Presenting your data: making figures in R

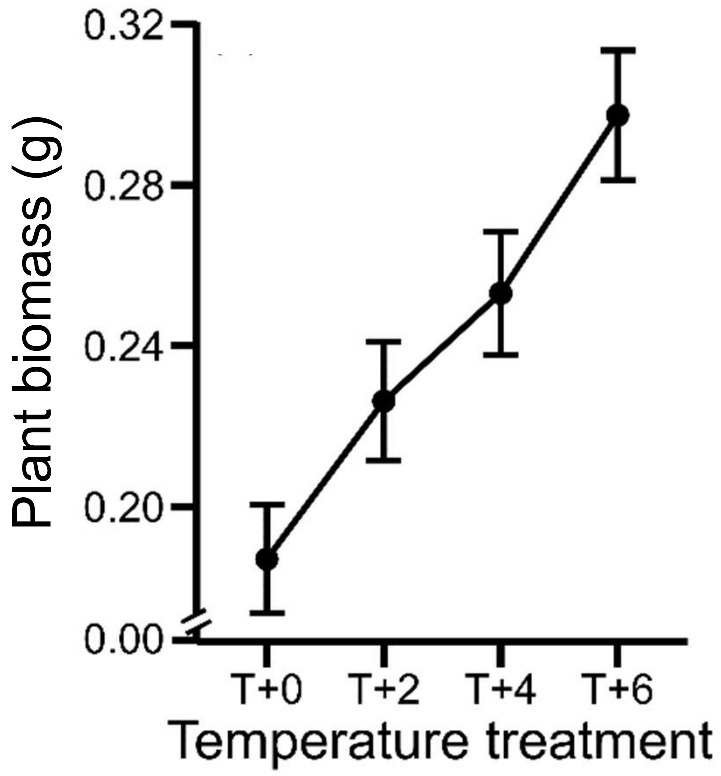
Emily Haeuser

Eva Malecore

Why are good figures important?

- Compared to text and tables, figures are:
  - Easier to understand
  - Easier to remember
  - More direct
  - More interesting
  - More persuasive

		Plant biomass (g)	
Fixed Effects	df	LRT	P
Initial size	1	17.219	<0.001
Temperature	1	13.289	<0.001
Random Effects	**Std. Dev**		**Levels**
Species	0.139		10

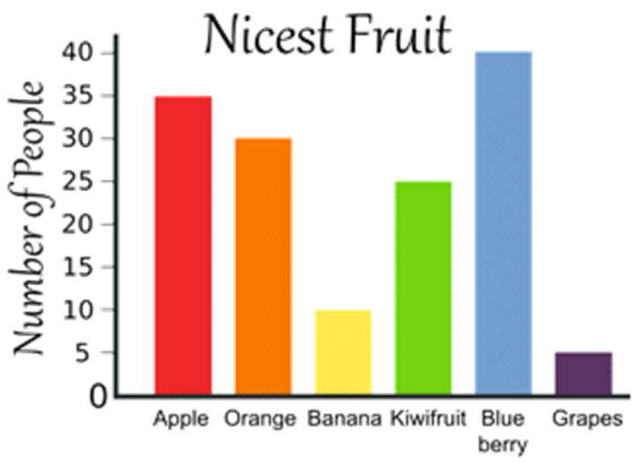


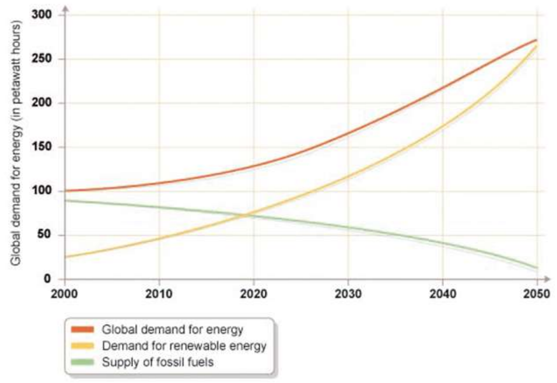
When making figures, consider:

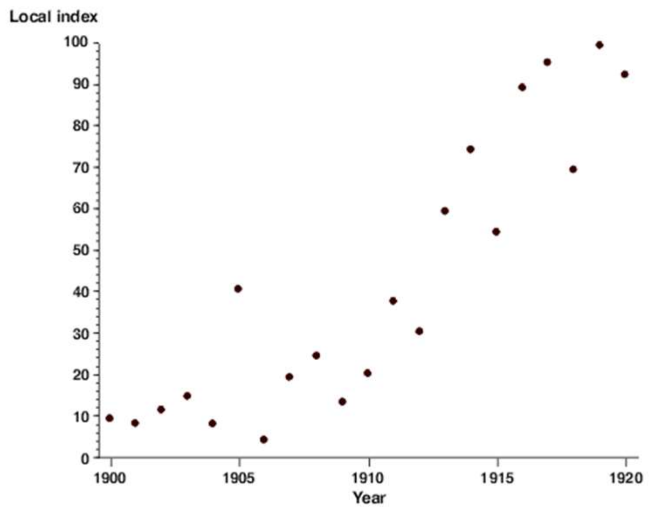
- What message do you want to convey?
- What is your response variable?
  - What kind of data is it?
- What are your predictor variables?
  - How many do you have?
  - What kind of data are they?
- Who is your audience?
  - How long will they have to digest the information?

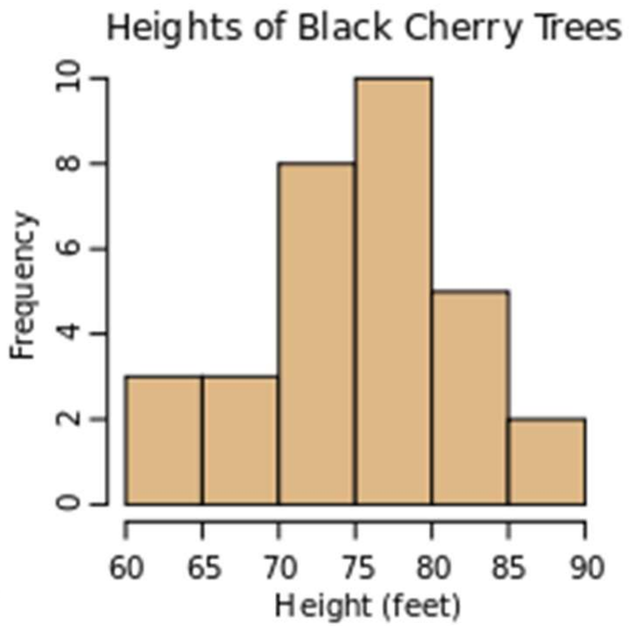
Common plot types

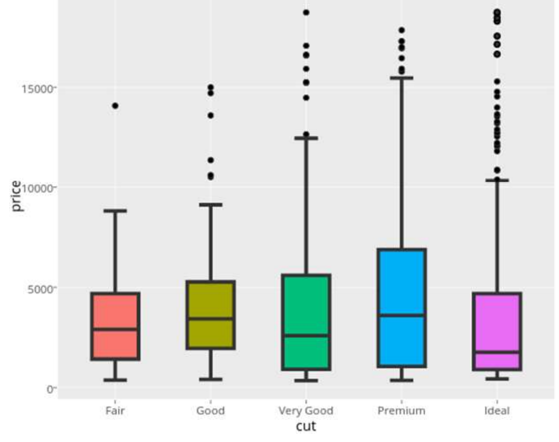
- Bar
- Line
- Point
- Histogram
- Boxplot











Always include:

- Axis labels
- Units
- Legend (usually)
- Information on both means and variances
- Informative caption

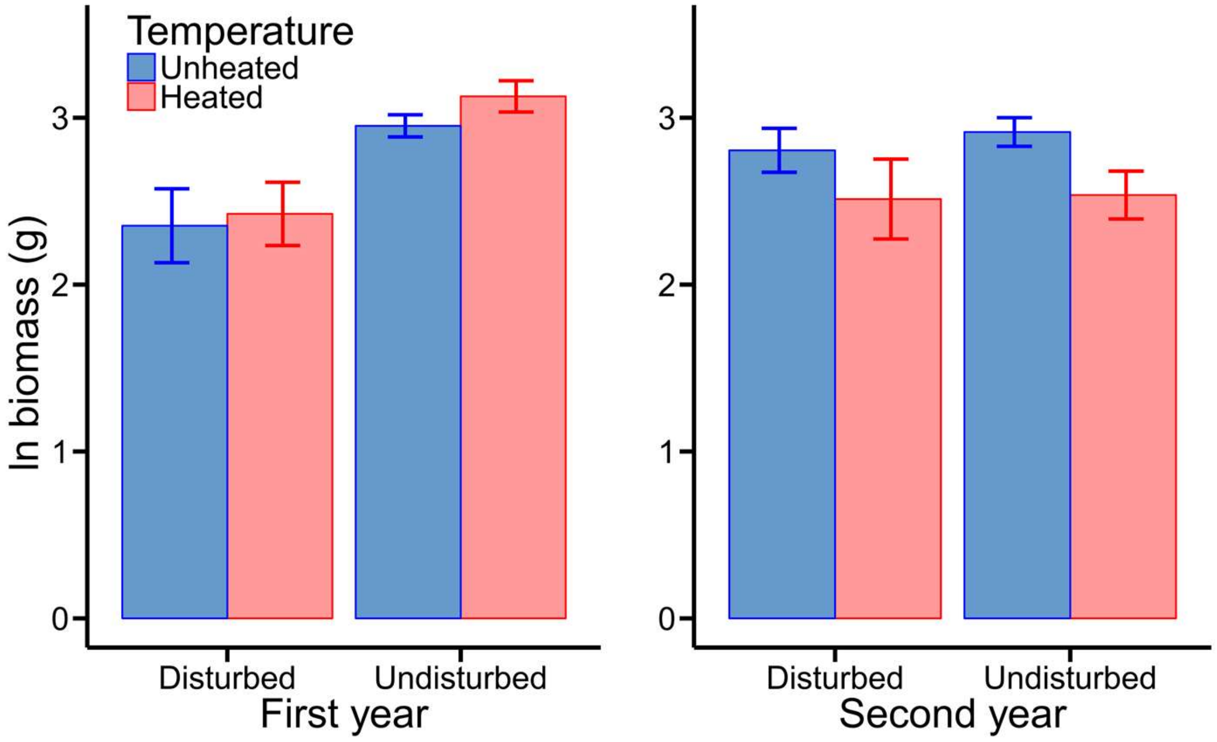
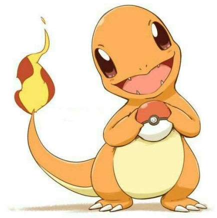


Figure 1: Plant biomass (log-transformed) during the first (left) and second (right) years of the study, in disturbed and undisturbed, as well as heated and unheated treatments



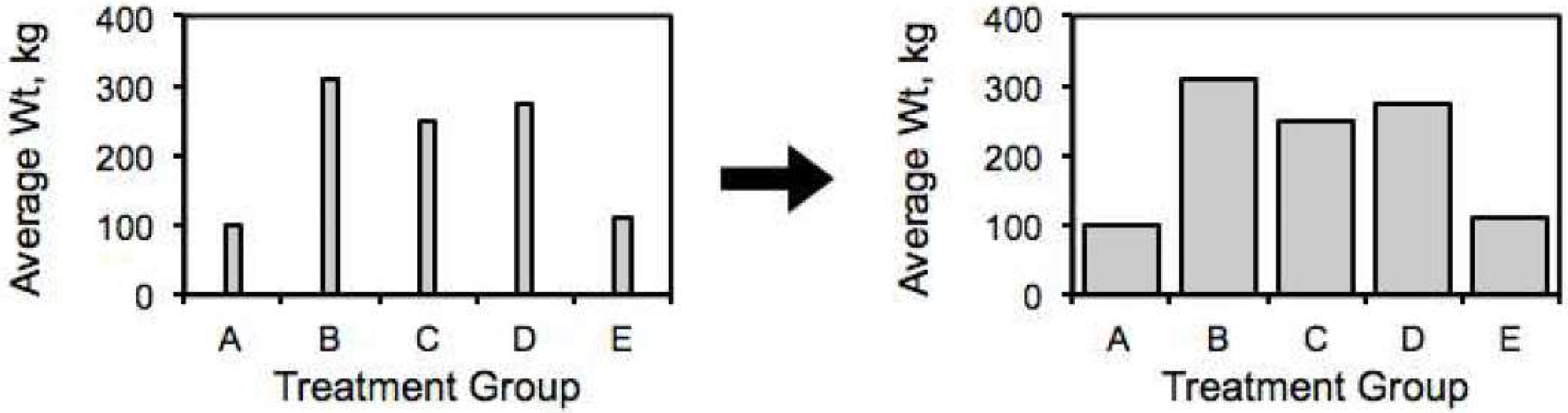
Identify these features on this graph?

