# ABRF 2017 Satellite workshop - Hands-on 2 : R markdown and simple statistics in R

*Meena Choi and Ting Huang*

*3/25/2017*

## Summary

- Start R markdown.
- Calculate simple statistics and visualize them using ggplot2.
- Saving your work

---

# 1. Basic statistical summaries in R

## 1.1 Calculate simple statistics

Let's start data with one protein as an example and calculate the mean, standard deviation, standard error of the mean across all replicates per condition. We then store all the computed statistics into a single summary data frame for easy access.

We can use the **aggregate** function to compute summary statistics

```r
#Load data from previous section
load(file = 'Section1.RData')
```

```r
# check what proteins are in dataset, show all protein names
unique(iprg$Protein)
```

```r
# Let's start with one protein, named "sp|P44015|VAC2_YEAST"
oneproteindata <- iprg[iprg$Protein == "sp|P44015|VAC2_YEAST", ]

# there are 12 rows in oneproteindata
oneproteindata
```

```
##                    Protein Log2Intensity                       Run
## 21096 sp|P44015|VAC2_YEAST      26.30163 JD_06232014_sample1_B.raw
## 21097 sp|P44015|VAC2_YEAST      26.11643 JD_06232014_sample1_C.raw
## 21098 sp|P44015|VAC2_YEAST      26.29089 JD_06232014_sample1-A.raw
## 21099 sp|P44015|VAC2_YEAST      25.81957 JD_06232014_sample2_A.raw
## 21100 sp|P44015|VAC2_YEAST      26.11527 JD_06232014_sample2_B.raw
## 21101 sp|P44015|VAC2_YEAST      26.08498 JD_06232014_sample2_C.raw
## 21102 sp|P44015|VAC2_YEAST      23.14806 JD_06232014_sample3_A.raw
## 21103 sp|P44015|VAC2_YEAST      23.32465 JD_06232014_sample3_B.raw
## 21104 sp|P44015|VAC2_YEAST      23.29555 JD_06232014_sample3_C.raw
## 21105 sp|P44015|VAC2_YEAST      20.94536 JD_06232014_sample4_B.raw
## 21106 sp|P44015|VAC2_YEAST      21.71424 JD_06232014_sample4_C.raw
## 21107 sp|P44015|VAC2_YEAST      20.25209 JD_06232014_sample4-A.raw
##        Condition BioReplicate Intensity
## 21096 Condition1            1  82714388
## 21097 Condition1           1  72749239
## 21098 Condition1           1  82100518
## 21099 Condition2           2  59219741
## 21100 Condition2           2  72690802
## 21101 Condition2           2  71180513
## 21102 Condition3           3   9295260
## 21103 Condition3           3  10505591
## 21104 Condition3           3  10295788
## 21105 Condition4           4   2019205
## 21106 Condition4           4   3440629
## 21107 Condition4           4   1248781
```

### 1.1.1 Calculate mean per groups

```
# splits 'oneproteindata' into subsets by 'Condition',
# then, compute 'FUN=mean' of 'log2Int'
sub.mean <- aggregate(Log2Intensity ~ Condition, data=oneproteindata, FUN=mean)
sub.mean
```

```
##    Condition Log2Intensity
## 1 Condition1      26.23632
## 2 Condition2      26.00661
## 3 Condition3      23.25609
## 4 Condition4      20.97056
```

### 1.1.2 Calculate SD(standard deviation) per groups

```
# The same as mean calculation above. 'FUN' is changed to 'sd'.
sub.sd <- aggregate(Log2Intensity ~ Condition, data=oneproteindata, FUN=sd)
sub.sd
```

```
##    Condition Log2Intensity
## 1 Condition1    0.10396539
## 2 Condition2    0.16268179
## 3 Condition3    0.09467798
## 4 Condition4    0.73140174
```

### 1.1.3 Count the number of observation per groups

```
# The same as mean calculation. 'FUN' is changed 'length'.
sub.len <- aggregate(Log2Intensity ~ Condition, data=oneproteindata, FUN=length)
sub.len
```

```
##    Condition Log2Intensity
## 1 Condition1             3
## 2 Condition2             3
## 3 Condition3             3
## 4 Condition4             3
```

