EDA LA-2

TITILE: GoodReads_100k_books

Deepika Reddy – 1NT19IS104

Jayanth A – 1NT19IS064

loading dataset

```
data<-
read.csv("C:/Users/JAYANTH/Desktop/GoodReads_100k_books.csv",header=TRUE,sep=
',')
```

loading the ggplot2 for using various graph functions

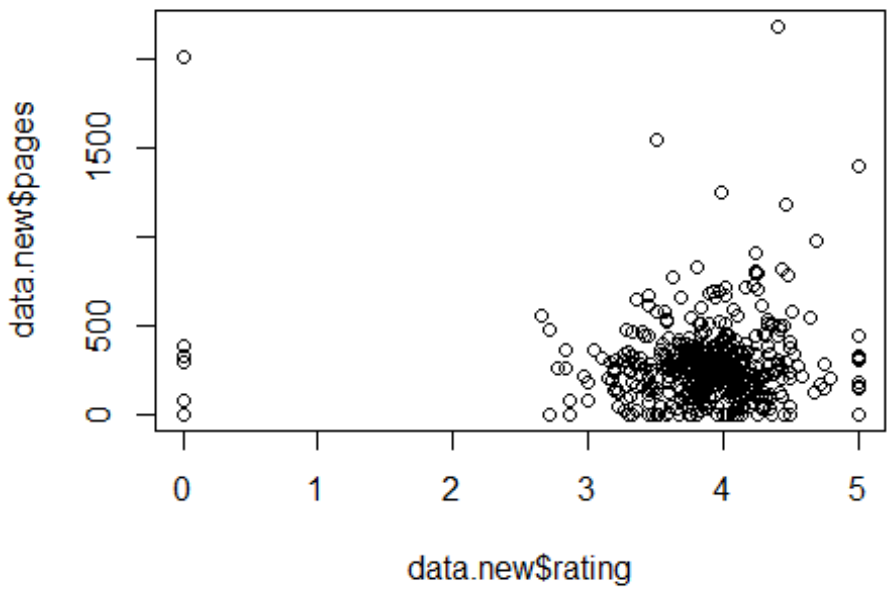
```
library(ggplot2)
```

Creating a Scatter Plot

To make a scatter plot , use plot() and pass it a vector of x values fol- lowed by a vector of y values

The
data.new
$rating returns the column named rating from the data.new data frame, and data.new$pages is the pages column

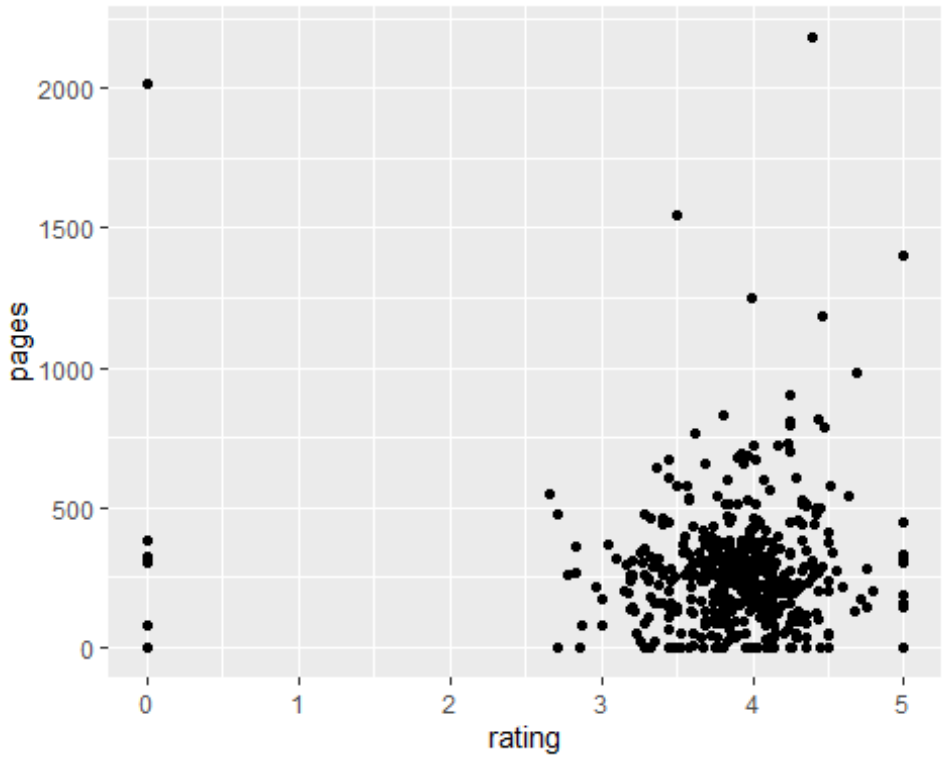
```
data.new <- data[1:500,]
plot(data.new$rating, data.new$pages)
```



With ggplot2, you can get a similar result using the ggplot() function

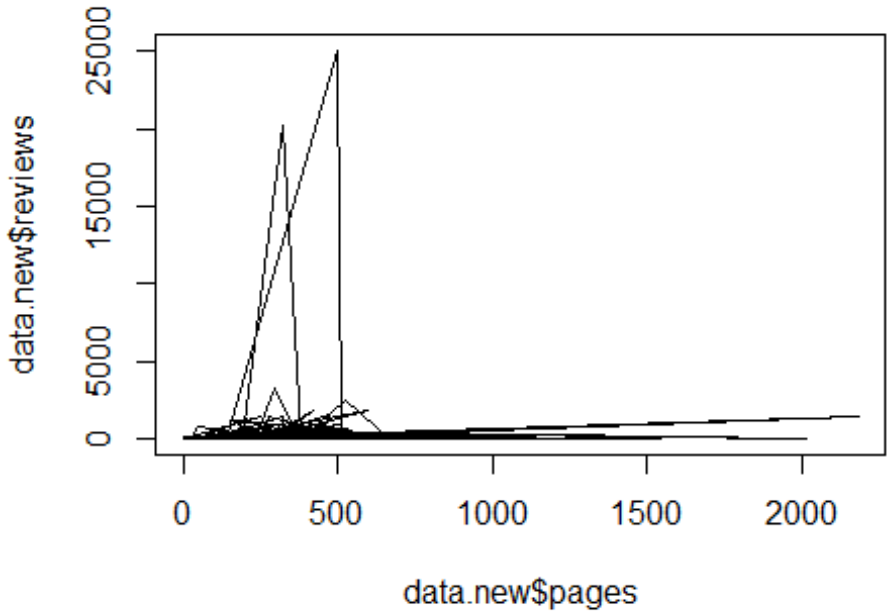
The first part, ggplot(), tells it to create a plot object, and the second part, geom_point(), tells it to add a layer of points to the plot.

```
ggplot(data.new, aes(x = rating, y = pages)) +
 geom_point()
```



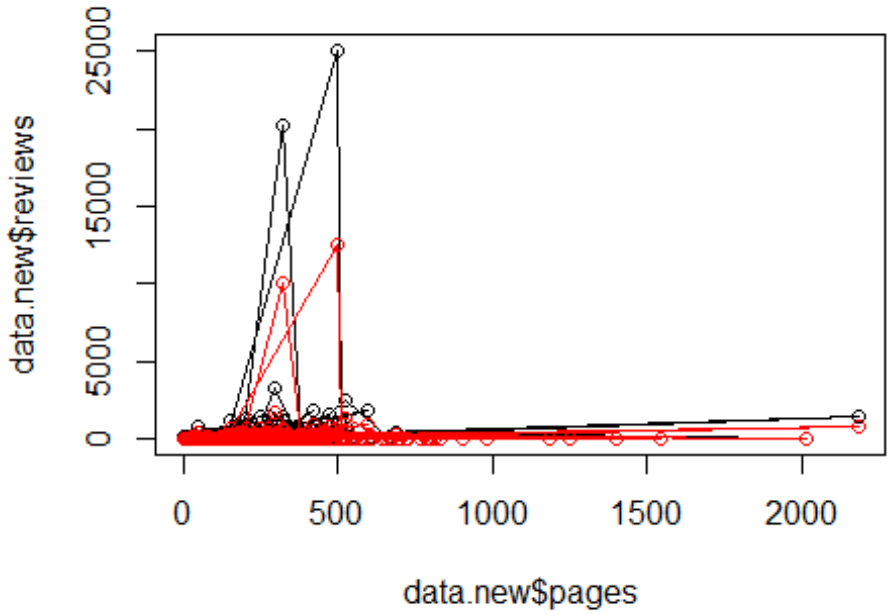
To make a line graph using plot() , pass it a vector of x values and a vector of y values, and use type = “l”:

```
plot(data.new$pages, data.new$reviews,type = "l")
```



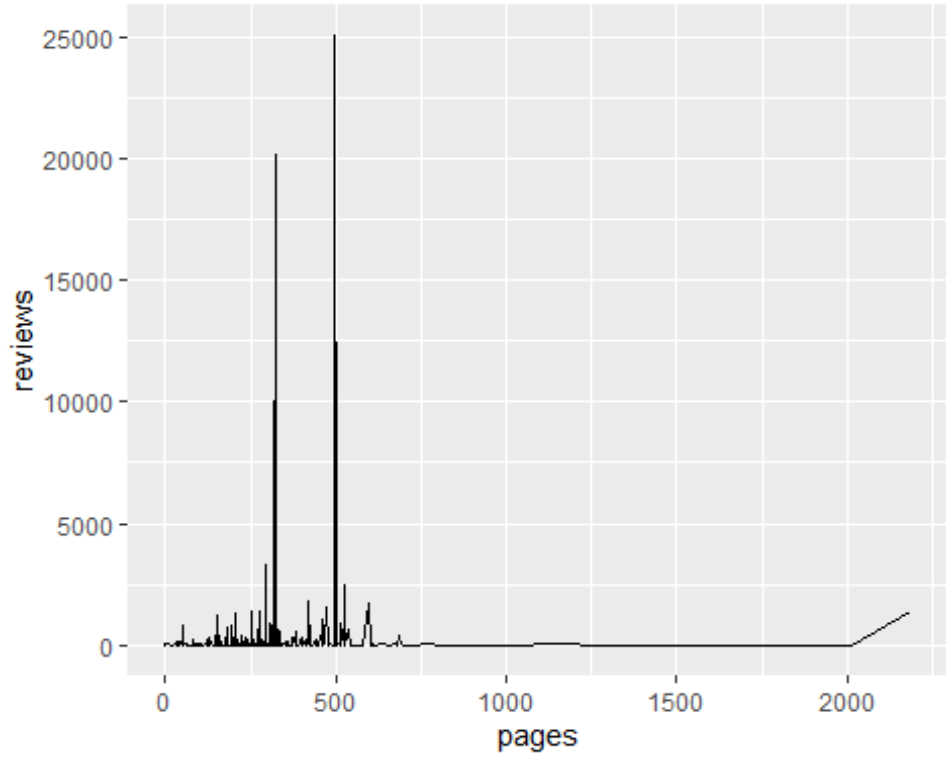
To add points and/or multiple lines , first call plot() for the first line, then add points with points() and additional lines with lines():

```
plot(data.new$pages, data.new$reviews, type = "l")
points(data.new$pages, data.new$reviews)
lines(data.new$pages, data.new$reviews/2, col = "red")
points(data.new$pages, data.new$reviews/2, col = "red")
```



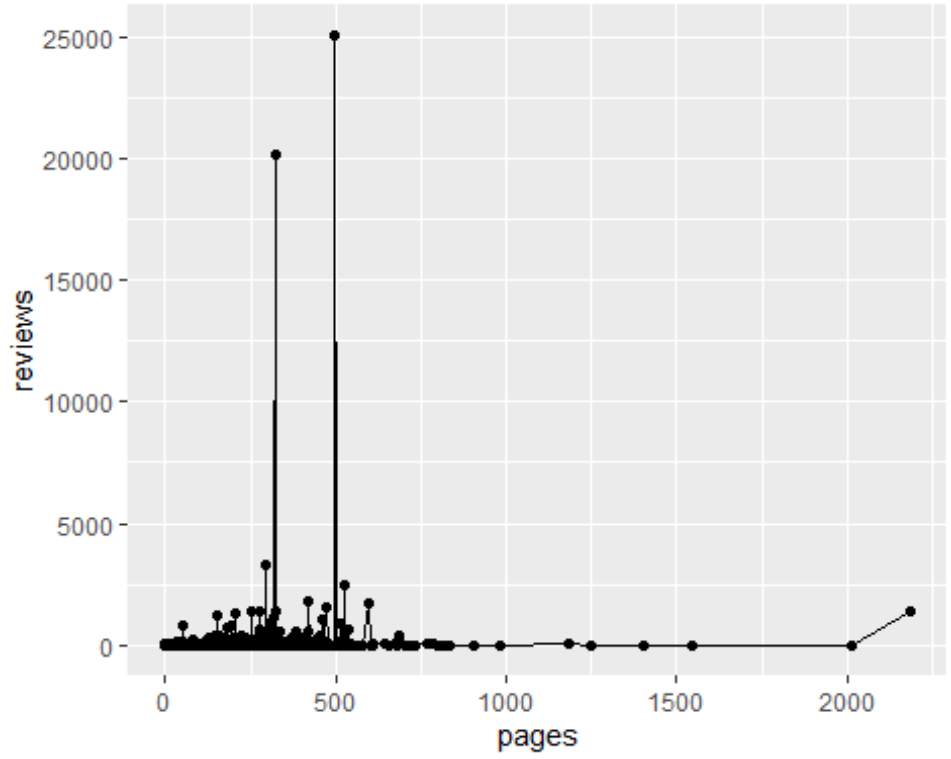
With ggplot2, you can get a similar result using geom_line()

```
ggplot(data.new, aes(x = pages, y = reviews)) +
 geom_line()
```



