Pair_Programming_Long_data_Module3

HD Sheets

2024-08-12

DSE5001 Module 3 pair programming exercise HD Sheets 8/13/2024 checked 01/03/2025

Pair Programming Long Data, Module 3

We will look at the World Phones Data set

This data set needs a lot of work

-it doesn't start out as a data frame -the regions and years are labels, not values -the rows and columns need to be flipped -we need to convert it to long form

This is an example of "data wrangling" in which we need to do a lot of data manipulation and structuring before we can do anything useful with it.

Watch the steps needed to do this

-changing from one data storage form to another -changing data types/formats -transposing-swapping rows and columns -more complex changes

Load the data and look at it:

```
data("WorldPhones")
WorldPhones
```

```
##      N.Amer Europe Asia S.Amer Oceania Africa Mid.Amer
## 1951  45939  21574 2876   1815    1646     89      555
## 1956  60423  29990 4708   2568    2366   1411      733
## 1957  64721  32510 5230   2695    2526   1546      773
## 1958  68484  35218 6662   2845    2691   1663      836
## 1959  71799  37598 6856   3000    2868   1769      911
## 1960  76036  40341 8220   3145    3054   1905     1008
## 1961  79831  43173 9053   3338    3224   2005     1076
```

Okay, so what are the problems here?

This is not a particularly unusual table, but it' still a mess.

The variable being measured is the number of phones (in units of a thousand phones)

They are recorded at different times and different locations

There is a composite key here, the region and the year with the measured variable being the number of phones

What type of data storage is this? Use str() to find out what we are dealing with

```
str(WorldPhones)
```

```
##  num [1:7, 1:7] 45939 60423 64721 68484 71799 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:7] "1951" "1956" "1957" "1958" ...
##   ..$ : chr [1:7] "N.Amer" "Europe" "Asia" "S.Amer" ...
```

I want to transpose this to flip the rows and columns and then put this into a data frame

t()- transpose, converting rows t columns

data.frame()- convert from a numerical matrix to a dataframe

```
phone_df=data.frame(t(WorldPhones))
phone_df
```

```
##          X1951 X1956 X1957 X1958 X1959 X1960 X1961
## N.Amer   45939 60423 64721 68484 71799 76036 79831
## Europe   21574 29990 32510 35218 37598 40341 43173
## Asia      2876  4708  5230  6662  6856  8220  9053
## S.Amer    1815  2568  2695  2845  3000  3145  3338
## Oceania   1646  2366  2526  2691  2868  3054  3224
## Africa      89  1411  1546  1663  1769  1905  2005
## Mid.Amer   555   733   773   836   911  1008  1076
```

That did odd things to the column names, they have an X in them now

we can use rename to rename all the columns, sorta annoying but not hard

```
library("tidyverse")
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
phone_df=phone_df |>rename("1951" = X1951,"1956"=X1956,"1957"=X1957,"1958"=X1958,"1959"=X1959,
                           "1960"=X1960,"1961"=X1961)
```

```
phone_df
```

```
##            1951  1956  1957  1958  1959  1960  1961
## N.Amer   45939 60423 64721 68484 71799 76036 79831
## Europe   21574 29990 32510 35218 37598 40341 43173
## Asia      2876  4708  5230  6662  6856  8220  9053
## S.Amer    1815  2568  2695  2845  3000  3145  3338
## Oceania   1646  2366  2526  2691  2868  3054  3224
## Africa      89  1411  1546  1663  1769  1905  2005
## Mid.Amer   555   733   773   836   911  1008  1076
```

Right now, the regions are row labels, not variables. Dang.

Notice that in the list of regions, there is no listed column name, that is because these values are not in a column, they are labels for each row.

We need to add a column that is equal to the regions

I want to pivot longer and to do that the regions have to be in a variable,

```
phone_df=phone_df |> mutate(region=rownames(phone_df))
```

```
phone_df
```

```
##            1951  1956  1957  1958  1959  1960  1961   region
## N.Amer   45939 60423 64721 68484 71799 76036 79831   N.Amer
## Europe   21574 29990 32510 35218 37598 40341 43173   Europe
## Asia      2876  4708  5230  6662  6856  8220  9053     Asia
## S.Amer    1815  2568  2695  2845  3000  3145  3338   S.Amer
## Oceania   1646  2366  2526  2691  2868  3054  3224  Oceania
## Africa      89  1411  1546  1663  1769  1905  2005   Africa
## Mid.Amer   555   733   773   836   911  1008  1076 Mid.Amer
```

Now let's convert this to Long form

In the long form all the variables except region are being converted to entries in the "year" column, with the associate values of those years being stored in "phones"

This is an example of key-value storage. There is an identifier "region" and then a key-value pair of the year (variable) and the number of phones (the value)

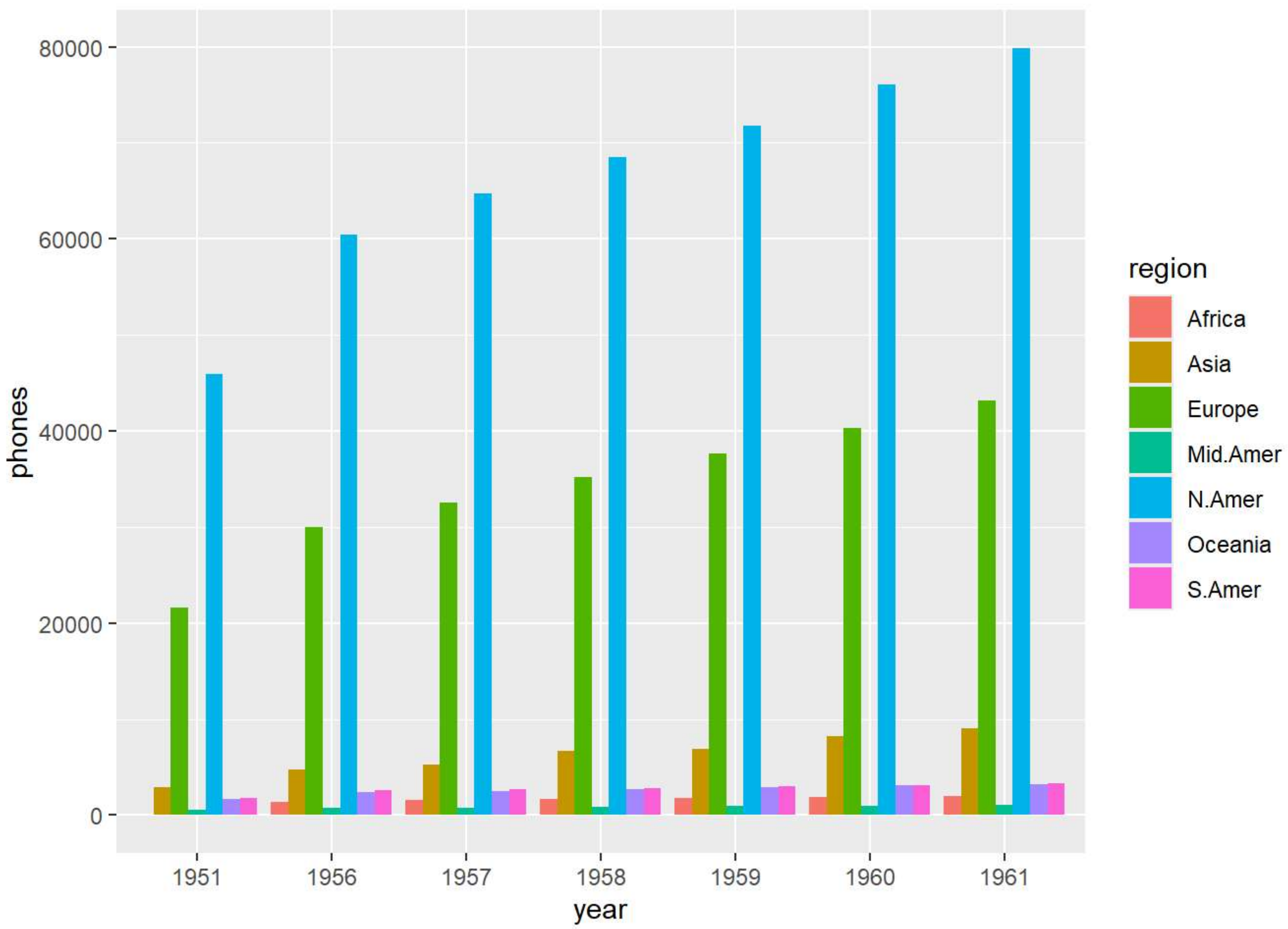
```
df_phones_long<-phone_df |> pivot_longer(!region,names_to="year",values_to="phones")

df_phones_long
```

```
## # A tibble: 49 × 3
##    region year  phones
##    <chr>  <chr>  <dbl>
##  1 N.Amer 1951   45939
##  2 N.Amer 1956   60423
##  3 N.Amer 1957   64721
##  4 N.Amer 1958   68484
##  5 N.Amer 1959   71799
##  6 N.Amer 1960   76036
##  7 N.Amer 1961   79831
##  8 Europe 1951   21574
##  9 Europe 1956   29990
## 10 Europe 1957   32510
## # ℹ 39 more rows
```

