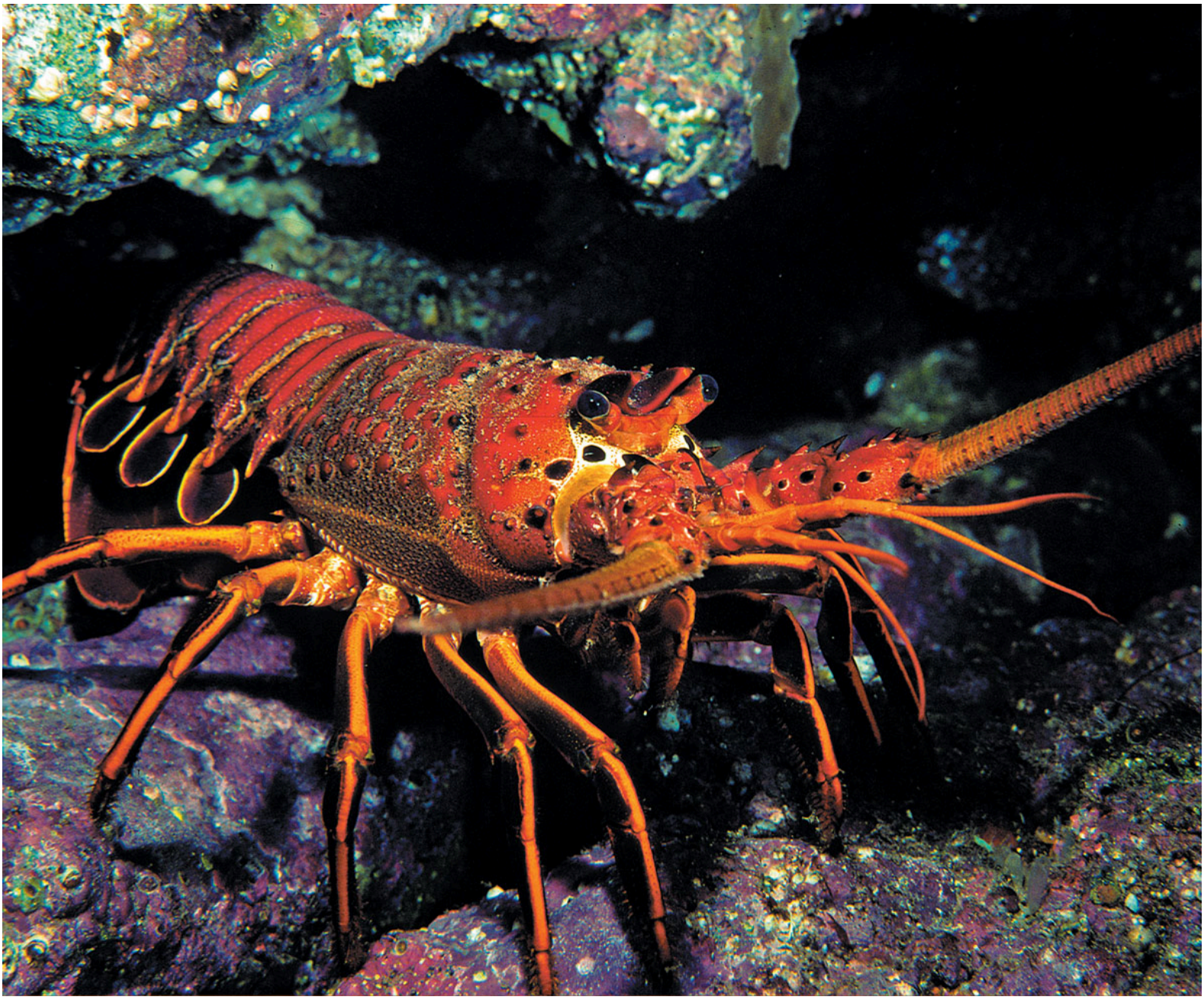
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

Assessing the Impact of Marine Protected Areas (MPAs) at 5 Reef Sites in Santa Barbara County

Ian Morris-Sibaja

1/8/2025 (Due 1/22)



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): Ian Morris-Sibaja, Haylee Oyleer

---

```
# Load libraries
library(tidyverse)
library(here)
library(janitor)
library(estimatr)
library(performance)
library(jtools)
library(gt)
library(gtsummary)
library(MASS)
library(interactions)
library(ggplot2)
library(ggridges)
library(beeswarm)
```

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. https://doi.org/10.6073/pasta/a593a675d644fdefb736750b291579a0 (https://doi.org/10.6073/pasta/a593a675d644fdefb736750b291579a0). Dataset accessed 11/17/2019.

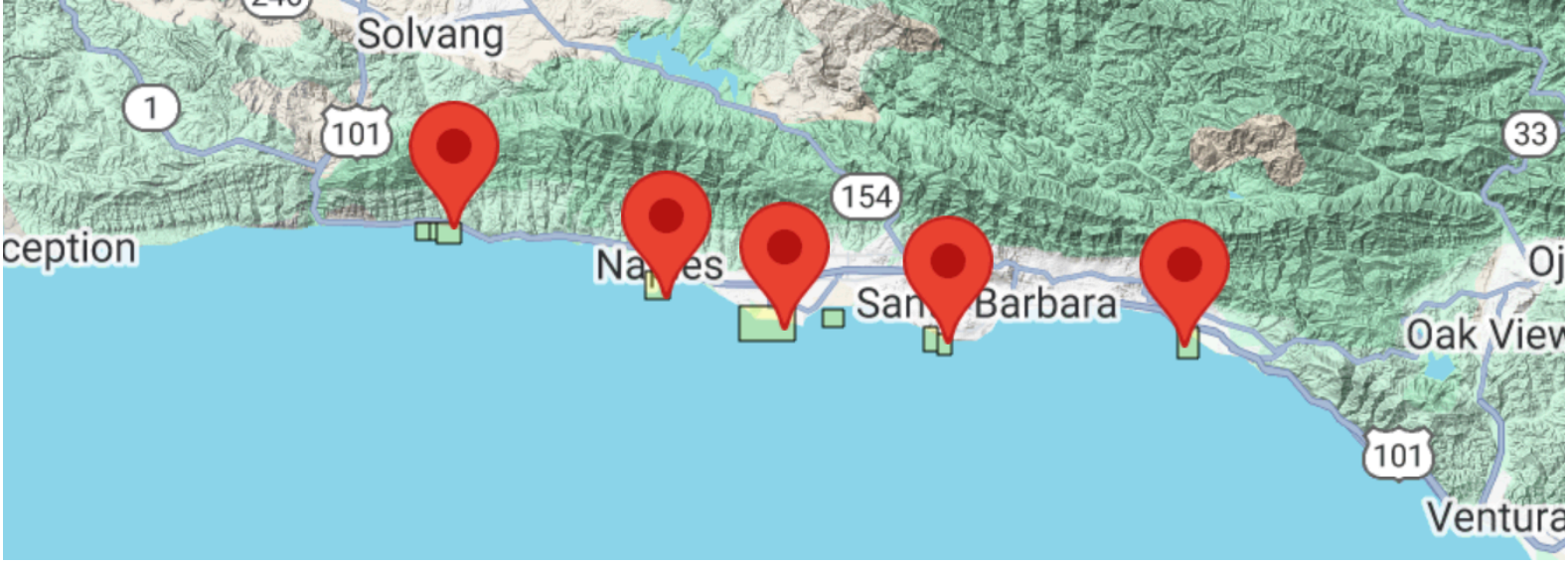
Introduction

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 centris paribus or whether selection bias is likely (be specific!).

I belive this comparison to be centris paribus, as they are in similar areas so environmental factors are not likely to be different between the two groups. This means that the only difference between the two groups is the MPA status, which is the only thing we are trying to measure.

Step 2: Read & wrangle data

a. Read in the raw data. Name the data.frame (`df`) `rawdata`

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

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

# Import spinky lobster data
rawdata <- read_csv("data/spiny_abundance_sb_18.csv", na = "-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"
```

```
# Create a new variable `reef` of type factor
tidydata <- rawdata %>%
    mutate(reef = factor(site, order = TRUE,
                         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 observe
d at each site-year-transect row-observation.

