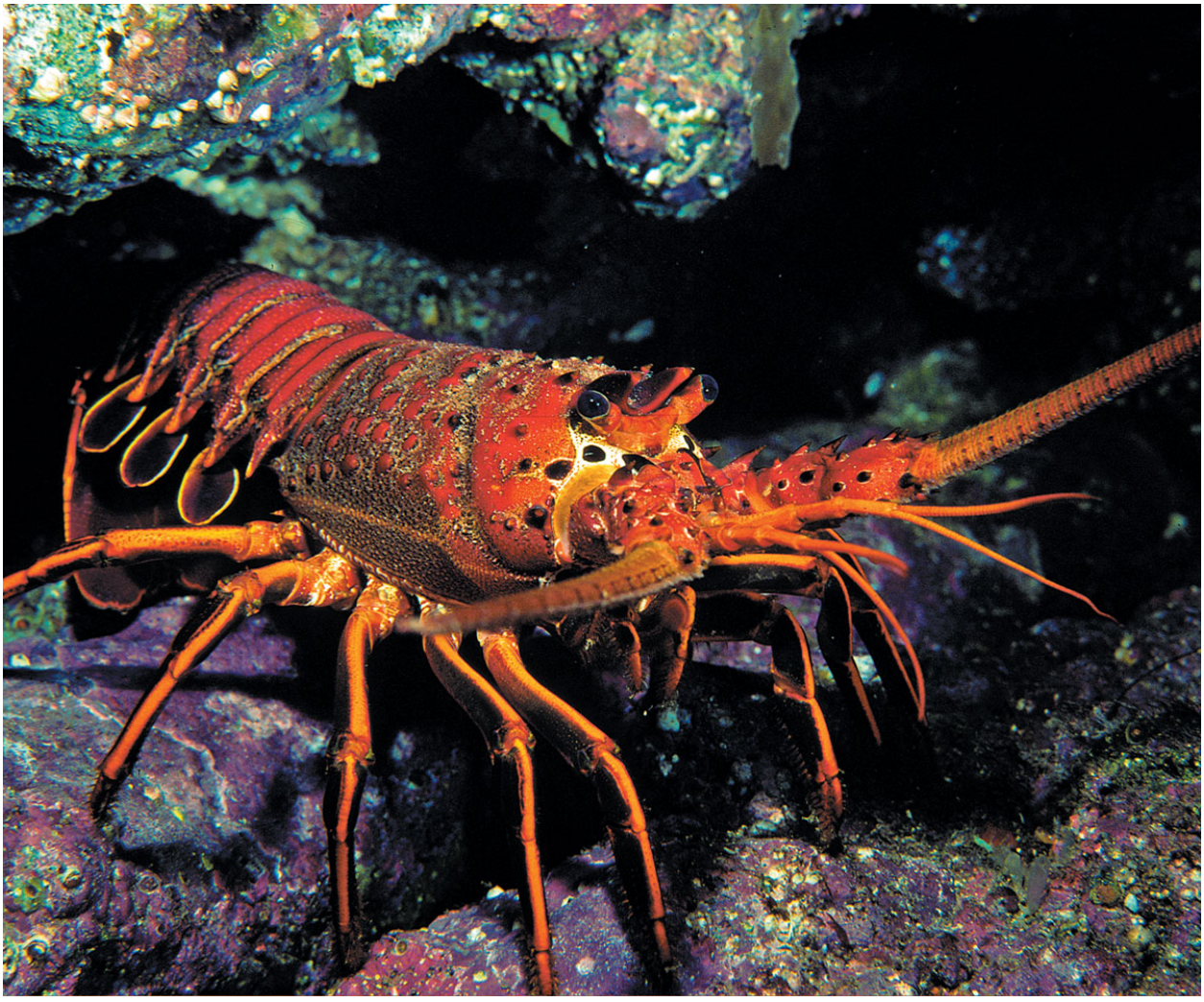
Assignment 1: California Spiny Lobster Abundance (*Panulirus Interruptus*)

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

EDS 241

2025-01-24



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 (Joshua Paul Cohen): _____

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

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.

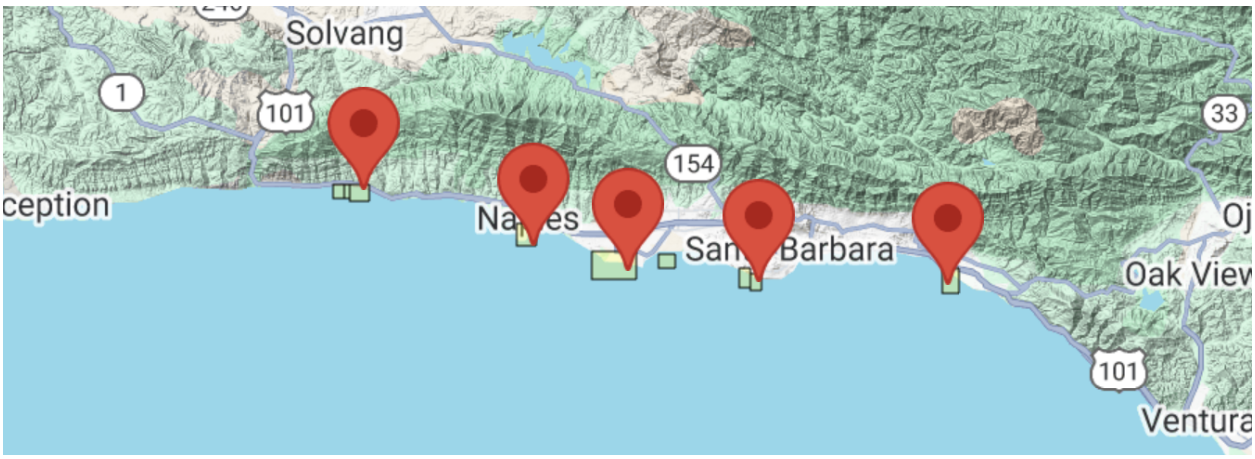
Introduction

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's potentially impossible to create an 'all other things equal' study in a field setting, because there are so many stochastic factors. Some degree of selection bias is likely always present in field studies, as you are selecting a transect to represent an entire ecosystem. But environmental conditions are heterogeneous and each transect could have it's own unique qualities. In this case, differences such as local human activity, development, slope/elevation of nearby land, unforeseen historical factors, etc, could skew the data. That being said, we can at least set up a study that's close enough to centris paribus for the results to be meaningful. Here, all sites are located in the same general ecological zone, and they are all south facing. They may not be the same size, but this can be acknowledged and accounted for.

Step 2: Read & wrangle data

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

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

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

rawdata <- read_csv(here('data', 'spiny_abundance_sb_18.csv')) %>%
    clean_names() %>%
    mutate(size_mm = na_if(size_mm, -99999))
```

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"
```

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

```
                                    "Carpenteria",
                                    "Mohawk",
                                    "Isla Vista",
                                    "Naples")))
```

Create new `df` named `spiny_counts`

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

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

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

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

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

spiny_counts <- tidydata %>%

    # classify sites by treatment
    mutate(mpa = case_when(
                  site %in% c("AQUE", "CARP", "MOHK") ~ 'non_MPA',
                  site %in% c("IVEE", "NAPL") ~ 'MPA'
              ),
              treat = case_when(
                  mpa == 'MPA' ~ 1,
                  mpa == 'non_MPA' ~ 0)) %>%


    group_by(site, year, transect) %>%
    summarize(counts = sum(count, na.rm = TRUE),
            mean_size = mean(size_mm, na.rm = TRUE),
            mpa = first(mpa),
            treat = first(treat))
```

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**:

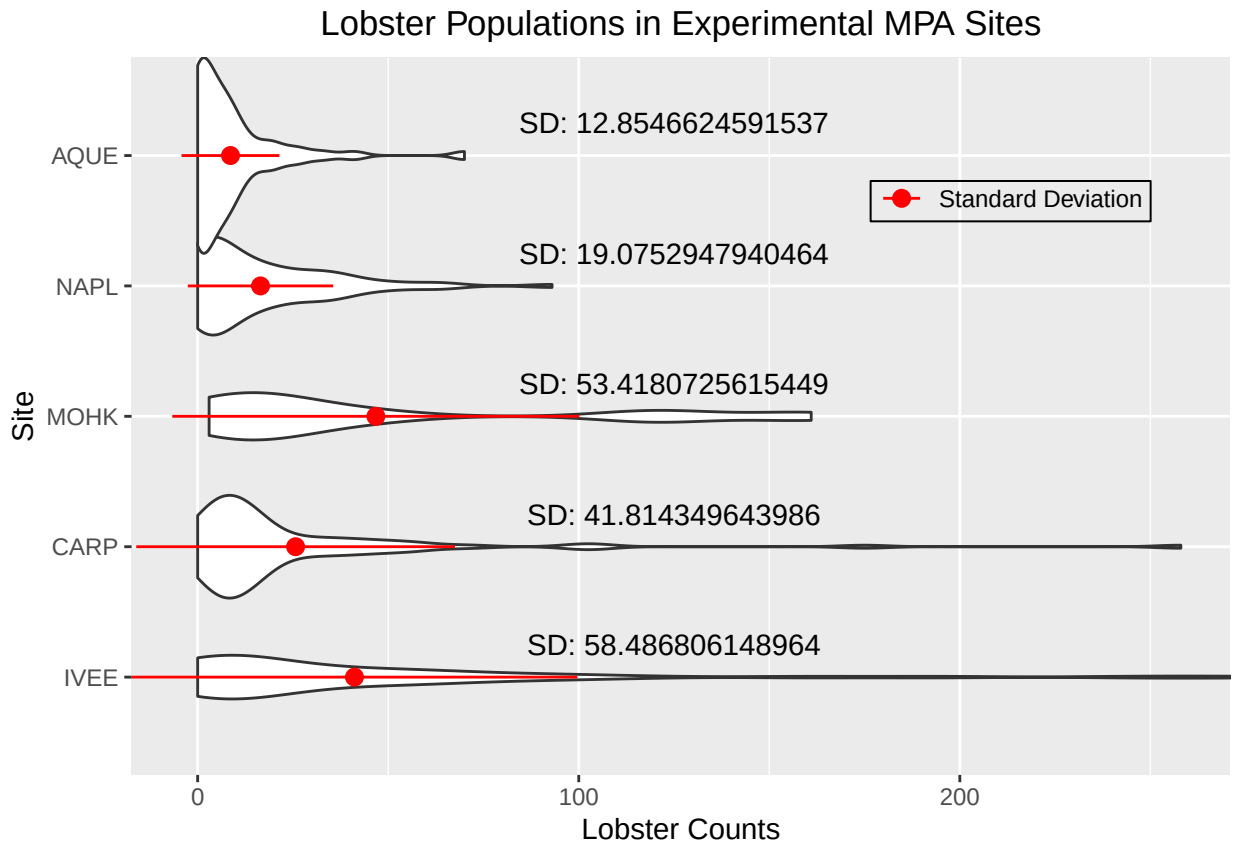
1) grouped by reef site

2) grouped by MPA status
3) grouped by year

Create a plot of lobster **size** :

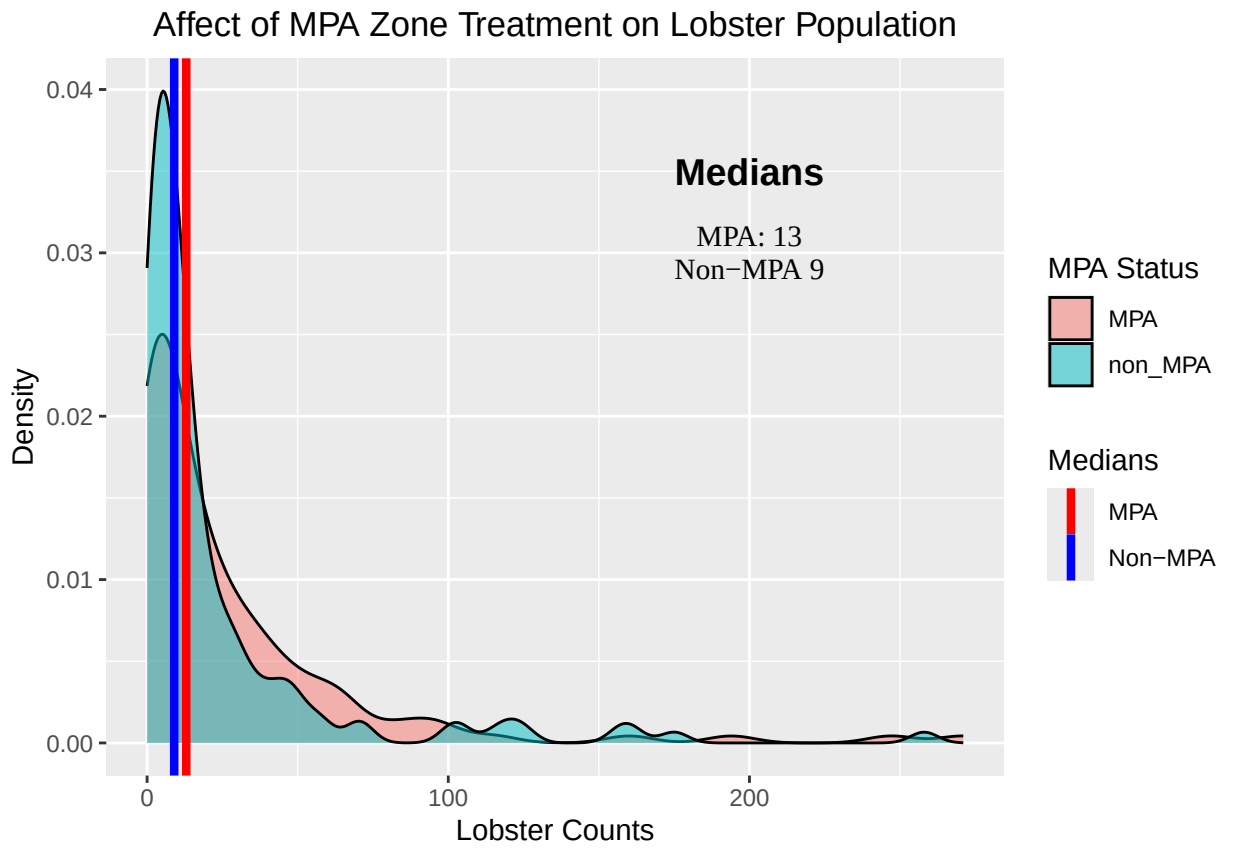
4) You choose the grouping variable(s)!

