Introduction to R (With Pokemon!)

Devon Edwards Joseph - Data Analytics, Lloyds Banking Group - AI Club for Gender Minorities Co-Organiser -

10/10/2017

Getting Started: Installing The Software

Your new desk isn't too shabby, you've already found the coffee shop, and your co-workers seem a friendly bunch. Sweet! Day 1 at the Pokemon Company Data Science department is going well.

Your thoughts quickly turn to the analysis that the CEO is counting on you for.

But - before the magic happens, you've got some downloading to do...

Check out the links to get set up.

Downloand R: https://cran.r-project.org/

Download RStudio IDE: www.rstudio.com/products/rstudio/download/

Download Pokemon Dataset: www.kaggle.com/abcsds/pokemon/data

Installing R Packages

RStudio is about to be your new best pal!

RStudio is the interactive development environment that you will be using to write and run your R code, so go ahead and open it up.

Before you can get to work, you need to set-up your workspace, which will involve importing your data and installing the R packages that you will need for your analysis.

R packages are libraries of pre-written code that you can use to speed up and simplify your work.

1. To use them, you first need to click the 'Packages' tab in the bottom right-hand box of RStudio (See the RStudio handout if you aren't sure where to look!), and then click 'Install' and type in the names of the packages shown in parenthesis in the code below. eg. `dplyr`

In the code snippets, anything with a '#' in front of it is a comment and is ignored by R. These are just instructions to help you!

2.

```
# Once installed, we load packages with the library() function, as below

library(dplyr) # A useful data cleaning package
library(ggplot2) # A data visualisation package
library(tidytext) # Another data cleaning package
```

Importing The Pokemon Data

Now that you have some shiny new packages installed and loaded in RStudio, it is time to import the Pokemon data to get started.

- R can load in data from many different sources, but today you will be loading the 'CSV' spreadsheet you downloaded and saved earlier
- To do this, we use a function called `read.csv()` by telling it where we saved the data earlier. Take a look at the example below and see if you can work out how to tell R where your data is saved on your computer

```
read.csv("C:/ConferenceDataFolder/Pokemon.csv")
```

The code above will read the CSV file into RStudio for you, however in order to make the data easier to work with, it is a good idea to assign it as a data object to a variable.

A variable acts like a container that will hold the data you give it, so that you can use the variable again and again in your code with ease and save changes you make to it instead of reading the data in each time.

Take a look at the code below, where the Pokemon data is assigned to the variable `Poke.Data`:

```
Poke.Data <- read.csv("C:/ConferenceDataFolder/Pokemon.csv")
```

Use the example above to import your data, changing the `"C:/ConferenceDataFolder/Pokemon.csv"` to match the file path and file name you used when you saved your Pokemon data. Write this code underneath your package loading code from the previous section.

To run your code once written, you can highlight it and either press 'Run' in the top right-hand corner, or hit CTRL + ENTER on your keyboard.

Exploring The Data

The first step in any data science project is to get a good look at your data.

You need to get an idea of:

- The amount of data you have - eg. the number of rows ('observations'), and columns ('variables') it has - Can you spot these details in the output when you run the code below?

The output of the code will appear in the console, the box underneath the one you're typing your code in!

```
str(Poke.Data)
```

- The layout of your data, and what is actually recorded in it as 'variables'

```
head(Poke.Data)
```

- Some summary (or 'descriptive') statistics to help you get a sense of your data - Can you run the code below and read the output to find out what the average ('mean') speed of all the Pokemon in your dataset is?

```
summary(Poke.Data)
```

Have a go at writing and running the data exploration functions above, and see if you can get a clearer idea of the data you have to work with. Don't worry if some of it doesn't make sense yet!

Visually Exploring The Data

Statistics are great, but sometimes you just need to see to believe.

Basic data visualisations can be really useful to help you get a better impression of your data before you begin analysing it.