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

# Create df spiny counts of summarized data by site, year, and transect
spiny_counts <- tidydata %>%
    group_by(site, year, transect) %>%
    summarise(mean_size = mean(size_mm, na.rm = TRUE),
              counts =  as.integer(sum(count, na.rm = TRUE))) %>%
    ungroup() %>%
    mutate(mpa = case_when(site %in% c("IVEE", "NAPL") ~ "MPA",
                           TRUE ~ "non_MPA"),
           treat = case_when(site %in% c("IVEE", "NAPL") ~ 1,
                           TRUE ~ 0)) %>%
    mutate(across(where(is.numeric), ~(ifelse(is.na(.), NA_real_, (.)))))
```

> 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 (https://r-charts.com/distribution/density-plot-group-ggplot2)
- Ridge plot (https://r-charts.com/distribution/ggridges/)
- Jitter plot (https://ggplot2.tidyverse.org/reference/geom_jitter.html)
- Violin plot (https://r-charts.com/distribution/violin-plot-group-ggplot2)
- Histogram (https://r-charts.com/distribution/histogram-density-ggplot2/)
- Beeswarm (https://r-charts.com/distribution/beeswarm/)

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

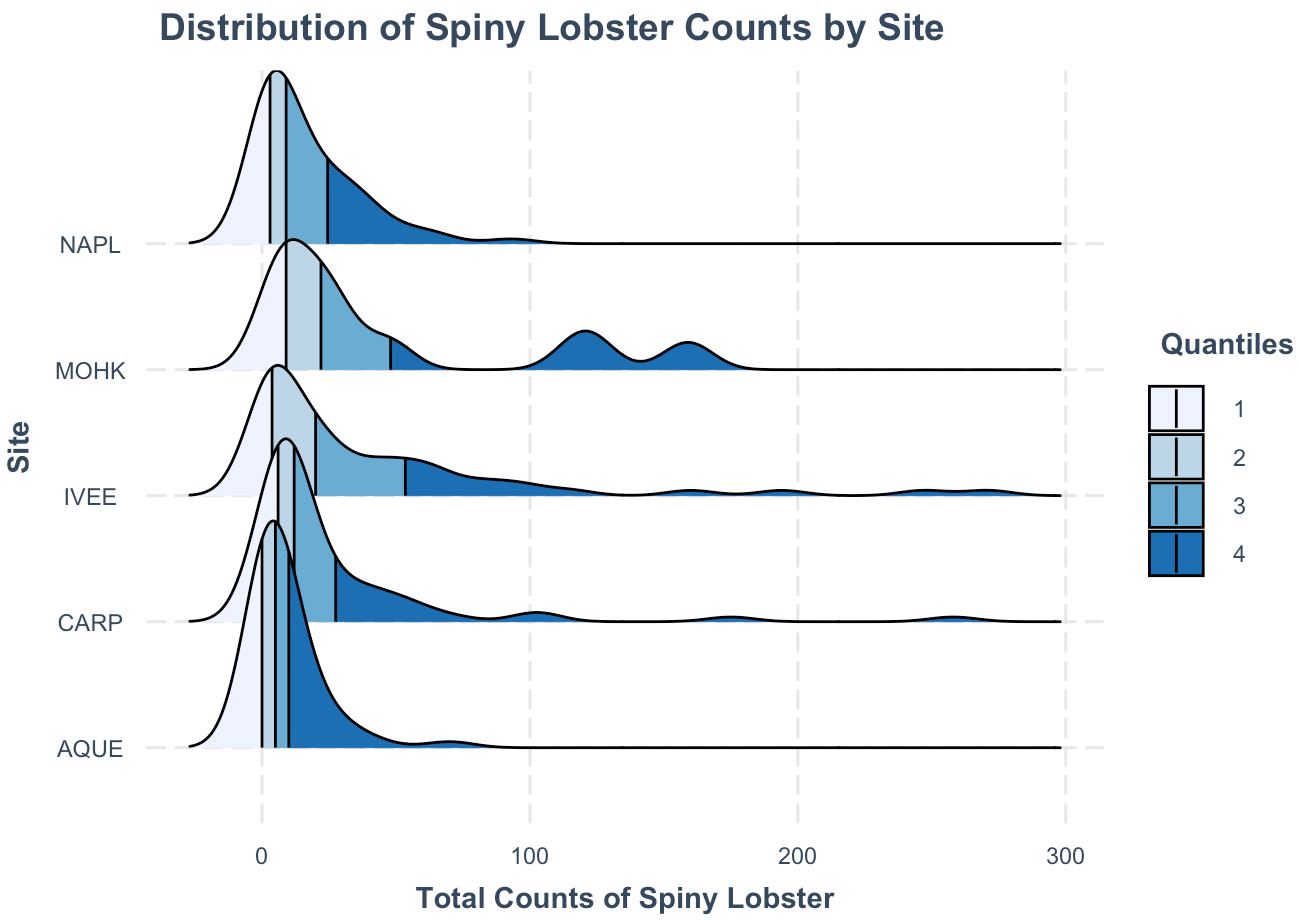
1. grouped by reef site

2. grouped by MPA status
3. grouped by year

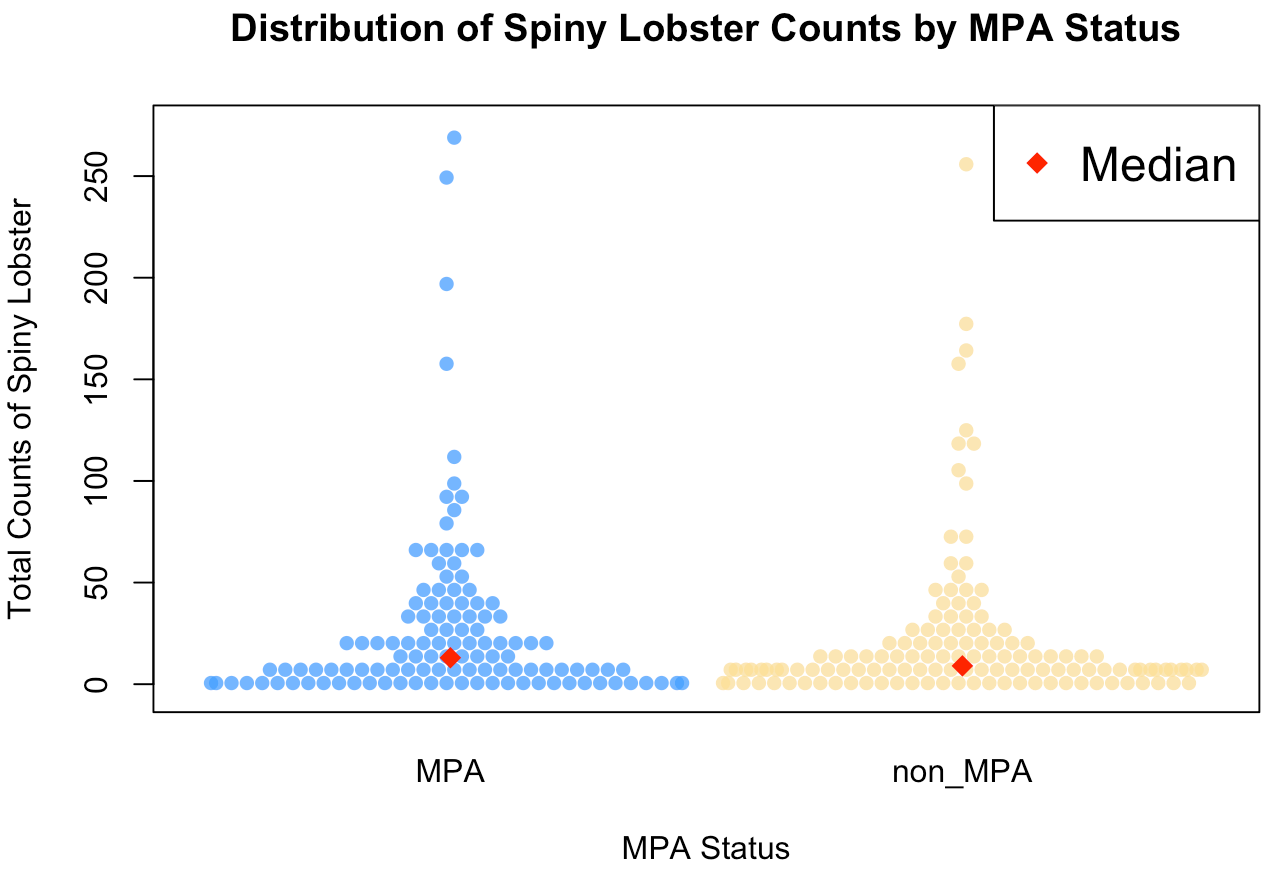
Create a plot of lobster **size** :

4. You choose the grouping variable(s)!

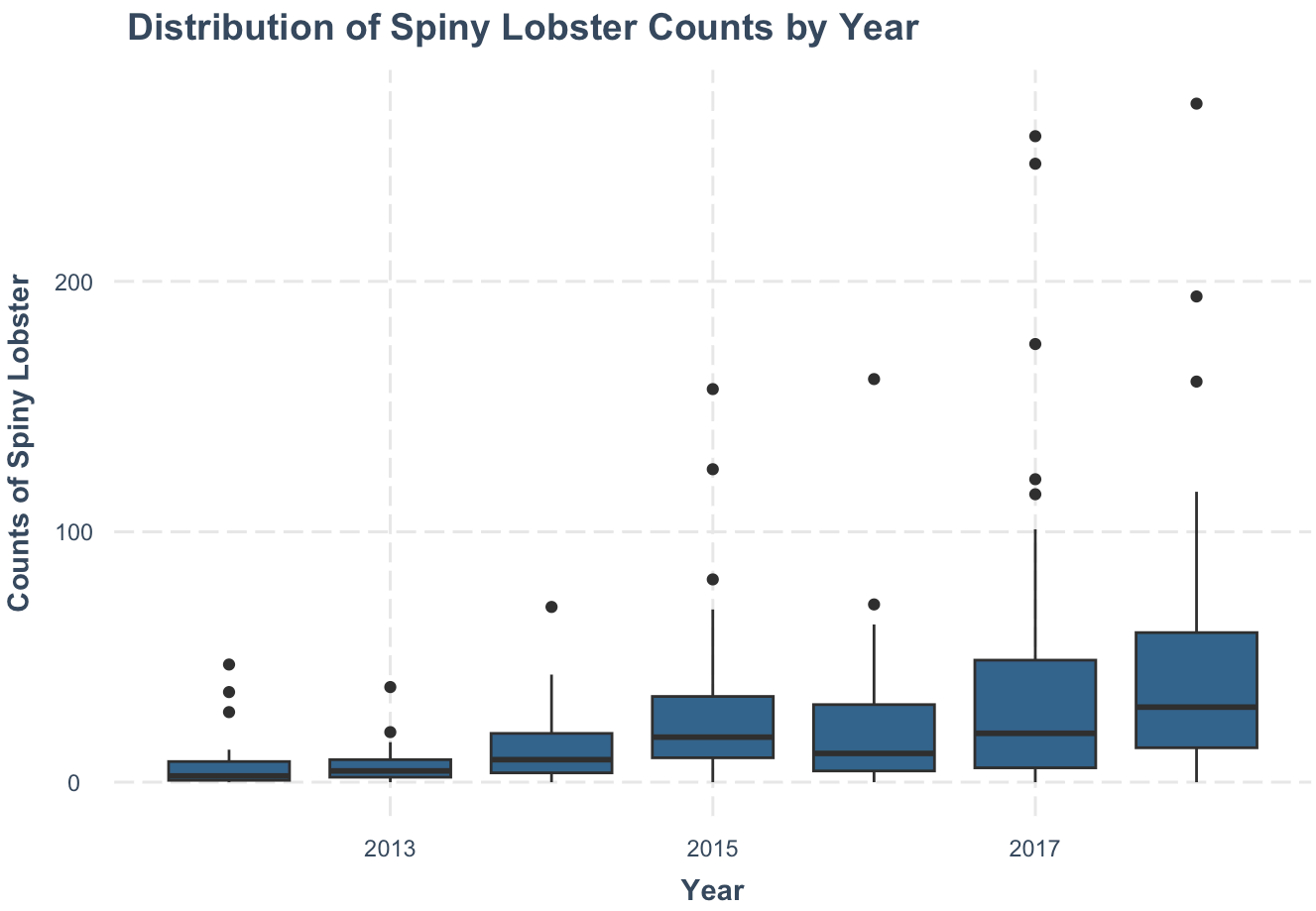
```
# plot 1: Ridgeline Plot of Spiny Lobster Count Distribution by Site
spiny_counts %>%
    ggplot(aes(x = counts, y = site, fill = after_stat(quantile))) +
  stat_density_ridges(quantile_lines = TRUE,
                      calc_ecdf = TRUE,
                      geom = "density_ridges_gradient") +
  scale_fill_brewer(name = "Quantiles") +
    labs( x = "Total Counts of Spiny Lobster",
          y = "Site",
          title = "Distribution of Spiny Lobster Counts by Site") +
    theme_nice()
```



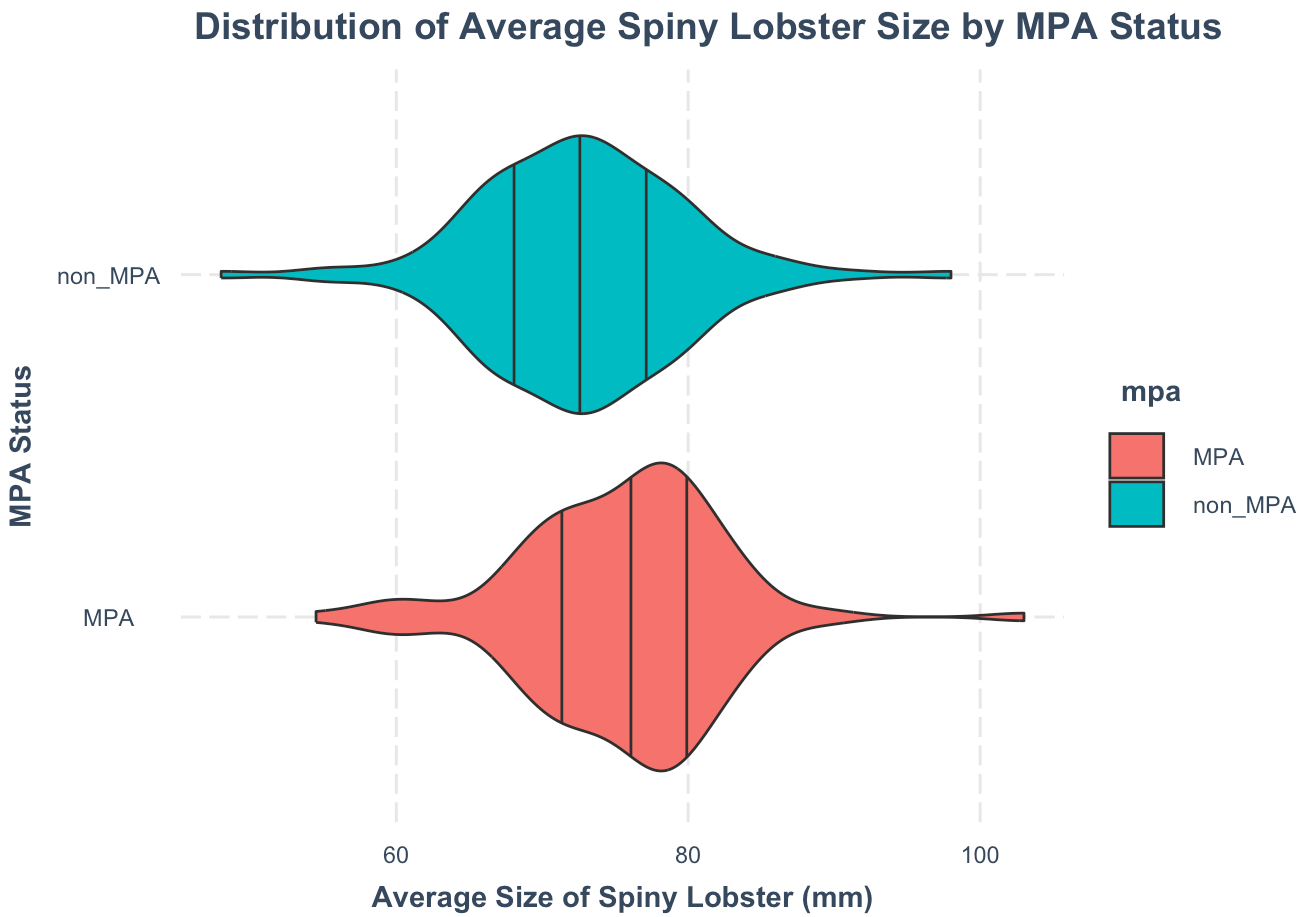
```r
# plot 2: Beeswarm Plot of Spiny Lobster Count Distribution by MPA Status

