Variations on Line Plots

Diardano Raihan

2022-07-21

Import the Libraries and Dataset

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

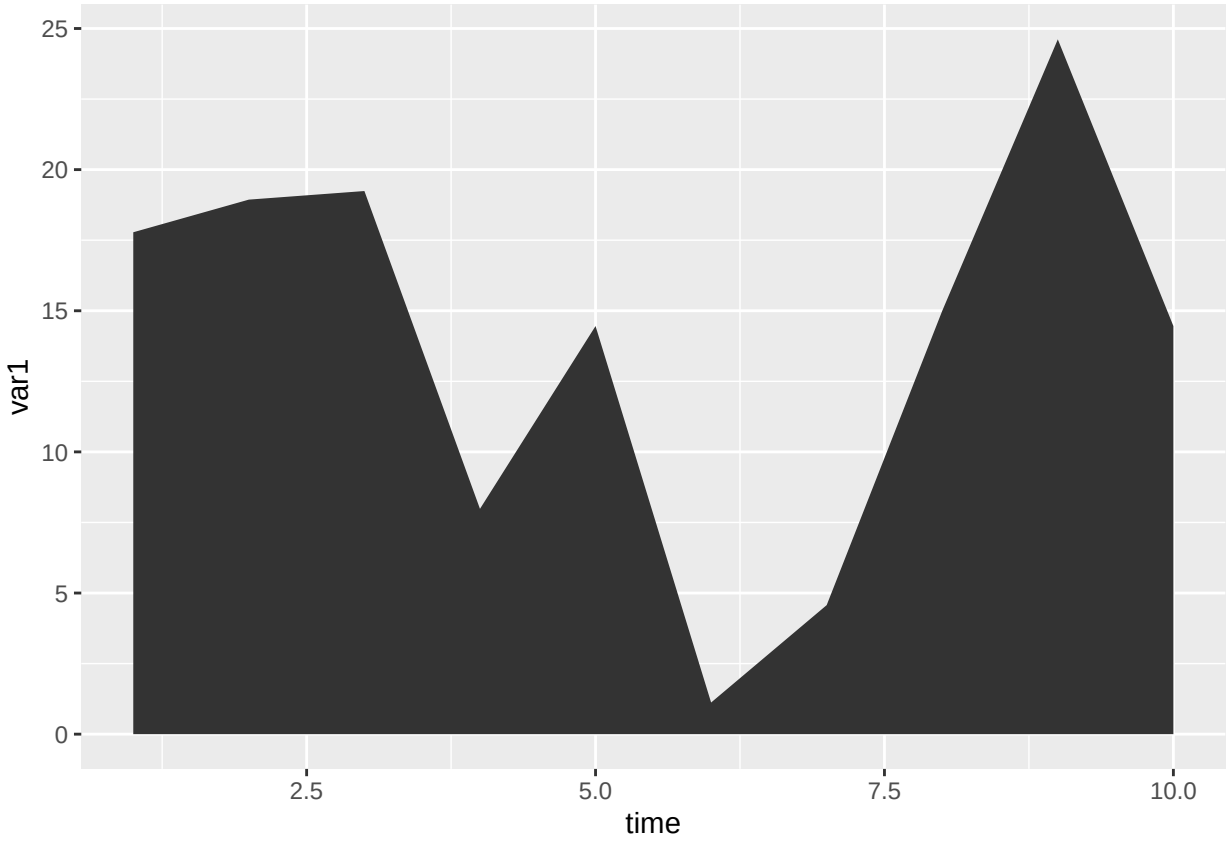
```
library(knitr)
library(ggthemes)
library(RColorBrewer)
```

