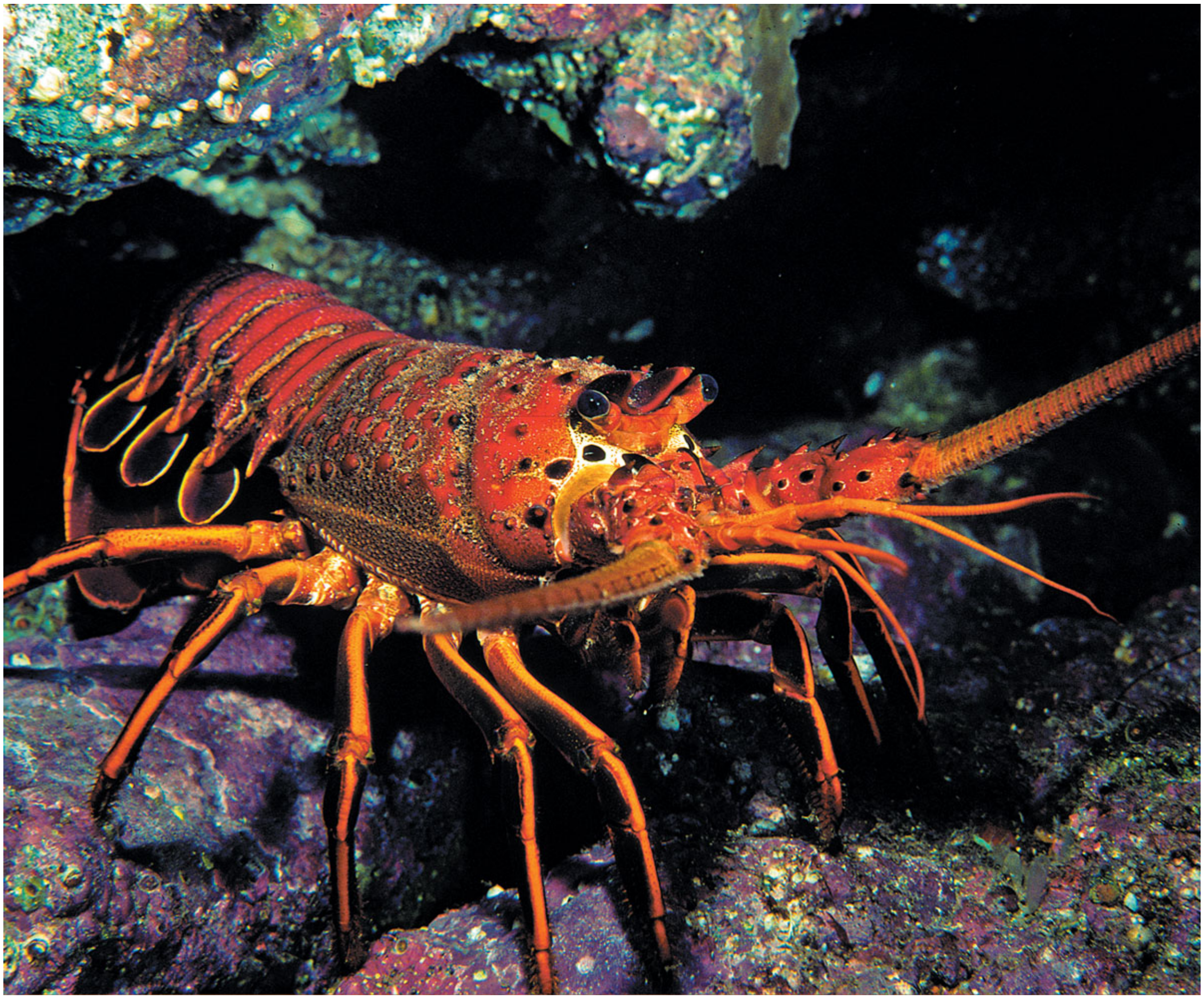
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

EDS 241

1/8/2024 (Due 1/22)



Assignment instructions:

- Working with partners to troubleshoot code and concepts is encouraged! If you work with a partner, please list their name next to yours at the top of your assignment so Annie and I can easily see who collaborated.
- All written responses must be written independently (**in your own words**).
- Please follow the question prompts carefully and include only the information each question asks in your submitted responses.
- Submit both your knitted document and the associated `RMarkdown` or `Quarto` file.
- Your knitted presentation should meet the quality you'd submit to research colleagues or feel confident sharing publicly. Refer to the rubric for details about presentation standards.

Assignment submission Josephine Cardelle: ______________________________

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

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.

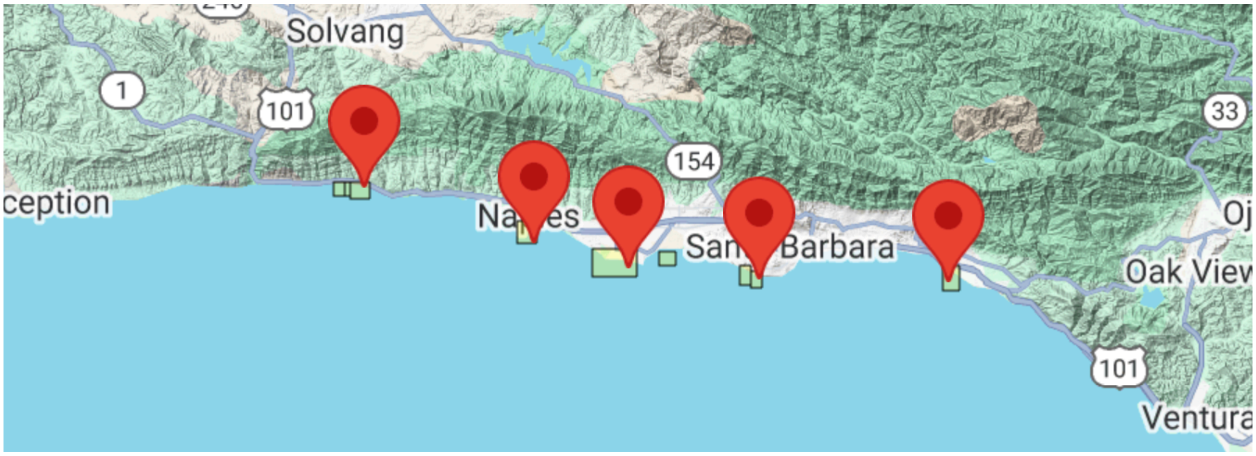
Introduction

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!).

There is likely some selection bias in this treatment. One possible source of selection bias is that the MPAs may have not been randomly selected. Naples and Isla Vista may have better conditions for lobster populations. Another source of selection bias is how close the locations are to large human populations and fishing centers. This could effect lobster abundance the designation of MPAs.

