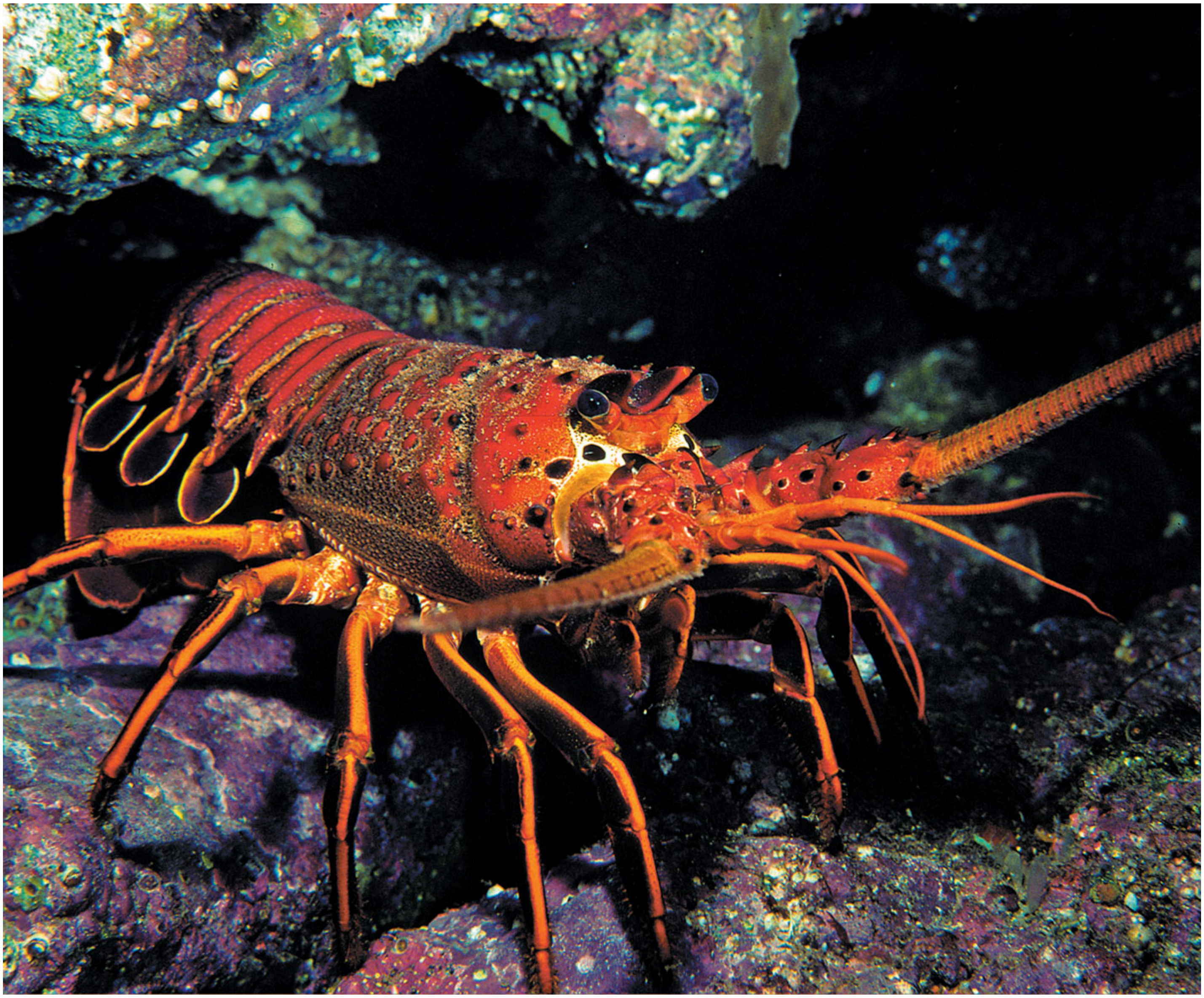
Assignment 1

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

Nathalie Bonnet

1/20/26



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 Nathalie Bonnet: ______________________________

```r
# Load libraries
library(tidyverse)
library(here)
library(janitor)
library(estimatr)
library(performance)
library(jtools)
library(gt)
library(gtsummary)
library(interactions)
library(ggridges)
library(ggbeeswarm)
```

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) Data accessed 11/17/2019.

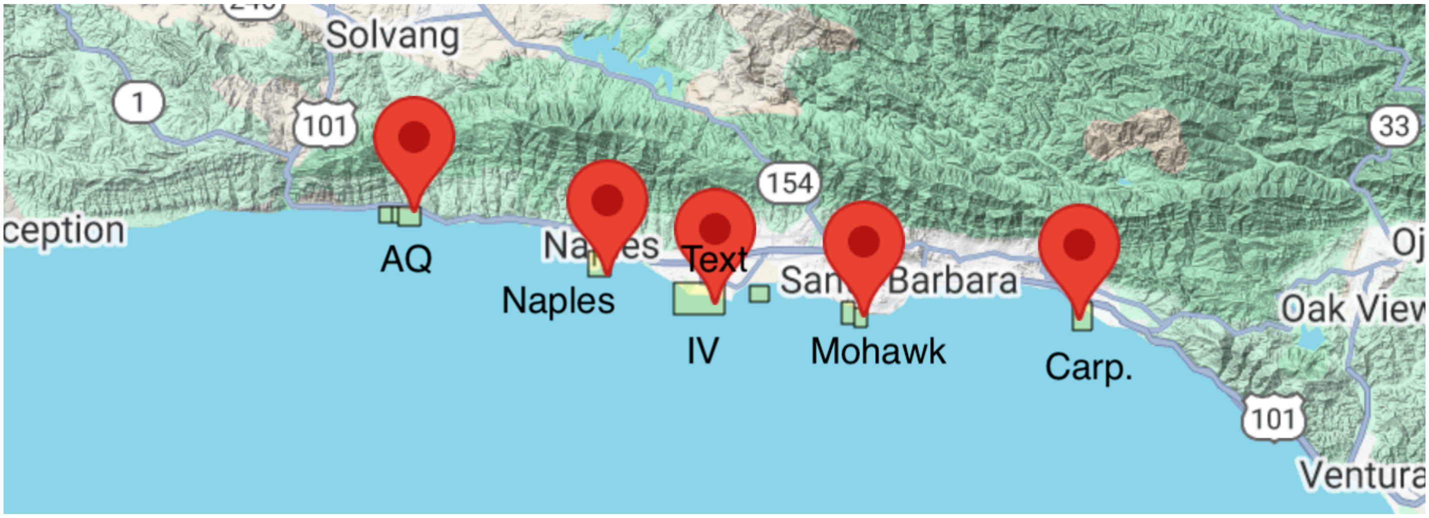
Introduction

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

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

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

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



Step 1: Anticipating potential sources of selection bias

a. Do the control sites (Arroyo Quemado, Carpenteria, and Mohawk) provide a strong counterfactual for our treatment sites (Naples, Isla Vista)? Write a paragraph making a case for why this comparison is ceteris paribus or whether selection bias is likely (be specific!). The counterfactual should provide an 'apples-to-apples' comparison, meaning the pre-treatment conditions should be the same. Looking at the site map, it seems that study sites differ in size and possibly oceanographic feature, which could introduce pre-treatment variation that results in selection bias. For example, the Isla Vista MPA is out on a point, which could potentially result in different tidal or depth dynamics. Also, the sites vary in proximity to relatively urban area, where some may have effects from more people using the area, pollution, etc. Based on these possibilities, it seems like the comparison may have selection bias.