Variations on Line Plots

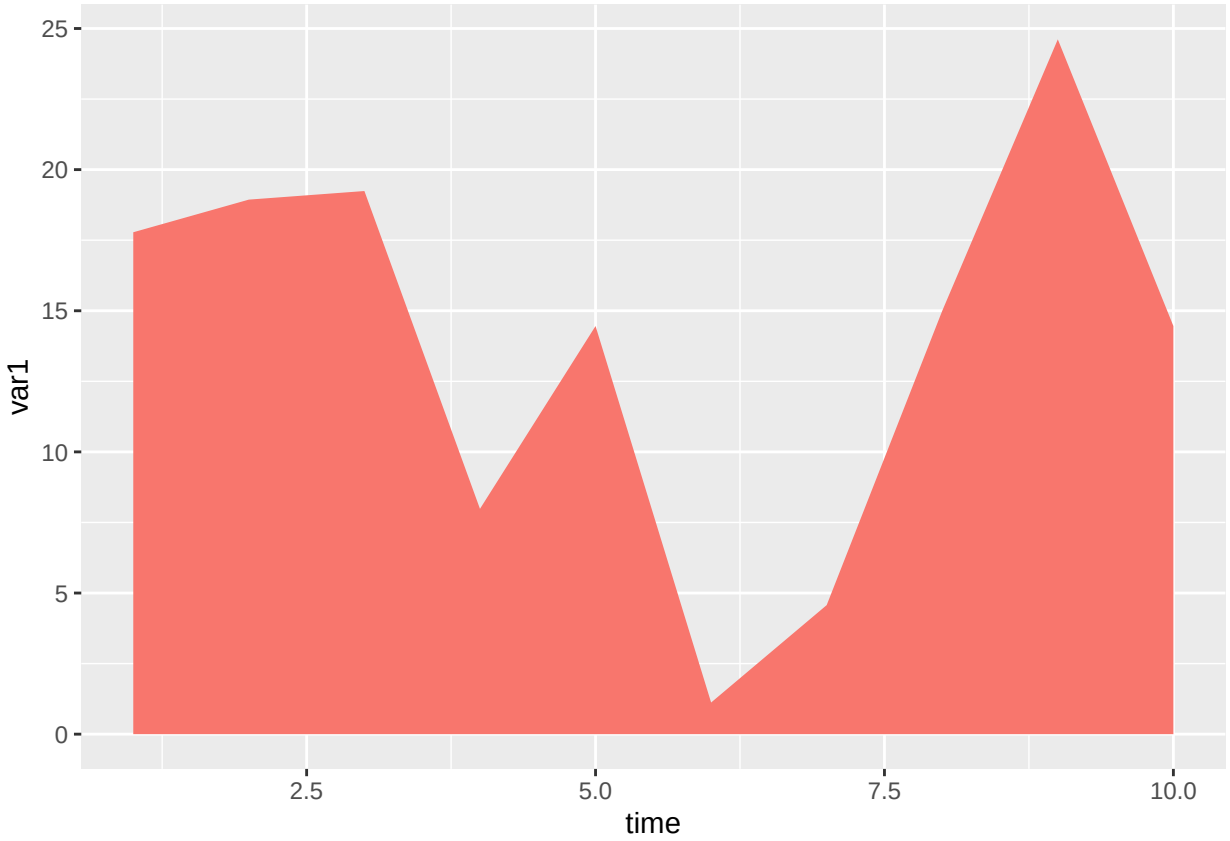
Area Plots

- You might think that you should use a fill aesthetic and geom_line(). It turns out that the best way to do this is by using geom_area().
- reference: https://r-graphics.org/recipe-line-graph-area

```
var1 <- runif(10, 0, 25)
time <- seq(1, 10)

df <- tibble(var1, time)

# just filling in the space under the line
ggplot(df, aes(x = time, y = var1))+
  geom_area()
```




```
# a litle aesthetic tweaking
ggplot(df, aes(x = time, y = var1, fill = 'red'))+
  geom_area()+
  guides(fill = "none")
```



