Analysis of Ecosystems homework week 3

Data visualization

The solution

Graph critique

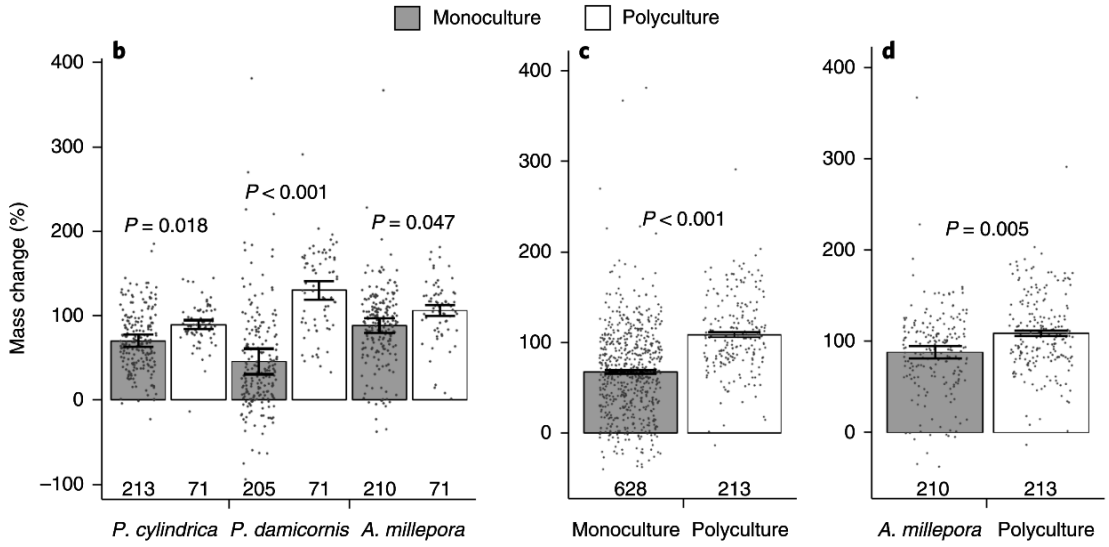


Figure 1: Clements & Hay. 2019. Biodiversity enhances coral growth, tissue survivorship and suppression of macroalgae. Nature ecology & evolution. DOI: https://doi.org/10.1038/s41559-018-0752-7

The **main issue** with these plots is that as the data are already *differences*—i.e., percent change—the dynamite plots obscure the fact that all but one bar include negative values. The extension of the bars into positive space give no indication that any observations were negative. The jittered data portray this, but the visual effect is still driven by the bars.
The data clearly have much wider distributions than the tight error bars suggest. The tight error bars and the P values are likely artifacts of large sample sizes.

Re-draw graphs

Load the data

```
corals <- read.csv(FilePath)
str(corals)
```

```
## 'data.frame':    841 obs. of  6 variables:
##  $ Plot          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Rep           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ CoralID       : int  1 2 3 4 5 6 7 8 9 1 ...
##  $ Species       : Factor w/ 3 levels "Acropora millepora",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Treatment     : Factor w/ 2 levels "Monoculture",..: 1 1 1 1 1 1 1 1 1 1 ...
```