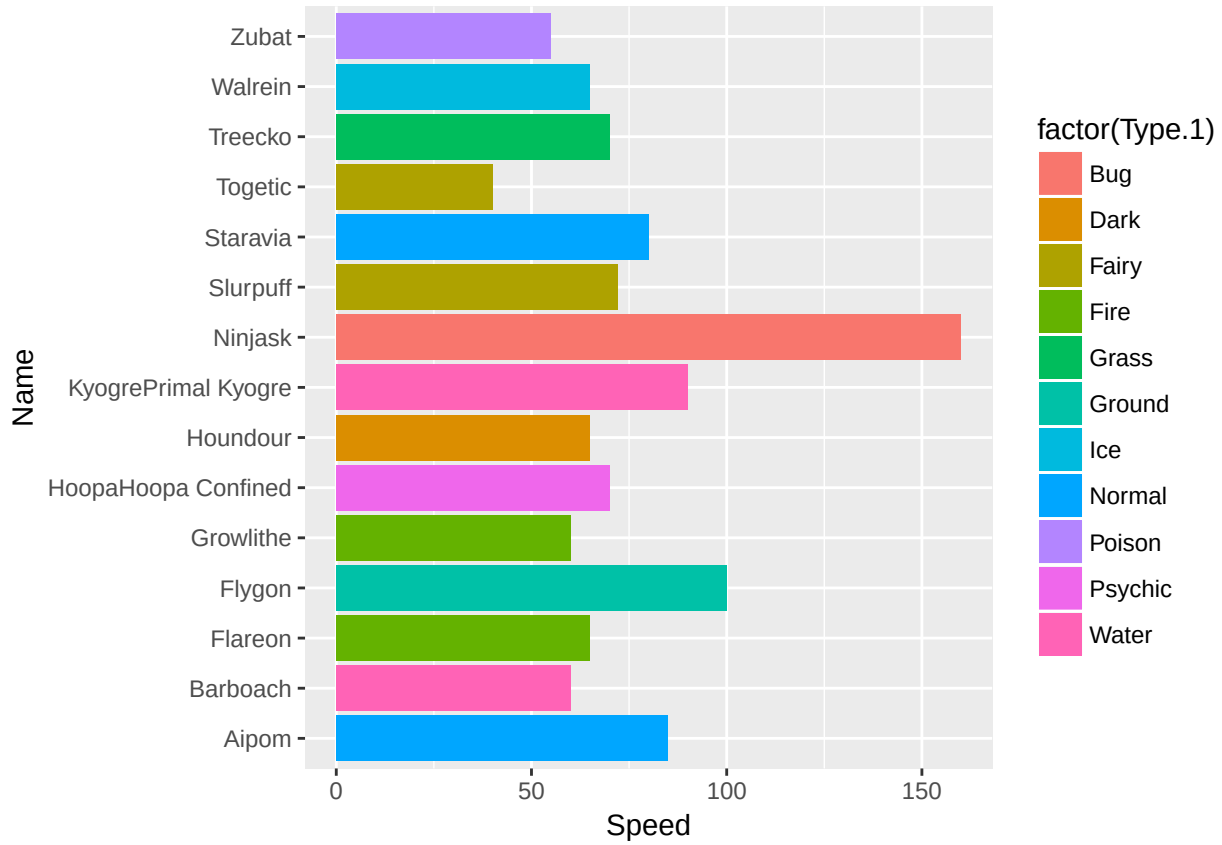
We are going to use the `ggplot2` package that you installed and loaded earlier. This code might look confusing for now, but don't worry - we're in this together!

Have a go at writing and running the code below.

Once you've written and run the code, click on the 'Plots' tab in RStudio to view your plots, and 'Zoom' to get a better look.

Your plot will look different to the one shown below, because we are taking a random sample of the dataset in the code.

```
# Take a random sample of 15 Pokemon from the dataset and add to a new variable
Poke.Data.Sample <- sample_n(Poke.Data, 15)

# Make a basic plot of each Pokemon's Speed and Type using the sample
ggplot(Poke.Data.Sample, aes(x = Name, y = Speed , fill = factor(Type.1))) +
  geom_col() +
  coord_flip()
```



Congratulations, you just made your first R code plot!

