LA2

sneha

2022-09-24

```
df = read.csv('stock.csv')
head(df)
```

```
##   Index       Date   Open   High    Low  Close Adj.Close Volume
## 1   NYA 31-12-1965 528.69 528.69 528.69 528.69    528.69      0
## 2   NYA 03-01-1966 527.21 527.21 527.21 527.21    527.21      0
## 3   NYA 04-01-1966 527.84 527.84 527.84 527.84    527.84      0
## 4   NYA 05-01-1966 531.12 531.12 531.12 531.12    531.12      0
## 5   NYA 06-01-1966 532.07 532.07 532.07 532.07    532.07      0
## 6   NYA 07-01-1966 532.60 532.60 532.60 532.60    532.60      0
```

```
library(dplyr)
```

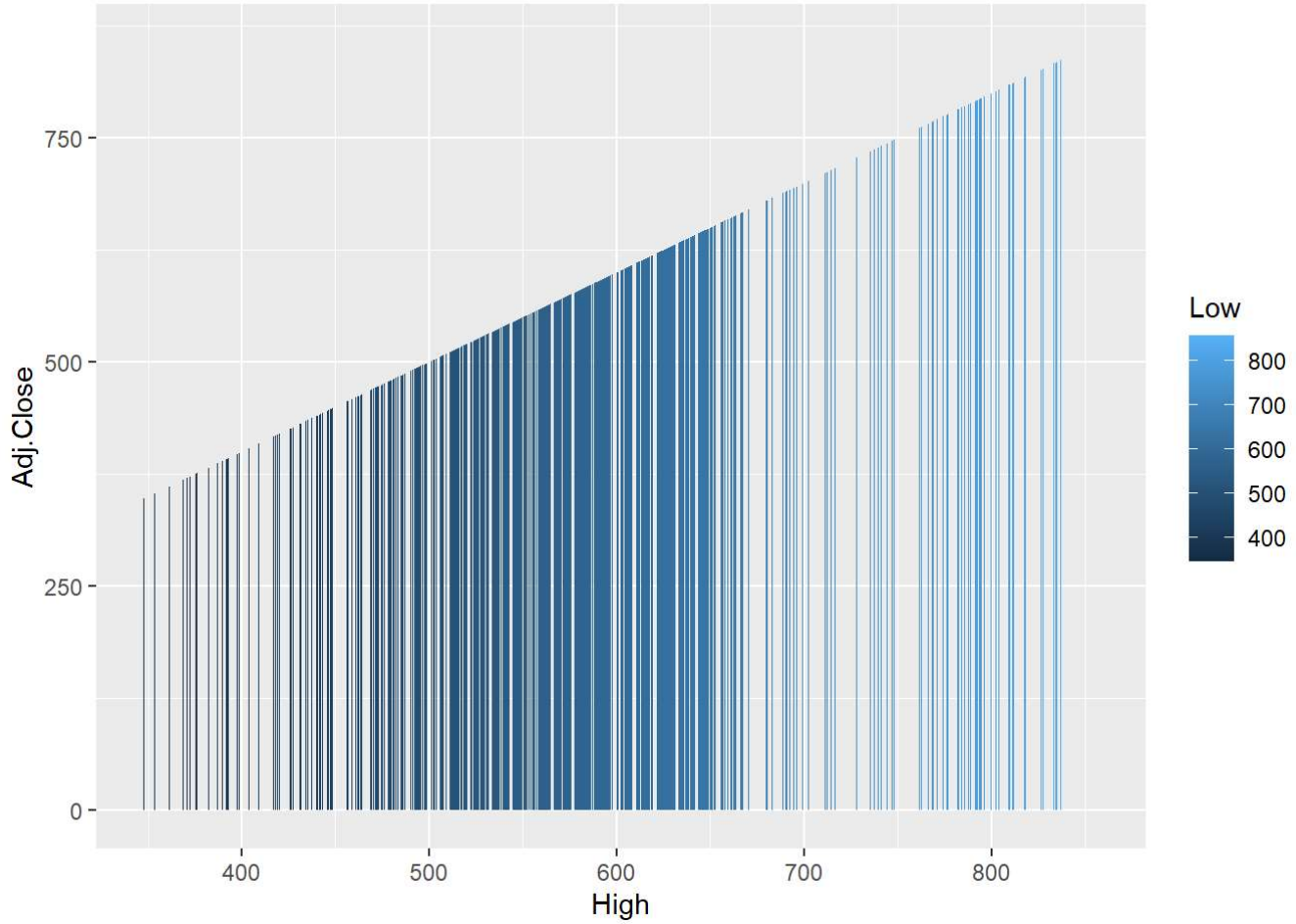
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

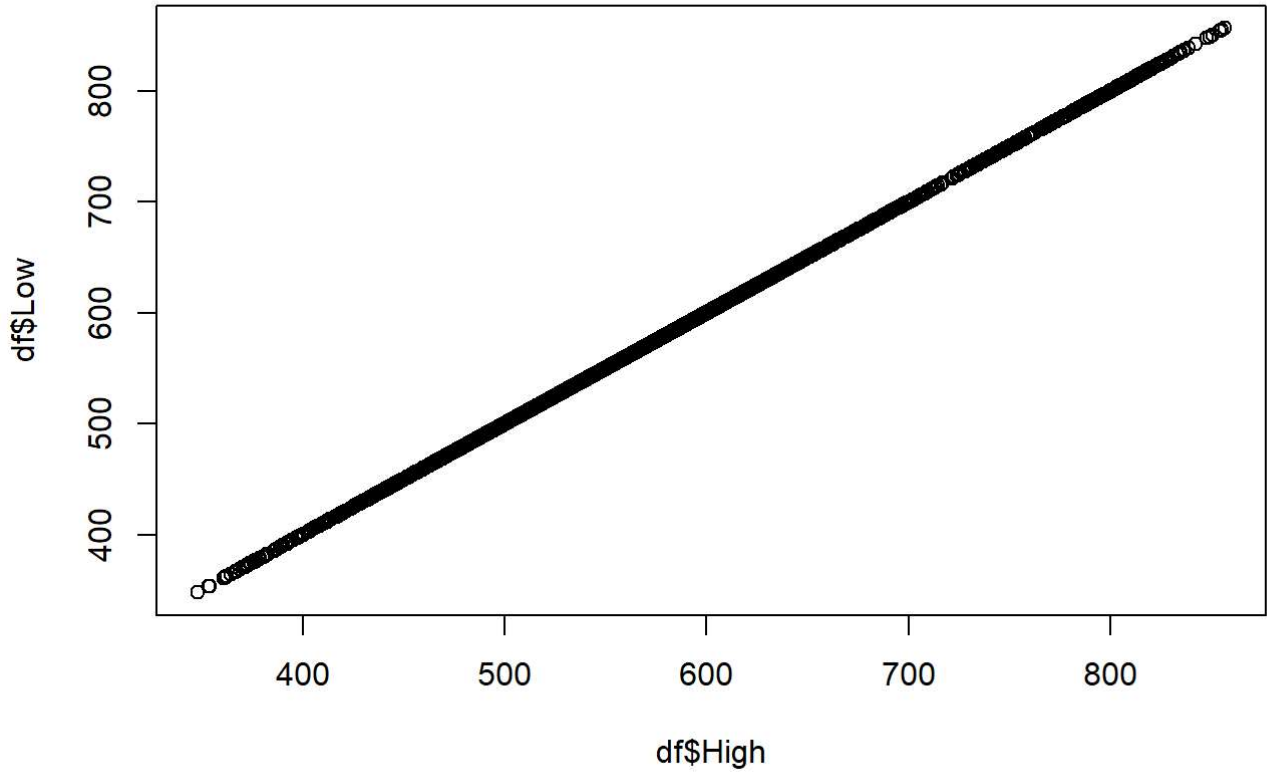
```
library(ggplot2)
```

```
ggplot(df,aes(x=High, fill=Low, y=Adj.Close))+geom_col(position ="dodge")
```



Creating a Scatter Plot

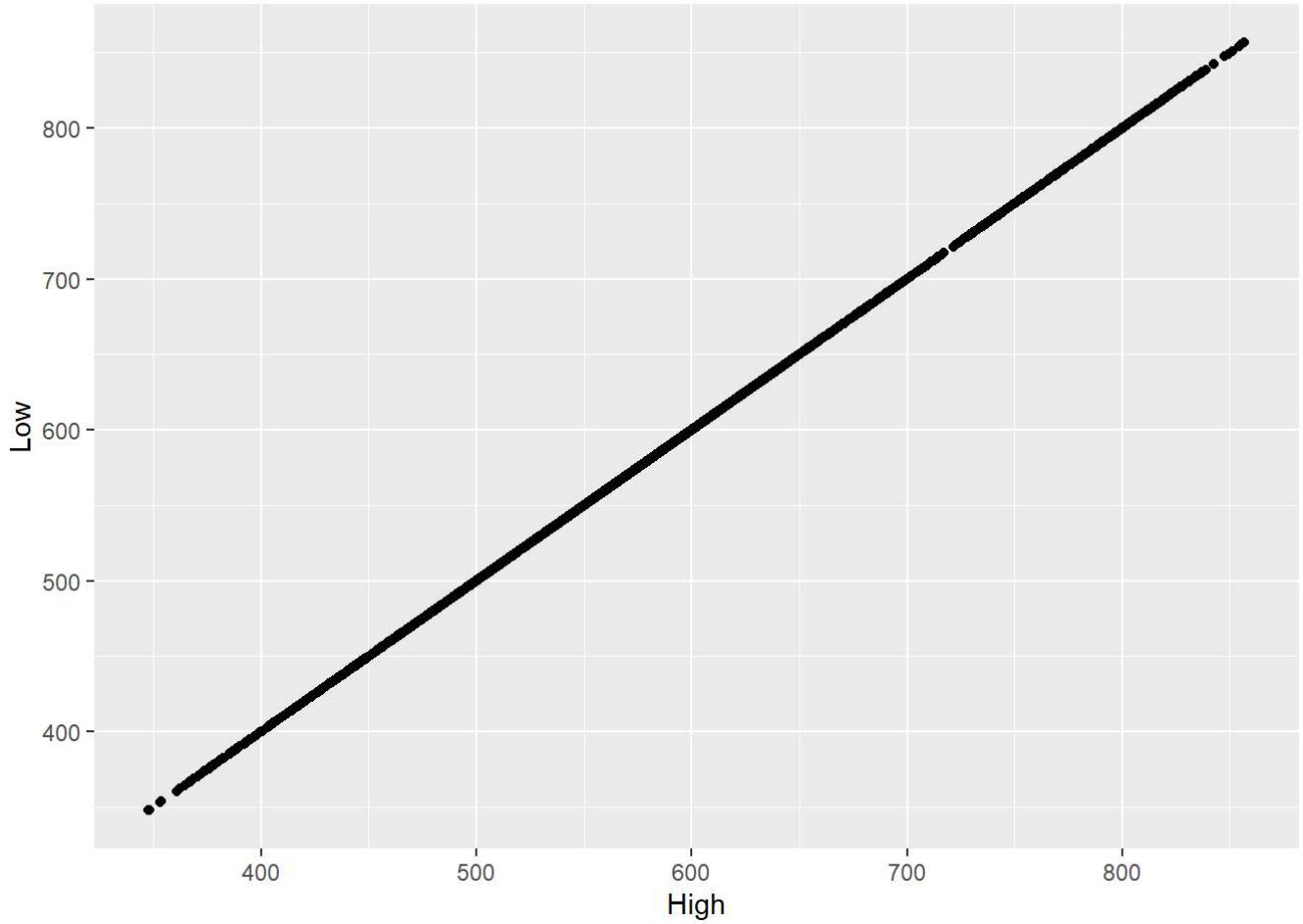
```
plot(df$High, df$Low)
```



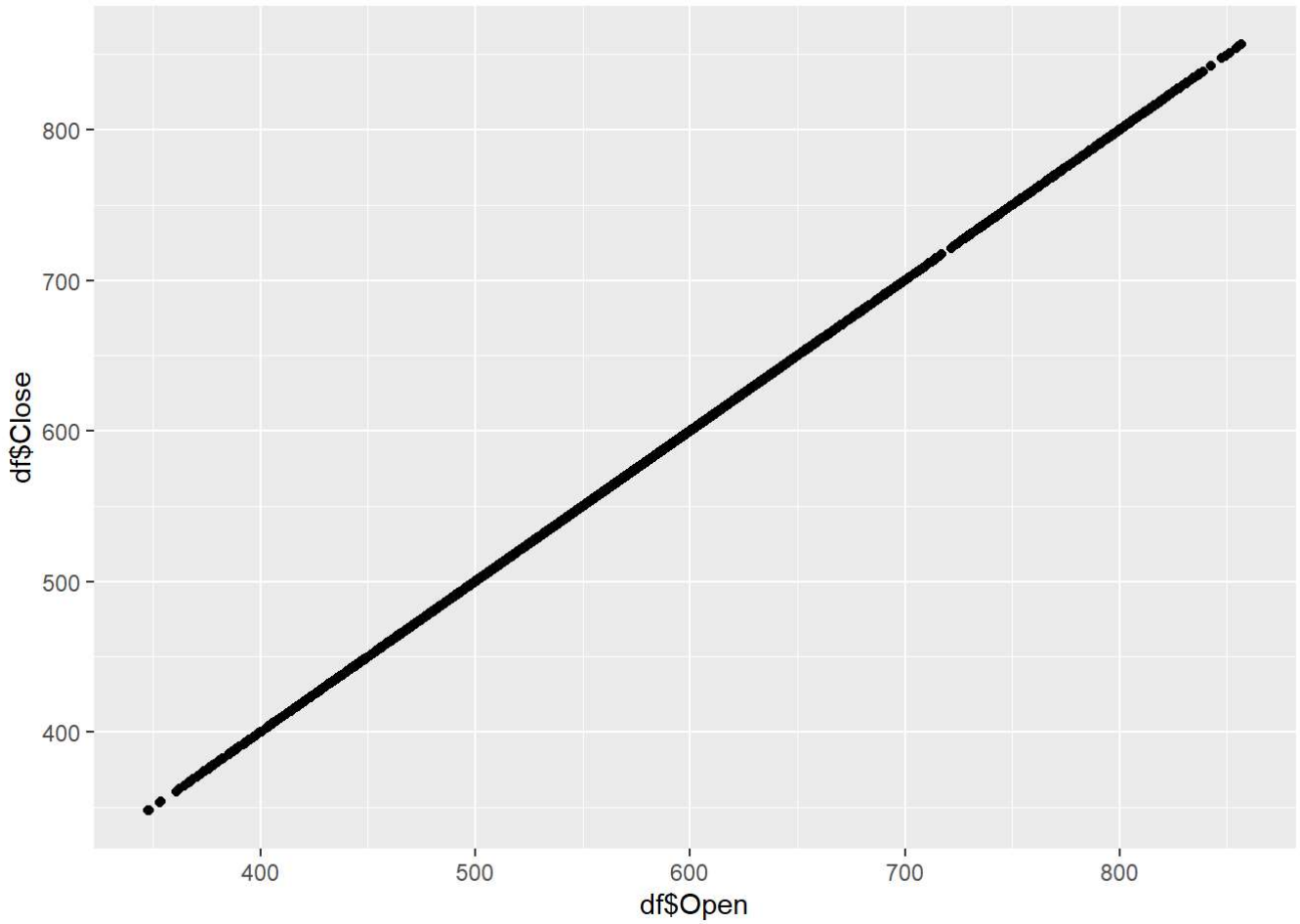

```
# Scatter plot with base graphics
```

Scatter plot with ggplot 2

```
library(ggplot2)
ggplot(df, aes(x = High, y = Low)) +geom_point()
```

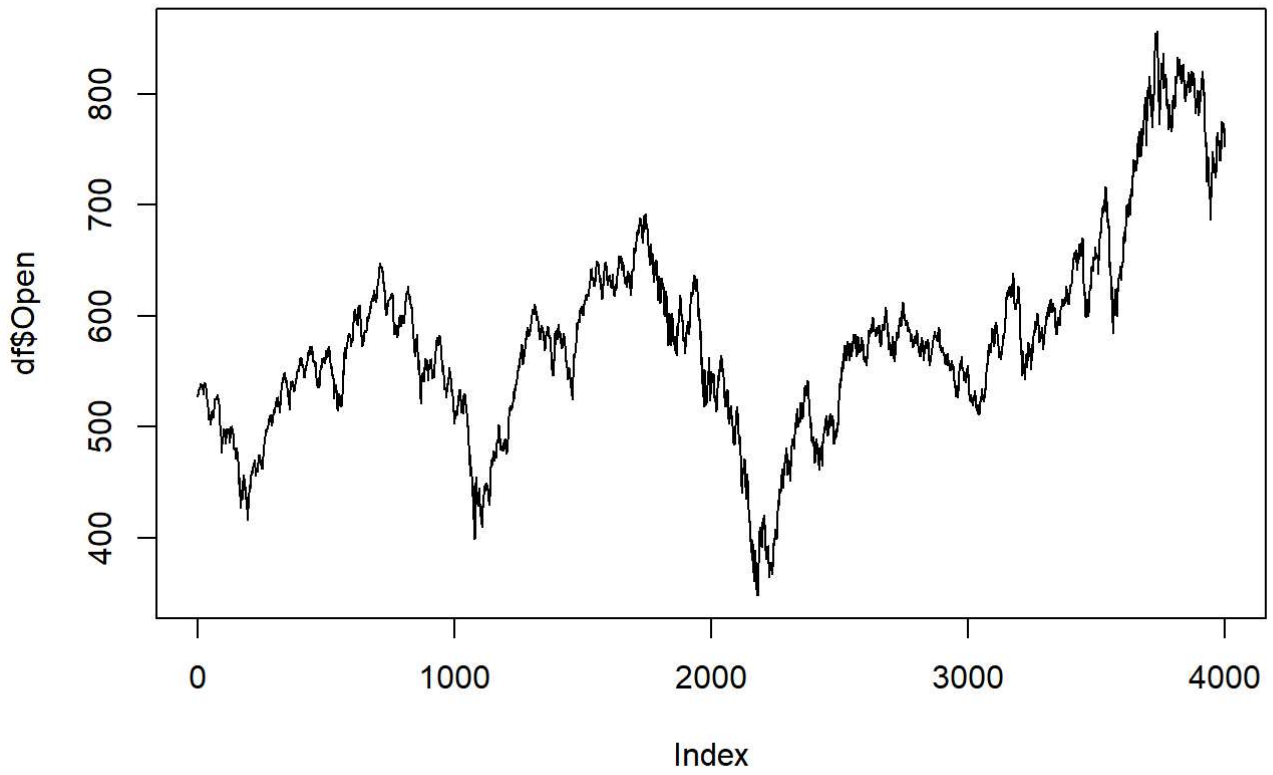



```
ggplot(data = NULL, aes(x =df$Open, y=df$Close)) +
geom_point()
```

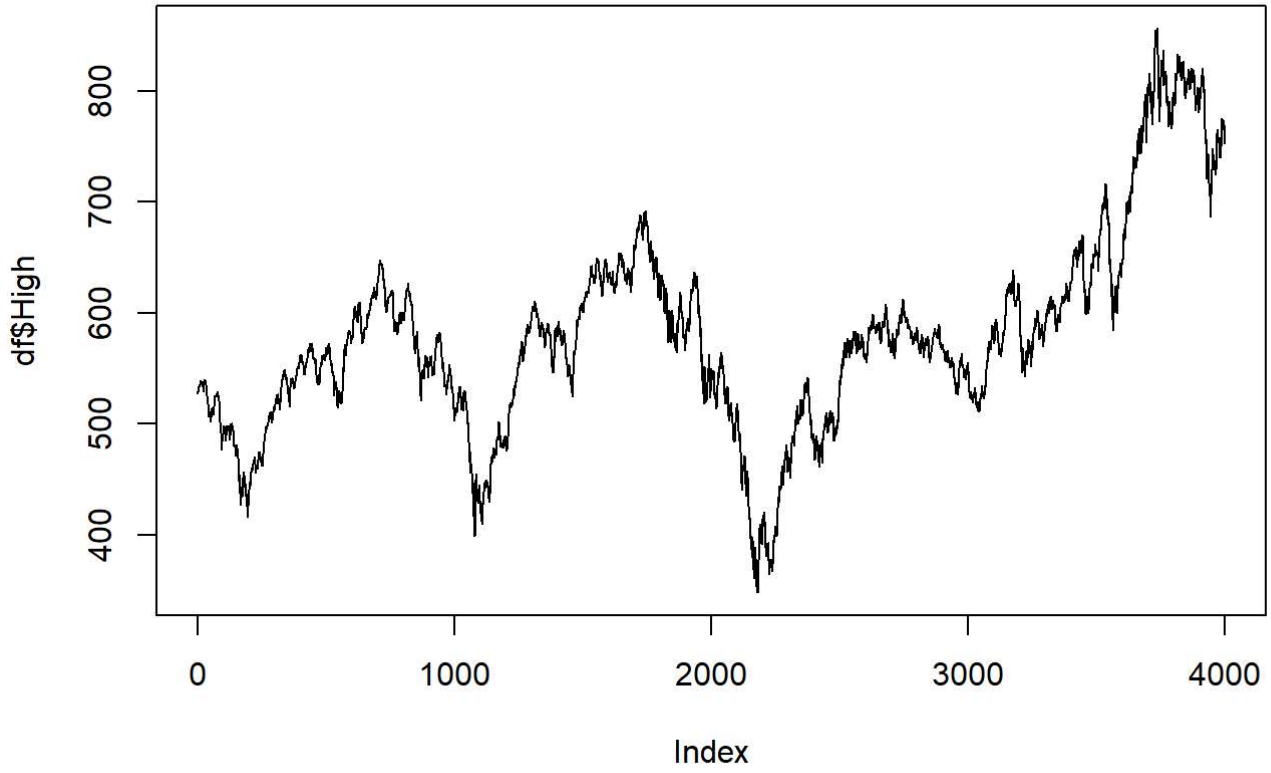


Creating a Line Graph

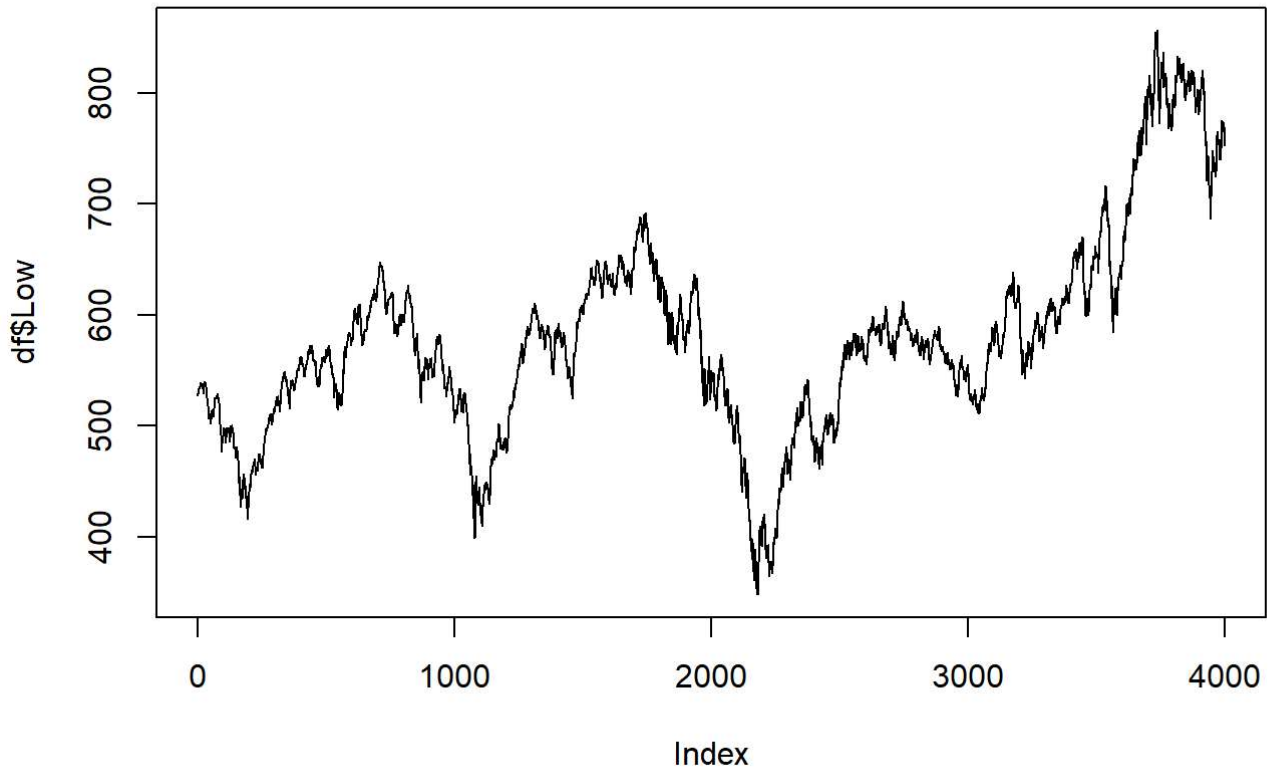
```
plot(df$Open,type = "l")
```



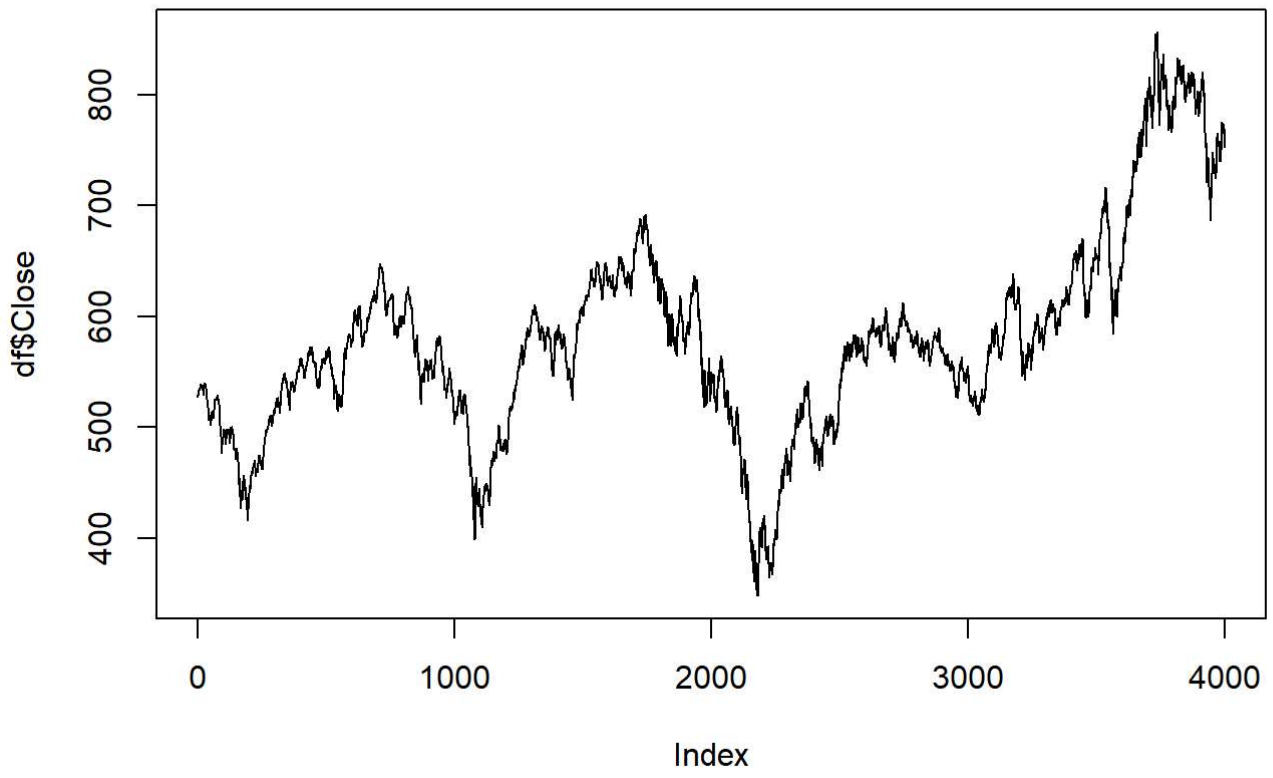

```
plot(df$High,type = "l")
```




```
plot(df$Low,type = "l")
```

