GGPlot Customization

Marco Torchiano
Version 1.0.1 - April 2021

License

Introduction

Sample data

```
knitr::kable(courses)
```

code	course	sem	credits	lecture_hours	study_hours
15AHM	Chemistry	1	8	80	120
12BHD	Computer science	1	8	80	120
16ACF	Calculus I	1	10	100	150
01PNN	Free Credits	2	6	60	90
01RKC	Linear Algebra	2	10	100	150
17AXO	Physics I	2	10	100	150

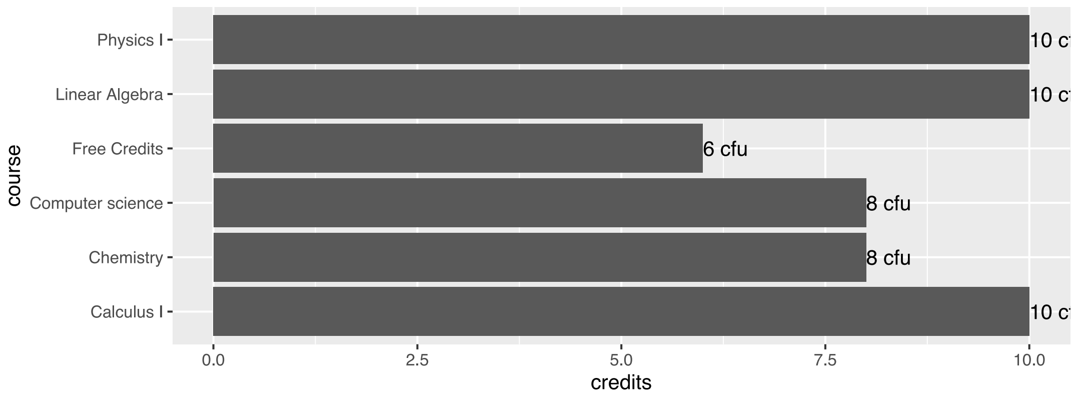
Aesthetics

Aesthetics can be defined:

- At plot level (`ggplot()`)
  - by mapping (`aes()`) to data
- At layer level (`geom_..()`)
  - by mapping (`aes()`) to date
  - at a fixed value (not scaled)

Aesthetics mapping at layer level

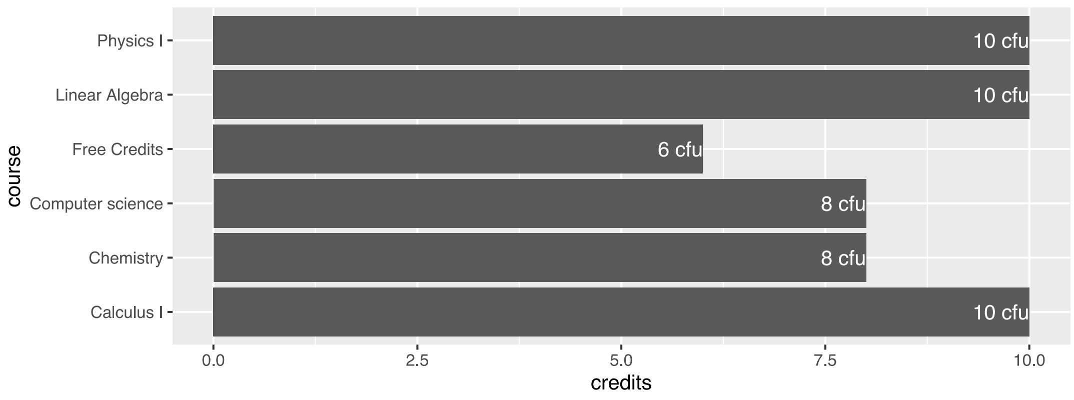
```
ggplot(courses,aes(y=course,x=credits))+
  geom_bar(stat="identity")+
  geom_text(aes(label=paste(credits,"cfu")),hjust=0)
```



Fixed aesthetics at layer level

No legend is shown for the fixed value.

