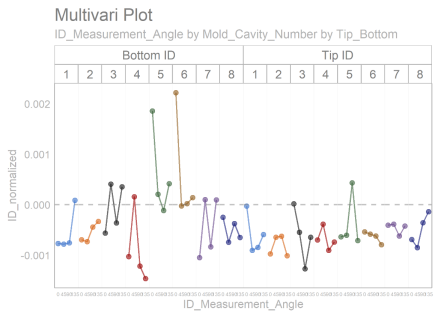
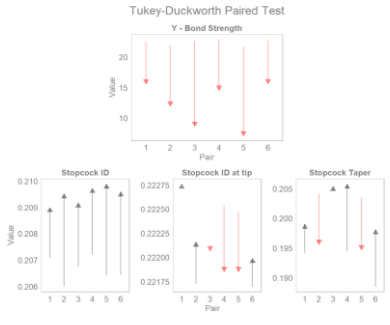
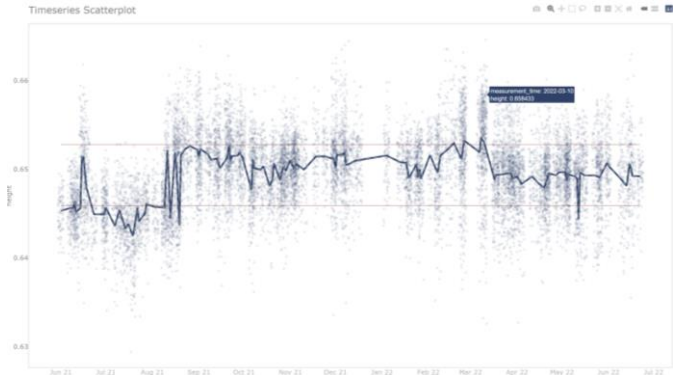
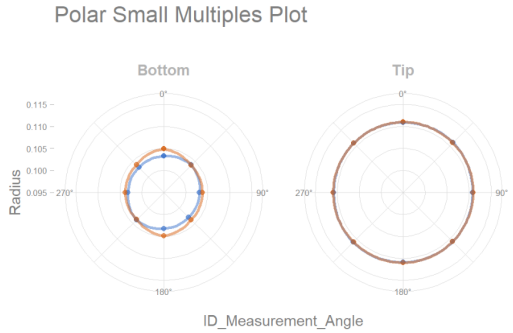
Intro to {sherlock}: an R Package for Problem Diagnosis by Gabor Szabo

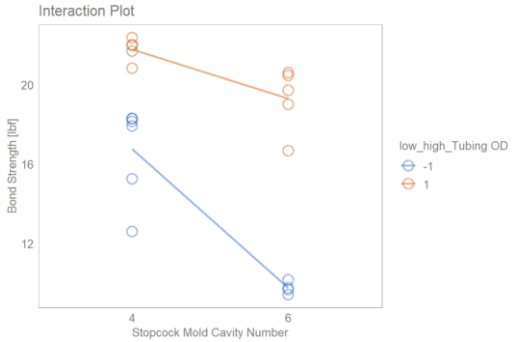
SoCal RUG, April 18th, 2023











About Gabor Szabo

Sr. Principal Engineer, Operations Excellence at Edwards Lifesciences

17 years of experience in engineering (quality and manufacturing)

Bachelors in Engineering Management

Have used R for 2 years

- Data analysis
- Shiny
- Package development: **sherlock**
- Taught fundamentals of R programming at Edwards Lifesciences
- **R for Engineering** newsletter on LinkedIn

From Excel to R to package development

Data analysis has always been my strong suit

Built custom applications in Excel

Tried to learn Python with no luck – haven't given up yet

Stumbled upon R two years ago – love at first sight

Decided to build a set of functionality for my own use

The idea of a package

Development and submission

sherlock

Provides graphical and statistical tools for **diagnosis of manufacturing-related problems**

Small multiples types of visuals

Typically small datasets from experiments

Target audience: engineers who are tasked with diagnosing quality or performance-related problems of products, machines or processes existing in the physical world

Has been on CRAN since November 2022. Current version is 0.6.0

Product Quality and Reliability

Important product characteristics

- Customer level: safe, affordable, reliable, comfortable, attractive
- Manufacturer level: form, fit, function

Variation is the enemy

- Raw materials, components
- Processing
- Environment

Product Quality and Reliability

Proactive

Product quality and reliability

Problems prevented

Robust design (quality and reliability “designed in”)

Reactive

Deviation from specified conditions

Problem needing resolution

Problem diagnosis – problem solving

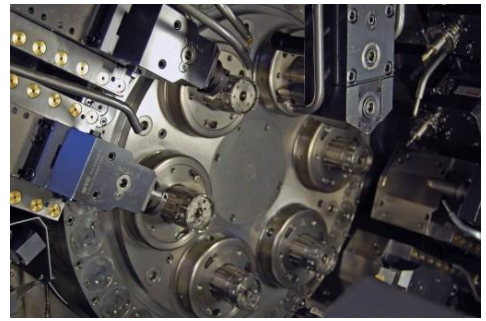
Why did this problem occur?

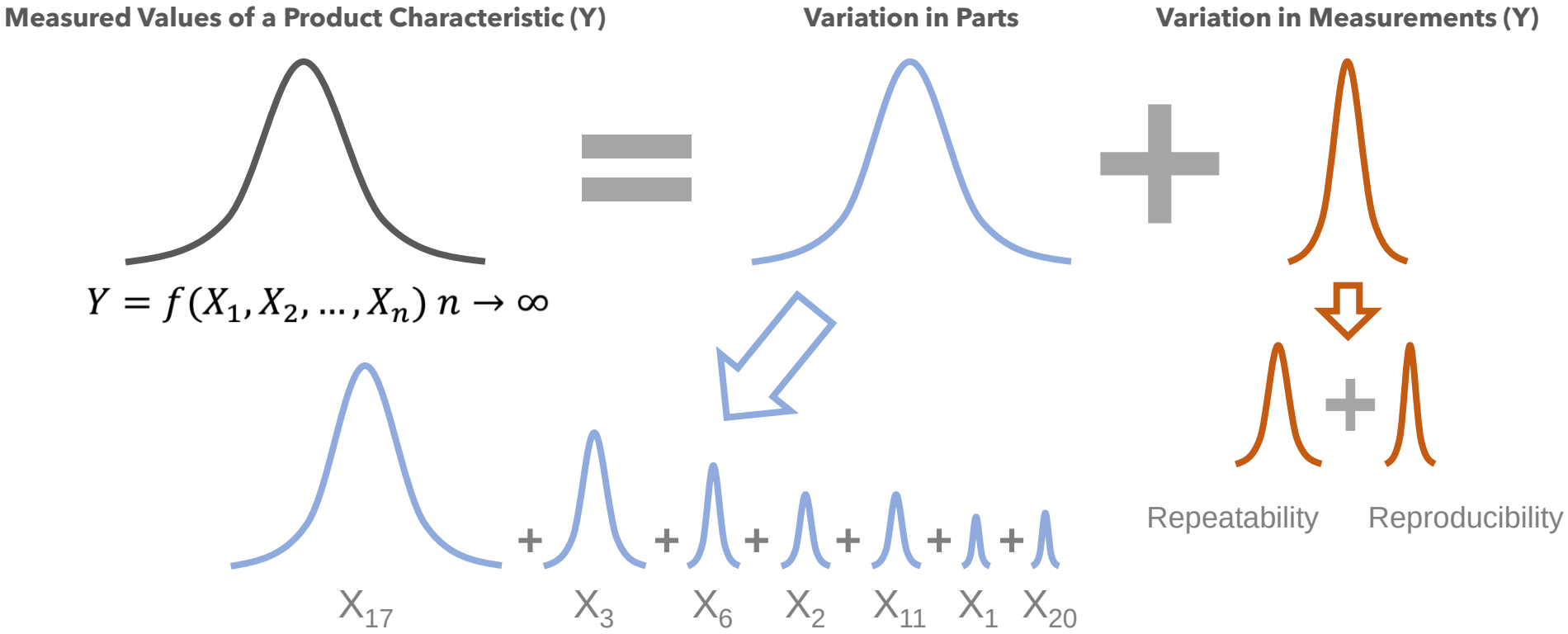
What drives variation in an important product characteristic?

