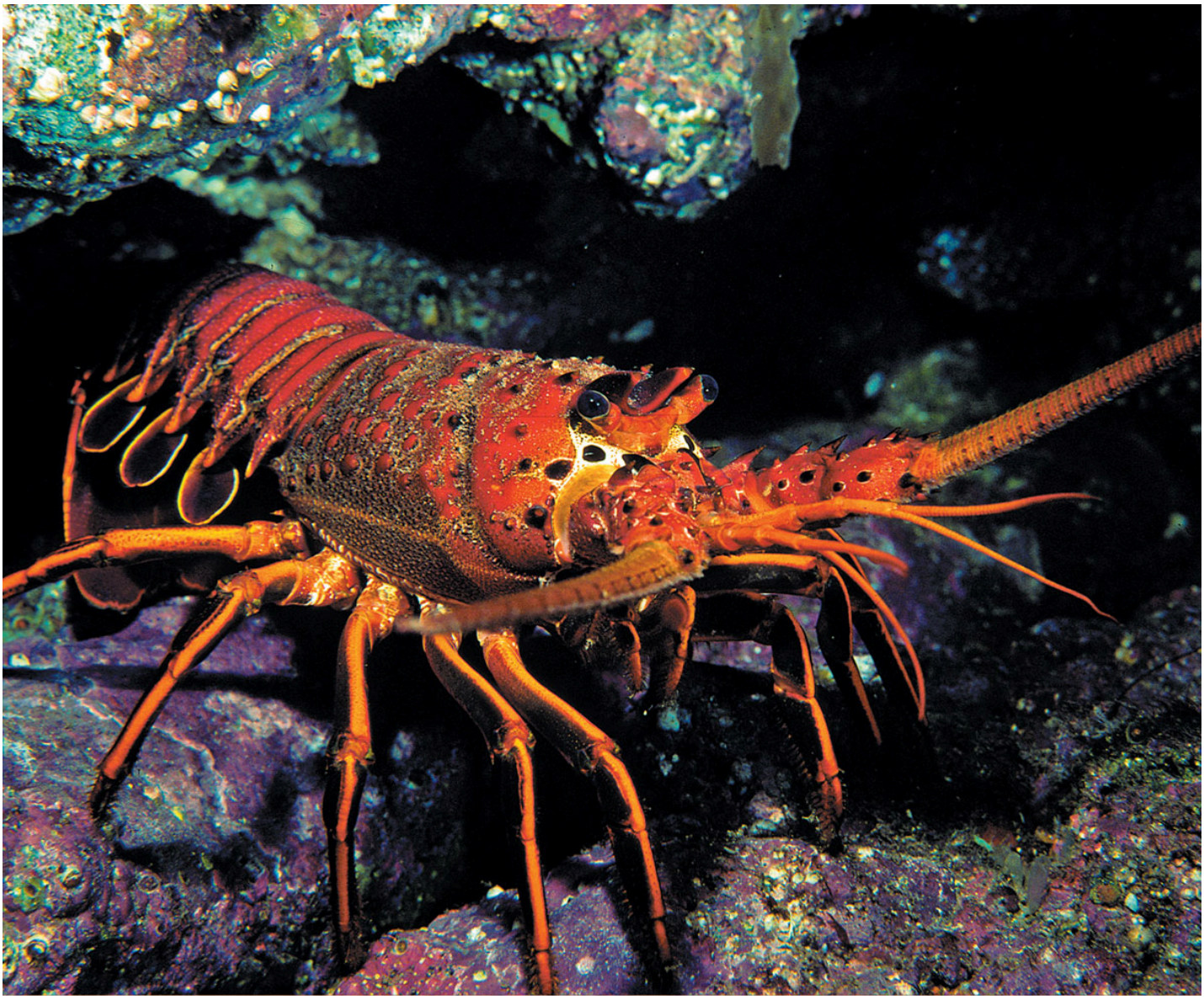
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

EDS 241

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: Kimberlee Wong

```
library(tidyverse)
library(here)
library(dplyr)
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(viridis)
```

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.

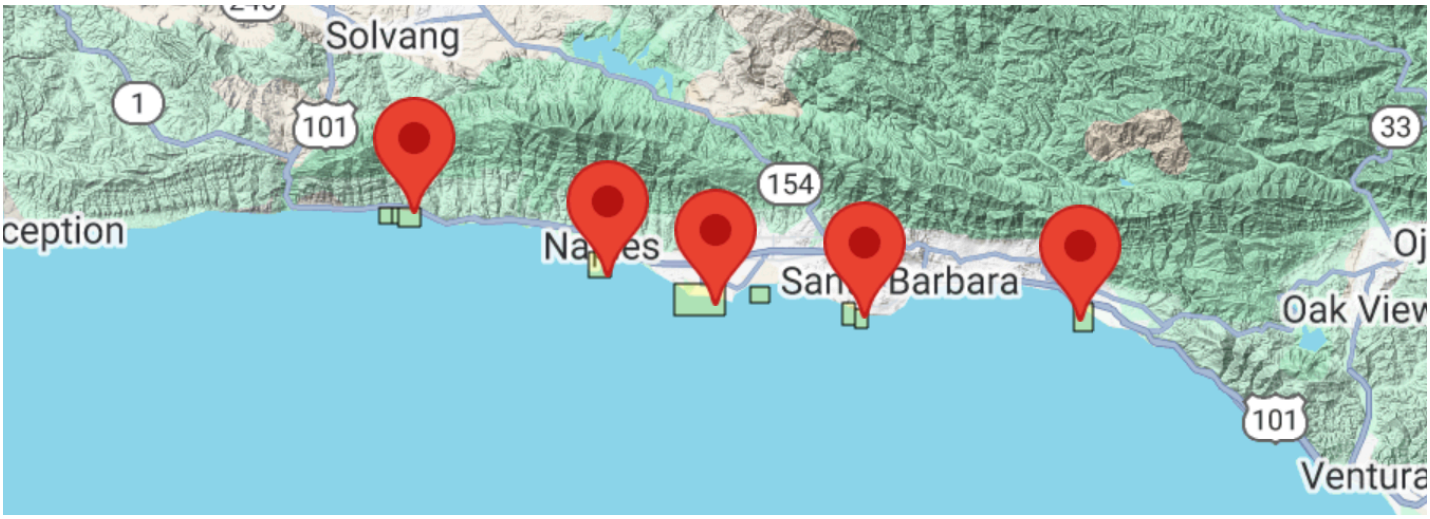
Introduction

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 think that this comparison is not centris paribus and the control sites do not provide a strong counterfactual for our treatment sites. As seen from the map above, the five sites have quite a bit of difference between them, so I think that there are potentially other environmental factors that could affect the health of the lobster population such as human interaction, temperature, or coastal conditions.

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"`)

rawdata <- read_csv(here("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"
```

```
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 lobster
s observed at each site-year-transect row-observation.