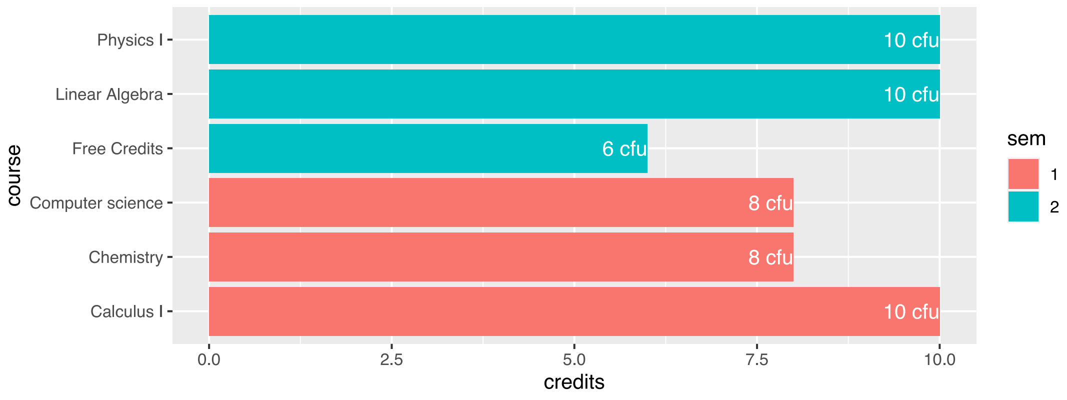
```
ggplot(courses,aes(y=course,x=credits))+
  geom_bar(stat="identity")+
  geom_text(aes(label=paste(credits,"cfu")),
            color="white", hjust=1)
```



Aesthetics at layer level

Aesthetics mapped to data (`aes()`) are scaled and produce a legend (*guide*)

```
ggplot(courses,aes(y=course,x=credits))+
  geom_bar(aes(fill=sem),stat="identity")+
  geom_text(aes(label=paste(credits,"cfu")),
            hjust=1, color="white")
```



Multiple data series

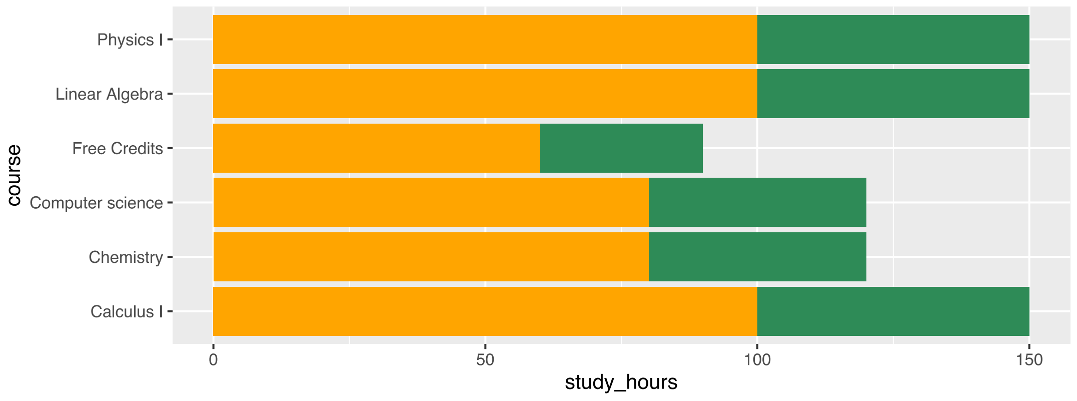
1. series are stored as distinct variables,
   - each variable is mapped in a different layer
2. series are stored as the same *value* variable, and another *type* variable tells them apart – usualy factor –
   - a single layer is used and series are separated by
     - mapping *type* variable to a visual or `group`, or/and
     - *faceting* (small multiples) by *type*

Case 1 may be trasformed into case 2 to via `pivot_longer()` and viceversa through `pivot_wider()`

Distinct variables

- layers are overlapped and independent, so upper (later layers) cover lower ones
- no legend is produced (because `fill` aesthetics is not scaled)

```
ggplot(courses,aes(y=course))+
  geom_bar(aes(x=study_hours),stat="identity",fill="seagre
  geom_bar(aes(x=lecture_hours),stat="identity",fill="oran
```



Type and value variables (data)

