HW 1

Shiloh Bradley

6/9/2020

Part A

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

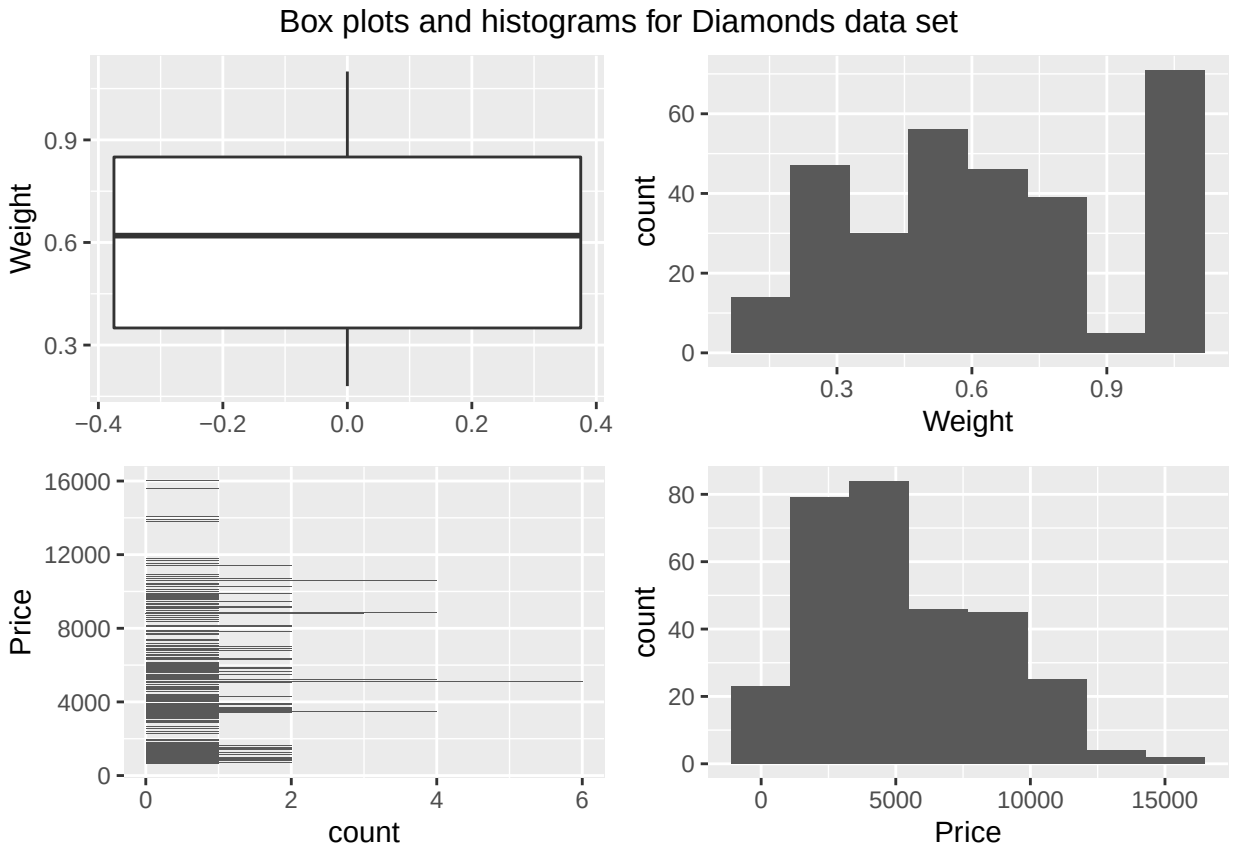
```
library(gridExtra)

setwd("~/Desktop/Personal Computer/MBA/HOA 732/")
```

```
D <- read.csv("Diamonds.csv")
head(D)
```

```
##   Weight Color Clarity Cert_Body Price
## 1   0.30     D     VS2       GIA  1302
## 2   0.30     E     VS1       GIA  1510
## 3   0.30     G    VVS1       GIA  1510
## 4   0.30     G     VS1       GIA  1260
## 5   0.31     D     VS1       GIA  1641
## 6   0.31     E     VS1       GIA  1555
```

```
B1 <- ggplot(D, aes(y = Weight)) +
      geom_boxplot()
H1 <- ggplot(D, aes(y = Weight)) +
      geom_histogram(bins = 8) +
      coord_flip()
B2 <- ggplot(D, aes(y = Price)) +
      geom_bar()
H2 <- ggplot(D, aes(y = Price)) +
      geom_histogram(bins=8) +
      coord_flip()
```

```
grid.arrange(B1, H1, B2, H2, ncol = 2, top = "Box plots and histograms for Diamonds data set")
```




```
tapply(D$Weight, D$Clarity, summary)
```

```
## $IF
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.2175  0.2650  0.3718  0.5050  1.0400
##
## $VS1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.4000  0.6000  0.6469  0.9000  1.0400
##
## $VS2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.5700  0.8200  0.7583  1.0000  1.1000
##
## $VVS1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.5075  0.6300  0.6398  0.7800  1.0100
##
## $VVS2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.5100  0.6750  0.6679  0.8900  1.0900
```

```
tapply(D$Weight, D$Cert_Body, summary)
```

```
## $GIA
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3000  0.5000  0.7000  0.6723  0.8950  1.1000
##
```

```
## $HRD
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5000  0.6550  0.8100  0.8129  1.0000  1.0900
##
## $IGI
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.2100  0.2900  0.3665  0.4775  1.0100
```

```
tapply(D$Weight, D$Clarity, summary)
```

```
## $IF
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.2175  0.2650  0.3718  0.5050  1.0400
##
## $VS1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.4000  0.6000  0.6469  0.9000  1.0400
##
## $VS2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.5700  0.8200  0.7583  1.0000  1.1000
##
## $VVS1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.5075  0.6300  0.6398  0.7800  1.0100
##
## $VVS2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.5100  0.6750  0.6679  0.8900  1.0900
```

