Lab 11b: Regression data exploration - scatterplots

Nathan Brouwer brouwern@gmail.com @lobrowR

2017-11-28

Lab 11 Part I continued: Data exploration with scatterplots

References

Meredith et al 1991 Repeated measures experiments in forestry: focus on analysis of response curves. Can. J. For. Res.

Load data

If you haven't already, load data_long.csv and turn conc.AL and week to factor variables. This won't be necessary if you are continuing directly from the previous handout.

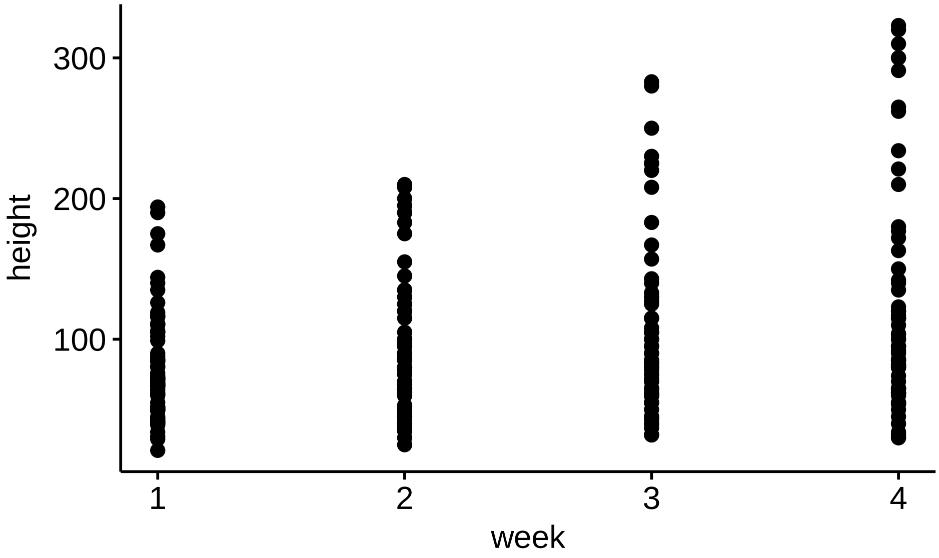
```
data.long <- read.csv(file = "data_long.csv")
data.long$conc.AL.FAC <- factor(data.long$conc.AL)
data.long$week.FAC <- factor(data.long$week)
```

Plotting regression data: scatterplots

Basic scatter plot for regression

Plot regression-style data with the ggpubr function ggscatter()

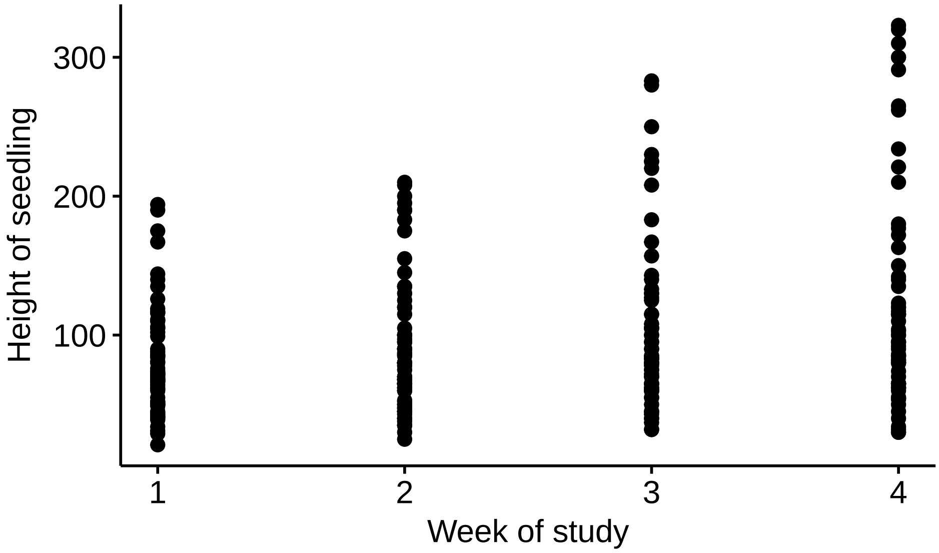
```
library(ggpubr)

ggscatter(data = data.long,
          y = "height",
          x = "week")
```