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
rawdata <- read_csv("data/spiny_abundance_sb_18.csv")

rawdata <- clean_names(rawdata)
```

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"
```

```
# Add reef column with full site name
tidydata <- rawdata %>%
    mutate(reef = case_when(site == "IVEE" ~ "Isla Vista",
                         site == "NAPL" ~ "Naples",
                         site == "AQUE" ~ "Arroyo Quemado",
                         site == "CARP" ~ "Carpenteria",
                         site == "MOHK" ~ "Mohawk"))

# Order based on reef column
tidydata$reef = factor(tidydata$reef, levels = c('Arroyo Quemado', 'Carpenteria', 'Mohawk', 'Isla Vista

tidydata = tidydata[order(tidydata$reef), ]
```

Create new `df` named `spiny_counts`

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

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

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

```
#HINT(d): Use `group_by()` & `summarize()` to provide the total number of lobsters observed at each sit
# Change negative sizes to NA
tidydata$size_mm <- replace(tidydata$size_mm, tidydata$size_mm <0, NA)

# Group by site,year,transect and add count and mean size column
spiny_counts <- tidydata %>%
    group_by(site, year, transect) %>%
    summarise(counts = n(),
              mean_size = mean(size_mm, na.rm = TRUE))

#HINT(e): Use `case_when()` to create the 3 new variable columns
# Add mpa and treat column
spiny_counts <- spiny_counts %>%
    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
- Ridge plot
- Jitter plot
- Violin plot
- Histogram
- Beeswarm

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

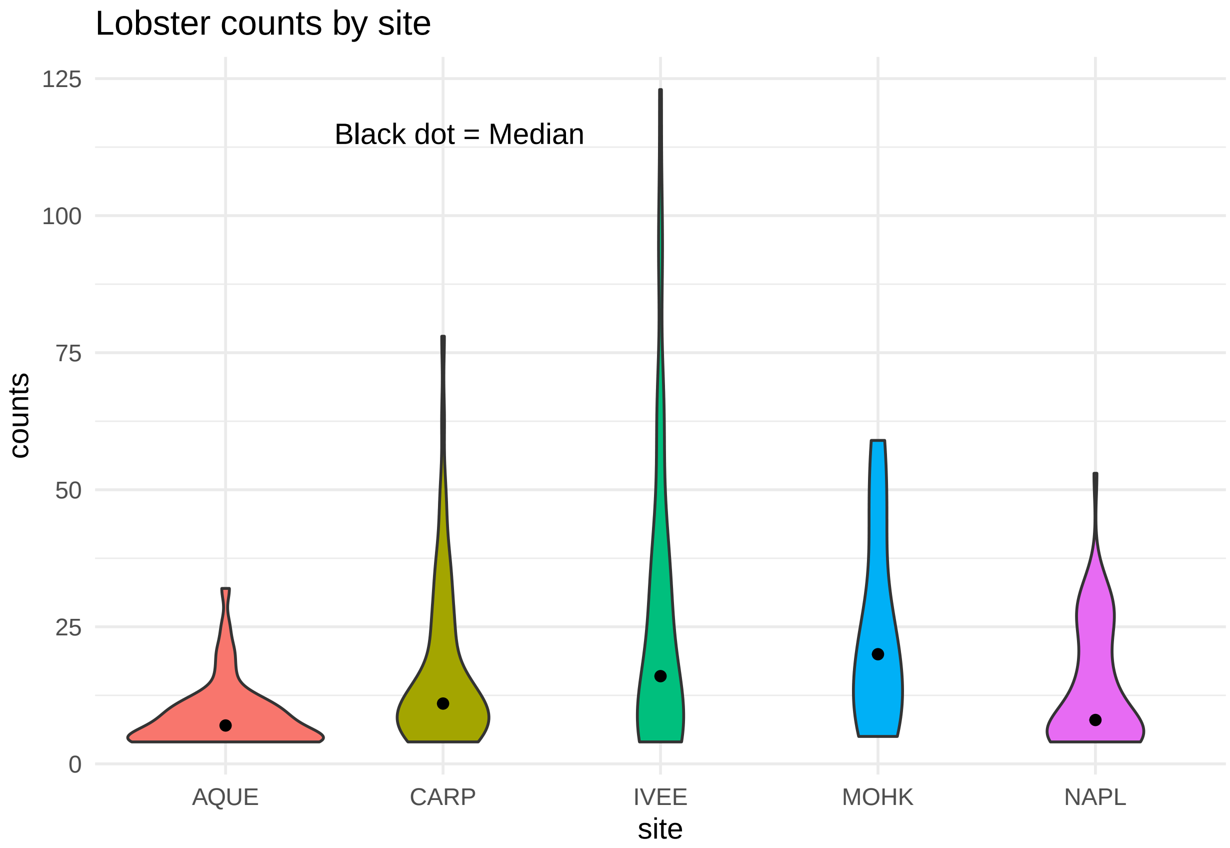
1) grouped by reef site

2) grouped by MPA status
3) grouped by year

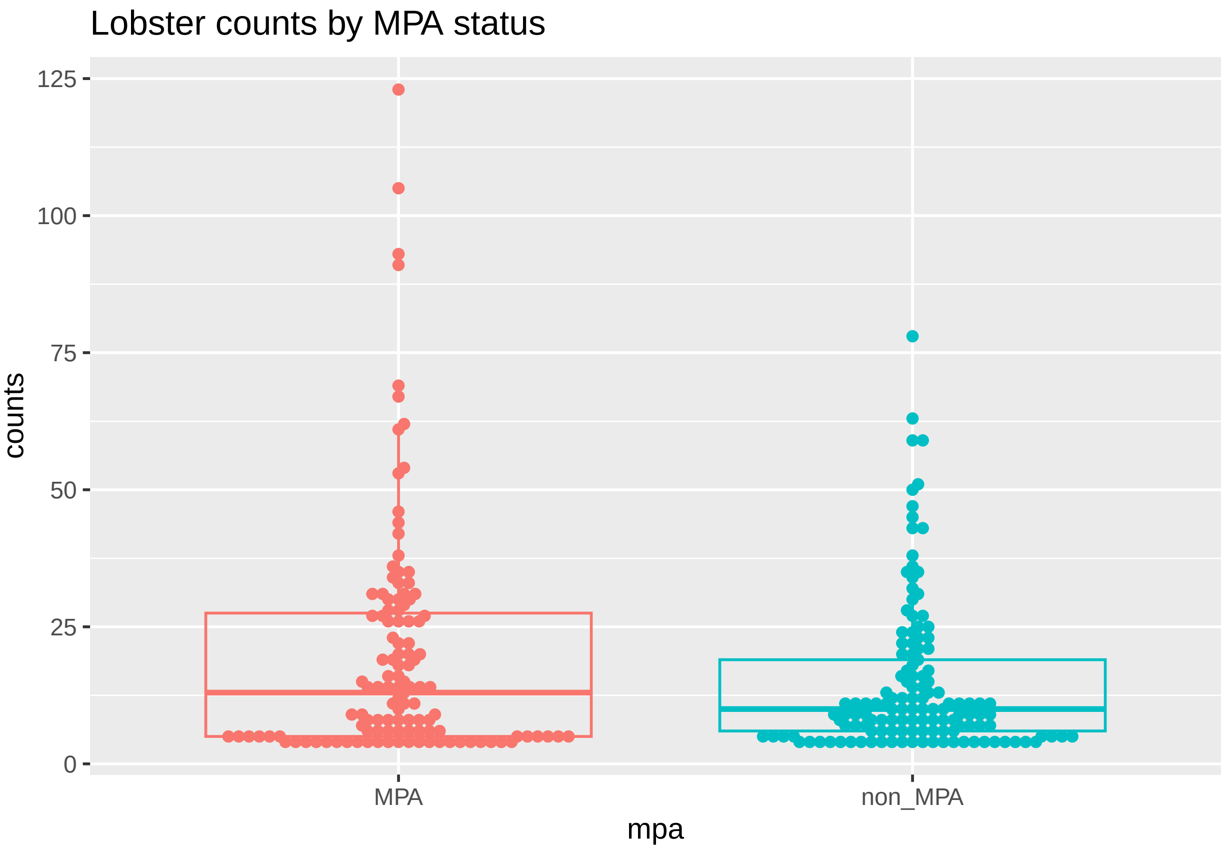
Create a plot of lobster **size** :

4) You choose the grouping variable(s)!