Causality in Manufacturing

Deterministic: laws of physics apply. Energy transformation.

Sparsity of effects: tells us that any effect cannot be evenly attributed to a large number of variables[1].





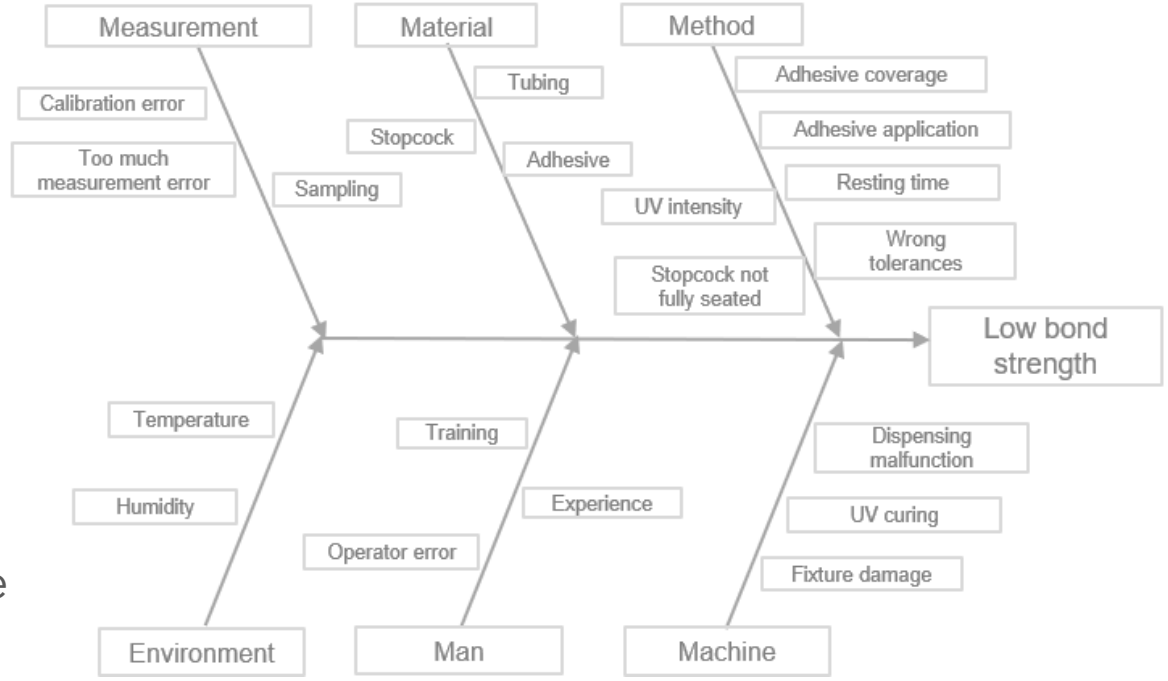
[1]Diagnosing Performance and Reliability, David J. Hartshorne and The New Science of Fixing Things, 2019

Diagnostic Approaches

Fishbone - Brainstorming

Hypothesis generators based on brainstorming and some amount of existing knowledge. The goal of a Fishbone exercise is to consider the potential impact of as many variables (Xs) as possible that we already know about or can think of. *This approach, however, is not effective and is definitely not efficient*.

1. Testing the effect of each of the potential variables may be possible, but most of it would be a waste of time and resources
2. Since these potential variables (Xs) are usually the result of brainstorming, a lot of times what we are searching for is not even on the list!
3. Consider how long it would take us to test the effect all of these potential causes; several weeks, if not months! We simply don't have that much time.



What's wrong?

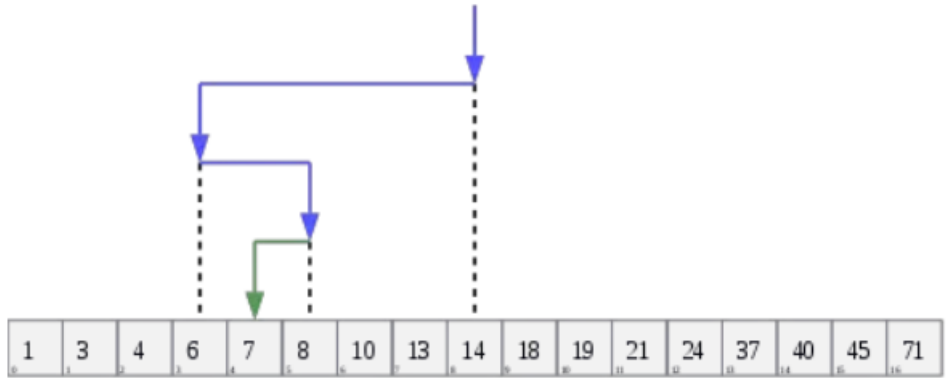
Diagnostic Approaches

Progressive search – Process of Elimination

In a progressive search, you start from the Y and converge on the X through posing and answering a series of questions

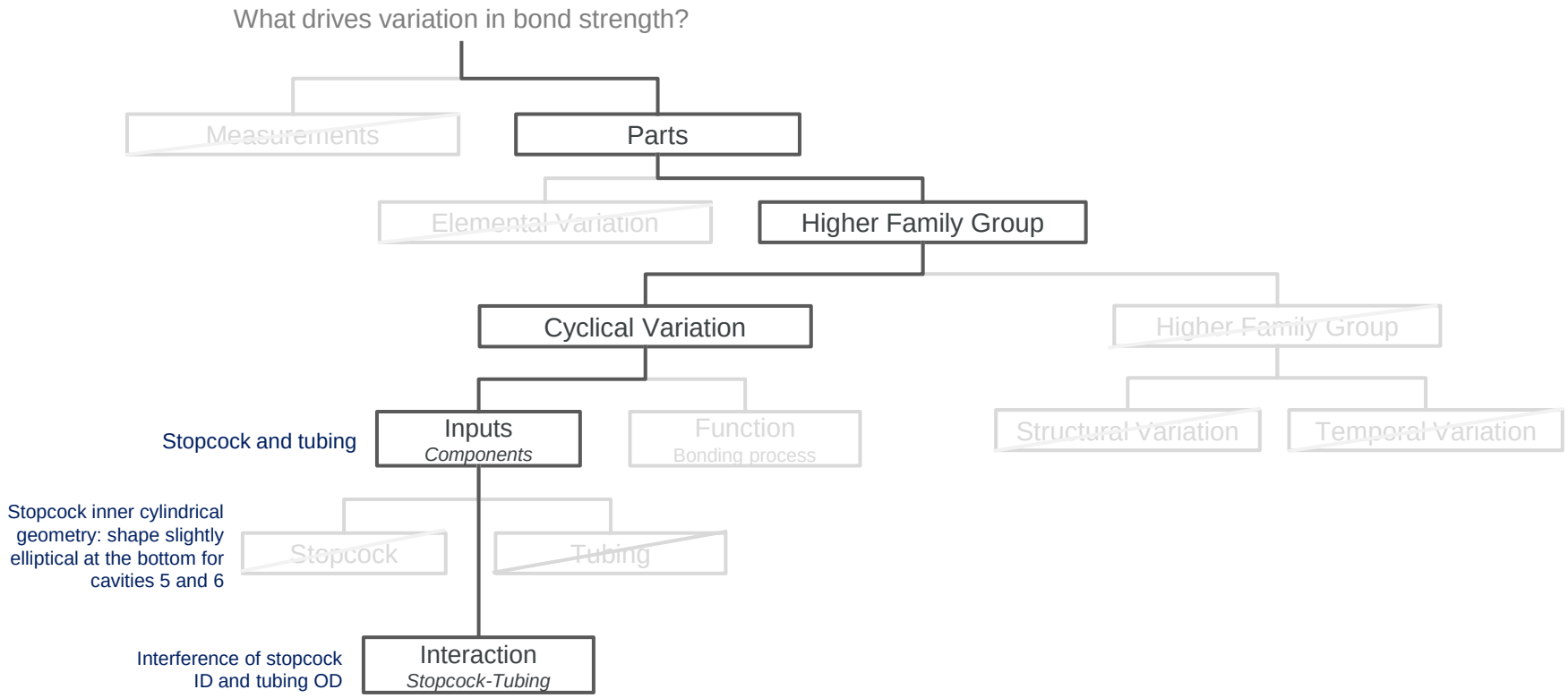
The questions must be phrased carefully so that all possibilities not yet eliminated are included thereby progressively reducing the *search space* – think binary search