#HINT(e): Use `case_when()` to create the 3 new variable columns
spiny_counts <- tidydata %>%
    group_by(year, site, transect) %>%
    summarize(counts = as.integer(sum(count, na.rm = TRUE)),
              mean_size = mean(size_mm, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(mpa = factor(site, levels = c("AQUE",
                                         "CARP",
                                         "MOHK",
                                         "IVEE",
                                         "NAPL"),
                               labels = c("non_MPA",
                                          "non_MPA",
                                          "non_MPA",
                                          "MPA",
                                          "MPA"))) %>%
    mutate(treat = case_when(mpa == "MPA" ~ 1,
                             mpa == "non_MPA" ~ 0))
```

> 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/)

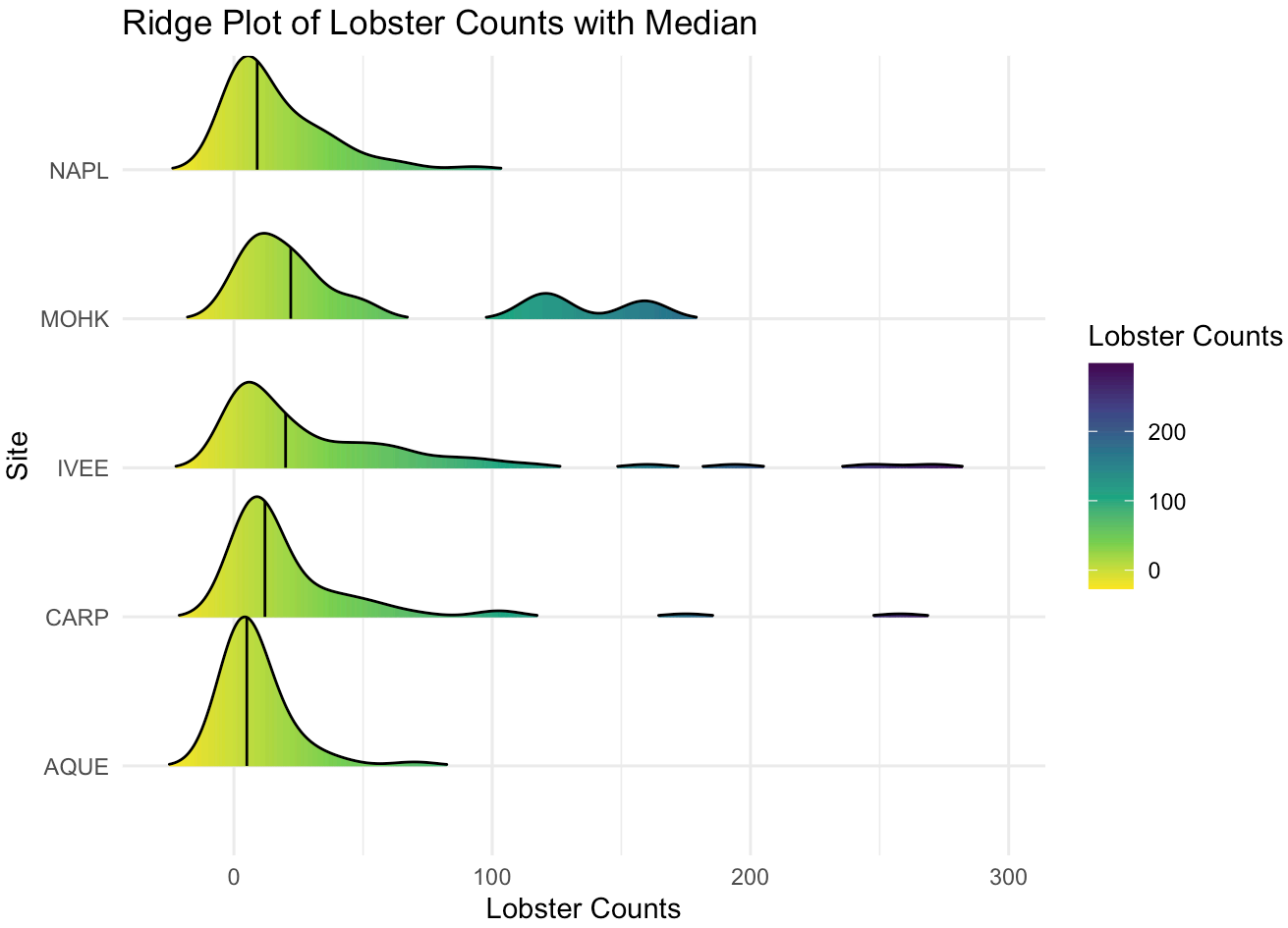
Create plots displaying the distribution of lobster **counts**:

1. grouped by reef site
2. grouped by MPA status
3. grouped by year

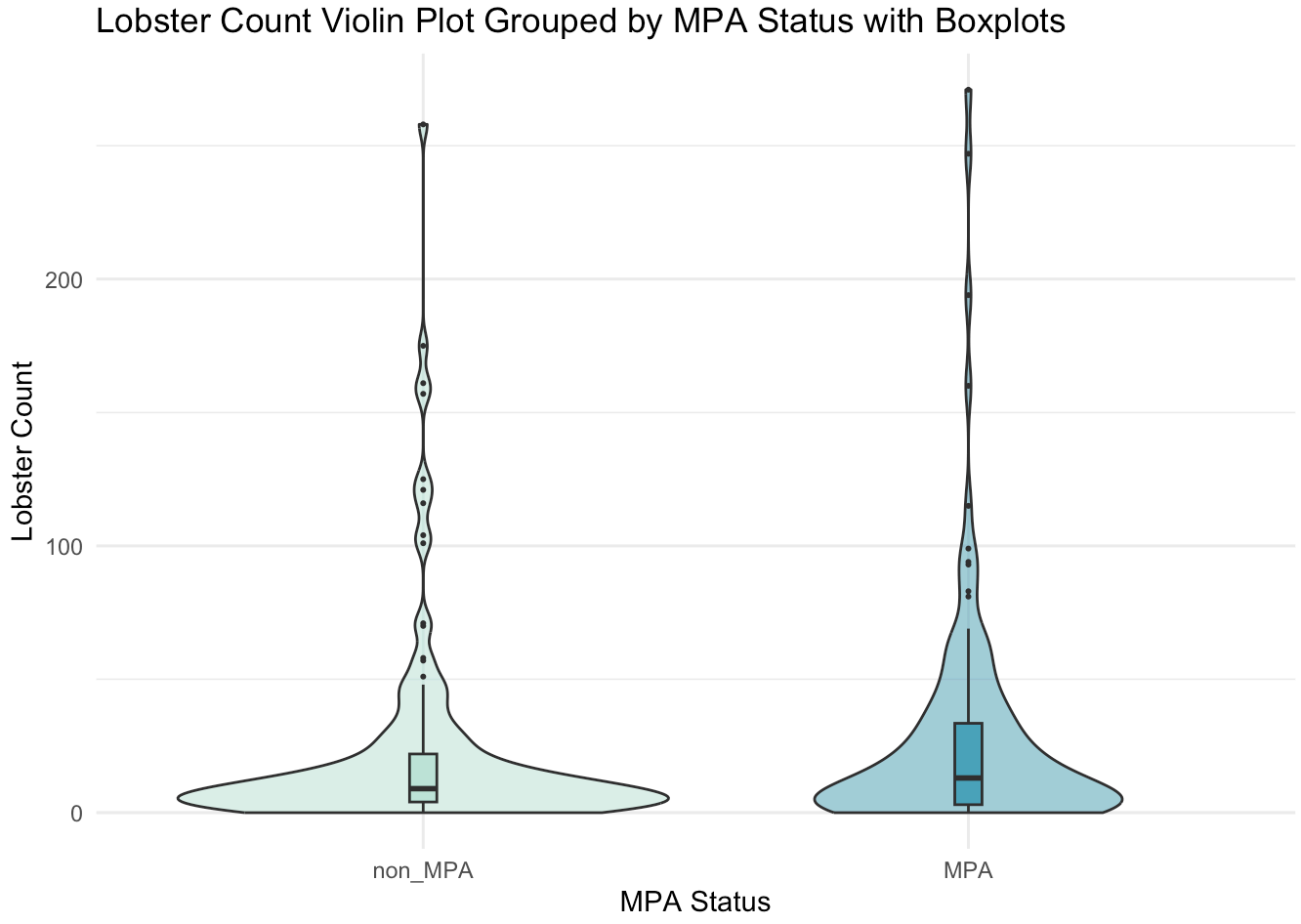
Create a plot of lobster **size** :

4. You choose the grouping variable(s)!

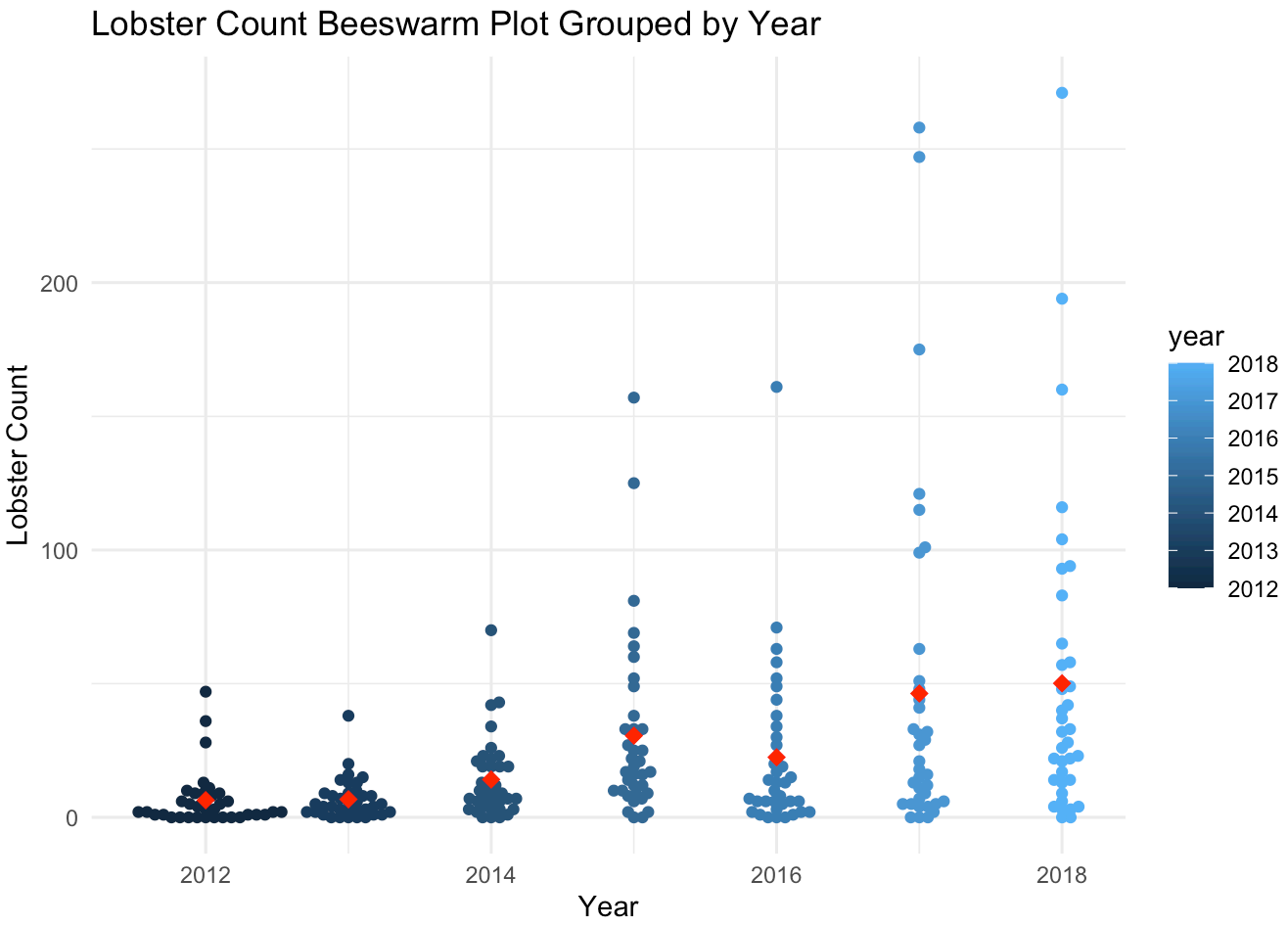
```
# plot 1: Lobster Count Ridge Plot Grouped by Site
spiny_counts %>%
    ggplot(aes(x = counts, y = site, fill = ..x..)) +
    geom_density_ridges_gradient(scale = 1, rel_min_height = 0.01,
                                 quantile_lines = TRUE, alpha = 0.75,
                                 quantiles = 2) +
    scale_fill_viridis_c(name = "Lobster Counts", option = "D", direction = -1) +
    labs(x = "Lobster Counts", y = "Site", title = "Ridge Plot of Lobster Counts
with Median") +
    theme_minimal()
```



