TUT-8

Uzair Mirza

14/03/2022

Introduction

This week's lecture we will be discussing topics from Ch:17. Chapter 17 is about **Tidying Data**. All the problems being discussed can be found on the PASIAS here

Specifically we will be focusing on `pivot_wider` for this week.

Q17.24 Jocko's Garage

Insurance companies are scepticale Joko is running a scam and giving higher estimates than the standard market. To investigate this sample of 10 cars involved in a crash are taken to his garage and another garage to get estimates.

a. Read and observe the data

```
my_url <- "http://ritsokiguess.site/datafiles/jocko.txt"
cars0 <- read_table(my_url, col_names = FALSE) # reads 1st row as data
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_double(),
##   X4 = col_double(),
##   X5 = col_double(),
##   X6 = col_double(),
##   X7 = col_double()
## )
```

```
cars0
```

```
## # A tibble: 6 x 7
##   X1    X2       X3    X4    X5    X6    X7
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 a     Car        1     2     3     4     5
## 2 a     Jocko   1375  1550  1250  1300   900
## 3 a     Other   1250  1300  1250  1200   950
## 4 b     Car        6     7     8     9    10
## 5 b     Jocko   1500  1750  3600  2250  2800
## 6 b     Other   1575  1600  3300  2125  2600
```

We see that we `Xi` are variables we have to make sense of and rename them as we move forward.

b. Make this data set tidy. That is, you need to end up with columns containing the repair cost estimates at each of the two garages and also identifying the cars, with each observation on one row. Describe your thought process.

Let us first make it longer and see what it looks like. We will keep variable `X1` `X2` and make the rest longer.

```
cars0 %>% pivot_longer(X3:X7, names_to="old_cols", values_to="values")
```

```
## # A tibble: 30 x 4
##    X1    X2    old_cols values
##    <chr> <chr> <chr>     <dbl>
##  1 a     Car   X3            1
##  2 a     Car   X4            2
##  3 a     Car   X5            3
##  4 a     Car   X6            4
##  5 a     Car   X7            5
##  6 a     Jocko X3         1375
##  7 a     Jocko X4         1550
##  8 a     Jocko X5         1250
##  9 a     Jocko X6         1300
## 10 a     Jocko X7          900
## # ... with 20 more rows
```

From 6 observations we have gone to 30.
Still no where to make much sense.

Let's work on it. It is now that we will be using `pivot_wider()`
What does `pivot_wider()` do? Takes a catgorical variable, makes the unique catogeries as a variable and fills in the related data under the new variables.
Hence the dimention in terms of rows decreases and cols increases.
Let's see what it looks like.

`names_from = col_name` is the catogorical variable we are interested to make into individual variables.

```
(cars0 %>% pivot_longer(X3:X7, names_to="names", values_to="values") %>%
pivot_wider(names_from = X2, values_from = values) -> cars)
```

```
## # A tibble: 10 x 5
##    X1    names   Car Jocko Other
##    <chr> <chr> <dbl> <dbl> <dbl>
##  1 a     X3        1  1375  1250
##  2 a     X4        2  1550  1300
##  3 a     X5        3  1250  1250
##  4 a     X6        4  1300  1200
##  5 a     X7        5   900   950
##  6 b     X3        6  1500  1575
##  7 b     X4        7  1750  1600
##  8 b     X5        8  3600  3300
##  9 b     X6        9  2250  2125
## 10 b     X7       10  2800  2600
```

```
(cars.1 <- cars %>% select(Car, Jocko, Other))
```

```
## # A tibble: 10 x 3
##      Car Jocko Other
##    <dbl> <dbl> <dbl>
##  1     1  1375  1250
##  2     2  1550  1300
##  3     3  1250  1250
##  4     4  1300  1200
##  5     5   900   950
##  6     6  1500  1575
##  7     7  1750  1600
##  8     8  3600  3300
##  9     9  2250  2125
## 10    10  2800  2600
```