What's different?



Diagnostic Approaches

Search Tree

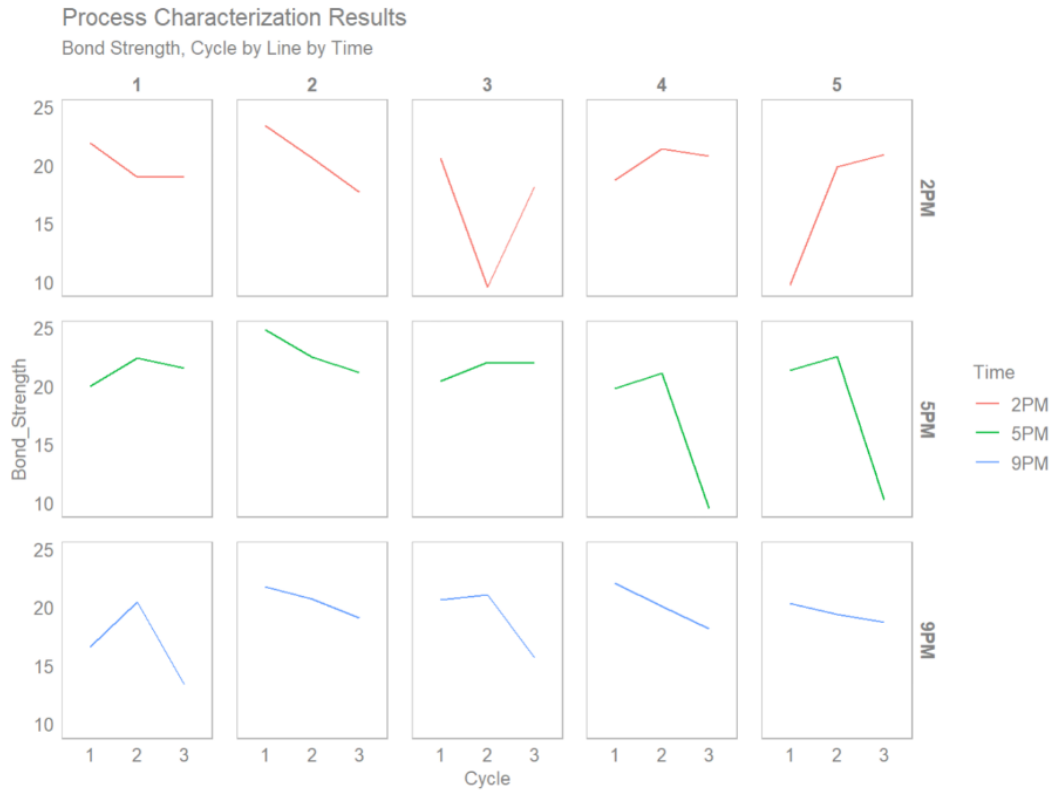


Small Multiples

Data visualization technique

Coined by data visualization pioneer and educator Edward Tufte

Instead of plotting all observations from a dataset in the same space, they are grouped and displayed in separate smaller displays using the same axes and scale



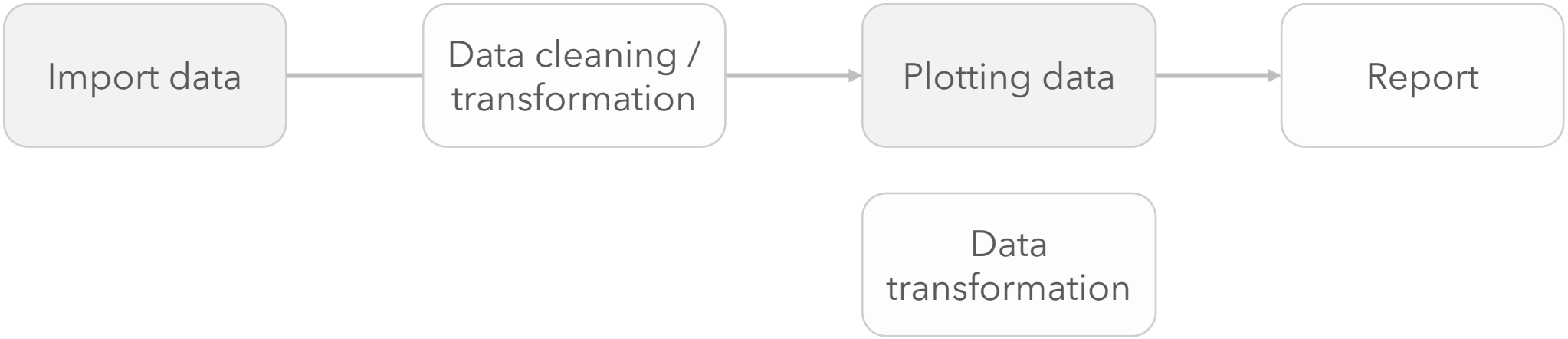
"That is to say, nature's laws are causal; they reveal themselves by comparison and difference, and they operate at every multi-variate space-time point" - Edward Tufte

Small multiples are codename for faceting in ggplot2

What's different?