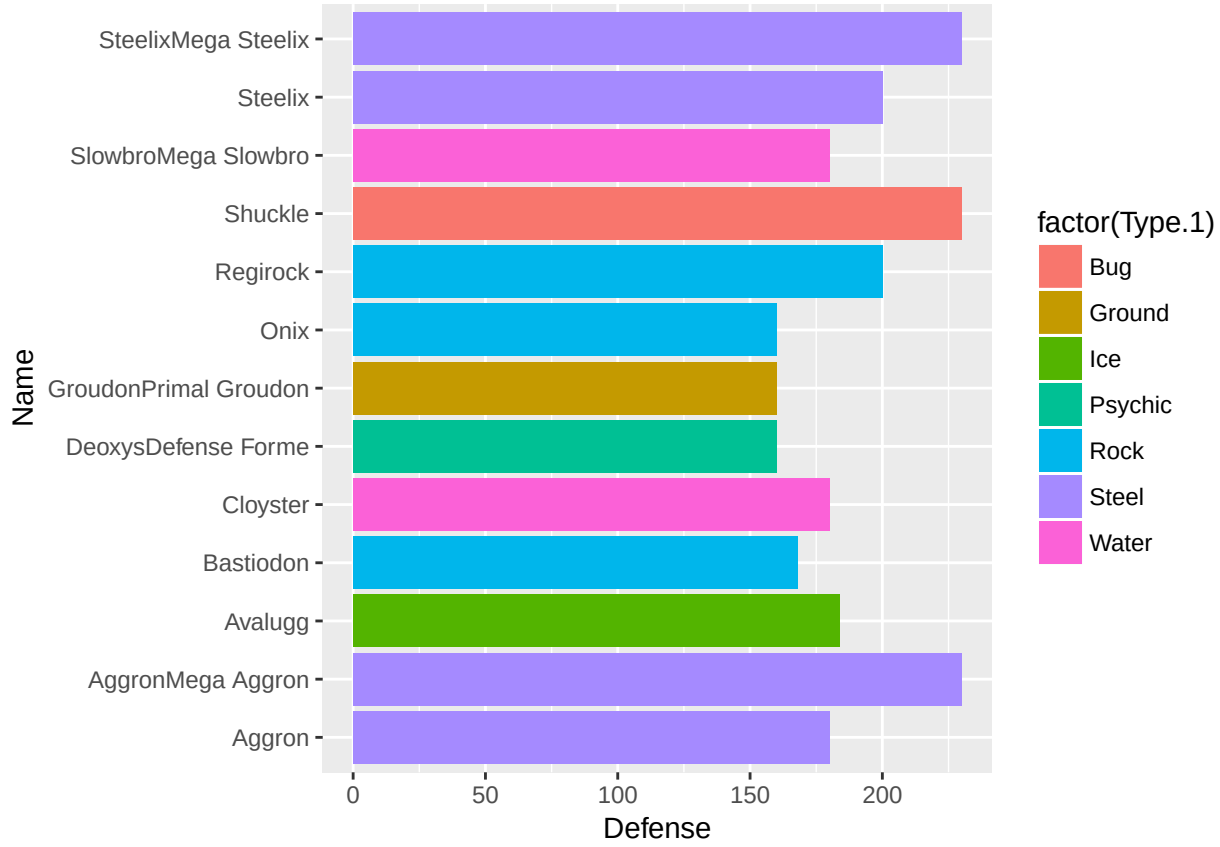
This plot is pretty, but we need to get a bit more specific if we want to get a handle on our data.

Let's answer some questions about the data using these basic plots.

See if you can understand what the code is doing in each of these examples, as you write and run them yourself.