```
# Violin plot grouped by reef site
spiny_counts %>%
ggplot(aes(x = site, y = counts, fill = site)) +
    geom_violin() +
    geom_point(stat = "summary", fun = median, color = "black") +
    annotate("text", x = 1.5, y = 115, label = "Black dot = Median", color = "black", size = 4, hjust =
    theme_minimal() +
    labs(title = "Lobster counts by site") +
    theme(legend.position = "none")
```

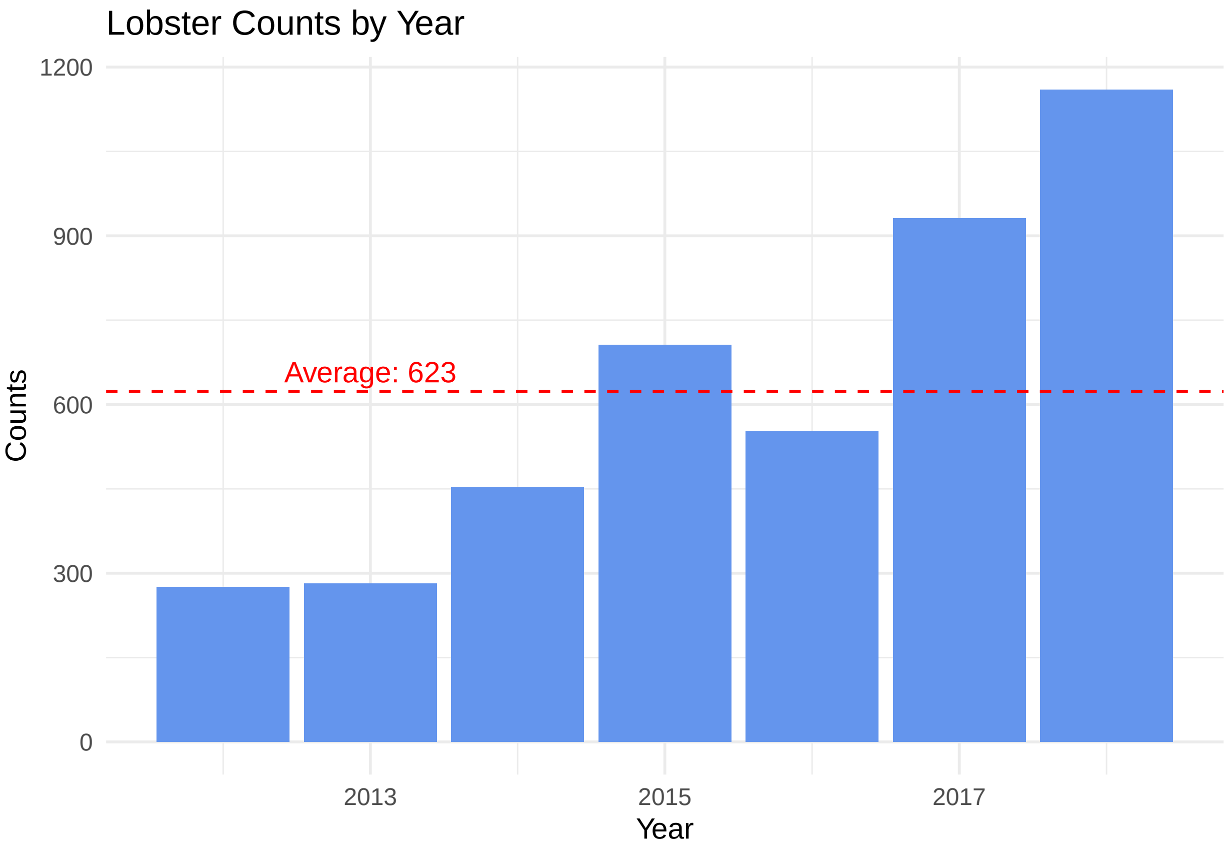


```
# Beeswarm plot grouped by MPA status
ggplot(spiny_counts, aes(x = mpa, y = counts, color = mpa)) +
    geom_boxplot(alpha = 0) +
    geom_beeswarm() +
    theme(legend.position =  "none") +
    labs(title = "Lobster counts by MPA status")
```

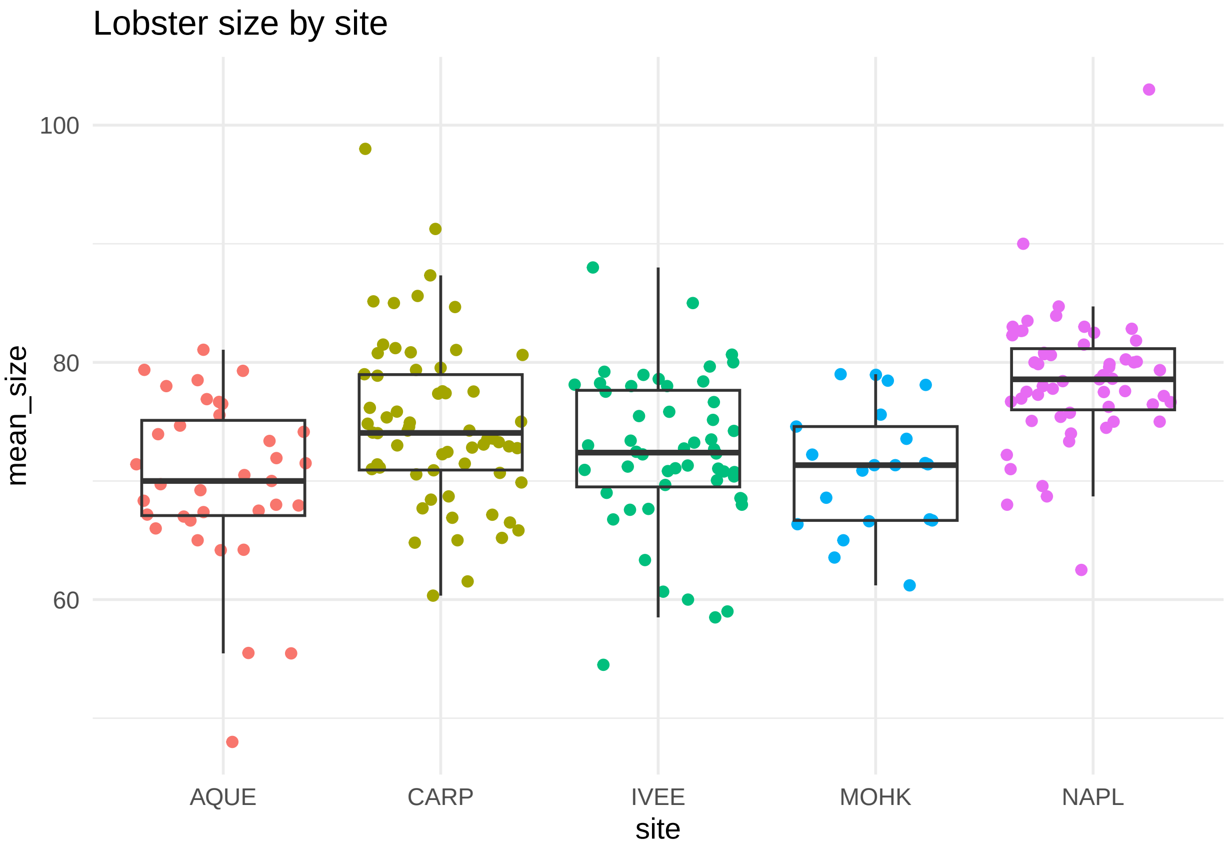


