Lab 11a: Regression data exploration - boxplots & histograms

Nathan Brouwer brouwern@gmail.com @lobrowR

2017-11-28

Introduction

In this lab exercise will explore regression modeling. We will use a dataset of tree seedlings treated with different concentrations of nutrients to see how it impacts growth over time.

Outline of Lab

There are numerous steps to this lab, each one divided into seperate handouts.. The key parts are

- Part I: Data exploration
- Part II: Basic regression modeling
- Part II: Anova Analysis
- Part IV: Diagnostics & Transformatiosn
- Part V: Multiple regression
- Part VI: Repeated measures analysis

Part I: Data exploration

For data exploration we'll use the whole dataset using all weeks of data.

- Data format: wide vs. long
- Data exploration with boxplots, histograms, and density plots
- Data exploratin with scatterplots & smoothers

Part II: Basic regression modeling

For regression modeling we'll consider just the last week of day (week 4).

- Regression model building for week 4
- Regression model testing
- Signficance Testing with hypothesis tests and p-values
- Assesing model fith with R^2
- Model comparison with AIC
- Reporting results of regression

Part III: 1-way ANOVA Analysis

For ANOVA analysis we'll also consider just last week of day (week 4).

- 1-way ANOVA with lm()
- Refitting models with aov()
- Multiple comparisons with TukeysHSD()
- Plotting effect sizes from TukeysHSD()

Part IV: Diagnostics & Transformatiosn

- Model diagnostics
- Transformation of variables

Part V: Multiple regression

- Multiple regression: regression with multiple predictory (x) variables

Part VI: Repeated measures analysis

- Repeated measures data: multiple measurements on the same individual

References

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

Data formatting

- Wide format
- Long form

Wide data format

- The 1st column is the concentration of aluminum (AL) that sugar maple seeds were treated with.

- Each ROW is a different tree
- Height growth was then measured for 4 weeks
- Each height was recorded in a seperate column for each week
- This is called "wide" format b/c data for an individual thing being studied is read left to right
- This is a common way to collect data and present it in a table and is easy to read
- These data are considered to be "repeated measures" data because each study object (tree seed) has had multiple measurements taken on it over time.
- Repeated measures data is often called longitudinal data.
- When repeated measures data are in long format its usuallty easy to identify that the same thing is being measured repeated

Load the data from a .csv file using read.csv()

```
dat.orig <- read.csv(file = "data_orig.csv")
```

The size of the dataframe

```
dim(dat.orig)
```

```
## [1] 67  6
```

Look at the data in wide format

```
head(dat.orig)
```

```
##   X conc.AL ht.wk.1 ht.wk.2 ht.wk.3 ht.wk.4
## 1 1       0      60      62      78     104
## 2 2       0      41      50      60      60
## 3 3       0      85      97     115     120
## 4 4       0      88      87      90      80
## 5 5       0      66      65      80      95
## 6 6       0     106     100     133     172
```

Experimental design

Seedlings were treated with different concentrations of Aluminum (AL). We can see some info about this using the summary() command.

```
summary(dat.orig)
```

```
##        X           conc.AL         ht.wk.1          ht.wk.2
##  Min.   : 1.0   Min.   :  0.0   Min.   : 21.00   Min.   : 25.00
##  1st Qu.:17.5   1st Qu.:100.0   1st Qu.: 51.50   1st Qu.: 56.50
##  Median :34.0   Median :300.0   Median : 72.00   Median : 80.00
##  Mean   :34.0   Mean   :253.7   Mean   : 80.97   Mean   : 90.42
##  3rd Qu.:50.5   3rd Qu.:450.0   3rd Qu.:105.00   3rd Qu.:110.00
##  Max.   :67.0   Max.   :600.0   Max.   :194.00   Max.   :210.00
##     ht.wk.3         ht.wk.4
##  Min.   : 32.0   Min.   : 30.0
##  1st Qu.: 62.0   1st Qu.: 65.0
##  Median : 83.0   Median : 95.0
##  Mean   :103.8   Mean   :120.3
##  3rd Qu.:126.0   3rd Qu.:141.0
##  Max.   :283.0   Max.   :323.0
```

