Assignment1

hbz

28/10/2021

Hi, Dr. Andrew, this is my work of **assignment$_1$**, hope you like it~

First, I’m going to load all the *packages* might use.

In this section, we use three main packages, which are:

1. `tidyverse`: used to tidy and wrangle data.

2. `Hmisc`: used to calculate formulation of the data.

3. `ggridges`: used to show the distributions of the data.

4. `ggstatsplot`: used to provide some basic statistic information to give us some hints in future discussion and exploration.

Notice: we have hidden the message and the warning of the following chunk to make it more readable and clearer for the audience

```
library(tidyverse) # Load the tidyverse package
library(Hmisc) # Load the Hmisc package
library(ggstatsplot) # Load the ggstatsplot package
library(ggridges) # Load the ggridges package
```

Then, let’s read in the data

`read.csv` is a basic function in R studio to read *.csv* format file. Here, I used it to read our target data of the experimental results and give it the name **experiment_data**, saving it as a variable in the environment. It is easy for us to reload and deal with it.

```
experiment_data <- read.csv("assignment_1_data.csv")
```

We start to wrangle and revise our data to make it more clear and to a more manipulated format.

In general, we use a wide format, because most statistical analyzing software requires to put the data every raw as a subject. However, in this experiment, we use a long format to make it clearer for the audience to understand the meaning of the data. Also, some software requires us to input long format data. In this part, we use three functions:

1. `pivot_longer`: to change it from wide format to long format.

Notice: we could also use function `gather` wchich might be used equivalently although might be more complete

2. `mutate`: mutate could change the variable name of the data, cause *Condition1/2/3/4* is hard for us and the audience to understand, and cost us energy our cognitive resources to recall in the following analyses. So, we change it to a more remembered name, corresponding to our condition - priming type (**Number/Letter**) and following displaying type (**Number/Letter**). Also, to make it clear, we use the abbreviation (**Num/Let**).

3. `filter`: used to filter the data without useful information (e.g., if our subjects have missed some trails or have no data so these subjects should be deleted). `!is.na` means to judge if the value in Condition is **NA**. In this case, it does nothing to our data.

```
experiment_data_longer <- experiment_data %>%
  pivot_longer(cols = c(Condition1, Condition2, Condition3, Condition4),
               names_to = "Condition",
               values_to = "RT") %>%
  mutate(Condition = recode(Condition,
                            "Condition1" = "Num_Num",
                            "Condition2" = "Num_Let",
                            "Condition3" = "Let_Num",
                            "Condition4" = "Let_Let")) %>%
  filter(!is.na(Condition))
```

We are going to summarise the data and show the profile of our data to our audience.

1. `group_by` is used to classify the data (in this case, is by **Condition**), then the following results will be given out as different **conditions**.

2. `summarise` is used to calculate and show some basic information of our data, including *mean*, *standard difference*, *median*, *quantile*, and *number*.

3. `knitr::kable` is used to generate a more formal and beautiful table to read for our audience.

```
table_original <- experiment_data_longer %>%
  group_by(Condition) %>%
  summarise(mean_RT = mean(RT), sd_RT = sd(RT), median = median(RT), number = n())

knitr::kable(
  table_original,
  caption = "Profile of reaction time (RT) in different conditions", col.names = c('Condition
name', 'Mean RT (ms)', 'SD RT (ms)', 'Median RT (ms)', 'N'), digits = 2, format.args = list(b
ig.mark = ",",
  scientific = FALSE), align = "lllll"
)
```

Profile of reaction time (RT) in different conditions

Condition name	Mean RT (ms)	SD RT (ms)	Median RT (ms)	N
Let_Let	995.85	29.66	997.5	48
Let_Num	1,251.73	27.99	1,252.5	48
Num_Let	1,245.73	29.80	1,239.5	48
Num_Num	995.42	26.51	995.5	48

In this section, we will visualize our data, which is the most interesting and creative part.

I create several pictures to profile this data following the logic to explore and show it in a real investigating process.

