Topic 2 Exercises Answers

Steve Paterson

2024-05-24

Please give the exercises a good try before looking at the answers.

> The file caffeine.txt contains the values of the urinary metabolic ratio of 5–acetylamino–6–formylamino–3–methyluracil to 1–methylxanthine (AFMU/1X) after oral administration of caffeine. Plot a histogram of the data and comment on its distribution.

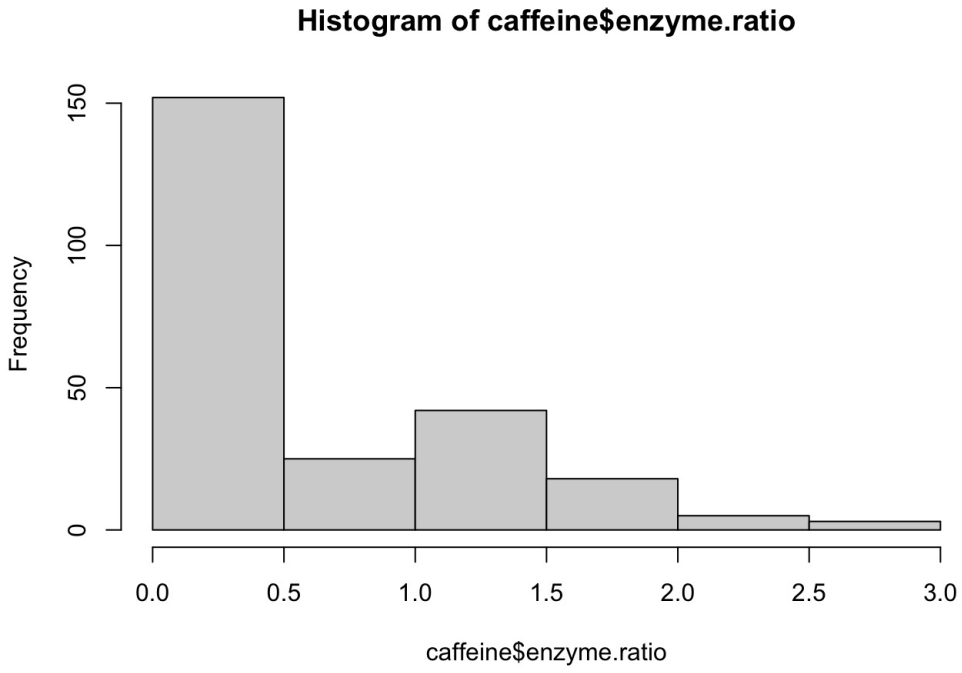
First read in the data. It's a .txt file, so use `read.table`. (Check spelling etc if it doesn't work.)

```
caffeine <- read.table("caffeine.txt", header=TRUE)
head(caffeine)
```

```
##   subject enzyme.ratio
## 1     S_1        0.130
## 2     S_2        0.080
## 3     S_3        1.261
## 4     S_4        0.224
## 5     S_5        0.132
## 6     S_6        1.052
```

You can do this in base R or in `ggplot`. In base R:

```
hist(caffeine$enzyme.ratio)
```