# Calculate medians
medians <- tapply(spiny_counts$counts, spiny_counts$mpa, median)

# Create beeswarm plot
beeswarm(spiny_counts$counts ~ spiny_counts$mpa,
         pch = 16,
         col = c(rgb(0.25, 0.63, 1, 0.75),
                 rgb(1, 0.88, 0.6, 0.75)),
         method = "hex", corral = "wrap",
         ylab = "Total Counts of Spiny Lobster", xlab = "MPA Status",
         main = "Distribution of Spiny Lobster Counts by MPA Status")
legend("topright", legend = c("Median"), col = "red", pch = 18, cex = 1.5)

# Overlay medians on plot
points(1:length(medians), medians, col = "red", pch = 18, cex = 1.5)
```



```
# plot 3: Bar Chart of Spiny Lobster Count Distribution by Year
spiny_counts %>%
  ggplot(aes(x = year, y = counts, group = year)) +
  theme_nice() +
  geom_boxplot(fill = "steelblue4") +
  labs(
    x = "Year",
    y = "Counts of Spiny Lobster",
    title = "Distribution of Spiny Lobster Counts by Year"
  )
```




```
# plot 4: Bar Chart of Spiny Lobster Size Distribution by Site
spiny_counts %>%
    ggplot(aes(x = mean_size, y = mpa, fill = mpa)) +
   geom_violin(draw_quantiles = c(0.25, 0.5, 0.75))  +
    labs( x = "Average Size of Spiny Lobster (mm)",
          y = "MPA Status",
          title = "Distribution of Average Spiny Lobster Size by MPA Status") +
    theme_nice()
```



c. Compare means of the outcome by treatment group. Using the `tbl_summary()` function from the package `gt_summary` (https://www.danieldsjoberg.com/gtsummary/articles/tbl_summary.html)

```
# USE: gt_summary::tbl_summary()
# Compare means by treatment
spiny_counts %>%
    dplyr::select(treat, counts, mean_size) %>%
    tbl_summary(by = treat,
                statistic = list(all_continuous() ~ "{mean} ({sd})"))
```

Characteristic	0 N = 133[1]	1 N = 119[1]
counts	23 (39)	28 (44)
mean_size	73 (7)	76 (7)
Unknown	15	12

[1] Mean (SD)

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` (https://jtools.jacob-long.com/) 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.

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

# Fit linear model
m1_ols <- lm(counts ~ treat, data = spiny_counts)

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

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

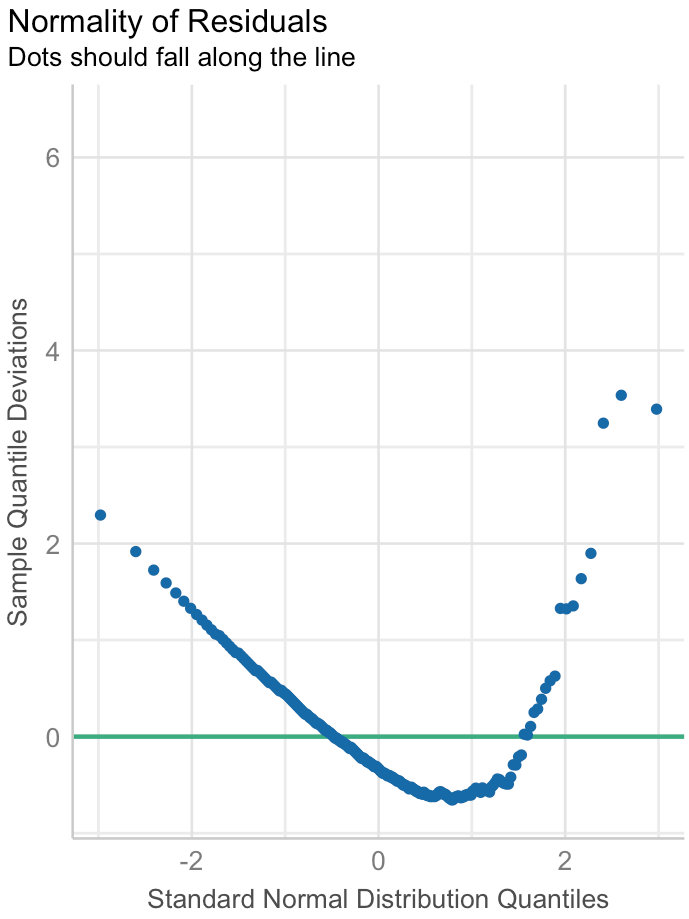
Standard errors: OLS

Non-MPA sites have ~23 lobster observations per sight. MPA sites have ~5 more lobsters than the non-MPA sights, or ~28 Lobsters.

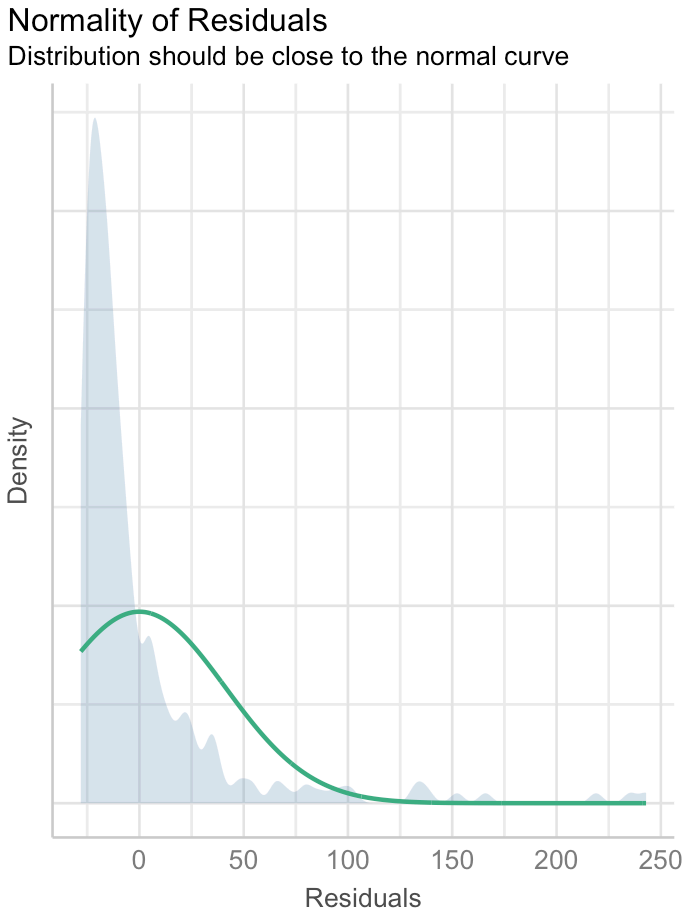
c. Check the model assumptions using the `check_model` function from the `performance` package

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

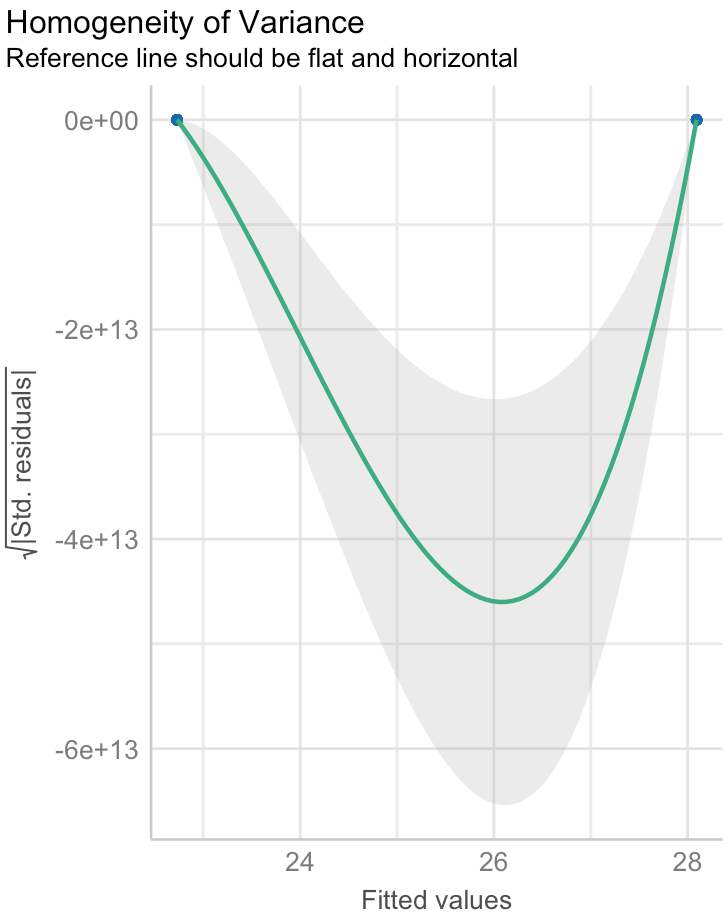
```
# Check qq plot
check_model(m1_ols,  check = "qq" )
```



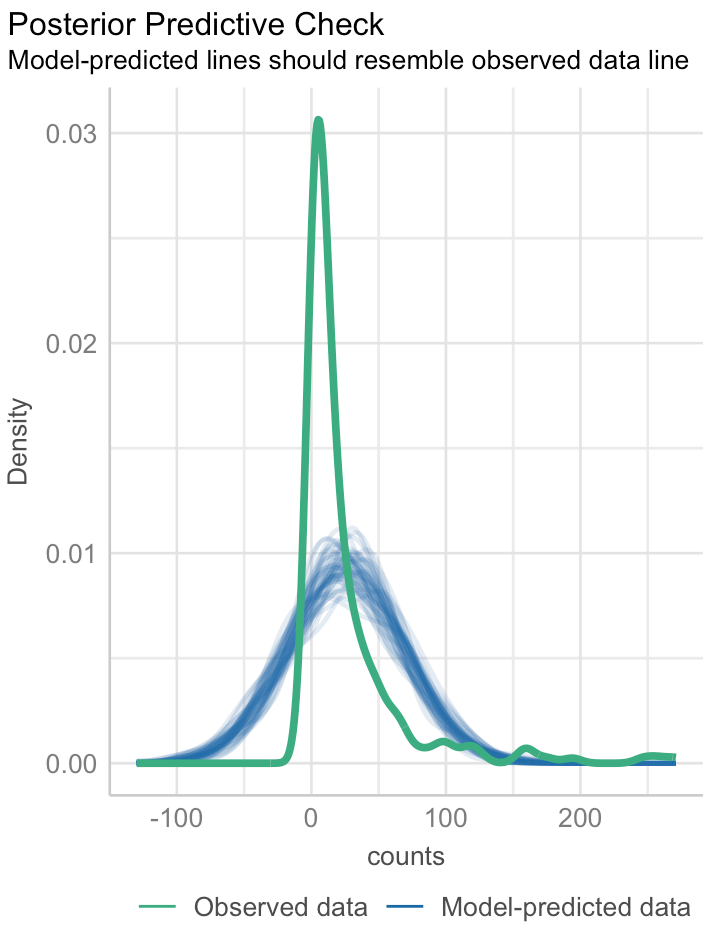