Step 2: Read & wrangle data

a. Read in the raw data from the “data” folder named `spiny_abundance_sb_18.csv` . Name the data.frame `rawdata`

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

```
# Read in data and correct NAs and column names
rawdata <- read.csv(here("data", "spiny_abundance_sb_18.csv"), na.strings = c("NA", "-99
9", "-99999", "-9999")) %>% 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 column with site names and reorder
tidydata <- rawdata %>%
  mutate(
    reef = factor(
      case_when(
        site == "AQUE" ~ "Arroyo Quemado",
        site == "CARP"  ~ "Carpenteria",
        site == "MOHK"  ~ "Mohawk",
        site == "IVEE" ~ "Isla Vista"   ,
        site == "NAPL"   ~ "Naples"),
      levels = 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`

```
# Create summary counts by site, year, and transect
spiny_counts <- tidydata %>%
    group_by(site, year, transect) %>%
    summarize(counts = sum(count),
                     mean_size = mean(size_mm, na.rm = TRUE)) %>%
    # Encode which sites are MPAs
    mutate(mpa = case_when(site == "IVEE"~"MPA",
                                site == "NAPL"~"MPA",
                                site == "AQUE"~"non_MPA",
                                site == "CARP"~"non_MPA",
                                site == "MOHK"~"non_MPA"),
                     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/)

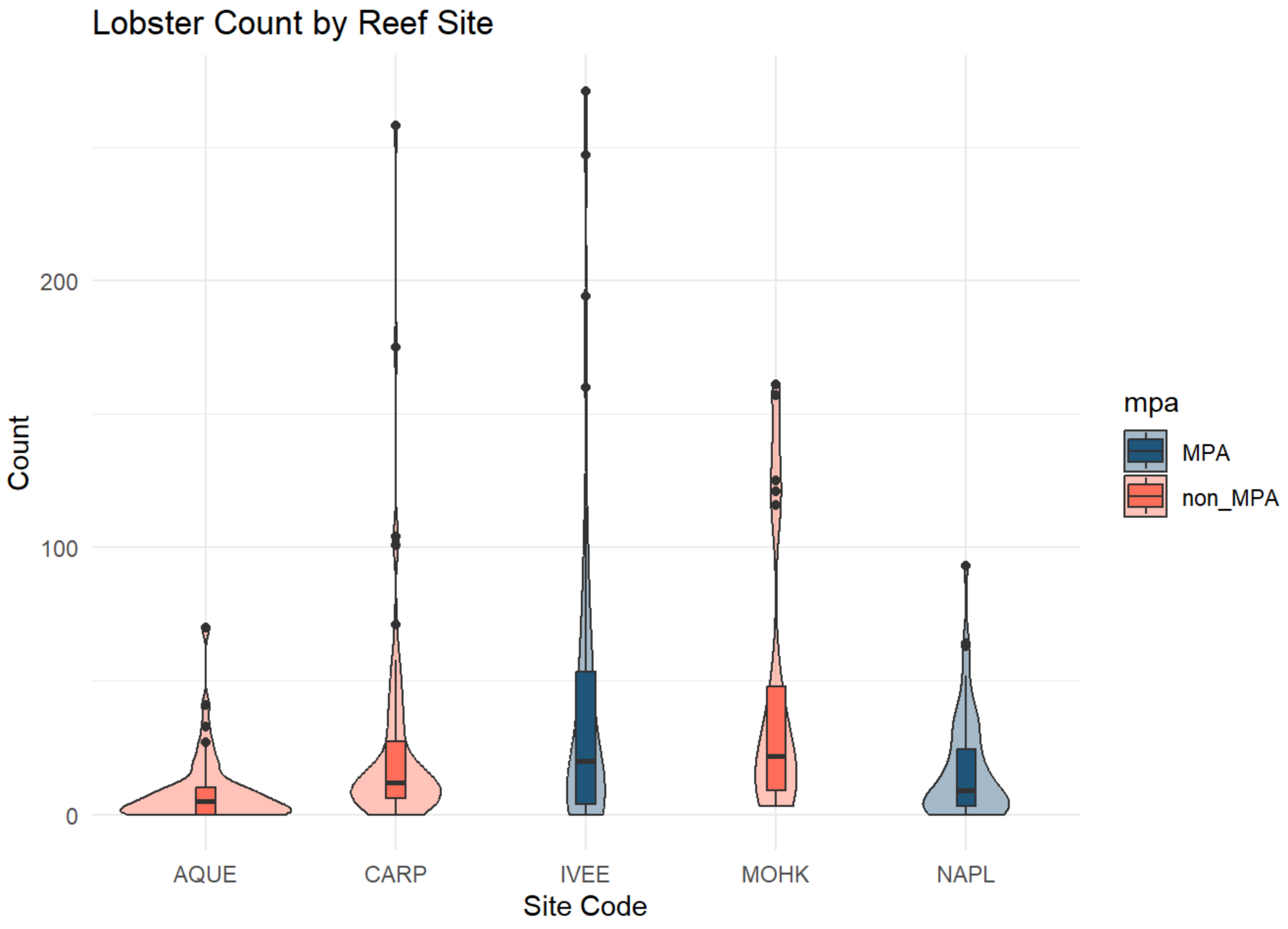
Create plots displaying the distribution of lobster **counts**:

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

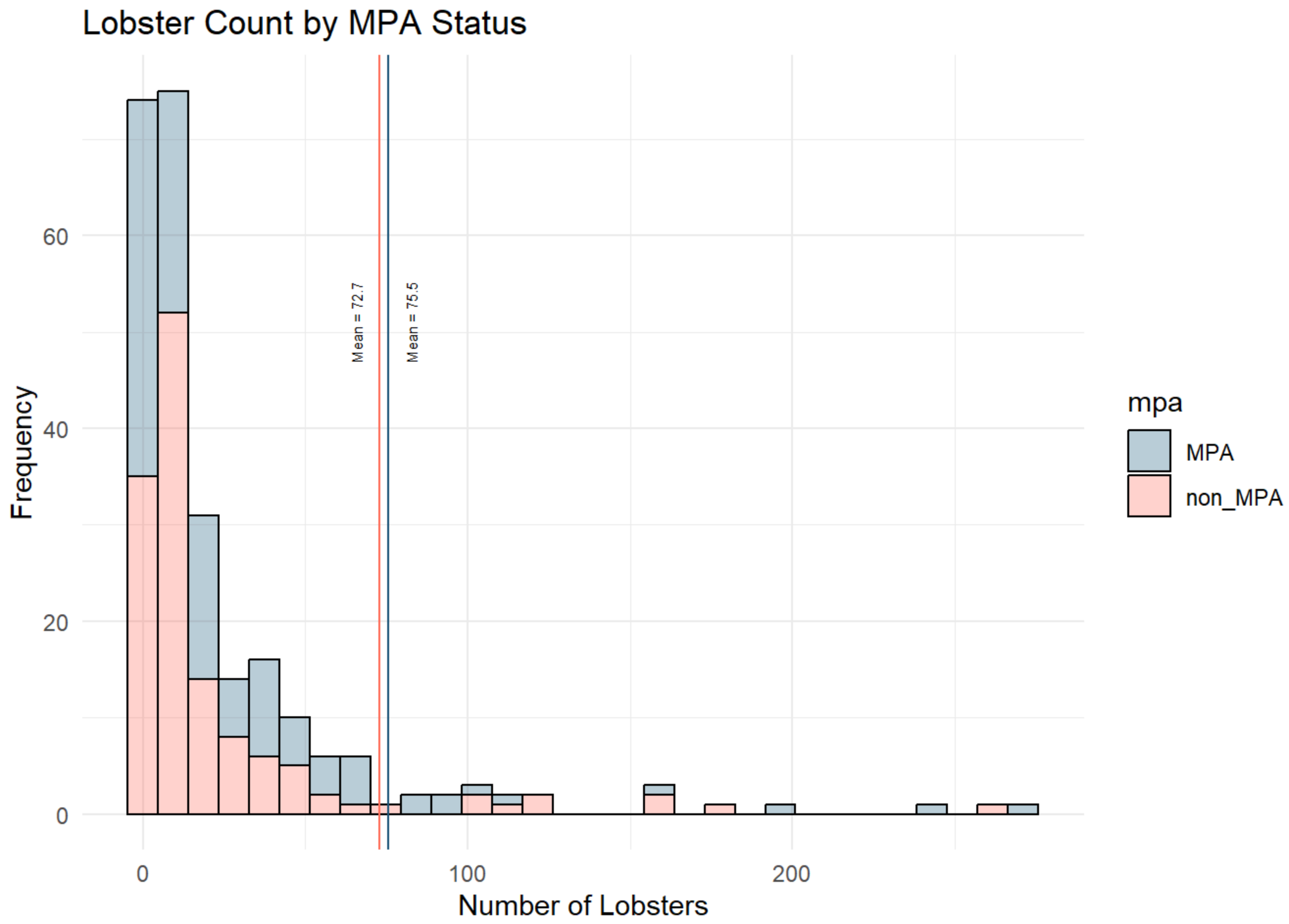
Create a plot of lobster **size** :