Long format data

- This is how R needs data to be formatted for regression and ANOVA (and pretty much everything else)
- t.test also uses data in this format
- Note that there are MANY rows of data
- Data in this format is not very easy to read by eye
- This format matches how the math gets done by the computer
- ALL respone data (y variables, “height”) are in a SINGLE column
- ALL predictor data (x variable, week) are in single columns
- When repeated measures data are in wide format its usuallty hard to identify that the same thing is being measured repeated - this is why a data dictionary can be key!

Read the data in and determine its size

```
data.long <- read.csv(file = "data_long.csv")

dim(data.long)
```

```
## [1] 268   4
```

Look at long-format data

```
head(data.long)
```

```
##   X height week conc.AL
## 1 1     60    1       0
## 2 2     41    1        0
## 3 3     85    1        0
## 4 4     88    1        0
## 5 5     66    1        0
## 6 6    106    1        0
```

Change numeric data to a factor

For this analysis we'll be focusing on regression analysis and so will be treating the variables height, conc.AL and week, at least intially, as continuous numeric variables. However, when a numeric variable (which in principal can take on any value) only takes on certain values it can be useful to treat it, at least during data exploration, as a categorical variable (aka "factor"). IN this study the concentration of AL was set experimetnally by the researchers. AL can in principla take on any value but only four valeus were used. Similarly, the study occurred over time and data was colelcted weekly. However, data could have been collected at different intervals (eg every 5 days instead of every 7), or at un-even intervals (eg at 7, 10, 12 and 25 days instead of at 7, 14, 21 , and 28)

First, We'll make a new columns called "conc.AL.FAC" where the "FAC" designates that its a factor

```
data.long$conc.AL.FAC <- factor(data.long$conc.AL)
```

We can then use summary() to see how many measurements there are per level of AL

```
summary(data.long$conc.AL.FAC)
```

```
##   0 100 300 600
##  64  68  68  68
```

Note that if we didn't want to make new column we could get this info in a single step like this

```
summary(factor(data.long$conc.AL))
```

```
##   0 100 300 600
##  64  68  68  68
```

We can similarly change the time variable "week" into a new column "week.FAC"

```
data.long$week.FAC <- factor(data.long$week)
```

We can see how it looks as a factor vs a numeric variable

```
summary(data.long$week)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    1.75    2.50    2.50    3.25    4.00
```

```
summary(data.long$week.FAC)
```

```
##  1  2  3  4
## 67 67 67 67
```

Optional / Advanced: the table() command

A great way to see how data fall out into muliple factors is to use the table() command.

```
table(data.long$week, data.long$conc.AL)
```

```
##
##      0 100 300 600
##    1 16  17  17  17
##    2 16  17  17  17
##    3 16  17  17  17
##    4 16  17  17  17
```

Note that table() can work with the original numeric data and doesn't needs the .FAC versions of the coulmn. However, it you give table a numeric variable that takes on lots of values, you will get a mess.

Optional / Advanced pro-tip: the with() command

In the code above I had to type "data.long.$. . . " twice. With the handy with() command and only need it once. Note that the table() function

```
with(data.long, table(week, conc.AL))
```

```
##     conc.AL
## week   0 100 300 600
##    1  16  17  17  17
##    2  16  17  17  17
##    3  16  17  17  17
##    4  16  17  17  17
```

General graphical data exploration

Its always good to thoroughly explore your data before analysis so that you can better understand its structure and also spot potential outliers or data entry errors. Boxplots and histograms are the best ways to do this, even if another type of plotting it more appropriate for presenting the analysis (eg a scatter plot for regression analyses)

Its good to make seperate boxplots and histograms of each continous numeric variable. Boxplots and histograms convey similar information, but I find thats it helpful to look at both.

In this study, there are three numeric variables, "height" , "conc.AL", and "week". However, conc.AL only takes on 4 different values so plotting it isn't that informative. We'll therefore just look at height. Instead, we'll use conc.AL as a factor and plot height at each level of AL.