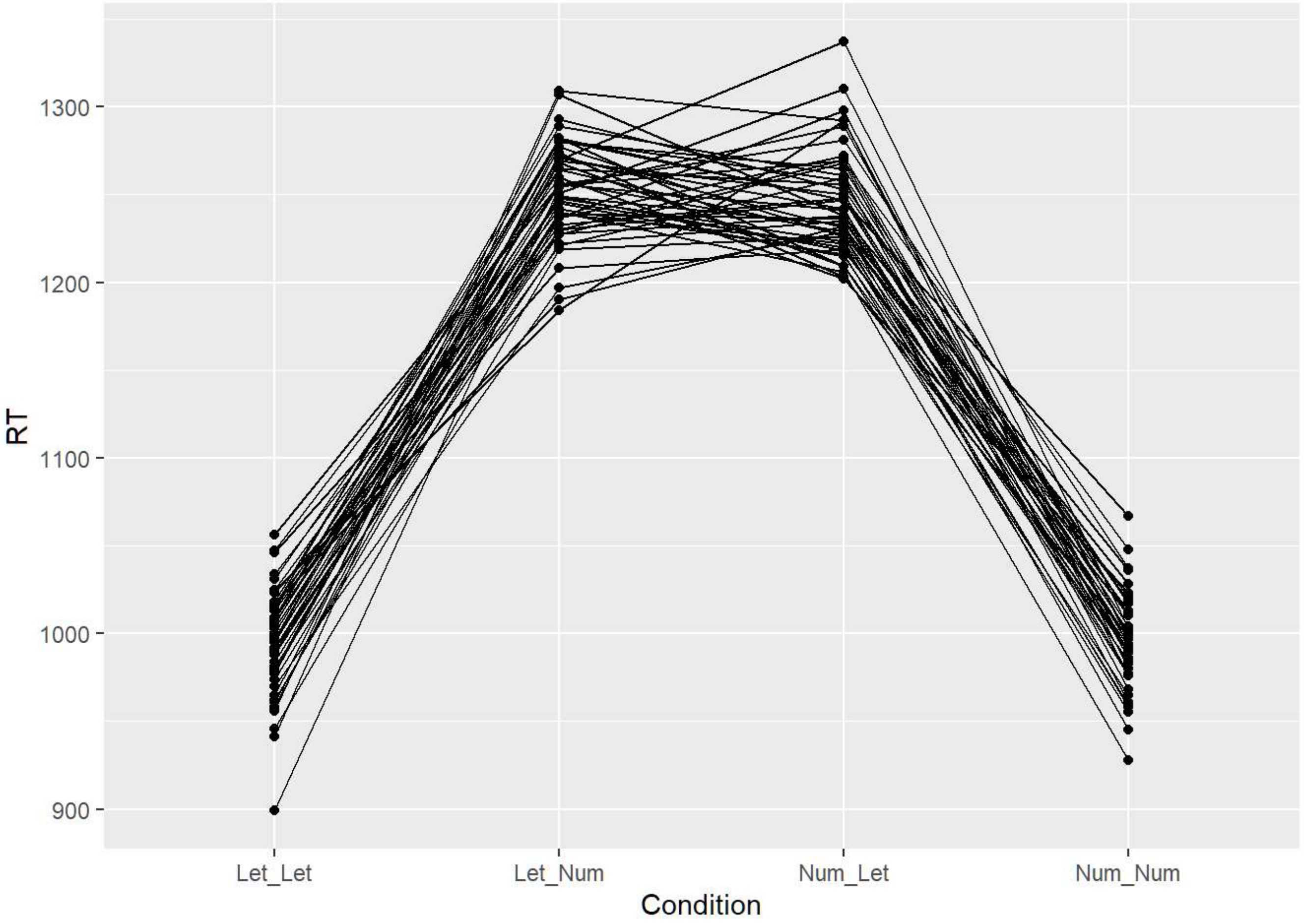
First, I think it is very important to draw all the data and see what it looks like. Considering the within-subject design of this experiment, I add lines between paired points for every subject, to compare the trend of changes among different conditions within-subject.

`ggplot` is a typical function used to draw pictures.

`geom_point` is a supplementary function corresponding for `ggplot` to draw the points of every data.

`geom_line` is a supplementary function corresponding for `ggplot` to draw the lines of paired data.

```
experiment_data_longer %>%
  ggplot(aes(x = Condition, y = RT)) +
  geom_point() +
  geom_line(aes(group = ID))
```



Then, it is necessary to see the distribution of the data and the density curve is the intuitive and convenient way for us to know it.

`geom_density_ridges` : used to generate the density(distribution) of the data.