```
# plot 2: Lobster Count Violin Plot Grouped by MPA Status
spiny_counts %>%
    ggplot(aes(x = mpa, y = counts, fill = mpa)) +
    geom_violin(aes(alpha = 0.5)) +
    geom_boxplot(width = 0.05, outlier.size = .4) +
    scale_fill_manual(values = c("#BCE4D8", "#49A4B9")) +
    labs(x = "MPA Status", y = "Lobster Count", title = "Lobster Count Violin Plo
t Grouped by MPA Status with Boxplots") +
    theme_minimal() +
    theme(legend.position = "none")
```

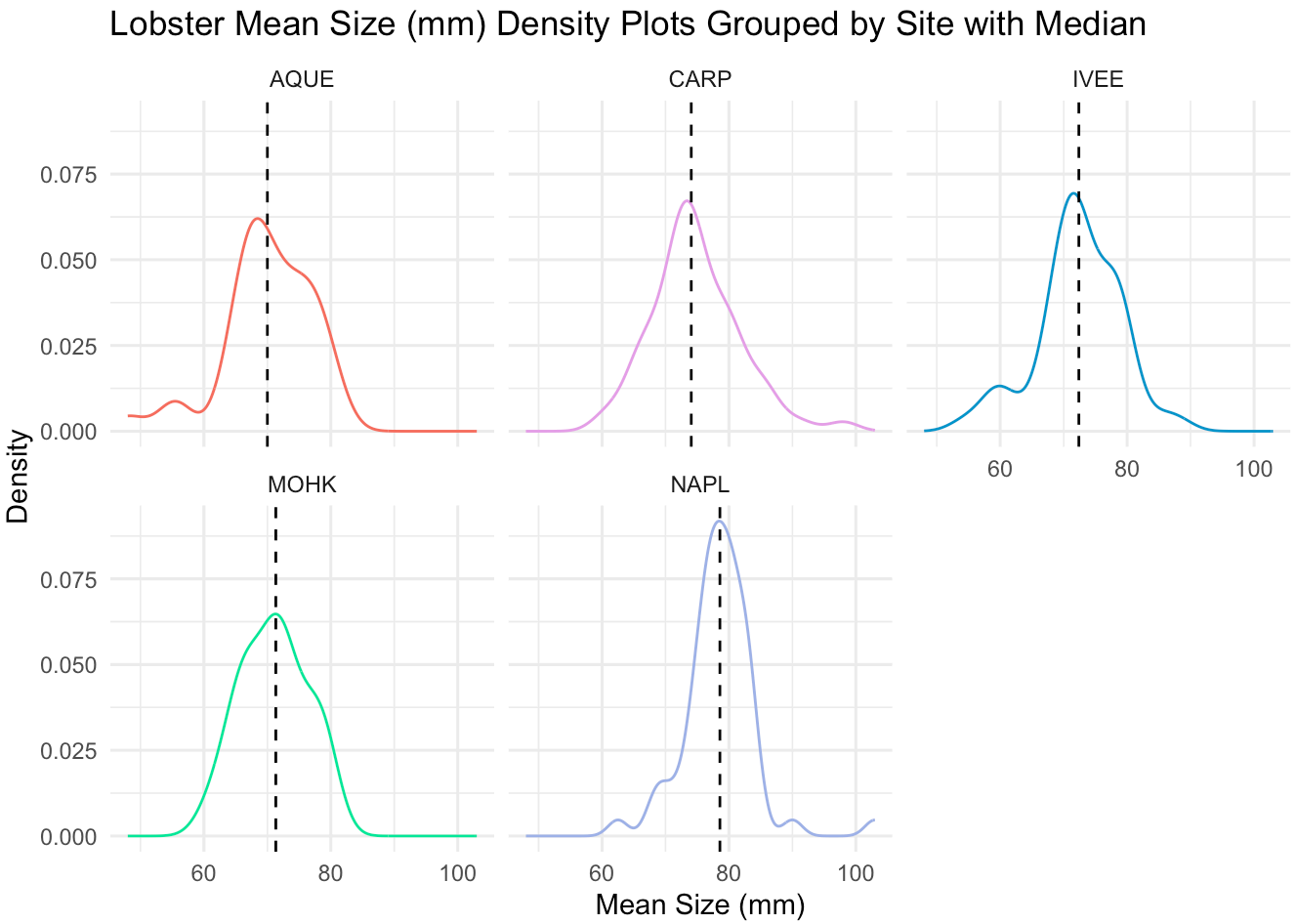



