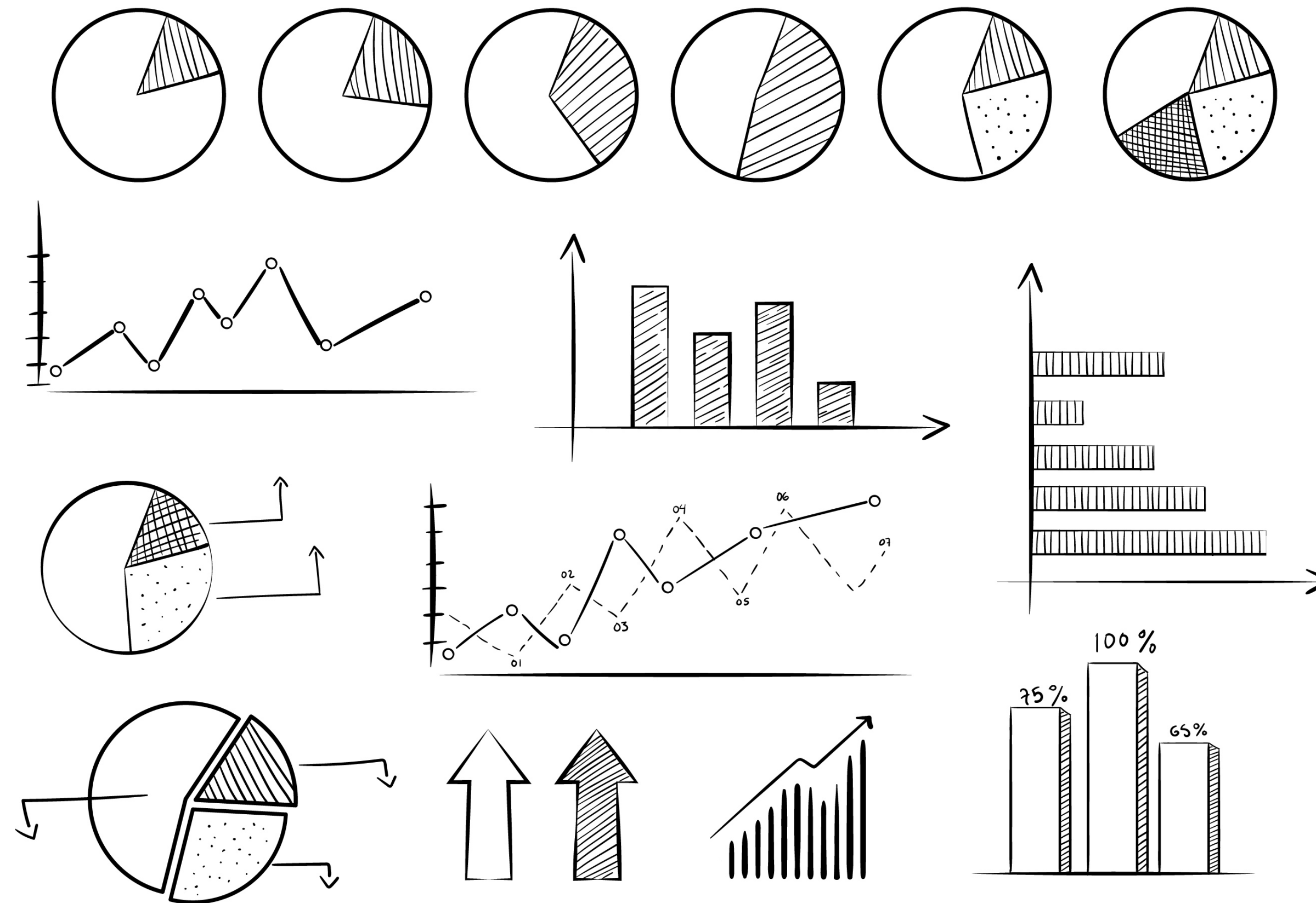


SwiftUI CHARTS

CocoaHeads Aachen, 2023-05-25
cocoaheads.de

DATA-VISUALIZATION

in a declarative way



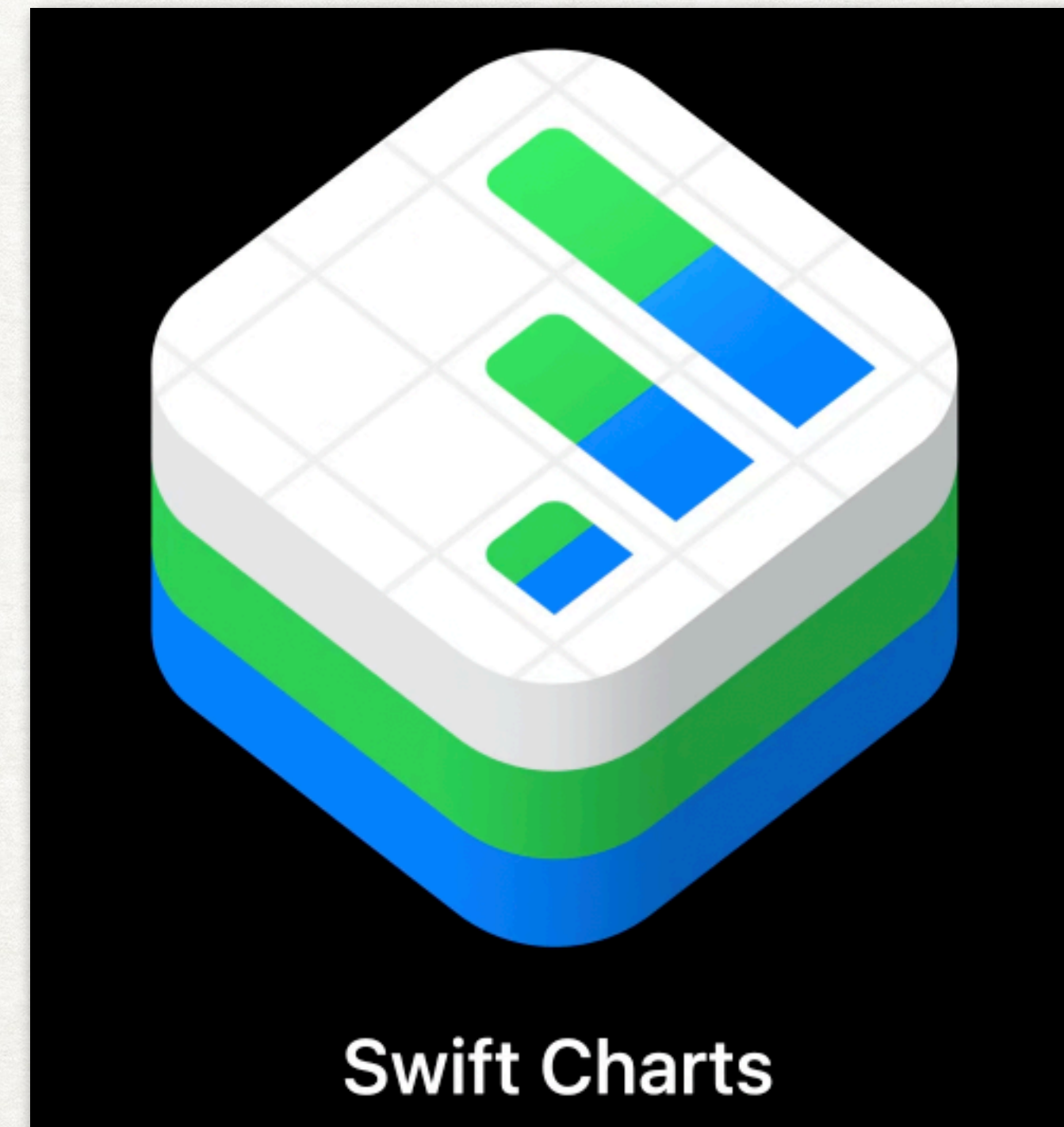
1. OVERVIEW