Load plotting functions

```
library(ggplot2)
library(ggpubr)
```

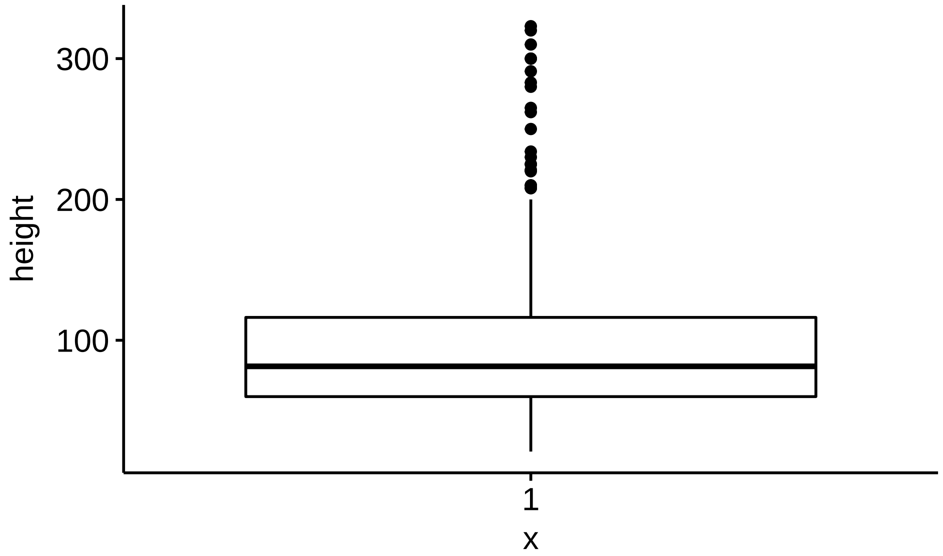
Boxplots

Boxplots in ggpubr are made with ggboxplot().

Boxplot of all of the data

We can look at all of the height data combined. Note that we only need to give it "y =" because we only have a single

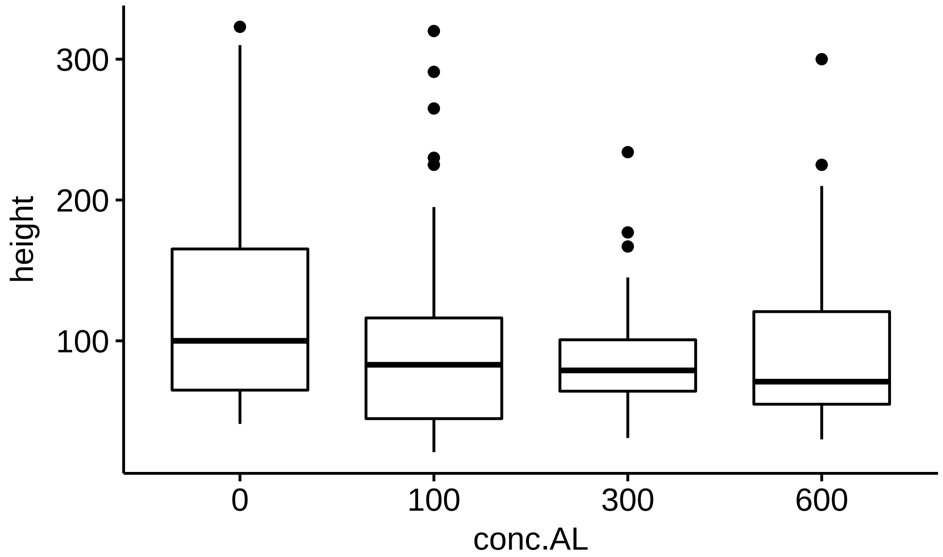
```
ggboxplot(data = data.long,
          y = "height")
```



Boxplots by experimental treatment

Its very informative to split the data up by experimental treatment. ggboxplot figures ot that conc.AL can work as a factor and plots the sperate groups. conc.AL.FAC would give the same results.