```
# plot 1: violin w SD

spiny_counts %>%
    ggplot(aes(x = counts, y = factor(site,
                                      levels = c('IVEE',
                                                 'CARP',
                                                 'MOHK',
                                                 'NAPL',
                                                 'AQUE')))) +
    geom_violin(width = 1.5) +

    #  plot standard deviation lines
    stat_summary(fun.data="mean_sdl", fun.args = c(mult = 1),
                 geom="pointrange", aes(color = 'Standard Deviation')) +


    labs(x = 'Lobster Counts',
         y = 'Site',
         title = 'Lobster Populations in Experimental MPA Sites') +

    # format line color and legend
    scale_color_manual(name = '', values = c('red')) +

    # shrink edges of graph
    scale_x_continuous(expand = c(0, NA)) +

    # place SD coefficients
    annotate('text', x = 125,
             y = 1.25:5.25,
             label = c(paste('SD:', sd(filter(spiny_counts, site == 'IVEE')$counts)),
                       paste('SD:', sd(filter(spiny_counts, site == 'CARP')$counts)),
                       paste('SD:', sd(filter(spiny_counts, site == 'MOHK')$counts)),
                       paste('SD:', sd(filter(spiny_counts, site == 'NAPL')$counts)),
                       paste('SD:', sd(filter(spiny_counts, site == 'AQUE')$counts)))) +


    theme(plot.title = element_text(hjust = 0.5), # center title
          legend.position = 'inside',
          legend.position.inside = c(0.8, 0.8),
          legend.background = element_blank(), # remove legend background
          legend.box.background = element_rect(color = 'black', linetype = 1), # recreate bg with borde
          legend.margin = margin(-18,3,0,3),
          legend.key = element_blank()) # remove legend icon backgrounds
```

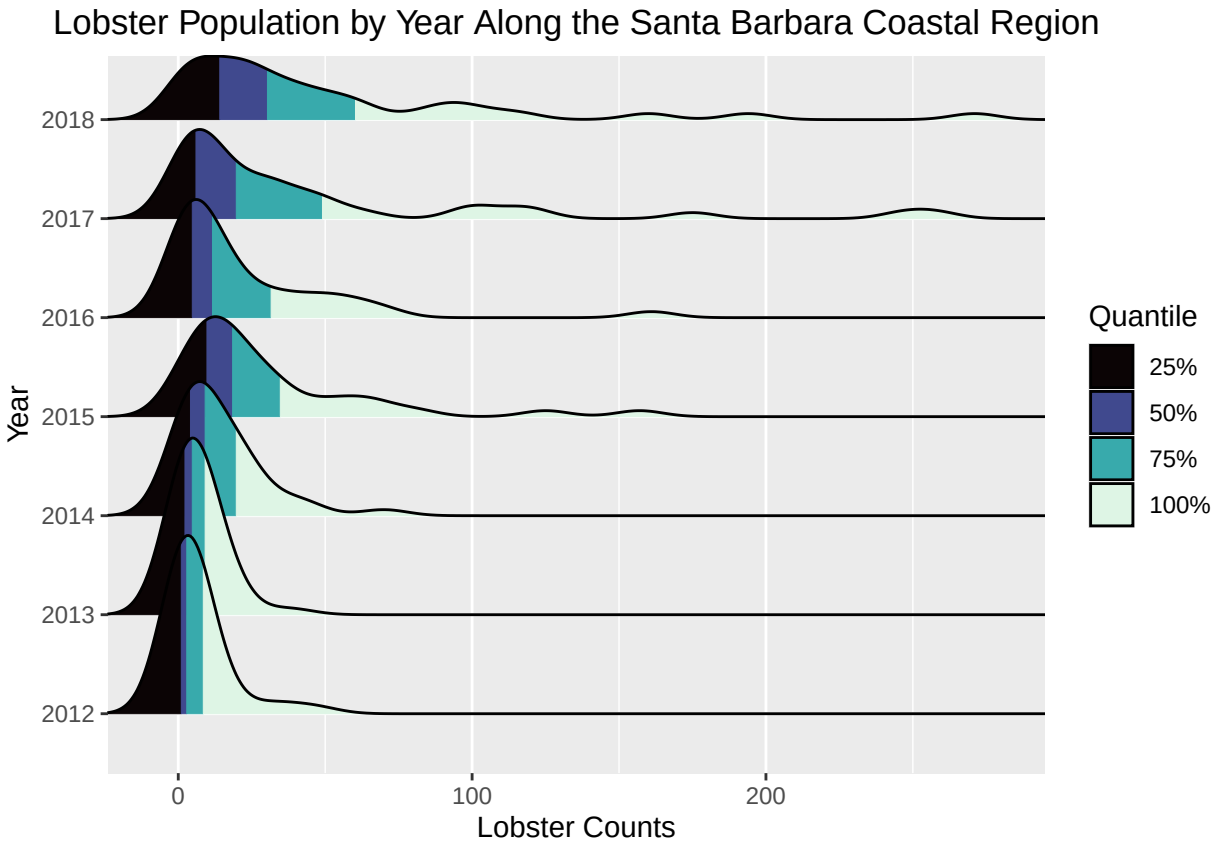



```
# plot 2: ridge

spiny_counts %>%
    ggplot(aes(x = counts, fill = mpa)) +
        geom_density(alpha = 0.5) +

    # MPA median line
    geom_vline(aes(xintercept = c(quantile(filter(spiny_counts,
                                                  mpa == 'MPA')$counts,
                                           na.rm = TRUE)[3]),
                   color = 'MPA'),
               lwd = 1.5) +

    # Non-MPA median line
    geom_vline(aes(xintercept = c(quantile(filter(spiny_counts,
                                                  mpa == 'non_MPA')$counts,
                                           na.rm = TRUE)[3]),
                   color = 'Non-MPA'),
               lwd = 1.5) +


    labs(y = 'Density',
         x = 'Lobster Counts',
         title = 'Affect of MPA Zone Treatment on Lobster Population') +

    # legend title
```

```
    guides(fill = guide_legend(title = 'MPA Status')) +

    # make and color legend for means
    scale_color_manual(name = 'Medians', values = c('MPA' = 'red', 'Non-MPA' = 'blue')) +

    # median text
    annotate('text',
             x = 200,
             y = 0.035,
             label = c('Medians'),
             size = 5,
             fontface = 'bold') +
    annotate('text',
             x = 200,
             y = c(0.031, 0.029),
             label = c(paste('MPA:',
                             median(filter(spiny_counts,
                                           mpa == 'MPA')$counts)),
                       paste('Non-MPA',
                             median(filter(spiny_counts,
                                           mpa == 'non_MPA')$counts))),
             family = 'serif') +

    # center title
    theme(plot.title = element_text(hjust = 0.5))
```

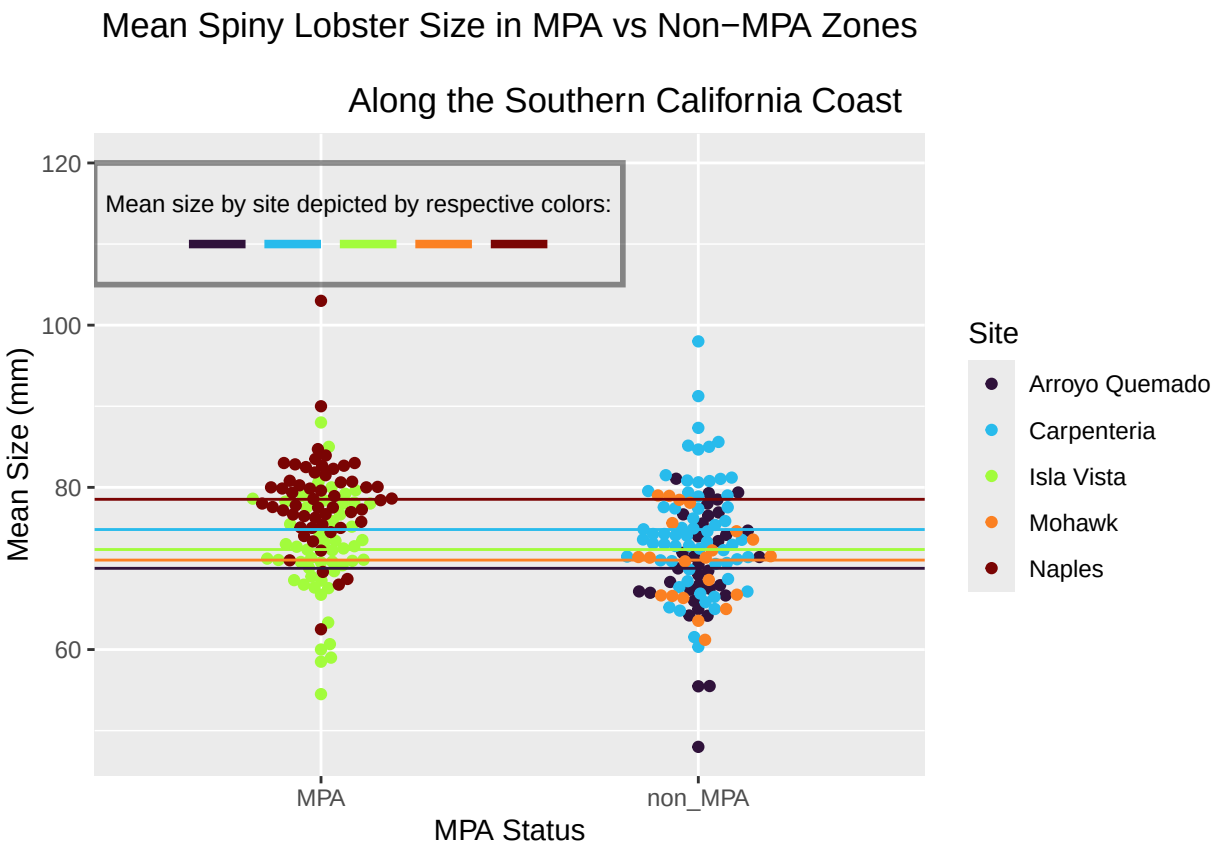


```r
# # plot 3: histogram
spiny_counts %>%
    ggplot(aes(x = counts, y = as.factor(year), fill = stat(quantile))) +

    # ridge plot with quantiles
    ggridges::stat_density_ridges(quantile_lines = FALSE,
                                  calc_ecdf = TRUE,
                                  geom = "density_ridges_gradient") +


    labs(x = 'Lobster Counts',
         y = 'Year',
         title = 'Lobster Population by Year Along the Santa Barbara Coastal Region') +

    # cut graph to edges
    scale_x_continuous(expand = c(0, NA)) +

    # rename, recolor, relabel quantile legend
    scale_fill_manual(name = "Quantile",
                      values = viridisLite::mako(4),
                      labels = c('25%', '50%', '75%', '100%')) +

    # center title
    theme(plot.title = element_text(hjust = 0.5))
```



```r
# plot 4: beeswarm
spiny_counts %>%
    ggplot(aes(x = mpa, y = mean_size, color = site)) +
        ggbeeswarm::geom_beeswarm(cex = 1.5) +

    # mean lines by site
    geom_hline(aes(yintercept = mean(filter(spiny_counts,
                                            site == 'IVEE')$mean_size,
                                     na.rm = TRUE)),
               color = viridisLite::turbo(5)[3]) +
    geom_hline(aes(yintercept = mean(filter(spiny_counts,
                                            site == 'CARP')$mean_size,
                                     na.rm = TRUE)),
               color = viridisLite::turbo(5)[2]) +
    geom_hline(aes(yintercept = mean(filter(spiny_counts,
                                            site == 'MOHK')$mean_size,
                                     na.rm = TRUE)),
               color = viridisLite::turbo(5)[4]) +
    geom_hline(aes(yintercept = mean(filter(spiny_counts,
                                            site == 'NAPL')$mean_size,
                                     na.rm = TRUE)),
               color = viridisLite::turbo(5)[5]) +
    geom_hline(aes(yintercept = mean(filter(spiny_counts,
                                            site == 'AQUE')$mean_size,
                                     na.rm = TRUE)),
               color = viridisLite::turbo(5)[1]) +