4. You choose the grouping variable(s)!

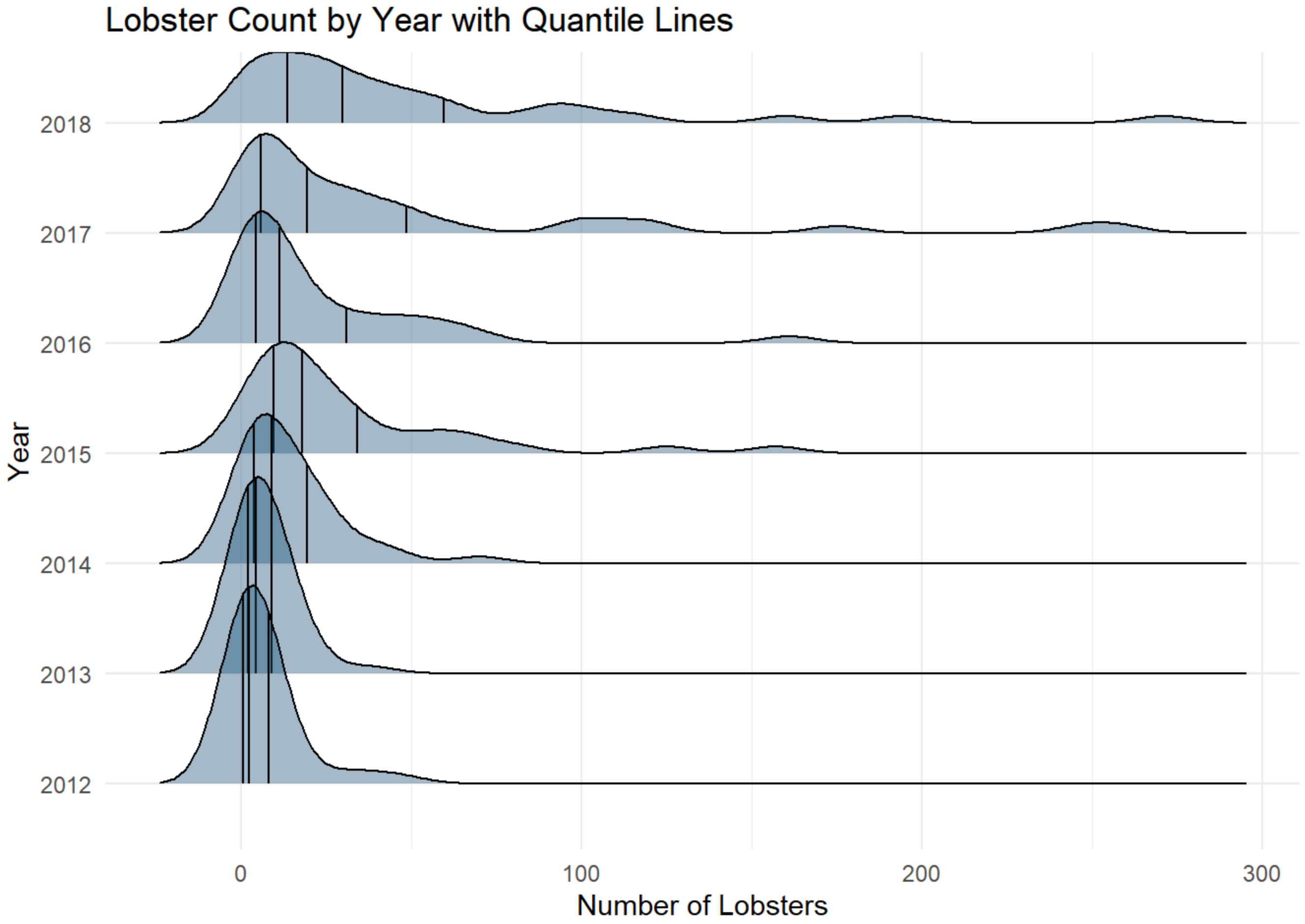
```
# Plot 1: reef site counts
spiny_counts %>%
    ggplot(aes(x = site, y = counts, fill = mpa)) +
    # Add violin plot
    geom_violin(alpha = 0.4) +
    # Layer boxplot
    geom_boxplot(width = 0.1) +
    # Fill custom colors
    scale_fill_manual(values = c("#22577A", "#FF715B")) +
    theme_minimal() +
    labs(title = "Lobster Count by Reef Site", x = "Site Code", y = "Count")
```