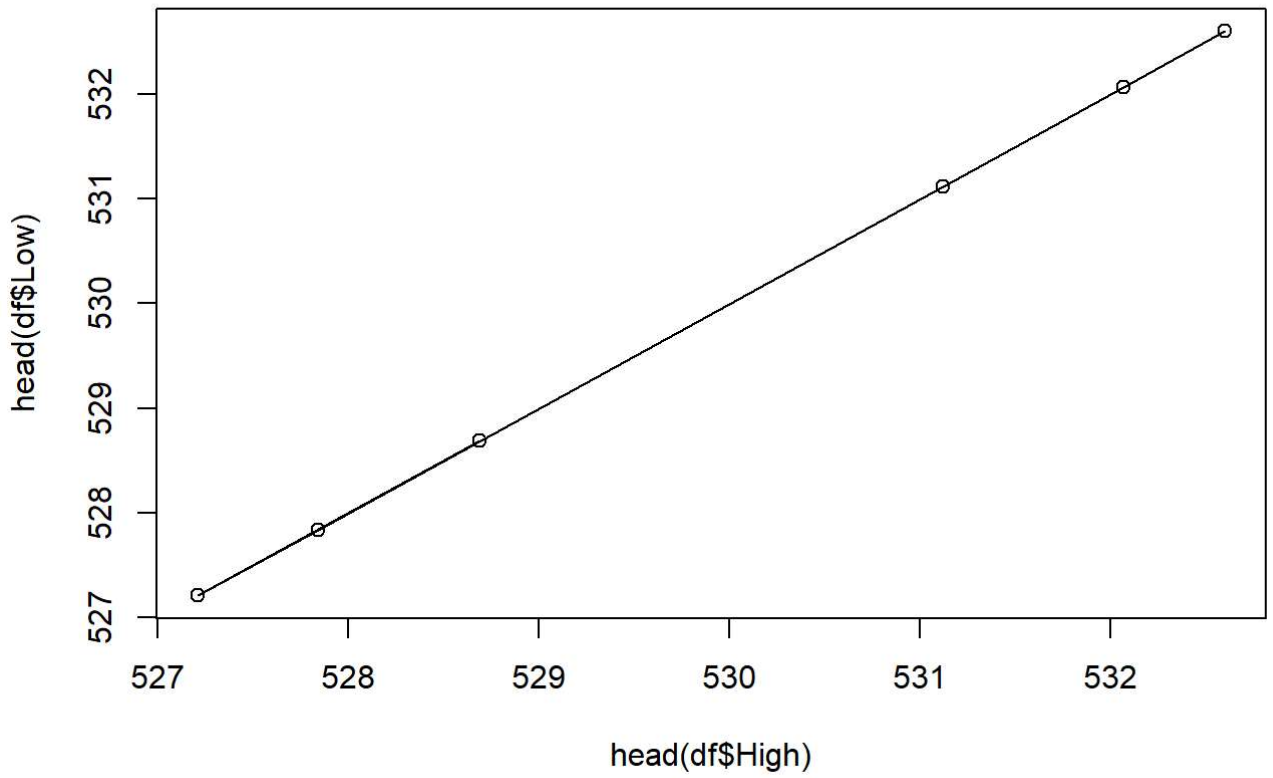


```
plot(df$Close,type = "l")
```



Line graph with base graphics

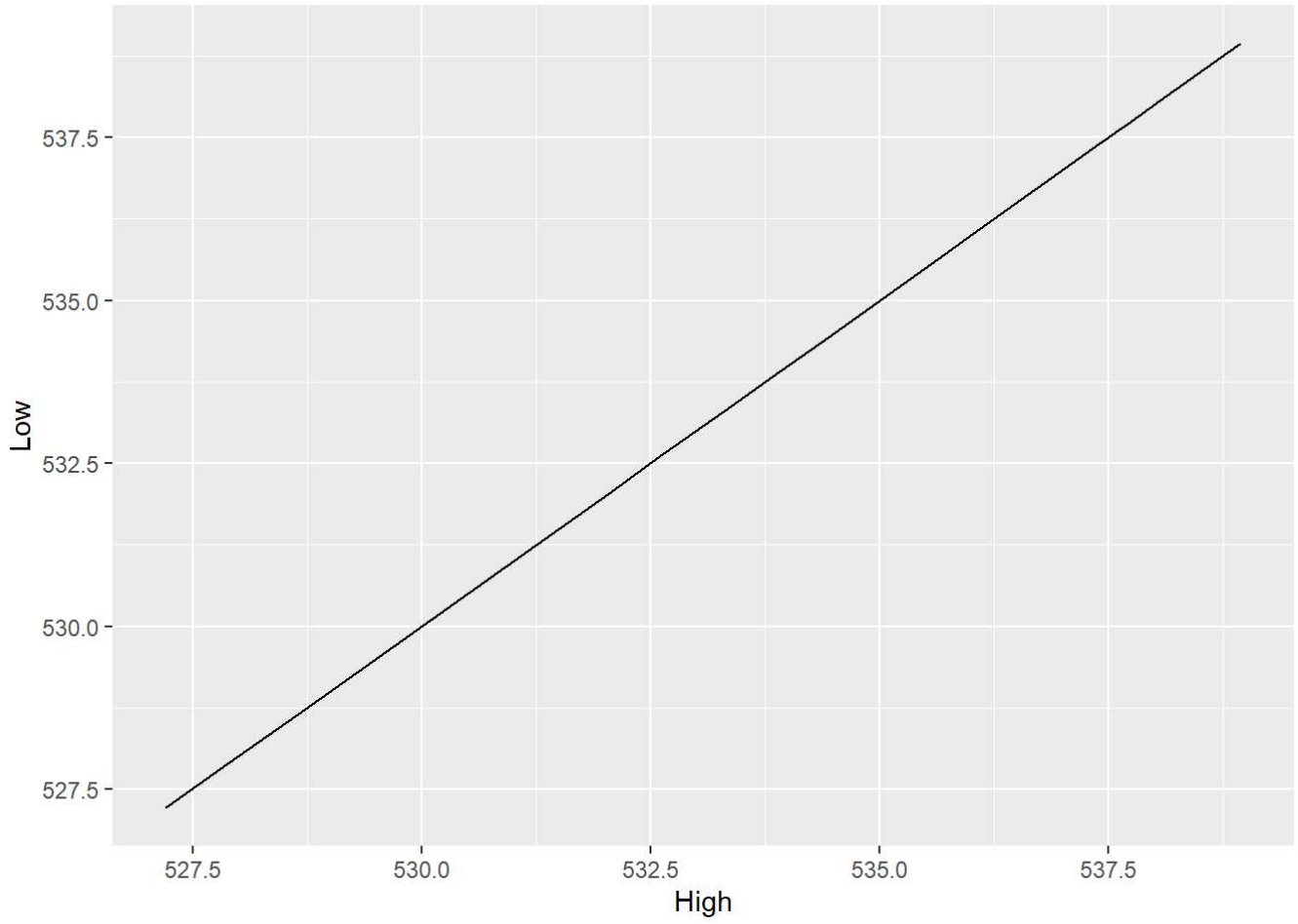
```
plot(head(df$High),head(df$Low),  type = "l")
points(head(df$Open),head(df$Close))
```



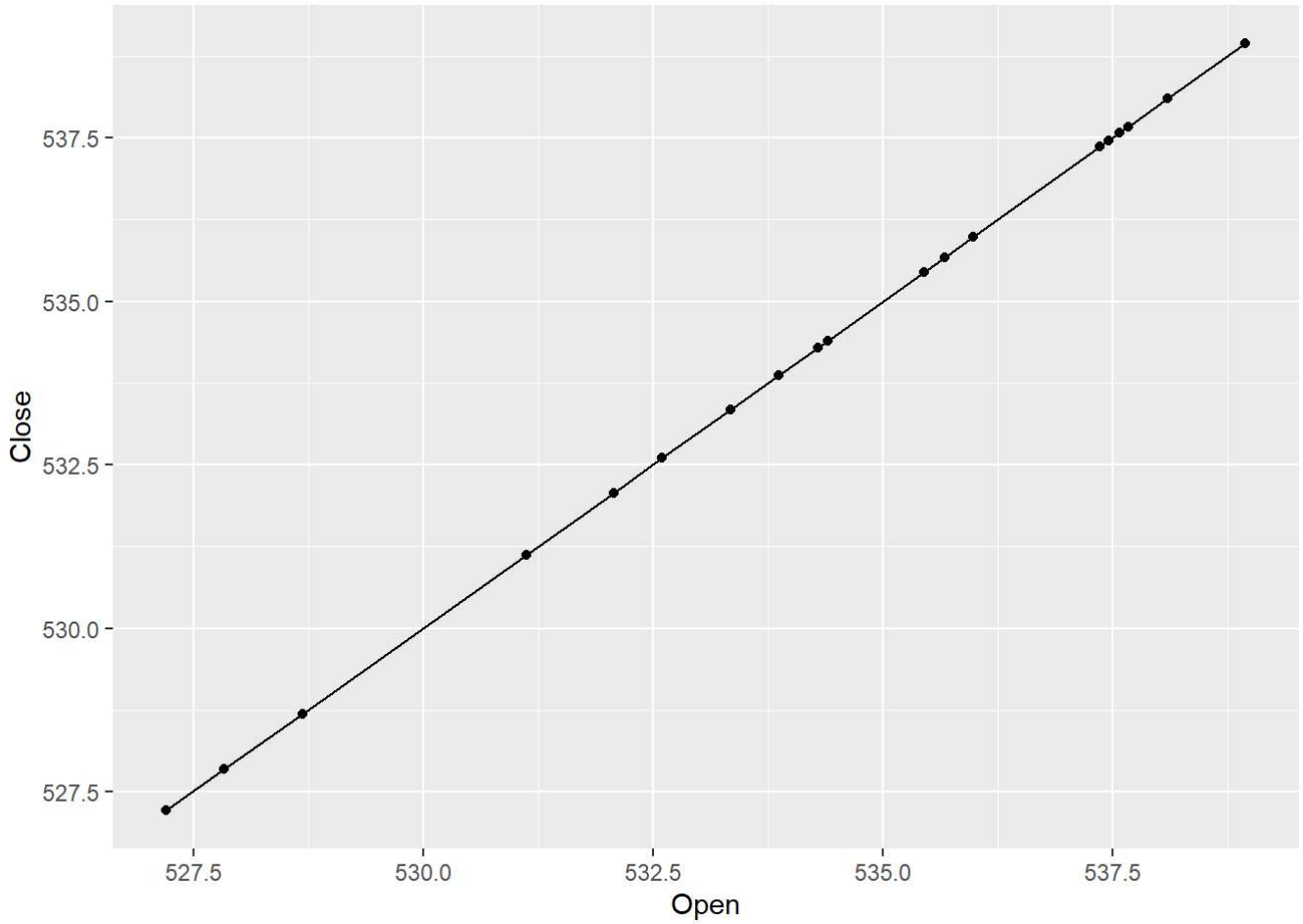
#

Line graph with ggplot() and With points added to ggplot()

```
ggplot(head(df,n=20), aes(x = High, y = Low)) +geom_line()
```



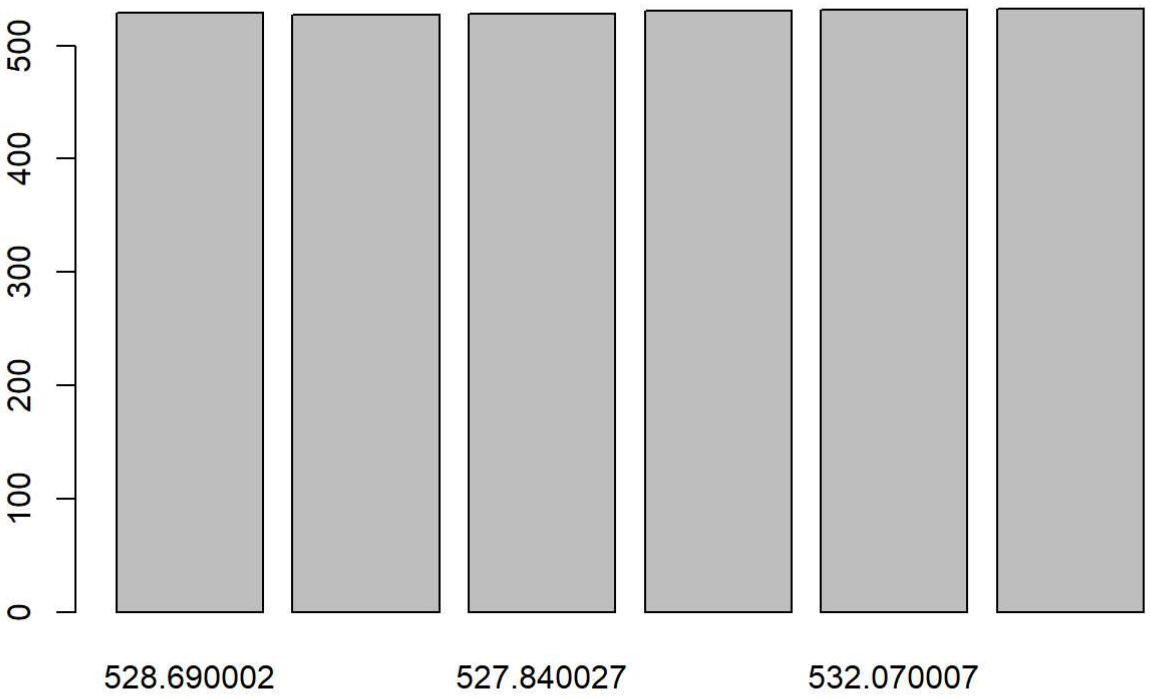
```
ggplot(head(df,n=20), aes(x = Open, y = Close)) +geom_line()+geom_point()
```



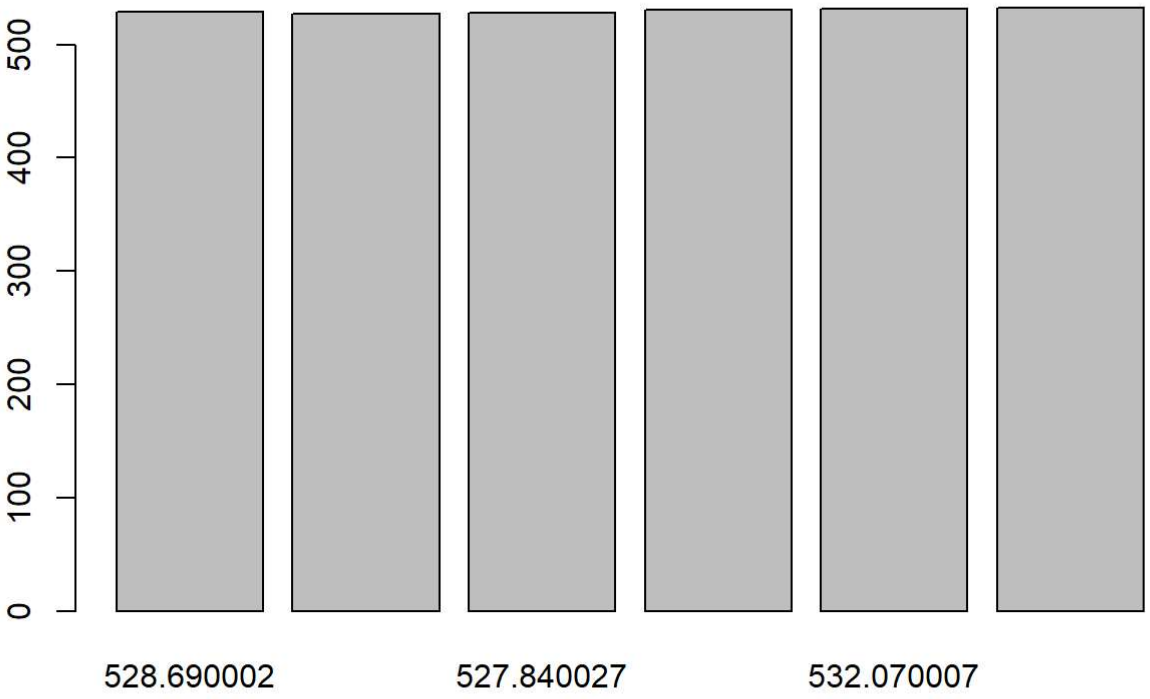
#

Creating a Bar Graph

```
barplot(head(df$High), names.arg = head(df$Low))
```



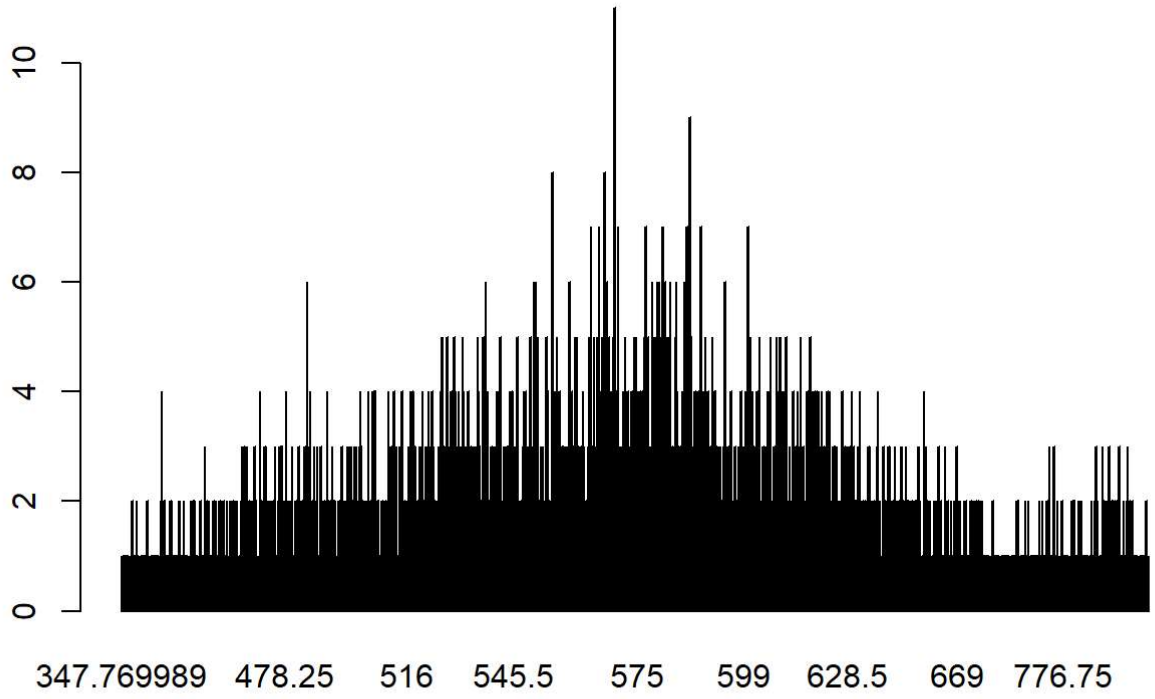

```
barplot(head(df$Open), names.arg = head(df$Close))
```



#

Generate a table of counts

```
barplot(table(df$Open))
barplot(table(df$High))
```




```
barplot(table(df$Low))
barplot(table(df$Close))
```

Laoding ggplot2() package

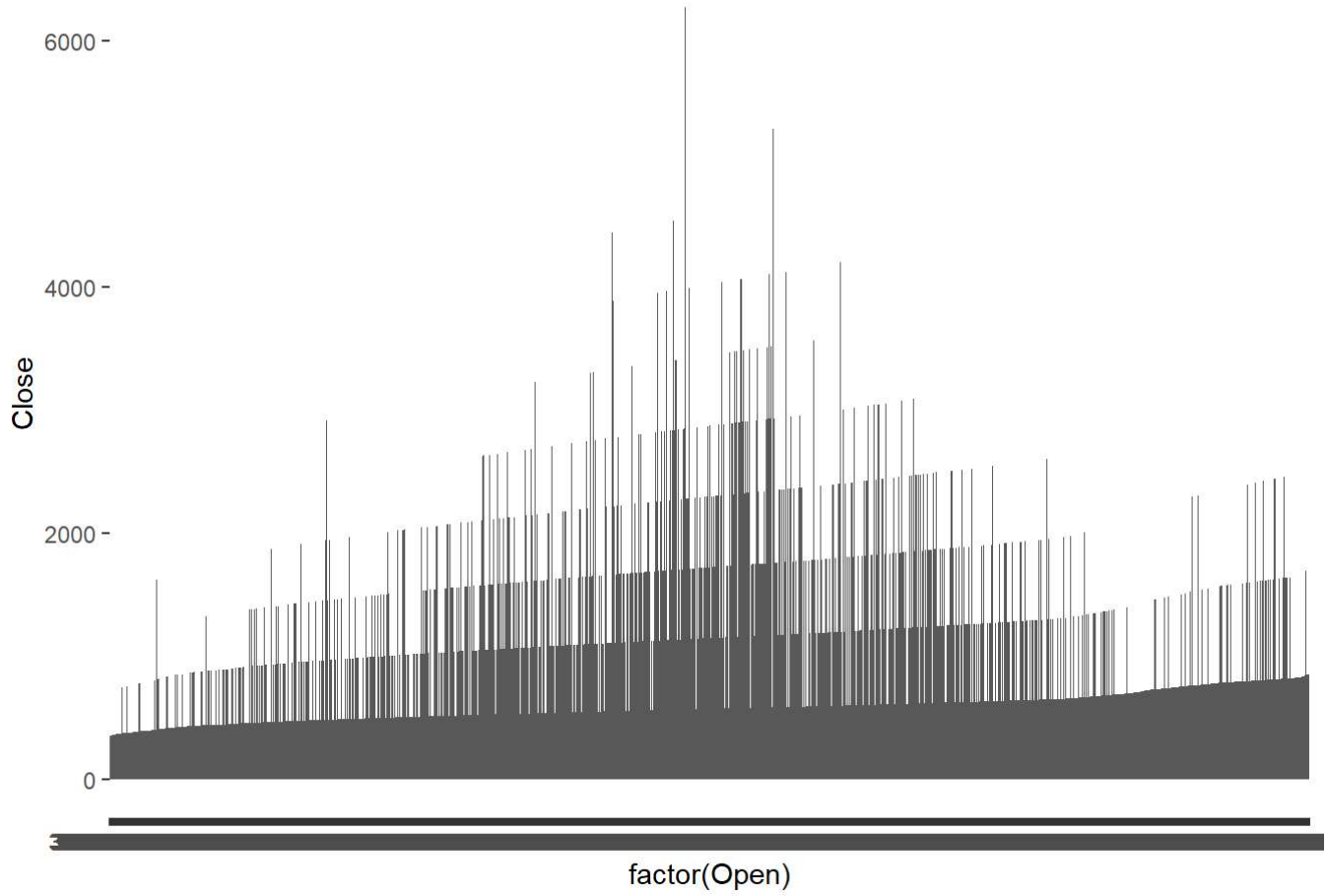
```
library(ggplot2)
```

Bar graph of values. This uses the dataset data frame, with the

“Open” column for x values and the “Close” column for y values.

` # Convert the x variable to a factor, so that it is treated as discrete

```
ggplot(df, aes(x = factor(Open), y = Close)) +geom_col()
```

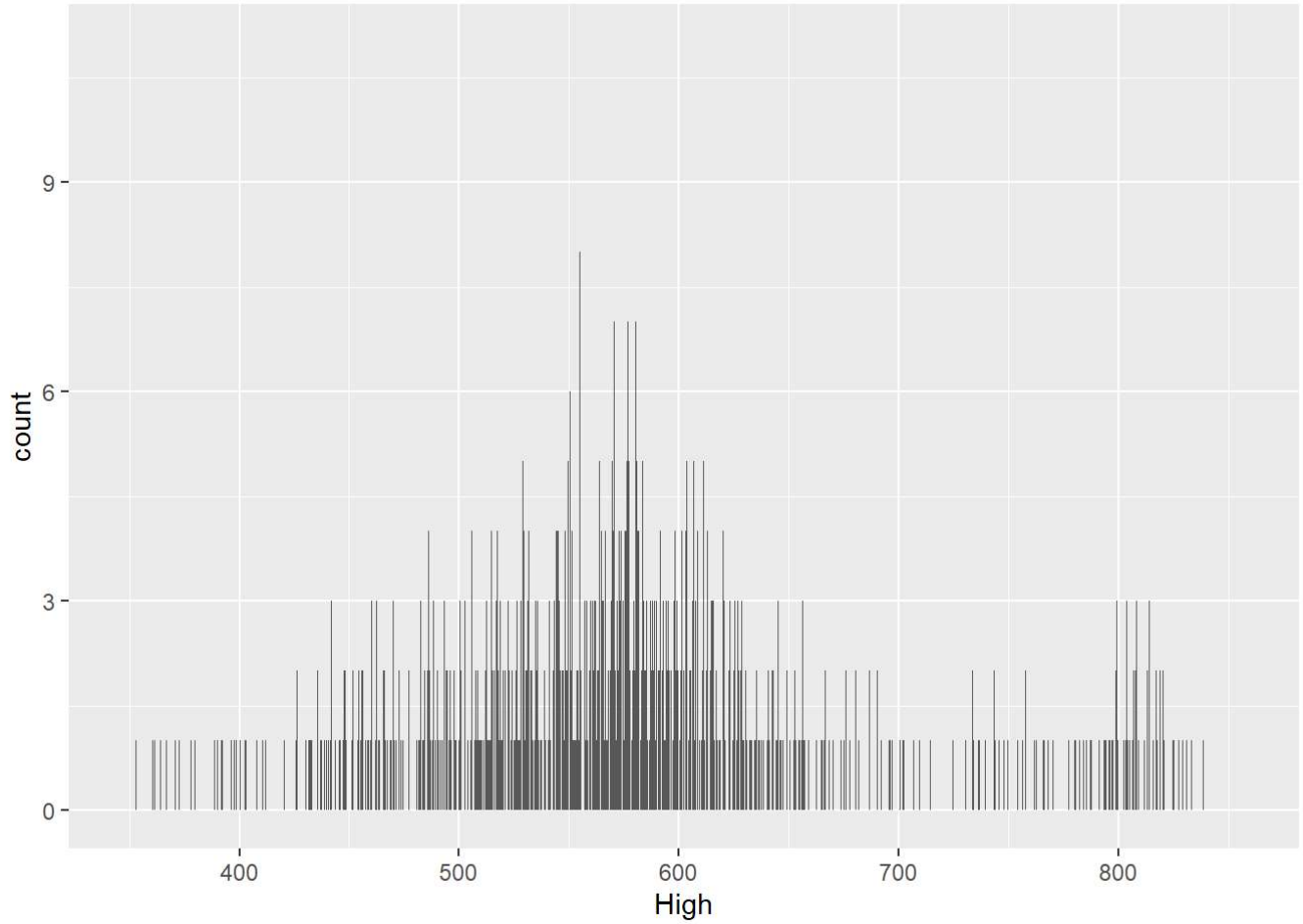


Bar graph of counts. This uses the dataset data frame, with the “High” column for

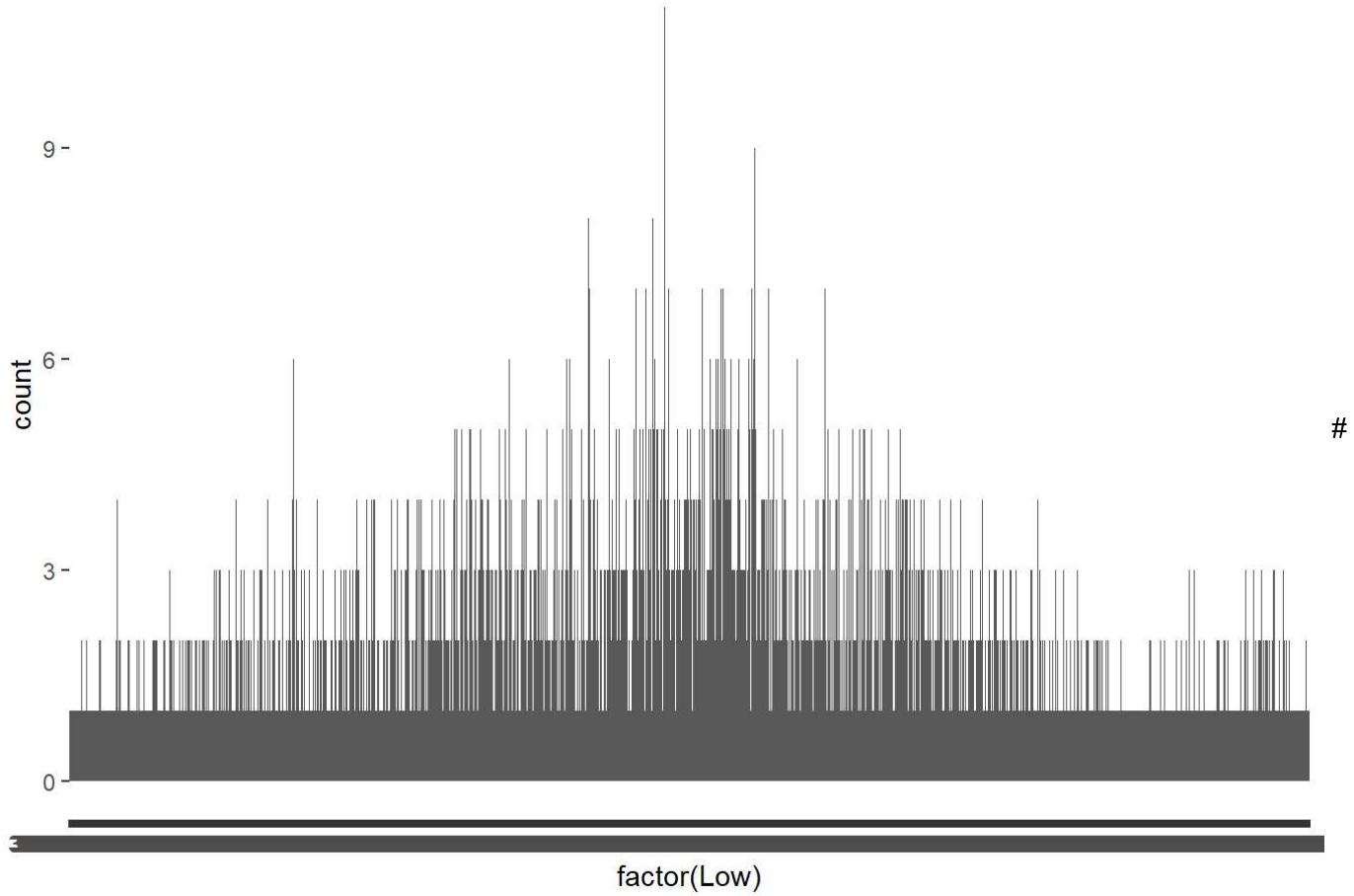
x position. The y position is calculated by counting the number of rows for

each value of pretest.

```
ggplot(df, aes(x = High)) +
geom_bar()
```



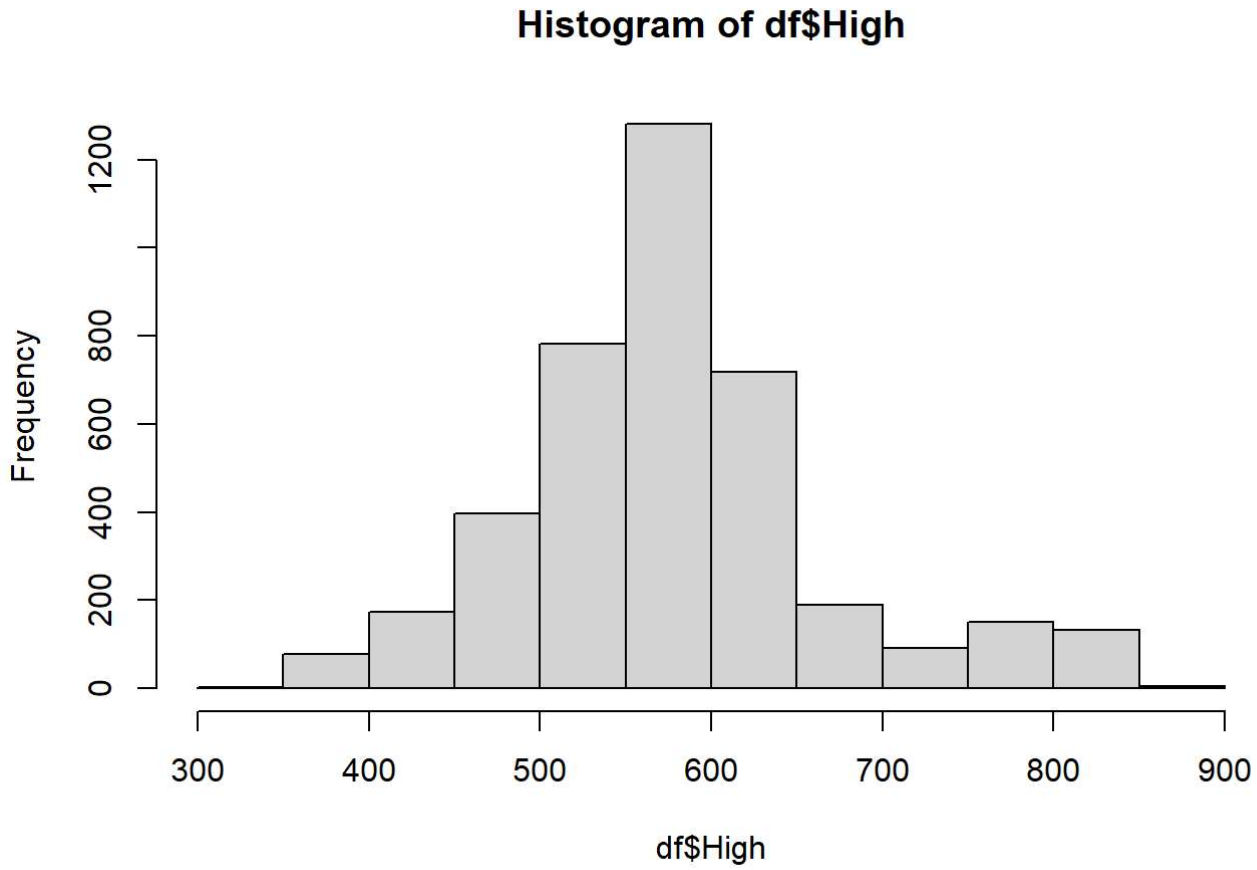

```
# Bar graph of counts
ggplot(df, aes(x = factor(Low))) +
geom_bar()
```