### 1.1.4 Calculate SE(standard error of mean) per groups

$SE = \sqrt{\frac{s^2}{n}}$

```
sub.se <- sqrt(sub.sd$Log2Intensity^2/sub.len$Log2Intensity)
sub.se
```

```
## [1] 0.06002444 0.09392438 0.05466236 0.42227499
```

```
# make the summary table including the results above (mean, sd, se and length).
summaryresult <- data.frame(Group=c("Condition1", "Condition2", "Condition3", "Condition4"),
                            mean=sub.mean$Log2Intensity,
                            sd=sub.sd$Log2Intensity,
                            se=sub.se,
                            length=sub.len$Log2Intensity)
summaryresult
```

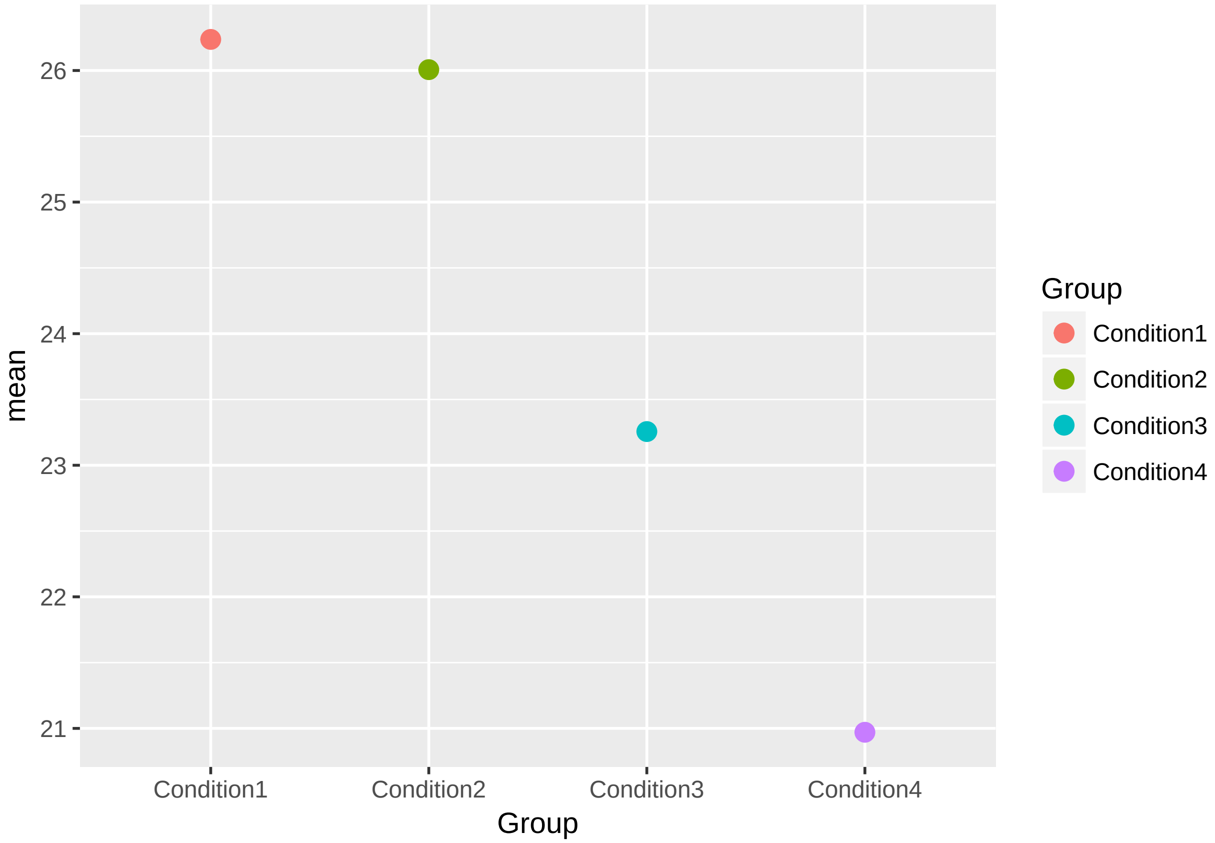
```
##        Group     mean         sd         se length
## 1 Condition1 26.23632 0.10396539 0.06002444      3
## 2 Condition2 26.00661 0.16268179 0.09392438      3
## 3 Condition3 23.25609 0.09467798 0.05466236      3
## 4 Condition4 20.97056 0.73140174 0.42227499      3
```

## 1.2 Visualization with error bars for descriptive purpose

'error bars' can have a variety of meanings or conclusions if what they represent is not precisely specified. Below we provide some examples of which types of error bars are common. We're using the summary of protein `sp|P44015|VAC2_YEAST` from the previous section and the `ggplot2` package as it provides a convenient way to make easily adaptable plots.