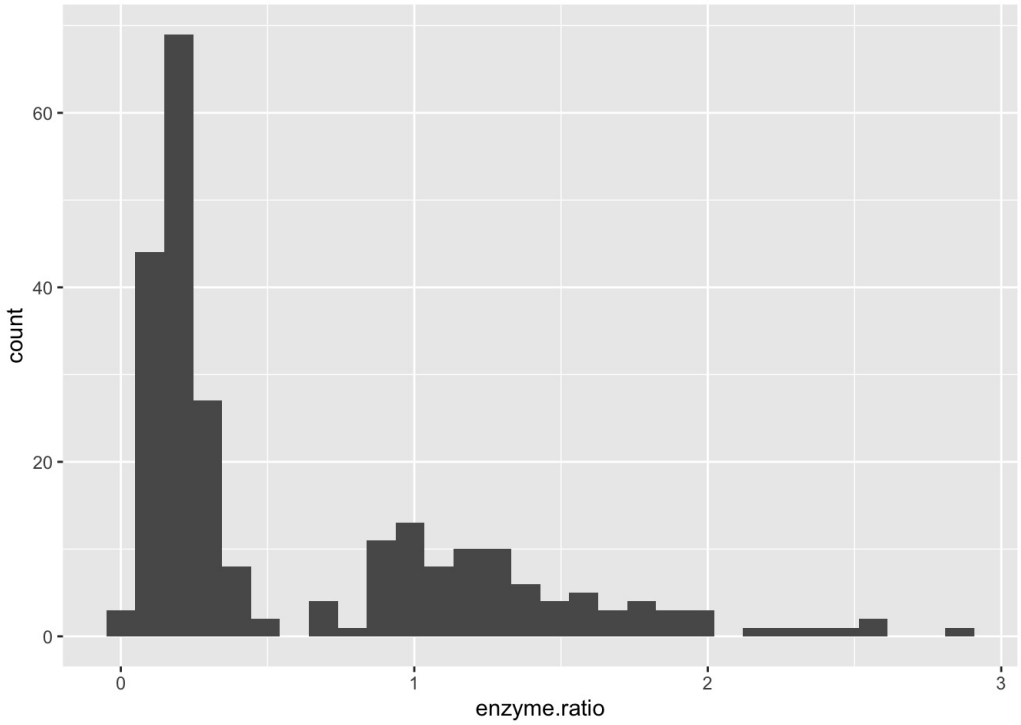


With `ggplot`

```
# What to plot
caffeine.plot <- ggplot(data=caffeine, aes(x=enzyme.ratio))

# How to plot it
caffeine.plot + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# can change the theme, x labels etc if you wish
# The default of 30 bins in geom_histogram probably gives a good enough impression of the distribution
```

Here enzyme.ratio seems to have a bimodal distribution (i.e. two distinct humps). This may reflect that the metabolism of a subset of individuals in the study respond differently to caffeine.

(If you're keen, you can replace `geom_histogram` with `geom_density` .)

> The file pancreatic.csv contains the concentrations of 2 biomarkers, CA19-9 and CA125 (in U/ml), in control (healthy) and diseased (diagnosed with pancreatic cancer) individuals. Plot these data to investigate the relationship of each biomarker to disease state.

This is a csv file, so needs `read.csv` .

```
pancreatic <- read.csv("pancreatic.csv") #read.csv assumes the first line is the header
head(pancreatic)
```

```
##   CA19.9 CA125  status
## 1   28.0  13.3 control
## 2   15.5  11.1 control
## 3    8.2  16.7 control
## 4    3.4  12.6 control
## 5   17.3   7.4 control
## 6   15.2   5.5 control
```

Let's make sure R knows that status is a factor; control or disease.

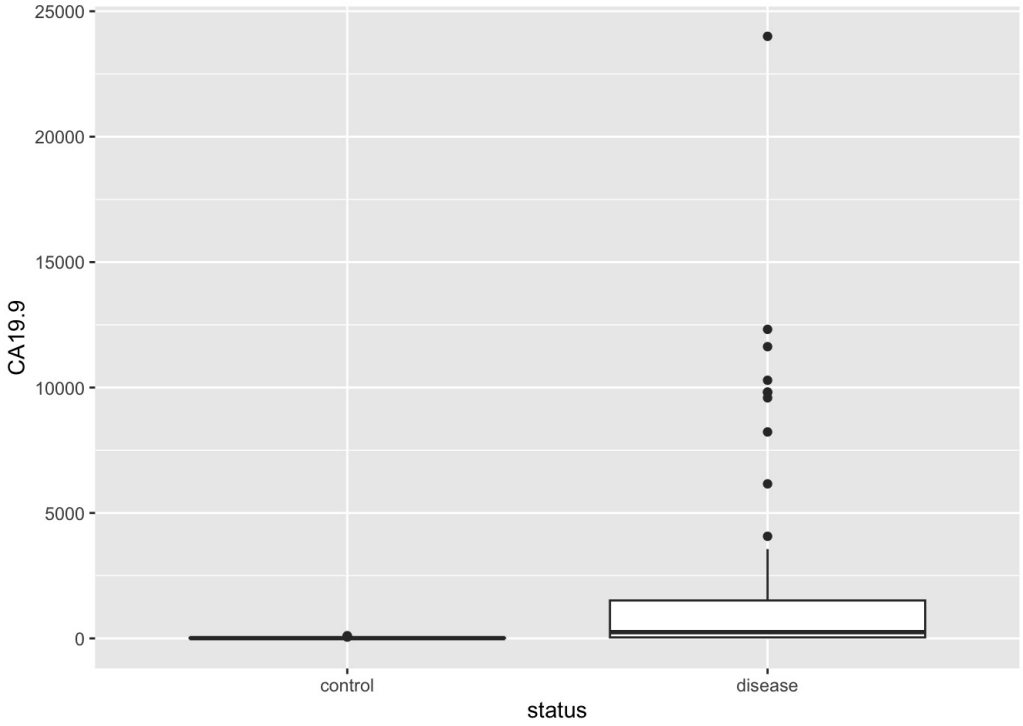
```
pancreatic$status <- factor(pancreatic$status)
```

We want to know whether either of the biomarkers differ between control and disease status. Let's use `ggplot` .