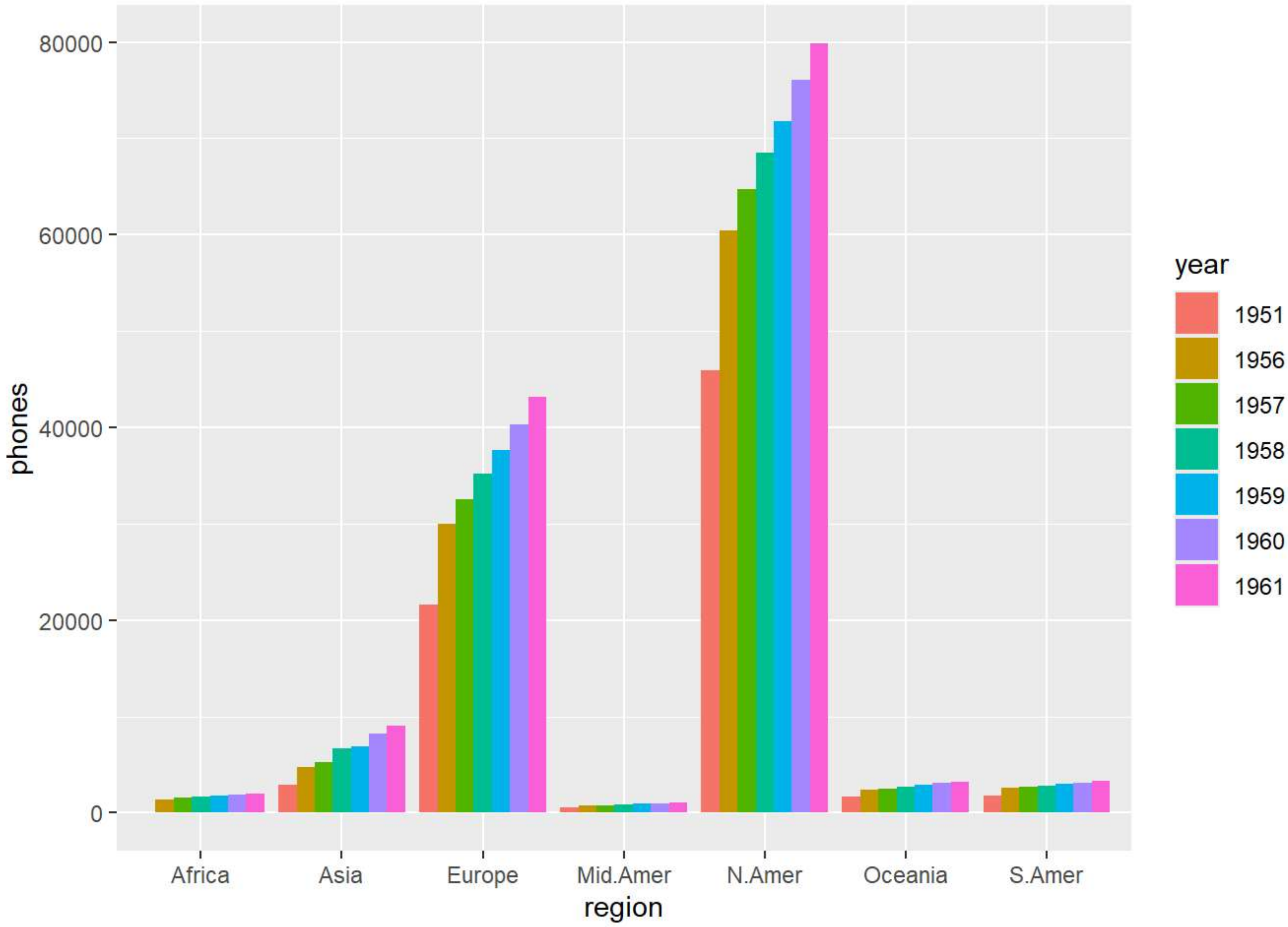
Okay, that's much better

We can easily create some interesting visuals now

```
library(ggplot2)

ggplot(df_phones_long,aes(x=year, y=phones,fill=region))+geom_bar(stat="identity",position="dodg
e")
```



```
library(ggplot2)

ggplot(df_phones_long,aes(x=region, y=phones,fill=year))+geom_bar(stat="identity",position="dodg
e")
```



Question/Action

The labels along the x-axis of the graph overlap each other and cannot be read.

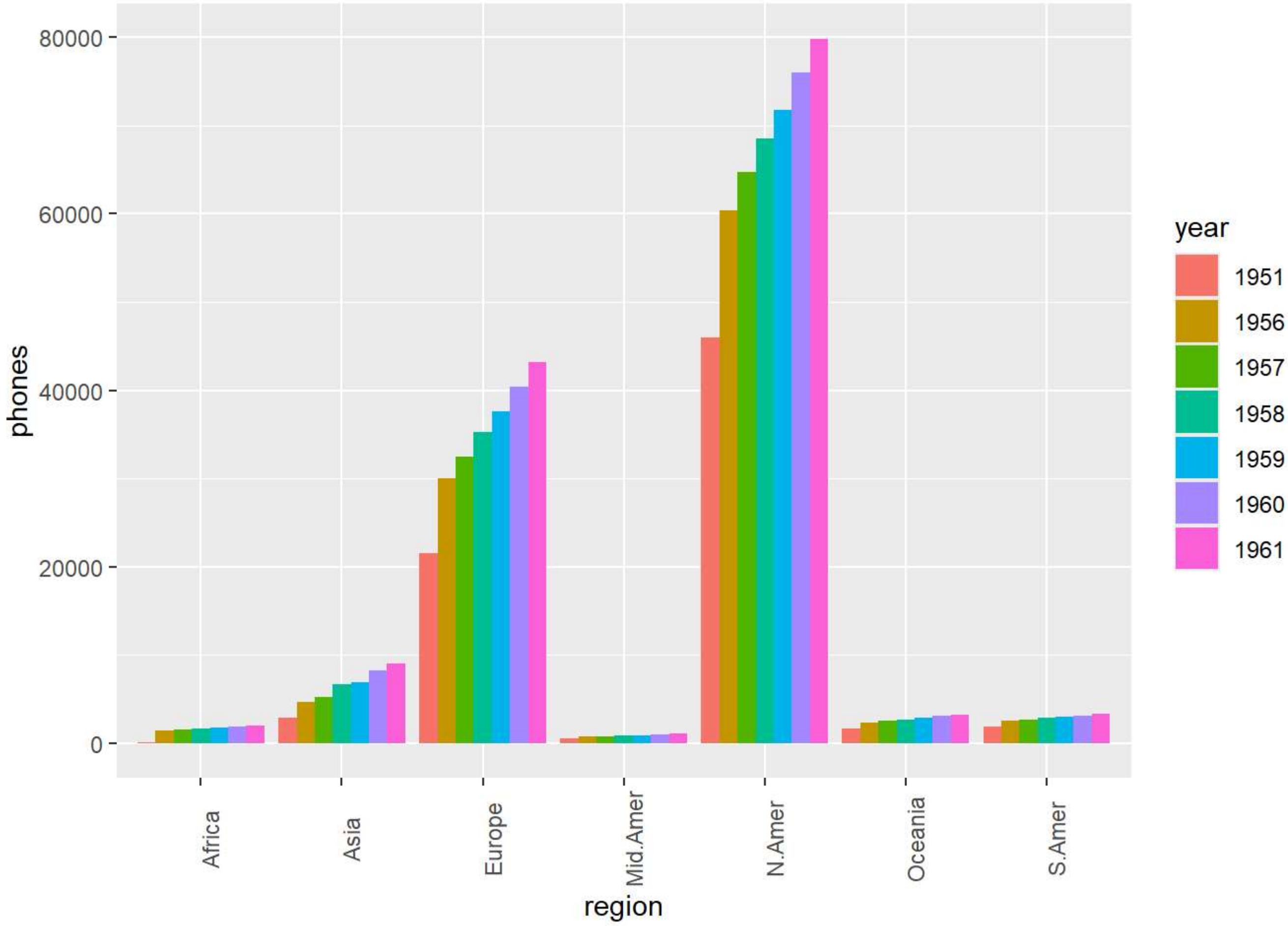
This is not “okay”, you can’t show anyone this graph like this.