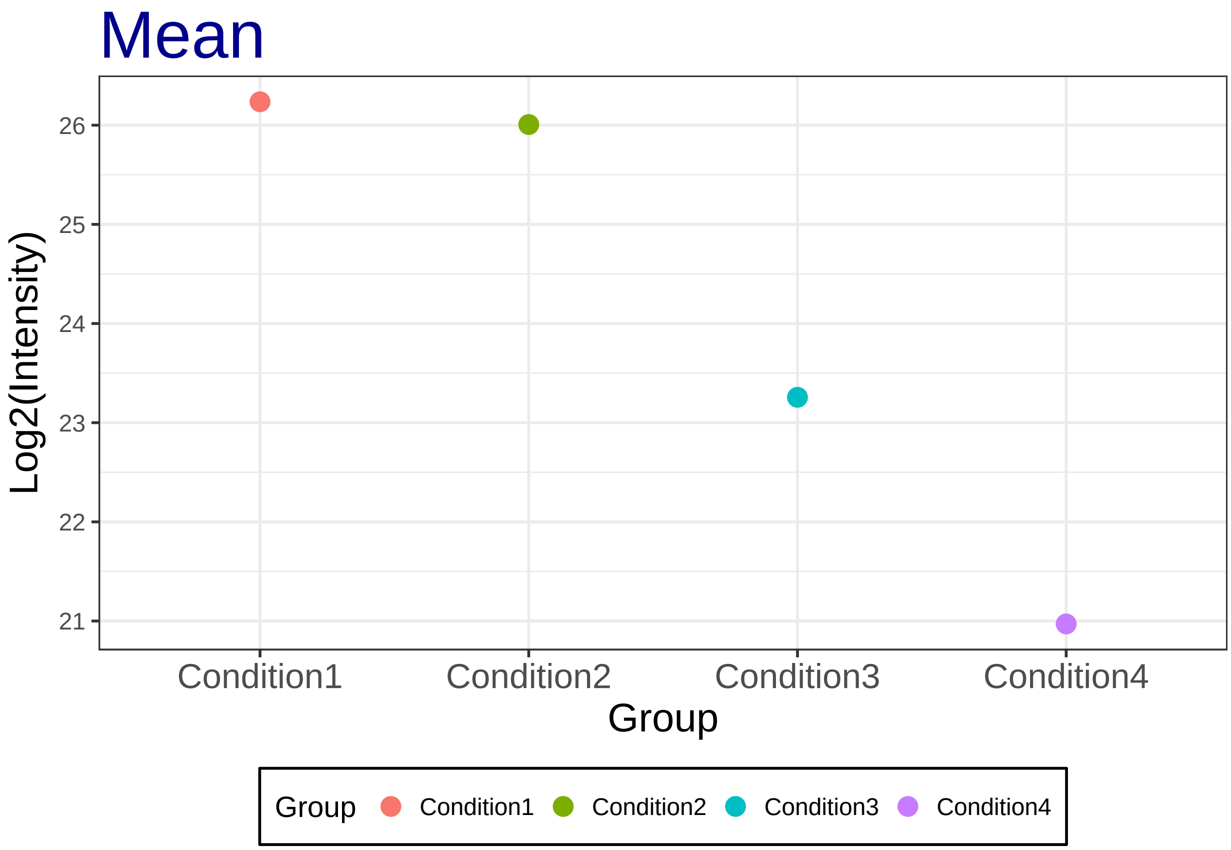
```
# Let's draw plots with mean and error bars
library(ggplot2)

# means without any errorbar
ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)
```

