient in your own words. Use full sentences and write your interpretation of the results to be as clear as possible to a non-academic audience.

With the poisson regression model used below, we can see that there is a roughly 32% higher number of lobsters counted at treated sites versus untreated sites. So assuming all-else-equal, MPAs should have a 32% higher lobster count, on average.

c. Explain the statistical concept of dispersion and overdispersion in the context of this model.

In this model, dispersion relates to the spread of data along the model's graph. Essentially, dispersion is the how wide of a graph we expect to see. Overdispersion is, as the name would imply, when there is more data spread than we would expect with a standard model. This model has a dispersion ratio of 21.510, we see that there is a very significant amount of overdispersion.

d. Compare results with previous model, explain change in the significance of the treatment effect

The key differences between the two models lies in their respective intercepts and treatments values. While the poisson model maintains the positive treatment change, it has a smaller intercept and treatment value than the previous model. Coupled with the high amount of overdispersion, it is unlikely to be a good fit for this data.

```
#HINT1: Incidence Ratio Rate (IRR): Exponentiation of beta returns coefficient which is interpreted as
```

```
#HINT2: For the second glm() argument `family` use the following specification option `family = poisson

m2_pois <- glm(counts ~ treat,
               data = spiny_counts,
               family = poisson(link = "log"))

# summary of pois
summ(m2_pois, model.fit = FALSE)
```

```
# check predictor coefficient
exp(coef(m2_pois))
```

```
## (Intercept)       treat
##   22.729323    1.235956
```

```
check_overdispersion(m2_pois)
```

```
## # Overdispersion test
##
##        dispersion ratio =    67.033
##   Pearson's Chi-Squared = 16758.289
##                 p-value =   < 0.001
```

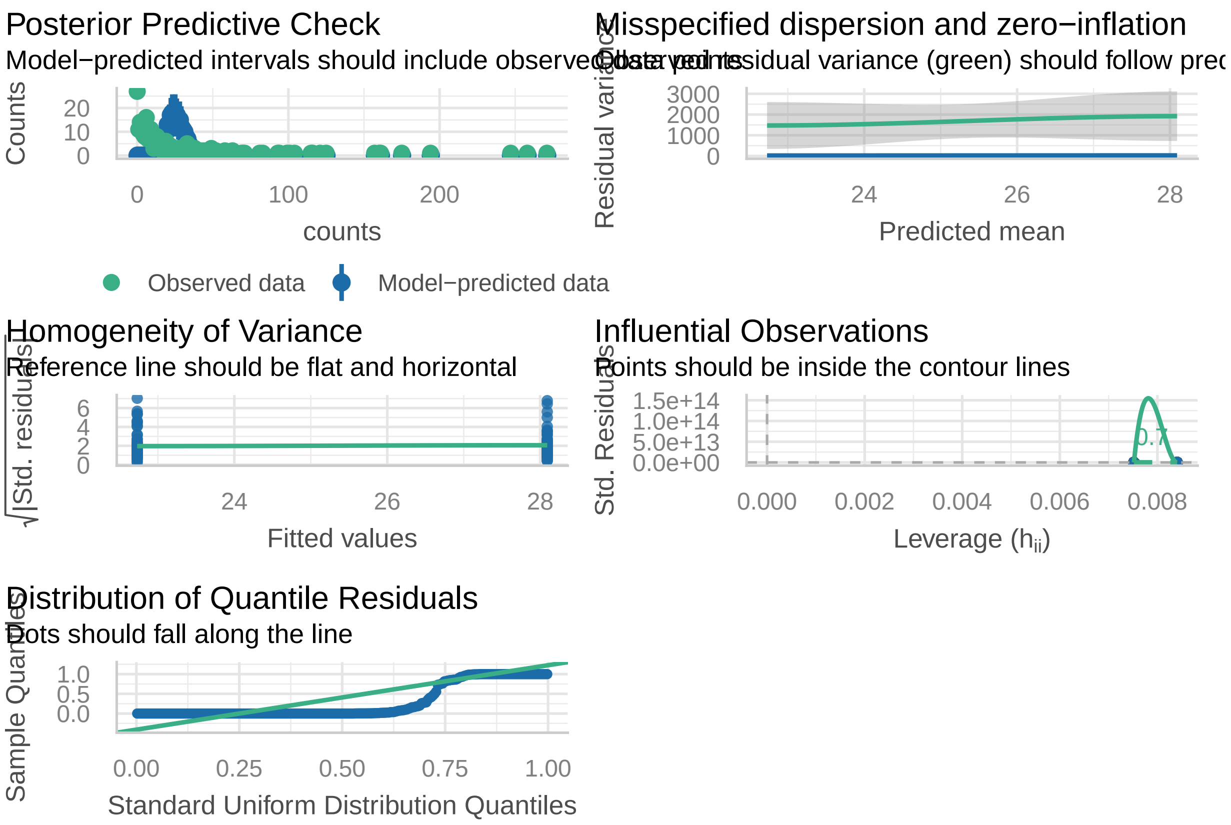
e. Check the model assumptions. Explain results.