We could fix the problem by figuring out how to rotate the labels along the x-axis by ninety degrees.

ggplot allows for fine control of graph elements, such as the x-axis label.

Google search and figure out how to rotate the x-axis labels by 90 degrees on this plot.
Create a new code cell and enter the corrected R code to create the plot above with the x-axis labels rotated by 90 degrees to make them readable.

```
ggplot(df_phones_long,aes(x=region, y=phones,fill=year))+geom_bar(stat="identity",position="dodg
e") + theme(axis.text.x = element_text(angle = 90))
```



Question/Action

Here is the Iris data set collected by Anderson and used in a famous paper by RA Fisher

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Question/Action

Do the following in a series of cells

-add the row number as data column, call it FlowerID

-convert this to long form, iris_long -you should have FlowerID and species as your two keys -names_to should be “flower part” -values_to should be “dimension”

-create a boxplot of values as y=dimension grouped by Species and flowerpart This is a group boxplot using dimension and Species as the grouping variables see https://r-graph-gallery.com/265-grouped-boxplot-with-ggplot2.html (https://r-graph-gallery.com/265-grouped-boxplot-with-ggplot2.html)

```
       use x= Species and color=flowerpart
```

-reverse the grouping order above, so you have y=dimension grouped by flowerpart and Species

```
iris_df=iris |> mutate(FlowerID=rownames(iris))
```

```
iris_df
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species FlowerID
## 1           5.1         3.5          1.4         0.2     setosa        1
## 2           4.9         3.0          1.4         0.2     setosa        2
## 3           4.7         3.2          1.3         0.2     setosa        3
## 4           4.6         3.1          1.5         0.2     setosa        4
## 5           5.0         3.6          1.4         0.2     setosa        5
## 6           5.4         3.9          1.7         0.4     setosa        6
## 7           4.6         3.4          1.4         0.3     setosa        7
## 8           5.0         3.4          1.5         0.2     setosa        8
## 9           4.4         2.9          1.4         0.2     setosa        9
## 10          4.9         3.1          1.5         0.1     setosa       10
## 11          5.4         3.7          1.5         0.2     setosa       11
## 12          4.8         3.4          1.6         0.2     setosa       12
## 13          4.8         3.0          1.4         0.1     setosa       13
## 14          4.3         3.0          1.1         0.1     setosa       14
## 15          5.8         4.0          1.2         0.2     setosa       15
## 16          5.7         4.4          1.5         0.4     setosa       16
## 17          5.4         3.9          1.3         0.4     setosa       17
## 18          5.1         3.5          1.4         0.3     setosa       18
## 19          5.7         3.8          1.7         0.3     setosa       19
## 20          5.1         3.8          1.5         0.3     setosa       20
## 21          5.4         3.4          1.7         0.2     setosa       21
## 22          5.1         3.7          1.5         0.4     setosa       22
## 23          4.6         3.6          1.0         0.2     setosa       23
## 24          5.1         3.3          1.7         0.5     setosa       24
## 25          4.8         3.4          1.9         0.2     setosa       25
## 26          5.0         3.0          1.6         0.2     setosa       26
## 27          5.0         3.4          1.6         0.4     setosa       27
## 28          5.2         3.5          1.5         0.2     setosa       28
## 29          5.2         3.4          1.4         0.2     setosa       29
## 30          4.7         3.2          1.6         0.2     setosa       30
## 31          4.8         3.1          1.6         0.2     setosa       31
## 32          5.4         3.4          1.5         0.4     setosa       32
## 33          5.2         4.1          1.5         0.1     setosa       33
## 34          5.5         4.2          1.4         0.2     setosa       34
## 35          4.9         3.1          1.5         0.2     setosa       35
## 36          5.0         3.2          1.2         0.2     setosa       36
## 37          5.5         3.5          1.3         0.2     setosa       37
## 38          4.9         3.6          1.4         0.1     setosa       38
## 39          4.4         3.0          1.3         0.2     setosa       39
## 40          5.1         3.4          1.5         0.2     setosa       40
## 41          5.0         3.5          1.3         0.3     setosa       41
## 42          4.5         2.3          1.3         0.3     setosa       42
## 43          4.4         3.2          1.3         0.2     setosa       43
## 44          5.0         3.5          1.6         0.6     setosa       44
## 45          5.1         3.8          1.9         0.4     setosa       45
## 46          4.8         3.0          1.4         0.3     setosa       46
## 47          5.1         3.8          1.6         0.2     setosa       47
## 48          4.6         3.2          1.4         0.2     setosa       48
## 49          5.3         3.7          1.5         0.2     setosa       49
## 50          5.0         3.3          1.4         0.2     setosa       50
## 51          7.0         3.2          4.7         1.4 versicolor       51
```