Biomarker CA19.9

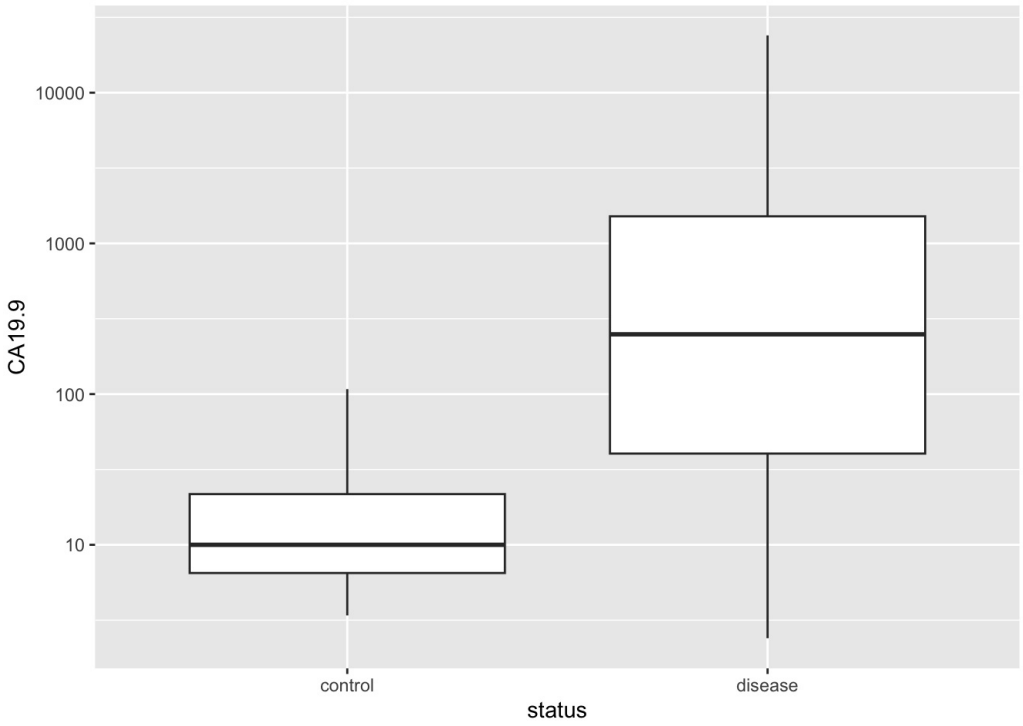
```
#what we want to plot
ca19.plot <- ggplot(data=pancreatic, aes(x=status, y=CA19.9))

#how we want to plot it
ca19.plot + geom_boxplot()
```



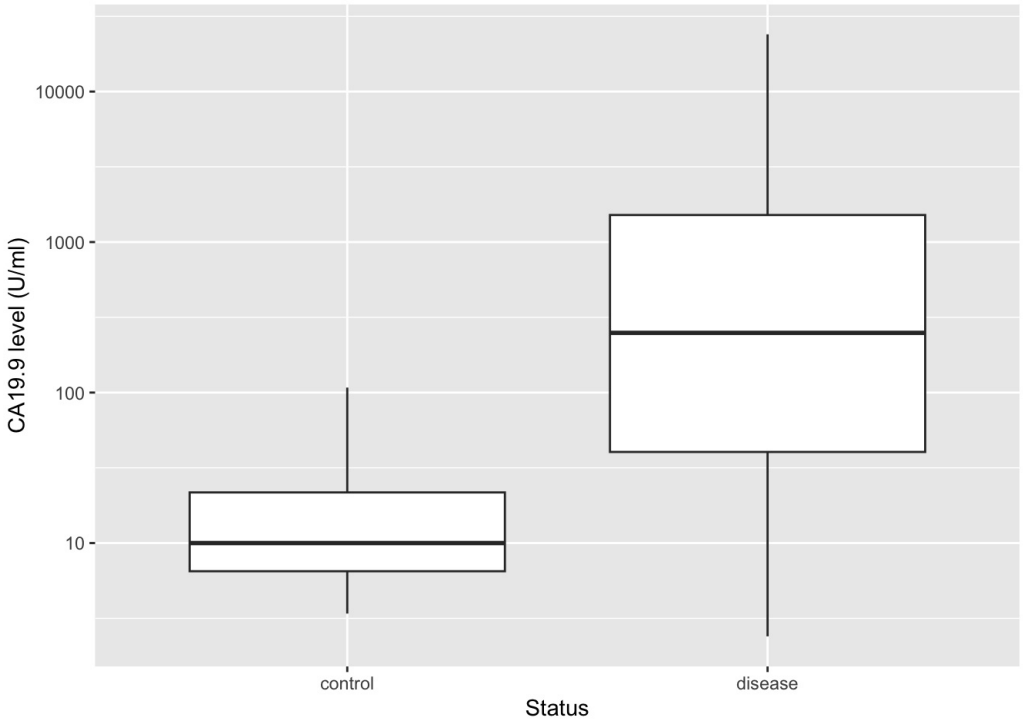
Here there does seem to be a difference, but most of the data are bunched up near zero. A log scale may be more appropriate.

```
ca19.plot + geom_boxplot() +
  scale_y_log10()
```