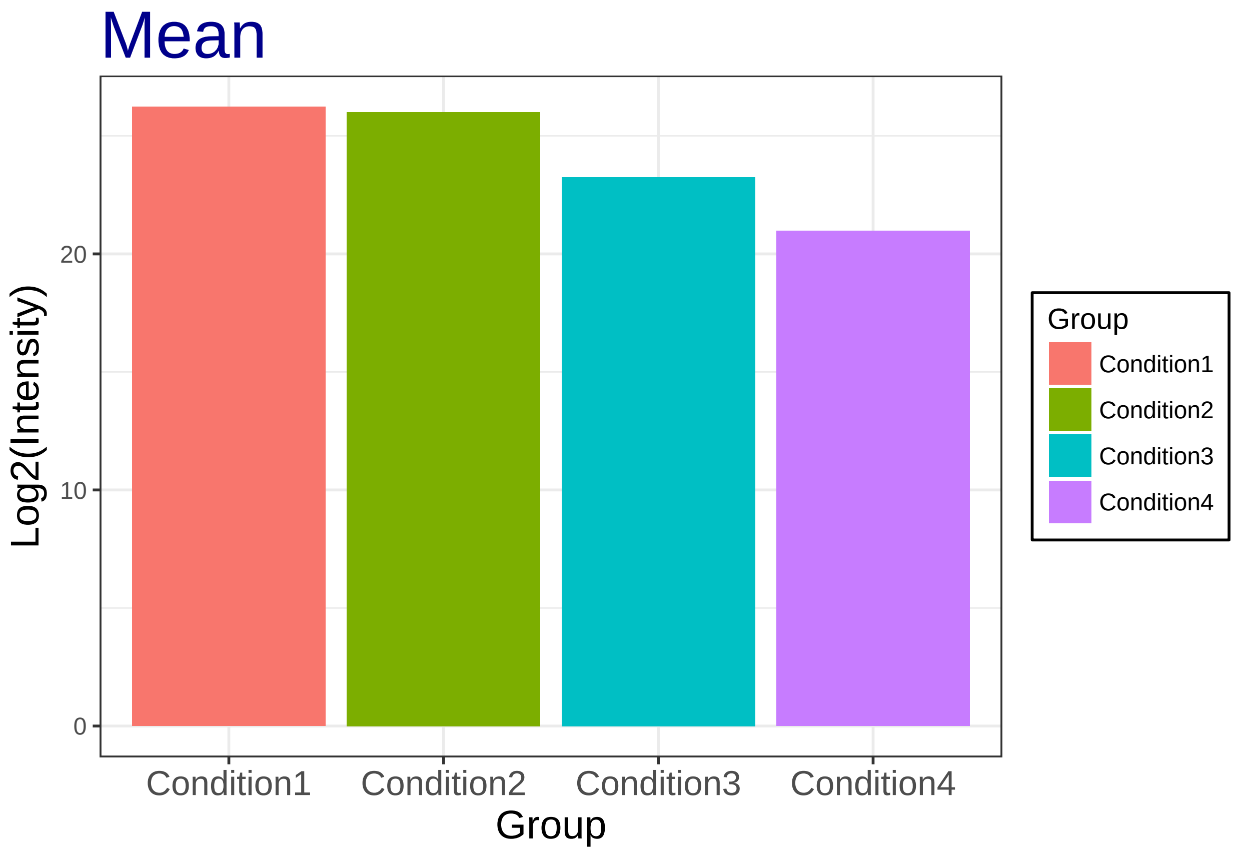


```r
# Let's change a number of visual properties to make the plot more atttractive
# Let's change the labels of x-axis and y-axis and title:
# add labs(title="Mean", x="Condition", y='Log2(Intensity)')
# Let's change background color for white : add theme_bw()
# Let's change size or color of labels of axes and title, text of x-axis : in theme
# Let's change the position of legend :'none' remove the legend
# Let's make the box for legend
# Let's remove the box for legend key.

ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)+
      labs(title="Mean", x="Group", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.position = 'bottom',
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```

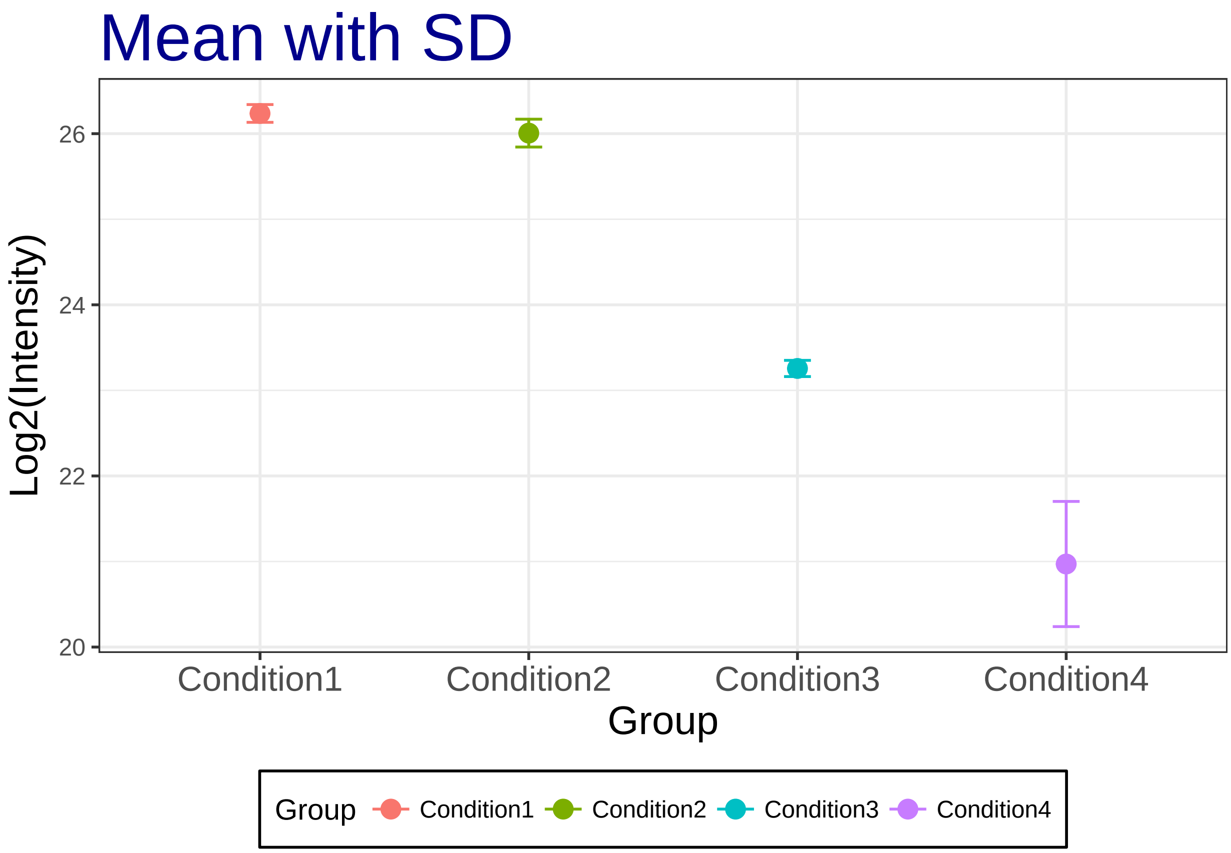