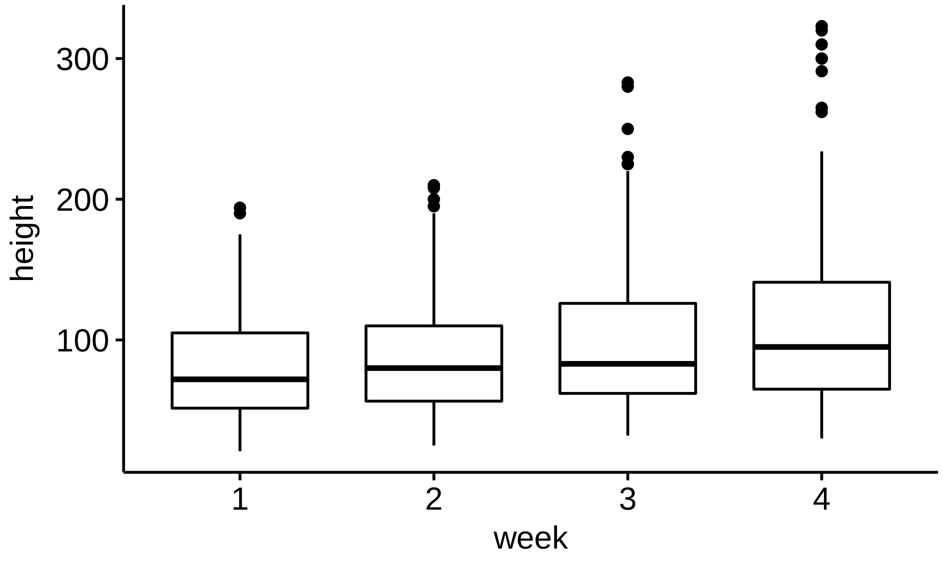
```
ggboxplot(data = data.long,
          y = "height",
          x = "conc.AL")
```



Boxplots by time

We can do the same thing using the time variable “week”.

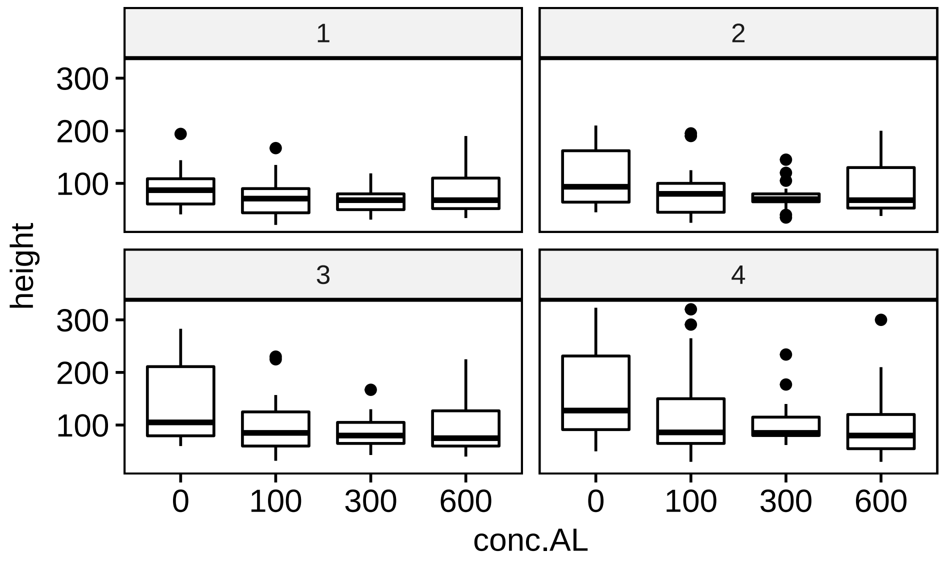
```
ggboxplot(data = data.long,
          y = "height",
          x = "week")
```