Based on the various model checks, and how poor the data points are fitting the predictive curves/lines, this is indicative of poor model fit for this data set.

f. Conduct tests for over-dispersion & zero-inflation. Explain results.

Given the high level of overdispersion, the variance of the lobster counts are larger than expected with Poisson distribution. Likewise the zero-inflation test shows that we received 27 zeros when we had only expected 0. Combined, this tells us that a Poisson model is a poor choice for this data.

```
check_model(m2_pois)
```



```
check_overdispersion(m2_pois)
```

```
## # Overdispersion test
##
##        dispersion ratio =    67.033
##   Pearson's Chi-Squared = 16758.289
##                 p-value =   < 0.001
```

```
check_zeroinflation(m2_pois)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 27
##   Predicted zeros: 0
##             Ratio: 0.00
```

g. Fit a negative binomial model using the function glm.nb() from the package `MASS` and check model diagnostics

h. In 1-2 sentences explain rationale for fitting this GLM model. We used a negative binomial model because Poisson had high levels of over dispersion. As shown with the below model checks, the negative binomial model is a much better fit.

i. Interpret the treatment estimate result in your own words. Compare with results from the previous model. Based on the findings, some aspects of the negative binomial are a much better fit then the poisson model above. First is that the disperation ratio is less than 1. While the p-value for it is very high, it does point to a better model fit, even if we cannot reject the null due the high p-val. Secondly, the zero-inflation score is much closer to 1. Since the predicted and actual zero values are nearly identical, we can say that the dispertion is more accurate. Coupled with the results from the model checks, its safe to say the model fit is a lot better.

```r
library(MASS) ## NOTE: The `select()` function is masked. Use: `dplyr::select()` ##
```

```r
# NOTE: The `glm.nb()` function does not require a `family` argument

m3_nb <- glm.nb(counts ~ treat,
                data = spiny_counts)

summary(m3_nb)
```

```
##
## Call:
## glm.nb(formula = counts ~ treat, data = spiny_counts, init.theta = 0.5500333101,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.1237     0.1183  26.399   <2e-16 ***
## treat         0.2118     0.1720   1.232    0.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.55) family taken to be 1)
##
##     Null deviance: 302.18  on 251  degrees of freedom
## Residual deviance: 300.66  on 250  degrees of freedom
## AIC: 2088.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.5500
##           Std. Err.:  0.0466
##
##  2 x log-likelihood:  -2082.5280
```