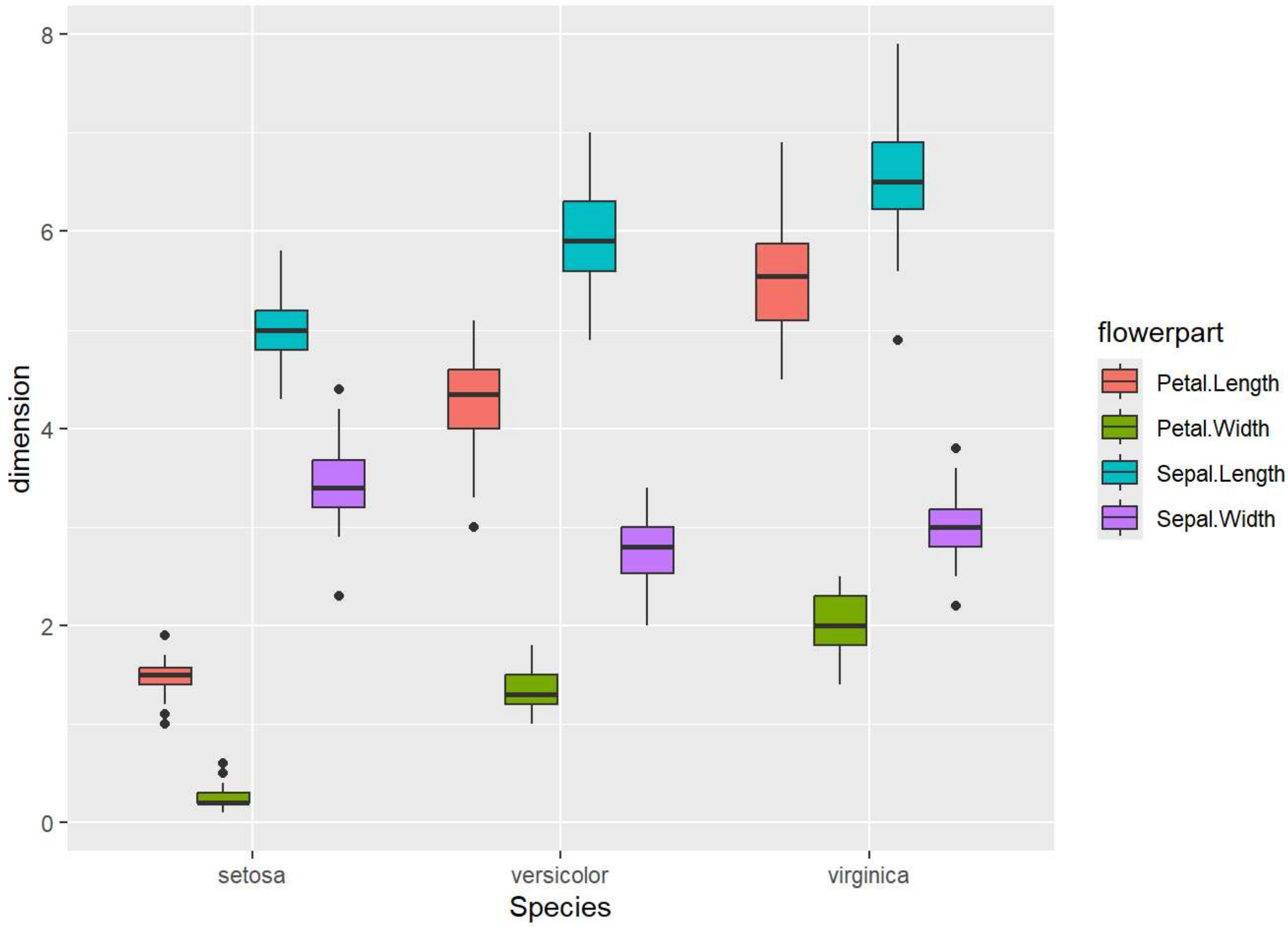
```
## 52           6.4         3.2          4.5         1.5 versicolor      52
## 53           6.9         3.1          4.9         1.5 versicolor      53
## 54           5.5         2.3          4.0         1.3 versicolor      54
## 55           6.5         2.8          4.6         1.5 versicolor      55
## 56           5.7         2.8          4.5         1.3 versicolor      56
## 57           6.3         3.3          4.7         1.6 versicolor      57
## 58           4.9         2.4          3.3         1.0 versicolor      58
## 59           6.6         2.9          4.6         1.3 versicolor      59
## 60           5.2         2.7          3.9         1.4 versicolor      60
## 61           5.0         2.0          3.5         1.0 versicolor      61
## 62           5.9         3.0          4.2         1.5 versicolor      62
## 63           6.0         2.2          4.0         1.0 versicolor      63
## 64           6.1         2.9          4.7         1.4 versicolor      64
## 65           5.6         2.9          3.6         1.3 versicolor      65
## 66           6.7         3.1          4.4         1.4 versicolor      66
## 67           5.6         3.0          4.5         1.5 versicolor      67
## 68           5.8         2.7          4.1         1.0 versicolor      68
## 69           6.2         2.2          4.5         1.5 versicolor      69
## 70           5.6         2.5          3.9         1.1 versicolor      70
## 71           5.9         3.2          4.8         1.8 versicolor      71
## 72           6.1         2.8          4.0         1.3 versicolor      72
## 73           6.3         2.5          4.9         1.5 versicolor      73
## 74           6.1         2.8          4.7         1.2 versicolor      74
## 75           6.4         2.9          4.3         1.3 versicolor      75
## 76           6.6         3.0          4.4         1.4 versicolor      76
## 77           6.8         2.8          4.8         1.4 versicolor      77
## 78           6.7         3.0          5.0         1.7 versicolor      78
## 79           6.0         2.9          4.5         1.5 versicolor      79
## 80           5.7         2.6          3.5         1.0 versicolor      80
## 81           5.5         2.4          3.8         1.1 versicolor      81
## 82           5.5         2.4          3.7         1.0 versicolor      82
## 83           5.8         2.7          3.9         1.2 versicolor      83
## 84           6.0         2.7          5.1         1.6 versicolor      84
## 85           5.4         3.0          4.5         1.5 versicolor      85
## 86           6.0         3.4          4.5         1.6 versicolor      86
## 87           6.7         3.1          4.7         1.5 versicolor      87
## 88           6.3         2.3          4.4         1.3 versicolor      88
## 89           5.6         3.0          4.1         1.3 versicolor      89
## 90           5.5         2.5          4.0         1.3 versicolor      90
## 91           5.5         2.6          4.4         1.2 versicolor      91
## 92           6.1         3.0          4.6         1.4 versicolor      92
## 93           5.8         2.6          4.0         1.2 versicolor      93
## 94           5.0         2.3          3.3         1.0 versicolor      94
## 95           5.6         2.7          4.2         1.3 versicolor      95
## 96           5.7         3.0          4.2         1.2 versicolor      96
## 97           5.7         2.9          4.2         1.3 versicolor      97
## 98           6.2         2.9          4.3         1.3 versicolor      98
## 99           5.1         2.5          3.0         1.1 versicolor      99
## 100          5.7         2.8          4.1         1.3 versicolor     100
## 101          6.3         3.3          6.0         2.5  virginica     101
## 102          5.8         2.7          5.1         1.9  virginica     102
## 103          7.1         3.0          5.9         2.1  virginica     103
```

```
## 104          6.3        2.9         5.6        1.8  virginica     104
## 105          6.5        3.0         5.8        2.2  virginica     105
## 106          7.6        3.0         6.6        2.1  virginica     106
## 107          4.9        2.5         4.5        1.7  virginica     107
## 108          7.3        2.9         6.3        1.8  virginica     108
## 109          6.7        2.5         5.8        1.8  virginica     109
## 110          7.2        3.6         6.1        2.5  virginica     110
## 111          6.5        3.2         5.1        2.0  virginica     111
## 112          6.4        2.7         5.3        1.9  virginica     112
## 113          6.8        3.0         5.5        2.1  virginica     113
## 114          5.7        2.5         5.0        2.0  virginica     114
## 115          5.8        2.8         5.1        2.4  virginica     115
## 116          6.4        3.2         5.3        2.3  virginica     116
## 117          6.5        3.0         5.5        1.8  virginica     117
## 118          7.7        3.8         6.7        2.2  virginica     118
## 119          7.7        2.6         6.9        2.3  virginica     119
## 120          6.0        2.2         5.0        1.5  virginica     120
## 121          6.9        3.2         5.7        2.3  virginica     121
## 122          5.6        2.8         4.9        2.0  virginica     122
## 123          7.7        2.8         6.7        2.0  virginica     123
## 124          6.3        2.7         4.9        1.8  virginica     124
## 125          6.7        3.3         5.7        2.1  virginica     125
## 126          7.2        3.2         6.0        1.8  virginica     126
## 127          6.2        2.8         4.8        1.8  virginica     127
## 128          6.1        3.0         4.9        1.8  virginica     128
## 129          6.4        2.8         5.6        2.1  virginica     129
## 130          7.2        3.0         5.8        1.6  virginica     130
## 131          7.4        2.8         6.1        1.9  virginica     131
## 132          7.9        3.8         6.4        2.0  virginica     132
## 133          6.4        2.8         5.6        2.2  virginica     133
## 134          6.3        2.8         5.1        1.5  virginica     134
## 135          6.1        2.6         5.6        1.4  virginica     135
## 136          7.7        3.0         6.1        2.3  virginica     136
## 137          6.3        3.4         5.6        2.4  virginica     137
## 138          6.4        3.1         5.5        1.8  virginica     138
## 139          6.0        3.0         4.8        1.8  virginica     139
## 140          6.9        3.1         5.4        2.1  virginica     140
## 141          6.7        3.1         5.6        2.4  virginica     141
## 142          6.9        3.1         5.1        2.3  virginica     142
## 143          5.8        2.7         5.1        1.9  virginica     143
## 144          6.8        3.2         5.9        2.3  virginica     144
## 145          6.7        3.3         5.7        2.5  virginica     145
## 146          6.7        3.0         5.2        2.3  virginica     146
## 147          6.3        2.5         5.0        1.9  virginica     147
## 148          6.5        3.0         5.2        2.0  virginica     148
## 149          6.2        3.4         5.4        2.3  virginica     149
## 150          5.9        3.0         5.1        1.8  virginica     150
```