Change x and y labels

Plot regression-style data with the ggpubr function ggscatter()

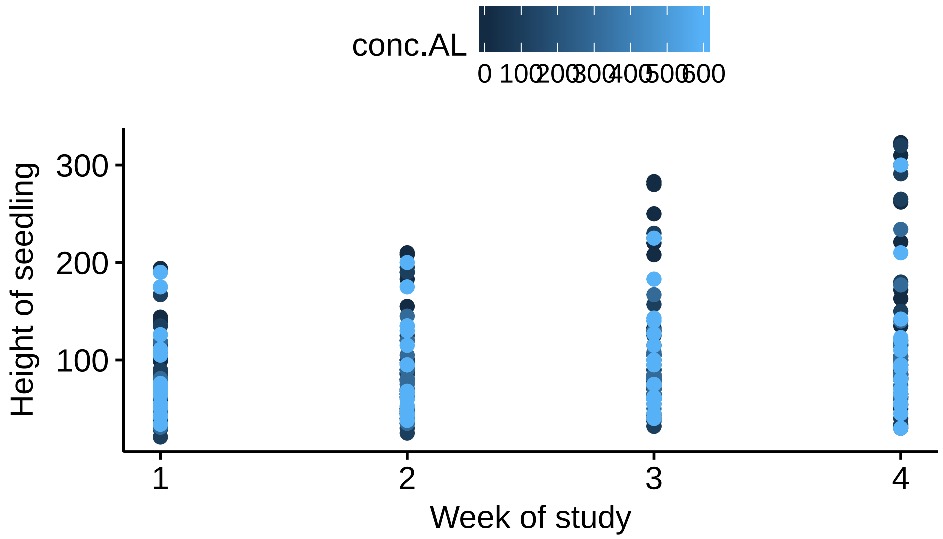
```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling")
```