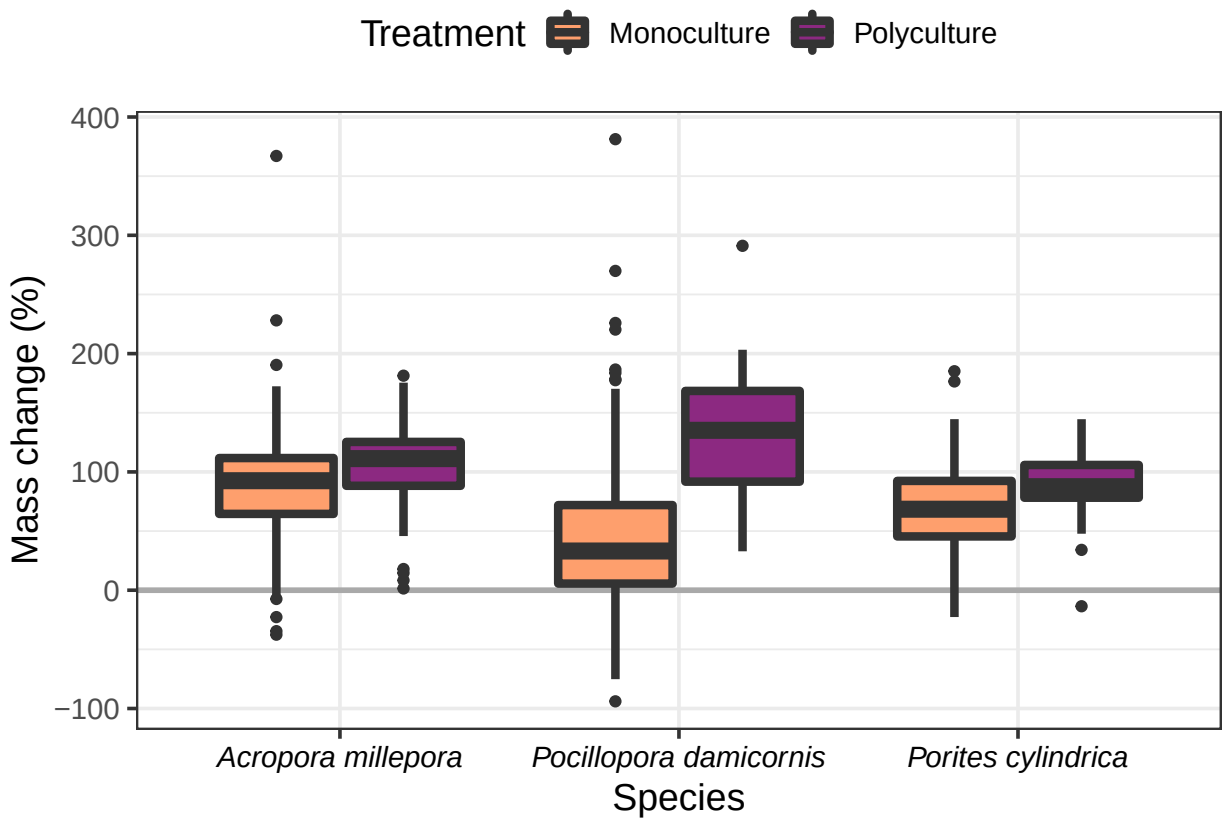
```
##  $ MassChangePrct: num  176.5 129.5 115.8 93.1 70.2 ...
```

`CoralID` is simply a number given to each coral in the plots so they can be monitored through time. The column that causes one to add to the critique of Fig. 1 is *Plot*. Replicates occur within plot, that's fine, but note that there are *multiple plots per species/treatment combination*. This means that the replicates—the individual points in Fig. 1—are not independent, but in Fig. 1 they are all plotted the same color.

