

# Data vis in ggplot2

Angela Trentacoste & Henriette Brandt  
May 2023

# Graphical grammar

- Call the `ggplot()` function which creates a blank canvas
- Specify **aesthetic mappings**, how you want to map variables to visual aspects = what you want on the x and y axes

```
ggplot(Data, aes(x,y))
```

- Add new layers that are **geometric objects** which will show up on the plot – what geometry you want assign to the info in the plot

```
ggplot(Data, aes(x,y)) + geom_point
```

# The grammar of graphics

**Example/R use** Leland Wilkinson (2005) designed the grammar upon which ggplot2 is based.

**Dataframe** • **Data:** variables mapped to aesthetic features of the graph.

**Scatter plot** • **Geoms:** objects/shapes on the graph.

**Conf ellipse** • **Stats:** statistical transformations that summarize data (e.g mean, confidence intervals)

**Colours, shapes** • **Scales:** mappings of aesthetic values to data values. Legends and axes display these mappings.

• **Coordinate systems:** the plane on which data are mapped on the graphic.

**Divide plots by site** • **Faceting:** splitting the data into subsets to create multiple variations of the same graph (paneling).

## Scatter plot with ellipses


the data df


Two columns from our df with the SI values to plot




```
ggplot(Data, aes(x=d|3C, y=d|5N))
```

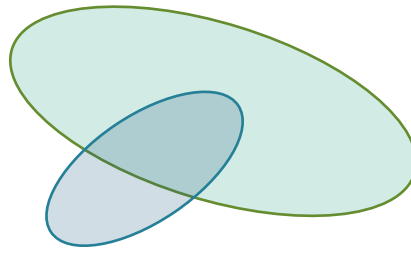
## Scatter plot with ellipses

```
ggplot(Data, aes(x=d13C, y=d15N)) +  This will apply to the following code,  
so we don't need define x and y again below  
geom_point(aes(colour = TaxonGroup, shape = TaxonGroup), size = 3)
```

  
The geometry,  
here a scatter plot

  
Scatter plot aesthetics: colours, shapes, etc.  
For things we want to change with groups  
(e.g. diff colours for diff sites) – must go inside aes()  
For things to apply equally to every (e.g. point size),  
that goes outside aes()

## Scatter plot with ellipses

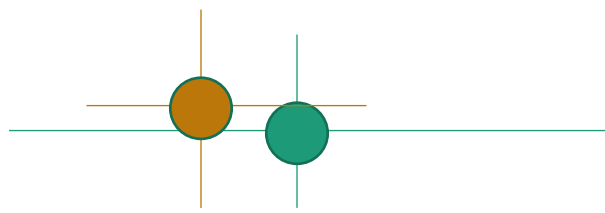


Each additional line adds or alters  
a visual element of the plot

```
ggplot(Data, aes(x=d13C, y=d15N)) +  
  geom_point(aes(colour = TaxonGroup, shape = TaxonGroup), size = 3) +  
  stat_ellipse(aes(fill = TaxonGroup), geom = "polygon", alpha = 0.2) +  
  facet_wrap(~Site) +  
  theme_bw(base_size = 15) +  
  theme(strip.background = element_rect(fill = "white", linetype = "blank")) +  
  labs(  
    y=expression(paste(delta^{15}, "N (\u2030)")),  
    x = expression(paste(delta^{13}, "C (\u2030)")),  
    colour = "Taxon", shape = "Taxon", fill = "Taxon" )
```

Let's see how it works...