Some principles of good graphing

- Make the data stand out
- Minimize clutter—less is more
  - Avoid irrelevant and uninformative additions
- Scale of axis values should fit with your data
- If plot symbols overlap, they should be visually distinguishable
- Captions should be comprehensive and informative

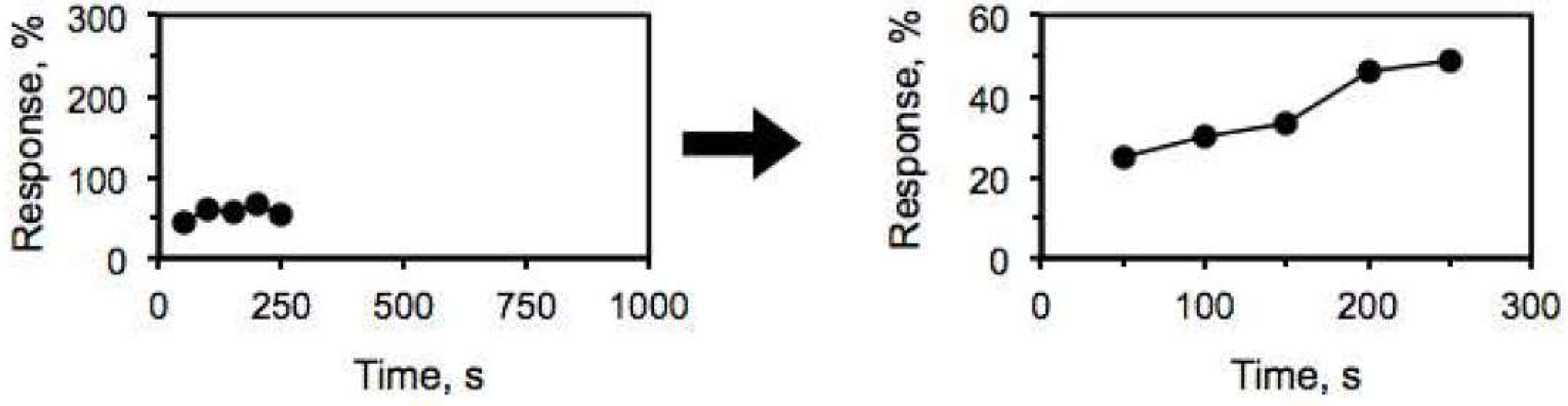
Bad graphs

Bad graphs...



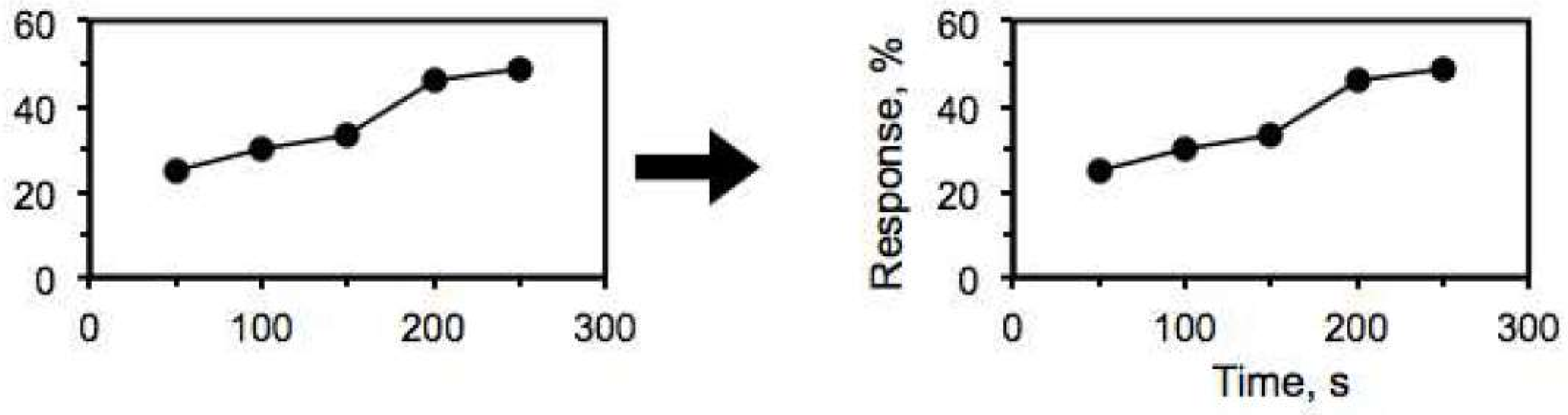
...misuse space

Bad graphs...



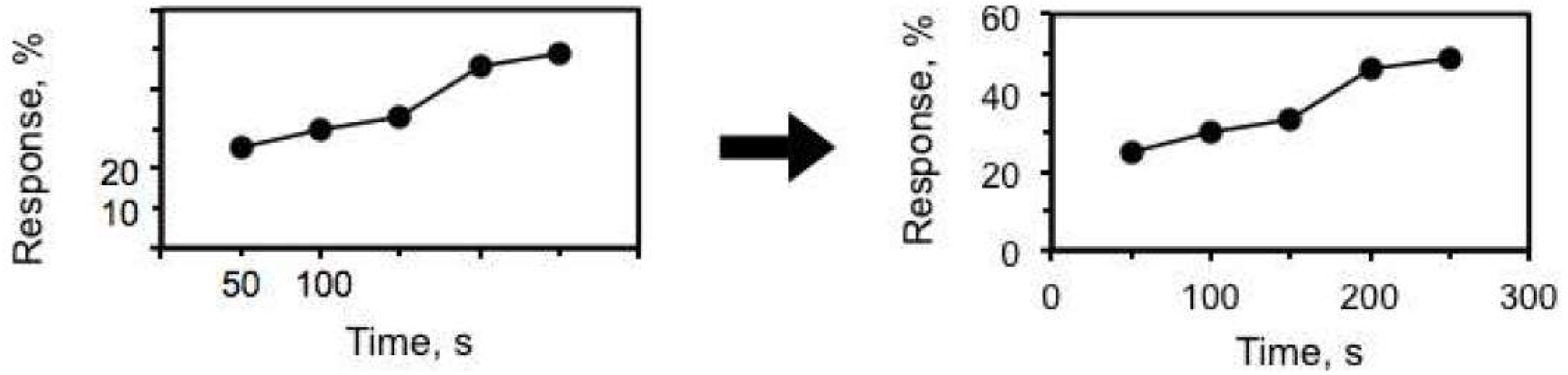
...misuse scale

Bad graphs...



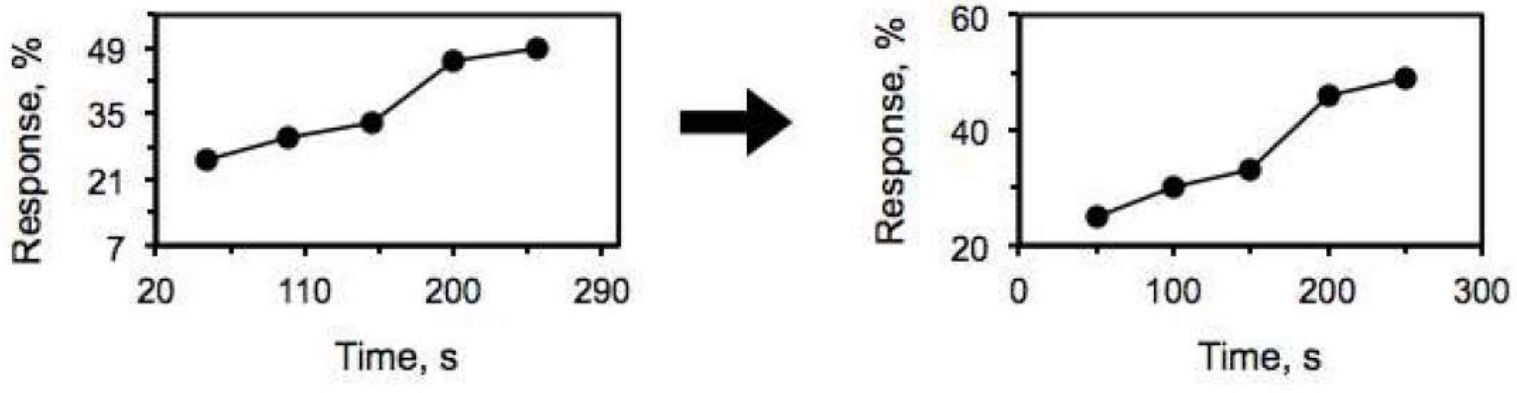
...lack labels

Bad graphs...



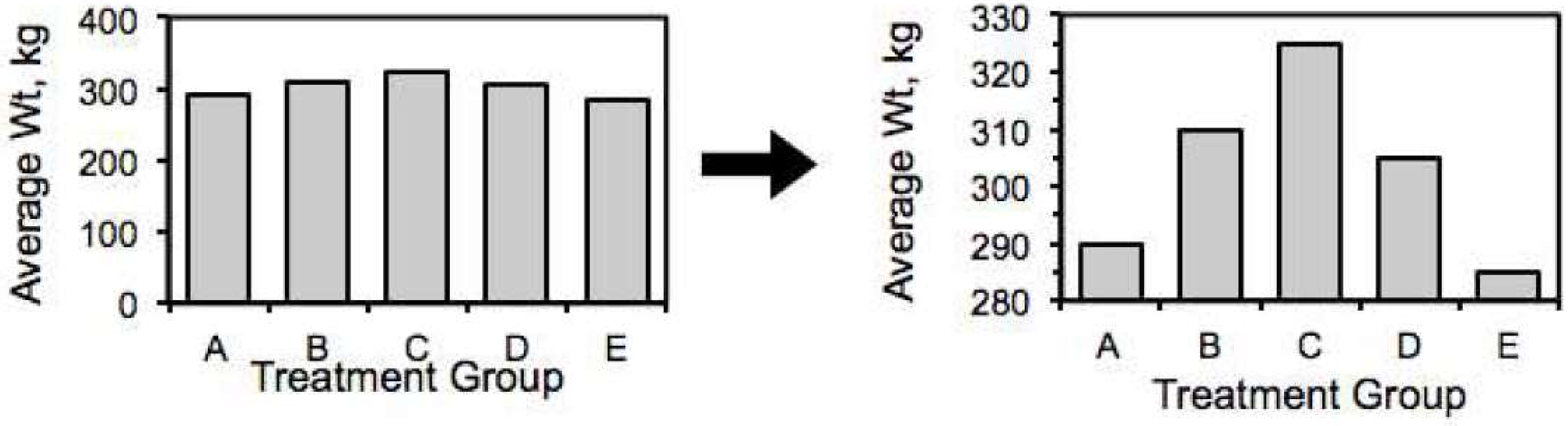
...use incomplete labels

Bad graphs...

