

# HW3\_tianjiang

Tian Jiang

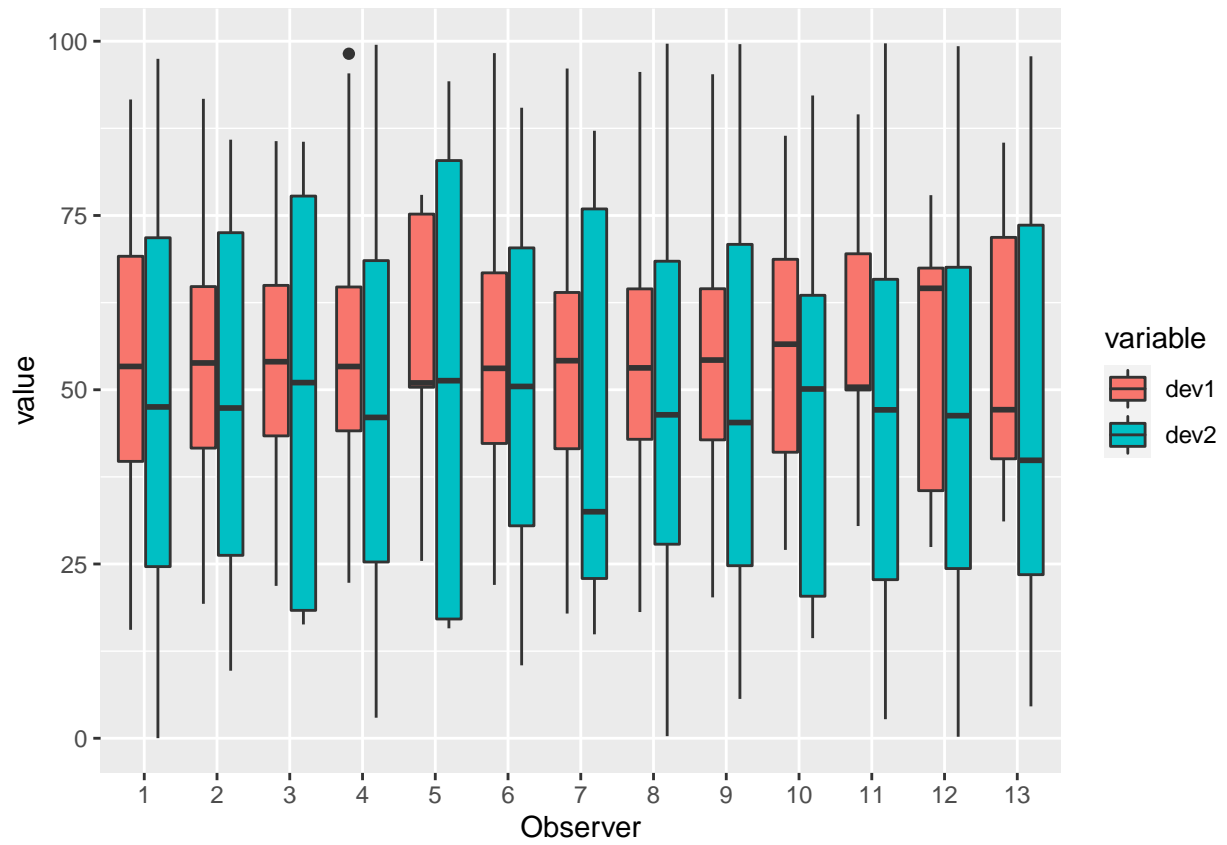
9/26/2020

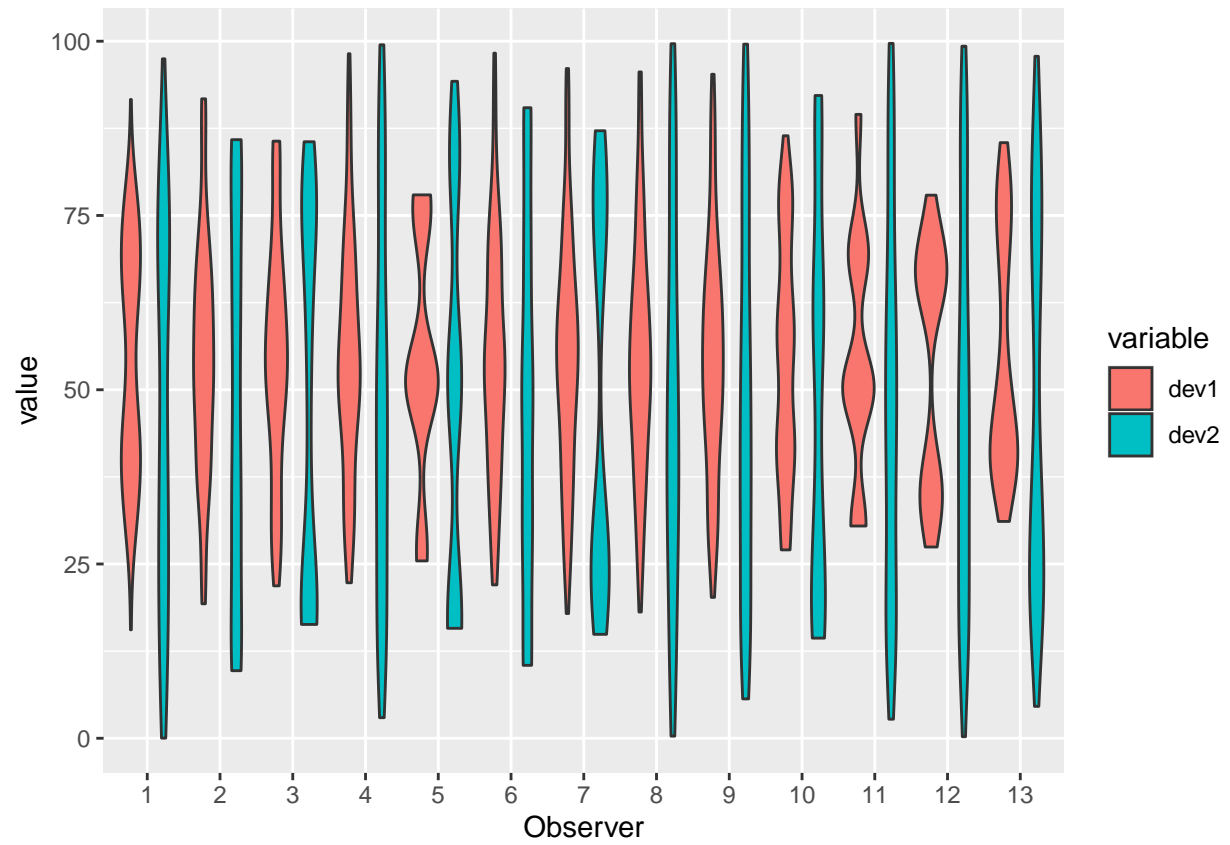
## P5

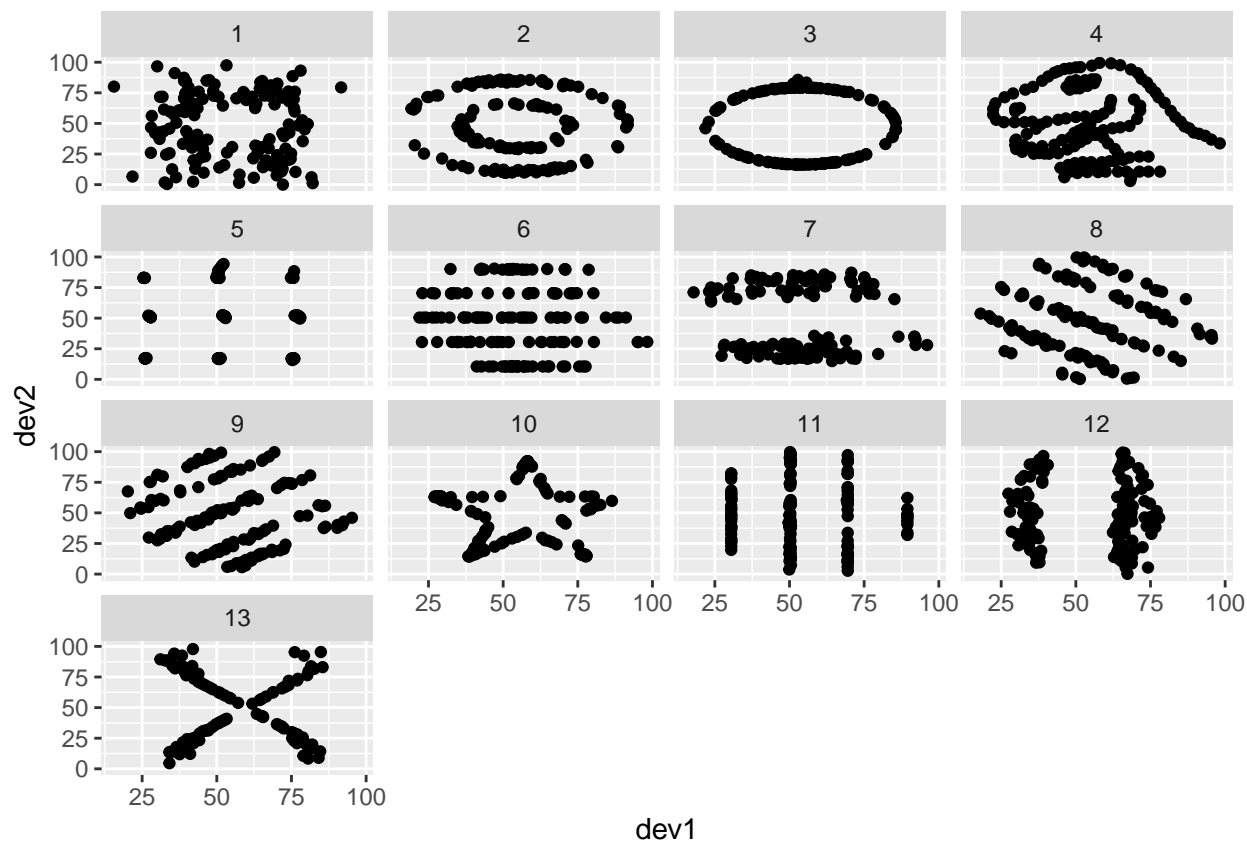
```
b0<-readRDS("/Users/mac/Downloads/HW3_data.rds")
p5 <- function(df) {
  table<-df %>% group_by(Observer)%>% summarize(MeanDev1 = mean(dev1),MeanDev2 = mean(dev2),SdDev1=sd(d
  print(kable(table))
  b <- melt(b0,id.vars=1)
  b$Observer<-factor(b$Observer)
  print(ggplot(b, aes(x=Observer, y=value, fill=variable)) +
    geom_boxplot())
  print(ggplot(b, aes(x=Observer, y=value, fill=variable)) +
    geom_violin())
  print(ggplot(b0, aes(x=dev1,y=dev2)) + geom_point() + facet_wrap(Observer~.))
}
p5(b0)
```