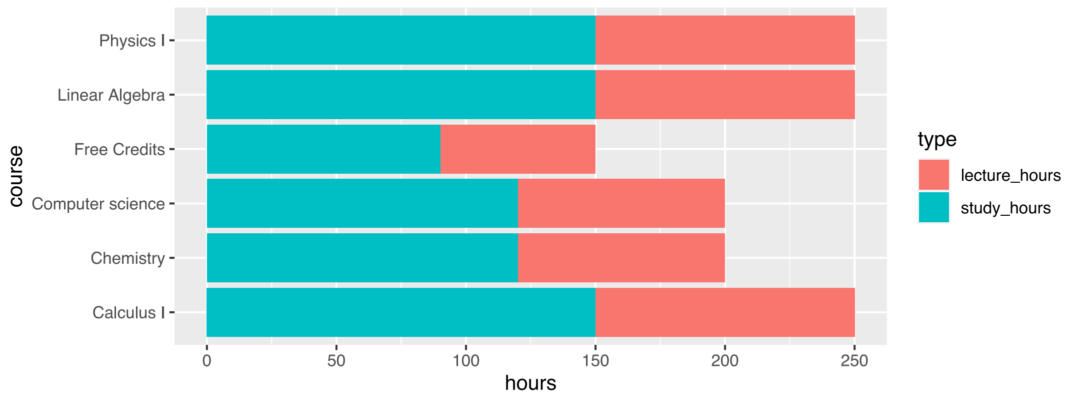
```
courses_long <- courses %>% pivot_longer(ends_with("_hours
knitr::kable(courses_long)
```

code	course	sem	credits	type	hours
15AHM	Chemistry	1	8	lecture_hours	80
15AHM	Chemistry	1	8	study_hours	120
12BHD	Computer science	1	8	lecture_hours	80
12BHD	Computer science	1	8	study_hours	120
16ACF	Calculus I	1	10	lecture_hours	100
16ACF	Calculus I	1	10	study_hours	150
01PNN	Free Credits	2	6	lecture_hours	60
0.. NN	Free Credits	2	6	study_hours	90

Type and value variables

- unique layers, bars are automatically stacked not to overlap
- a legend is produced by the (implicit) `scale_fill_discrete`

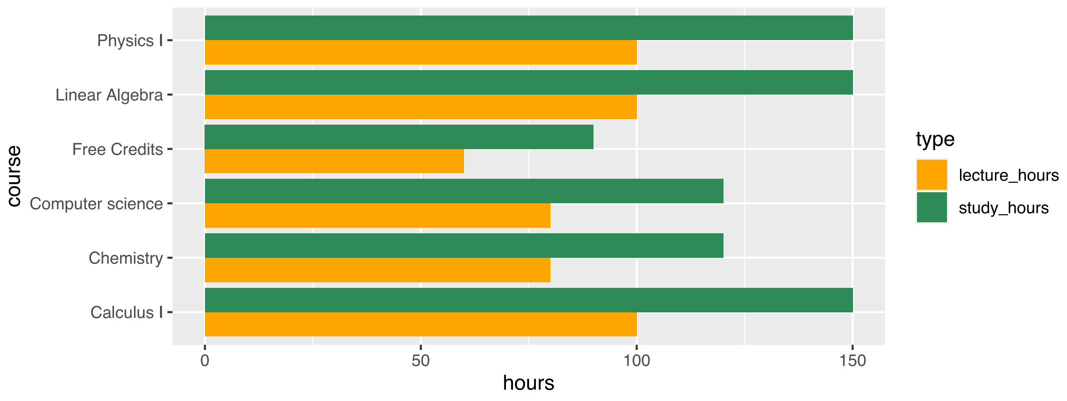
```
ggplot(courses_long, aes(y=course,x=hours,fill=type))+
  geom_bar(stat="identity")
```



Type and value variables (dodging)

- unique layers, bars are explicitly *dodged*
- color scaled by `scale_fill_manual`