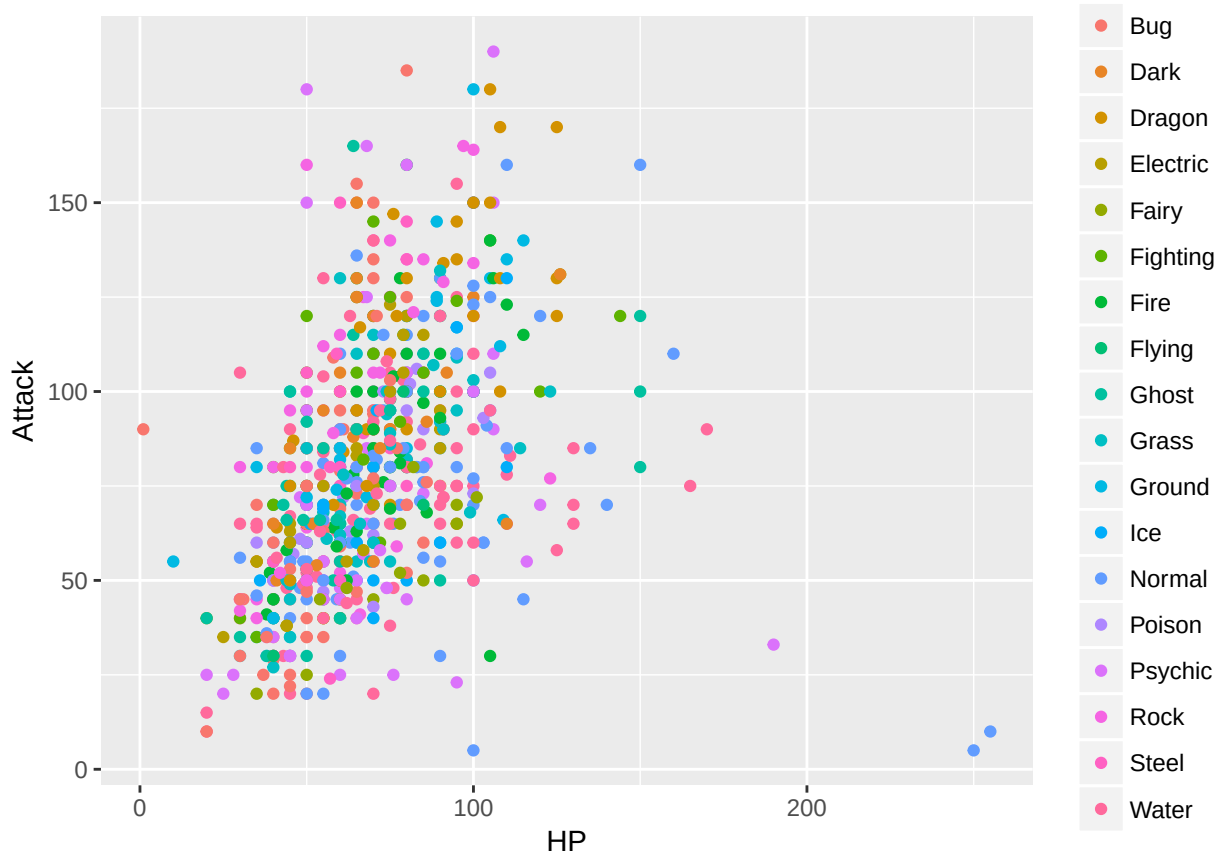
- Which `Type` of pokemon tend to have a `Defense` score of over 150?

```
Fast.Pokemon <- filter(Poke.Data, Defense > 150)

ggplot(Fast.Pokemon, aes(x = Name, y = Defense, fill = factor(Type.1))) +
  geom_col() +
  coord_flip()
```



