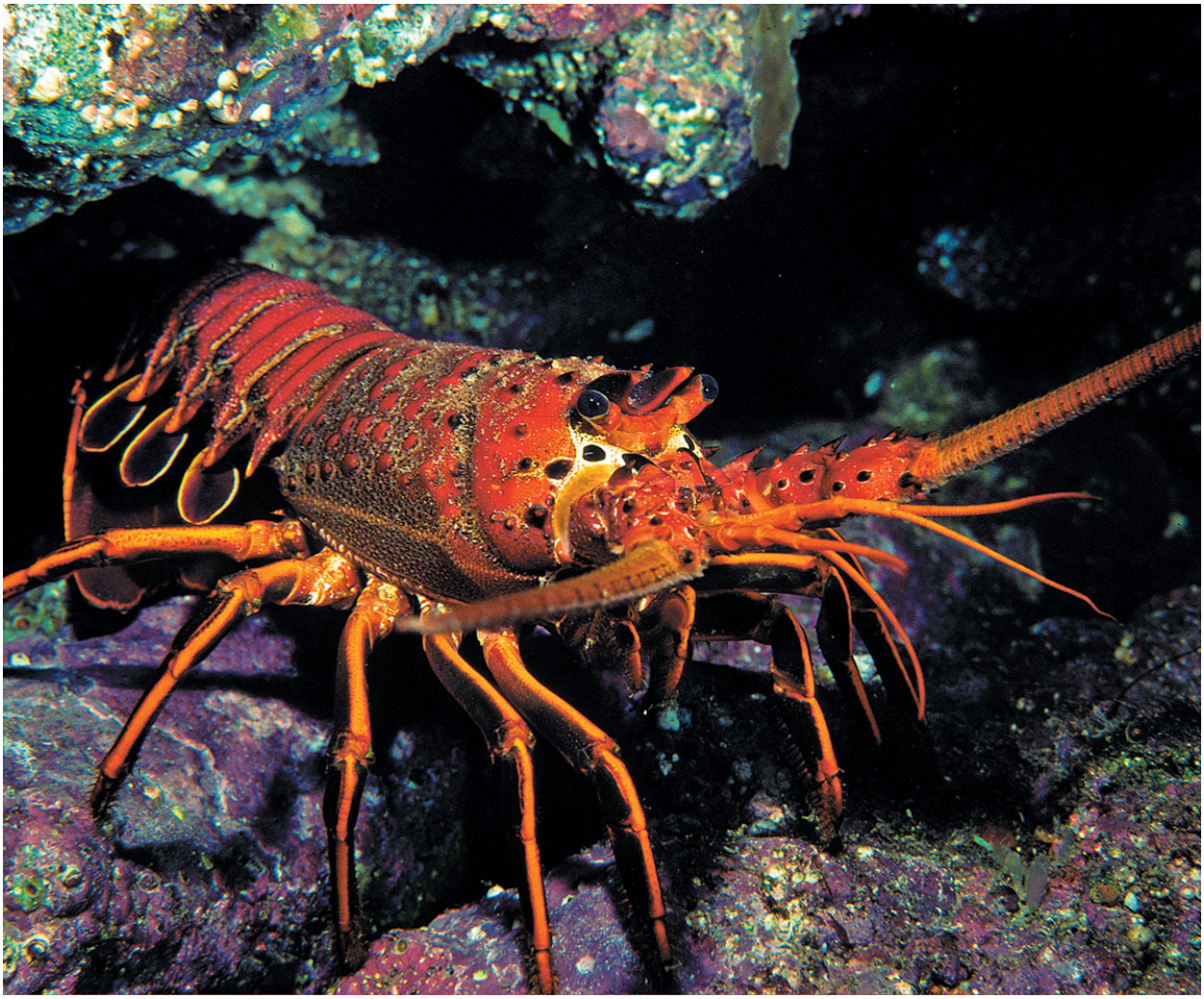
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

Emma Bea Mitchell

1/8/2024 (Due 1/26)



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 Emma Bea Mitchell: _____

```
# Load libraries
library(tidyverse)
library(here)
library(janitor)
library(estimatr)
library(performance)
library(jtools)
library(gt)
library(gtsummary)
library(MASS) ## NOTE: The `select()` function is masked. Use: `dplyr::select()` ##
library(interactions)
library(ggridges)
library(ggbeeswarm)
library(gtsummary)
```

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. Dataset accessed 11/17/2019.

Introduction

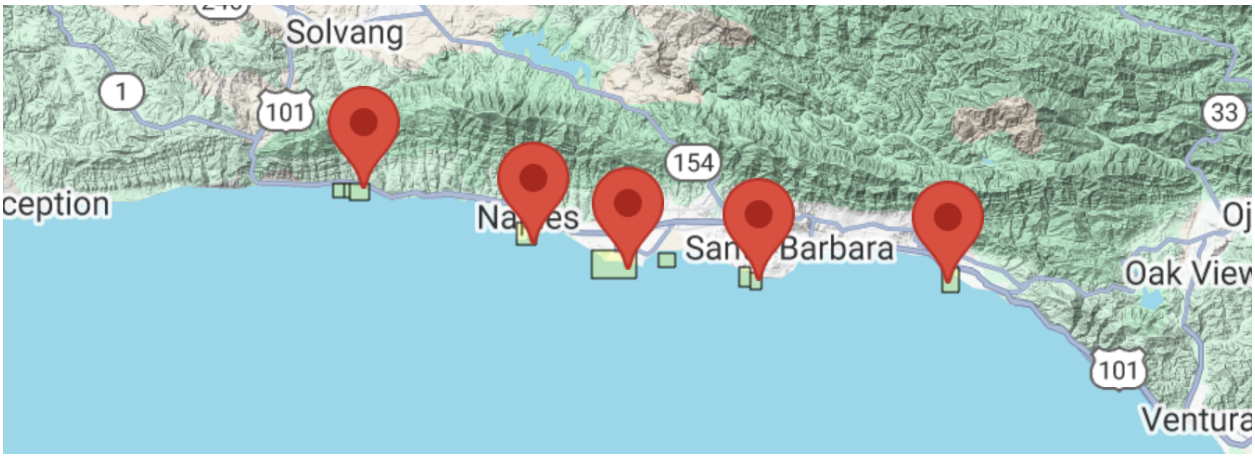
You're about to dive into some deep data collected from five reef sites in Santa Barbara County, all about the abundance of California spiny lobsters! Data was gathered by divers annually from 2012 to 2018 across Naples, Mohawk, Isla Vista, Carpinteria, and Arroyo Quemado reefs.

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!).

Although we can see on the map above that the control sights (Arroyo Quemado, Carpenteria, and Mohawk) are relatively close along to the coast line to the treatment sites (Naples, Isla Vista), we can't assume that these sights are completely identical. Without further information about how these sights are nearly identical in relevant attributes, I wouldn't assume that there is no selection bias. Each area has a different human population, which may impact lobster populations. They may also have varying ecosystems and habitats such as sandy, rocky, more shallow areas, more predators, etc. Although selection bias is most likely present, it appears to be a similar enough comparison to make accurate assumptions about the effects of MPAs on lobster populations.

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"`)
# Load data and clean
rawdata <- read_csv(here("data", "spiny_abundance_sb_18.csv")) |>
    clean_names() |>
    naniar::replace_with_na(replace = list(size_mm = -99999))
```

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 new column
 tidydata <- rawdata |>
    mutate(reef = factor(site, order = TRUE, levels = c("AQUE", "CARP", "MOHK", "IVEE", "NAPL"), labels
```

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
# Create new variables

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") ~ "non_MPA"
    ), treat = case_when(mpa == "MPA" ~ 1,
                         mpa == "non_MPA" ~ 0))

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

> 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**:

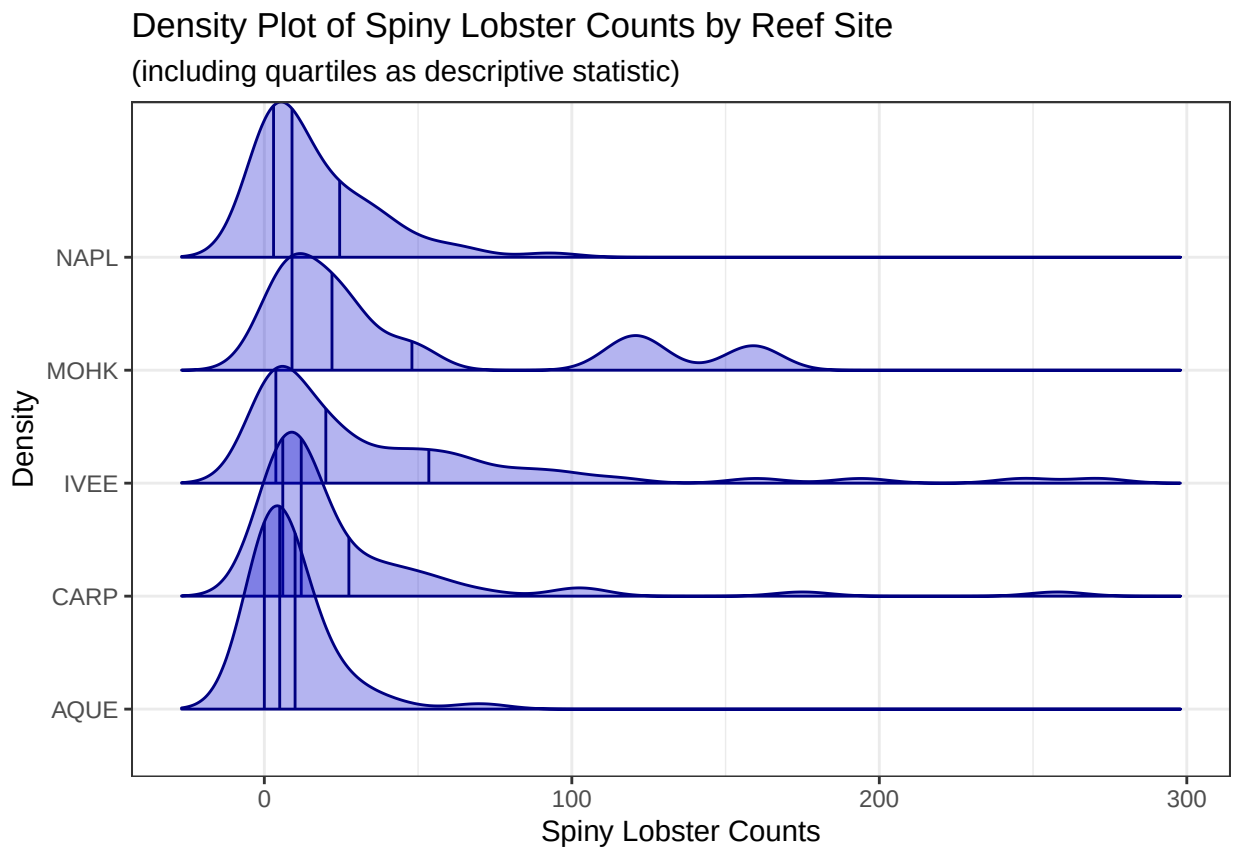
1) grouped by reef site

2) grouped by MPA status
3) grouped by year

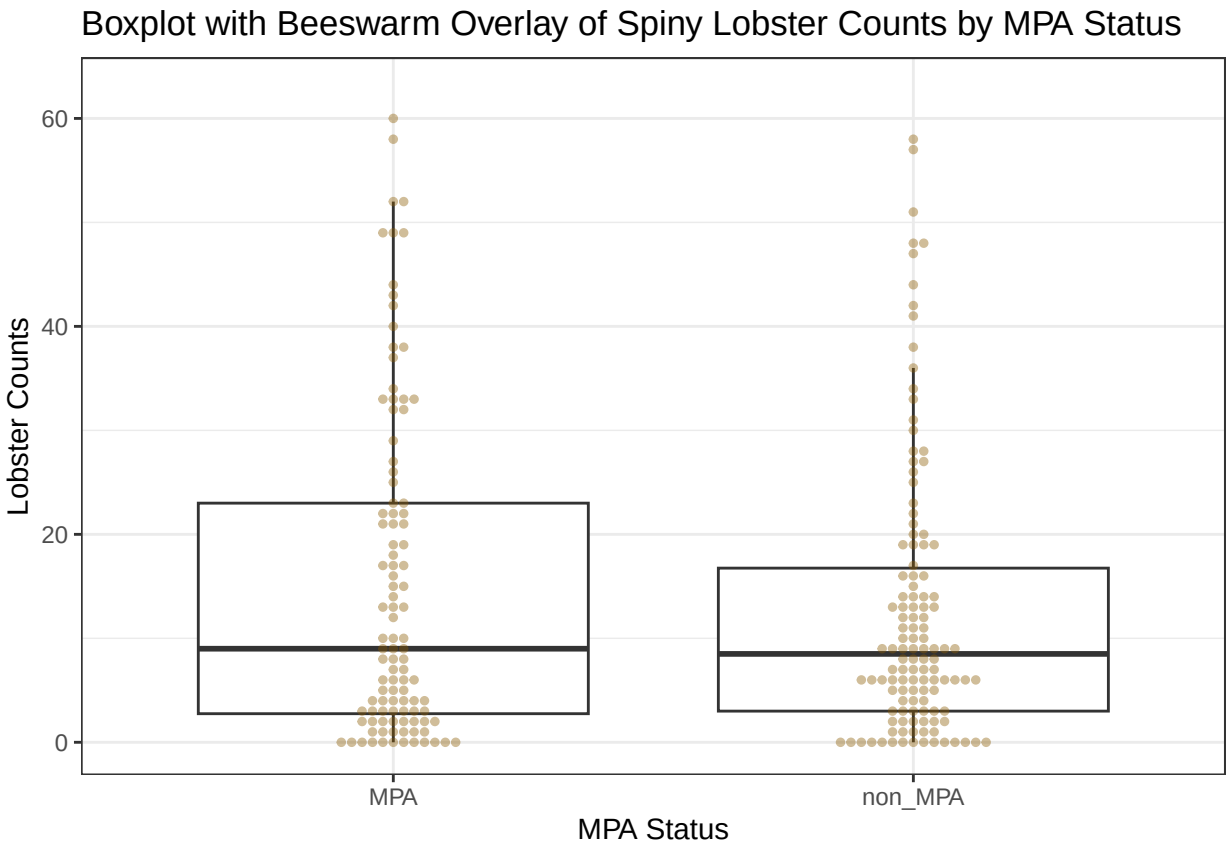
Create a plot of lobster **size** :

4) You choose the grouping variable(s)!

