Day 5 - Descriptive Statistics

Chandan Kumar Pandey

2022-10-09

In this section we will finally see the basic application of R programming for our data analysis. After getting the cleaned data set, the first set need us to describe the data. This must be done using graphical representation. Using out Student performance data set.

```
##reading the data set.
##note that if you directory is not set to place where data is stores.
## you need to use setwd() to set it first.
Student_performance <- read.csv("StudentsPerformance.csv",header = T)
head(Student_performance)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B           bachelor’s degree     standard
## 2 female        group C                some college     standard
## 3 female        group B             master’s degree     standard
## 4   male        group A          associate’s degree free/reduced
## 5   male        group C                some college     standard
## 6 female        group B          associate’s degree     standard
##   test.preparation.course math.score reading.score writing.score
## 1                    none         72            72            74
## 2               completed         69            90            88
## 3                    none         90            95            93
## 4                    none         47            57            44
## 5                    none         76            78            75
## 6                    none         71            83            78
```

Descriptive Statistics

As the name suggest, it is use to describe the basic feature of the data. They provide simple summary for your data set without inferring any outcome to it. We can summaries the data in tabular or graphical for.

For Example.

```
## average marks of the students.
Student_performance$average <- round(rowSums(Student_performance[,6:8])/3,2)
##grouping the data by gender.
mean_avg_marks <- tapply(Student_performance$average,Student_performance$gender,FUN = mean)
sd_avg_marks <- tapply(Student_performance$average,Student_performance$gender,FUN = sd)
med_avg_marks <- tapply(Student_performance$average,Student_performance$gender,FUN = median)
count_numbers <- tapply(Student_performance$average,Student_performance$gender,FUN = length)
marks_avg_gender <- data.frame(mean_avg_marks,sd_avg_marks,med_avg_marks,count_numbers)
## getting results as output
write.csv(marks_avg_gender,"mark_by_gender.csv")
```

A help from pacakge dplyr.

A package called dplyr can help to summaries your data more effectively.

```
#checking if dplyr is installed or not. If not then installing it.
if(!require(dplyr)){
    install.packages("dplyr",dependencies = T)
    library(dplyr)
}
```

```
## Loading required package: dplyr
```

```
##
## 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
```

```
mark_summary <- group_by(Student_performance,gender)%>%
  summarise(Math.avg = mean(math.score), Math.sd = sd(math.score),
            writing.mean = mean(writing.score),writng.sd = sd(writing.score),
            reading.mean = mean(reading.score),reading.sd = sd(reading.score),
            overall.mean = mean(average),overall.sd = sd(average),
            count = n())%>%
  as.data.frame()
write.csv(mark_summary,"mark_summary.csv")
```