sherlock - basic workflow

Simplicity

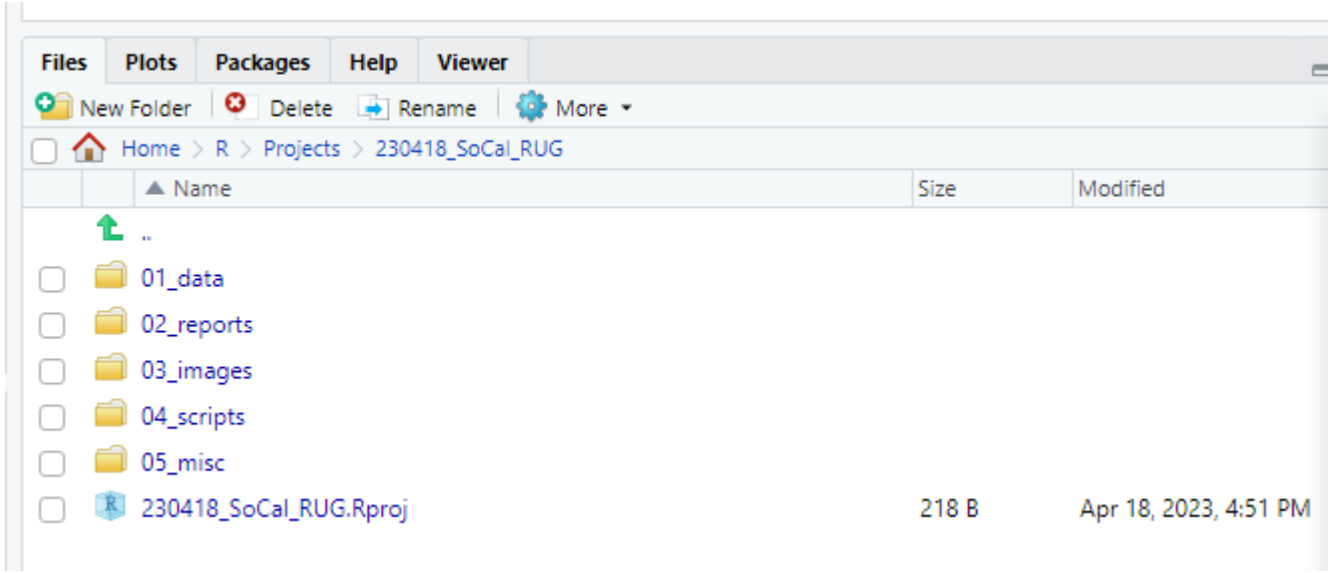


Functionality

sherlock::create_project_folder()

Helper function for setting up a project folder

```
create_project_folder(folder_name = "230418_SoCal_RUG", path = "Projects/")
```



sherlock::load_file(), load_files()

Wrapper functions around common functions for reading in .xlsx, .txt and .csv files

The ability to read in and clean multiple files (integration with a custom data cleaning function)

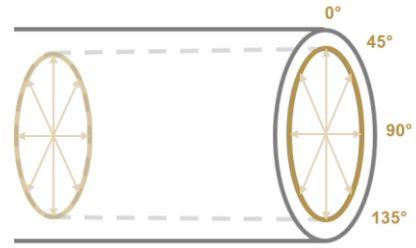
sherlock::draw_multivari_plot()

Small multiples plot

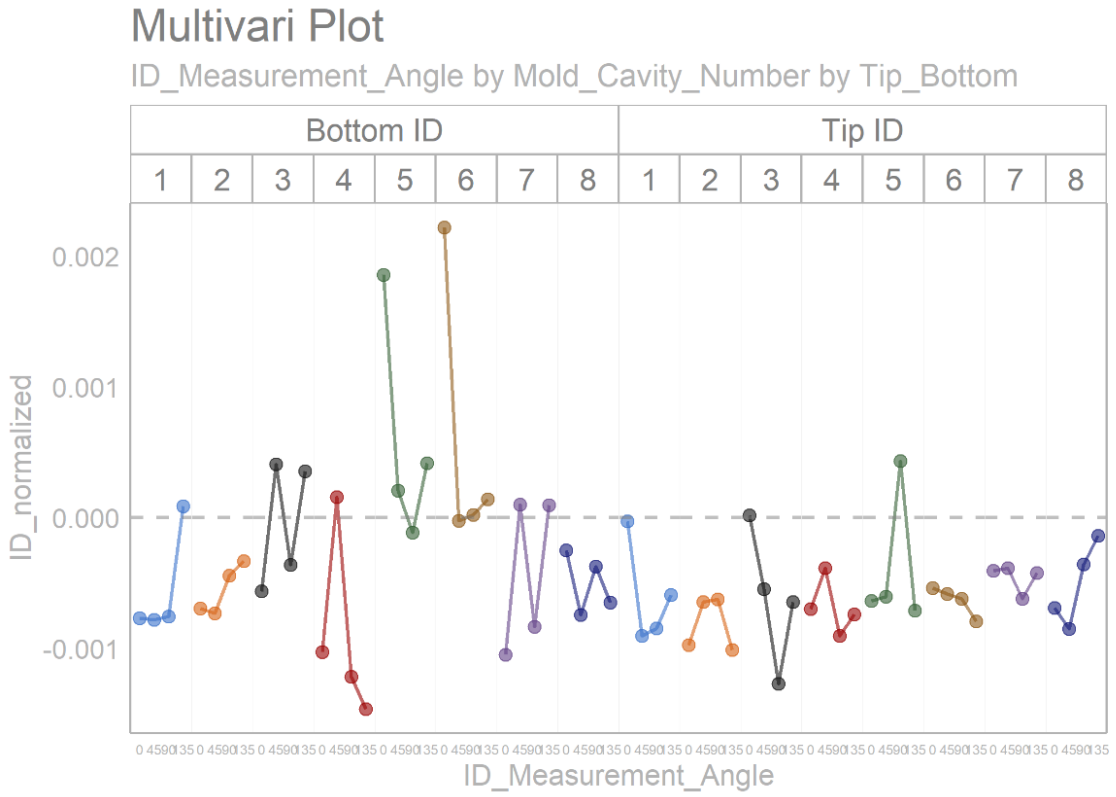
Takes a dataframe that includes the Y characteristic of interest and levels (groups) of X variables and outputs a multivari plot

Can handle up to 3 levels (groups)

Can be used with other helper functions such as *draw_horizontal_reference_line()* or *normalize_observations()*



ID at bottom ID at tip




```r
normalized_tbl %>%
  draw_multivari_plot(response     = ID_normalized,
                      factor_1     = ID_Measurement_Angle,
                      factor_2     = Mold_Cavity_Number,
                      factor_3     = Tip_Bottom,
                      x_axis_text  = 6) +
  draw_horizontal_reference_line(reference_line = 0)
```

sherlock::normalize_observations()

Normalizing observations can be helpful to show deviation from an expected value, especially when multiple groups, each having a different expected value, are being plotted

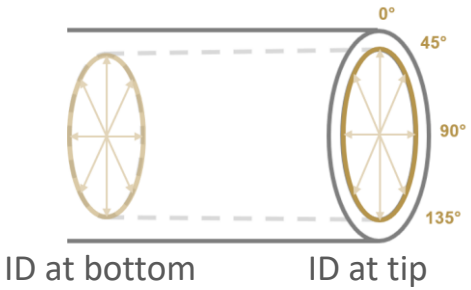
Takes a dataframe and outputs a dataframe adding a column with observations normalized for the Y characteristic of interest

The new dataframe with the normalized values can then be plotted

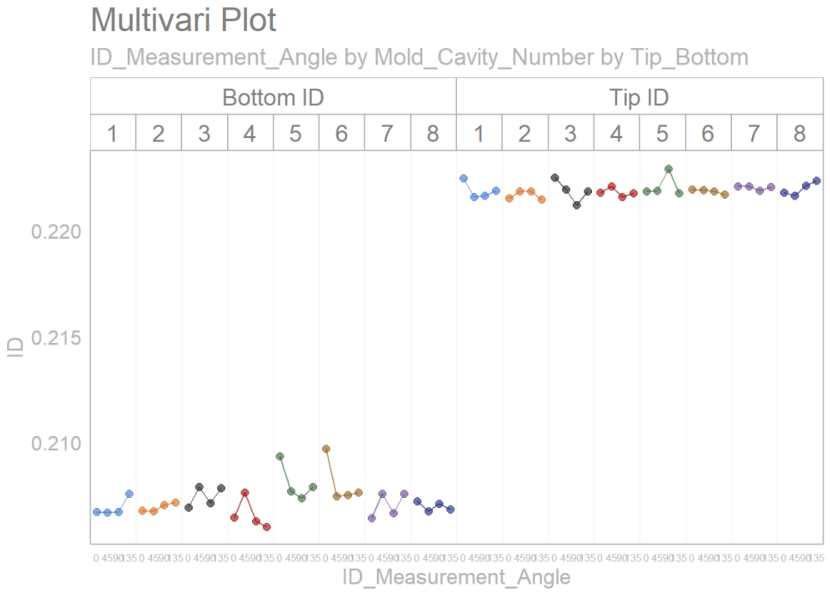
```
normalized_tbl <- polar_small_multiples_data %>%
    select(-ID_2) %>%
  filter(ID_Measurement_Angle %in% c(0, 45, 90, 135)) %>%
  normalize_observations(response = ID, grouping_var = Tip_Bottom, ref_values = c(0.2075, 0.2225))

## Joining, by = "Tip_Bottom"

normalized_tbl
```