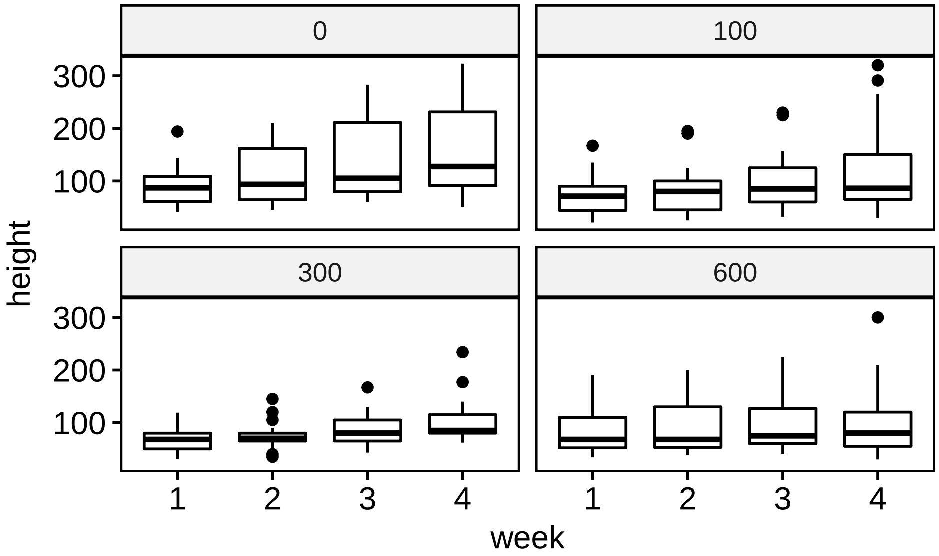
Facetting boxplots: plotting by week & time

We can do the same thing using the time variable “week”. Note that the x-axis is the different AL concentrations and the four panels are numbeed 1 through 4 for the four weeks.

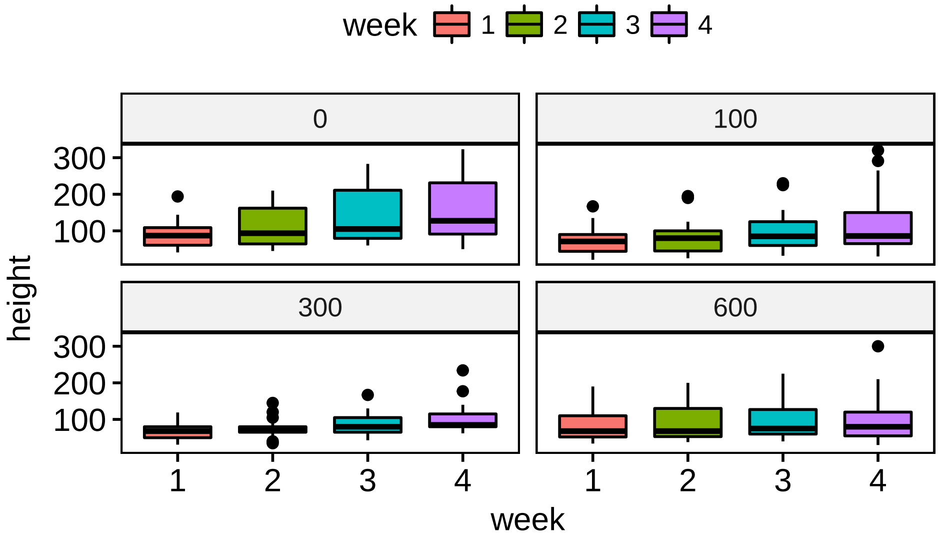
```r
ggboxplot(data = data.long,
          y = "height",
          x = "conc.AL",
          facet.by = "week")
```



We can produce this plot with week on the x-axis and the panels being different concentrations by swapping conc.AL and week.