2. HOW TO
3. HANDS ON

OVERVIEW

SWIFT CHARTS

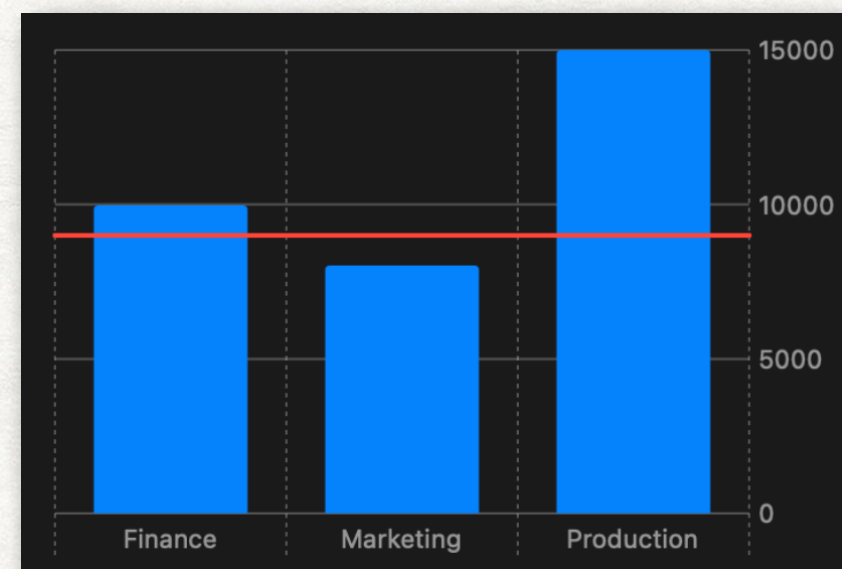
SWIFTUI FRAMEWORK

- Several marks
- Customisable scales, axes, and legends
- Animatable, accessible and dynamic