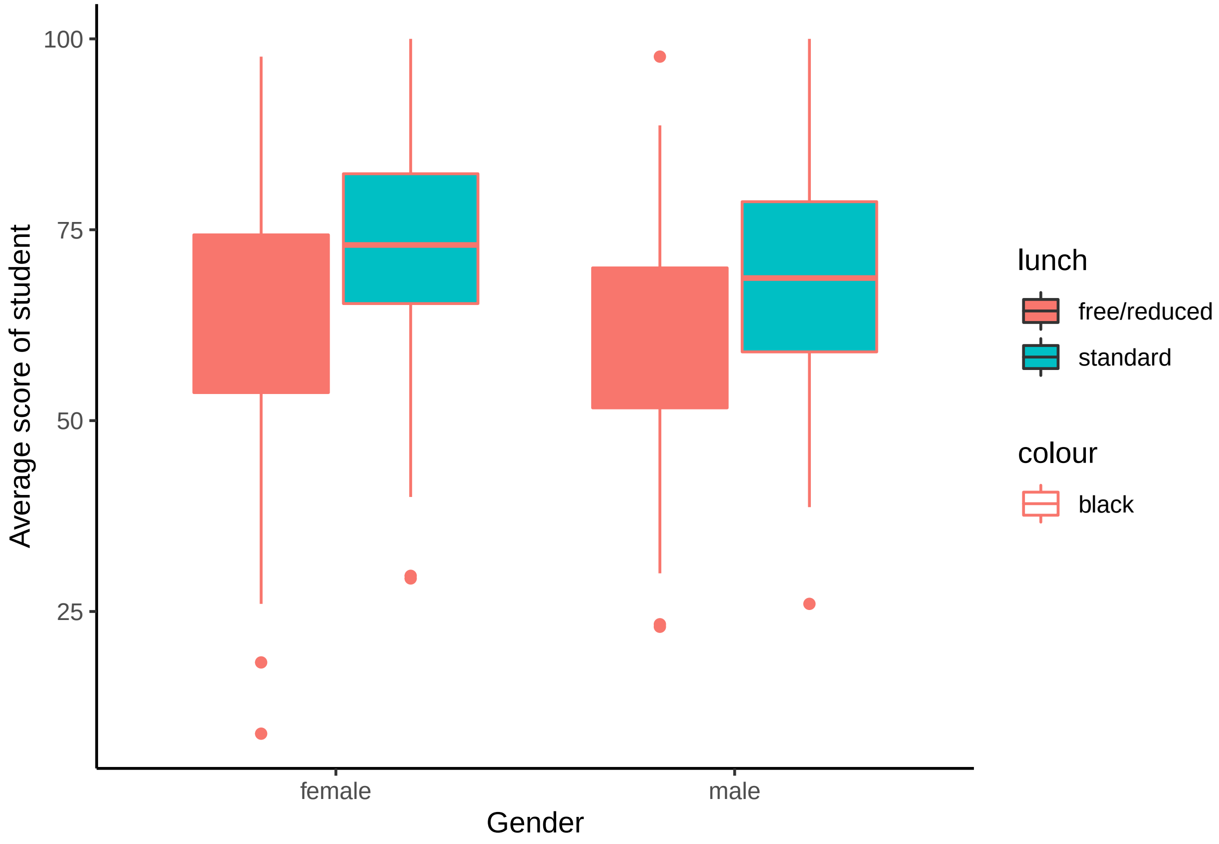
Plots for summary

We will use another package for piloting which is very common these days. It is called ggplot2

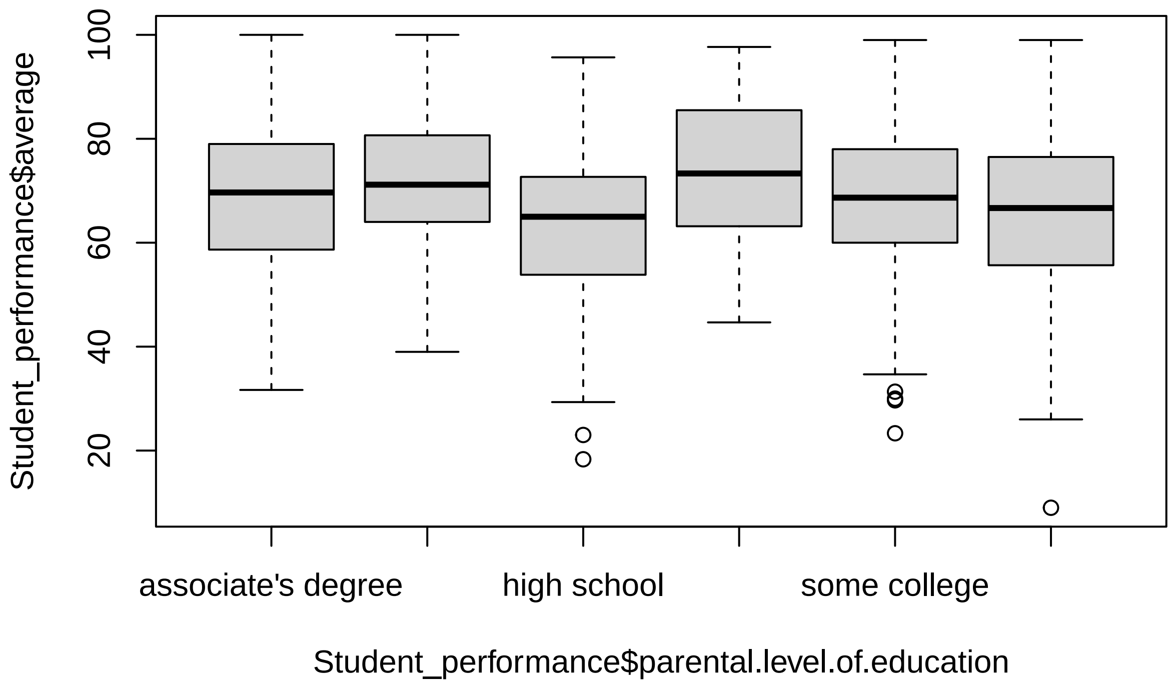
```
#checking if ggplot2 is installed or not. If not then installing it.
if(!require(ggplot2)){
    install.packages("ggplot2",dependencies = T)
    library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```
##Box plot
box_plot<-ggplot(data = Student_performance,aes(x=gender,y=average,fill=lunch))
box_plot+geom_boxplot(aes(color="black"),show.legend = T)+
  theme_classic()+
  labs(y="Average score of student",x="Gender")
```