Its doesn't add too much info but we can set the colors to be different using fill = "week"

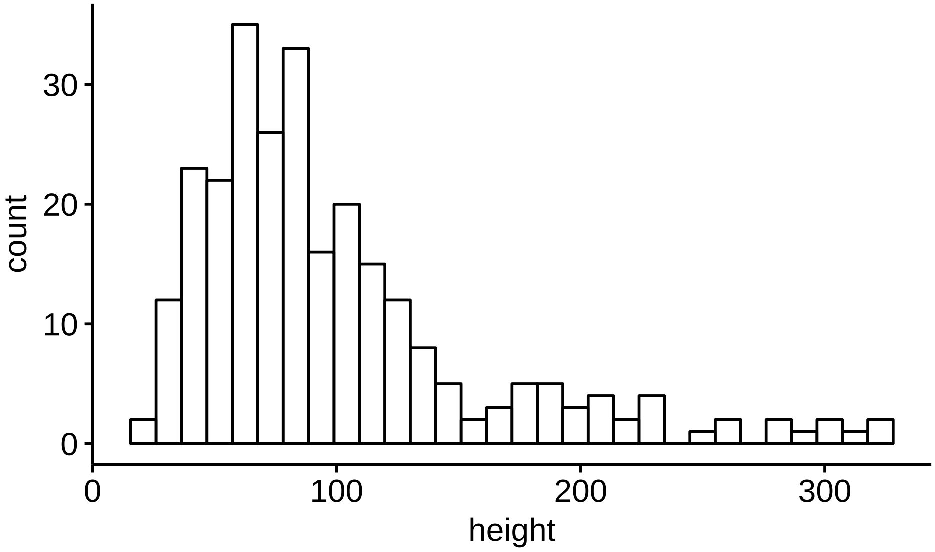


Histograms

Histogram of all of the data. Note that there is no "y = .."" for gghistogram, only "x = . . . "

```
gghistogram(data = data.long,
            x = "height")
```

```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```



Faceting histograms

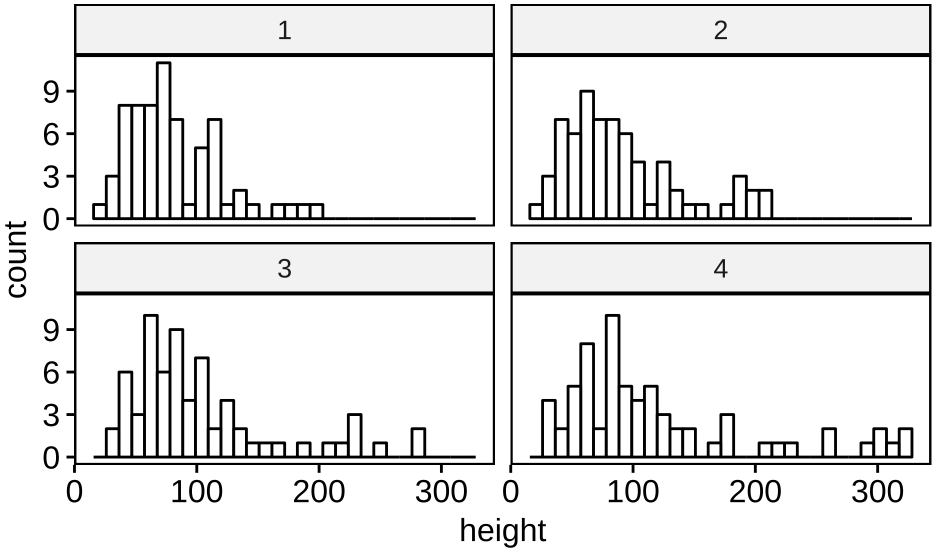
To look at different groupings of data using histograms you have to use faceting

Facetting by 1 variable

We can fact by week like this

```r
gghistogram(data = data.long,
            x = "height",
            facet.by = "week")
```

```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```



You can swap week with conc.AL

```
gghistogram(data = data.long,
            x = "height",
            facet.by = "conc.AL")
```