```
# plot 3: Lobster Count Beeswarm Plot Grouped by Year with Mean
spiny_counts %>%
    ggplot(aes(x = year, y = counts, color = year)) +
    geom_beeswarm() +
    stat_summary(fun = "mean", geom = "point", shape = 18, size = 3, color = "re
d") +  # Mean indicator
    labs(x = "Year", y = "Lobster Count", title = "Lobster Count Beeswarm Plot Gr
ouped by Year") +
    theme_minimal()
```




```
# plot 4: Lobster Mean Size Density Plot Grouped by Site
size_median <- spiny_counts %>%
    group_by(site) %>%
    summarise(median_size = median(mean_size, na.rm = TRUE))

spiny_counts %>%
    ggplot() +
    geom_density(aes(x = mean_size, color = site), ) +
    facet_wrap(~site) +
    scale_color_manual(values = c("#F76D5E", "#e6a0e7", "#0696cc", "#03ea9d", "#a
0b2e7")) +
    geom_vline(data = size_median, aes(xintercept = median_size), color = "blac
k", linetype = "dashed") +
    labs(y = "Density", x = "Mean Size (mm)", title = "Lobster Mean Size (mm) Den
sity Plots Grouped by Site with Median") +
    theme_minimal() +
    theme(legend.position = "none")
```



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)