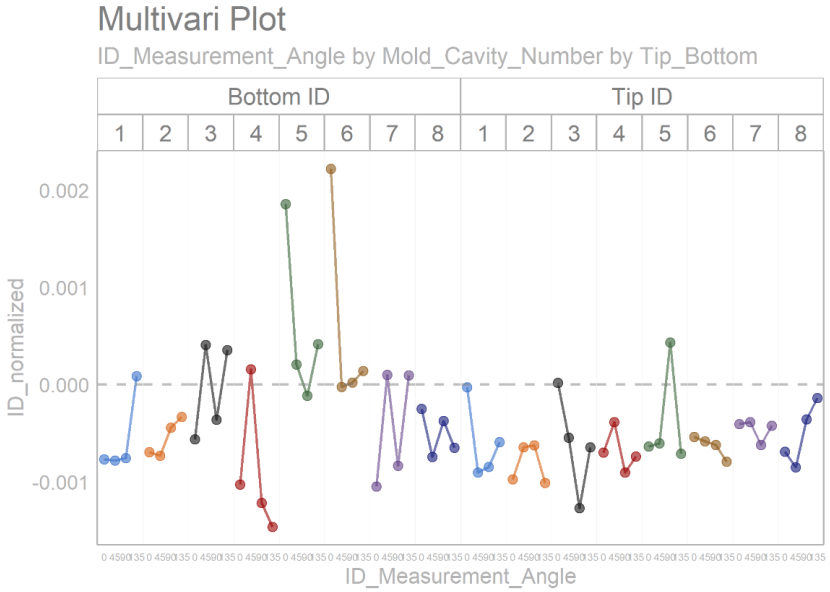
Creates new column with normalized values

```
## # A tibble: 64 × 6
##    ID_Measurement_Ang… Mold_Cavity_Num… Tip_Bottom    ID ref_value ID_normalized
##                  <dbl>            <dbl> <chr>      <dbl>     <dbl>         <dbl>
##  1                   0                1 Bottom ID  0.207     0.208   -0.000770
##  2                  45                1 Bottom ID  0.207     0.208   -0.000780
##  3                  90                1 Bottom ID  0.207     0.208   -0.000759
##  4                 135                1 Bottom ID  0.208     0.208    0.0000882
##  5                   0                2 Bottom ID  0.207     0.208   -0.000694
##  6                  45                2 Bottom ID  0.207     0.208   -0.000732
##  7                  90                2 Bottom ID  0.207     0.208   -0.000444
##  8                 135                2 Bottom ID  0.207     0.208   -0.000333
##  9                   0                3 Bottom ID  0.207     0.208   -0.000565
## 10                  45                3 Bottom ID  0.208     0.208    0.000406
## # … with 54 more rows
```



ID at bottom ID at tip





You can easily turn this… → …into this!

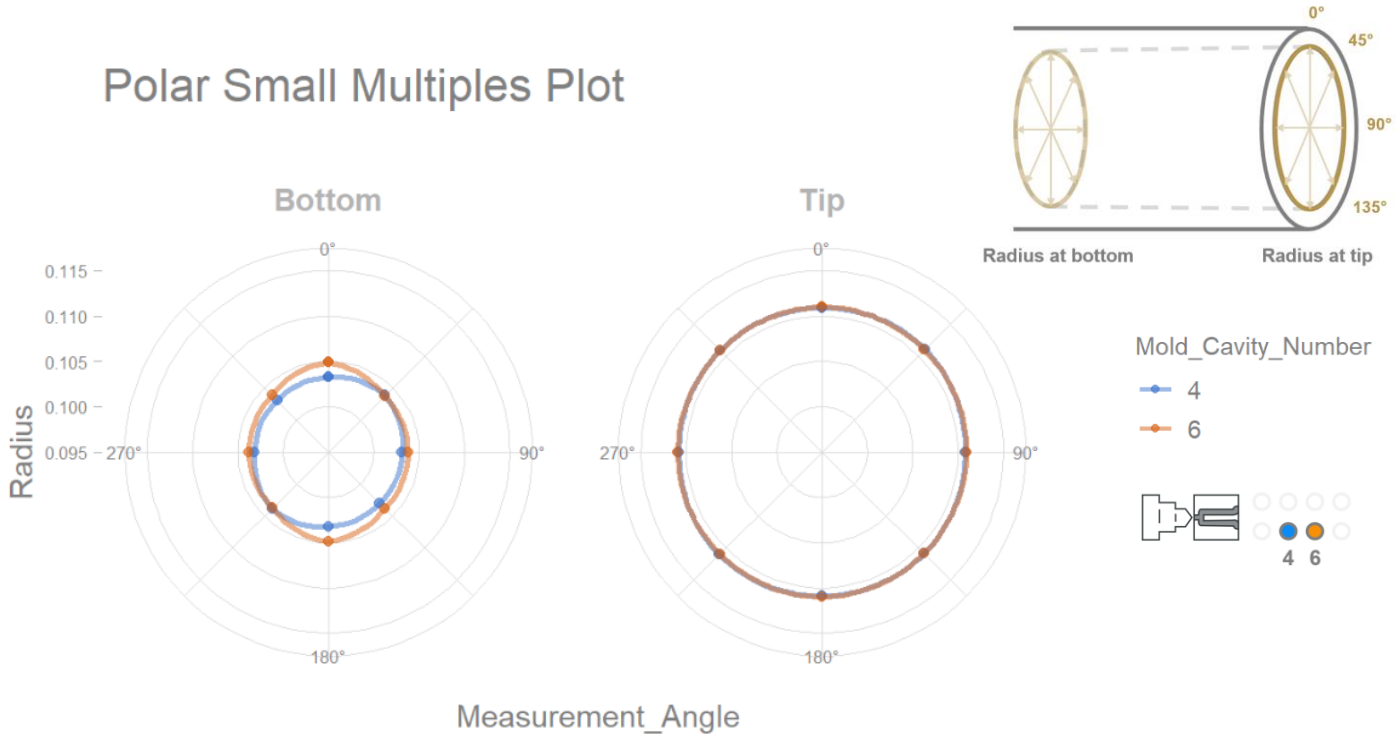
sherlock::draw_polar_small_multiples()

Useful for plotting rotational or cylindrical data to understand shape or positional deviations

Information carried on the parts as to how their geometry is created[1]