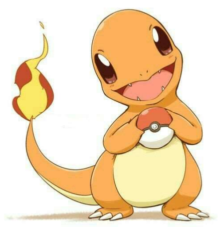


...use inappropriate labels

Bad graphs...

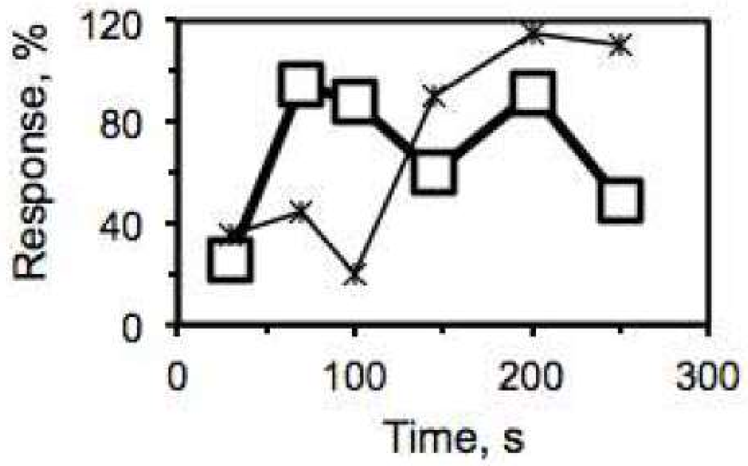


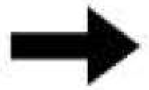
...hide meaningful differences

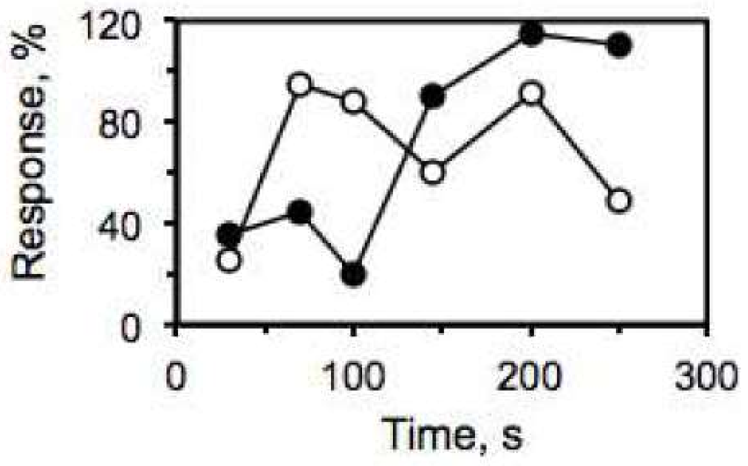


But… what information is still missing from the improved figure?

Bad graphs...

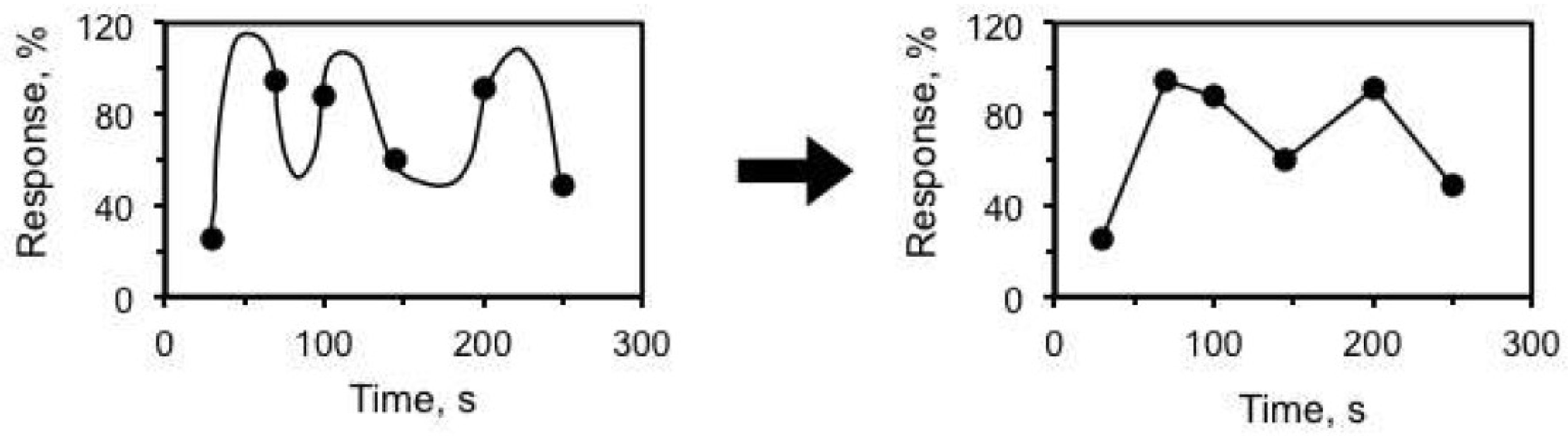






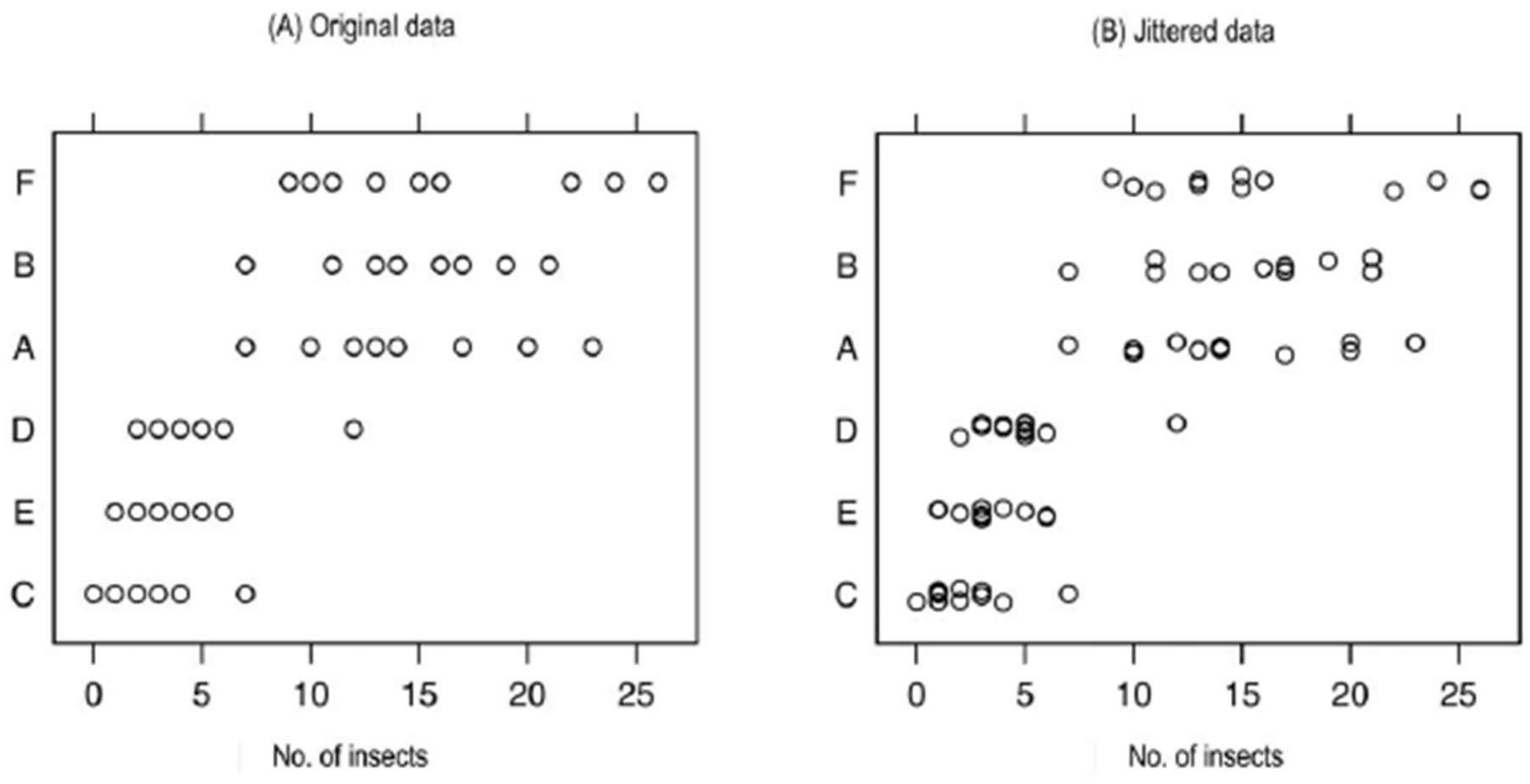
...overcomplicate symbology

Bad graphs...

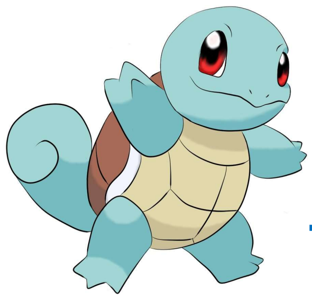


...misrepresent the data

Bad graphs...



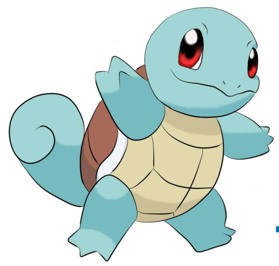
...present incomplete data



Graphing in R

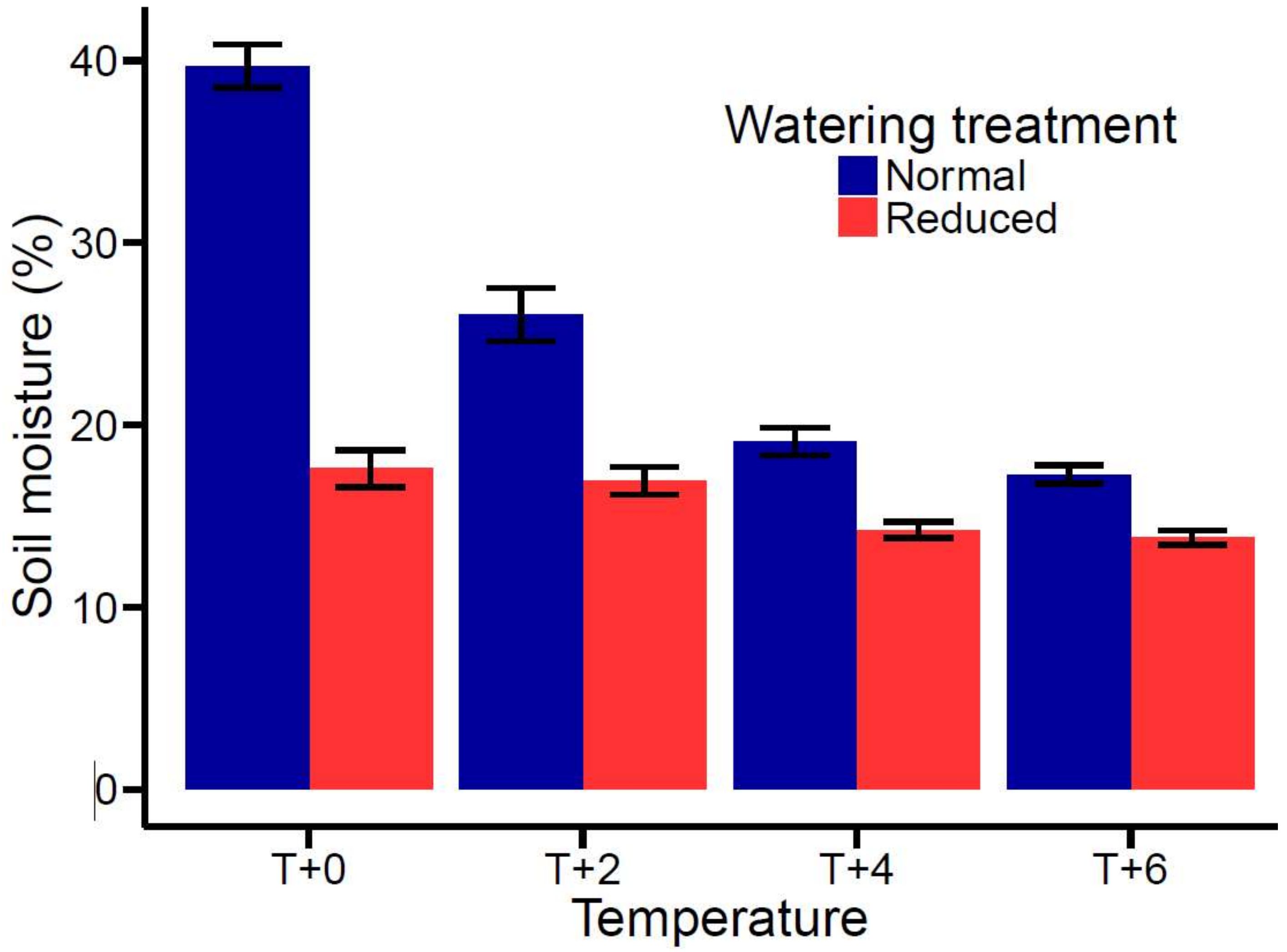
Graphing package: '*ggplot2*'