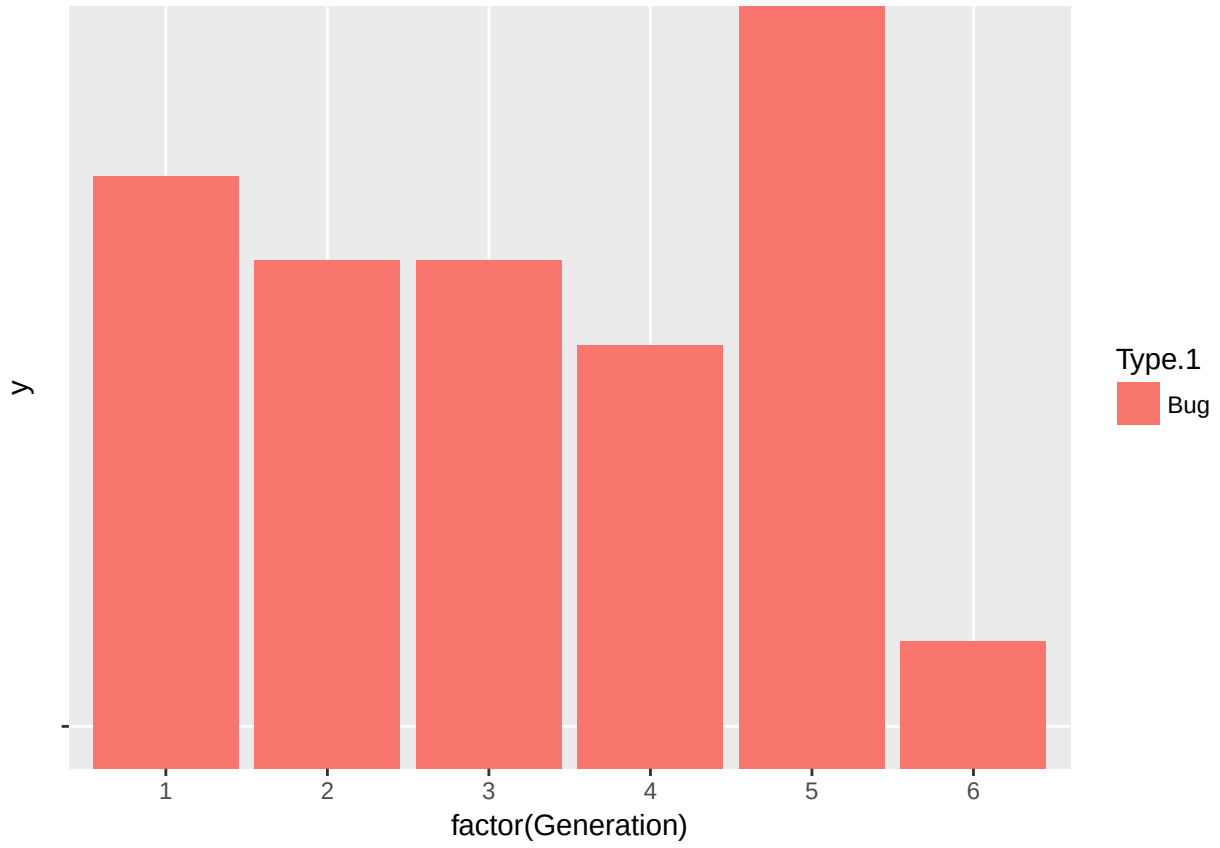
- Is there a correlation between `Attack` and `HP`?

```
qplot(HP, Attack, data = Poke.Data, color = Type.1)
```



- Which `Generation` do most of the Bug Pokemon belong to?

```
# Filter the data to only include Bug Type Pokemon
Bug.Pokemon <- filter(Poke.Data, Type.1 == "Bug")

ggplot(Bug.Pokemon, aes(x = factor(Generation), y = " ", fill = Type.1)) +
  geom_col()
```



The First Assignment!

By now you should have a much better idea of the data you're working with.

Duty calls, and the CEO is getting impatient, so you'd better get on with that analysis!

