Post1-Pingjing-Wang.Rmd

Pingjing Wang

2017/10/28

The Extend of the `ggplot2`

Introduction of the ggplot2

A lot of work needed to be done if we want to use the R's traditional graphics to get a nice set of plots or view the same plot for different groups in a data set. Therefore, learning the `ggplot2` seems to be essential and important for us due to the reason that it will provide a much easier way to get beautiful graphs. The `ggplot2` is "a data visualization package for the statistical programming language R" (Wikipedia). The "gg" in `ggplot2` represents the "Grammer of Graphics," a comprehensive theory of graphics raised by Leland Wilkinson in 2005. In his book "Leland Wilkinson," Wilkinson shows how to describe plots not just as discrete simple types, but using a "grammar" that would work both for plots we usually use and for almost any conceivale graphic. Nowadys, the `"ggplot2"` package is probably the most popular package in R because it is extremely flexible and easy to learn as well as to use.

Motivation for the post

In previous labs, we have already learned the main function `ggplot()`, inputs, consturntion, and some auxiliary functions of the `ggplot2`. Today I will go beyond what we've coverd before and introduce more intresting usages and functions of the `ggplot2`:

- Add error bars to bar and line plots
- Adding different labels above error bars
- Using line segments to compare values

Adding error bars to bar and line plots

In this part, I will show how to add error bars to something that we are familiar with like bar and line plots.

An error bar is similar to a "pointrange (minus the point, plus the whisker)" or a "linerange (plus the whiskers)". In simply terms, `geom_error`, the main function of error bars, is used to show means, standard errors, confidence intervals, and standard deviations of a set of data. Error bars are often combined with other fucntions.

Preparing for the data

First after installing and evoking the `ggplot()` and `dplyr()` package, I'm going to create a new data set in order to graph bar and line charts as well as error bars.

```
# Install the package "ggplot2" and "dplyr"
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Creating a simple new data set
df <- data.frame(phonecall = factor(c(2, 2, 2, 5, 5, 5, 6, 6,
                                      6)),
                 response = c(2, 1, 2, 5, 4, 5, 3, 5, 6),
                 familymember = factor(c(3, 2, 1, 3, 2, 1, 3, 2,
                                         1)),
                 se = c(0.4, 0.2, 0.4, 0.5, 0.3, 0.2, 0.4, 0.6,
                        0.7))

# Checking the data set by the function `head()`
head(df)
```

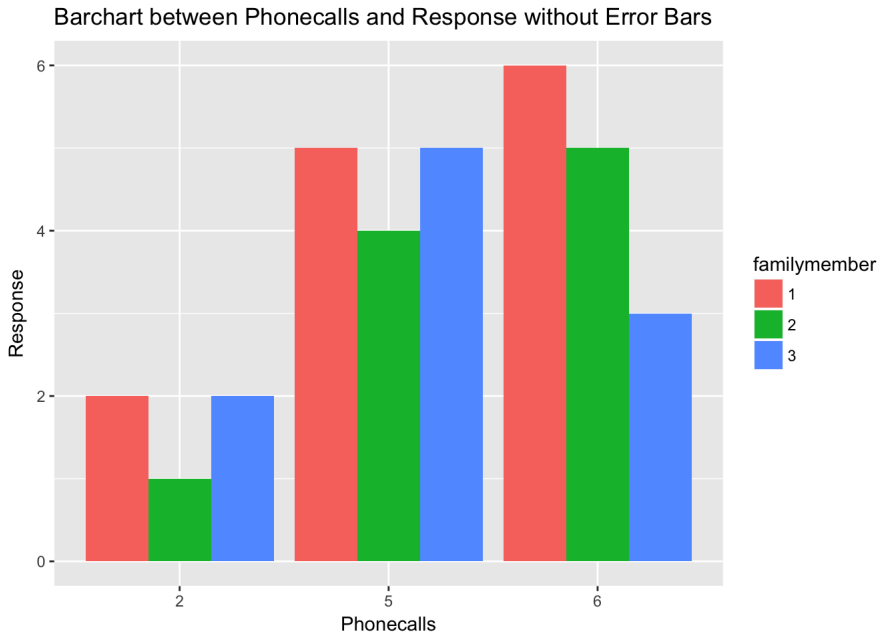
```
##   phonecall response familymember  se
## 1         2        2            3 0.4
## 2         2        1            2 0.2
## 3         2        2            1 0.4
## 4         5        5            3 0.5
## 5         5        4            2 0.3
## 6         5        5            1 0.2
```

Barplot with error bars

Second, I'm going to show what the function `geom-errorbar` looks like and how to add it to bar plots.