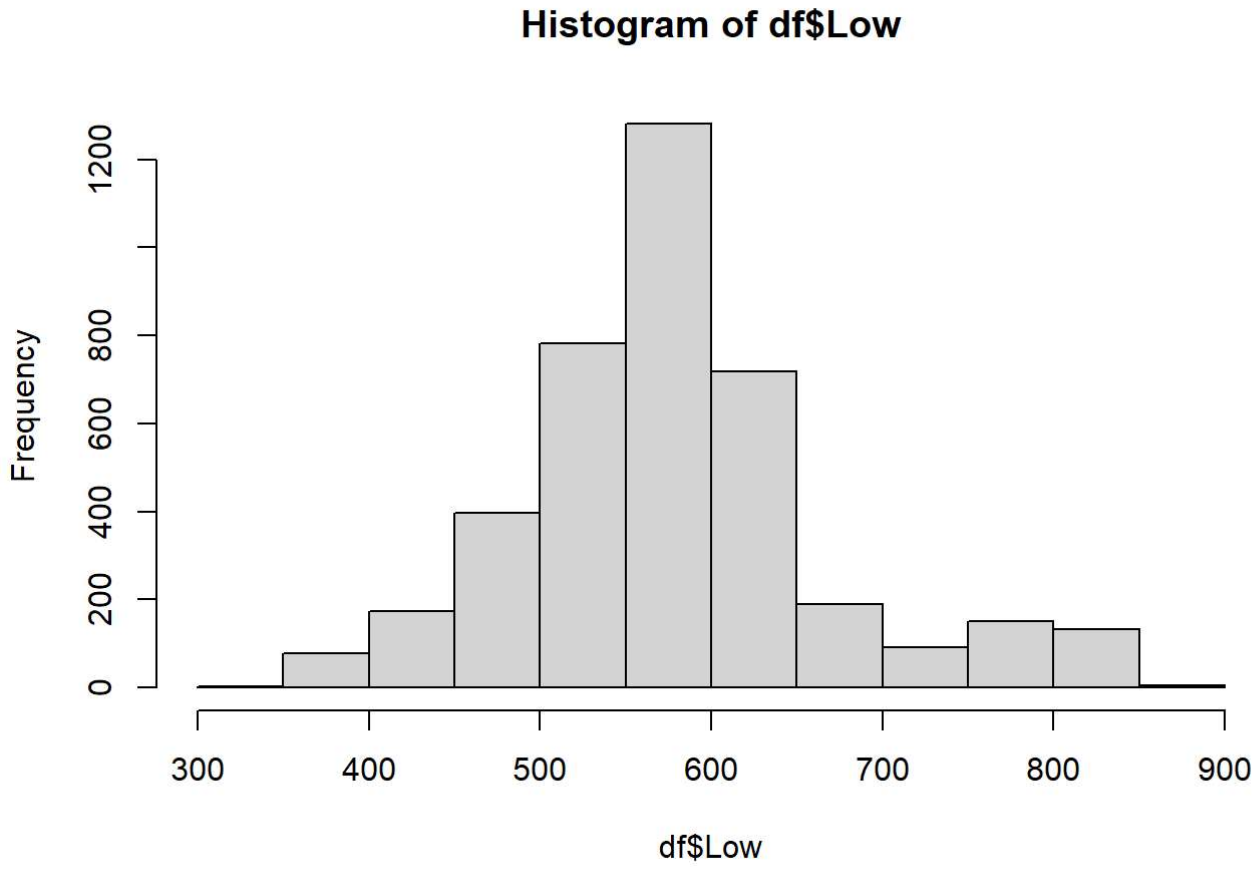
#

Creating a Histogram

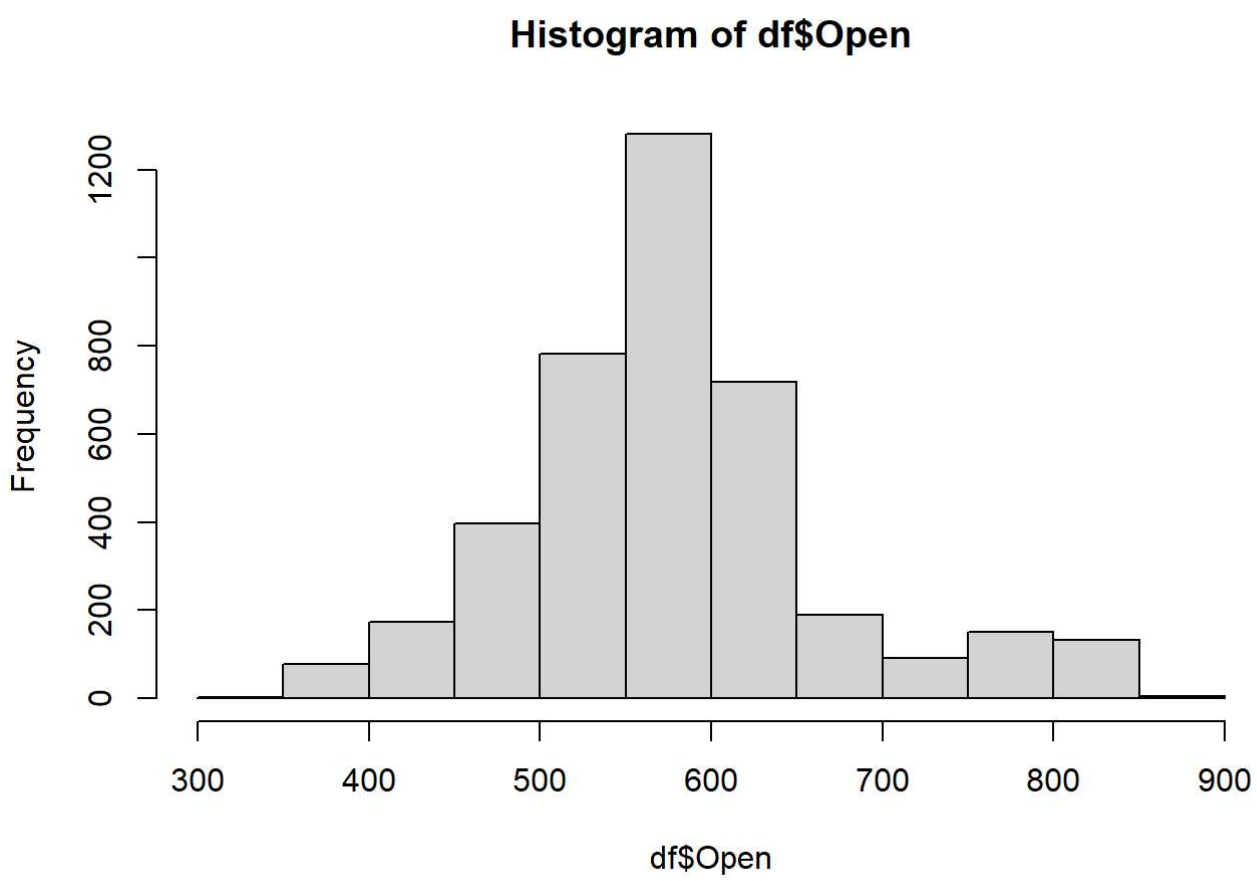
```
hist(df$High)
```



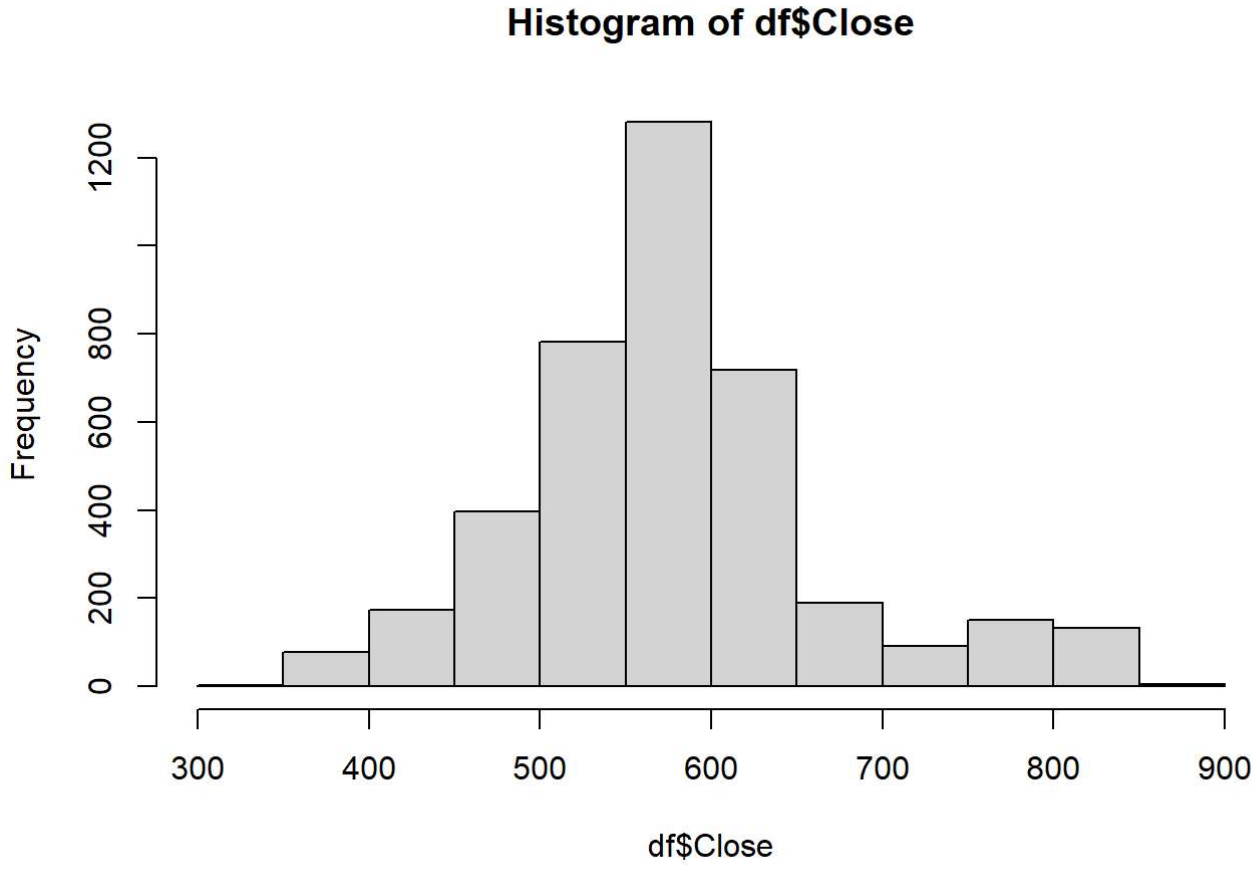
```
hist(df$Low)
```




```
hist(df$Open)
```




```
hist(df$Close)
```

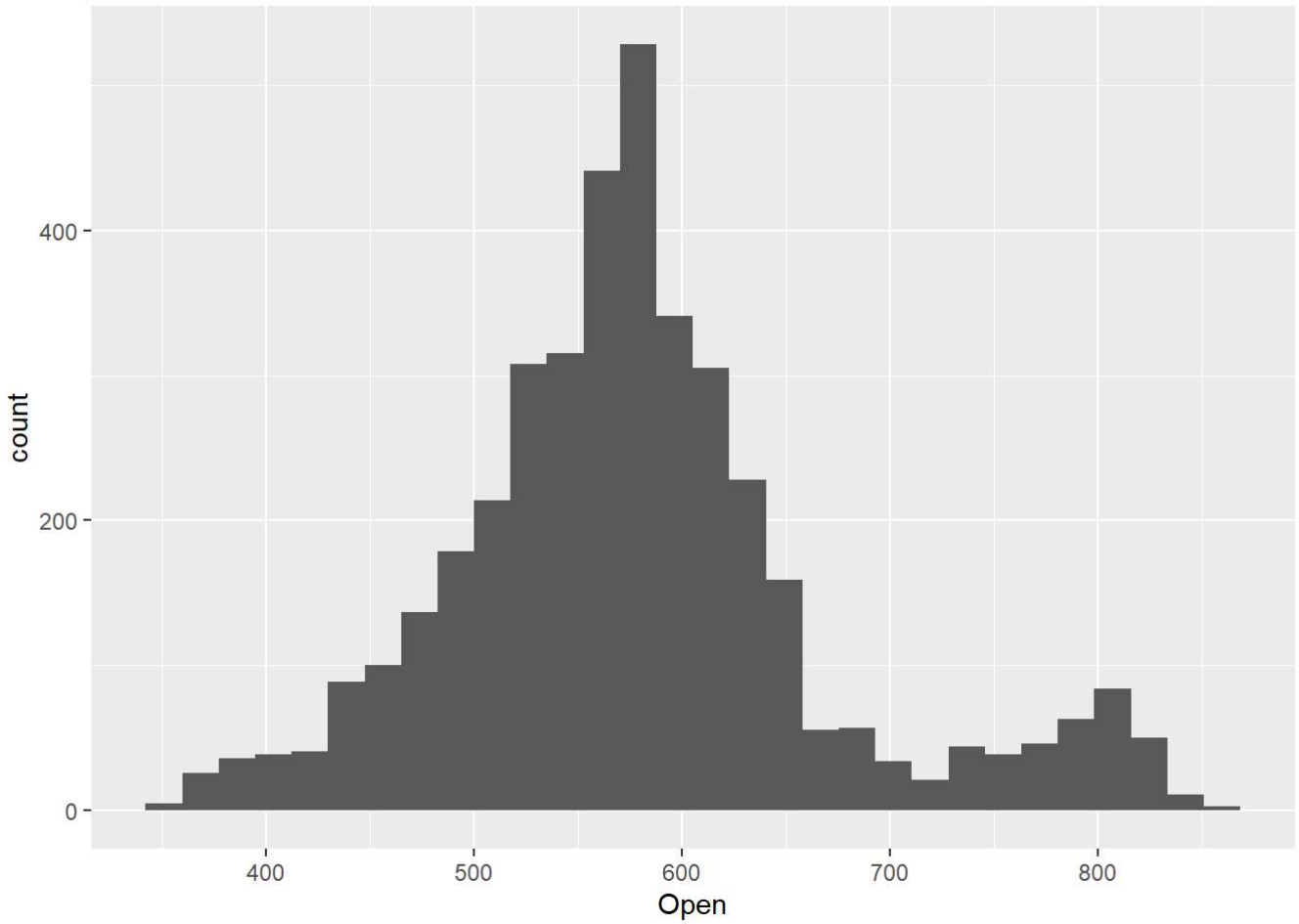


```
# Specify approximate number of bins with breaks
```

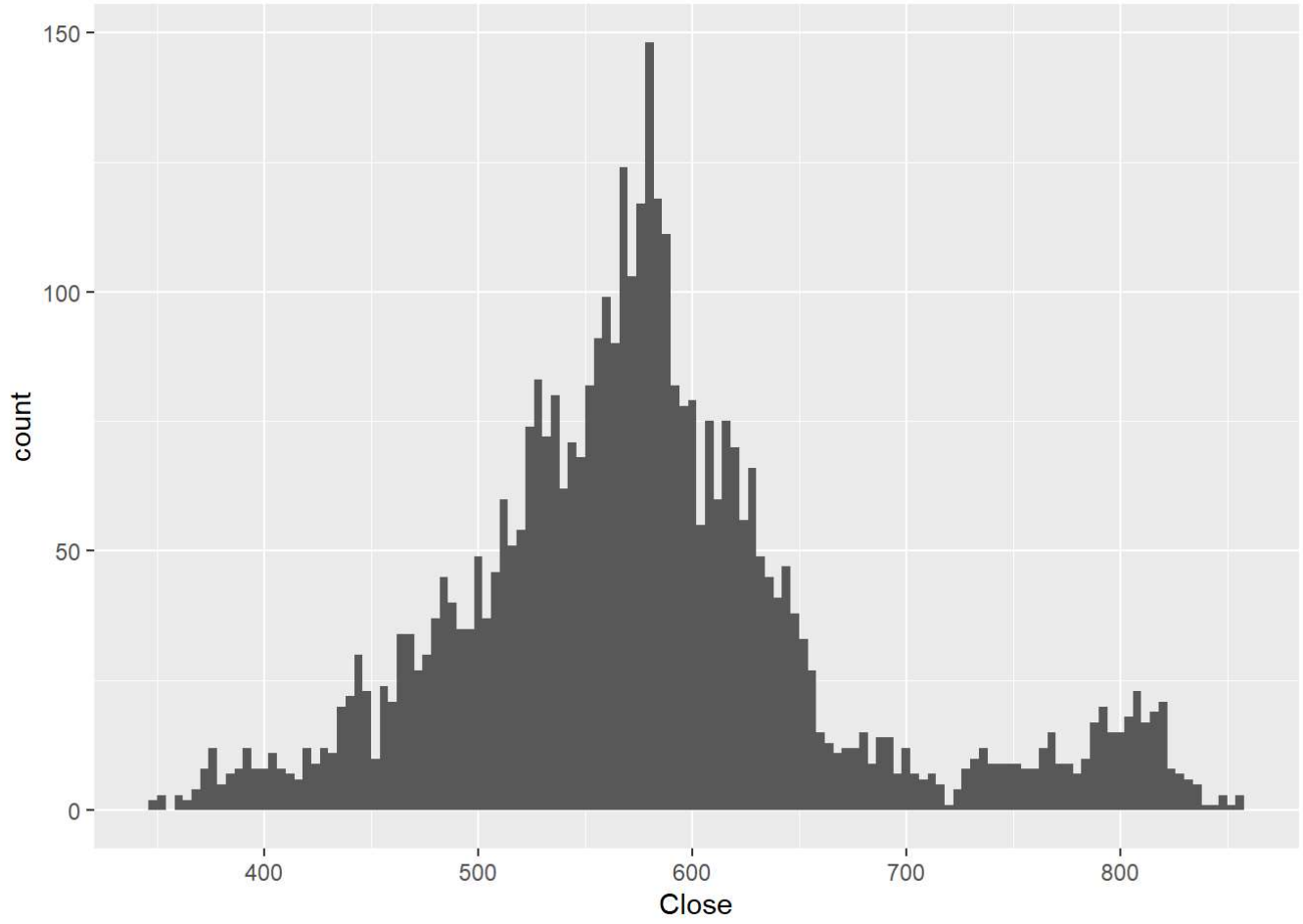
#ggplot2 histogram with default bin width (left); With wider bins (right)

```
ggplot(df, aes(x = Open)) +
geom_histogram()
```

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

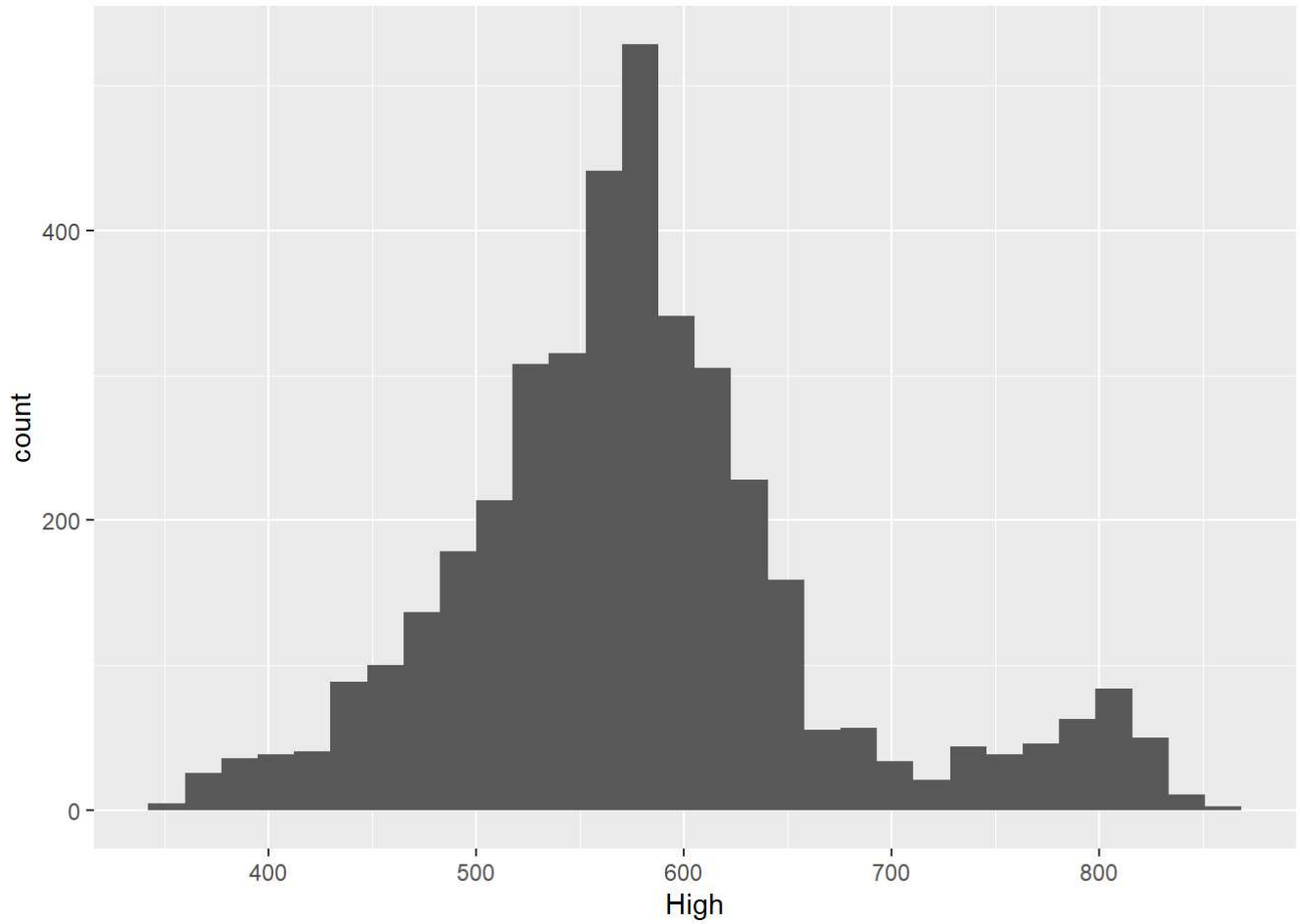



```
ggplot(df, aes(x = Close)) +
geom_histogram(binwidth = 4)
```

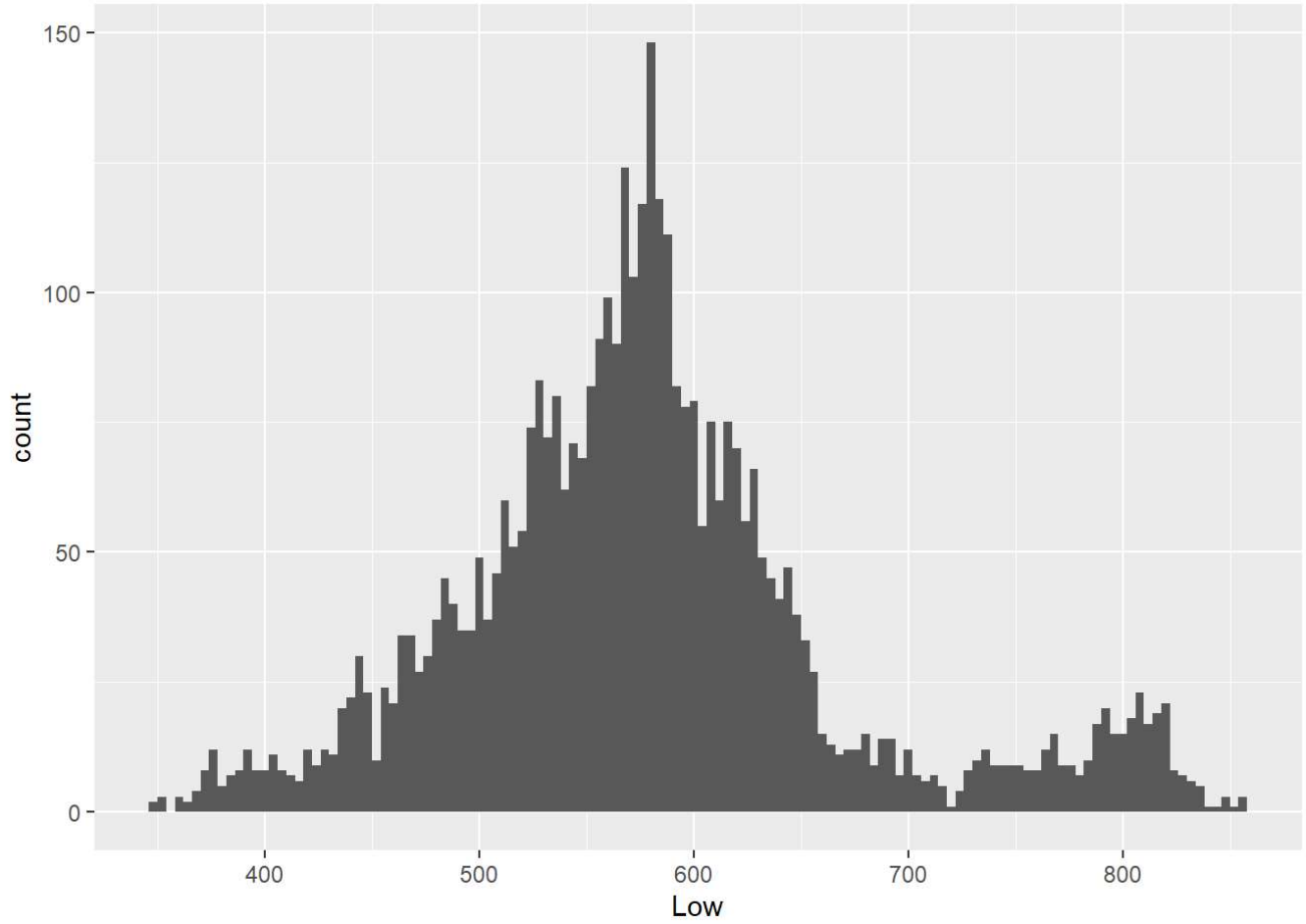



```
ggplot(df, aes(x = High)) +
geom_histogram()
```