```
# calculate average by year
mean_counts_per_year <- spiny_counts %>%
  group_by(year) %>%
  summarise(counts_by_year = sum(counts, na.rm = TRUE))

# Histogram grouped by year
ggplot(spiny_counts, aes(x = year, y = counts)) +
  geom_histogram(stat = "identity", fill = "cornflowerblue", binwidth = 1) +
  geom_hline(aes(yintercept = mean(mean_counts_per_year$counts_by_year)), color = "red", linetype = "das
  annotate("text", x = 2013, y = mean(mean_counts_per_year$counts_by_year) + 35,
           label = paste("Average:", round(mean(mean_counts_per_year$counts_by_year), 0)), color = "red
  theme_minimal() +
  labs(title = "Lobster Counts by Year", x = "Year", y = "Counts") +
  theme(legend.position = "none")
```



```r
# Jitter plot grouped by site showing size
ggplot(spiny_counts, aes(x = site, y = mean_size)) +
    geom_jitter(aes(colour = site)) +
    theme_minimal() +
    labs(title = "Lobster size by site") +
    theme(legend.position = "none") +
    geom_boxplot(alpha = 0)
```



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

```
# USE: gt_summary::tbl_summary()
spiny_counts %>% tbl_summary(by = treat) %>%
    as_kable()
```

Characteristic	**0** N = 133	**1** N = 119
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%)
year		
2012	19 (14%)	17 (14%)
2013	19 (14%)	17 (14%)
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%)
transect		
1	21 (16%)	14 (12%)
2	21 (16%)	14 (12%)
3	21 (16%)	14 (12%)

Characteristic	**0** N = 133	**1** N = 119
4	14 (11%)	14 (12%)
5	14 (11%)	14 (12%)
6	14 (11%)	14 (12%)
7	14 (11%)	14 (12%)
8	7 (5.3%)	14 (12%)
9	7 (5.3%)	7 (5.9%)
counts	10 (6, 19)	13 (5, 28)
mean_size	73 (68, 77)	77 (71, 80)
Unknown	15	12
mpa		
MPA	0 (0%)	119 (100%)
non_MPA	133 (100%)	0 (0%)

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

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

```
## MODEL INFO:
## Observations: 252
## Dependent Variable: counts
## Type: OLS linear regression
##
## Standard errors:OLS
## ------------------------------------------------
##                      Est.   S.E.   t val.      p
## ----------------- ------- ------ -------- ------
## (Intercept)         14.93   1.57     9.52   0.00
## treat                5.03   2.28     2.20   0.03
## ------------------------------------------------
```

The coefficient on the intercept (14.93) represents the lobster count when the there is no treatment (`treat` = 0). The coefficient on the predictor(treat) is how much the lobster count increases with treatment. So lobster count increases by about 5 lobster from non-MPA to MPA.

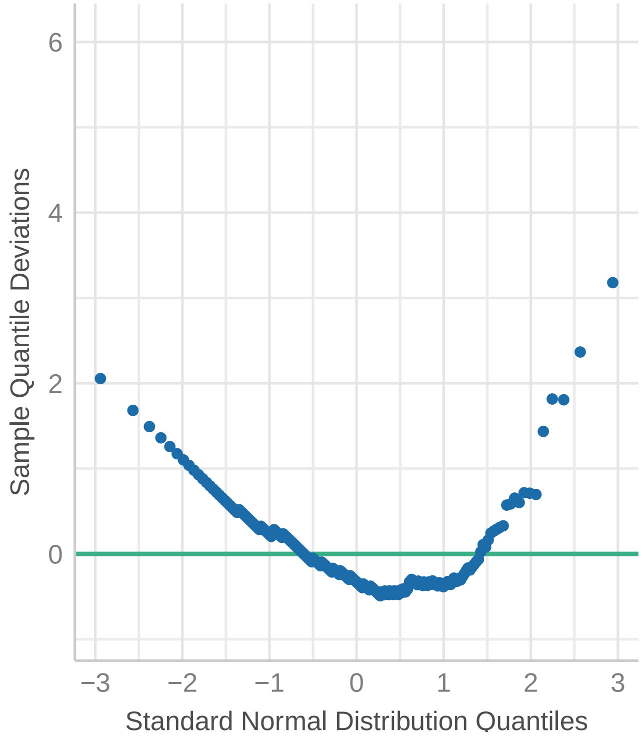
c. Check the model assumptions using the `check_model` function from the `performance` package

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

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

Normality of Residuals

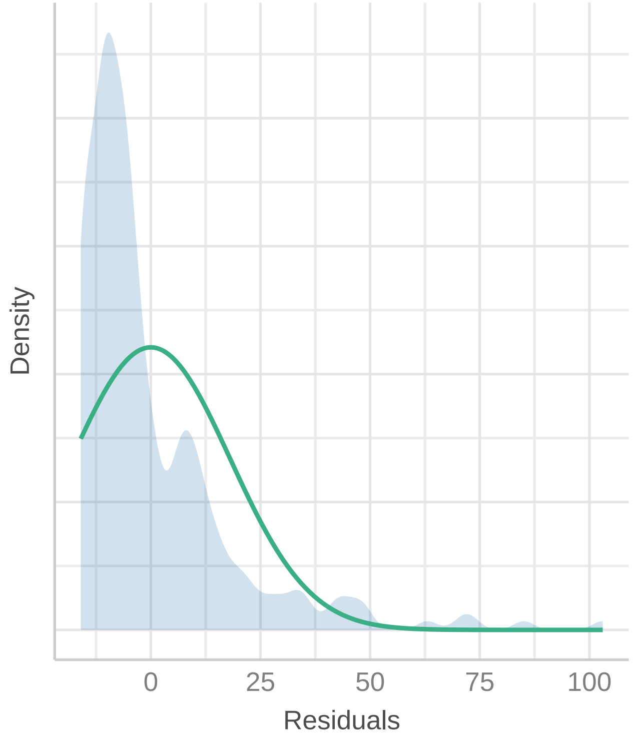
Dots should fall along the line



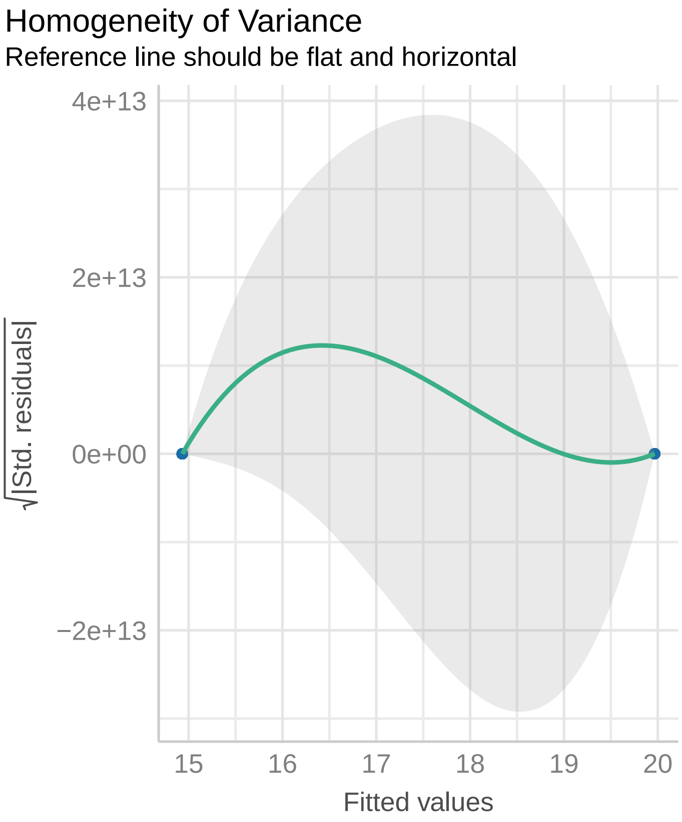
```r
check_model(m1_ols, check = "normality")
```

Normality of Residuals

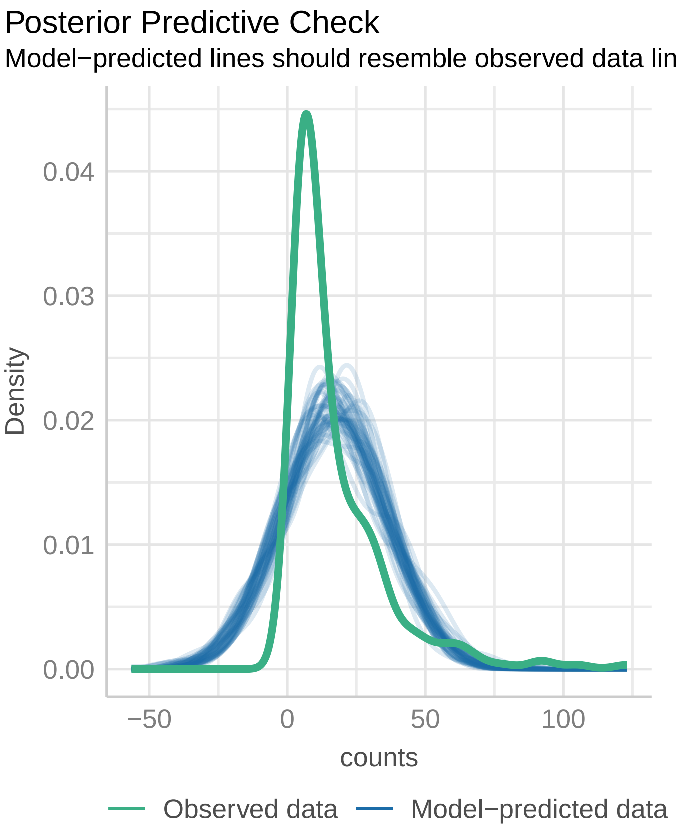
Distribution should be close to the normal curve



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



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



After checking the model, we can see our model is not consistent with the actual data distribution. All the check_model results showed that the actual data residuals are not normal and the variance is not homogeneous. We may be getting this because the data does not follow a linear model.

Step 5: Fitting GLMs

a. Estimate a Poisson regression model using the `glm()` function

b. Interpret the predictor coefficient in your own words. Use full sentences and write your interpretation of the results to be as clear as possible to a non-academic audience.

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

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