```
# Check normality
check_model(m1_ols, check = "normality")
```



```
# Check Homogeneity
check_model(m1_ols, check = "homogeneity")
```




```
# Posterior Pedictive Check
check_model(m1_ols, check = "pp_check")
```



If the distribution were normal, the residuals would fall along a normal curve. The first two plots show they do not. Rather, they vary in residual distribution greatly. An assumption of OLS is that the residuals must be normally distributed. The third plot shows that it violates the OLS assumption of homoscedasticity. This states the data must have the same variance across all values of the independent variables. The last plot predicts what the model should look like if well fitting to our data. The curve does not fit well, showing that this model likely does not fit our data. These all show that a Linear Model does not fit our data well.

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.

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

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

```
#HINT1: Incidence Ratio Rate (IRR): Exponentiation of beta returns coefficient which is in terpreted as the 'percent change' for a one unit increase in the predictor

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

# Fit a Poisson Regression Model
m2_pois <- glm(counts ~treat,
               data = spiny_counts,
               family=poisson(link = "log"))

summ(m2_pois, model.fit = FALSE)
```

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

```
print(paste("Percent Change: ", (exp(m2_pois$coefficients["treat"]) - 1) * 100))
```

```
## [1] "Percent Change:  23.5955712089655"
```

MPA sites have 23% more lobsters than the non-MPA sights on average.

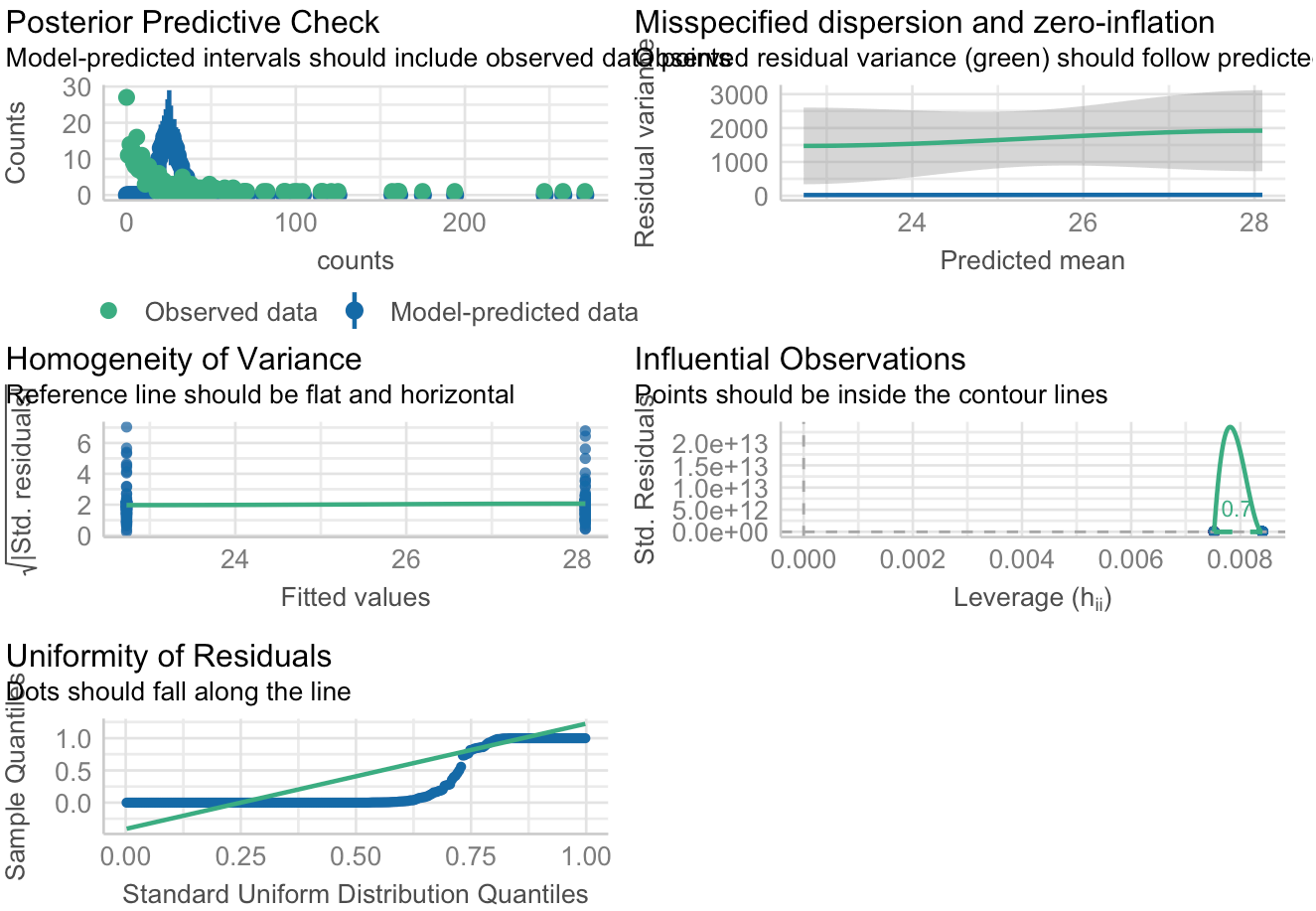
Dispersion refers to the relationship between the variance of the observed data and the mean. In our model, this involves examining whether the variance in the treated values is relatively low or high compared to the mean, which theoretically should follow a Poisson distribution. Overdispersion occurs when the variance exceeds the mean. This indicates that the model underestimates the variability in the data, potentially due to missing variables, dependency among observations (e.g., clustered or correlated counts), or an excess of zero counts in the data. In our case, overdispersion would suggest that the variance in the observed lobster counts is greater than the mean, violating the Poisson model's assumption of equidispersion and undermining its suitability for the data.

In terms of the treatment effect of the current model, the p value is significant in the poisson regression, while in the linear regression model it is not significant. This suggest that the model that is proportionally changing is much more suited to our data, instead of an absolute change in treatments.

e. Check the model assumptions. Explain results.

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

```
# Multiple model check
check_model(m2_pois)
```



Posterior Predictive Check in Poisson models check if the model capture relevant data correctly, such as median, mean and SDs. As we can see, the model predicted data and observed data do not overlap well, so it does not capture correctly. Additionally, we can see in the dispersion and zero inflation check (which checks for expected variability in the modle) that once again, the model does not capture the data well. This time, we see the residual variance is much higher in our observed model than what we would expect. Our homogeneity of Variance captures if the expected standardized residuals are equal between the predicted and observed values. We can see form the data that this holds true. Additionally, influential observations (important observations that would greatly affect important statistical calculations) are within the threshold we would expect if the model were a good fit. Lastly, under uniformity of residuals, which measures if the residuals are evenly distributed across the model, we see that our model does not capture the residuals well.

