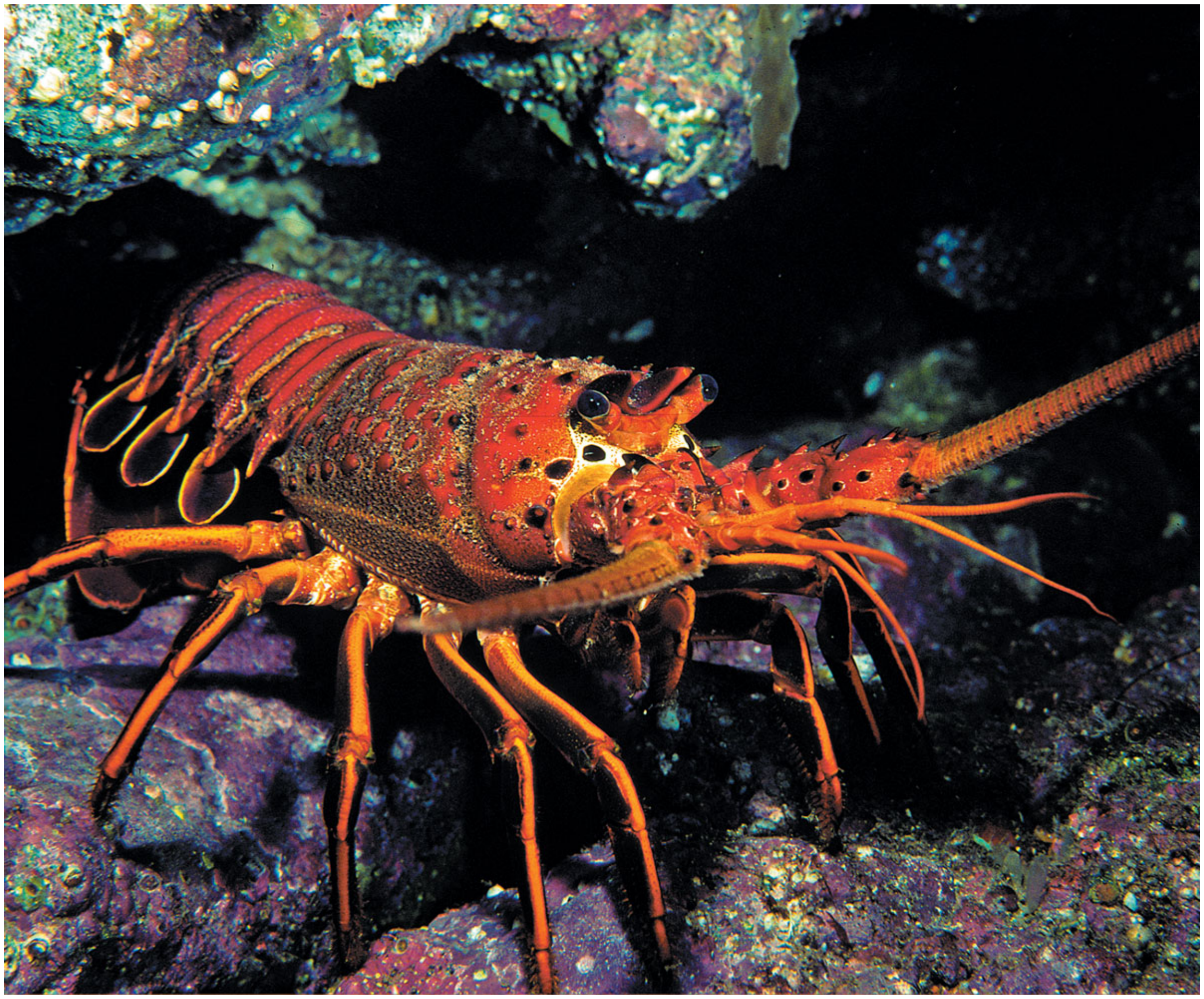
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

Kaiju Morquecho

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 (YOUR NAME): ***Kaiju Morquecho***

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(here)
```

```
## here() starts at /Users/kaiju/MEDS/EDS241/EDS241_HW1
```

```r
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(estimatr)
library(performance)
library(jtools)
library(gt)
library(gtsummary)
library(MASS) ## NOTE: The `select()` function is masked. Use: `dplyr::select()` ##
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:gtsummary':
##
##     select
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(interactions)
library(ggridges)
library(ggrepel)
library(dplyr)
```

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.

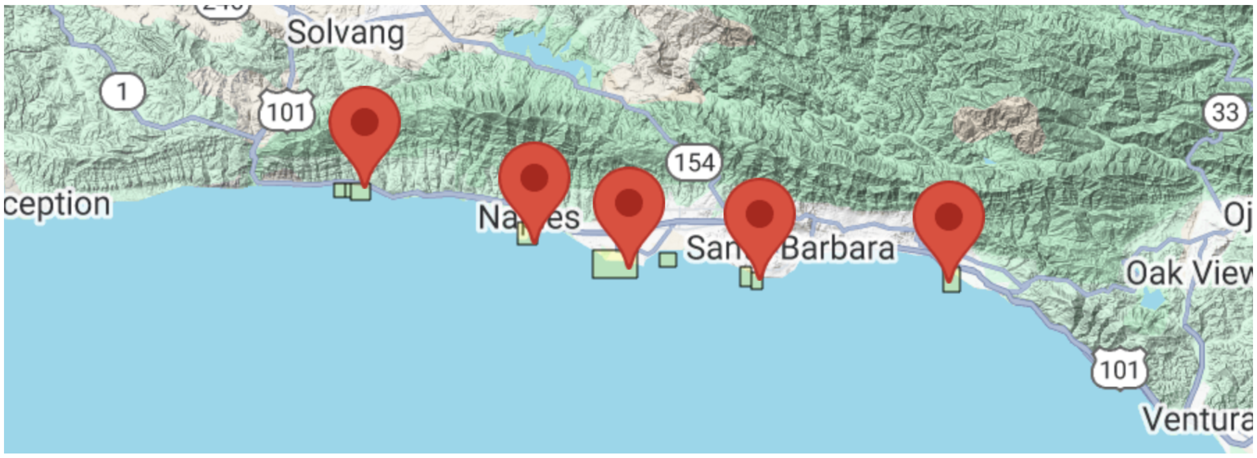
Introduction

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!).

It is unlikely that the comparison between the controls and the treatment sites in this case are centris paribus, at least not perfectly. There are factors at play that we likely have not accounted for. Human factors such as population density inland (near a reef), and environmental factors such as differences in water temperature, food availability, and recent weather events that affect the overall health of the reefs and of its lobsters. Furthermore, we do not know if the conditions within both of the MPAs are themselves centris paribus, nor do we know what the fishing pressures and lobster health conditions were before the reefs were designated as MPAs. These are all factors pointing toward selection bias.

There are, however, characteristics that make this analysis promising - the 5 reefs geographically relatively close to one another. They likely experience similar ocean currents and temperatures and face similar weather conditions. These are significant in helping mitigate selection bias and make modeling more feasible.

Step 2: Read & wrangle data

a. Read in the raw data. Name the data.frame (`df`) `rawdata`

```r
rawdata <- read_csv(here("data","spiny_abundance_sb_18.csv"),
                    na = "-99999")
```

```
## Rows: 4362 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): SITE, REPLICATE
## dbl  (7): YEAR, MONTH, TRANSECT, SIZE_MM, COUNT, NUM_AO, AREA
## date (1): DATE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
sum(is.na(rawdata))
```

```
## [1] 350
```

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

```r
# HINT: check for coding of missing values (`na = "-99999"`)

rawdata <- clean_names(rawdata)
```

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"
```

```r
tidydata <- rawdata %>%
  mutate(reef = factor(site,
                       levels = c("AQUE","CARP","MOHK","IVEE","NAPL"),
                       labels = c("Arroyo Quemado","Carpinteria","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`)!