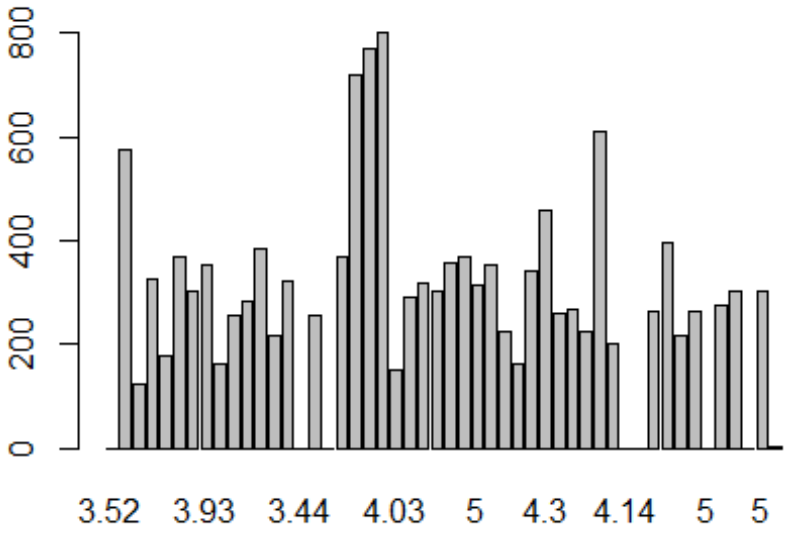
With points added for line graph using geom_point

```
ggplot(data.new, aes(x = pages, y = reviews)) +
 geom_line()+
  geom_point()
```



To make a bar graph of values , use barplot() and pass it a vector of values for the height of each bar and (optionally) a vector of labels for each bar. If the vector has names for the elements, the names will automatically be used as labels:

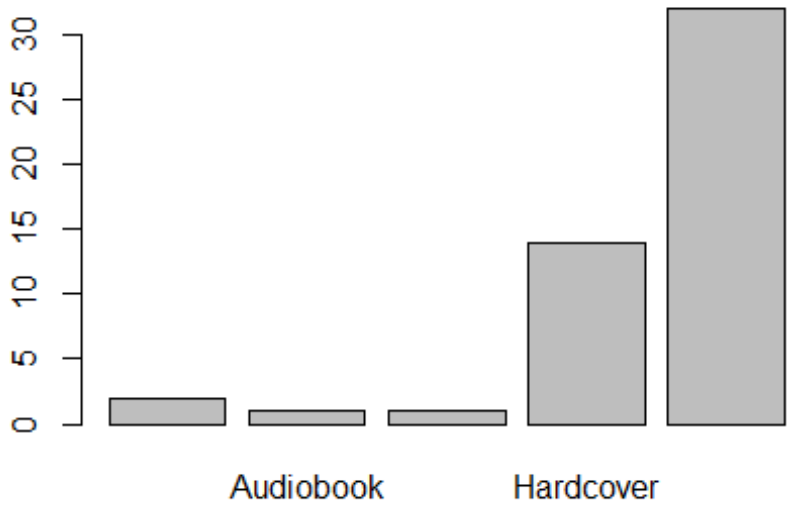
```
data.new1 <- data[1:50, ]
barplot(data.new1$pages, names.arg = data.new1$rating)
```



To generate the count of each unique value in a vector, use the table() function: Then pass the table to barplot() to generate the graph of counts

Gives the count of each group of bookformat

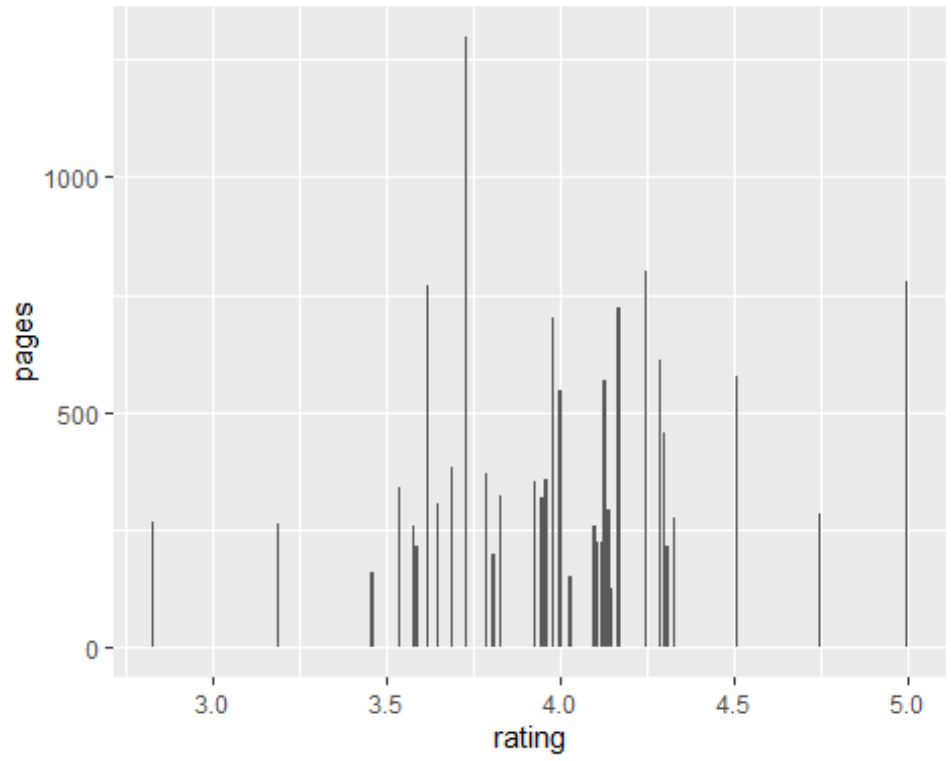
```
barplot(table(data.new1$bookformat))
```



With ggplot2, you can get a similar result using geom_col() (Figure 2-6). To plot a bar graph of values, use geom_col()

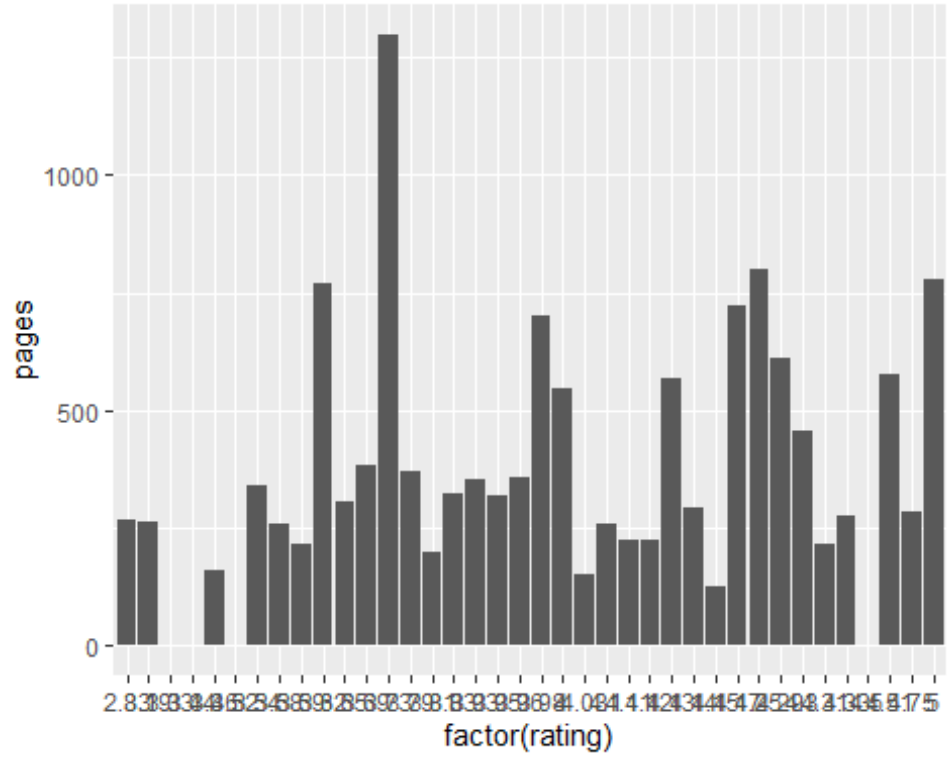
This uses the data.new1 data frame, with the “rating” column for x values and the “pages” column for y values.

```
ggplot(data.new1, aes(x = rating, y = pages)) +
 geom_col()
```



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

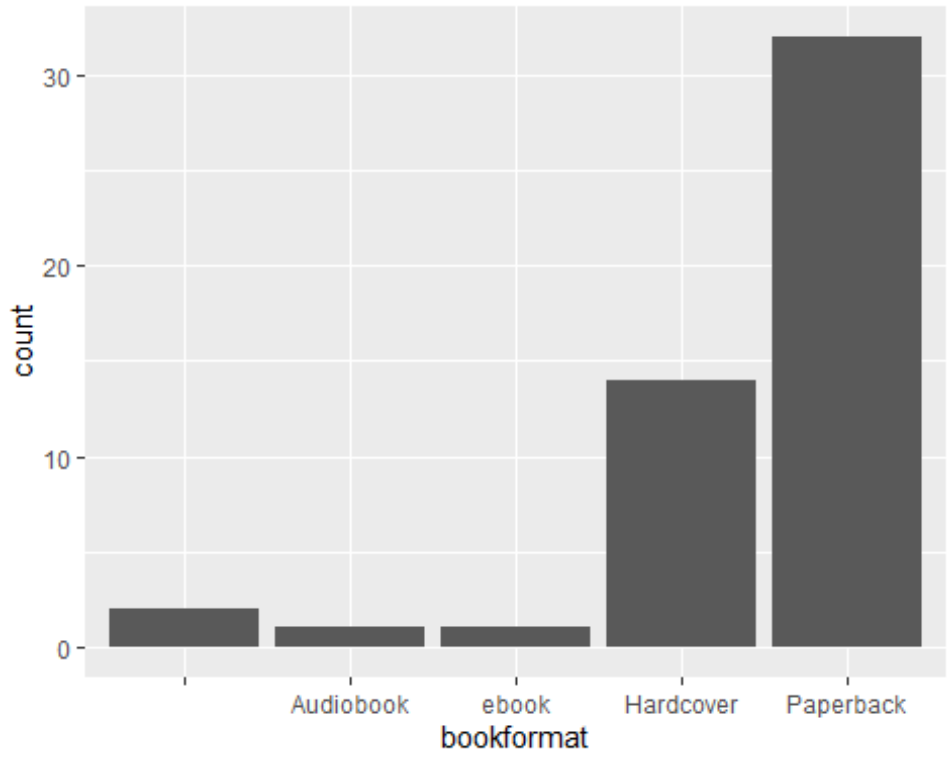
```
ggplot(data.new1, aes(x = factor(rating), y = pages)) +
 geom_col()
```



ggplot2 can also be used to plot the count of the number of data rows in each category , by using geom_bar() instead of geom_col().

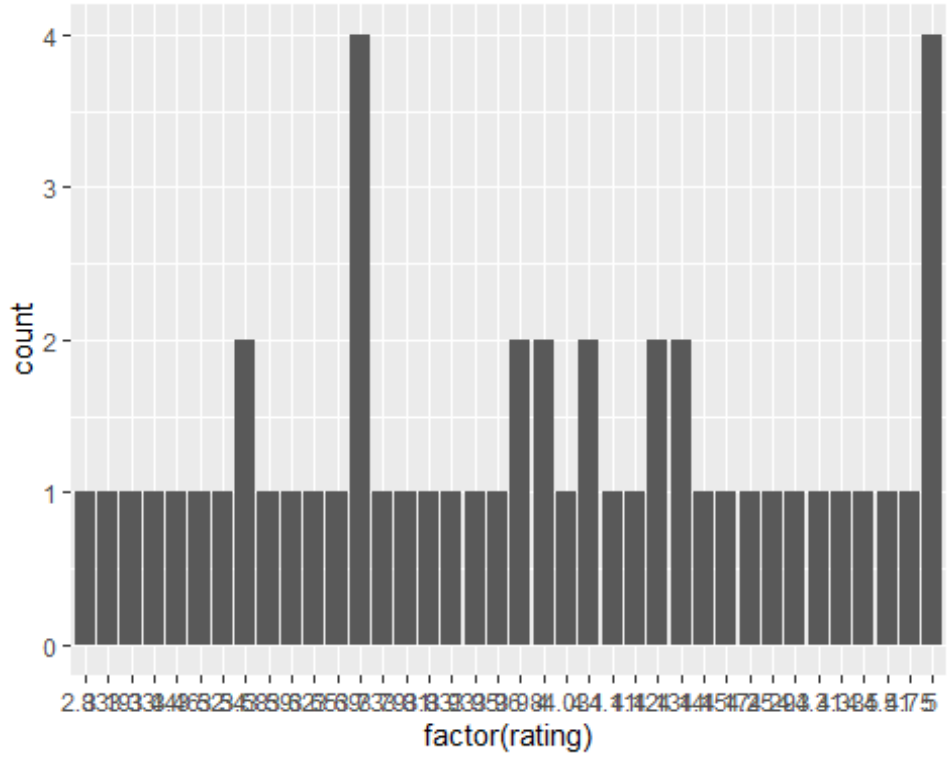
Bar graph of counts. This uses the data.new1 data frame, with the "bookformat" column for x position. The y position is calculated by counting the number of rows for each value of bookformat.

```
ggplot(data.new1, aes(x = bookformat)) +
 geom_bar()
```