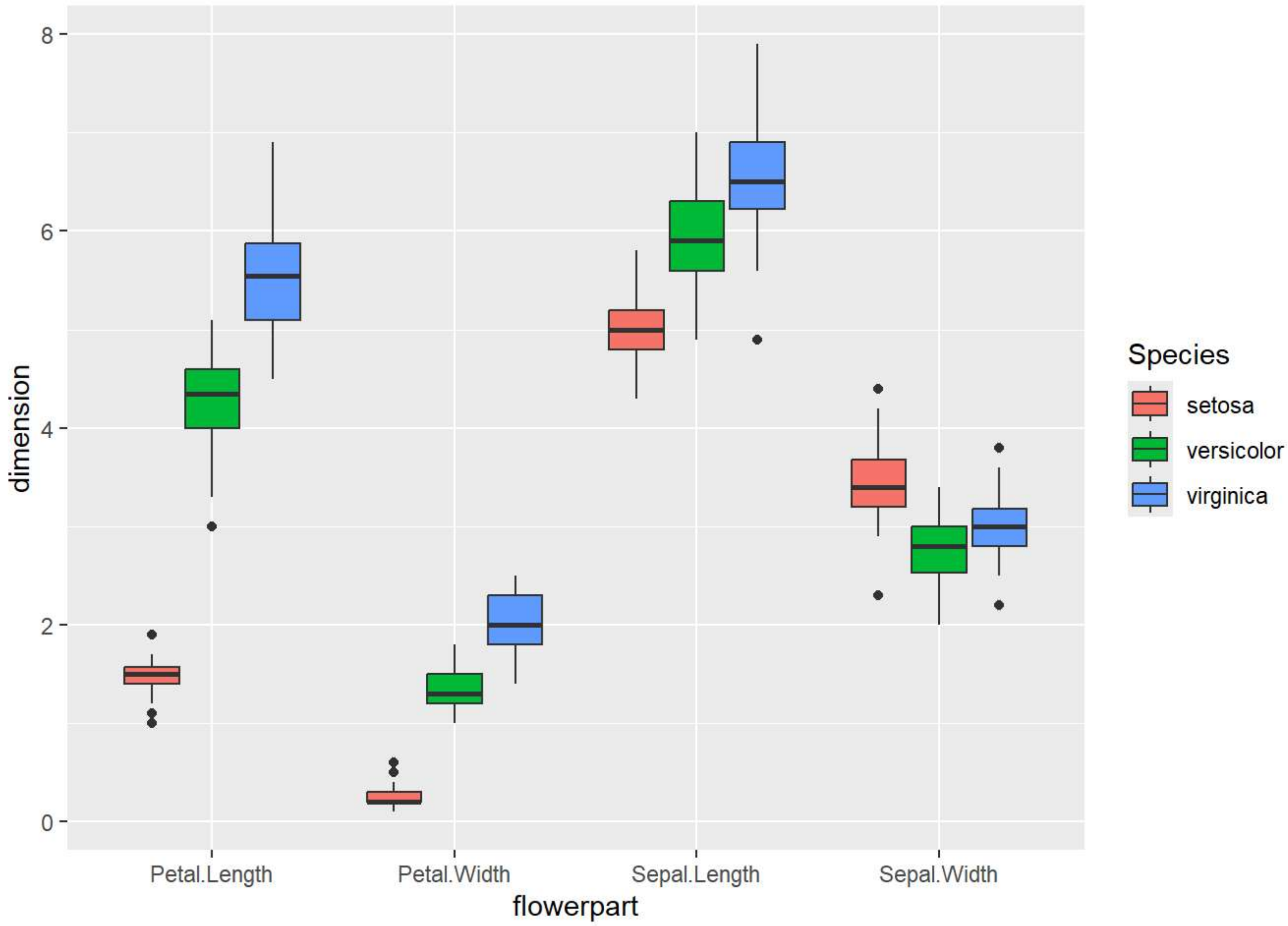
```
iris_df_long<-iris_df |> pivot_longer(!FlowerID & !Species,names_to="flowerpart",values_to="dime
nsion")

iris_df_long
```

```
## # A tibble: 600 × 4
##    Species FlowerID flowerpart   dimension
##    <fct>   <chr>    <chr>            <dbl>
##  1 setosa  1        Sepal.Length       5.1
##  2 setosa  1        Sepal.Width        3.5
##  3 setosa  1        Petal.Length       1.4
##  4 setosa  1        Petal.Width        0.2
##  5 setosa  2        Sepal.Length       4.9
##  6 setosa  2        Sepal.Width        3
##  7 setosa  2        Petal.Length       1.4
##  8 setosa  2        Petal.Width        0.2
##  9 setosa  3        Sepal.Length       4.7
## 10 setosa  3        Sepal.Width        3.2
## # ℹ 590 more rows
```

```
ggplot(
  iris_df_long,
  mapping=aes(x=Species,y=dimension,fill=flowerpart)
       ) + geom_boxplot()
```




```
ggplot(
  iris_df_long,
  mapping=aes(x=flowerpart,y=dimension,fill=Species)
      ) + geom_boxplot()
```



Correlation

We'll work with the Iris data set again

We want just one species, not all of them, we'll just select setosa

We want to start with the wide data frame

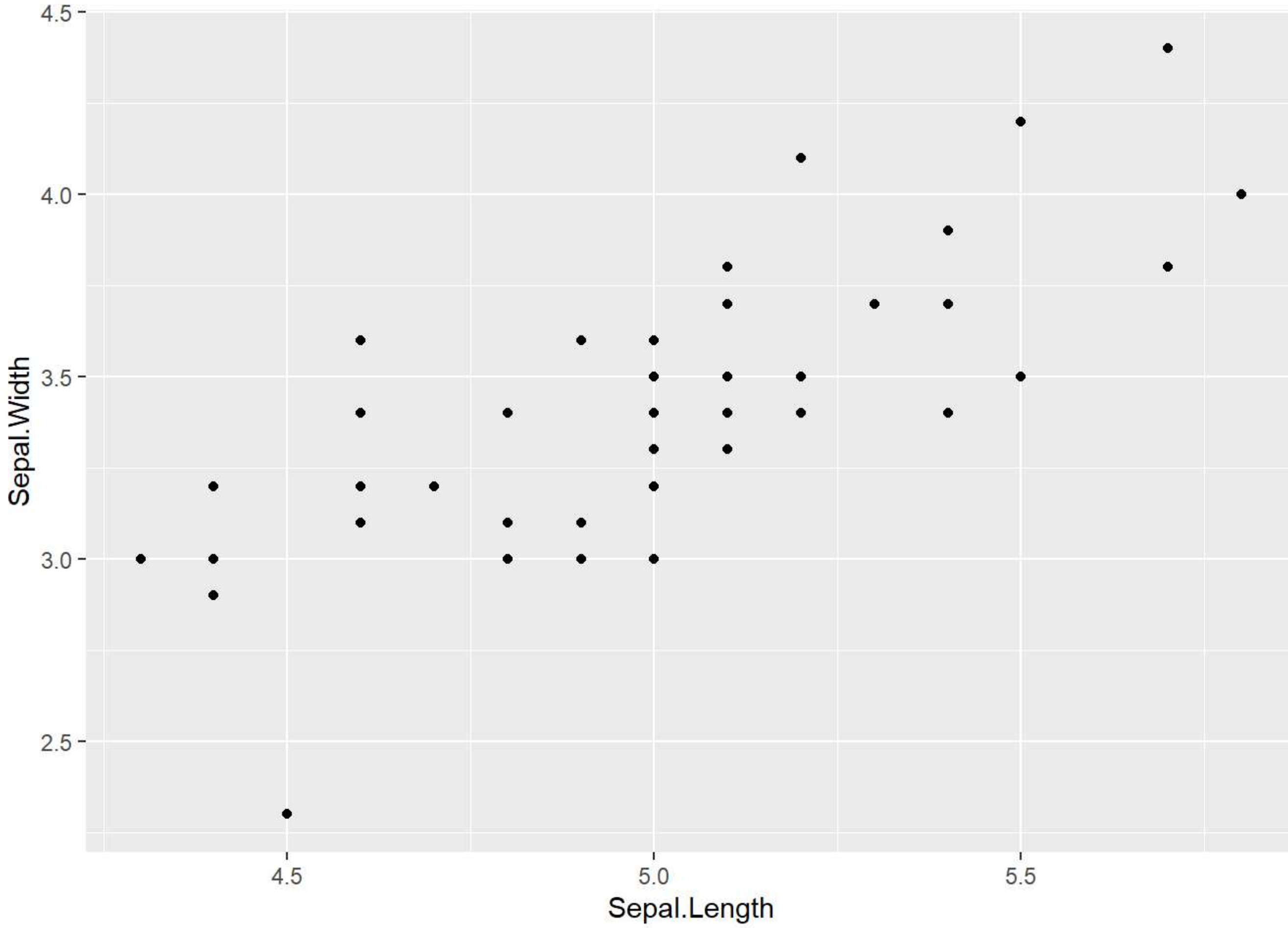
```
df_setosa= iris |> filter(Species=='setosa')

head(df_setosa)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Now let's look at correlation of Sepal.Length and Sepal.Width

Plotting first