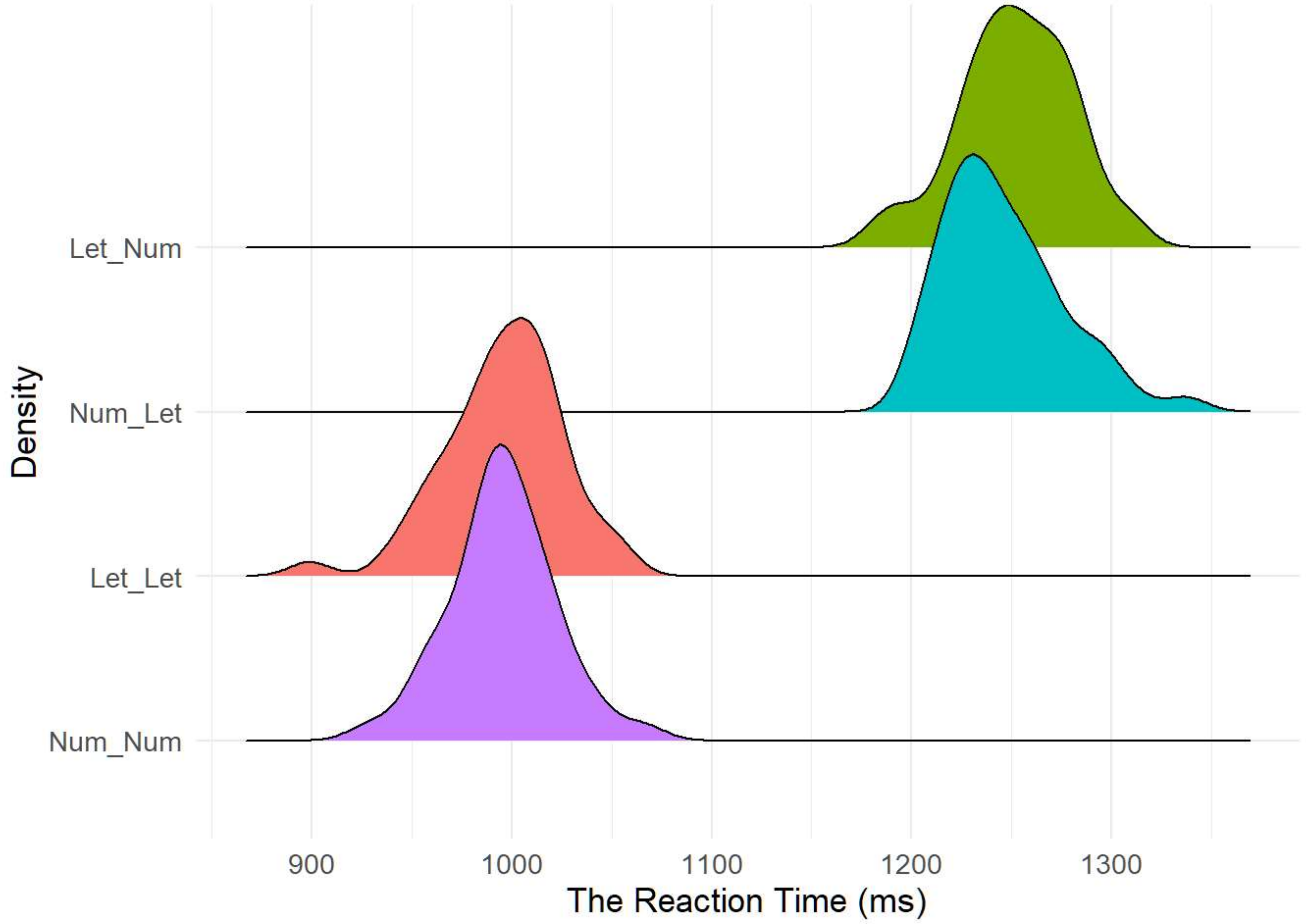
`guides` : used to set the properties of legends.

`theme_minimal` : used to clear the shade in background grid.

`them` : combied with `axis.text.x` to set the properties of text in x axis.

`labs` : used to set the properties of the main title, x/y titles.

```
experiment_data_longer %>%
  ggplot(aes(x = RT, y = fct_reorder(Condition, .fun = mean, RT))) +
  geom_density_ridges(height = .5, aes(fill = Condition)) +
  theme_minimal() +
  theme(text = element_text(size = 13)) +
  guides(fill = 'none') +
  labs(x = "The Reaction Time (ms)",
       y = "Density")
```



Next, I tend to draw data that can give comprehensive details and information for our audience to realize the data.