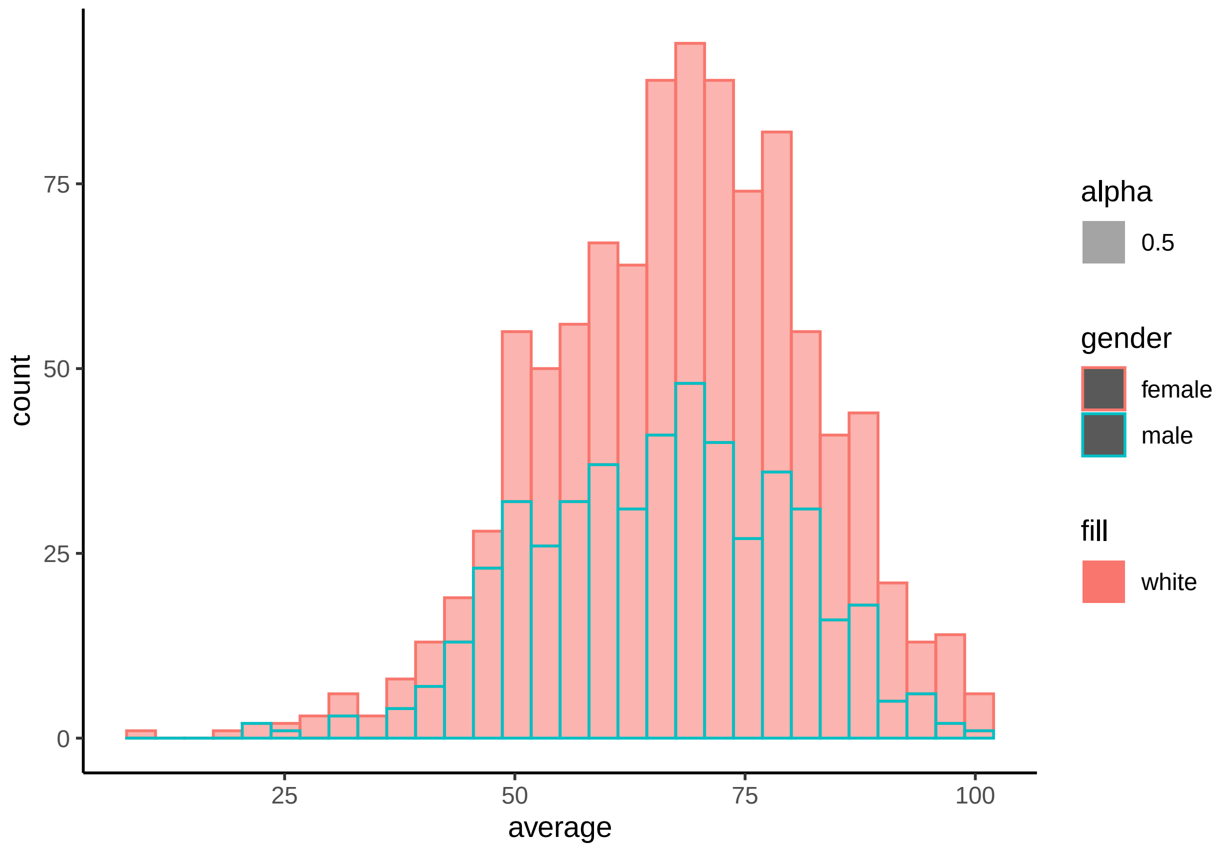
```
##another way for box plot
boxplot(Student_performance$average~Student_performance$parental.level.of.education)
```



```
##histogram
hist_marks <- ggplot(data = Student_performance, aes(x=average,col=gender))
hist_marks + geom_histogram(aes(position="identity",fill="white",binwidth = 10,alpha=0.5),
                            show.legend = T)+theme_classic()
```

```
## Warning: Ignoring unknown aesthetics: position, binwidth
```

