One Data Science Programme Week 3

Introduction to statistical models

Nanaki Maitra

Learning outcomes:

- Recap: dataset
- Recap: data wrangling
- Recap: data visualisation
- Introduction simple linear regression models
- Building on data visualisation

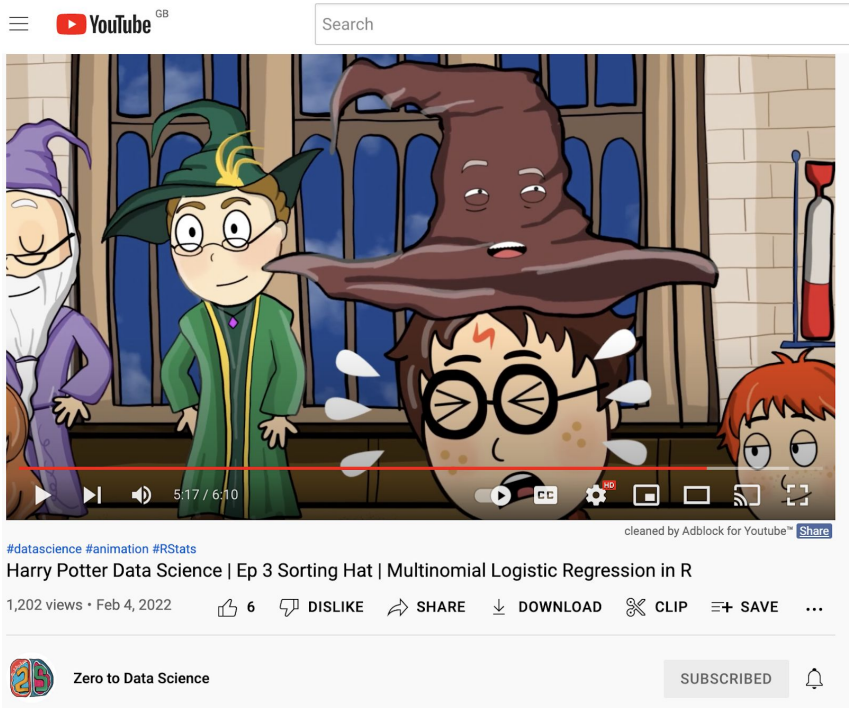
Important packages

```
> library(RCurl)       # http interface
> library(ggplot2)     # data visualisation
> library(dplyr)       # needed for the select() and summarise() functions
> library(magrittr)    # forward pipe operator %>%
```

Dataset

List of variables:

- House
- Name
- Birthday
- Best hand
- Arithmancy
- Muggle Studies
- Defence Against the Dark Arts



Task #1: Load dataset

```
data <- read.csv("https://raw.githubusercontent.com/kai-lim/One-Data-Science/main/data/Hogwarts_enrolment_data.csv")
```

https://raw.githubusercontent.com/kai-lim/One-Data-Science/main/data/Hogwarts_enrolment_data.csv

Recap: Data wrangling

Example:

Task #2: Filter by house and subjects

	Charms	Herbology
1	-246.4272	6.061064
2	-251.0625	-4.997610
3	-250.9119	-2.208650
4	-253.0216	-8.390447
5	-252.3844	-4.492272
6	-244.7478	-1.841579
7	-247.9081	-5.019345
8	-253.2316	-3.234020
9	-247.7641	-4.789794
10	-249.0736	-3.930535

Slytherin.data <- data %>% filter(Hogwarts.House == "Slytherin") %>% select(Charms, Herbology)

> summary(Slytherin.data)