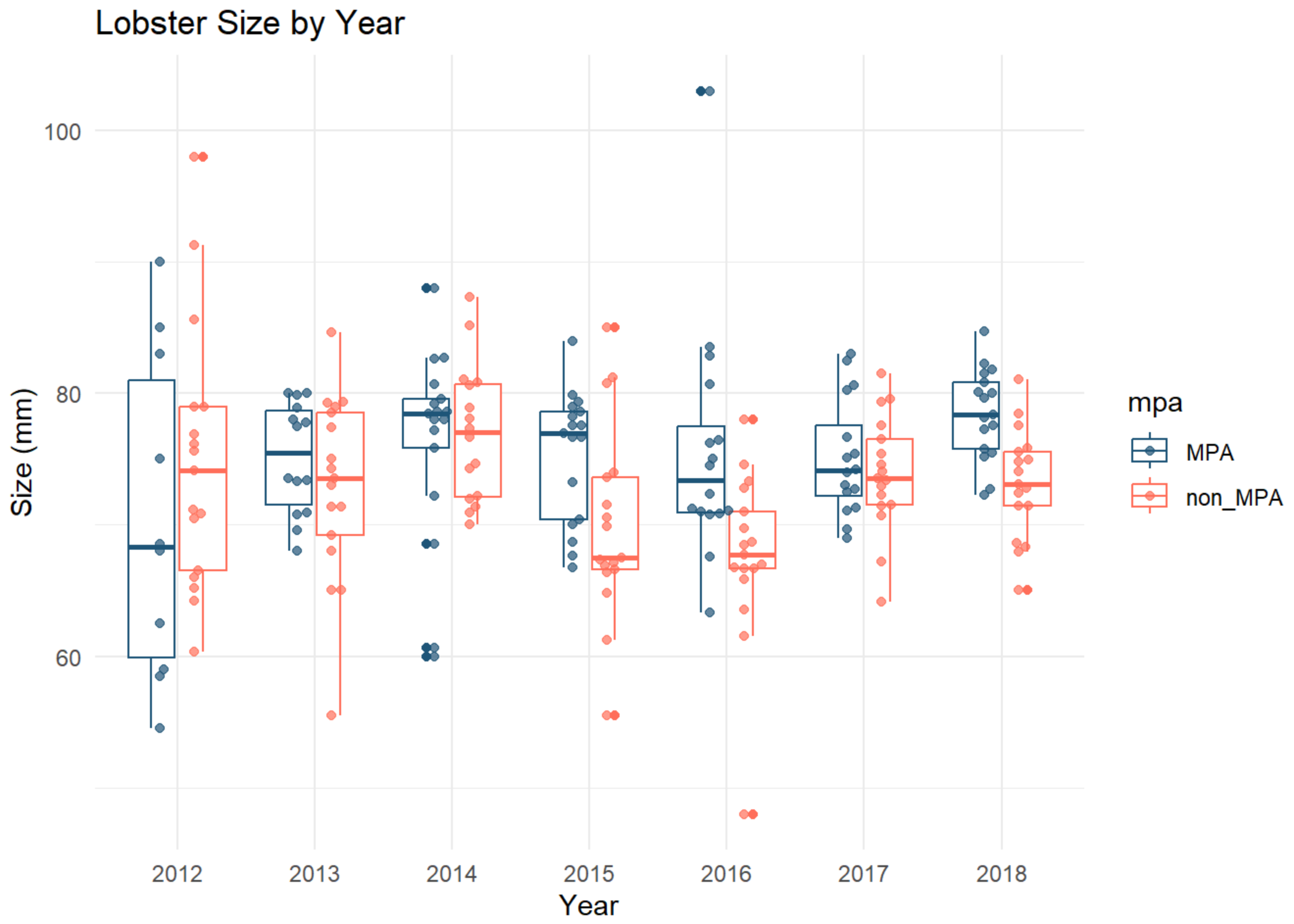
```
#Find mean lobster size for mpas vs. non-mpa
mean_mpa <- spiny_counts %>%
    group_by(mpa) %>%
    summarise(mean = mean(mean_size, na.rm = TRUE))

# Plot 2: MPA status
spiny_counts %>%
    ggplot(aes(x = counts, fill = mpa)) +
    geom_histogram(color = "black", alpha = 0.3) +
    # Add vertical lines for means
    geom_vline(xintercept = mean_mpa$mean[[1]], color = "#22577A") +
    geom_vline(xintercept = mean_mpa$mean[[2]], color = "#FF715B") +
    # Add MPA annotation
    annotate(
        "text",
        x = (mean_mpa$mean[[1]] + 7),
        y = 51,
        label = paste0("Mean = ", round(mean_mpa$mean[[1]], 1)),
        angle = 90,
        size = 2
    ) +
    # Add non-MPA annotation
    annotate(
        "text",
        x = (mean_mpa$mean[[2]] - 7),
        y = 51,
        label = paste0("Mean = ", round(mean_mpa$mean[[2]], 1)),
        angle = 90,
        size = 2
    ) +
    scale_fill_manual(values = c("#22577A", "#FF715B")) +
    theme_minimal() +
    labs(title = "Lobster Count by MPA Status", x = "Number of Lobsters", y = "Frequenc
y")
```



```
# Plot 3: year
spiny_counts %>%
    ggplot(aes(x = counts, y = factor(year))) +
    # Add ridge plot with quantile lines
    geom_density_ridges(
        fill = "#22577A",
        alpha = 0.4,
        quantile_lines = TRUE,
        quantiles = c(0.25, 0.5, 0.75),
        panel_scaling =  TRUE
    ) +
    theme_minimal() +
    labs(title = "Lobster Count by Year with Quantile Lines", x = "Number of Lobsters", y
= "Year")
```



```r
# Plot 4: size
spiny_counts %>%
    ggplot(aes(x = factor(year), y = mean_size, color = mpa)) +
    # Add boxplot
    geom_boxplot() +
    # Layer beeswarm plot to visualize distribution
    geom_beeswarm(stat = "identity",
                  alpha = 0.7,
                  dodge.width = 0.5) +
    scale_color_manual(values = c("#22577A", "#FF715B")) +
    theme_minimal() +
    labs(title = "Lobster Size by Year", x = "Year", y = "Size (mm)")
```



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)

```
# Create gtsummary for table output
lobster_summary <- spiny_counts %>%
    drop_na(mean_size) %>%
    gtsummary::tbl_summary(
        # Group by MPA
        by = mpa,
        # Include columns in table
        include = c(mean_size, counts),
        # Summary stats
        statistic = list(all_continuous() ~ "{mean} ({sd})")
    ) %>%
    # Add p-val
    add_p()

# Call gt table and add details
lobster_summary %>%
    as_gt() %>%
    # Add a Title and Subtitle
    tab_header(title = "Santa Barbara Lobster Size and Count", subtitle = "Comparing MPA
vs. non MPA Areas") %>%
    # Add a Spanner to group the data columns
    tab_spanner(label = "Treatment", columns = c(stat_1, stat_2)) %>%
    # Change column labels
    cols_label(
        label = "Variable",
        p.value = "P-Value",
        stat_1 = "MPA",
        stat_2 = "non MPA"
    )
```

Santa Barbara Lobster Size and Count

Comparing MPA vs. non MPA Areas

	Treatment		
Variable	MPA[1]	non MPA[1]	P-Value[2]
mean_size	76 (7)	73 (7)	<0.001
counts	31 (45)	26 (40)	0.3

[1] Mean (SD)

[2] Wilcoxon rank sum test

According to the summary table, there is a significant difference in lobster size between MPA and non-MPA sites, with the mean for MPA being 76mm, and non-MPA lobsters being an average 73mm long. The table does not immediately reflect a signficant difference in mean counts however.

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)

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

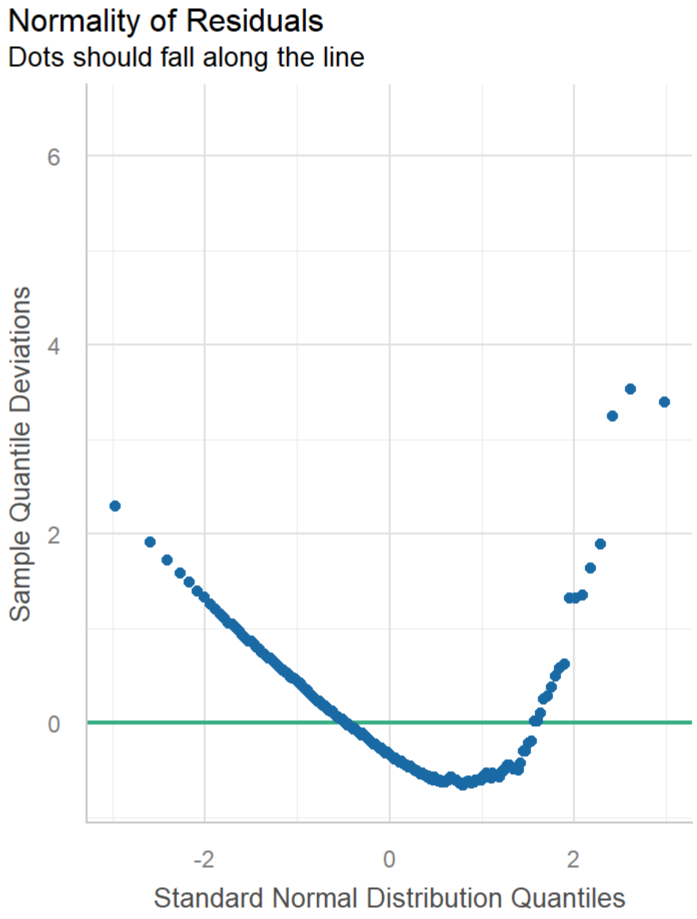
Standard errors: OLS

Based on the model's intercept estimate, in non-MPA sites we would expect to count an average of 23 lobsters per transect. The model estimates an average of 5 more lobsters per transect in MPA sites than in non-MPA sites.