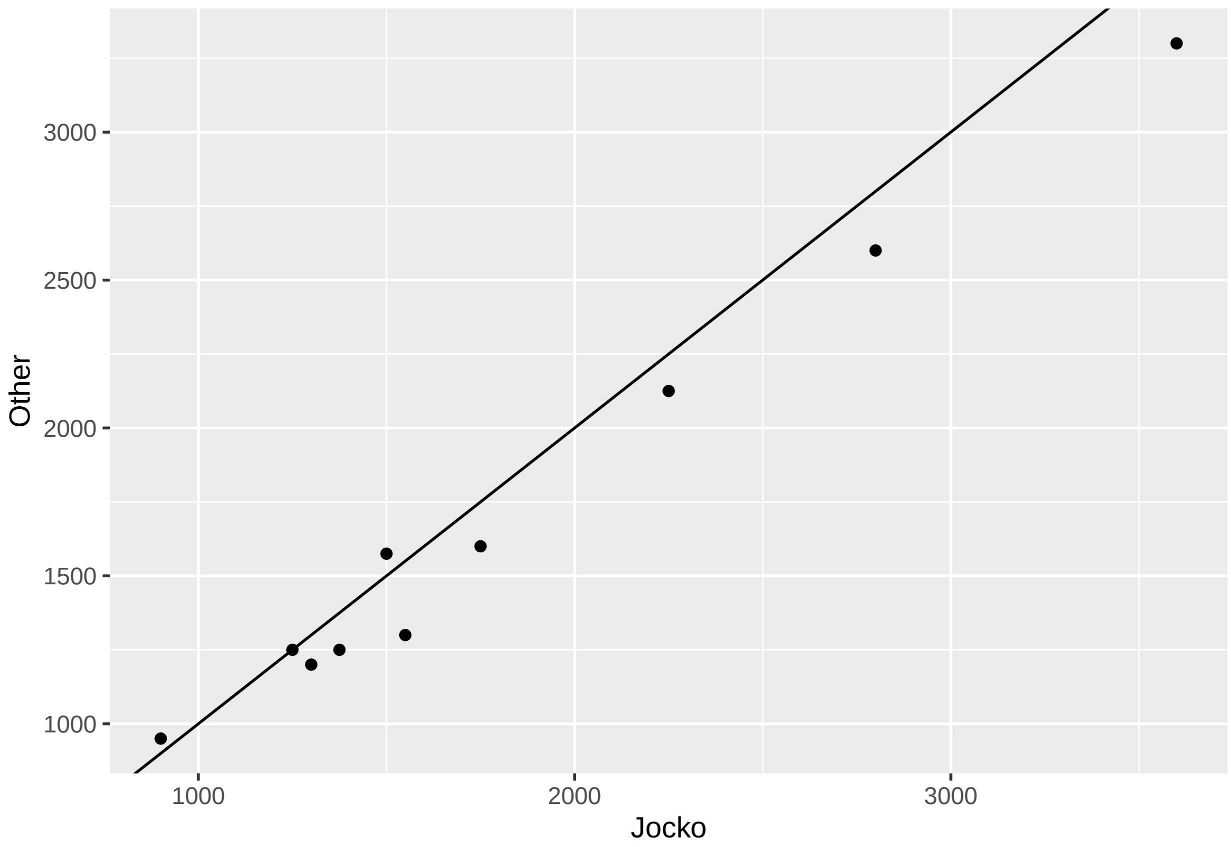
What can we observe?
Especially for the `X2` col. Note the number of new variables generated.

Note with this data we con perform our hyp testing or make sutiable plots to investigate the trend.

c. Now observe the trend and investigate if he is charging more.

Now if we get a scatter plot and plot a $y = x$ line would tell us if he is charging extra.
Depending on the region where most of the points lie. either `y>x` or `y<x`

```
ggplot(cars, aes(x=Jocko, y=Other)) + geom_point() + geom_abline(slope = 1, intercept = 0)
```