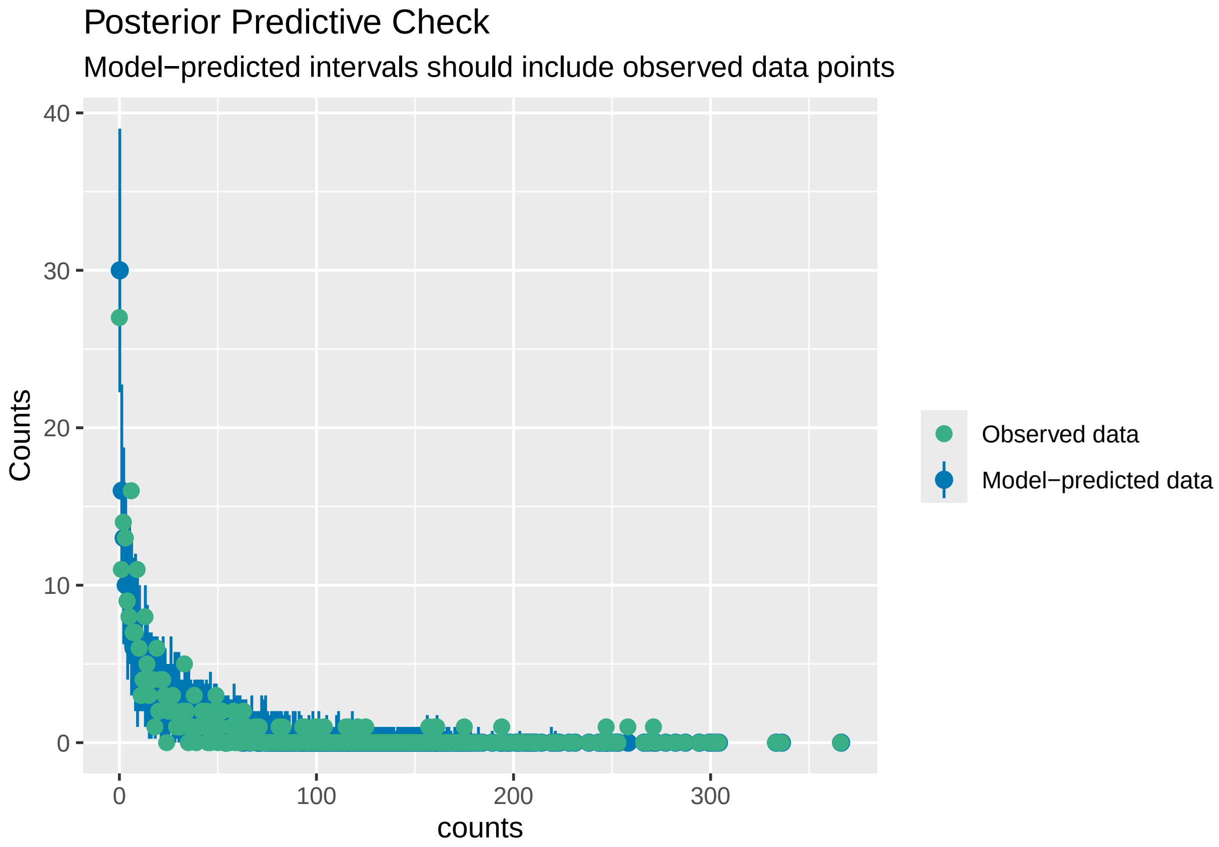
```r
check_overdispersion(m3_nb)
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.398
##           p-value = 0.088
```

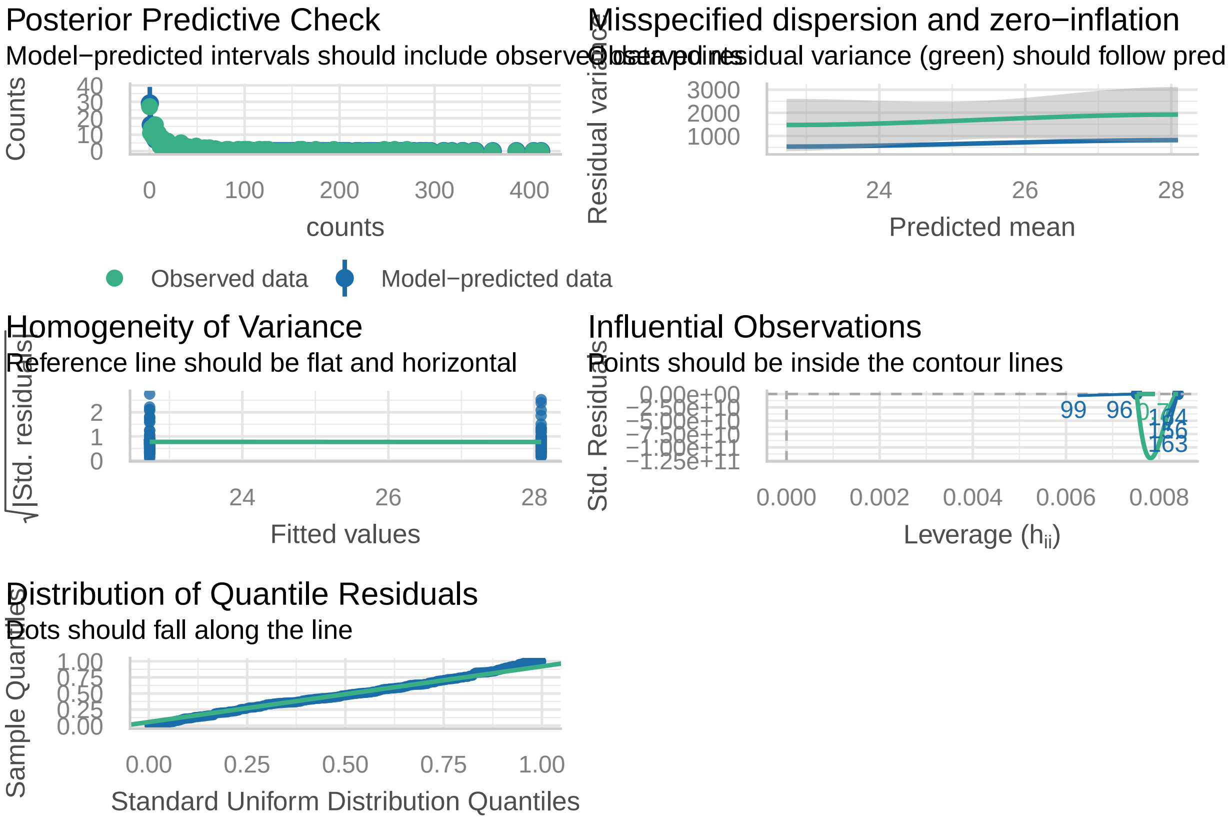
```r
check_zeroinflation(m3_nb)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 27
##   Predicted zeros: 30
##             Ratio: 1.12
```

```r
check_predictions(m3_nb)
```



```
check_model(m3_nb)
```



