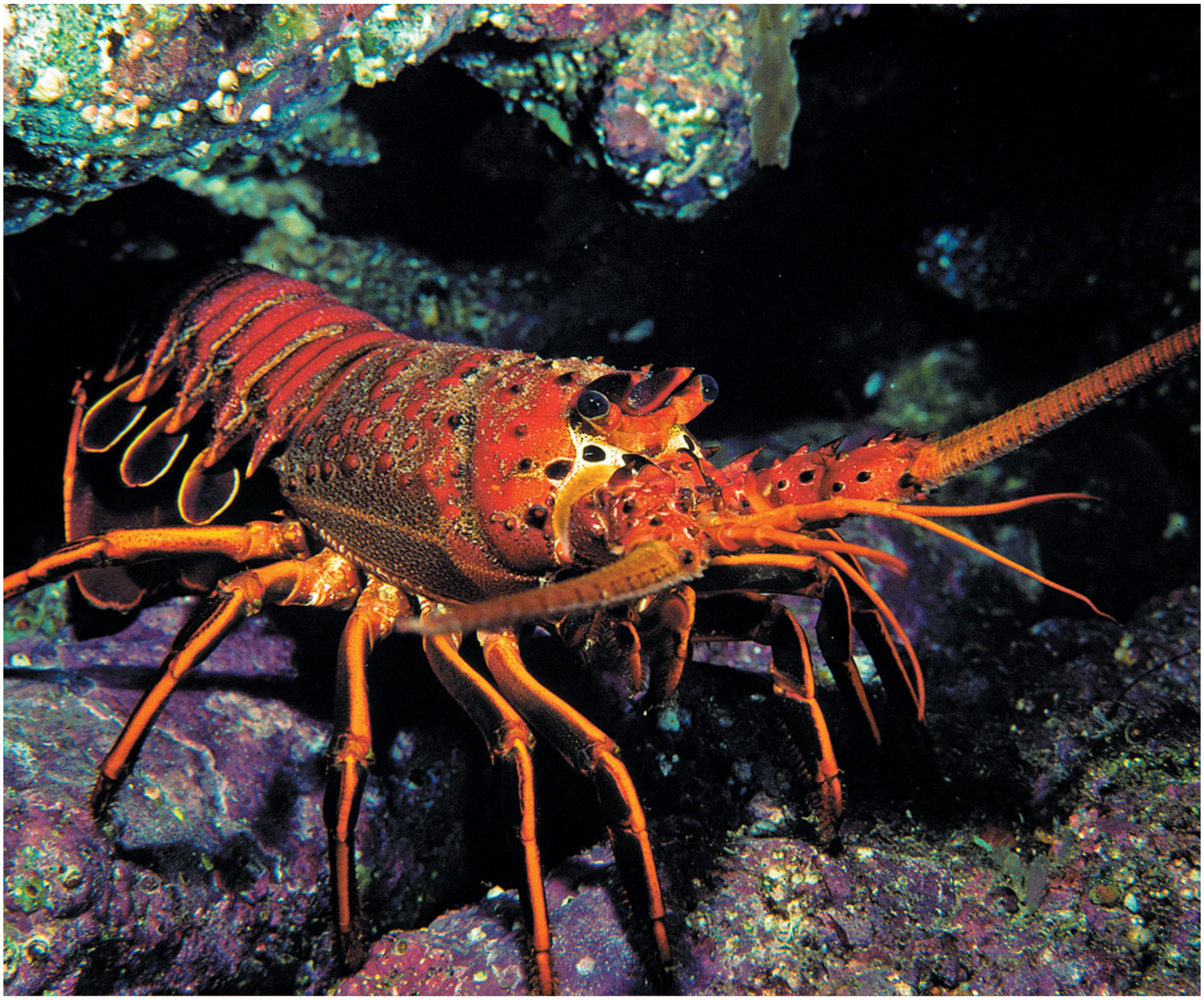
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

Haylee Oyler

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 : Haylee Oyler

With Emma Bea Mitchell, Tom Gibbens-Matsuyama, Ian Morris-Sibaja

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

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.

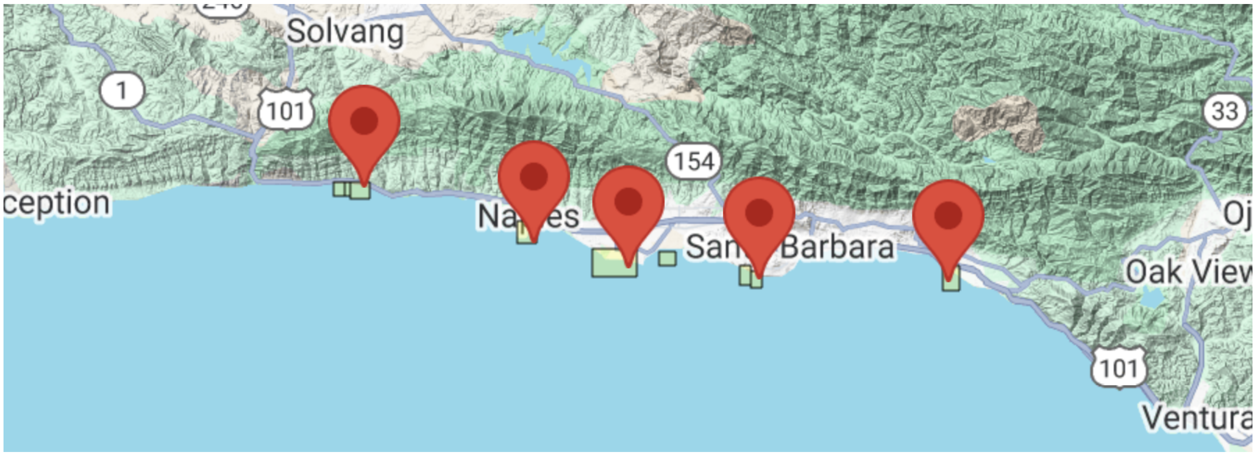
Introduction

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!).

The control sites do not provide a strong counterfactual and the comparison is *not* centris paribus. Even though the sites are relatively close together, there will still be variation in substrate, water temperature, habitat, etc. that means the control and treatment sites are not 100% identical. This means selection bias is likely because we are picking control sites that are introducing variables that differ from our treatment. That being said, it is still a very close comparison, all things considered.

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"`)
# Read in data and clean names and NAs
rawdata <- read_csv(here("data", "spiny_abundance_sb_18.csv"), na = "-99999") %>%
    clean_names()
```

c. Create a new `df` named `tidydata`. 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"`