A Stacked Line Graph

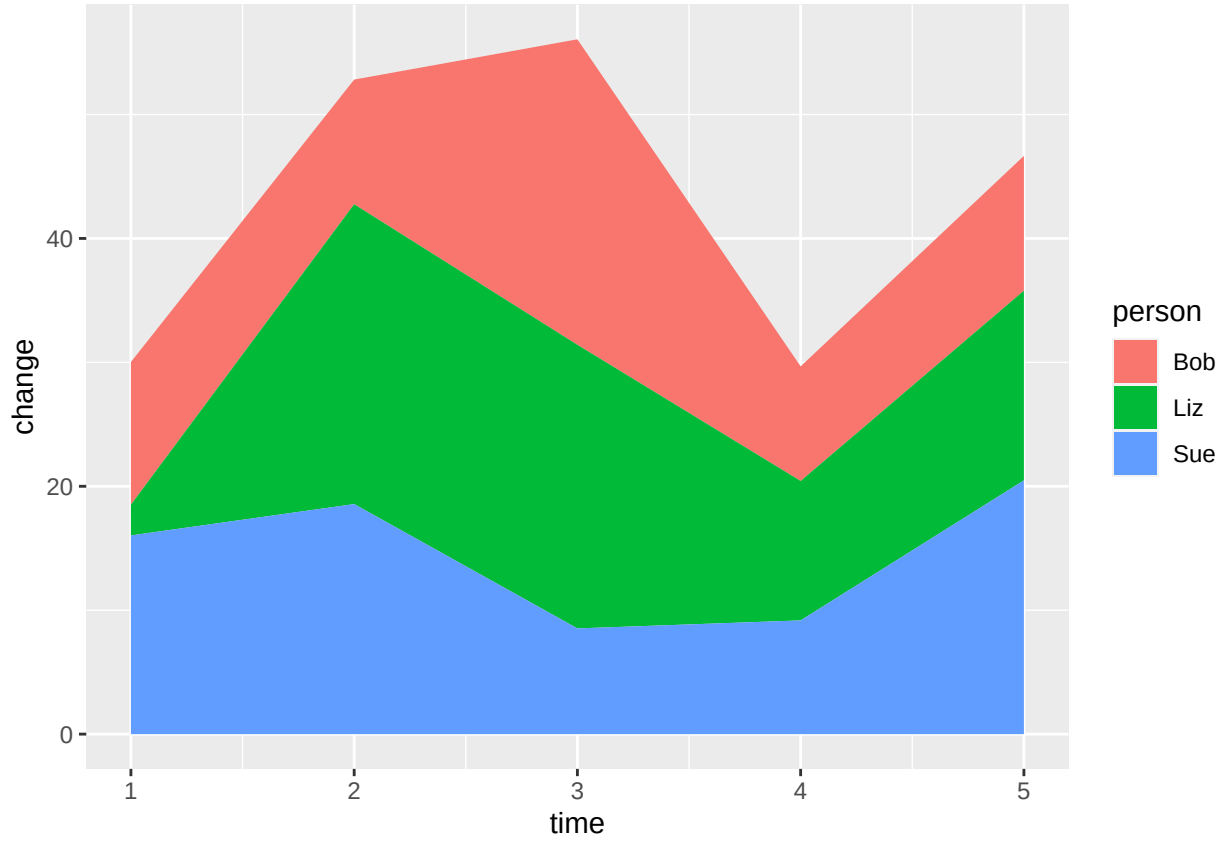
- Adapted form https://r-graphics.org/recipe-line-graph-stacked-area

```
# Create fake data for three people at three different time points
Bob <- tibble(person = rep("Bob", 5), time = seq(1:5), change = runif(5, 0, 25))
Sue <- tibble(person = rep("Sue", 5), time = seq(1:5), change = runif(5, 0, 25))
Liz <- tibble(person = rep("Liz", 5), time = seq(1:5), change = runif(5, 0, 25))

df <- bind_rows(Bob, Sue, Liz)
df
```

```
## # A tibble: 15 x 3
##    person  time change
##    <chr>  <int>  <dbl>
##  1 Bob        1  11.5
##  2 Bob        2  10.1
##  3 Bob        3  24.7
##  4 Bob        4   9.25
##  5 Bob        5  10.9
##  6 Sue        1  16.0
##  7 Sue        2  18.6
##  8 Sue        3   8.53
##  9 Sue        4   9.18
## 10 Sue        5  20.5
```

```
## 11 Liz        1   2.48
## 12 Liz        2  24.2
## 13 Liz        3  22.9
## 14 Liz        4  11.2
## 15 Liz        5  15.3
```

```
# plot the stacked lineplot
ggplot(df, aes(x = time, y = change, fill = person))+
  geom_area()
```




```
# test your understanding of what the plot is showing
df_wide <- df %>%
  pivot_wider(id_cols = time, values_from = change, names_from = person)
df_wide
```

```
## # A tibble: 5 x 4
##    time   Bob   Sue   Liz
##   <int> <dbl> <dbl> <dbl>
## 1     1 11.5  16.0   2.48
## 2     2 10.1  18.6  24.2
## 3     3 24.7   8.53 22.9
## 4     4  9.25  9.18 11.2
## 5     5 10.9  20.5  15.3
```

```r
# note that the stacked areas add up to the totals
df_wide$total <- df_wide$Bob + df_wide$Sue + df_wide$Liz
df_wide
```

```
## # A tibble: 5 x 5
##    time   Bob   Sue   Liz total
##   <int> <dbl> <dbl> <dbl> <dbl>
## 1     1 11.5  16.0   2.48  30.0
## 2     2 10.1  18.6  24.2   52.8
## 3     3 24.7   8.53 22.9   56.1
## 4     4  9.25  9.18 11.2   29.7
## 5     5 10.9  20.5  15.3   46.7
```