```
# Overdispersion test
check_overdispersion(m2_pois)
```

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

```
# Zero Inflation Test
check_zeroinflation(m2_pois)
```

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

Our checks show that there is evidence of zero-inflation, over dispersion, and non-uniformity of residuals in our model. All checks violate assumptions within a poisson model.

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.

This GLM model discrete data over a positive range whose sample variance exceeds the sample mean, helping in those cases when there is zero-inflation, similar to our data. Additionally, this model is used when the data is over dispersed (variance > mean), which is the case in our data.

i. Interpret the treatment estimate result in your own words. Compare with results from the previous model.

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

# Fit a binomal model
m3_nb <- MASS::glm.nb(counts ~ treat,
                      data = spiny_counts)

# Summarize Model statistics
summ(m3_nb, model.fit = FALSE)
```

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.55)
Link	log

	Est.	S.E.	z val.	p
(Intercept)	3.12	0.12	26.40	0.00
treat	0.21	0.17	1.23	0.22

Standard errors: MLE

```
# Print percent change
print(paste("Percent Change: ", (exp(m3_nb$coefficients["treat"]) - 1) * 100))
```

```
## [1] "Percent Change:  23.5955712089665"
```

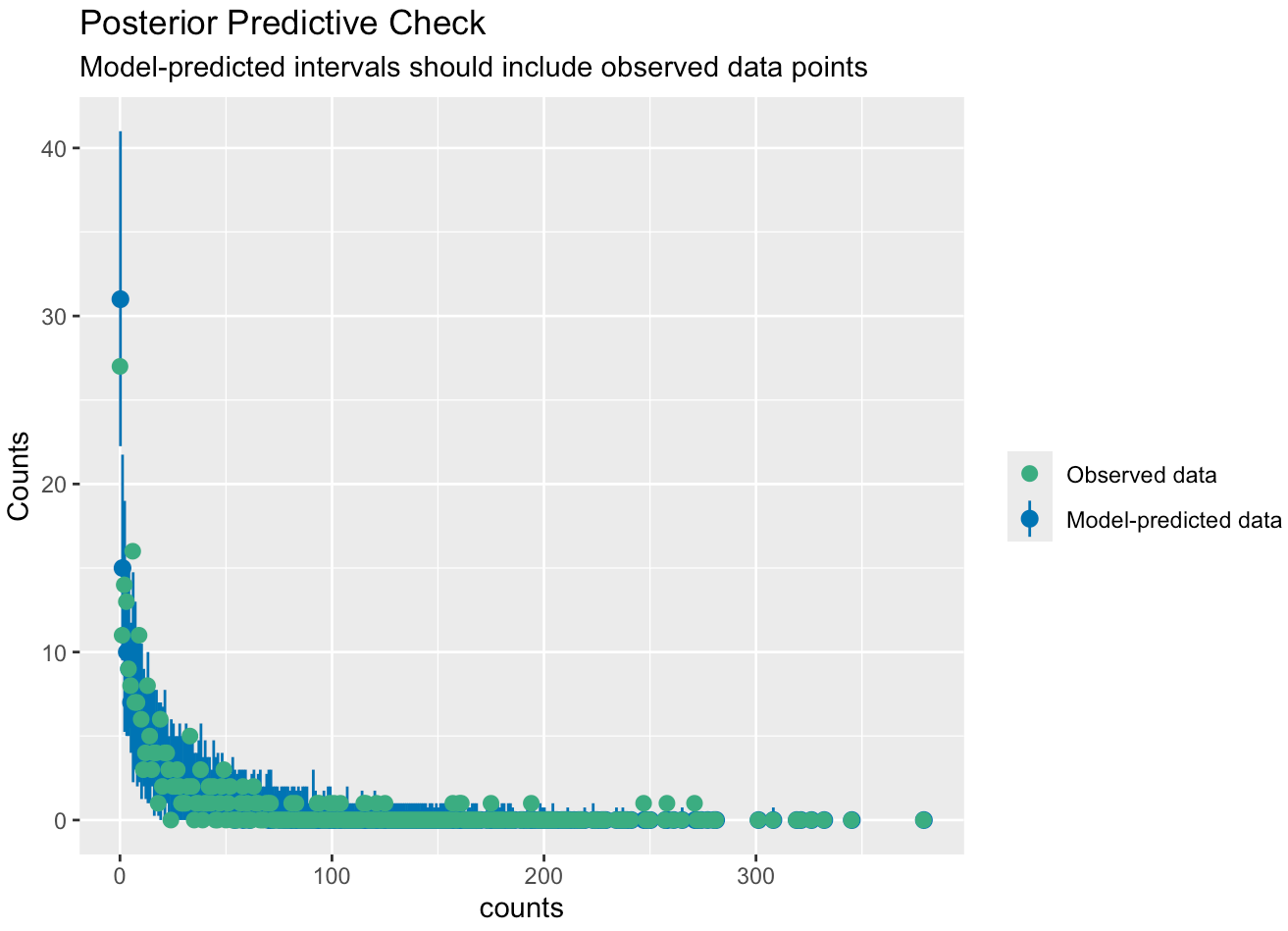
```
# Overdispersion test
check_overdispersion(m3_nb)
```

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

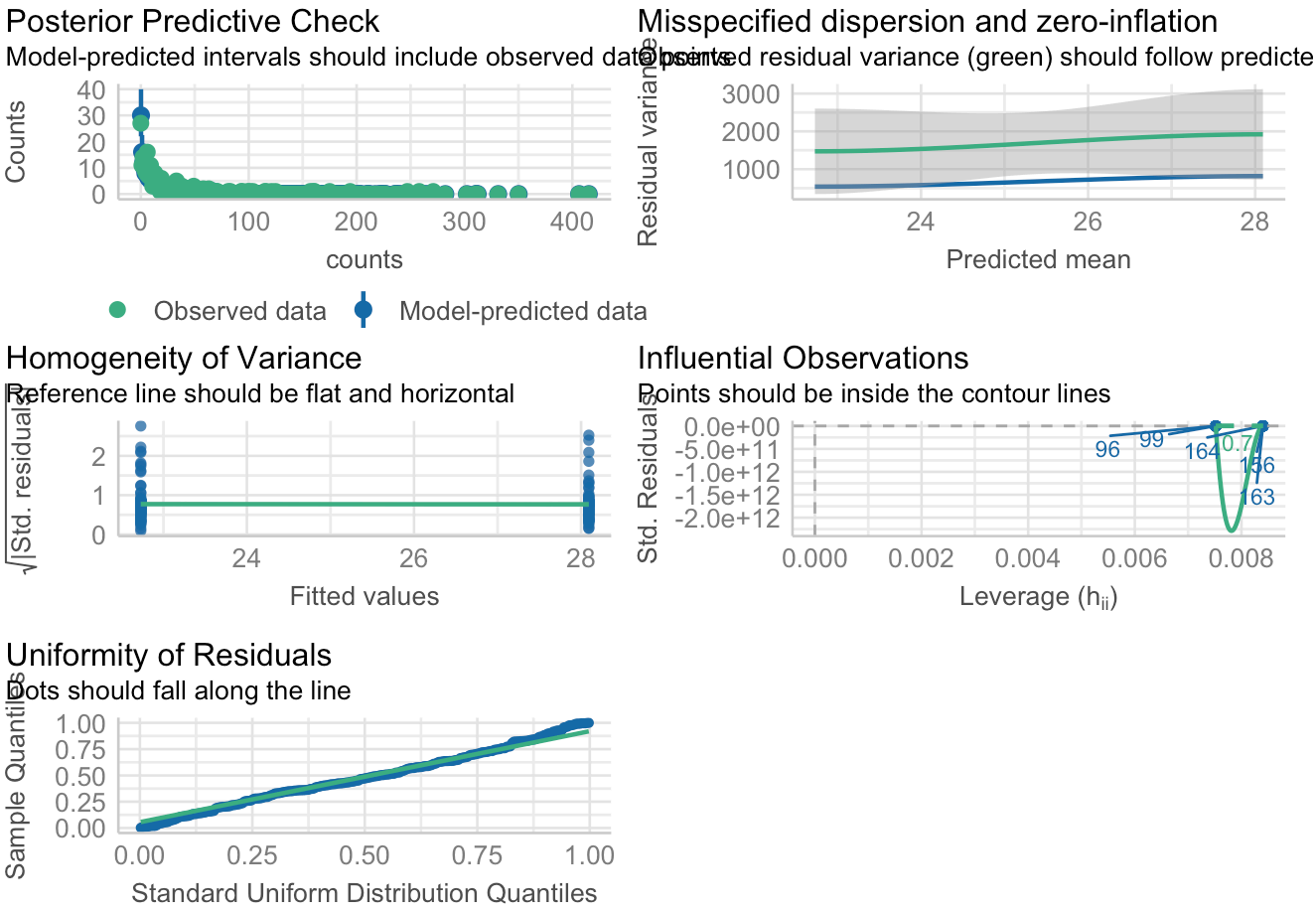
```
# Zero inflation test
check_zeroinflation(m3_nb)
```

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

```
# Posterior Predictive Test
check_predictions(m3_nb)
```




```
# Model Multiple check
check_model(m3_nb)
```



We can see that the negative binomial model is a better fit for our data than the poisson model. The negative binomial model also has a lower dispersion ratio, which is closer to 1 than the poisson model. This suggests that the negative binomial model is a better fit for our data than the poisson model. One thing to note is that the P-value is insignificant in our summary table, which means that while the model fits the data better statistically, it is likely due to chance rather than a meaningful correlation. This may be visible by looking at the misspecified dispersion and zero-inflation check, which shows that our observed residual variance is still consistently greater than our predicted residual variance.

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.

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

	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.

```
# Percent Change in each model
paste("OLS: ", (m1_ols$coefficients[2]/m1_ols$coefficients[1])*100, "% Change")
```

```
## [1] "OLS:  23.5955712089663 % Change"
```

```
paste("Poisson: ", (exp(m2_pois$coefficients[2]) - 1) * 100, "% Change")
```

```
## [1] "Poisson:  23.5955712089655 % Change"
```

```
paste("Negative Binomial: ", (exp(m3_nb$coefficients[2])-1)*100, "% Change")
```

```
## [1] "Negative Binomial:  23.5955712089665 % Change"
```

The results show that Poisson distribution is the only treatment with a significantly significant result. Both OLS and Negative Binomial Regression show that the models effect of the treated values are not stable across model specifications. However, the change in the treatment effect is similar across all models, suggesting that the treatment effect is robust across all models.