```
spiny_counts %>%
    dplyr::select(treat, year, site, mean_size, counts) %>%
    tbl_summary(by = treat,
                statistic = list(all_continuous() ~ "{mean}")) %>%
    modify_caption("**Differences in mean statistics between sites that are MPAs
(1) and non MPAs (0)**")
```

Differences in mean statistics between sites that are MPAs (1) and non MPAs (0)

Characteristic	0 N = 133[1]	1 N = 119[1]
year		
2012	19 (14%)	17 (14%)
2013	19 (14%)	17 (14%)

[1] n (%); Mean

Characteristic	0 N = 133[1]	1 N = 119[1]
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%)
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%)
mean_size	73	76
Unknown	15	12
counts	23	28

[1] n (%); Mean

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-squa
re)

m1_ols <- lm(counts ~ treat, data = spiny_counts)

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

Observations	252
Dependent variable	counts

Type	OLS linear regression

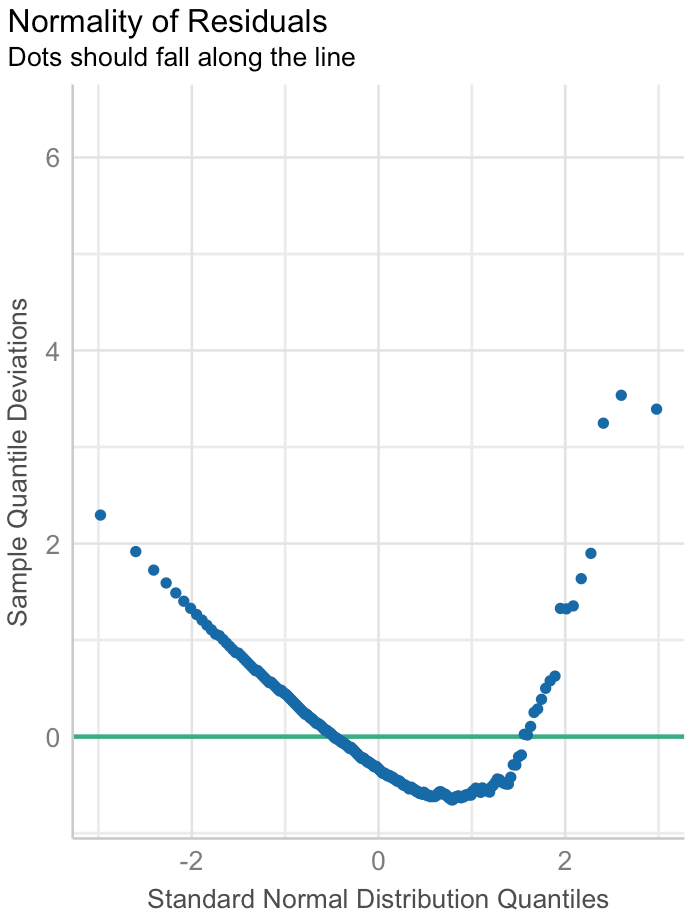
	Est.	S.E.	t val.	p
(Intercept)	22.73	3.57	6.36	0.00
treat	5.36	5.20	1.03	0.30

Standard errors: OLS

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

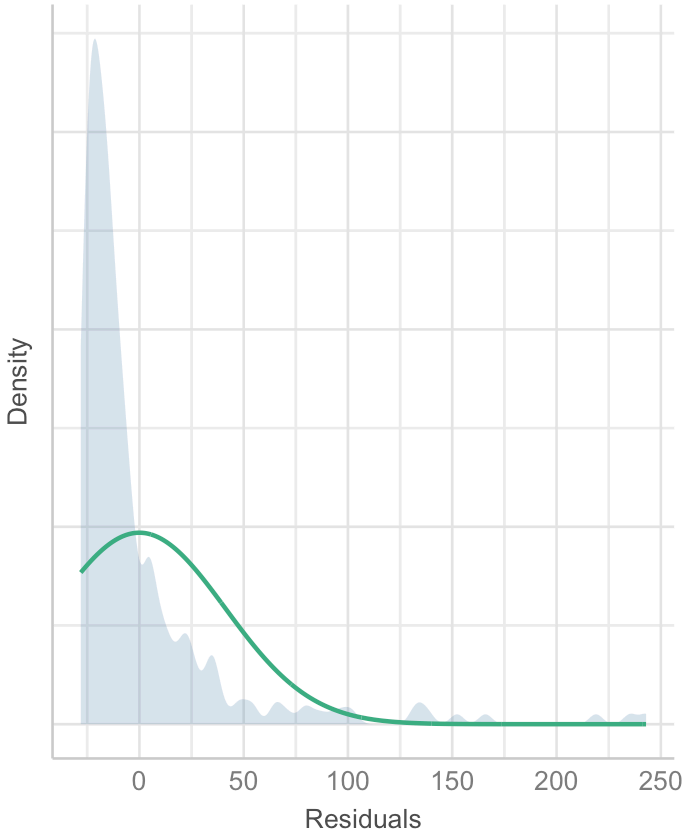
d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

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



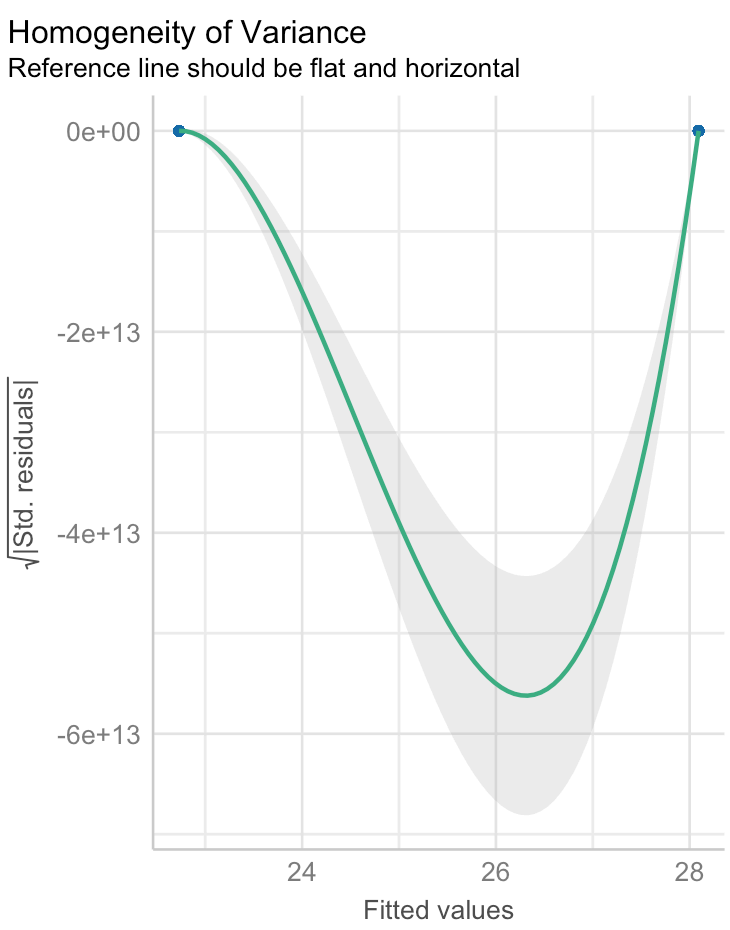
Because the dots on this qq plot do not fall on the green line, this means that are residuals are not normally distributed. This could be an indicator that OLS is not a good model to fit our data.

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



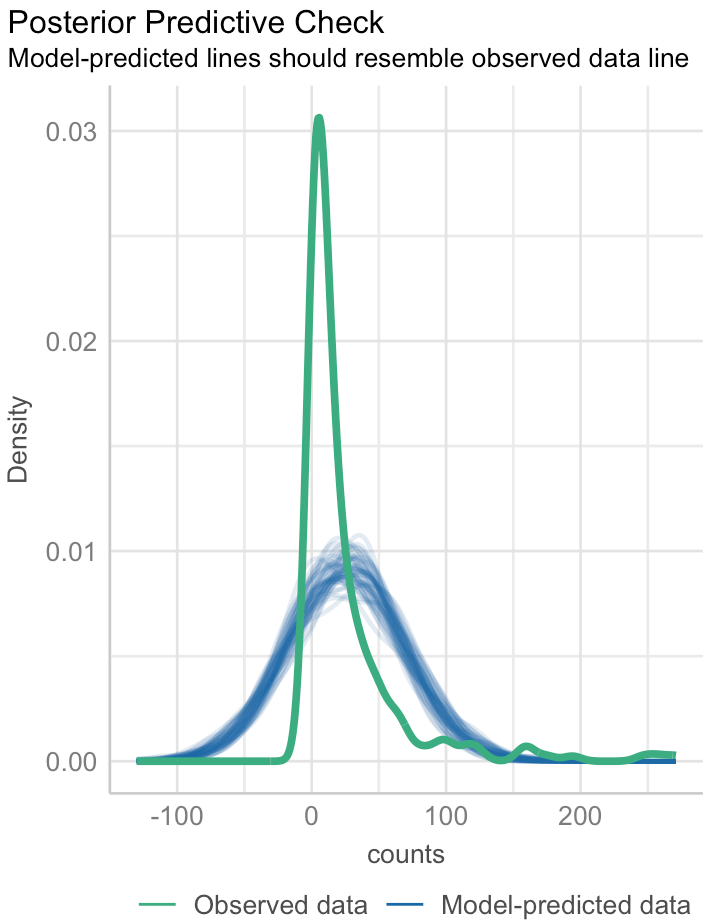
Again, similar to the last plot, this plot shows how our residuals are quite far from the normal, green curve that is used as a reference. We can come to the same conclusion that OLS might be a poor choice.

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



This plot is checking for the homogeneity of variance. According to the note, the reference line should be flat and horizontal, but ours is significantly curved like a U. So we are experiencing heteroscedasticity that could've come from biased data or again the wrong model choice.

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



Similar to the prior three plots, our model-predicted data is very far off from the ideal reference line. This plot is checking how well our model can predict aspects of the data like mean and standard deviations. In our case, our model does it very poorly once again showing that OLS is most likely a bad choice for our data.

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.

Because the Poisson model output from the glm() function returns coefficients in the log-scale, we need to convert it to percent change in order to properly interpret it. After doing that we get, 0.2337, which means that there is a 23.37% increase in lobster count in MPAs (treated site).

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

In the Poisson model, the assumption is equal dispersion. This means that the mean of the lobster count is equal to the variance of the count. Overdispersion means that the variance is greater than the mean. In our context, this would mean that the variance of the lobster count is greater than the mean of the lobster count.

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

The OLS model had an insignificant p-value (0.3) for the treatment effect. This means that any relationship we found could have been random chance. But in this Poisson model, the p-value was significant (0.0) and found that there is about a 23.37% increase in lobster counts in MPAs.

```
#HINT1: Incidence Ratio Rate (IRR): Exponentiation of beta returns coefficient wh
ich is interpreted as the 'percent change' for a one unit increase in the predict
or

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

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

```
# convert from log to percent change
print(exp(0.21) - 1)
```