We know that the construction of a ggplot is done by *adding layers* with the `+` operator. Therefore, I will plot a barchart same as what we've learned in lectures and labs first and then add a the `geom-errorbar` onto it. In this way, we will clearly see the differences between two graphs and codes.

```
# Plotting the barchart without error bars
ggplot(data = df, aes(x = phonecall, y = response, fill =
                      familymember)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Barchart between Phonecalls and Response without Error Bars")
```

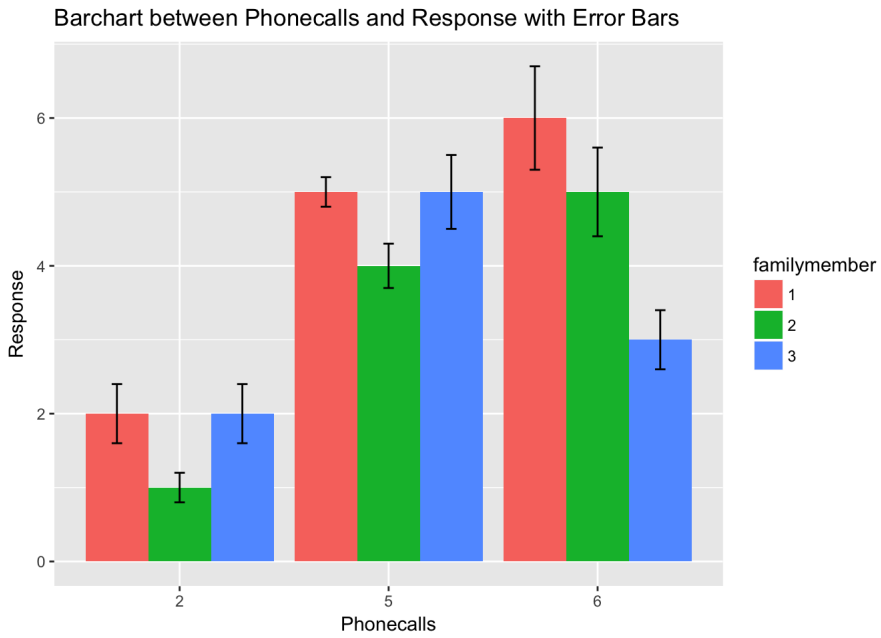


The next step is adding the `geom_errorbar()` on the above barchart by using the `+` operator.

The `geom_errorbar` indicates the error or uncetainty in a measurement. In other words, the `geom_errorbar` gives a general idea of how precise this measurement is, or conversely, how far from the reported value the true (error free) value might be.