```
library("ggplot2")
ggplot(df_setosa,aes(x=Sepal.Length,y=Sepal.Width))+geom_point()
```



There looks to be a trend (ie correlation) here, with quite a bit of noise

What is the correlation

```
cor(df_setosa$Sepal.Length,df_setosa$Sepal.Width)
```

```
## [1] 0.7425467
```

We have an R of 0.745, reasonably high but not extreme

We could look at the correlation of of Sepal length and width in all 3 species

```
iris |>group_by(Species) |>summarize(R=cor(Sepal.Length,Sepal.Width))
```

```
## # A tibble: 3 × 2
##   Species        R
##   <fct>      <dbl>
## 1 setosa     0.743
## 2 versicolor 0.526
## 3 virginica  0.457
```

#Looking at all Pairwise plots for the Setosa species data

The function ggpairs from GGally gives us a fast visual summary of the data

We get histograms of each variable, boxplots of each variable, biplots of each pair and the correlation of each pair

This is a handy tool for exploratory analysis, but is too much at once for presentations

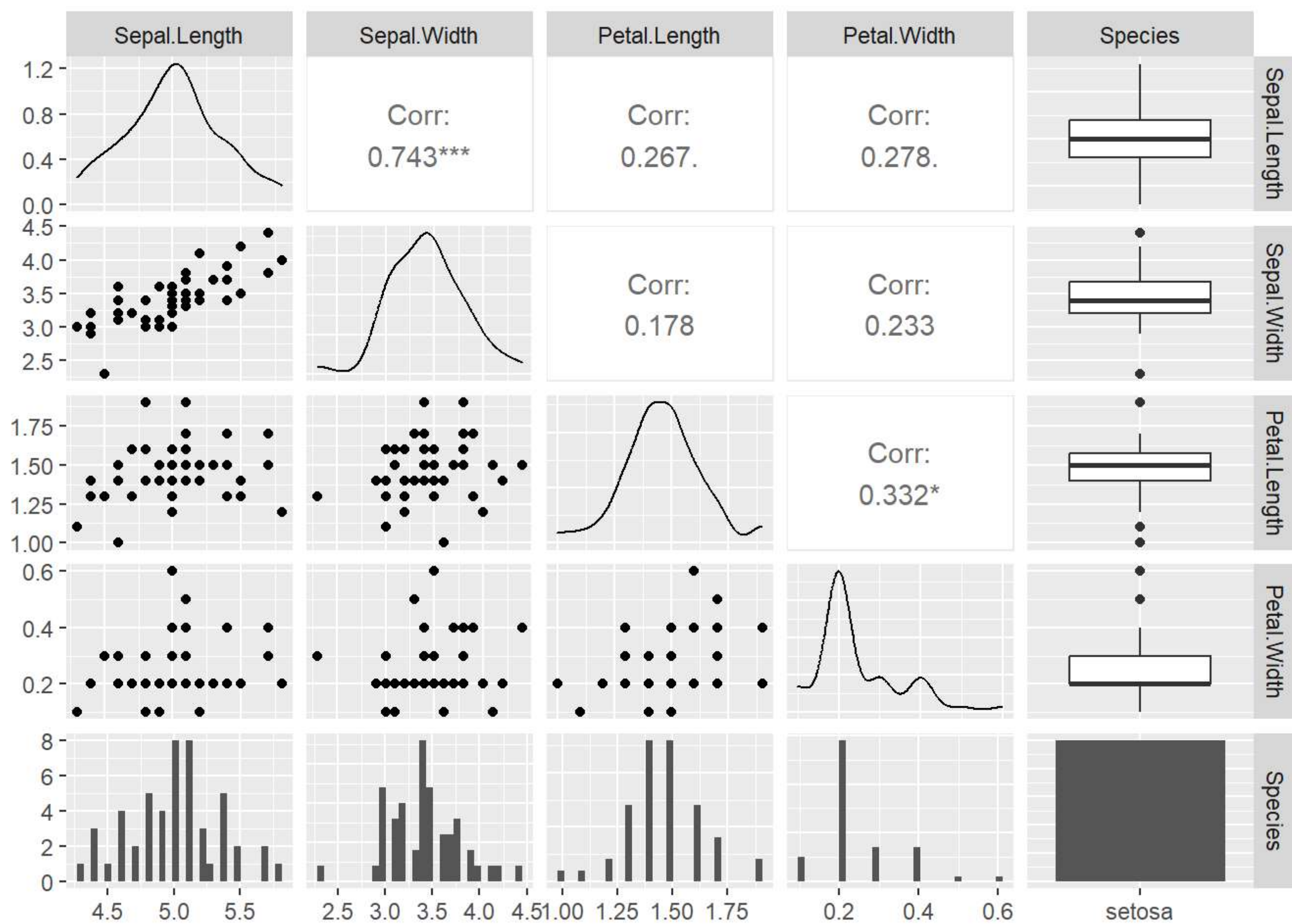
```
library('GGally')
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(df_setosa)
```

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

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



Question/Action

-Do all the distributions look normal/gaussian/bell curve? Explain why or why not

All but the petal width look normal. The petal width is deceiving because the resolution on the x-axis is very low, and one value had many instances, creating a false "bell curve".

-Which biplots look like they show a trend?

Sepal length vs. sepal width as well as petal length vs. petal width show some correlation.

-Which two correlations are the highest?

Sepal Length vs. Sepal Width and Petal Length vs Petal Width.

-Which two variables seem to have the most outliers?

Petal Length and Sepal Width. Sepal Width has a higher resolution along the x axis, so it likely has a higher quantity of outliers compared to petal width.

Question/Action

Load the mtcars built-in data set