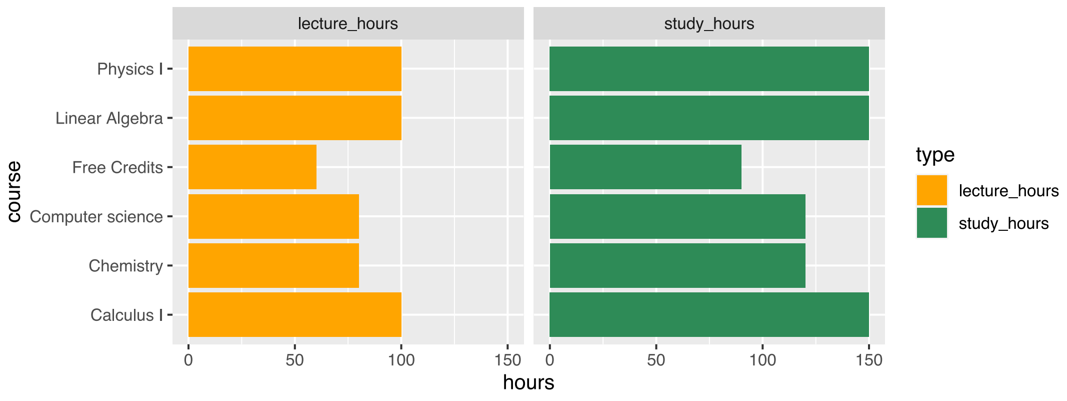
```
ggplot(courses_long, aes(y=course,x=hours,fill=type))+
  geom_bar(stat="identity",position="dodge") +
  scale_fill_manual(values=c("orange","seagreen"))
```



Faceting multiple series

Each facet contains a separate series

```
ggplot(courses_long, aes(y=course,x=hours,fill=type))+
  geom_bar(stat="identity",position="dodge") +
  scale_fill_manual(values=c("orange","seagreen")) +
  facet_wrap(type~.)
```



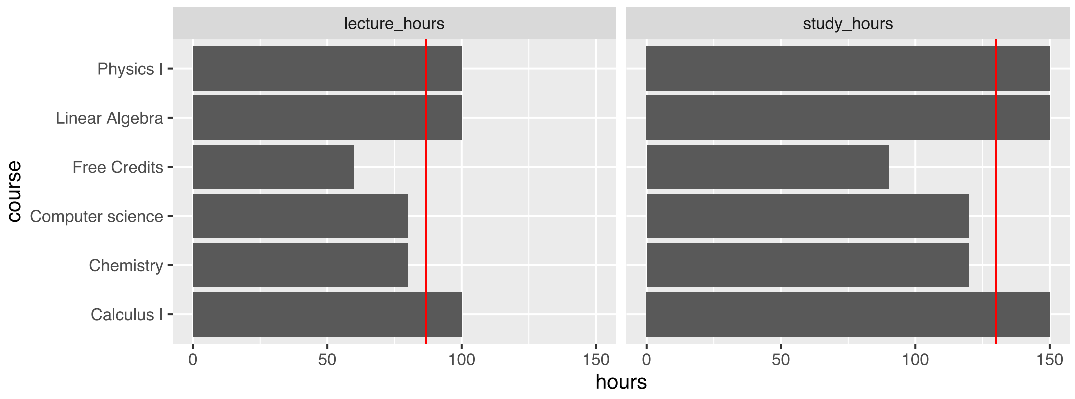
Data

Data can be defined:

- At plot level (`ggplot()`)
- At layer level (`geom_..()`)
    - allow adding other data

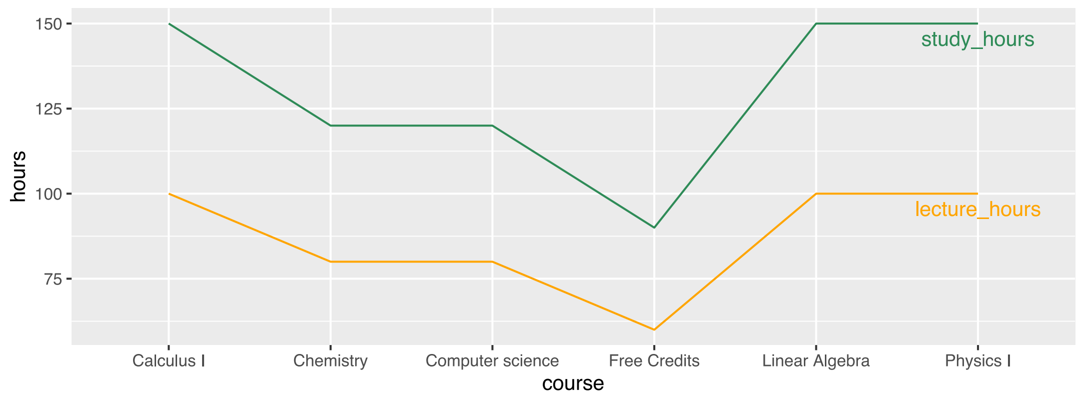
Layer specific data

```
average <- courses_long %>% group_by(type) %>%
           summarize(hours=mean(hours))
ggplot(courses_long, aes(y=course,x=hours))+
  geom_bar(stat="identity",position="dodge") +
  geom_vline(aes(xintercept=hours),data=average,color="red
  facet_wrap(type~.)
```