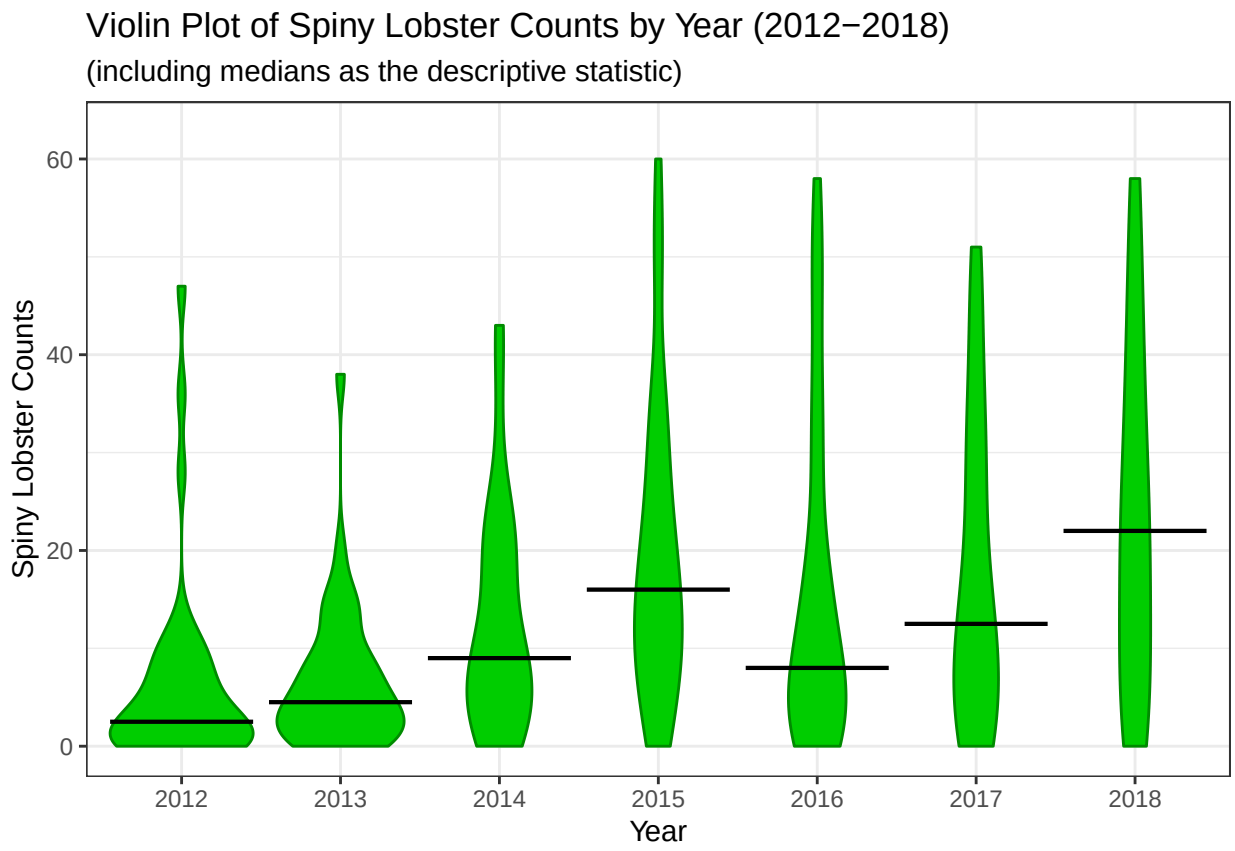
```
# plot 1: density ridge plot
plot1 <- spiny_counts |>
    ggplot(aes(x = counts, y = site)) +
    geom_density_ridges2(quantile_lines = TRUE,
                         alpha = 0.3,
                         fill = "blue3",
                         color = "navy") +
    labs(
        title = "Density Plot of Spiny Lobster Counts by Reef Site",
        subtitle = "(including quartiles as descriptive statistic)",
        x = "Spiny Lobster Counts",
        y = "Density") +
    theme_bw()

print(plot1)
```



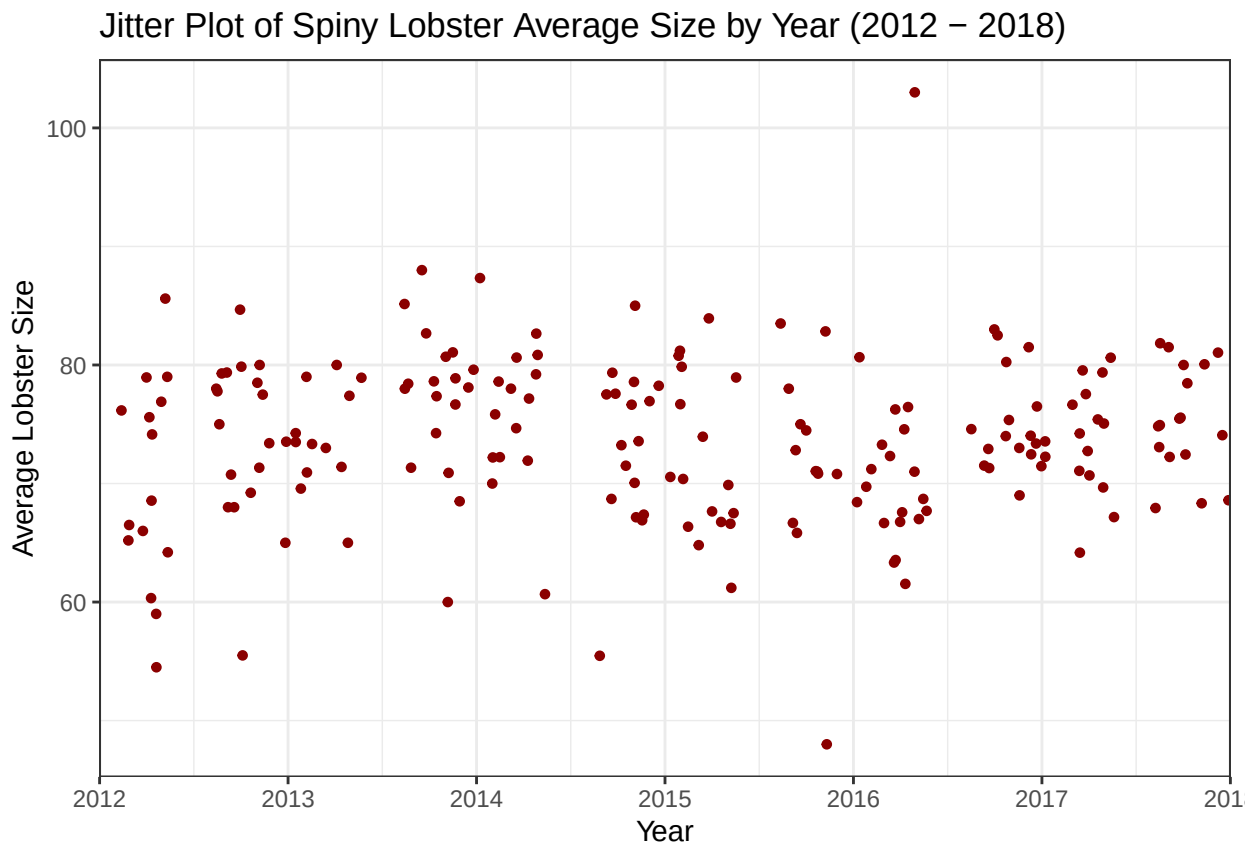

```
# plot 2: beeswarm (with boxplot)

plot2 <- ggplot(spiny_counts, aes(x = mpa, y = counts)) +
    geom_boxplot(outlier.shape = NA) +
    ggbeeswarm::geom_beeswarm(size = 1, alpha = .4, color = "orange4") +
    scale_y_continuous(limits = quantile(spiny_counts$counts, c(0.1, 0.9))) +
    theme_bw() +
    labs(
        title = "Boxplot with Beeswarm Overlay of Spiny Lobster Counts by MPA Status",
        x = "MPA Status",
        y = "Lobster Counts")
print(plot2)
```




```
# plot 3: violin plot

plot3 <- ggplot(spiny_counts, aes(x = as.factor(year), y = counts)) +
    geom_violin(color = "green4", fill = "green3") +
    stat_summary(fun.y=median, geom="crossbar", size=.3, color="black") +
    scale_y_continuous(limits = quantile(spiny_counts$counts, c(0.1, 0.9))) +
    theme_bw() +
    labs(
        title = "Violin Plot of Spiny Lobster Counts by Year (2012-2018)",
        subtitle = "(including medians as the descriptive statistic)",
        x = "Year",
        y = "Spiny Lobster Counts")

print(plot3)
```




```
# plot 4: jitter plot

plot4 <- ggplot(spiny_counts, aes(x = year, y = mean_size)) +
    geom_jitter(color = "darkred", size = 1.2) +
    theme_bw() +
    labs(
        title = "Jitter Plot of Spiny Lobster Average Size by Year (2012 - 2018)",
        x = "Year",
        y = "Average Lobster Size") +
     scale_x_continuous(limits=c(2012, 2018), expand = c(0,NA))


print(plot4)
```



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 |>
    dplyr::select(counts, mean_size, mpa) |>
    tbl_summary(by = mpa,
                statistic = list(all_continuous() ~ "{mean}")) |>
    modify_caption("**Comparing the mean counts and mean sizes of California Spiny Lobsters at MPA and
```

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.

```
# NOTE: We will not evaluate/interpret model fit in this assignment (e.g., R-square)
# Fit OLS model
m1_ols <- lm(counts ~ treat, data = spiny_counts)

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

Table 1: **Comparing the mean counts and mean sizes of California Spiny Lobsters at MPA and non-MPA sites**

Characteristic	**MPA** N = 119[1]	**non_MPA** N = 133[1]
site		
AQUE	0 (0%)	49 (37%)
CARP	0 (0%)	63 (47%)
IVEE	56 (47%)	0 (0%)
MOHK	0 (0%)	21 (16%)
NAPL	63 (53%)	0 (0%)
year		
2012	17 (14%)	19 (14%)
2013	17 (14%)	19 (14%)
2014	17 (14%)	19 (14%)
2015	17 (14%)	19 (14%)
2016	17 (14%)	19 (14%)
2017	17 (14%)	19 (14%)
2018	17 (14%)	19 (14%)
counts	28	23
mean_size	76	73
Unknown	12	15

[1] n (%); Mean

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

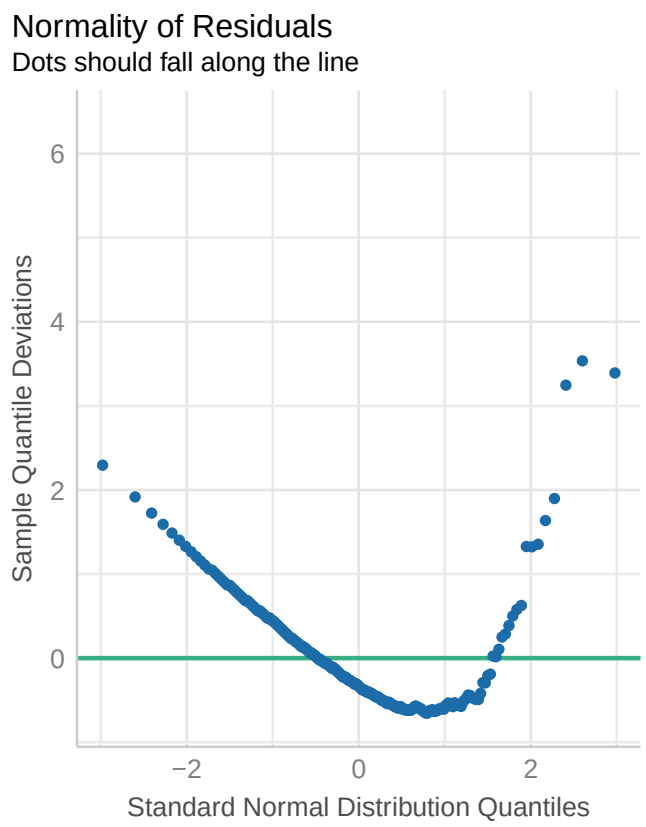
Standard errors: OLS

The intercept coefficient is the lobster count when the site is not an MPA (22.73). The treatment plus the intercept is the count when the site is an MPA (28.09). The p-value of the intercept is 0, which is means it is significant. The treat coefficient has a p value of .3, meaning that the coefficient is not statistically significant.