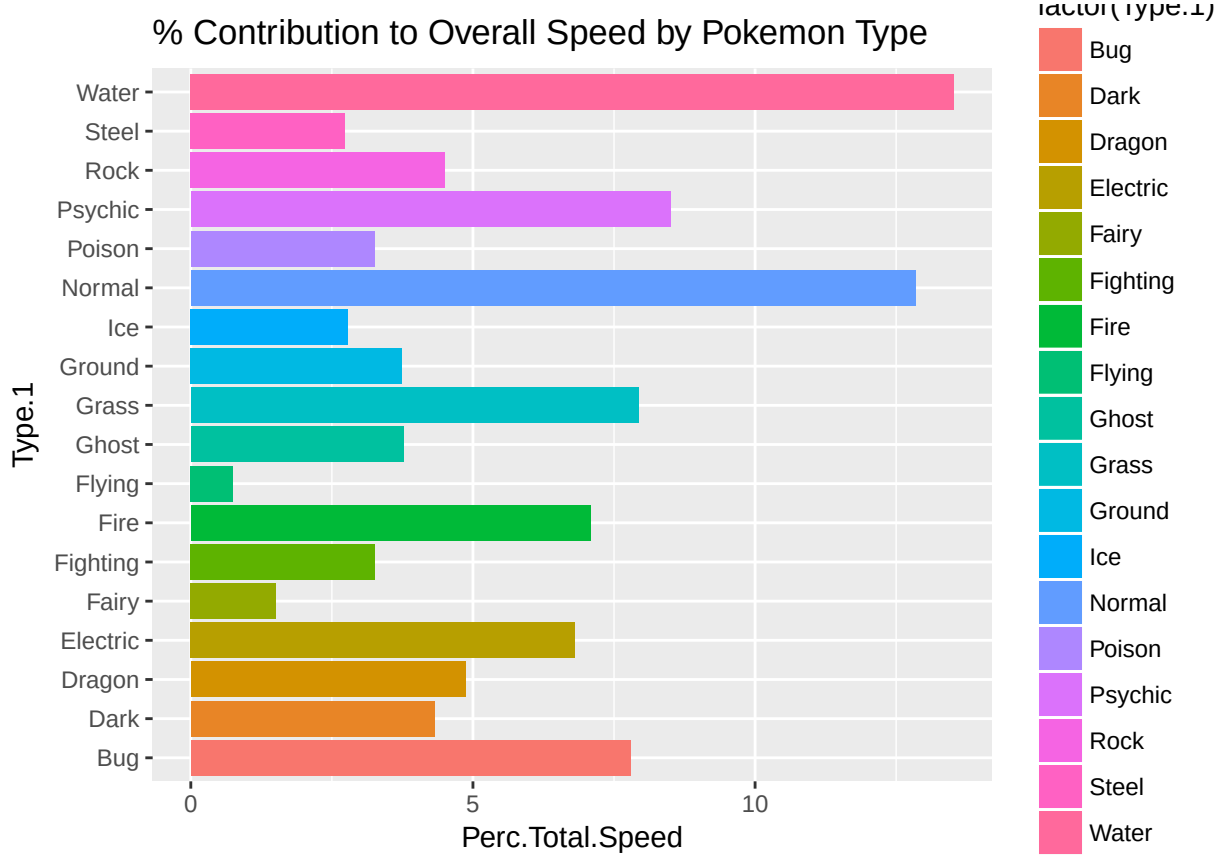
The CEO wants you to answer the following question first:

1. What contribution is each Pokemon `Type.1` making to the overall `Speed` profile of the Pokemon Company? Which Pokemon `Type.1` accounts for most of our `Speed`?