Direct labeling with layer data

```
rightmost <- courses_long %>% group_by(type) %>%
          summarize(course=last(course),hours=last(hours))
ggplot(courses_long, aes(x=course,y=hours,color=type))+
  geom_line(aes(group=type))+
  scale_color_manual(values=c("orange","seagreen"),guide=F
  geom_text(aes(label=type),data=rightmost,
            hjust=0.5,vjust=1.5,show.legend=FALSE)
```



Geometry layers

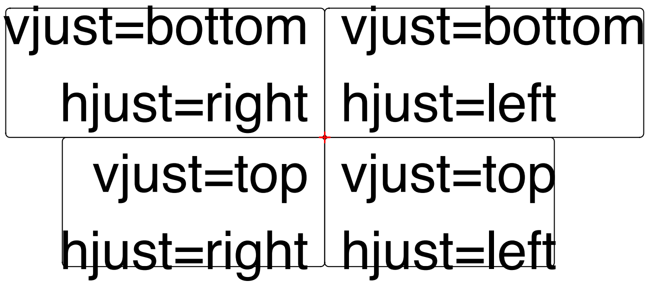
Text annotations

Text annotation layer is created with `geom_text()` and `geom_label()` with the following aesthetics

- `label`: the text label
- `hjust` : horizontal alignment
    - `"left"`, `"middle"`, `"right"`
- `vjust`: vertical alignment
    - `"top"`, `"center"`, `"bottom"`
- `family`: font family
- `fontface`: type of font, e.g. “bold”, “italic”