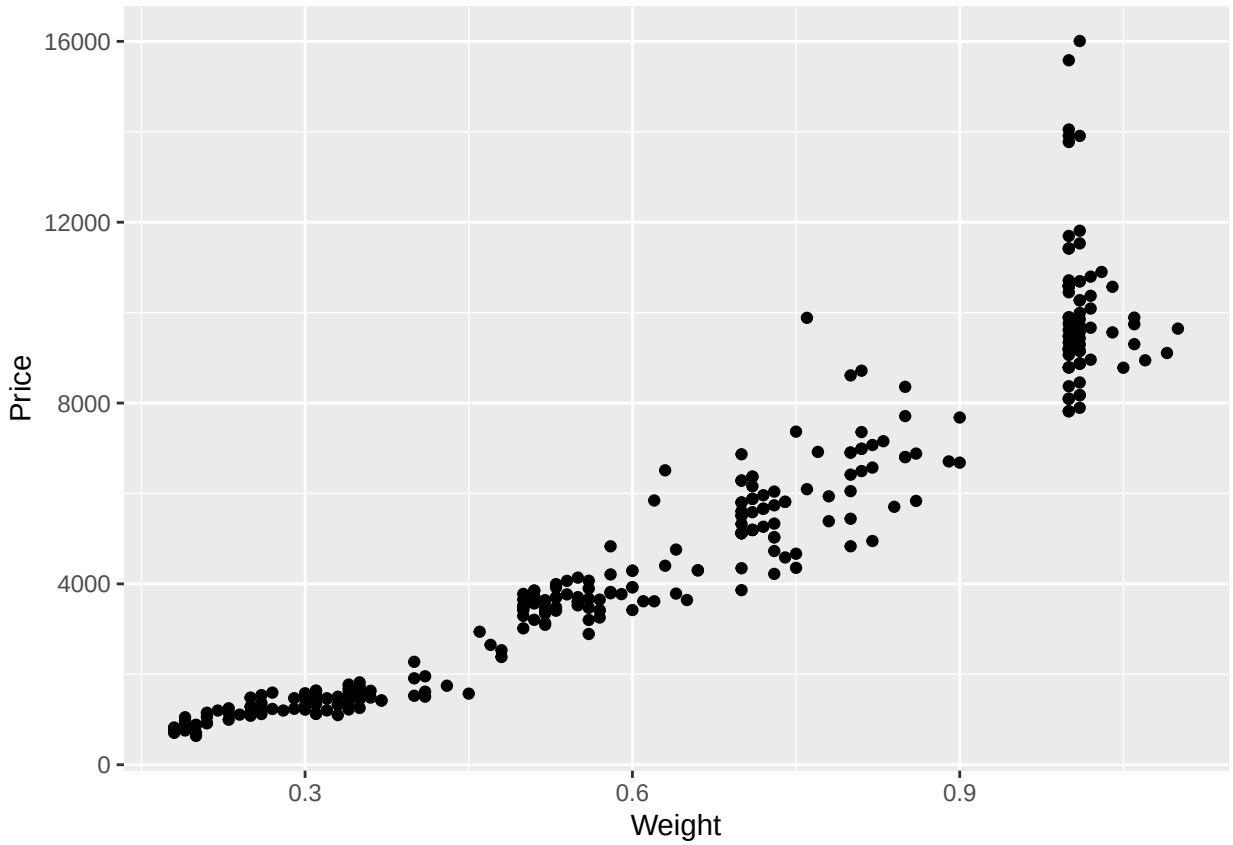
```
tapply(D$Weight, list(D$Clarity, D$Cert_Body), mean)
```

```
##            GIA       HRD       IGI
## IF   0.7633333 0.7250000 0.2611765
## VS1  0.6091803 0.8784615 0.5457143
## VS2  0.7480556 0.8033333 0.6050000
## VVS1 0.6853333 0.7613043 0.3914286
## VVS2 0.6836364 0.8475000 0.4380952
```