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

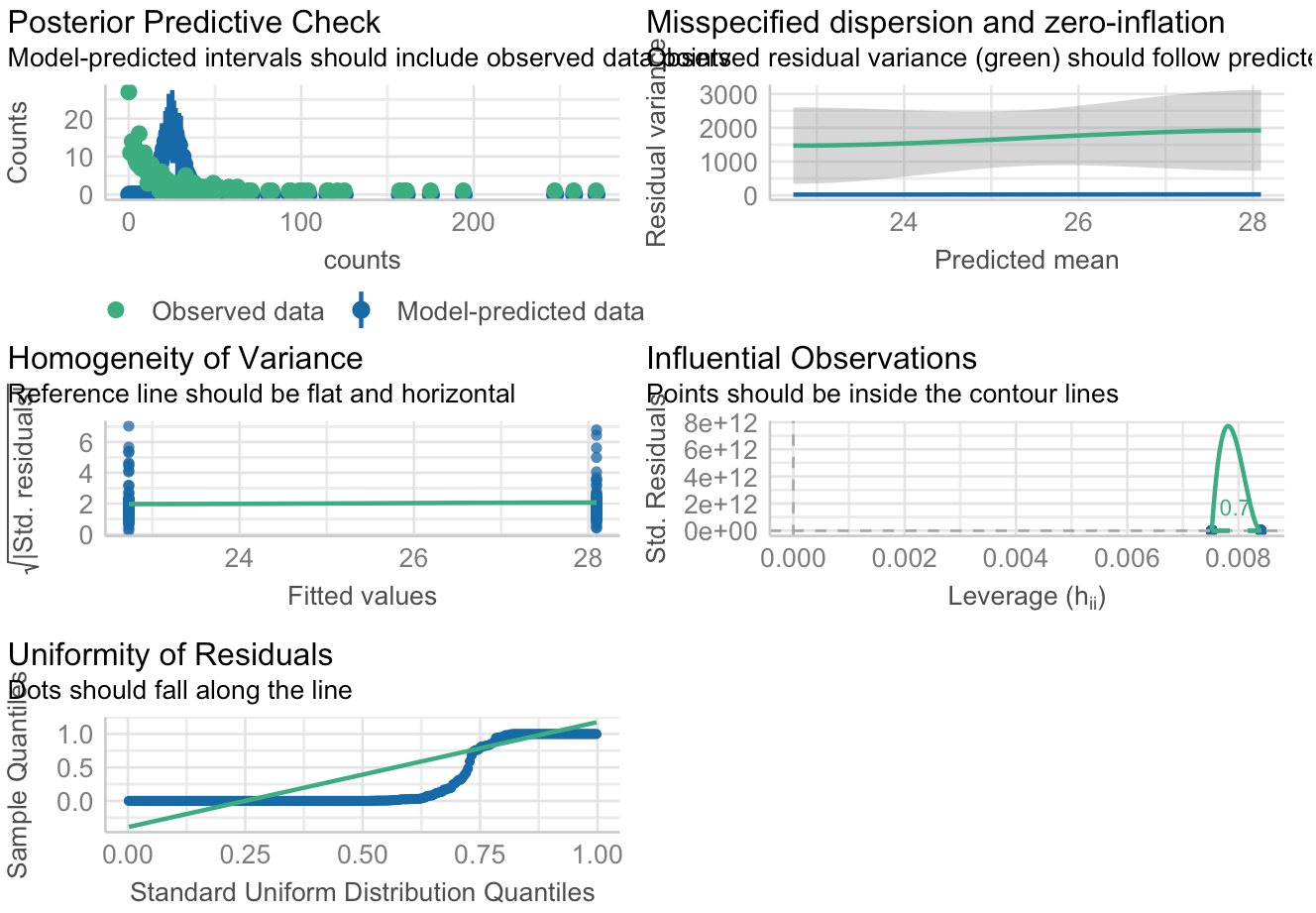
e. Check the model assumptions. Explain results.

Similar to OLS, it seems that the Poisson model is also not the ideal choice for our data. While the posterior predictive check does seem to better than the OLS, showing improvement in the model, it still deviates quite significantly from the reference. Our residuals do not align with the reference, suggesting this is the wrong model, and there are signficant zero inflation in our model.

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

**The over-dispersion tests shows that our model is overdispersed, meaning the lobster count variance is larger the mean. The p-value is also < 0.001, so this test is significant. The zero-inflation test also shows that we have 27 more zeros than predicted, so this also shows the model is not ideal.

```
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.

After both the OLS and Poisson models proved to be not ideal, for the assumptions deviated greatly from the predictions, a negative bionomial model is a good choice to try next. This model adds a dispersion parameter, so it can account for the overdispersion that was observed earlier.

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

Very similar to the Poisson model, this model finds a 23.37% increase in lobster counts in MPAs compared to non-MPAs, but this p-value is statistically insignificant unlike the Poisson model

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

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

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