Example 1: Shape of the inner cylinder of an injection molded component

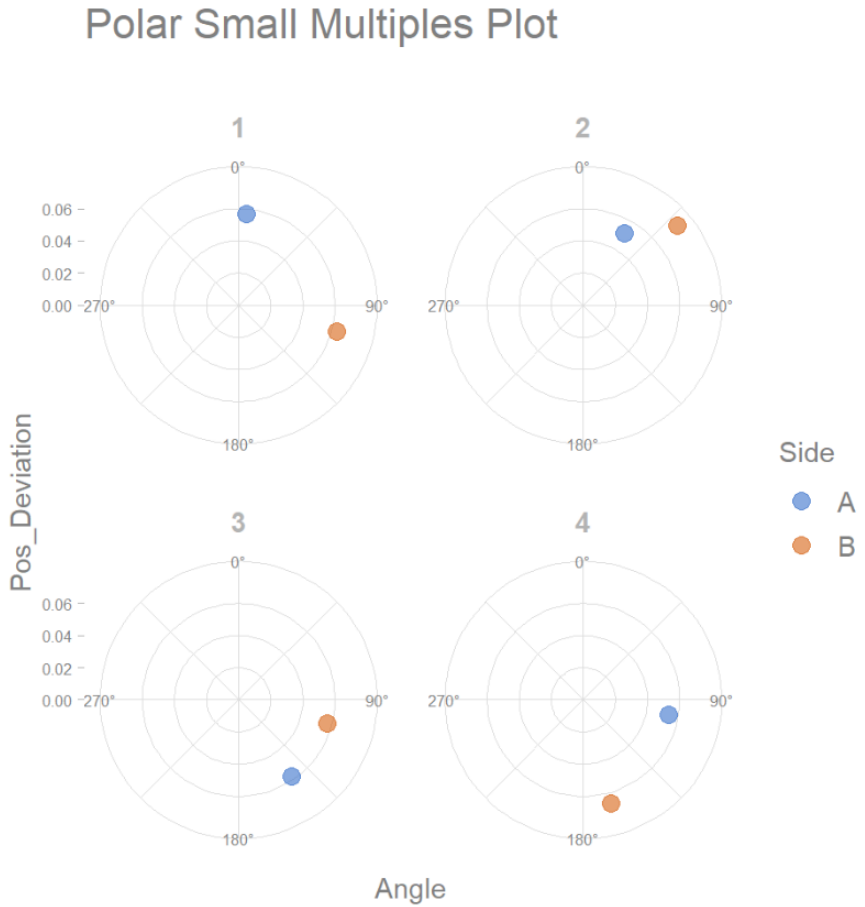
```r
```{r}
polar_small_multiples_data %>%
  mutate(Tip_Bottom = case_when(Tip_Bottom == "Bottom ID" ~ "Bottom",
                                Tip_Bottom == "Tip ID" ~ "Tip")) %>%
  filter(Mold_Cavity_Number %in% c(4, 6)) %>%
  rename(Radius = "ID_2") %>%
  draw_polar_small_multiples(angular_axis    = ID_Measurement_Angle,
                             x_y_coord_axis = Radius,
                             grouping_var    = Mold_Cavity_Number,
                             faceting_var_1 = Tip_Bottom,
                             point_size      = 1.5,
                             line_size       = 1,
                             connect_with_lines     = TRUE,
                             x_y_coord_axis_limits = c(0.095, 0.115)) +
    theme(axis.text = element_text(size = 6))
```
```



[1]Diagnosing Performance and Reliability, David J. Hartshorne and The New Science of Fixing Things, 2019

sherlock::draw_polar_small_multiples()

Example 2: Positional deviation of the two tips of a long, cylindrical component



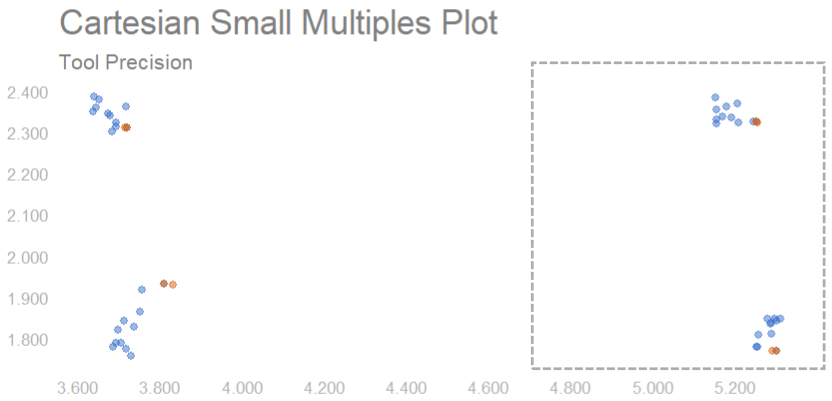

```r
polar_plot_tbl %>%
  group_by(Needle, Datum) %>%
  filter(Pos_Deviation == max(Pos_Deviation)) %>%
  ungroup() %>%
  rename(Side = "Datum") %>%
  filter(Needle %in% 1:4) %>%
  draw_polar_small_multiples(angular_axis    = Angle,
                             x_y_coord_axis = Pos_Deviation,
                             grouping_var    = Side,
                             faceting_var_1 = Needle,
                             point_size      = 3,
                             label_text_size = 6)
```

sherlock::draw_cartesian_small_multiples()

A powerful plot for visualizing data in a cartesian coordinate system (x-y)

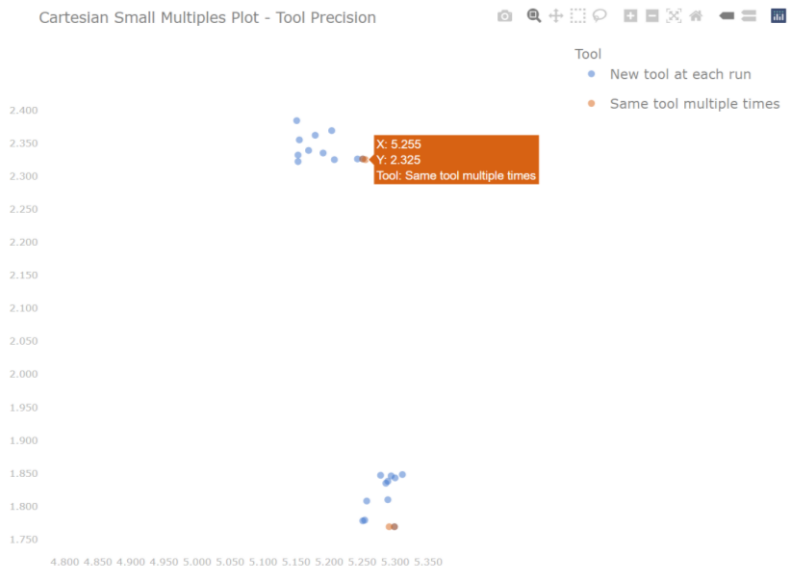
Custom grouping and faceting

Full interactivity (built-in Plotly view)