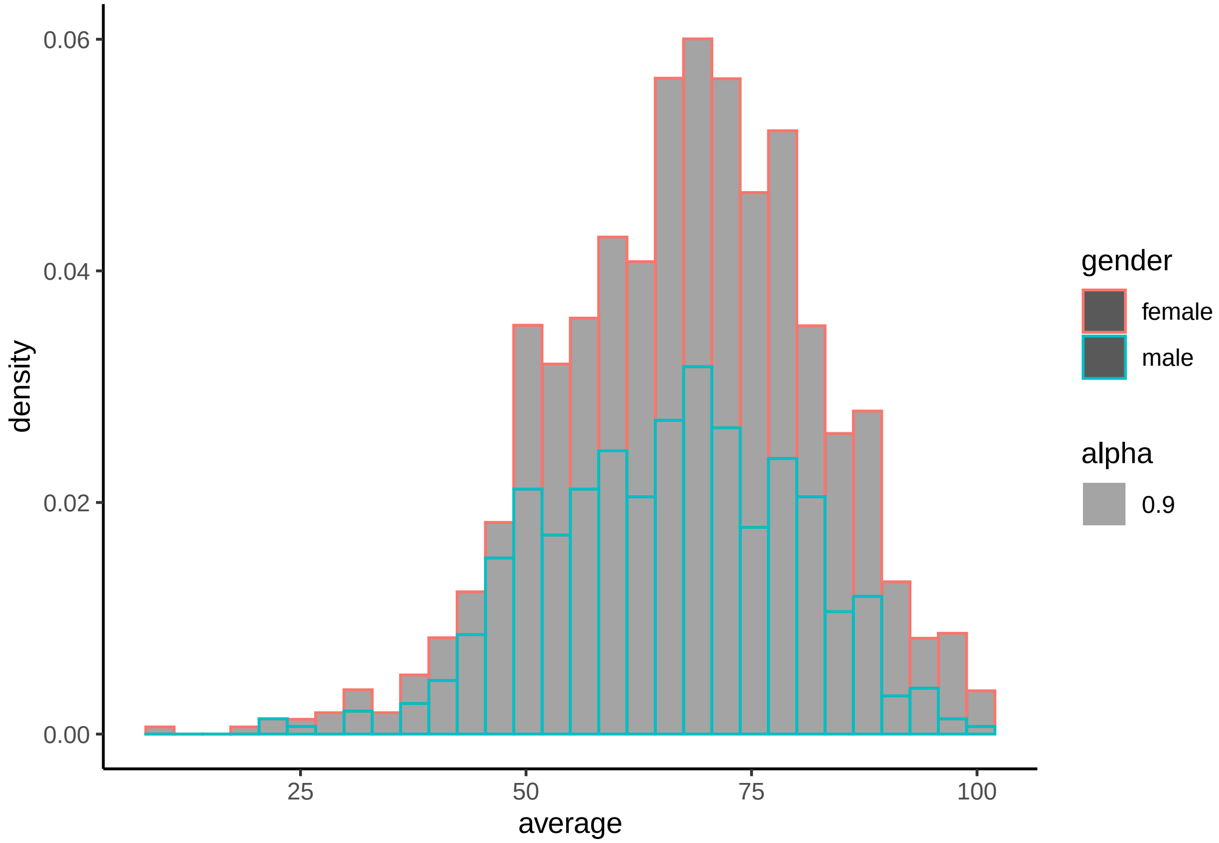
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
##with density not actual count
hist_marks +
  geom_histogram(aes(y = ..density..,
                                 position="identity",binwidth = 10,alpha=0.9),