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



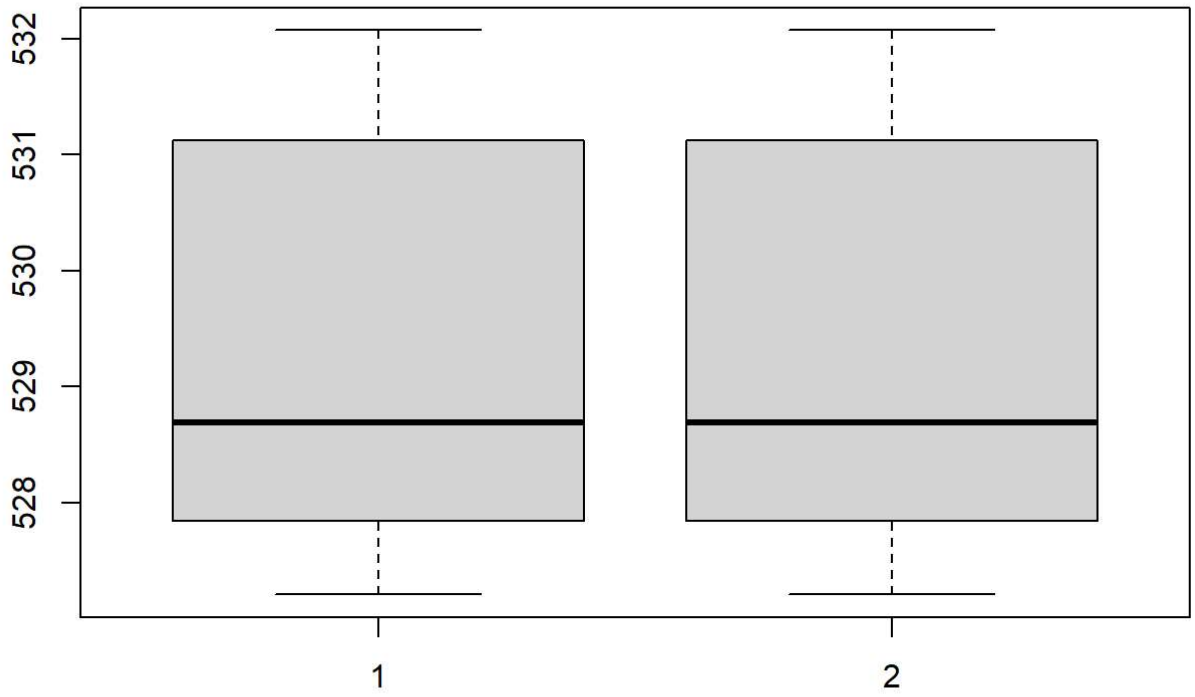

```
ggplot(df, aes(x = Low)) +
geom_histogram(binwidth = 4)
```



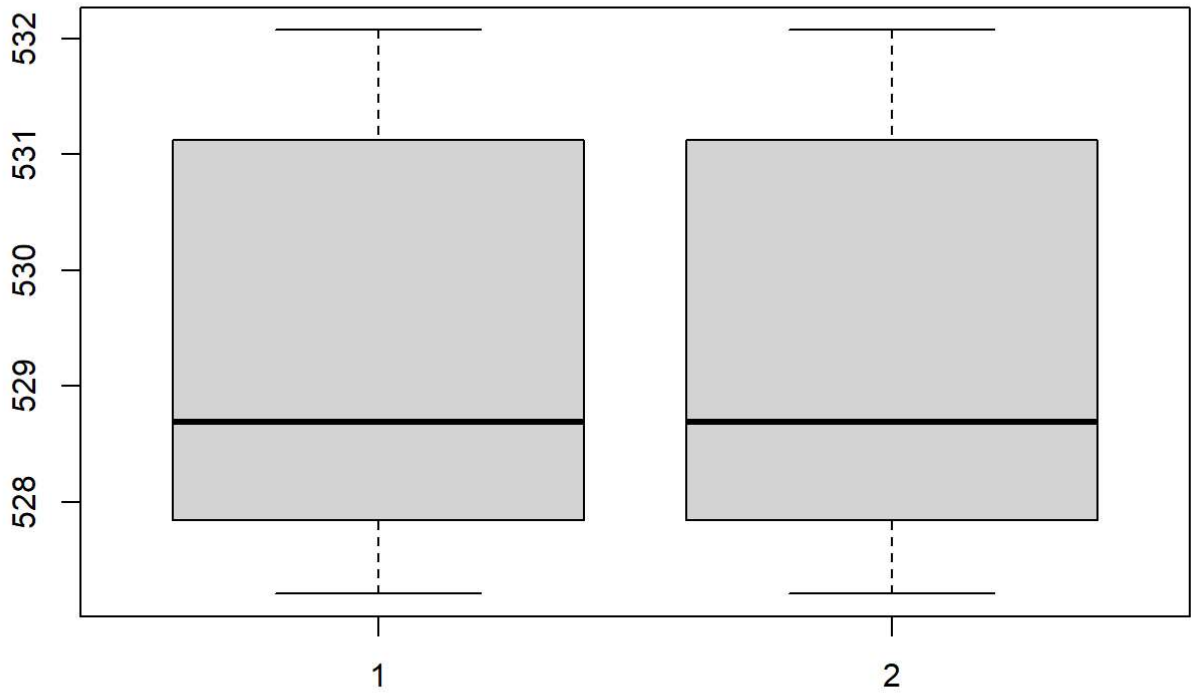
#

Creating a Box Plot

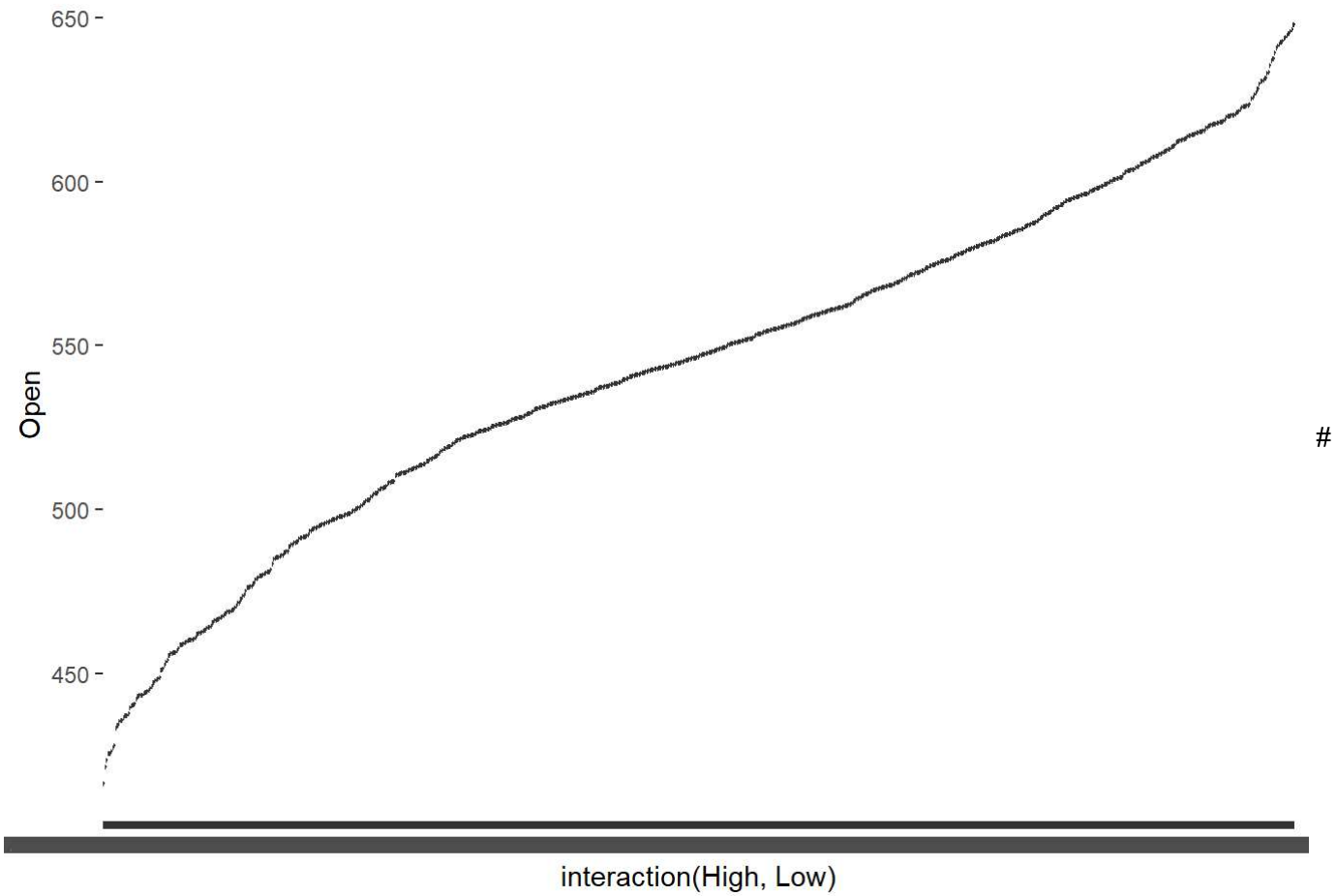
```
boxplot(head(df$High,n=5), head(df$Low,n=5))
```



```
boxplot(head(df$Open,n=5), head(df$Close,n=5))
```



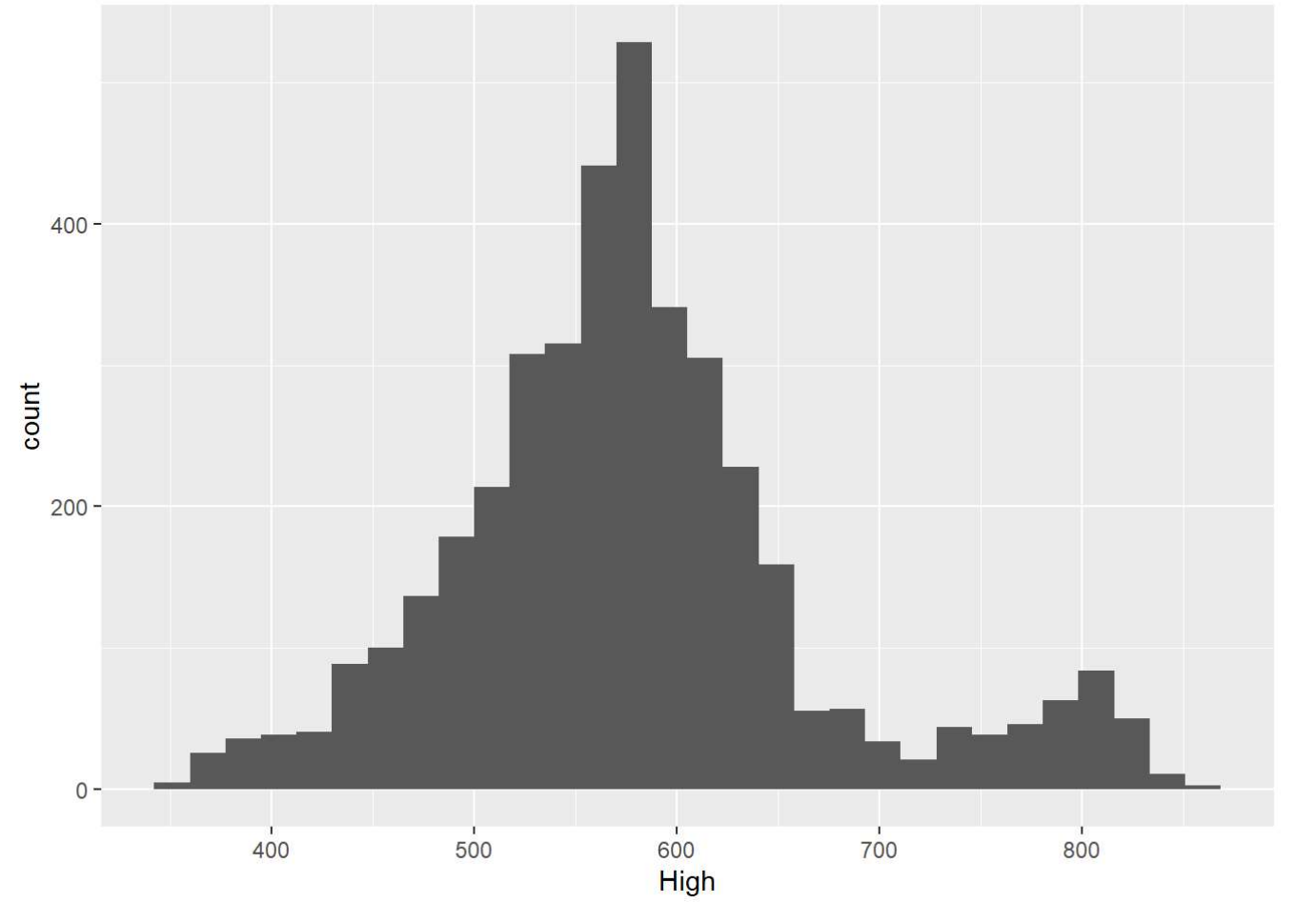

```
ggplot(head(df,n=1000), aes(x = interaction(High, Low), y = Open)) +
geom_boxplot()
```



#

Making a Basic Histogram

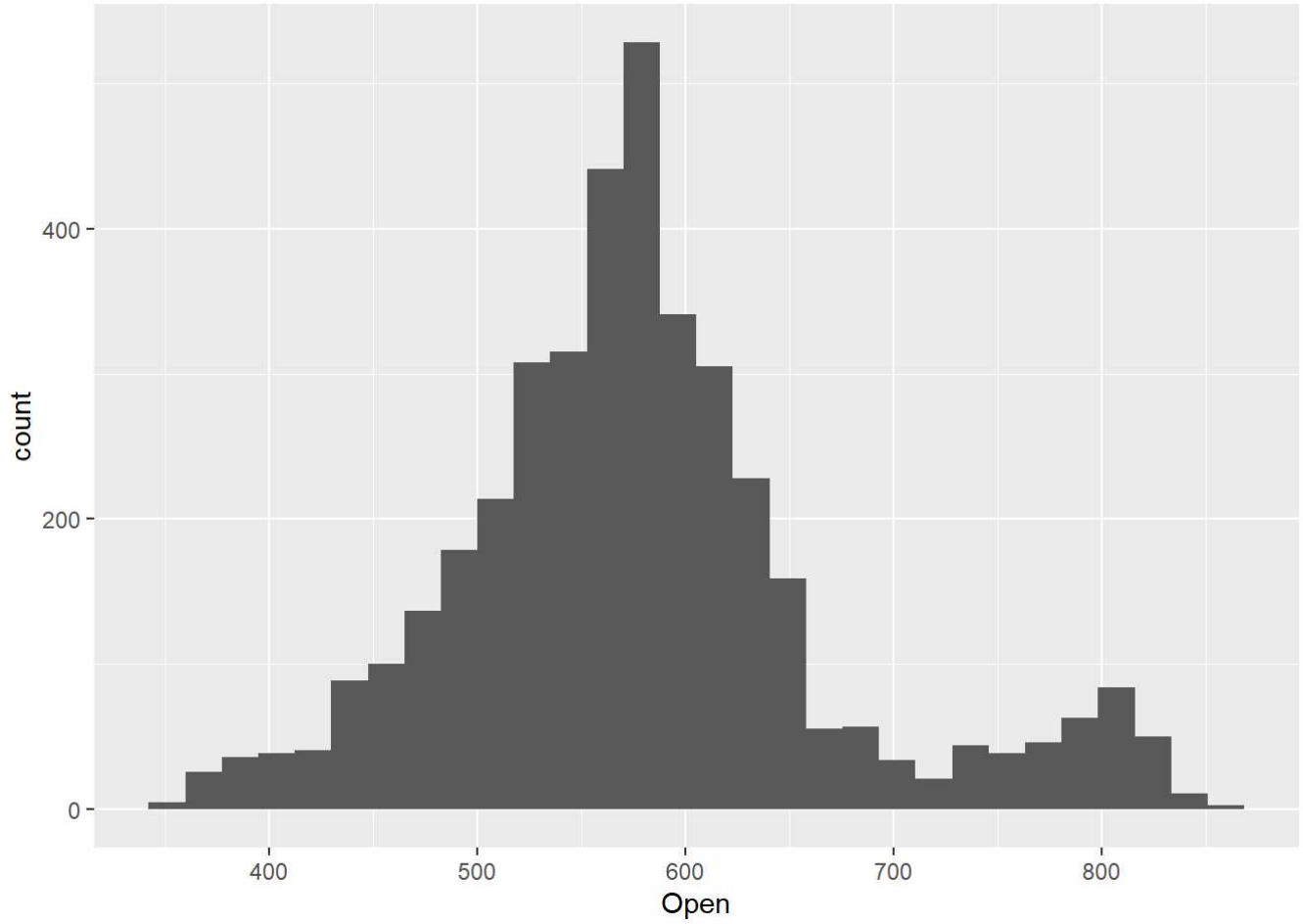
```
ggplot(df, aes(x = High)) +geom_histogram(bins=30)
```



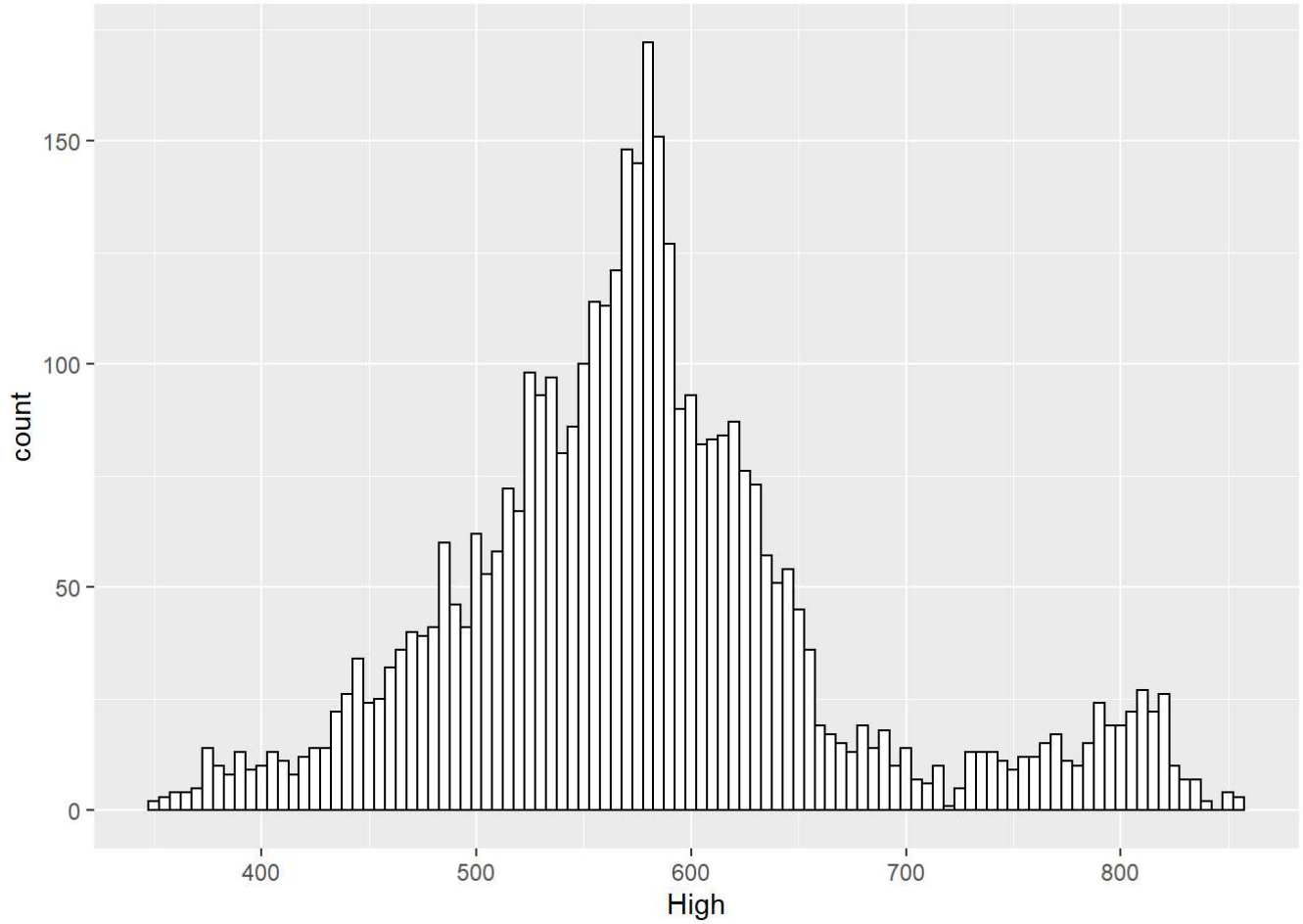
```
ggplot(df, aes(x = Open)) +geom_histogram(bin = 30)
```

```
## Warning: Ignoring unknown parameters: bin
```

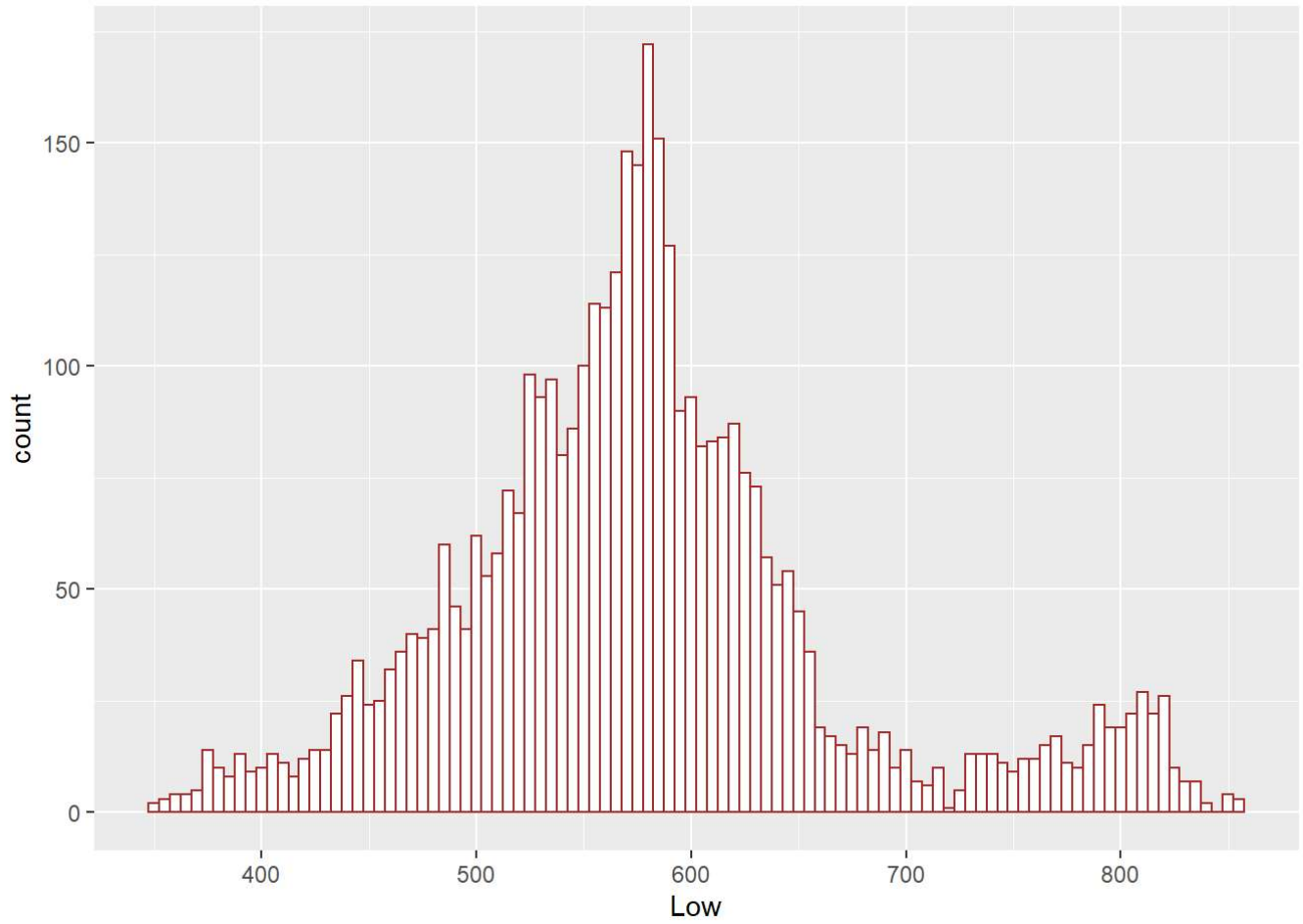
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



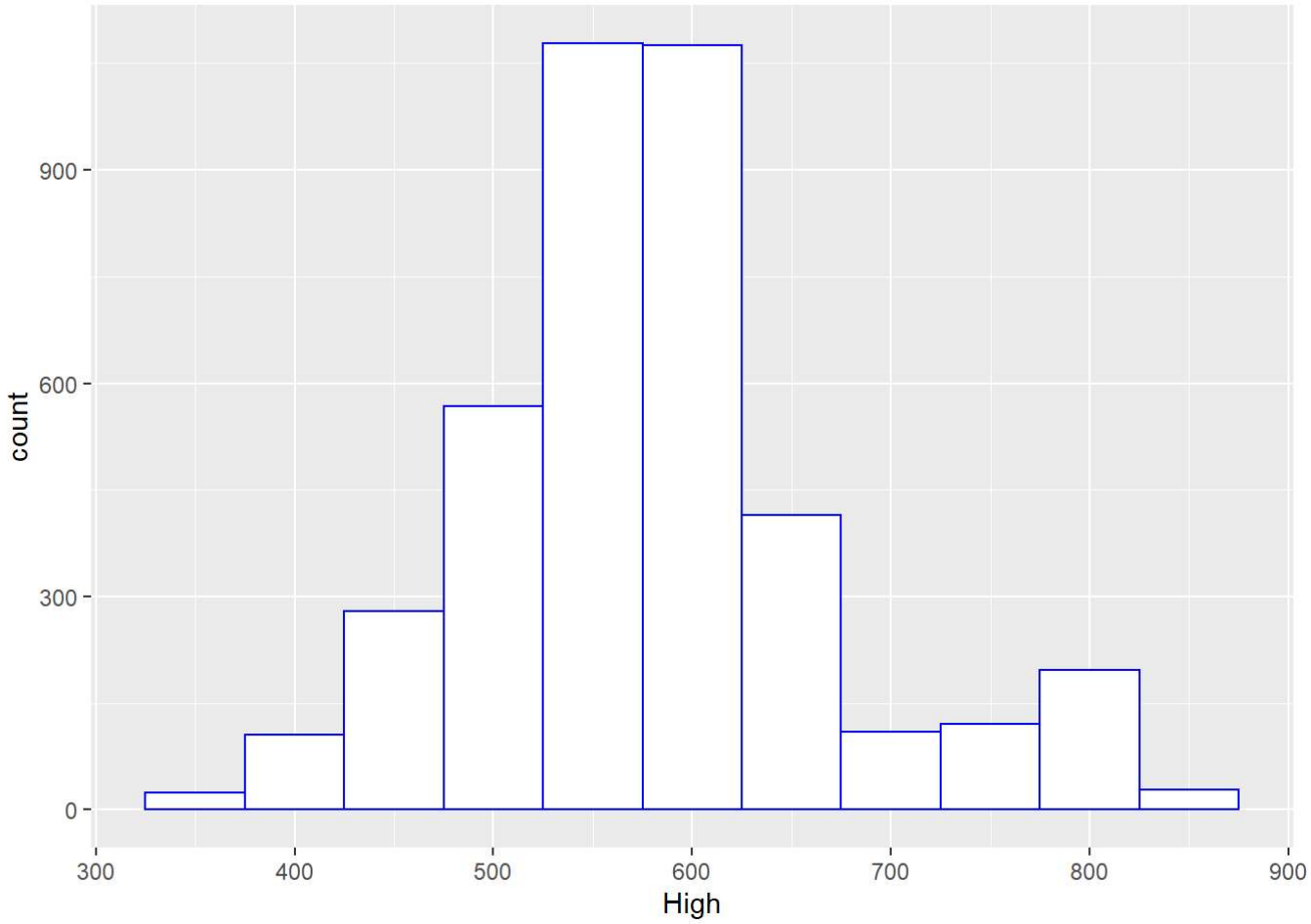

```
ggplot(df, aes(x= High)) +
geom_histogram(binwidth = 5, fill = "white", colour = "black")
```



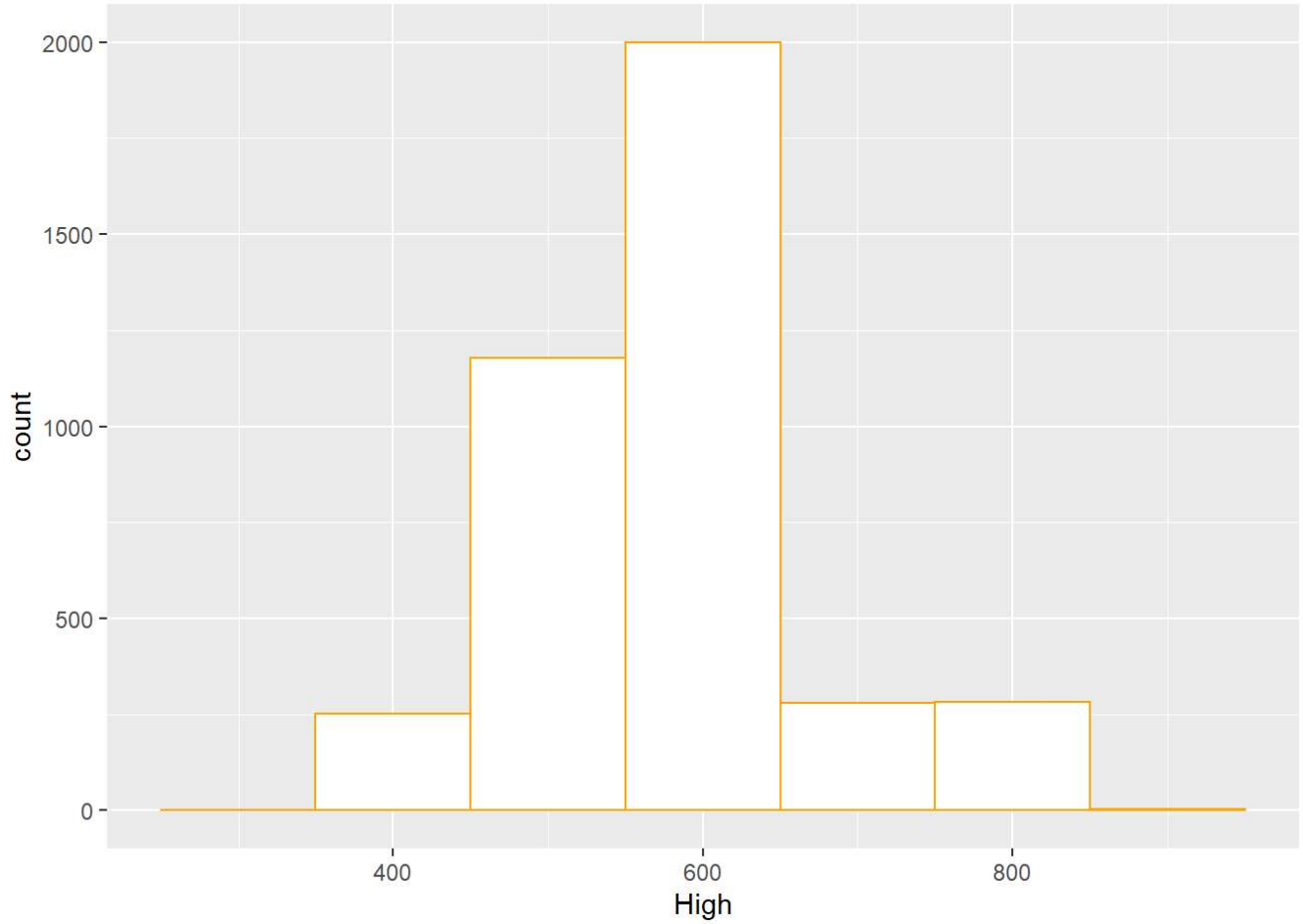

```
ggplot(df, aes(x= Low)) +
geom_histogram(binwidth = 5, fill = "white", colour = "brown")
```



```
ggplot(df, aes(x= High)) +
geom_histogram(binwidth = 50, fill = "white", colour = "blue")
```