```r
spiny_counts <- tidydata %>%
  group_by(site,year,transect) %>%
  summarize(mean_size = mean(size_mm,
                             na.rm = TRUE),
            counts = sum(count,
                         na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'site', 'year'. You can override using the
## `.groups` argument.
```

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`

```r
#HINT(d): Use `group_by()` & `summarize()` to provide the total number of lobsters observed at each sit

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

spiny_counts <- spiny_counts %>%
  mutate(mpa = case_when(site %in% c("IVEE","NAPL") ~ "MPA",
                         .default = "non_MPA")) %>%
  mutate(treat = case_when(mpa == "MPA" ~ 1,
                           .default = 0)) %>%
  ungroup()
```

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`)

```
dim(tidydata)
```

```
## [1] 4362   11
```

```
dim(spiny_counts)
```

```
## [1] 252   7
```

```
head(spiny_counts)
```

```
## # A tibble: 6 x 7
##   site   year transect mean_size counts mpa     treat
##   <chr> <dbl>    <dbl>     <dbl>  <dbl> <chr>   <dbl>
## 1 AQUE   2012        1      64.2      5 non_MPA     0
## 2 AQUE   2012        2      66        9 non_MPA     0
## 3 AQUE   2012        3     NaN        0 non_MPA     0
## 4 AQUE   2012        4      74.1      9 non_MPA     0
## 5 AQUE   2012        5      76.9     11 non_MPA     0
## 6 AQUE   2012        6     NaN        0 non_MPA     0
```

```
head(tidydata)
```

```
## # A tibble: 6 x 11
##    year month date       site  transect replicate size_mm count num_ao  area
##   <dbl> <dbl> <date>     <chr>    <dbl> <chr>       <dbl> <dbl>  <dbl> <dbl>
## 1  2012     8 2012-08-20 IVEE         1 A              NA     0      0   300
## 2  2012     8 2012-08-20 IVEE         1 B              NA     0      0   300
## 3  2012     8 2012-08-20 IVEE         1 C              NA     0      0   300
## 4  2012     8 2012-08-20 IVEE         1 D              NA     0      0   300
## 5  2012     8 2012-08-20 IVEE         2 A              NA     0      0   300
## 6  2012     8 2012-08-20 IVEE         2 B              NA     0      0   300
## # i 1 more variable: reef <fct>
```

```
site_mean <- spiny_counts %>%
  group_by(mpa) %>%
  summarize(mean_counts = mean(counts))
```

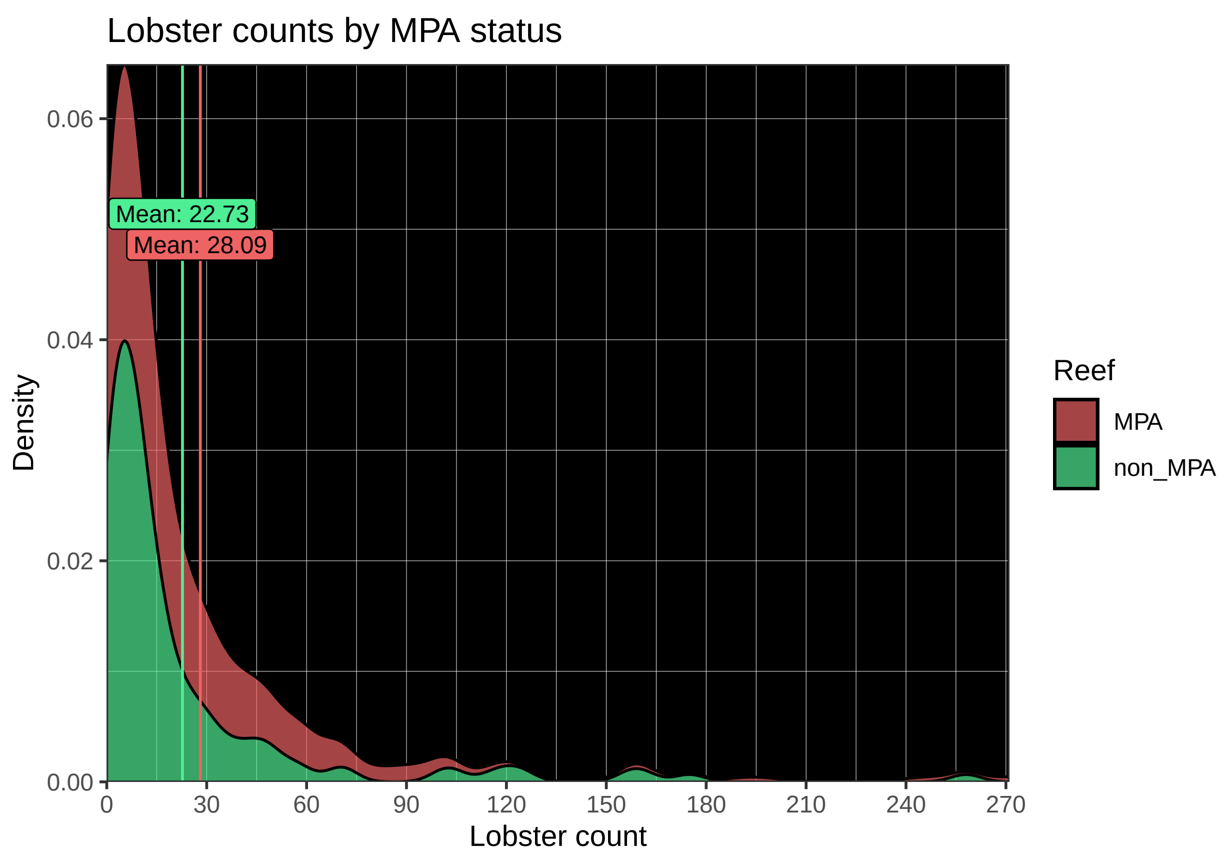
b. We will focus on the variables `count`, `year`, `site`, and `treat(mpa)` to model lobster abundance. Create the following 4 plots using a different method each time from the 6 options provided. Add a layer (`geom`) to each of the plots including informative descriptive statistics (you choose; e.g., 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**:

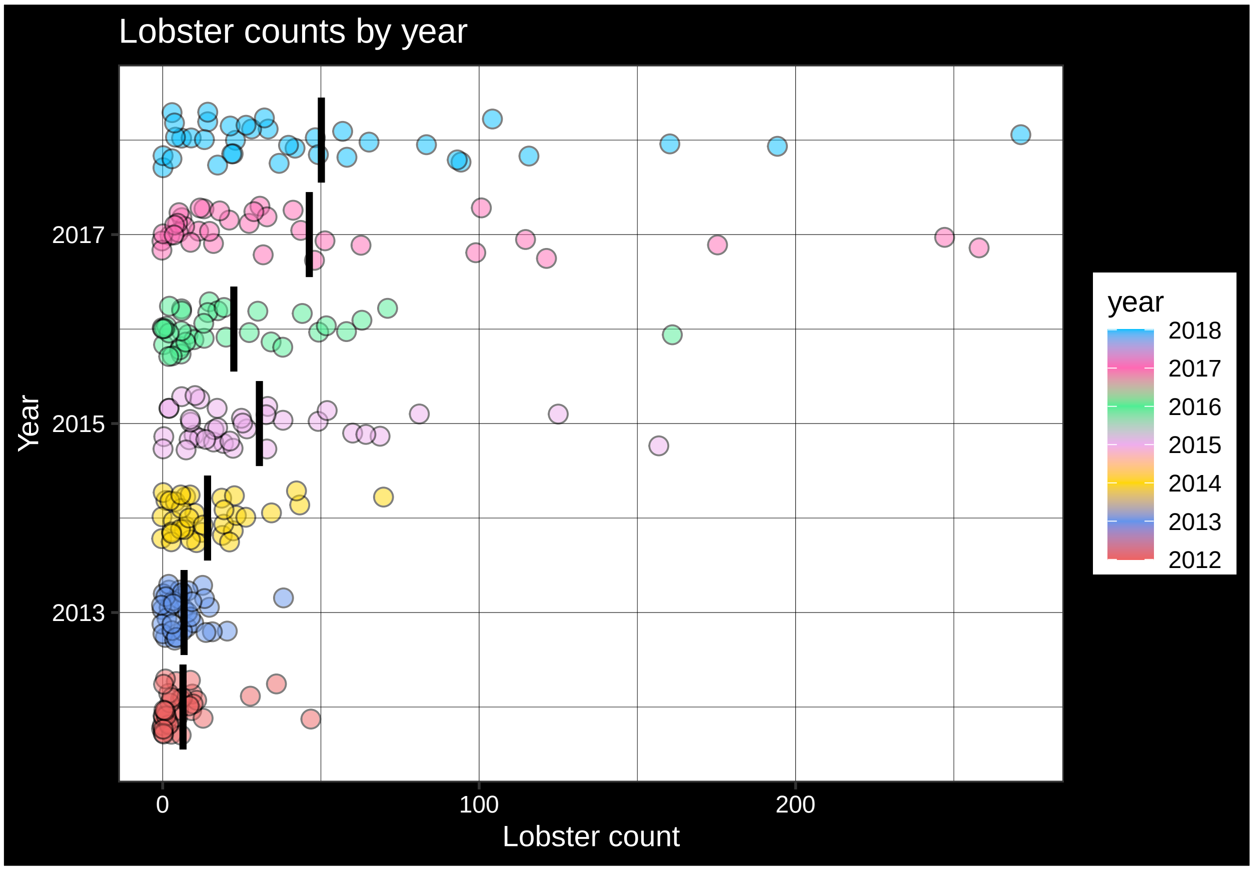
1) grouped by MPA status