Dumbell Chart

- Adapted from https://rkabacoff.github.io/datavis/Time.html#dummbbell-charts

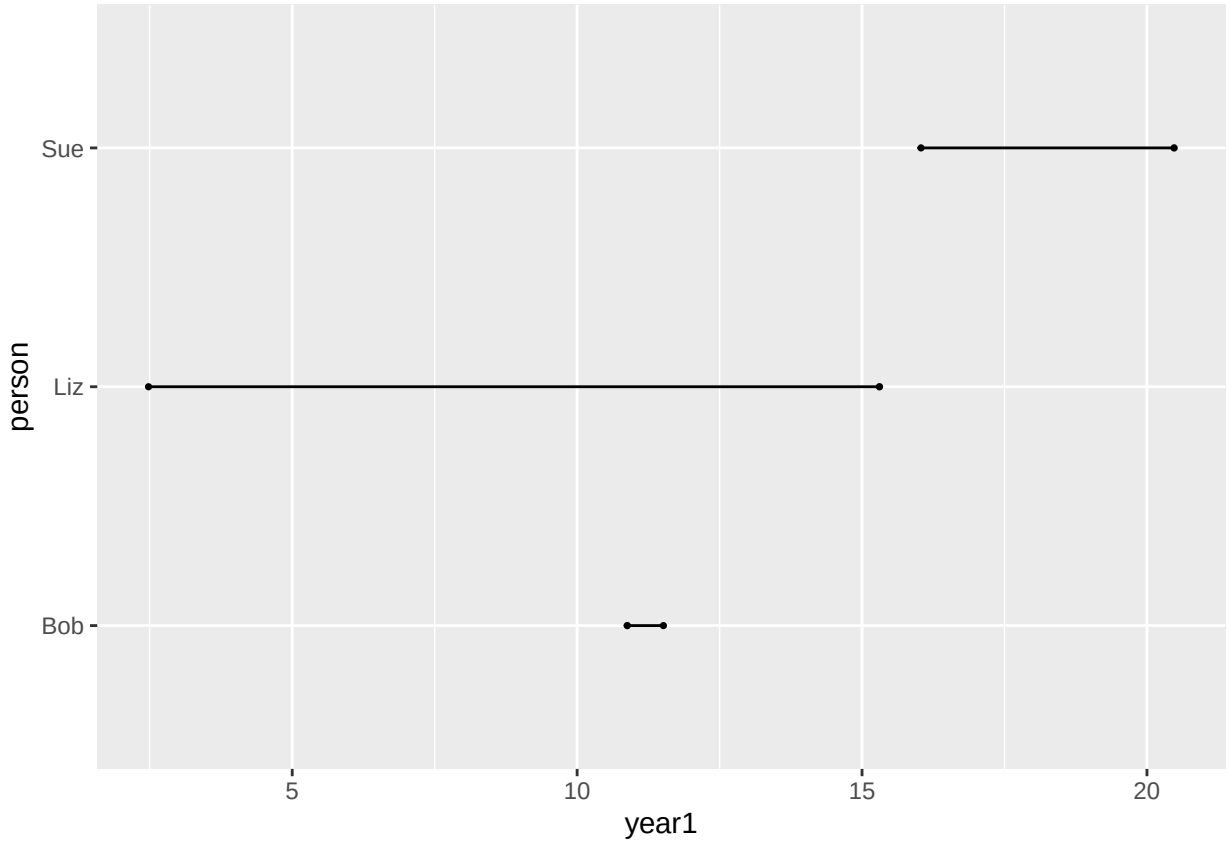
```r
# install.packages("ggalt")
library(ggalt)
```

```
## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2
```