                 show.legend = T)+theme_classic()
```

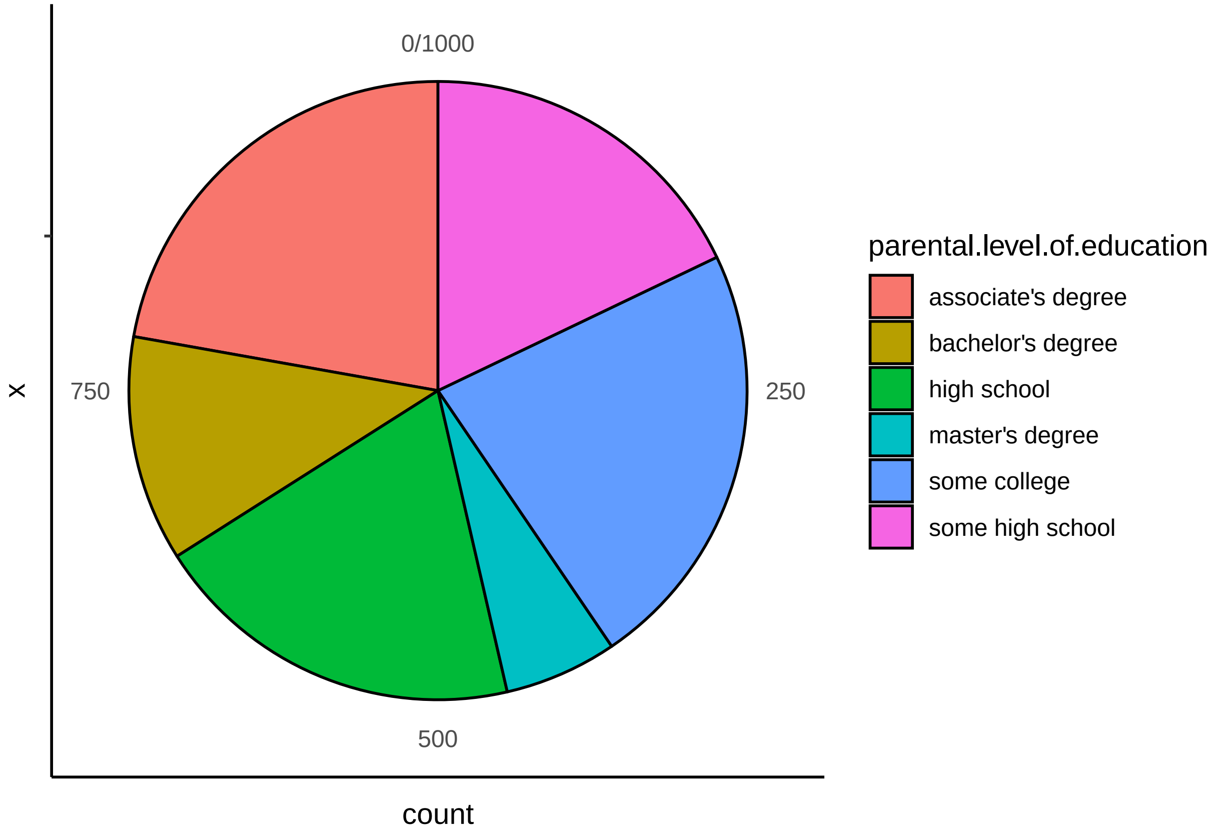
```
## Warning: Ignoring unknown aesthetics: position, binwidth
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