```r
density_plot <- spiny_counts %>%
  ggplot(aes(x = counts, fill = mpa)) +
  geom_density(alpha = 0.7,
               position = "stack") +
  geom_vline(data = site_mean,
             aes(xintercept = mean_counts,
                 color = mpa),
             show.legend = FALSE) +
  geom_label(data = site_mean,
             aes(x = mean_counts,
                 y = 0.05,
                 label = paste0("Mean:", " ", round(mean_counts, digits = 2)),
                 vjust = c("top","bottom")),
             size = 3,
             show.legend = FALSE) +
  labs(title = "Lobster counts by MPA status",
       x = "Lobster count",
       y = "Density",
       fill = "Reef") +
  scale_fill_manual(values =c("indianred2",
                              "seagreen2")) +
  scale_color_manual(values =c("indianred2",
                              "seagreen2")) +
  scale_x_continuous(expand = c(0,0),
                     breaks = seq(0,280,30)) +
  scale_y_continuous(expand = c(0,0)) +
  theme_bw() +
  theme(
        panel.grid = element_line(color = "white",
                                  linewidth = 0.02),
        panel.background = element_rect(fill = "black"))



print(density_plot)
```



2) grouped by year

```r
jitter_plot <- spiny_counts %>%
  ggplot(aes(x = year, y = counts,
             fill = year)) +
  geom_jitter(width = 0.3,
              alpha = 0.5,
              shape = 21,
              size = 3) +
  stat_summary(fun = "mean",
               geom = "crossbar",
               color = "black",
               fill = "white") +
  labs(title = "Lobster counts by year",
       x = "Year",
       y = "Lobster count") +
  scale_fill_gradientn(colors = c("indianred2",
                                  "cornflowerblue",
                                  "gold1",
                                  "plum2",
                                  "seagreen2",
                                  "hotpink",
                                  "deepskyblue1")) +
  coord_flip() +
  theme_bw() +
  theme(text = element_text(color = "white"),
        axis.text = element_text(color = "white"),
        legend.text = element_text(color = "black"),
        legend.title = element_text(color = "black"),
        panel.grid = element_line(color = "black",
                                  linewidth = 0.1),
        panel.background = element_rect(fill = "white"),
        plot.background = element_rect(fill = "black"))

print(jitter_plot)
```

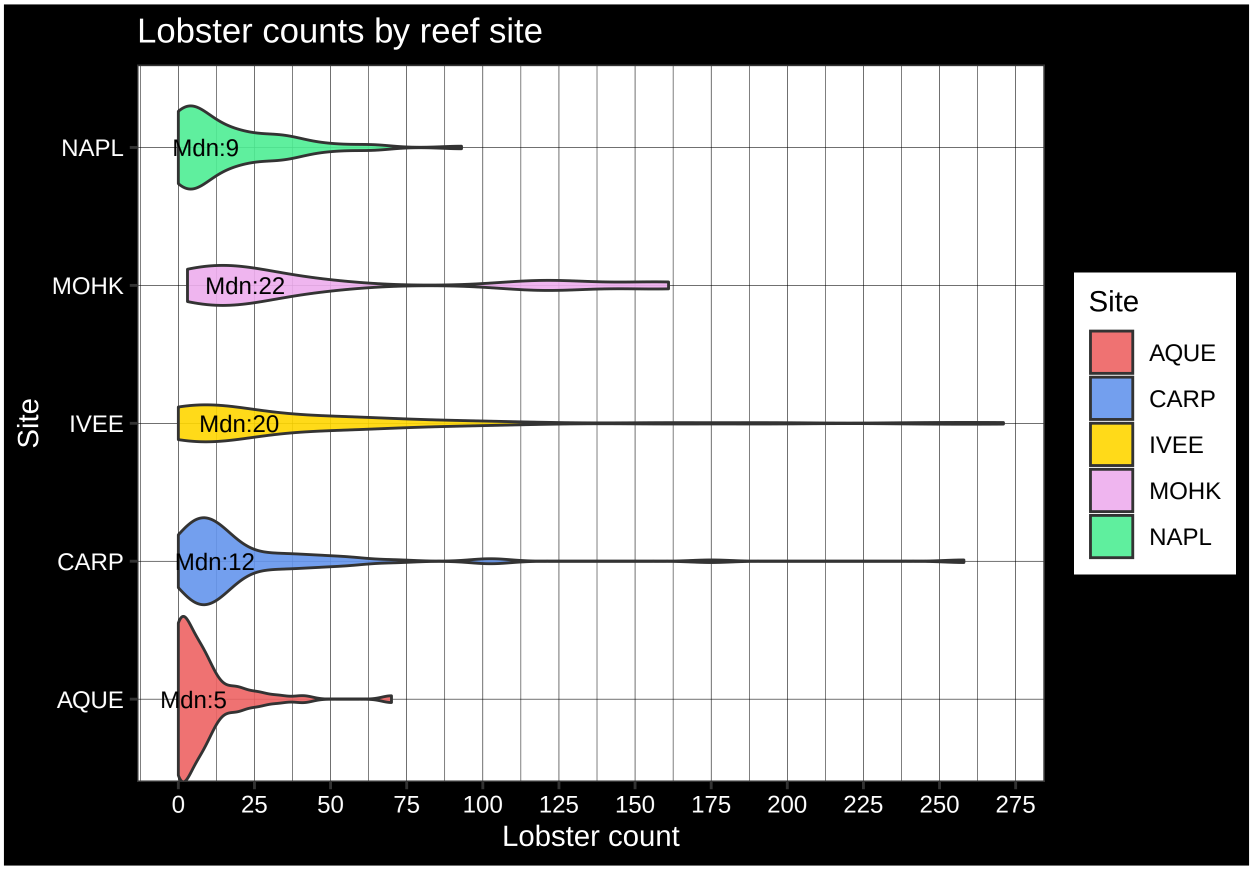


3) grouped by site