    # retitle legend
    guides(color = guide_legend(title = 'Site')) +

    # make legend for mean lines
    annotate('text',
             x = 1.1,
             y = 115,
             label = 'Mean size by site depicted by respective colors:',
             size = 3) +
    annotate("segment",
             x = c(0.65, 0.85, 1.05, 1.25, 1.45),
             xend = c(0.8, 1, 1.2, 1.4, 1.6),
             y = 110,
             yend = 110,
  color = viridisLite::turbo(5),
  lwd = 1.5) +
    annotate('rect',
             xmin = 0.4,
             xmax = 1.8,
             ymin = 105,
             ymax = 120,
             fill = NA,
             color = '#00000070',
             lwd = 1) +
```

```
    labs(title = 'Mean Spiny Lobster Size in MPA vs Non-MPA Zones\n
                        Along the Southern California Coast',
         x = 'MPA Status',
         y = 'Mean Size (mm)') +

    # center title
    theme(plot.title = element_text(size = 13, hjust = 0.5)) +

    # relabel and recolor legend
    scale_color_manual(labels = c('Arroyo Quemado',
                                  'Carpenteria',
                                  'Isla Vista',
                                  'Mohawk',
                                  'Naples'),
                       values = viridisLite::turbo(5))
```



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

```
# USE: gt_summary::tbl_summary()
tbl_summary(spiny_counts,
            treat,
            statistic = list(all_continuous() ~ "{mean}"),  # just the mean
            include = c(counts, mean_size),  # select columns
            missing = 'no') # don't include unknown value row
```

Characteristic	**0** N = 133[1]	**1** N = 119[1]
counts	23	28
mean_size	73	76

[1] Mean

Step 4: OLS regression- building intuition

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

This model would suggest that we cannot make any deterministic claims about if there is a relationship between the MPA treatment and lobster populations. That being said, if there was a relationship, it can be interpreted as there being an average difference of 5.36 in the lobster counts between the treatment sites and the non-treatment sites, with treatment sites generally having higher counts. Alternatively, it's possible that a linear model is just not appropriate to predict the relationship between these two variables.

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

m1_ols <- lm(counts ~ treat, spiny_counts)

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

Observations	252
Dependent variable	counts
Type	OLS linear regression

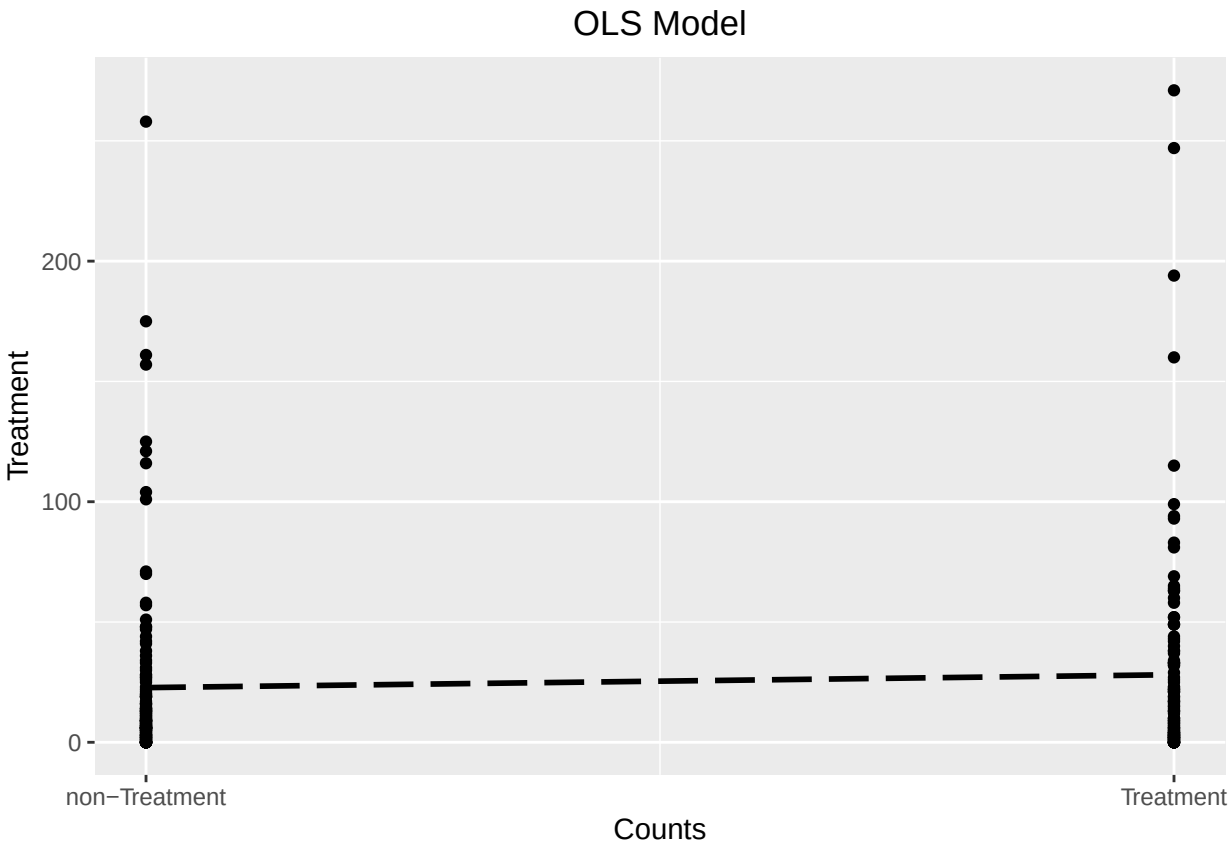
	Est.	S.E.	t val.	p
(Intercept)	22.73	3.57	6.36	0.00
treat	5.36	5.20	1.03	0.30

Standard errors: OLS

```
cat('r_squared:', summary(m1_ols)$r.squared)
```

```
## r_squared: 0.004236826
```

```
# plot OLS line

spiny_counts %>%
    ungroup() %>%
    mutate(reg = fitted(m1_ols)) %>%
ggplot(aes(x = treat, y = counts)) +
    geom_point() +
    geom_line(aes(x = treat, y = reg), linetype = 5, linewidth = 1) +
    scale_x_continuous(breaks = c(0,1), labels = c('non-Treatment', 'Treatment'))  +
    labs(x = 'Counts',
         y = 'Treatment',
         title = 'OLS Model') +
    theme(plot.title = element_text(hjust = 0.5))
```

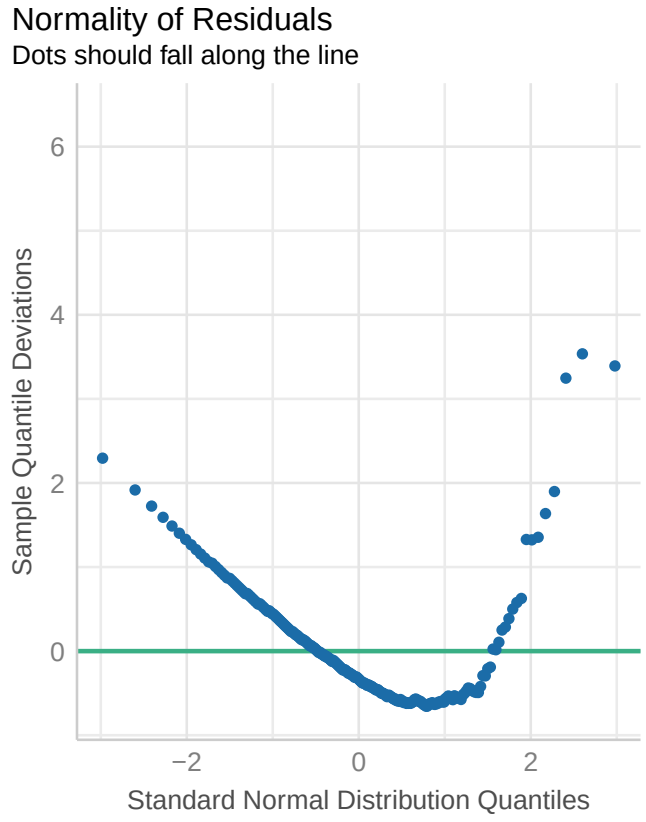


c. Check the model assumptions using the `check_model` function from the `performance` package

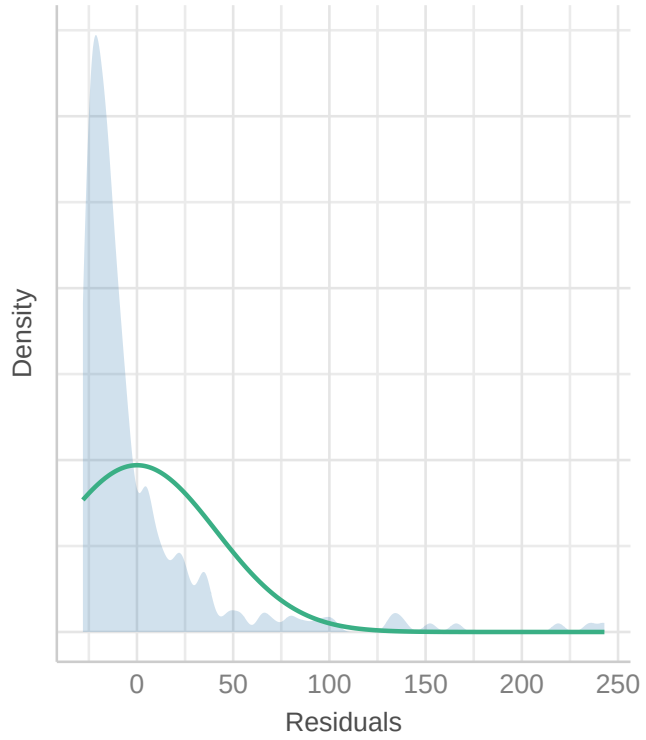
d. Explain the results of the 4 diagnostic plots. Why are we getting this result?

- For the normality of residuals, when you run a linear regression for a model with a binary variable, you simply end up with points that have a very high residual, and some with coincidentally very low residuals because of the relationship between a binary variable and a straight line. This can be shown in both graphs, where there are specific regions in the data that are very far from the horizontal line, and some that are closer.
- The homogeneity of variance graph suggests that the relationship is heteroscedastic, because points do not at all form a straight line. Likely because the relationship to the line is so strange, it appears that the assumption of equal variance between all points is not true.
- The posterior predictive check graph assumes normal distribution of the data, which does not appear to be the case when you plot a linear model between counts and treatment. Because treatment is binary, all the data is bunched up around two points, 1 and 0. This can have strange results when attempting to plot a density graph.

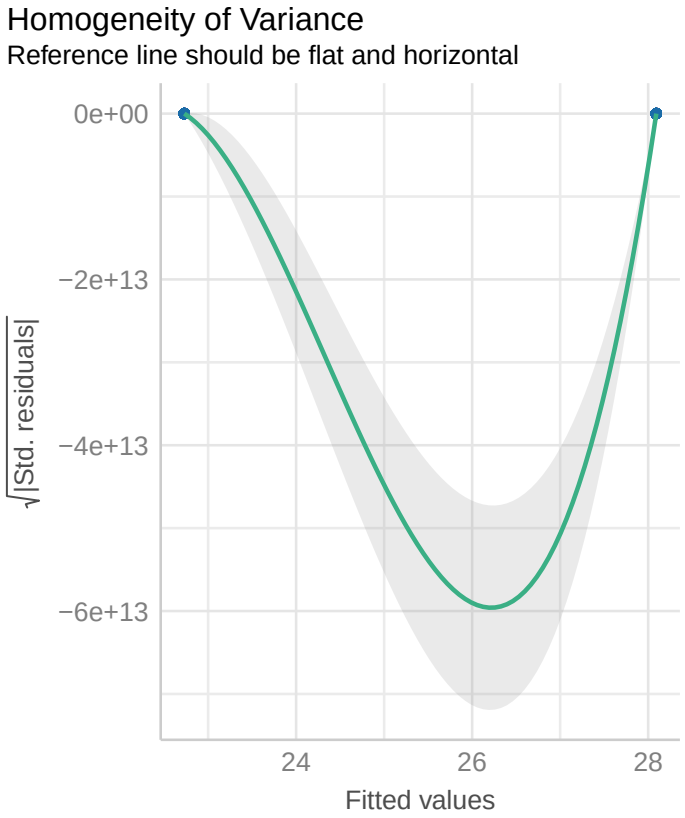
```
check_model(m1_ols,  check = "qq" )
```



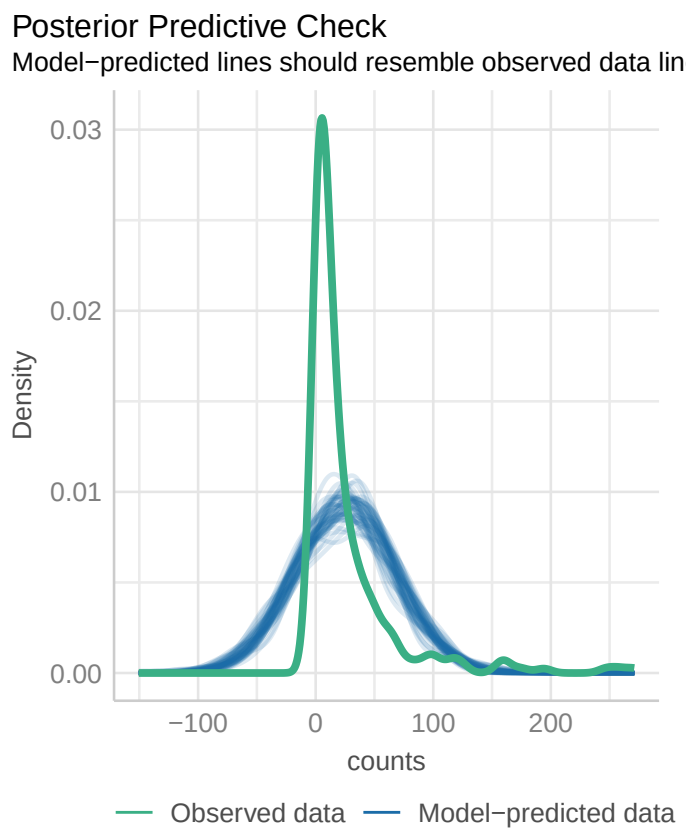

```
check_model(m1_ols, check = "normality")
```




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




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



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 poisson glm shows a very close relationship between the treatment and lobster populations. It can be interpreted as the chance of a lobster count being from a treatment site or a non-treatment site being different by about 21%, with treatment sites being more likely to have more lobsters than the non-treatment sites.

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

Dispersion is another word for variance. Overdispersion is when there is greater dispersion in the data than was expected to be the case based on the predictions of a model. For a poisson regression, there is an assumption that variance = mean, but this may not be true in the actual data, and variance > mean, which is overdispersion. In the case of this model, overdispersion could occur either from spatial autocoorelation between transects, or ommitted variable bias.

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

The poisson model more accurately explains the relationship between the treatment and and dependent variable, as rather there being a defined change between MPA and non-MPA zones, there is a chance that a count observation is in either. The p-value become significant from the linear model to the poisson model. This is likely due to the produced line just being a better fit along the data.