```
data(mtcars)
```

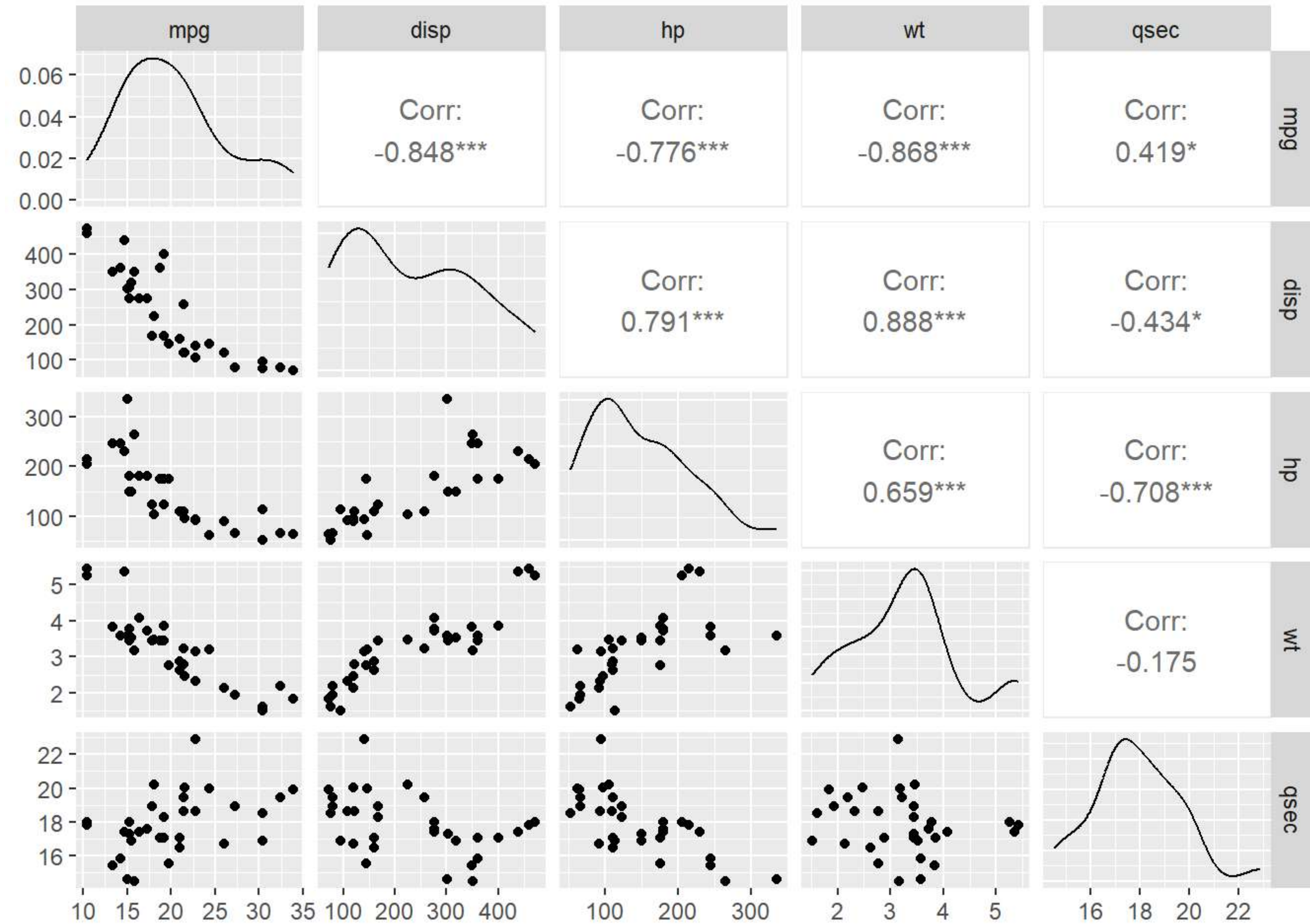
Select only mpg,disp, hp, wt and qsec from the data frame, call it mtcars_few

```
mtcars_few <- mtcars |> select(mpg, disp, hp, wt, qsec)
mtcars_few
```

```
##                      mpg  disp  hp    wt  qsec
## Mazda RX4           21.0 160.0 110 2.620 16.46
## Mazda RX4 Wag       21.0 160.0 110 2.875 17.02
## Datsun 710          22.8 108.0  93 2.320 18.61
## Hornet 4 Drive      21.4 258.0 110 3.215 19.44
## Hornet Sportabout   18.7 360.0 175 3.440 17.02
## Valiant             18.1 225.0 105 3.460 20.22
## Duster 360          14.3 360.0 245 3.570 15.84
## Merc 240D           24.4 146.7  62 3.190 20.00
## Merc 230            22.8 140.8  95 3.150 22.90
## Merc 280            19.2 167.6 123 3.440 18.30
## Merc 280C           17.8 167.6 123 3.440 18.90
## Merc 450SE          16.4 275.8 180 4.070 17.40
## Merc 450SL          17.3 275.8 180 3.730 17.60
## Merc 450SLC         15.2 275.8 180 3.780 18.00
## Cadillac Fleetwood  10.4 472.0 205 5.250 17.98
## Lincoln Continental 10.4 460.0 215 5.424 17.82
## Chrysler Imperial   14.7 440.0 230 5.345 17.42
## Fiat 128            32.4  78.7  66 2.200 19.47
## Honda Civic         30.4  75.7  52 1.615 18.52
## Toyota Corolla      33.9  71.1  65 1.835 19.90
## Toyota Corona       21.5 120.1  97 2.465 20.01
## Dodge Challenger    15.5 318.0 150 3.520 16.87
## AMC Javelin         15.2 304.0 150 3.435 17.30
## Camaro Z28          13.3 350.0 245 3.840 15.41
## Pontiac Firebird    19.2 400.0 175 3.845 17.05
## Fiat X1-9           27.3  79.0  66 1.935 18.90
## Porsche 914-2       26.0 120.3  91 2.140 16.70
## Lotus Europa        30.4  95.1 113 1.513 16.90
## Ford Pantera L      15.8 351.0 264 3.170 14.50
## Ferrari Dino        19.7 145.0 175 2.770 15.50
## Maserati Bora       15.0 301.0 335 3.570 14.60
## Volvo 142E          21.4 121.0 109 2.780 18.60
```

Create a ggpairs plot