- For producing handsome, customized, publication-quality plots
- Approach graphs from a visual perspective, considers how each component of the data is represented on the final plot
- Superimpose multiple layers (points, lines, maps, tiles, box plots) from different data sources with automatically adjusted common scales
- Add customizable smoothers that use powerful modeling capabilities of R, such as loess, linear models, generalized additive models, and robust regression



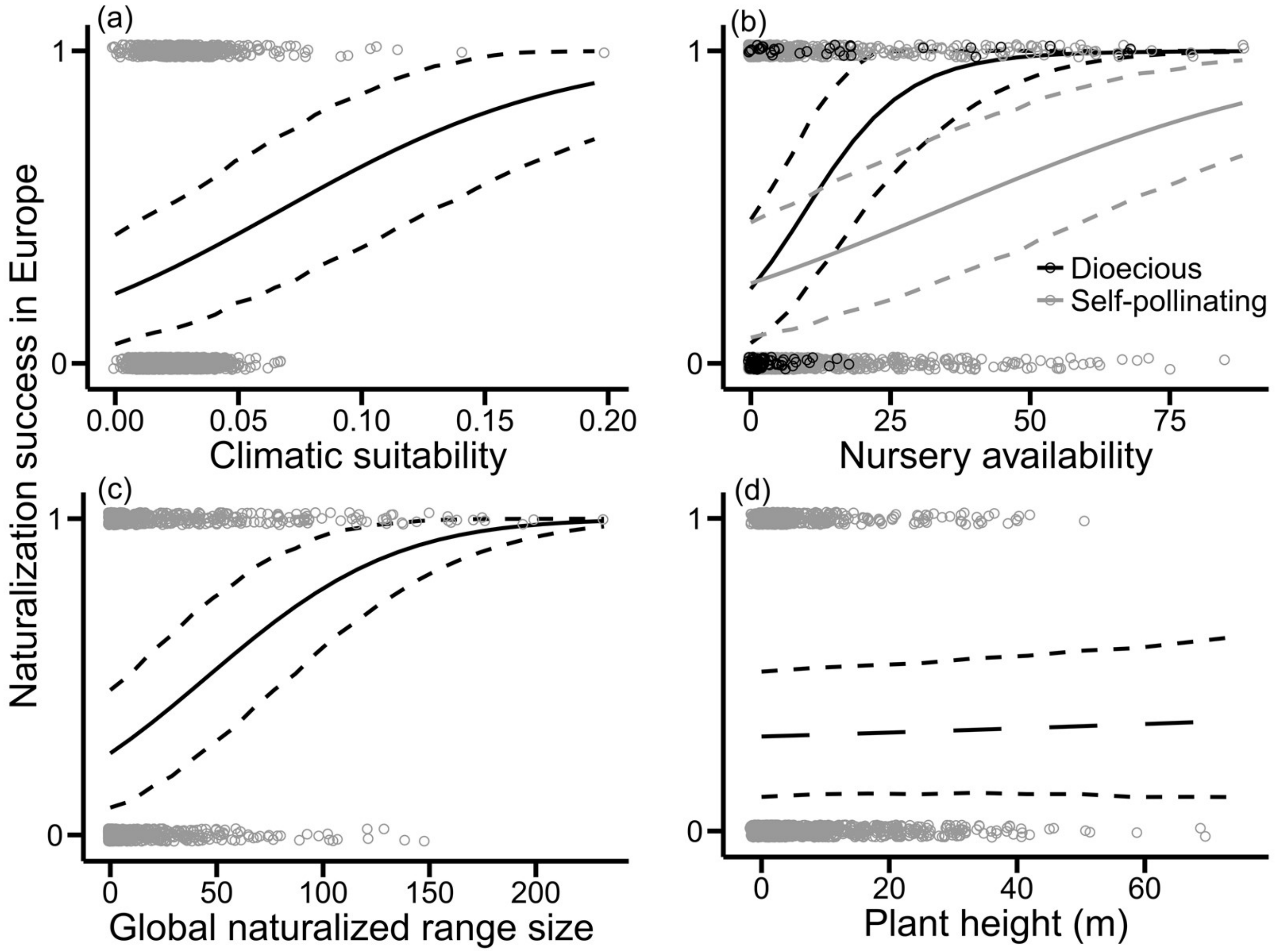
ggplot2 figure examples

Effects of a watering and warming treatments on soil moisture: Haeuser *et al.* 2018



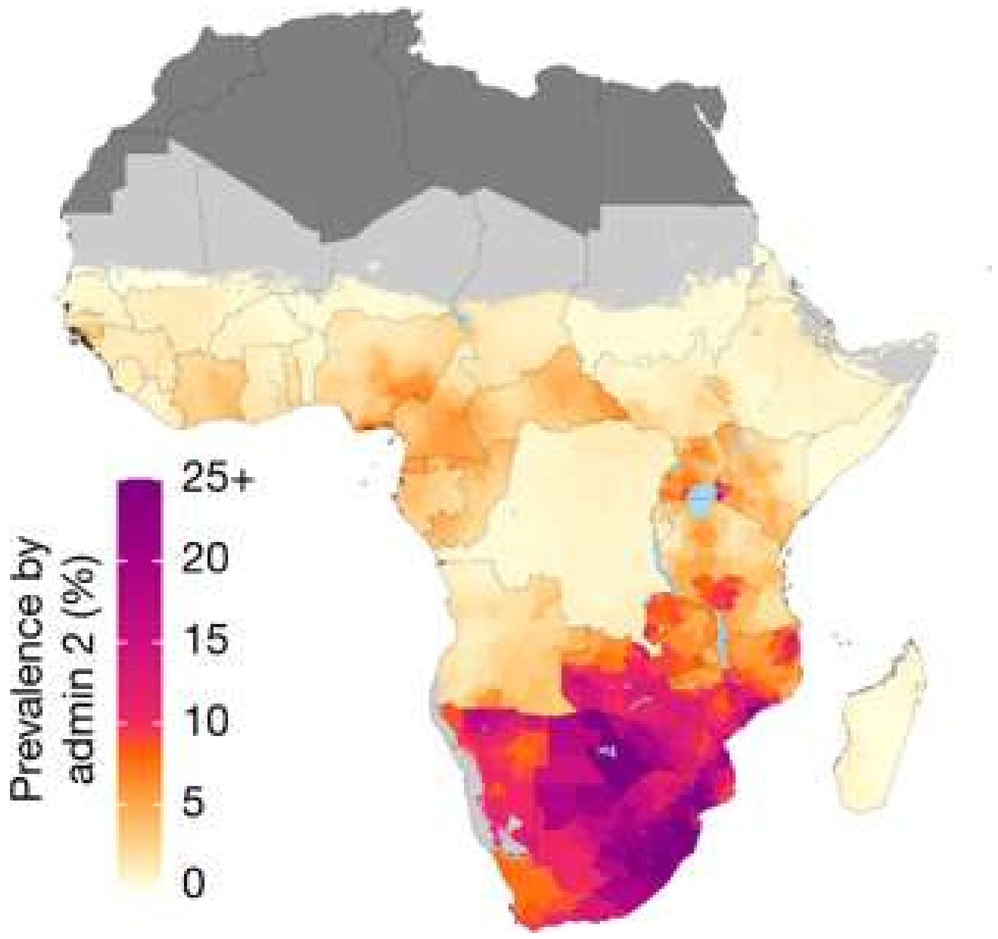
ggplot2 figure examples

Effects of different variables on alien plant naturalizations: Haeuser *et al.* 2018

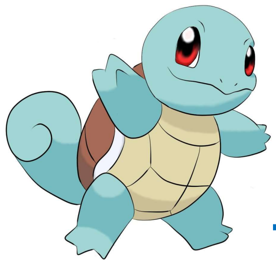


ggplot2 figure examples

HIV prevalence in Africa in 2017: Dwyer-Lindgren *et al.* 2019

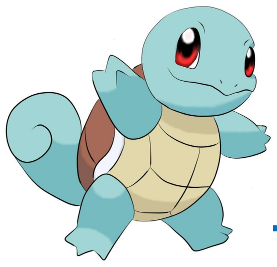


```
a<-ggplot(data=fitCMed,
aes(x=CMed_orig/1000, y=fit))+
  geom_line(size=1)+
  geom_line(lty=2, size=1, aes(y = lwr))+
  geom_line(lty=2, size=1, aes(y = upr))+
  geom_jitter(data=dat_minad,
aes(x=curmedian/1000, y=Nat_in_E),
              shape=1, size=2,
color="#999999", width = 0.005,
height=0.02)+
  theme_bw() +

scale_y_continuous(breaks=number_ticks(
1), limits=c(-0.03, 1.03))+
  theme(text = element_text(size=20)) +
  theme(panel.grid.major =
element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border = element_blank(),
        axis.line=element_line(size=1.2),
        axis.ticks.length=unit(0.3, "cm"),
        axis.ticks = element_line(size = 1.2,
color="#000000"))+

theme(axis.text=element_text(colour="blac
k"))+
  xlab("Median climate suitability") +
  ylab("European naturalization success")
b<-ggplot(data=fitHt, aes(x=Ht_orig,
y=fit))+
  geom_line(size=1)+
  geom_line(lty=2, size=1, aes(y = lwr))+
  geom_line(lty=2, size=1, aes(y = upr))+
  geom_jitter(data=dat_minad,
aes(x=Height, y=Nat_in_E),
              shape=1, size=2,
color="#999999", width = 2, height=0.02)+
  theme_bw() +

scale_y_continuous(breaks=number_ticks(
  theme(text = element_text(size=20)) +
  theme(panel.grid.major =
element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border = element_blank(),
        axis.line=element_line(size=1.2),
        axis.ticks.length=unit(0.3, "cm"),
        axis.ticks = element_line(size = 1.2,
color="#000000"))+

theme(axis.text=element_text(colour="blac
k"))+
  xlab("Maximum plant height") +
  ylab("")
c<-ggplot(data=fitNGR, aes(x=NGR_orig,
y=fit))+
  geom_line(size=1)+
  geom_line(lty=2, size=1, aes(y = lwr))+
  geom_line(lty=2, size=1, aes(y = upr))+
  geom_jitter(data=dat_minad,
aes(x=NE_gl_reg, y=Nat_in_E),
             shape=1, size=2,
color="#999999", width = 1, height=0.02)+
  theme_bw() +

scale_y_continuous(breaks=number_ticks(
1), limits=c(-0.03, 1.03))+
  theme(text = element_text(size=20)) +
  theme(panel.grid.major =
element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border = element_blank(),
        axis.line=element_line(size=1.2),
        axis.ticks.length=unit(0.3, "cm"),
        axis.ticks = element_line(size = 1.2,
color="#000000"))+

theme(axis.text=element_text(colour="blac
k"))+
  xlab("Global naturalization extent") +
d<-ggplot(data=fitVDV, aes(x=VDV_orig,
y=fit, color=as.factor(Mono_orig)))+
  geom_line(size=1)+
  geom_line(lty=2, size=1, aes(y = lwr))+
  geom_line(lty=2, size=1, aes(y = upr))+
  geom_jitter(data=dat_minad,
aes(x=VDV_all, y=Nat_in_E,
color=as.factor(Mono)),
              shape=1, size=2,  width = 0.5,
height=0.02)+
  theme_bw() +

scale_y_continuous(breaks=number_ticks(
1), limits=c(-0.03, 1.03))+
  theme(text = element_text(size=20)) +
  theme(panel.grid.major =
element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border = element_blank(),
        axis.line=element_line(size=1.2),
        axis.ticks.length=unit(0.3, "cm"),
        axis.ticks = element_line(size = 1.2,
color="#000000"))+

theme(axis.text=element_text(colour="blac
k"))+
  xlab("Nurseries") +
  ylab("")+
  theme(legend.position = c(0.8, 0.3),
        legend.title=element_blank())+
  scale_color_manual(values=c("#000000",

"#000000","#999999","#999999" ),
                    labels=c("Dioecious",
"Dioecious",
                           "Monoecious",
"Monoecious"))


library(gridExtra)
```