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

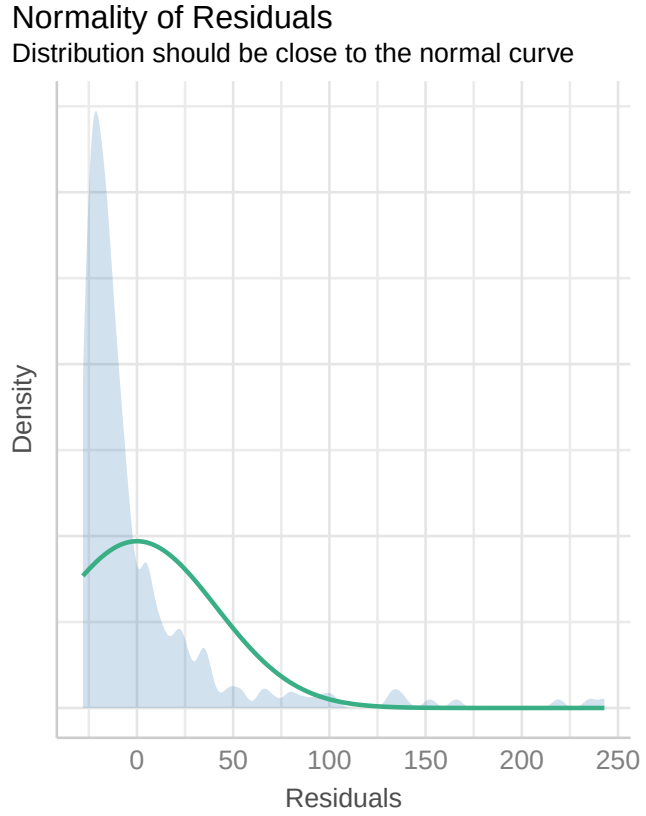
d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

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



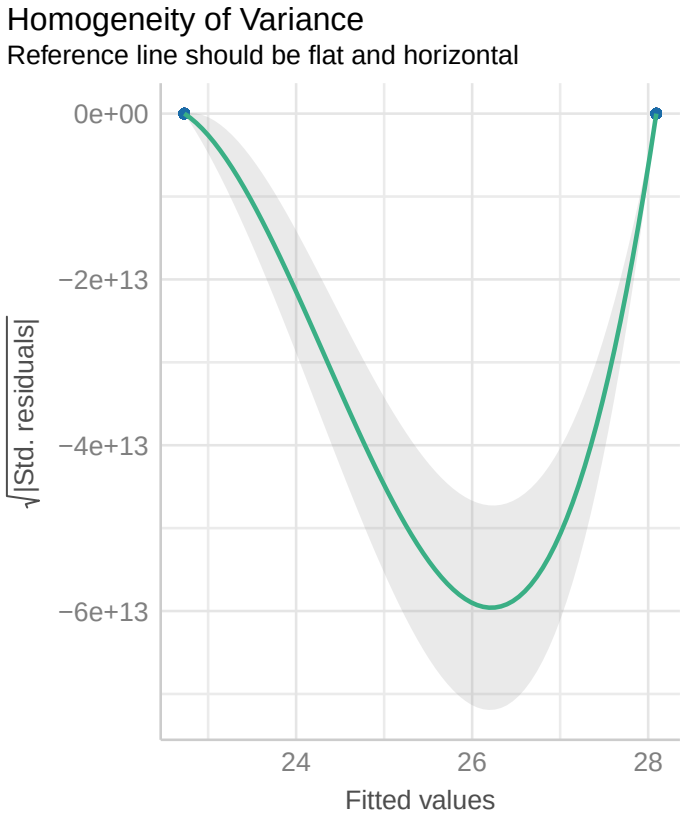
This model check is looking at the normality of residuals. If our residuals were normal, the dots would fall along the green line. Our results mean that are residuals are very much not normal. Because we are using a linear regression model, it's important that our residuals are normal. Our results show that our OLS model is not a good fit.

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



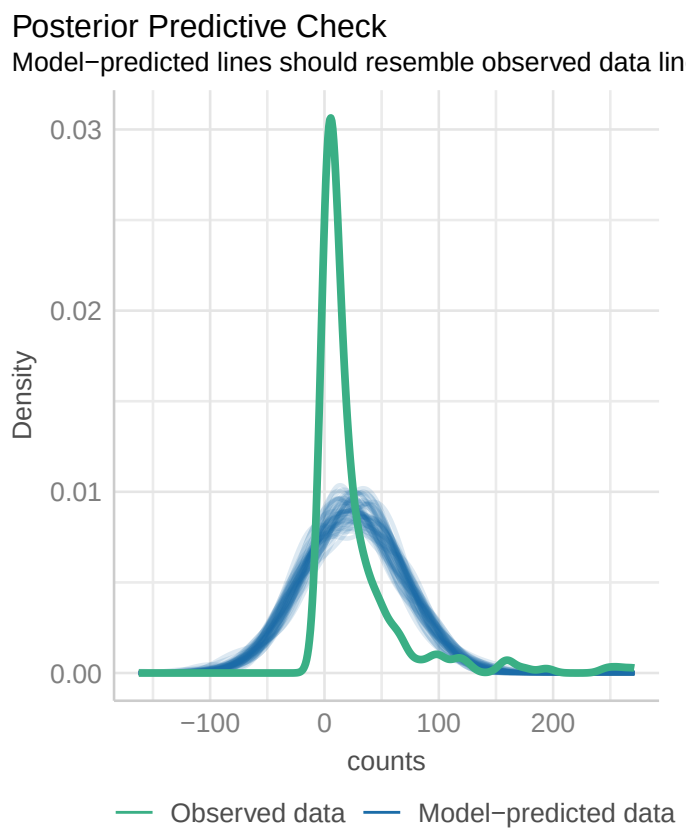
This model is also looking at the normality of residuals, this time looking at the shape of the curve. If our distributions fell close to the green curve on the graph, we would have a normally distributed curve as expected if our model was linear. Because it's far from being close to the curve, it's clear our distribution is not normal and we should not be using a linear regression model.

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



This model is looking at the homogeneity of variance, meaning that we're looking at the equality of the variance across our samples. The subtitle tells us that the reference line should be flat and horizontal, but when we look at the graph we can tell that the line is extremely curved in a U shape. This is a sign that our data could be skewed or biased. This is another sign that a linear regression is not the right choice.

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



This model is looking at the posterior predictive check, meaning that it's looking at how well the model shows means, standard deviations, and quantiles. If the model predicted data lines lined up with the green observed data line, our model would successfully capture relevant aspects. We are seeing that our lines are not lining up because our model is not showing these aspects successfully. This is another sign that our linear regression model is not the right choice.

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 interpreted as

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

# Fit poisson model
m2_pois <- glm(counts ~ treat,
               spiny_counts,
               family = poisson(link = "log"))
```

```
summ(m2_pois)
```

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

$\chi^2(1)$	71.36
p	0.00
Pseudo-R^2 (Cragg-Uhler)	0.25
Pseudo-R^2 (McFadden)	0.01
AIC	11365.62
BIC	11372.68

	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(exp(.21) - 1)
```

```
## [1] 0.2336781
```

b. The predictor coefficient is .21 on a log scale. When using a poisson regression model we take the exponent of that predictor coefficient and then subtract one. This gives us the percent changed when the treatment is present (it is an mpa site). The results are a 23.36 percent increase.

c. The poisson model makes the assumption that the mean is equal to the dispersion (variance). When there is overdispersion that means that the dispersion (variance) is greater than the mean.

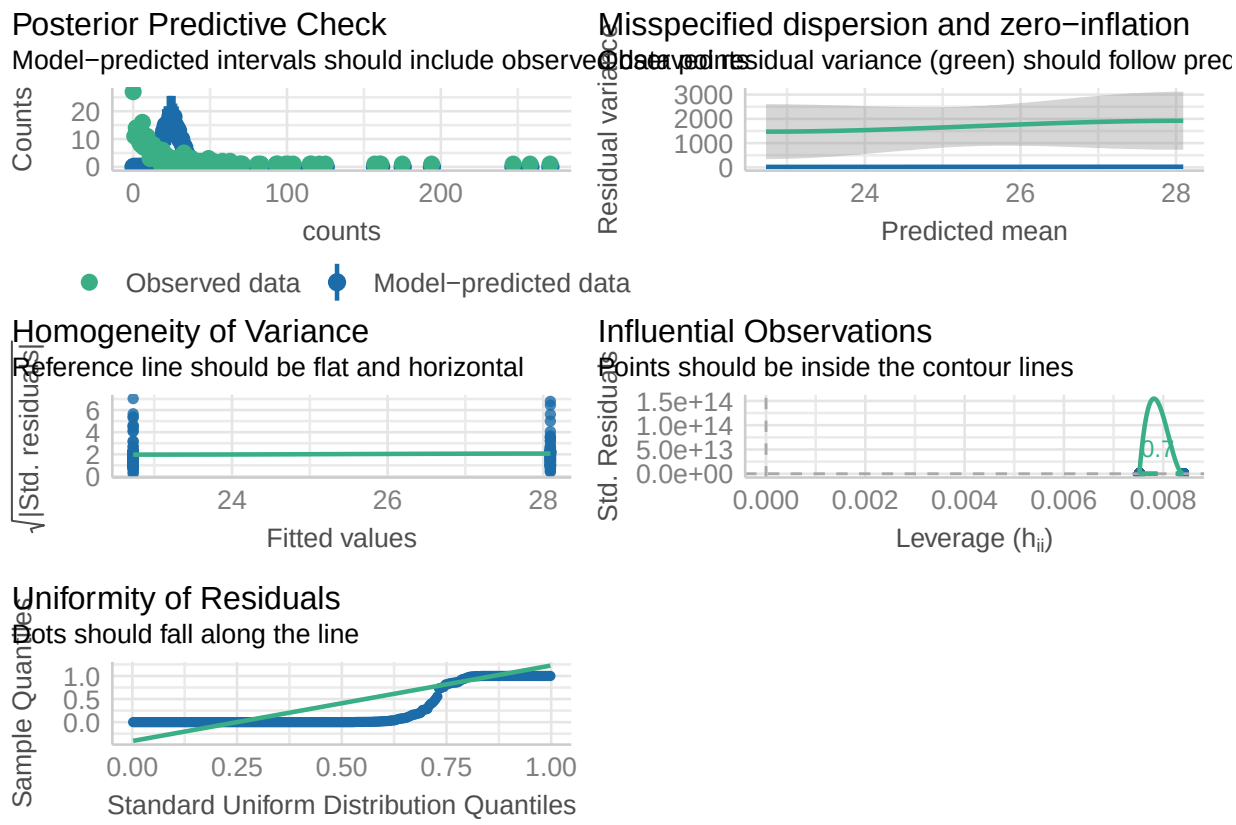
d. In the previous model, the treatment effect was not statistically significant. Its p-value was .3, which is over the .05 threshold for significance. In this model, the p-value of the treatment is 0.00, meaning that it is significant in this model. In the previous model, the treatment effect was 5.36, translating to about a 23.58 percent increase from non-mpa sites. In this model the predictor coefficient tells us that the treament effect is 23.36 percent. Both results are extremely close, although there is a slight increase in signficance of effect in the poisson model.

e. Check the model assumptions. Explain results.

The model assumes that the mean is equal to the distribution. By using the overdispersion test we can tell that the dispersion does not equal the mean, meaning that poisson is not a good fit. We also checked the zero inflation, which showed that there were too many zeros in our model, another sign that poisson is not a good fit.

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

```
print(check_model(m2_pois))
```




```
print(check_overdispersion(m2_pois))
```

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

This test shows that overdispersion was detected in the model. The p-value is less than .001, meaning that the test results are statistically significant. We now know that the dispersion is greater than the mean.

```
print(check_zeroinflation(m2_pois))
```

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

This test shows us that the model is underfitting zeros and that zero-inflation is probable. That means that the model is allowing too many zeros. This is a sign to us that a glm model was not the right choice.

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.