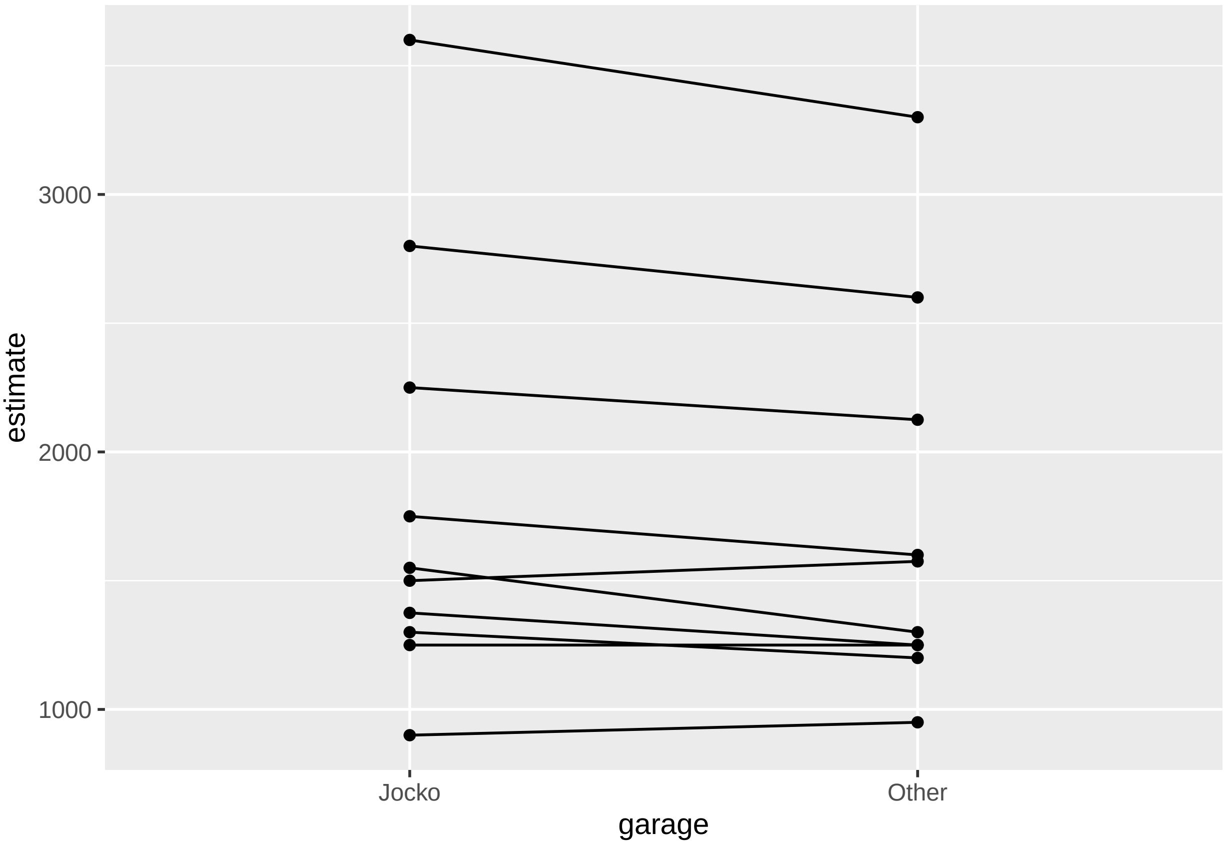


Let us also look at Spegetti Plots.
Used to observe trends or observe the changes as we progress in diff catogeries.
Lets get the data in the right format first. In order to do this let's make it longer first because we are in 2-d lets get the catagorical variables(The garages) into 1 variable which goes on the x-axis and the estimates will be on the y-axis.

```
# Making the data longer
(cars.1 %>% pivot_longer(-Car, names_to = "garage", values_to = "estimate" ) -> cars.12)
```

```
## # A tibble: 20 x 3
##      Car garage estimate
##    <dbl> <chr>     <dbl>
##  1     1 Jocko      1375
##  2     1 Other      1250
##  3     2 Jocko      1550
##  4     2 Other      1300
##  5     3 Jocko      1250
##  6     3 Other      1250
##  7     4 Jocko      1300
##  8     4 Other      1200
##  9     5 Jocko       900
## 10     5 Other       950
## 11     6 Jocko      1500
## 12     6 Other      1575
## 13     7 Jocko      1750
## 14     7 Other      1600
## 15     8 Jocko      3600
## 16     8 Other      3300
## 17     9 Jocko      2250
## 18     9 Other      2125
## 19    10 Jocko      2800
## 20    10 Other      2600
```

```
#Making the spegetti plot
ggplot(cars.12, aes(x=garage, y=estimate, group=Car)) + geom_point() + geom_line()
```



Majority of the lines are going downhill hence we have slight visual evidence that Jocko is messing around.

Might want to do a t.test to further verify?
(Well CLT check before)
If fails then perhaps median or variance test.