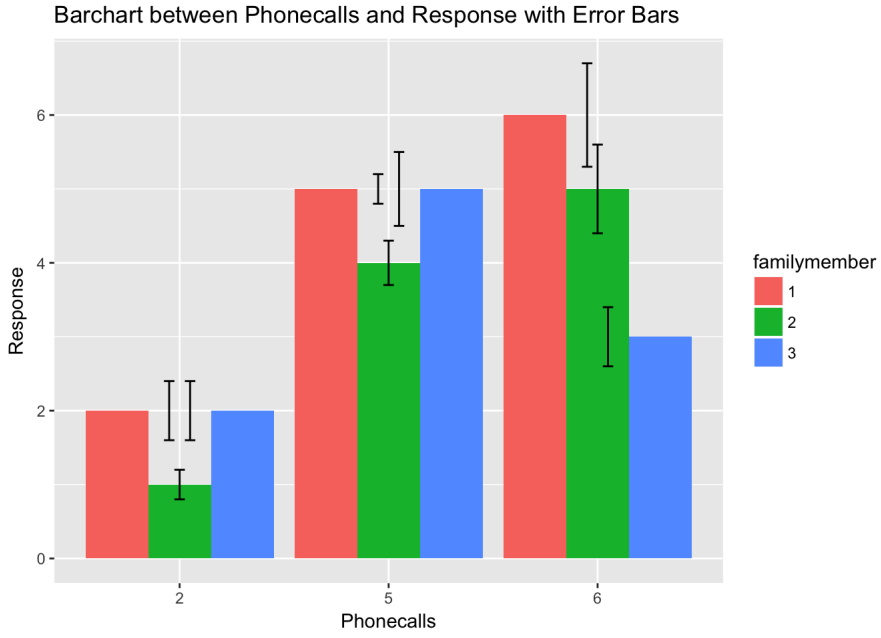
What we actually do here is use `geom_errorbar()` to map variables to values for *parameters* `ymin` and `ymax`. There are another two *parameters* needed to be mentioned. One *parameter* is `width =`. We should specified the width of the ends of the error bars. For example, we can do `width = 0.2`; otherwise, the error bars will be very wide, spanning all the space between items on the x-axis. Except the `width =`, another *parameter* is `position = position_dodge()`. For a bar graph with groups of bars, the error bars must also be *dodged*; otherwise, they'll have the exact same x coordinate and line up with the bars. The default dodge width for `geom_bar()` is 0.9, and we have to tell the error bars to be dodged the same width.

```
# Plotting the barchart with the error bars with the specified dodge width and the value of width
ggplot(data = df, aes(x = phonecall, y = response, fill =
                      familymember)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymax = response + se, ymin = response - se),
                position = position_dodge(0.9), width = 0.15) +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Barchart between Phonecalls and Response with Error Bars")
```



The graph below shows that if we fail to specify the dodge width, it will default to dodging by the width of the error bars, which is usually less than the width of the bars.

```
# Plotting the barchart with the error bars without setting the dodge width
ggplot(data = df, aes(x = phonecall, y = response, fill =
                      familymember)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymax = response + se, ymin = response - se),
                position = position_dodge(), width = 0.15) +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Barchart between Phonecalls and Response with Error Bars")
```

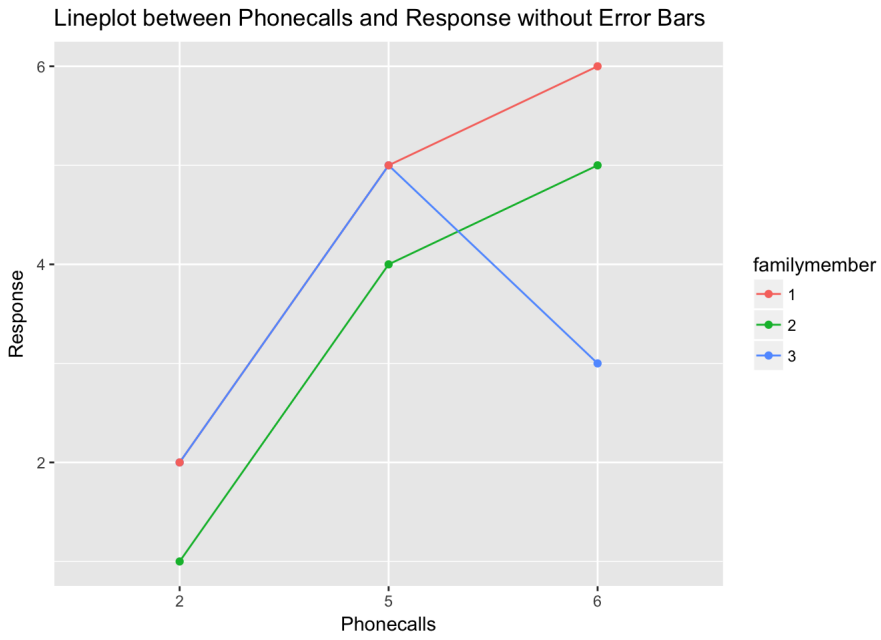


Lineplot with error bars

Third, I'm going to explore how to add error bars to line plots.