pie chart

```
pie_data<-group_by(Student_performance,parental.level.of.education)%>%
  summarise(count=n())
pie_parent_edu <- ggplot(data = pie_data,aes(x="",y=count,fill=parental.level.of.education))
pie_parent_edu+geom_col(color="black")+coord_polar(theta = "y")+theme_classic()
```



ANNOVA test.

Analysis of variance is statically test which is used to see if two group are different from each other. There are some assumption in the which we need to take consideration of

1. Independence of case
2. Normality
3. Homogeneity

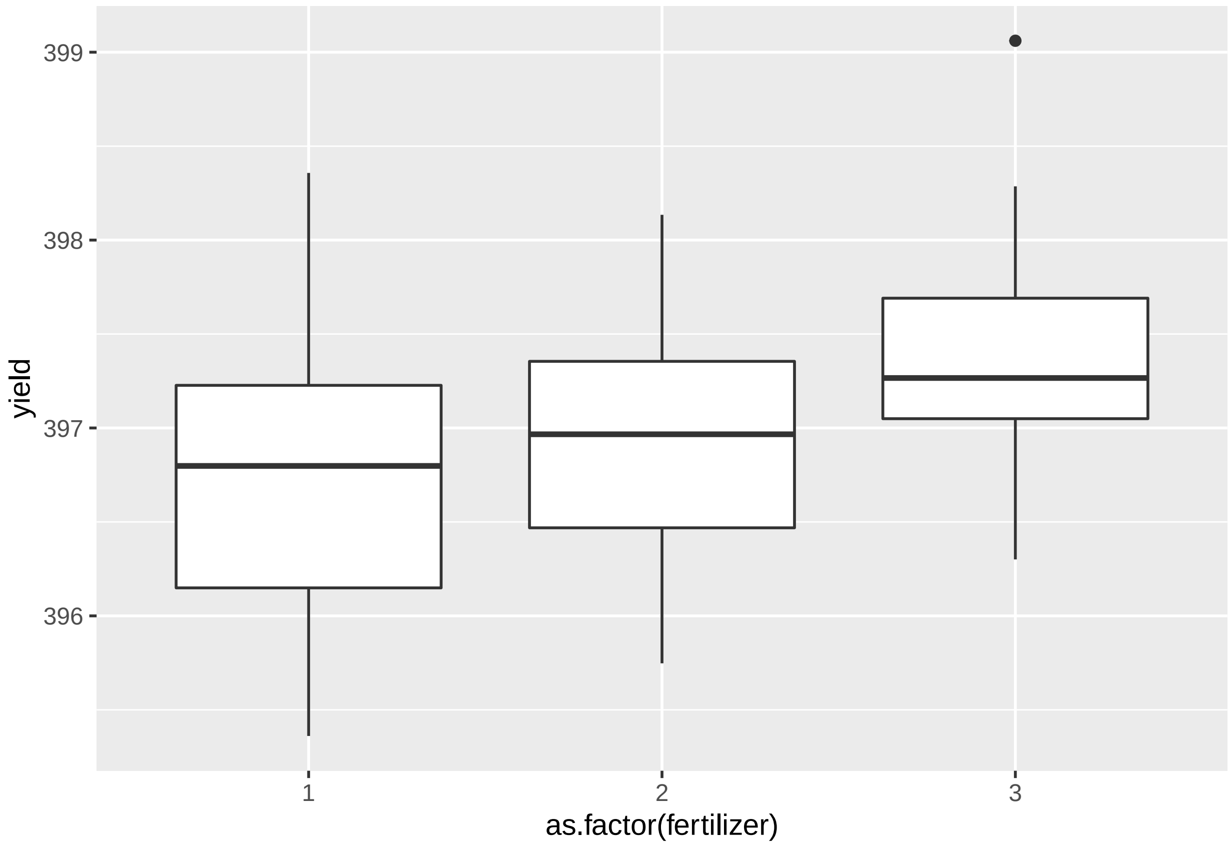
```
##load the data
Yeild <- read.csv("./crop_yeild/field_data.csv",header = T)
## checking the summary of the data
summary(Yeild)
```

```
##      height          temp          humidity       fertilizer      yield
##  Min.   :1.0   Min.   :60.00   Min.   :60.00   Min.   :1    Min.   :395.4
##  1st Qu.:1.0   1st Qu.:62.00   1st Qu.:64.75   1st Qu.:1    1st Qu.:396.5
##  Median :1.5   Median :66.00   Median :69.00   Median :2    Median :397.1
##  Mean   :1.5   Mean   :65.16   Mean   :70.03   Mean   :2    Mean   :397.0
##  3rd Qu.:2.0   3rd Qu.:68.00   3rd Qu.:77.00   3rd Qu.:3    3rd Qu.:397.4
##  Max.   :2.0   Max.   :70.00   Max.   :80.00   Max.   :3    Max.   :399.1
##      region
##  Min.   :1.00
##   1st Qu.:1.75
##   Median :2.50
##   Mean   :2.50
##   3rd Qu.:3.25
##   Max.   :4.00
```