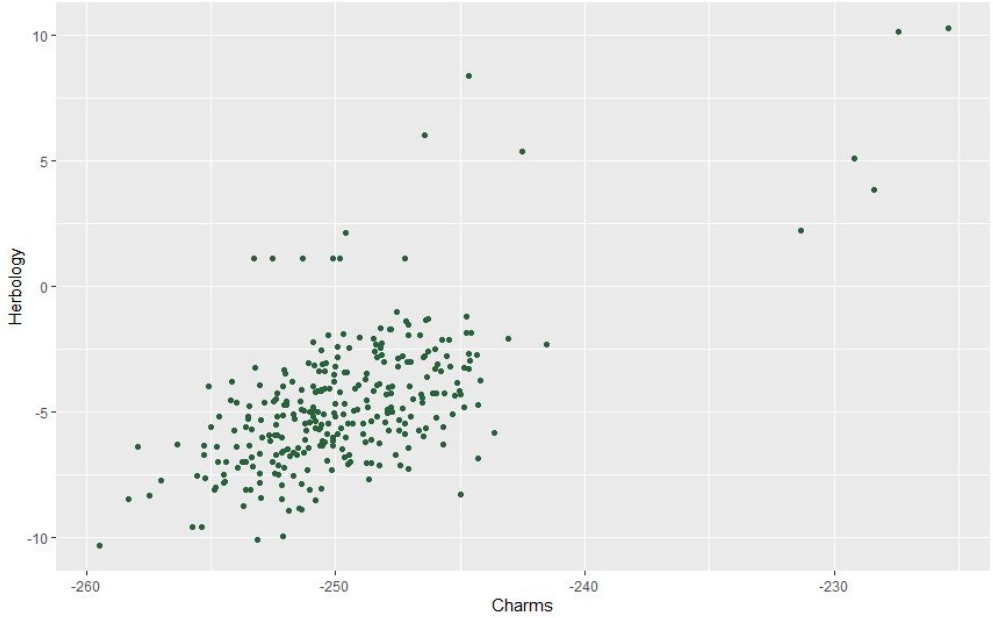
```
     Charms          Herbology      
 Min.   :-259.5   Min.   :-10.296  
 1st Qu.:-252.1   1st Qu.: -6.362  
 Median :-250.1   Median : -4.998  
 Mean   :-249.6   Mean   : -4.658  
 3rd Qu.:-247.5   3rd Qu.: -3.375  
 Max.   :-225.4   Max.   : 10.297  
```

Recap: Data visualisation

Example:

```
> ggplot(Slytherin.data, aes(x=Charms, y=Herbology)) + geom_point(colour = "#2a623d")
```

Using geom_point() from ggplot2 allows us to make a scatter plot



Introduction to simple linear regression

- Establish the relationship between two continuous variables
- Forecast a new observation

$y = mx + c$ ⟹ $Y = \beta_0 + \beta_1 x$

Y = observed values for dependent variable

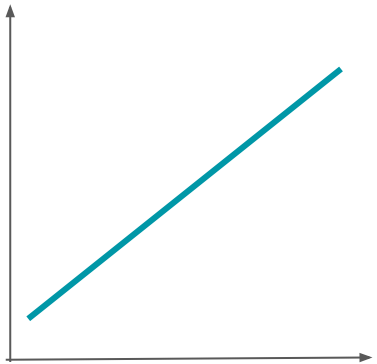
β_0 = y intercept

β_1 = gradient / slope

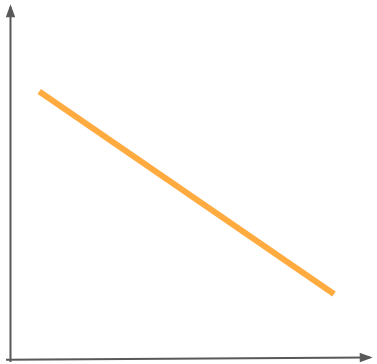
x = all observed values for independent variable

Introduction to simple linear regression

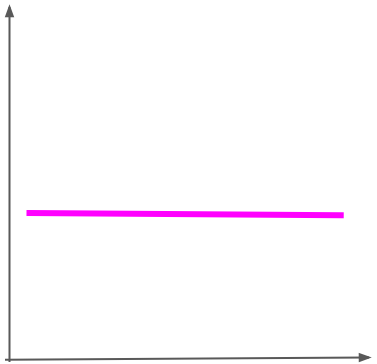
How do we interpret our results?