```
# Very similar but now as a bar plot.
ggplot(aes(x=Group, y=mean, fill=Group), data=summaryresult)+
      geom_bar(position=position_dodge(), stat='identity')+
      scale_x_discrete('Group')+
      labs(title="Mean", x="Group", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```

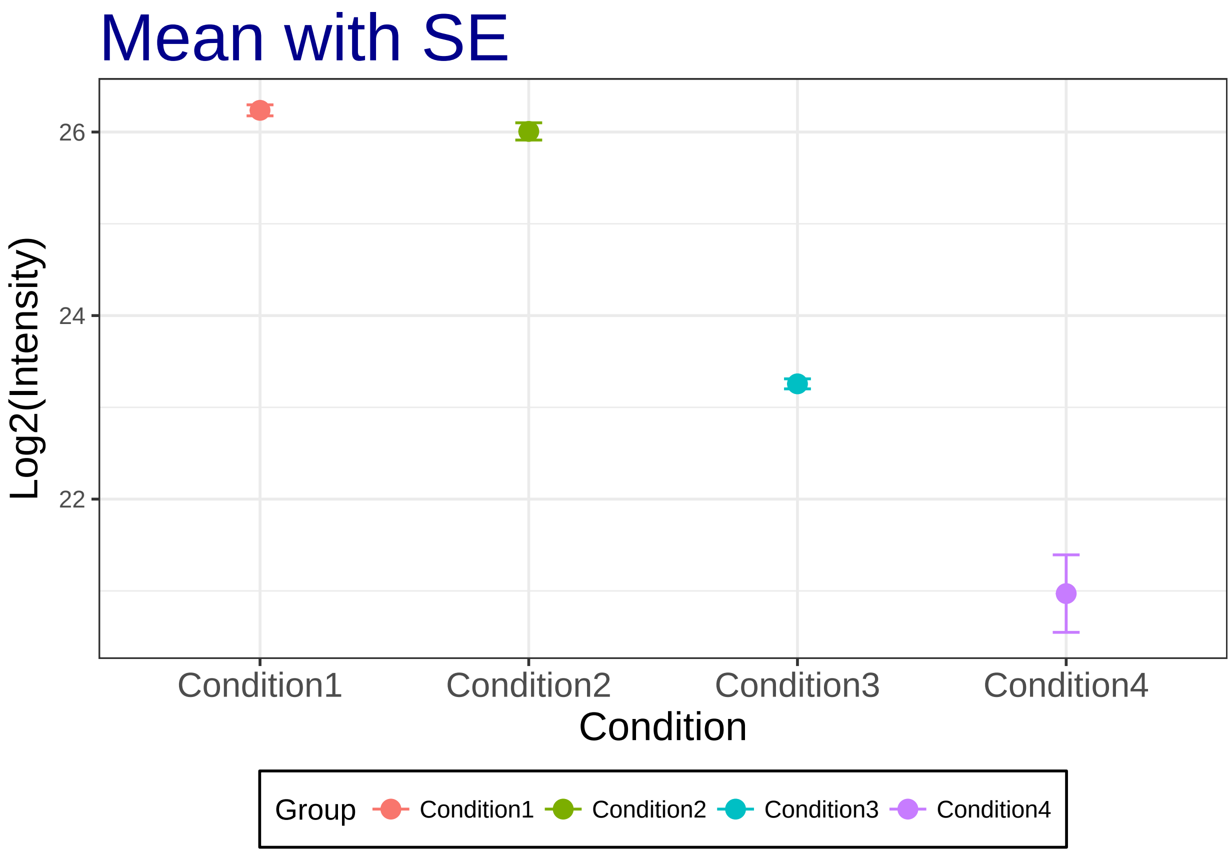


For the sake of this tutorial we'll continue adding error bars for different statistics with the point plots. We'll leave it as an exercise to add error bars to the barplots. Let's first add the standard deviation, then the standard error of the mean. Which one is smaller?

```
# mean with SD
ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)+
      geom_errorbar(aes(ymax = mean + sd, ymin=mean - sd), width=0.1)+
      scale_x_discrete('Group')+
      labs(title="Mean with SD", x="Group", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.position = 'bottom',
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```



```
# mean with SE
ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)+
      geom_errorbar(aes(ymax = mean + se, ymin=mean - se), width=0.1)+
      labs(title="Mean with SE", x="Condition", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.position = 'bottom',
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```



***Note*** : The SE is narrow than the SD!

### 1.3 Calculate the confidence interval

Now that we've covered the standard error of the mean and the standard deviation, let's investigate how we can add custom confidence intervals (CI) for our measurement of the mean. We'll add these CI's to the summary results we previously stored for protein `sp|P44015|VAC2_YEAST`