```
#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
```

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

summ(m2_pois)
```

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	poisson
Link	log

$\chi^2(1)$	71.36
p	0.00
Pseudo-R^2 (Cragg-Uhler)	0.25
Pseudo-R^2 (McFadden)	0.01
AIC	11365.62
BIC	11372.68

	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
# plot poisson curve

spiny_counts %>%
    ungroup %>%
    mutate(reg = fitted(m2_pois)) %>%
ggplot(aes(x = treat, y = counts)) +
    geom_point() +
    geom_line(aes(x = treat, y = reg), linetype = 5, linewidth = 1) +
    scale_x_continuous(breaks = c(0,1), labels = c('non-Treatment', 'Treatment')) +
    labs(x = 'Counts',
         y = 'Treatment',
         title = 'Poisson Model') +
    theme(plot.title = element_text(hjust = 0.5))
```

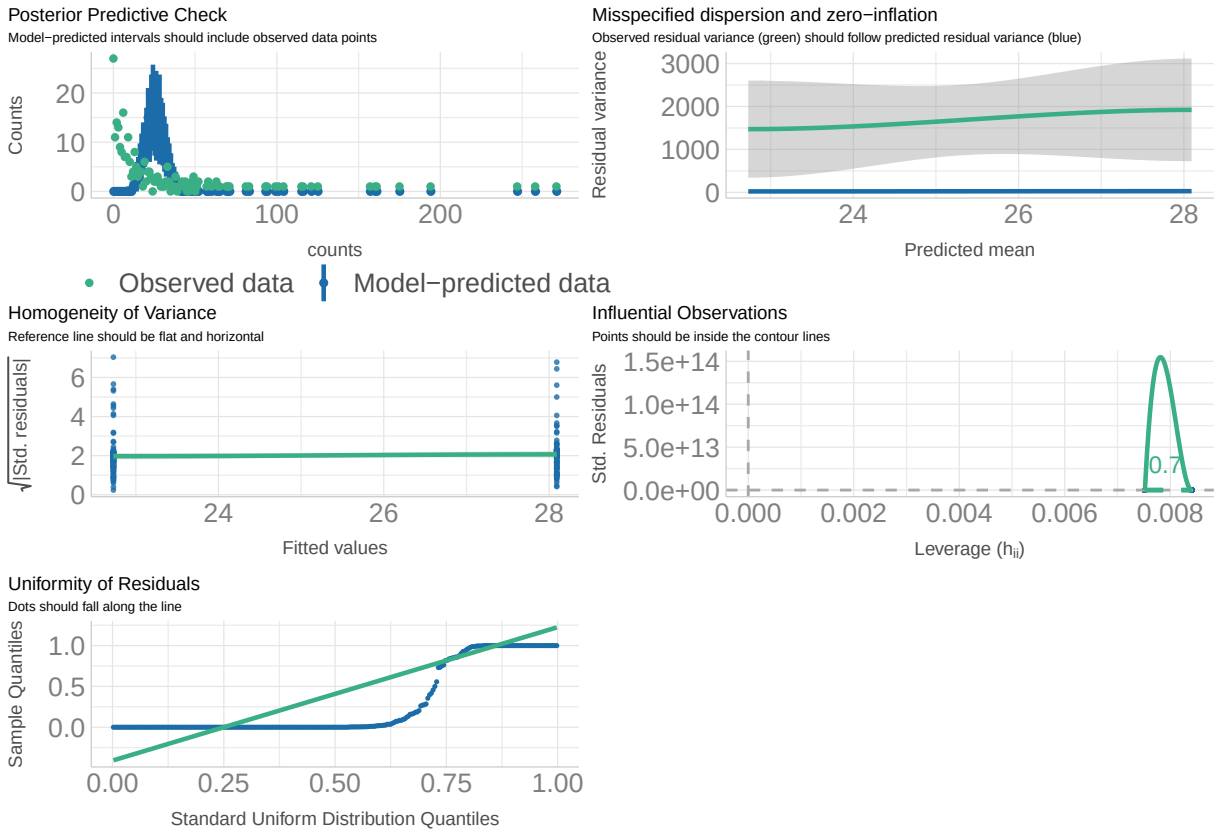
e. Check the model assumptions. Explain results.

Because a poisson model better represents the relationships in the data, the line generated is a better fit and therefore the observations are approximately homoscedastic, as suggested by the homogeneity of variance graph. But the poisson regression still doesn't distribute the residuals normally, which may be causing overdispersion.

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

There is a high degree of over-dispersion, potentially due to omitted variable bias, or the model just not being appropriate. For the zero-inflation test, while the actual count data has some zeros in it, the model is predicting zero count events to be more common than they actually are.

```
check_model(m2_pois, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```




```
check_overdispersion(m2_pois)
```

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

```
check_zeroinflation(m2_pois)
```

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

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

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

The data violates an assumption about poisson distributions, being mean != variance (overdispersion). A negative binomial model is an alternative in which this violation are accounted for.

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

The relationship is found to be not significant at a p-value of 0.22. But if it was, the model could be interpreted as when selecting a count value at random, there is approximately a 21% difference in the chance of that count value being from either the treatment sites or the non-treatment sites, with it being more likely to be from the treatment sites if it's higher, and vice versa if it's lower. The poisson model actually had the

same coefficient with a significant p-value, but because many assumptions about poisson were violated, it was not a good model in the sense of it being statistically conclusive.

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

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

```
summ(m3_nb)
```

Observations	252
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.55)
Link	log

$\chi^2(250)$	1.52
p	0.22
Pseudo-R^2 (Cragg-Uhler)	0.01
Pseudo-R^2 (McFadden)	0.00
AIC	2088.53
BIC	2099.12

	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

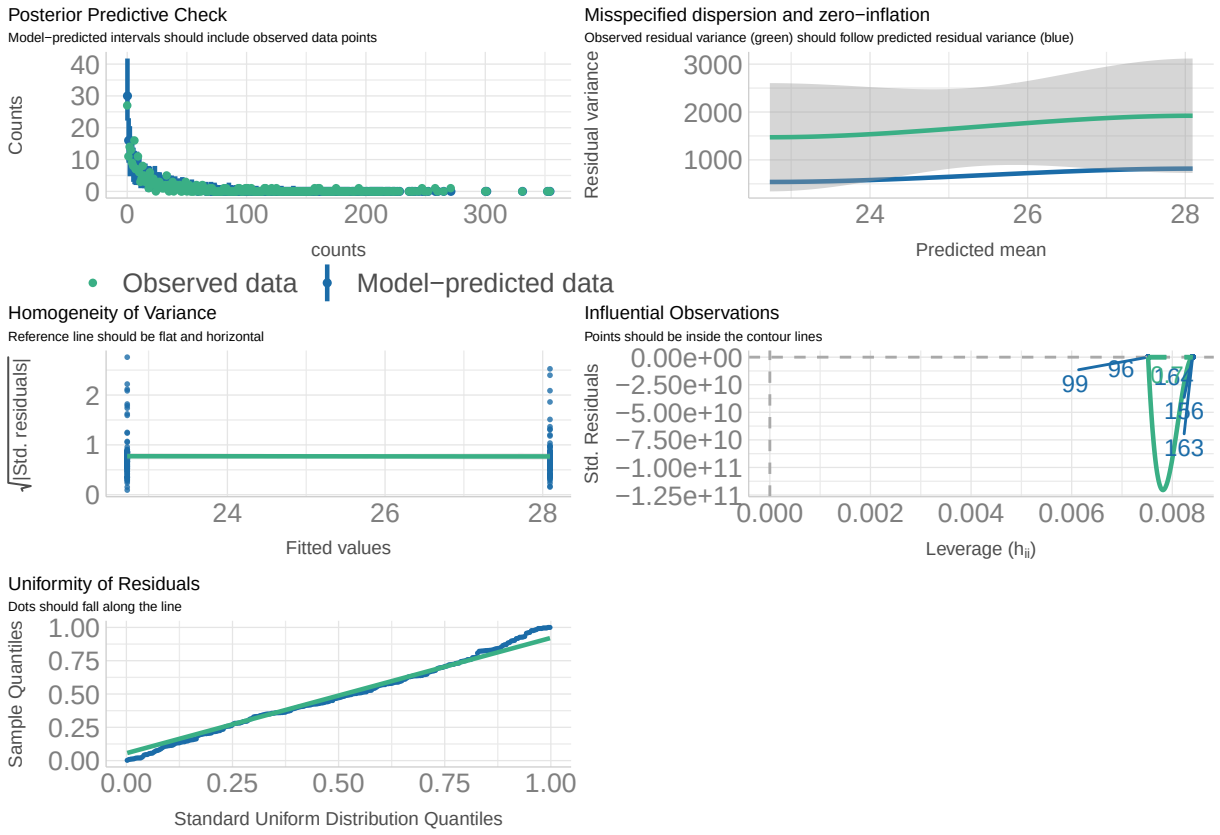
```
check_overdispersion(m3_nb)
```

```
## # 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_model(m3_nb, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```



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.

The negative binomial model is the best fit for the data, even though it is not significant. This is because the poisson model violates assumptions, and the OLS model is simply a poor way to represent the relationship. The conclusions about the relationship between the two variables change between models, so the treatment effect is not robust.

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

Step 7: Building intuition - fixed effects

a. Create new `df` with the `year` variable converted to a factor

b. Run the following OLS model using `lm()`

- Use the following specification for the outcome `log(counts+1)`
- Estimate fixed effects for `year`
- Include an interaction term between variables `treat` and `year`

	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.

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)

Overall, the coefficients for treatment suggests that there is a significant negative relationship between MPAs and lobster counts when accounting for current year, with y-intercept of the treated sites being 1.23 lower than that of the non-treated sites. The year20XX coefficients show the total difference between the y-intercept and the current log count value for that year for the non-treatment sites. The treat:year20XX coefficients show the same, but for the treatment sites.

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

The main treatment coefficient is negative because the treatment effect in this parallel slopes model is negative, as in there is a significant negative difference between the treatment and non-treatment sites for lobster populations where treatment sites are correlated with lower counts. I suppose this could make sense in the model itself. In real life, it would depend if time is a deterministic variable for lobster populations as it relates to the MPA treatment.

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

m5_fixedeffs <- lm(
    log(counts+1) ~ treat*year,
    data = ff_counts)

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

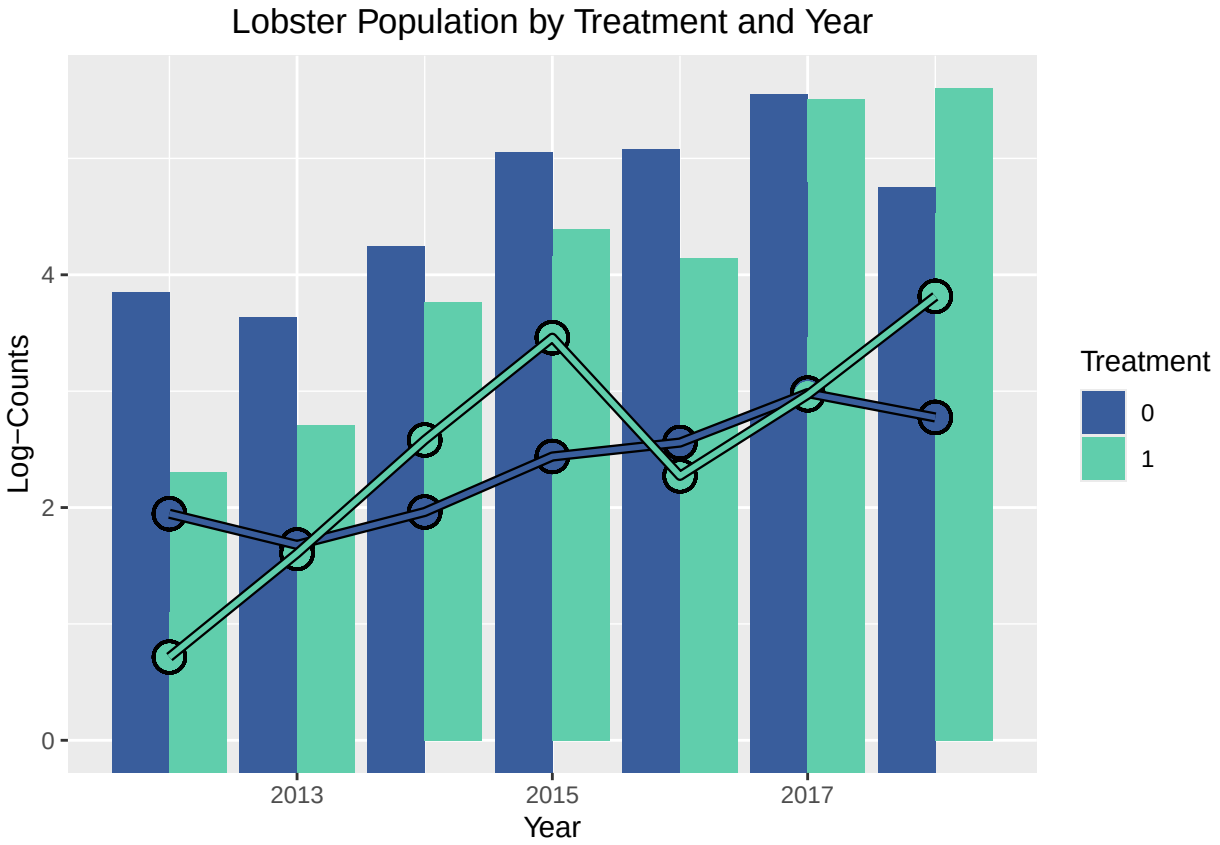
Observations	252
Dependent variable	log(counts + 1)
Type	OLS linear regression