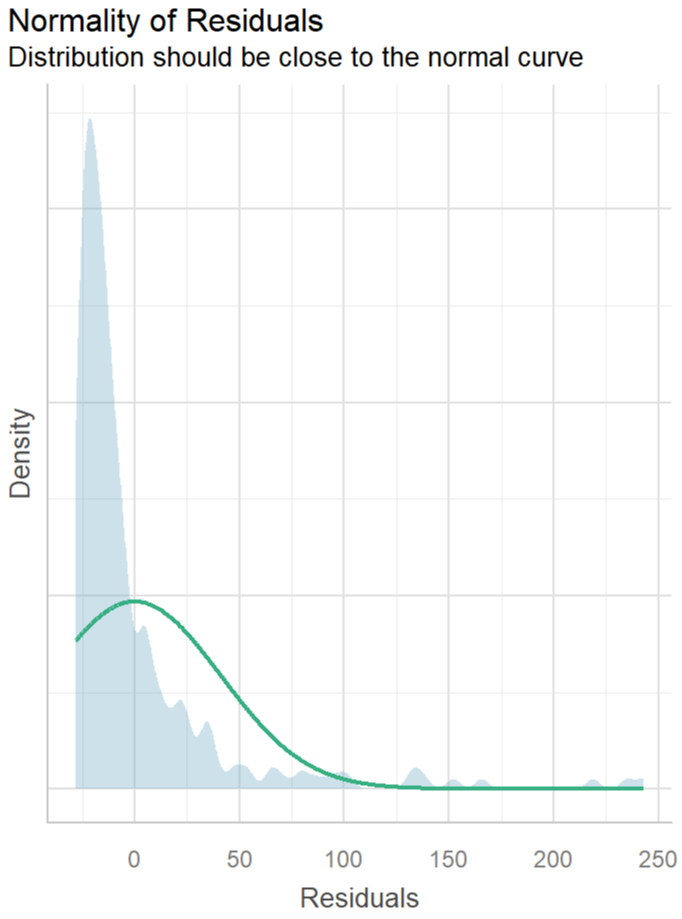
c. Check the model assumptions using the `check_model` function from the `performance` package

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

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

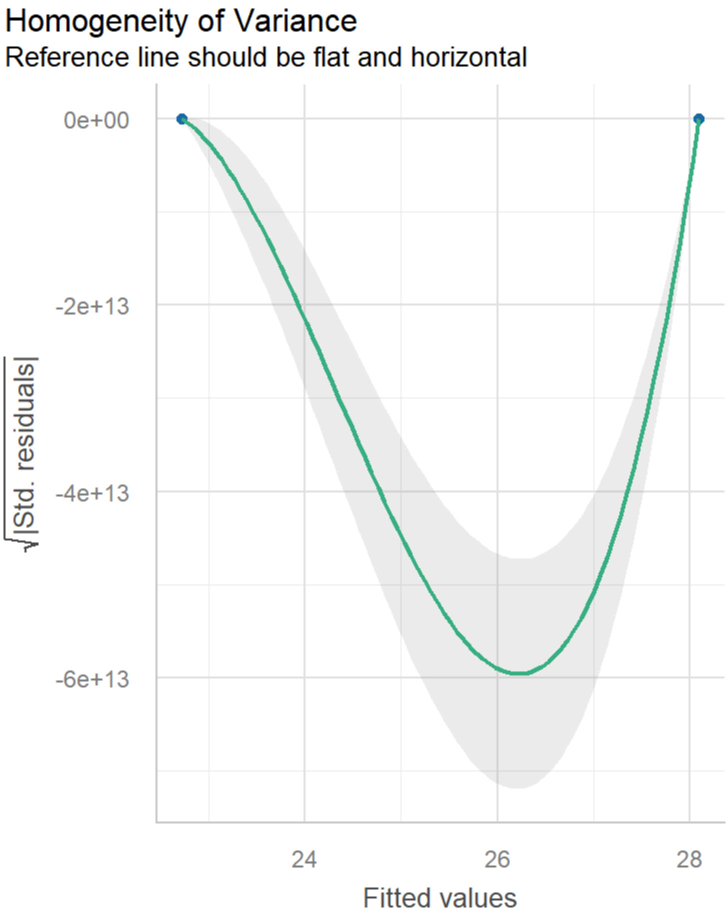


```
check_model(m1_ols, check = "normality")
```

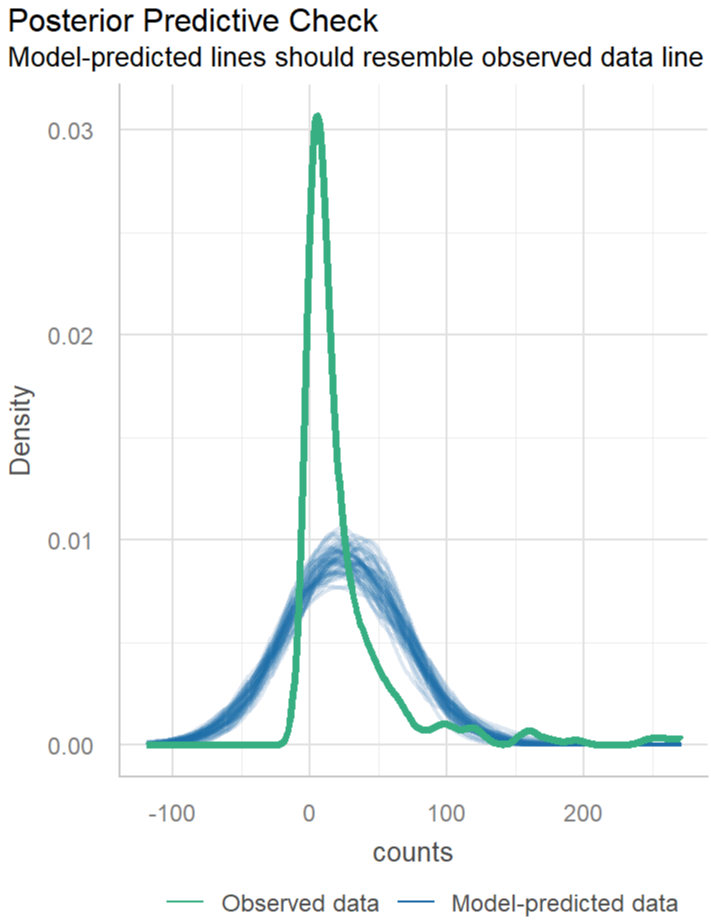

Normality of Residuals

Distribution should be close to the normal curve

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



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



OLS models assume that the variance across residuals is the same. Here, the residuals do not show homogeneity as we would expect for data that is well fitted with an OLS model. This suggests to us that a linear model is not best suited for use with these data, and we should select another based on the trends in the 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.

The model's intercept term predicts an average of 23 lobsters per transect in non-MPA site transects (exp(3.12)). The treatment coefficient predicts a significant increase in average lobster count of 23% in MPA sites compared to non-MPA sites.

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

Dispersion explains spread of data (variance), and over dispersion describes when the data has more variance than is predicted in a given statistical model. Over dispersion is important to check for when choosing a model because if selected wrong it will not represent the data well.