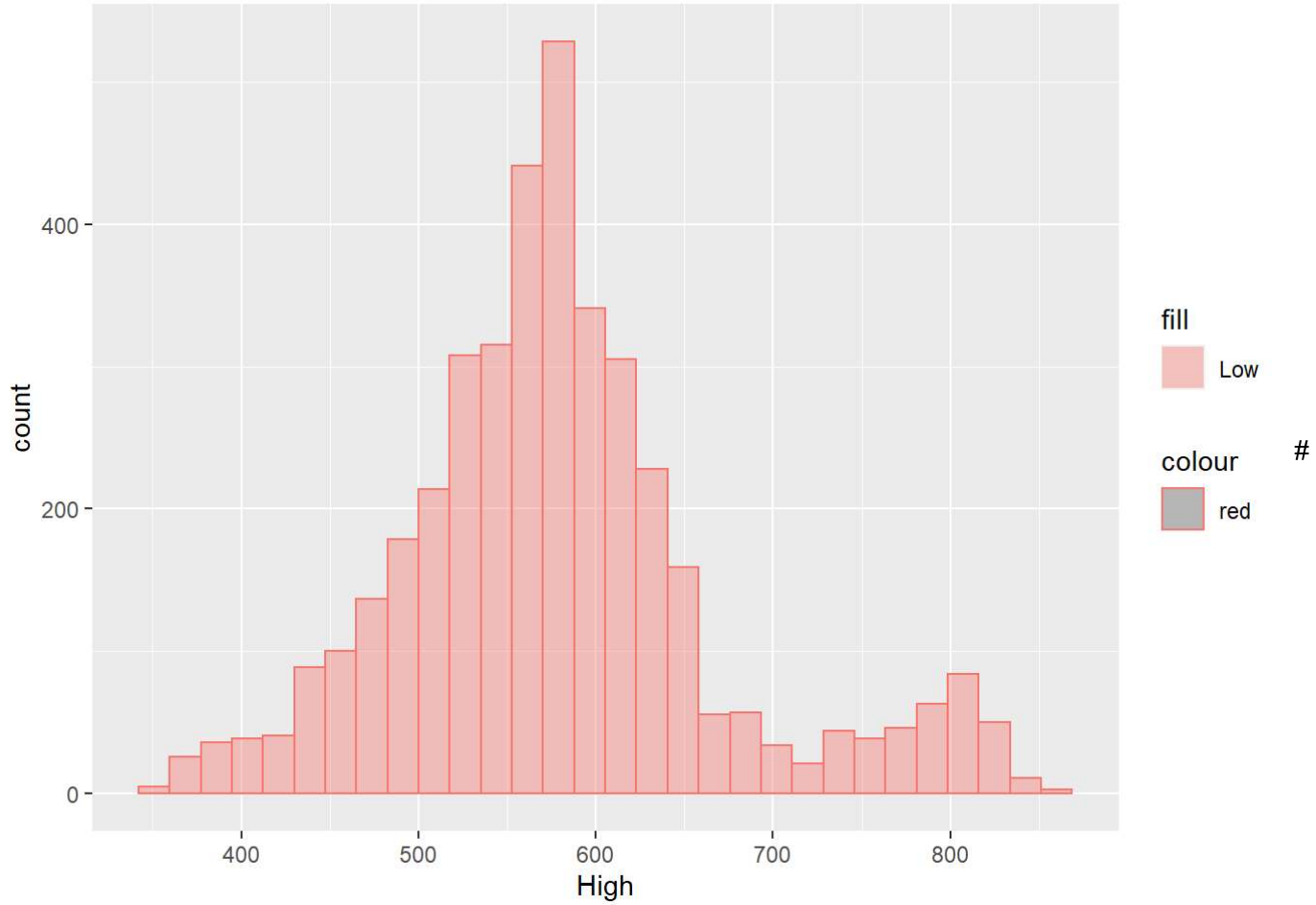

```
ggplot(df, aes(x= High)) +
geom_histogram(binwidth = 100, fill = "white", colour = "orange")
```



Map pretest to fill, make the bars NOT stacked, and make them semitransparent

```
ggplot(df, aes(x = High, fill ='Low',colour='red' )) +
geom_histogram(position = "identity", alpha = 0.4)
```

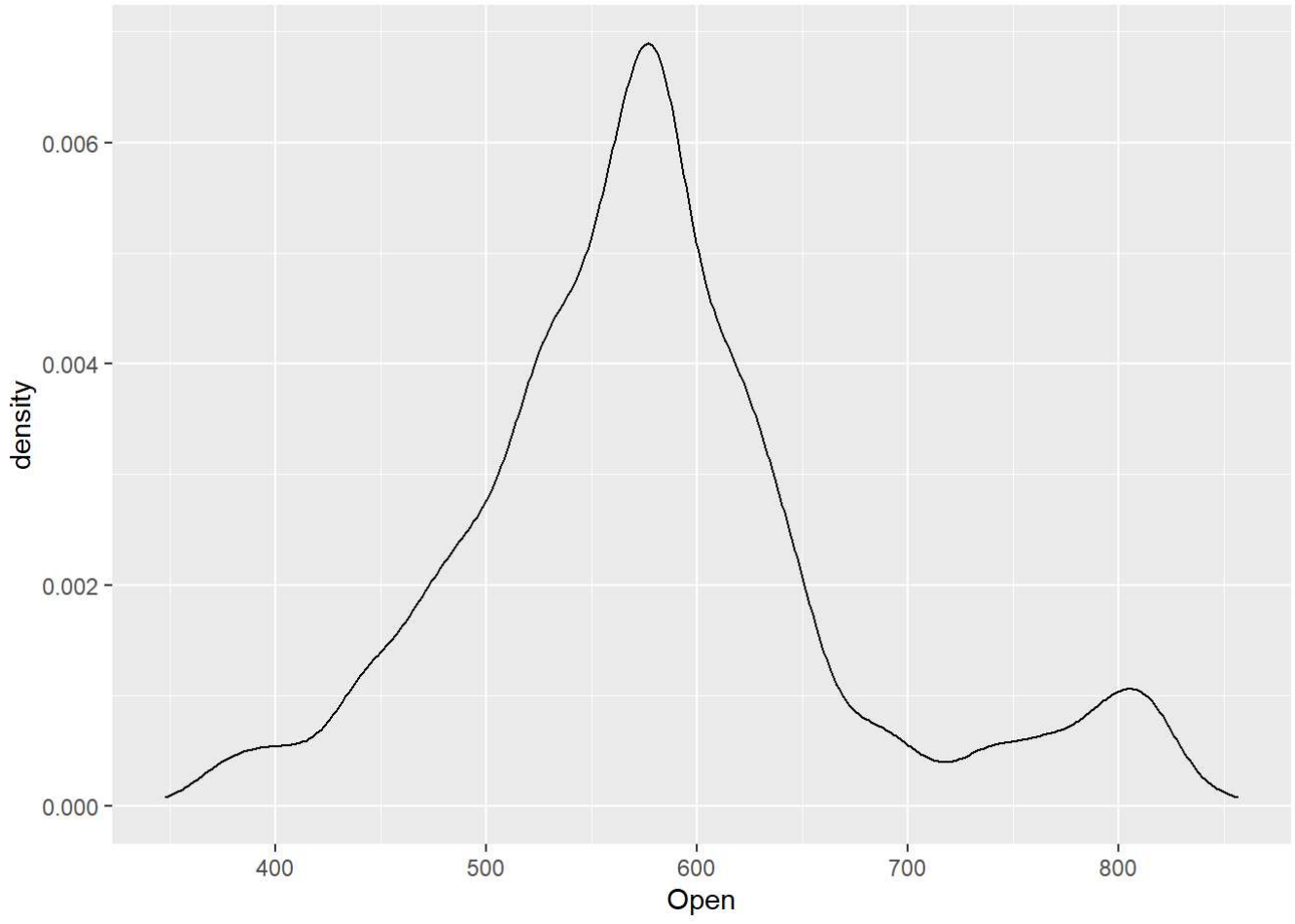
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



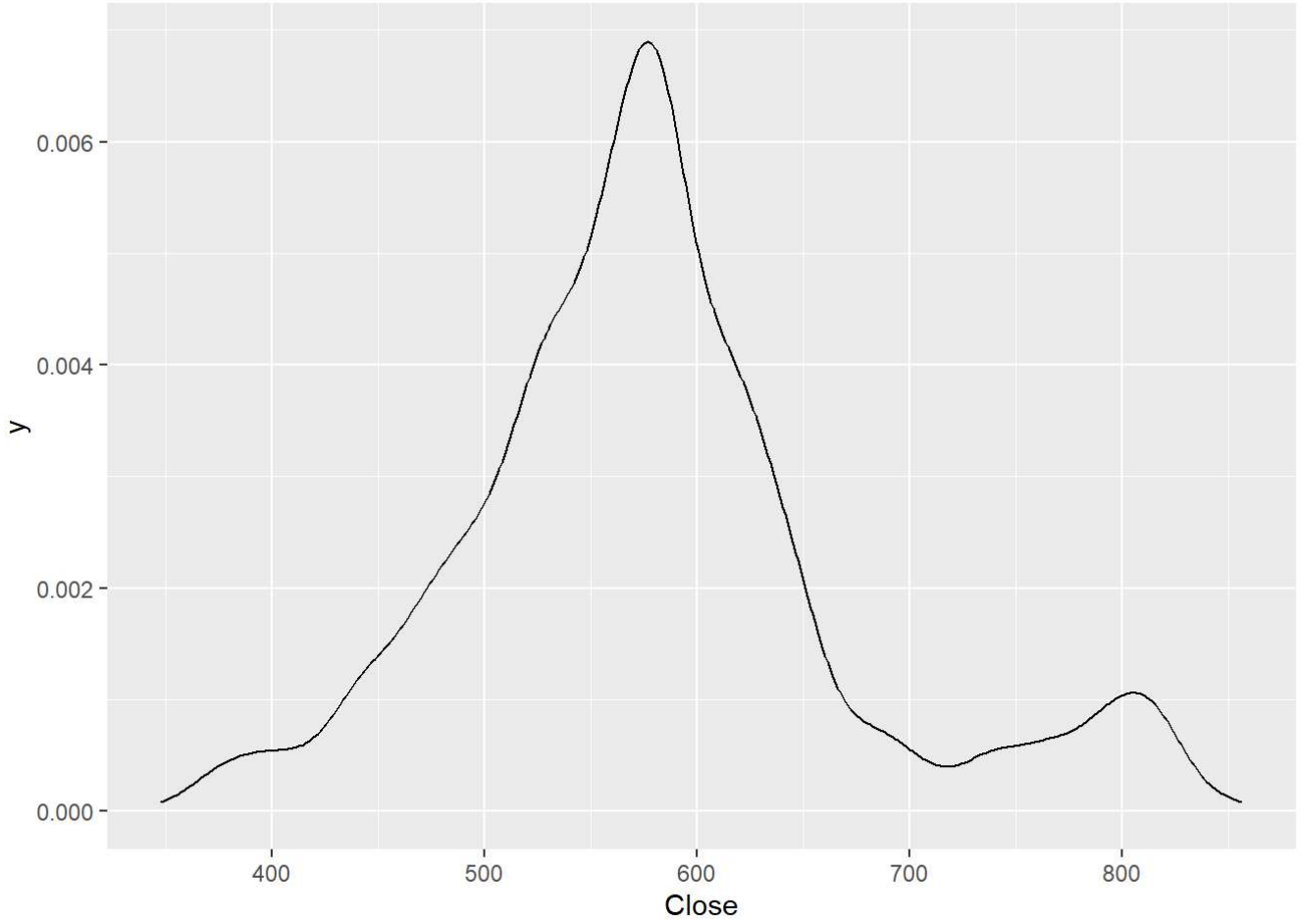
#

Making a Density Curve

```
ggplot(df, aes(x = Open)) +geom_density()
```



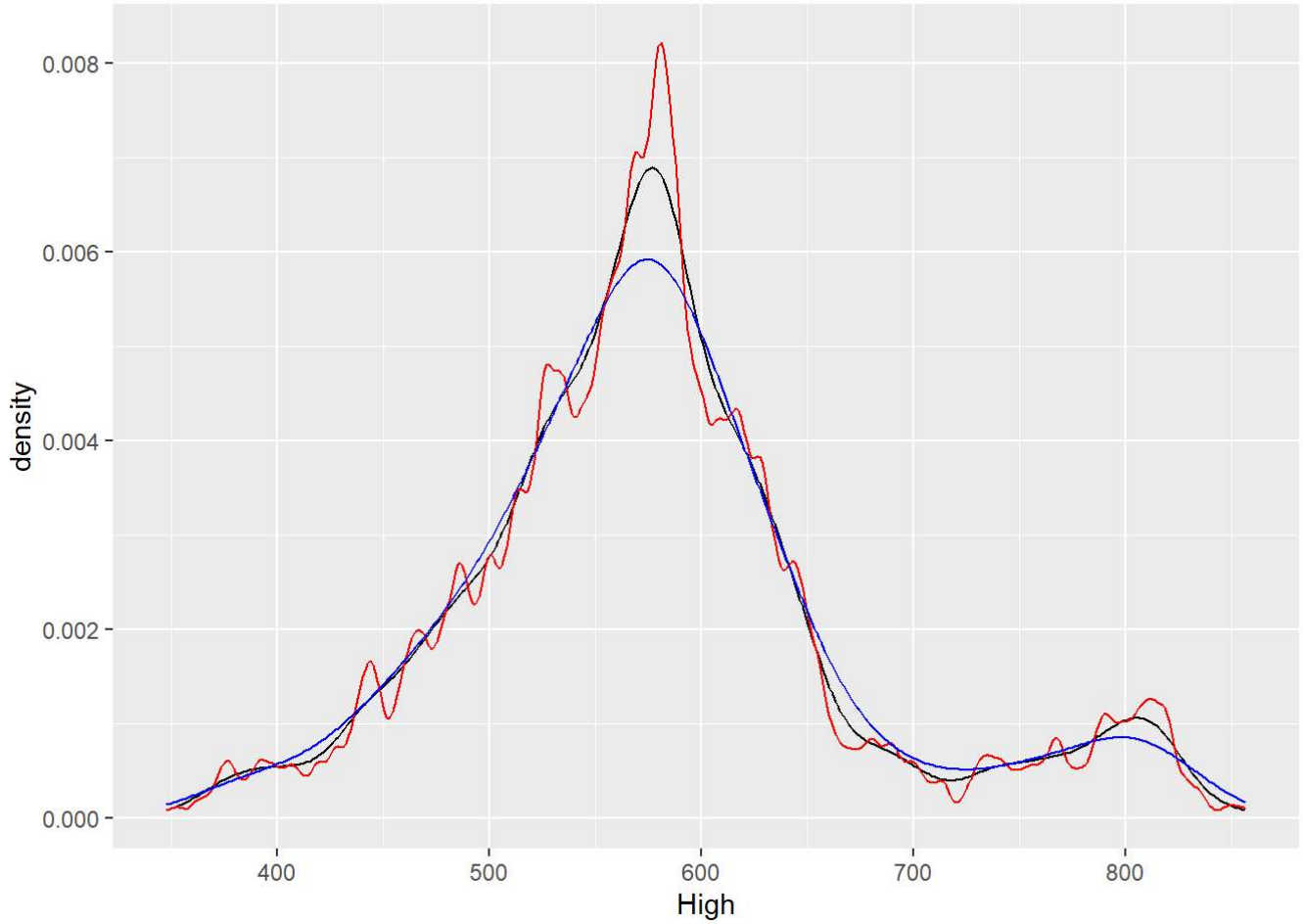
```
# expand_limits() increases the y range to include the value 0
ggplot(df, aes(x = Close)) +geom_line(stat = "density") +expand_limits(y = 0)
```



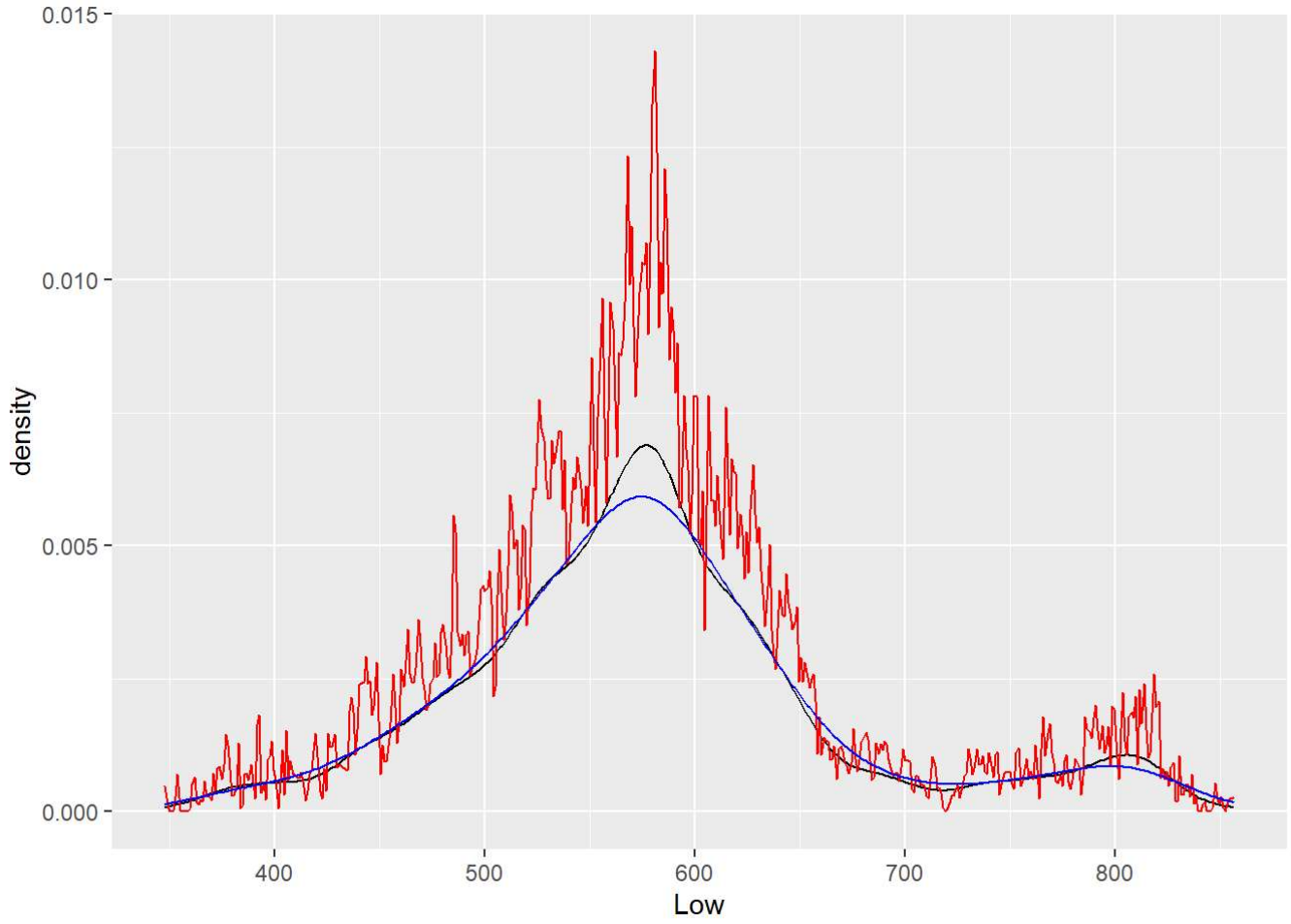
#

Density curve with a smaller and larger value of adjust:

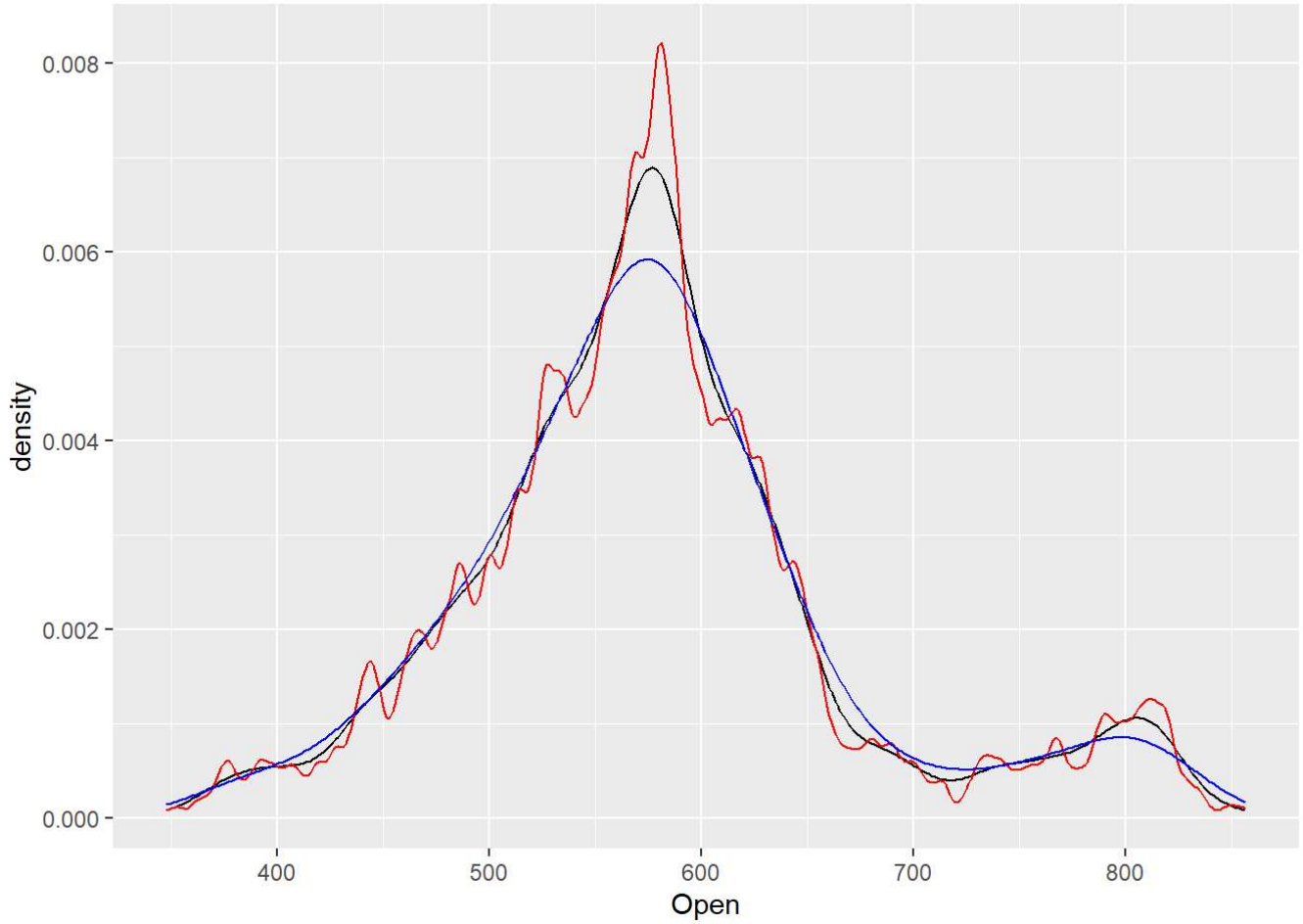
```
ggplot(df, aes(x = High)) +geom_line(stat = "density") +
geom_line(stat = "density", adjust = .25, colour = "red") +
geom_line(stat = "density", adjust = 2, colour = "blue")
```



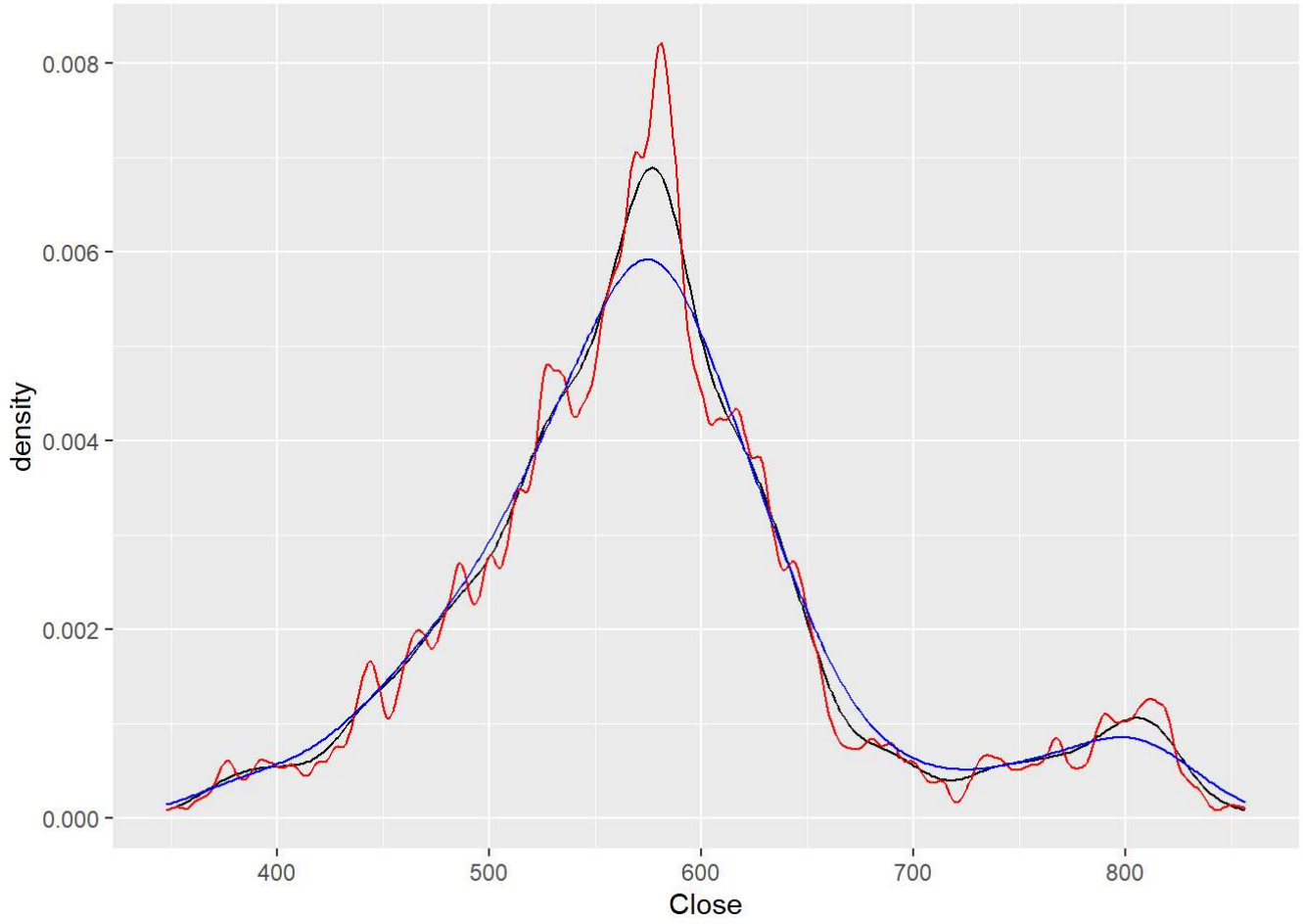

```
ggplot(df, aes(x = Low)) +geom_line(stat = "density") +
geom_line(stat = "density", adjust = .025, colour = "red") +
geom_line(stat = "density", adjust = 2, colour = "blue")
```




```
ggplot(df, aes(x = Open)) +geom_line(stat = "density") +
geom_line(stat = "density", adjust = .25, colour = "red") +
geom_line(stat = "density", adjust = 2, colour = "blue")
```