```
# Refactor our sites into a clean data frame
tidydata <- rawdata |>
    mutate(reef = factor(site, order = TRUE,
                         levels = c("AQUE",
                                    "CARP",
                                    "MOHK",
                                    "IVEE",
                                    "NAPL"),
                         labels = c("Arroyo Quemado",
                                    "Carpenteria",
                                    "Mohawk",
                                    "Isla Vista",
                                    "Naples")))
```

Create new `df` named `spiny_counts`

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

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

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

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

# Add a treatment variable by MPA site and a mean size variable
spiny_counts <- tidydata %>%
    group_by(site, year, transect) %>%
    summarize(counts = as.integer(sum(count, na.rm = TRUE)),
              mean_size = mean(size_mm, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(mpa = case_when(site %in% c("IVEE", "NAPL") ~ "MPA",
                           site %in% c("CARP", "MOHK", "AQUE") ~ "non_MPA"),
           treat = case_when(mpa == "MPA" ~ 1,
                             mpa == "non_MPA" ~ 0)) %>%
    mutate(across(where(is.numeric), ~(ifelse(is.na(.), NA_real_, (.)))))


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

NAN is not a number NA real is still a numeric missing value.

> 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**:

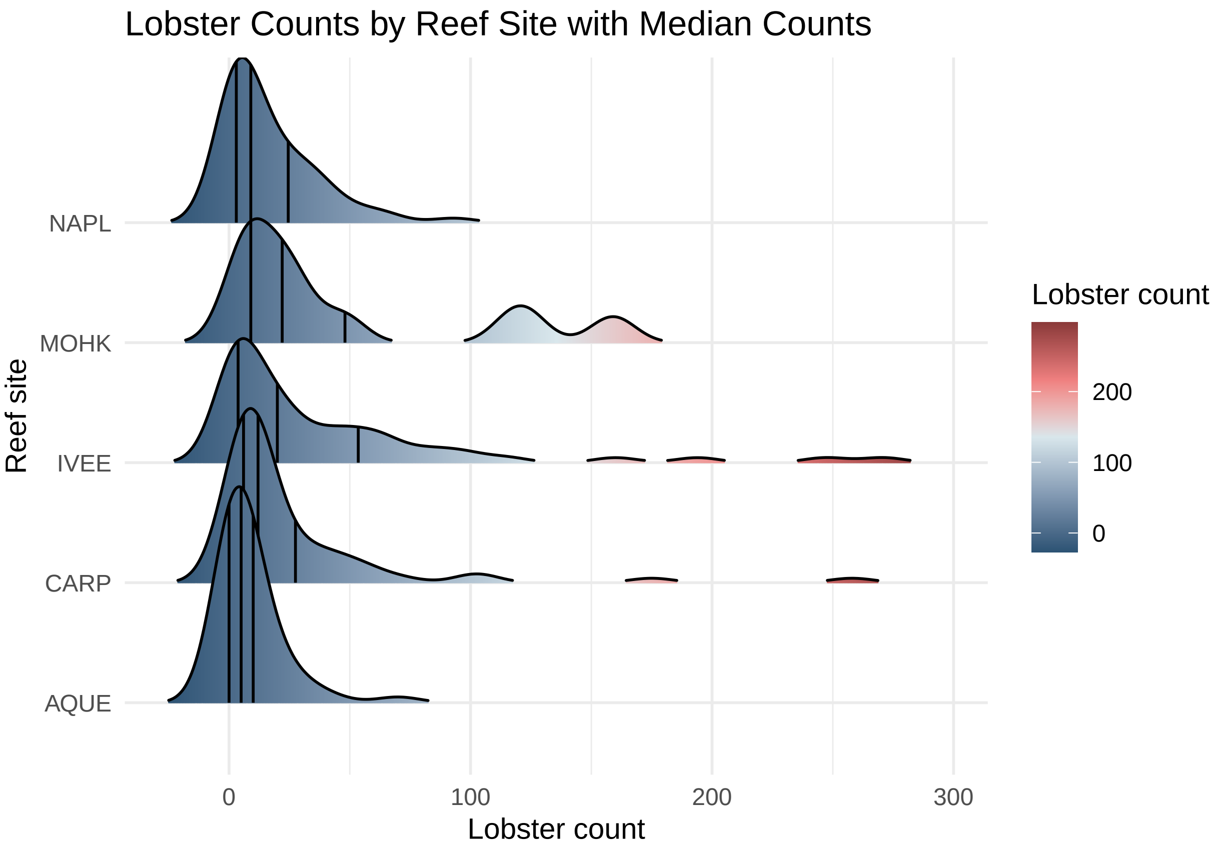
1) grouped by reef site

2) grouped by MPA status
3) grouped by year

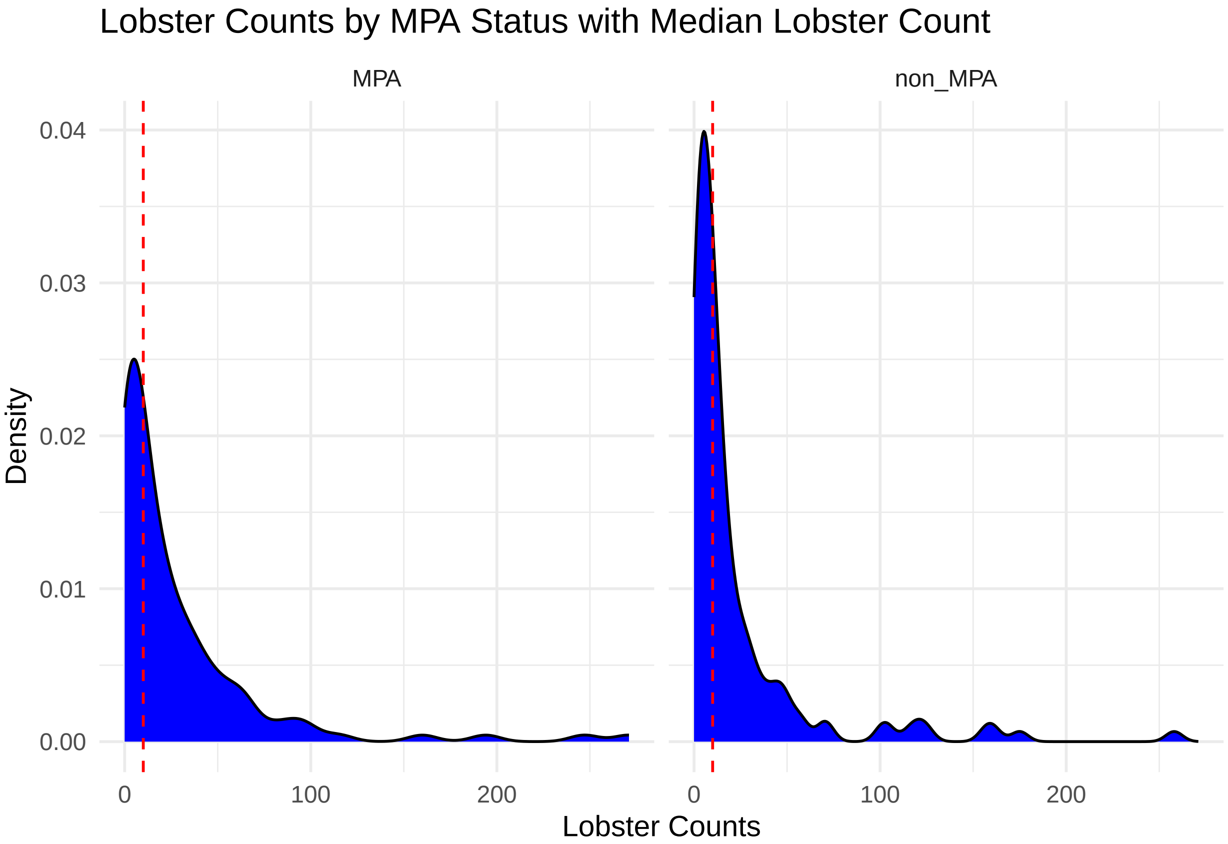
Create a plot of lobster **size** :

4) You choose the grouping variable(s)!

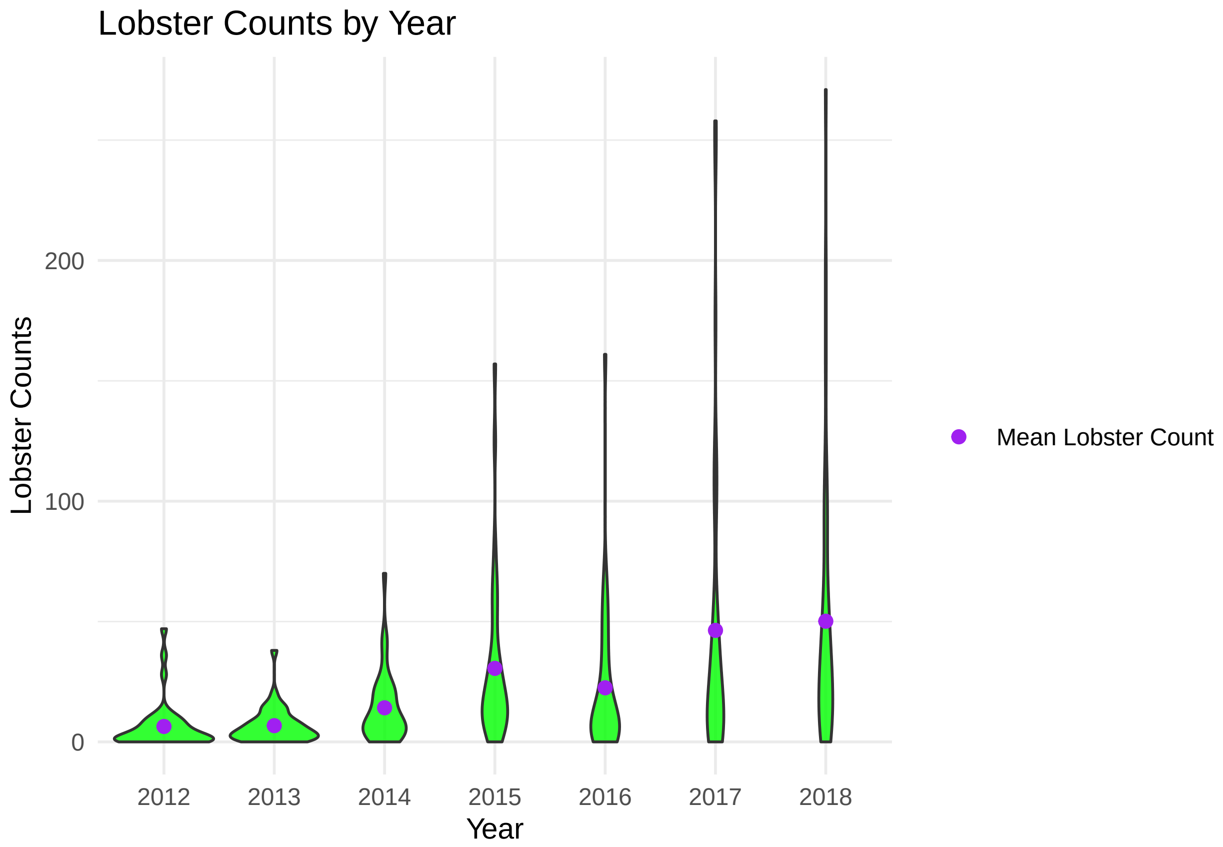
```
# plot 1: Ridge plot of counts grouped by reef site
spiny_counts %>%
    ggplot(aes(x = counts, y = site, fill = after_stat(x))) +
    geom_density_ridges_gradient(quantile_lines = TRUE,
                         rel_min_height = 0.01,
                         quantiles = 4,
                         alpha = 0.5,
                         scale = 1.8) +
    scale_fill_gradientn(colors = c("#2C5374","#849BB4", "#D9E7EC", "#EF8080", "#8B3A3A")) +
    labs(title = "Lobster Counts by Reef Site with Median Counts",
         x = "Lobster count",
         y = "Reef site",
         fill = "Lobster count") +
    theme_minimal()
```



```
# plot 2: Density of counts grouped by MPA status
spiny_counts %>%
    ggplot(aes(x = counts)) +
    geom_density(fill = "blue") +
    facet_wrap(~mpa) +
    geom_vline(aes(xintercept = median(counts), fill="Median Lobster Count"), color = "red", linetype =
    labs(title = "Lobster Counts by MPA Status with Median Lobster Count",
         x = "Lobster Counts",
         y = "Density") +
    theme_minimal()
```



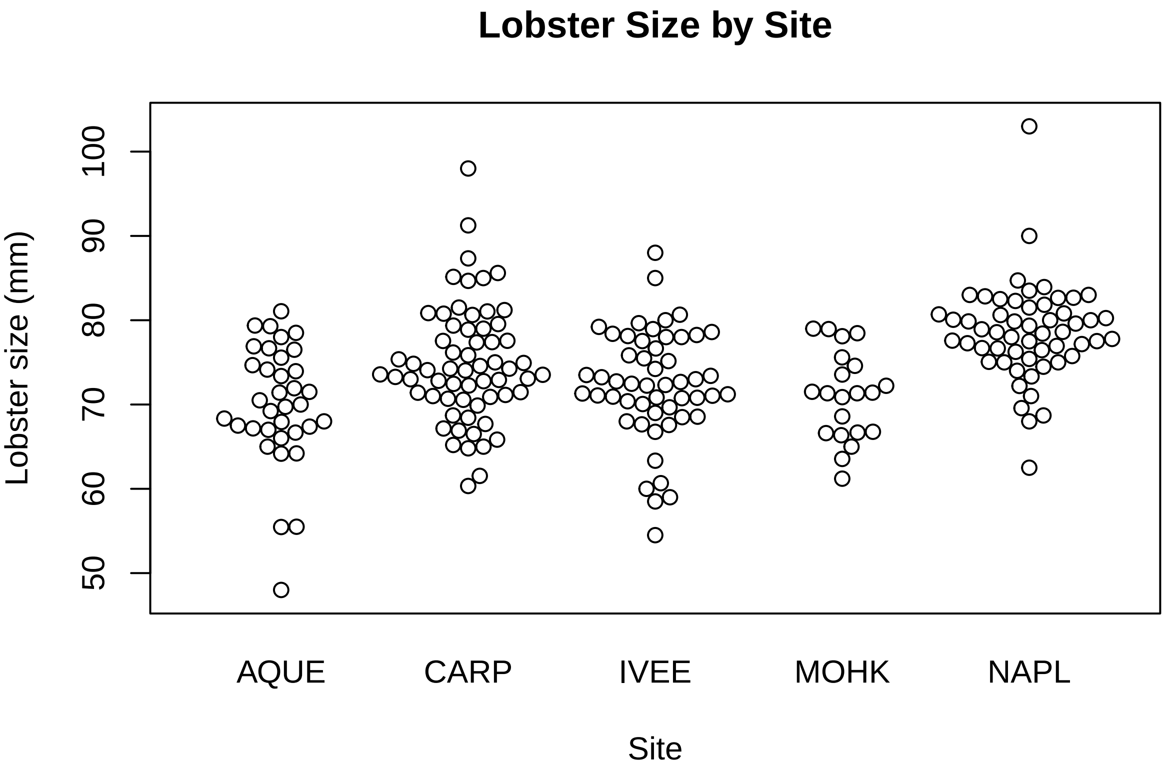
```
# plot 3: Violin plot of counts grouped by year
spiny_counts %>%
    ggplot(aes(x = as.factor(year), y = counts)) +
    geom_violin(fill = "green", trim = TRUE, alpha=0.8) +
    stat_summary(fun.y = mean, geom = "point", color = "purple", size = 2, aes(fill="Mean Lobster Count
    labs(title = "Lobster Counts by Year",
         x = "Year",
         y = "Lobster Counts",
         fill = " ") +
    theme_minimal()
```



```
# plot 4: Lobster size  grouped by site
beeswarm(mean_size ~ site, data = spiny_counts,
         xlab = "Site",
         ylab="Lobster size (mm)",
         main="Lobster Size by Site")
```

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

[1] Mean (SD)



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