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 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")`

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

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

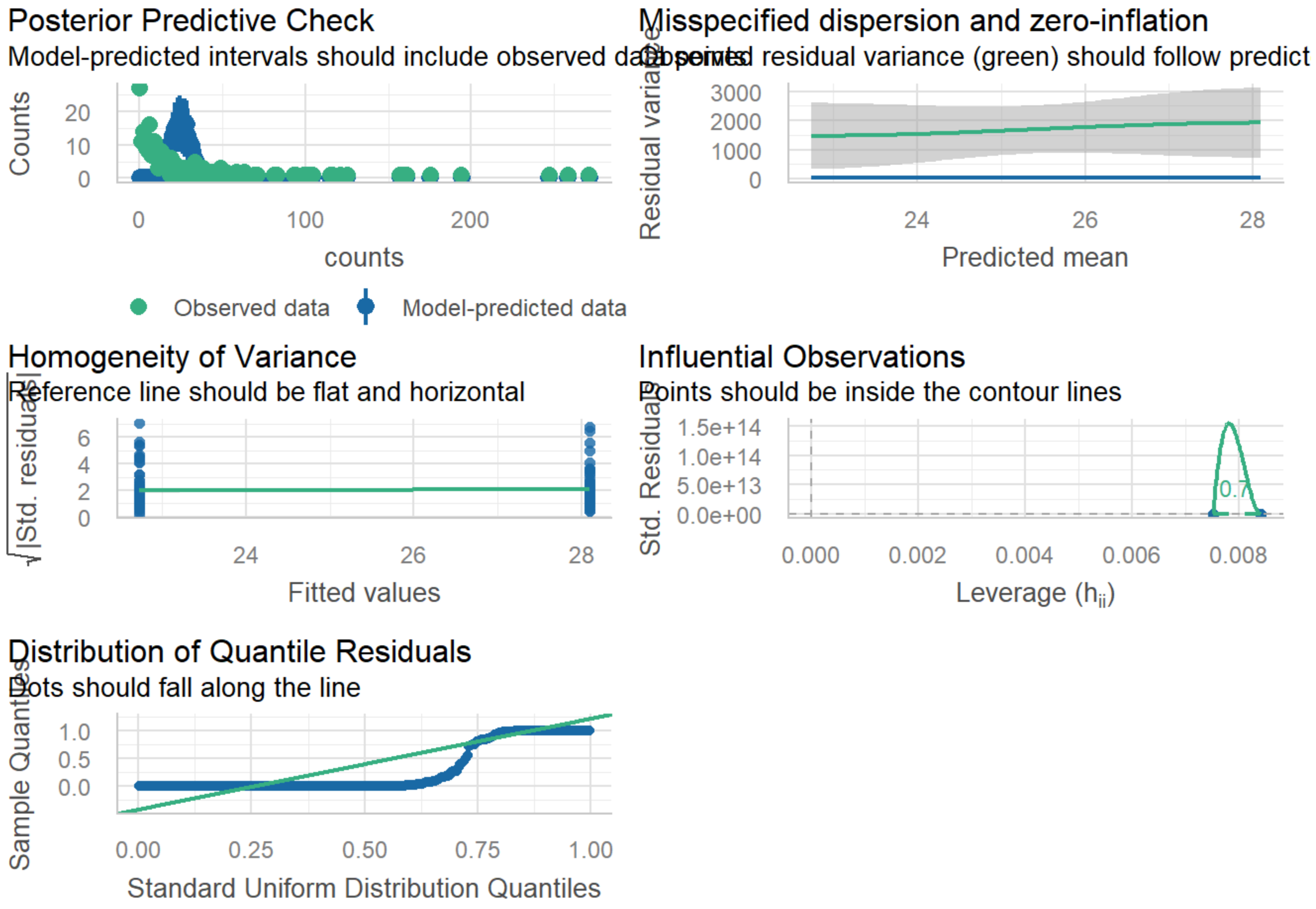
Rounding to the nearest lobster, both models predict the same average number of lobsters per transect in non-MPA sites, but converting coefficient for the effect of treatment, the treatment effect is slightly less in the poisson model, with a predicted number of 4.76 compared to the 5.36 predicted by the ols model.

e. Check the model assumptions. Explain results.

The model overall does not appear to fit the data well because the predicted variance does not follow the observed residuals in any of the model fit checks run. This suggests that the data is over dispersed for the poisson model, and needs to be fit with a different one to account for the larger variance in the data.

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

```
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
```

According to the over dispersion check, the dispersion ratio is significantly larger than 67, which tells us that the data is over dispersed. The zero inflation check predicts no zeros, where there are 27 observed, which tells us that the data is in fact zero inflated (more zeros than the model leads us to expect).

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.

The negative binomial model accounts better for count data containing many zeros.

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

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

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

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

The result of the negative binomial model matches the predictions of the poisson model, where there are an average of 23 lobsters per transect in non-MPA area, and a 23% increase in average lobster counts in MPA transects (average of 5). However, the p value for the treatment coefficient in this model does not suggest that this is a significant increase where the poisson does. Checking the over dispersion, posterior predictive check, and residuals, the negative binomial model much more accurately predicts zeros and dispersion in the data than the poisson.