```
# plot fixed effects model

spiny_counts %>%
    ungroup() %>% # get errors without ungrouping
    mutate(reg = fitted(m5_fixedeffs)) %>% # create column with fitted line
ggplot(aes(x = year, y = log(counts),
           fill = factor(treat))) +
    geom_col(position = 'dodge') +
    geom_point(aes(x = year, y = reg),
               size = 5.5,
               show.legend = FALSE) +
    geom_point(aes(x = year, y = reg, color = factor(treat)),
```

	Est.	S.E.	t val.	p
(Intercept)	1.95	0.27	7.26	0.00
treat	-1.23	0.39	-3.16	0.00
year2013	-0.27	0.38	-0.71	0.48
year2014	0.02	0.38	0.04	0.97
year2015	0.49	0.38	1.30	0.20
year2016	0.61	0.38	1.61	0.11
year2017	1.04	0.38	2.73	0.01
year2018	0.83	0.38	2.18	0.03
treat:year2013	1.16	0.55	2.10	0.04
treat:year2014	1.85	0.55	3.35	0.00
treat:year2015	2.25	0.55	4.08	0.00
treat:year2016	0.95	0.55	1.71	0.09
treat:year2017	1.22	0.55	2.20	0.03
treat:year2018	2.27	0.55	4.12	0.00

Standard errors: OLS

```
            size = 4,
            show.legend = FALSE) +
  ggborderline::geom_borderline(aes(x = year,
                                    y = reg,
                                    color = factor(treat),
                                    bordercolour = 'black'),
                                lwd = 1,
                                show.legend = FALSE) +
  scale_fill_manual(values = viridisLite::mako(2, begin = 0.4, end = 0.8)) +
  scale_color_manual(values = viridisLite::mako(2, begin = 0.4, end = 0.8)) +
  labs(x = 'Year',
       y = 'Log-Counts',
       title = 'Lobster Population by Treatment and Year') +
  guides(fill = guide_legend(title = 'Treatment')) +
  theme(plot.title = element_text(hjust = 0.5))
```

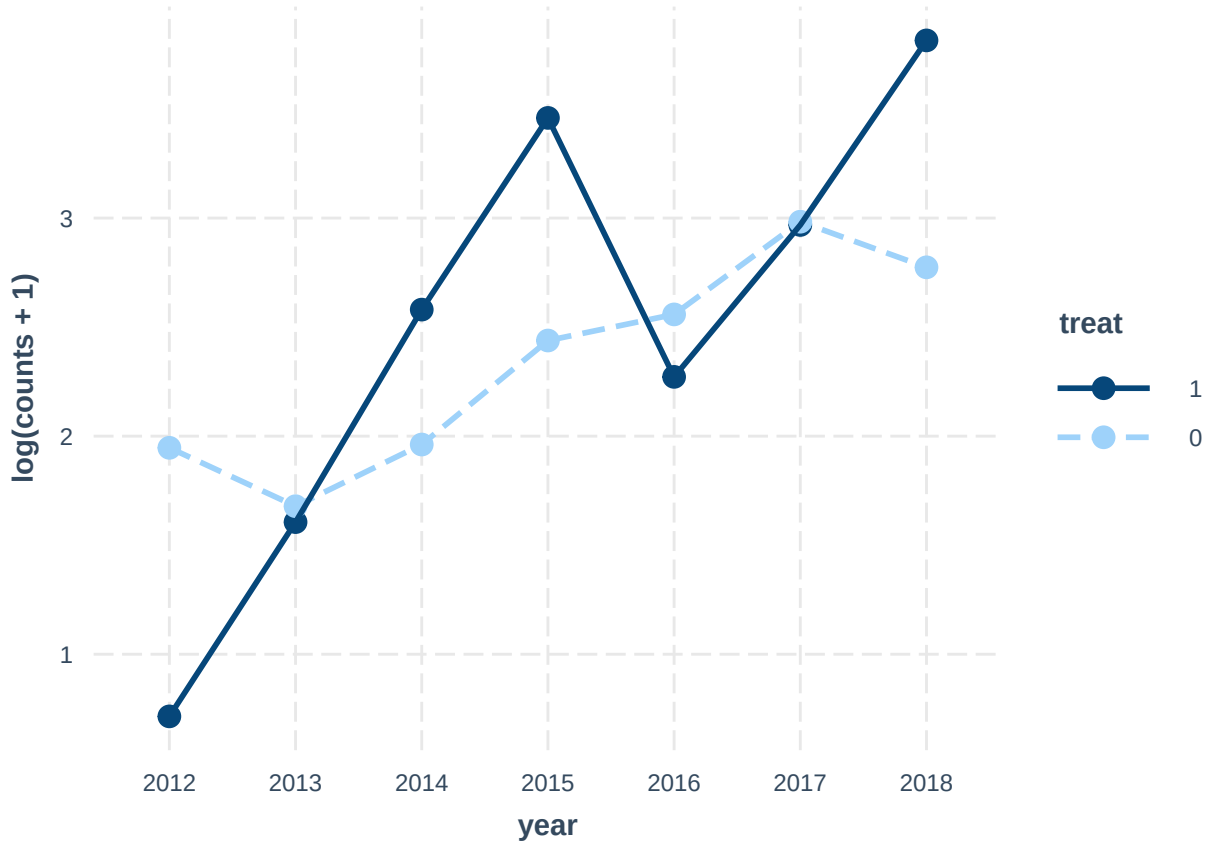


e. Look at the model predictions: Use the `interact_plot()` function from package `interactions` to plot mean predictions by year and treatment status.

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

I had used the above visualization to answer the question initially, so I stand by my answer.

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

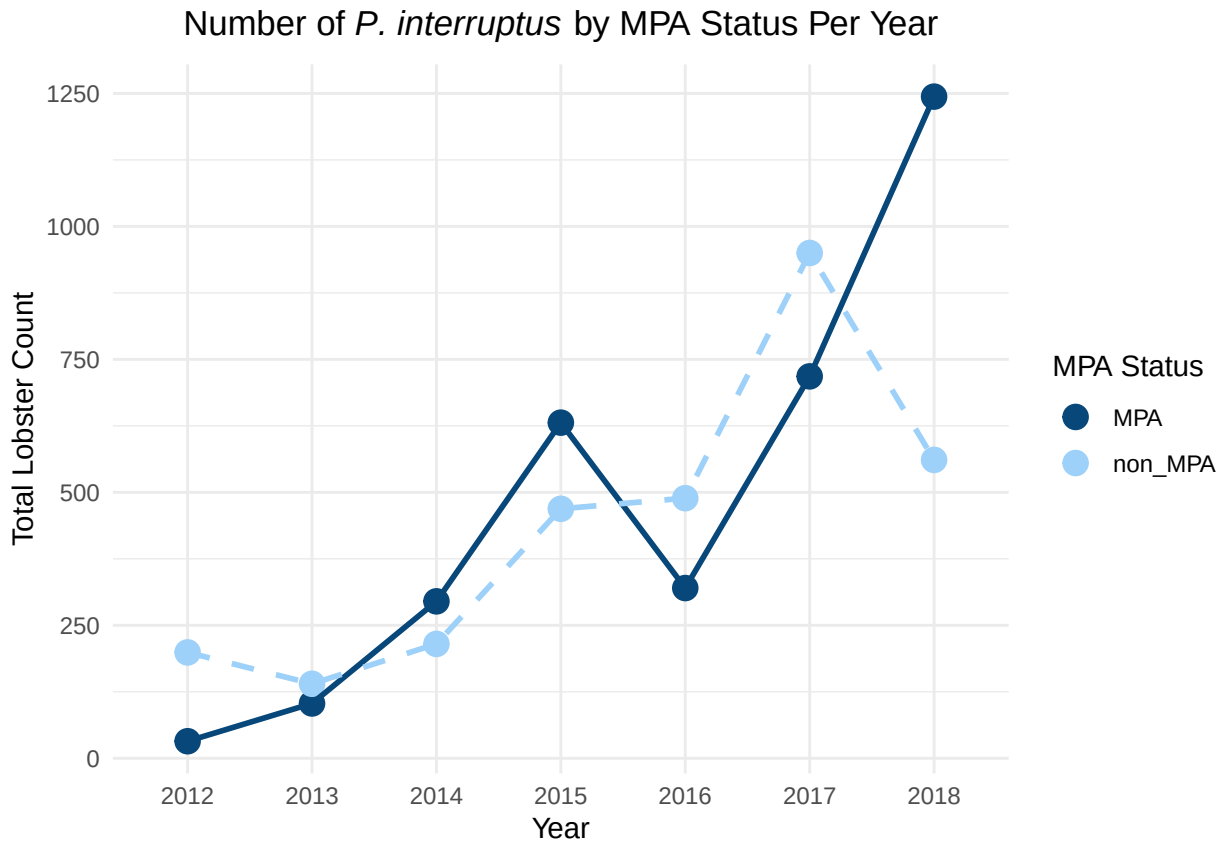


g. Using `ggplot()` create a plot in same style as the previous `interaction plot`, but displaying the original scale of the outcome variable (lobster counts). This type of plot is commonly used to show how the treatment effect changes across discrete time points (i.e., panel data).

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

```
# 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) %>%
    mutate(tot_count = sum(counts), # total counts by year and mpa
           year = as_factor(year)) %>% # make year a factor variable
    summarize(tot_count = first(tot_count)) # crunch df to one row per year and treatment

# plot_counts %>% ggplot() ...

plot_counts %>%
    ggplot(aes(x = year, y = tot_count, group = mpa, color = mpa)) +
    geom_point(size = 4) +

    # plot model lines by treatment visualized by line type
    geom_line(aes(linetype = mpa), lwd = 1, show.legend = FALSE) +


    labs(title = expression(paste('Number of ',
                                  italic('P. interruptus'),
```