```
#HINT1: Incidence Ratio Rate (IRR): Exponentiation of beta returns coefficient which is interpreted as

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

# Poisson model
m2_pois <- glm(counts ~ treat, family = poisson, data = spiny_counts)

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

```
## MODEL INFO:
## Observations: 252
## Dependent Variable: counts
## Type: Generalized linear model
##   Family: poisson
##   Link function: log
##
## Standard errors:MLE
## -----------------------------------------------
##                       Est.   S.E.   z val.      p
## ----------------- ------ ------ -------- ------
## (Intercept)           2.70   0.02   120.48   0.00
## treat                 0.29   0.03     9.56   0.00
## -----------------------------------------------
```

The coefficient on the intercept (2.70) represents the lobster count when the there is no treatment (`treat` = 0). The coefficient on the predictor(treat) is how much the lobster count increases with treatment. So lobster count increases by about 0.3 lobster from non-MPA to MPA.

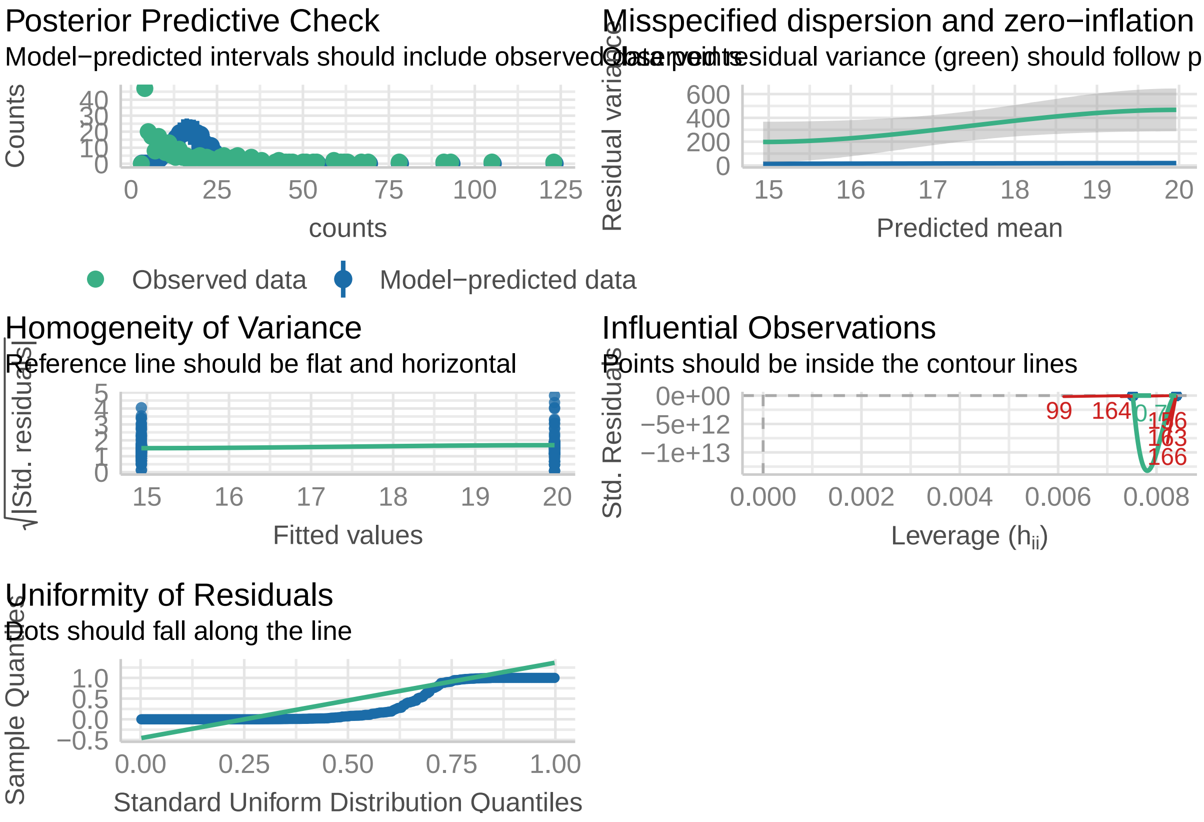
Dispersion is the different between the mean and variance of the data and in a Poisson distribution they should be equal. Overdisperson is when the variance is greater than the mean, so the data is too widely spread out.