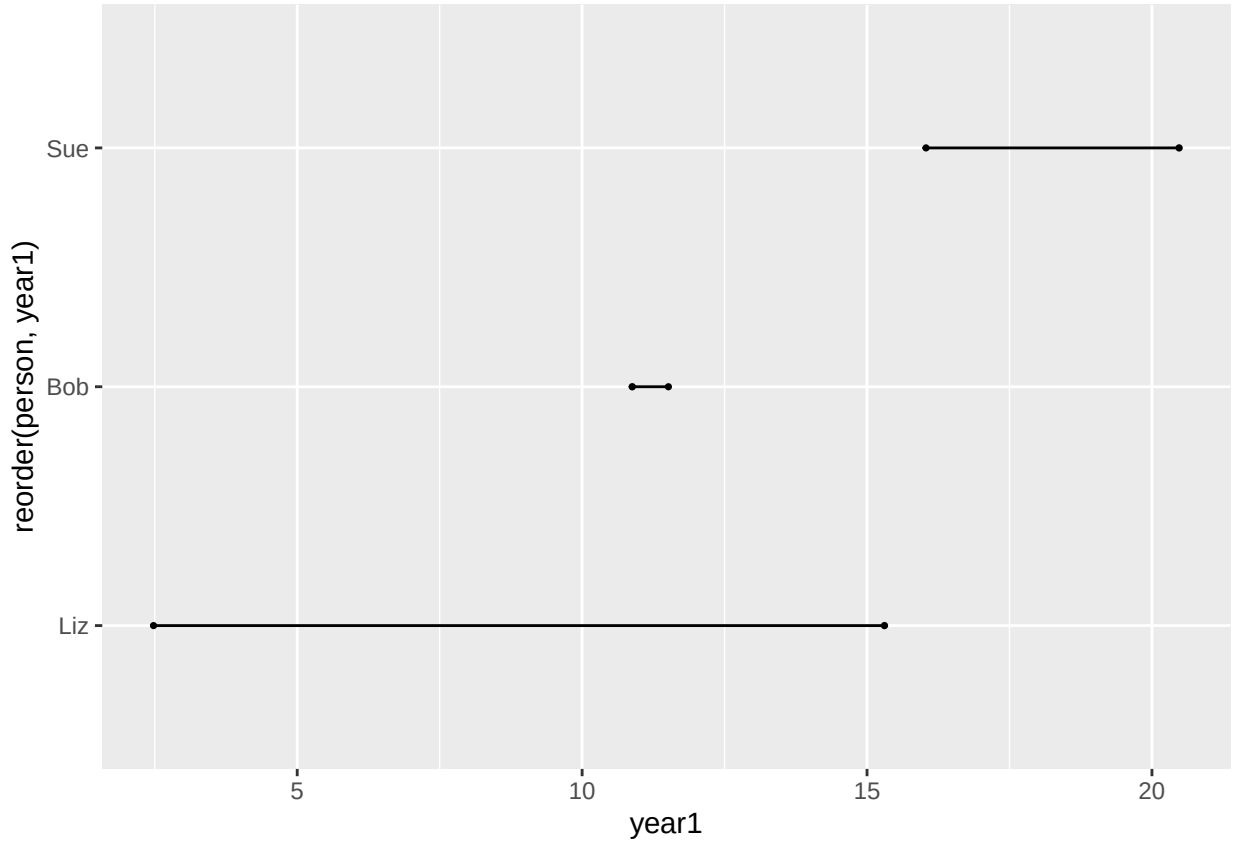
```r
# reformat the fake data from the stacked area chart, so we are comparing times at 1 and 5 for Bob, Sue

df2 <- df %>%
  filter(time == 1 | time == 5) %>%
  pivot_wider(names_from = time, values_from = change) %>%
  rename(year1 = "1", year5 = "5")

df2
```

```
## # A tibble: 3 x 3
##   person year1 year5
##   <chr>  <dbl> <dbl>
## 1 Bob    11.5   10.9
## 2 Sue    16.0   20.5
## 3 Liz     2.48  15.3
```