```
# USE: gt_summary::tbl_summary()

# View mean outcomes by site treatment
spiny_counts %>%
    dplyr::select(treat, counts, mean_size) %>%
    tbl_summary(by = treat,
                statistic = list(all_continuous() ~ "{mean} ({sd})"))
```

Step 4: OLS regression- building intuition

a. Start with a simple OLS estimator of lobster counts regressed on treatment. Use the function `summ()` from the `jtools` 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)
```

```r
# Linear model of counts as a function of treatment
m1_ols <- lm(counts ~ treat, data = spiny_counts)

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

Observations	252
Dependent variable	counts
Type	OLS linear regression

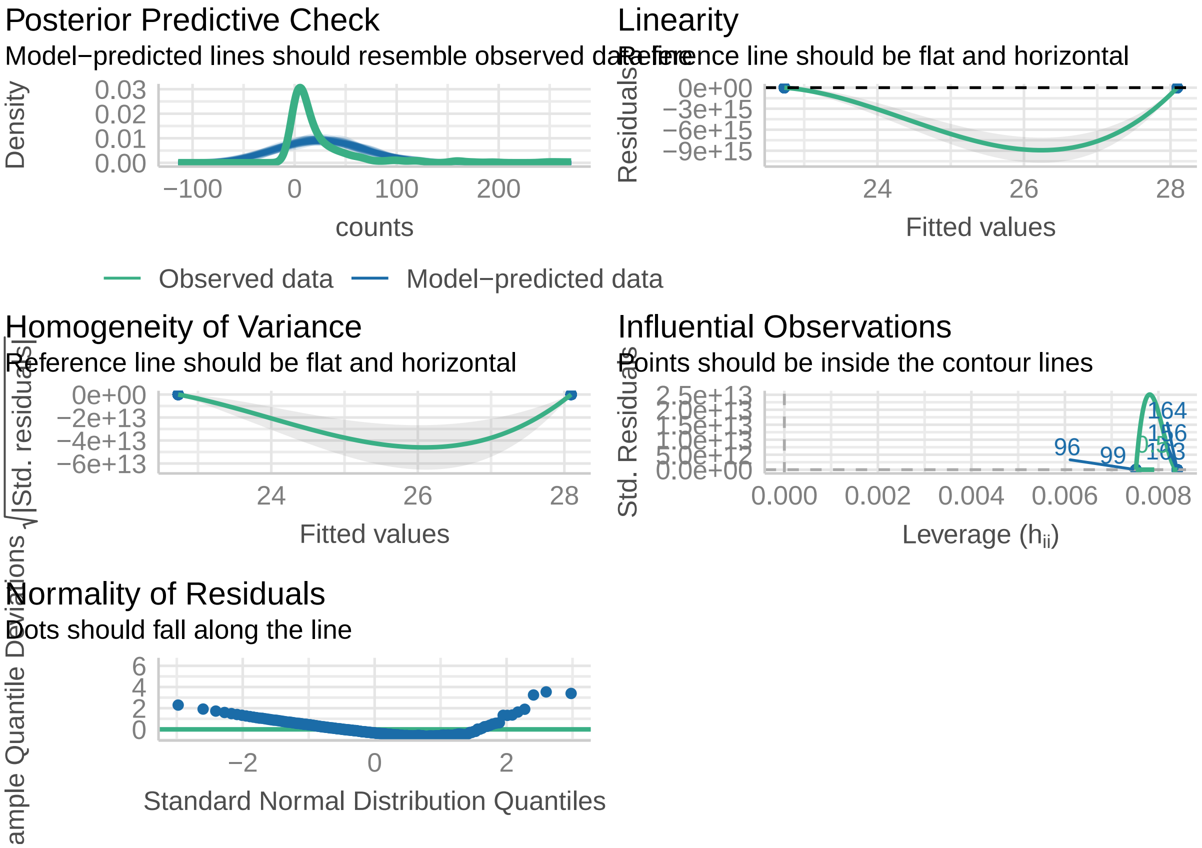
	Est.	S.E.	t val.	p
(Intercept)	22.73	3.57	6.36	0.00
treat	5.36	5.20	1.03	0.30

Standard errors: OLS

Our intercept coefficient tells us that for non-MPA sites, the estimated lobster count is approximately 23 lobsters. The predictor coefficient tells us that for MPA sites, the estimated lobster count for MPA sites is approximately 28 lobsters, which is The estimated treatment effect of MPA sites is 5 lobsters.

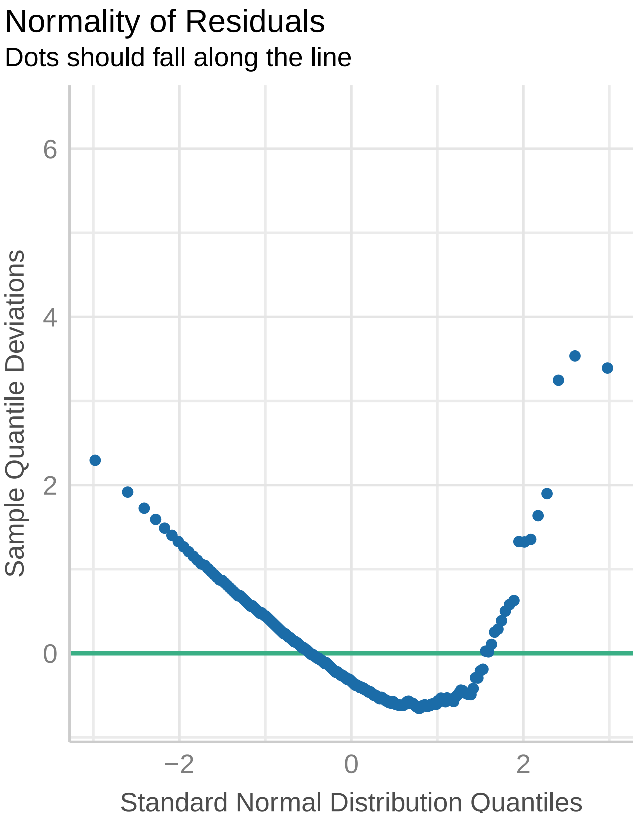
c. Check the model assumptions using the `check_model` function from the `performance` package

```r
check_model(m1_ols)
```

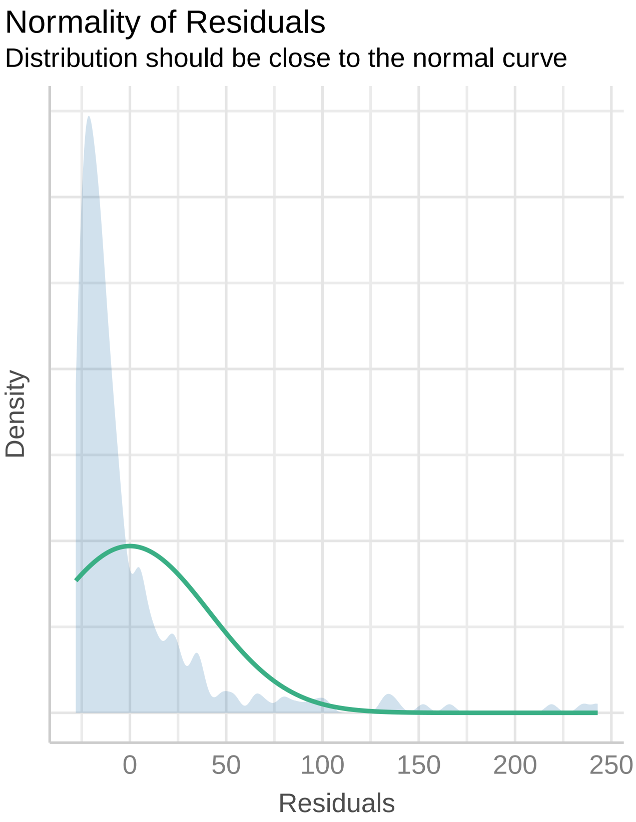


d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

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



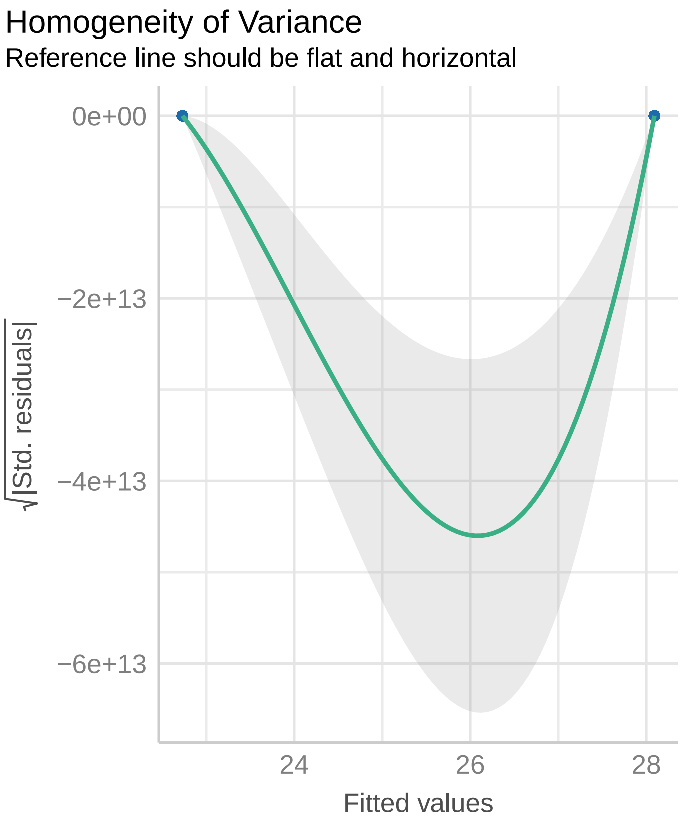
```
check_model(m1_ols, check = "normality")
```



Normality of Residuals

Distribution should be close to the normal curve

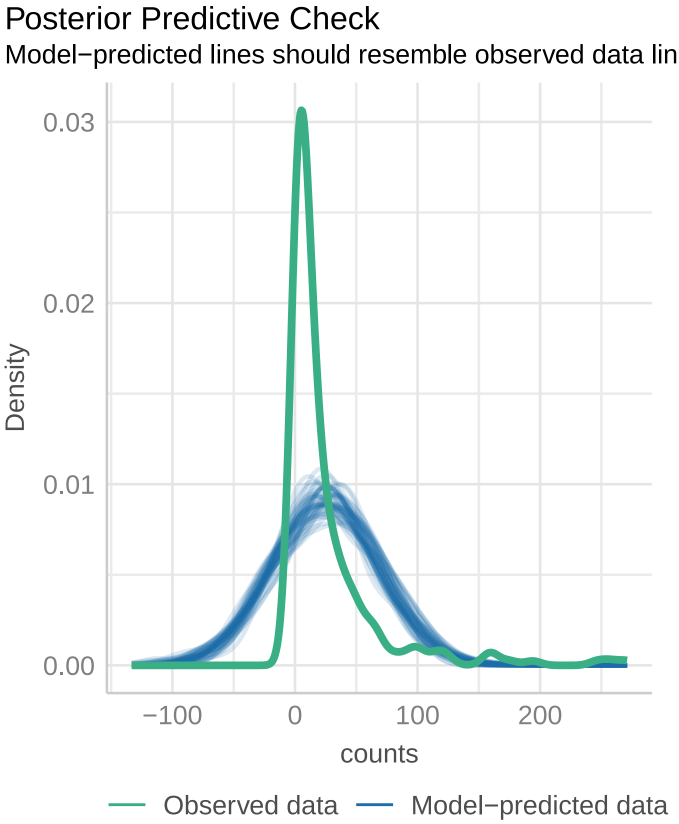
```r
check_model(m1_ols, check = "homogeneity")
```



Homogeneity of Variance

Reference line should be flat and horizontal

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



The first two plots tell us that our residuals are not normally distributed, which violates an assumption of OLS. The third plot tells us that we do not have the same variance across all values of X, or in other words, it is not homoscedastic. The posterior predictive check did not accurately predict our data. The combination of all of these things is likely because we have chosen an incorrect model for our dataset and system.

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 predictor coefficient for the poisson model tells us that MPA sites have 23% more lobsters than the non-MPA sites.

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

Dispersion refers to the how much variability we see amongst out lobster counts, or how "dispersed" they are in magnitude. In the context of this model, overdispersion means that the variance in the lobster counts between the treatment sites is greater than the overall mean of the lobster counts.

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

The insignificant p-value for the OLS model tells us that any relationship we observed between the site treatment and the lobster counts could just be due to random chance. However, our

poisson model does have a significant treatment effect, indicating it is very unlikely this result is due to chance alone.