Positive relationship



Negative relationship



No relationship

Example

Task #3:

> slytherin.lm <- lm(formula = Herbology ~ Charms, data = Slytherin.data)

> summary(slytherin.lm)

```
Call:
lm(formula = Herbology ~ Charms, data = Slytherin.data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6678 -1.2214 -0.1992  1.0212 10.8520

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.30465    7.39505   14.51   <2e-16 ***
Charms        0.44858    0.02962   15.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.129 on 299 degrees of freedom
Multiple R-squared:  0.434,     Adjusted R-squared:  0.4321
F-statistic: 229.3 on 1 and 299 DF,  p-value: < 2.2e-16
```

Coefficients table:
Row 1 “Intercept” - this is the y intercept
Row 2 “Charms” - Here we have the slope of the equation and error

Y = 107.3 + 0.449x
OR
Herbology = 107.3 + 0.449*Charms

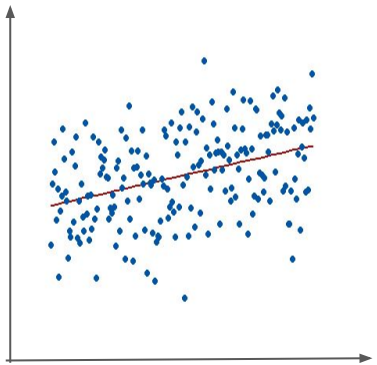
Introduction to simple linear regression

How good is our linear regression line?

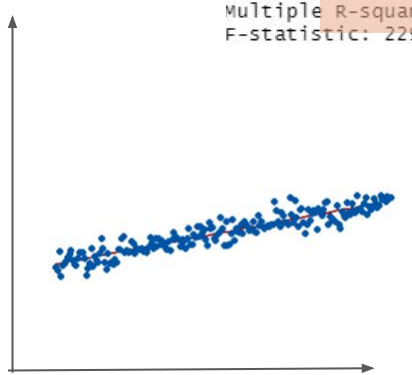
Let's look at the **r^2 value!**

r^2 is the coefficient of determination.

```
Call:
lm(formula = Herbology ~ Charms, data = Slytherin.data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6678 -1.2214 -0.1992  1.0212 10.8520

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.30465    7.39505   14.51   <2e-16 ***
Charms        0.44858    0.02962   15.14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.129 on 299 degrees of freedom
Multiple R-squared:  0.434,     Adjusted R-squared:  0.4321
F-statistic: 229.3 on 1 and 299 DF,  p-value: < 2.2e-16
```



Smaller r^2 value

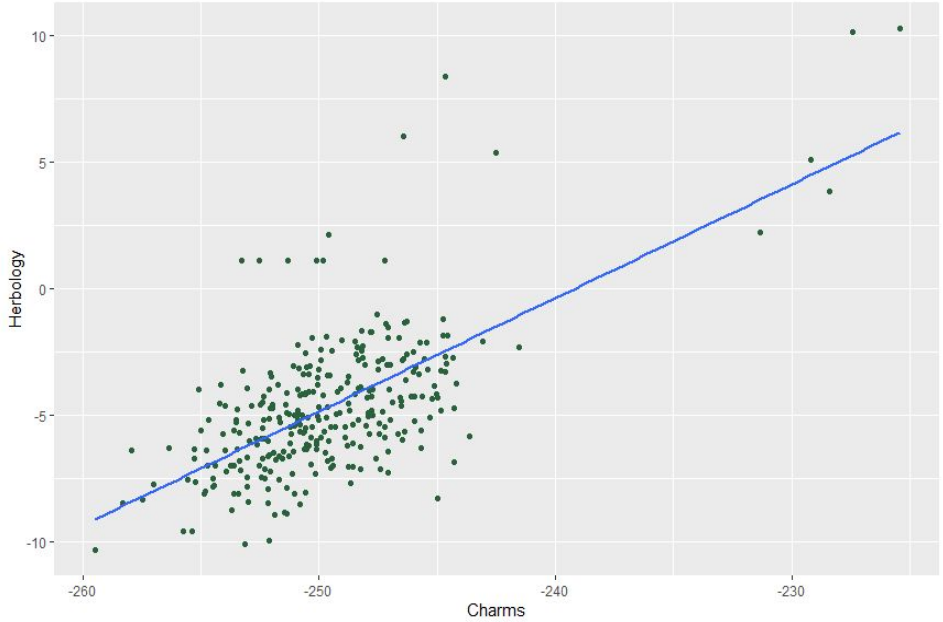


Larger r^2 value

Example

```
Call:
lm(formula = Herbology ~ Charms, data = Slytherin.data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6678 -1.2214 -0.1992  1.0212 10.8520

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.30465    7.39505   14.51   <2e-16 ***
Charms        0.44858    0.02962   15.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.129 on 299 degrees of freedom
Multiple R-squared:  0.434,     Adjusted R-squared:  0.4321
F-statistic: 229.3 on 1 and 299 DF,  p-value: < 2.2e-16
```