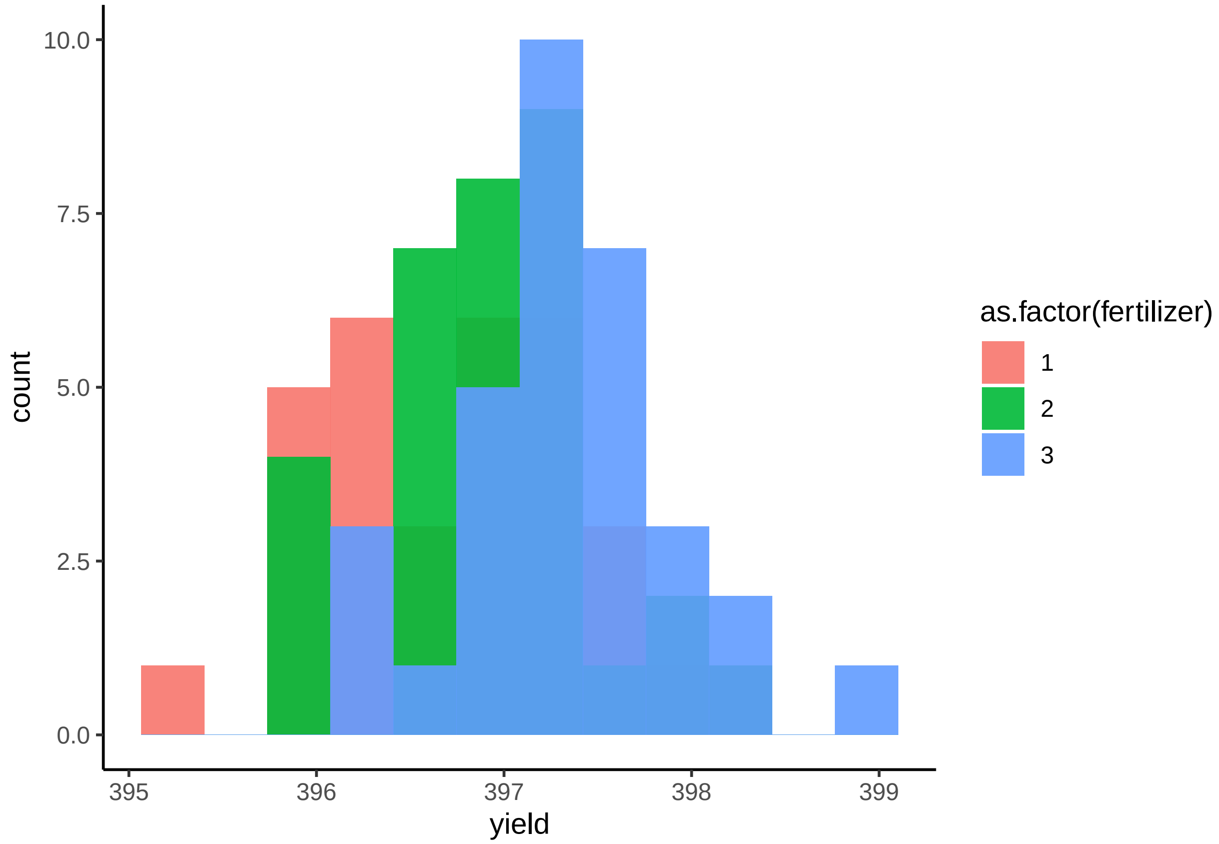
```
##checking the structure of the data
str(Yeild)
```

```
## 'data.frame':    96 obs. of  6 variables:
##  $ height    : int  1 2 1 2 1 2 1 2 1 2 ...
##  $ temp      : int  70 68 69 69 70 68 67 64 70 64 ...
##  $ humidity  : int  72 76 63 65 76 67 77 74 67 62 ...
##  $ fertilizer: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yield     : num  397 398 396 398 397 ...
##  $ region    : int  1 2 3 4 1 2 3 4 1 2 ...
```

```
##basic description of the data.
library(ggplot2)
#boxplot
plot_box<-ggplot(data = Yeild,aes(x=as.factor(fertilizer),y=yield))
plot_box+geom_boxplot()
```



```r
#histogram
plot_hist<-ggplot(data = Yeild,aes(x=yield,fill=as.factor(fertilizer)))
plot_hist+geom_histogram(stat = "bin",position="identity",bins=12,alpha=0.9,
                         show.legend = T)+
  theme_classic()
```