```r
#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

# Possion model of counts as a function of treatment
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

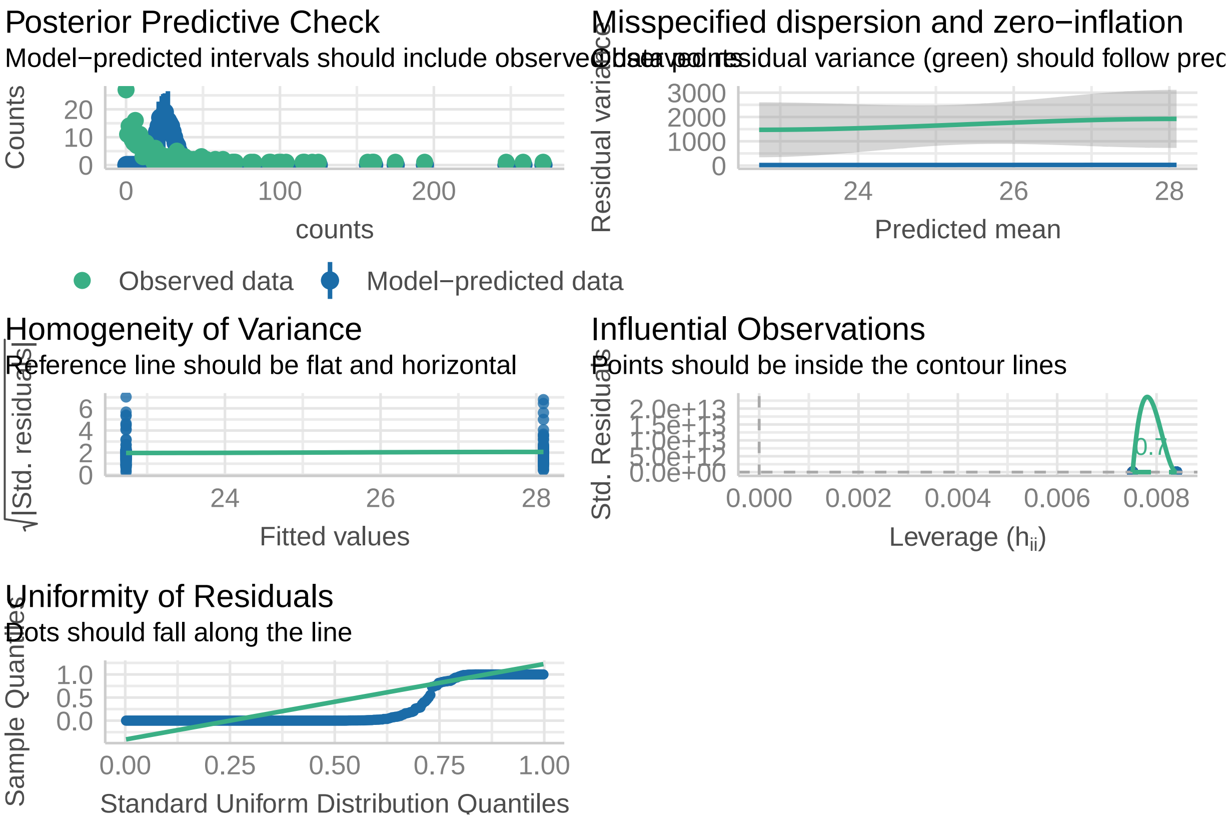
```r
# Un-log our treatment into percent change so we can interpret it
(exp(m2_pois$coefficients["treat"])-1)*100
```

```
##    treat
## 23.59557
```

e. Check the model assumptions. Explain results.

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

```r
check_model(m2_pois)
```



Checking assumptions, it is clear that we are still is violating many assumptions of the poisson model. Our residuals are not uniform and there seems to be zero-inflation. Additionally, our data does not follow our posterior distribution. However, the variance does seem to be much more homogeneous, meaning we've improved slightly from our OLS model. There are also no influential observations. Overall, these results suggest we should search for a better model.

```
check_overdispersion(m2_pois)
```

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

Our model is very overdispersed, indicating that we our variance is greater than our mean.

```
check_zeroinflation(m2_pois)
```

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

Our model also displays zero-inflation because the number of observed zeroes is larger than the number of predicted data. This tells us we should consider a negative binomial model or a zero-inflated model.

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

h. In 1-2 sentences explain rationale for fitting this GLM model.

We choose to fit with a negative binomial because our poisson model showed overdispersion. The negative binomial is a generalization of the poisson, but allows for the variance to be larger than the mean, which helps account for the overdispersion.

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

The treatment estimate for our negative binomial model esimates a 23% increase in lobster counts in the MPA sites. This is almost identical to that of our poisson model; however, this estimate is statistically significant whereas the poisson was not.

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

# Negative binomial model of counts as a function of treatment
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

```
# Un-log our treatment into percent change so we can interpret it
(exp(m3_nb$coefficients["treat"])-1)*100
```

```
##    treat
## 23.59557
```