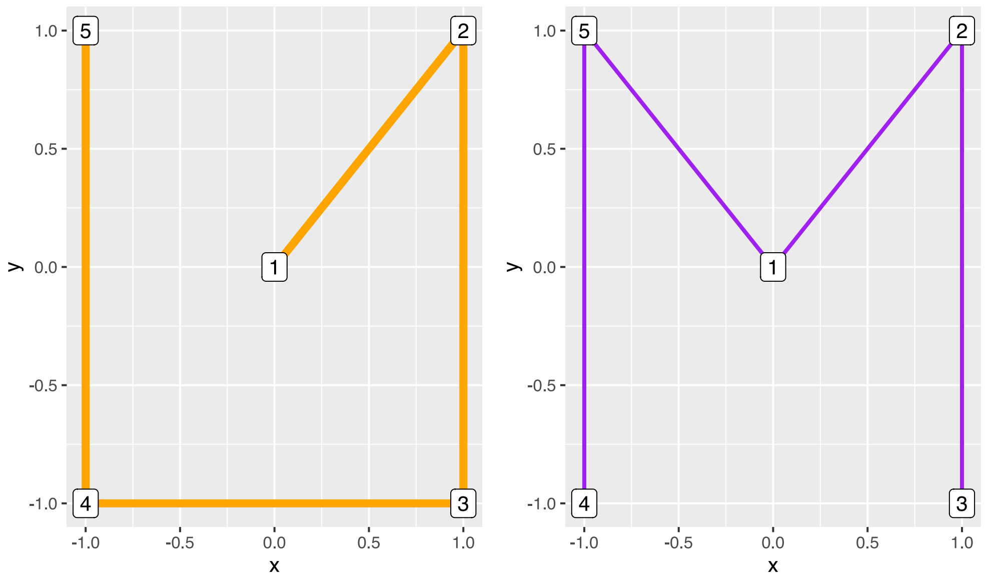
Text annotations alignment

Note: e.g., `"left"` means align *to the left* of the reference position



Paths vs. Line

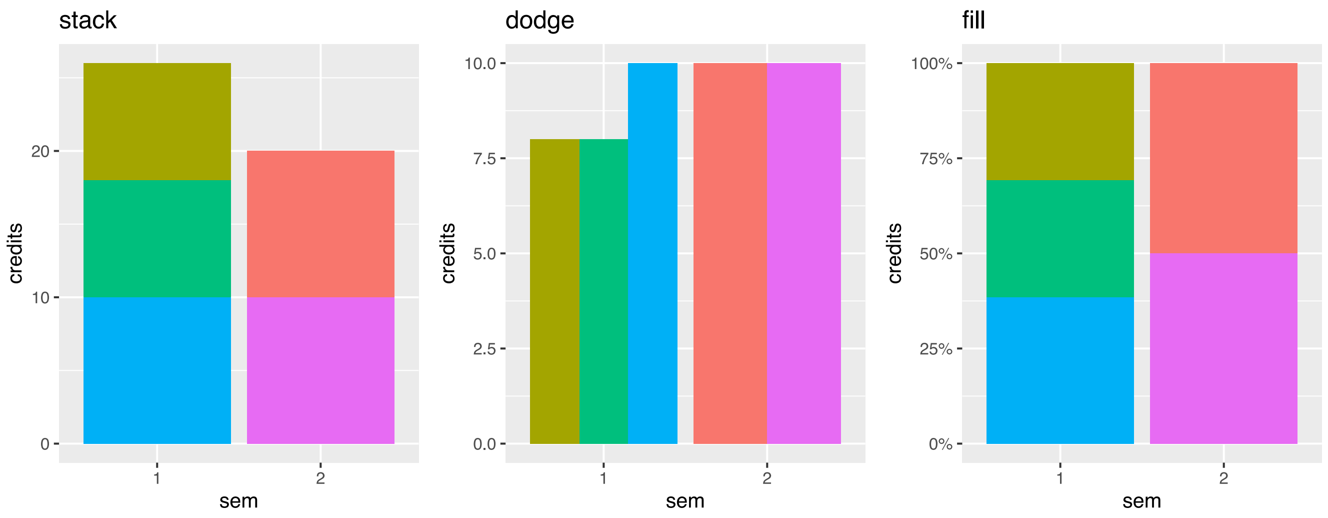
- Path obeys the order of points in the data
- Line sorts the points by x



Barplots

Use `geom_bar(stat="identity")`, with aethetics:

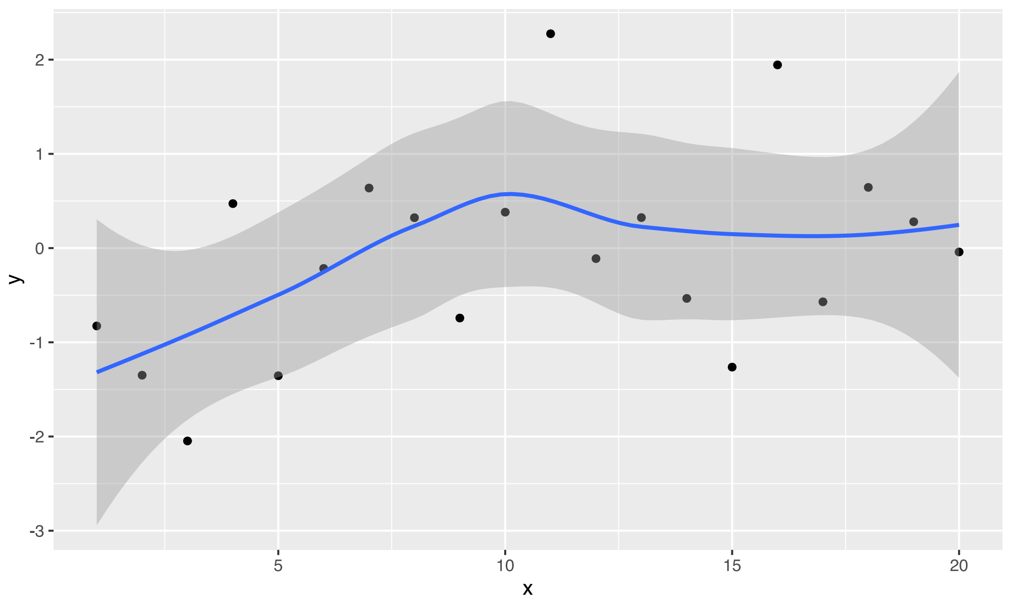
- `position`: `"stack"`, `"dodge"`, `"fill"`



Smoothing

`geom_smooth()` provides a least square interpolation of points.

```
set.seed(1793); data.frame(x=1:20,y=rnorm(20)) %>%
  ggplot(aes(x,y))+geom_point()+geom_smooth()
```



Scales

Position scales

Position scales are `scale_`*`x/y`*..