```
                                ' by MPA Status Per Year')),
       x = 'Year',
       y = 'Total Lobster Count',
       color = 'MPA Status') +


  theme(plot.title = element_text(hjust = 0.5), # center title
        panel.background = element_rect(fill = 'white'), # make background blend in
        panel.grid = element_line(color = 'grey92'), # make grid visible
        axis.ticks = element_blank()) + # remove axis ticks


  scale_color_manual(values = c('#084779', '#9ED1F9')) +

  # specify mpa statuses different linetypes
  scale_linetype_manual(values = c(1, 2))
```



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)

There could be spillover effects, depending on a couple factors. One is the ecology of spiny lobsters, as well as ocean ecology broadly. The ocean is a big 3D environment, where individual organisms can travel long

distance. Spiny lobsters themselves are known for seasonal migration, specifically from shallow waters to deep waters and back. There is a possibility of spiny lobsters migrating between sites. Potentially, if the treatment sites did experience an increase in population, that increase was nullified by reaching carrying capacity, or attracting predators of the species.

Another possible factor is the enforcement of the MPA itself. MPA's may suffer from the edge affect, in which significant fishing activities occur at the edge of MPA zones reducing biodiversity therein (Tel-Aviv University, 2021). Such activities may have a significant effect on the MPA zone itself, despite being fully legal. The affect this has on neighboring zones is uncertain, but because the ocean ecosystem is largely interconnected, there is a possibility it can.

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

With causal analysis, there is an assumption that the effect of the treatment is exclusive to the subjects that the treatment was a target of. When this possibly isn't the case, it becomes difficult to argue that the treatment itself was the cause of the effect of it.

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

The goal of this experiment was to say for certain whether MPA's increase spiny lobster populations or not. Causal research in a field setting is already difficult, but because of the conditions of the ocean ecosystem being what they are, the possibility of spillover between sites is too great to make a definitive conclusion about causality.

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
   - There are two assumptions of SUTVA, no hidden variation and no interference.
   - It's is theoretically true that there is not hidden variation, as in a legal sense the MPA laws were applied equally. But there is always the possibility of certain stocastic differences rearing their head, like minute regional differences in the likelihood of fishing in restricted zones and the ability of local enforcement to enforce the MPAs.
   - The assumption of no interference is the troublesome one however. As mentioned in the previous questions, the tendency of ocean species to travel long distances means that it is unlikely that this assumption holds up in it's entirety.
2) Excludability assumption
   - Generally speaking, this assumption is one that is impossible to prove in any field setting, especially in an ecological setting.
   - The ocean ecosystem is one that is normally very static but can randomly undergo sudden shifts in conditions. Meaning while the difference in the counts could potentially be attributed to the MPAs, unpredictable abiotic shifts can muddy the waters, unless these alternative variables are accounted for in the model. Given that it can be difficult to know when these shifts occur, this can be difficult, and therefore so is validating the excludability assumption.

Citations:

Tel-Aviv University. (2021, August 2). Overfishing and other human pressures are severely harming many marine protected areas around the world, study finds. ScienceDaily. Retrieved January 19, 2025 from www.sciencedaily.com/releases/2021/08/210802114940.htm

---

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

```
lobst_24 <- read_csv(here::here('data', 'lobster_sbchannel_24.csv')) %>%
    clean_names() %>%
    mutate(size_mm = na_if(size_mm, -99999)) # make null values NA

# reorder and relabel data by site
tidydata_24 <- lobst_24 %>%
    mutate(reef = factor(site,
                         levels = c("AQUE",
                                    "CARP",
                                    "MOHK",
                                    "IVEE",
                                    "NAPL"),
                         labels = c("Arroyo Quemado",
                                    "Carpenteria",
                                    "Mohawk",
                                    "Isla Vista",
                                    "Naples")))

spiny_counts_24 <- tidydata_24 %>%

    # classify sites by treatment
    mutate(mpa = case_when(
                  site %in% c("AQUE", "CARP", "MOHK") ~ 'non_MPA',
                  site %in% c("IVEE", "NAPL") ~ 'MPA'
              ),
              treat = case_when(
                  mpa == 'MPA' ~ 1,
                  mpa == 'non_MPA' ~ 0)) %>%
    group_by(site, year, transect) %>%
    summarize(counts = sum(count, na.rm = TRUE),
            mean_size = mean(size_mm, na.rm = TRUE),
            mpa = first(mpa),
            treat = first(treat))
```

b. Run at least 3 regression models & assess model diagnostics

```
lob_24_ols <- lm(counts ~ treat, spiny_counts_24)

summ(lob_24_ols)
```

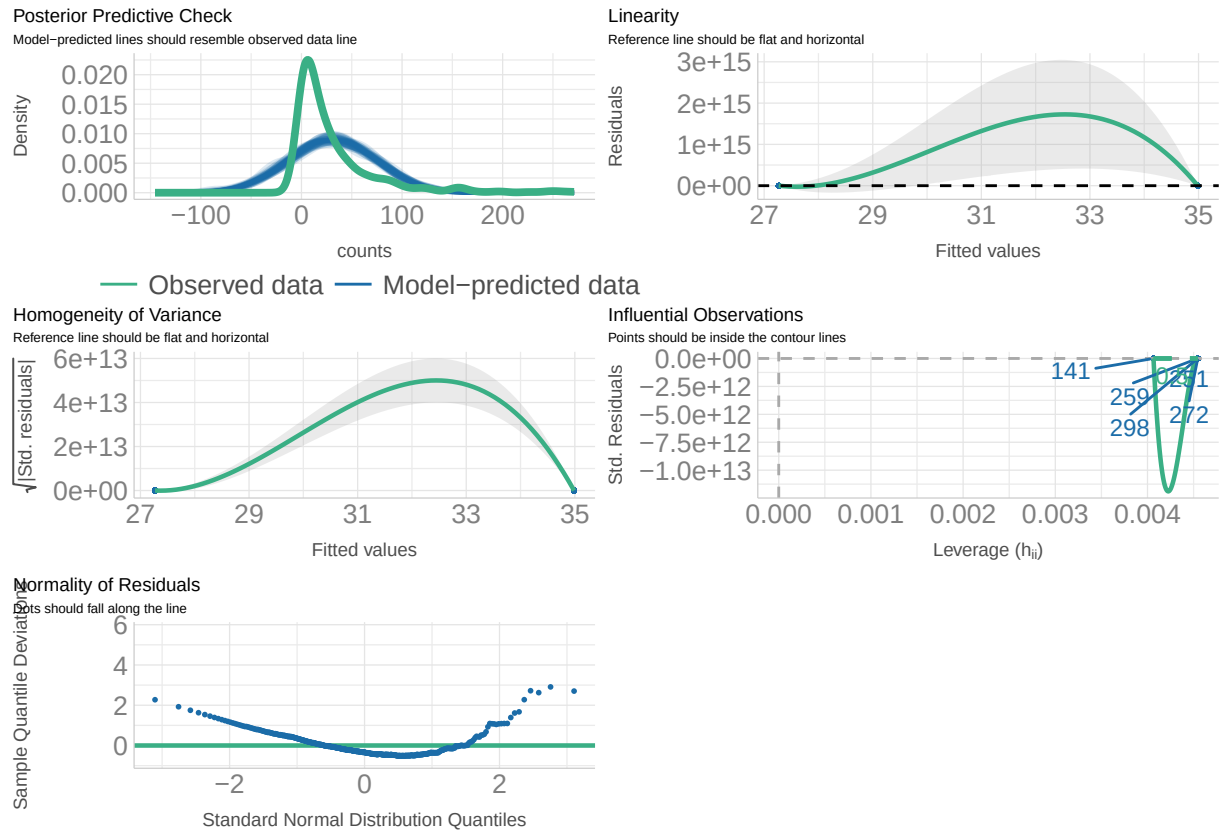
Observations	466
Dependent variable	counts
Type	OLS linear regression

F(1,464)	3.90
R^2	0.01
Adj. R^2	0.01

	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

```
check_model(lob_24_ols, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```




```
lob_24_poisson <- glm(counts ~ treat,
                      family = poisson(link = 'log'),
                      data = spiny_counts_24)

summ(lob_24_poisson)
```

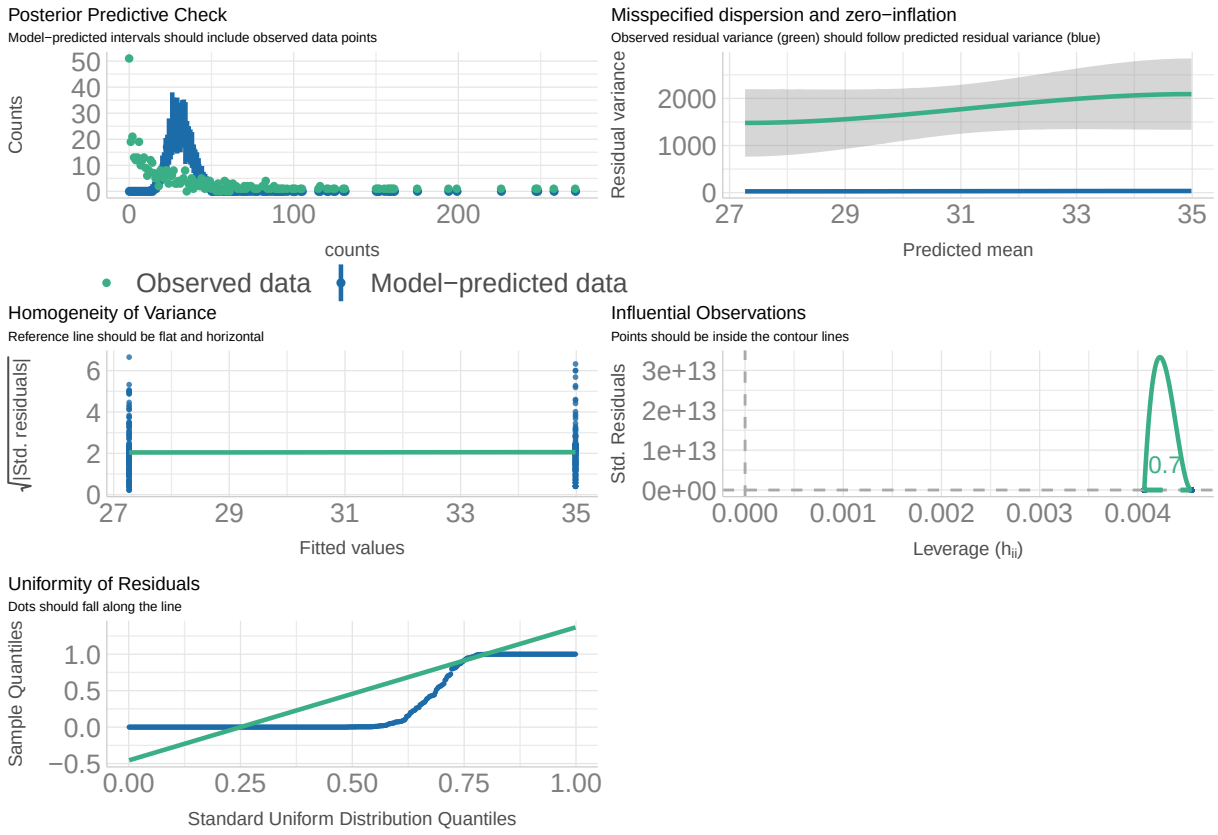
Observations	466
Dependent variable	counts
Type	Generalized linear model
Family	poisson
Link	log

```
check_model(lob_24_poisson, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```

$\chi^2(1)$	223.34
p	0.00
Pseudo-R^2 (Cragg-Uhler)	0.38
Pseudo-R^2 (McFadden)	0.01
AIC	21530.09
BIC	21538.38

	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




```
check_overdispersion(lob_24_poisson)
```

```
## # Overdispersion test
##
##        dispersion ratio =    57.103
##   Pearson's Chi-Squared = 26496.023
##                   p-value =   < 0.001
```

```
check_zeroinflation(lob_24_poisson)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 51
```

```
##   Predicted zeros: 0
##               Ratio: 0.00
```

```
lob_24_nb <- glm.nb(counts ~ treat, data = spiny_counts_24)

summ(lob_24_nb)
```

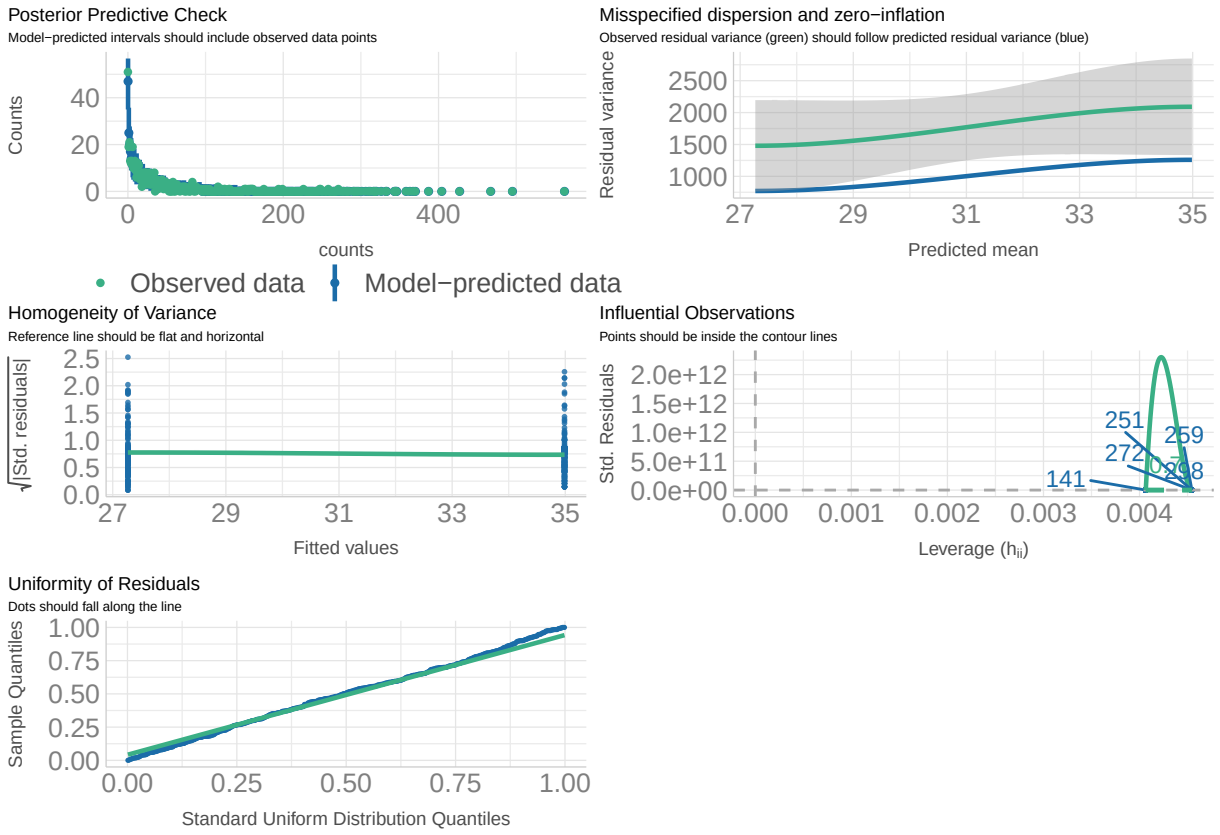
Observations	466
Dependent variable	counts
Type	Generalized linear model
Family	Negative Binomial(0.5769)
Link	log

$\chi^2(464)$	4.08
p	0.04
Pseudo-R^2 (Cragg-Uhler)	0.01
Pseudo-R^2 (McFadden)	0.00
AIC	4058.04
BIC	4070.48

	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