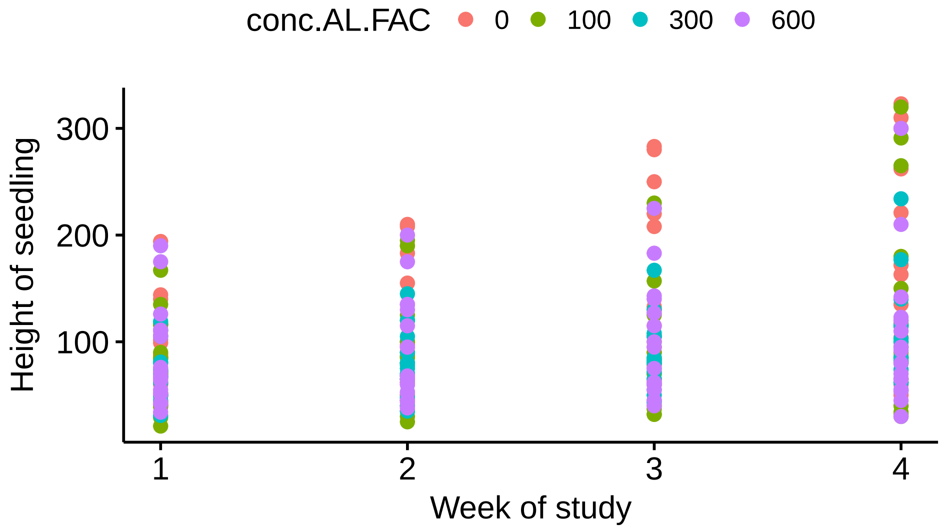
Change color based on treatment

We can use color = "conc.AL" to color code the data points

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          color = "conc.AL")
```

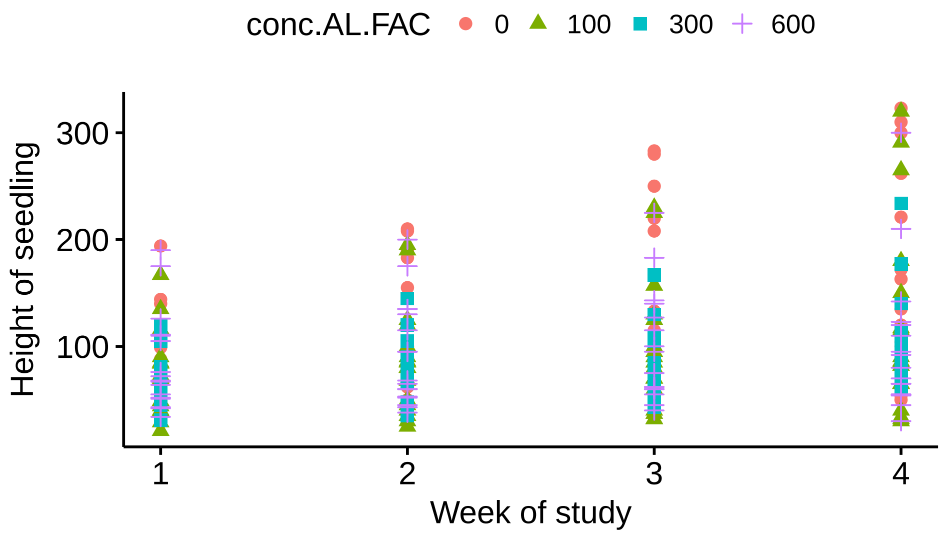


"conc.AL" is numeric data and when ggplot and/or ggpubr use a numeric variable to set colors they change the shade gradually from dark to light blue. Since we have only a 4 different amounts of AL used in the study it makes more sense to use AL as a factor. Change color = "conc.AL" to color = "conc.AL.FAC" to produce this graph



Change shape based on treatment

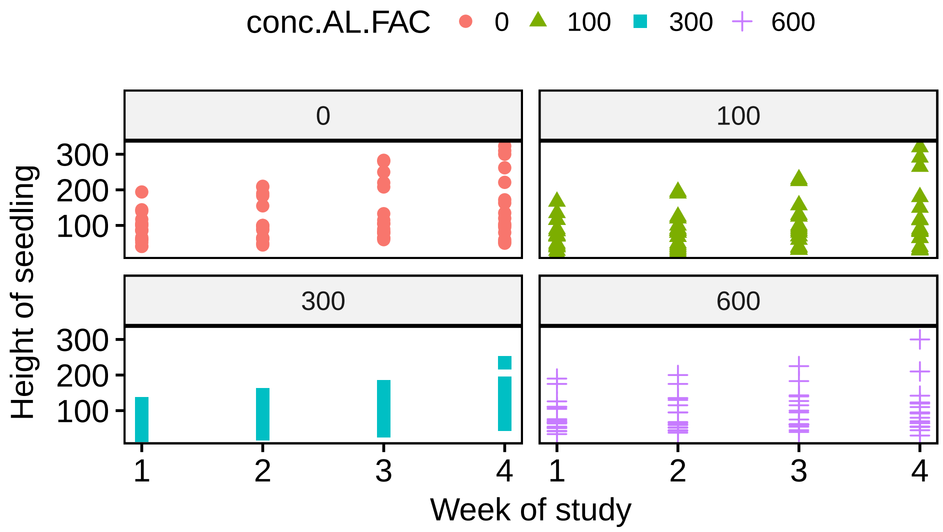
We can chane the shape also to make to even clearr what the treatmetns are. Setting shape = "conc.AL.FAC" will produce the following graph



Facet data

There is a lot of overlap in the points. We can clean things up by faceting the data so that the treatments are in different panels.

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          color = "conc.AL.FAC",
          shape = "conc.AL.FAC",
          facet.by = "conc.AL.FAC")
```