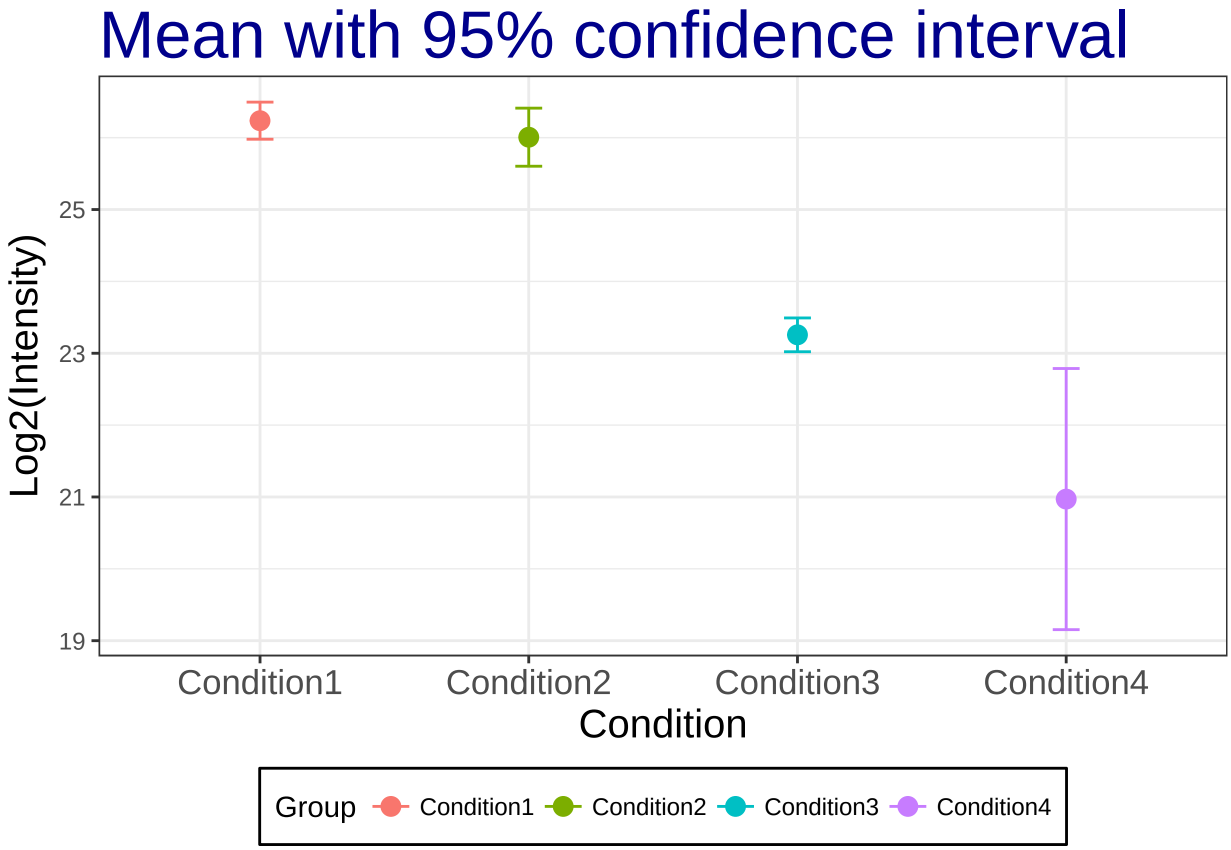
Confidence interval : mean $\pm$ $(SE \times \alpha/2$ quantile of t distribution)

```
# 95% confident interval
# Be careful for setting quantile for two-sided. need to divide by two for error.
# For example, 95% confidence interval, right tail is 2.5% and left tail is 2.5%.

summaryresult$ciw.lower.95 <- summaryresult$mean - qt(0.975,summaryresult$len-1)*summaryresult$se
summaryresult$ciw.upper.95 <- summaryresult$mean + qt(0.975,summaryresult$len-1)*summaryresult$se
summaryresult
```

```
##        Group     mean         sd         se length ciw.lower.95
## 1 Condition1 26.23632 0.10396539 0.06002444      3     25.97805
## 2 Condition2 26.00661 0.16268179 0.09392438      3     25.60248