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 NB model
m3_nb <- MASS::glm.nb(counts ~ treat,
                      spiny_counts)

summ(m3_nb)
```

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

$\chi^2(250)$	1.52
p	0.22
Pseudo-R^2 (Cragg-Uhler)	0.01
Pseudo-R^2 (McFadden)	0.00
AIC	2088.53
BIC	2099.12

	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(exp(.21) - 1)
```

```
## [1] 0.2336781
```

```
print(check_overdispersion(m3_nb))
```

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

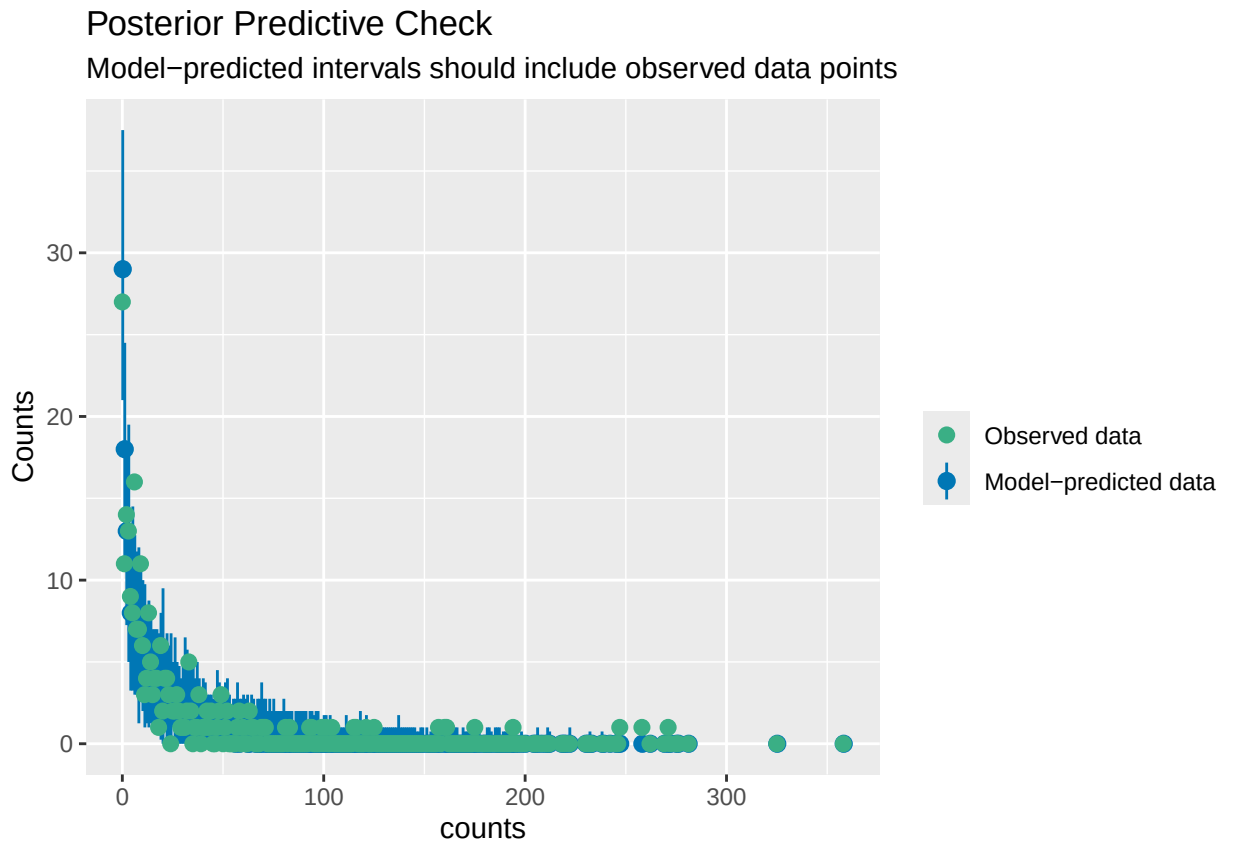
As expected, this model controls for overdispersion, so there is none present.

```
print(check_zeroinflation(m3_nb))
```

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

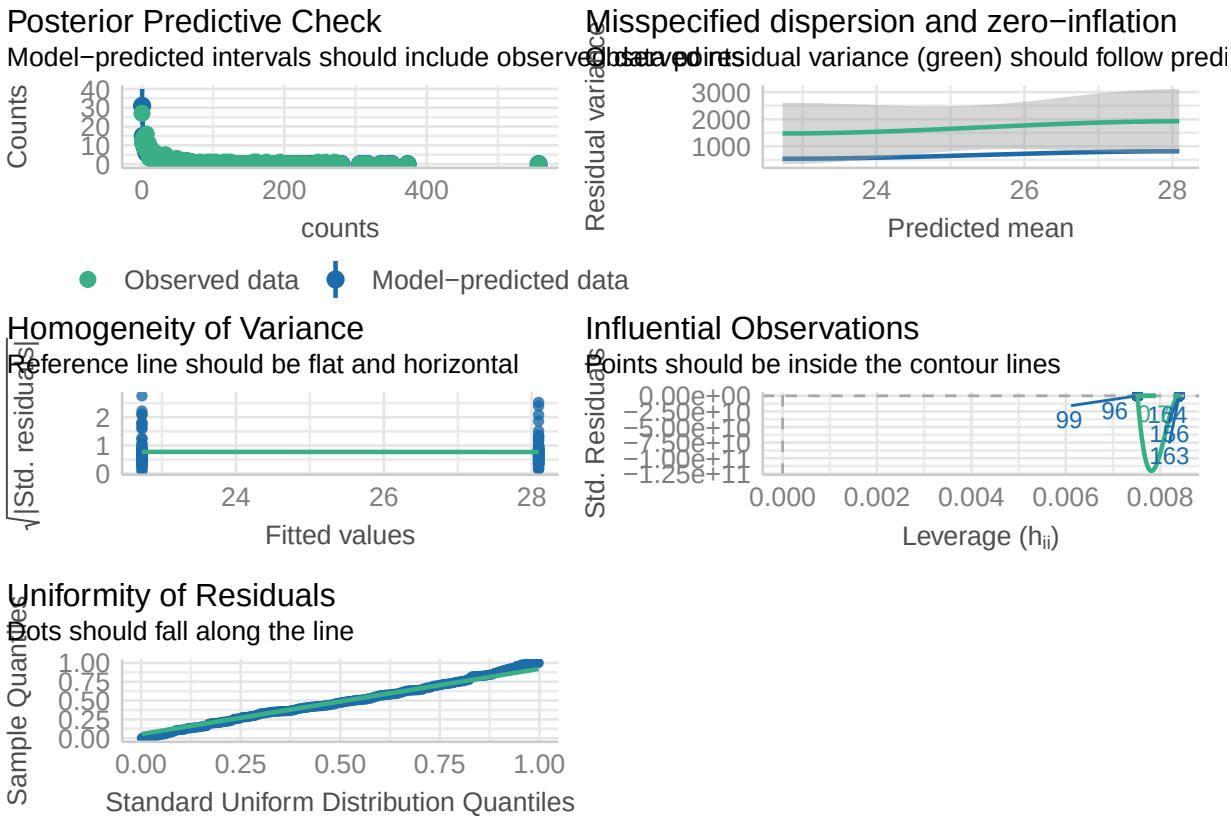
This model does not control/ have a parameter for zero inflation, so it makes sense that overfitting of zeros would still be present.

```
print(check_predictions(m3_nb))
```



This model's observed data and model-predicted data seem to line up in the graph. Meaning that the model represents mean, quantiles, and standard deviation well.

```
print(check_model(m3_nb))
```



h. The rationale for fitting a negative binomial model is that it accounts for over dispersion by adding a parameter. We know from the poisson model checks that there is over dispersion in our model, so we know that over dispersion is something we need to account for.

i. The treatment estimate is .21, and similar to the poisson model this one also shows log-scale coefficients. When performing the equation we can tell that the treatment effect is 23.36 percent. The p-value of the treatment coefficient is .22, meaning it's not statistically significant. The treatment effect is very similar to the first model (23.58), and the same as the poisson model (23.36)

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.

```
# Create comparison table
export_summs(m1_ols, m2_pois, m3_nb,
             model.names = c("OLS","Poisson", "NB"))
```

The results of all three models are actually quite similar. The OLS regression looks different in numbers, but that is only because the poisson and negative binomial are log scale coefficients. In actuality, the treatment effect from all the models are extremely similar. The poisson and negative binomial models have the same summary outputs. This signifies to me that the treatment effect is robust/stable across models.

	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)
N	252	252	252
R2	0.00		
AIC	2593.35	11365.62	2088.53
BIC	2603.94	11372.68	2099.12
Pseudo R2		0.25	0.01

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

Step 7: Building intuition - fixed effects

a. Create new `df` with the `year` variable converted to a factor

b. Run the following OLS model using `lm()`

- Use the following specification for the outcome `log(counts+1)`
- Estimate fixed effects for `year`
- Include an interaction term between variables `treat` and `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)

The model has estimated the treatment effect, but has also included the years. There is a coefficient for each year for both treatment and non treatment sites. The intercept coefficient represents the non-treatment sites in 2012 (that's the year that is left out of the coefficients), and the treat coefficient represents the treatment sites in 2012.

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

This result does make sense because the main treatment effect does not actually represent the main effect without any years, it actually represents the treatment effect in 2012. The negative treatment coefficient makes sense because it just means a decrease in counts in the treatment group in 2012.

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

m5_fixedeffs <- lm(
    log(counts+1) ~ treat*year,
    data = ff_counts)

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

Observations	252
Dependent variable	log(counts + 1)
Type	OLS linear regression

	Est.	S.E.	t val.	p
(Intercept)	1.95	0.27	7.26	0.00
treat	-1.23	0.39	-3.16	0.00
year2013	-0.27	0.38	-0.71	0.48
year2014	0.02	0.38	0.04	0.97
year2015	0.49	0.38	1.30	0.20
year2016	0.61	0.38	1.61	0.11
year2017	1.04	0.38	2.73	0.01
year2018	0.83	0.38	2.18	0.03
treat:year2013	1.16	0.55	2.10	0.04
treat:year2014	1.85	0.55	3.35	0.00
treat:year2015	2.25	0.55	4.08	0.00
treat:year2016	0.95	0.55	1.71	0.09
treat:year2017	1.22	0.55	2.20	0.03
treat:year2018	2.27	0.55	4.12	0.00

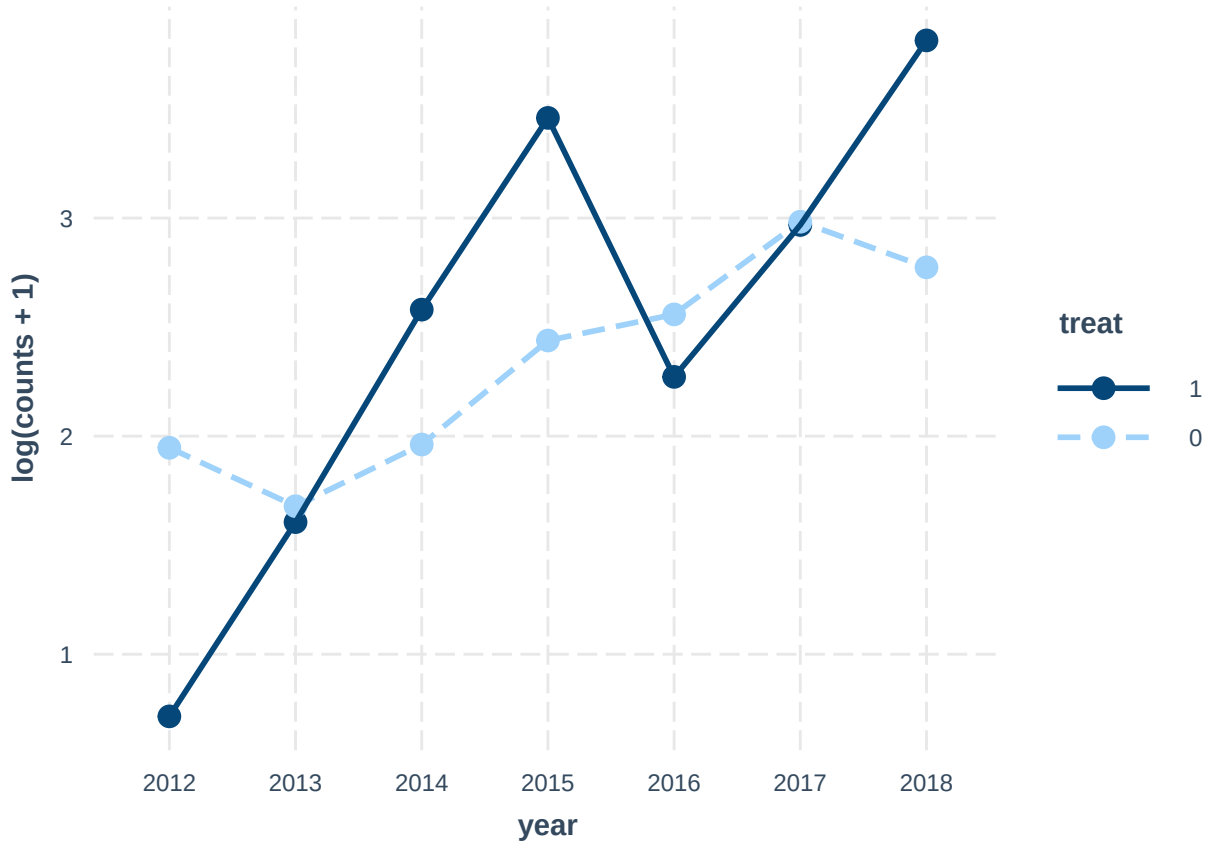
Standard errors: OLS

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.

When re-evaluating my responses based on the plot, I can see that the year 2012 is significantly lower than in change than the rest of the years. This lines up with the negative treat coefficient. In 2012 and 2013, the treatment group was actually lower than the non-treatment group. After that the treatment group is higher than the non-treatment group the majority of the time.