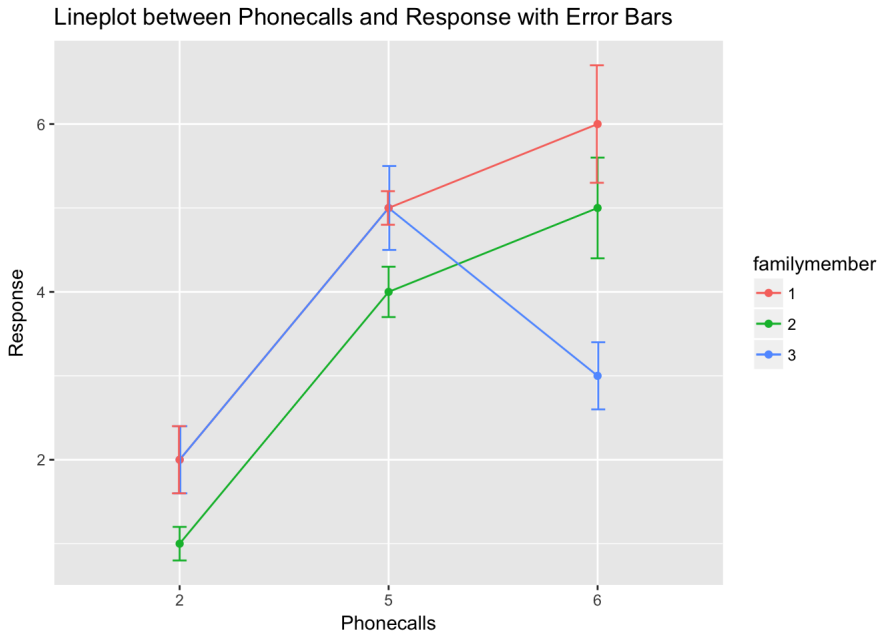
As the previous example, I will graph a line plot without error bars first and then graph the second line plot with error bars so that we can see clearly the differences between two graphs and codes. I will still use the previous data set.

```
# Graphing the lineplot without error bars
ggplot(data = df, aes(x = phonecall, y = response, group = familymember, fill
                      = familymember, color = familymember)) +
  geom_line() +
  geom_point() +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Lineplot between Phonecalls and Response without Error Bars")
```



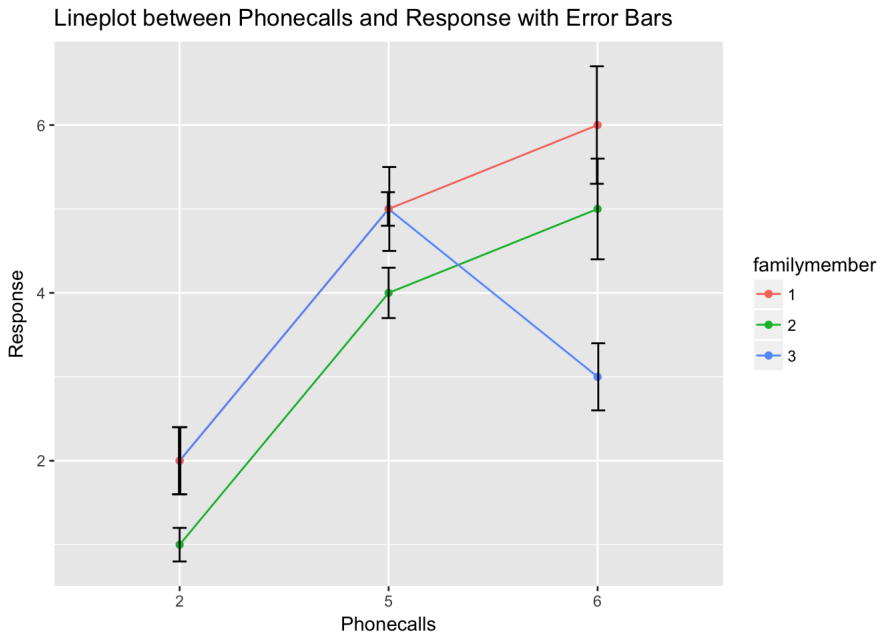
Right now, I'm going to use the function `geom_errorbar()` to add the error bars onto the above line plots.

```
# Graphing the lineplot with error bars
ggplot(data = df, aes(x = phonecall, y = response, group = familymember, fill
                      = familymember, color = familymember)) +
  geom_line() +
  geom_point() +
  geom_errorbar(aes(ymax = response + se, ymin = response - se),
                position = position_dodge(0.01), width = 0.2) +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Lineplot between Phonecalls and Response with Error Bars")
```