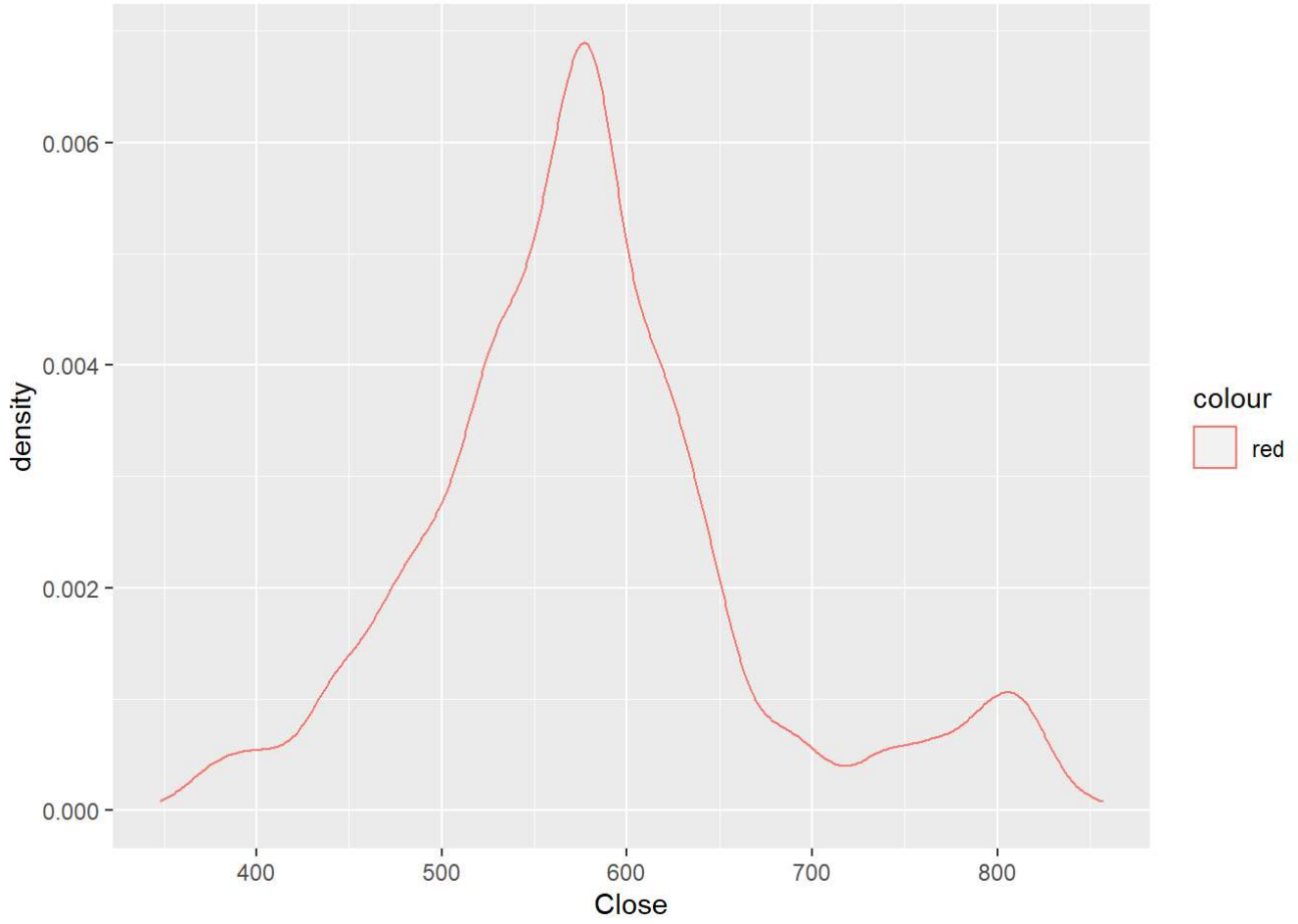



```
ggplot(df, aes(x = Close)) +geom_line(stat = "density") +
geom_line(stat = "density", adjust = .25, colour = "red") +
geom_line(stat = "density", adjust = 2, colour = "blue")
```

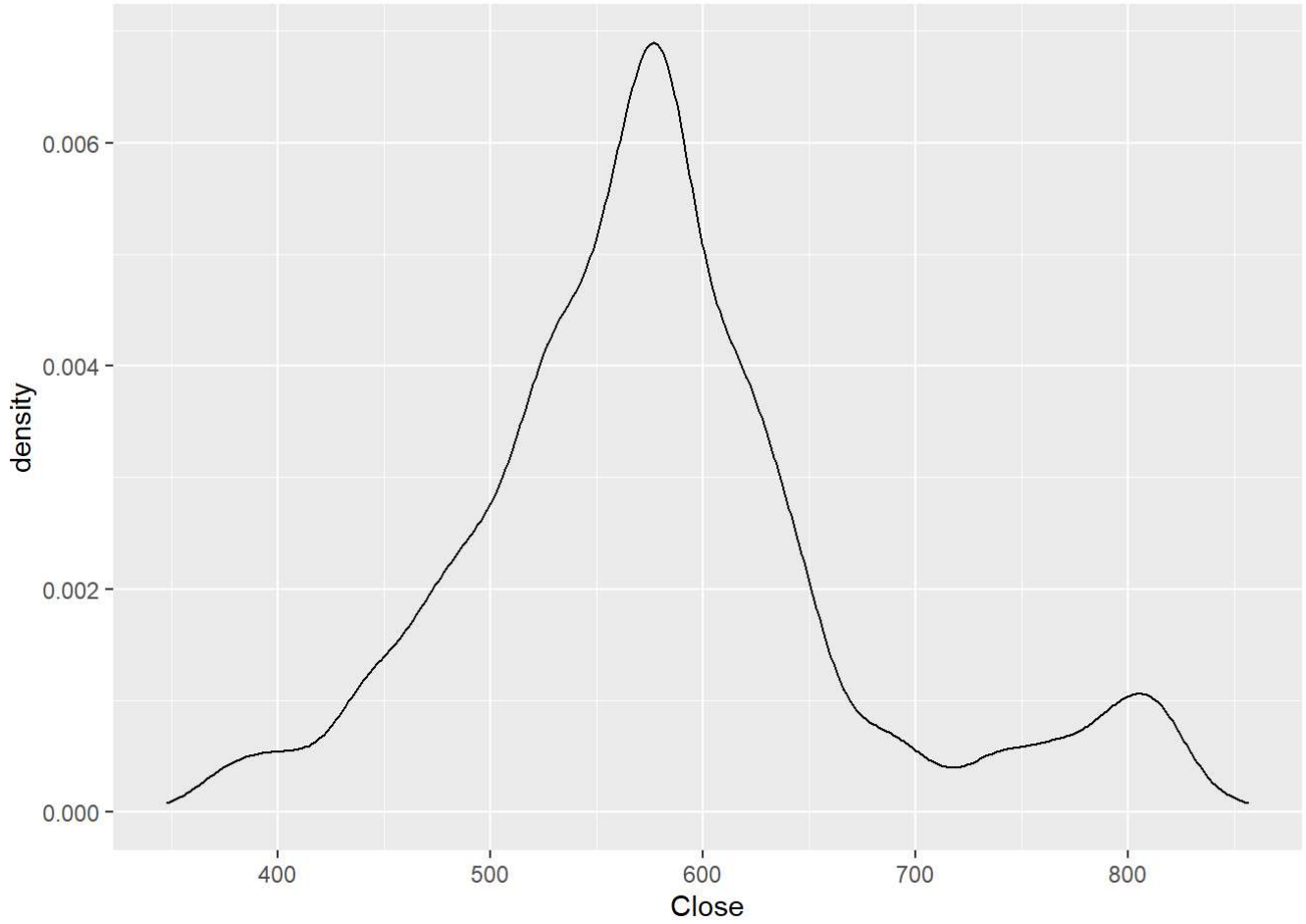


Making Multiple Density Curves from Grouped Data

```
data5 <- df %>%
mutate(A = as.factor(Open)) # Convert n_student to a factor
# Map n_student to colour
ggplot(data5, aes(x = Close, colour = "red")) +geom_density()
```




```
# Map n_student to fill and make the fill semitransparent by setting alpha
ggplot(data5, aes(x = Close, fill = Open)) +geom_density(alpha = .3)
```

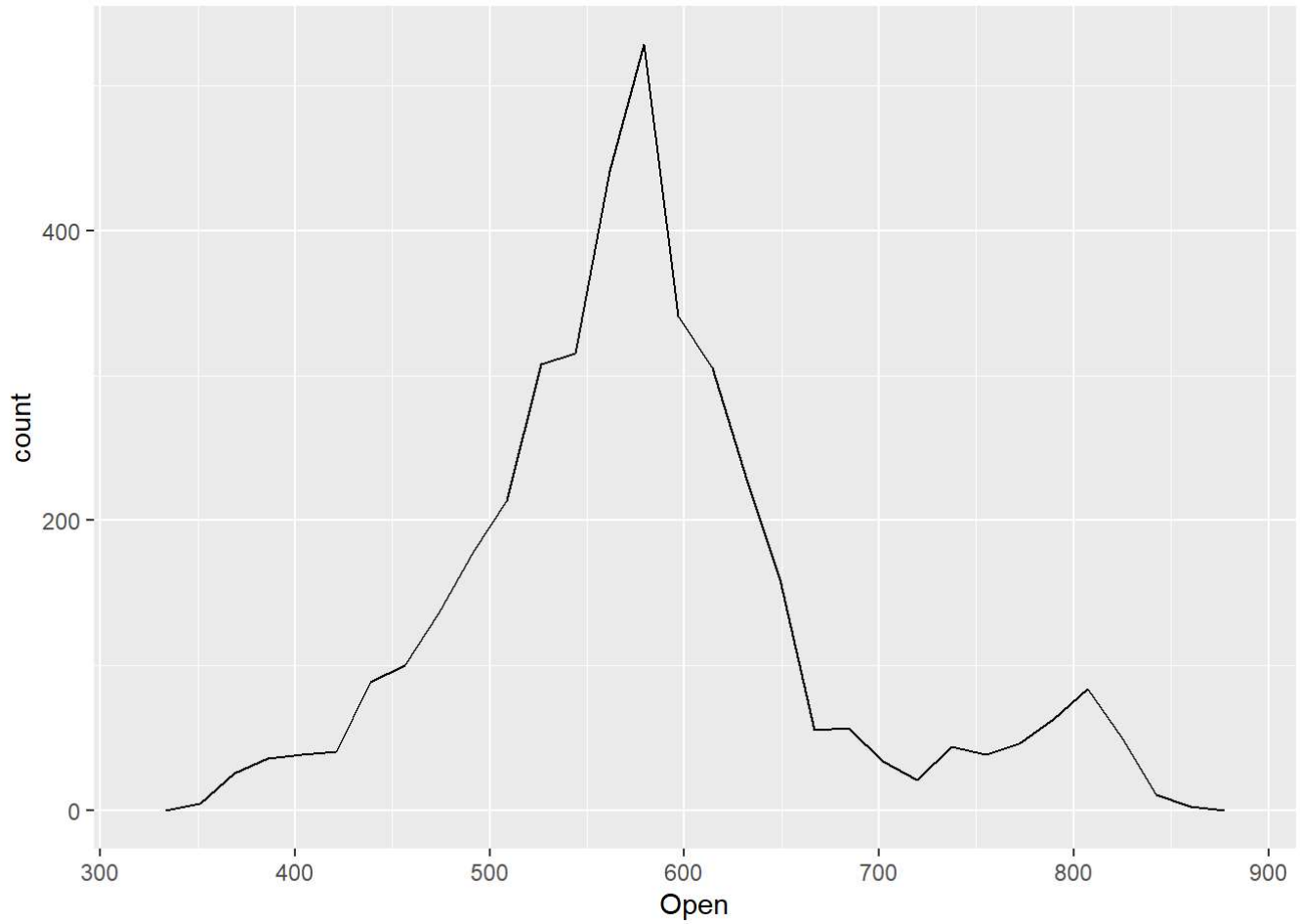


#

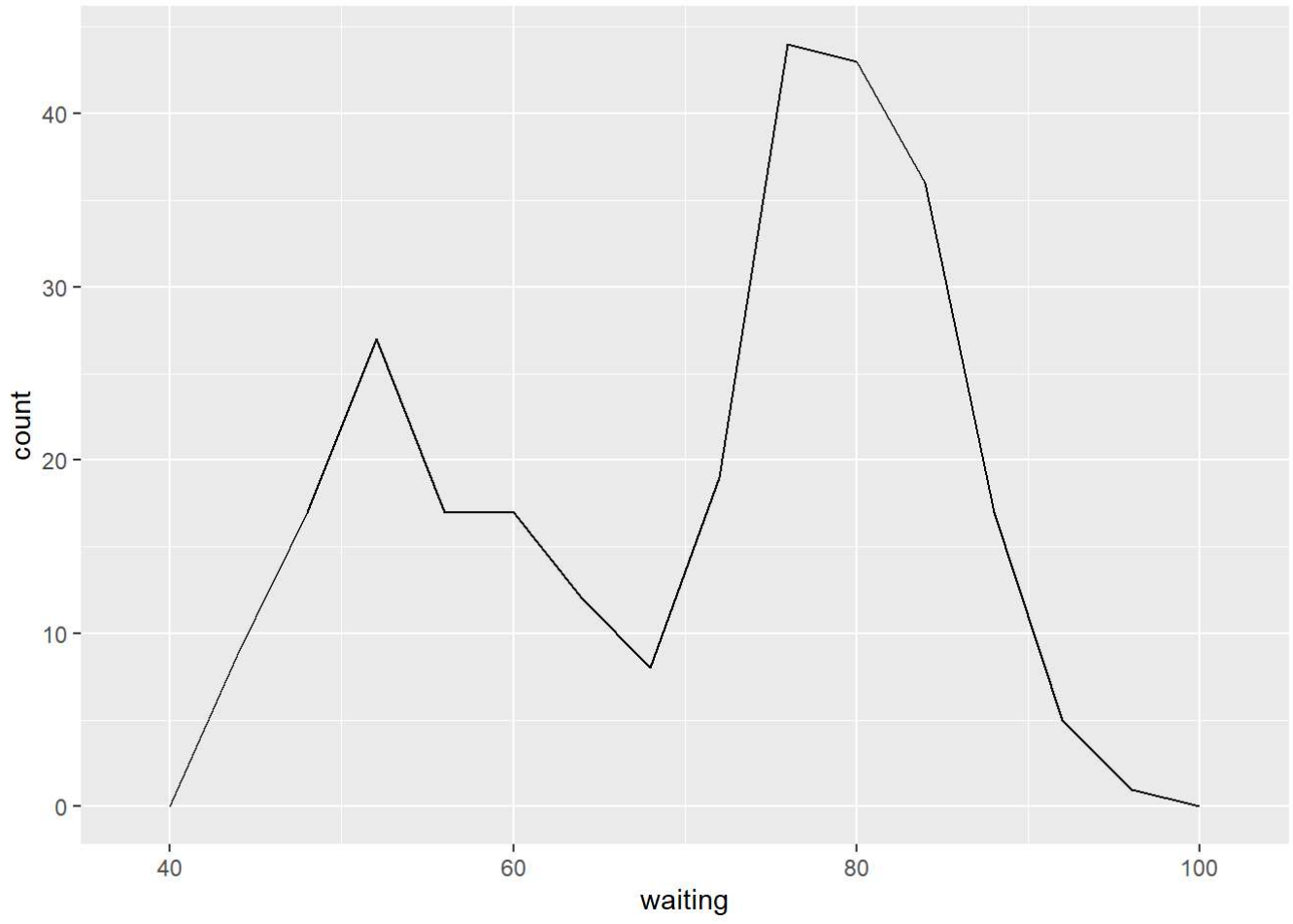
Making a Frequency Polygon

```
ggplot(df, aes(x=Open)) +
geom_freqpoly()
```

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



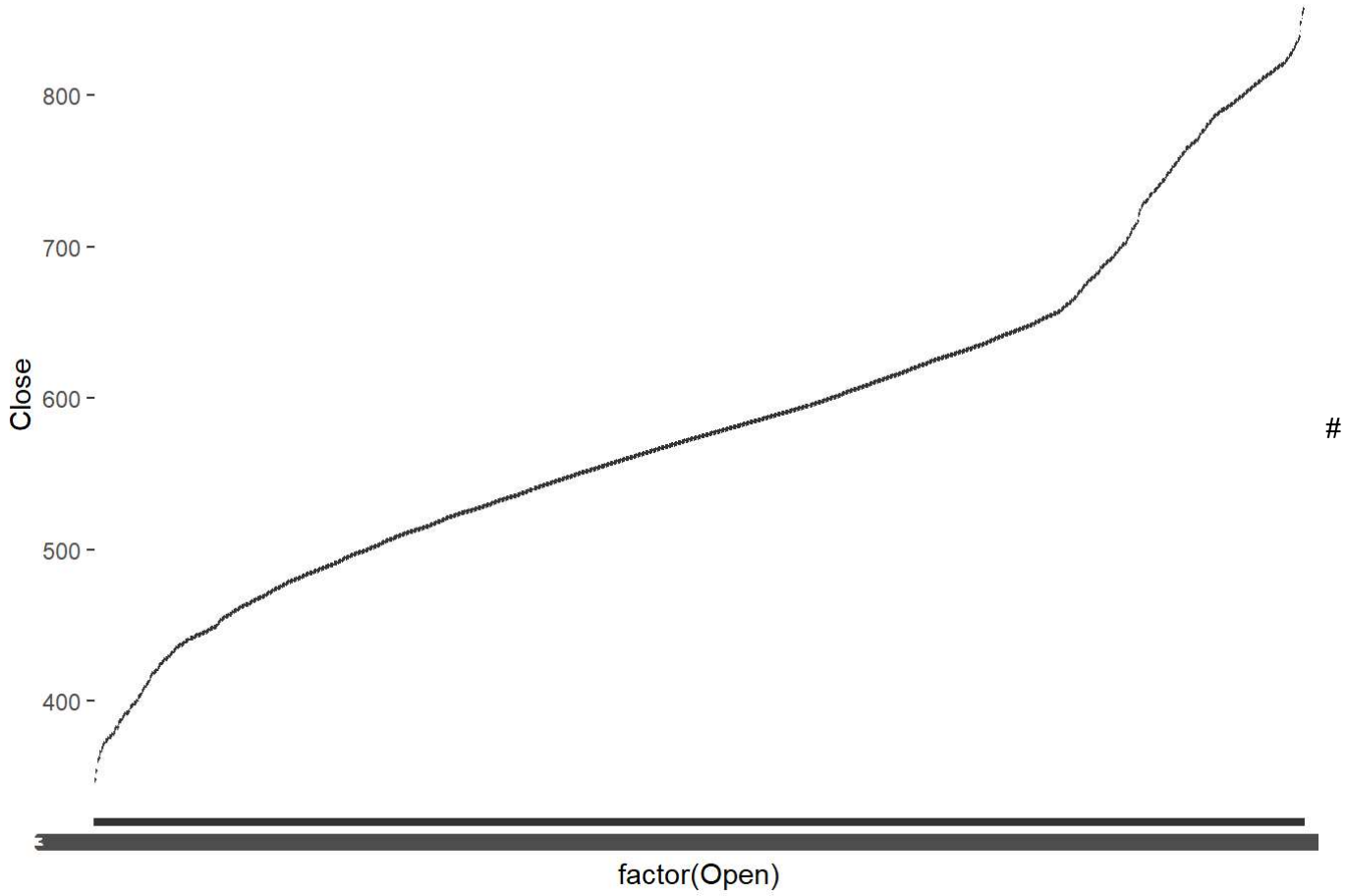

```
ggplot(faithful, aes(x = waiting)) +
geom_freqpoly(binwidth = 4)           #controlling bin width
```



Making a Basic Box Plot

```
ggplot(df, aes(x = factor(Close), y = Open)) +
geom_boxplot()
```

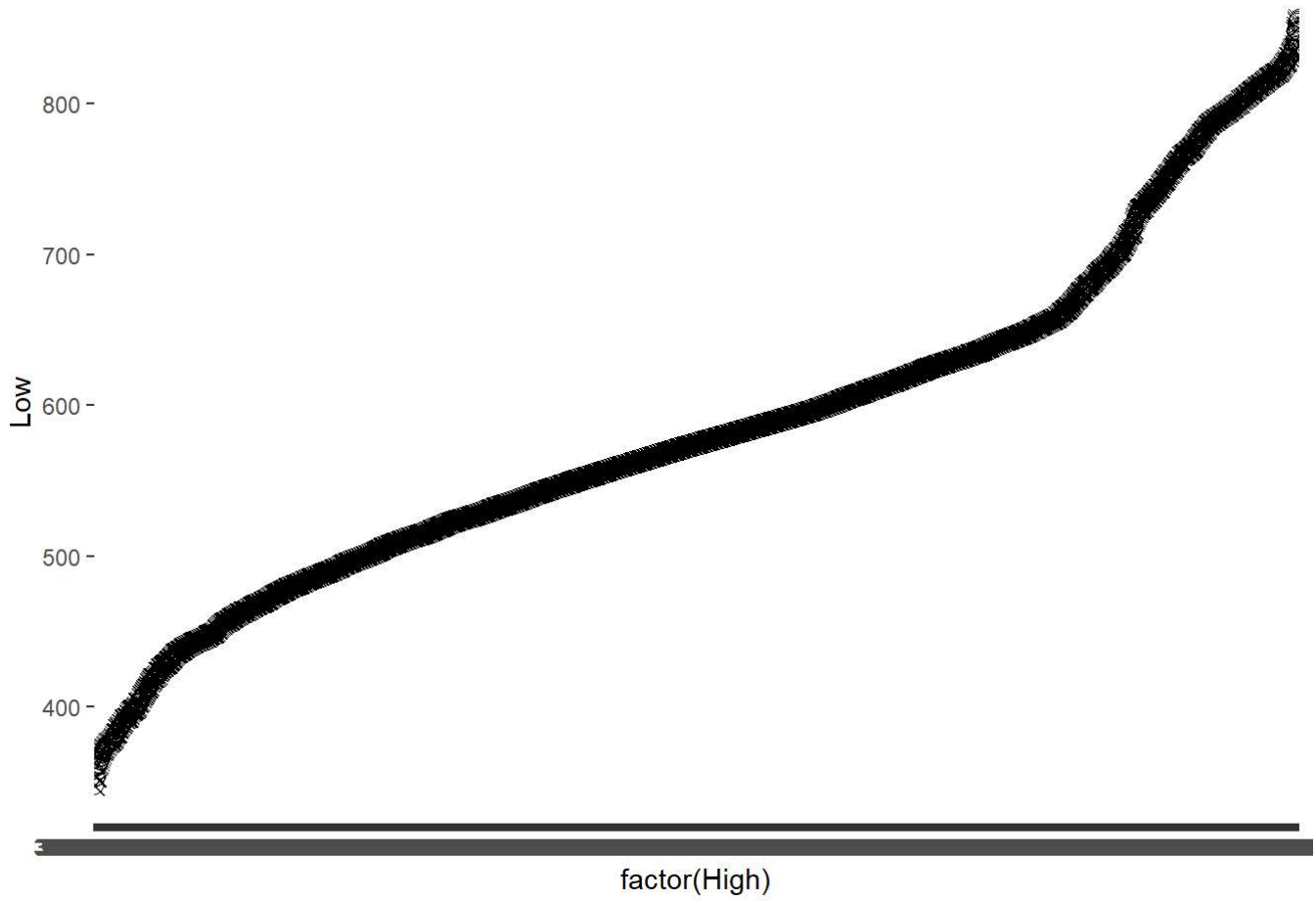

```
ggplot(df, aes(x = factor(Open), y = Close)) +geom_boxplot(notch = FALSE)
```



#

Adding means to box plot

```
ggplot(df, aes(x = factor(High), y = Low)) +geom_boxplot() +stat_summary(fun = "mean", geom =
"point", shape = 13, size = 3,
fill = "white")
```

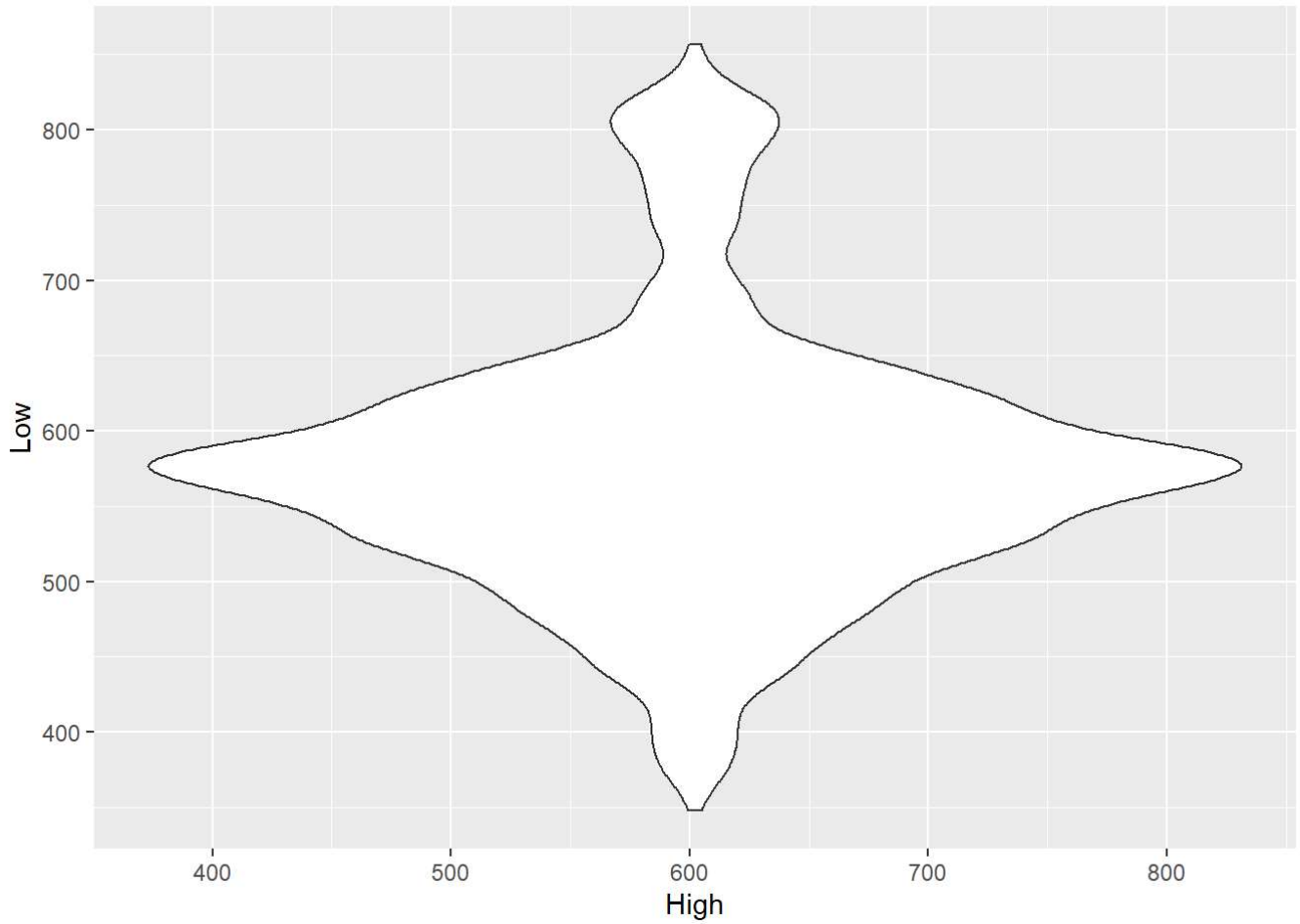



```
ggplot(df, aes(x = factor(Open), y = Close)) +geom_boxplot() +stat_summary(fun = "mean", geom
= "point", shape = 43, size = 6,
fill = "white")
```

#

Making a Violin Plot

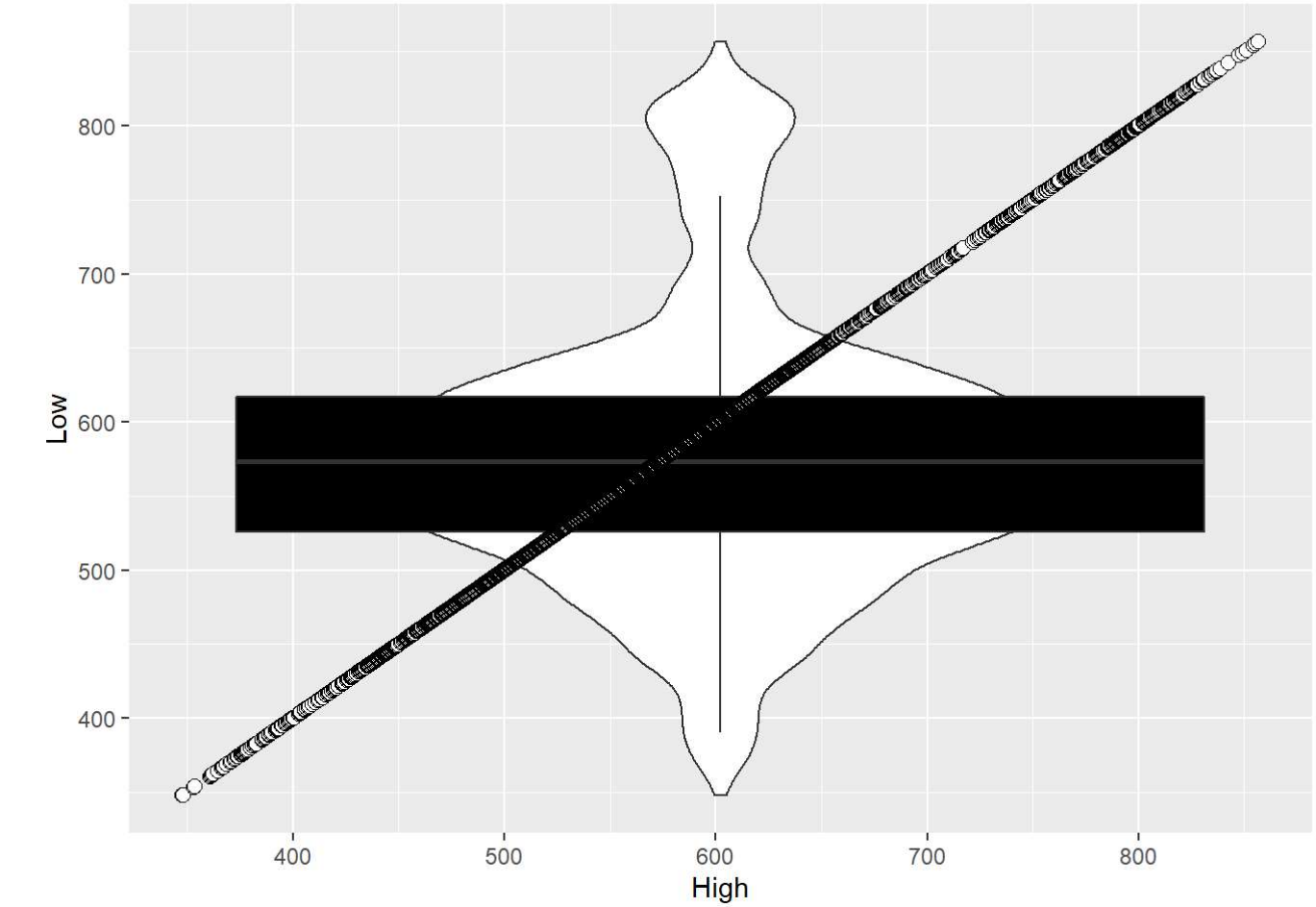
```
data6 <- ggplot(df, aes(x = High, y = Low))
data6+geom_violin()
```