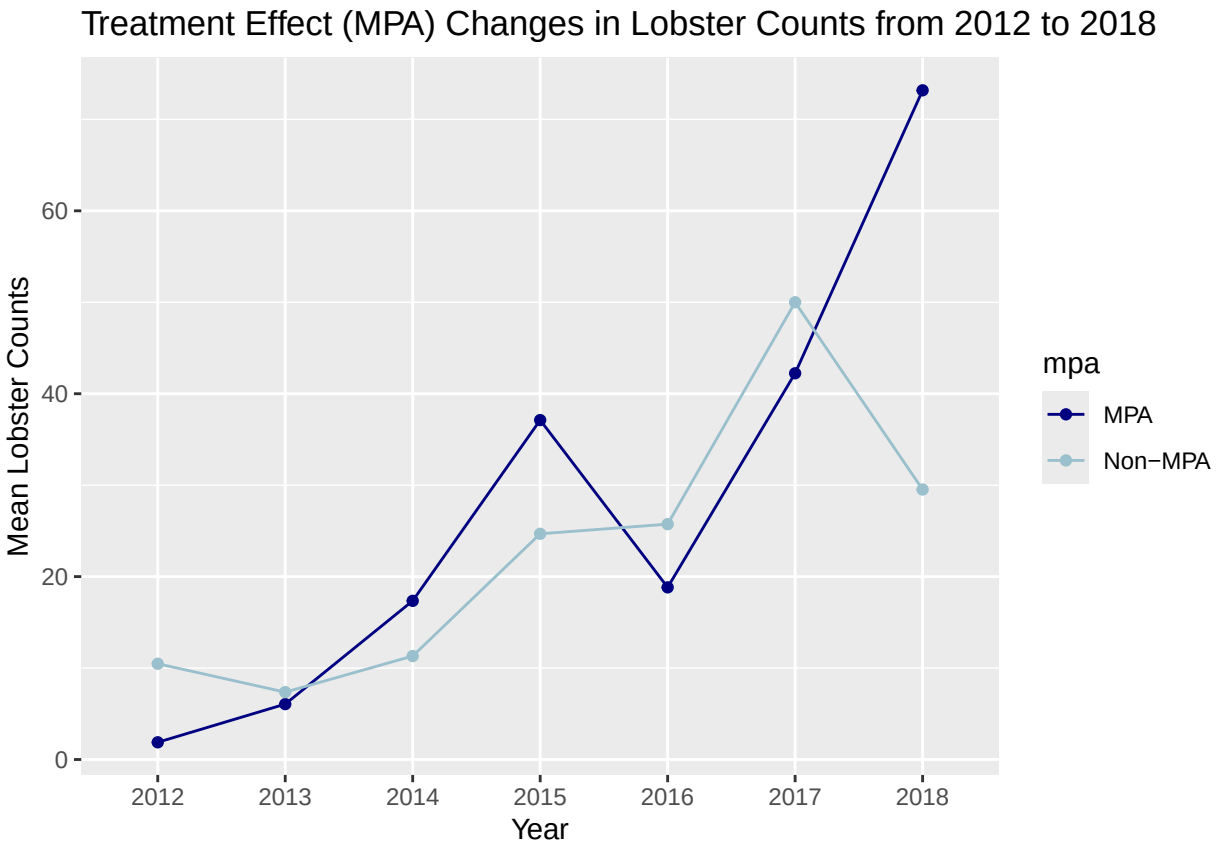
```
# Hint 1: Group counts by `year` and `mpa` and calculate the `mean_count`
# Hint 2: Convert variable `year` to a factor
# Plot interact plot
print(interact_plot(m5_fixedeffs, pred = year, modx = treat,
              outcome.scale = "response"))
```



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 remake of interaction plot using ggplot
plot_counts <- spiny_counts |>
    group_by(year, mpa) |>
    summarize(mean_counts = mean(counts)) |>
    mutate(year = as.factor(year)) |>
    ggplot(aes(x = year, y = mean_counts, color = mpa)) +
    geom_point() +
    geom_line(aes(group = mpa)) +
    labs(
        title = "Treatment Effect (MPA) Changes in Lobster Counts from 2012 to 2018",
        x = "Year",
        y = "Mean Lobster Counts"
    ) +
    scale_color_manual(values = c("navy", "lightblue3" ),
                       labels = c("MPA", "Non-MPA")) +
    scale_linetype_manual(values = c("solid", "longdash"))

print(plot_counts)
```



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 that there is likely at least some spillover in this research context. The sites are all fairly close together (both MPA and non-MPA), meaning that an increase in lobsters at an MPA site may have an effect on the non-MPA sites, it could be as simple as lobsters moving from one site to another.

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

Spillover is an issue for identifying causal effects because it would mean that the treatment and control groups are not independent. The treatment is also somehow affecting the control.

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

Spillover relates to impact because we can't trust that our counts at our sites are independent of each other. The rise in counts in non-MPA sites could be due to the MPA sites being nearby. We can't be sure of the true impact of MPA sites.

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

SUTVA is not a reasonable assumption because we can assume there is a spillover effect. The Stable Unit Treatment Value assumes that the control and treatment groups are unaffected by each other. Because there is spillover, it violates the key assumption in the SUTVA that there is no interference of effect on MPAs on non-MPAs.

The Excludability assumption is that the treatment influences the lobster counts at MPA sites and that the treatment (the MPA regulations) are the only things influencing the treatment sites. This is not a reasonable assumption in our research because if there is a spillover effect we can't be sure that the treatment is actually influencing lobster count differences between sites.

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

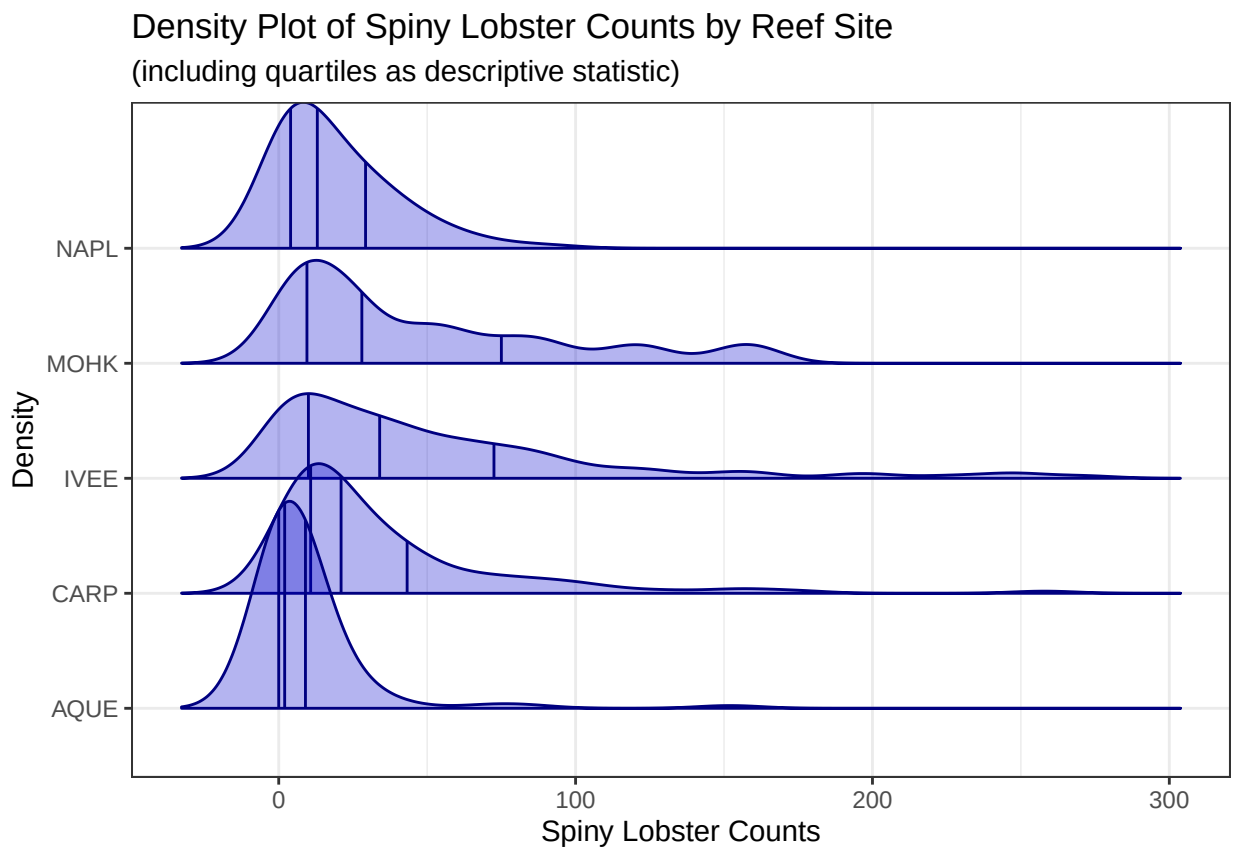
```
recent_lobster <- read_csv(here("data", "lobster_sbchannel_24.csv")) |>
    clean_names() |>
    naniar::replace_with_na(replace = list(size_mm = -99999))
```

```
recent_lobster_clean <- recent_lobster |>
    mutate(reef = factor(site, order = TRUE, levels = c("AQUE", "CARP", "MOHK", "IVEE", "NAPL"), labels
```

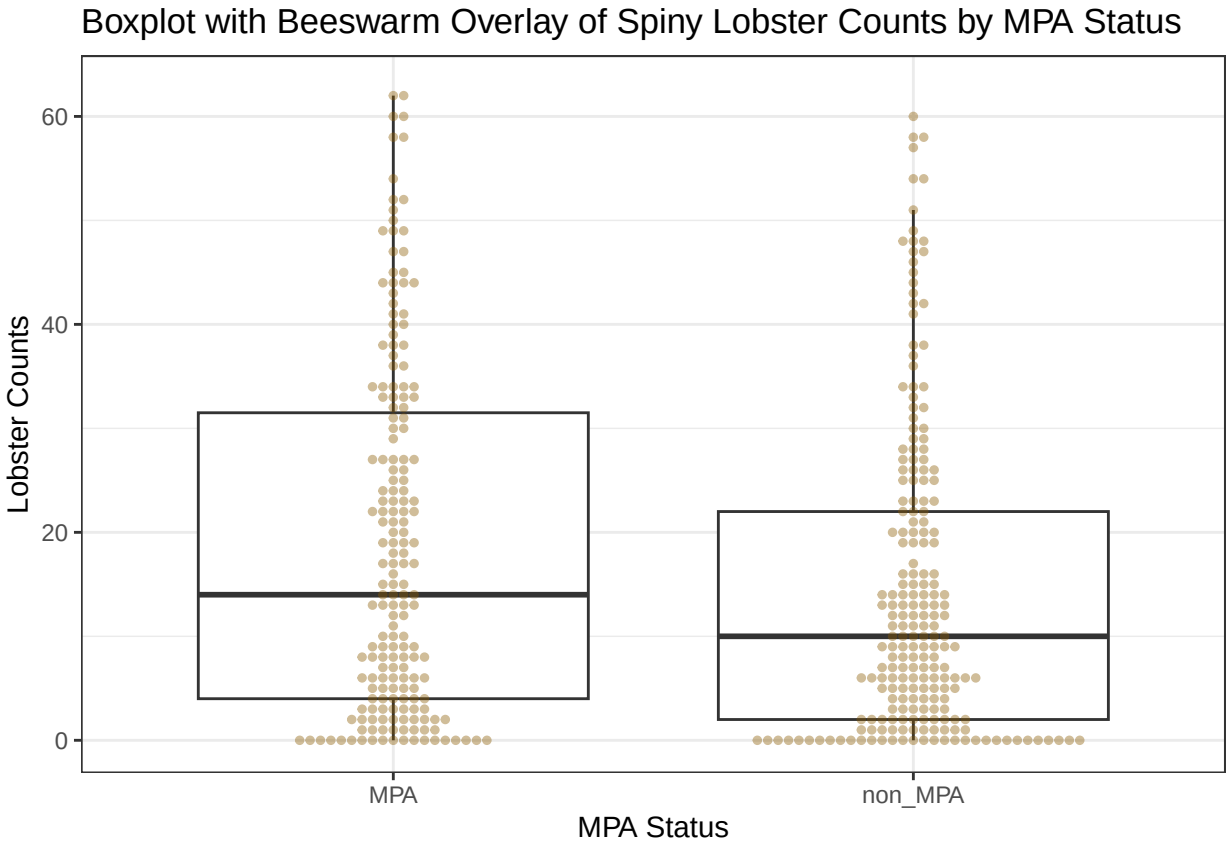
```
recent_counts <- recent_lobster_clean |>
    group_by(site, year, transect) |>
    summarize(counts = sum(count, na.rm = TRUE),
              mean_size = mean(size_mm, na.rm = TRUE)) |>
    mutate(mpa = case_when(
        site %in% c("IVEE", "NAPL") ~ "MPA",
        site %in% c("AQUE", "CARP", "MOHK") ~ "non_MPA"
    ), treat = case_when(mpa == "MPA" ~ 1,
                         mpa == "non_MPA" ~ 0))
```