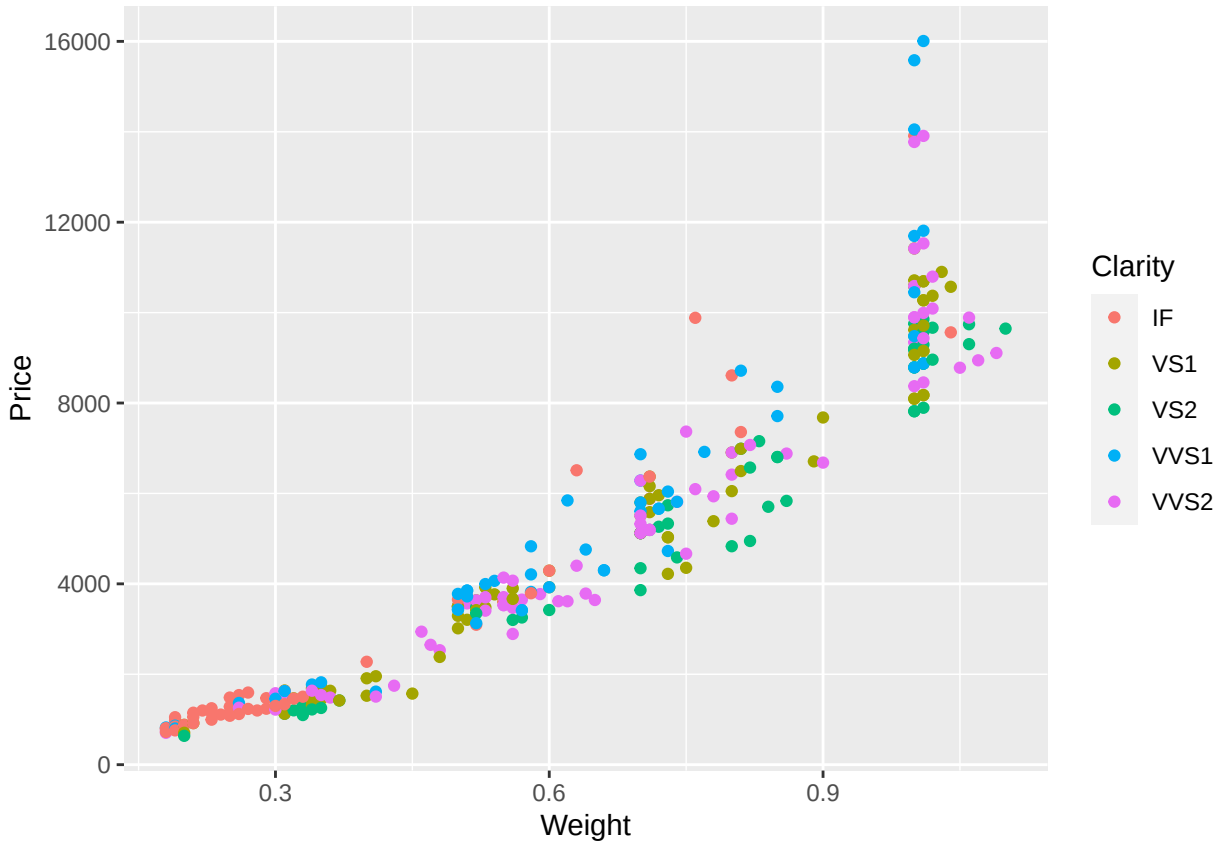
```
tapply(D$Weight, list(D$Clarity, D$Cert_Body), sd)
```

```
##            GIA       HRD       IGI
## IF   0.2109660 0.1066146 0.0806701
## VS1  0.2455015 0.1618047 0.3683683
## VS2  0.2778779 0.1818031 0.5727565
## VVS1 0.2027267 0.1932569 0.1706935
## VVS2 0.2108735 0.1868329 0.2272800
```