```r
violin_plot <- spiny_counts %>%
  ggplot(aes(x = site, y = counts,
             fill = site)) +
  geom_violin(alpha = 0.9,
              width = 1.2) +
  theme_bw() +
  theme(text = element_text(color = "white"),
        axis.text = element_text(color = "white"),
        legend.text = element_text(color = "black"),
        legend.title = element_text(color = "black"),
        panel.grid = element_line(color = "black",
                                  linewidth = 0.1),
        panel.background = element_rect(fill = "white"),
        plot.background = element_rect(fill = "black")) +
  labs(title = "Lobster counts by reef site",
       fill = "Site",
       x = "Site",
       y = "Lobster count") +
  stat_summary(
    aes(label = paste0("Mdn:","",round(..y.., 1))),
    size = 3,
    fun = "median",
    geom = "text",
    colour = "black",
    show.legend = FALSE) +
  scale_fill_manual(values = c(
    "indianred2", "cornflowerblue", "gold1", "plum2", "seagreen2"
  )) +
  scale_color_manual(values = c(
    "indianred2", "cornflowerblue", "gold1", "plum2", "seagreen2"))+
  scale_y_continuous(breaks = seq(0,300,25)) +
  coord_flip()


print(violin_plot)
```

```
## Warning: The dot-dot notation (`..y..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(y)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `position_dodge()` requires non-overlapping x intervals.
```



Create a plot of lobster **size** :

4) You choose the grouping variable(s)!

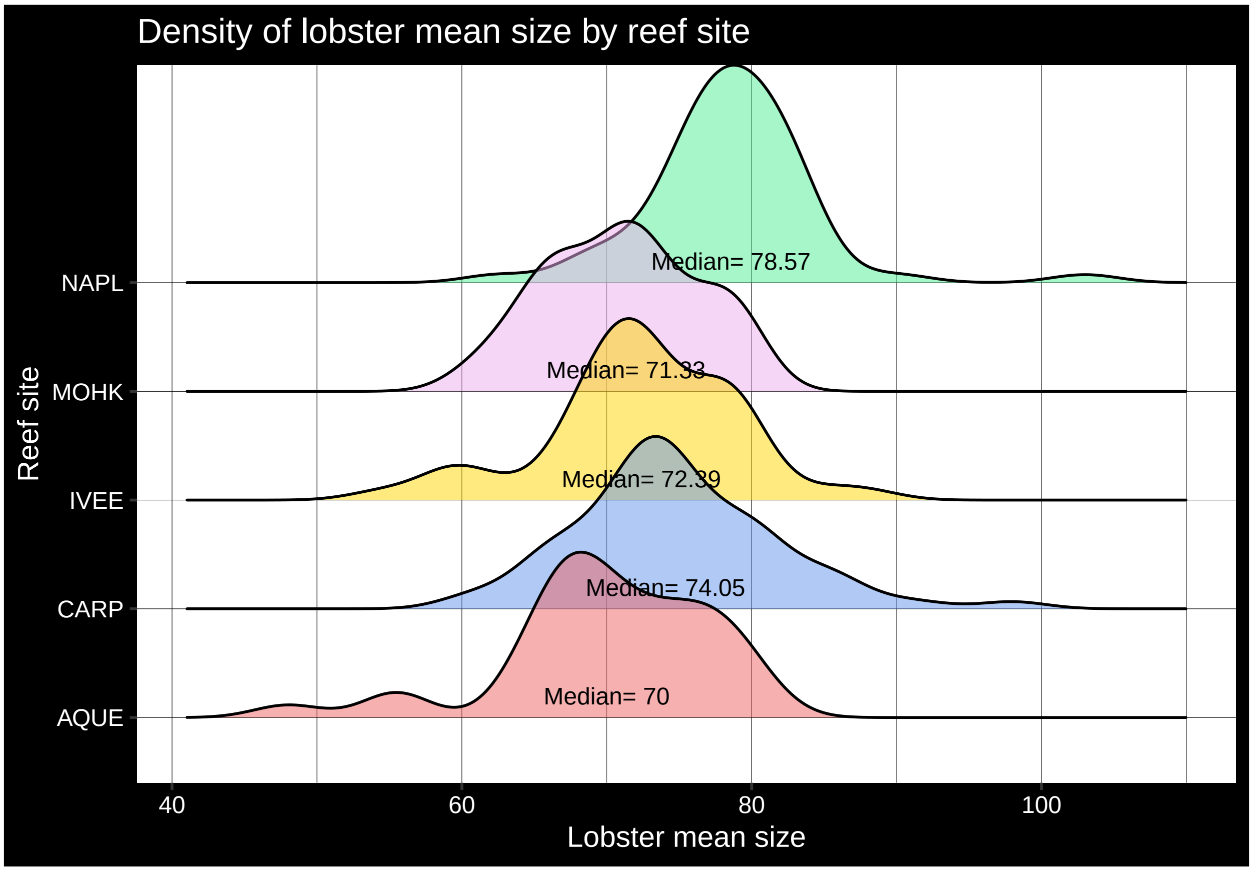
```
medians <- spiny_counts %>%
  group_by(site) %>%
  summarize(median_size = median(mean_size, na.rm = TRUE))

spiny_counts %>%
  ggplot(aes(x = mean_size, y = site, fill = site)) +
  geom_density_ridges(
    alpha = 0.5,
    size = 1,
    scale = 2
  ) +
  geom_text(data = medians,
            aes(x = median_size,
                y = site,
                label = paste0("Median="," ",(round(median_size, 2)))),
            color = "black",
            size = 3,
            vjust = -0.75) +
  theme(text = element_text(color = "white"),
        axis.text = element_text(color = "white"),
        legend.position = "none",
        panel.grid = element_line(color = "black",
```

```r
                                  linewidth = 0.1),
        panel.background = element_rect(fill = "white"),
        plot.background = element_rect(fill = "black")) +
  labs(
    title = "Density of lobster mean size by reef site",
    x = "Lobster mean size",
    y = "Reef site",
    fill = "Site"
  ) +
  scale_fill_manual(values = c(
    "indianred2", "cornflowerblue", "gold1", "plum2", "seagreen2"
  ))
```

```
## Warning in geom_density_ridges(alpha = 0.5, size = 1, scale = 2): Ignoring
## unknown parameters: ‘size‘
```

```
## Picking joint bandwidth of 2.32
```

```
## Warning: Removed 27 rows containing non-finite outside the scale range
## (‘stat_density_ridges()‘).
```



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

Characteristic	**MPA** N = 119[1]	**non_MPA** N = 133[1]
site		
AQUE	0 (0%)	49 (37%)
CARP	0 (0%)	63 (47%)
IVEE	56 (47%)	0 (0%)
MOHK	0 (0%)	21 (16%)
NAPL	63 (53%)	0 (0%)

[1] n (%)

```r
tbl_summary(data = spiny_counts,
            by = "mpa",
            statistic = list(all_continuous() ~ "{mean}"),
            include = "site")
```

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

```r
# NOTE: We will not evaluate/interpret model fit in this assignment (e.g., R-square)

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

tbl_1 <- summ(model = m1_ols,
     model.fit = FALSE)