```
# plot 1: density ridge plot
plot1 <- recent_counts |>
    ggplot(aes(x = counts, y = site)) +
    geom_density_ridges2(quantile_lines = TRUE,
                         alpha = 0.3,
                         fill = "blue3",
                         color = "navy") +
    labs(
        title = "Density Plot of Spiny Lobster Counts by Reef Site",
        subtitle = "(including quartiles as descriptive statistic)",
        x = "Spiny Lobster Counts",
```

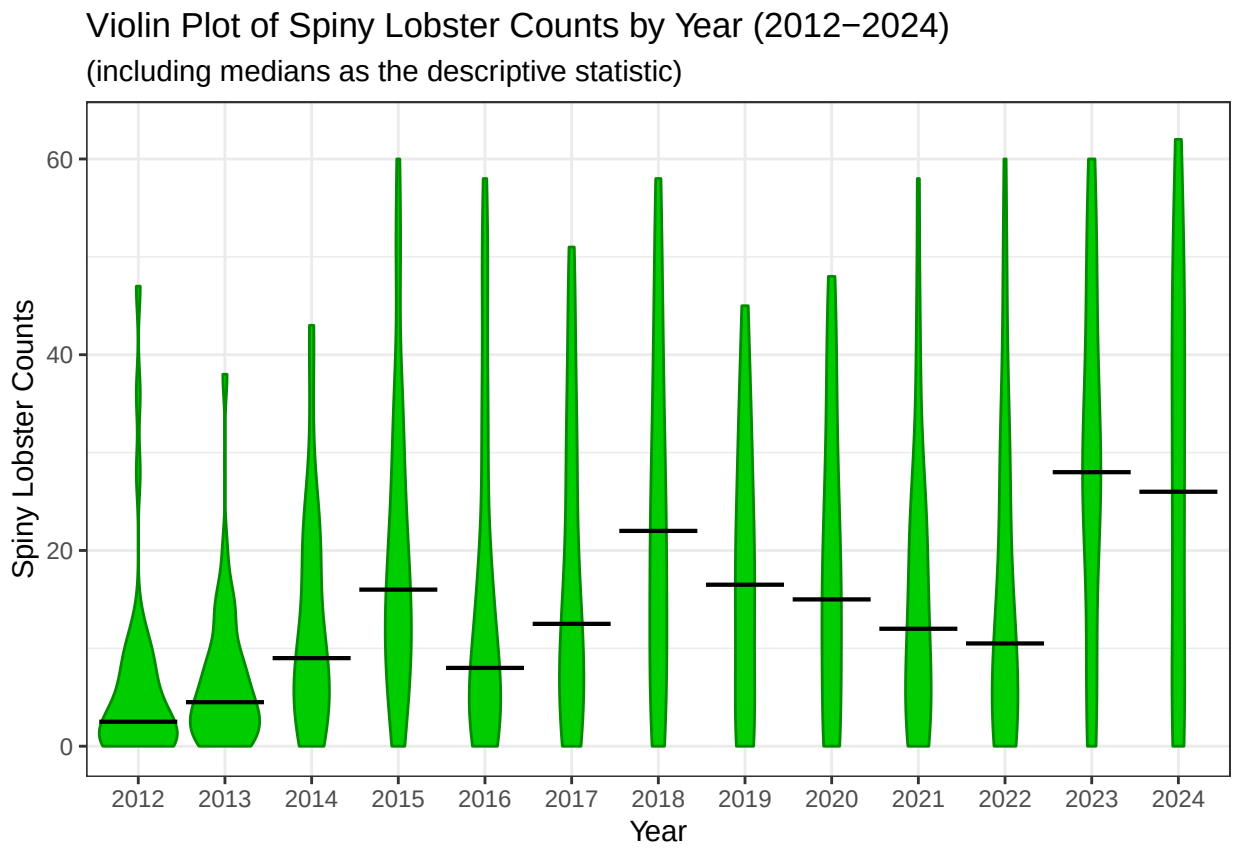
```
        y = "Density") +
    theme_bw()

print(plot1)
```



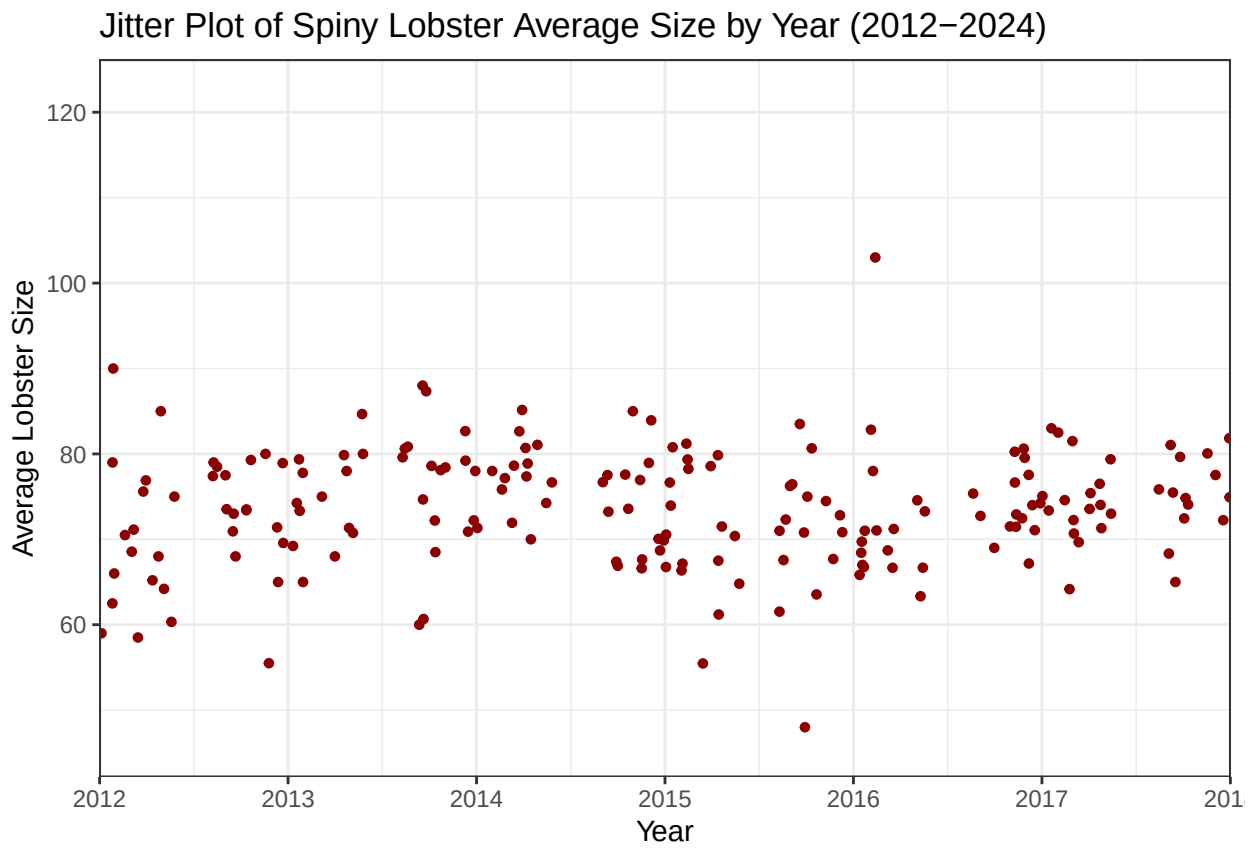

```
# plot 2: beeswarm (with boxplot)

plot2 <- ggplot(recent_counts, aes(x = mpa, y = counts)) +
    geom_boxplot(outlier.shape = NA) +
    ggbeeswarm::geom_beeswarm(size = 1, alpha = .4, color = "orange4") +
    scale_y_continuous(limits = quantile(spiny_counts$counts, c(0.1, 0.9))) +
    theme_bw() +
    labs(
        title = "Boxplot with Beeswarm Overlay of Spiny Lobster Counts by MPA Status",
        x = "MPA Status",
        y = "Lobster Counts")
print(plot2)
```




```
# plot 3: violin plot

plot3 <- ggplot(recent_counts, aes(x = as.factor(year), y = counts)) +
    geom_violin(color = "green4", fill = "green3") +
    stat_summary(fun.y=median, geom="crossbar", size=.3, color="black") +
    scale_y_continuous(limits = quantile(spiny_counts$counts, c(0.1, 0.9))) +
    theme_bw() +
    labs(
        title = "Violin Plot of Spiny Lobster Counts by Year (2012-2024)",
        subtitle = "(including medians as the descriptive statistic)",
        x = "Year",
        y = "Spiny Lobster Counts")

print(plot3)
```




```
# plot 4: jitter plot

plot4 <- ggplot(recent_counts, aes(x = year, y = mean_size)) +
    geom_jitter(color = "darkred", size = 1.2) +
    theme_bw() +
    labs(
        title = "Jitter Plot of Spiny Lobster Average Size by Year (2012-2024)",
        x = "Year",
        y = "Average Lobster Size") +
     scale_x_continuous(limits=c(2012, 2018), expand = c(0,NA))


print(plot4)
```




```
recent_counts |>
    dplyr::select(counts, mean_size, mpa) |>
    tbl_summary(by = mpa,
                statistic = list(all_continuous() ~ "{mean}")) |>
    modify_caption("**Comparing the mean counts and mean sizes of California Spiny Lobsters at MPA and
```

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

```
m1_ols_recent <- lm(counts ~ treat, data = recent_counts)

summ(m1_ols_recent, model.fit = FALSE)
```

```
print(check_model(m1_ols_recent,  check = "qq" ))
```

Table 2: **Comparing the mean counts and mean sizes of California Spiny Lobsters at MPA and non-MPA sites**

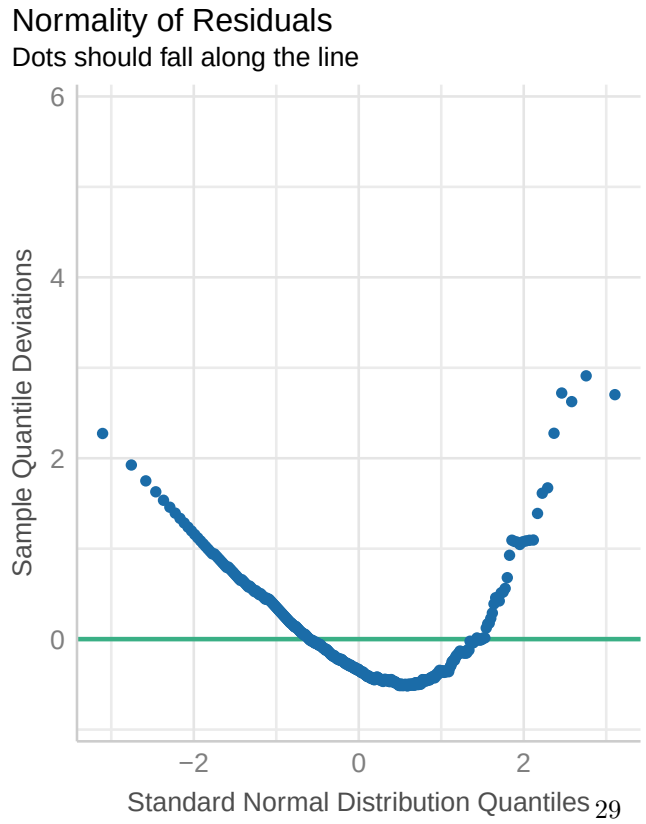
Characteristic	**MPA** N = 220[1]	**non_MPA** N = 246[1]
site		
AQUE	0 (0%)	91 (37%)
CARP	0 (0%)	116 (47%)
IVEE	104 (47%)	0 (0%)
MOHK	0 (0%)	39 (16%)
NAPL	116 (53%)	0 (0%)
year	2,018.0	2,018.0
counts	35	27
mean_size	80	74
Unknown	19	32

[1] n (%); Mean

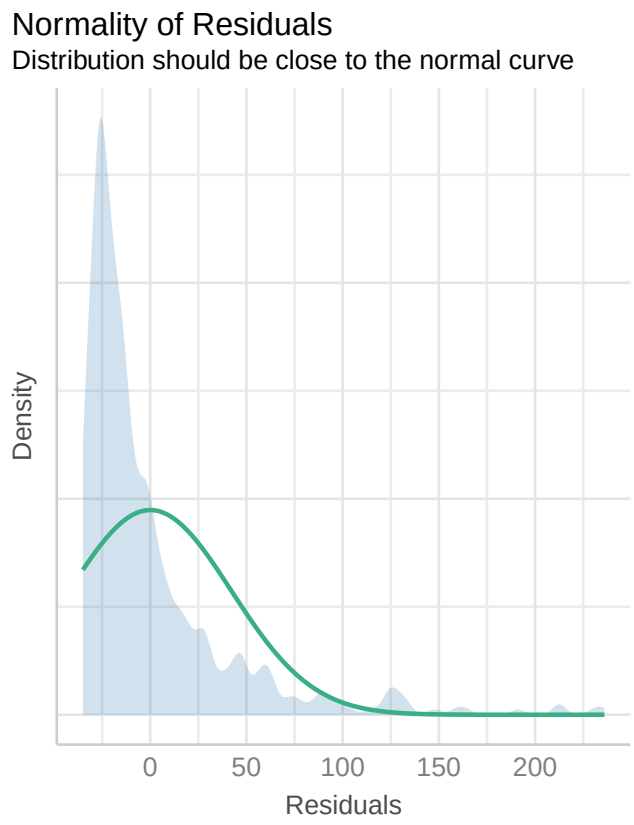
Observations	466
Dependent variable	counts
Type	OLS linear regression

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

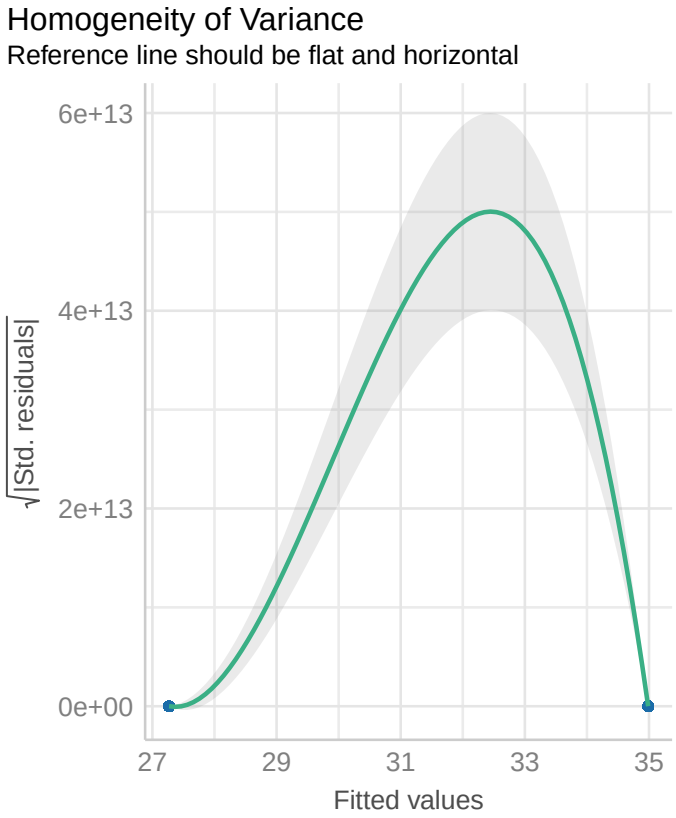
Standard errors: OLS



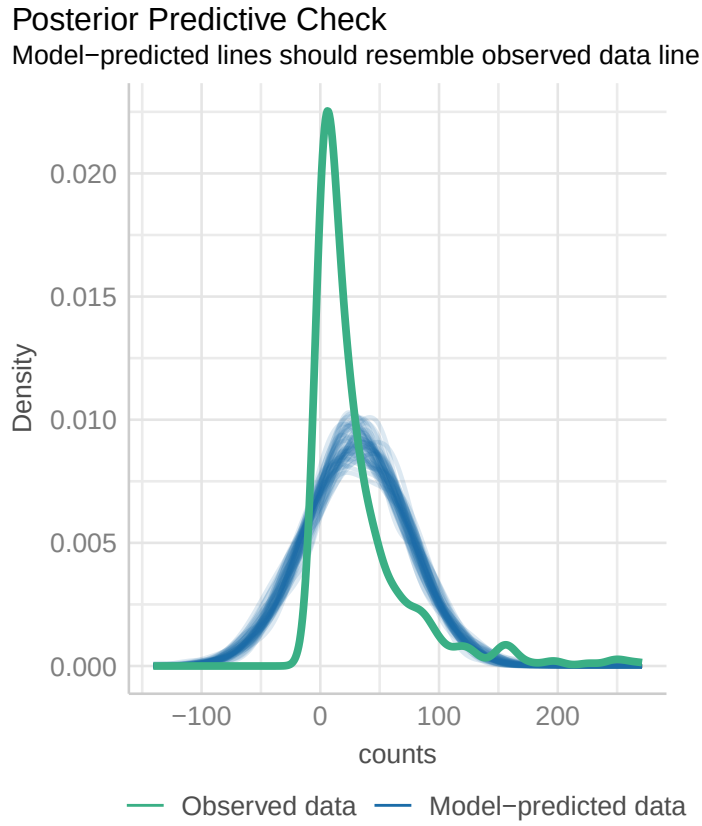
```
print(check_model(m1_ols_recent, check = "normality"))
```




