

# Data visualisation I

O Rodriguez de Rivera Ortega, PhD  
SE@K (Statistical Ecology @ Kent), University of Kent

null



Introduction

Introduction  
to ggplot

Now it is  
your turn!  
With the  
data  
available  
create the  
most  
*horrible*  
*plot* using  
ggplot

# Introduction

# Introduction I

## Introduction

### Introduction to ggplot

Now it is  
your turn!  
With the  
data  
available  
create the  
most  
*horrible*  
*plot* using  
ggplot

- Data visualization is part art and part science. The challenge is to get the art right without getting the science wrong, and vice-versa.
- A data visualization first and foremost has to accurately convey the data.
- At the same time, a data visualization should be aesthetically pleasing.  
Good visual presentations tend to enhance the message of the visualization.

## Introduction II

To provide a simple visual guideline of which examples should be emulated and which should be avoided, I am labeling problematic figures as “ugly,” “bad,” or “wrong”:

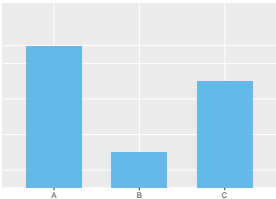
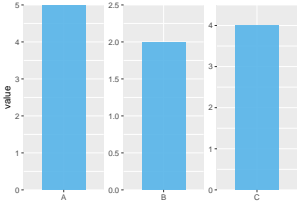
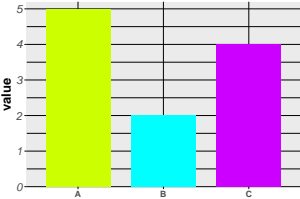
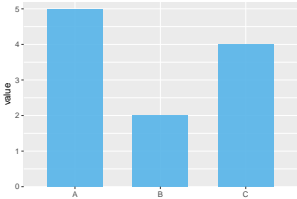
- Ugly - A figure that has aesthetic problems but otherwise is clear and informative
- Bad - A figure that has problems related to perception; it may be unclear, confusing, overly complicated, or deceiving
- Wrong - A figure that has problems related to mathematics; it is objectively incorrect

Introduction

Introduction  
to ggplot

Now it is  
your turn!  
With the  
data  
available  
create the  
most  
*horrible*  
plot using  
ggplot

Introduction-Example



# Introduction to ggplot

# A simple example

- The functions in the ggplot2 package build up a graph in layers.
- We'll build a a complex graph by starting with a simple graph and adding additional elements, one at a time. # #

We are going to download data from the III Forest Inventory (Murcia)

```
##load data
library(curl)
FI_Murcia <- read.csv(curl("https://raw.githubusercontent.com/orrone"))
attach(FI_Murcia)
names(FI_Murcia)
```

```
## [1] "Species"      "Shape"        "Height"       "Diameter1"    "Diameter2"
```

```
##specify dataset and mapping  
library(ggplot2)  
ggplot(data = FI_Murcia,  
        mapping = aes(x = Diameter1, y = Height))
```

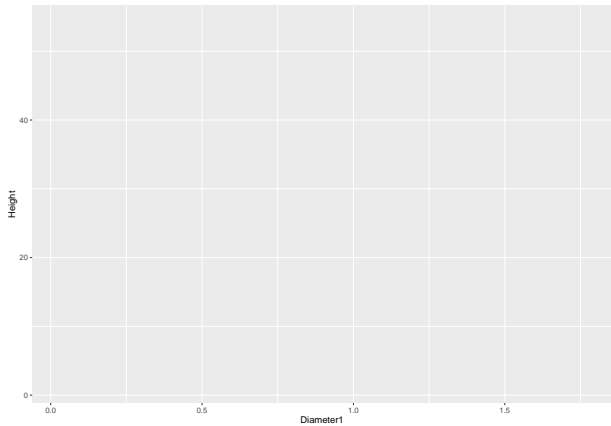


Figure 1: Map variables

We need to specify what we wanted placed on the graph



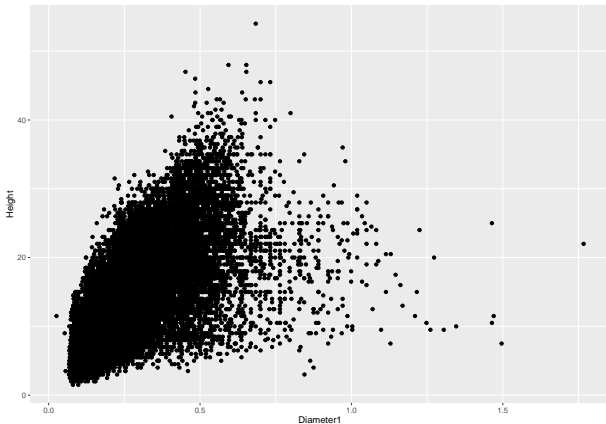
```
##add geom_point
```

```
library(ggplot2)
```