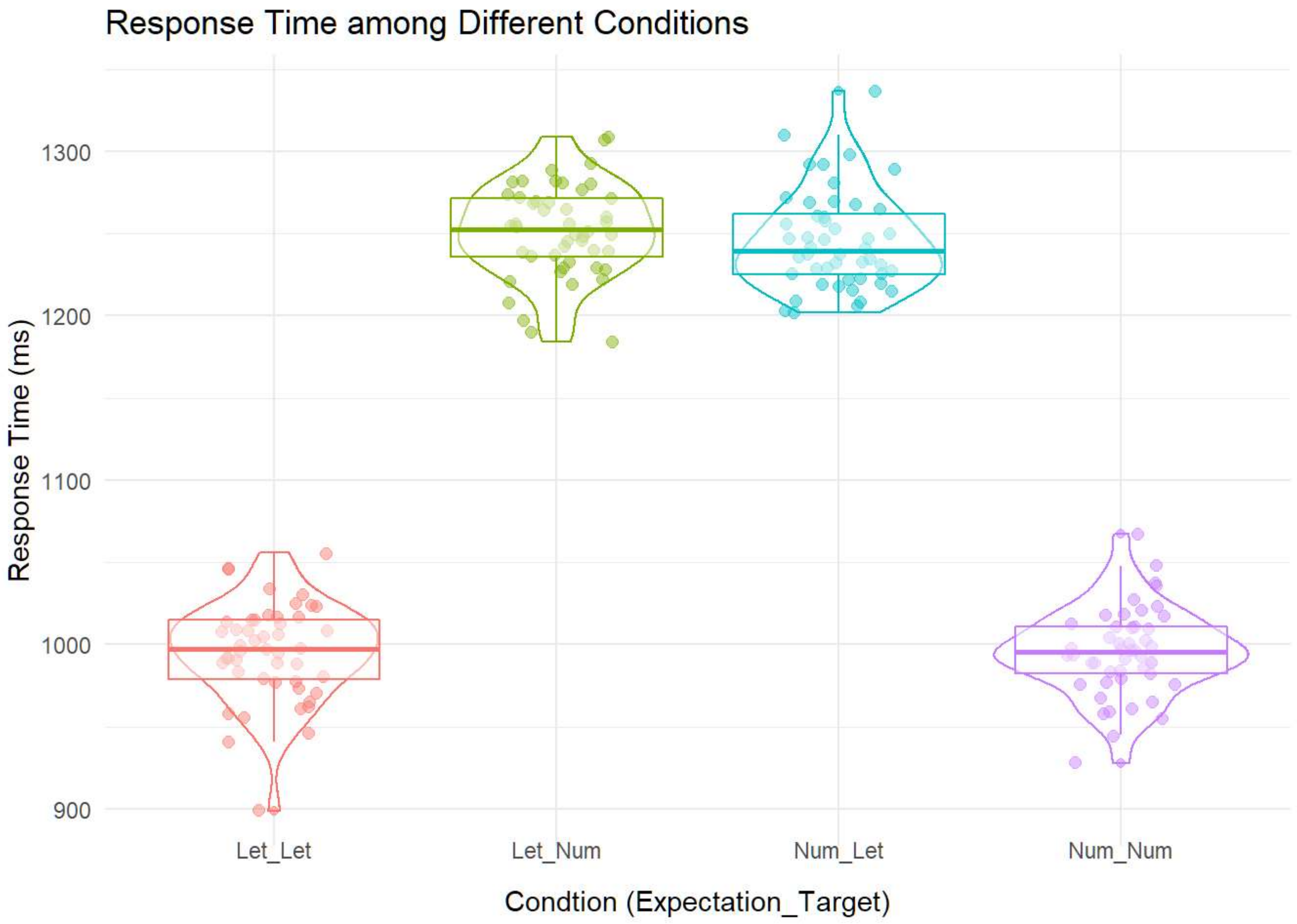
`geom_violin` : is a supplementary function corresponding for `ggplot` to draw the violin format picture for data.

`geom_jitter` : is a supplementary function corresponding for `ggplot` to draw the jittering point of data to make every point available to be seen.

`geom_boxplot` : is a supplementary function corresponding for `ggplot` to draw the box format picture of data.

other functions of packages see the section above.

```
experiment_data_longer %>%
  ggplot(aes(x = Condition, y = RT, colour = Condition)) +
  geom_violin() +
  geom_jitter(width = .2, alpha = .45, size = 2) +
  geom_boxplot(alpha = .5) +
  guides(colour = 'none') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, vjust = 5, hjust = .5)) +
  labs(title = "Response Time among Different Conditions",
       x = "Condtion (Expectation_Target)",
       y = "Response Time (ms)")
```