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)
- Estimate fixed effects for `year`
- 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)

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

```
# Create df with factorized year
ff_counts <- spiny_counts %>%
    mutate(year=as_factor(year))

# Create a negative binomial with factorized years
m5_fixedeffs <- glm.nb(
    counts ~
        treat +
        year +
        treat*year,
    data = ff_counts)

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

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

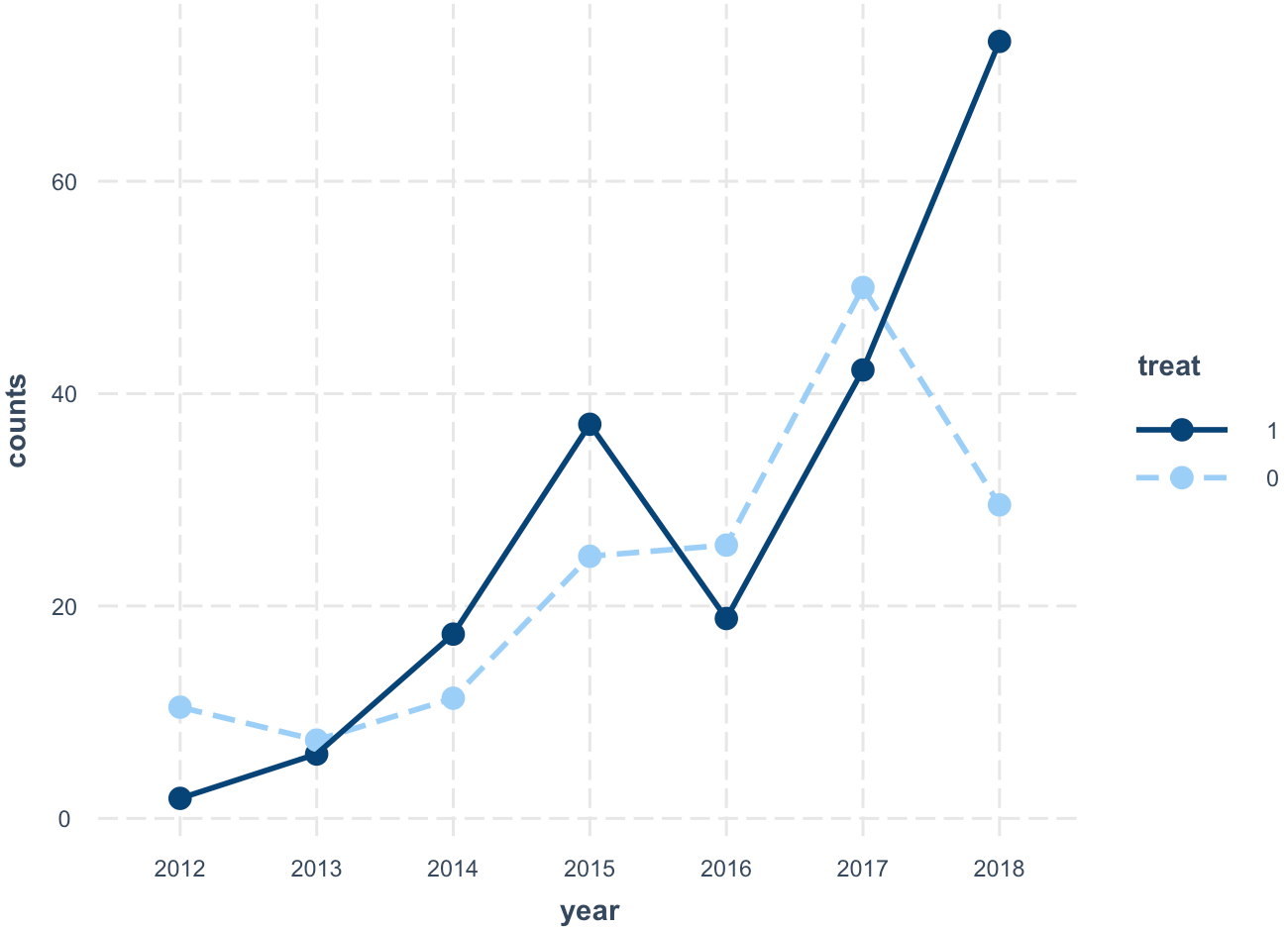
The negative binomial model predicts how the treatment effect changes over time and the affects of their interactions `treat:year` . Overall, this model suggests that the treatment variable has a negative effect on the counts of lobsters initially, meaning at year 2012, the MPA sites have fewer lobsters than the non-MPA sites. However, as time goes on, the

treatment effect becomes positive, meaning that the MPA sites generally have more lobsters than the non-MPA sites. This suggests that the MPA sites are recovering from the effects of fishing and are now supporting larger lobster populations.

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.

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

interact_plot(m5_fixedeffs, pred = year, modx = treat,
              outcome.scale = "response")
```

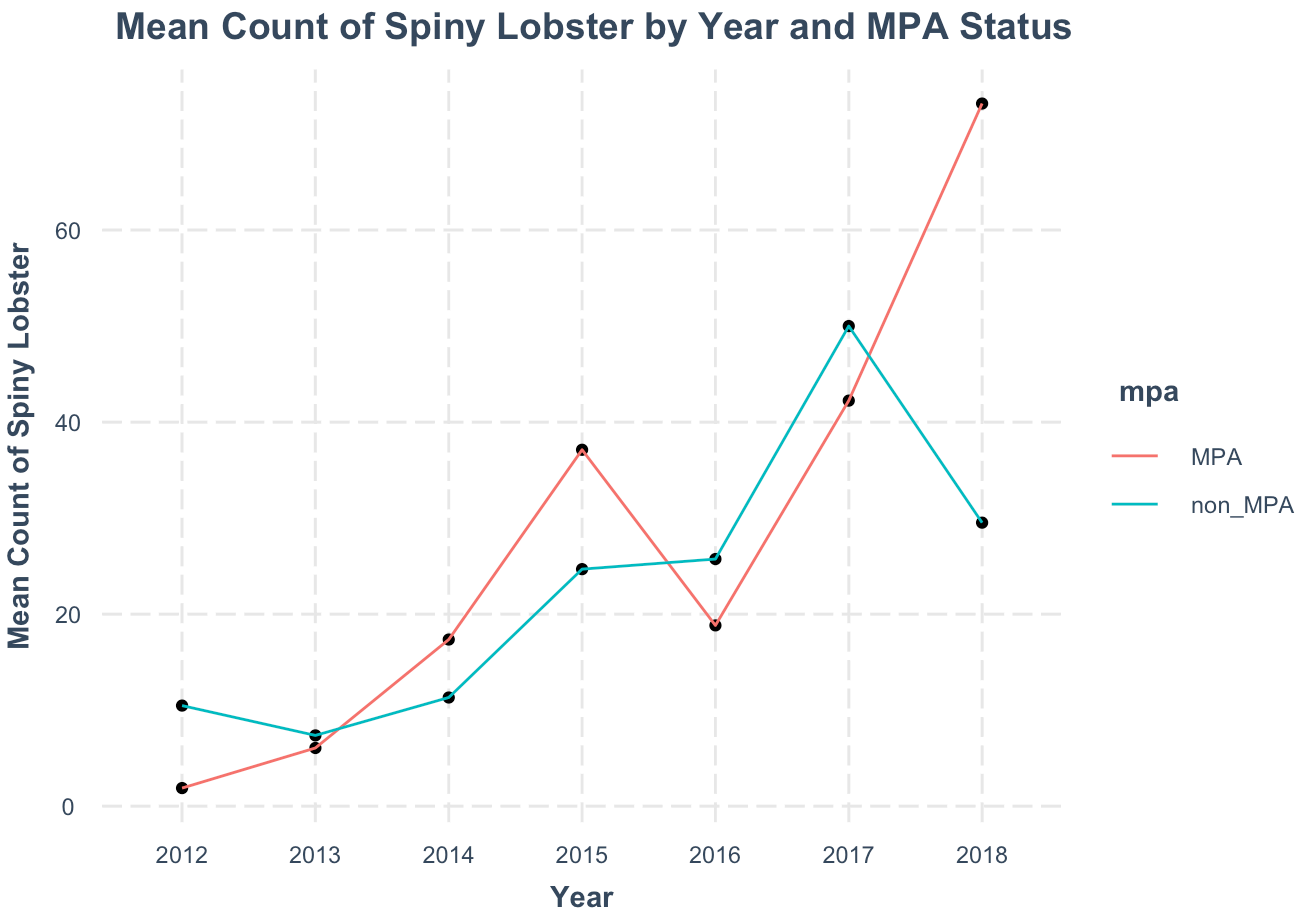


After re-evaluating the model, we can see that the treatment effect is not as stable as we once thought. The treatment effect is negative at the beginning of the study, becomes positive as time goes on, decreases between 2015-2106, then the populations begin to rise again. It is possible that MPA sites are recovering from the effects of fishing and are now supporting larger lobster populations, but there is not a clear trajectory on their population with this range.

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 <- ff_counts %>%
    group_by(year, mpa) %>%
    summarize(mean_count = mean(counts, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(year = as.factor(year))

plot_counts %>% ggplot(aes(x = year, y = mean_count, group = mpa)) +
    geom_point() +
    geom_line(aes(color = mpa)) +
    labs(x = "Year", y = "Mean Count of Spiny Lobster",
         title = "Mean Count of Spiny Lobster by Year and MPA Status") +
    theme_nice()
```



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 (https://docs.google.com/document/d/1RIudsVcYhWGpqC-Uftk9UTz3PIq6stVyEpT44EPNgpE/edit?usp=sharing))

I believe that spillover effects are likely. The MPA and non-MPA sites are located very closely to each other. The MPA sites are located near the non-MPA sites, so lobster populations can easily disperse between sites.

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

Spillover is an issue for the identification of causal effects because it violates Stable Unit Treatment Value Assumption (SUTVA). When SUTVA is violated, it means that the treatment effect is normalizing the effects across all individuals in the study equally, leading to confounding bias.

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

With sites situated so closely, the possible population intermixing could cause the treatment to not be a causation as once thought. There may be other factors (predators, fishing, etc.) that affect the lobster populations, but are not obvious within our analysis.

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 The SUTVA assumption may be violated because of the locality of the MPA and non-MPA sites, and the possibility of their populations intermixing.
2. Excludability assumption The Excludability assumption may be violated due to the other mechanisms mention above (predators, fishing, etc.) which could be affecting the lobster populations, when our model is only taking in account the MPA and non-MPA sites as the treatment.

EXTRA CREDIT

> Use the recent lobster abundance data with observations collected up until 2024 (`lobster_sbchannel_24.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

```
# Import new data
rawdata_new <- read_csv(here("data", "lobster_sbchannel_24.csv"), na = "-99999") %>%
    clean_names()

# Refactor sites
tidydata_new <- rawdata_new %>%
    mutate(reef = factor(site, order = TRUE,
                         levels = c("AQUE", "CARP",
                                    "MOHK", "IVEE", "NAPL"),
                         labels = c("Arroyo Quemado", "Carpenteria",
                                    "Mohawk", "Isla Vista", "Naples")))