Step 6: Compare models **a.** Use the `export_summ()` function from the `jtools` package to look at the three regression models you fit side-by-side.

c. Write a short paragraph comparing the results. Is the treatment effect `robust` or stable across the model specifications.

Across all three models, treatment remained positive, indicating consistent model behavior with expected findings when treatments is applied to spiny lobster habitat locations. Given this consistency in behavior, it is fair to say that treatment effect is robust. At the same time, given the issues shown across the different models, such as overdispersion, zero-inflation scores, and differences in treatment effect, these models cannot determine the true effect of the treatment beyond that it positively impacts lobster count.

```
export_summs(m1_ols, m2_pois, m3_nb,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none")
```

Step 7: Building intuition - fixed effects **a.** Create new `df` with the `year` variable converted to a factor

b. Run the following negative binomial model using `glm.nb()`

- Add fixed effects for `year` (i.e., dummy coefficients)
- Include an interaction term between variables `treat` & `year` (`treat*year`)

c. Take a look at the regression output. Each coefficient provides a comparison or the difference in means for a specific sub-group in the data. Informally, describe the what the model has estimated at a conceptual level

(NOTE: you do not have to interpret coefficients individually)

Unlike the above models which had simpler syntax that did not specify dummy coefficients or an interaction term, this model does have those components. What this model shows is the effect of treatment vs non-treatment on lobster counts across different years.

d. Explain why the main effect for treatment is negative? *Does this result make sense?