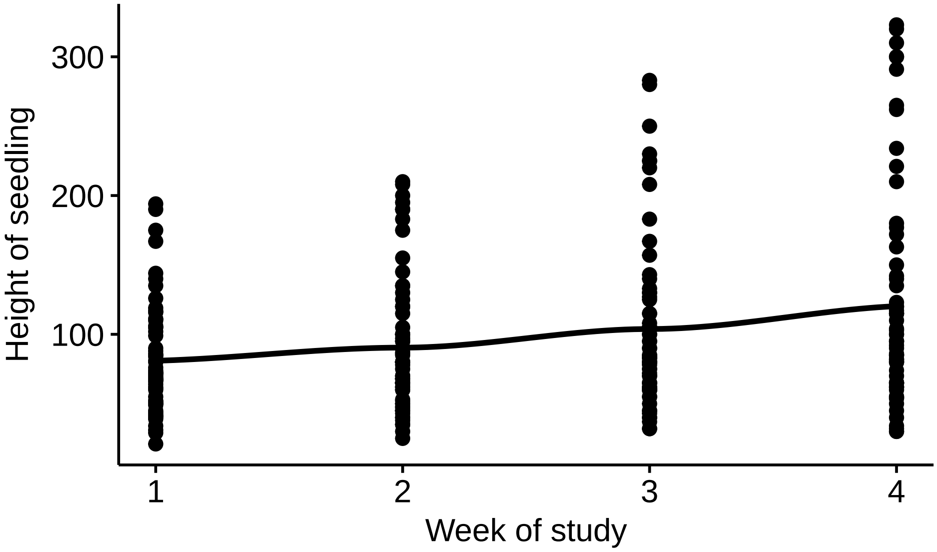


Add smoothers & regression lines

Add smoothers lines

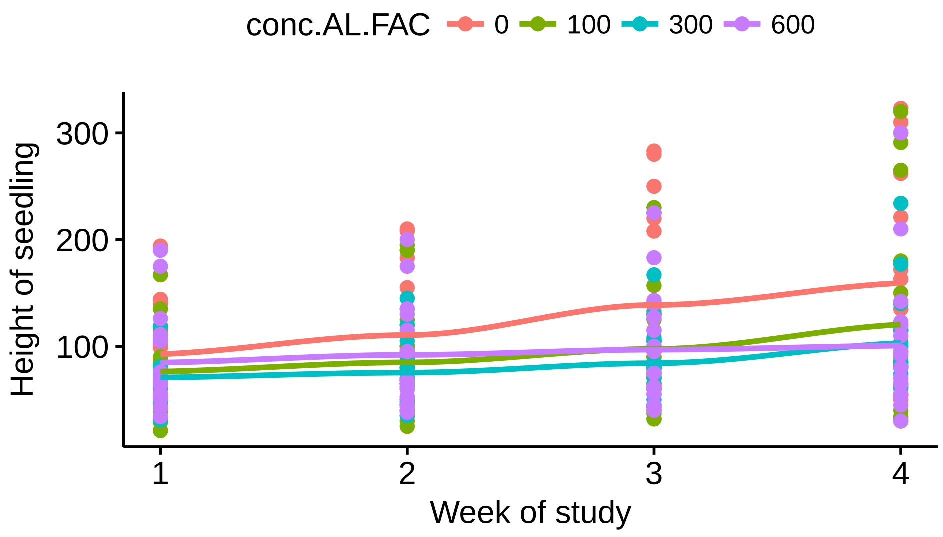
Smoothers show you the general shape of the data. A common type of smoother is a "loess smoother." We can add one by adding add = "loess"

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "loess")
```



What happens when we add color?

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "loess",
          color = "conc.AL.FAC")
```

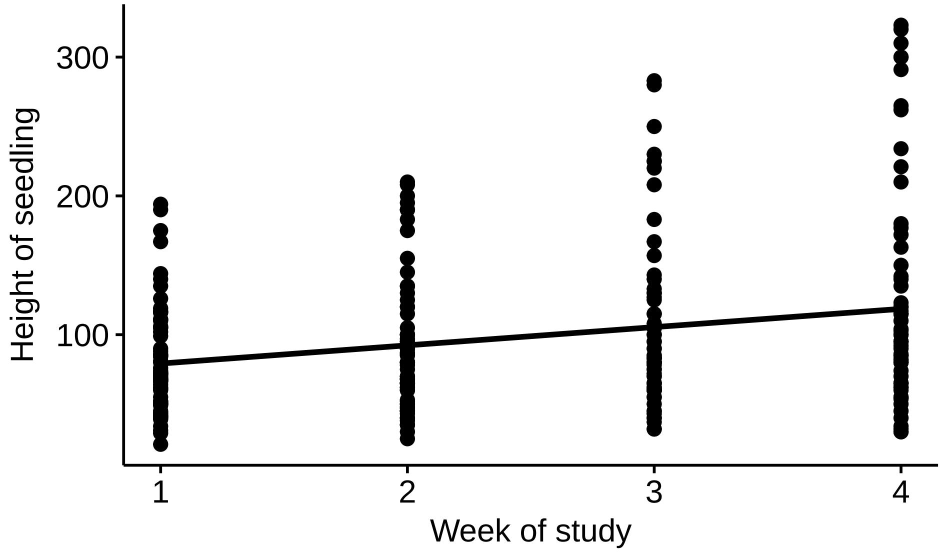


Most of these lines are fairly straight, meaning that standard regression will probably work pretty well. When smoother lines curve more complicate models might be appropriate.