```
print(check_model(m1_ols_recent, check = "homogeneity"))
```




```
print(check_model(m1_ols_recent, check = "pp_check"))
```




```
m2_pois_recent <- glm(counts ~ treat,
                      recent_counts,
                      family = poisson(link = "log"))

summ(m2_pois_recent)
```

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

$\chi^2(1)$	223.34
p	0.00
Pseudo-R^2 (Cragg-Uhler)	0.38
Pseudo-R^2 (McFadden)	0.01
AIC	21530.09
BIC	21538.38

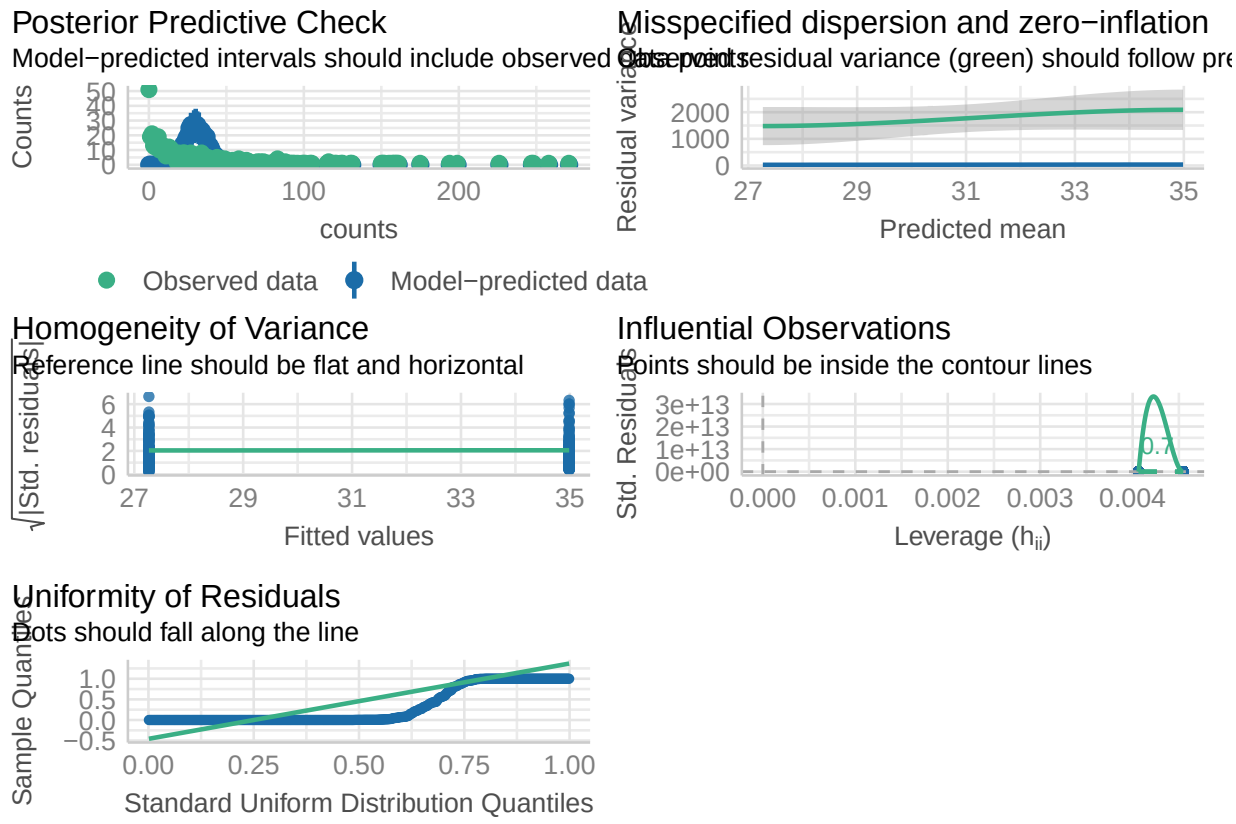
```
print(exp(.25) - 1)
```

```
## [1] 0.2840254
```

	Est.	S.E.	z val.	p
(Intercept)	3.31	0.01	270.75	0.00
treat	0.25	0.02	14.92	0.00

Standard errors: MLE

```
print(check_model(m2_pois_recent))
```




```
print(check_overdispersion(m2_pois_recent))
```

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

```
print(check_zeroinflation(m2_pois_recent))
```

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

```r
m3_nb_recent <- MASS::glm.nb(counts ~ treat,
                recent_counts)

summ(m3_nb_recent)
```

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

$\chi^2(464)$	4.08
p	0.04
Pseudo-R^2 (Cragg-Uhler)	0.01
Pseudo-R^2 (McFadden)	0.00
AIC	4058.04
BIC	4070.48

	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

```r
print(exp(.25) - 1)
```

```
## [1] 0.2840254
```

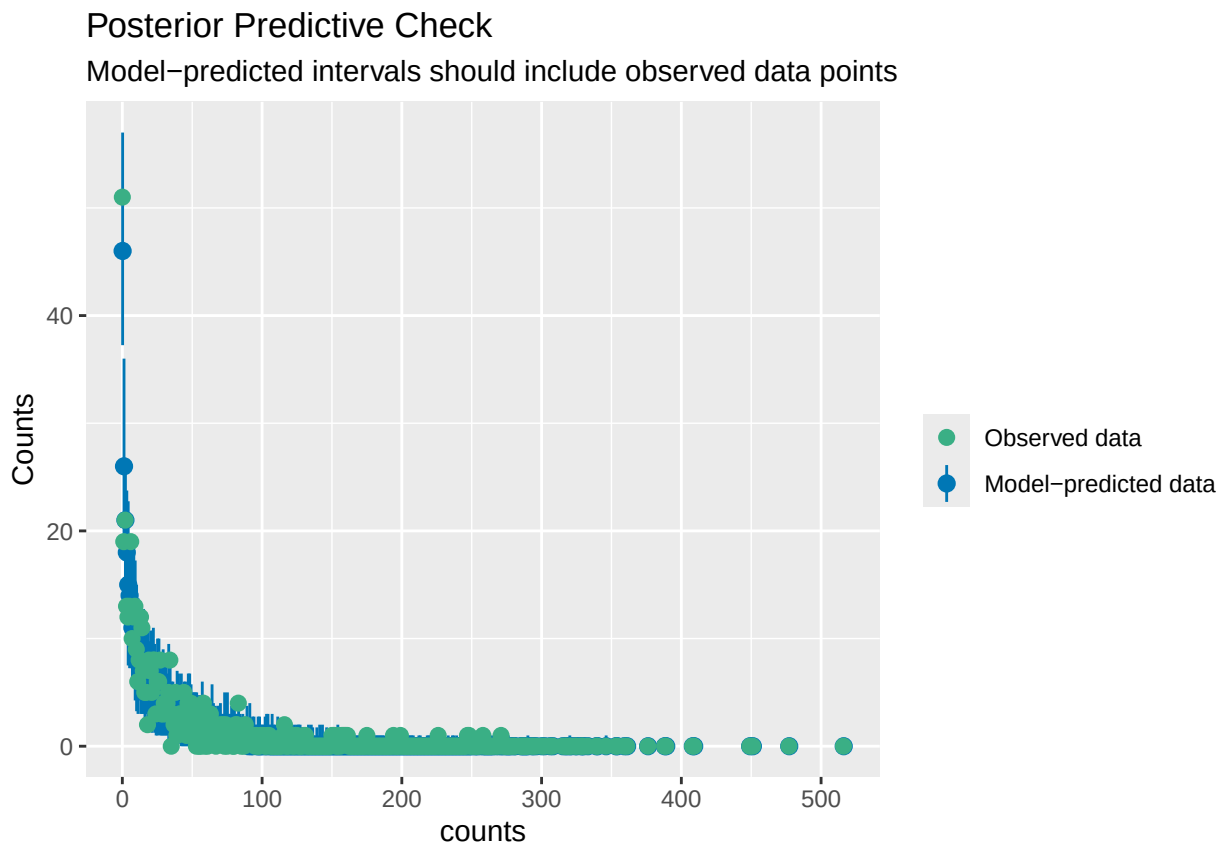
```r
print(check_overdispersion(m3_nb_recent))
```

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

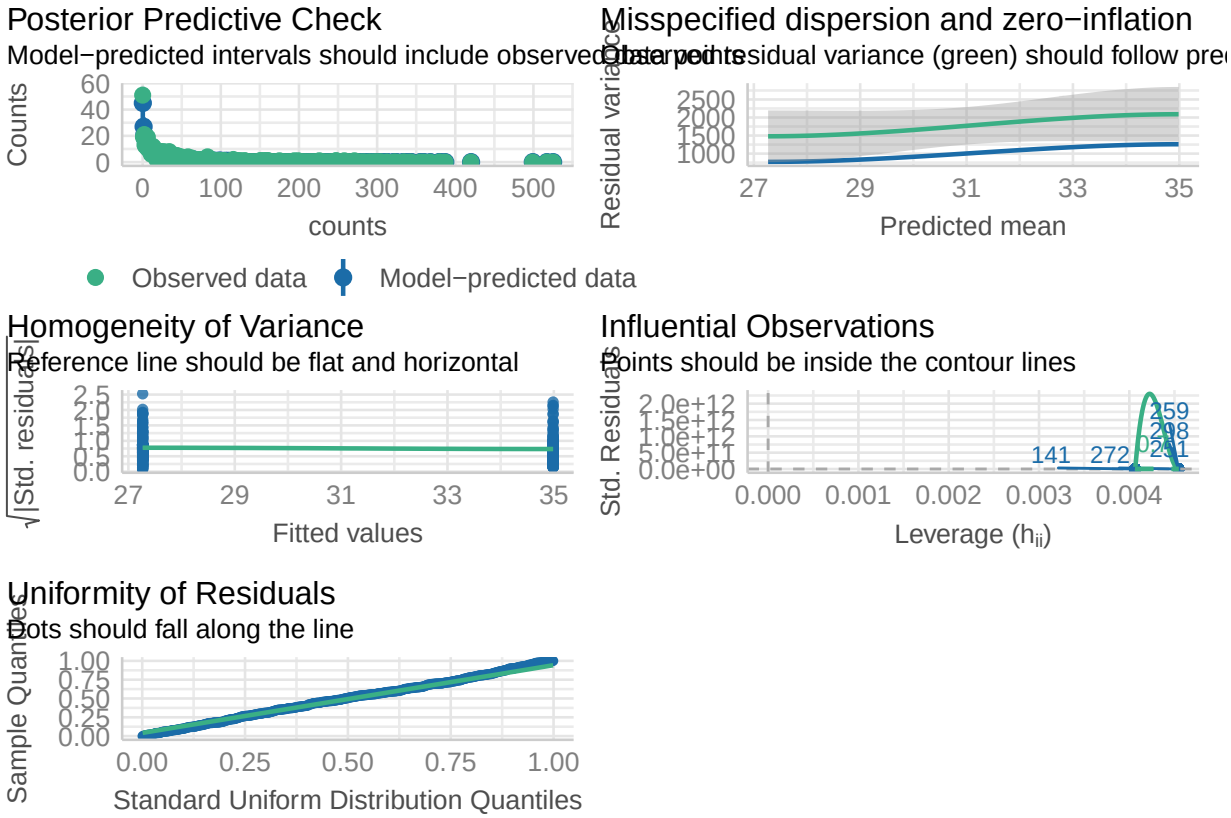
```r
print(check_zeroinflation(m3_nb_recent))
```

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