Both the intercept and treatment coefficients are much smaller for this model than for the OLS model. The change in significance of the treatment effect is due to the fact that OLS assumes normally distributed errors with leads to larger significance.

e. Check the model assumptions. Explain results.

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

```
check_model(m2_pois)
```



```
check_overdispersion(m2_pois)
```

```
## # Overdispersion test
##
##        dispersion ratio =   18.179
##   Pearson's Chi-Squared = 4544.739
##                 p-value =  < 0.001
```

```
check_zeroinflation(m2_pois)
```

```
## Model has no observed zeros in the response variable.
```

```
## NULL
```

After checking the model assumptions we can see the model does not fit the data well. The posterior predictive check showed the model predicted data and observed data were not similar. The misspecified dispersion and zero inflation test showed the residual variance did not follow the predicted mean. The homogeneity of variance test did show that the variance was fairly homgeneous throughout. The influential observations and uniformity of residuals test also howed the model was not a good fit.

The overdispersion test deteced overdispersion, meaning our variance is significantly greter than our mean. The zero-inflation test showed us that there are no zeros in the counts column so that is not affecting our 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.

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

```
# NOTE: The `glm.nb()` function does not require a `family` argument
# Negative binomial model
m3_nb <- glm.nb(counts ~ treat, data = spiny_counts)

print(summ(m3_nb, model.fit = FALSE))
```

```
## MODEL INFO:
## Observations: 252
## Dependent Variable: counts
## Type: Generalized linear model
##   Family: Negative Binomial(1.5081)
##   Link function: log
##
## Standard errors:MLE
## ------------------------------------------------
##                     Est.   S.E.   z val.      p
## ------------------ ------ ------ -------- ------
## (Intercept)          2.70   0.07    36.49   0.00
## treat                0.29   0.11     2.71   0.01
## ------------------------------------------------
```

```
check_overdispersion(m3_nb)
```

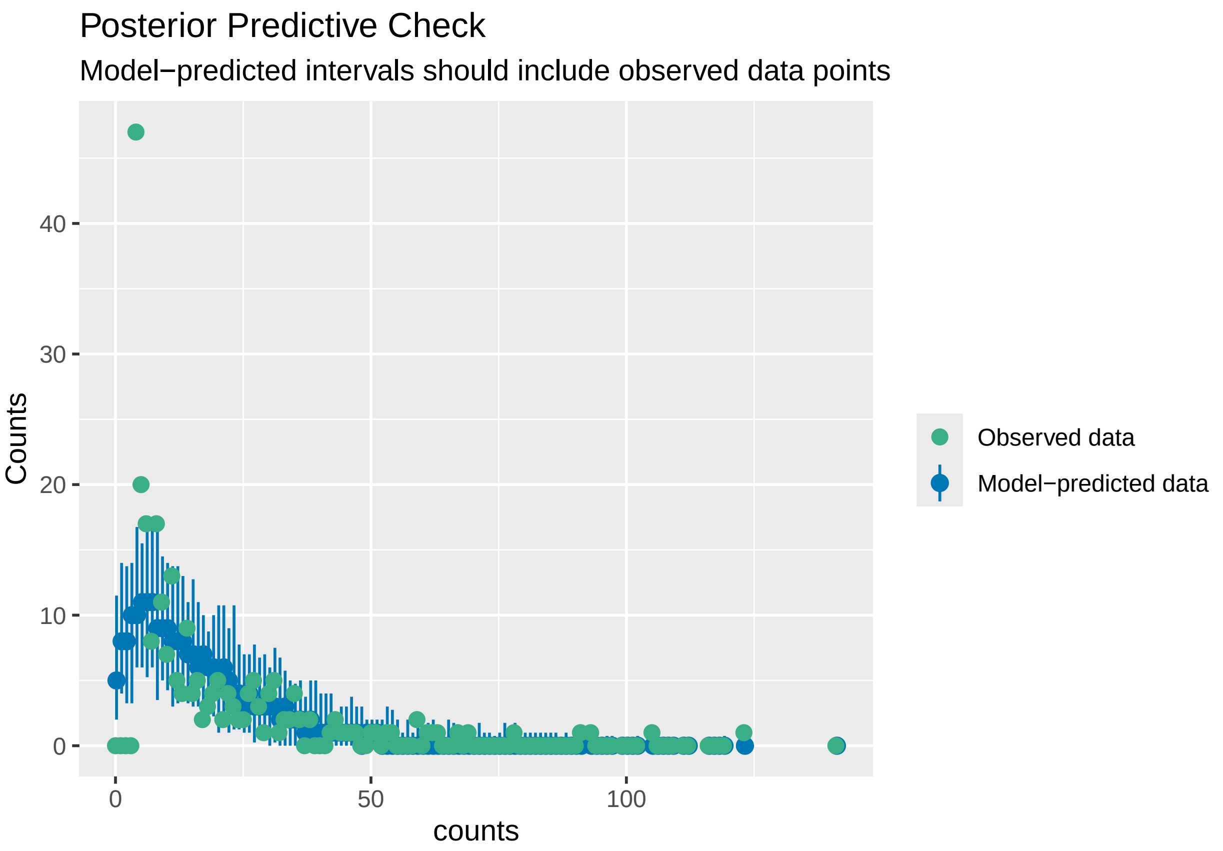
```
## # Overdispersion test
##
##   dispersion ratio = 1.476
##            p-value = 0.016
```

```
check_zeroinflation(m3_nb)
```