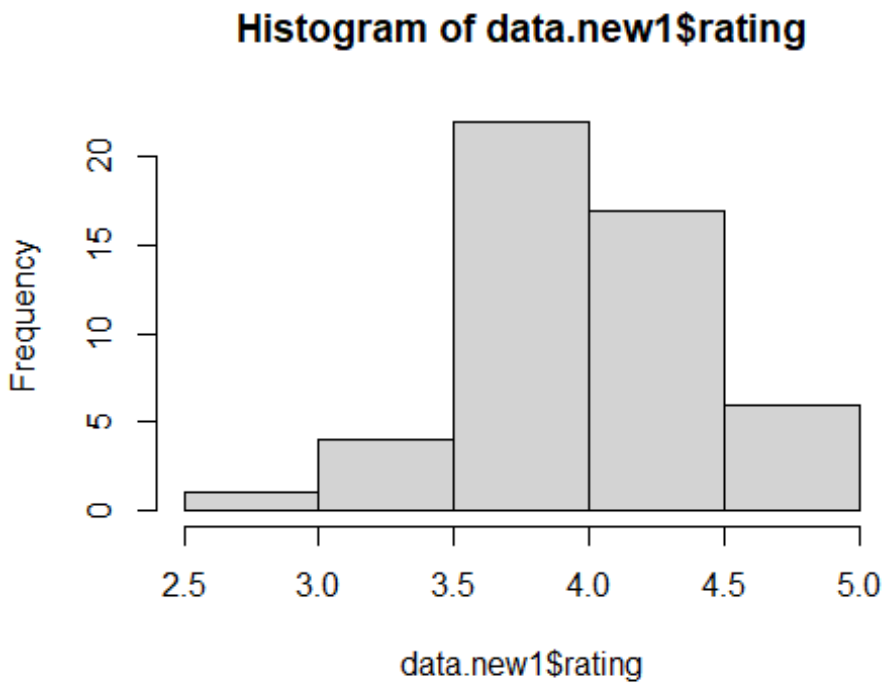
With x variable(here rating is used) converted to a factor

```
ggplot(data.new1, aes(x = factor(rating))) +
 geom_bar()
```



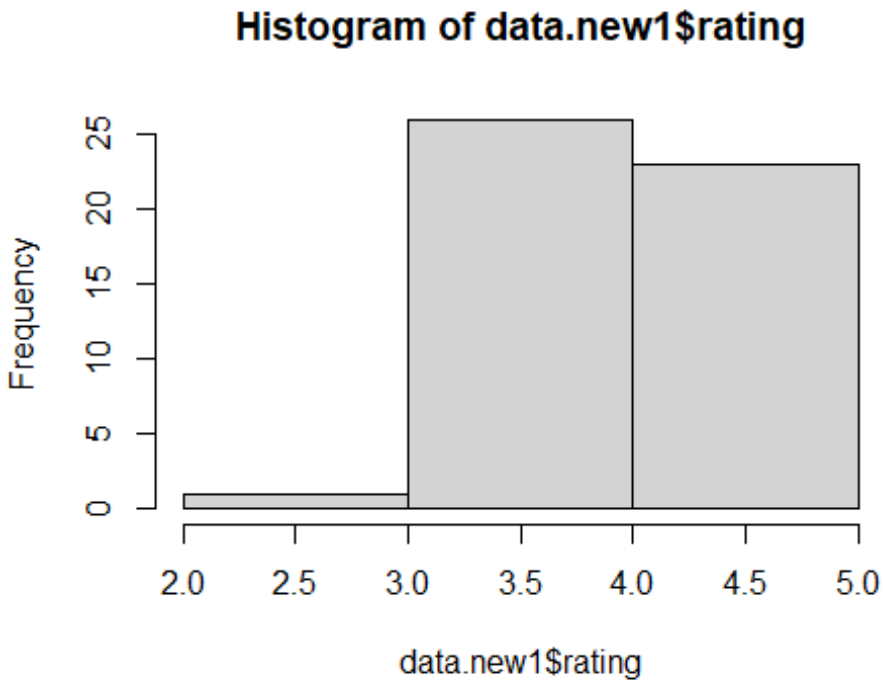
To make a histogram use hist() and pass it a vector of values

```
hist(data.new1$rating)
```



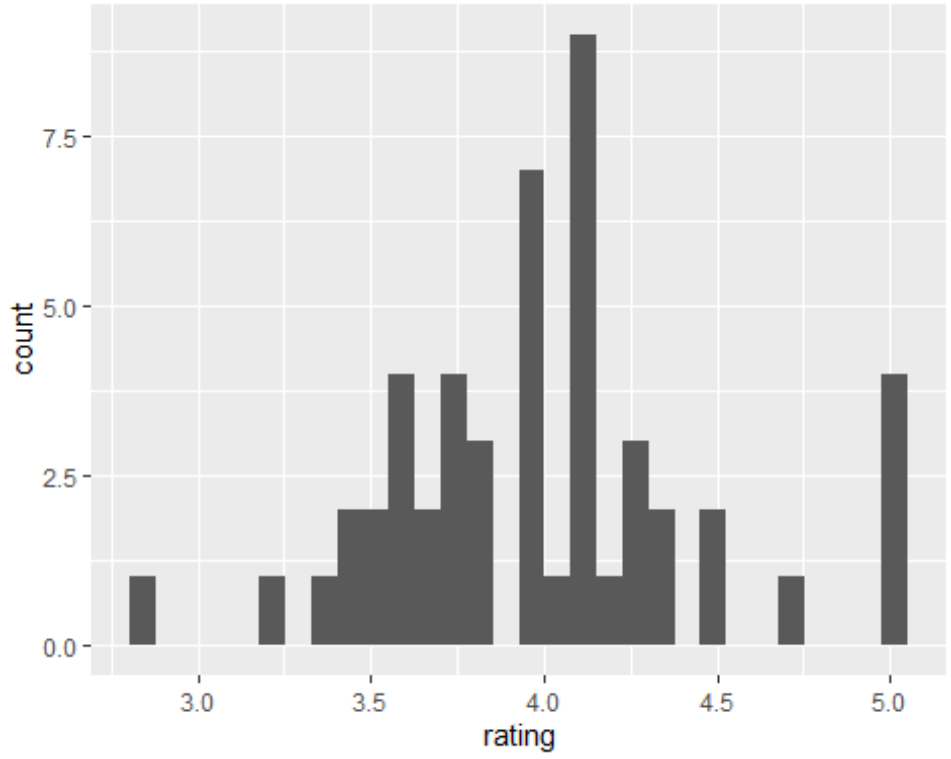
Specify approximate number of bins with breaks

```
hist(data.new1$rating, breaks = 2)
```



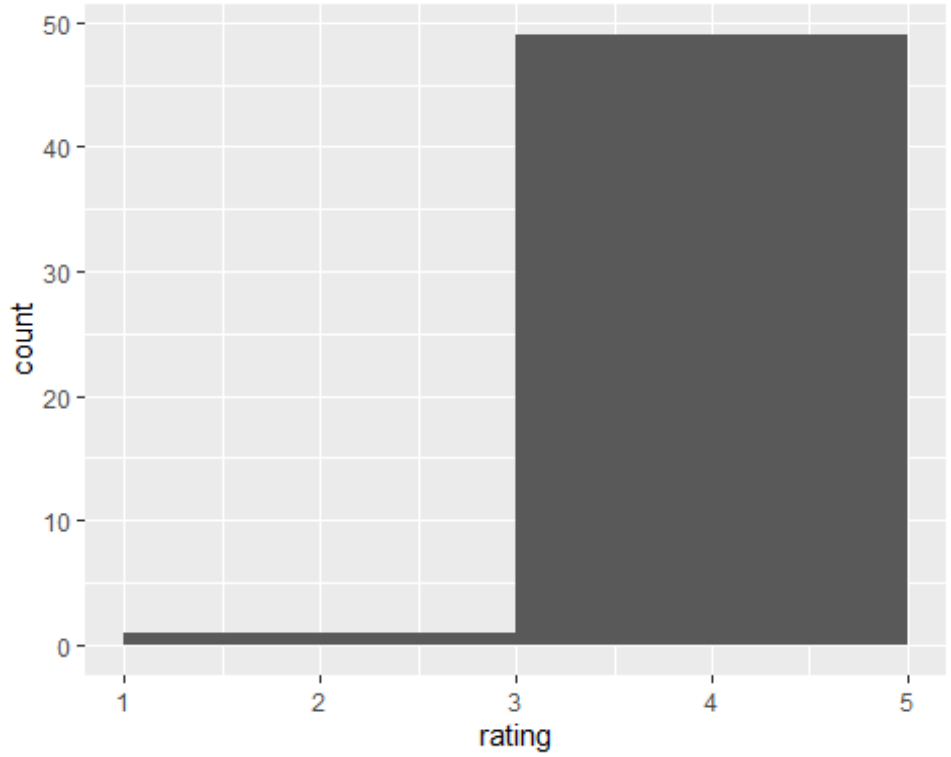
With the ggplot2, you can get a similar result using geom_histogram()

```
ggplot(data.new1, aes(x = rating)) +
 geom_histogram()

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



With wider bins

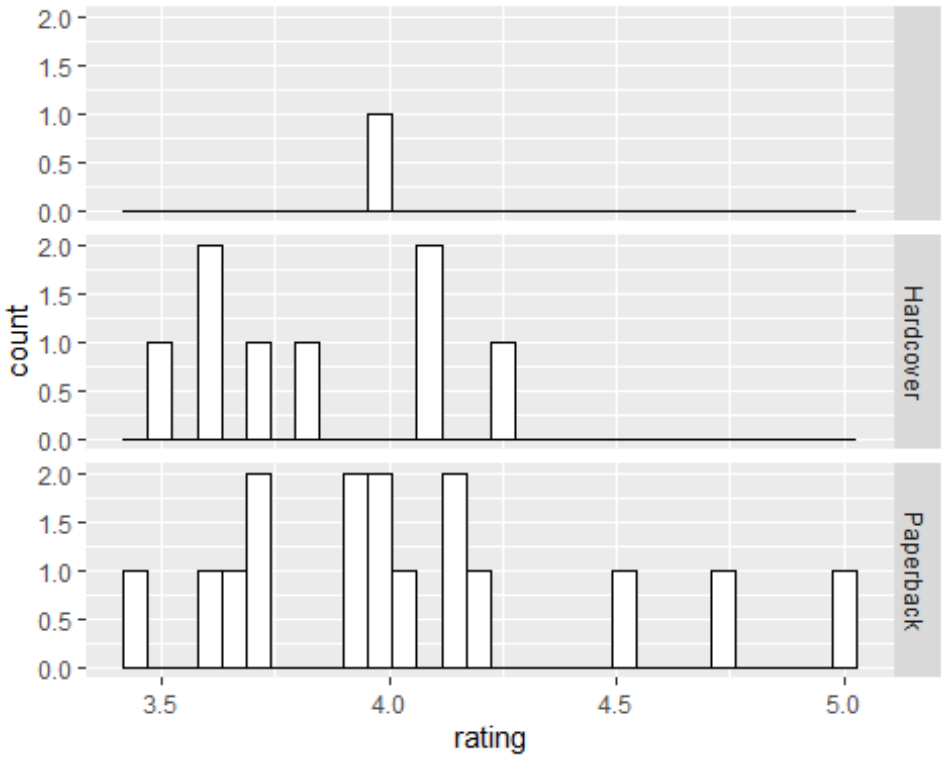
```
ggplot(data.new1, aes(x = rating)) +
 geom_histogram(binwidth = 2)
```



Making Multiple Histograms from Grouped Data

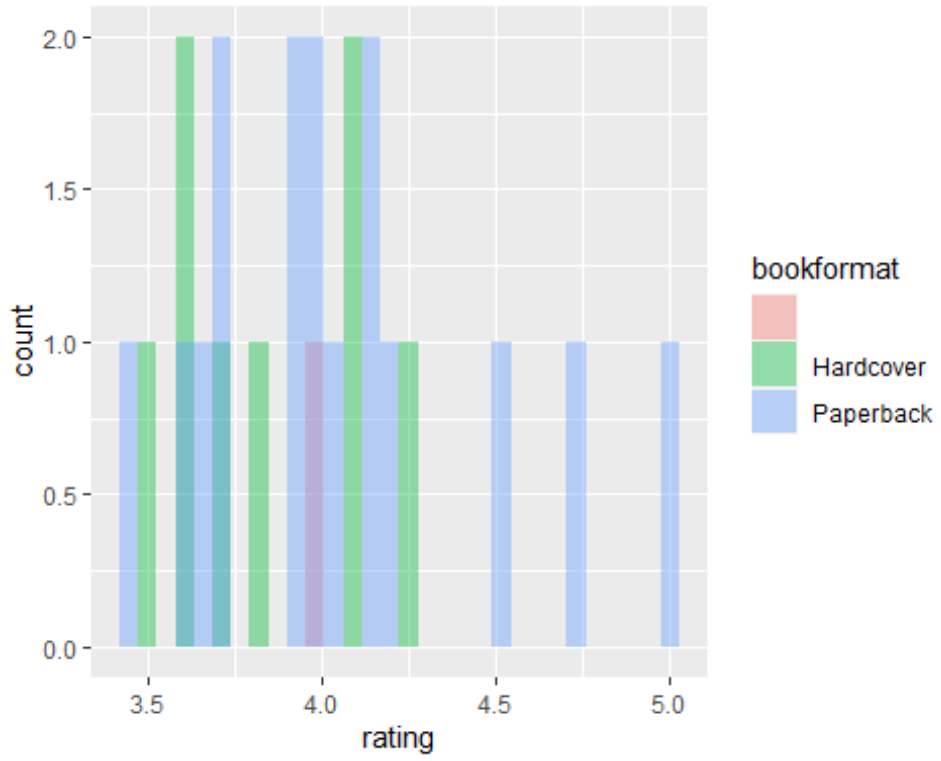
#method-1

```
 ggplot(data.new1[1:25,], aes(x = rating)) + geom_histogram(fill = "white",
colour = "black") +
 facet_grid( bookformat~ .)