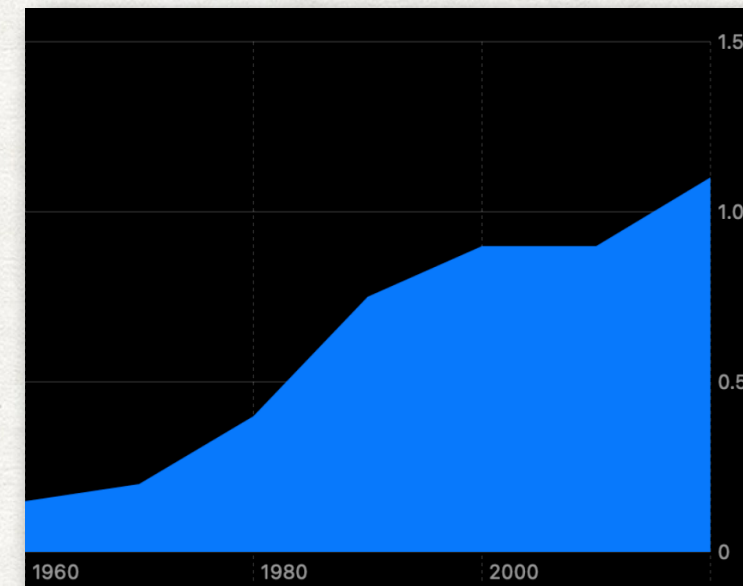


OVERVIEW

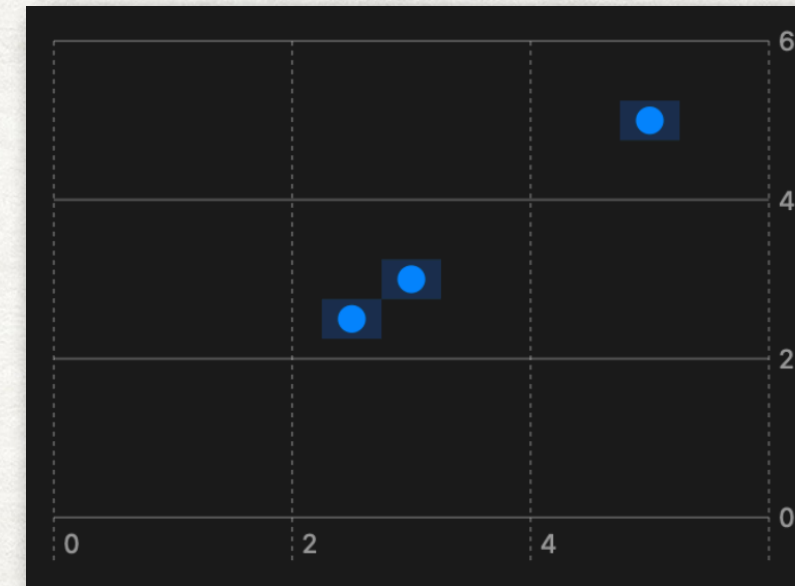
EXAMPLES - MARKS



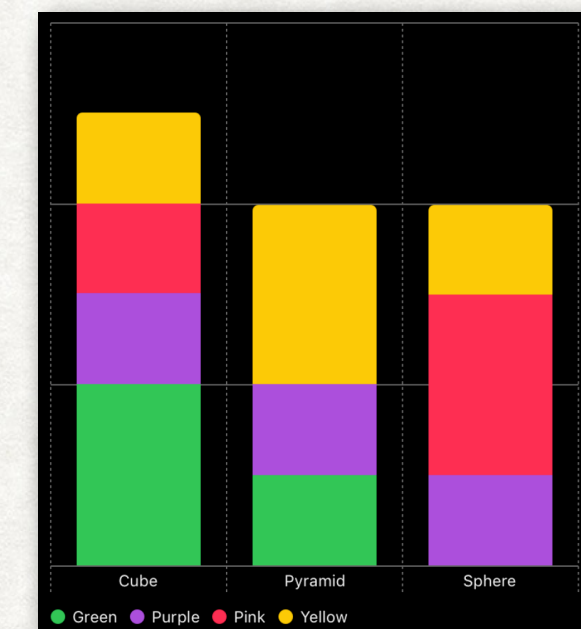
Bar / Ruler



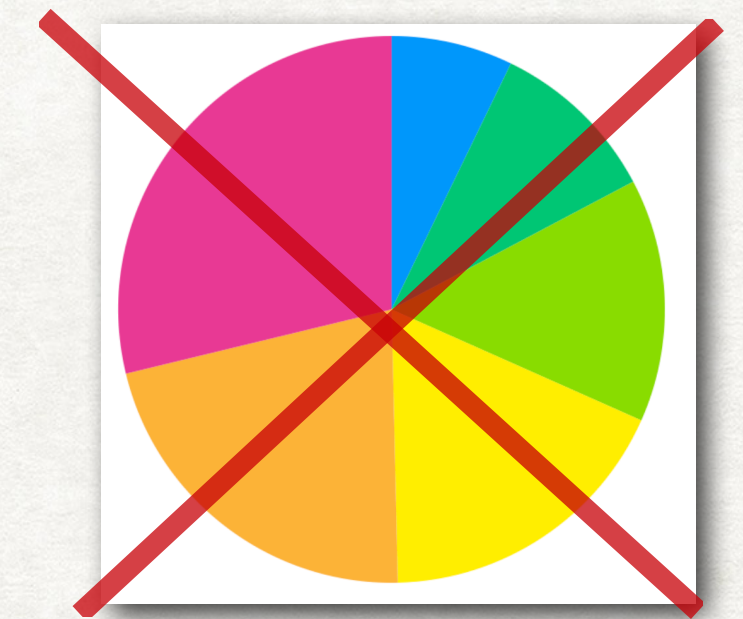
Area



Points



Stacked bars



Piechart 🤔

OVERVIEW

DATA

PLOTTABLE PROTOCOL