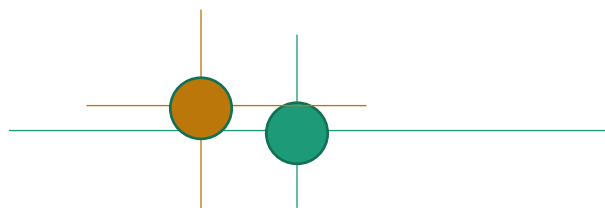
## Scatter plot with sd bars



- To plot sd bars, we need to calculate the mean and sd of each set of SI values
- We can do this using `group_by()` to make groups and `summarise()` to summarise the data

```
DataSummary <- Data %>%      # make new object called DataSummary. From Data then....  
  group_by(Site,TaxonGroup) %>% # make groups  
  summarise(                  # for each group summarise...  
    Cmean = mean(dI3C),      # make new column Cmean, in it put the mean() of dI3C  
    Csd = sd(dI3C),          # make new column Csd, in it put the sd() of dI3C  
    Nmean = mean(dI5N),  
    Nsd = sd(dI5N))
```

## Scatter plot with sd bars



```
ggplot() + # just call empty ggplot() because we are using different df for big and small points
  geom_point(data = Data, # small points for individuals. Info in df Data
    aes(x=d13C, y=d15N, colour = TaxonGroup, shape = TaxonGroup), size = 2) +