- `_continuous()` linear, with variations:
  - `_log10()`
  - `_sqrt()`
  - `_reverse()`
- `_discrete()`
- `_date()` with variations
  - `_datetime()`
  - `_time()`

Position scales

Parameters:

- `name` the name/label of the axis
- `breaks` the breaks (ticks) of the axis, a vector or a function
  - also `minor_breaks` for continuous and date scales
  - also `date_breaks` define distance between breaks, e.g. `2 days`
- `labels` the labels, a vector or a function
  - can use `label_` functions in `scales` package e.g. `label_percent()`

Position scales

- `limits`: define the limits (or list of values for discrete scale)
- `expand`: define how much space is added at extremes of axis
  - use function `expansion()`
    - `mult`: multiplicative expansion, default: 0.05 for continuous
    - `add`: additive expansion, default: 0.6 for discrete
    - both can be 1 or 2 elements long

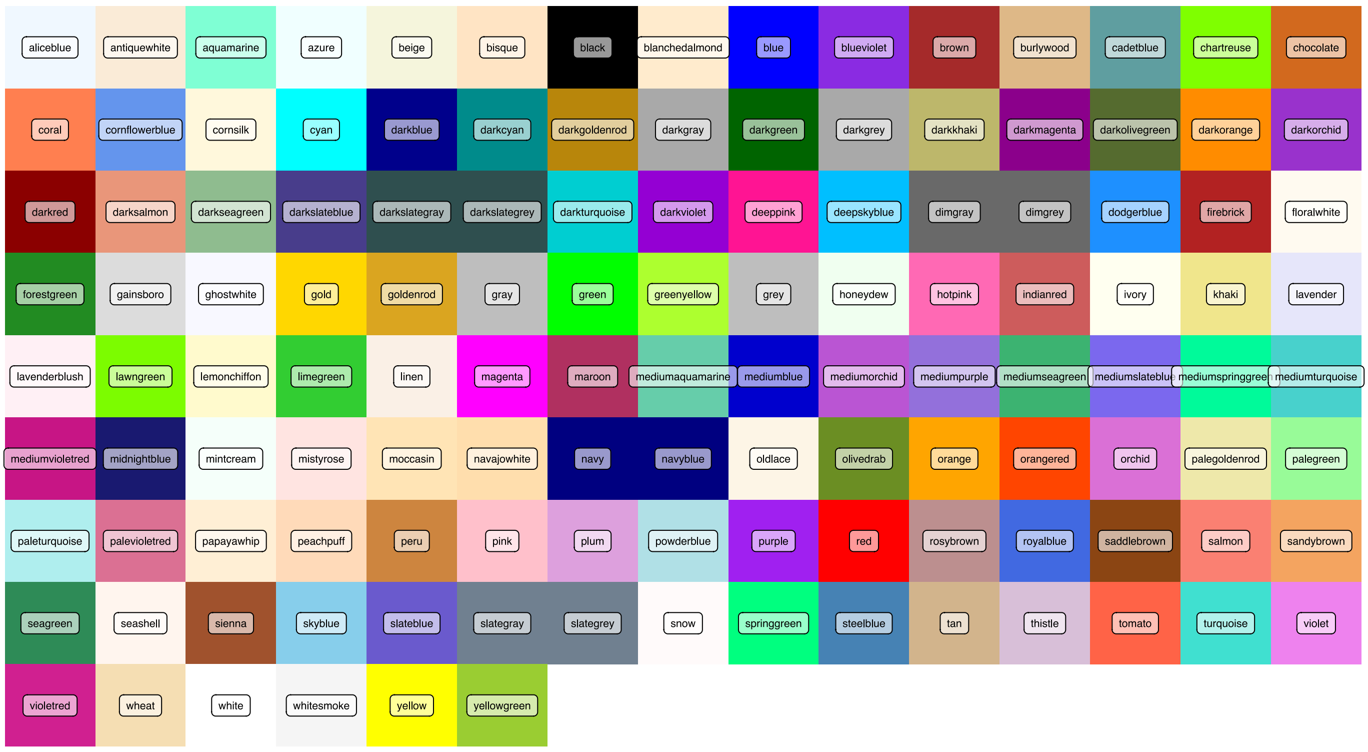
Color scales

Works for both `color` and `fill` aesthetics.