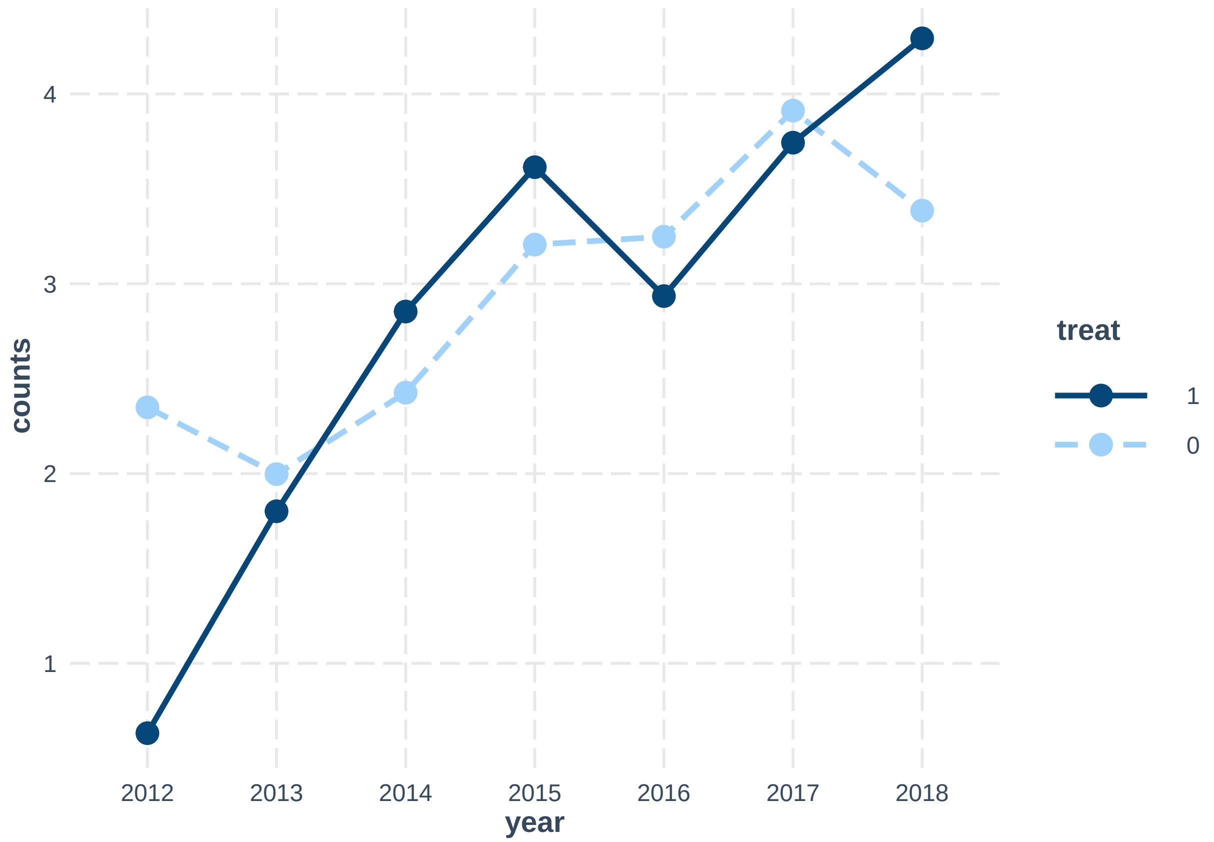
Yes. Since we added year as a dummy coefficient, we are now essentially only comparing it to the reference year, which is the earliest year. While both plots begin with treated counts below untreated, it would initially say that the treatment is ineffectual. But since estimate trends more positively for treated than un-treated zones, on average, as the years increase, we can assume a positive impact as the treatment's effect occurs year over year. This is likely due to the interaction term.

```
ff_counts <- spiny_counts %>%
    mutate(year=as_factor(year))

m5_fixedeffs <- glm.nb(
    counts ~
        treat +
        year +
        treat*year,
    data = ff_counts)

summ(m5_fixedeffs, model.fit = FALSE)
```

e. Look at the model predictions: Use the `interact_plot()` function from package `interactions` to plot mean predictions by year and treatment status.

f. Re-evaluate your responses (c) and (b) above.

```
interact_plot(m5_fixedeffs, pred = year, modx = treat,
              outcome.scale = "link") # NOTE: y-axis on log-scale
```