```r
print(check_predictions(m3_nb_recent))
```




```
print(check_model(m3_nb_recent))
```




```
export_summs(m1_ols_recent, m2_pois_recent, m3_nb_recent,
             model.names = c("OLS","Poisson", "NB"))
```

	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)
N	466	466	466
R2	0.01		
AIC	4813.06	21530.09	4058.04
BIC	4825.50	21538.38	4070.48
Pseudo R2		0.38	0.01

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

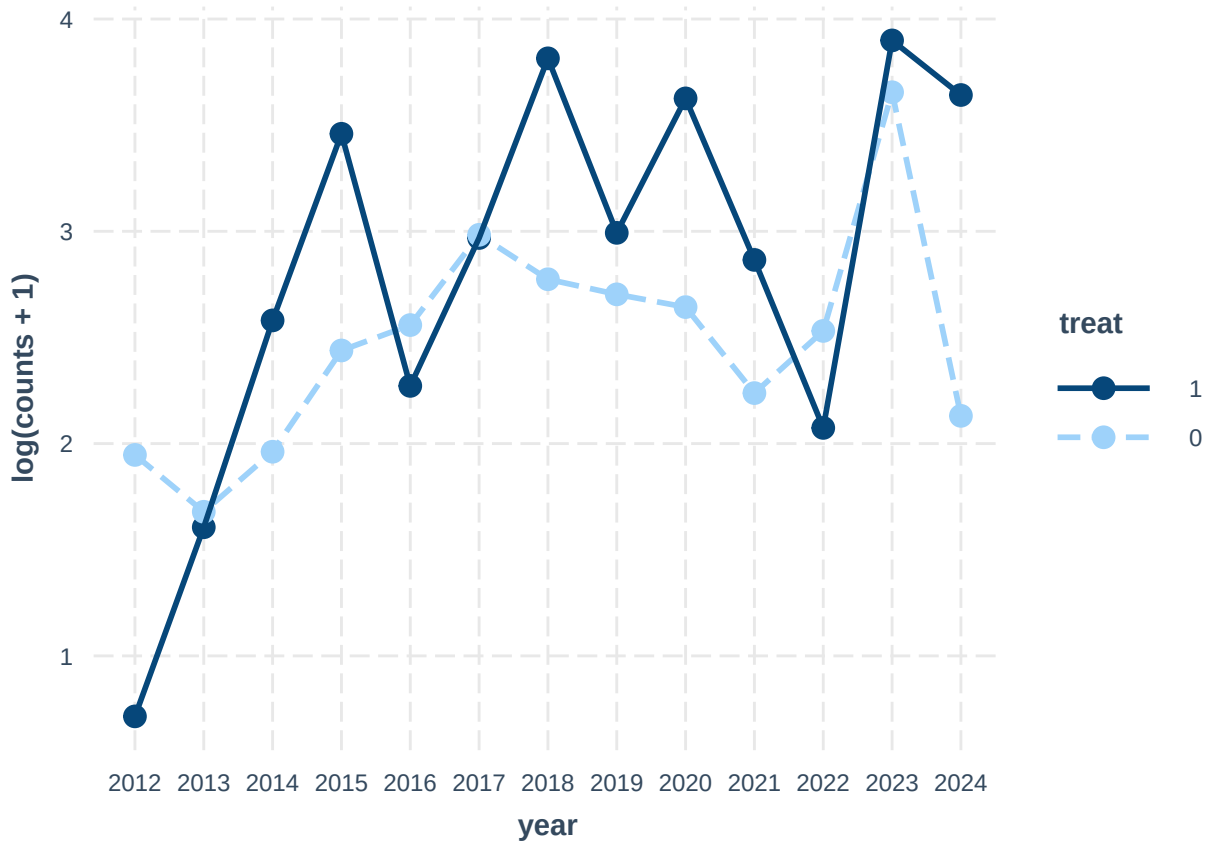
```r
ff_counts_recent <- recent_counts %>%
    mutate(year=as_factor(year))

m5_fixedeffs_recent <- lm(
    log(counts+1) ~ treat*year,
    data = ff_counts_recent)

summ(m5_fixedeffs_recent, model.fit = FALSE)
```

Observations	466
Dependent variable	log(counts + 1)
Type	OLS linear regression

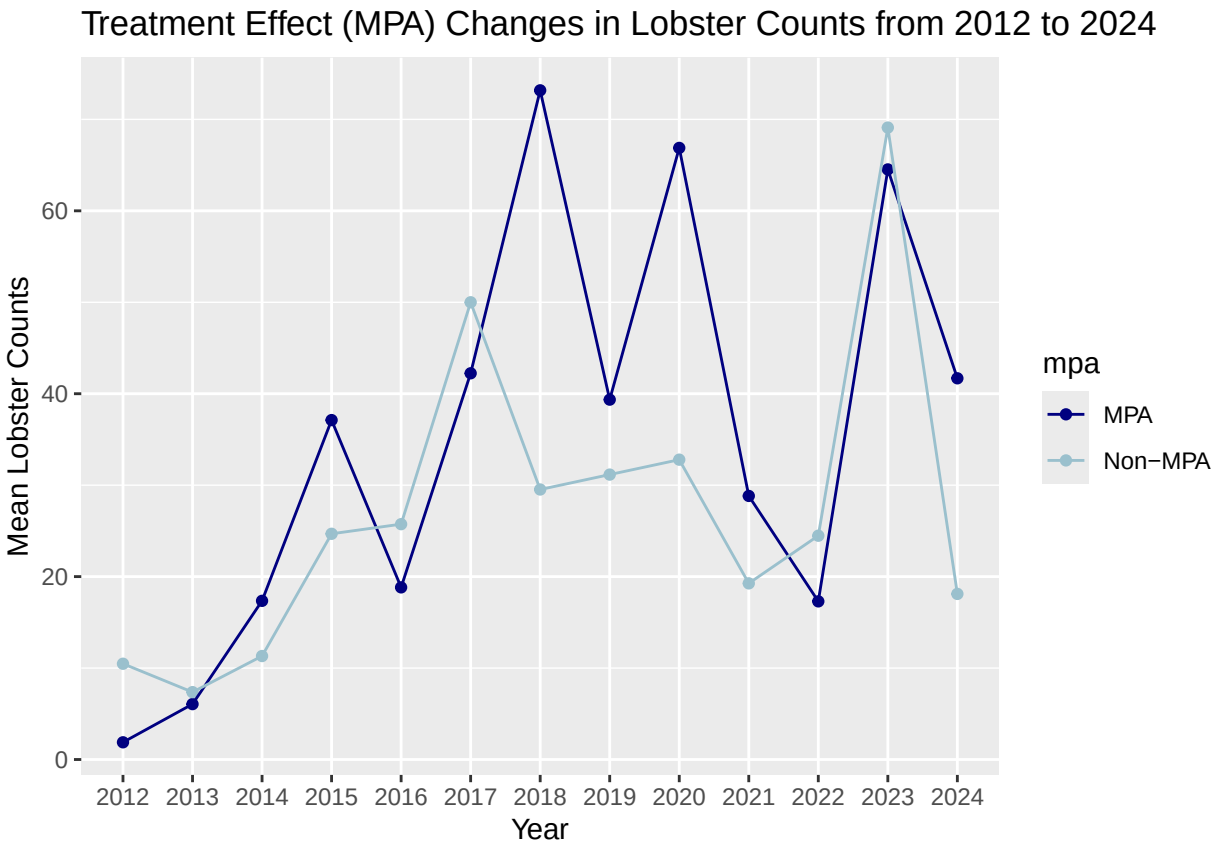
	Est.	S.E.	t val.	p
(Intercept)	1.95	0.29	6.71	0.00
treat	-1.23	0.42	-2.92	0.00
year2013	-0.27	0.41	-0.65	0.51
year2014	0.02	0.41	0.04	0.97
year2015	0.49	0.41	1.20	0.23
year2016	0.61	0.41	1.49	0.14
year2017	1.04	0.41	2.53	0.01
year2018	0.83	0.41	2.02	0.04
year2019	0.76	0.41	1.84	0.07
year2020	0.70	0.41	1.70	0.09
year2021	0.29	0.41	0.71	0.48
year2022	0.58	0.41	1.42	0.15
year2023	1.71	0.41	4.17	0.00
year2024	0.18	0.42	0.44	0.66
treat:year2013	1.16	0.60	1.94	0.05
treat:year2014	1.85	0.60	3.10	0.00
treat:year2015	2.25	0.60	3.77	0.00
treat:year2016	0.95	0.60	1.58	0.11
treat:year2017	1.22	0.60	2.04	0.04
treat:year2018	2.27	0.60	3.81	0.00
treat:year2019	1.52	0.60	2.55	0.01
treat:year2020	2.21	0.60	3.71	0.00
treat:year2021	1.86	0.60	3.12	0.00
treat:year2022	0.77	0.60	1.30	0.19
treat:year2023	1.48	0.60	2.47	0.01
treat:year2024	2.74	0.61	4.53	0.00

Standard errors: OLS

```r
print(interact_plot(m5_fixedeffs_recent, pred = year, modx = treat,
              outcome.scale = "response"))
```




```
plot_counts_recent <- recent_counts |>
    group_by(year, mpa) |>
    summarize(mean_counts = mean(counts)) |>
    mutate(year = as.factor(year)) |>
    ggplot(aes(x = year, y = mean_counts, color = mpa)) +
    geom_point() +
    geom_line(aes(group = mpa)) +
    labs(
        title = "Treatment Effect (MPA) Changes in Lobster Counts from 2012 to 2024",
        x = "Year",
        y = "Mean Lobster Counts"
    ) +
    scale_color_manual(values = c("navy", "lightblue3" ),
                       labels = c("MPA", "Non-MPA")) +
    scale_linetype_manual(values = c("solid", "longdash"))

print(plot_counts_recent)
```



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

```
export_summs(m1_ols, m2_pois, m3_nb, m1_ols_recent, m2_pois_recent, m3_nb_recent,
             model.names = c("OLS","Poisson", "NB", "2024 OLS", "2024 Poisson", "2024 NB"))
```

The results of the original OLS analysis versus this new one differs by 2.36. This result isn't surprising because we see a significant spike in 2018 - 2020 for MPA sites over non-MPA sites. This spike isn't seen in the original OLS analysis because it ends at 2018. The results of the original Poisson analysis versus this one differs by 5.03%. Once again, it makes sense that the recent poisson model has a bigger treatment effect because of the spike that is unaccounted for in the first model. Like the previous model, this recent Negative Binomial model has the same treatment effect as the Poisson model.

When looking at the plots, we can see that after the spike from 2018-2020, the MPA site actually dips lower in counts than the non-MPA sites. There is also a significant spike in both MPA and non-MPA sites in 2023. My theory is that the spillover effect that we are seeing in the original data is continuing even more here. As the spillover builds up, it makes the non-MPA sites follow the trends of the MPA sites.

	OLS	Poisson	NB	2024 OLS	2024 Poisson	2024 NB
(Intercept)	22.73 ***	3.12 ***	3.12 ***	27.27 ***	3.31 ***	3.31 ***
	(3.57)	(0.02)	(0.12)	(2.69)	(0.01)	(0.08)
treat	5.36	0.21 ***	0.21	7.72 *	0.25 ***	0.25 *
	(5.20)	(0.03)	(0.17)	(3.91)	(0.02)	(0.12)
N	252	252	252	466	466	466
R2	0.00			0.01		
AIC	2593.35	11365.62	2088.53	4813.06	21530.09	4058.04
BIC	2603.94	11372.68	2099.12	4825.50	21538.38	4070.48
Pseudo R2		0.25	0.01		0.38	0.01

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