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



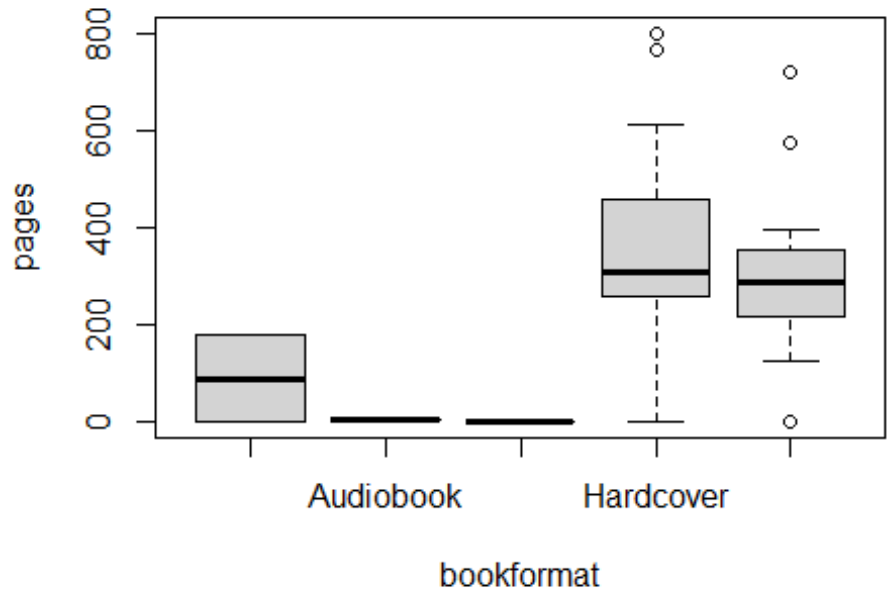
#method-2

```
ggplot(data.new1[1:25,], aes(x = rating, fill = bookformat))+
  geom_histogram(position = "identity", alpha = 0.4)

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



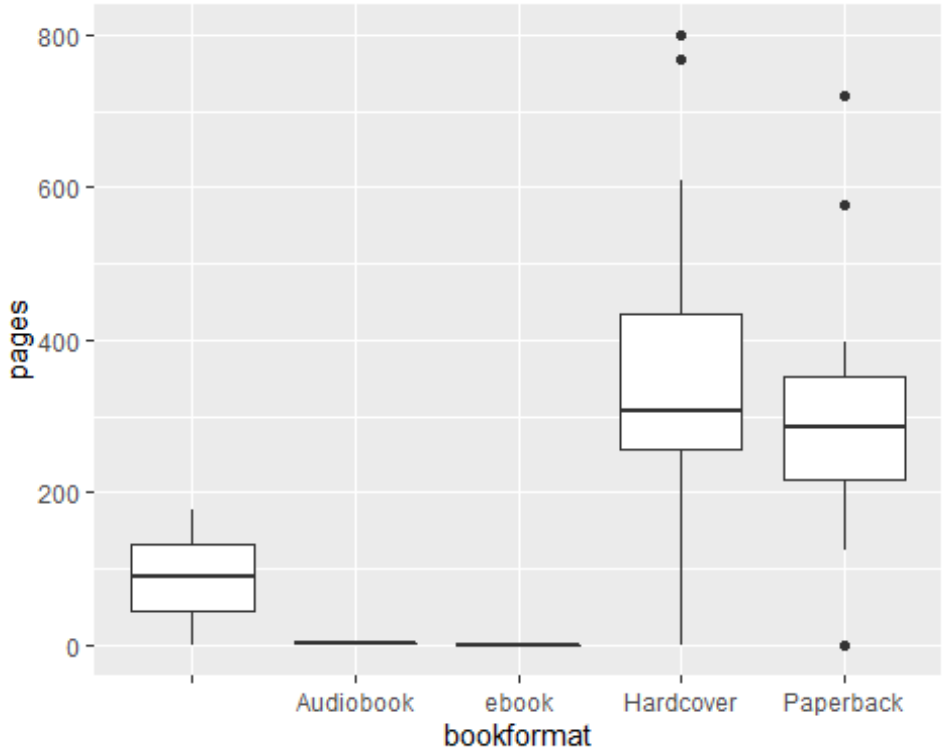
to create a box plot for comparing distributions. use the boxplot() function with formula syntax

```
boxplot(pages ~ bookformat, data = data.new1)
```



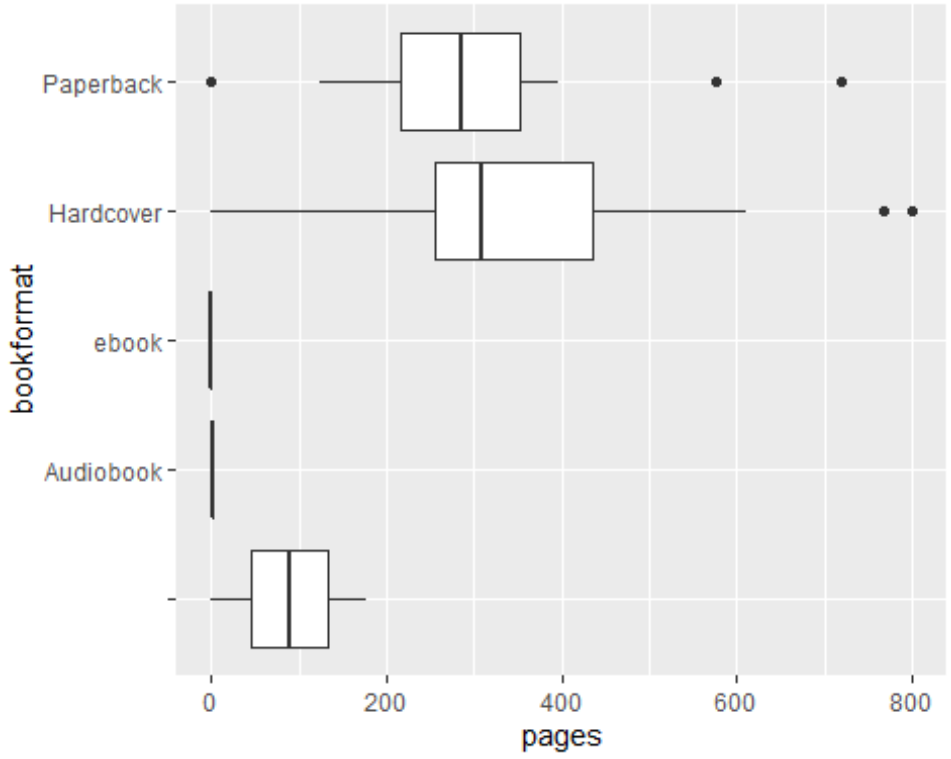
With the ggplot2 package, you can get a similar result with geom_boxplot():

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()
```



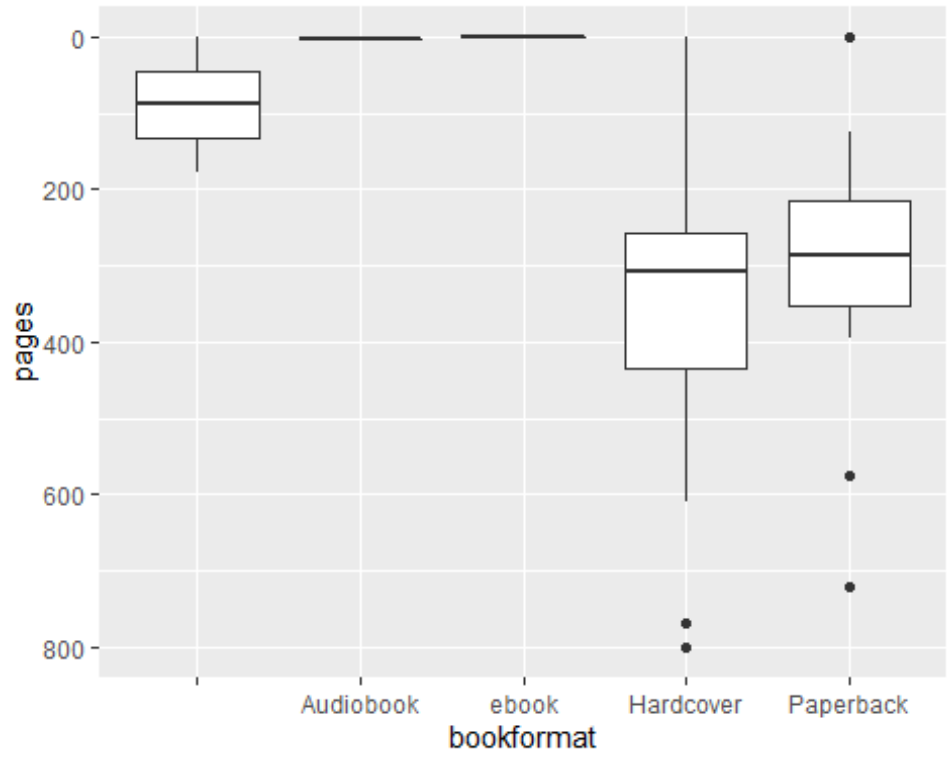
To swap the x- and y-axes on a graph. Solution Use coord_flip() to flip the axes

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()+
  coord_flip()
```



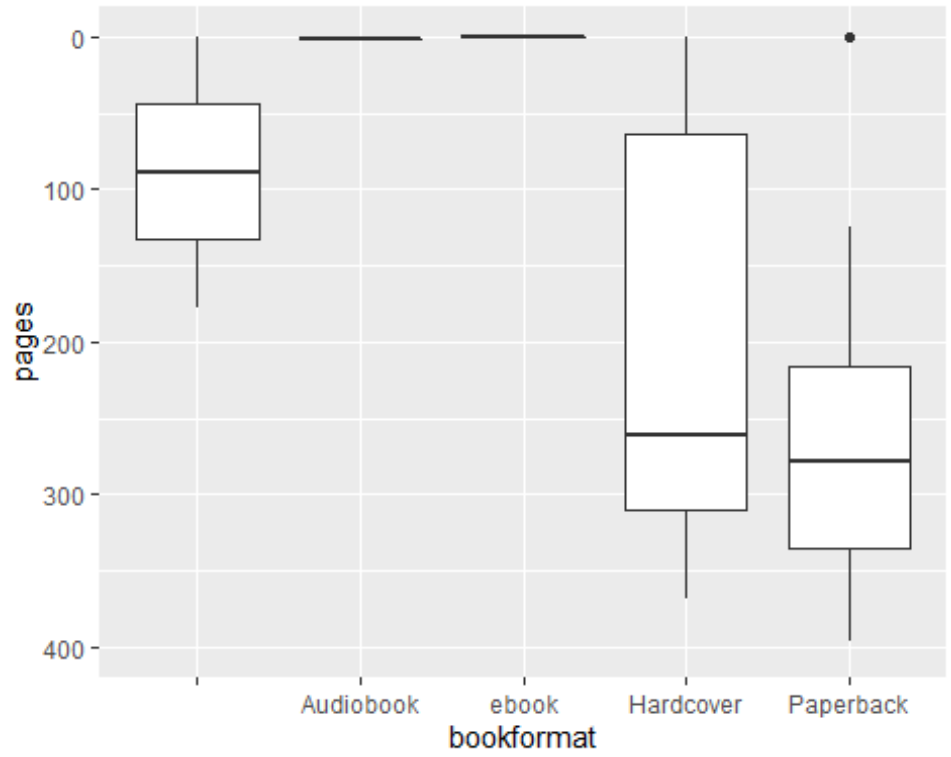
To reverse the direction of a continuous axis. Solution Use scale_y_reverse() or scale_x_reverse(). The direction of an axis can also be reversed by specifying the limits in reversed order, with the maximum first, then the minimum:

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()+
  scale_y_reverse()
```



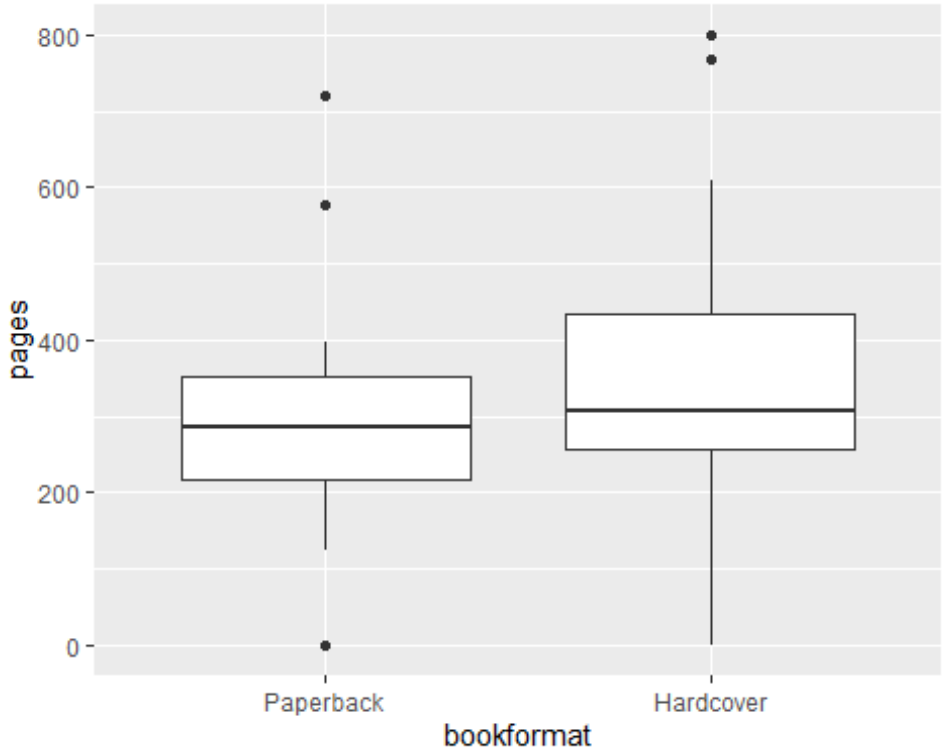
Similar effect by specifying limits in reversed order