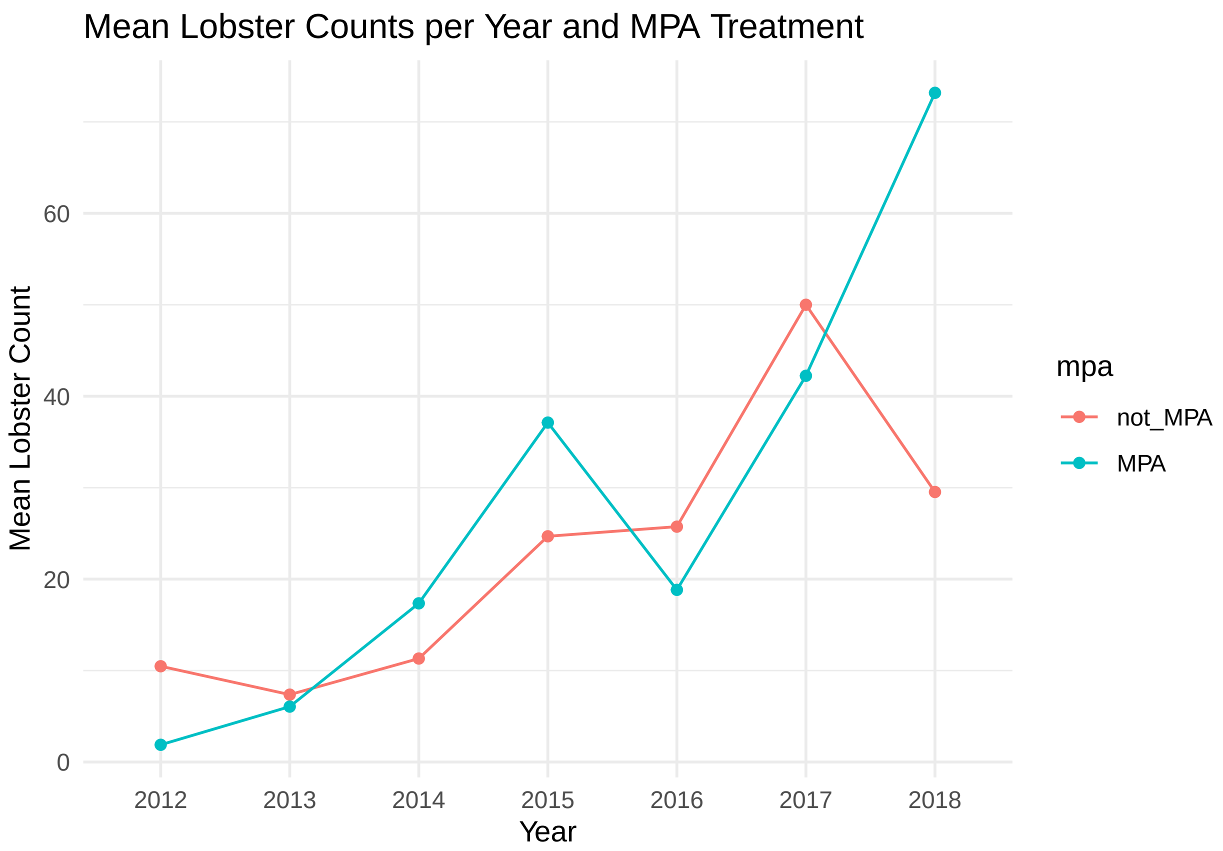


```
# HINT: Change `outcome.scale` to "response" to convert y-axis scale to counts
```

g. Using `ggplot()` create a plot in same style as the previous `interaction plot`, but displaying the original scale of the outcome variable (lobster counts). This type of plot is commonly used to show how the treatment effect changes across discrete time points (i.e., panel data).

The plot should have… - `year` on the x-axis - `counts` on the y-axis - `mpa` as the grouping variable

```
# Hint 1: Group counts by `year` and `mpa` and calculate the `mean_count`
# Hint 2: Convert variable `year` to a factor

plot_counts <- spiny_counts %>%
    mutate(year = as_factor(year)) %>%
    group_by(year, mpa) %>%
    summarize(mean_count = mean(counts), .groups = "drop")

plot_counts %>% ggplot(aes(x = year,
                           y = mean_count,
                           color = mpa,
                           group = mpa)) +
    geom_line() +
    geom_point() +
    labs(title = "Mean Lobster Counts per Year and MPA Treatment",
         x = "Year",
         y = "Mean Lobster Count") +
    theme_minimal()
```



Step 8: Reconsider causal identification assumptions

a. Discuss whether you think `spillover effects` are likely in this research context (see Glossary of terms; https://docs.google.com/document/d/1RIudsVcYhWGpqC-Uftk9UTz3PIq6stVyEpT44EPNgpE/edit?usp=sharing)

I think its certainly possible. As discussed in lecture, and based on the size, shape, and location of the MPA and non-MPA areas of interest, spillover is highly likely when observing lobster populations. "Fishing the line" can be an issue, which could impact lobsters likely to be seen in MPA areas, and potential weather events could also lead to unexpected population movements impacting how we evaluate the treatment.

b. Explain why spillover is an issue for the identification of causal effects

Spillover is an issue because it can create discrepancies in how impactful a treatment is viewed vs the control. If, for example, a day before lobster pops were counted a freak stormed pushed a bunch of lobsters into the MPA, scientists would see an outsized impact for the treatment.

c. How does spillover relate to impact in this research setting?