## 3 Condition3 23.25609 0.09467798 0.05466236      3     23.02090
## 4 Condition4 20.97056 0.73140174 0.42227499      3     19.15366
##   ciw.upper.95
## 1     26.49458
## 2     26.41073
## 3     23.49128
## 4     22.78746
```

```
# mean with 95% two-sided confidence interval
ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)+
      geom_errorbar(aes(ymax = ciw.upper.95, ymin=ciw.lower.95), width=0.1)+
      labs(title="Mean with 95% confidence interval", x="Condition", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.position = 'bottom',
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```

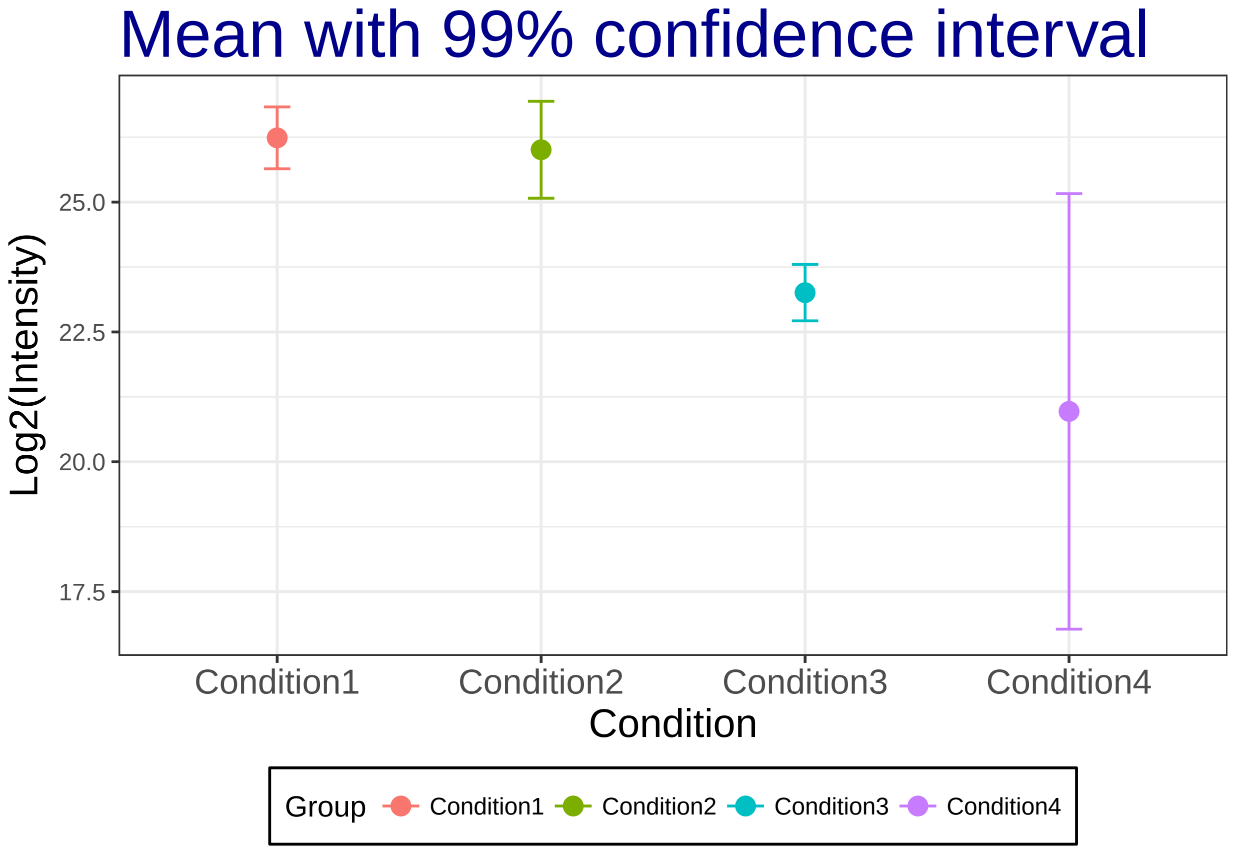


Let’s repeat that one more time for the 99% two-sided confidence interval.