Basic parts of *ggplot2* code

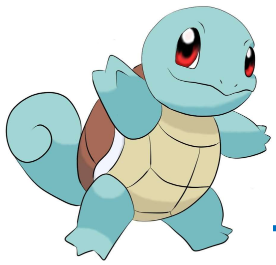
- **Your data**
  - Dataframe to work from
  - X & Y variables
  - Any grouping variables
- **Your chosen graphic type(s) (e.g. boxplot, point, line)**
- Axis labels
- Plot styles
- Etc.: Font sizes, axis lengths, annotations, facets, more



Basic parts of *ggplot2* code

1. Identify your data
   a) Dataframe to work from
   b) X & Y variables
   c) Any grouping variables

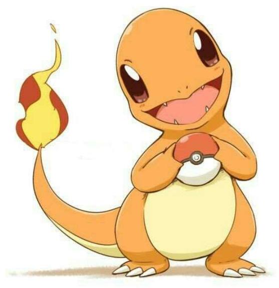
```
ggplot(DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE))
```



Basic parts of *ggplot2* code

1. Identify your data
   a) Dataframe to work from
   b) X & Y variables
   c) Any grouping variables
2. Identify your chosen graphic type(s) (e.g. boxplot, point, line, bar, errorbar)

```
ggplot(DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE))
+
geom_GRAPHICTYPE()
```



Time for an exercise

- With our leaf traits dataset, make a plot showing:
  - The relationship between species’ N-fixing ability (yes/no), and leaf N-content

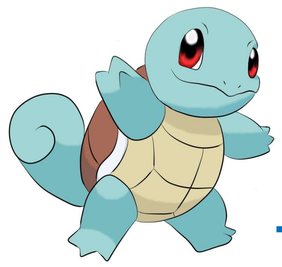
Steps:

1) Load ‘leaftraits.csv’ (if it is not already loaded)
2) Create dataframe where ‘NA’ observations in N2 fixing are removed

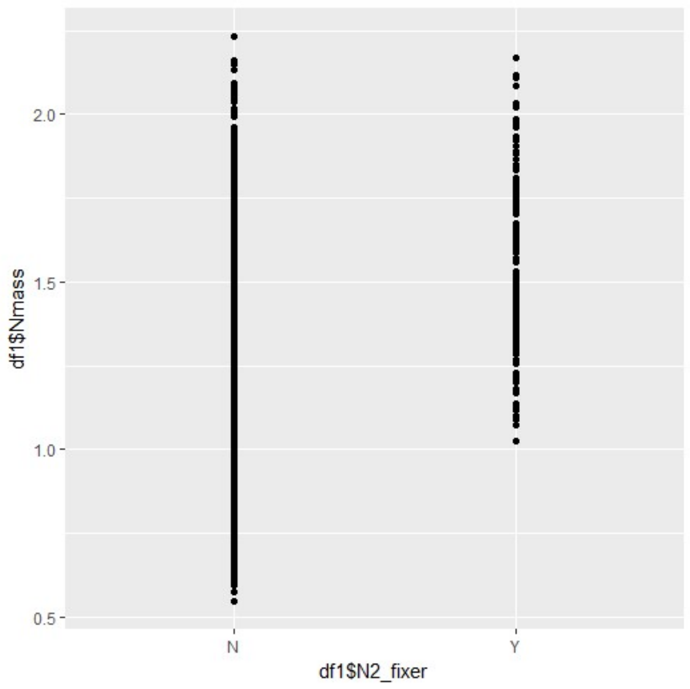
```
newdata=subset(leaftraits, N2_fixed!=NA)
```

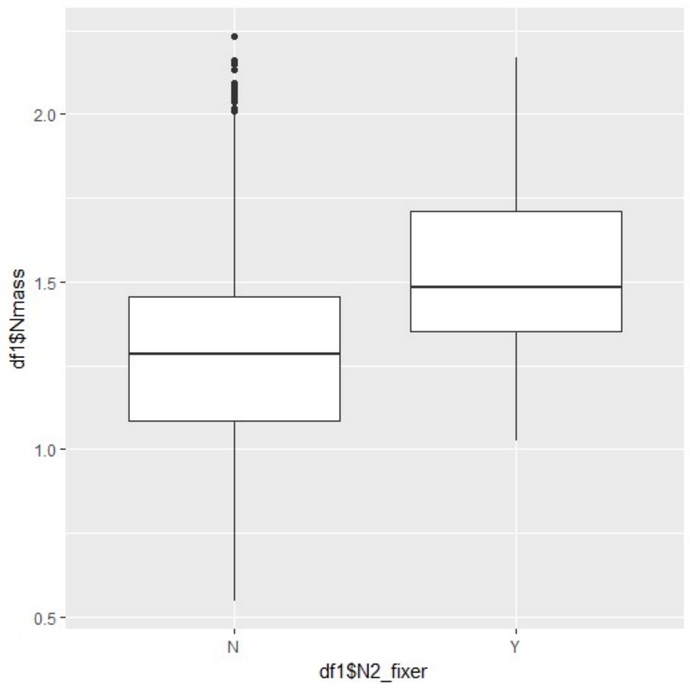
3) Load package ‘ggplot2’
4) Plot N-fixing ability by leaf N-content—Try first using ‘point’ geometry, then ‘boxplot’ geometry

Tip: What happens if you don’t exclude the NAs?

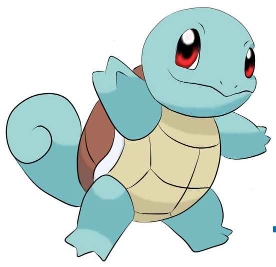


First figures





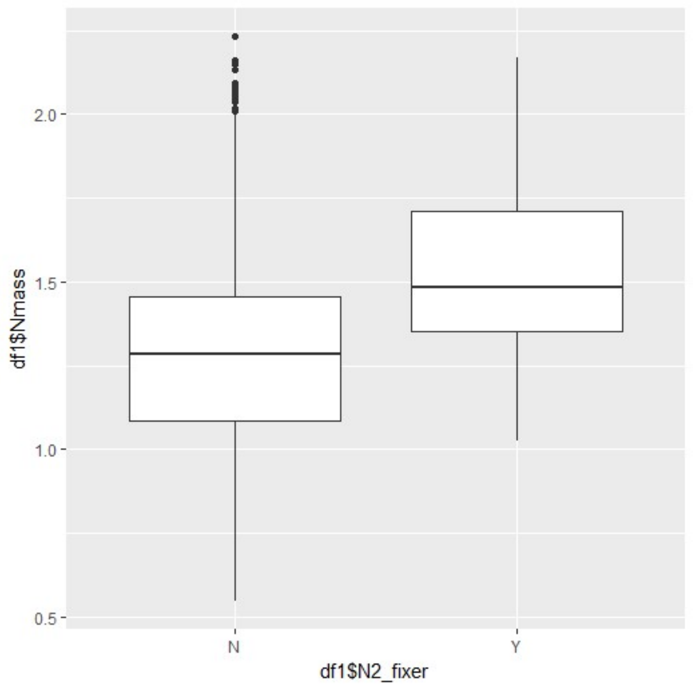
Problems with these graphs?

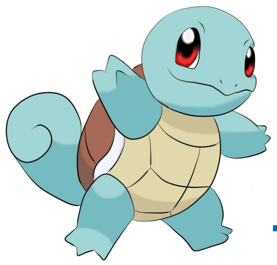


First figures

How can we improve the boxplot?

- Increase the font size
- Add axis lines
- Remove gridlines
- Remove gray background
- Change axis labels



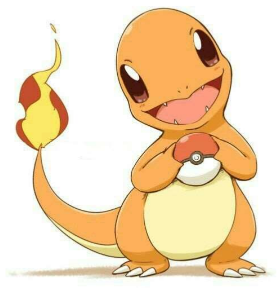


Basic figure modifications

Goal	ggplot syntax
Increase font size	theme(text = element_text(size=20))
Change font color	theme(text = element_text(color="black"))
Use a style with white background, axis lines, no gridlines	theme_classic()
Change X axis label	xlab("INSERTLABELHERE")
Change Y axis label	ylab("INSERTLABELHERE")
Change size of points, lines, boxplots, etc.—done within the 'geom_XXXX()' line	geom_XXXX(size=3)

When adding these lines to your script, don't forget to add a **+** to the line before it!

Tip: for the elements changed within 'theme()': you can add them all within the theme parentheses, separating them by a comma.