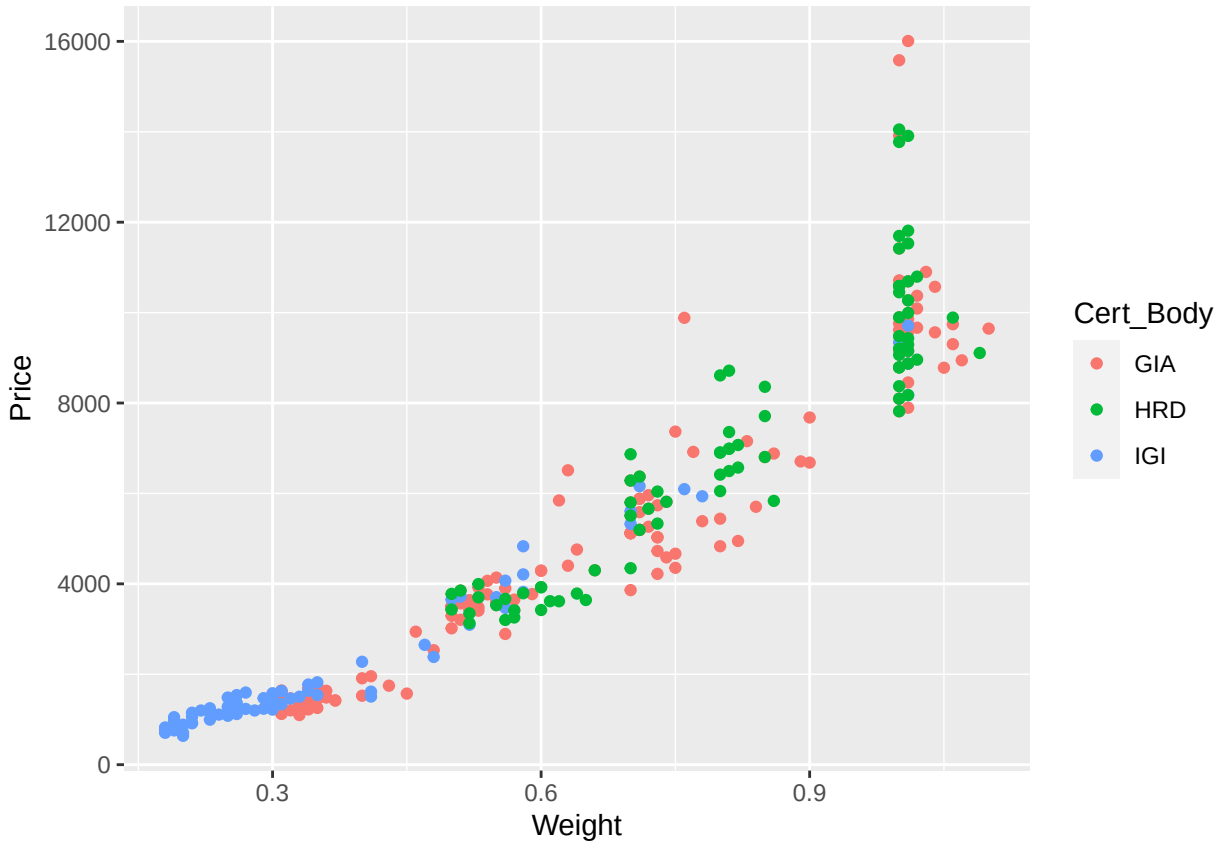
```
ggplot(D, aes(x = Weight, y = Price)) +
  geom_point()
```




```
ggplot(D, aes(x = Weight, y = Price, color = Clarity)) +
  geom_point()
```




```
ggplot(D, aes(x = Weight, y = Price, color = Cert_Body)) +
  geom_point()
```