print(tbl_1)
```

```
## MODEL INFO:
## Observations: 252
## Dependent Variable: counts
## Type: OLS linear regression
##
## Standard errors:OLS
## -------------------------------------------------
##                      Est.   S.E.   t val.      p
## ----------------- -------- ------ -------- ------
## (Intercept)         22.73   3.57     6.36   0.00
## treat                5.36   5.20     1.03   0.30
## -------------------------------------------------
```

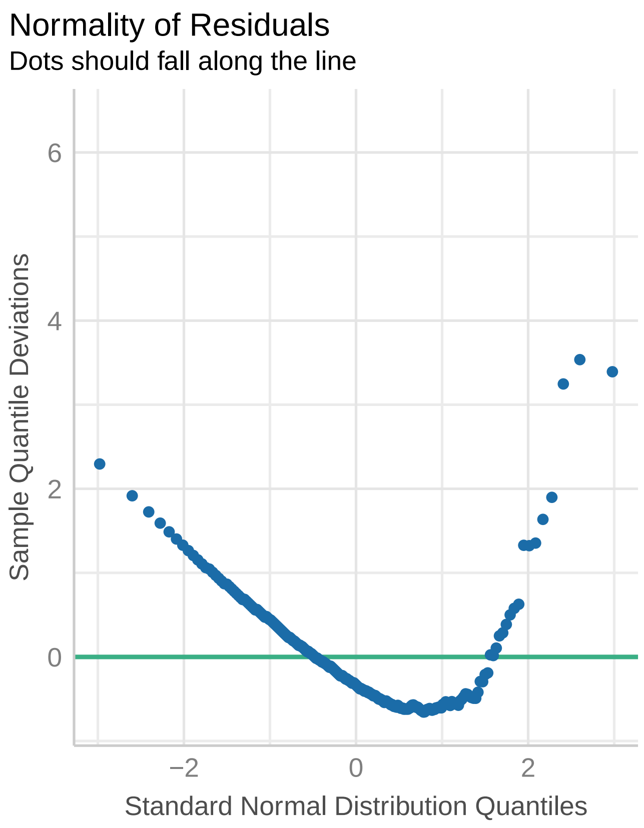
b. Interpret the intercept & predictor coefficients *in your own words*. Use full sentences and write your interpretation of the regression results to be as clear as possible to a non-academic audience.

The intercept = 22.73 is what the lm model estimates the lobster count will be when the treatment is not being applied (when the site is not an MPA site). The predictor coeffient = 5.36 tells us that when the treatment IS applied (the site is an MPA) the estimated lobster count will increase from 22.73 by 5.36. The predictor tells us that sites that are MPAs have a positive effect on lobster counts (lobsters are more abundant).

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

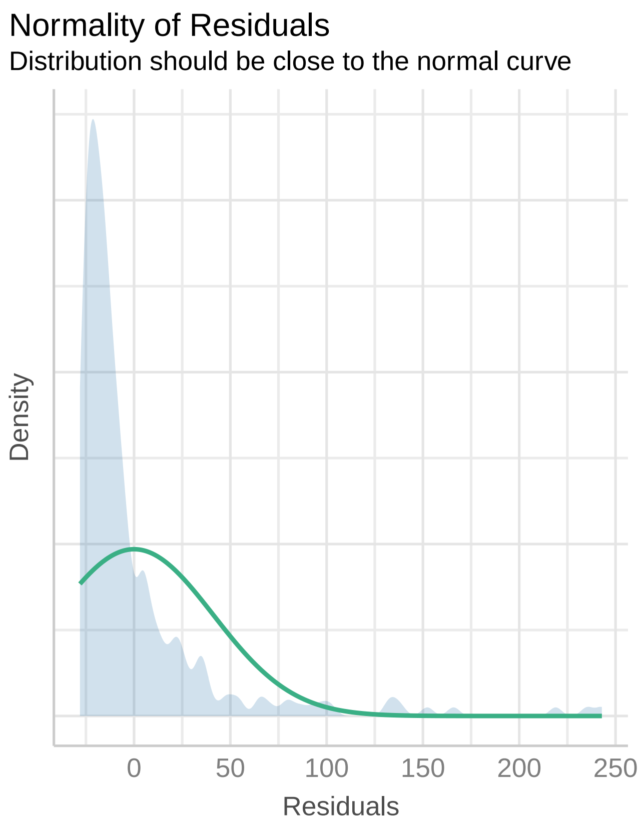
d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

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



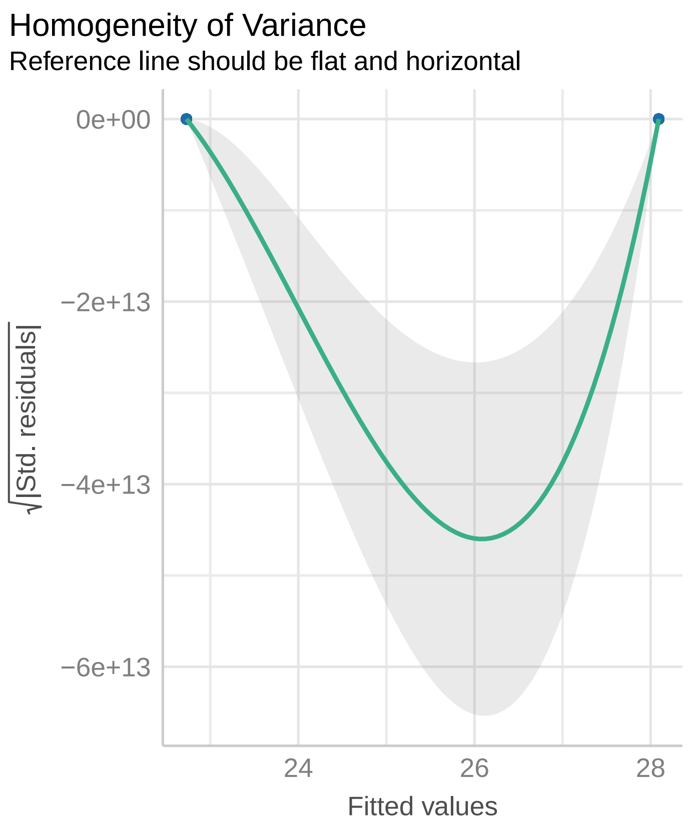
QQ plot explanation The straight line represents a normal distribution and the model's residuals diverge significantly from it. There is a noticeable pattern in our residuals when the distribution of normal residuals should have no pattern/be randomly distributed. This tells us we are more than likely using the wrong model for our data and we are not capturing the effect of possible patterns in it.

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



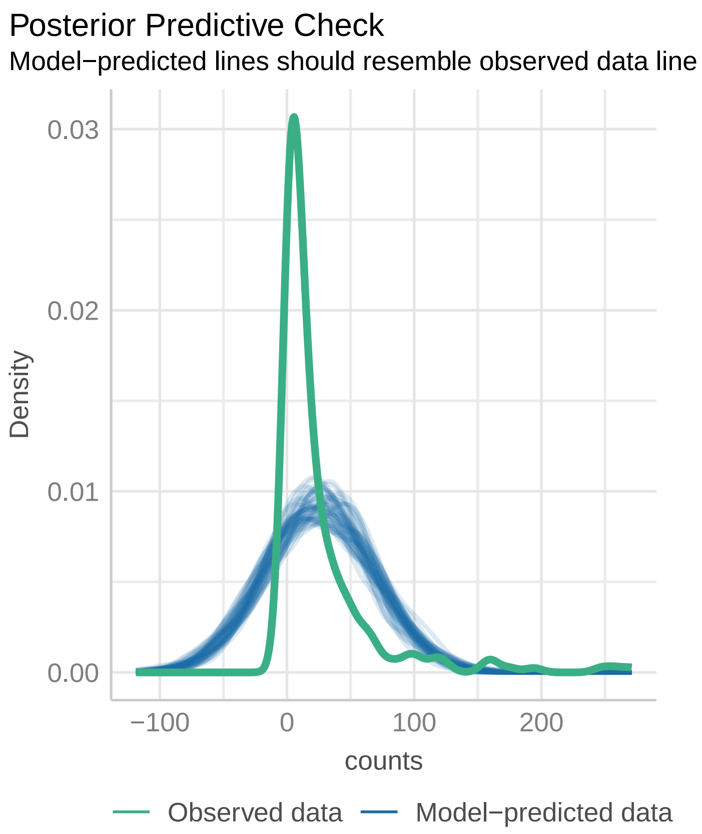
Normality of residuals density plot explanation Plot indicates a departure from normality and it allows us to see HOW our model predictions deviate from a normal distribution. Our data is likely skewed (not symmetrically distributed) and has a tail.

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



Homogeneity of variance plot explanation This plot shows that the residuals of our fitted values do not display constant variance across all levels, this means that the model does not accurately capture the variability of all levels in our data.

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



Posterior predictive check explanation This plot tells us that the distribution of our model data predictions do not match those that were actually observed. It speaks to a poor model fit. The model does not seem to be accurately representing and replicating the complexity of the observed data.

Step 5: Fitting GLMs

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