17.26 Tidy blood pressure

Basic study that measures patients systolic heart pressure before and after an appoinment.

a. Read and display the data

```
my_url <- "http://ritsokiguess.site/datafiles/blood_pressure2.csv"
(bp0 <- read_csv(my_url))
```

```
## Rows: 2 Columns: 11-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): time
## dbl (10): p1, p2, p3, p4, p5, p6, p7, p8, p9, p10
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 2 x 11
##   time      p1    p2    p3    p4    p5    p6    p7    p8    p9   p10
##   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 before   132   135   149   133   119   121   128   132   119   110
## 2 after    118   137   140   139   107   116   122   124   115   103
```

Why is this data not tidy?
values under `time` should have their own col. -> `pivot_wider()` ? `pi` should be observation in rows.

b. make it tidy

Lets first make our data longer ie. make the `pi` as rows and assign the values into 1 column. This would result in increasing the rows of the dataset.

```
(bp0 %>% pivot_longer(-time, names_to="person", values_to="bp") ->bp0.1)
```

```
## # A tibble: 20 x 3
##    time   person    bp
##    <chr>  <chr>  <dbl>
##  1 before p1       132
##  2 before p2       135
##  3 before p3       149
##  4 before p4       133
##  5 before p5       119
##  6 before p6       121
##  7 before p7       128
##  8 before p8       132
##  9 before p9       119
## 10 before p10      110
## 11 after  p1       118
## 12 after  p2       137
## 13 after  p3       140
## 14 after  p4       139
## 15 after  p5       107
## 16 after  p6       116
## 17 after  p7       122
## 18 after  p8       124
## 19 after  p9       115
## 20 after  p10      103
```

Lets make the data abit wider now and get the `before` & `after` variable.

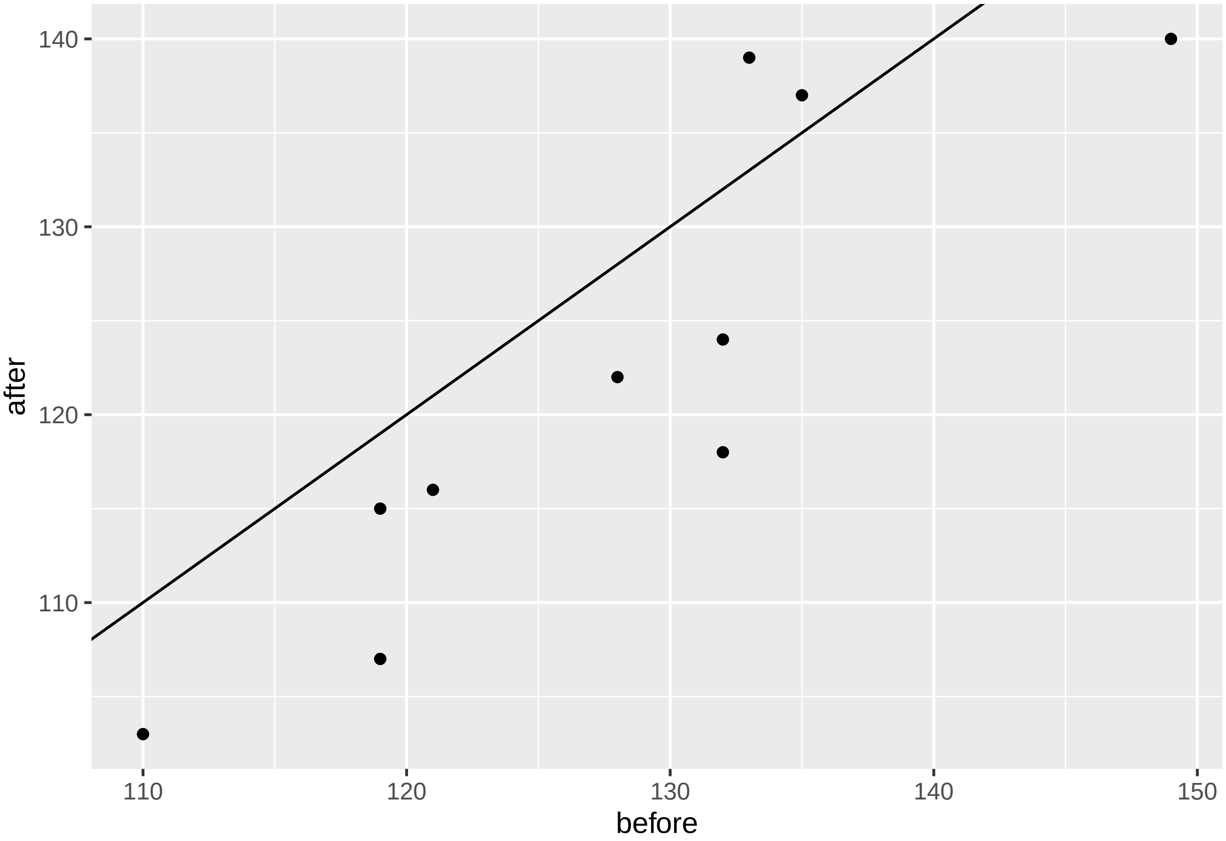
```
(bp0.1 %>% pivot_wider(names_from = time, values_from = bp) -> blood_pressure)
```

```
## # A tibble: 10 x 3
##    person before after
##    <chr>   <dbl> <dbl>
##  1 p1        132   118
##  2 p2        135   137
##  3 p3        149   140
##  4 p4        133   139
##  5 p5        119   107
##  6 p6        121   116
##  7 p7        128   122
##  8 p8        132   124
##  9 p9        119   115
## 10 p10       110   103
```