```
check_model(lob_24_nb, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```




```
check_overdispersion(lob_24_nb)
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.035
##           p-value = 0.808
```

```
check_zeroinflation(lob_24_nb)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 51
##   Predicted zeros: 47
##             Ratio: 0.91
```

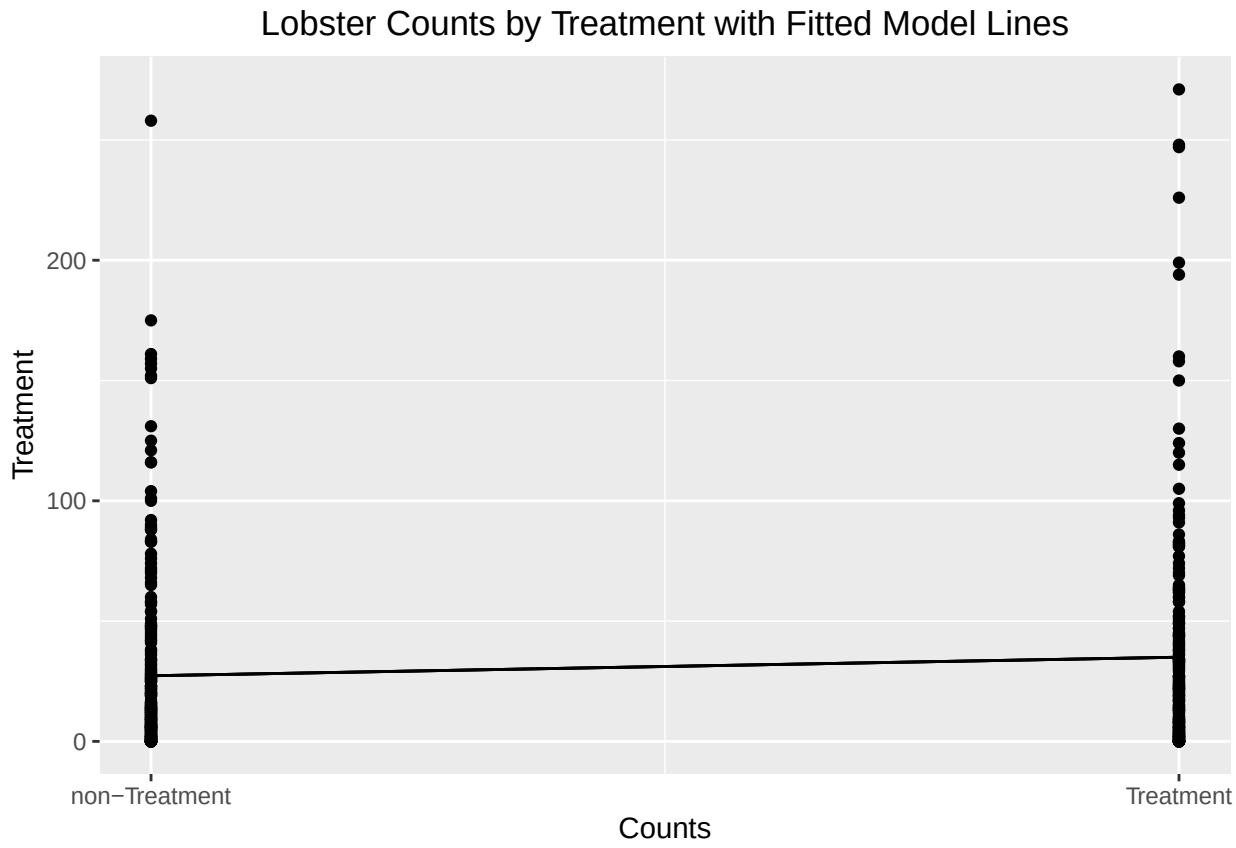
```
export_summs(lob_24_ols, lob_24_poisson, lob_24_nb,
             model.names = c("OLS","Poisson", "NB"),
             statistics = "none")
```

```
spiny_counts_24 %>%
    ungroup() %>%
    mutate(reg_ols = fitted(lob_24_ols),
           reg_poisson = fitted(lob_24_poisson),
           reg_nb = fitted(lob_24_nb)) %>%
    ggplot(aes(x = treat, y = counts)) +
    geom_point() +
    geom_line(aes(x = treat, y = reg_ols)) +
    geom_line(aes(x = treat, y = reg_poisson)) +
```

	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.

```
    geom_line(aes(x = treat, y = reg_nb)) +
    scale_x_continuous(breaks = c(0,1), labels = c('non-Treatment', 'Treatment')) +
    labs(x = 'Counts',
         y = 'Treatment',
         title = 'Lobster Counts by Treatment with Fitted Model Lines') +
    theme(plot.title = element_text(hjust = 0.5))
```



c. Compare and contrast results with the analysis from the 2012-2018 data sample (~ 2 paragraphs)

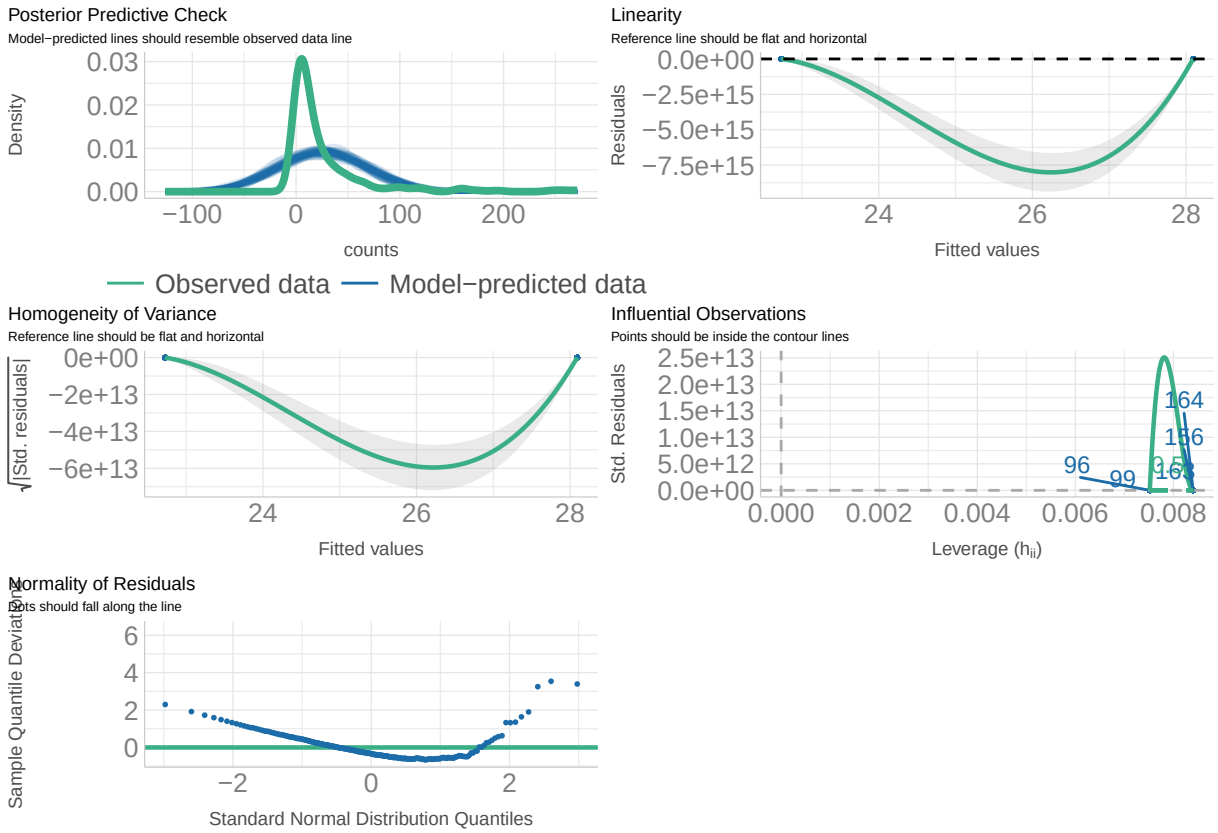
```
export_summs(m1_ols, lob_24_ols, m2_pois,
             lob_24_poisson, m3_nb, lob_24_nb,
             model.names = c("OLS 2018","OLS 2024", "Poisson 2018",
                             'Poisson 2024', 'NB 2018', 'NB 2024'),
             statistics = "none")
```

	OLS 2018	OLS 2024	Poisson 2018	Poisson 2024	NB 2018	NB 2024
(Intercept)	22.73 ***	27.27 ***	3.12 ***	3.31 ***	3.12 ***	3.31 ***
	(3.57)	(2.69)	(0.02)	(0.01)	(0.12)	(0.08)
treat	5.36	7.72 *	0.21 ***	0.25 ***	0.21	0.25 *
	(5.20)	(3.91)	(0.03)	(0.02)	(0.17)	(0.12)

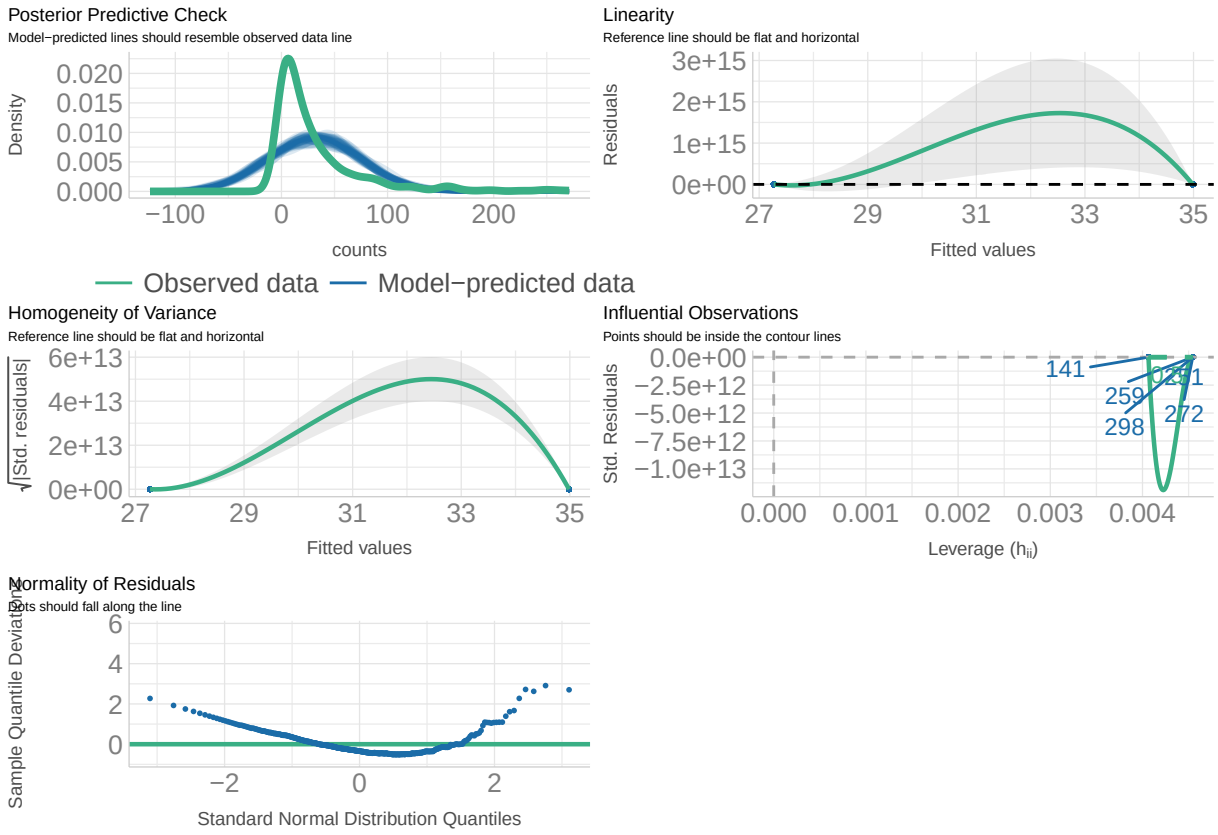
*** p < 0.001; ** p < 0.01; * p < 0.05.