Probably more useful. Let's add better axes labels.

```
ca19.plot + geom_boxplot() +
  scale_y_log10() +
  xlab("Status") + ylab("CA19.9 level (U/ml)")
```

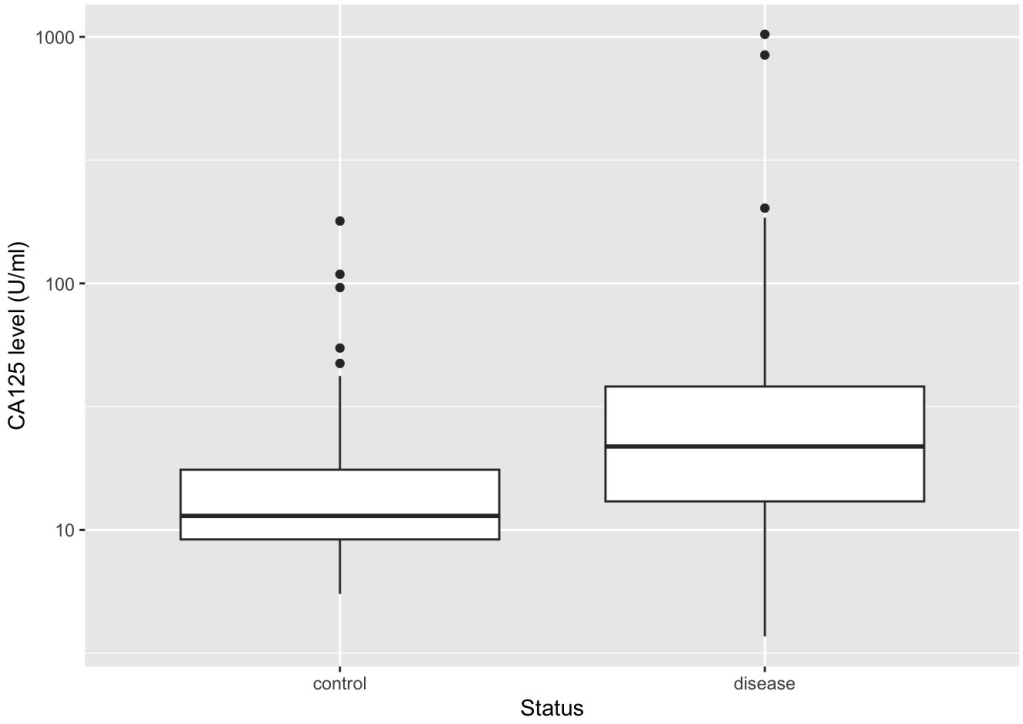


So there seems to be a difference. Patients with pancreatic cancer tend to have higher levels of CA19.9 than controls. There is still overlap though. If you're using this to make a clinical diagnosis, it's important to realise that some patients with cancer still have low CA19.9 levels. So we might be able to say that patients with levels above about 100 are likely to have cancer but we may not be able to say that patients below this don't. Even patients with very low levels might have cancer, particularly if they have arrived in a clinic experiencing symptoms.

Repeat for the other biomarker

```
# change the data we want to pplot
ca125.plot <- ggplot(data=pancreatic, aes(x=status, y=CA125))

#how we want to plot it
ca125.plot + geom_boxplot() +
  scale_y_log10() +
  xlab("Status") + ylab("CA125 level (U/ml)")
```



This is a similar plot. There is a difference between the two groups, but also a lot of overlap and it would probably be difficult to suggest any threshold that is indicative of having pancreatic cancer or not.

The file plant_height.csv contains data on the heights of different plants, plus various ecological factors from where grow.

First, produce a plot to compare the height (in metres) of different growthforms (herb, shrub or tree).

Second, produce a plot of how the height of these different growthforms varies with rain (the column 'rain', in mm/yr). Label the axes appropriately. What does `geom_smooth()` do? Try adding it to your plot.

```
plants <- read.csv("plant_height.csv")
head(plants)
```