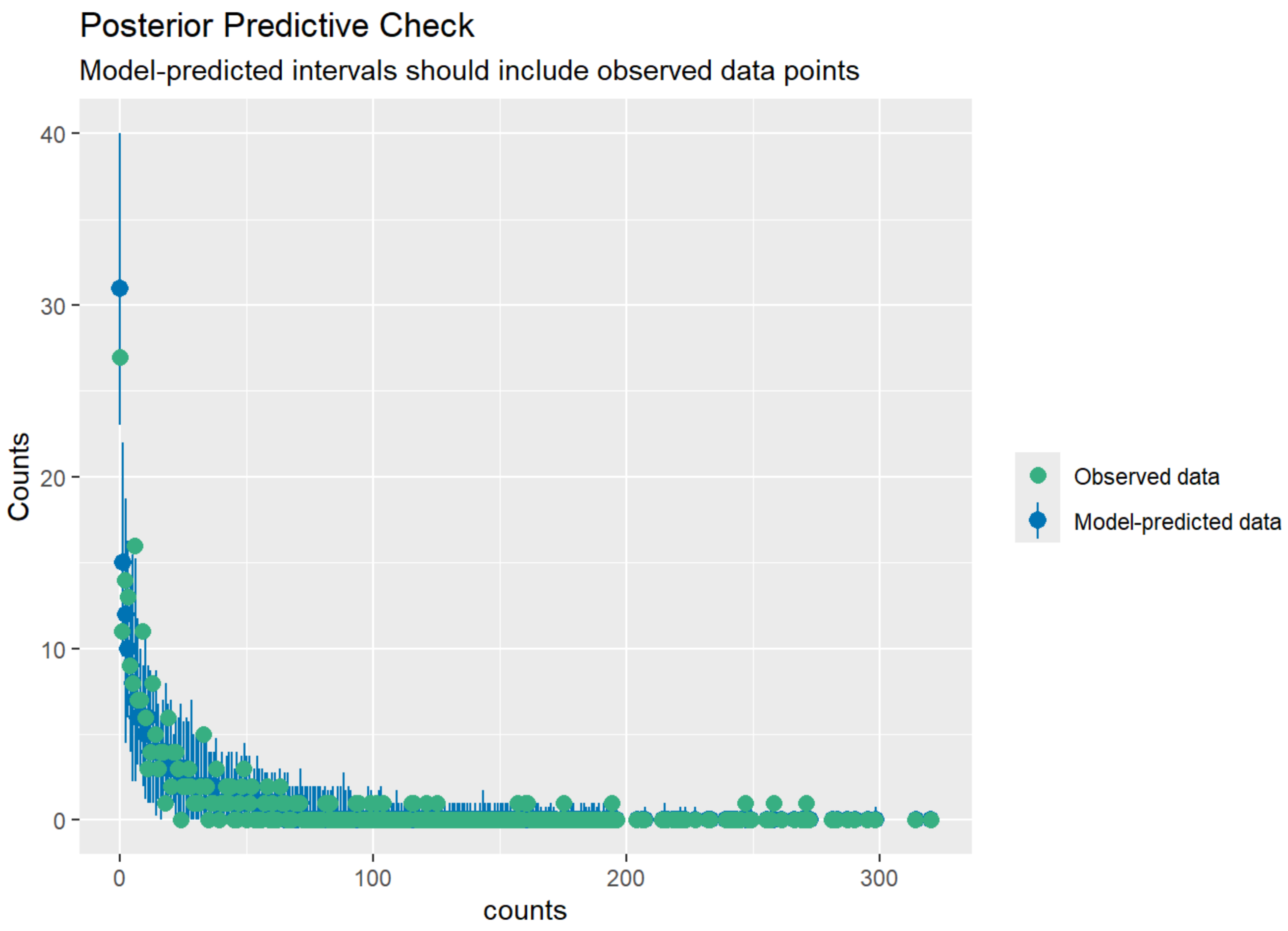
```
check_overdispersion(m3_nb)
```

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

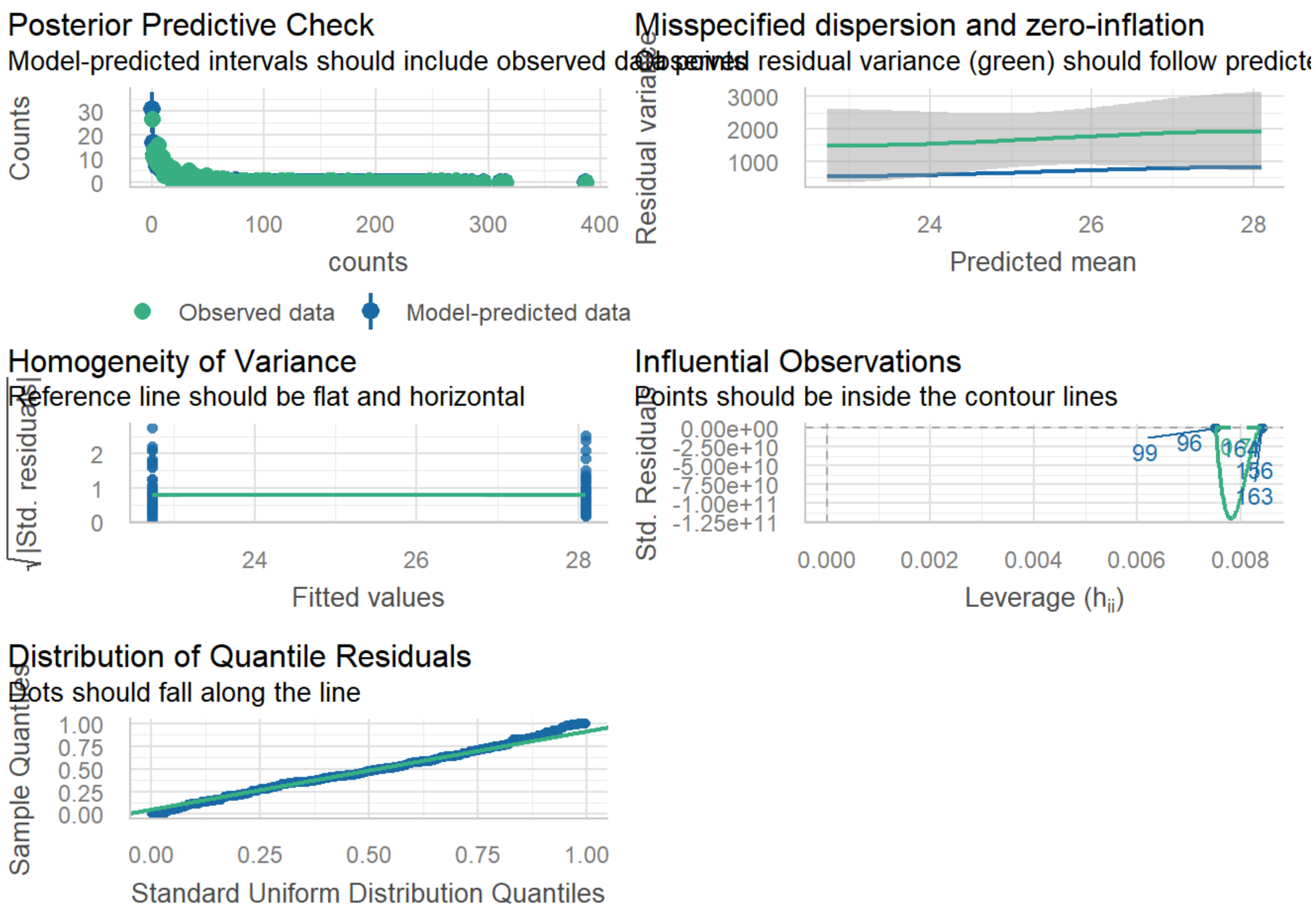
```
check_zeroinflation(m3_nb)
```

```
## # 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.

```
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.

Each model predicts between 22-23 lobsters on average for each transect in non-MPAs, and approximately 5 more lobsters per transect in treatment transects. However, the only model that shows a significant treatment effect is the poisson model. Thus, the treatment effect is not robust across all models. Looking at the residuals and predictions versus observed data, the negative binomial model did not show evidence of over dispersion or zero inflation, and the observed data fit the predicted data. Therefore, the negative binomial is the best fit model here, despite the treatment not showing significant effect.

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)