Distribution graphs

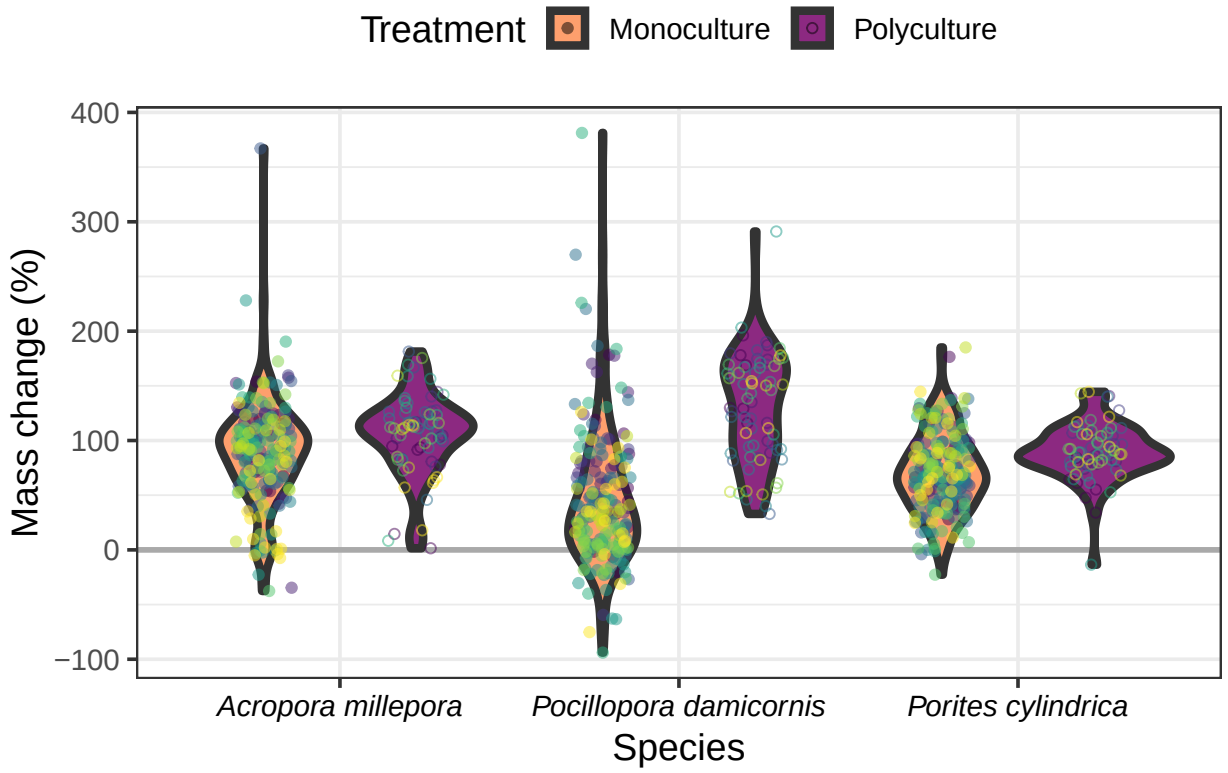
```
# Get a couple colours from the magma gradient in the viridis package
prov.col2 <- viridis::magma(n=2, begin=0.4, end=0.8, direction = -1)

ggplot(corals, aes(x=Species, y=MassChangePrct)) + theme_bw(14) +
  geom_hline(yintercept=0, color="darkgrey", size=1) +
  geom_boxplot(aes(fill=Treatment), size=1.5) +
  scale_fill_manual(values=prov.col2) +
  labs(y="Mass change (%)") +
  theme(legend.position = "top",
        axis.text.x = element_text(color="black", face="italic"))
```




```
ggplot(corals, aes(x=Species, y=MassChangePrct)) + theme_bw(14) +
  geom_hline(yintercept=0, color="darkgrey", size=1) +
  geom_violin(aes(fill=Treatment), size=1.5) +
  geom_jitter(aes(shape=Treatment, color=Rep),
              alpha=0.5, # color="black",
```

```
            position = position_jitterdodge(jitter.width=0.33,
                                            dodge.width = 0.9)) +
  scale_shape_manual(values=c(19, 1)) +
  scale_color_viridis(guide=FALSE) +
  scale_fill_manual(values=prov.col2) +
  labs(y="Mass change (%)",
       caption= "Colors denote unique plots.") +
  theme(legend.position = "top",
        axis.text.x = element_text(color="black", face="italic"))
```



Both the boxplots and violin plots highlight the two critiques of Fig. 1 given above: They reveal the breadth of data distributions, call attention to potentially-skewing outliers, and offer emphasis on the fact that many observations were negative. The jittered points are colored by Plot to indicate potential clusters—and overall non-independence—in the data.

Calculate summary statistics

```
ddply(corals, .(Species, Treatment),
              summarize,
              mean = mean(MassChangePrct),
              se = sd(MassChangePrct)/sqrt(length(MassChangePrct)) ) %>%
  knitr::kable(digits=1, caption="Summary statistics for the coral data.")
```

Table 1: Summary statistics for the coral data.

Species	Treatment	mean	se
Acropora millepora	Monoculture	88.0	3.2
Acropora millepora	Polyculture	106.0	4.3
Pocillopora damicornis	Monoculture	44.0	4.2
Pocillopora damicornis	Polyculture	129.9	5.8
Porites cylindrica	Monoculture	70.3	2.4
Porites cylindrica	Polyculture	89.8	3.0

Note how easily this nicely-formatted table was created by piping the results of `ddply` into the `kable` function, available in the `knitr` package.