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()+
  ylim(400,0)

## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```



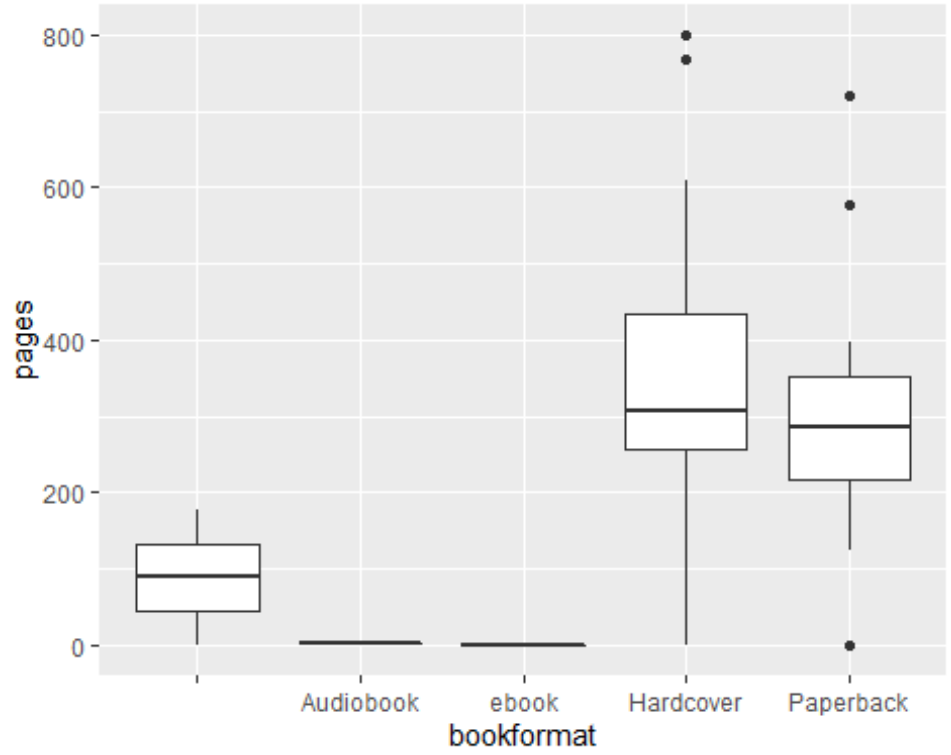
To change the order of items on a categorical axis and displaying subset of items on x-axis use scale_x_discrete() by setting limits inside it.

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()+
   scale_x_discrete(limits = c("Paperback", "Hardcover"))

## Warning: Removed 4 rows containing missing values (stat_boxplot).
```



Similarly to reverse the order, set limits = rev(levels(…)), and put the factor inside

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot()+
   scale_x_discrete(limits =rev(levels(data.new$bookformat)))
```

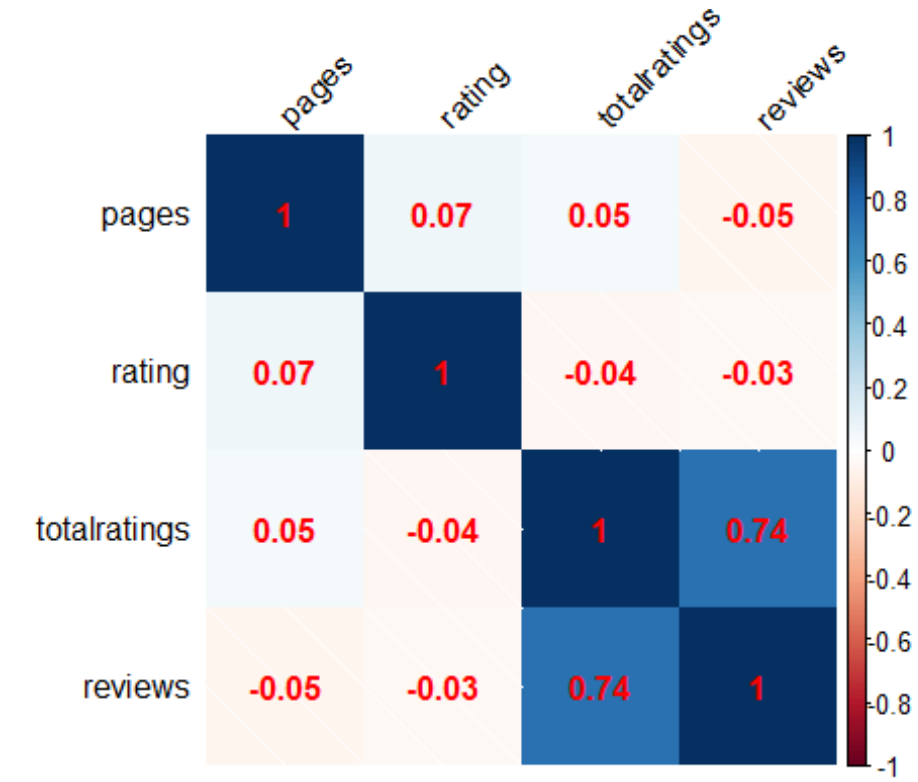


Plotting a correlation matrix

```
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.2.1

## corrplot 0.92 loaded

data.new2 <- subset(data.new1, select =
c("pages","rating","totalratings","reviews") )
mcor <- cor(data.new2)
round(mcor, digits = 2)

##              pages rating totalratings reviews
## pages         1.00   0.07         0.05   -0.05
## rating        0.07   1.00        -0.04   -0.03
## totalratings  0.05  -0.04         1.00    0.74
## reviews      -0.05  -0.03         0.74    1.00
```

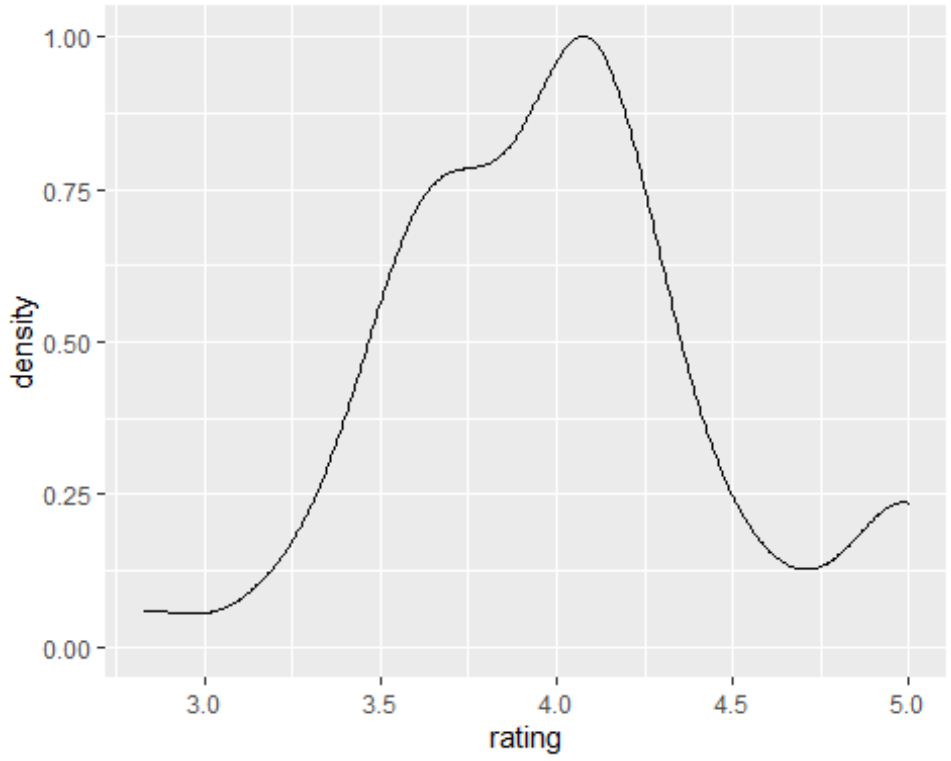
Plotting a correlation matrix

```
corrplot(mcor,tl.col="black",method="shade",tl.srt=45,addCoef.col = "red")
```



Making a density curve

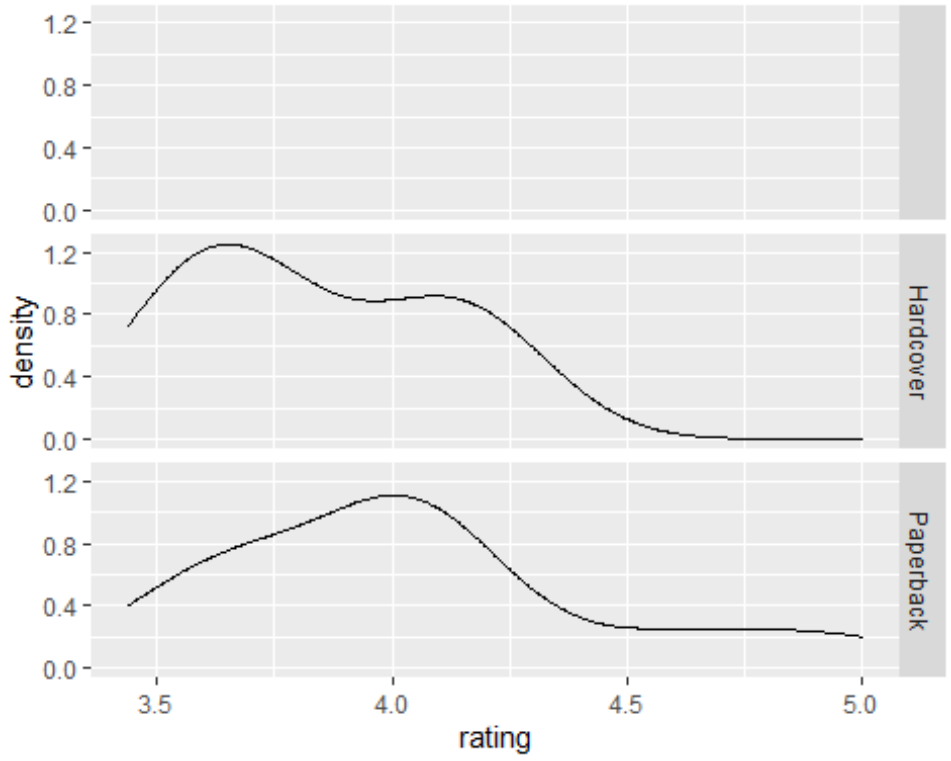
```
ggplot(data.new1, aes(x = rating)) + geom_density()
```



Making Multiple Density Curves from Grouped Data

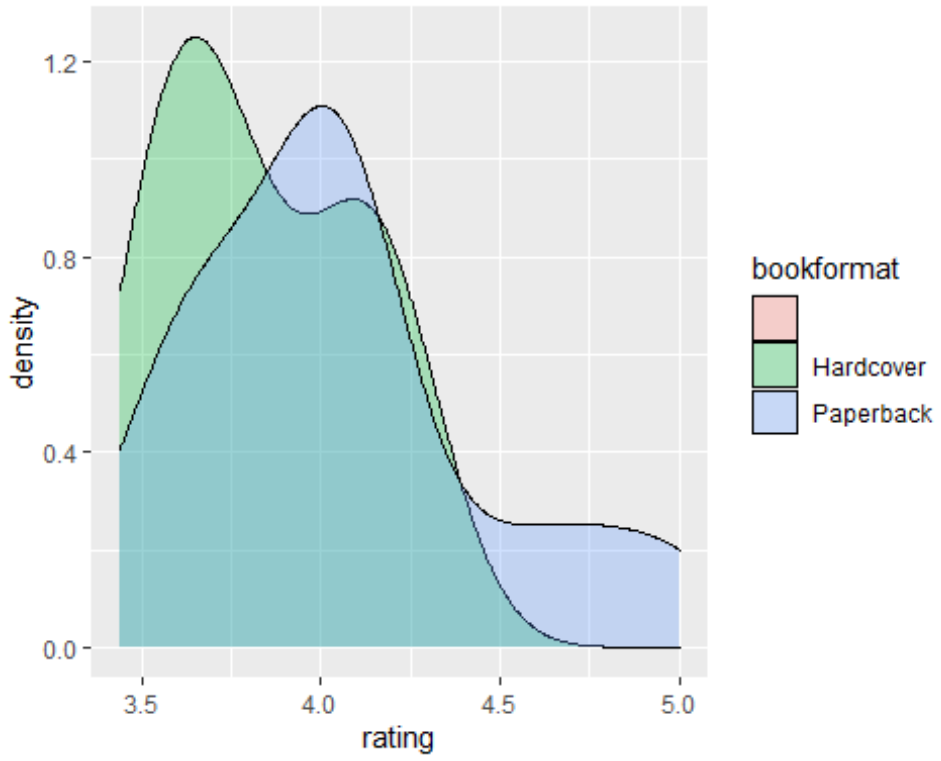
#method-1

```
ggplot(data.new1[1:25,], aes(x = rating)) + geom_density()+
  facet_grid(bookformat ~ .)

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max;
returning
## -Inf
```