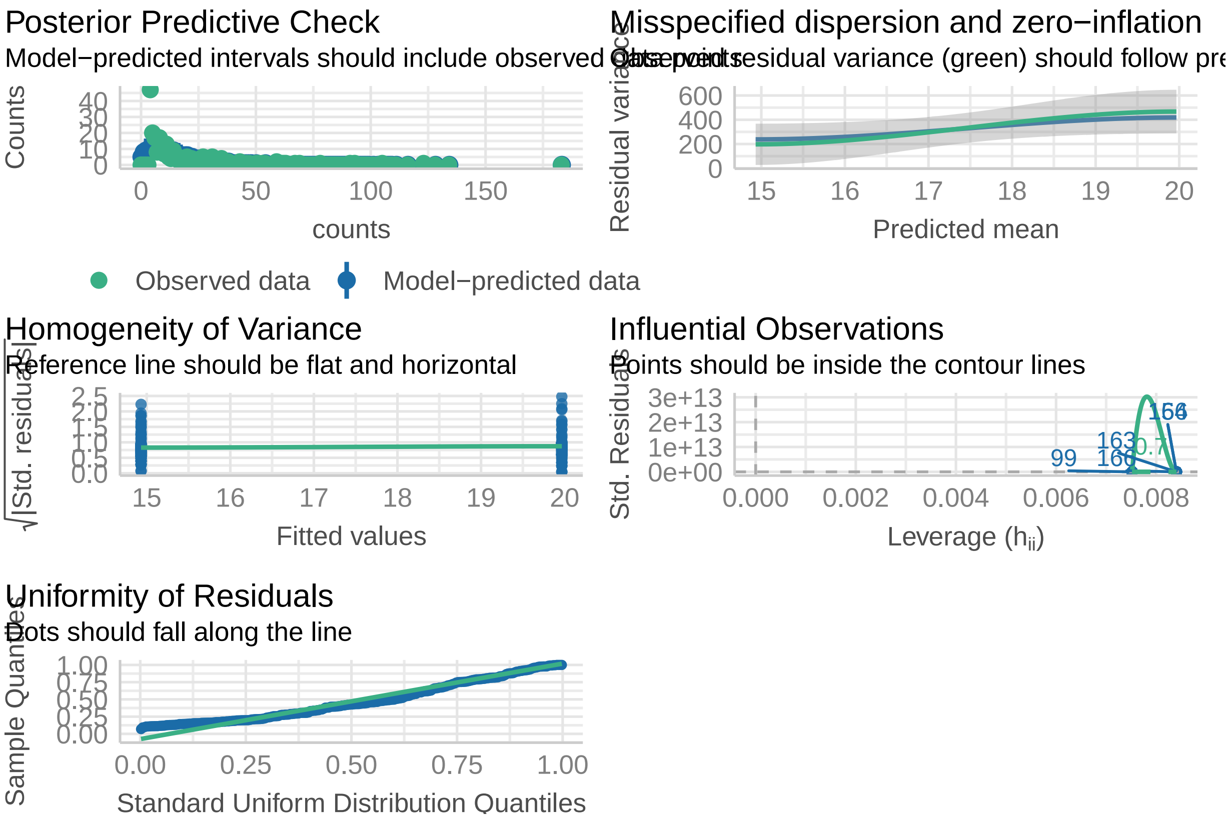
```
## Model has no observed zeros in the response variable.
```

```
## NULL
```

```
check_predictions(m3_nb)
```



```
check_model(m3_nb)
```



The coefficient on the intercept (2.70) represents the lobster count when the there is no treatment (`treat` = 0). The coefficient on the predictor(treat) is how much the lobster count increases with treatment. So lobster count increases by about 0.3 lobster from non-MPA to MPA. These number are similar to the numbers for the poisson model. However, the standard deviation and z values differ from the poisson model.

A negative binomial model is appropriate when the variance is significantly greater than the mean (overdispersion) which we saw is happening when we used the poisson model.

All of the checks for this model show it is a much better fit than the 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.

```
# View models together
print(export_summs(m1_ols, m2_pois, m3_nb,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none"))
```

```
##
##                                    OLS        Poisson        NB
##
##                  (Intercept)    14.93 ***     2.70 ***     2.70 ***
##                              (1.57)      (0.02)      (0.07)
##                 treat         5.03 *      0.29 ***    0.29 **
##                              (2.28)      (0.03)      (0.11)
##
##                 *** p < 0.001; ** p < 0.01; * p <
##                 0.05.
##
## Column names: names, OLS, Poisson, NB
```

The results for the Poisson and Negative Binomial models are very similar. However, the results from the OLS model differ greatly from them. So the treatment effect is robust in regards to Poisson and NB but not OLS.

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?

```r
# Negative binomial model with fixed effects for year and interaction term
ff_counts <- spiny_counts %>%
    mutate(year=as_factor(year))

m5_fixedeffs <- glm.nb(
    counts ~
        treat +
        year +
        treat*year,
    data = ff_counts)

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

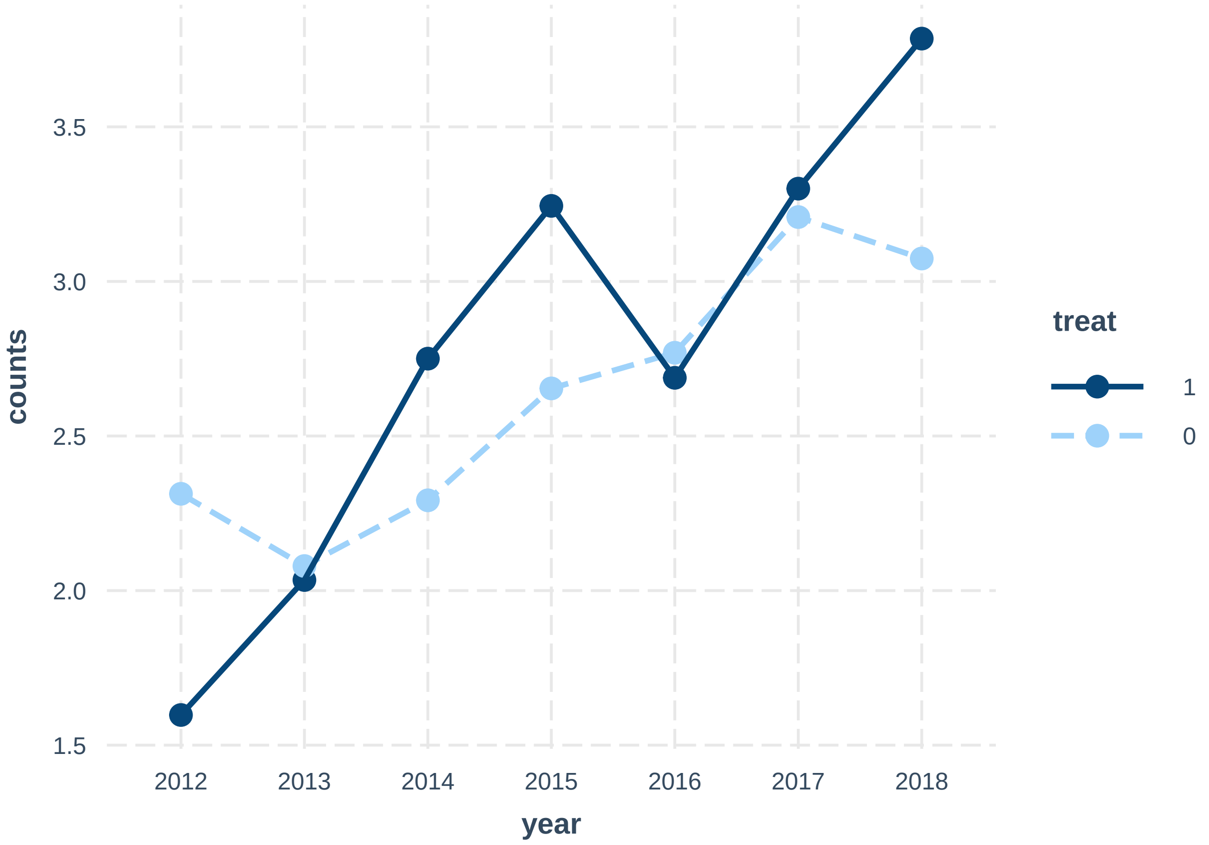
```
## MODEL INFO:
## Observations: 252
## Dependent Variable: counts
## Type: Generalized linear model
##   Family: Negative Binomial(2.4165)
##   Link function: log
##
## Standard errors:MLE
## ------------------------------------------------------
##                             Est.   S.E.   z val.      p
## -------------------- ------- ------ -------- ------
## (Intercept)                 2.31   0.16    14.08   0.00
## treat                  -0.72   0.25    -2.85   0.00
## year2013               -0.23   0.24    -0.99   0.32
## year2014               -0.02   0.23    -0.09   0.93
## year2015                0.34   0.23     1.49   0.14
## year2016                0.46   0.23     2.00   0.05
## year2017                0.90   0.23     3.97   0.00
## year2018                0.76   0.23     3.36   0.00
## treat:year2013          0.67   0.35     1.91   0.06
## treat:year2014          1.17   0.34     3.41   0.00
## treat:year2015          1.31   0.34     3.84   0.00
## treat:year2016          0.63   0.34     1.86   0.06
## treat:year2017          0.81   0.34     2.39   0.02
## treat:year2018          1.43   0.34     4.24   0.00
## ---------------------------------------------------
```

The intercept coefficient represents the count (log transformed) for the year 2012 untreated. The treat coefficient show the year 2012 change from untreated to treated. And the year coefficients show how the counts change from the baseline year to that year untreated. And the treat:year coefficients show how the counts change from the baseline to that year with treatment, added to the treat coefficient. It makes sense that treat is negative because it represents the 2012 year, when the MPAs were first established, so there wasnt much effect yet. However, for later years you need to include the interaction term, which are all positive, so increase the -0.72.

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.