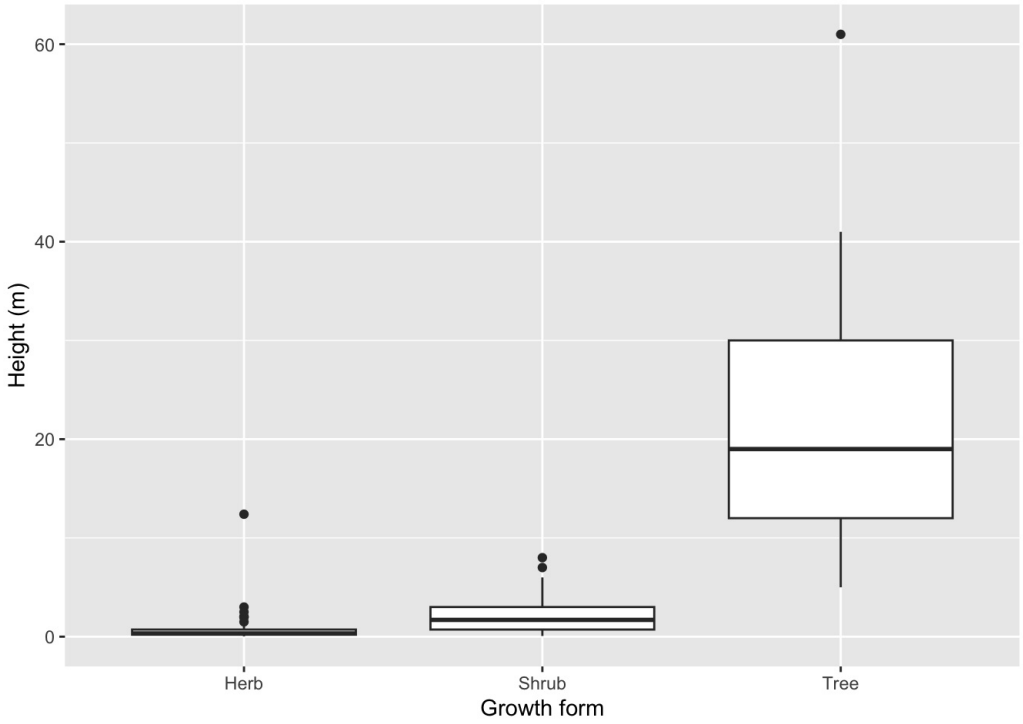
```
##   X sort_number site         Genus_species        Family growthform height
## 1 1        1402  193    Acer_macryophyllum   Sapindaceae       Tree   28.0
## 2 2       25246  103    Quararibea_cordata     Malvaceae       Tree   26.6
## 3 3       11648   54    Eragrostis_dielsii       Poaceae       Herb    0.3
## 4 4        8168  144    Cistus_salvifolius     Cistaceae      Shrub    1.6
## 5 5       22422  178          Phlox_bifida Polemoniaceae       Herb    0.2
## 6 6       15925   59 Homalium_betulifolium    Salicaceae      Shrub    1.7
##        loght       Country              Site    lat     long alt temp
## 1  1.4471580           USA    Oregon - McDun 44.600 -123.334 179 10.8
## 2  1.4248816          Peru              Manu 12.183  -70.550 386 24.5
## 3 -0.5228787     Australia Central Australia 23.800  133.833 553 20.9
## 4  0.2041200        Israel           Hanadiv 32.555   34.938 115 19.9
## 5 -0.6989700           USA     Indiana Dunes 41.617  -86.950 200  9.7
## 6  0.2304489 New Caledonia              <NA> 21.500  165.500  95 22.6
##   diurn.temp isotherm temp.seas temp.max.warm temp.min.cold temp.ann.range
## 1       11.8      4.4       5.2          27.0           0.3           26.7
## 2       10.8      7.4       0.9          31.2          16.7           14.5
## 3       16.3      4.8       6.0          37.0           3.6           33.4
## 4        9.7      4.4       4.9          30.7           8.7           22.0
## 5       10.7      2.8       9.7          28.6          -9.5           38.1
## 6        7.4      5.4       2.2          29.0          15.5           13.5
##   temp.mean.wetqr temp.mean.dryqr temp.mean.warmqr temp.mean.coldqr rain
## 1             4.9            17.4             17.6              4.5 1208
## 2            25.1            23.2             25.3             23.1 3015
## 3            28.1            14.8             28.1             12.8  278
## 4            13.6            25.3             25.7             13.6  598
## 5            21.6            -3.3             21.6             -3.3  976
## 6            25.4            20.4             25.4             19.7 1387
##   rain.wetm rain.drym rain.seas rain.wetqr rain.dryqr rain.warmqr rain.coldqr
## 1       217        13        69        601         68          75         560
## 2       416        99        45       1177        340         928         359
## 3        37         9        42        109         35         109          42
## 4       159         0       115        408          0           2         408
## 5       104        44        23        299        165         299         165
## 6       216        59        46        600        186         600         212
##    LAI  NPP hemisphere
## 1 2.51  572          1
## 2 4.26 1405         -1
## 3 1.32  756         -1
## 4 1.01  359          1
## 5 3.26 1131          1
## 6 6.99 1552         -1
```

```
#make sure growthform is a factor
plants$growthform <- factor(plants$growthform)
```

First, let's plot height vs growthform. This is just the same structure as the pancreatic example; it's a continous variable vs a factor.