c. observe the trend

again if it's equal most points should lie CLOSE to or on the $y = x$ line.

```
ggplot(blood_pressure, aes(x=before, y=after)) + geom_point() +
geom_abline(intercept = 0, slope = 1)
```



We can see that before entering their BP is higher.

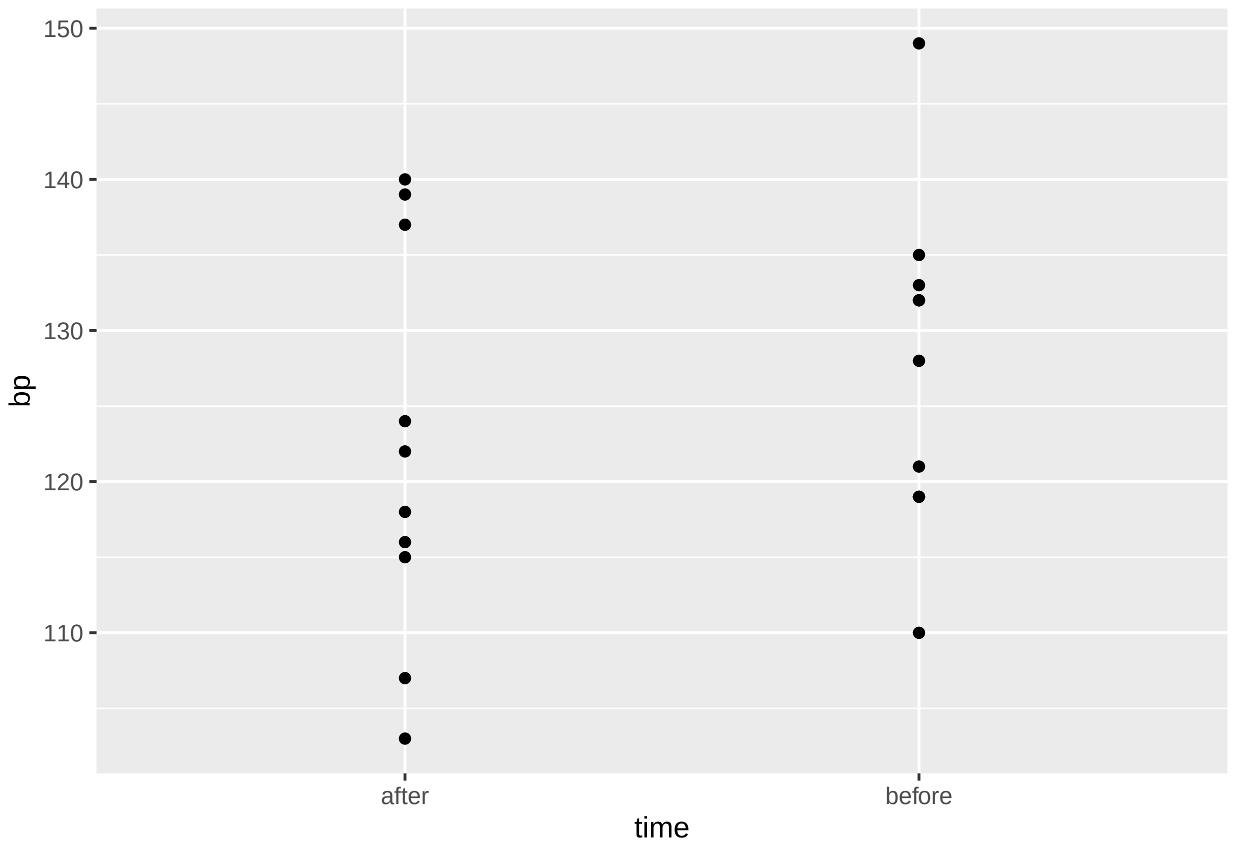
d. get data in the right format to make spegetti plots and plot the spegetti plot.