Color coding makes it look nicer, but at this stage doesn't add much

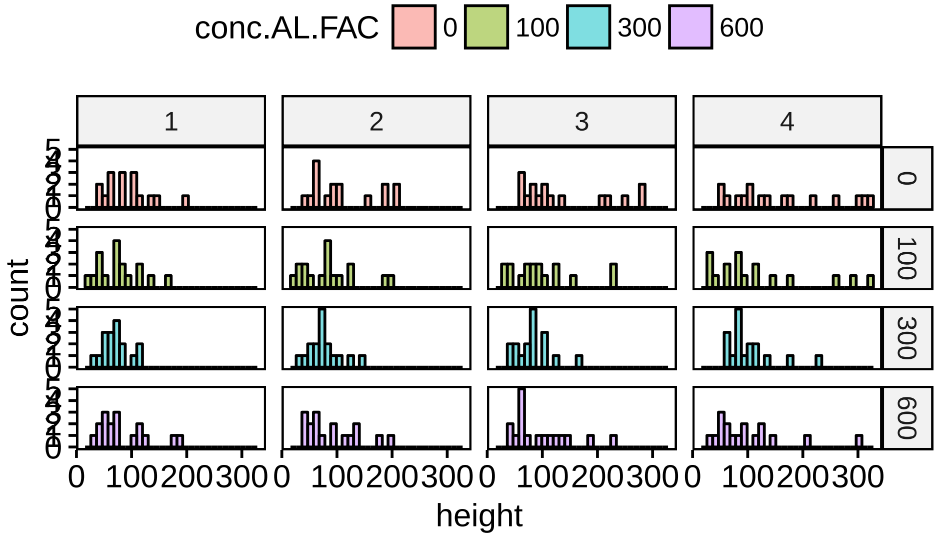
```
gghistogram(data = data.long,
            x = "height",
            facet.by = "conc.AL",
            fill = "conc.AL.FAC")
```