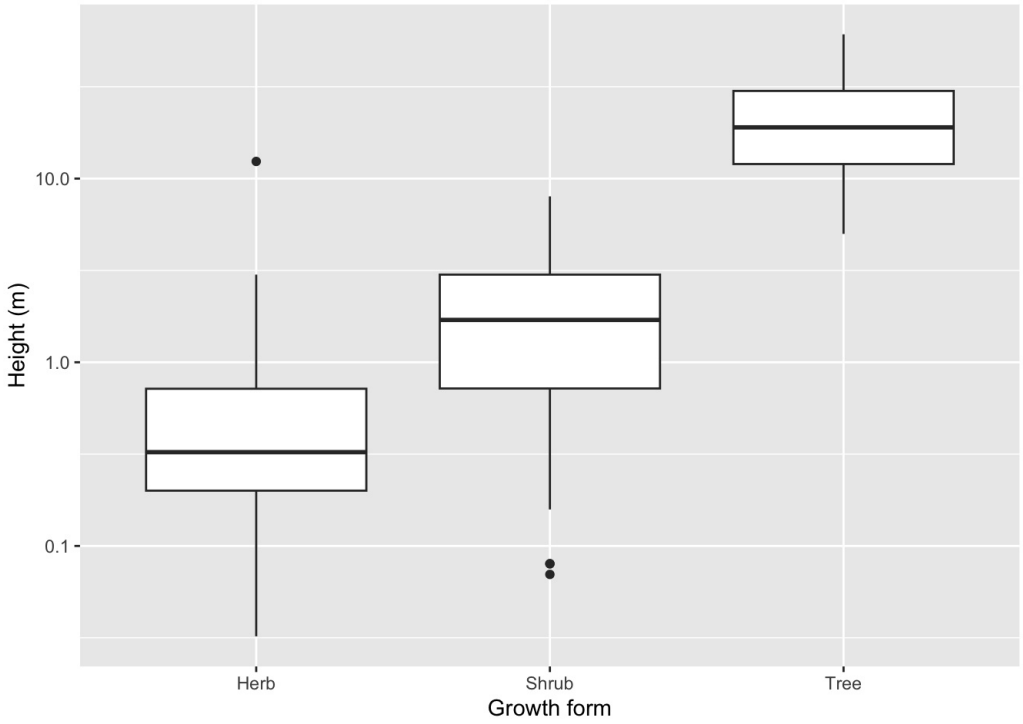
```
#what to plot
plant.plot1 <- ggplot(data = plants, aes(x=growthform, y=height))

#how to plot it
plant.plot1 + geom_boxplot() +
  xlab("Growth form") + ylab("Height (m)")
```



Again it may be easier to interpret on a log scale, we just add `scale_y_log10()`.

```
plant.plot1 + geom_boxplot() +
  xlab("Growth form") + ylab("Height (m)") +
  scale_y_log10()
```

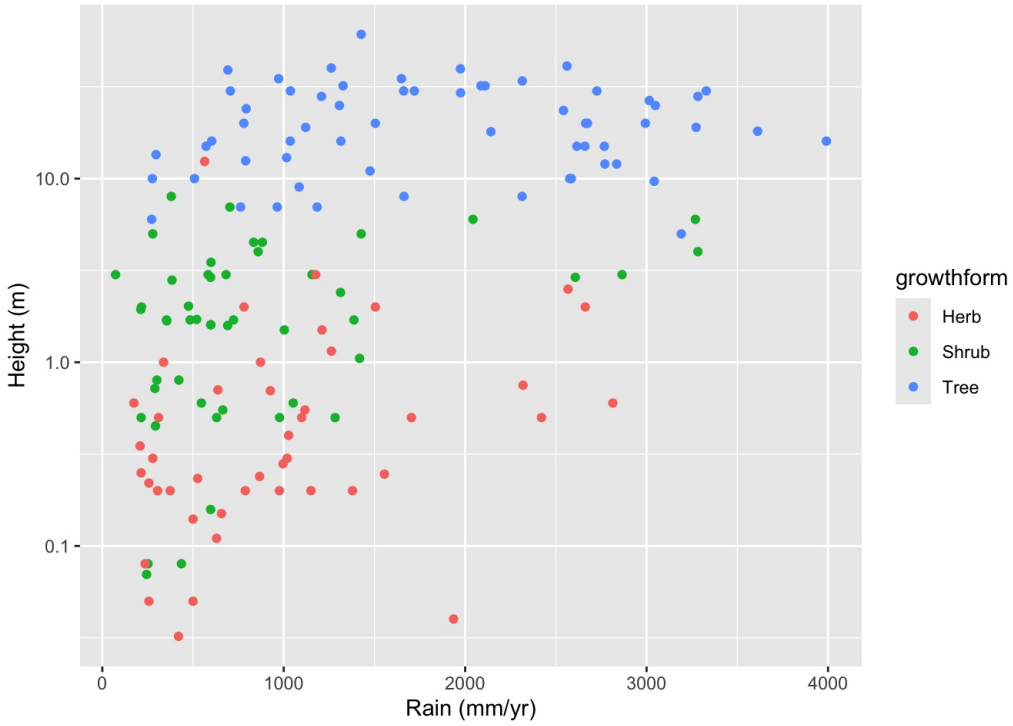


Trees are bigger than shrubs are bigger than herbs. That's not earth-shattering biology. But hopefully you're finding it easier to do the plots now.