# Treatment variables MPA site and mean size
spiny_counts_new <- tidydata_new %>%
    group_by(site, year, transect) %>%
    summarise(mean_size = mean(size_mm, na.rm = TRUE),
              counts =  as.integer(sum(count, na.rm = TRUE))) %>%
    ungroup() %>%
    mutate(mpa = case_when(site %in% c("IVEE", "NAPL") ~ "MPA",
                           TRUE ~ "non_MPA"),
           treat = case_when(site %in% c("IVEE", "NAPL") ~ 1,
                           TRUE ~ 0)) %>%
    mutate(across(where(is.numeric), ~(ifelse(is.na(.), NA_real_, (.)))))
```

b. Run at least 3 regression models & assess model diagnostics

Linear Model

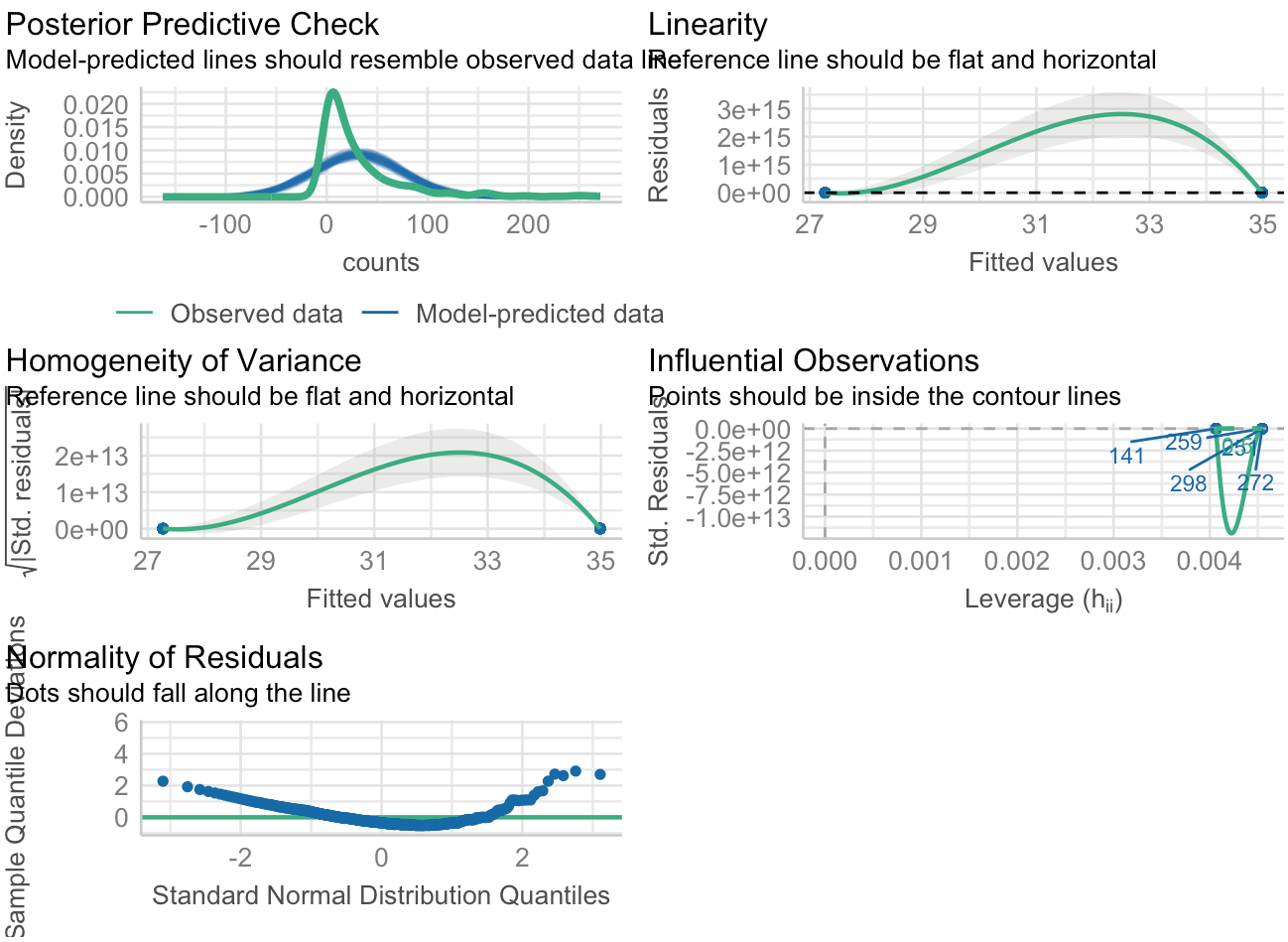
```
# Linear model
m1_ols_new <- lm(counts ~ treat, data = spiny_counts_new)

summ(m1_ols_new, model.fit = FALSE)
```

Observations	466
Dependent variable	counts
Type	OLS linear regression

	Est.	S.E.	t val.	p
(Intercept)	27.27	2.69	10.15	0.00
treat	7.72	3.91	1.97	0.05

Standard errors: OLS

```
check_model(m1_ols_new)
```



Again, the linear model does not fit our data well. Regardless of the significant P-Value in our treated variable, The residuals are not uniform, our data is overdispersed, and there is zero-inflation present. Additionally, our data does not follow our posterior distribution. The influential observations are also not within the threshold we expect. In conclusion, this is a poor fitting model.

Poisson Model

```
# Possion model of counts as a function of treatment
m2_pois_new <- glm(counts ~ treat,
                   data = spiny_counts_new,
                   family = poisson(link = "log"))

summ(m2_pois_new, model.fit = FALSE)
```

Observations	466
Dependent variable	counts
Type	Generalized linear model
Family	poisson
Link	log

	Est.	S.E.	z val.	p
(Intercept)	3.31	0.01	270.75	0.00

Standard errors: MLE

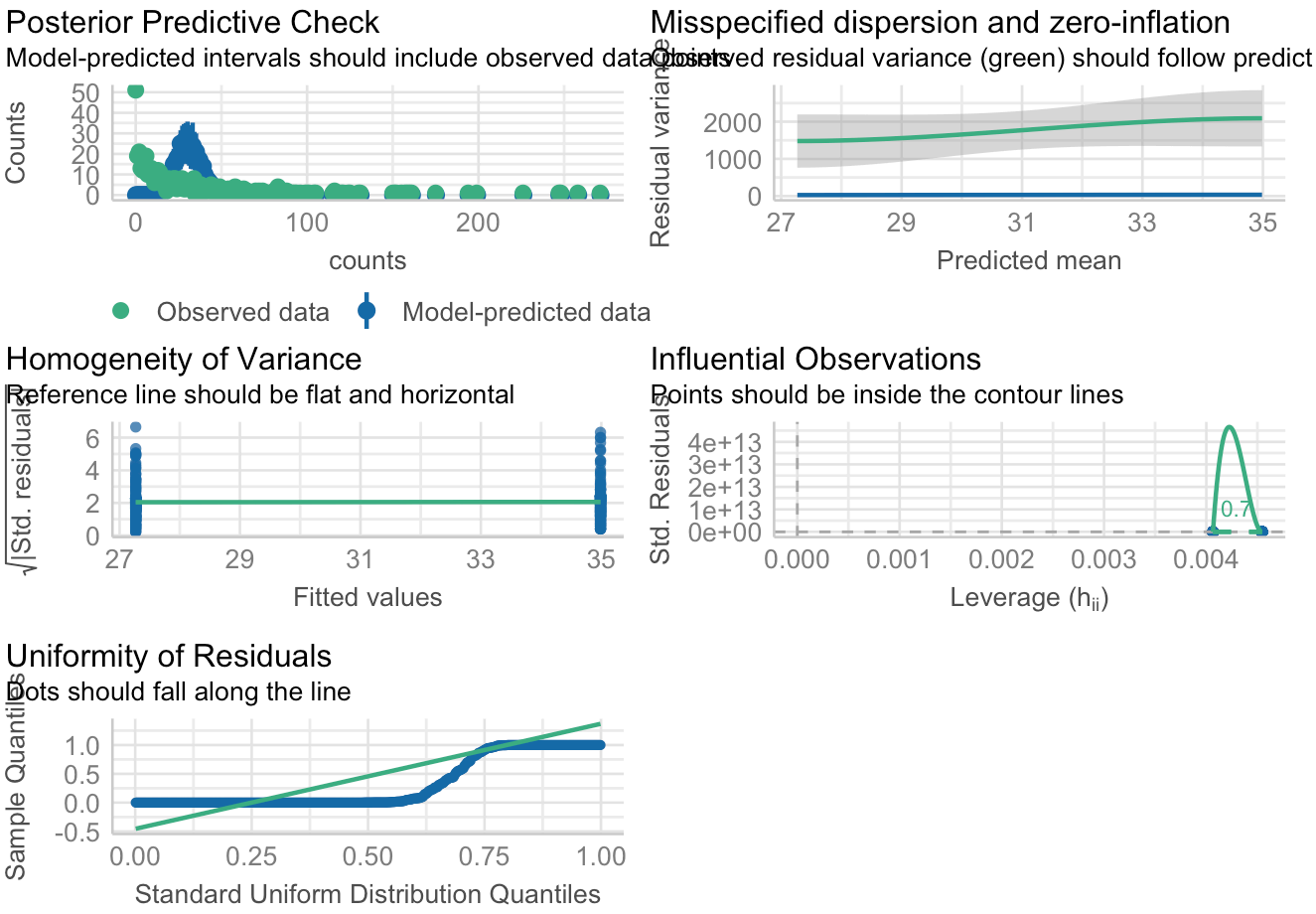
	Est.	S.E.	z val.	p
treat	0.25	0.02	14.92	0.00

Standard errors: MLE

```
print(paste("Percent Change: ", (exp(m2_pois_new$coefficients["treat"]) - 1) * 100))
```

```
## [1] "Percent Change:  28.304195804195"
```

```
check_model(m2_pois_new)
```




```
check_overdispersion(m2_pois_new)
```

```
## # Overdispersion test
##
##        dispersion ratio =    57.103
##   Pearson's Chi-Squared = 26496.023
##                 p-value =   < 0.001
```

```
check_zeroinflation(m2_pois_new)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 51
##   Predicted zeros: 0
##             Ratio: 0.00
```

While there are less violations of our model assumptions in the Poisson model and our p-value states that our results are statistically significant, there is still zero-inflation and over dispersion present in our data. Additionally, the uniformity of residuals still does not follow along the expected line. This suggests that the model is not a good fit for our data.

Negative Binomial Model