A violin plot with box plot overlaid on it

```
data6+geom_violin() +geom_boxplot(width = .1, fill = "black", outlier.colour = NA) +
stat_summary(fun= median, geom = "point", fill = "white", shape = 21,
size = 2.5)
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

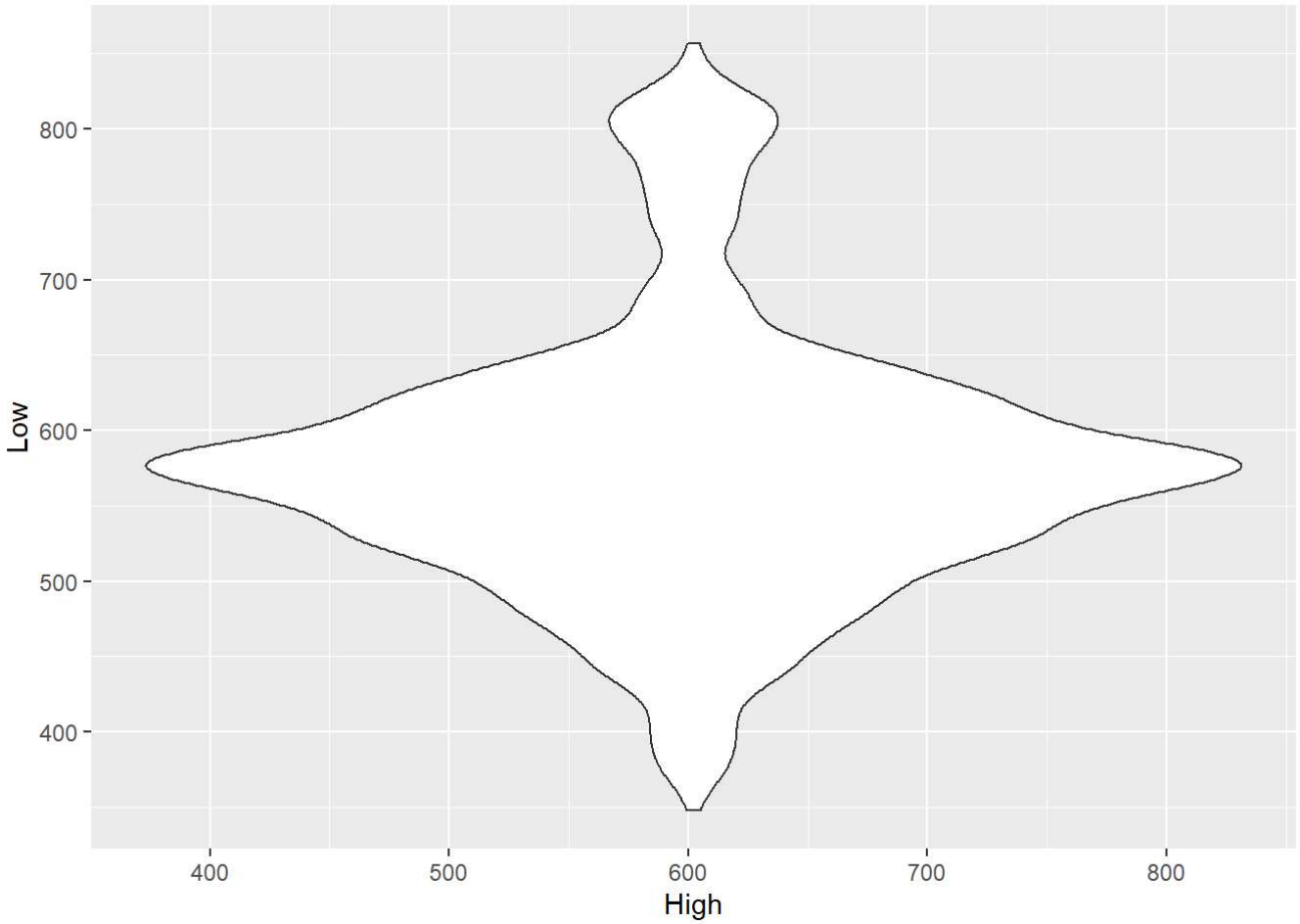


#

A violin plot with tails

```
data6+geom_violin(trim = FALSE)
```

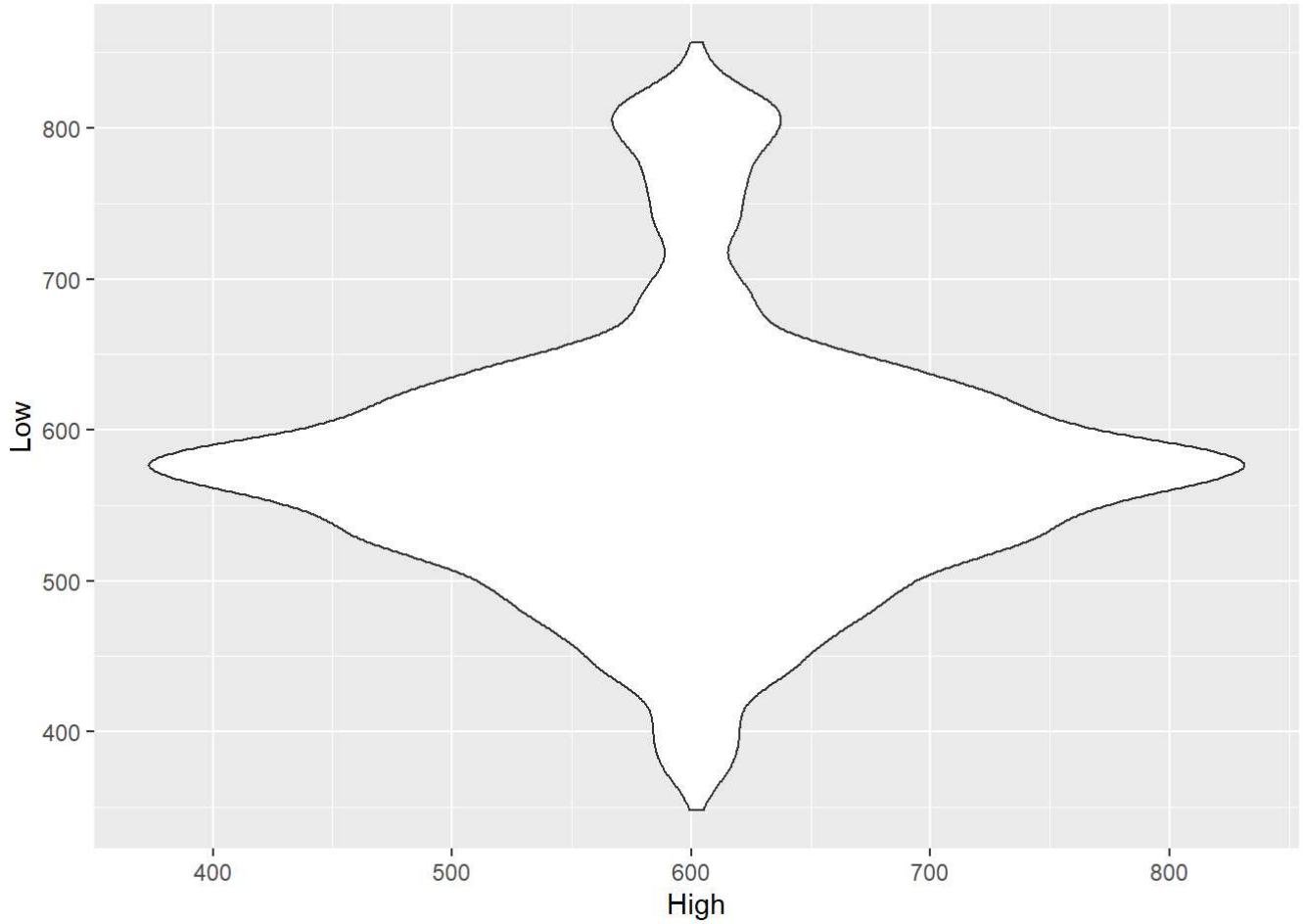
```
data6+geom_violin(trim = TRUE)
```



#

Violin plot with area proportional to number of observations

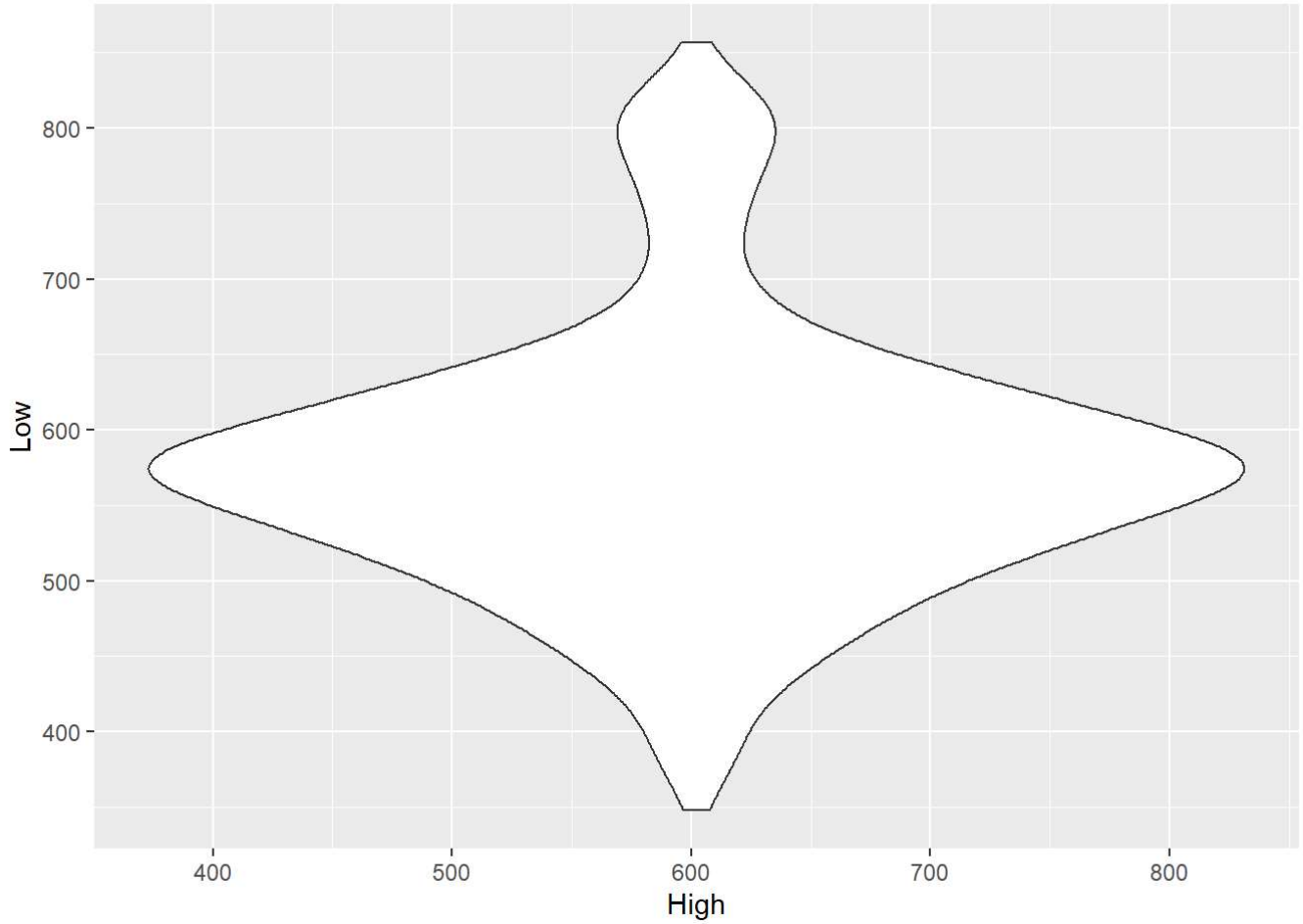
```
data6 +geom_violin(scale = "count")
```



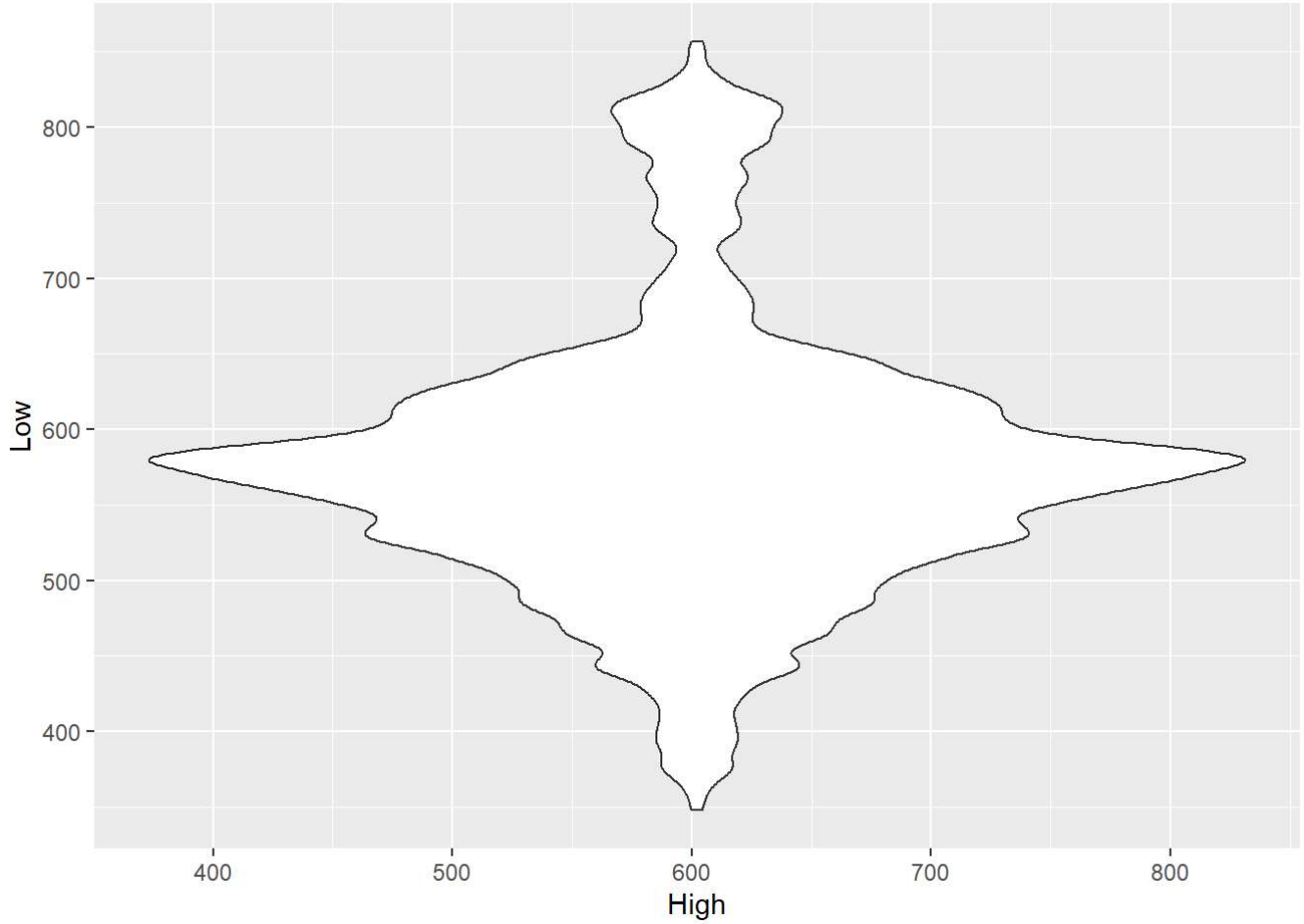
#

Violin plot with

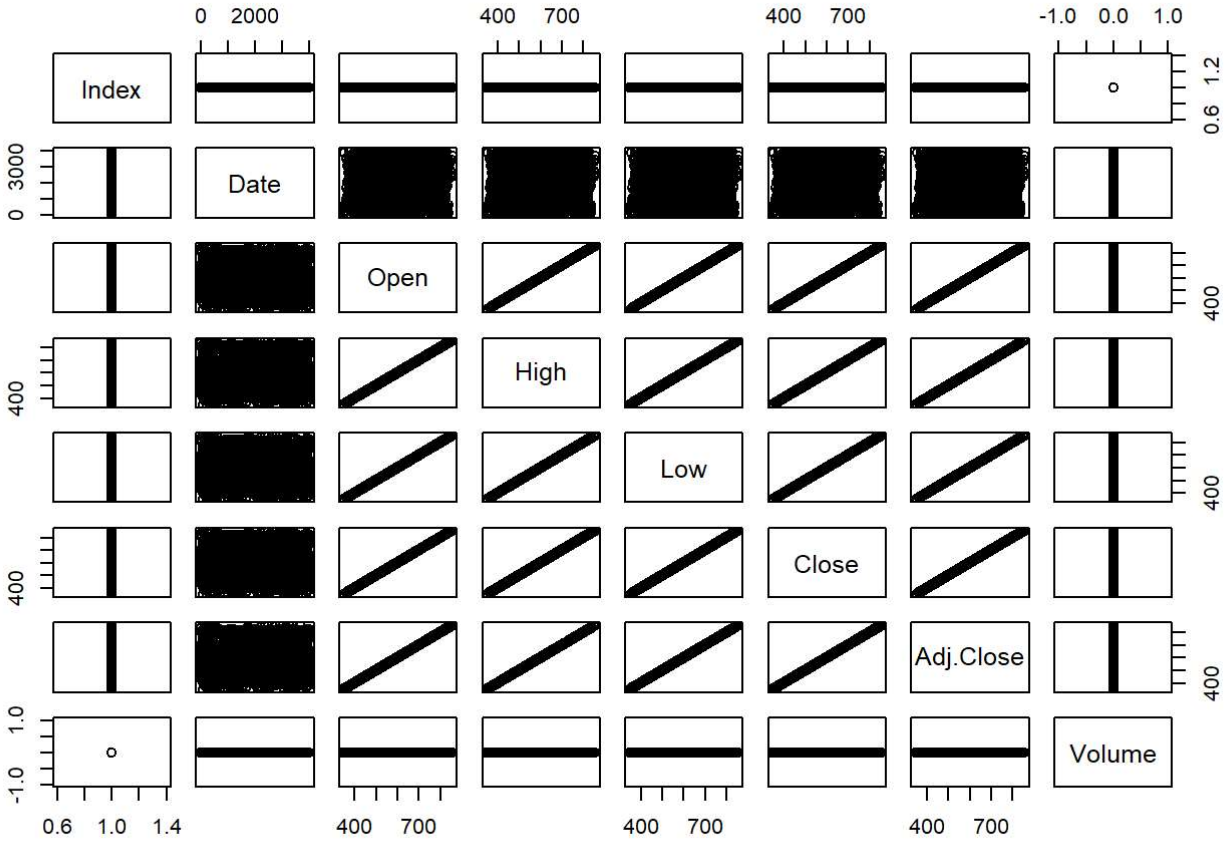
```
# More smoothing
data6+geom_violin(adjust = 2)
```




```
# Less smoothing
data6 +geom_violin(adjust = .5)
```

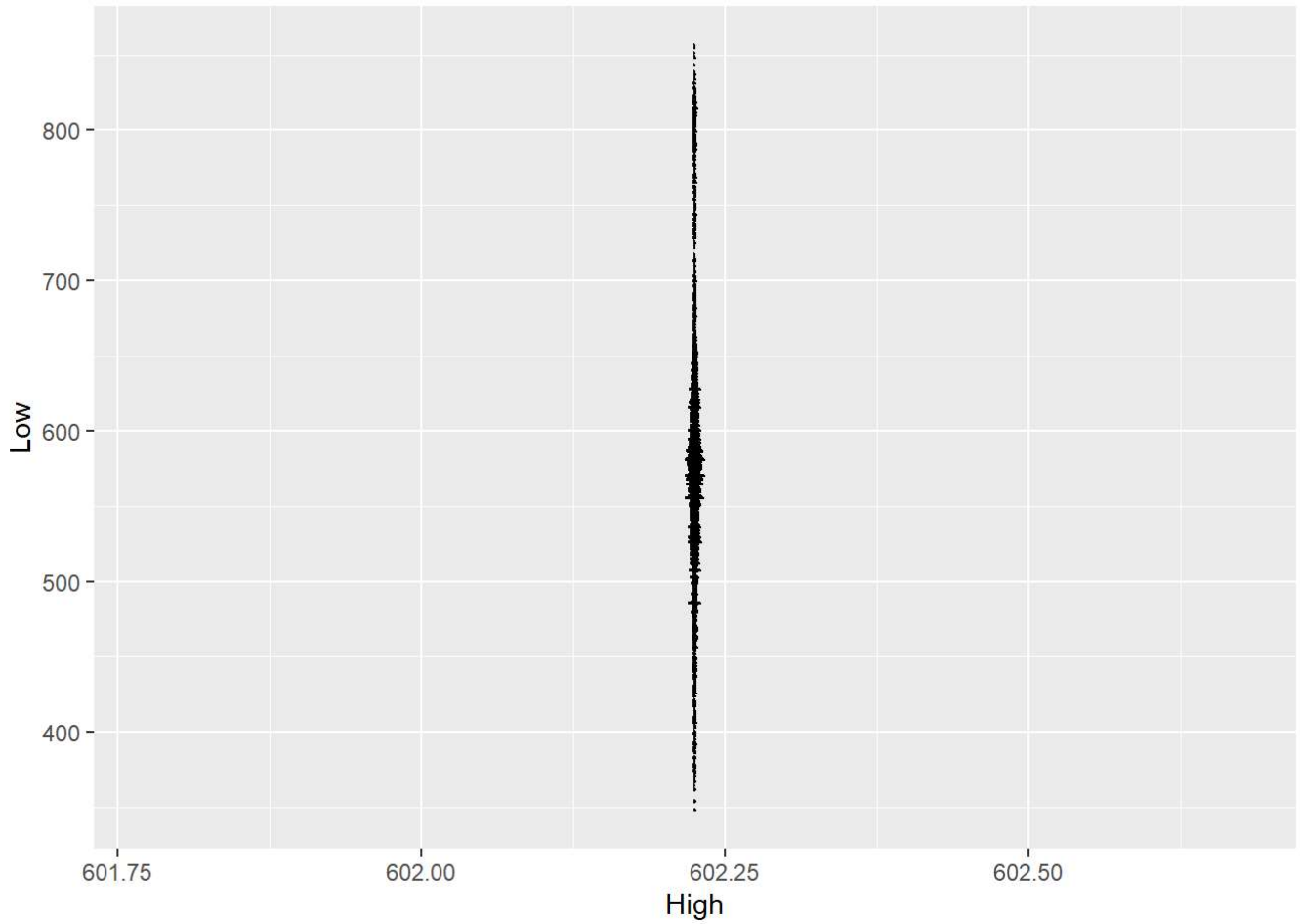


```
plot(df)
```

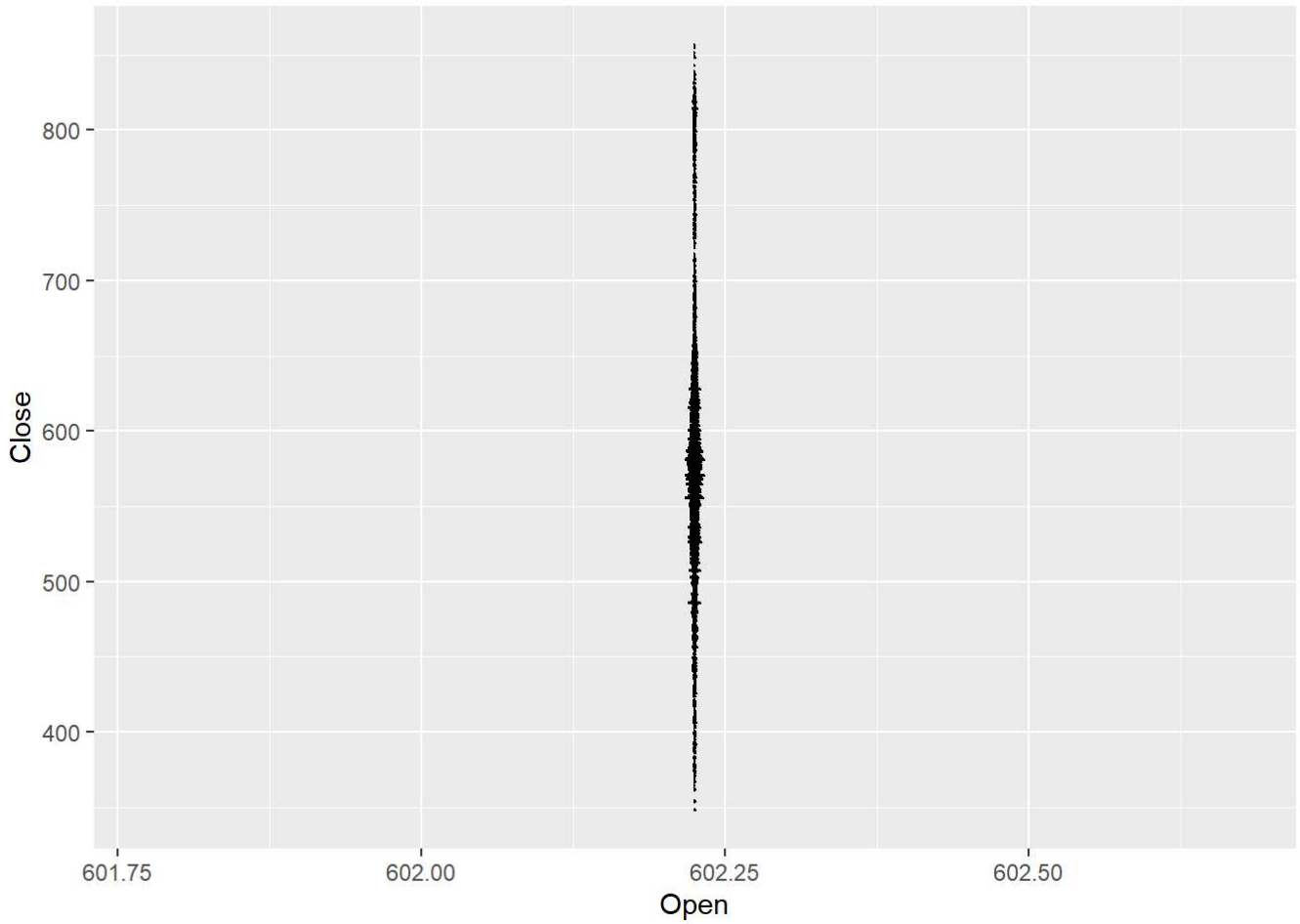


Making Multiple Dot Plots for Grouped Data

```
ggplot(df, aes(x = High, y = Low)) +
geom_dotplot(binaxis = "y", binwidth = .5, stackdir = "center")
```