- Define marks with any value that conforms to Plottable protocol
- Default types: Int, String, Double, Date and Decimal
- Own types to the possible

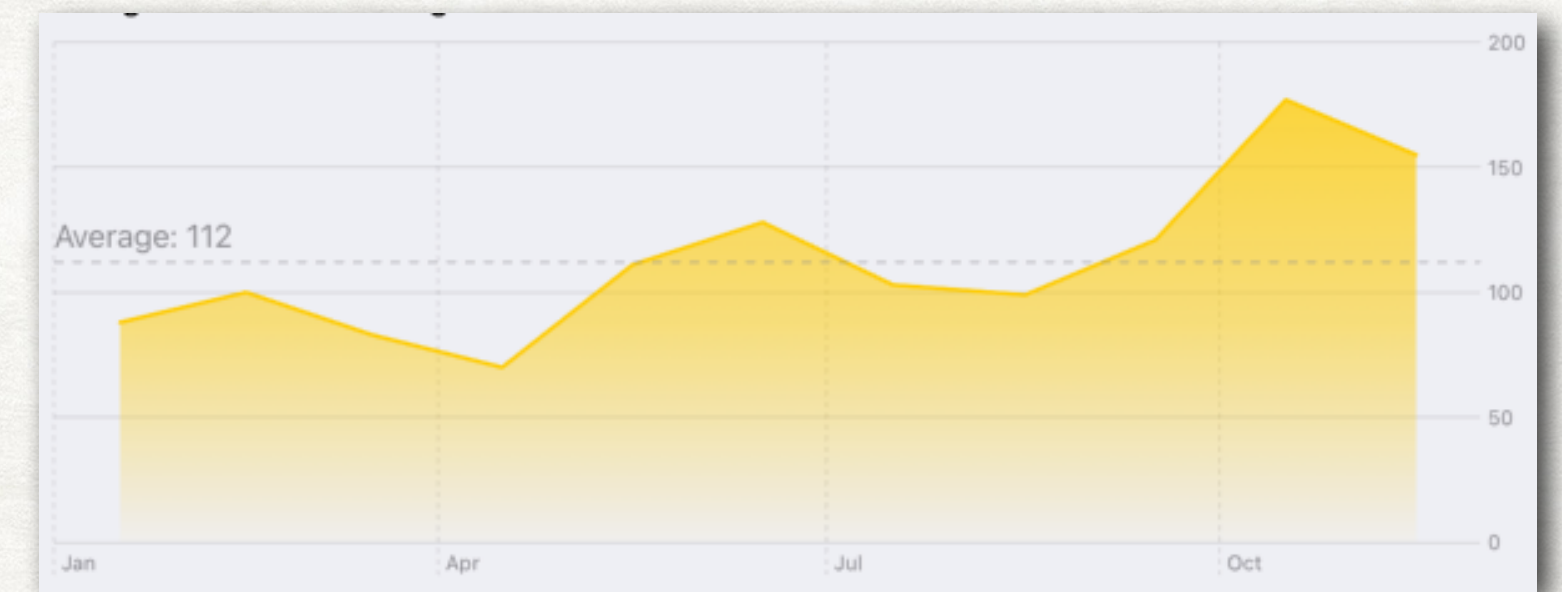
DATA TYPES

- Quantitative (e.g. Double ... number of attendees)
- Categorical (e.g. String ... country)
- Temporal (e.g. Date ... duration)