## 'summarise()' ungrouping output (override with '.groups' argument)

```
##
##
## | Observer| MeanDev1| MeanDev2| SdDev1| SdDev2| Correlation|
## |-----:|-----:|-----:|-----:|-----:|-----:|
## | 1| 54.26610| 47.83472| 16.76983| 26.93974| -0.0641284|
## | 2| 54.26873| 47.83082| 16.76924| 26.93573| -0.0685864|
## | 3| 54.26732| 47.83772| 16.76001| 26.93004| -0.0683434|
## | 4| 54.26327| 47.83225| 16.76514| 26.93540| -0.0644719|
## | 5| 54.26030| 47.83983| 16.76774| 26.93019| -0.0603414|
## | 6| 54.26144| 47.83025| 16.76590| 26.93988| -0.0617148|
## | 7| 54.26881| 47.83545| 16.76670| 26.94000| -0.0685042|
## | 8| 54.26785| 47.83590| 16.76676| 26.93610| -0.0689797|
## | 9| 54.26588| 47.83150| 16.76885| 26.93861| -0.0686092|
## |10| 54.26734| 47.83955| 16.76896| 26.93027| -0.0629611|
## |11| 54.26993| 47.83699| 16.76996| 26.93768| -0.0694456|
## |12| 54.26692| 47.83160| 16.77000| 26.93790| -0.0665752|
## |13| 54.26015| 47.83972| 16.76996| 26.93000| -0.0655833|
```







Analysis: The means of two variables don't vary much among different groups. Also Dev1 and Dev2 have slightly negative correlations and the correlation coefficient of different groups are very close. Boxplots support these conclusions, too. And there's no outlier through out the data except dev1 of group 4. Also from violin plots we see the range of dev2 is always larger than that of dev1 in every group. And frequencies of different values spread more evenly for dev2.

Comprehensively we conclude that data of different groups are distributed very similarly.

However, as the last plot shows, my conclusion is totally wrong. The lesson is that we should have an viewable and intuitive understanding of the data in the first place so that we hace a basic idea of the relationship between variables, i.e do the job from the very basic step. Data visualisation is very important! Also only looking at a small number of statistics might result in wrong understanding. We should refer to more kinds of analysis results and think about the situation from different perspectives.

## P6

```
solution<-0.3413*(2*pi)^0.5
j<-c()
Probability <- function(start_value, end_value, step)
{g<-seq(start_value,end_value,step)
Kdens<-exp(-g^2/2)
probability = sum(Kdens * step) # Approximate the integral of the PDF
print("width")
print(step)
print("result")}
```

```

print(probability)
print("error")
error<-abs(probability-solution)
print(error)}
for (i in 2:8)
{re<-Probability(0,1,10^(-i))
#if (abs(re-solution)<10^(-6))
#{j<-c(j,re)}
}