#method-2

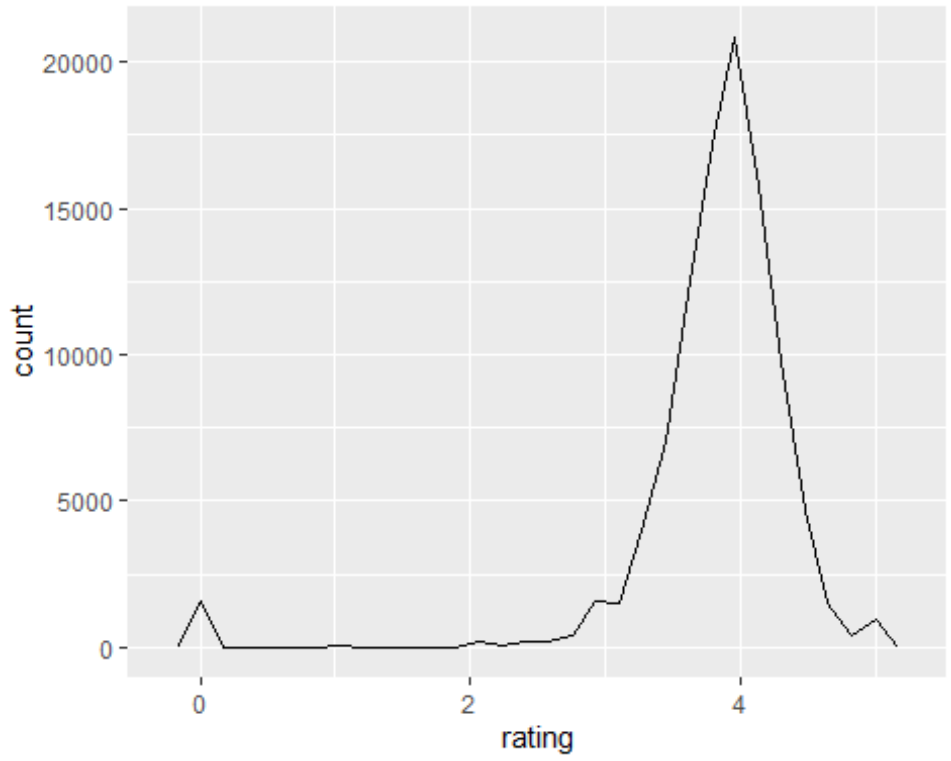
```
ggplot(data.new1[1:25,], aes(x = rating ,fill = bookformat))+
  geom_density(alpha = .3)

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max;
returning
## -Inf
```



Making a frequency polygon

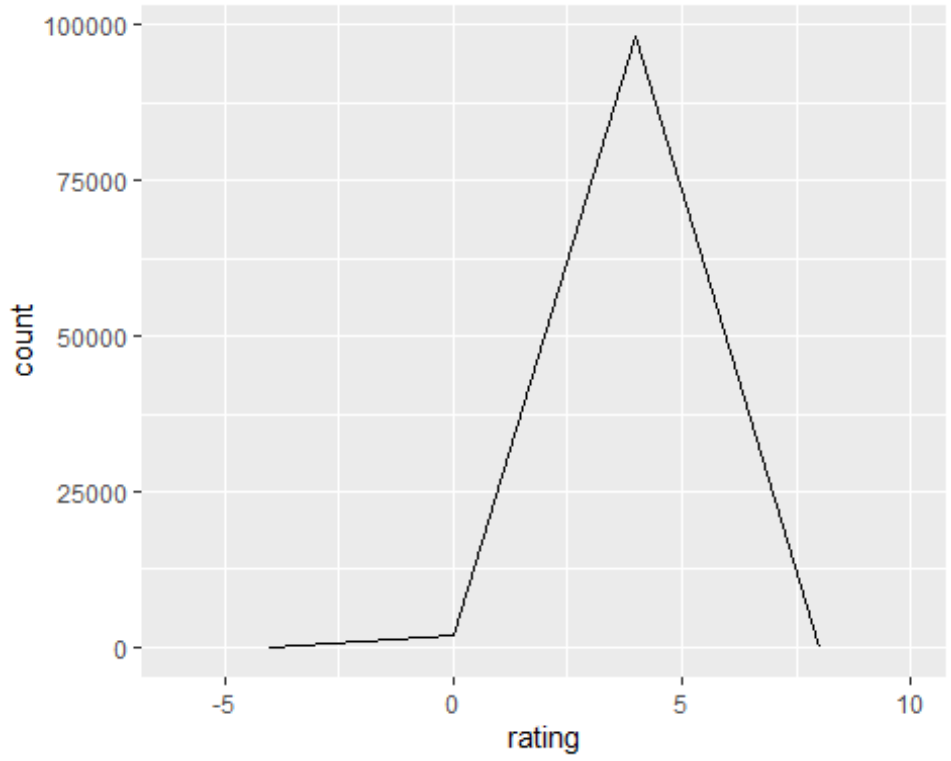
```
ggplot(data, aes(x=rating)) + geom_freqpoly()
```

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



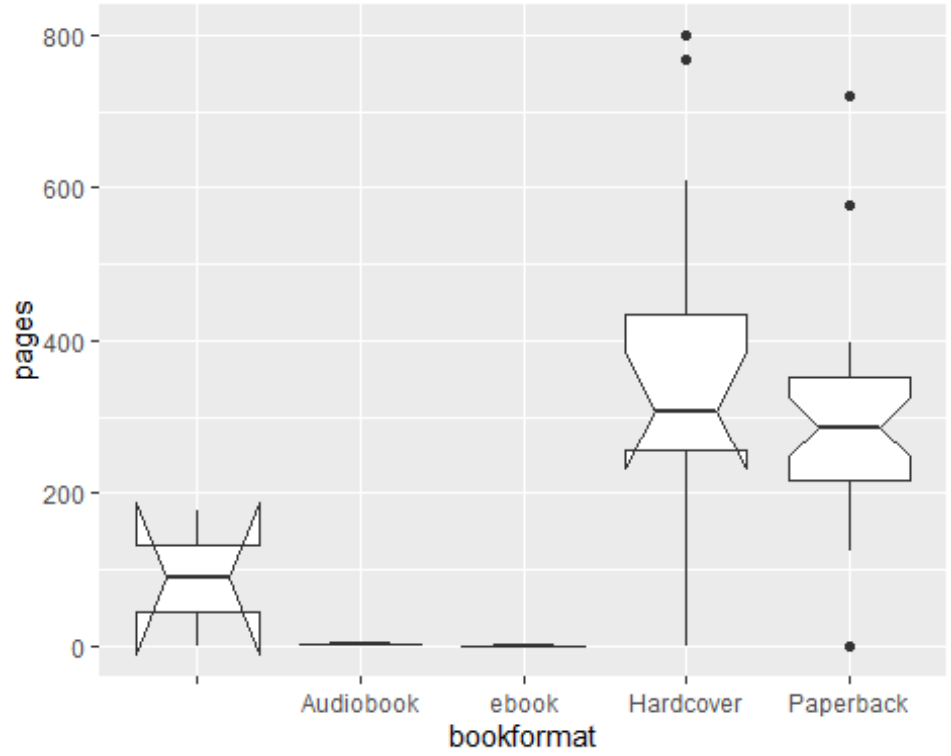
Making a frequency polygon with bin width

```
ggplot(data, aes(x=rating)) +geom_freqpoly(binwidth=4)
```



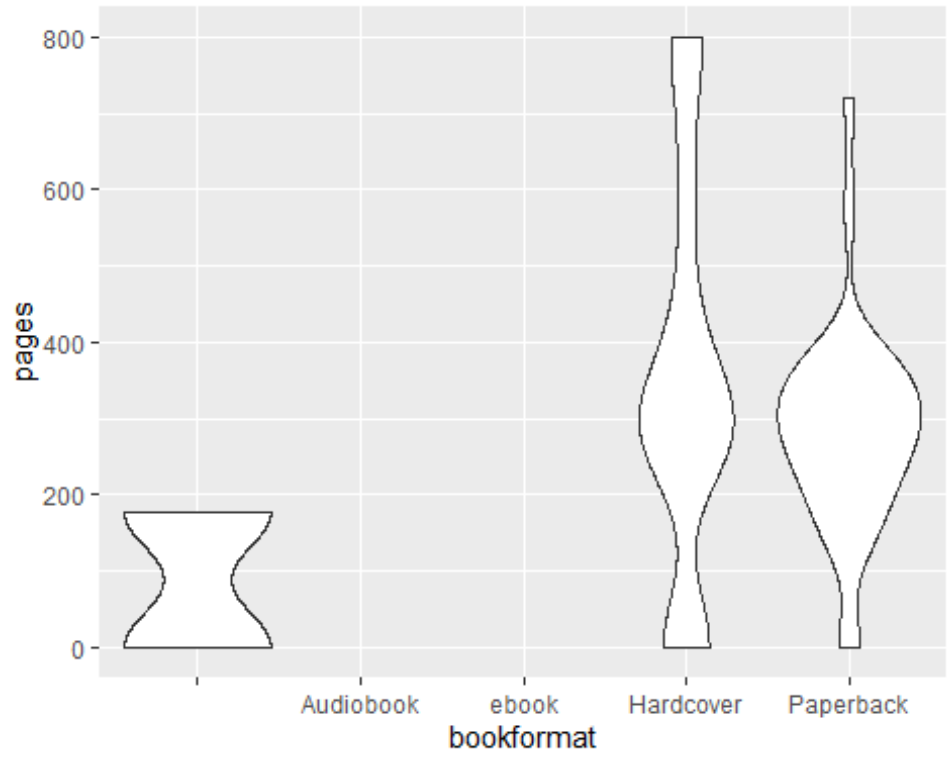
Making a box plot with notches

```
ggplot(data.new1, aes(x = bookformat, y = pages)) +
 geom_boxplot(notch = TRUE)

## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



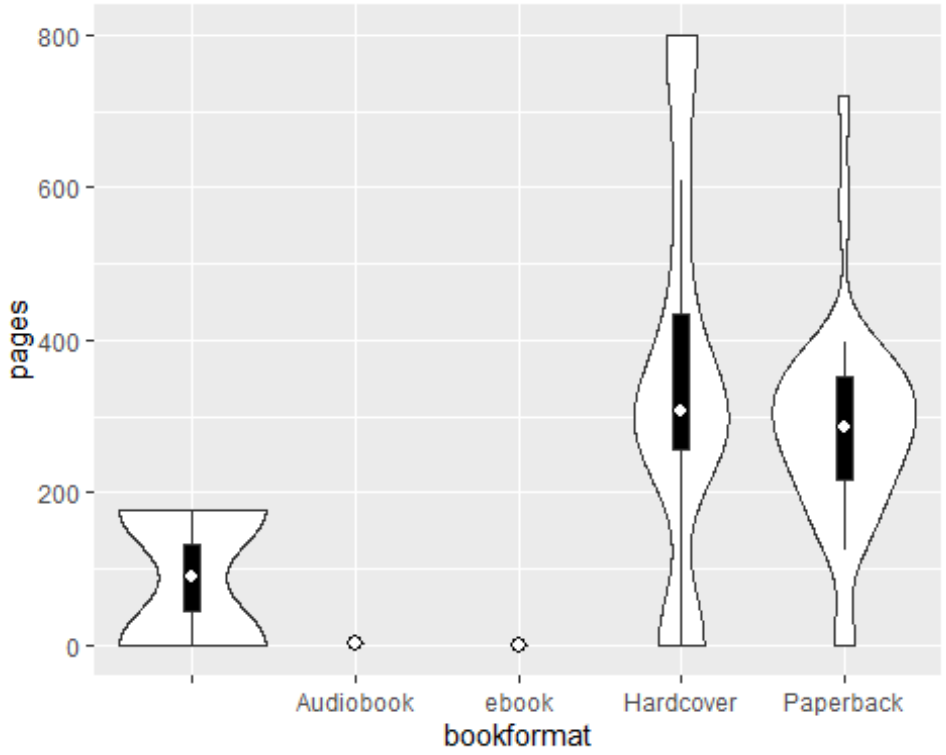
Making a violin plot

```
data_p <- ggplot(data.new1, aes(x = bookformat, y = pages))
data_p + geom_violin()