HOW TO MARKS STYLING

- `foregroundStyle(by:)`
- `interpolationMethod()`
- `lineStyle(by:)`
- `position(by: , axis: )`
- `symbol(by:)`
- `symbolSize()`

```
.foregroundStyle(yellowGradient)  
.lineStyle(StrokeStyle(dash: [5]))
```



```
.position(by: .value("Name", data), axis: .vertical)
```

```
.position(by: .value("Name", data), axis: .horizontal)
```



```
.symbol() {  
  Rectangle()  
  .fill(.red)  
  .frame(width: 10, height: 10)  
}
```



HOW TO INTERPOLATION METHODS

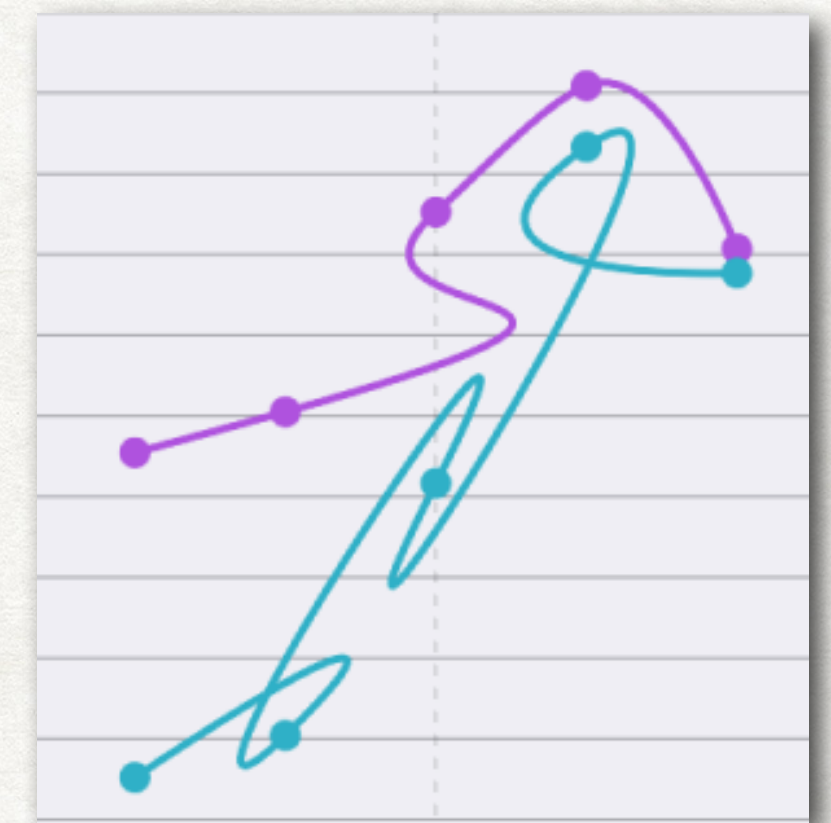
- Linear
- Monotone
- Step-start / Step-end / Step-centre
- Cardinal
- Catmul-rom



step-start / step-end



linear / monoton



cardinal / catmul-rom

HOW TO AXES

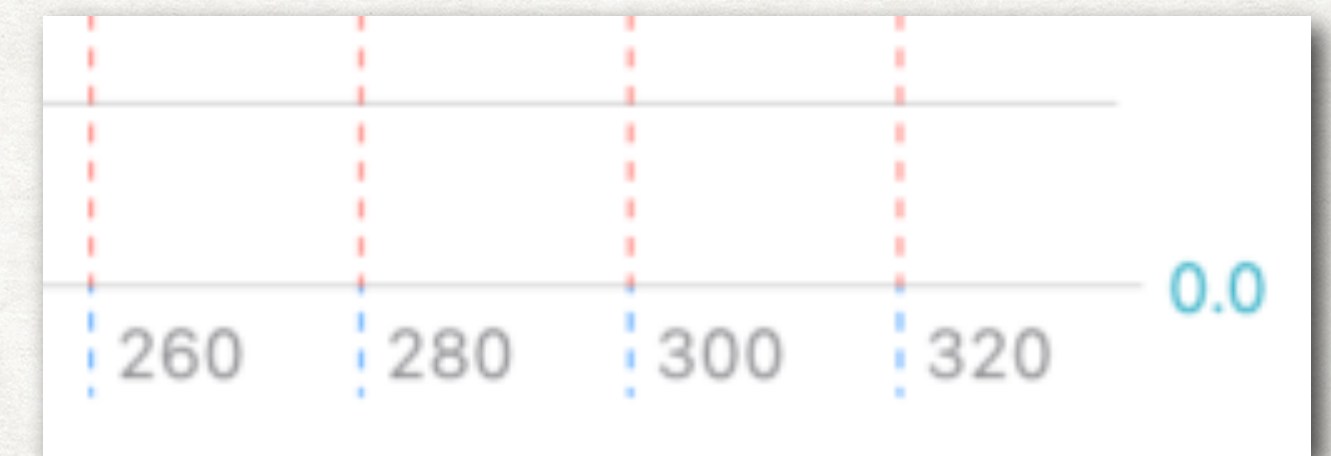
- `.chartXAxis/.chartYAxis(content:)` modifier
 - Takes one or more `AxisMarks` instances or `.hidden`
 - With `.automatic(desiredCount:roundLowerBound:roundUpperBound:)` or specific values or individual content

```
.chartYAxis { AxisMarks(values: .automatic(desiredCount: 3)) }  
.chartYAxis { AxisMarks(values: [0, 50, 100]) }
```