R-squared for our model is 43.3%



Task #4: Plot regression line

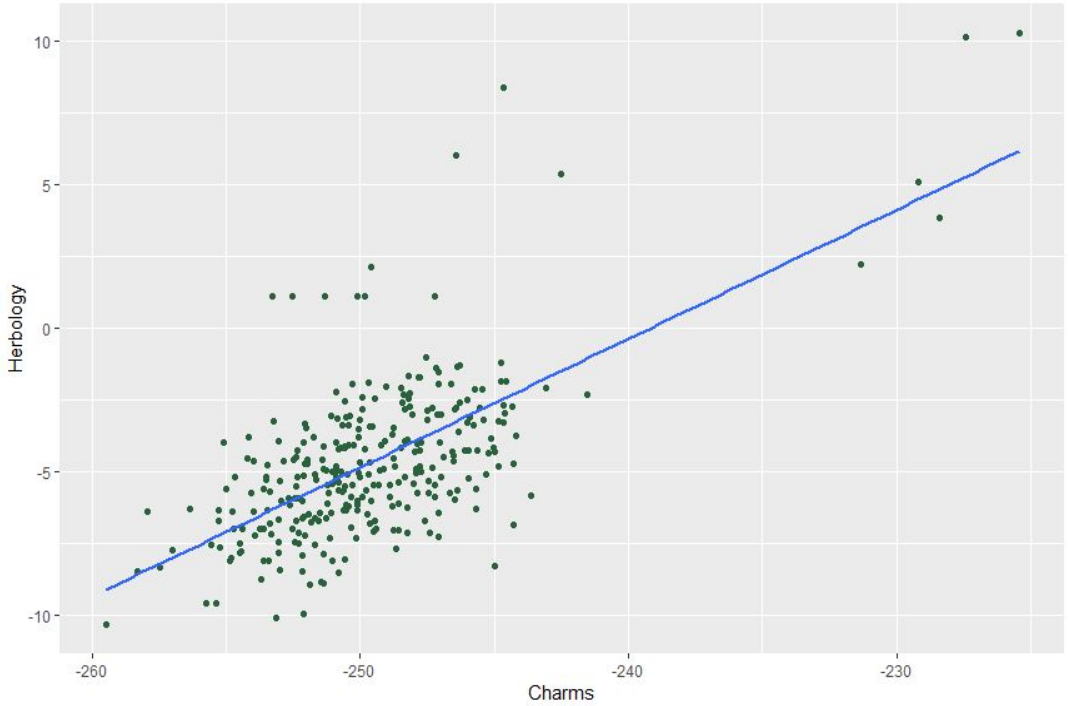
```
ggplot(Slytherin.data, aes(x=Charms, y=Herbology)) + geom_point(colour = "#2a623d") +
+ geom_smooth(method = lm, formula = y ~ x, se = FALSE)
```

Building on data visualisation

A good graph conveys all required information about the data to the viewer.

Therefore, presentation is key!

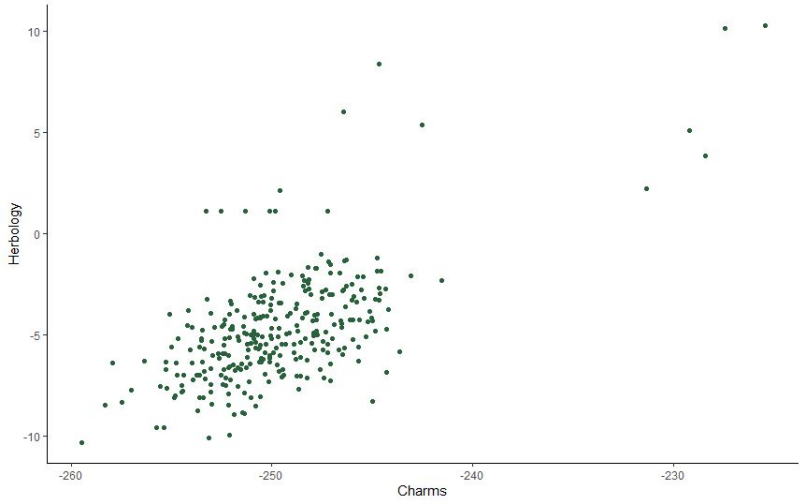
A lot of data presentation can come down to personal preferences but including somethings, like chart title and axis labels, is good practice:



Themes

ggplot(Slytherin.data, aes(x=Charms, y=Herbology)) + geom_point(colour = "#2a623d") +

theme_classic()



theme_light()

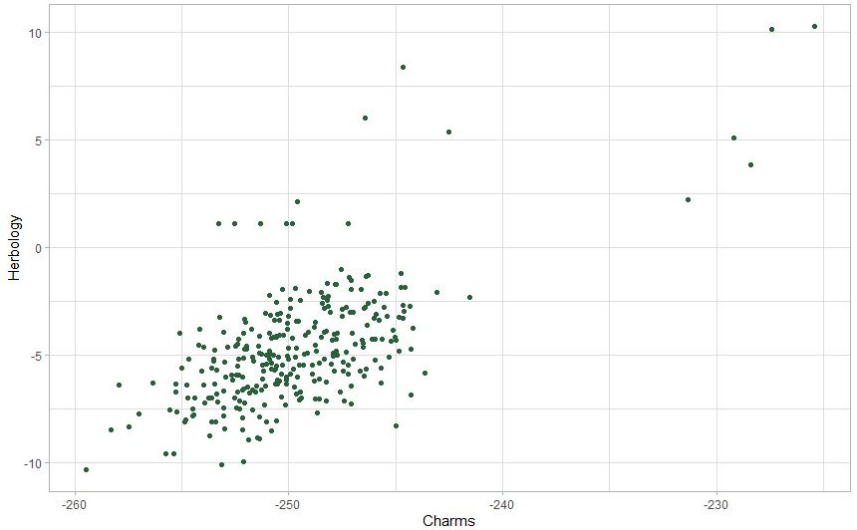
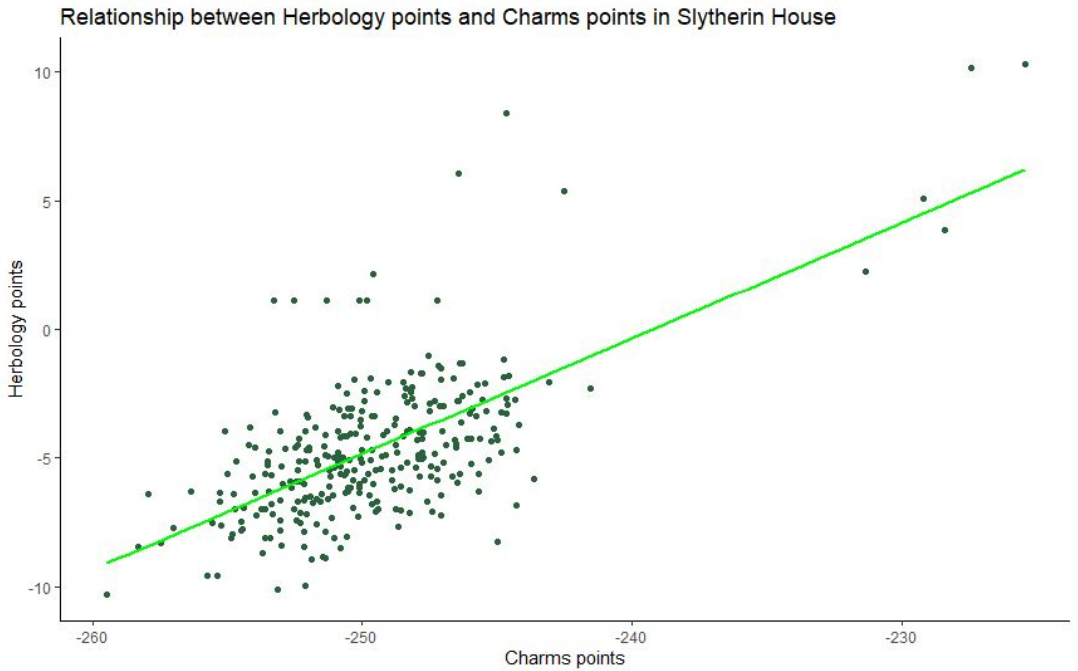