- `_manual`, assign levels to color names in `values`
- `_gradient`, define a color gradient from `low` to `high`
- `_brewer` picks from a *colorbewer* predefined palette
  - `type`: `"seq"`, `"div"`, `"qual"`
  - `palette`: names (e.g., “Greens”) or number
- `_viridis_`*b*|*c*|*d* binned,continuous, discrete predefined palettes
  - `option`: “A” to “E”

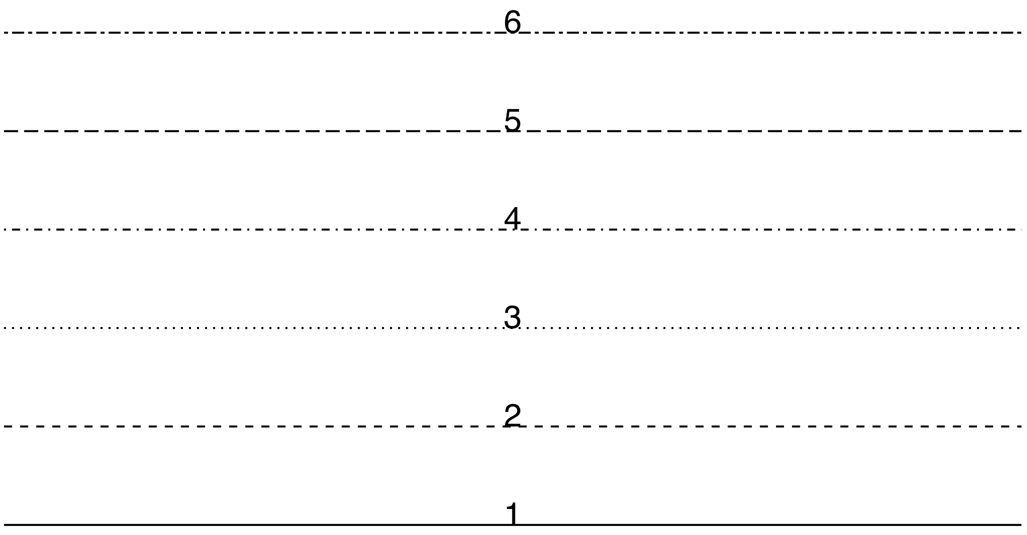
See documentation for further details

Color names (a sample)



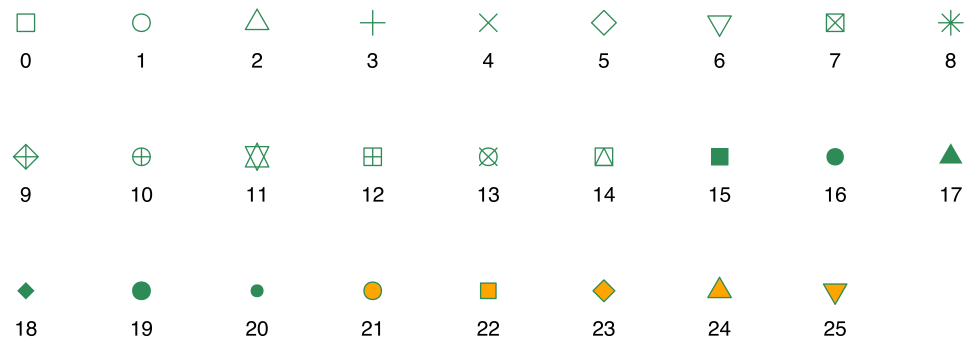
Unscaled aethetics (linetype)

- `linetype` type of line for `geom_line()` and `geom_path()`



Unscaled aethetics (shape)

- Maps to *at most* 6 different levels of a discrete variable
- Symbols can be selected with `scale_shape_manual()`
    - 0 to 14 are outlined shapes
    - 15 to 20 are solid shapes
    - 21 to 25 are filled shapes



References

- Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen. “ggplot2: Elegant Graphics for Data Analysis”, in-prograss
    - https://ggplot2-book.org/
- Winston Chang, “R Graphics Cookbook” O’Reilly, 2013