- Customize with composition of stylable elements
 - `AxisGridLine`
 - `AxisTick`
 - `AxisValueLabel`

```
AxisGridLine()  
  .foregroundStyle(.red)
```

```
AxisTick()  
  .foregroundStyle(.blue)
```



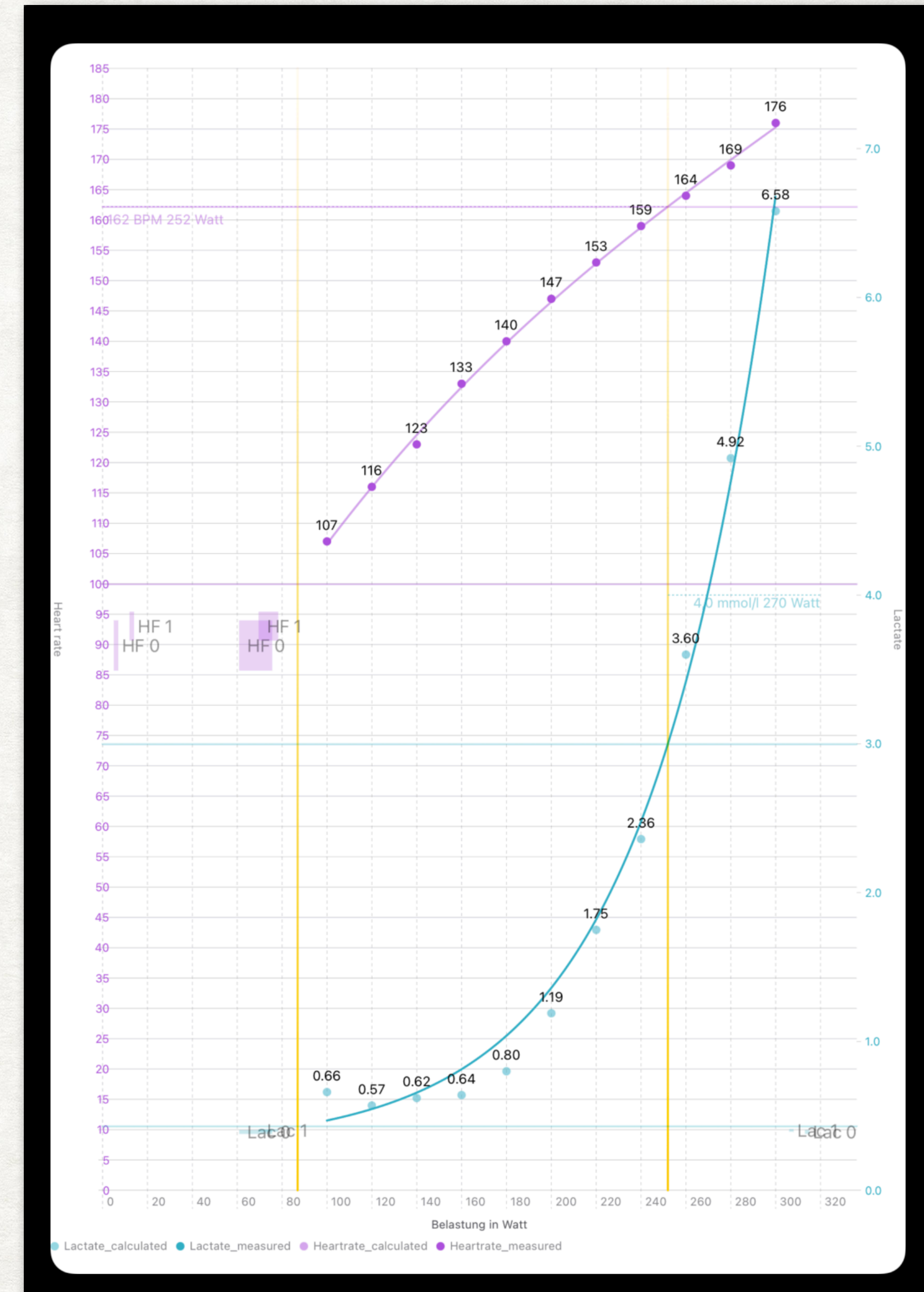
HOW TO SCALES

- Scale = mapping abstract data to mark properties (e.g. coordinates in screen space)
- Input values often “scaled” by some factor to screen coordinates
- Swift Charts infers scales automatically from given data
- All scales can be customised, e.g.:
 - `.chartYScale(domain:type:) modifier`