OLS

```
check_model(m1_ols, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```

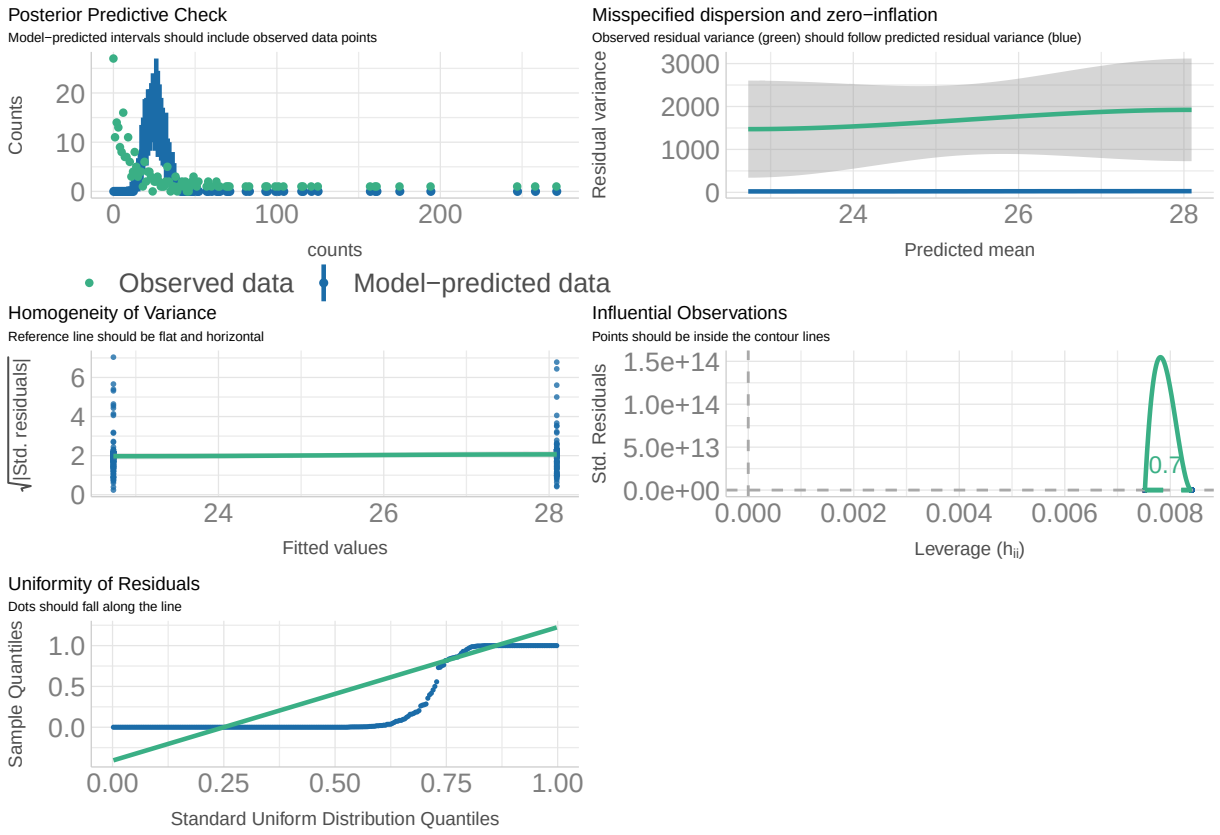



```
check_model(lob_24_ols, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```

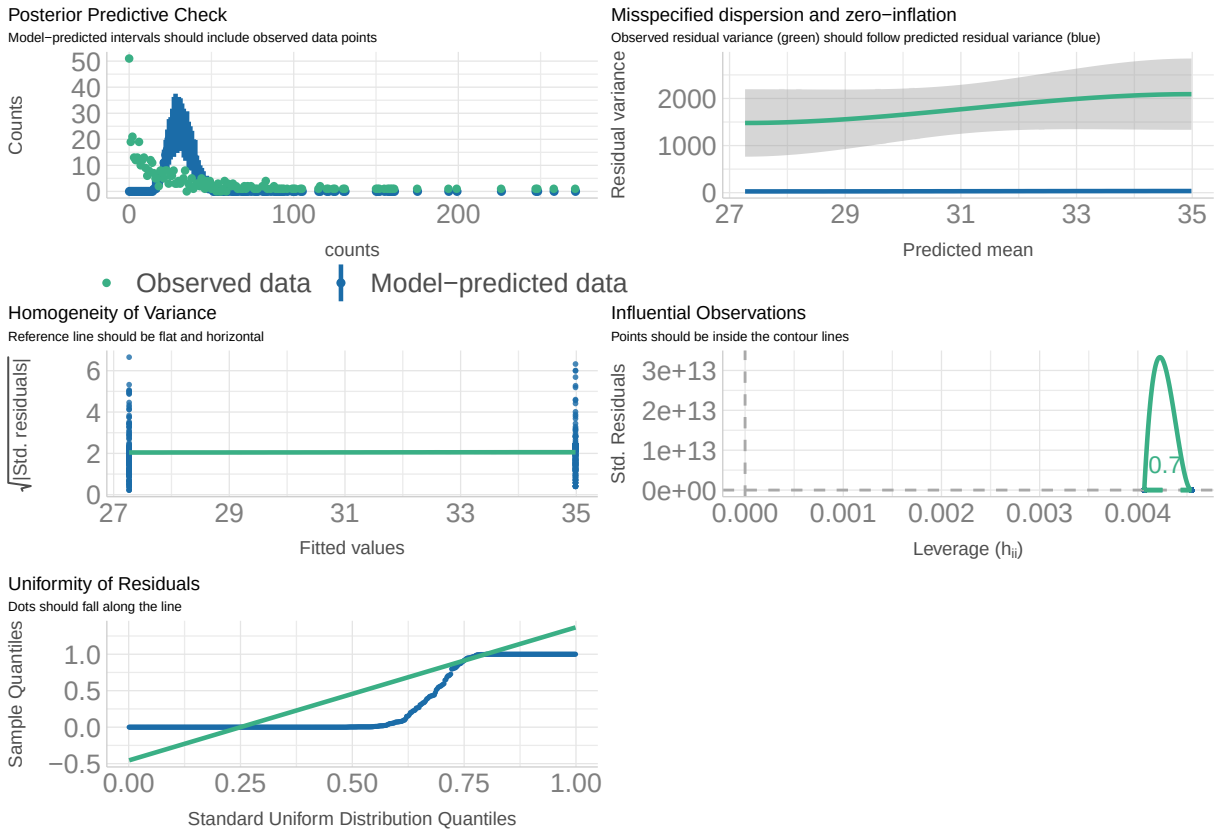


Poisson

```
check_model(m2_pois, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```

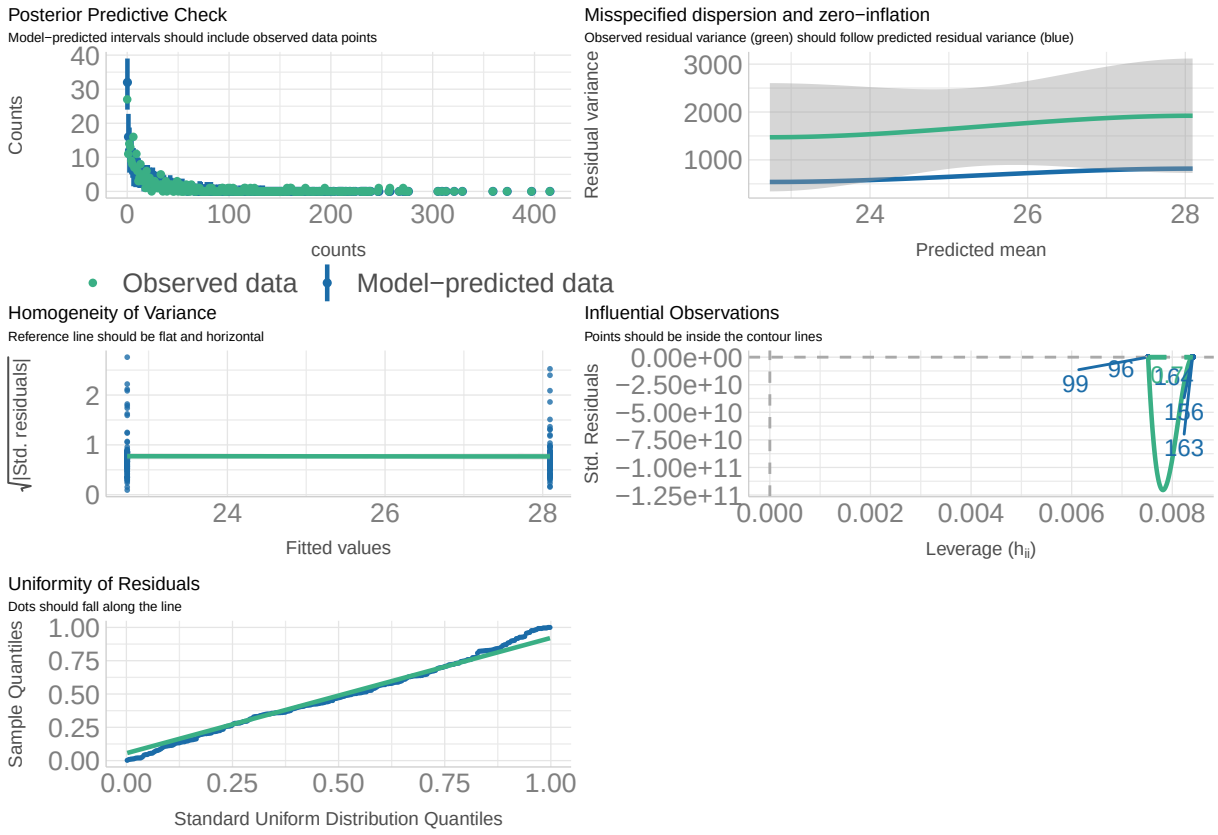



```
check_model(lob_24_poisson, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```

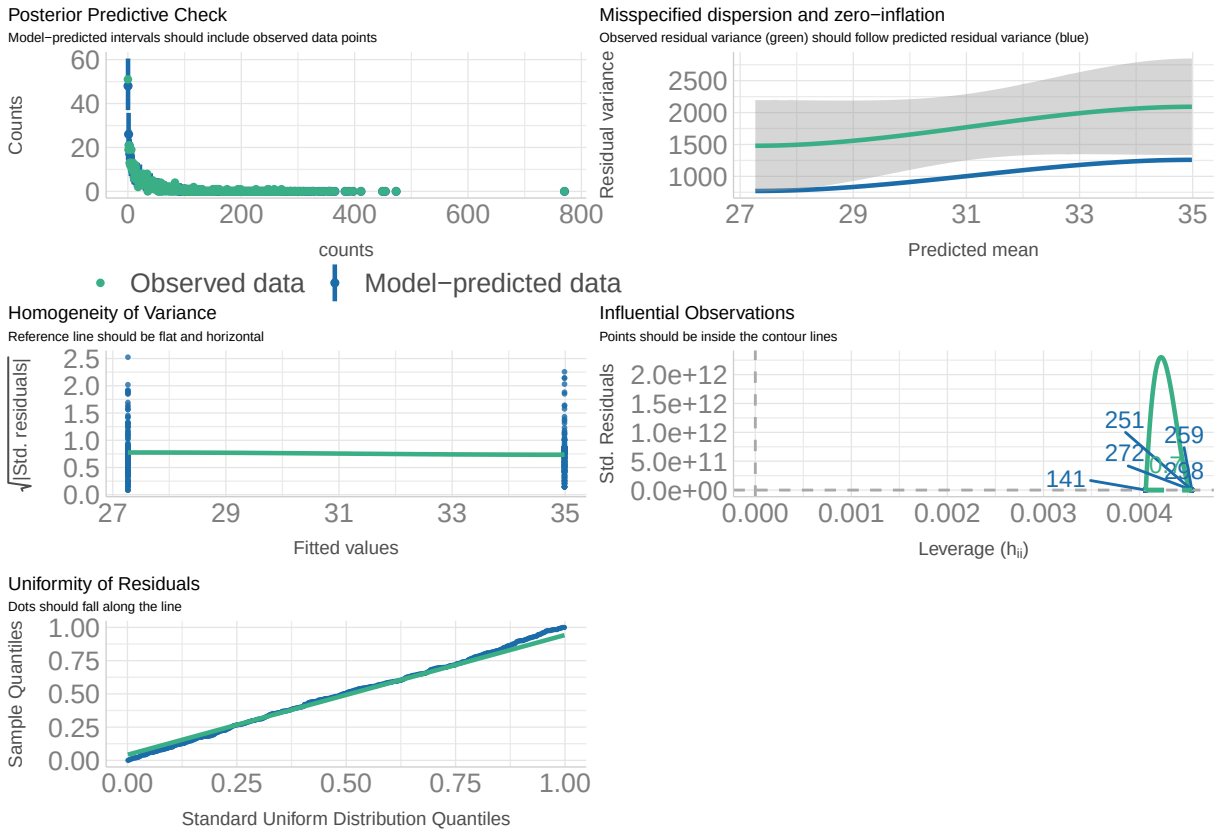


Negative Binomial

```
check_model(m3_nb, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```




```
check_model(lob_24_nb, title_size = 7, axis_title_size = 7, base_size = 5, dot_size = 1)
```



The to-2024 data seems to follow the same patterns as the to-2018 data, in the sense of the OLS being a
The most notable difference, however, was that the added data allowed the new negative binomial model t

---