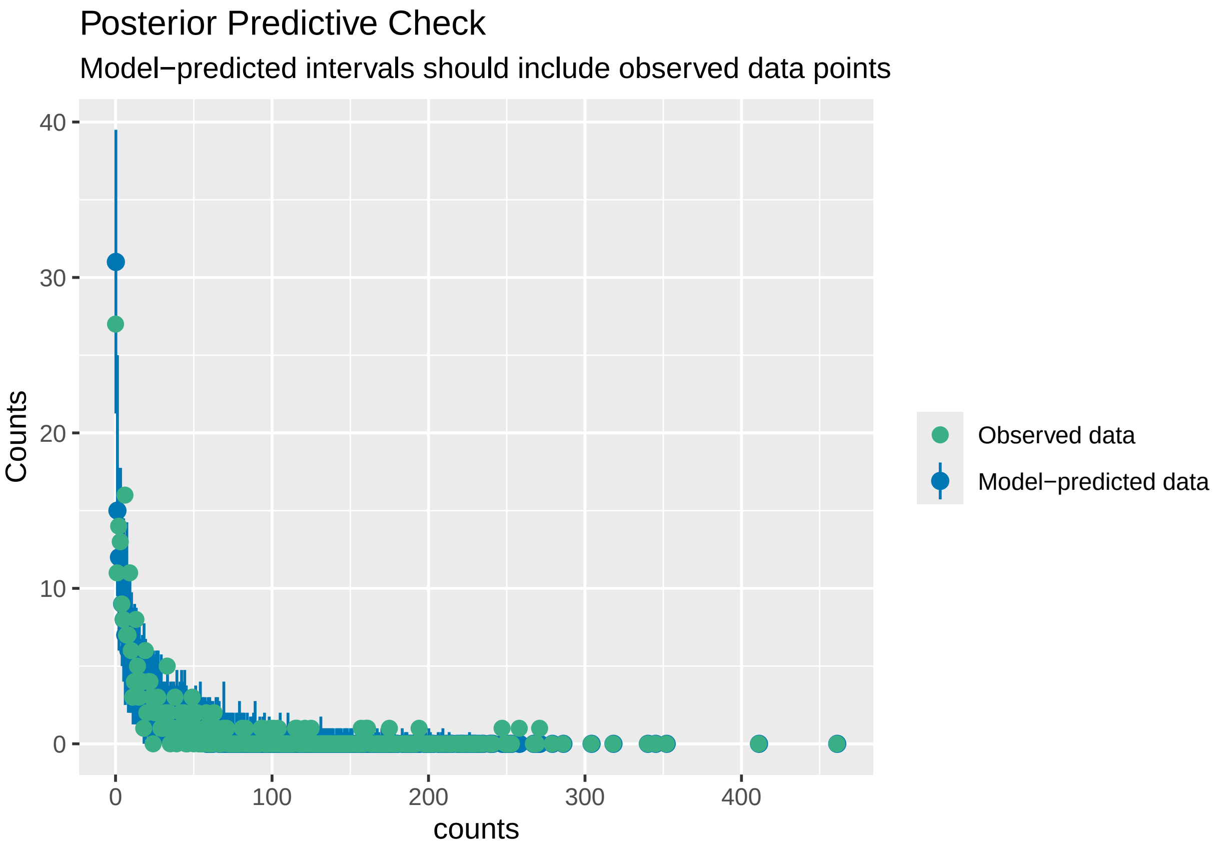
```
check_overdispersion(m3_nb)
```

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

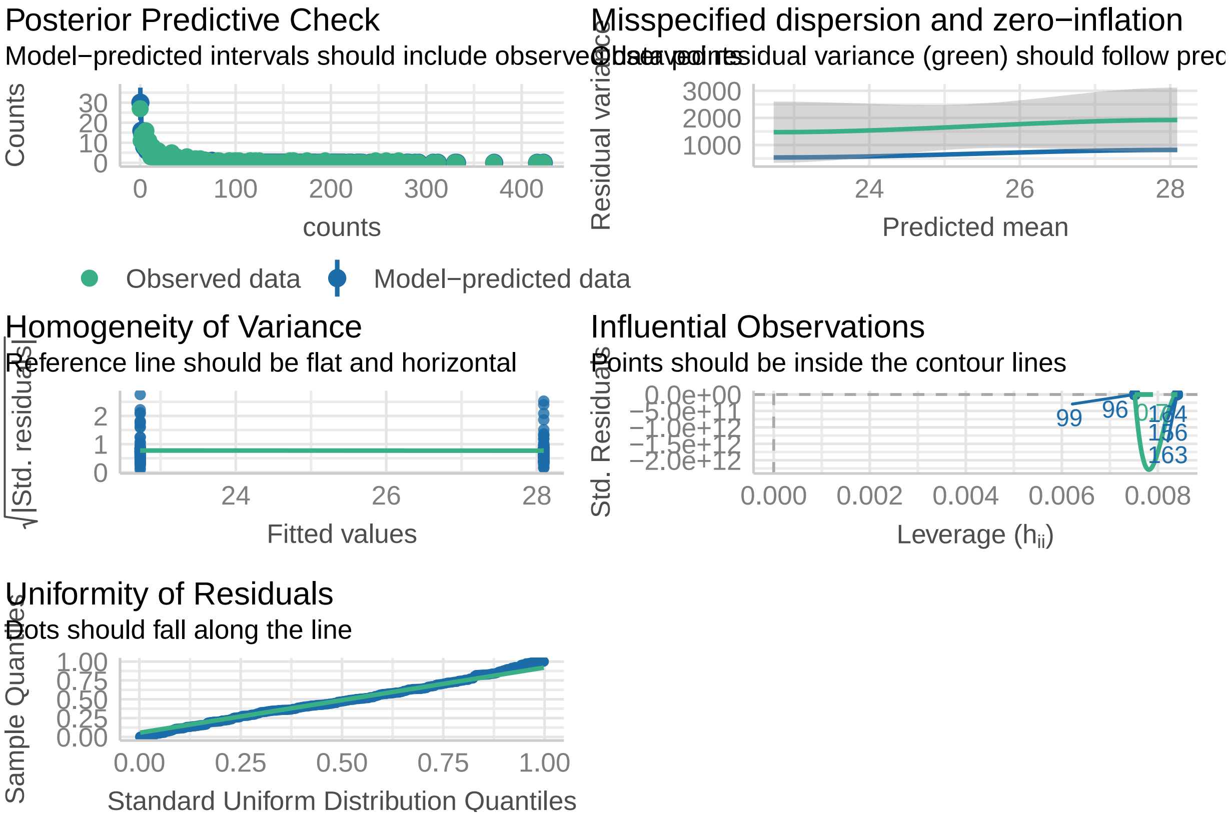
```
check_zeroinflation(m3_nb)
```

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

```
check_predictions(m3_nb)
```



```
check_model(m3_nb)
```



We can see that our data fits the assumptions of the negative binomial model much better than the poisson model. Our residuals are uniform, we have very little overdispersion and zero-inflation, and our data follows our posterior distribution.

We can also see that the percent change in lobster observation as a result of the MPA site treatment is also 23%, which is the same as our poisson model.

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.

```r
# View all three model results side by side
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)

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

```r
# Calculate percent change in lobster counts in each model:

# ratio of treatment beta coeff / intercept
m1_est_ols = (5.36/22.73)*100  # % change = 23.58%

m2_est_poi = (exp(0.21)-1)*100  # % change = -23.37%%
m3_est_log = (exp(0.21)-1)*100  # % change = -23.37%
```

Across all three models, we see a percent change of approximately a 23% increase in lobster counts in MPA sites compared to non-MPA sites. It appears our treatment effect is stable. When we calculate the percent change in lobster counts, we get approximately 23% across all three models we ran. This tells us that the treatment effect is stable across different model specifications.

As a whole, the treatment effect does not seem to be robust.

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)

The model is estimating the effect that site treatment and year have on lobster counts, both individually and as an interaction.

When you look at the estimate for year alone, it seems like the coefficient is slowly increasing over time, indicating that there lobster counts are also slowly increasing over time. However, when you consider treatment and year together, you see and increase, a decrease, then an increase, indicating that lobster counts have more variability from year to year than the previous term suggested. Additionally, all terms see an increase in lobster counts except for year 2013 and 2012, our reference level.

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

The main effect for treatment is negative because this is our reference level of 2012, which was at the beginning of the MPA designation. The negative is telling us that there were more lobsters in non-MPA sites than MPA ones, which is very probable given that it takes time for any benefits of marine protection to trickle down to the lobster species. So lobster counts did not start increasing in MPAs until later years.

```r
# Convert year to factor
ff_counts <- spiny_counts %>%
    mutate(year=as_factor(year))

# Negative binomial model with fixed effects
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

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

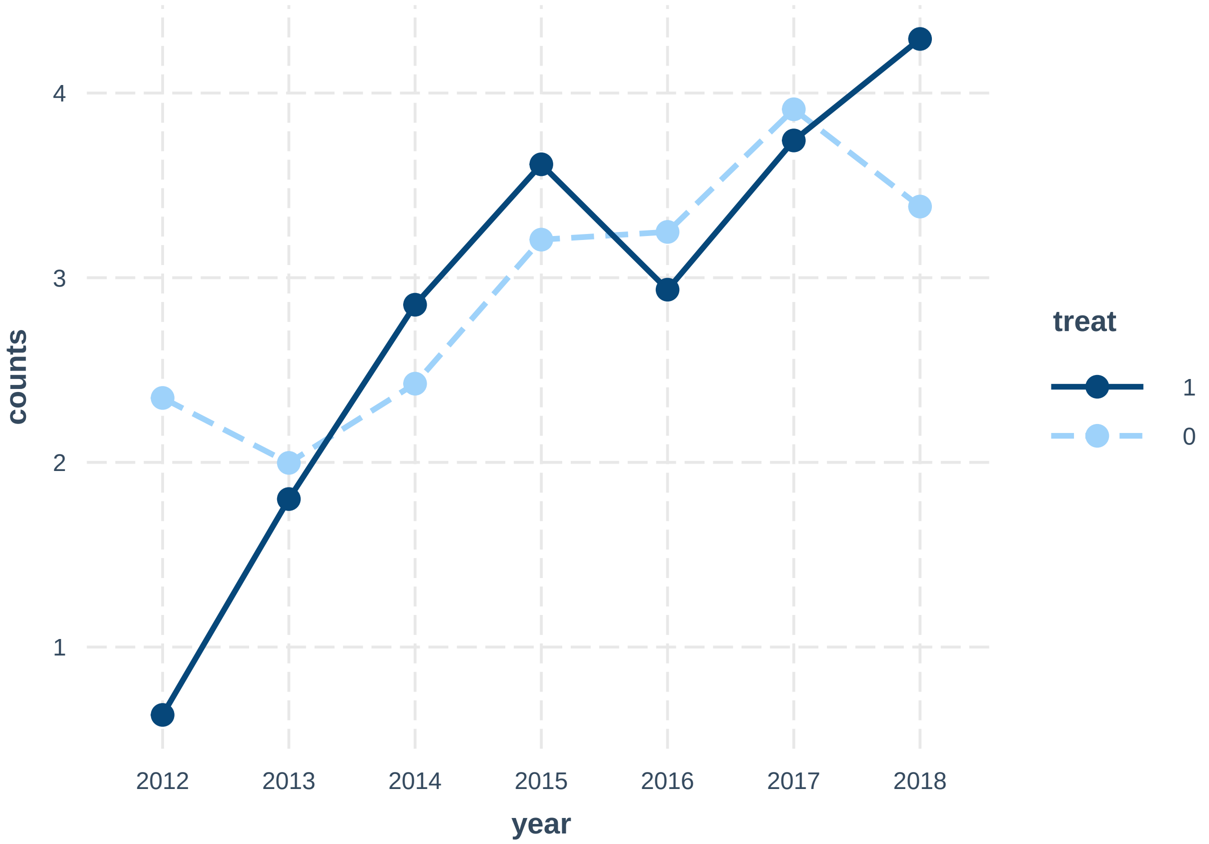
f. Re-evaluate your responses (c) and (b) above.

This plot affirms my responses in c and d. When the y-axis is converted back to counts, you can very clearly see the low starting point of MPA sites in 2012 compared to non-MPA sites, which explains the negative `treat` coefficient. You can also see that increasing, decreasing, increasing trend that I described when looking at the coefficients for interacting treatment and year.

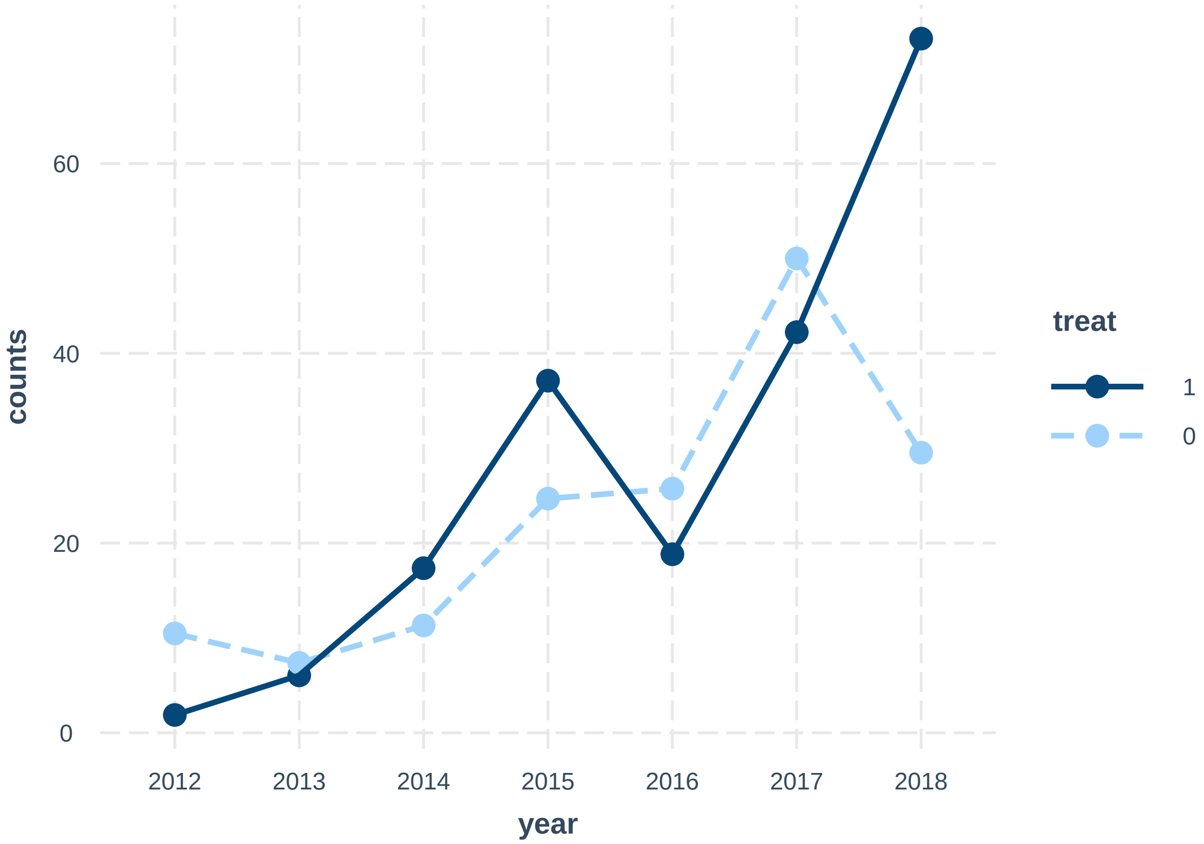
```r
# Looking at the logged outcome variable
interact_plot(m5_fixedeffs, pred = year, modx = treat,
              outcome.scale = "link") # NOTE: y-axis on log-scale
```