  geom_point(data = DataSummary, # big points for means. Info in df DataSummary
    aes(x = Cmean, y=Nmean, colour = TaxonGroup), size = 5) + # large size for points
  # now add vertical and horizontal bars using info from DataSummary.
  # length of each bar is set by an equation and then have it change colour with TaxonGroup
  geom_errorbar(data = DataSummary, aes(x=Cmean, ymin=Nmean-Nsd, ymax=Nmean+Nsd,
    colour = TaxonGroup)) +
  geom_errorbarh(data = DataSummary, aes(y = Nmean, xmin=Cmean-Csd, xmax=Cmean+Csd,
    colour = TaxonGroup)) +
```



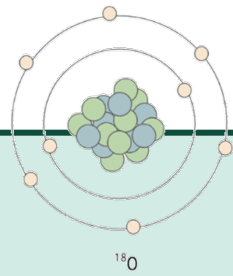
# TEETH TIME



Hot from the oven – isotope results!

## Getting things organised

- We normally start with 2-3 tables, which we want to link by sample and ASIL nos.



**ASIL + SI value**  
(and redundant info)  
via Nils



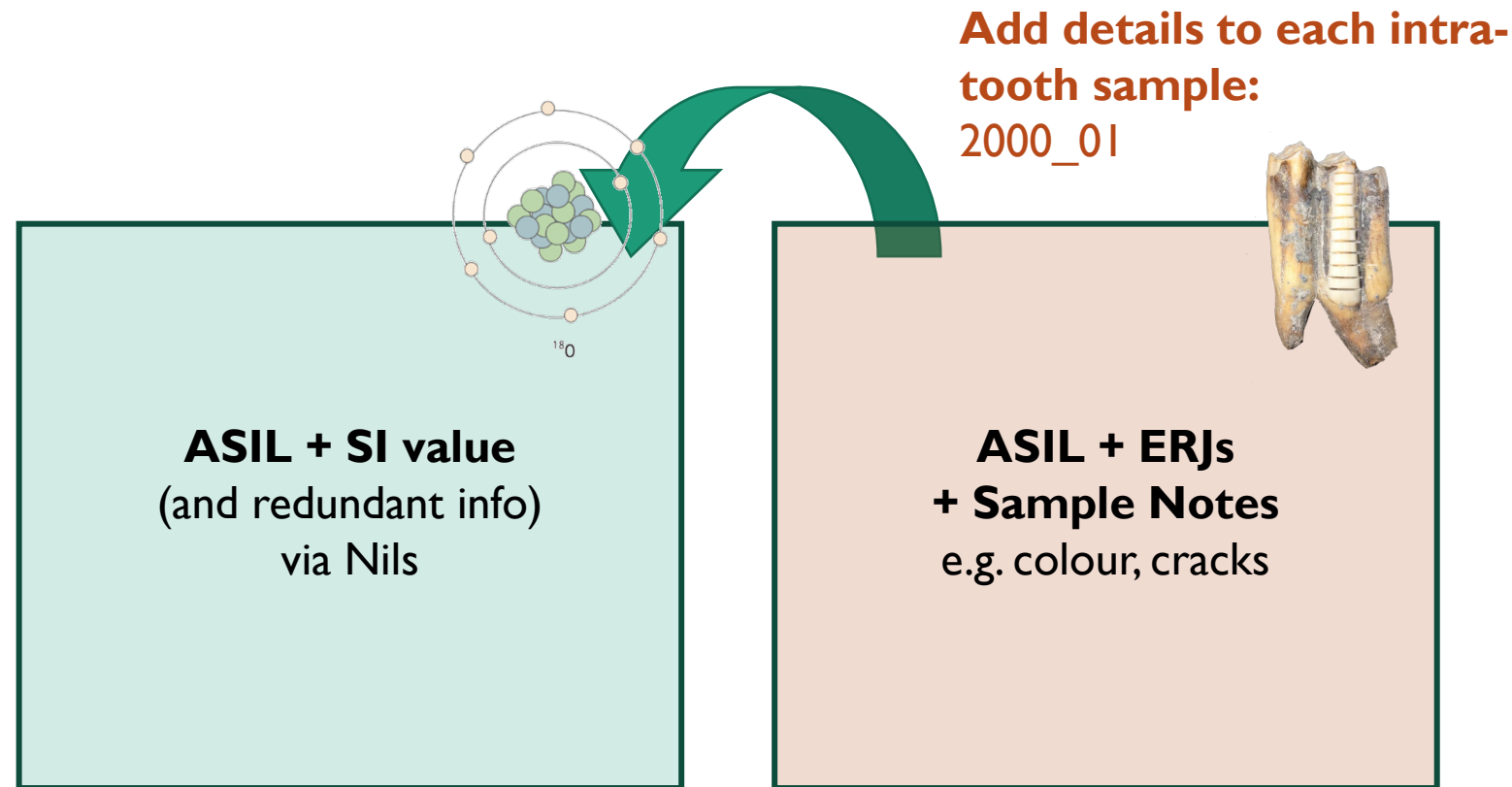
**ASIL + ERJs**  
**+ Sample Notes**  
e.g. colour, cracks