Time for an exercise

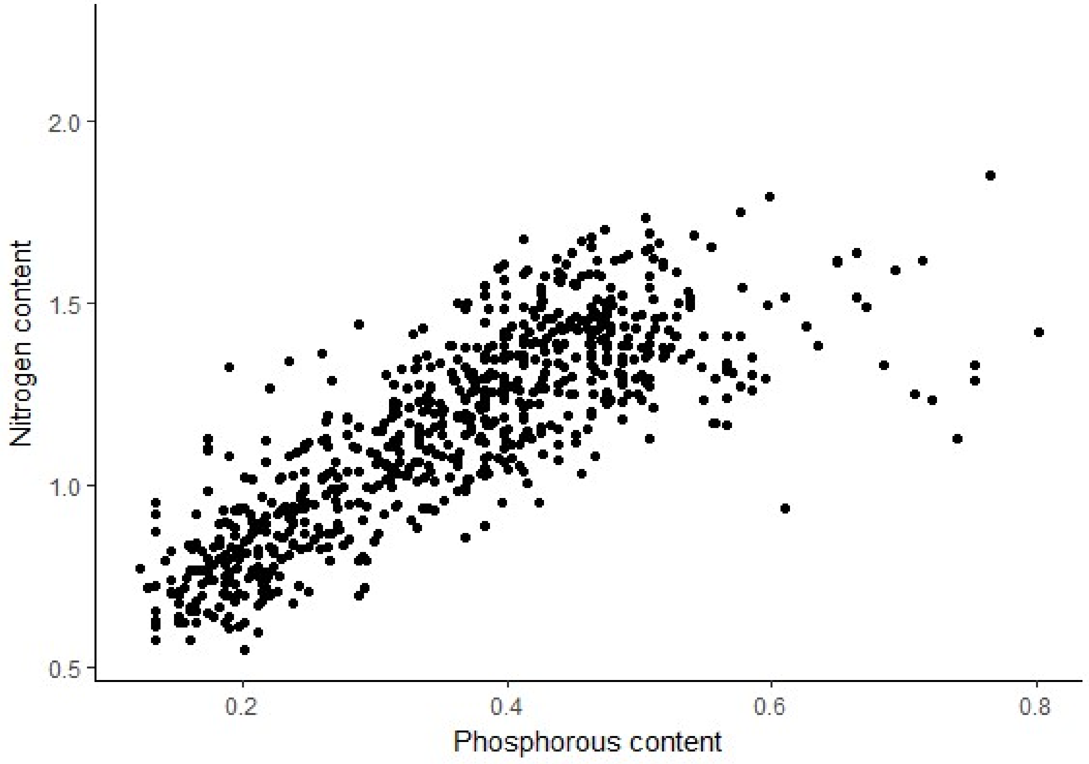
- With our leaf traits dataset, make a NICE plot showing:
  - The relationship between species' N-fixing ability (yes/no), and leaf N-content

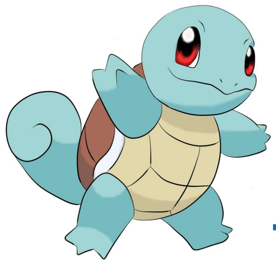
Steps:

Add some of the modifiers seen on the previous slide

What will the following code plot?

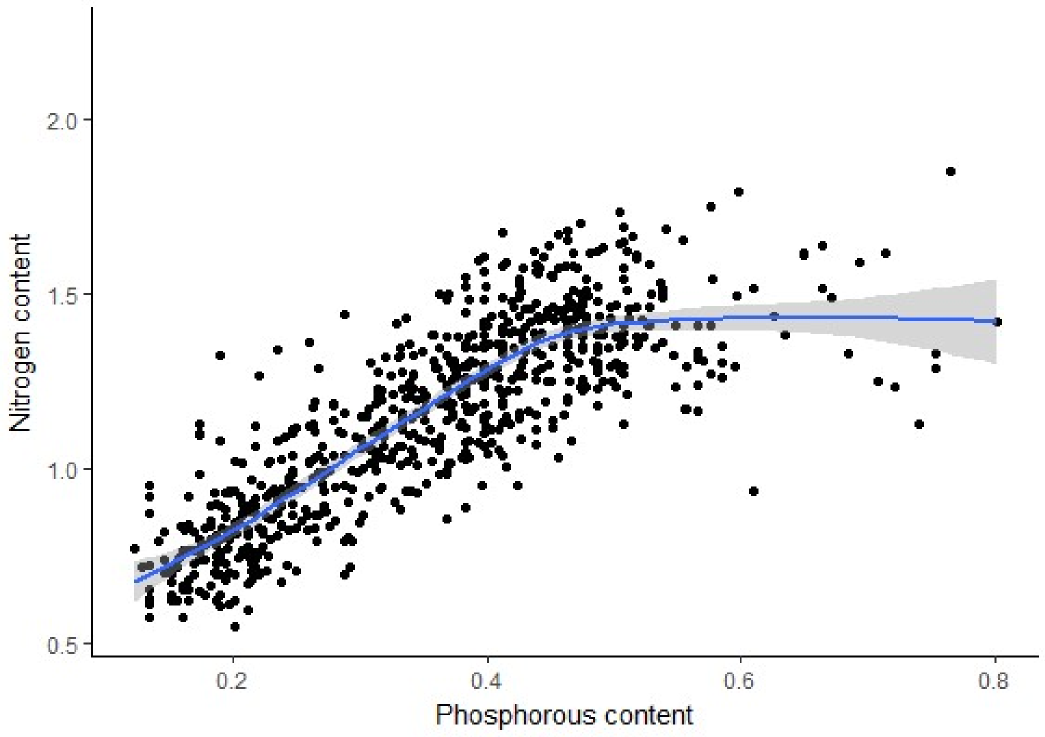
```
ggplot(data=leaftraits, aes(x=Pmass, y=Nmass)) +
        geom_point() +
        ylab("Nitrogen content") +
        xlab("Phosphorous content") +
        theme_classic()
```





Adding a smoothing line

- You can kind of see the trend in the points here, ggplot can also estimate one for you
- For this we can add the geometry call, **geom_smooth()**

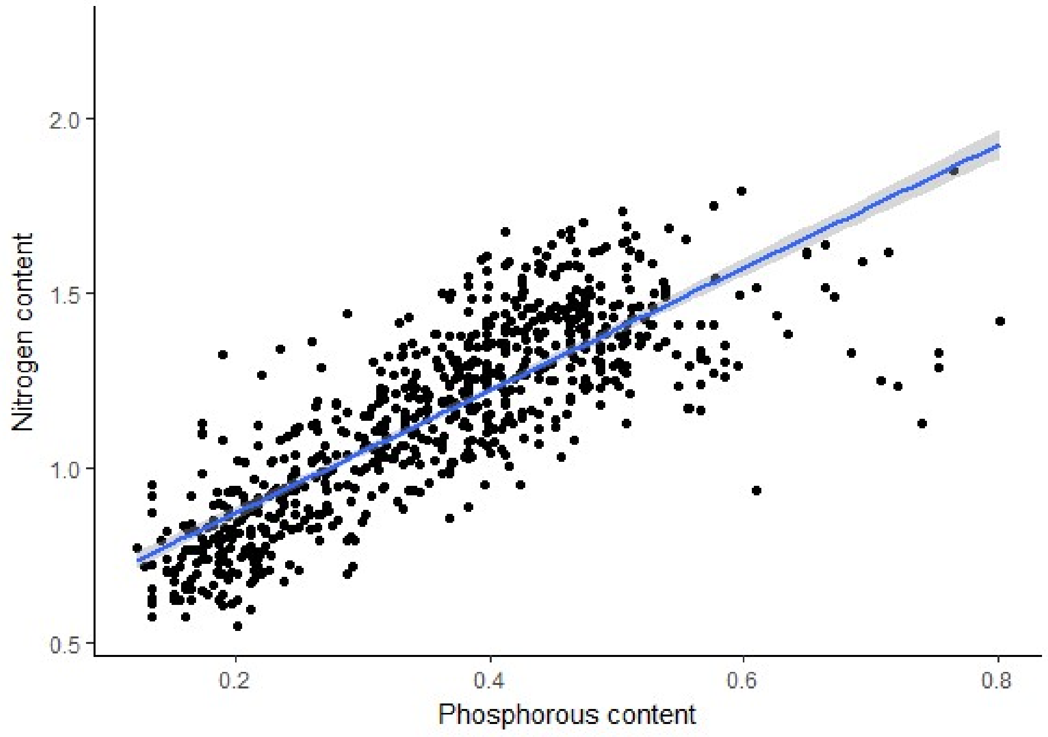


Try adding this to the code for the last plot!

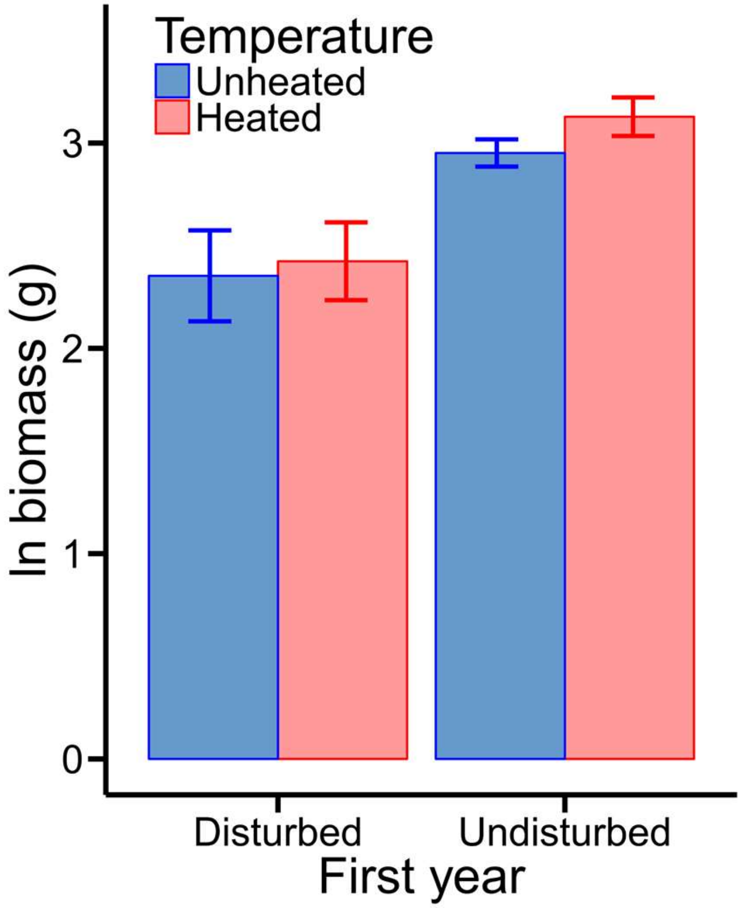
Adding a smoothing line

- By default, **geom_smooth()** fits a model using a non-linear “GAM” model method. If we want a linear fit, we can adjust it like so:

geom_smooth(method=“lm”)



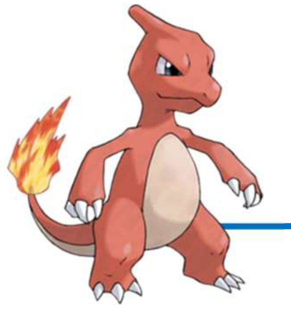
Plots with levels grouped by color



Plots with levels grouped by color

1. Identify your data
   a) Dataframe to work from
   b) X & Y variables
   c) **Grouping variables**

```
ggplot(DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE,
        color=GROUPINGVARIABLE)) +
geom_GRAPHICTYPE()
```