```
# mean with 99% two-sided confidence interval
summaryresult$ciw.lower.99 <- summaryresult$mean - qt(0.995,summaryresult$len-1)*summaryresult$se
summaryresult$ciw.upper.99 <- summaryresult$mean + qt(0.995,summaryresult$len-1)*summaryresult$se
summaryresult
```

```
##        Group     mean         sd         se length ciw.lower.95
## 1 Condition1 26.23632 0.10396539 0.06002444      3     25.97805
## 2 Condition2 26.00661 0.16268179 0.09392438      3     25.60248
## 3 Condition3 23.25609 0.09467798 0.05466236      3     23.02090
## 4 Condition4 20.97056 0.73140174 0.42227499      3     19.15366
##   ciw.upper.95 ciw.lower.99 ciw.upper.99
## 1     26.49458     25.64058     26.83205
## 2     26.41073     25.07442     26.93879
## 3     23.49128     22.71357     23.79860
## 4     22.78746     16.77955     25.16157
```

```
ggplot(aes(x=Group, y=mean, colour=Group), data=summaryresult)+
      geom_point(size=3)+
      geom_errorbar(aes(ymax = ciw.upper.99, ymin=ciw.lower.99), width=0.1)+
      labs(title="Mean with 99% confidence interval", x="Condition", y='Log2(Intensity)')+
      theme_bw()+
      theme(plot.title = element_text(size=25, colour="darkblue"),
            axis.title.x = element_text(size=15),
            axis.title.y = element_text(size=15),
            axis.text.x = element_text(size=13),
            legend.position = 'bottom',
            legend.background = element_rect(colour='black'),
            legend.key = element_rect(colour='white'))
```



***comment*** : Let's compare all three versions of the means with different error bars. Every gap between error bars is different. Furthermore, error bars with SD and CI are overlapping between groups! Error bars for SD show the spread of the population while error bars based on SE reflect the uncertainty in the mean and depend on the sample size. Confidence intervals of `n` on the other hand mean that the interval captures the population mean `n %` of the time. When the sample size increases, CI and SE are getting closer to each other.