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



Making a violin plot where the box plot outliers are not displayed

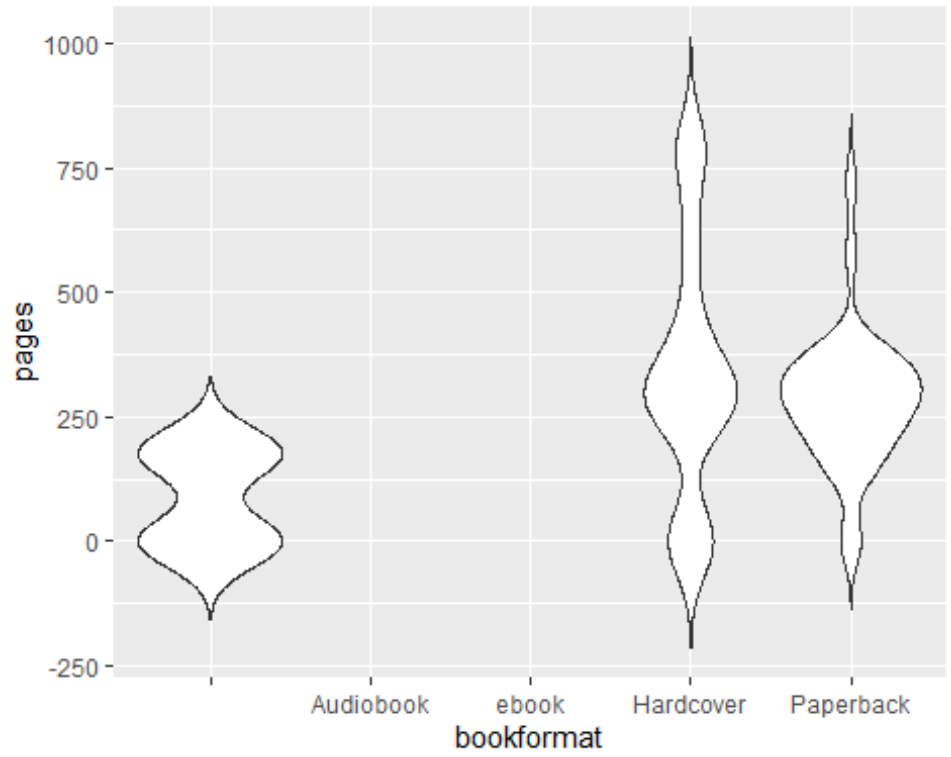
```
data_p +
 geom_violin() + geom_boxplot(width = .1, fill = "black", outlier.colour =
NA) + stat_summary(fun = median, geom = "point", fill = "white", shape = 21,
size = 2.5)

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



Making a violin plot by keeping the tails

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

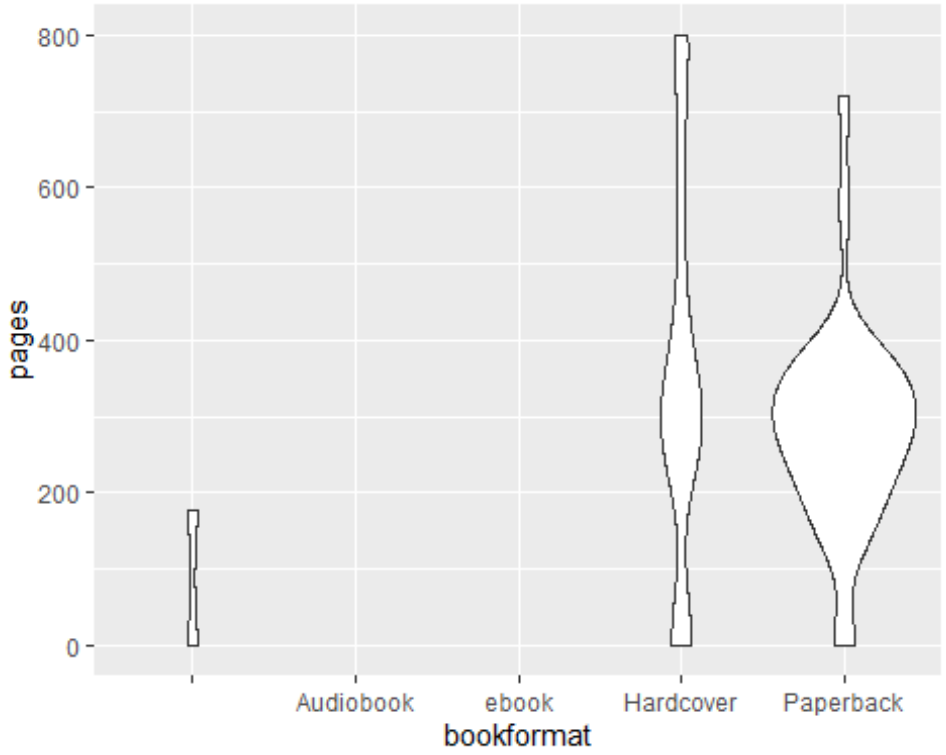
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



Making a violin plot with area proportional to number of observations

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

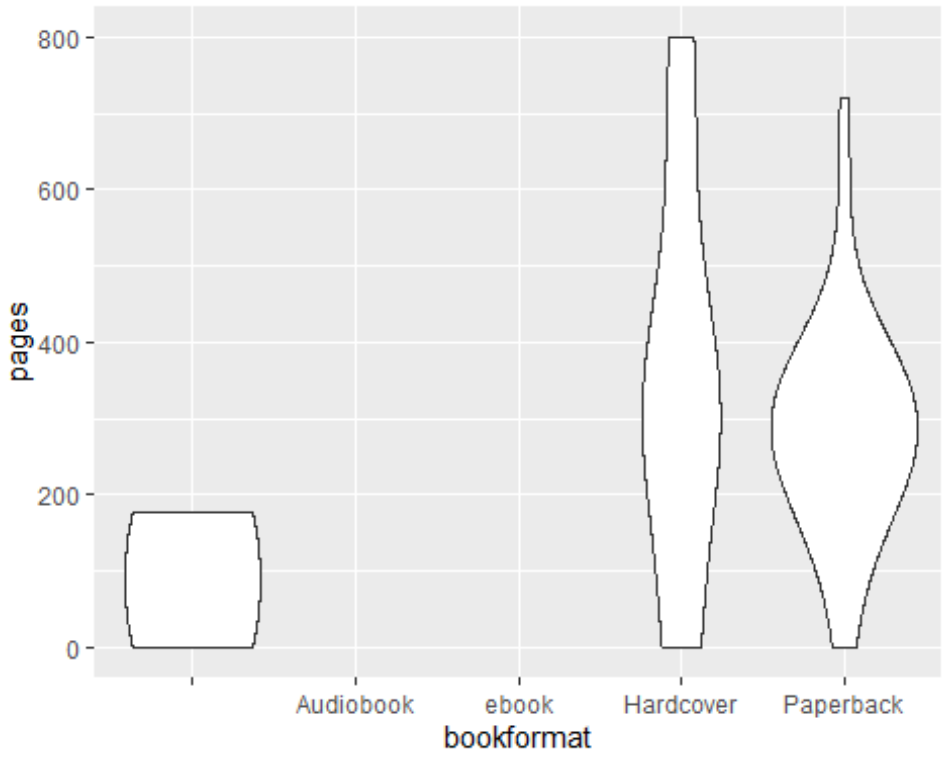
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



Making a violin plot with more smoothing

```
data_p + geom_violin(adjust = 2)
```

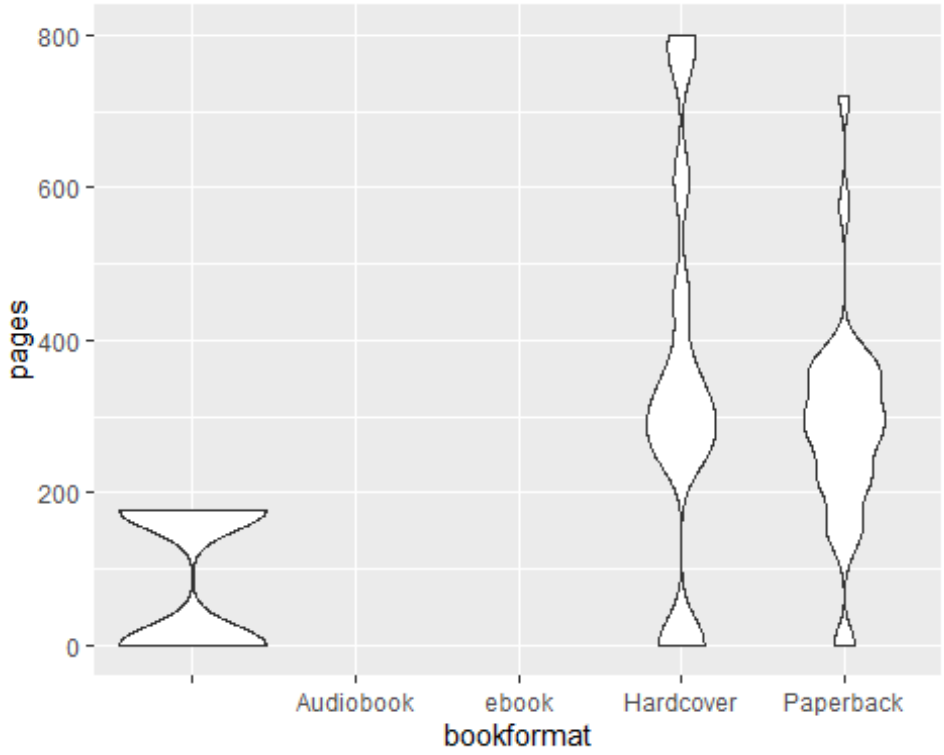
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



Violin plot with less smoothing

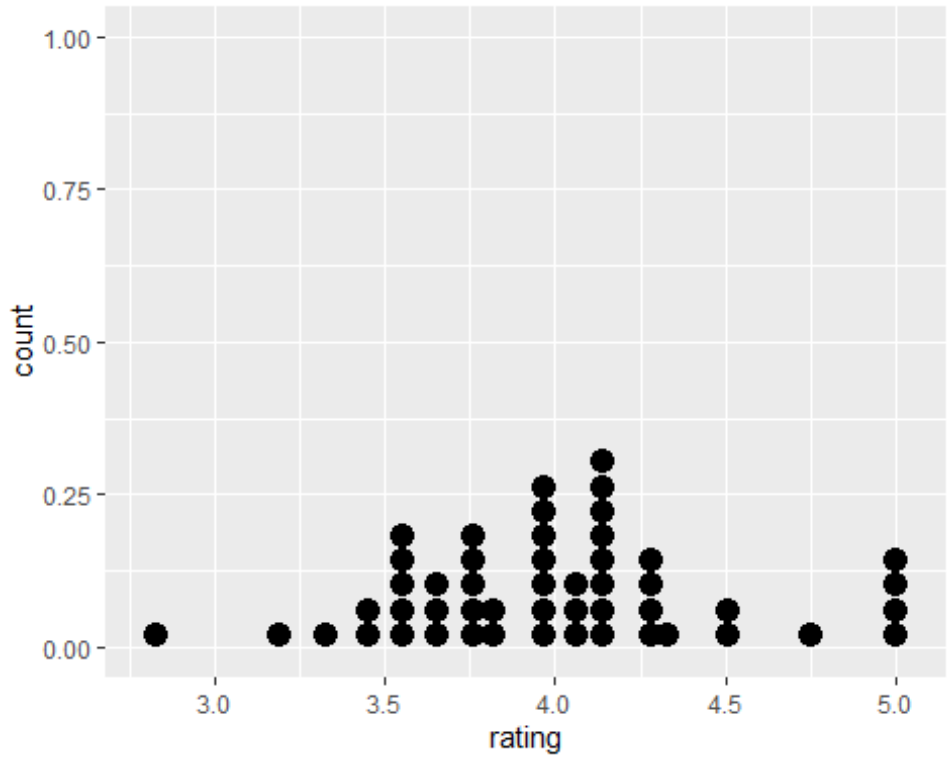
```
data_p + geom_violin(adjust = .5)
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



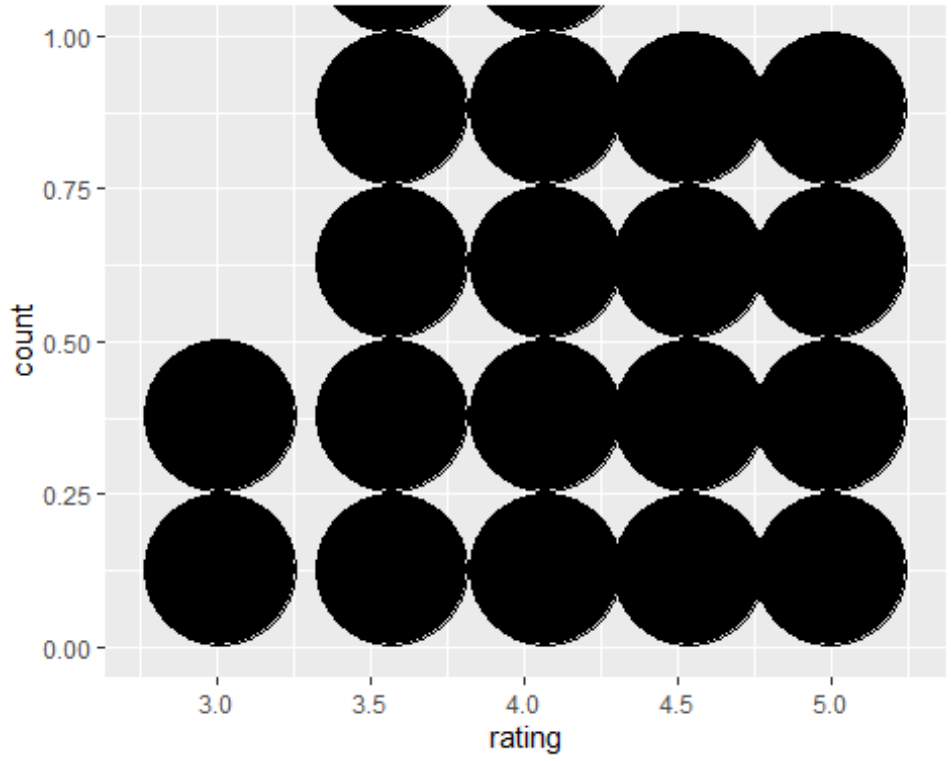
Making a wilkinson dot plot

```
data_d <- ggplot(data.new1, aes(x = rating))
data_d + geom_dotplot()

## Bin width defaults to 1/30 of the range of the data. Pick better value
with `binwidth`.
```



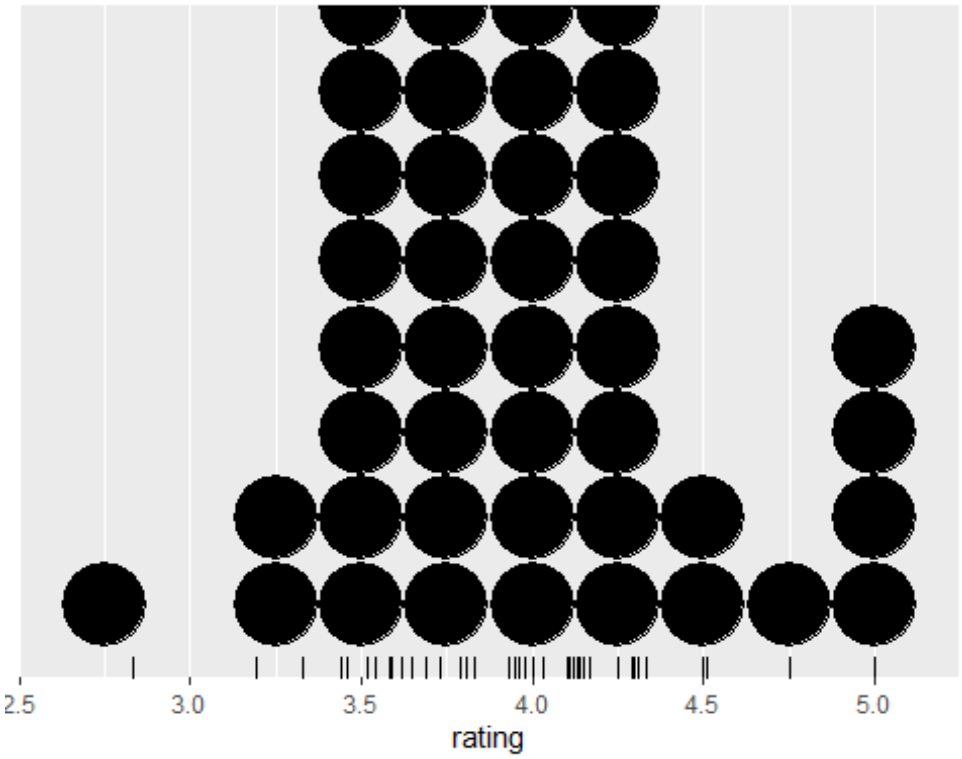
Making a wilkinson dot plot with binwidth

```
data_d + geom_dotplot(binwidth=.5)
```