`.chartYScale(domain: [0, 100])`
 - `.chartForegroundStyleScale(mapping:) modifier`
`.chartYScale(["City A": .orange, "City B": .blue])`

HOW TO ANNOTATIONS

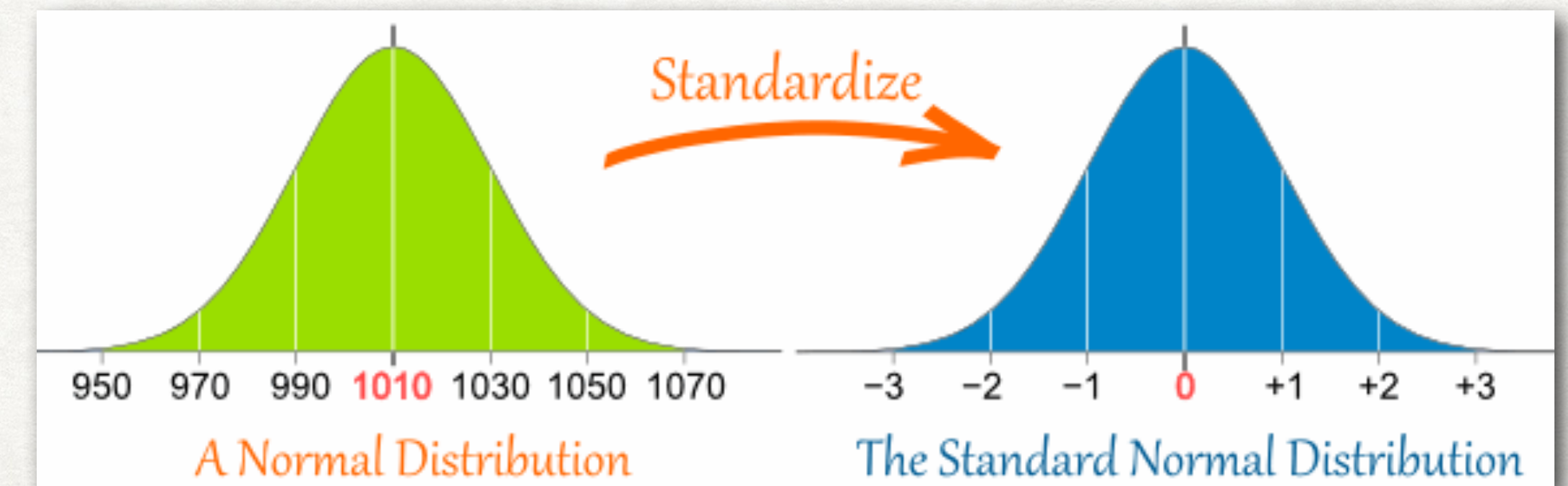
- annotation modifier
 - Annotates a mark or collection of marks with a view positioned relative to its bounds
- chartPlotStyle modifier
 - Returns the content of whole the plot area (e.g. for styling purposes)

```
.chartPlotStyle { plotArea in
    plotArea.frame(height: 130)
    .background(.pink.opacity(0.2))
    .border(.pink, width: 1)
}
```