Static Plot

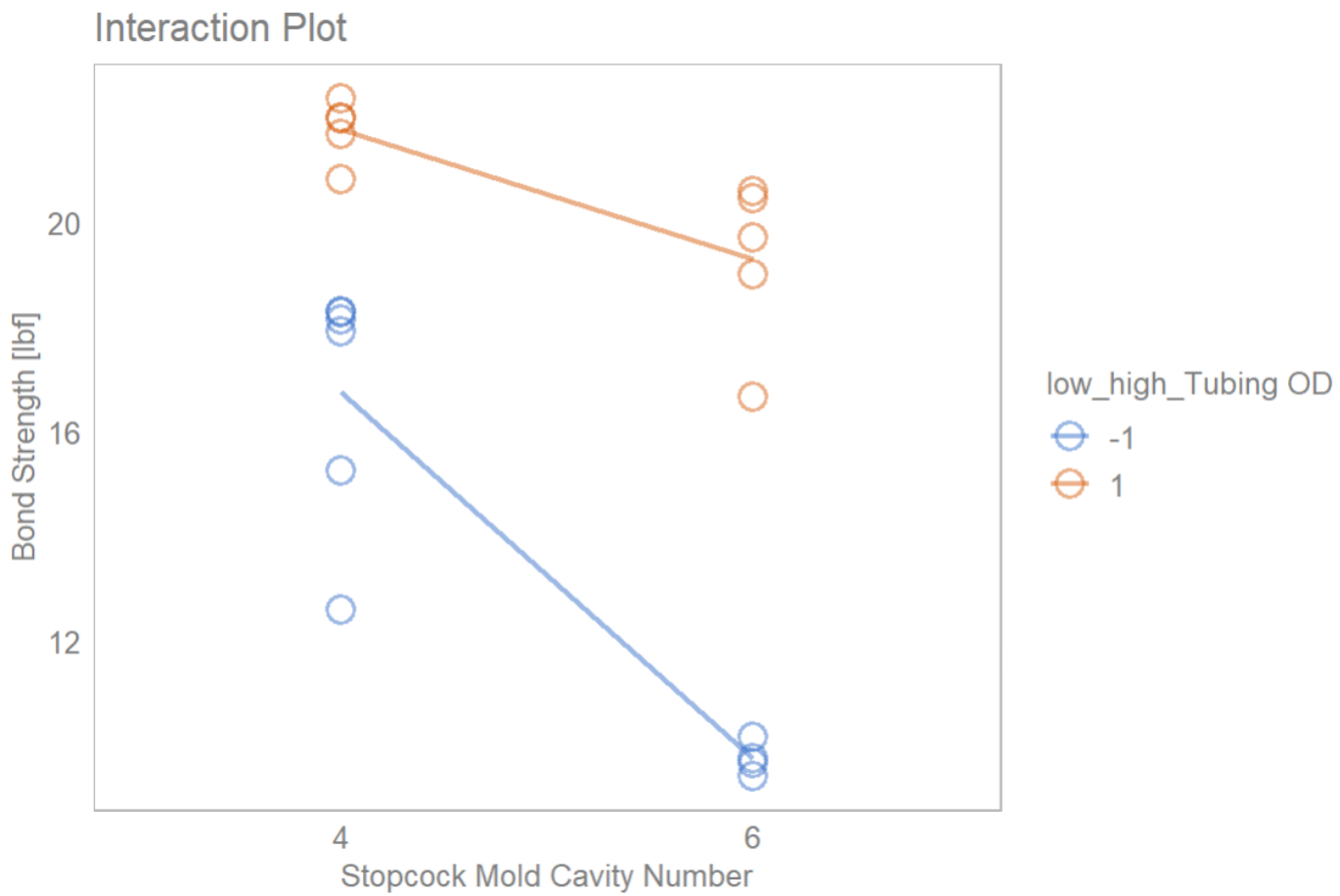
Interactive Plotly Plot



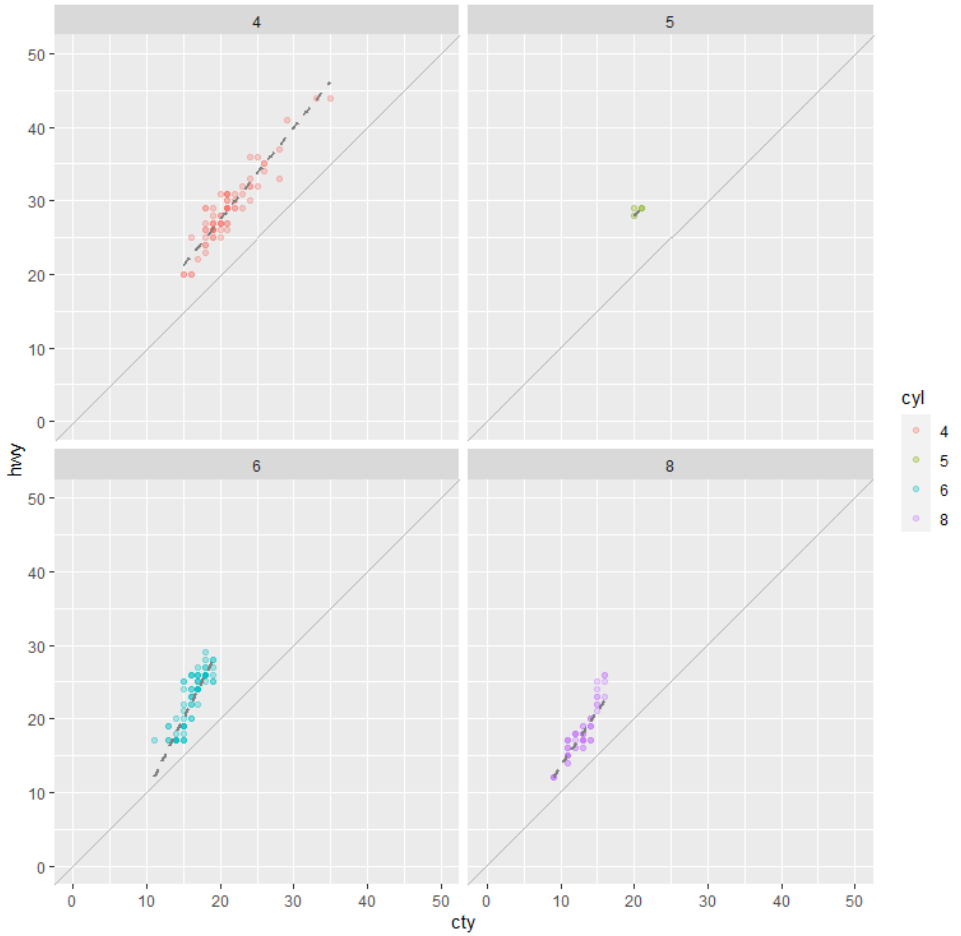
sherlock::draw_interaction_plot()

Used to visualize two-way interactions

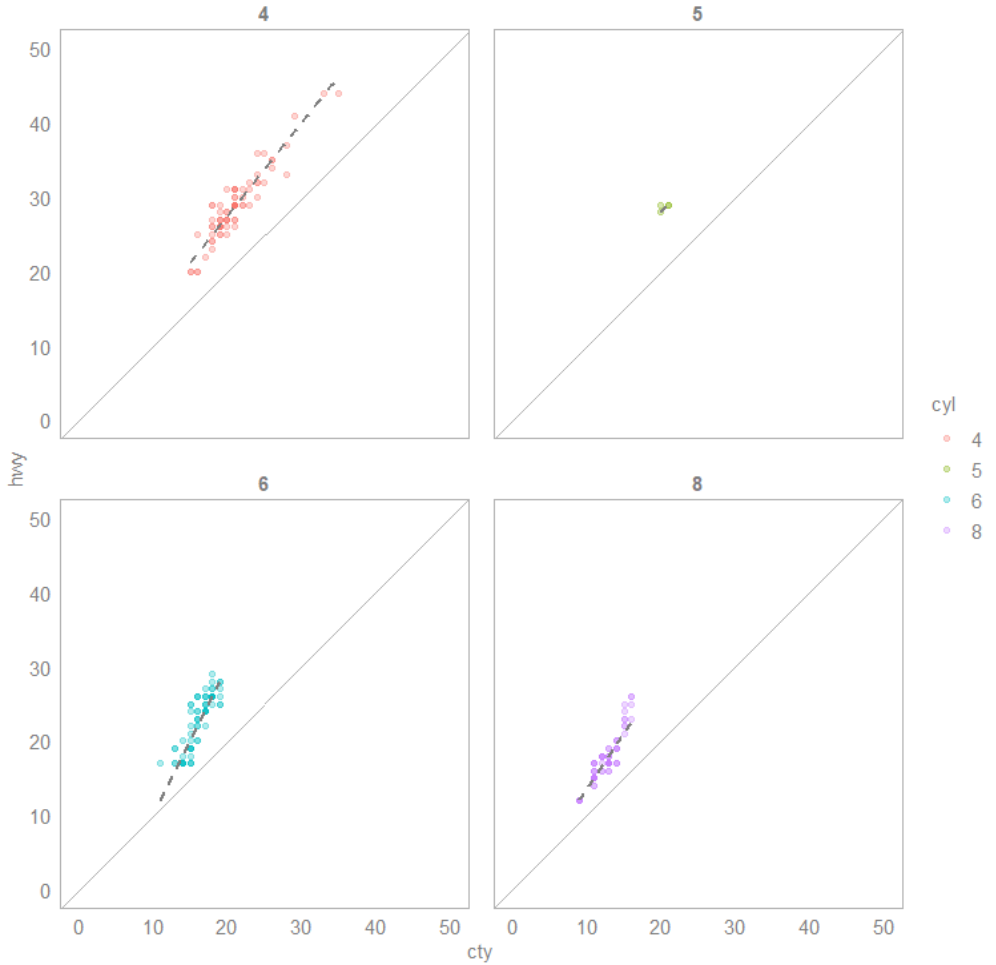
Displays both main effects (means) and individual values



sherlock::theme_sherlock()