Since the MPA boundaries are not physical, and the lobsters cannot be explicitly told to stay in their respective areas, spillover can impact how effective MPA sites are as a treatment in protecting spiny lobster populations.

d. Discuss the following causal inference assumptions in the context of the MPA treatment effect estimator. Evaluate if each of the assumption are reasonable:

1) SUTVA: Stable Unit Treatment Value assumption
2) Excludability assumption

1. SUTVA means that the outcome of one unit is not affected by the treatments status of another. I think in this case, SUTVA is a reasonable assumption for our target. Lobsters are not going to be impacted, or change their behavior simply because they have been marked as either a control or treated lobster.
2. For the Exogeneity assumption, its a bit harder to tell. Since there are so many factors that may play in role in a study such as this, where you are applying a geographic boundary to an area where no such boundary is physically present, its hard to say definitively that there is no impact on the treatment.

EXTRA CREDIT

Use the recent lobster abundance data with observations collected up until 2024 (`extracredit_sblobstrs24.csv`) to run an analysis evaluating the effect of MPA status on lobster counts using the same focal variables.

a. Create a new script for the analysis on the updated data
b. Run at least 3 regression models & assess model diagnostics
c. Compare and contrast results with the analysis from the 2012-2018 data sample (~ 2 paragraphs)

Characteristic	**0** N = 133[1]	**1** N = 119[1]
site		
AQUE	49 (37%)	0 (0%)
CARP	63 (47%)	0 (0%)
IVEE	0 (0%)	56 (47%)
MOHK	21 (16%)	0 (0%)
NAPL	0 (0%)	63 (53%)
year		
2012	19 (14%)	17 (14%)
2013	19 (14%)	17 (14%)
2014	19 (14%)	17 (14%)
2015	19 (14%)	17 (14%)
2016	19 (14%)	17 (14%)
2017	19 (14%)	17 (14%)
2018	19 (14%)	17 (14%)
transect		
1	21 (16%)	14 (12%)
2	21 (16%)	14 (12%)
3	21 (16%)	14 (12%)
4	14 (11%)	14 (12%)
5	14 (11%)	14 (12%)
6	14 (11%)	14 (12%)
7	14 (11%)	14 (12%)
8	7 (5.3%)	14 (12%)
9	7 (5.3%)	7 (5.9%)
Mean Lobster Count	22.73 (38.52)	28.09 (44.03)
mean_size	73 (68, 77)	77 (71, 80)
Unknown	15	12
mpa		
not_MPA	133 (100%)	0 (0%)
MPA	0 (0%)	119 (100%)

[1] n (%); Mean (SD); Median (Q1, Q3)

Observations	252
Dependent variable	counts
Type	OLS linear regression

	Est.	S.E.	t val.	p
(Intercept)	22.73	3.57	6.36	0.00
treat	5.36	5.20	1.03	0.30

Standard errors: OLS

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	poisson
Link	log

	Est.	S.E.	z val.	p
(Intercept)	3.12	0.02	171.74	0.00
treat	0.21	0.03	8.44	0.00

Standard errors: MLE

	OLS	Poisson	NB
(Intercept)	22.73 ***	3.12 ***	3.12 ***
	(3.57)	(0.02)	(0.12)
treat	5.36	0.21 ***	0.21
	(5.20)	(0.03)	(0.17)

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.8129)
Link	log

	Est.	S.E.	z val.	p
(Intercept)	2.35	0.26	8.89	0.00
treat	-1.72	0.42	-4.12	0.00
year2013	-0.35	0.38	-0.93	0.35
year2014	0.08	0.37	0.21	0.84
year2015	0.86	0.37	2.32	0.02
year2016	0.90	0.37	2.43	0.01
year2017	1.56	0.37	4.25	0.00
year2018	1.04	0.37	2.81	0.00
treat:year2013	1.52	0.57	2.66	0.01
treat:year2014	2.14	0.56	3.80	0.00
treat:year2015	2.12	0.56	3.79	0.00
treat:year2016	1.40	0.56	2.50	0.01
treat:year2017	1.55	0.56	2.77	0.01
treat:year2018	2.62	0.56	4.69	0.00

Standard errors: MLE