```
m2_pois <- glm(counts ~ treat,
                data = spiny_counts,
                family = poisson)

exp(coef(m2_pois))
```

```
## (Intercept)       treat
##   22.729323    1.235956
```

```
summary(m2_pois)
```

```
##
## Call:
## glm(formula = counts ~ treat, family = poisson, data = spiny_counts)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.12366    0.01819 171.744   <2e-16 ***
## treat        0.21184    0.02510   8.441   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10438  on 251  degrees of freedom
## Residual deviance: 10366  on 250  degrees of freedom
## AIC: 11366
##
## Number of Fisher Scoring iterations: 6
```

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 estimates how the treatment affects the lobster count of any given site. Meaning, in a reef site that is not an MPA the lobster count is estimated to be aprox. 23. When the treatment is applied (when the reef site is an MPA), model estimates a multiplicative factor of approx. 1.24. This means the model predicts an increase in lobster count when the treatment is applied (when the reef is an MPA)

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

Dispersion is the spread/distribution of variability around the mean. A poisson model makes the assumption that the mean and variance of the data are equal to each other. Overdispersion in this case would mean that the variability predicted by the model is greater than the mean of the data. This means that the model may not be a good fit - it may not account for all the variability occurring in the data. There may be other interactions at play across sites and lobster counts that are unaccounted for.

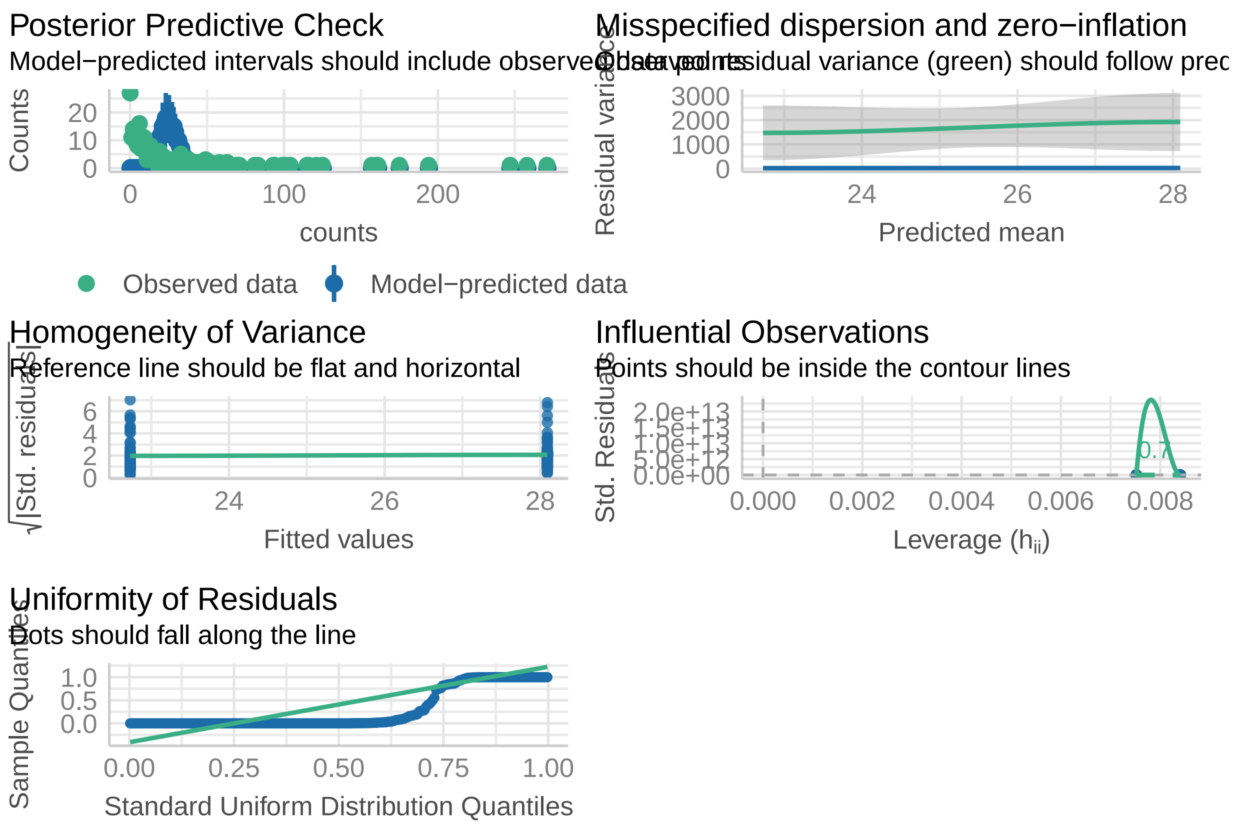
d. Compare results with previous model, explain change in the significance of the treatment effect

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

e. Check the model assumptions. Explain results.

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

```
check_model(m2_pois)
```



```r
check_overdispersion(m2_pois)
```

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

```
## Overdispersion detected.
```

Overdispersion test explanation The test shows a dispersion ratio of 67.033. A dispersion > 1 is considered overdispersion. The p-value of our test is very small, certainly significant. This means that we can confidently say there is more variability in the data than the model can predict.

```r
check_zeroinflation(m2_pois)
```

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

```
## Model is underfitting zeros (probable zero-inflation).
```

Zero-inflation explanation The results of the test say that the model did not predict/account for ANY zeros being present in our data. However, our data did contain 27 observations = 0. This underfitting can result in an inflation of our model estimates (making them inaccurate) and therefore leading to unreliable predictions.

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

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

summary(m3_nb)
```

```
##
## Call:
## glm.nb(formula = counts ~ treat, data = spiny_counts, init.theta = 0.5500333101,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.1237     0.1183  26.399   <2e-16 ***
## treat         0.2118     0.1720   1.232    0.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.55) family taken to be 1)
##
##     Null deviance: 302.18  on 251  degrees of freedom
## Residual deviance: 300.66  on 250  degrees of freedom
## AIC: 2088.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.5500
##           Std. Err.:  0.0466
##
##  2 x log-likelihood:  -2082.5280
```

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

A negative binomial model allows for overdispersion, whereas a regular glm poission model does not. Since we tested for dispersion and found a significantly large dispersion ratio, we need to fit a model that can accommodate for this overdispersion.

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

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

summary(m3_nb)
```

```
##
## Call:
## glm.nb(formula = counts ~ treat, data = spiny_counts, init.theta = 0.5500333101,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.1237     0.1183  26.399   <2e-16 ***
## treat         0.2118     0.1720   1.232    0.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.55) family taken to be 1)
##
##     Null deviance: 302.18  on 251  degrees of freedom
## Residual deviance: 300.66  on 250  degrees of freedom
## AIC: 2088.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.5500
##           Std. Err.:  0.0466
##
##  2 x log-likelihood:  -2082.5280
```

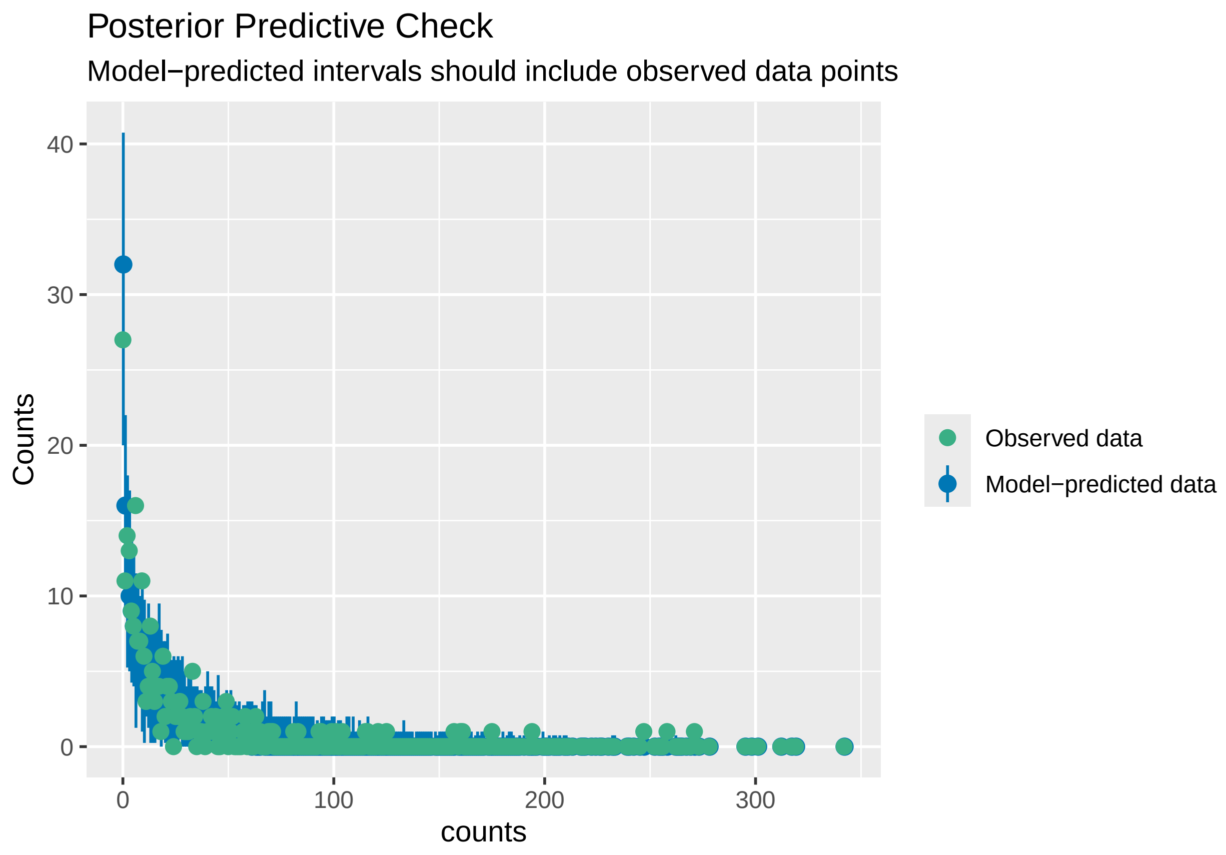
```r
check_overdispersion(m3_nb)
```

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

## No overdispersion detected.
```