Graph the mean effect

```
  corals %>%
    group_by(Species, Treatment) %>%
      summarize(mean = mean(MassChangePrct),
                se = sd(MassChangePrct)/sqrt(length(MassChangePrct)),
                count = n() ) %>%
  ggplot(aes(x=Species, y=mean, color=Treatment)) + theme_bw(14) +
    geom_hline(yintercept=0, color="darkgrey", size=1) +
    geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
                  size=1.5, width = 0.25,
                  position= position_dodge(0.5)) +
    geom_point(aes(shape=Treatment, fill=Treatment),
                color="black", size=4, stroke=1.5,
                  position= position_dodge(0.5)) +
    scale_shape_manual(values=c(21, 24)) +
    scale_color_manual(values=prov.col2) +
    scale_fill_manual(values=prov.col2) +
    labs(y="Mean mass change (%)") +
    coord_cartesian(ylim = c(0,150)) +
    geom_text(aes(y=10, group=Treatment, label=count),
              color="black",
              position = position_dodge(0.5)) +
    theme(legend.position = "top",
          axis.text.x = element_text(color="black", face="italic")) -> means

RepMeans <-
  corals %>%
    group_by(Species, Treatment, Rep) %>%
      summarize(mean = mean(MassChangePrct),
                se = sd(MassChangePrct)/sqrt(length(MassChangePrct)) )
TreatMeans <-
corals %>%
    group_by(Species, Treatment) %>%
      summarize(mean = mean(MassChangePrct),
                se = sd(MassChangePrct)/sqrt(length(MassChangePrct)),
                count = n() )
```

```
  ggplot(data=RepMeans,
         aes(x=Species, y=mean, color=Treatment)) + theme_bw(14) +
    geom_hline(yintercept=0, color="darkgrey", size=1) +
    geom_linerange(data=RepMeans,
                   aes(ymin=mean-se, ymax=mean+se,
                       group=interaction(Rep, Treatment)),
                  size=1.5, alpha=0.25,
                 position= position_jitterdodge(dodge.width=0.5, seed=2)) +
    geom_point(data=RepMeans,
               aes(color=Treatment,
                    group=interaction(Rep, Treatment)),
               size=4, stroke=1.5, alpha=0.25,
                  position= position_jitterdodge(dodge.width=0.5, seed=2)) +
    scale_shape_manual(values=c(21, 24)) +
    scale_color_manual(values=prov.col2) +
    scale_fill_manual(values=prov.col2) +
    labs(y="Mean mass change (%)") +
    geom_errorbar(data = TreatMeans,
        aes(x=Species, ymin=mean-se, ymax=mean+se, color=Treatment),
         size=1.5, width=0.25,
        position = position_dodge(width=0.5)) +
    geom_point(data=TreatMeans,
           aes(fill=Treatment,
               shape=Treatment),
           color="black", size=5, stroke=1.5,
        position = position_dodge(width=0.5)) +
    geom_text(data=TreatMeans,
              aes(y=10, group=Treatment, label=count),
              color="black",
              position = position_dodge(0.5)) +
    theme(legend.position = "top",
          axis.text.x = element_text(color="black", face="italic")) -> means2
# Some advanced ggplotting using packages gridExtra and cowplot
 # Prepare Grobs
  grid.grob <- arrangeGrob(means + theme(legend.position = "none",
                                         axis.title=element_blank()),
                           means2 + theme(legend.position = "none",
                                         axis.title=element_blank()))
# create common labels
  x.grob <- textGrob("Species",
                     gp=gpar(fontface="bold", fontsize=14))
  y.grob <- textGrob("Mean (± s.e.) mass change (%)", rot=90,
                  gp=gpar(fontface="bold", fontsize=14))

# Combine panels and labels
  plot.grob <- grid.arrange(arrangeGrob(grid.grob, left = y.grob,
                                                   bottom = x.grob))
# Grab a legend
  legend.grob <- cowplot::get_legend(means + theme(legend.position="top"))

# Combine labeled plot grid & legend
  plot_grid(legend.grob, plot.grob,
            ncol=1, rel_heights = c(0.1, 0.9))
```