Can you work out what's going on in the code below?

```
# First, let's pick out the columns relating to Speed and Type
Speed.Type.Columns <- select(Poke.Data, Speed, Type.1)
print(Speed.Type.Columns)

# Secondly, we need to work out how to add up the Speed scores for each Type
# What do you think 'group_by()' and 'summarize()' are doing here?
# What do you think the '%>%' is doing here?
# Write and run the code first to help you work it out!
Add.Speed.Per.Type <- Speed.Type.Columns %>%
  group_by(Type.1) %>%
  summarize(Total.Speed = sum(Speed))
```

```r
print(Add.Speed.Per.Type)

# Thirdly, percentages would be much more useful than total counts of Speed
# seeing as the CEO is interested in the proportional contribution that each
# Pokemon Type makes to Speed - Let's convert to percentages
# What do you think 'mutate()' has done here?
Percentage.Contribution <- Add.Speed.Per.Type %>%
  mutate(Perc.Total.Speed = Total.Speed/sum(Total.Speed) * 100)

print(Percentage.Contribution)

# Now, which Type has the greatest % contribution to total speed?
# Order the Percentages in descending order and pick off the top one
Highest.Contribution <- Percentage.Contribution %>%
  arrange(desc(Perc.Total.Speed)) %>%
  top_n(1)

print(Highest.Contribution)

# Let's visualise the percentages of Speed contribution by Pokemon Type we calculated
ggplot(Percentage.Contribution, aes(x = Type.1, y = Perc.Total.Speed, fill = factor(Type.1))) +
  geom_col() +
  ggtitle('% Contribution to Overall Speed by Pokemon Type') +
  coord_flip()
```



Have a think about how you would make this chart better before sending it on to the CEO to review.

Assignment Number Two

Ok, now that you're all warmed up, things are about to get a little more tricky around here.

Panic! Your Data Science Assistant, Jim, comes crashing over, sending Pokeballs flying in all directions!

'Boss, Boss you have to help me!' Jim cries in dismay, 'I've accidentally deleted our `Attack` data for Articuno!'

In his attempts to take a look at Articuno's record, Jim ran the code below and accidentally deleted the `Attack` value instead of assigning it to a new variable!

Silly Jim.

Make sure that you run Jim's code below as well so that you can figure out how to correct his mistake!

```
Poke.Data[144, "Attack"] <- 0
```

Any mere mortal would panic at the thought of explaining this to the CEO.

But not you - you remember that earlier you found a correlation between `Attack` and `HP` and wonder if you can use this new found knowledge to predict Articuno's missing `Attack` score...

One way to solve this problem could be to use a linear regression model, so let's see if this is the answer.

```
# Take a look at line 144 (Articuno) to see for yourself that its Attack score has gone!
Poke.Data[144,]
```