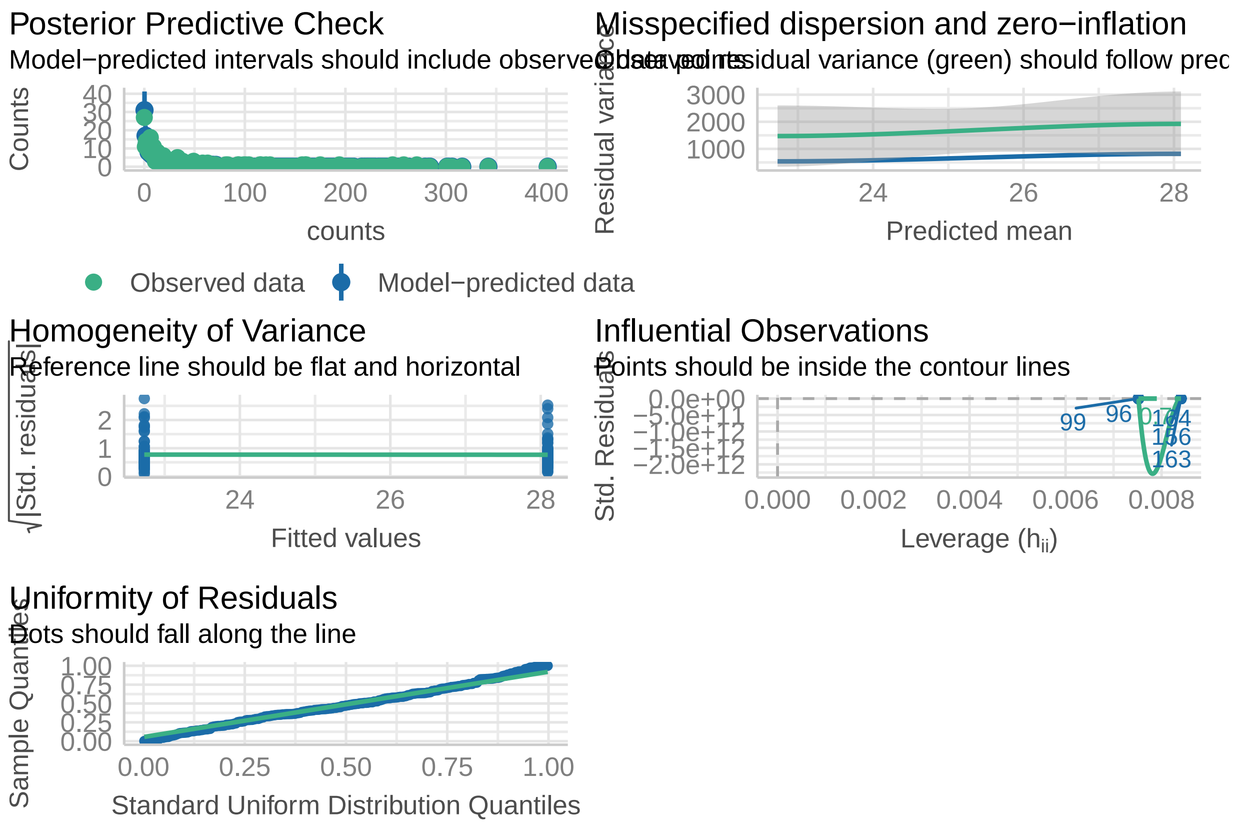
```r
check_zeroinflation(m3_nb)
```

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

## Model is overfitting zeros (p = 0.600).
```

```r
check_predictions(m3_nb)
```



```
check_model(m3_nb)
```



Comparison of m2_pois & m3_nb The estimated coefficients of models m2_pois and m3_nb are very similar. However, m2_pois does not allow for zero-inflation and overdispersion. On the other hand, m3_nb tests did not detect overdispersion and the zero-inflation test detected not an underfitting, but rather a small OVERfitting of zeros.

Despite these encouraging m3_nb results, the m3_nb coefficient p-values indicate that the model does not produce statistically significant estimates. Thus, even if the negative binomial model makes for a better fit, it does not allow us to confidently say that treatment has an effect on lobster count.

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
export_summs(m1_ols, m2_pois, m3_nb,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none")
```

```
## Warning in (function (..., error_format = "({std.error})", error_pos = c("below", : Unrecognized sta
## Try setting ‘statistics‘ explicitly in the call to ‘huxreg()‘

## Warning in build_tabular(ht): Multiple horizontal border widths in a single
## row; using the maximum.
```

	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.

Comparing 3 models The estimated treatment effects are similar between m2_pois and m3_nb, but otherwise vary across models. In the OLS model the treatment coefficient is relatively large, however it is not statistically significant. In the Poisson model the coefficients are smaller and statistically significant but tests revealed both overdispersion and zero-inflation. Lastly, in the NB model tests showed no overdispersion and only a slight overestimation of zero-inflation, but the estimates were not statistically significant. When comparing the models, it is clear that the treatment effect is not stable and not 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 shows how the effect of the treatment is affected by the variable year, what the treatment effect was in the baseline year (2012) and how it changed and compared across following years. In general, the model gives us a more complex and nuanced look at the mechanisms going on in the data. In the baseline year, 2012, MPAs had significantly fewer lobster than non-MPAs. This is also the year when these MPAs were established. And, since lobster counts go up over the years in these MPA sites, it shows that the treatment has had a positive effect on reef sites even if the lobster counts of these aren't dramatically higher than those of non-MPAs.

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

This does make sense - the main effect for treatment is negative in the baseline year (2012). In 2012, the lobster count in now-MPA sites was lower because the benefits of the treatment were not yet tangible.

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

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

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

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.8129)
Link	log

	Est.	S.E.	z val.	p
(Intercept)	2.35	0.26	8.89	0.00
treat	-1.72	0.42	-4.12	0.00
year2013	-0.35	0.38	-0.93	0.35
year2014	0.08	0.37	0.21	0.84
year2015	0.86	0.37	2.32	0.02
year2016	0.90	0.37	2.43	0.01
year2017	1.56	0.37	4.25	0.00
year2018	1.04	0.37	2.81	0.00
treat:year2013	1.52	0.57	2.66	0.01
treat:year2014	2.14	0.56	3.80	0.00
treat:year2015	2.12	0.56	3.79	0.00
treat:year2016	1.40	0.56	2.50	0.01
treat:year2017	1.55	0.56	2.77	0.01
treat:year2018	2.62	0.56	4.69	0.00