```
# Negative binomial model
m3_nb_new <- glm.nb(counts ~ treat,
                data = spiny_counts_new)

summ(m3_nb_new, model.fit = FALSE)
```

Observations	466
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.5769)
Link	log

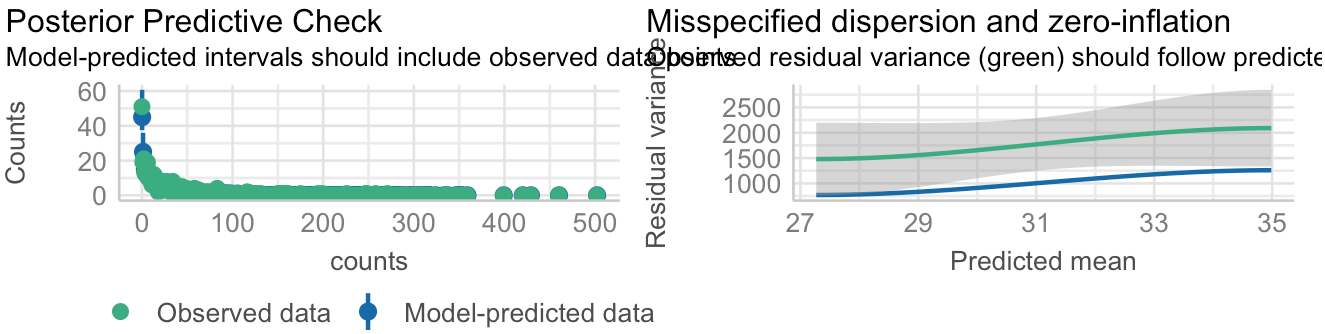
	Est.	S.E.	z val.	p
(Intercept)	3.31	0.08	38.97	0.00
treat	0.25	0.12	2.02	0.04

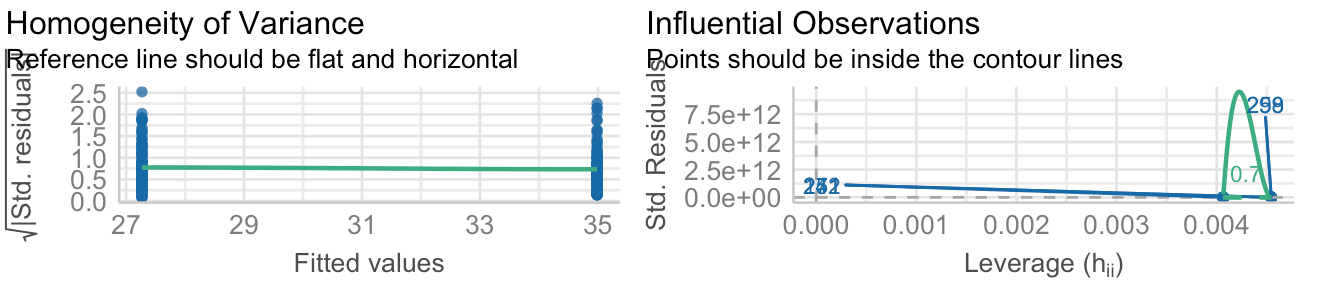
Standard errors: MLE

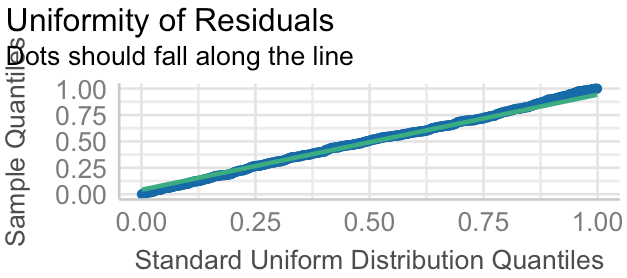
```
print(paste("Percent Change: ", (exp(m3_nb_new$coefficients["treat"]) - 1) * 100))
```

```
## [1] "Percent Change:  28.3041958041958"
```

```
check_model(m3_nb_new)
```








```
check_overdispersion(m3_nb_new)
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.035
##           p-value = 0.808
```

```
check_zeroinflation(m3_nb_new)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 51
##   Predicted zeros: 47
##             Ratio: 0.91
```

Again, the negative binomial model performed the best out of all our model selections. This tracks with what we discovered when model-fitting for the 2018 lobster data. However this time, the P-value in our treatment variable is significant. The residuals are uniform, we have very little over dispersion (with a p value of .8) and little zero-inflation, and our data follows our posterior distribution. This suggests that the negative binomial model is the best fit for our data, and is not statistically insiginificant as suggested before

c. Compare and contrast results with the analysis from the 2012-2018 data sample (~ 2 paragraphs)

```
# View all three model results side by side
export_summs(m1_ols_new, m2_pois_new, m3_nb_new,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none")
```

	OLS	Poisson	NB
(Intercept)	27.27 ***	3.31 ***	3.31 ***
	(2.69)	(0.01)	(0.08)
treat	7.72 *	0.25 ***	0.25 *
	(3.91)	(0.02)	(0.12)
Observations	466		

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

```
# Percent Change in each model
paste("OLS: ", (m1_ols_new$coefficients[2]/m1_ols_new$coefficients[1])*100, "% Change")
```

```
## [1] "OLS:  28.3041958041957 % Change"
```

```
paste("Poisson: ", (exp(m2_pois_new$coefficients[2]) - 1) * 100, "% Change")
```

```
## [1] "Poisson:  28.304195804195 % Change"
```

```
paste("Negative Binomial: ", (exp(m3_nb_new$coefficients[2])-1)*100, "% Change")
```

```
## [1] "Negative Binomial:  28.3041958041958 % Change"
```

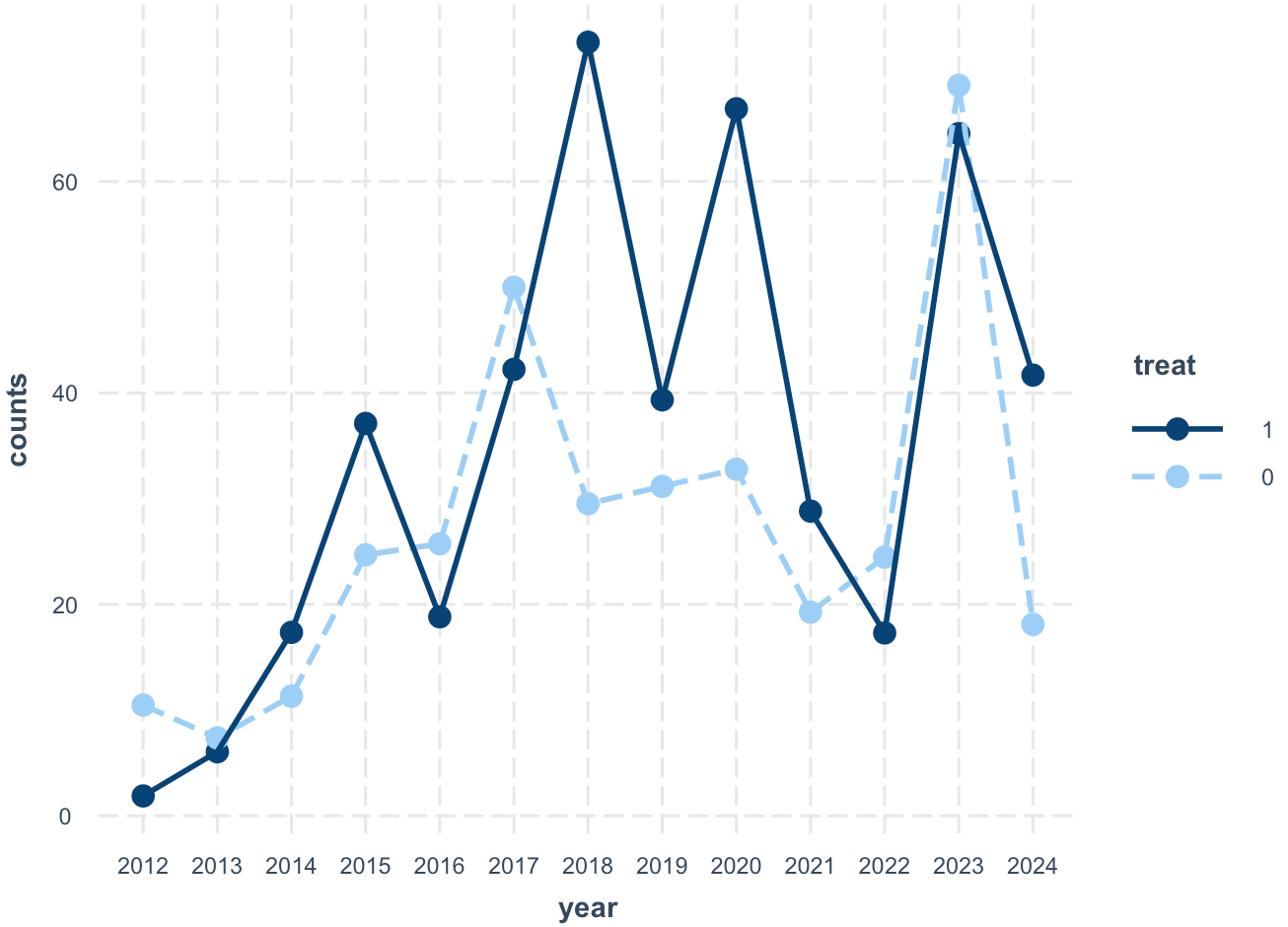
```
# Create df with factorized year
ff_counts_new <- spiny_counts_new %>%
    mutate(year=as_factor(year))

# Create a negative binomial with factorized years
m5_fixedeffs_new <- glm.nb(
    counts ~
        treat +
        year +
        treat*year,
    data = ff_counts_new)

summ(m5_fixedeffs_new, model.fit = FALSE)
```

Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.7762)
Link	log

	Est.	S.E.	z val.	p
(Intercept)	2.35	0.27	8.70	0.00
treat	-1.72	0.42	-4.05	0.00
year2013	-0.35	0.38	-0.91	0.36
year2014	0.08	0.38	0.20	0.84
year2015	0.86	0.38	2.27	0.02
year2016	0.90	0.38	2.38	0.02
year2017	1.56	0.38	4.15	0.00
year2018	1.04	0.38	2.75	0.01
year2019	1.09	0.38	2.89	0.00
year2020	1.14	0.38	3.03	0.00
year2021	0.61	0.38	1.61	0.11
year2022	0.85	0.38	2.25	0.02
year2023	1.89	0.38	5.02	0.00
year2024	0.55	0.38	1.43	0.15
treat:year2013	1.52	0.58	2.61	0.01
treat:year2014	2.14	0.58	3.72	0.00
treat:year2015	2.12	0.57	3.71	0.00
treat:year2016	1.40	0.57	2.45	0.01
treat:year2017	1.55	0.57	2.71	0.01
treat:year2018	2.62	0.57	4.60	0.00
treat:year2019	1.95	0.57	3.41	0.00
treat:year2020	2.43	0.57	4.25	0.00
treat:year2021	2.12	0.57	3.70	0.00
treat:year2022	1.37	0.57	2.39	0.02
treat:year2023	1.65	0.57	2.89	0.00
treat:year2024	2.55	0.58	4.40	0.00

Standard errors: MLE

```
interact_plot(m5_fixedeffs_new, pred = year, modx = treat,
              outcome.scale = "response")
```



With new data, we see increased significance across all of our models. This suggests that the new data modeling is similarly robust when compared the old data. Additionally, the negative binomial model is still the best fit for our data, as it has the lowest dispersion ratio and the most uniform residuals. This suggests that the negative binomial model is the best fit for our data, and is not statistically insignificant as suggested before. Additionally, the graph intuitively indicates that there is a trend of higher counts of lobster in the MPA sites than the non-MPA sites, which is consistent with our model results.