```
# Convert from log to percent change
print(exp(0.21) - 1)
```

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

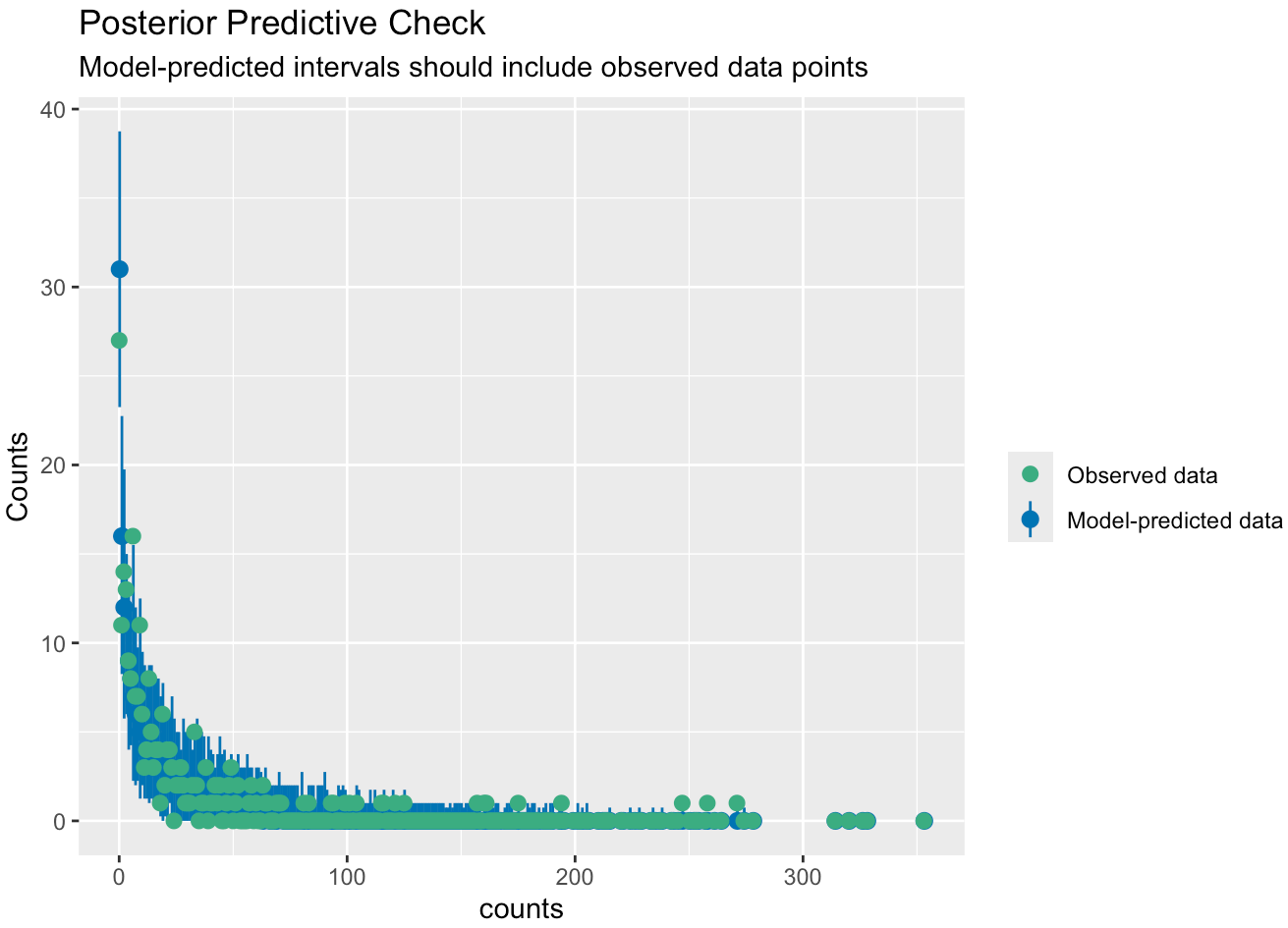
```
check_overdispersion(m3_nb)
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.400
##           p-value = 0.064
```

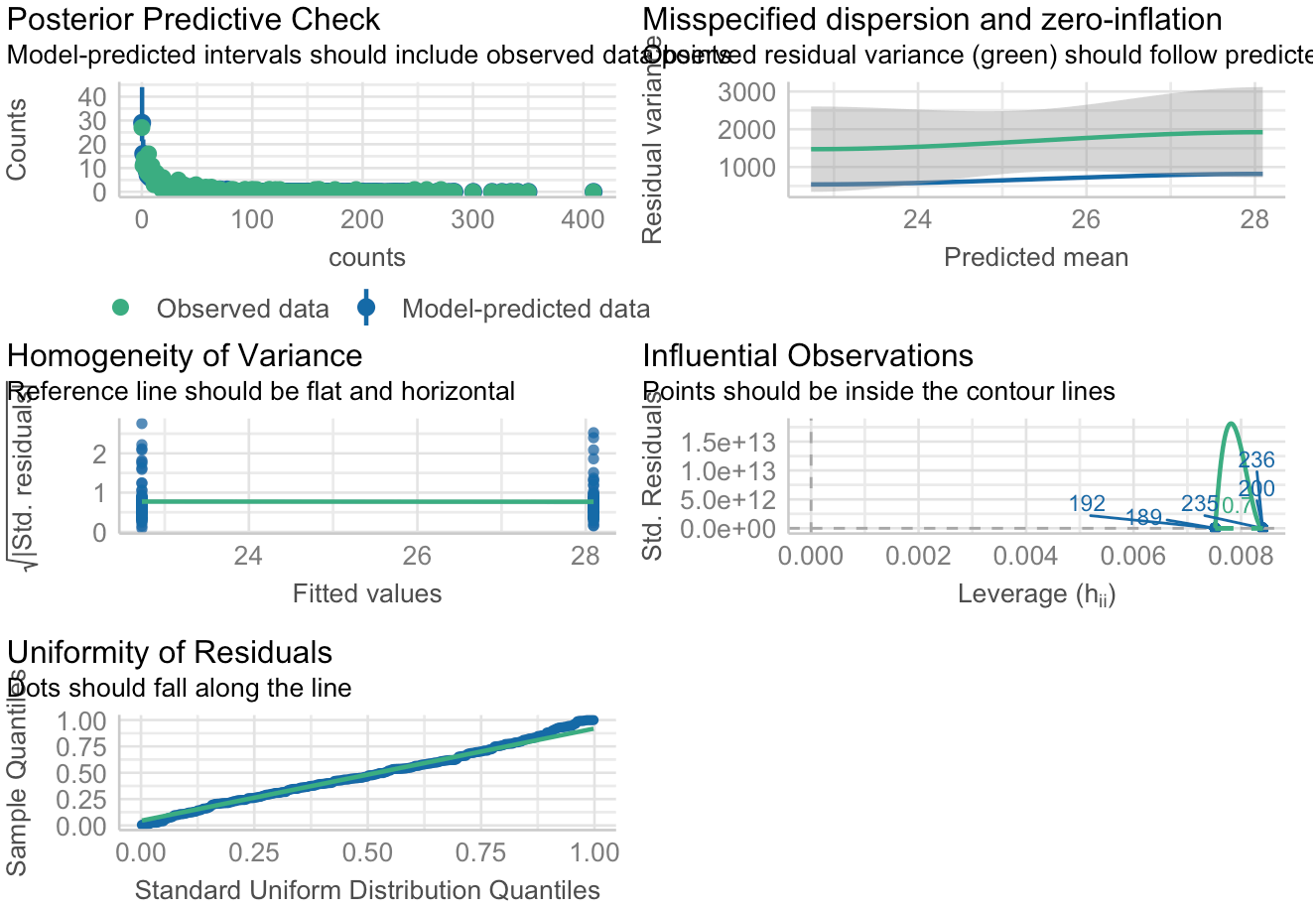
```
check_zeroinflation(m3_nb)
```

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

```
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.

After converting all of the treatment estimates from all three models to percent change, the result shows that all three are very similar. This suggests that the treatment effect is stable and robust, for it stays consistent across the different model specifications

```
jtools::export_summs(m1_ols, m2_pois, m3_nb,
                     model.names = c("OLS","Poisson", "NB"))
```

	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.

```
# Find percent change of lobster counts in each model:

# ratio of treatment beta coeff / intercept
ols_pc = (5.36/22.73)  # % change = + 23.58%

pois_pc = (exp(0.21) - 1)  # % change = + 23.37%
nb_pc = (exp(0.21) - 1)  # % change = + 23.37%
```

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)

This model has estimated the treatment effect by itself and in an interaction with year as a factor. When there is no treatment, the non MPA populations are steadily increasing, while the treatments experience a more dynamic increase, decrease, and then increase pattern.

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

The main effect for treatment is negative because it represents the year 2012, for this is our reference level. In this year, there were more lobsters in non-MPAs than MPAs. This makes sense because this was the first year the MPAs were implemented, so the population wouldn't have increased that quickly. In addition to this, its possible that the original populations before the MPAs were implemented were lower than the non-MPAs and that's why they were implemented in the first place.

```
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)
```