```
ggplot(df, aes(x = Open, y = Close)) +
geom_dotplot(binaxis = "y", binwidth = .5, stackdir = "center")
```

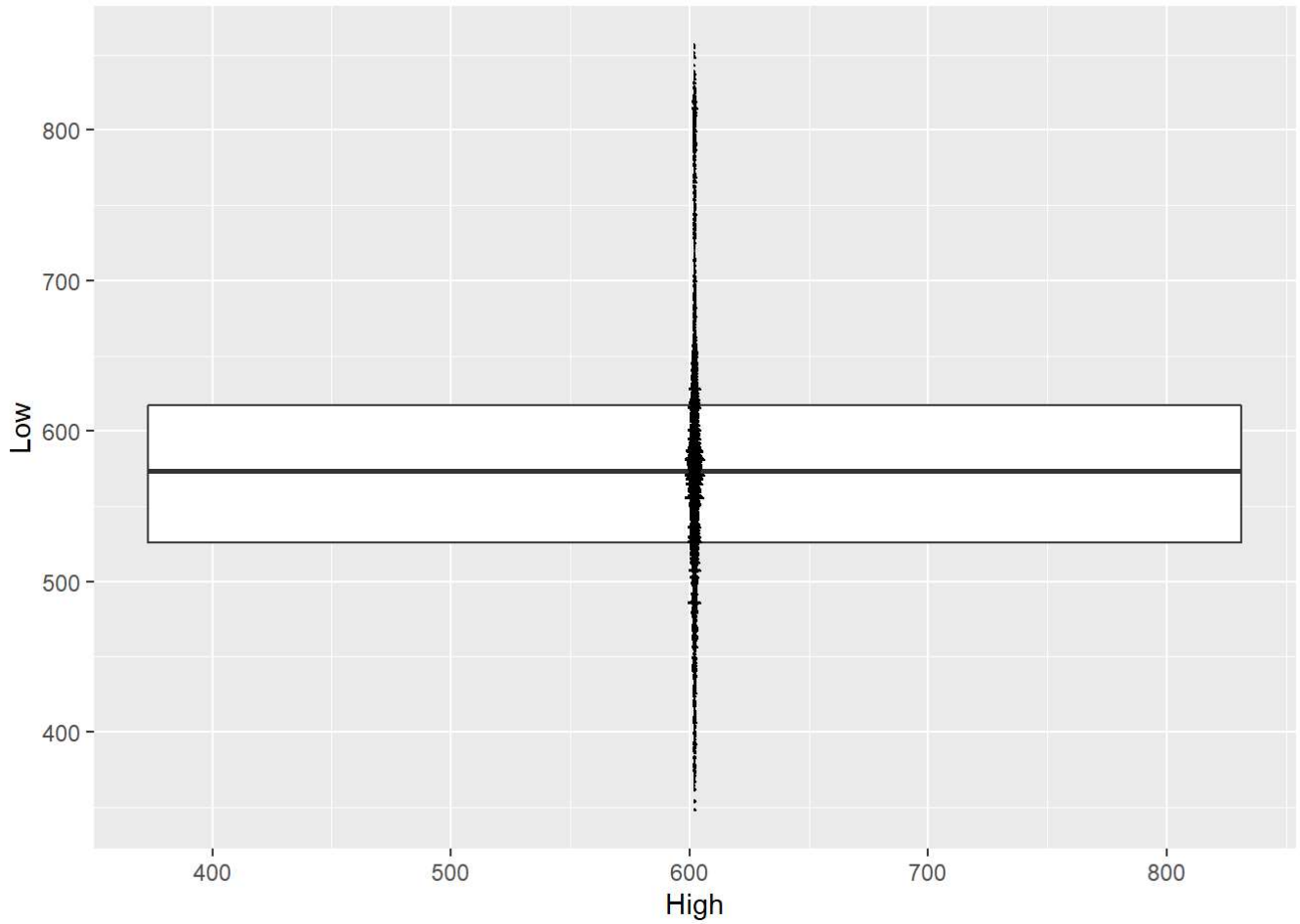


#

Dot plot overlaid on box plot

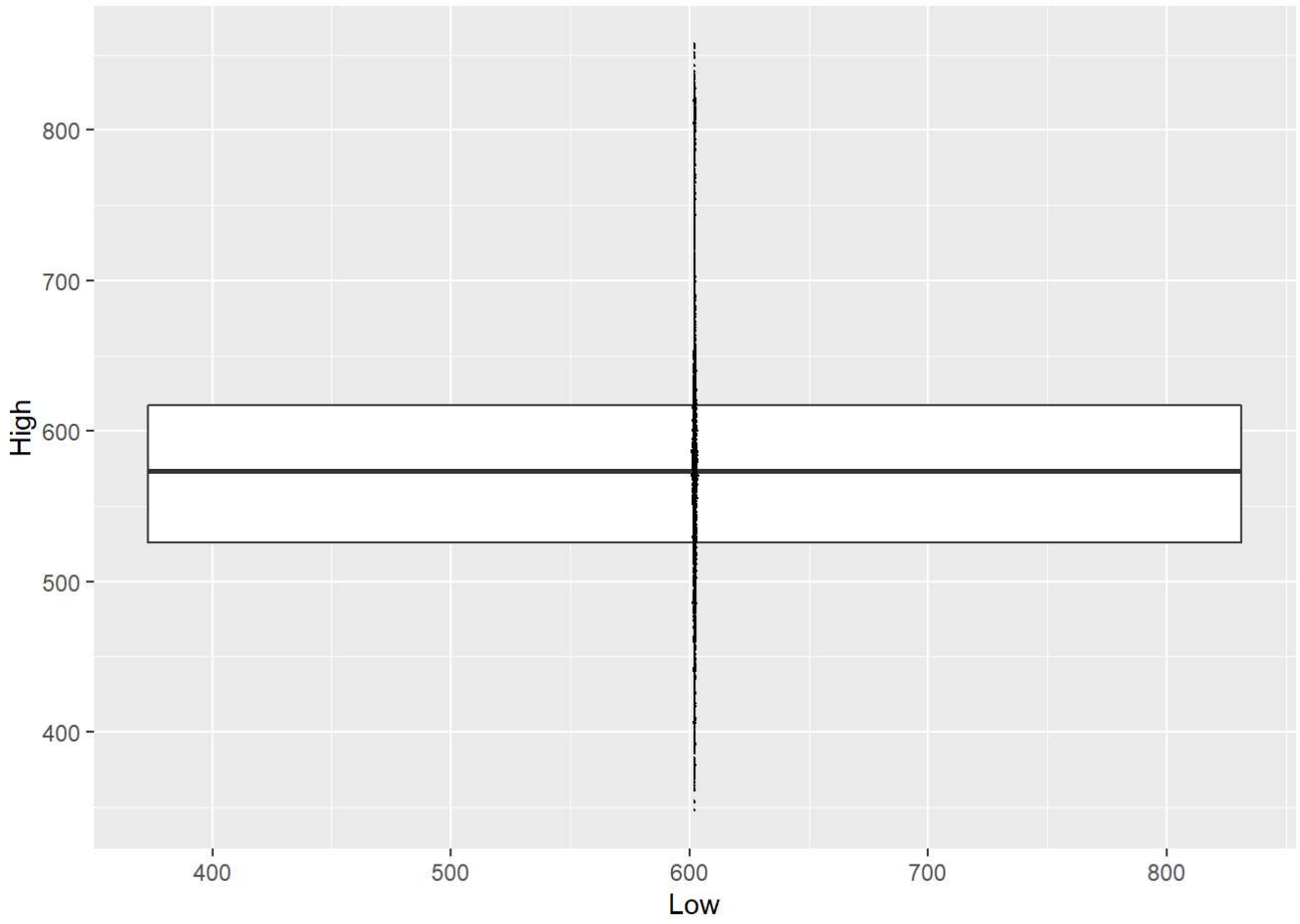
```
ggplot(df, aes(x = High, y = Low)) +
geom_boxplot(outlier.colour = NA, width = .4) +
geom_dotplot(binaxis = "y", binwidth = .5, stackdir = "center", fill = NA)
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



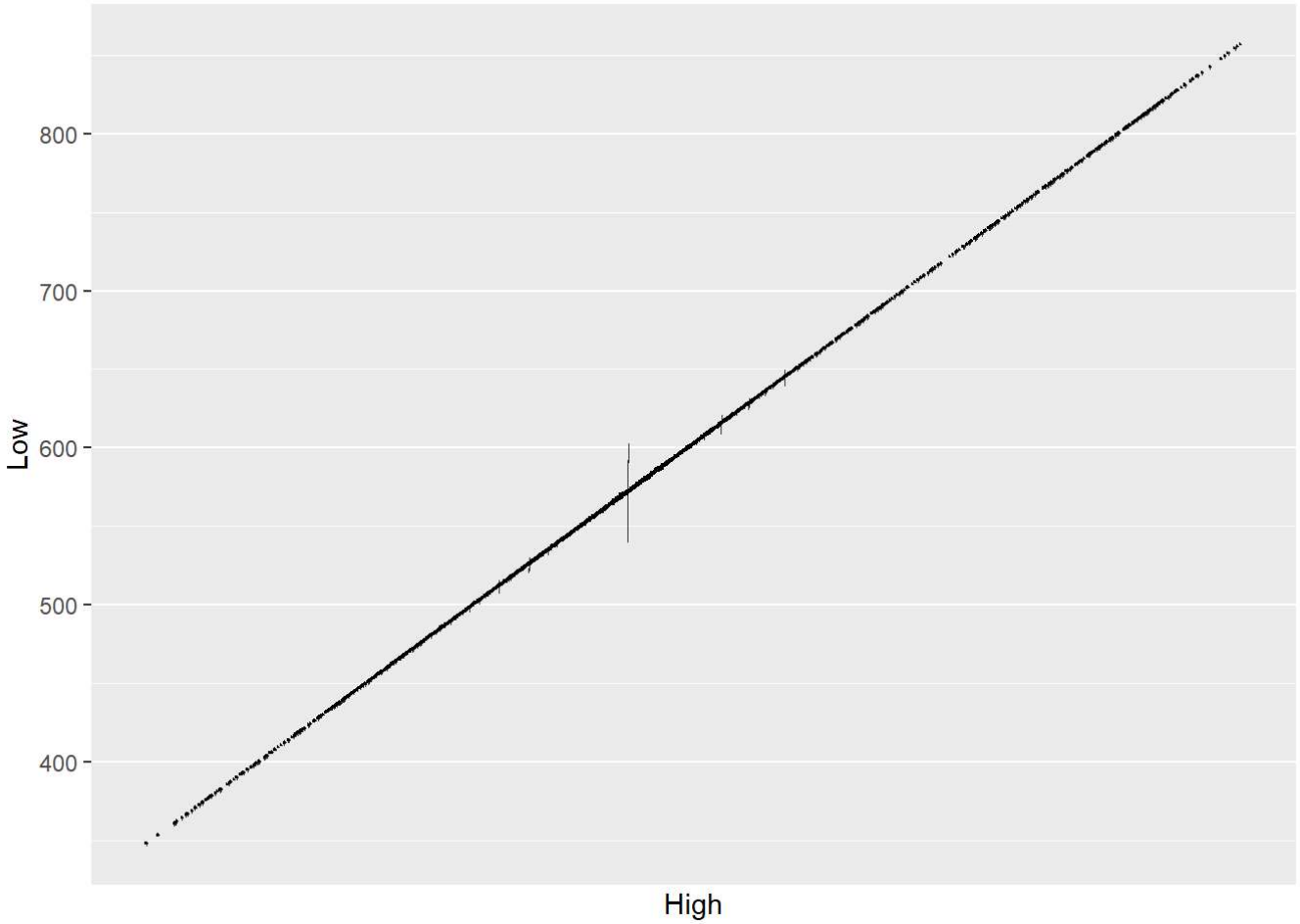

```
ggplot(df, aes(x = Low, y = High)) +
geom_boxplot(outlier.colour = NA, width = .4) +
geom_dotplot(binaxis = "y", binwidth = .25, stackdir = "center", fill = NA)
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Dot plot next to box plot

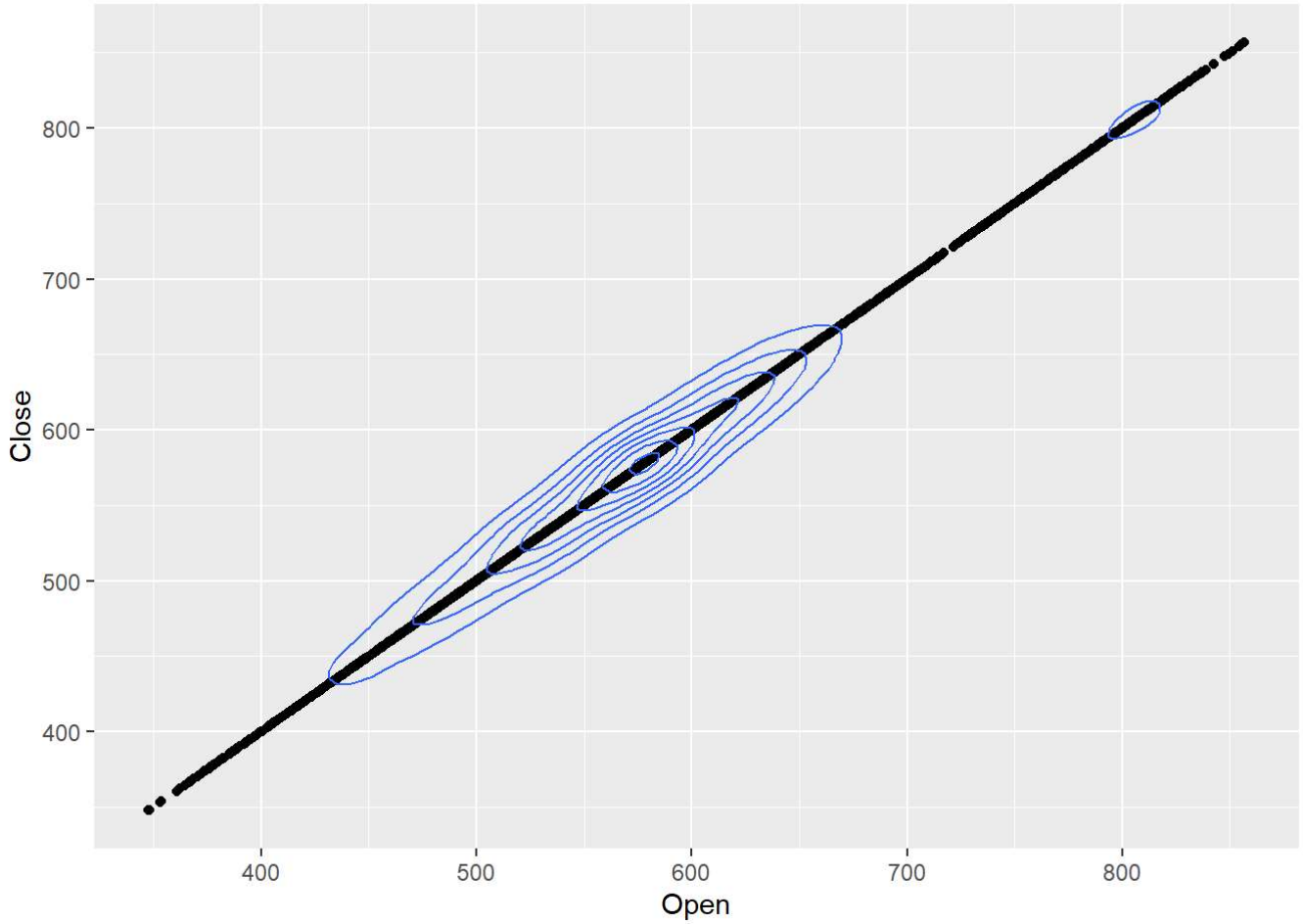
```
ggplot(df, aes(x = High, y = Low)) +
geom_boxplot(aes(x = as.numeric(High) + .2, group = High), width = .25) +geom_dotplot(aes(x =
as.numeric(High) - .2, group = High),binaxis = "y",binwidth = .5,stackdir = "center") +scale_
x_continuous(breaks = 1:nlevels(df$pretest),labels=levels(df$KDA)
)
```



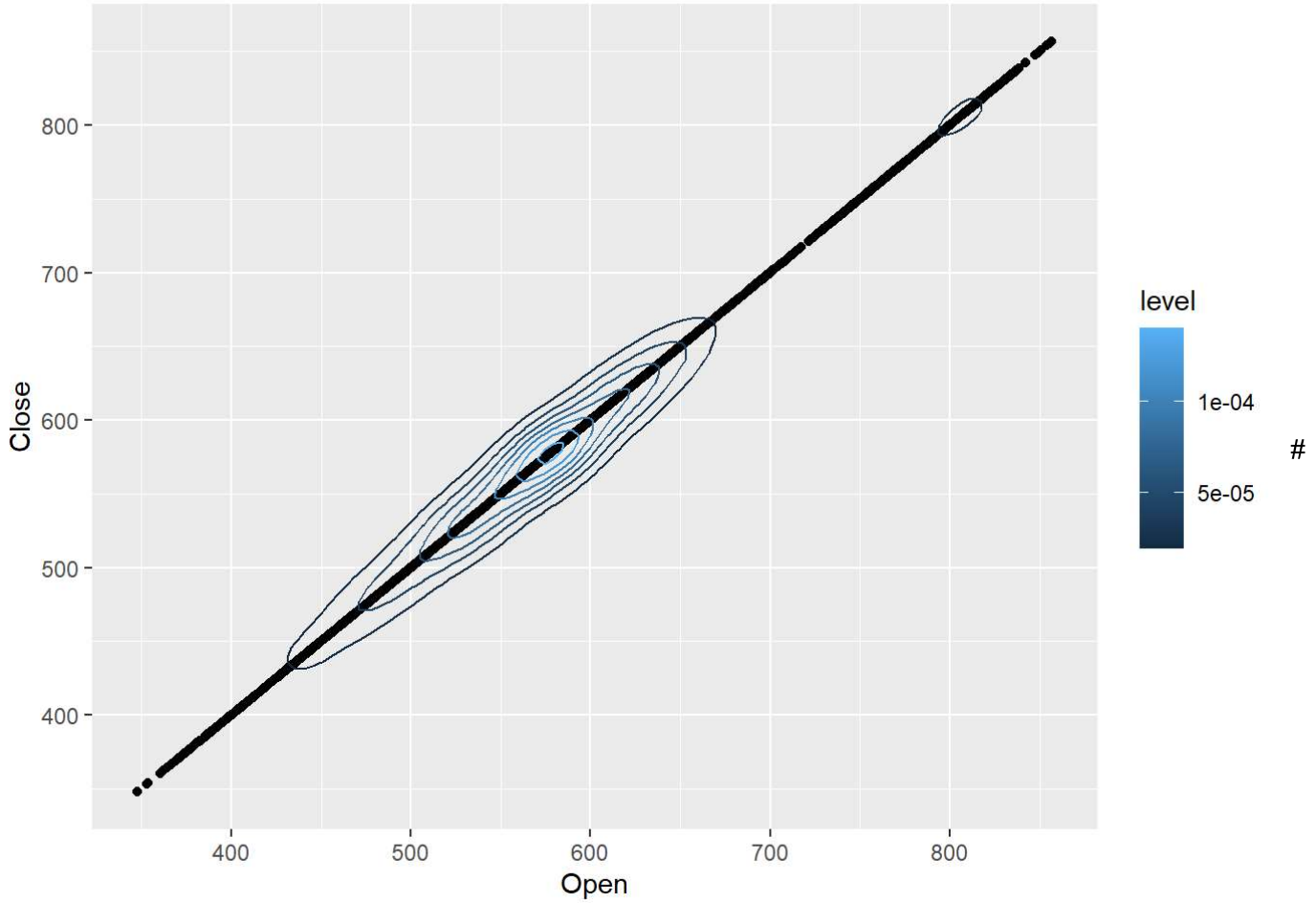
#

Making a Density Plot of Two-Dimensional Data

```
# Save a base plot object
 ggplot(df, aes(x = Open, y = Close))+geom_point() +stat_density2d()
```

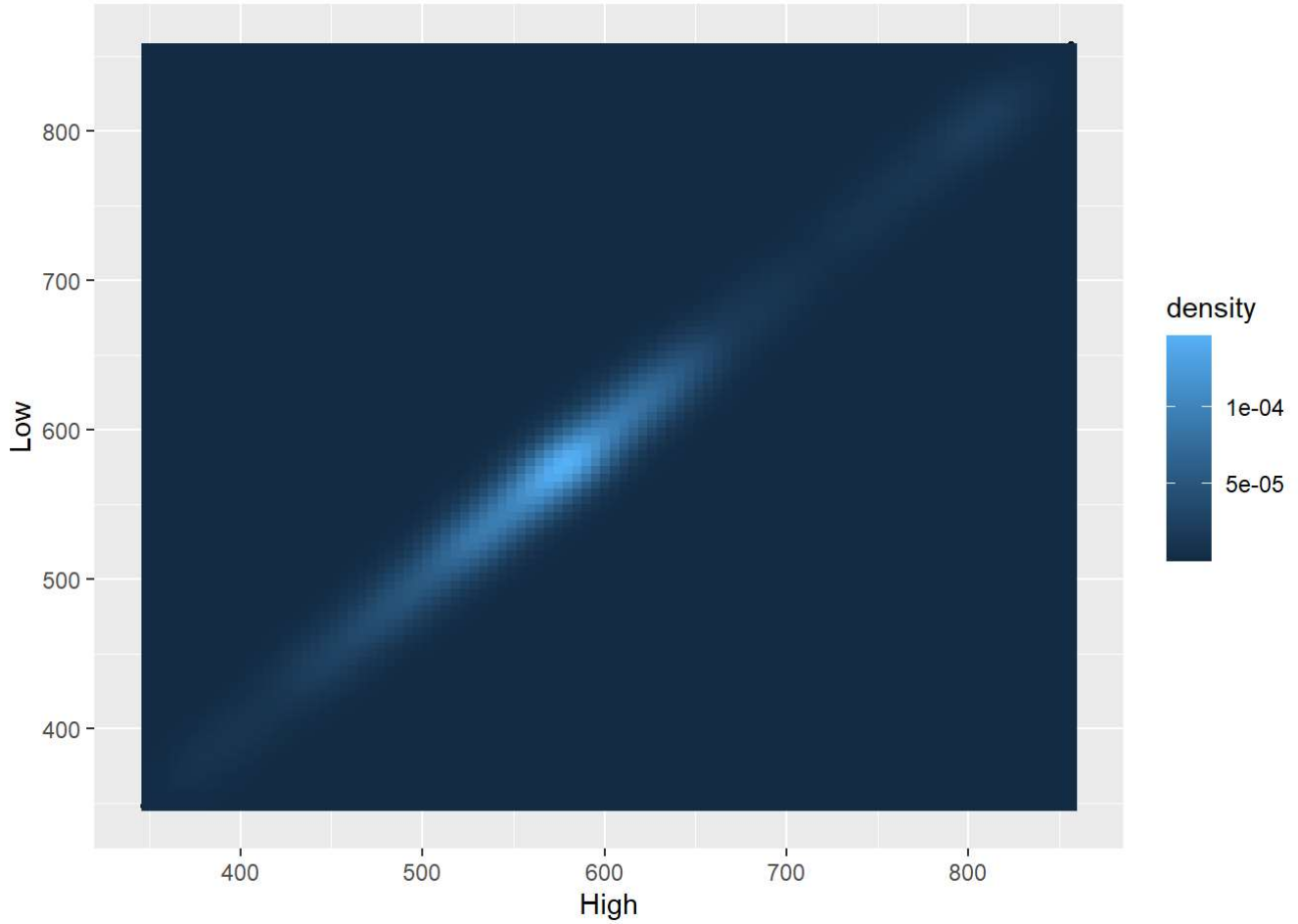



```
# Contour lines, with "height" mapped to color
ggplot(df, aes(x = Open, y = Close))+geom_point() +stat_density2d()+stat_density2d(aes(colour
= ..level..))
```

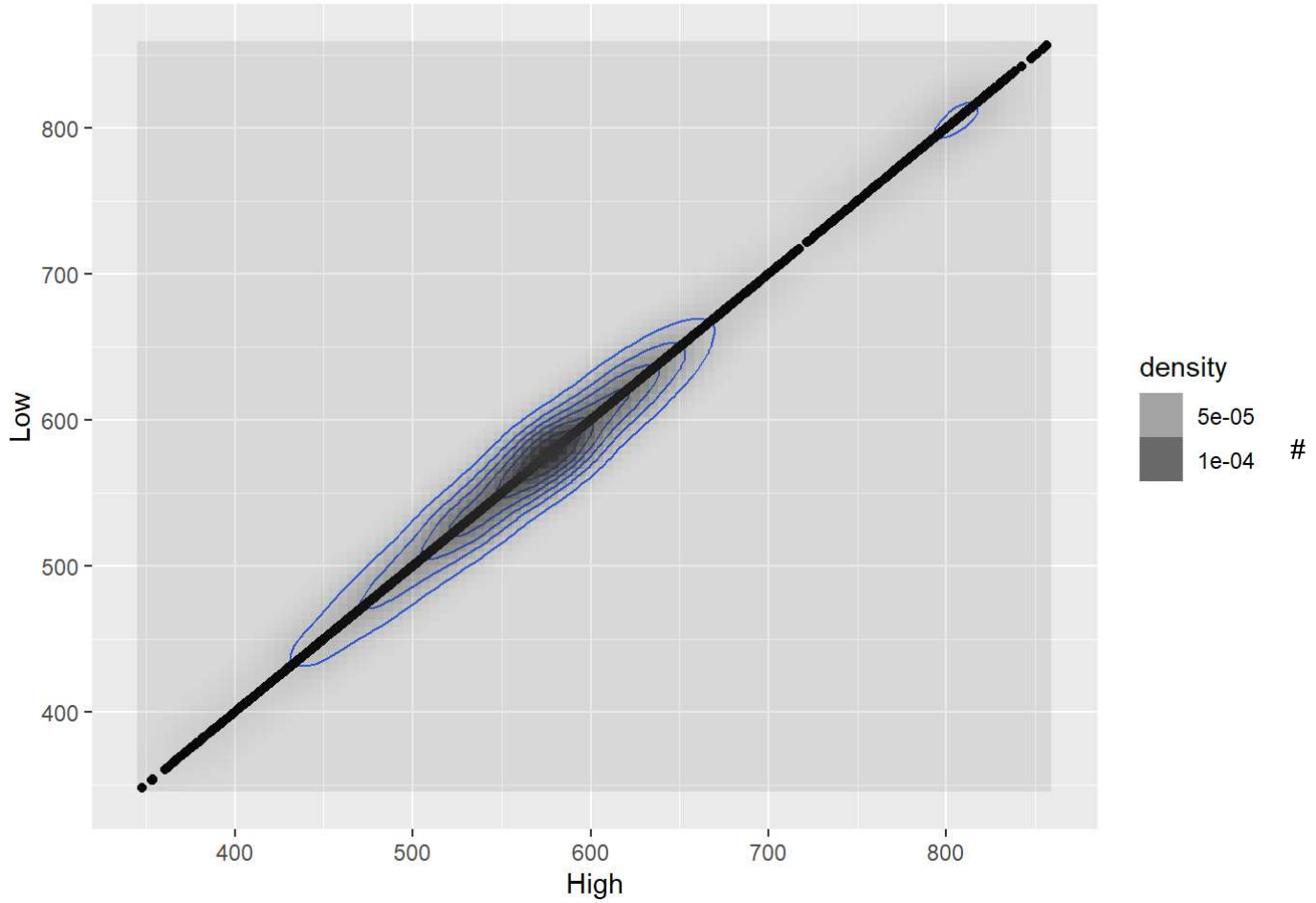


With ..density.. mapped to fill (1) # With points, and ..density.. mapped to alpha(2)

```
# Map density estimate to fill color
ggplot(df, aes(x = High, y = Low))+geom_point() +stat_density2d()+stat_density2d(aes(fill =
 ..density..), geom = "raster", contour = FALSE)
```

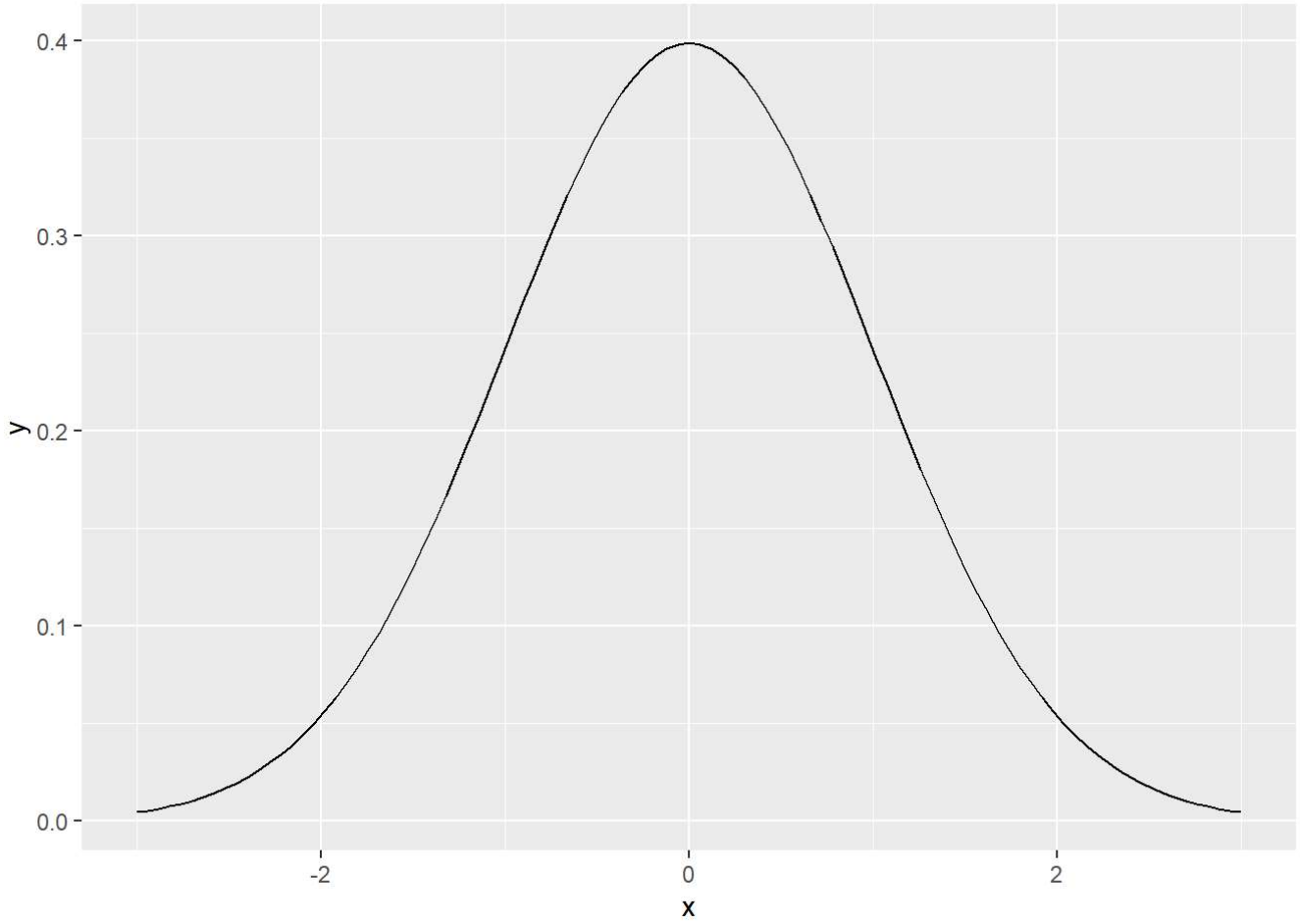



```
# With points, and map density estimate to alpha
ggplot(df, aes(x = High, y = Low))+geom_point() +stat_density2d()+geom_point() +stat_density2
d(aes(alpha = ..density..), geom = "tile", contour = FALSE)
```



Plotting a Function # # The data frame is only used for setting the range

```
# The normal distribution
ggplot(data.frame(x = c(-3, 3)), aes(x = x))  + stat_function(fun = dnorm)
```




```
# The t-distribution with df=2
ggplot(data.frame(x = c(-3, 3)), aes(x = x)) + stat_function(fun = dt, args = list(df = 2))
```