Standard errors: MLE

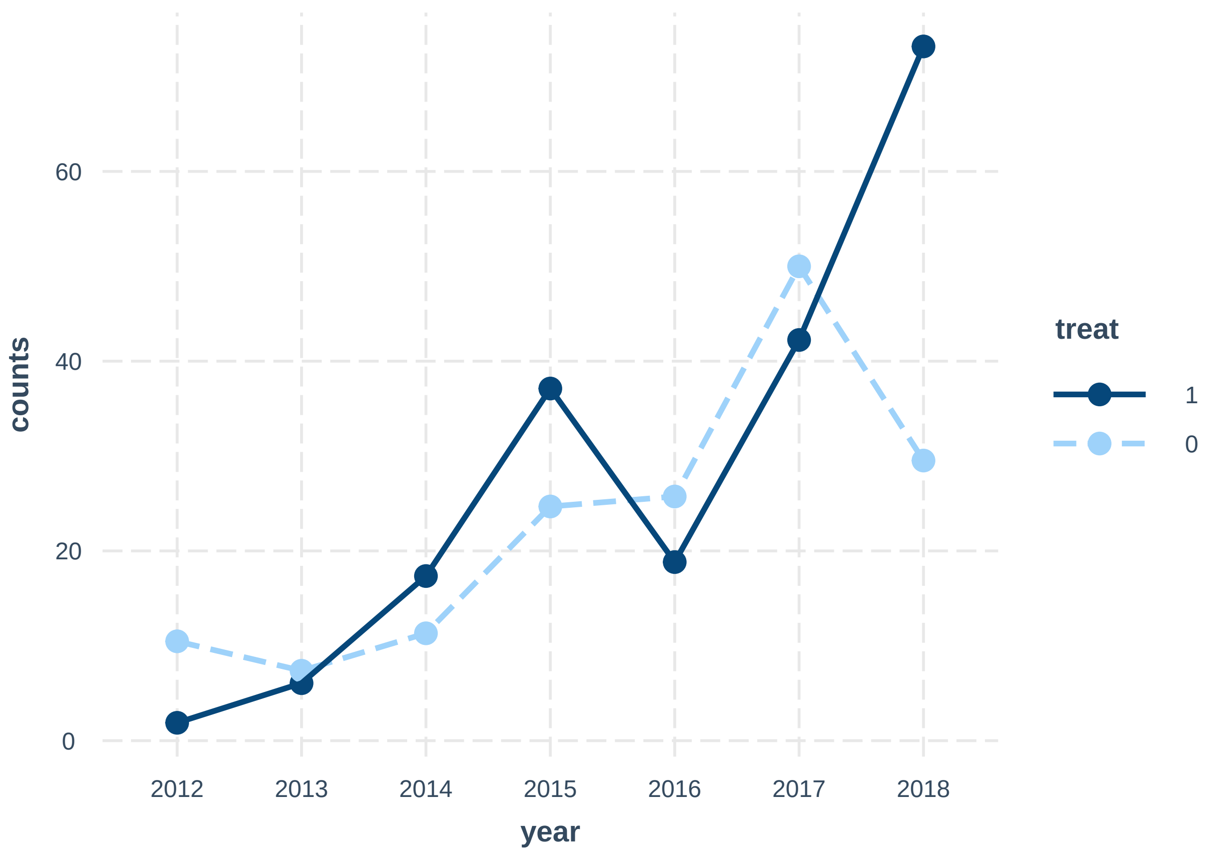
e. Look at the model predictions: Use the `interact_plot()` function from package `interactions` to plot mean predictions by year and treatment status.

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

The interactions plot supports my responses in c and d above.

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

```
## x Detected factor predictor.
## i Plotting with cat_plot() instead.
## i See '?interactions::cat_plot()' for full details on how to specify models
##   with categorical predictors.
## i If you experience errors or unexpected results, try using cat_plot()
##   directly.
```



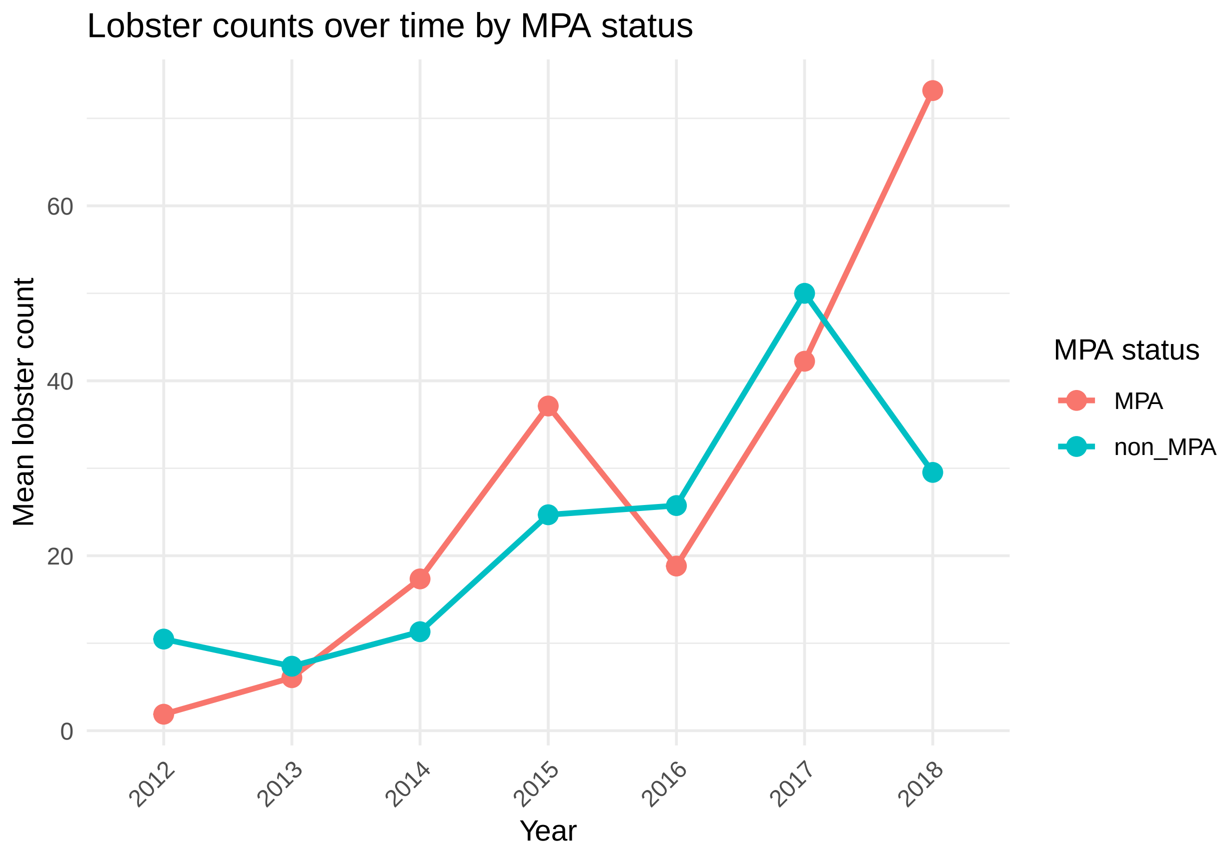
```r
# HINT: Change `outcome.scale` to "response" to convert y-axis scale to counts
```

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

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

```r
# 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) %>%
  summarize(mean_count = mean(counts), .groups = "drop") %>%
  mutate(year = as.factor(year))

plot_counts %>% ggplot(aes(x = year,
                           y = mean_count,
                           group = mpa, color = mpa)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  labs(
    title = "Lobster counts over time by MPA status",
    x = "Year",
    y = "Mean lobster count",
    color = "MPA status"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



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)

Yes, spillover effects are relevant in this research context. Lobster can move without any limits between one MPA to another. When fishing pressures change (in both MPAs and non), lobster respond to these ecological and environmental changes in several ways - they may migrate due to increased competition amongst them, they may move toward other reefs due to temp changes, food availability etc.

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

The spillover effect is an issue for the identification of causal effects because blurs the differences between treated and non-treated reef sites. It complicates our ability to determine if the observed changes in lobster populations are a result of treatment or of unrelated/unintented lobster movement between reef sites.

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

Spillover can lead researchers to underestimate or overestimate the efficacy of the treatment, leading to an inaccurate understanding of the benefits of MPAs. This, in turn, would affect the efforts toward creating more 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
   2) Excludability assumption

Neither of these assumptions are reasonable in the context of the MPA treatment effect estimator. The SUTVA is more than likely violated by the spillover effect since the changes in lobster count do not solely depend on the treatment effect.The lobster count of a non-treated site is affected by the treatment of another site. Lastly, the excludability assumption is violated because other than the spillover effect, there are countless other factors, human and and non-human-related, that affect the health of reefs and the abundance of lobster.

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)