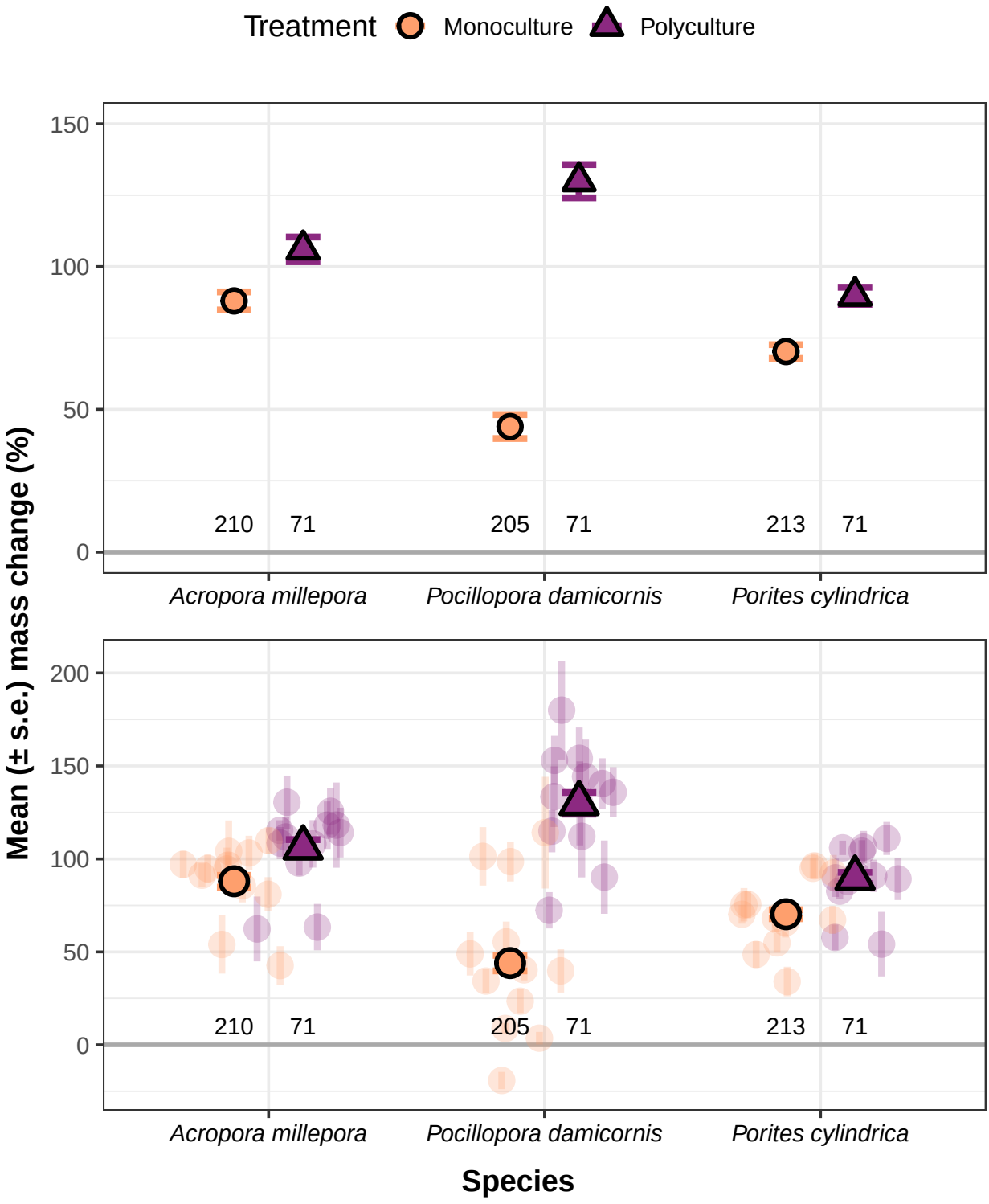


Figure 2: Two takes on plotting the mean. The bottom graph shows the same means and standard errors as the first, but overlaid on plot means that show the variability among plots within each treatment group.

Reflection

Of the two graphs above, the bottom one is a better way to present the means of these non-independent data. It is arguably disingenuous to show an overall mean for each treatment when in fact the study design has an obvious hierarchy: the sample sizes don't reflect just a bunch of corals, but instead the corals were grouped as replicates within discrete plots. We might then be interested in variability among plots, as well as among treatments.
The mean $\pm$ s.e. graph is technically the best replacement for Fig. 1 because the original paper intends to use the graph to illustrate the comparison between treatments in support of the reported P-values. But the mean $\pm$ s.e. graphs only improve upon the original dynamite plots in conveying the same information but with less (digital) ink. Negative values are still obscured. The most publication-friendly figure would be the boxplot with outliers or even violin plots with symbols denoting the means, which is what the statistical results are testing. I would not include jittered data in the publication. The figure legend should specify that the reported P-values are based on a statistical model that properly accounts for the non-indpendence of data from different plots (the caption in the original figure includes this information).