Add regression lines

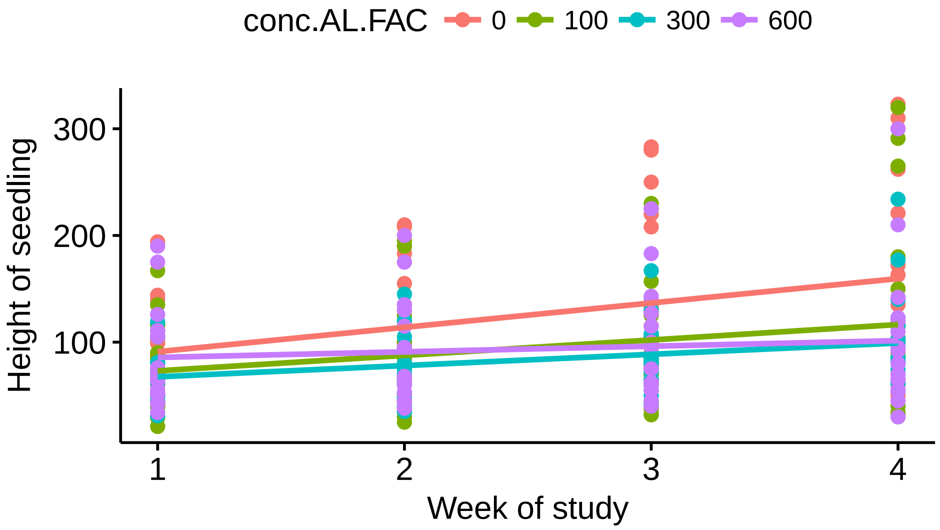
A least square regression line can be added to the plot using add = "reg.line" for "Add regression line." This line is fit behind the scenes using the lm() function.

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "reg.line")
```



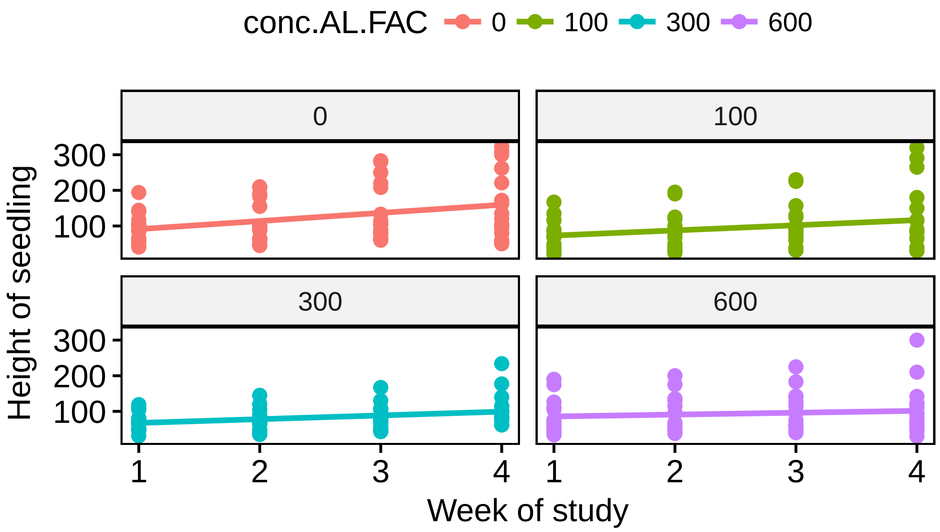
What happens when we add color using color = "conc.AL.FAC"?

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "reg.line",
          color = "conc.AL.FAC")
```



And then we facet?