	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



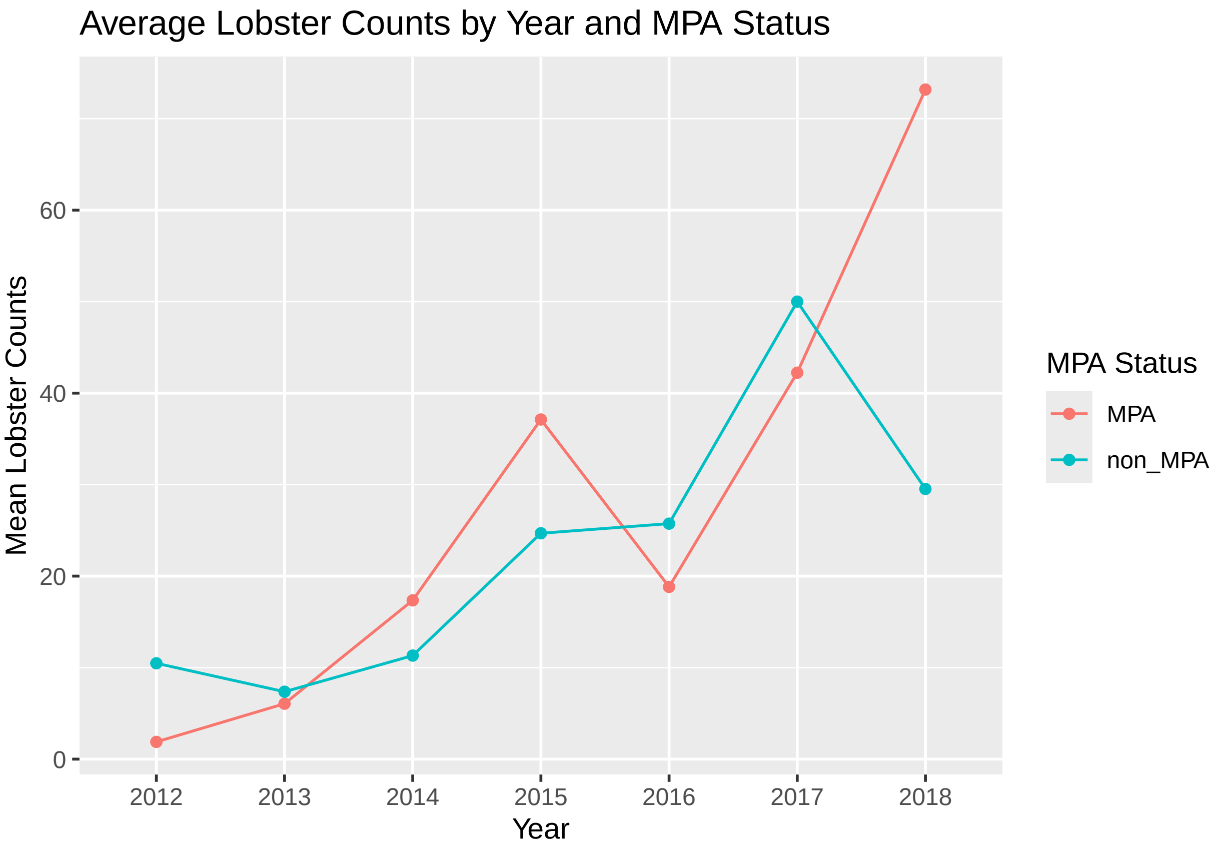
```
# HINT: Change `outcome.scale` to "response" to convert y-axis scale to counts
interact_plot(m5_fixedeffs, pred = year, modx = treat,
              outcome.scale = "response")
```



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

```r
# 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(year, mpa) %>%
    summarize(mean_count = mean(counts, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(year = as.factor(year))

# Plot mean lobster counts by year and MPA status
plot_counts %>% ggplot(aes(x = year, y = mean_count, color = mpa, group=mpa)) +
    geom_point() +
    geom_line() +
    labs(title = "Average Lobster Counts by Year and MPA Status",
         x = "Year",
         y = "Mean Lobster Counts",
         color = "MPA Status")
```



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 spillover effects are likely in this research context. The MPA and non-MPA sites are close together and it would be easy for the lobsters to move from site to site. So any supposed habitat benefit provided to the lobsters by living in a protected area could be detected as "spillover" in the non-protected area as the MPA lobsters move around. However, the magnitude of the spillover would be hard to determine without knowing more about lobster movement patterns and behavior.

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

   Spillover is an issue for identification of causal effects because it makes it difficult to determine whether the observed effect is actually caused by the treatment, or it's caused by the spillover. The treatment and control groups should always be independent of each other when estimating casual effects, and spillover violates some of that independence.

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

   In general, spillover has a positive impact because it means benefits to lobster populations in the MPA sites can spread more generally around the central coast region, including the non-MPA sites as well. The overall effect is a net positive for lobster populations. In the case of this specific research question, spillover has a negative impact because it means that we cannot clearly delineate where lobster population counts are actually higher.

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

In this context, SUTVA refers to the idea that our treatment effect estimator of lobster counts is only influenced by the one assigned treatment effect (MPA site or non-MPA site). This assumption would be violated if there are spillover effects, something that we discussed earlier is a possibility given the geographic proximity of all the sites as a whole. As we have no evidence as to whether spillover is occurring and I personally don't know enough about lobster migration to say whether it's likley or not, I would say it's reasonable but I would be very cautious about interpreting results. Really, it would be great to get more information to know for sure.

1) Excludability assumption

In this context, the excludability assumption refers to the idea that our outcome of lobster counts is affected *only* by the treatment. As we discussed earlier, the MPA sites and non-MPA sites are not perfect counterfactuals. This means that there will be microsite variation in habitat, food, climate, etc. that can influence our lobster population counts. But given that true counterfactuals almost never actually exist in the real world, I would say the excludabililty assumption is reasonable becuase the sites are close enough.

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)

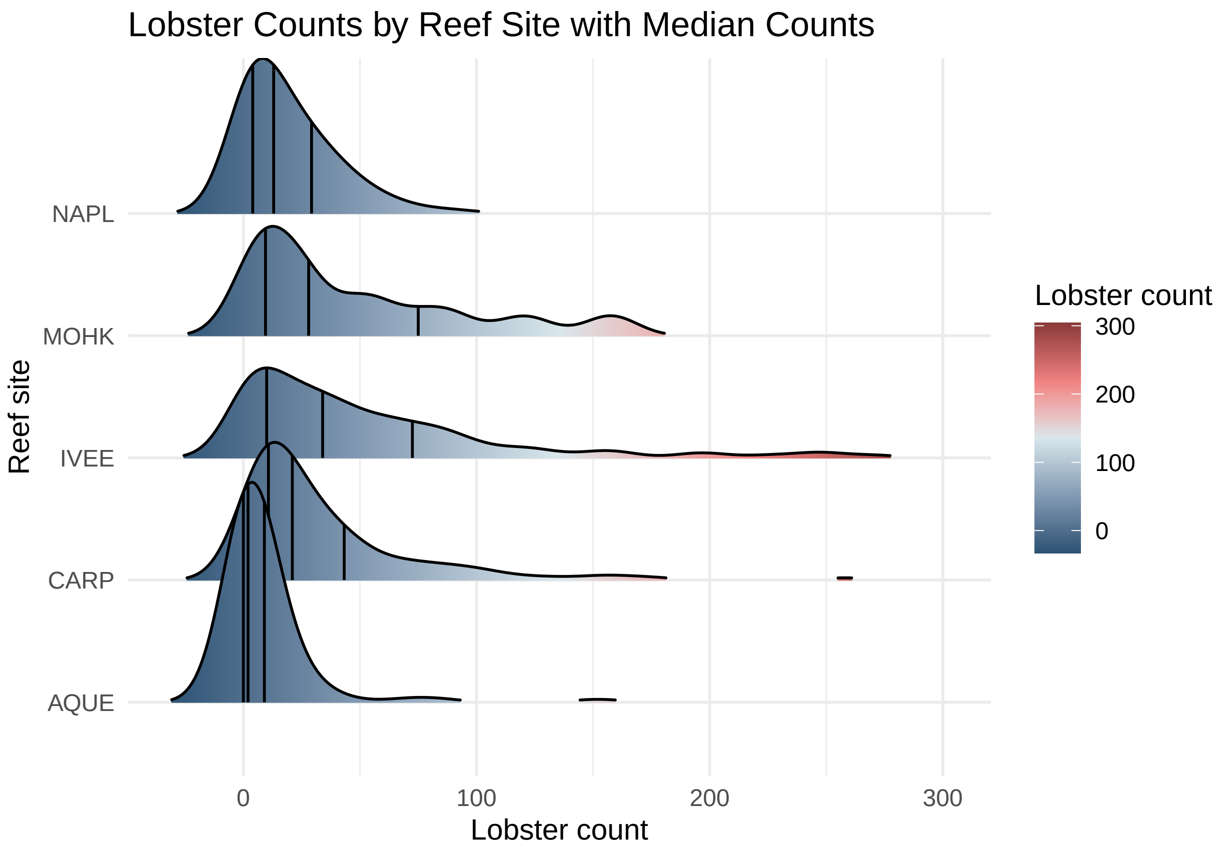
Extra Credit

```r
# Read in data and clean names and NAs
rawdata24 <- read_csv(here("data", "lobster_sbchannel_24.csv"), na = "-99999") %>%
    clean_names()
```