```

```

## [1] "width"
## [1] 0.01
## [1] "result"
## [1] 0.863652
## [1] "error"
## [1] 0.008139761
## [1] "width"
## [1] 0.001
## [1] "result"
## [1] 0.8564276
## [1] "error"
## [1] 0.0009153765
## [1] "width"
## [1] 1e-04
## [1] "result"
## [1] 0.8557047
## [1] "error"
## [1] 0.0001924878
## [1] "width"
## [1] 1e-05
## [1] "result"
## [1] 0.8556324
## [1] "error"
## [1] 0.0001201944
## [1] "width"
## [1] 1e-06
## [1] "result"
## [1] 0.8556252
## [1] "error"
## [1] 0.000112965
## [1] "width"
## [1] 1e-07
## [1] "result"
## [1] 0.8556245
## [1] "error"
## [1] 0.0001122421
## [1] "width"
## [1] 1e-08
## [1] "result"
## [1] 0.8556244
## [1] "error"
## [1] 0.0001121698

```

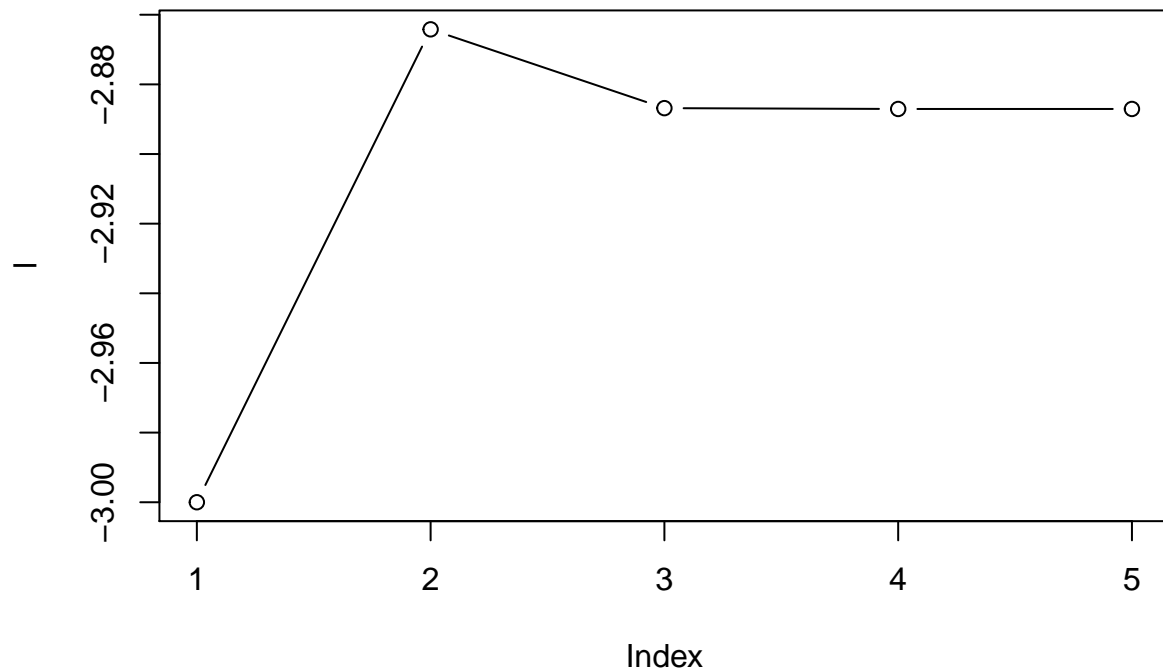
```
#print("j")
#print(j)
```

My R dies everytime I set the step less than  $10^{-8}$ . So I'm sorry it can't generate a result with the error less than  $10^{-6}$

## P7

```
g <- function(x) {3^x -sin(x)+cos(5*x)}
gPrime <- function(x) {3^x*log(3)-cos(x)-5*sin(5*x)}
guess <- (-3)
tolerance <- .00001
l<-c()
root <- function(g, gPrime, guess, tolerance) {
  x = guess
  l<-c(1,x)
  print(x)
  while ((abs(g(x))) > tolerance) {
    x = x - g(x)/gPrime(x)
    print(x)
    l<-c(1,x)}
  print(x)
  l<-c(1,x)
  plot(l,type="b")}
root(g, gPrime, guess, tolerance)
```

```
## [1] -3
## [1] -2.864196
## [1] -2.886825
## [1] -2.887058
## [1] -2.887058
```



## P8

```
set.seed(1)
X <- cbind(rep(1,100),rep.int(1:10,time=10))
beta <- c(4,5)
y <- X%%beta + rnorm(100)
ybar<-mean(y)
sum=0
calsum<-function(x){
  for (i in 1:100)
  {sum<-sum+(y[i]-ybar)^2}
  sum}
calmat<-function(o)
{a<-(t(as.vector(y)))%*%((diag(100))-1/100*matrix(1, 100, 100))%*%(as.vector(y))
a}
print("Results")
```

```
## [1] "Results"
```

```
calsum(x)
```

```
## [1] 20628.38
```

```
calmat(x)[1,1]
```

```
## [1] 20628.38
```

```
times<-microbenchmark(calsum(x),calmat(x),times=100)  
autoplot(times)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