Chart and axis titles

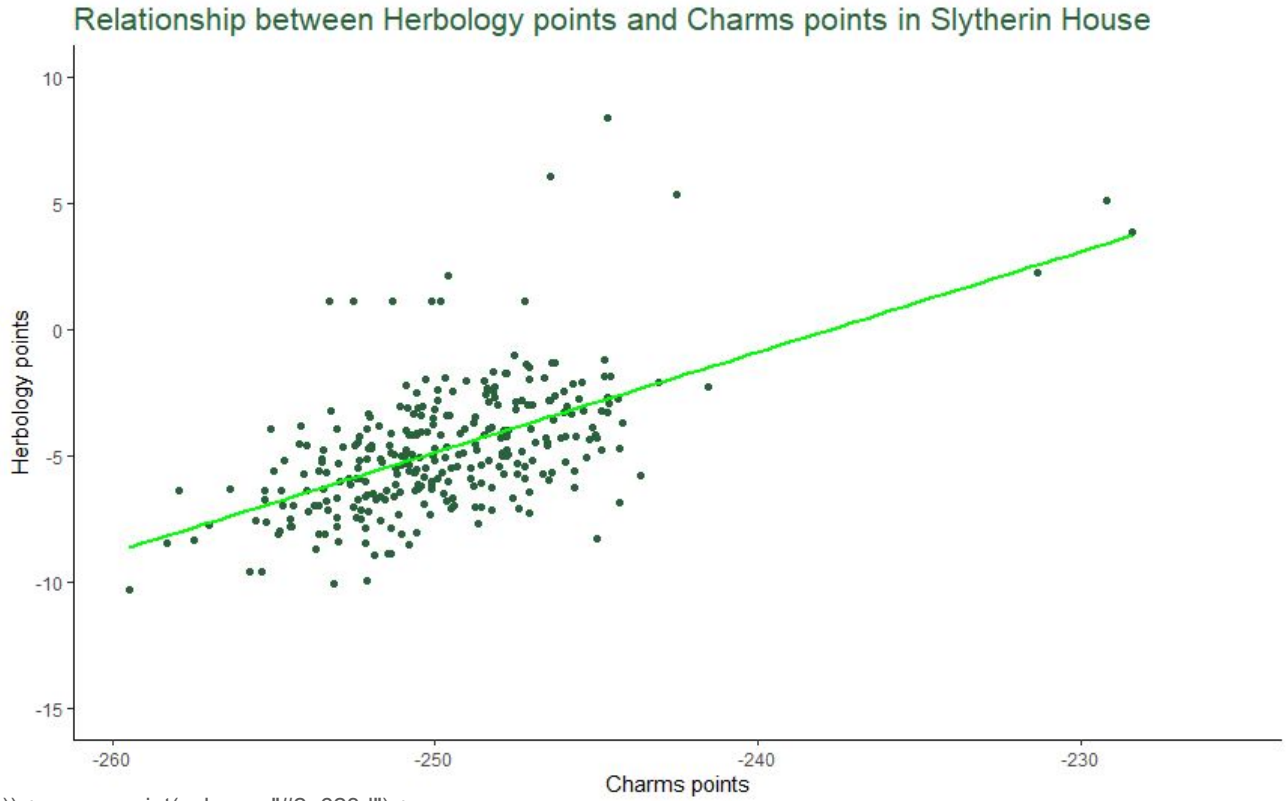
```
ggplot(Slytherin.data, aes(x=Charms, y=Herbology))
+ geom_point(colour = "#2a623d") +

+ theme_classic() +

+ ggtitle("Relationship between Herbology points
and Charms points in Slytherin House") +

+ xlab("Charms points") + ylab("Herbology points") +

+ geom_smooth(method = lm, formula = y ~ x, se =
FALSE, colour = "green")
```