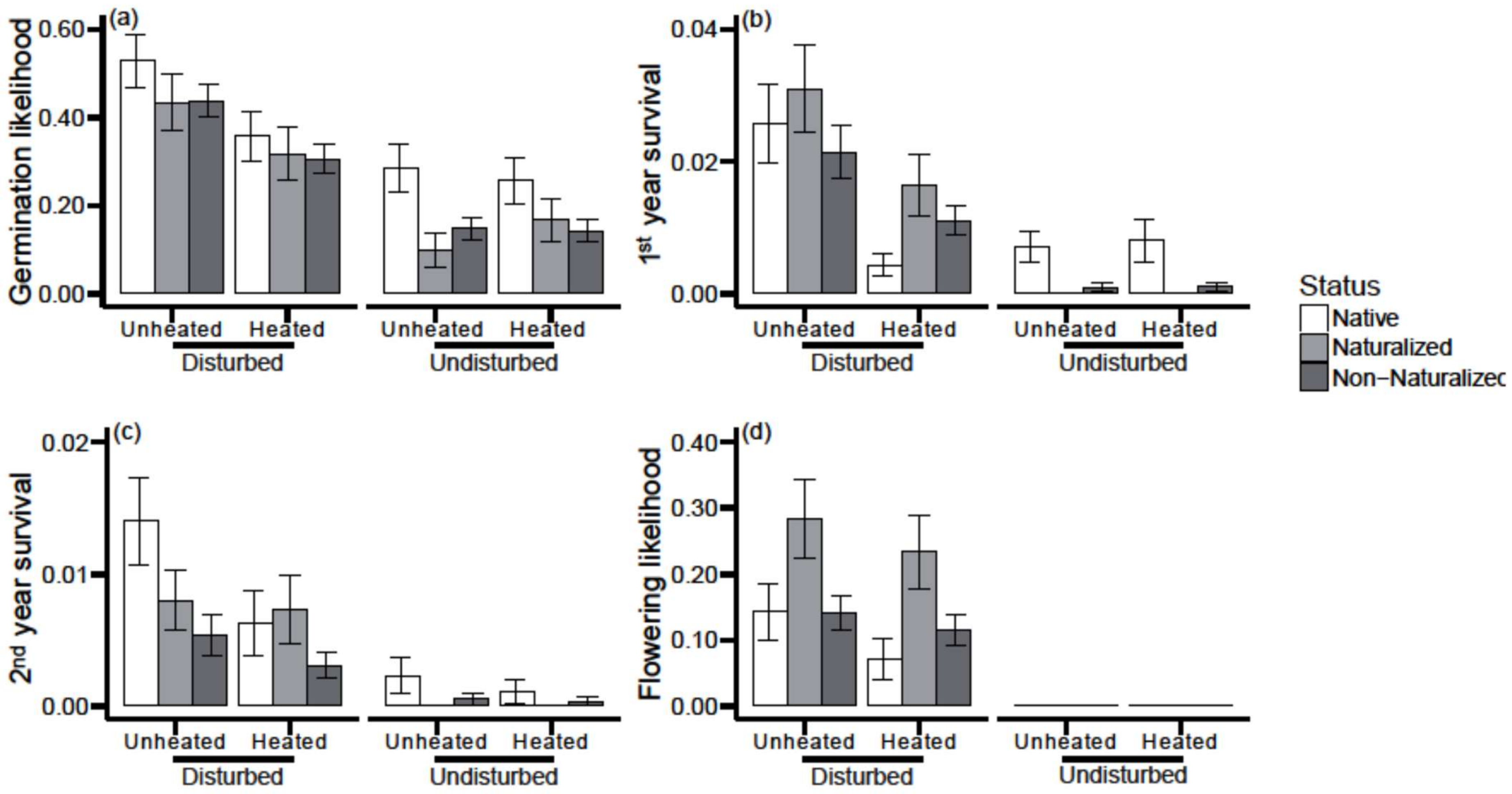
Time for an exercise

- With our leaf traits dataset, make a plot showing:
  - The difference between Nitrogen content levels for Nitrogren-fixing and non-Nitrogren-fixing species, and how that differs for Deciduous and Evergreen species

Tip: There are still NAs in the ‘Decid_Evergreen’ column; make a subset of your data that excludes them!

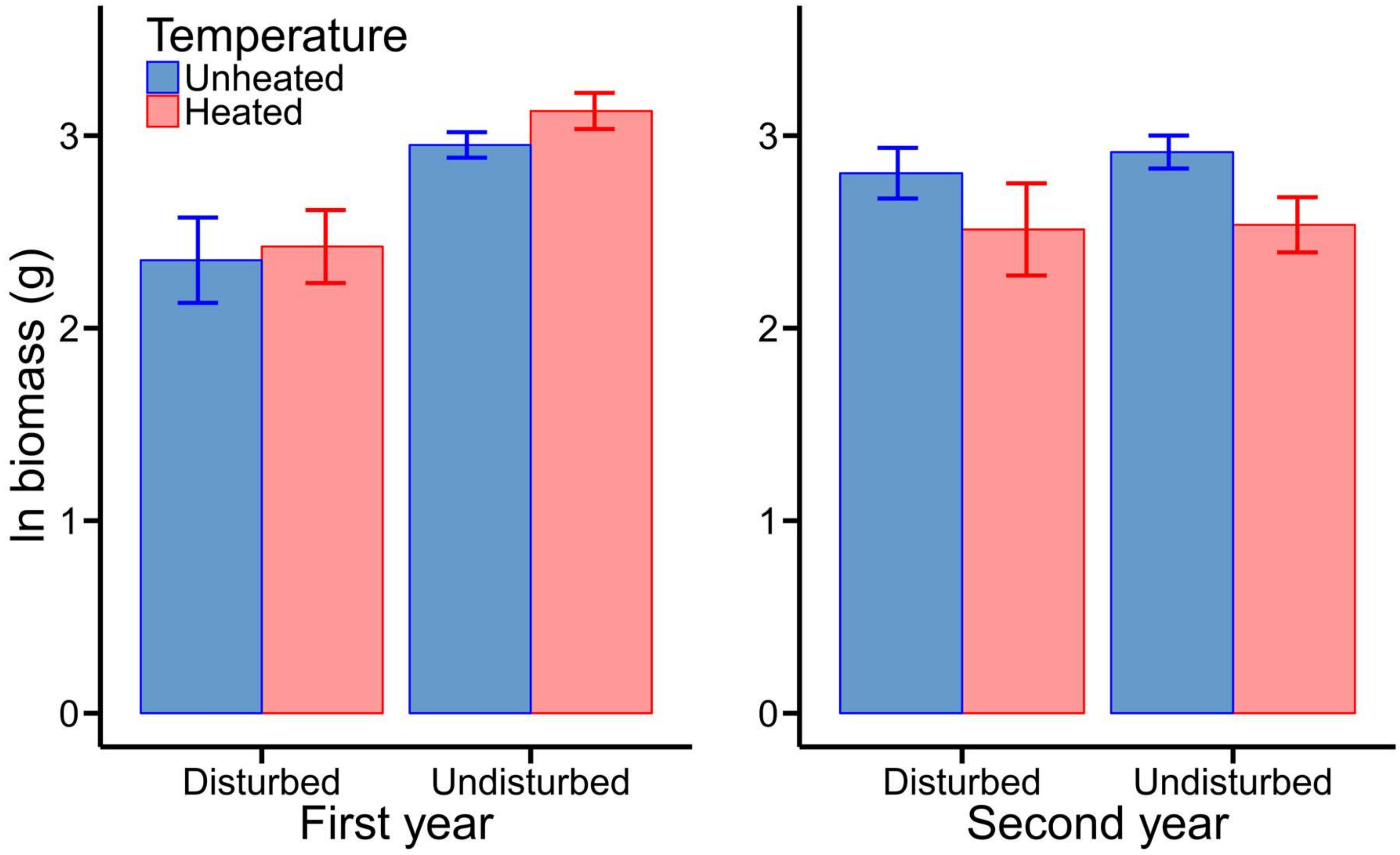
Bonus: what happens if you replace ‘color’ with ‘fill’?

Facets and multipanel graphs



Facets

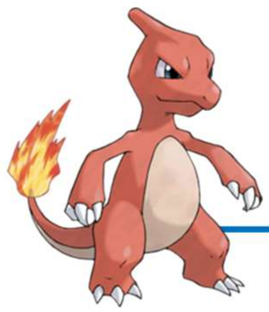
Useful when you have two interacting explanatory variables to show



Faceting code

1. Identify your data
   a) Dataframe to work from
   b) X & Y variables
   c) **Grouping variables—two or more**

```
ggplot(DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE,
        fill=GROUPINGVARIABLE)) +
geom_GRAPHICTYPE()+
facet_grid(.~FACETINGVARIABLE)
```



Time for an exercise

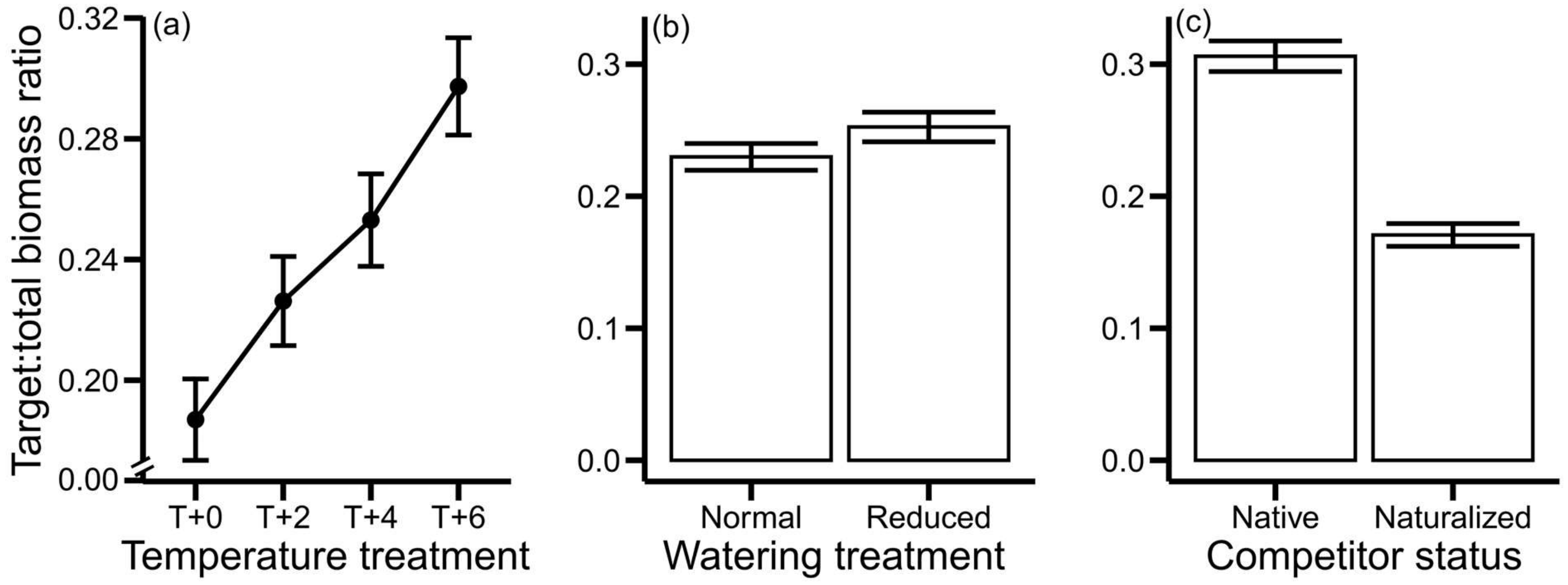
- With our leaf traits dataset, make a plot showing:
  - The difference between Nitrogen content levels for Nitrogren-fixing and non-Nitrogren-fixing species, and how that differs for Deciduous and Evergreen species, AND broadleaf vs needle species

Tip: There are still NAs in the ‘Needle_broadlf’ column!

Bonus: what happens if you switch the ‘.’ with the faceting variable?