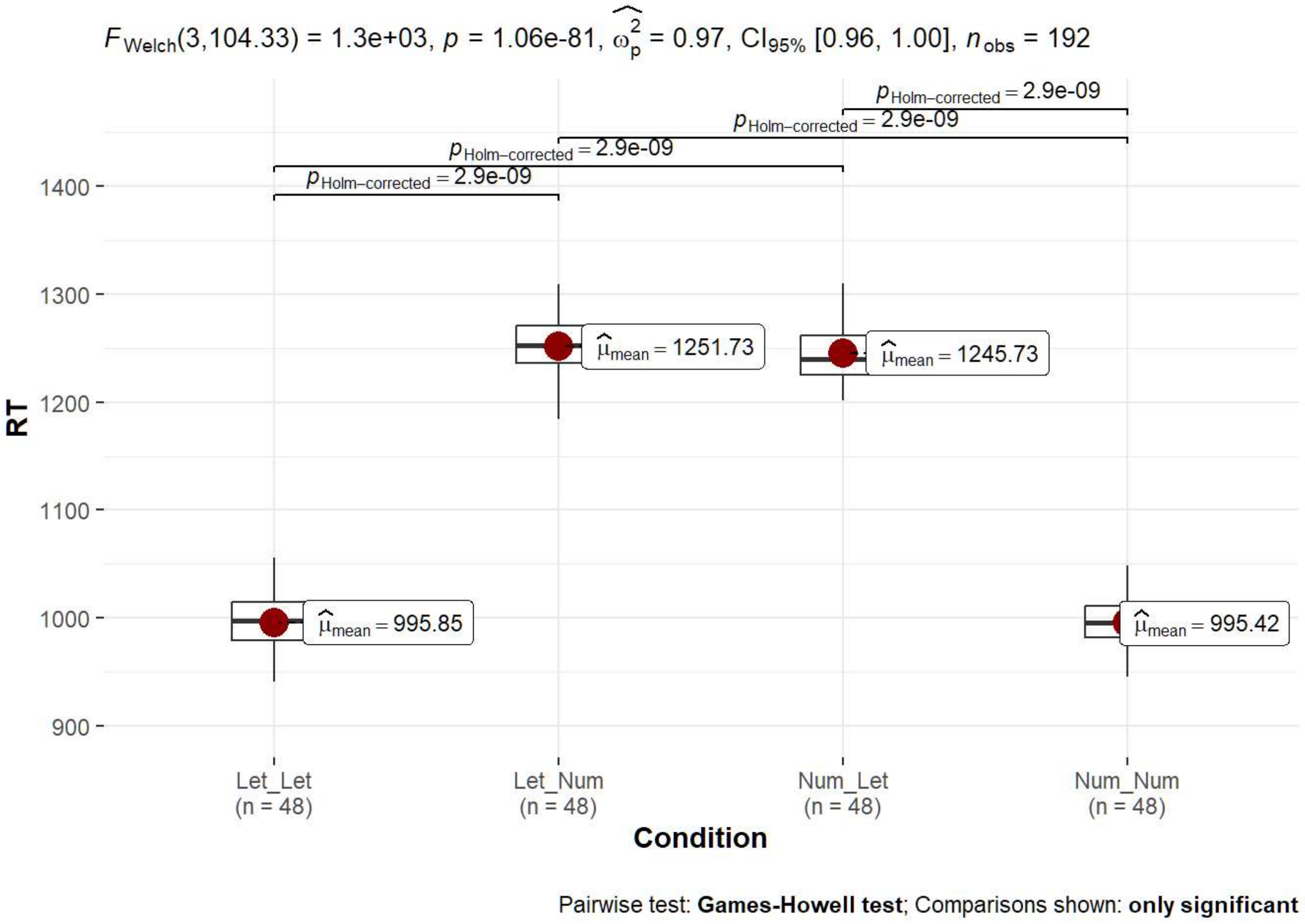


Finally, I tend to use an easily generated way to see the potential statistical value of this data.

`ggbetweenstats` : used to generate combinational plots to show the data with some statistical details.

Notice, I know it is not suitable here to use an independent t-test because of the design of this experiment (I use this just to help me prepare to develop later analyzing strategies). In this case, I believe the repeated-measure ANOVA is the best way to analyze this data with the function of `aov` *or* `anova` *.*

```
ggbetweenstats(experiment_data_longer, x = Condition, y = RT, plot.type = "box",
               point.args = list(alpha = 0))
```



In general, we could also give audience a link to our some materials, including lab website, other supplementary materials, video description or any related information.

Here, I recommend you to use Youtube (http://youtube.com), you could find anything you like. (just for fun, hhhh)

Finally, I would like to finish with a meme to myself.

an image caption Source: network