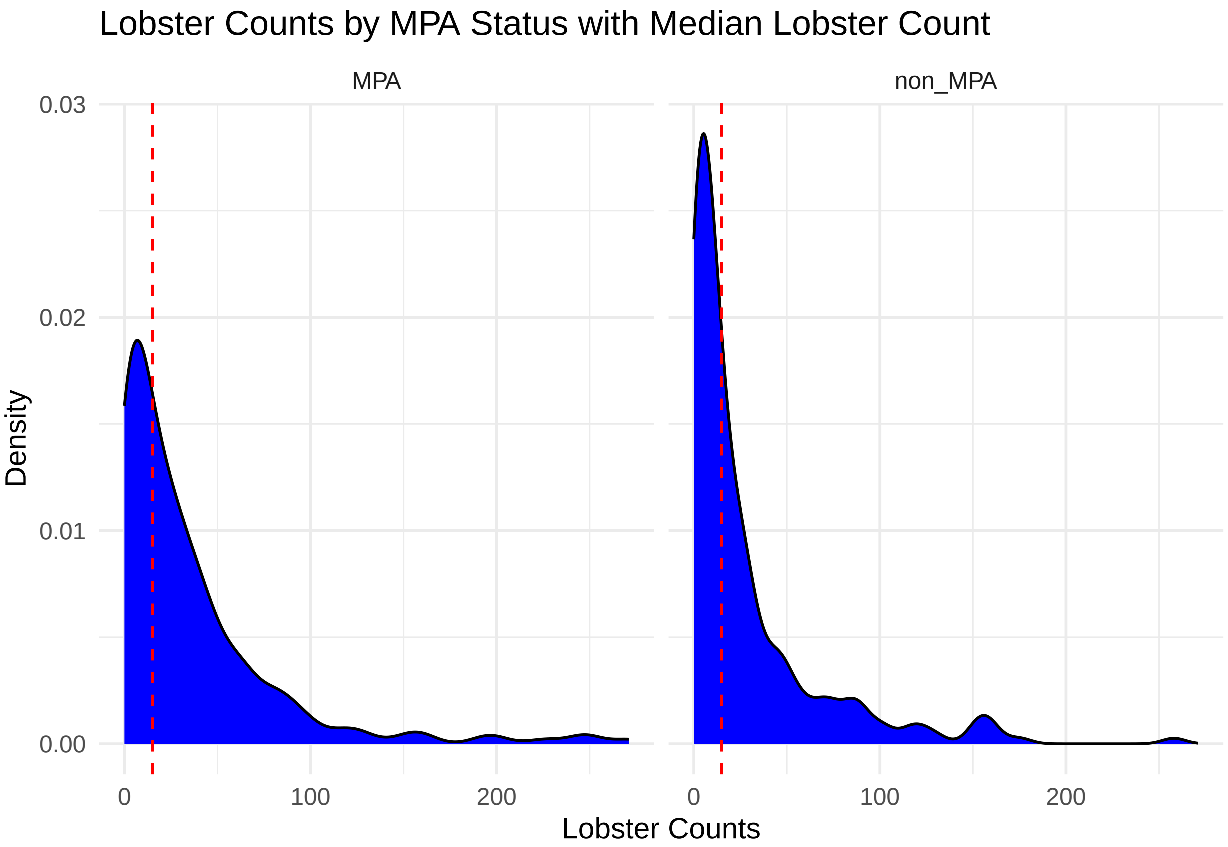
```r
# Refactor our sites into a clean data frame
tidydata24 <- rawdata24 |>
    mutate(reef = factor(site, order = TRUE,
                         levels = c("AQUE",
                                    "CARP",
                                    "MOHK",
                                    "IVEE",
                                    "NAPL"),
                         labels = c("Arroyo Quemado",
                                    "Carpenteria",
                                    "Mohawk",
                                    "Isla Vista",
                                    "Naples")))
```

```r
# Add a treatment variable by MPA site and a mean size variable
spiny_counts24 <- tidydata24 %>%
    group_by(site, year, transect) %>%
    summarize(counts = as.integer(sum(count, na.rm = TRUE)),
              mean_size = mean(size_mm, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(mpa = case_when(site %in% c("IVEE", "NAPL") ~ "MPA",
                           site %in% c("CARP", "MOHK", "AQUE") ~ "non_MPA"),
           treat = case_when(mpa == "MPA" ~ 1,
                             mpa == "non_MPA" ~ 0)) %>%
    mutate(across(where(is.numeric), ~(ifelse(is.na(.), NA_real_, (.)))))
```

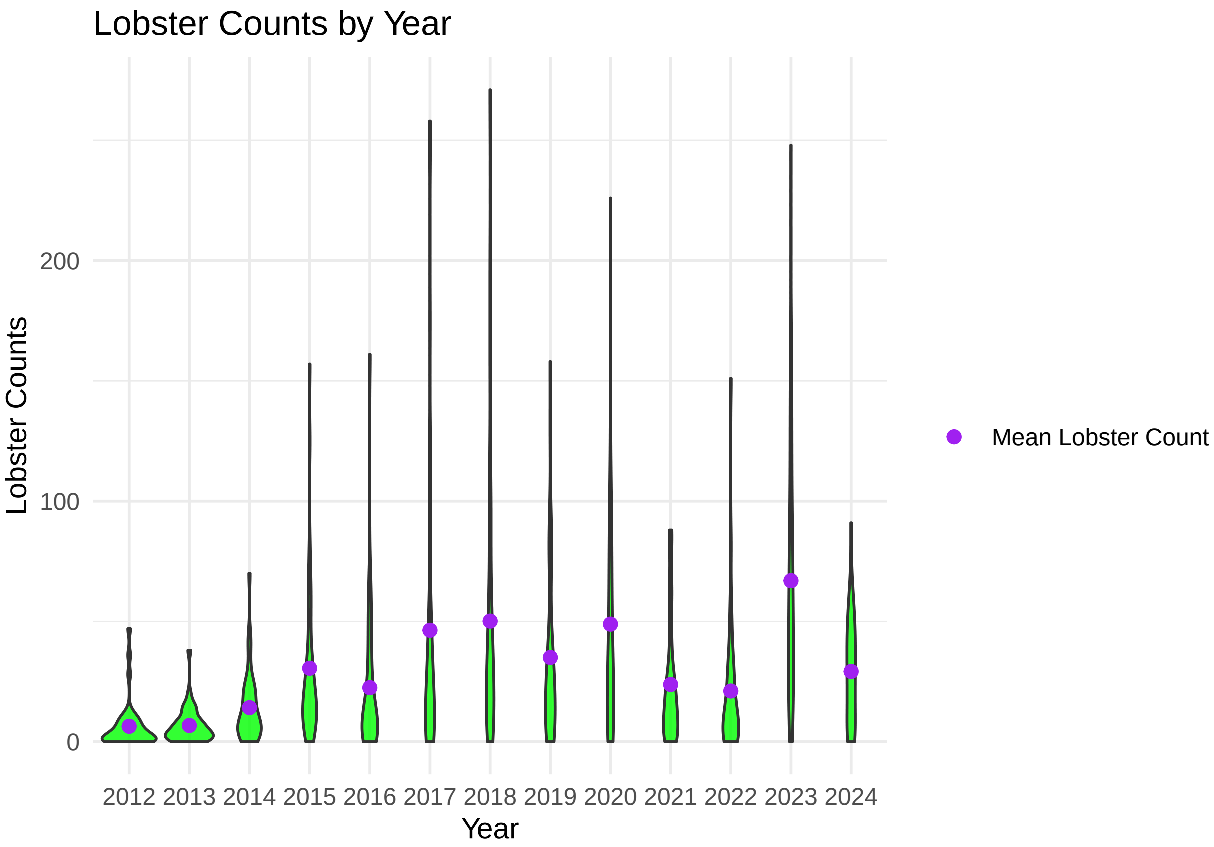
```r
# plot 1: Ridge plot of counts grouped by reef site
spiny_counts24 %>%
    ggplot(aes(x = counts, y = site, fill = after_stat(x))) +
    geom_density_ridges_gradient(quantile_lines = TRUE,
                         rel_min_height = 0.01,
                         quantiles = 4,
                         alpha = 0.5,
                         scale = 1.8) +
    scale_fill_gradientn(colors = c("#2C5374","#849BB4", "#D9E7EC", "#EF8080", "#8B3A3A")) +
    labs(title = "Lobster Counts by Reef Site with Median Counts",
         x = "Lobster count",
         y = "Reef site",
         fill = "Lobster count") +
    theme_minimal()
```



```
# plot 2: Density of counts grouped by MPA status
spiny_counts24 %>%
    ggplot(aes(x = counts)) +
    geom_density(fill = "blue") +
    facet_wrap(~mpa) +
    geom_vline(aes(xintercept = median(counts)), color = "red", linetype = "dashed") +
    labs(title = "Lobster Counts by MPA Status with Median Lobster Count",
         x = "Lobster Counts",
         y = "Density") +
    theme_minimal()
```



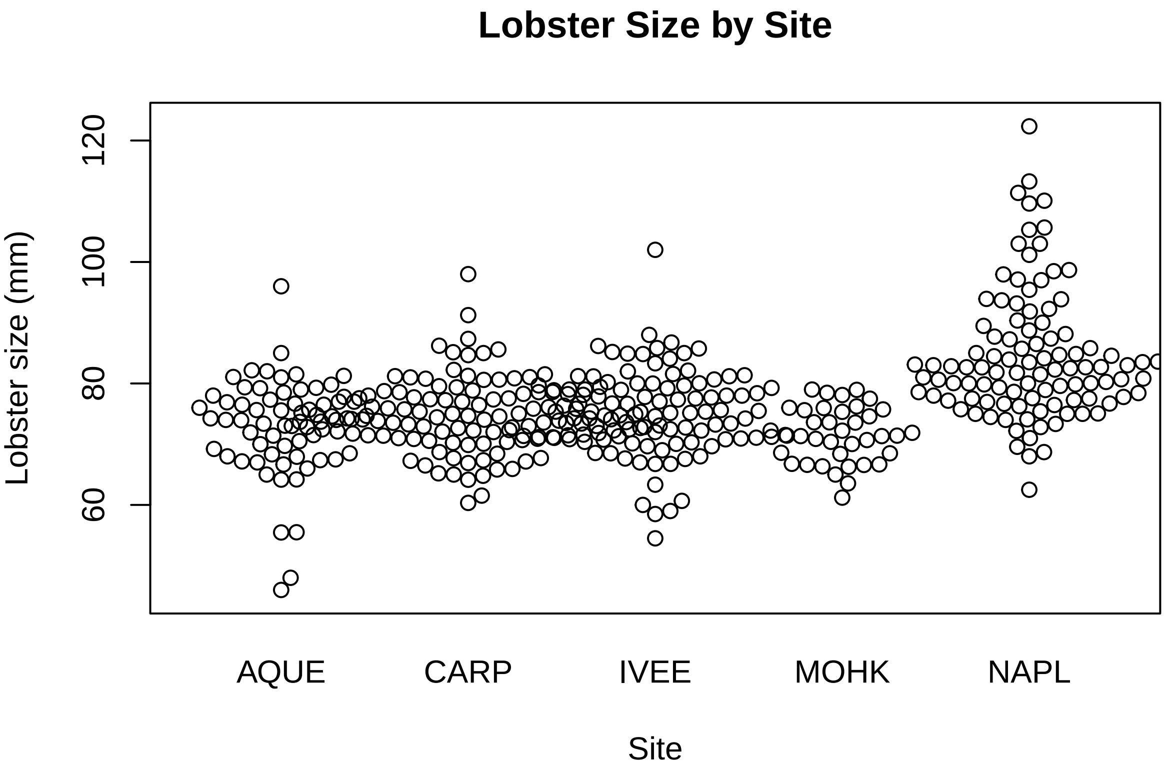
```
# plot 3: Violin plot of counts grouped by year
spiny_counts24 %>%
    ggplot(aes(x = as.factor(year), y = counts)) +
    geom_violin(fill = "green", trim = TRUE, alpha=0.8) +
    stat_summary(fun.y = mean, geom = "point", color = "purple", size = 2, aes(fill="Mean Lobster Count
    labs(title = "Lobster Counts by Year",
         x = "Year",
         y = "Lobster Counts",
         fill = " ") +
    theme_minimal()
```



```
# plot 4: Lobster size  grouped by site
beeswarm(mean_size ~ site, data = spiny_counts24,
         xlab = "Site",
         ylab="Lobster size (mm)",
         main="Lobster Size by Site")
```

Characteristic	**0** N = 246[1]	**1** N = 220[1]
counts	27 (39)	35 (46)
mean_size	74 (7)	80 (10)
Unknown	32	19

[1] Mean (SD)



```r
# View mean outcomes by site treatment
spiny_counts24 %>%
    dplyr::select(treat, counts, mean_size) %>%
    tbl_summary(by = treat,
                statistic = list(all_continuous() ~ "{mean} ({sd})"))
```