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

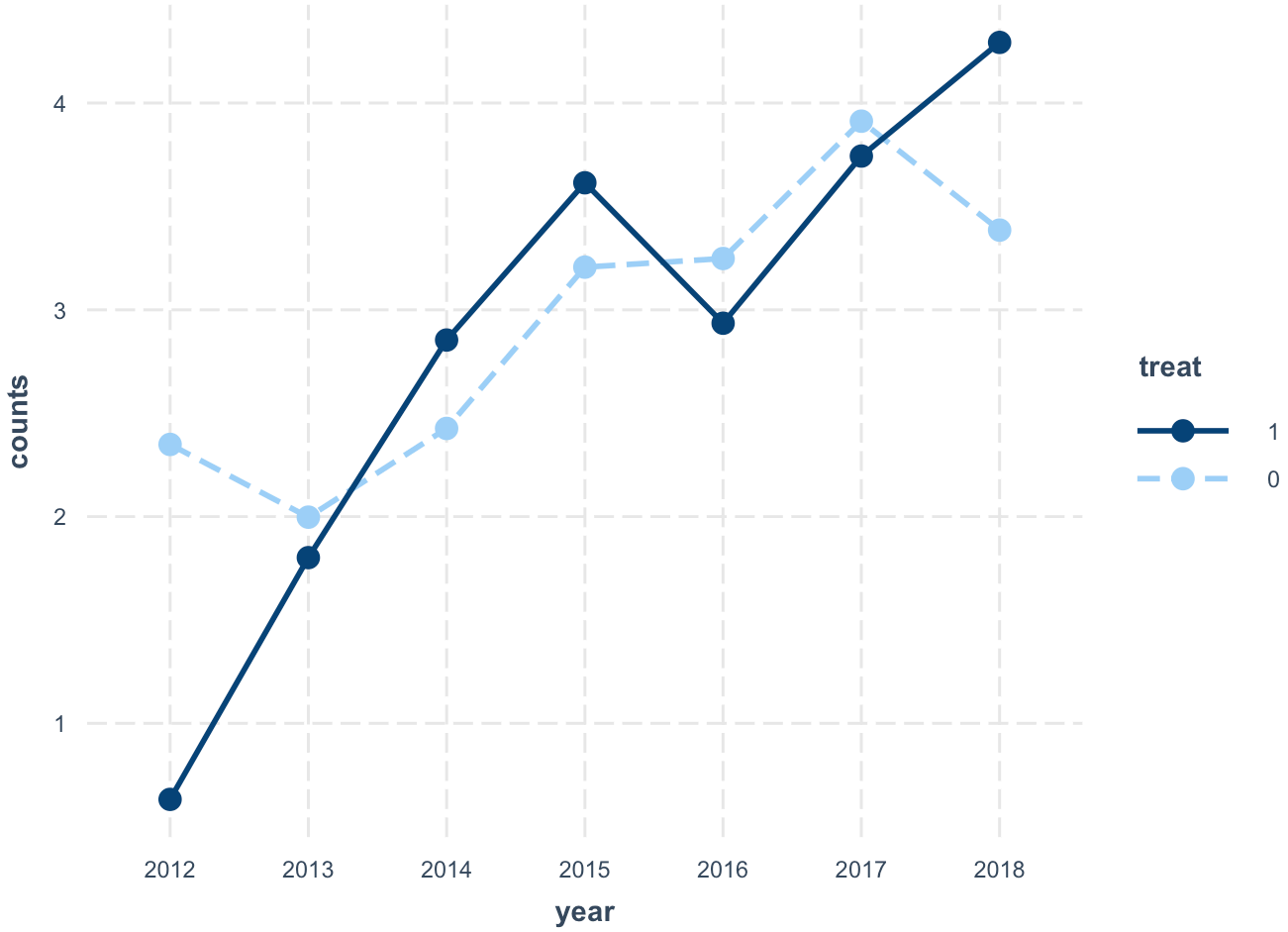
Standard errors: MLE

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.

This plot confirms what I had mentioned earlier. The MPA populations start lower than non-MPAs, which is why the initial treatment effect is negative. The patterns I described earlier also seem to fit the plot.

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



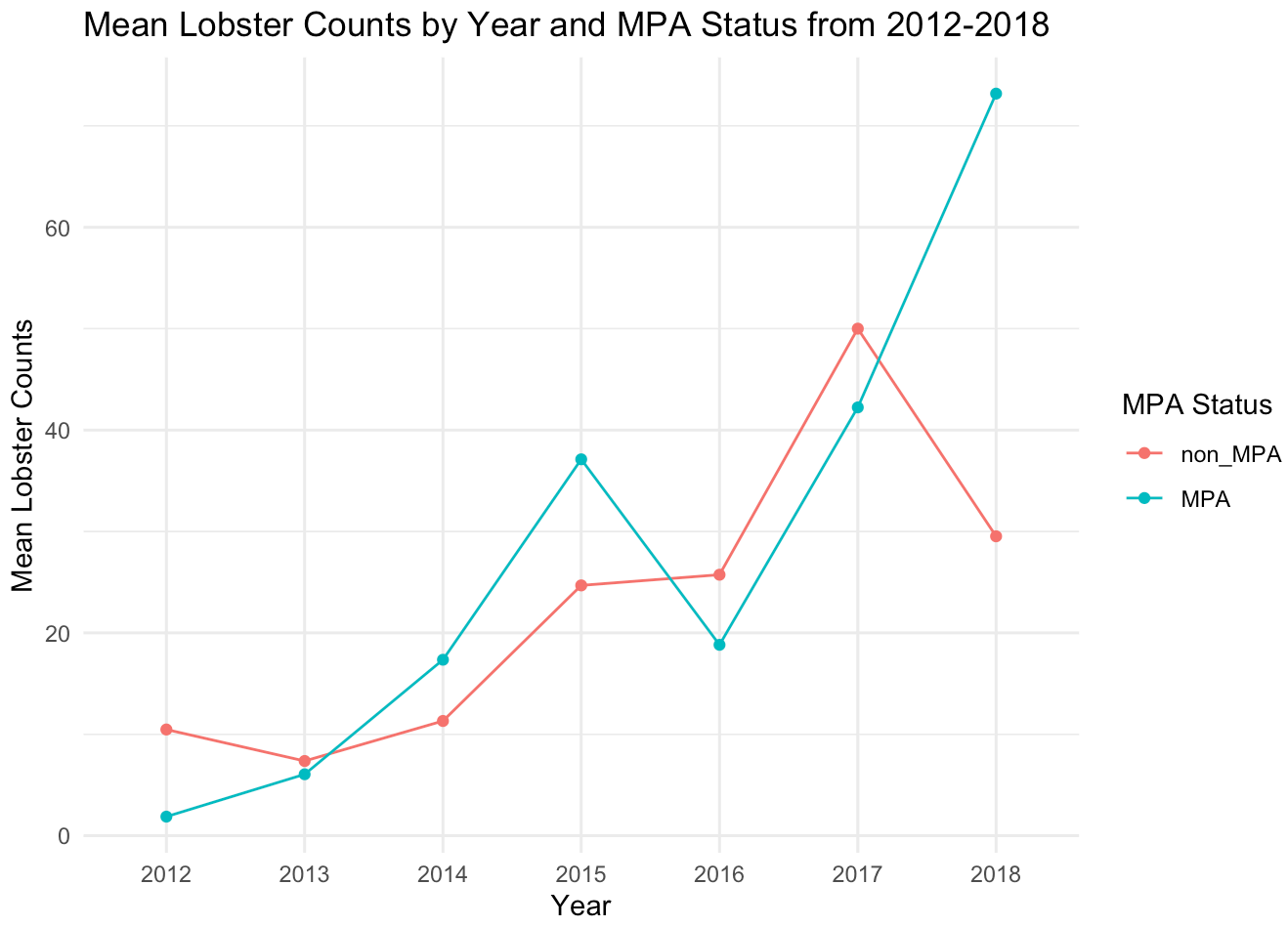

```
# HINT: Change `outcome.scale` to "response" to convert y-axis scale to counts
```

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
# Calculate the mean lobster count by year and site
plot_counts <- ff_counts %>%
    group_by(mpa, year) %>%
    summarize(mean_count = mean(counts, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(year = as.factor(year))

# Plot
plot_counts %>% ggplot(aes(x = year, y = mean_count, color = mpa, group = mpa)) +
    geom_point() +
    geom_line() +
    labs(title = "Mean Lobster Counts by Year and MPA Status from 2012-2018",
         x = "Year",
         y = "Mean Lobster Counts",
         color = "MPA Status") +
    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 (https://docs.google.com/document/d/1RIudsVcYhWGpqC-Uftk9UTz3PIq6stVyEpT44EPNgpE/edit?usp=sharing)) **I think spillover is possible in our case study. Because the sites are pretty close to eachother, it seems feasible that the lobsters would be able to move between sites. If this occurred, then the MPA sites could be unintentionally affecting the control groups (non-MPAs).**

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 key assumptions that were made for the study. It also messes up the counterfactual (outcome if the treated group wasn't treated) because if the control group gets affected by the treatment group, then the control group is no longer a true counterfactual. Overall, it causes problem for the validity of the results. c. How does spillover relate to impact in this research setting?

In our case, it can bias our treatment estimate or distort the differences between the treatment and control group. If spillover wasn't taken into account, we could come to an incorrect conclusion about the effectiveness of MPAs for lobster populations in this area.

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

SUTVA states that there is no interference between control and treatment group, and there is a clear treatment status. If spillover was occurring as discussed above, then this assumption would be violated. Because I don't know for sure that spillover did or did not occur, then for now, I can assume this assumption is reasonable

2. Excludability assumption

The excludability assumptions assumes that the status of the treatment is the only thing affecting the lobster populations. As stated at the beginning, our study does not have perfect counterfactuals, for there are possible environmental factors that affect population as well. In addition to this, if spillover was occurring, this would also violate this assumption. So, I'll say that this one is less reasonable.

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
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)