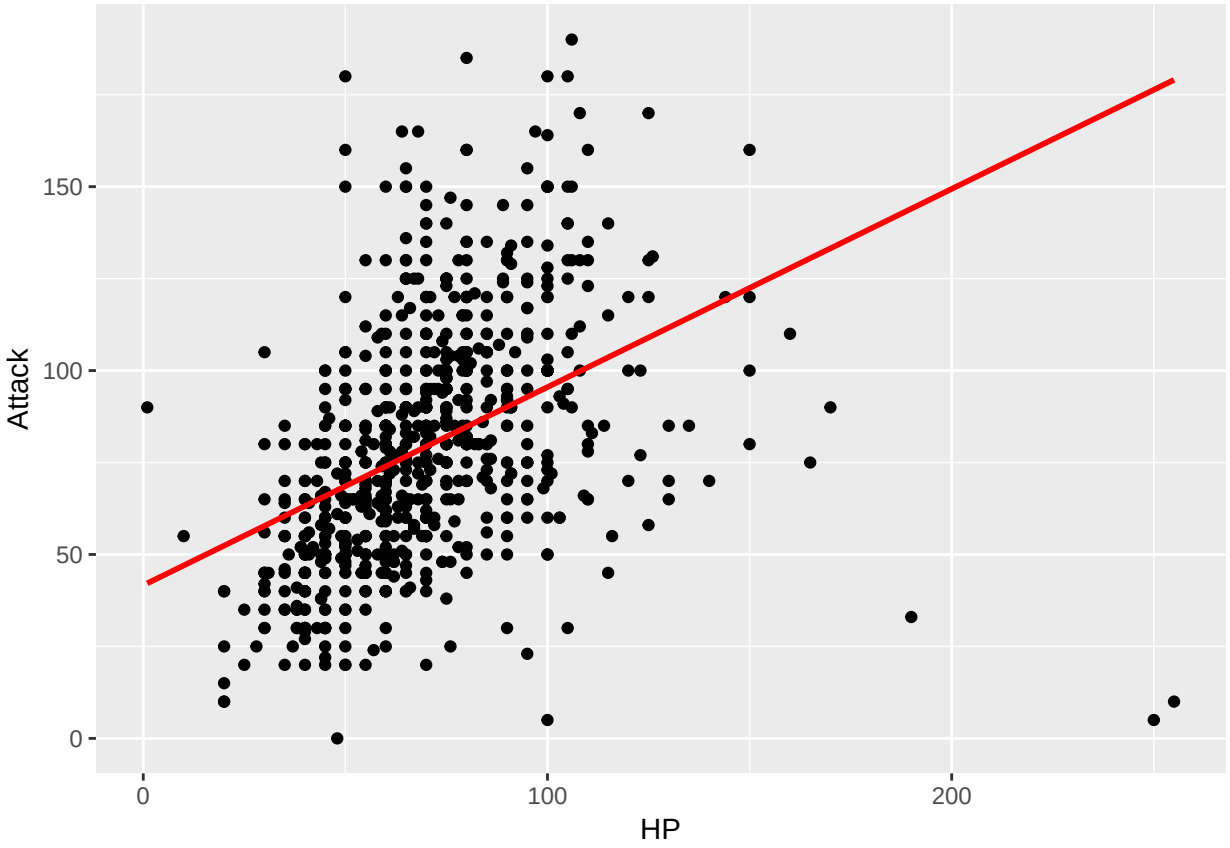
```
##      X.  Name Type.1 Type.2 Total HP Attack Defense Sp..Atk Sp..Def Speed
## 144 132 Ditto Normal          288 48      0      48      48      48    48
##     Generation Legendary
## 144          1     FALSE
```

```
# We know from our scatter plot earlier that there's some relationship between Pokemon HP & Attack
# But, to build linear regression models that are effective in prediction
# We need to be minful of the correlation coefficient, or how strong the correlation is
Poke.Data %>%
  summarize(corr.coeff = cor(HP, Attack))
```

```
##   corr.coeff
## 1  0.4226043
```

What does the result of this code tell you about the strength of the correlation? (Use Google to help you if you need it!)

Let's use some visualisation to help us out here.



Hmm, it was a good idea but it looks as though the correlation just isn't strong enough for you to build a predictive linear regression model.

So, on to option B!

Can you beg, borrow and steal from the code you've already written in this workshop, and that of those around you, to work out how you might work out what the average `Attack` score for a Pokemon of Articuno's `Type.1` would be, and then assign Articuno that value?

Psst, a hint for you - to assign a value to Articuno's `Attack` variable, you can use the code below:

```
Poke.Data[144, "Attack"] <- #insert your value here#

# Then, check that it's worked like this
Poke.Data[144, "Attack"]
```

Continuing Your Data Science Journey with R

I really hope that you've enjoyed the workshop and taking a brief peek into the world of R and some of the cool things that it allows you to do with data.

If you are keen to keep learning about R and data science, I can recommend the resources below:

- Join the 'AI Club for Gender Minorities' Meet Up group! As a Co-Organiser I am slightly biased, but definitely not wrong in saying that this is an incredibly friendly, supportive and knowledgeable group of women all either working in or towards careers in Data Science.

- There is also another fantastic group named 'R Ladies London' which sounds right up your street.
- DataCamp - https://www.datacamp.com/home DataCamp played a huge part in my process of self-teaching R, and I would recommend it as a useful introduction to many of the things you'd want to do with R and statistics, all with a useful in-browser IDE so that you can get used to writing code.
- Modern Dive Online Open Source R Textbook - http://moderndive.com/ A really accessible introduction to basic statistical concepts using R code.

Thank you for coming along to my workshop today. I would love to hear about how you found it, and how you get on with learning R, so please do feel free to connect with me on Linked In :) : https://www.linkedin.com/in/devonej/