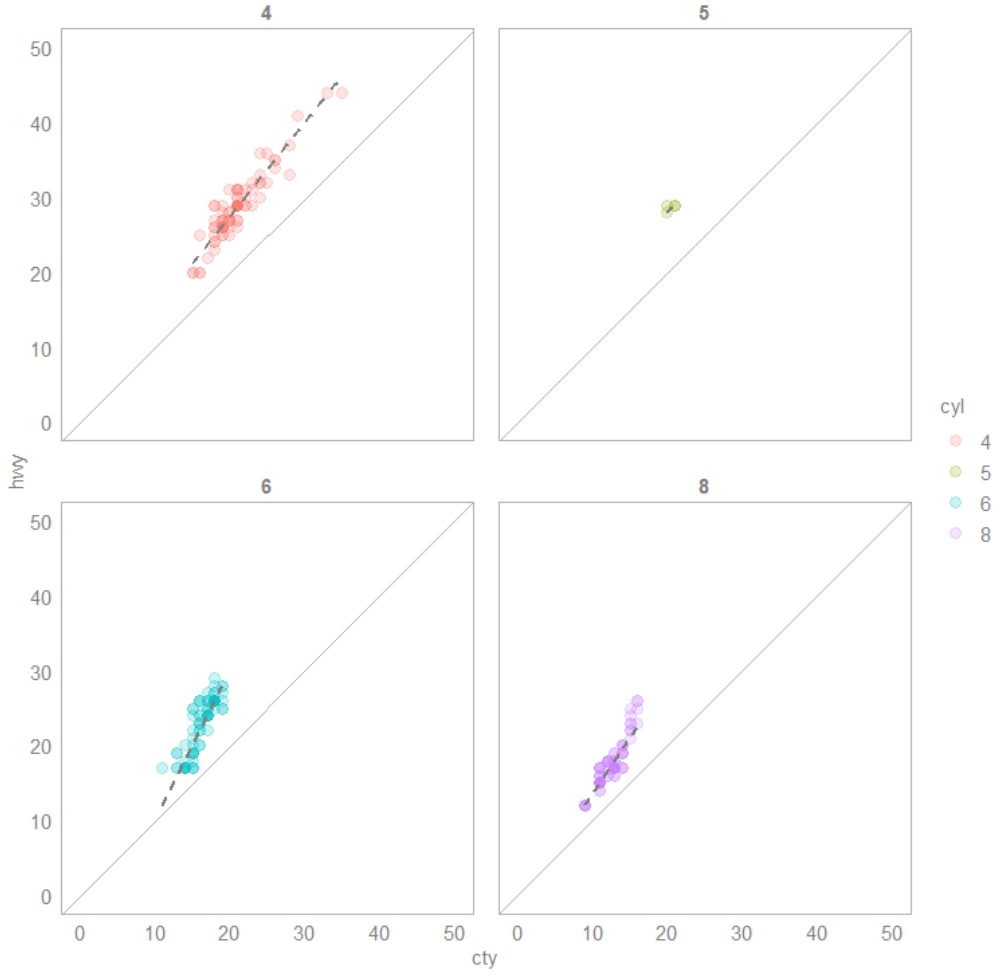


Default

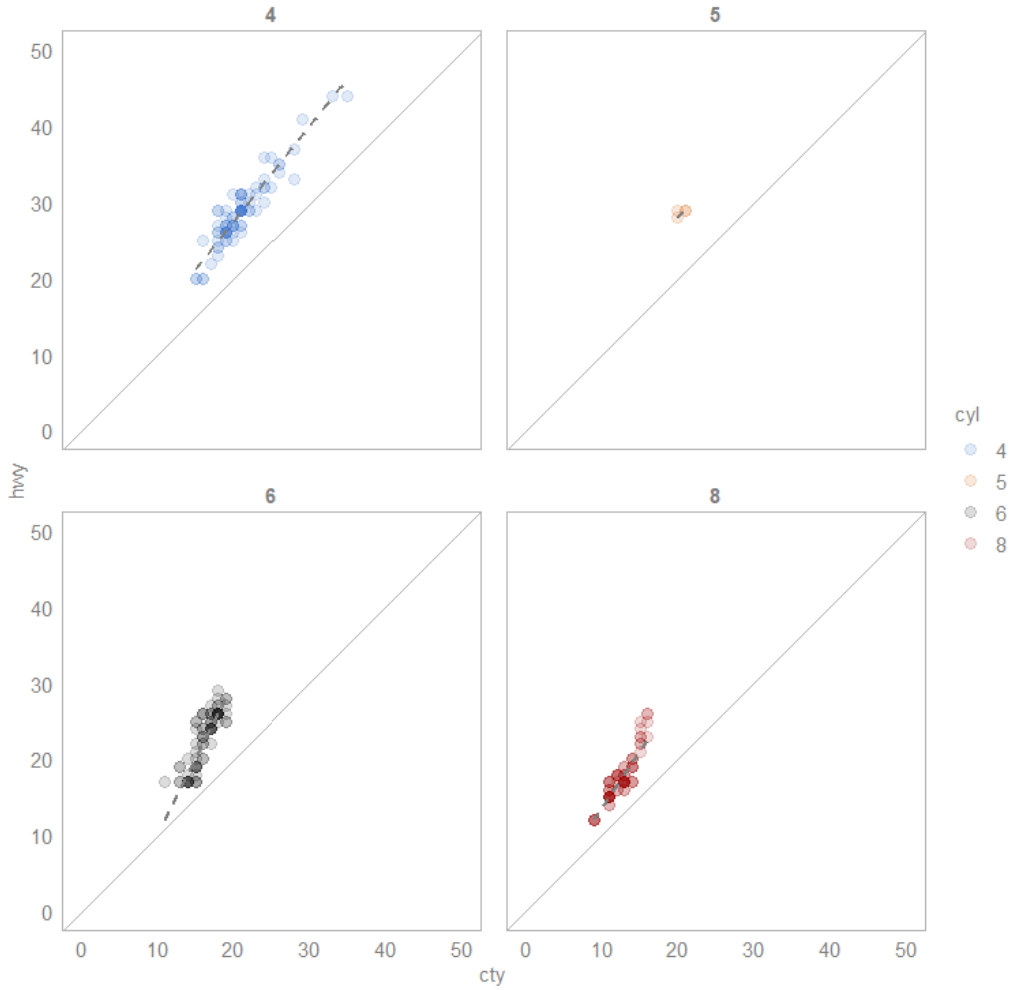


theme_sherlock()

sherlock::scale_color_sherlock(), scale_fill_sherlock()



Default



scale_color_sherlock()

Package Development

Package development

Always design with the end user in mind

Standard argument naming conventions

- Y variable
- Grouping variables
- Faceting variables
- Aesthetics (color, alpha, size etc.)
- Labels (x, y axis, subtitle etc.)
- Interactive: Boolean

It **is** a software development project, and you are the developer, project manager, tester, technical writer, marketing specialist

Thank you for your attention!

Connect with me on LinkedIn

install.packages("sherlock")

https://github.com/gaboraszabo/sherlock

https://gaboraszabo.github.io/sherlock