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

```
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

Standard errors: MLE

	Est.	S.E.	z val.	p
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

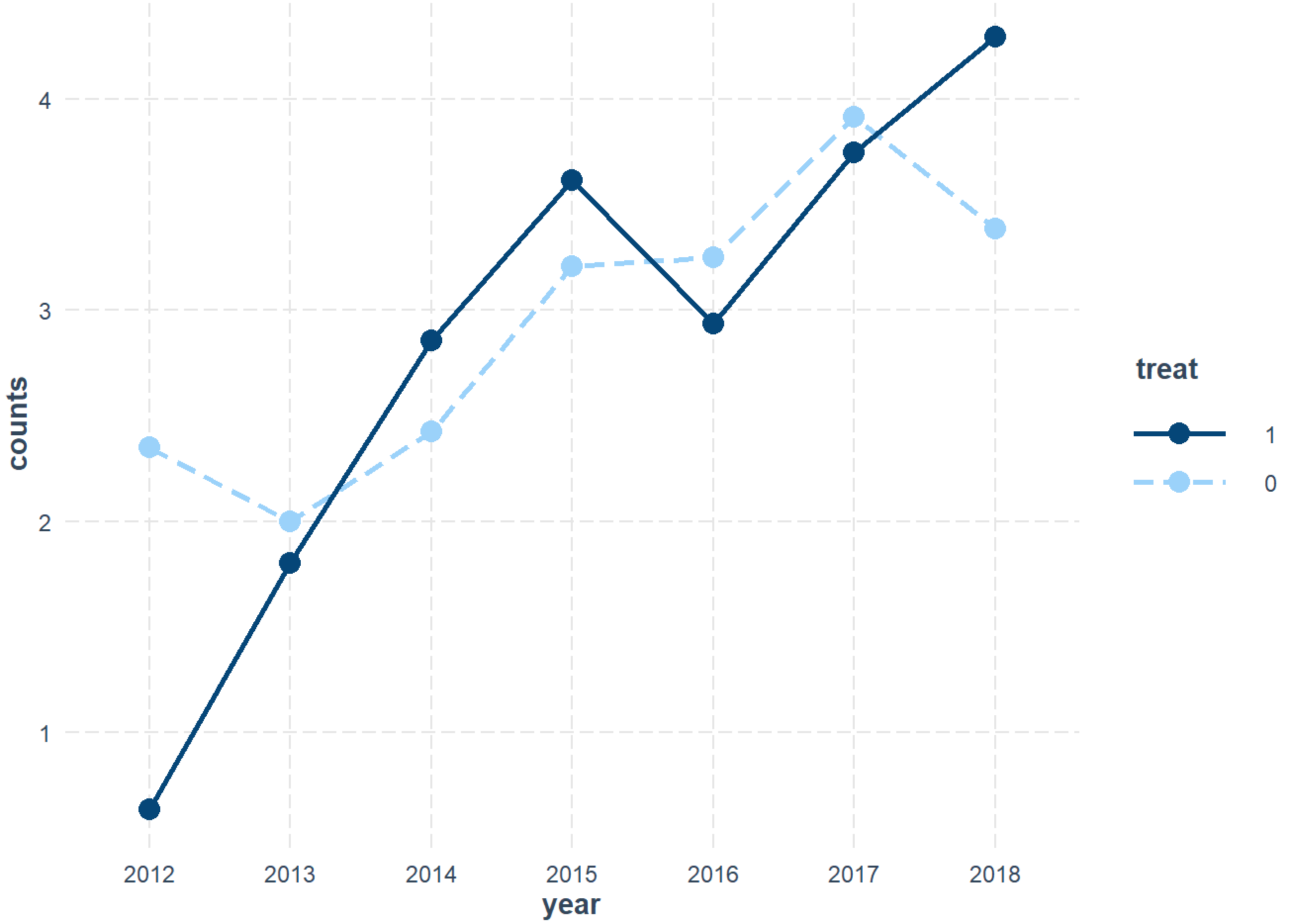
Standard errors: MLE

The main treatment effect coefficient is negative because the reference level is non-MPA sites in 2012, indicating that MPA sites had 82% fewer lobsters than non-MPA sites in 2012. The treatment effect increases over time, seen as the sign flips after 2013. The interaction terms between treatment status and year were positive and statistically significant, indicating that lobster abundance increased more in MPA sites than non MPA sites.

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

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

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



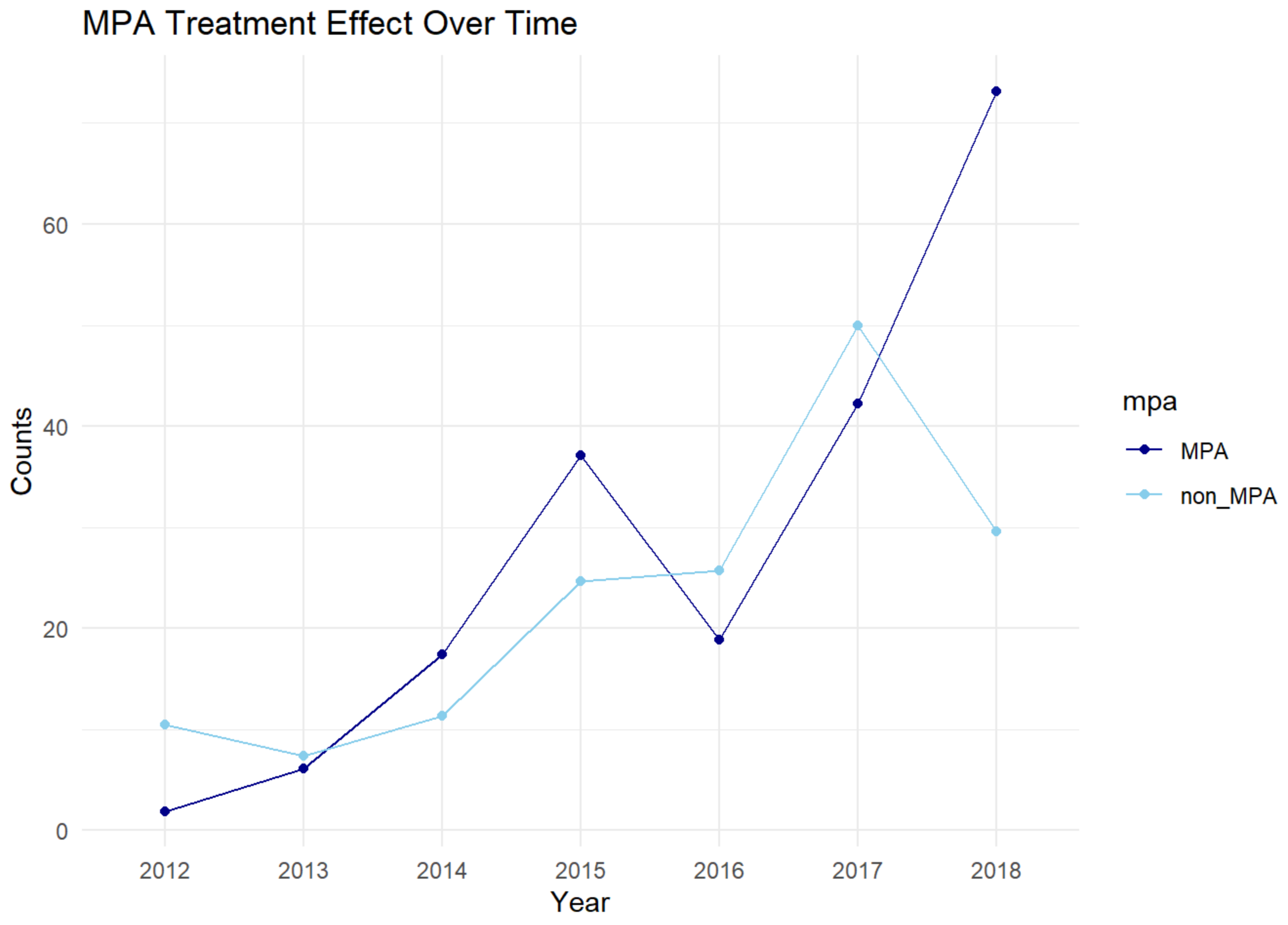
```
# HINT: Change `outcome.scale` to "response" to convert y-axis scale to counts
```

This interaction plot confirms the interpretation from the model above, where the treatment plots (MPA) initially have fewer lobsters than the non-MPA sites, but the treatment effect increases the expected lobster count over time, flipping the MPA's initial predicted effect.

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

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

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

plot_counts <- spiny_counts %>%
    group_by(year, mpa) %>%
    summarise(mean_count = mean(counts))

ggplot(data = plot_counts, aes(
    x = factor(year),
    y = mean_count,
    color = mpa,
    group = mpa
)) +
    geom_point() +
    geom_line() +
    scale_color_manual(values = c("darkblue", "skyblue"))+
    labs(title = "MPA Treatment Effect Over Time",
         x = "Year",
         y = "Counts") +
    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))

Spillover effects are fairly likely in this context because if lobsters are being protected in MPAs that are geographically close to non-MPA areas, there may be an overall increase in lobster abundance even if not all areas are technically protected.

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

Spillover is a problem for causal effect identification because spillover from treatment units could obscure differences caused by the treatment itself, making it difficult to determine the actual treatment effect.

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

Spillover in the MPA research context could result in the determination that implementing MPAs does not result in a significant increase in lobster abundance, which has the potential to change policy decisions, where it is possible that MPAs are simply conferring benefits for lobster abundance inside and outside MPAs.

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 implies that treatment affects each treated observation equally, or that the outcome of a treatment unit is not linked to the treatment status of other units.

In the context of a spatial study, I think it would be difficult to say for certain that treatment is applied directly and consistently. Even if regulations are enforced equally, there is likely environmental variation that could result in differential treatment effects, such as higher lobster recruitment in different reefs based on various factors.

```
2) Excludability assumption
```

Excludability requires that intervention influences the outcome only through the mechanism explained in a study/model. In this case we would be assuming that fishing effort is excluded as assumed from the MPA, so fishermen are not changing behaviors that could influence the treatment effect, and illegal lobster catches are not occurring.

```
3) Exogeneity assumption
```

Exogeneity assumes that the treatment variable is not related to unobserved variables that would otherwise affect the outcome of a treatment, essentially that model variables are not linked to the error in the model. In this case, this would assume that the effectiveness of MPA treatment is not related to better habitat quality in treatment areas, pre-existing lobster population densities, or change in fishing behavior. This seems like a potential issue in the study as MPAs are often placed in previously assessed suitable habitat. There is also the likelihood that fishing practices would be displaced outside the MPAs, potentially increasing fishing pressure on outside lobster populations.

EXTRA CREDIT

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

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