Facetting by 2 variables

We can "facet by" 2 variables. The syntax here require that we give facet.by 2 things contained in a c(thing1, thing2), eg facet.by = c("conc.AL", "week"). Color coding with "fill =" is very useful here for highlight similarlies in the data (each, each row is the same conc. of AL)

```
gghistogram(data = data.long,
            x = "height",
            facet.by = c("conc.AL","week"),
            fill = "conc.AL.FAC")
```

```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```



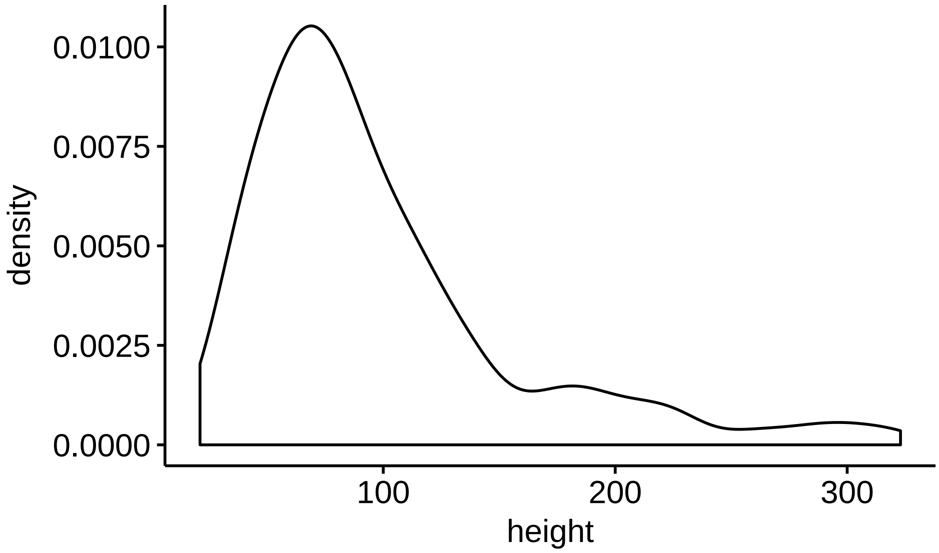
Density plots

Histograms are a fabulous tool. However, sometiems when you start making lots of facets its gets hard to make comparisons. Boxplots can work better for this. Another option is a density plot using ggdensity()

Basic density plot

ggdensity() works very similar to gghistogram()

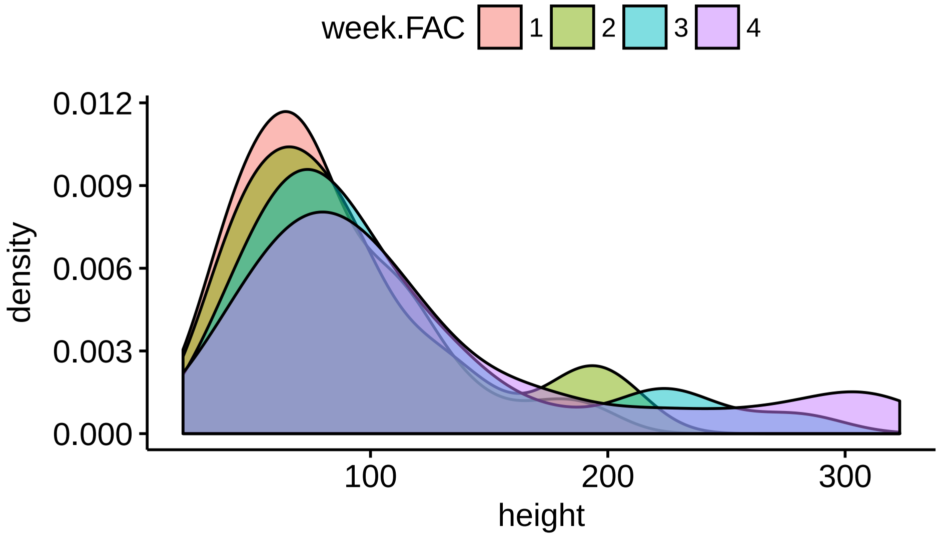
```
ggdensity(data = data.long,
            x = "height")
```



Multiple density plots

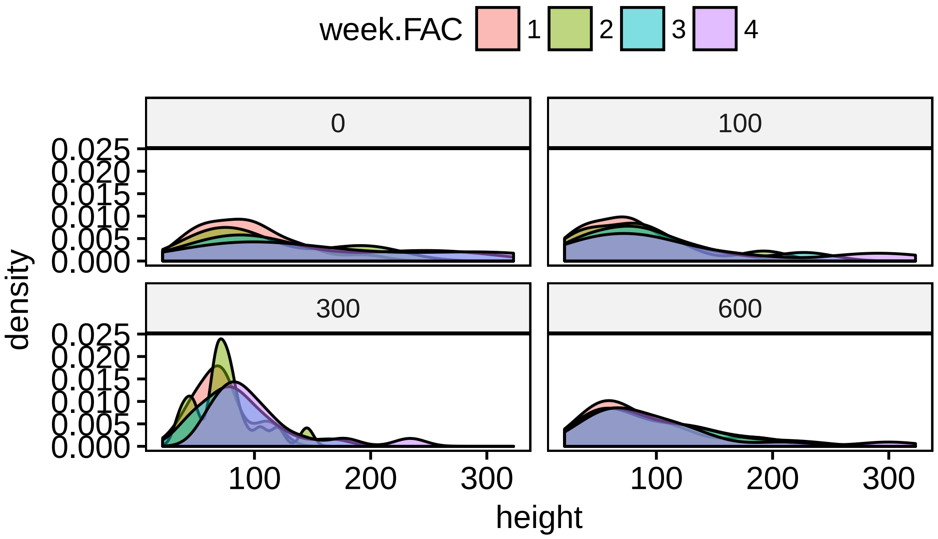
Density plots work well when you want to overlay multple distributions. HEre, I plot a seperate desnity plot of each weeks data and use a different color for fill. NOte that I'm using "week.FAC" (not just "week", which didn't seem to work)

```
ggdensity(data = data.long,
          x = "height",
          fill = "week.FAC")
```



You can also facet density plots

```
ggdensity(data = data.long,
          x = "height",
          fill = "week.FAC",
          facet.by = "conc.AL.FAC")
```



Challenge: What happens when we facet by both “conc.AL.FAC” and “week.FAC”? Do you understand why it does this?