**ASIL + Tooth Into**  
**+ Tooth Notes**  
taxon, site, man/max,  
measurements, wear stage etc

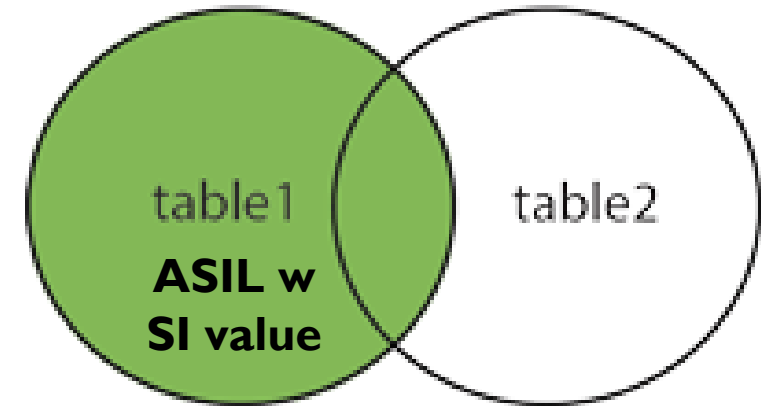
## Getting things organised

- Left join sample into to sample number



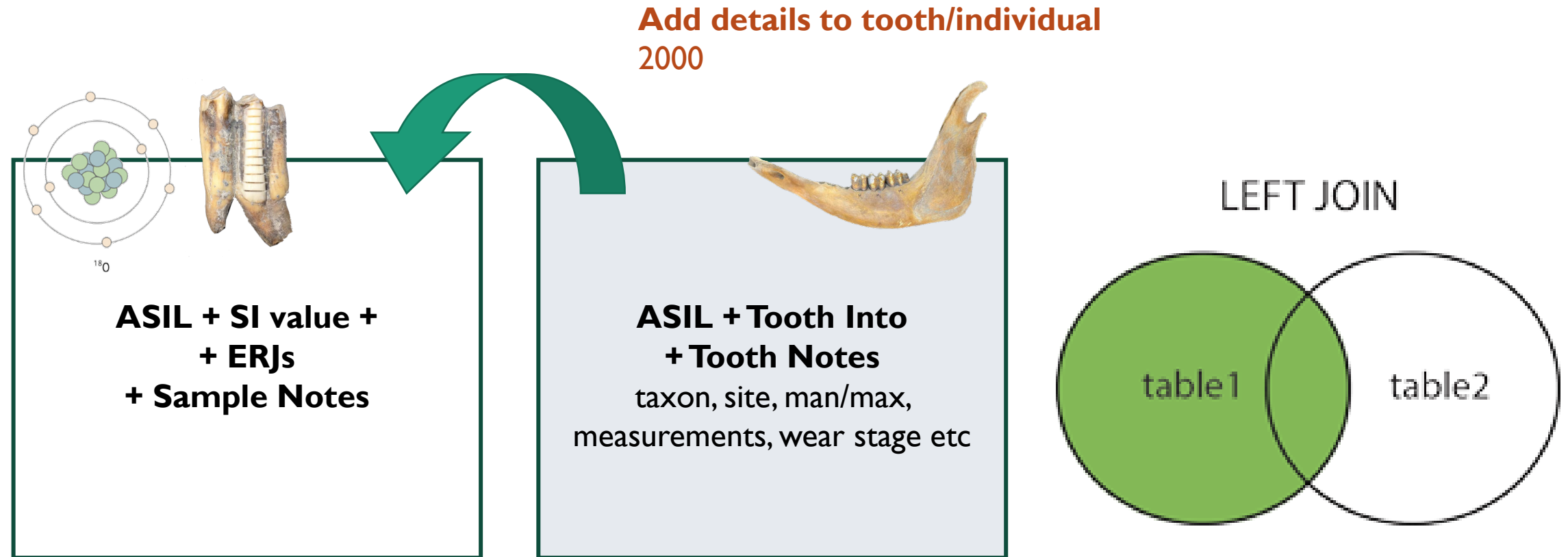
All values in left table &  
only values in R table that are  
matched

LEFT JOIN



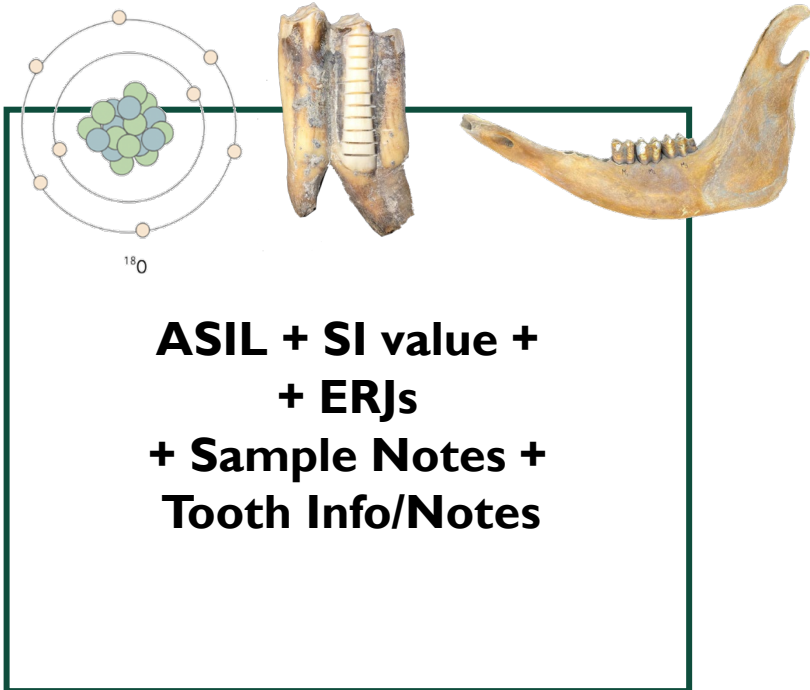
## Getting things organised

- Left join again with new df and tooth into



## Getting things organised

- Now we have a mega-table with all the info for each sample



# Time to plot some curves

the data df



Two columns from our df with the ERJs and SI values



```
ggplot(Enameldata, aes(ERJ, dI8O)) +
```

```
geom_line() +
```