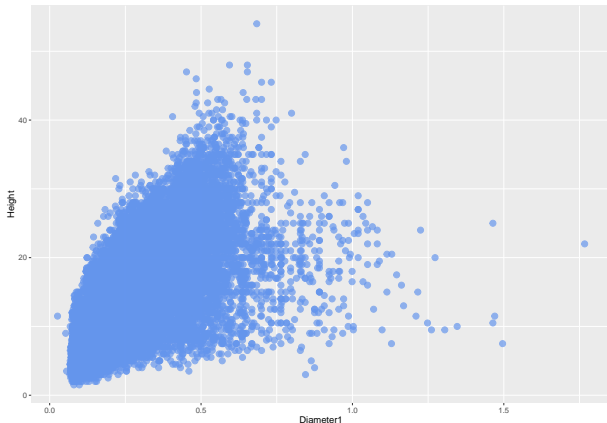
```
ggplot(data = FI_Murcia,
```

```
  mapping = aes(x = Diameter1, y = Height)) +
```

```
  geom_point()
```

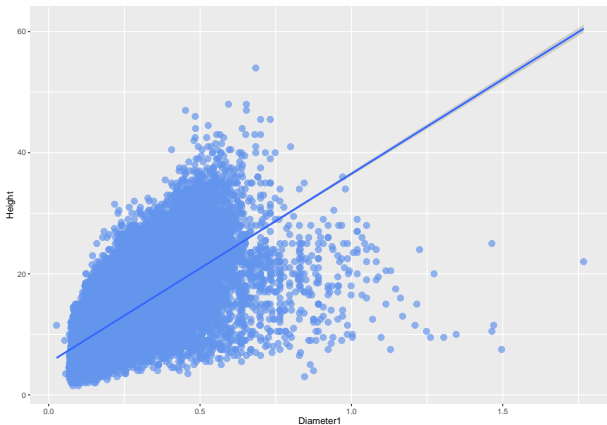


```
##make points blue, larger, and semi-transparent  
ggplot(data = FI_Murcia,  
mapping = aes(x = Diameter1, y = Height)) +  
  geom_point(color = "cornflowerblue", alpha = .7,  
    size = 3)
```



```
##add a line of best fit
```

```
ggplot(data = FI_Murcia,  
mapping = aes(x = Diameter1, y = Height)) +  
  geom_point(color = "cornflowerblue", alpha = .7, size = 3) +  
  geom_smooth(method = "lm")
```



```
##indicate species using color
FI_Murcia_sub = subset(FI_Murcia, Species == c("Castanea sativa", "Fagus sylvatica", "Pinus nigra", "Quercus robur"))
ggplot(data = FI_Murcia_sub,
mapping = aes(x = Diameter1, y = Height, color = Species)) +
  geom_point(alpha = .7, size = 3) +
  geom_smooth(method = "lm", se = FALSE, size = 1.5)
```



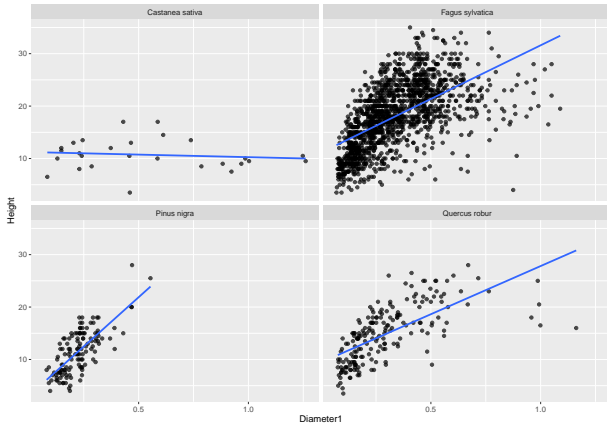
```
# reproduce plot for each species
```

```
ggplot(data = FI_Murcia_sub,  
mapping = aes(x = Diameter1, y = Height)) +  
  geom_point(alpha = .7) +  
  geom_smooth(method = "lm", se = FALSE) +  
  scale_x_continuous(breaks = seq(0, 2, 0.5)) +  
  scale_y_continuous(breaks = seq(0, 60, 10)) +  
  scale_color_manual(values = c("indianred3", "cornflowerblue")) +  
  facet_wrap(~Species)
```

## Introduction

## Introduction to ggplot

Now it is  
your turn!  
With the  
data  
available  
create the  
most  
horrible  
plot using  
ggplot



Now it is your turn! With the data available create the  
most *horrible plot* using ggplot