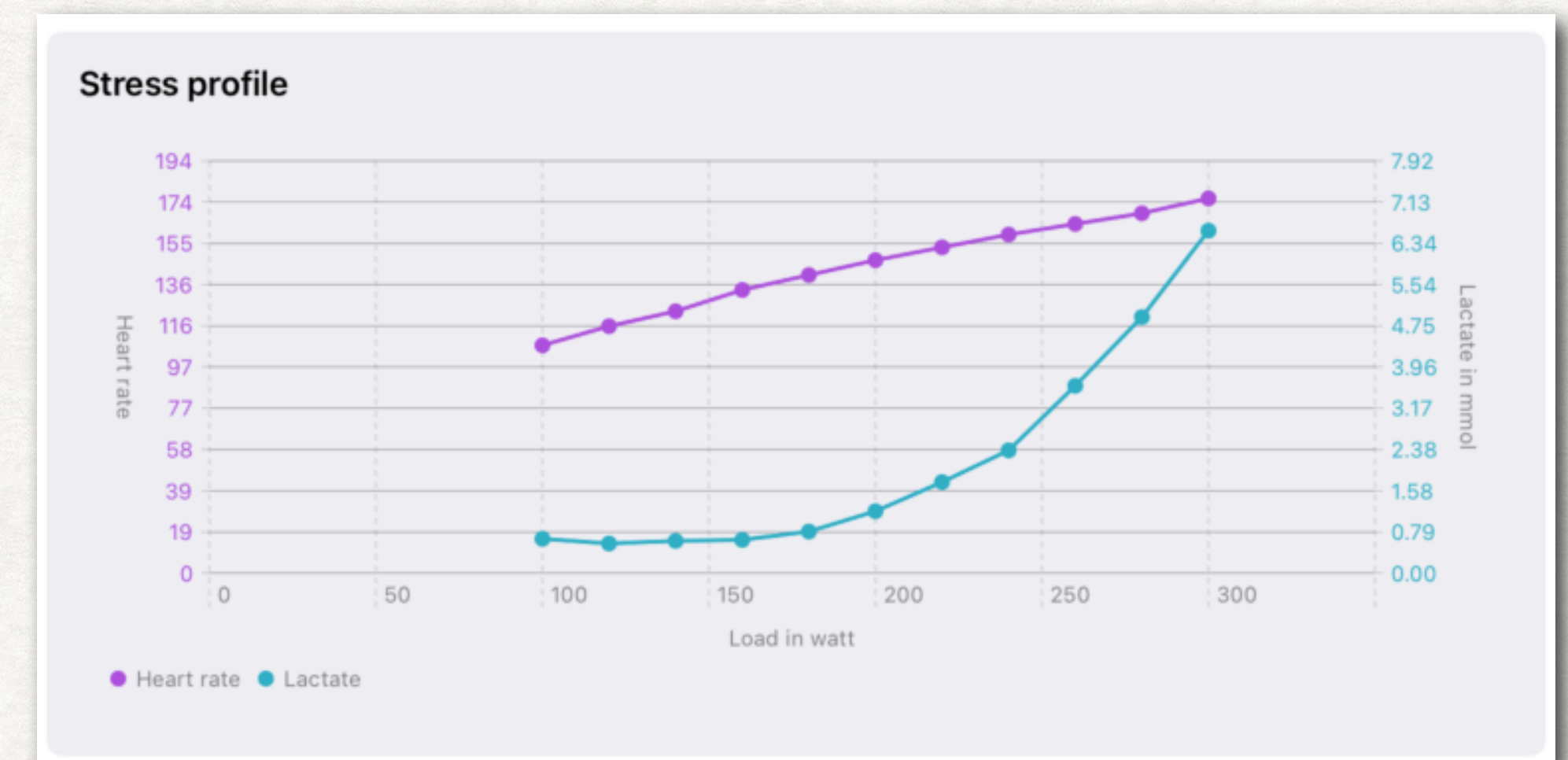


HOW TO DATA PREPROCESSING

- Standardisation
mean value = 0
 $x' = (x - \text{mean}) / \text{standard_deviation}$
- Min/Max Normalisation
all values between 0 ... 1
 $x' = (x - \text{min}) / (\text{max} - \text{min})$



<https://www.mathsisfun.com/data/standard-normal-distribution.html>



HANDS ON...

HANDS ON

LETS BUILD A CHART

Date	Attendees	Pizza
12.01.2023	107	NO
14.02.2023	116	NO
12.03.2023	123	NO
14.04.2023	133	NO
12.05.2023	140	NO
14.06.2023	147	NO
12.07.2023	153	YES
14.08.2023	159	NO
12.09.2023	164	NO
14.10.2023	169	NO
12.11.2023	176	YES

CocoaHeads Aachen

Date	Attendees	Pizza
22.01.2023	88	YES
24.02.2023	100	YES
10.03.2023	83	YES
16.04.2023	70	YES
02.05.2023	111	YES
18.06.2023	128	NO
19.07.2023	103	YES
24.08.2023	99	YES
09.09.2023	121	YES
11.10.2023	177	NO
18.11.2023	155	YES

CocoaHeads Cologne

THANK YOU

Stephan Partzsch

Mostly correct

Rarely wrong

Consistently authentic



Gerd Bernhardt

No Blog

No Twitter

No Facebook

No Instagram

but social

<https://developer.apple.com/documentation/charts/chart/>

<https://developer.apple.com/wwdc22/10137>