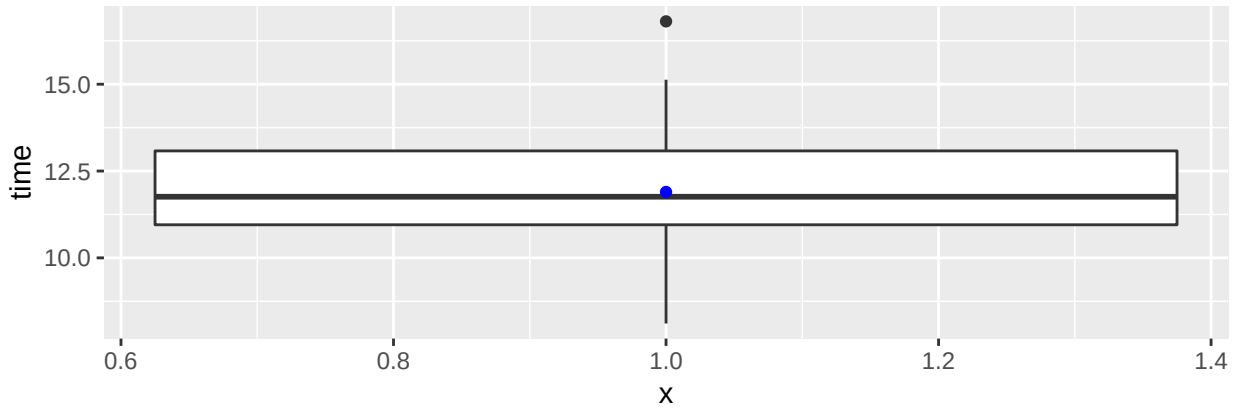


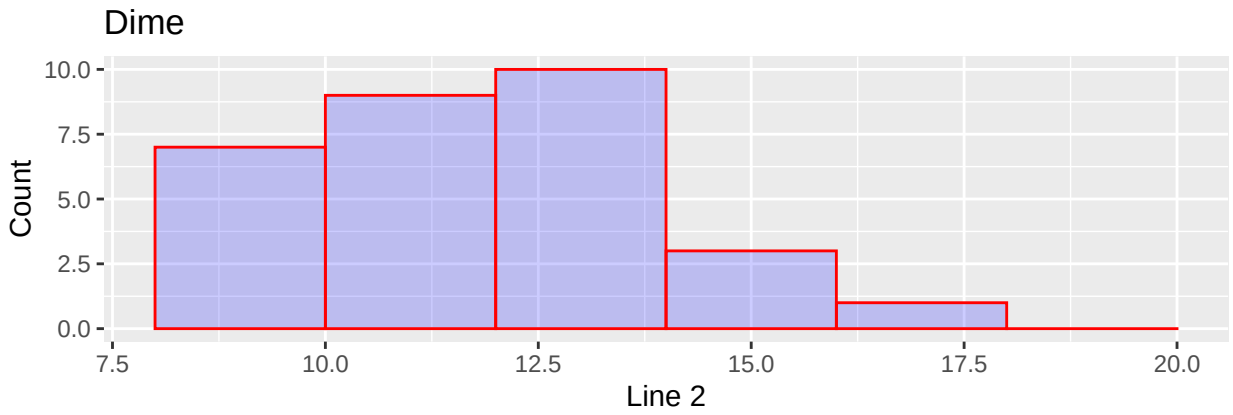
Part B

```
D <- read.csv("teller time.csv", header=TRUE)
BX2 <- ggplot(D,aes(x=1,y=time)) + geom_boxplot()+
stat_summary(fun.y = "mean", geom="point",color="blue")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
HISD2 <- ggplot(data=D, aes(time)) + geom_histogram(breaks=seq(8, 20, by = 2),
                 col="red",
                 fill="blue",
                 alpha = .2)+
                 labs(title="Dime") +
                  labs(x="Line 2", y="Count")

grid.arrange(BX2,HISD2,ncol=1, nrow =2)
```






```
summary(D)
```