From the above lineplot we can find that the error bars are in the same colors of the lines and points. However, if we want to set the error bars be in a diferent color than the lines and points, we should draw the error bars first. By doing this way, the error bars are underneath the lines and points, and the error bars will not inherit colors.

```
ggplot(data = df, aes(x = phonecall, y = response, group = familymember, fill
                      = familymember, color = familymember)) +
  geom_line() +
  geom_point() +
  geom_errorbar(aes(ymax = response + se, ymin = response - se),
                position = position_dodge(0.01), width = 0.2, color =
                "black") +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Lineplot between Phonecalls and Response with Error Bars")
```



Adding different labels above error bars

Sometimes although we've already distinguish barcharts by different colors, we still have problems telling or finding out what exactly each bars with the same color represents. In order to solve this problem, we can add some labels to make a better distinction between every bars even, especially for bars which are in the same colors.

Let's look at the previous barchart. There're three green bars, three red bars, and three blue bars. For example, if we want to distinguish these three green bars, one possible way for us is figuring out what the values of x and y axis are. This way indeed works here. However, if we have a really large data set, then we need a much simplier way. At this moment, adding labels to every bar becomes critial for the further use for statisticians.

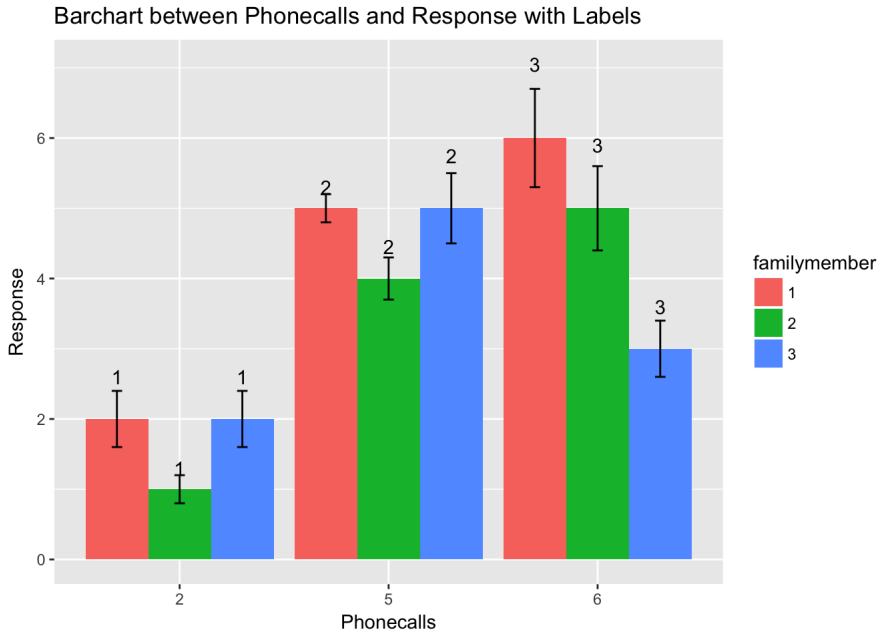
```
# Adding different labels above error bars to make a better distinction between each bar
# Here you can change any label you like. For example, you can label them as "*" or other characters you like.

label <- c(label <- c("1", "1", "1", "2", "2", "2", "3", "3",
                      "3"))

ggplot(data = df, aes(x = phonecall, y = response, fill =
                      familymember)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymax = response + se, ymin = response - se),
                position = position_dodge(0.9), width = 0.15) +
  geom_text(aes(y = response +  1.5 * se,
                label = label, familymember = familymember), position =
                position_dodge(0.9), width = 0.5) +
  labs(x = "Phonecalls", y = "Response") +
  ggtitle("Barchart between Phonecalls and Response with Labels")
```

```
## Warning: Ignoring unknown parameters: width
```

```
## Warning: Ignoring unknown aesthetics: familymember
```



Using line segments to compare values

Now let's switch our pespectives. What should we do if we want to see the change between the old and new value of one variable just in one glance? The `ggplot2` provides a really cool function `geom_segment()`, which will allow us to see the original value of variables, new value of variables, and the change between them. The `geom_segment()` function draws a stright line between points (x, y) and (xend, yend). The simplified format of the function is:

- `geom_segment(aes(x, y, xend, yend))`.