```r
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
```

The counts on average increase with treatment. The year that it decreases the most is 2012, which is when the MPAs were first established so there was not much effect yet. In 2013 there is only then a slight decrease. The counts then greatly increase with treatment in 2014 and 2015. It slightly decreases in 2016 but then goes back to increasing with the next two years.

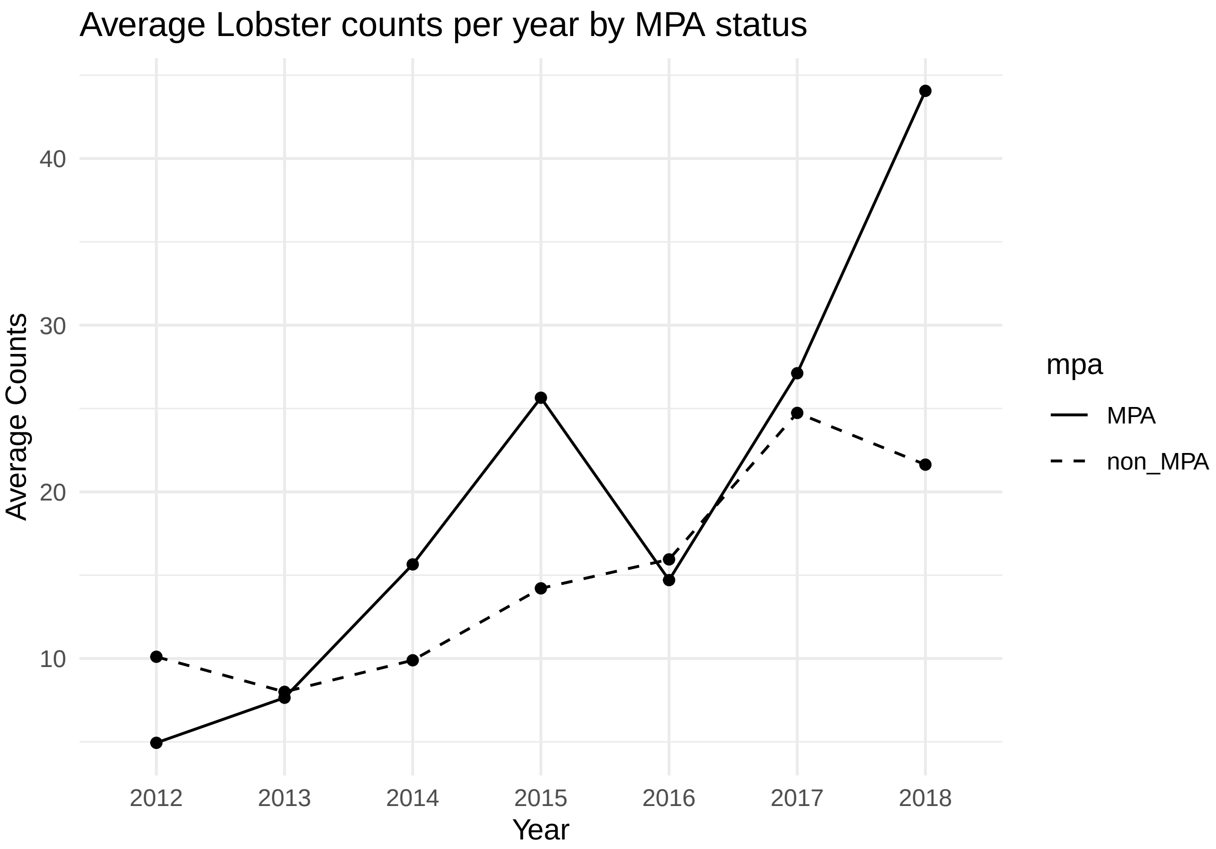
g. Using `ggplot()` create a plot in same style as the previous `interaction plot`, but displaying the original scale of the outcome variable (lobster counts). This type of plot is commonly used to show how the treatment effect changes across discrete time points (i.e., panel data).

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

# View average lobster counts per year by treatment
plot_counts <- spiny_counts %>%
    group_by(year, mpa) %>%
    summarise(mean_count = mean(counts)) %>%
    mutate(year = as_factor(year))

plot_counts  %>%
    ggplot(aes(x = year, y = mean_count, group = mpa, linetype = mpa)) +
    geom_line() +
    geom_point() +
    labs(title = "Average Lobster counts per year by MPA status",
         color = "treat",
         x = "Year",
         y = "Average Counts") +
    theme_minimal() +
    scale_linetype_manual(values = c("solid", "dashed"))
```



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 do believe spillover is likely in this research context. I believe this is likely due to the proximity of the locations. Protecting lobsters in one location will likely effect the amount of lobsters in a close location.

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

Spillover is an issue because then the treatment is not only affecting the treated unit. It is also affecting the "untreated" unit.

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

I believe protecting the lobsters in the treated areas has a positive impact on lobster counts in other areas. We can see from the graphs that the lobster counts increased in the non-MPAs as well as in the 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 I do not believe this is a reasonable assumption in this context. I believe the treatment of the MPAs had an effect on the surrounding areas and therefore the non-MPAS. However, I do believe all units received the same treatment.
    2) Excludability assumption I do not believe this is a reasonable assumption. I believe there could be other ways lobster count is effected besides MPA designation.

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)