Changing scales

xlim(min value, max value)

ylim(min value, max value)

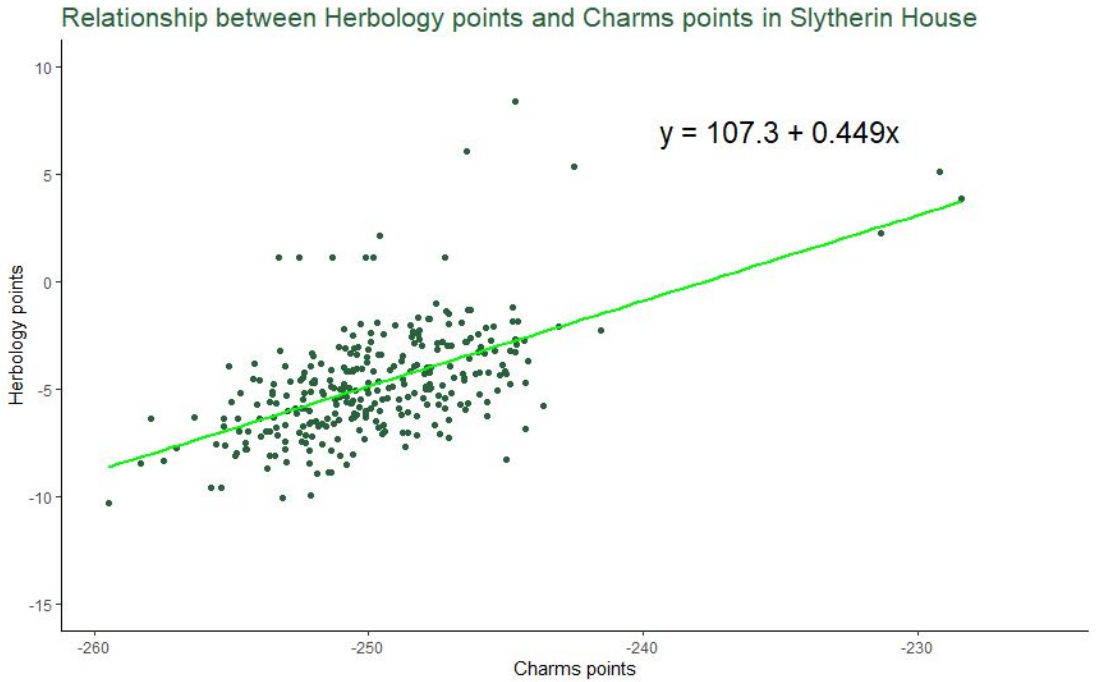



```
> ggplot(Slytherin.data, aes(x=Charms, y=Herbology)) + geom_point(colour = "#2a623d") +
+ geom_smooth(method = lm, formula = y ~ x, se = FALSE, colour = "green") +
+ xlab("Charms points") + ylab("Herbology points") +
+ ggtitle("Relationship between Herbology points and Charms points in Slytherin House") +
+ theme_classic() +
+ theme(plot.title = element_text(colour = "#2a623d", size = 16)) +
+ ylim(-15, 10)
```

Adding annotations on the graph

+annotate(geom = “text”, x axis position, y axis position, colour = , size = , label = “ ”)

Useful if you want to add the regression equation or the r^2 value



Hogwarts Colors

Griffindor	Slytherin	Ravenclaw	Hufflepuff
Griffindor Dark Red #740001	Slytherin Dark Green #1a472a	Ravenclaw Dark Blue #0e1a40	Hufflepuff Canary #ecb939
Gryffindor Red #ae0001	Slytherin Green #2a623d	Ravenclaw Blue #222f5b	Hufflepuff Light Canary #f0c75e
Griffindor Yellow #eeba30	Slytherin Dark Silver #5d5d5d	The Grey Lady #bebebe	Hufflepuff Light Brown #726255
Griffindor Gold #d3a625	Slytherin Light Silver #aaaaaa	Ravenclaw Gold #946b2d	Hufflepuff Dark Brown #372e29

Introducing Task

Send us an email if you have any questions:

one.data.science.program@gmail.com