Preparing for the data

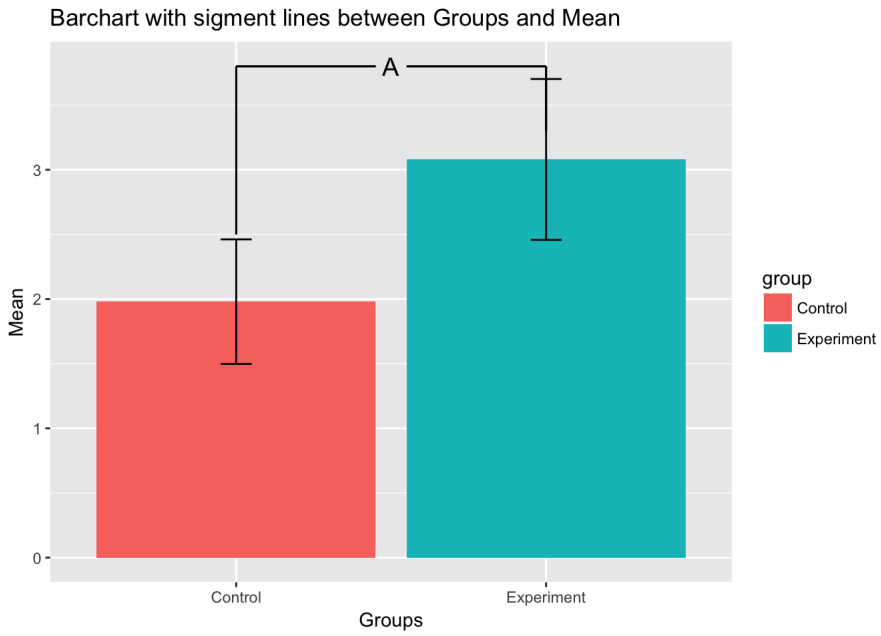
First I will create a simple data set for the following steps.

```
# Creating a simple new data set
control <- c(1.4, 2.5, 2.0, 2.4, 1.6)
experiment <- c(2.0,3.5,3.5,3.2,3.2)
mean <- c(mean(control), mean(experiment))
sd <- c(sd(control), sd(experiment))
df <- data.frame(group = c("Control", "Experiment"), mean = mean, sd = sd)
df$V <- factor(df$group, levels = c("Control", "Experiment"))
```

Barplot with error bars

Next, I will use the funtion `geom_sigment()` combining the `geom_errorbar()` to draw a barchart with sigment lines between groups and mean. I will show step by step how to draw the sigment lines by the function `geom_sigment()`

```
ggplot(data = df, aes(x = group, y = mean, fill = group)) +
  geom_bar(stat = "identity", position = position_dodge(0.9)) +
  geom_errorbar(aes(ymax = mean +sd, ymin = mean - sd), width = 0.1) +
  # graph the verticle line above the variable control
  geom_segment(aes(x = 1, y = 2.5, xend = 1, yend = 3.8)) +
  # graph the verticle line above the varaivle experiment
  geom_segment(aes(x = 2, y = 3.3, xend = 2, yend = 3.8)) +
  # graph the horizontal line between two vertile lines
  geom_segment(aes(x = 1, y = 3.8, xend = 1.45, yend = 3.8)) +
  geom_segment(aes(x = 1.55, y = 3.8, xend = 2, yend = 3.8)) +
  # add label to this function
  annotate("text", x=1.5, y=3.8, label="A", size=5) +
  labs(x = "Groups", y = "Mean") +
  ggtitle("Barchart with sigment lines between Groups and Mean")
```



Conclusion (Take-home Message)

After reading this blog post, I wish that I could help reviewers understand the following three usages of `ggplot()`:

- Using the `geom_errorbar()` function to gives a general idea of how precise this measurement is
- Using *parameter* `label` in the function `geom_text()` to make plots more clear
- Using the function `geom_sigment()` see the change between the old and new value of one variable

References

1. Wikipedia
2. Information of Leland Wilkinson and his book
3. geom_errorbar
4. Center error bars (geom_error)
5. Adding labels
6. Changing colors of error bars
7. geom_segment