We need before and after under 1 variable and their values under 1 variable

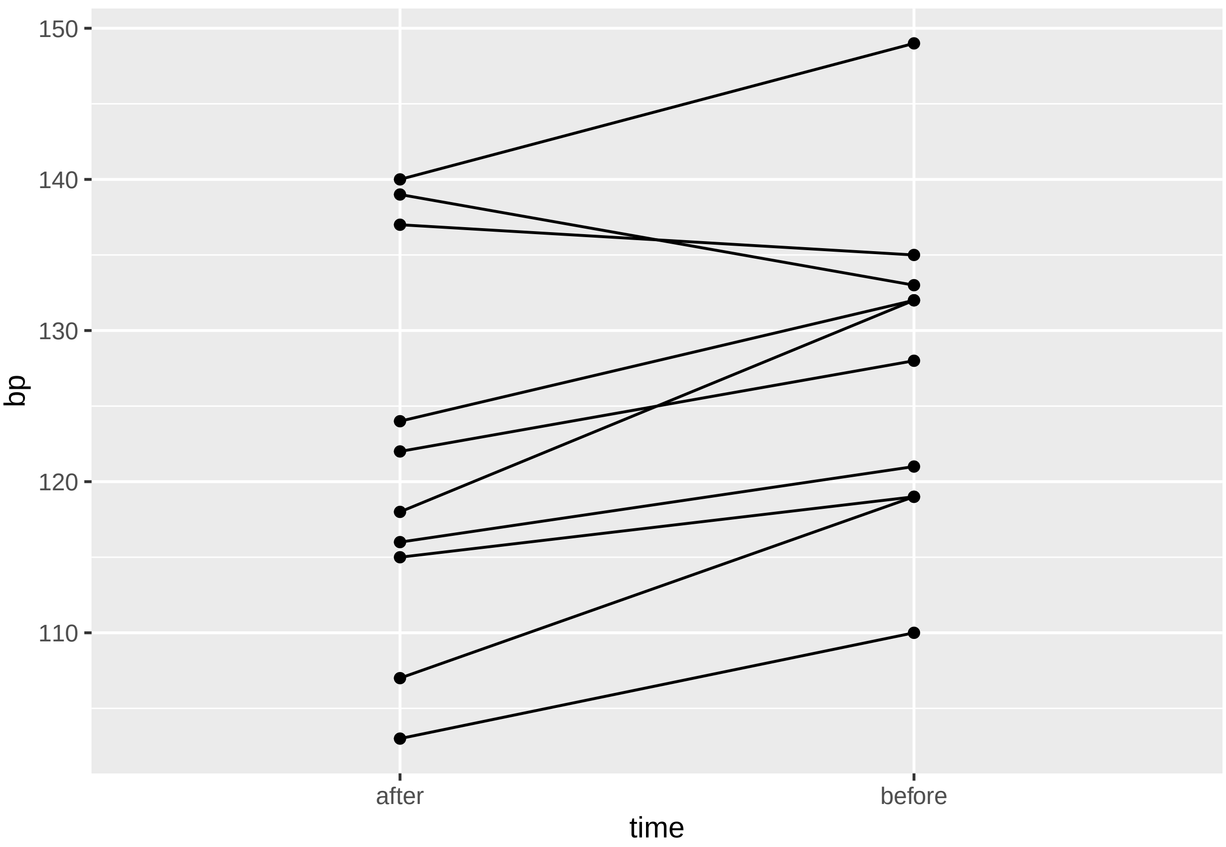
```
# getting data in the right format
(blood_pressure %>% pivot_longer(-person, names_to = "time", values_to = "bp" ) -> bp.1)
```

```
## # A tibble: 20 x 3
##    person time      bp
##    <chr>  <chr>  <dbl>
##  1 p1     before   132
##  2 p1     after    118
##  3 p2     before   135
##  4 p2     after    137
##  5 p3     before   149
##  6 p3     after    140
##  7 p4     before   133
##  8 p4     after    139
##  9 p5     before   119
## 10 p5     after    107
## 11 p6     before   121
## 12 p6     after    116
## 13 p7     before   128
## 14 p7     after    122
## 15 p8     before   132
## 16 p8     after    124
## 17 p9     before   119
## 18 p9     after    115
## 19 p10    before   110
## 20 p10    after    103
```

```
# making the spegetti plot
## lets get the dots on the plot
bp.1 %>% ggplot(aes(x=time, y=bp)) + geom_point()
```



```
## lets connect these dots for each person
bp.1 %>% ggplot(aes(x=time, y=bp, group=person)) + geom_point() + geom_line()
```


150
140
130
120
110
bp
after
before
time