Next, how do these forms differ with rain (we'll just take the 'rain' variable). In this case we need to specify height on the y axis and rain on the x axis. The first way you saw to do this was to plot points in different colours (note the american spelling below).

```
plant.plot2 <- ggplot(data=plants, aes(x=rain, y = height, color= growthform))

plant.plot2 + geom_point() +
  scale_y_log10() +
  xlab("Rain (mm/yr)") + ylab("Height (m)")
```

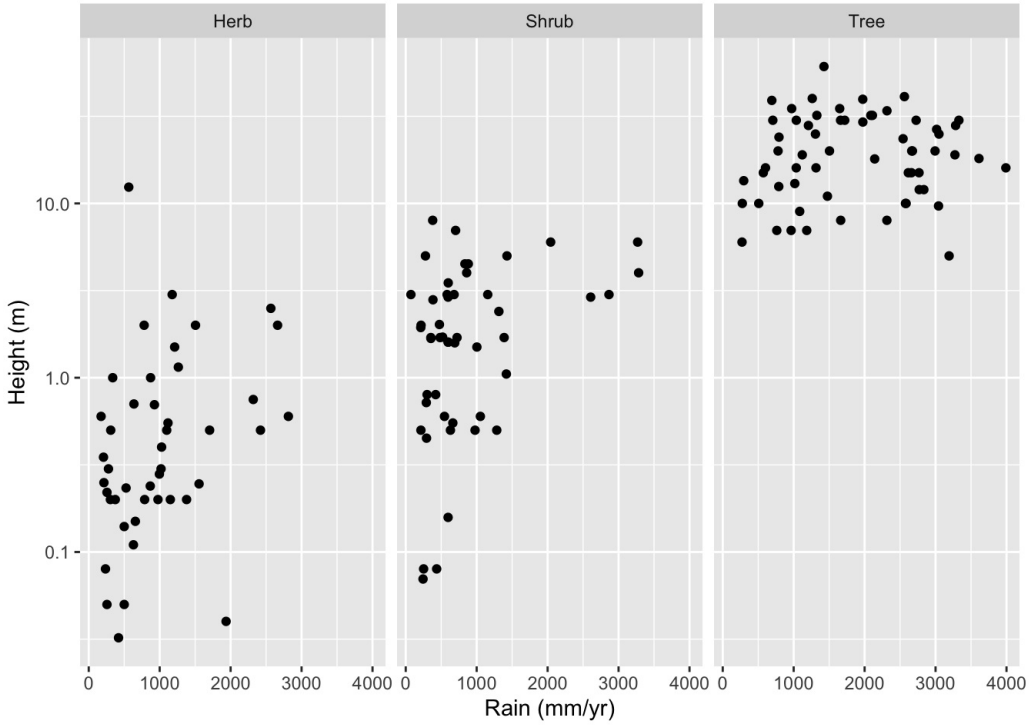


(One could also try logging the x axis as well by adding `scale_x_log10()`, but I'm not sure it really helps a lot since the points are reasonably well spread along the x axes.)

An alternative to the plot above is to produce separate plots, You can either keep the different colours or, as below, just have black points.

```
plant.plot3 <- ggplot(data=plants, aes(x=rain, y = height)) # deleted the color=growthform argument

plant.plot3 + geom_point() +
  scale_y_log10() +
  xlab("Rain (mm/yr)") + ylab("Height (m)") +
  facet_wrap(vars(growthform))
```