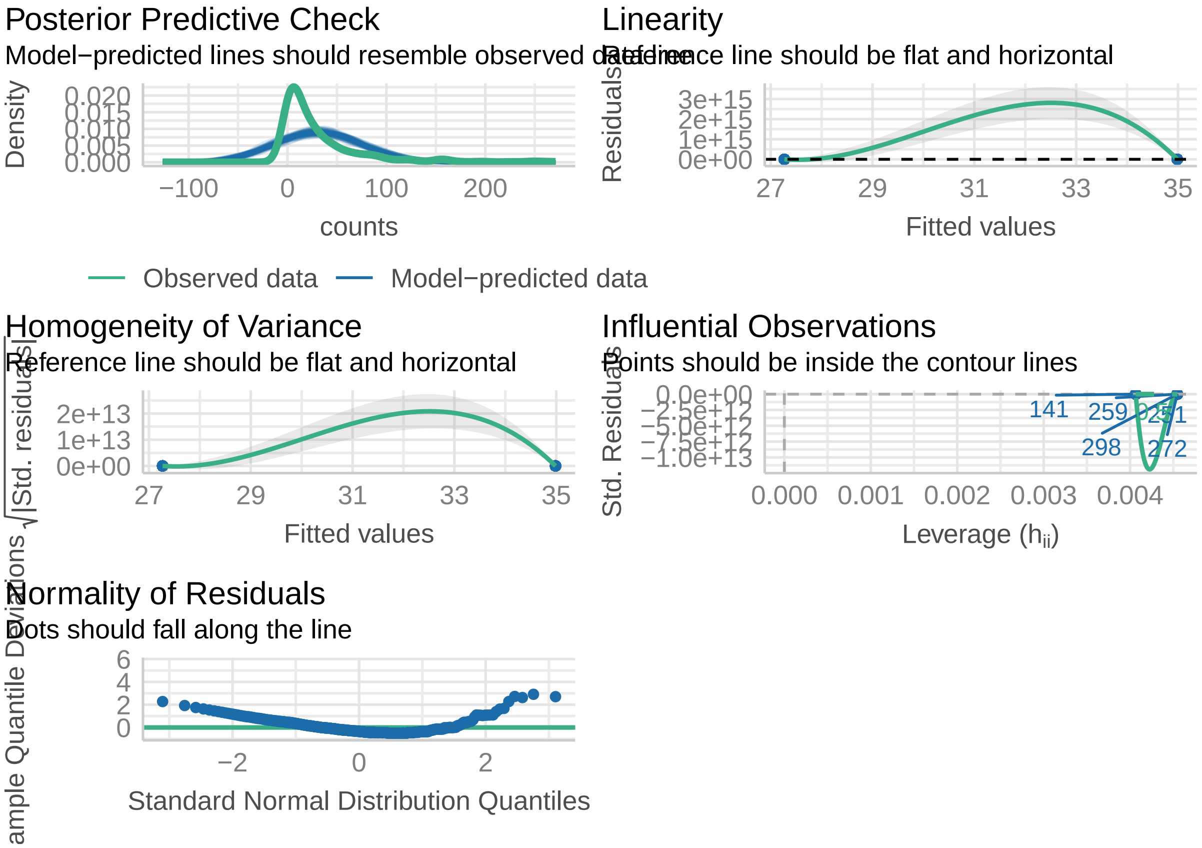
```r
# Linear model of counts as a function of treatment
m1_ols_24 <- lm(counts ~ treat, data = spiny_counts24)

summ(m1_ols_24, model.fit = FALSE)
```

Observations	466
Dependent variable	counts
Type	OLS linear regression

```r
check_model(m1_ols_24)
```

	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



```
# Possion model of counts as a function of treatment
m2_pois_24 <- glm(counts ~ treat,
               data = spiny_counts24,
               family = poisson(link = "log"))

summ(m2_pois_24, model.fit = FALSE)
```

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

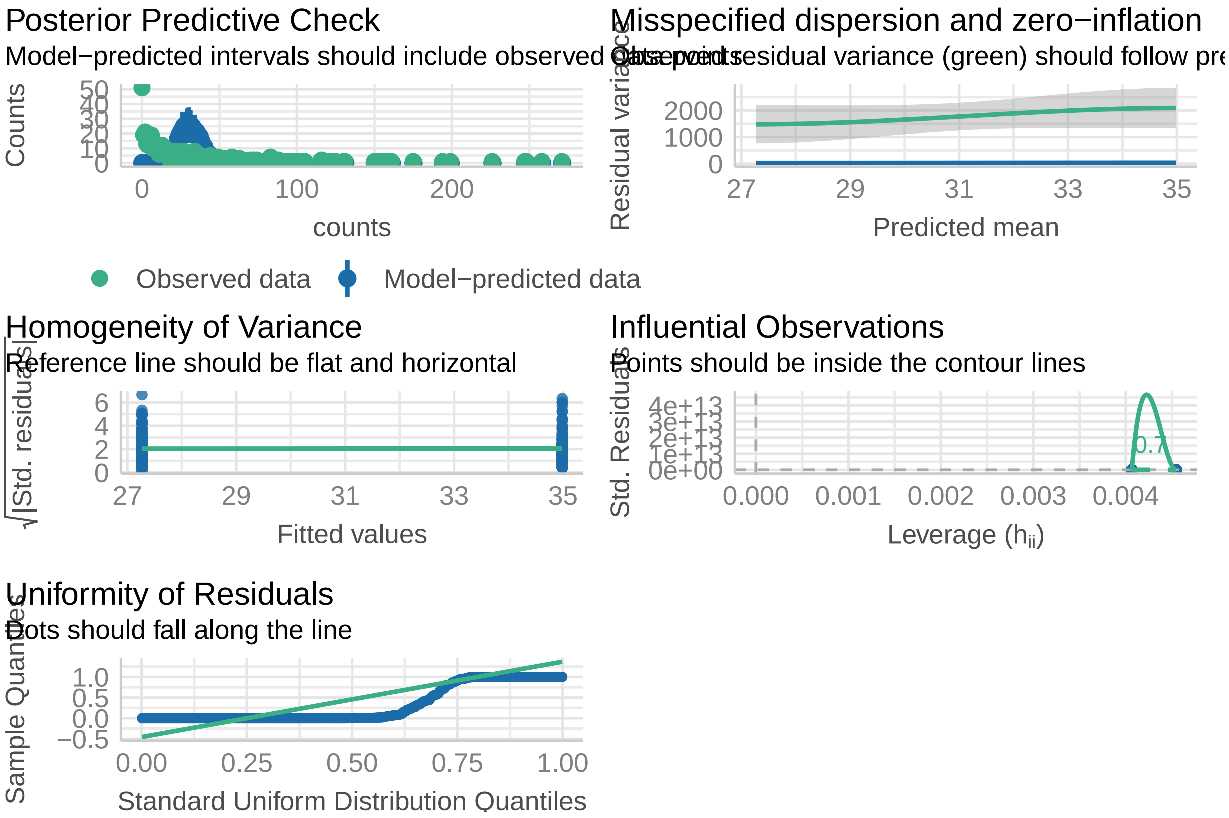
	Est.	S.E.	z val.	p
(Intercept)	3.31	0.01	270.75	0.00
treat	0.25	0.02	14.92	0.00

Standard errors: MLE

```
# Un-log our treatment into percent change so we can interpret it
(exp(m2_pois_24$coefficients["treat"])-1)*100
```

```
##   treat
## 28.3042
```

```
check_model(m2_pois_24)
```



```
# Negative binomial model of counts as a function of treatment
m3_nb_24 <- glm.nb(counts ~ treat,
                   data = spiny_counts24)

summ(m3_nb_24, model.fit = FALSE)
```

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

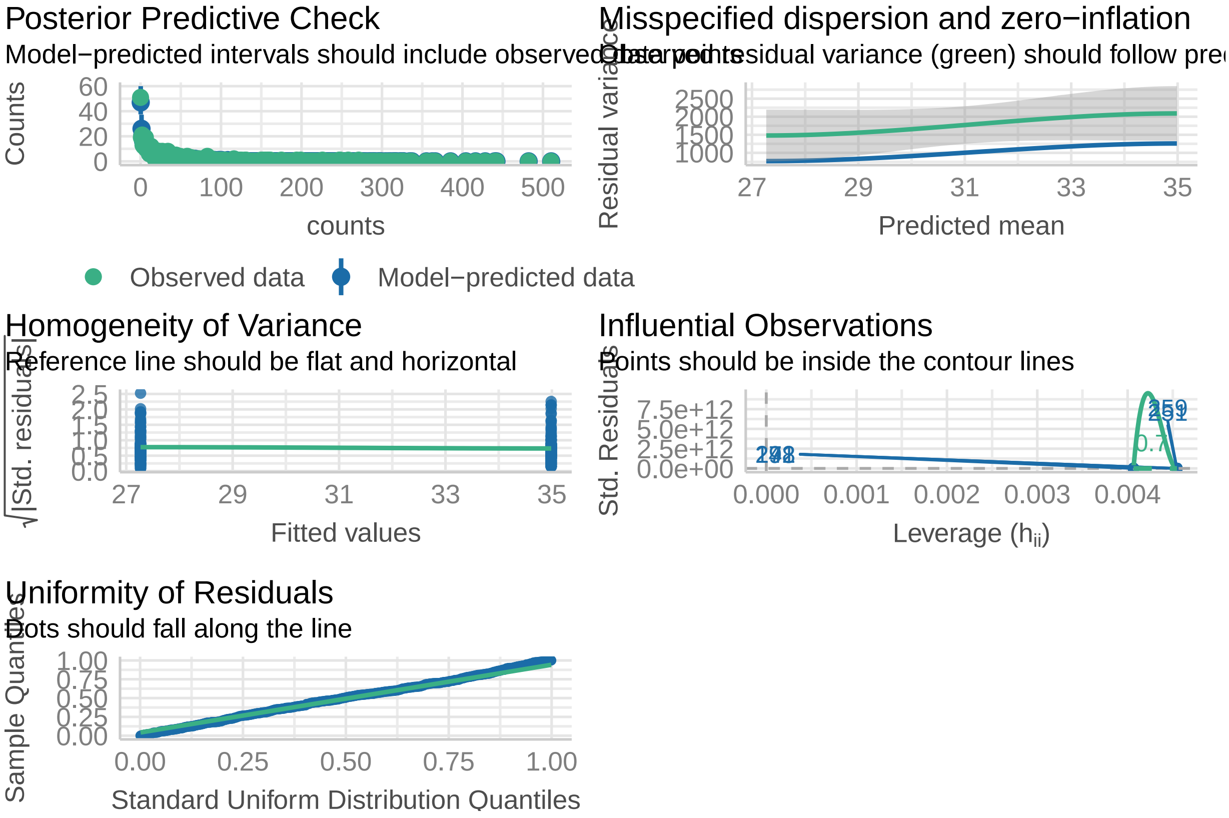
	Est.	S.E.	z val.	p
(Intercept)	3.31	0.08	38.97	0.00
treat	0.25	0.12	2.02	0.04

Standard errors: MLE

```
# Un-log our treatment into percent change so we can interpret it
(exp(m3_nb_24$coefficients["treat"])-1)*100
```

```
##   treat
## 28.3042
```

```
check_model(m3_nb_24)
```



```
# View all three model results side by side
export_summs(m1_ols_24, m2_pois_24, m3_nb_24,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none")
```

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

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

```
# Calculate percent change in lobster counts in each model:

# ratio of treatment beta coeff / intercept
m1_est_ols = (7.72/27.27)*100  # % change = 28.31%

m2_est_poi = (exp(0.25)-1)*100  # % change = -28.40%%
m3_est_log = (exp(0.25)-1)*100  # % change = -28.40%
```