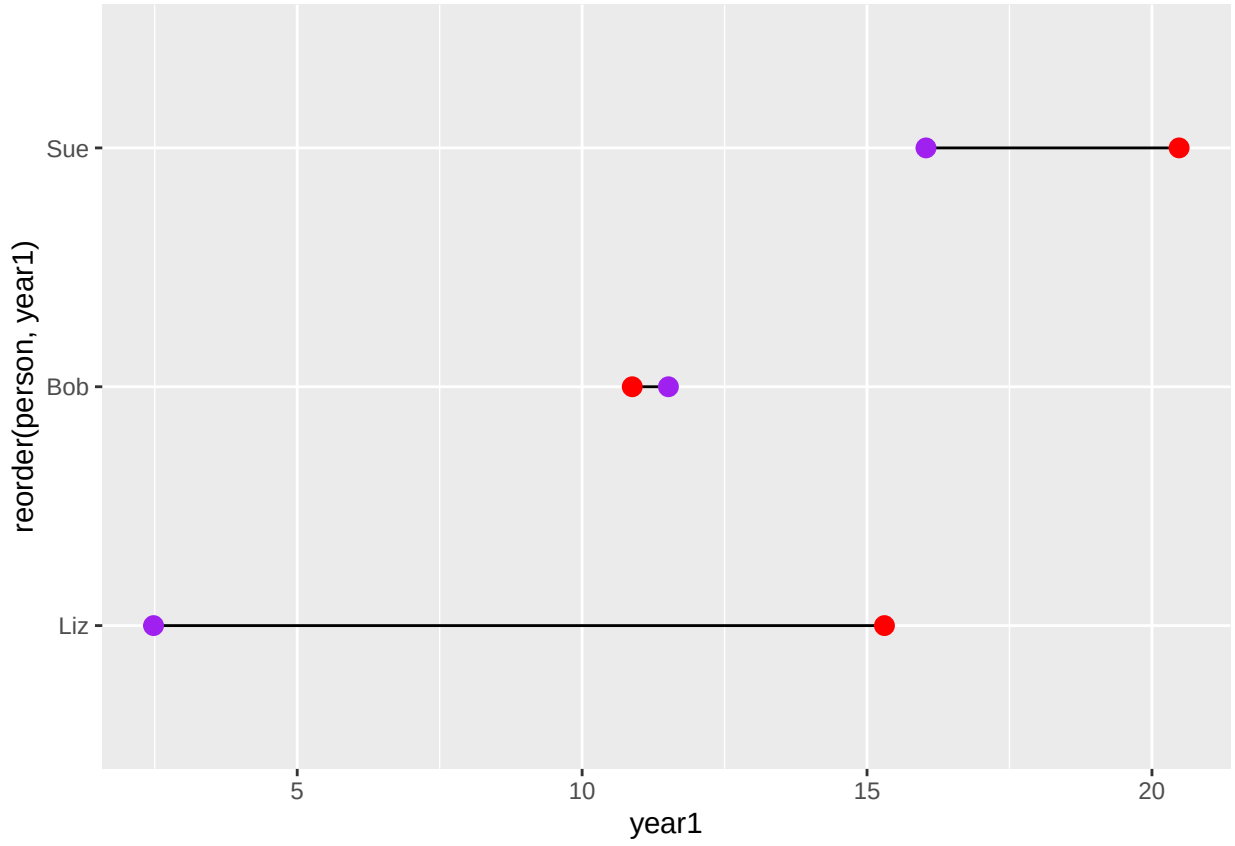
```r
# basic dumbell chart
ggplot(df2,
       aes( y = person,     # the starting point, earliest time
            x = year1,      # the ending point, final time
            xend = year5))+ # connects the points
  geom_dumbbell()
```




```
# reorder the y axis
ggplot(df2,
       aes( y = reorder(person, year1), # reorder the y axis
            x = year1,
            xend = year5))+
  geom_dumbbell()
```




```
ggplot(df2,
       aes( y = reorder(person, year1), # reorder the y axis
            x = year1,
            xend = year5))+
  geom_dumbbell(
    colour_x = "purple",
    colour_xend = "red",
    size_x = 3,
    size_xend = 3
  )
```




```
# change the color and sizes
```