Add line geometry

```
geom_point(colour = "darkblue") +
```

Add point geometry (note order and colour)

```
facet_wrap(vars(ASIL)) +
```

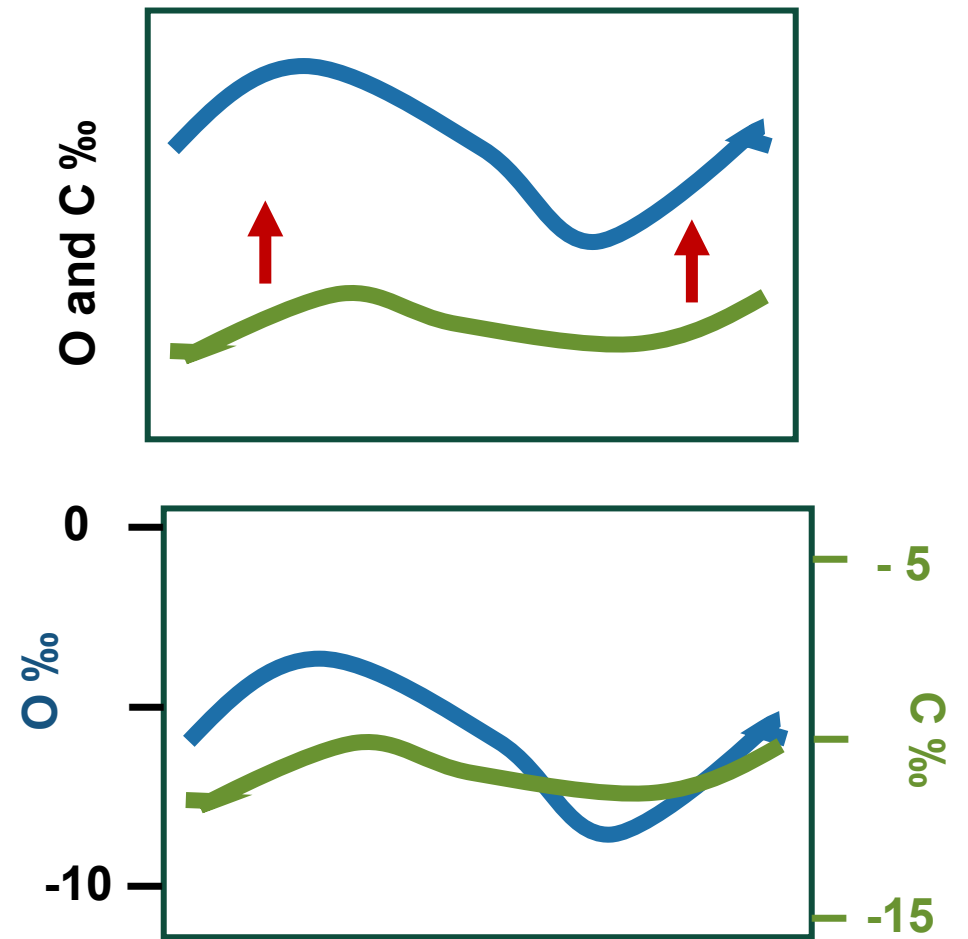
Facet by ASIL

```
theme_minimal()
```

Nice white minimal theme

## Second axis

- R uses one y-axis for both variables
- Want to make 2<sup>nd</sup> axis, e.g. **dI3C**
- But right **y axis** is pegged to left y
- In ggplot can only do this by transforming the left y axis – you can't assign a scale, only manipulate the existing y scale
  - So we shift **dI3C** up ↑
  - And then create a new y right axis subtracting ↓
  - So the **right labels** and **dI3C values** line up



## Basic data viz principles

- Be honest
  - R makes beautiful plots that can hide message data
  - Don't hide the data or mispresent them
  - e.g. don't use box plots for non-normal data
- Keep it simple and clear
  - Manipulate colours, points and text size to improve clarity (colour blindness)
  - Don't clutter
  - Put legend info in the legend, not the caption