```
ggpairs(mtcars_few)
```



-Which variables, if any, look normal?

mpg, qsec, and it could be argued wt as well.

-Which variables seem to have skew?

All variables have skew.

From the plots, which variables have positive correlation, which have negatitve? Do any appear to have little or no correlation?

Positive Correlation: mpg vs. qsec, disp vs. hp, disp vs. wt, and hp vs. wt

Negative COrrelation: mpg vs. disp, mpg vs. hp, mpg vs. wt, and hp vs. qsec

Little or No Correlation qsec vs. wt

-Which pair has the highest positive correlation?

disp vs. wt

-Which has the most extreme negative correlation?

wt vs. mpg

Question/Action

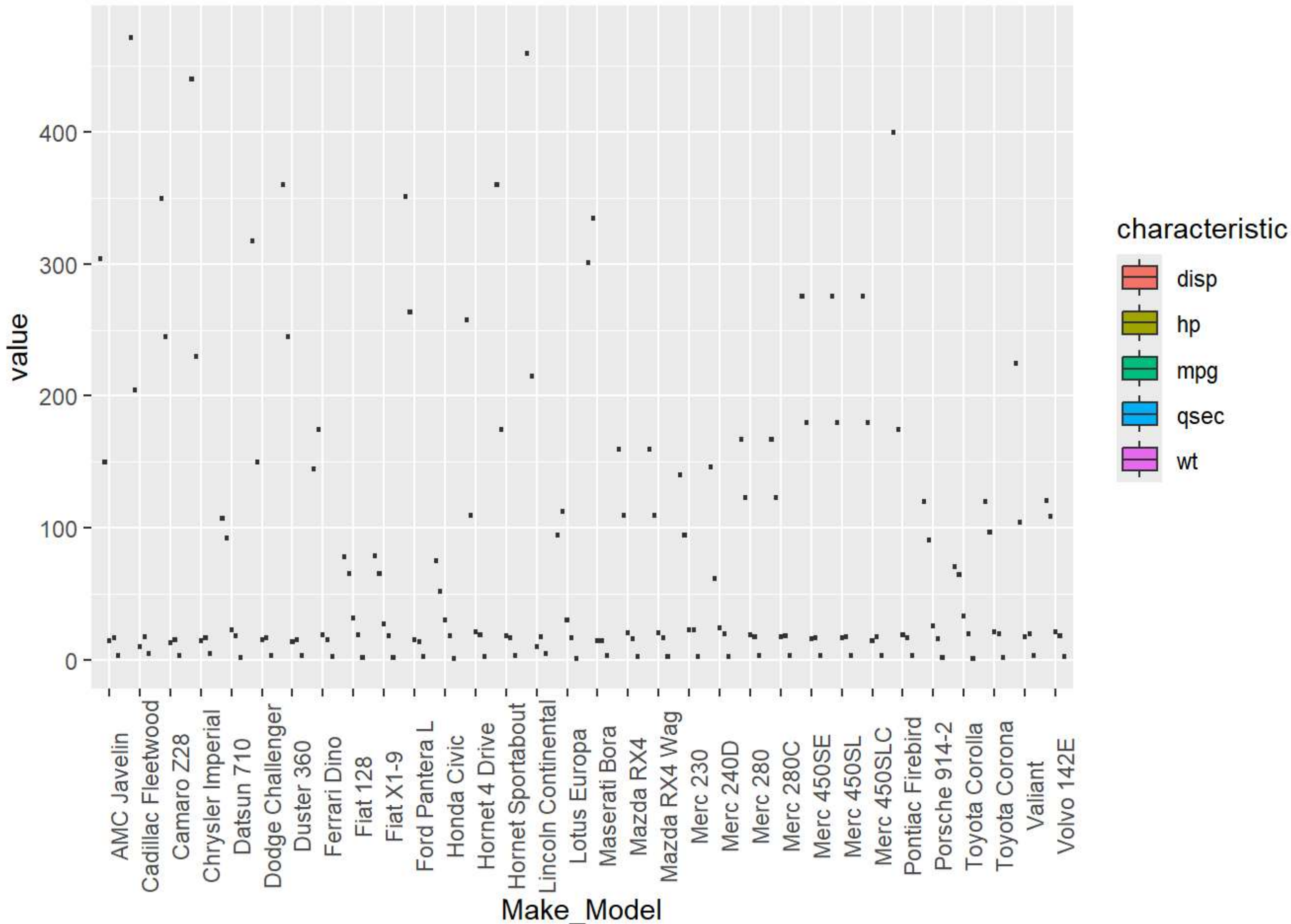
Convert mtcars_few to a long version

```
mtcars_few_long<-mtcars_few |>
  mutate(Make_Model=rownames(mtcars_few)) |>
  pivot_longer(!Make_Model,names_to="characteristic",values_to="value")

mtcars_few_long
```

```
## # A tibble: 160 × 3
##    Make_Model    characteristic  value
##    <chr>         <chr>           <dbl>
##  1 Mazda RX4     mpg             21
##  2 Mazda RX4     disp           160
##  3 Mazda RX4     hp             110
##  4 Mazda RX4     wt               2.62
##  5 Mazda RX4     qsec            16.5
##  6 Mazda RX4 Wag mpg             21
##  7 Mazda RX4 Wag disp           160
##  8 Mazda RX4 Wag hp             110
##  9 Mazda RX4 Wag wt               2.88
## 10 Mazda RX4 Wag qsec            17.0
## # ℹ 150 more rows
```

Create a boxplot that shows all 5 variables in one plot

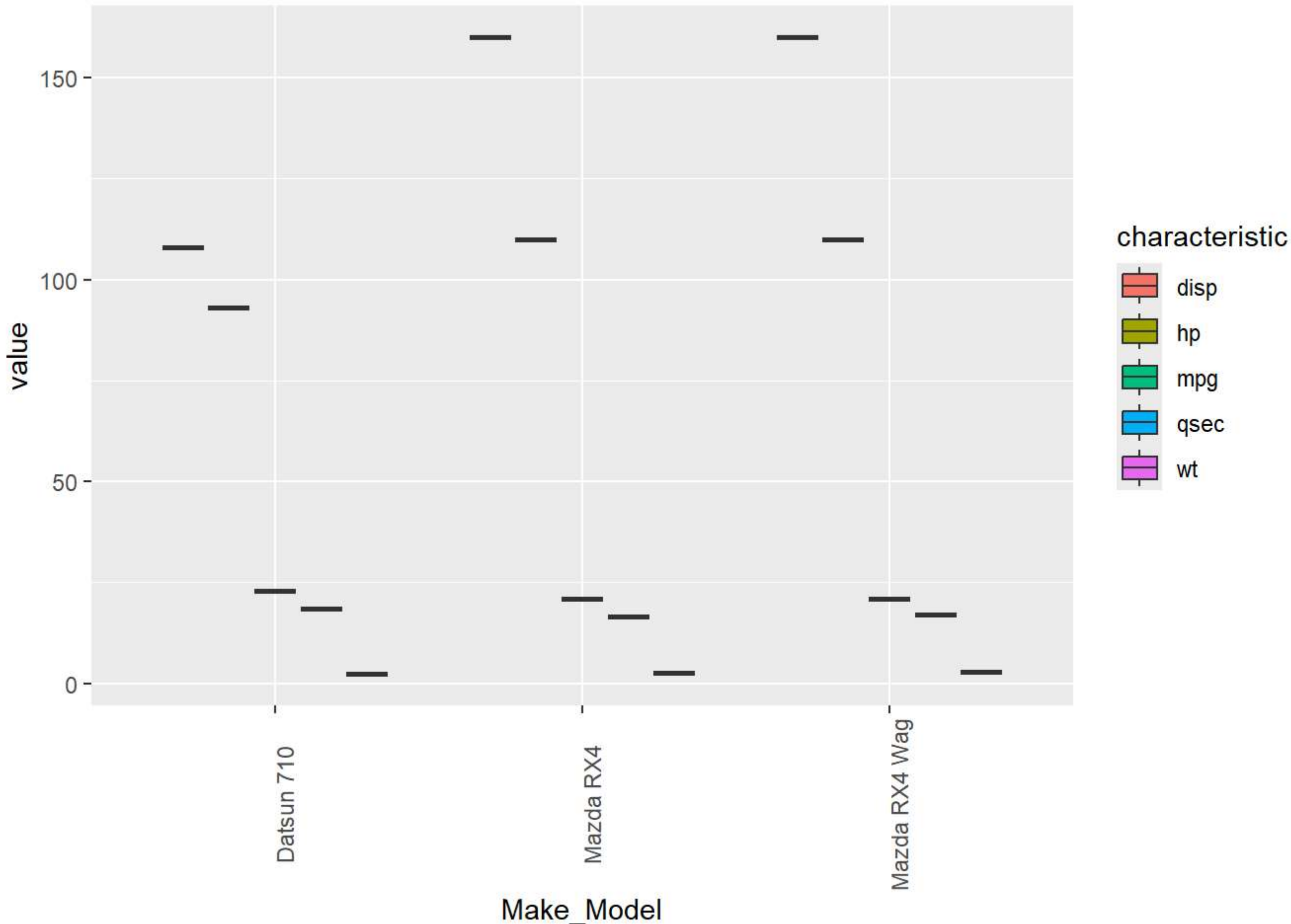
```
ggplot(
  mtcars_few_long,
  mapping=aes(x=Make_Model,y=value,fill=characteristic)
      ) + geom_boxplot() + theme(axis.text.x = element_text(angle = 90))
```



well… that certainly isn't ideal, let's select a smaller set of the data

```
mtcars_few_long<-head(mtcars_few_long,15)
```

```
ggplot(
  mtcars_few_long,
  mapping=aes(x=Make_Model,y=value,fill=characteristic)
      ) + geom_boxplot() + theme(axis.text.x = element_text(angle = 90))
```



I can’t really argue that this is any better, there is just too much variance in the values across the characteristics.