Multipanel plots

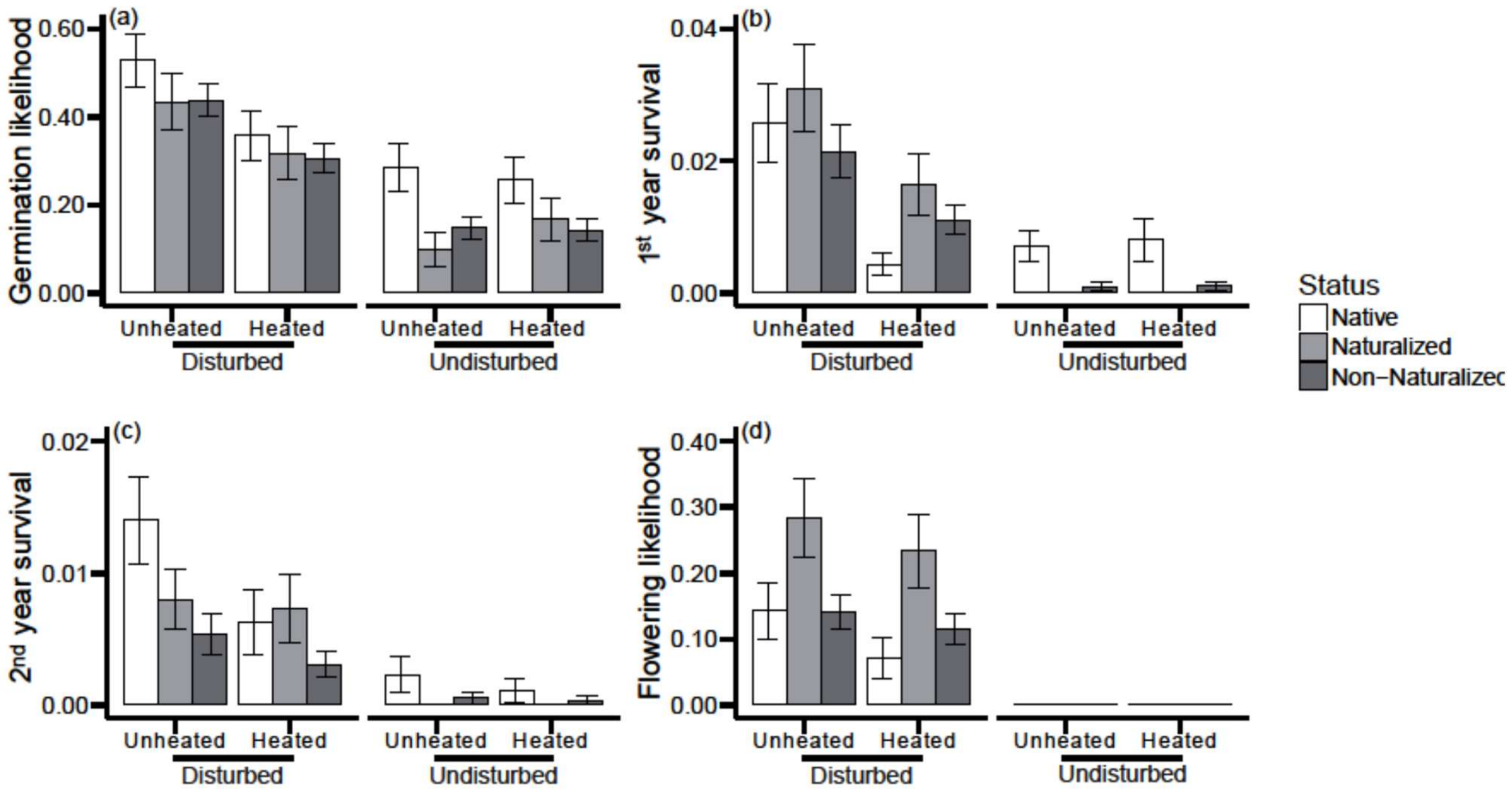
Useful for displaying non-interacting explanatory variables



How does this differ from faceting?

Multipanel graphs

Useful for displaying same explanatory effects on different response variables



Multipanel plots

Step 1: Save the plots you want to put together

```
plot1<-ggplot(………
plot2<-ggplot(……..
```

Step 2: Load the package ‘gridExtra’ (which you have hopefully installed!)

Step 3: Use the function ‘grid.arrange’

```
grid.arrange(plot1, plot2)
```

Try this with some of the plots from earlier

Bonus: what happens if you add the argument ‘ncol=2’?)

More *ggplot2* resources

Cheat sheet 1

Cheat sheet 2

Picking colors

Acknowledgements

People:

Noelie Maurel
Wayne Dawson
Fränzi Körner

International Max Planck Research School for Organismal Biology

Ecology Konstanz

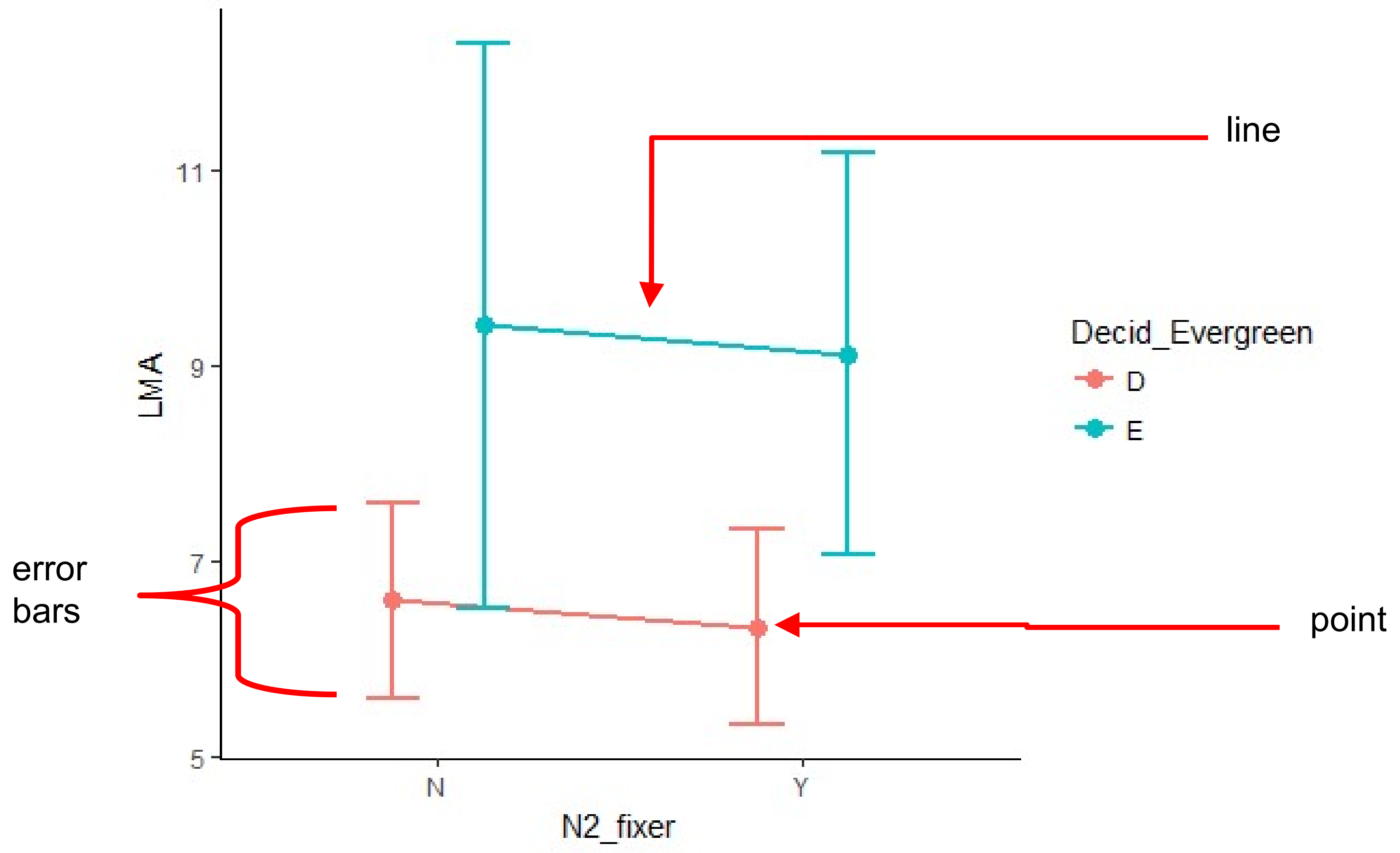
Institute for Health Metrics and Evaluation

TReND

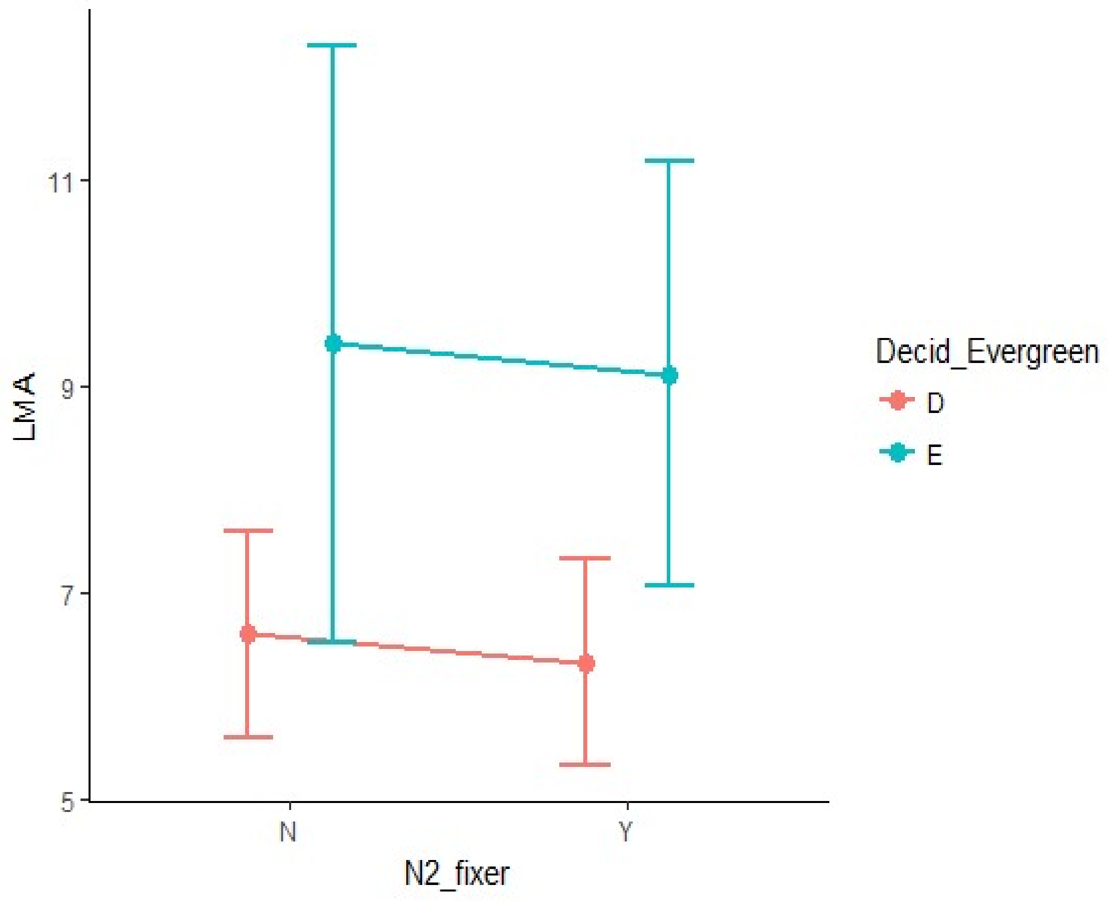
The Company of Biologists

Bonus

Geometric types: you can include more than one!



Extra:Sometimes you will have to **aggregate** your data to get what you want in your plot…



This plot shows the mean and
standard deviation of LMA for Nfixing and non-N-fixing species,
separated out by deciduous or evergreen species.

We plot standard deviation using the
geom_errorbar() geometry.

But, we don't have the means or standard deviations calculated.

How can we make a dataset with this info?

Aggregate! (Twice!)

```
newdata4<-aggregate(LMA~N2_fixer+Decid_Evergreen, data=newdata3, mean)
#^^Makes a simplified dataset showing means for LMA according to grouping variables

newdata5<-aggregate(LMA~N2_fixer+Decid_Evergreen, data=newdata3, sd)
 #Make a simplified dataset for standard deviations for LMA according to grouping variables

#We need to add the standard deviation of LMA as a column to df4
newdata4$LMAsd<-newdata5$LMA #Rename LMA so you don't have two columns named LMA.


#Now we use the dataframe 'newdata4' to make our ggplot.
```