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "reg.line",
          color = "conc.AL.FAC",
          facet.by = "conc.AL.FAC")
```

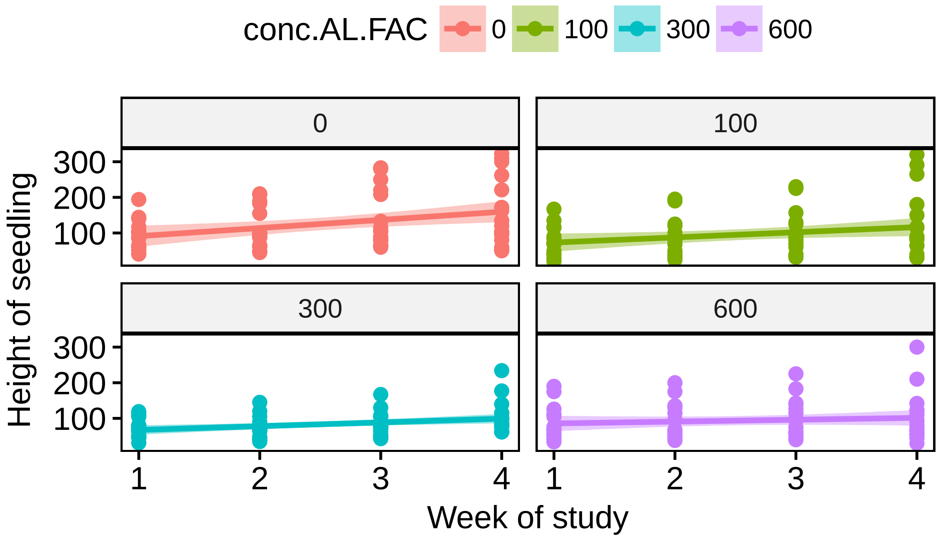


Add confidence intervals

Data in a scatterplot are used to estimate an intercept and a slope for the best fit line running through the scatterplot. The slope and intercept are estimated with uncertainty; therefore the slope and intercept both have standard errors. This uncertainty can be transformed into an error band or 95% confidence band that represents uncertainty in the true location of the line.

We can easily add this "conf.int = TRUE". Each line has its own 95% confidence interval.

```
ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "reg.line",
          conf.int = TRUE,
          color = "conc.AL.FAC",
          facet.by = "conc.AL.FAC")
```



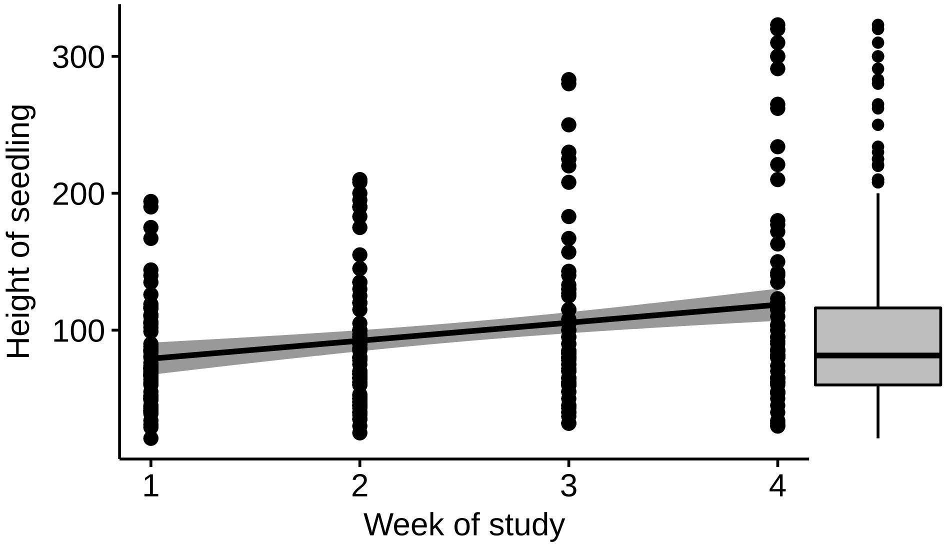
Advanced / Optional: Combine scatter plots and histograms

If we download the ggExtra package we can combine our scatterplot with our previous exploratory boxplots. Note that the syntax here is a big different. We have to save the plot to an object (my.plot) and then use a function on that object: ggMarginal(my.plot).

Note: If this doesn't work, don't worry.

NOte: Don't ask for help getting this to work - its totally cherry on the top plotting.

```
#load ggExtra
library(ggExtra)

#save the scatterplot
my.plot <- ggscatter(data = data.long,
          y = "height",
          x = "week",
          xlab = "Week of study",
          ylab = "Height of seedling",
          add = "reg.line",
          conf.int = TRUE)

#make boxplot
ggMarginal(my.plot, type = "boxplot",margins = "y")
```


300
200
100
Height of seedling
1
2
3
4
Week of study