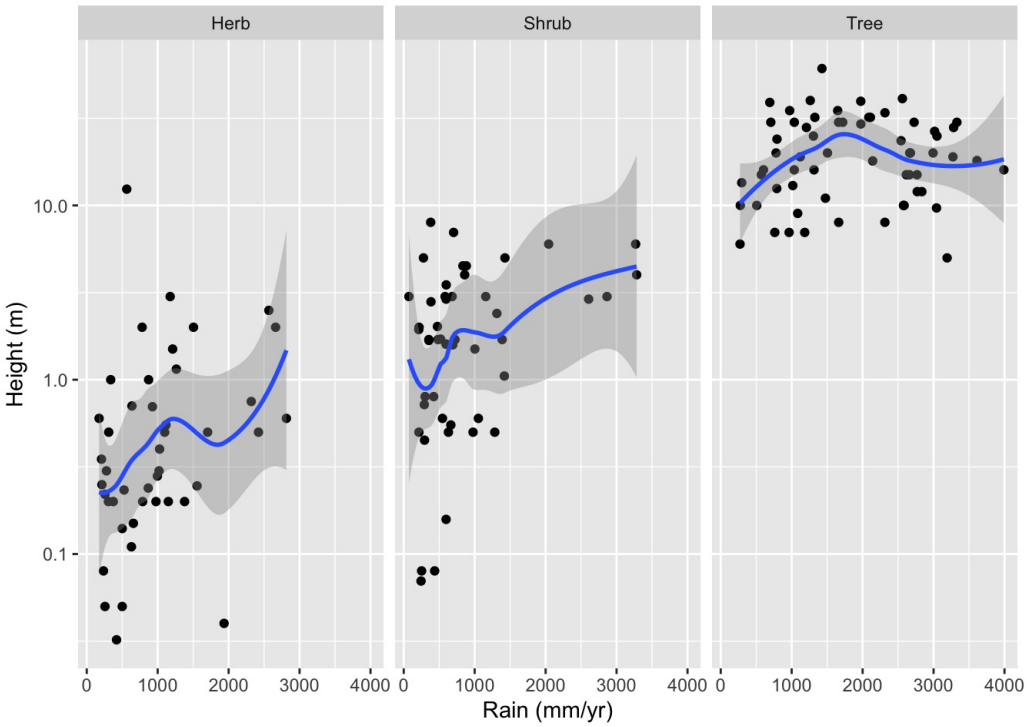


So there is possibly a positive relationship between rain and height in herbs and shrubs but not trees. Testing such relationships statistically is the subject of later topics.

What does `geom_smooth` do? To see just add it to the code:

```
plant.plot3 + geom_point() +
  scale_y_log10() +
  xlab("Rain (mm/yr)") + ylab("Height (m)") +
  facet_wrap(vars(growthform)) +
  geom_smooth()
```

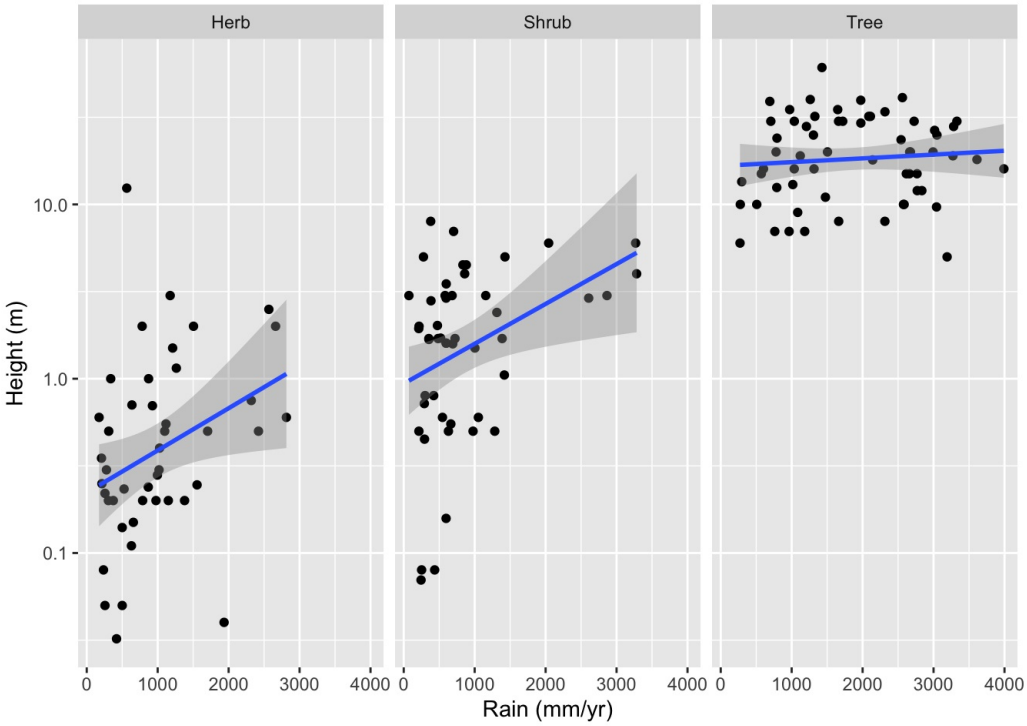
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



The default 'loess' function here plots a wiggly line for each plot (in blue), plus standard errors around this line. Type `?geom_smooth` to get the help page. To plot a straight line, use `geom_smooth(method='lm')`.

```
plant.plot3 + geom_point() +
  scale_y_log10() +
  xlab("Rain (mm/yr)") + ylab("Height (m)") +
  facet_wrap(vars(growthform)) +
  geom_smooth(method='lm')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



If you've managed to do these you've done well. The syntax for `ggplot` can be off-putting at first but is very powerful once you get the hang of it. For a report, being able to present data clearly, with appropriately labelled axes and figure legends is something that you'll be assessed on, both in this module and elsewhere on your programme.