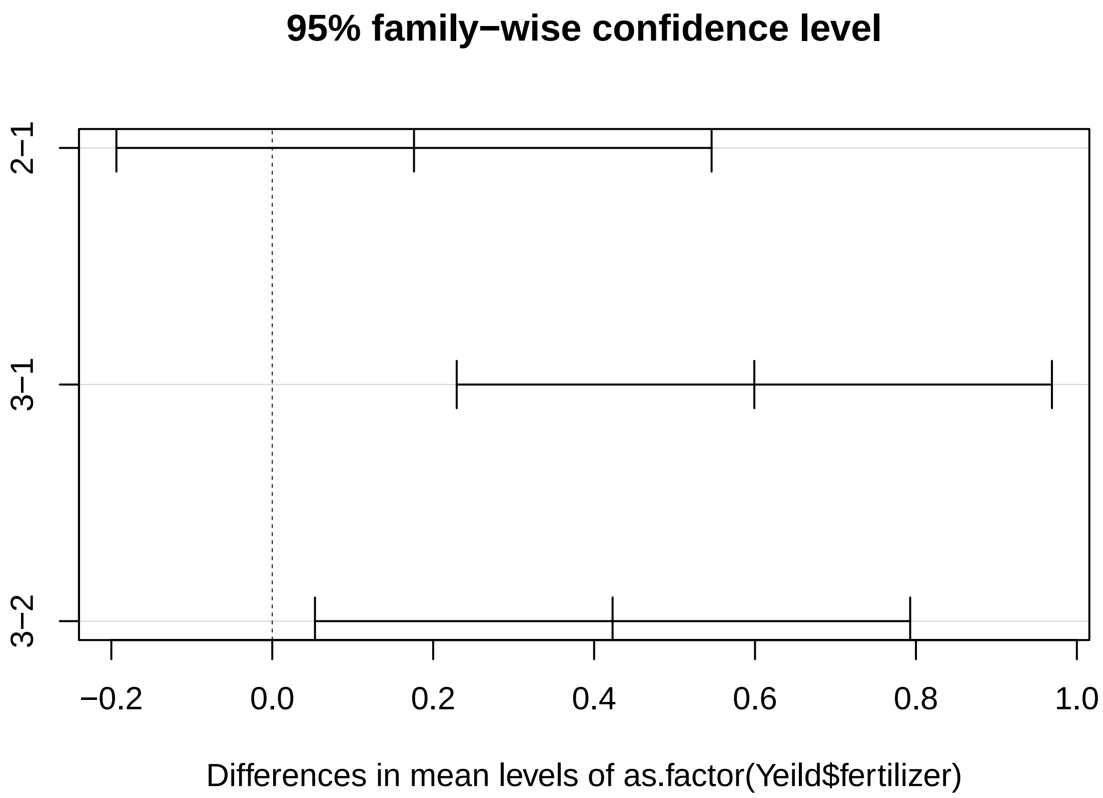
```r
#Shapiro-Wilk normality test
shapiro.test(Yeild$yield)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Yeild$yield
## W = 0.989, p-value = 0.6135
```

```r
mod1<-aov(formula = Yeild$yield ~ as.factor(Yeild$fertilizer))
summary(mod1)
```

```
##                             Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Yeild$fertilizer)  2   6.07  3.0340   7.863  7e-04 ***
## Residuals                   93  35.89  0.3859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#posthoc
Post_hoc <- TukeyHSD(mod1)
plot(Post_hoc)
```


95% family−wise confidence level
Differences in mean levels of as.factor(Yeild$fertilizer)