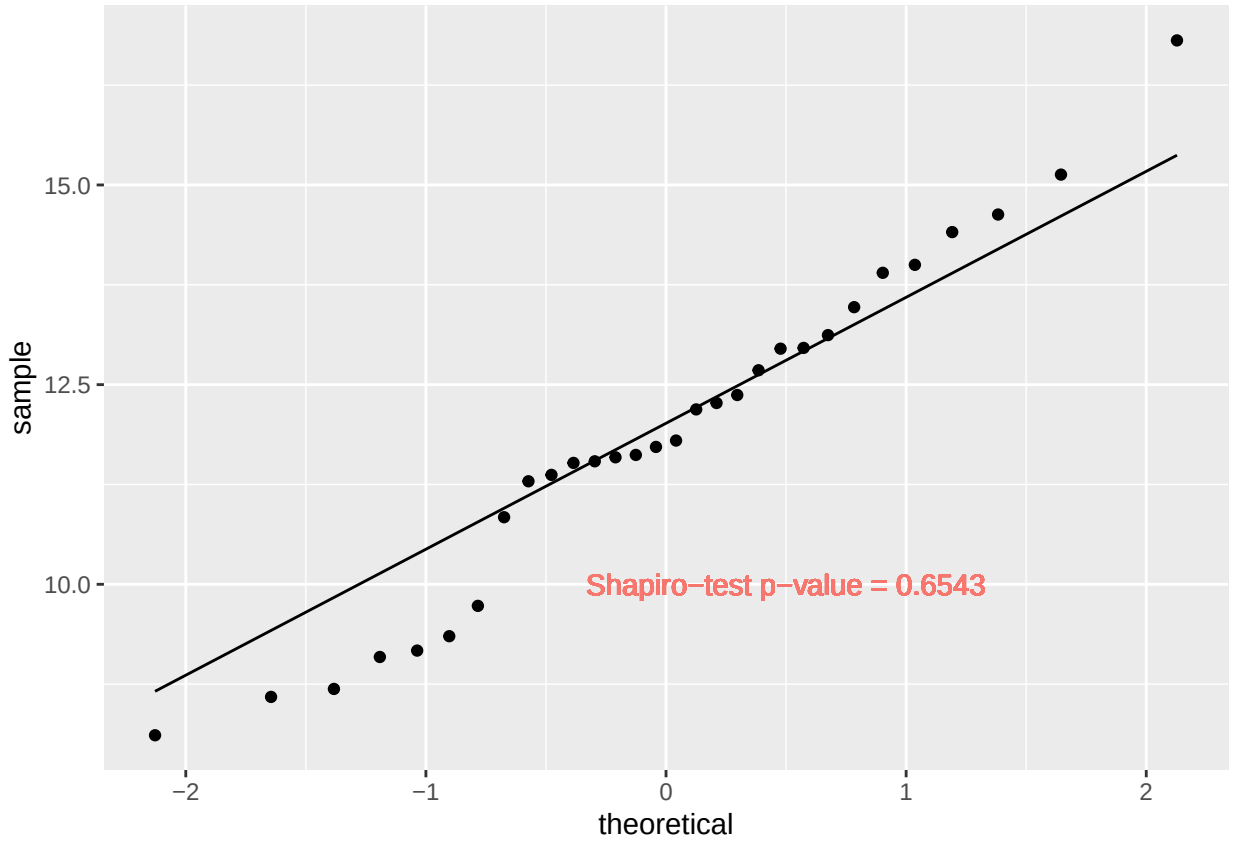
```
##       time
##  Min.   : 8.11
##  1st Qu.:10.95
##  Median :11.76
##  Mean   :11.90
##  3rd Qu.:13.08
##  Max.   :16.81
```

```
#Test h0:mu=10, H1:mu > 10
t.test(D$time,mu=10, alternative = "greater" )
```

```
##
##  One Sample t-test
##
## data:  D$time
## t = 4.9382, df = 29, p-value = 1.506e-05
## alternative hypothesis: true mean is greater than 10
## 95 percent confidence interval:
##  11.24428      Inf
## sample estimates:
## mean of x
##    11.897
```

```
#qq plot with normal line (normality test)
ggplot(D)+stat_qq(aes(sample=time)) +
  geom_qq_line(aes(sample=time))+
```

```
  geom_text(aes(x=0.5, y=10, color="red", label="Shapiro-test p-value = 0.6543"))+
  theme(legend.position="none")
```




```
shapiro.test(D$time)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  D$time
## W = 0.97373, p-value = 0.6453
```

```
x <- sum(D$time > 10)
n <- nrow(D)
#test if p > 0.80
prop.test(x, n, .80, alternative = "greater")          # continuity correction DRUE by default
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  x out of n, null probability 0.8
## X-squared = 0.052083, df = 1, p-value = 0.5903
## alternative hypothesis: true p is greater than 0.8
## 95 percent confidence interval:
##  0.6030711 1.0000000
## sample estimates:
##         p
## 0.7666667
```