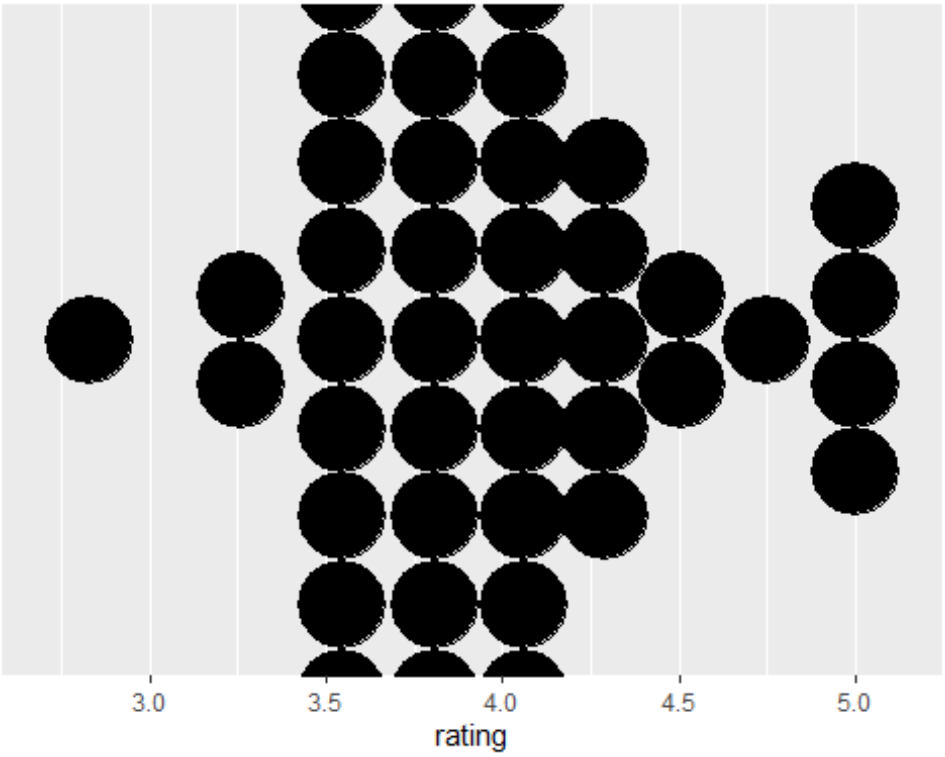
Making a wilkinson dot plot with histodot (fixed-width) binning

```
data_d +
 geom_dotplot(method = "histodot", binwidth = .25) +
 geom_rug() +
 scale_y_continuous(breaks = NULL) +
 theme(axis.title.y = element_blank())
```



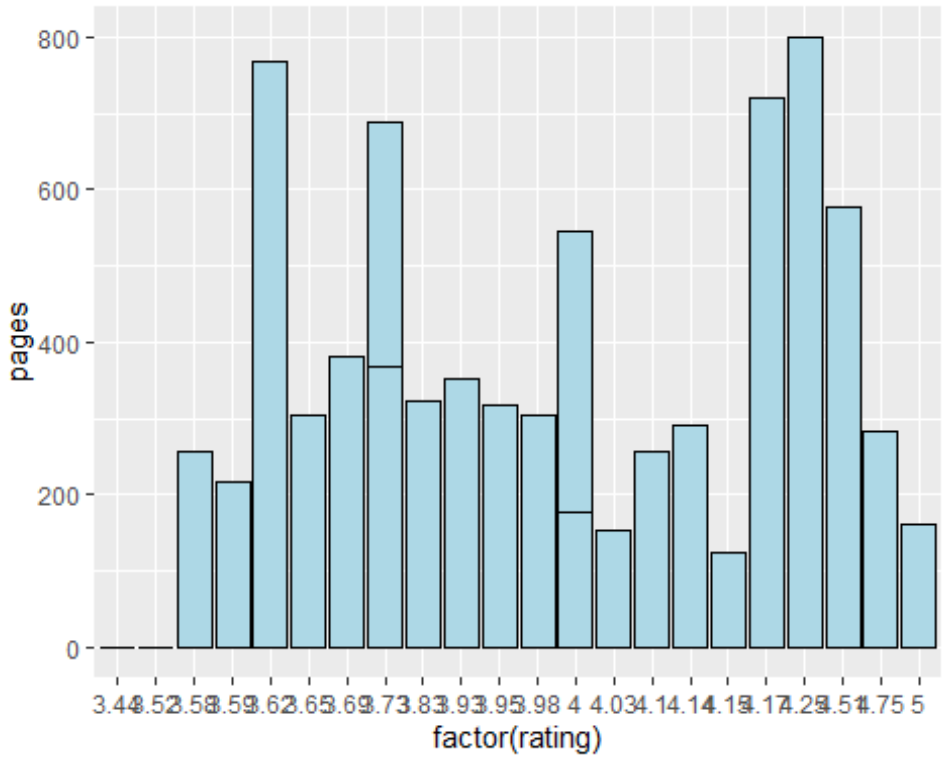
Making a wilkinson dot plot with stackdir = “center”

```
data_d +
 geom_dotplot(binwidth = .25, stackdir = "center") +
 scale_y_continuous(breaks = NULL) +
 theme(axis.title.y = element_blank())
```



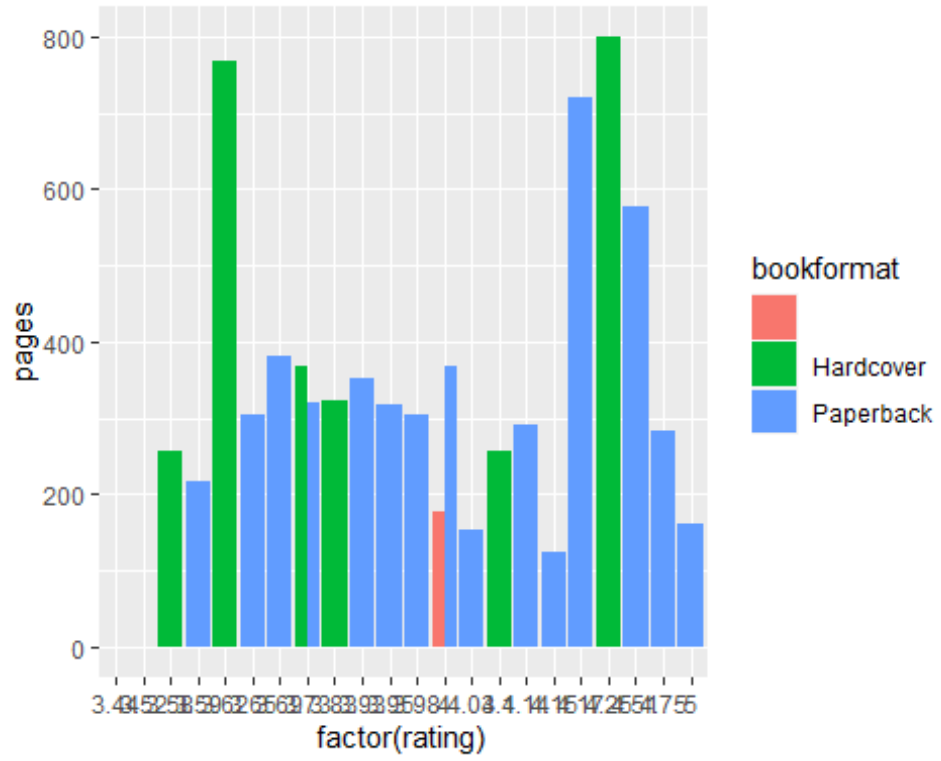
Adding color fill and outline to a bar graph using 'fill' and 'colour' command

```
data.new2 <- data.new1[1:25,]
bar <- ggplot(data.new2, aes(x = factor(rating), y = pages))
bar + geom_col(fill = "lightblue", colour = "black")
```



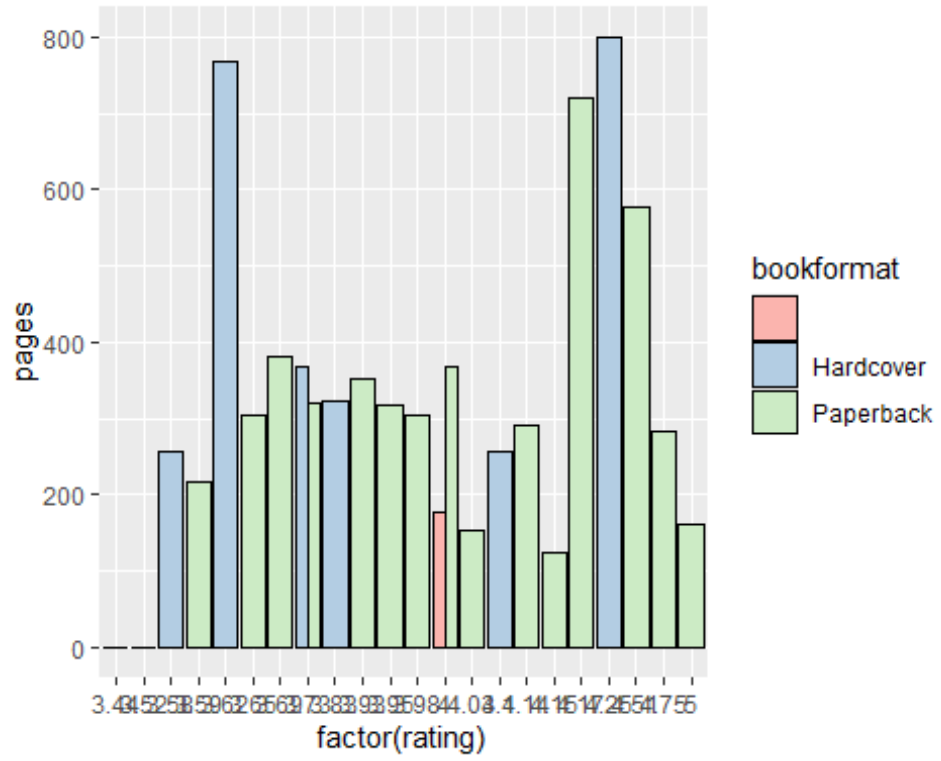
Grouping Bars Together Map a variable to fill, and use geom_col(position = “dodge”)

```
bar1 <- ggplot(data.new2, aes(x = factor(rating), y = pages, fill =
bookformat))
bar1+geom_col(position = "dodge")
```



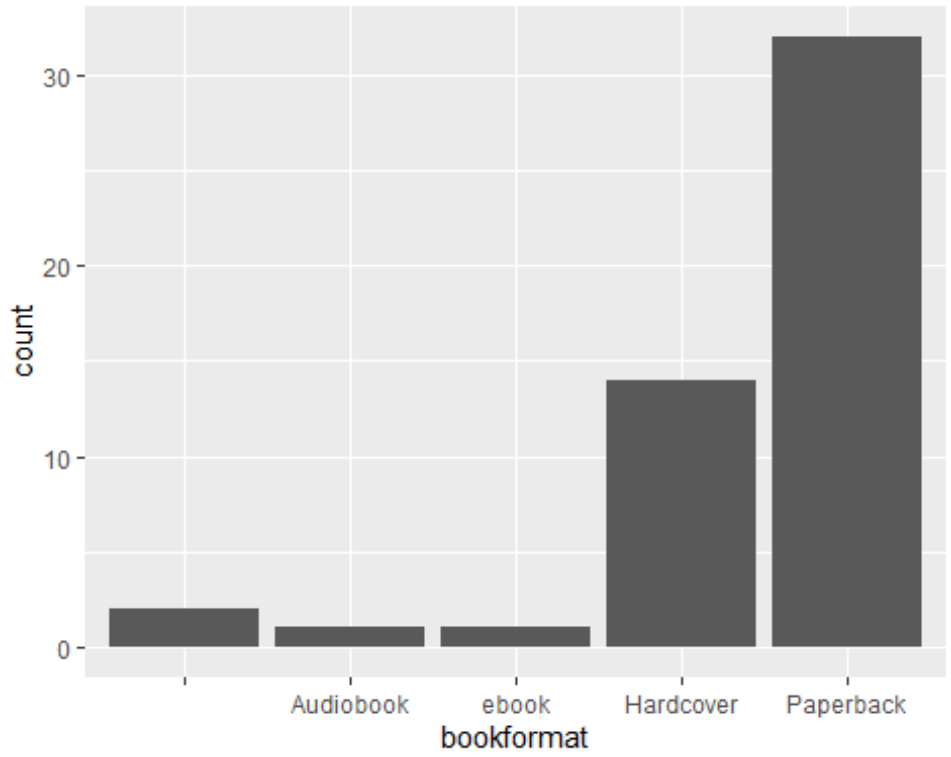
Grouped bars with black outline and a different color palette

```
bar1+ geom_col(position = "dodge", colour = "black") +
 scale_fill_brewer(palette = "Pastel1")
```



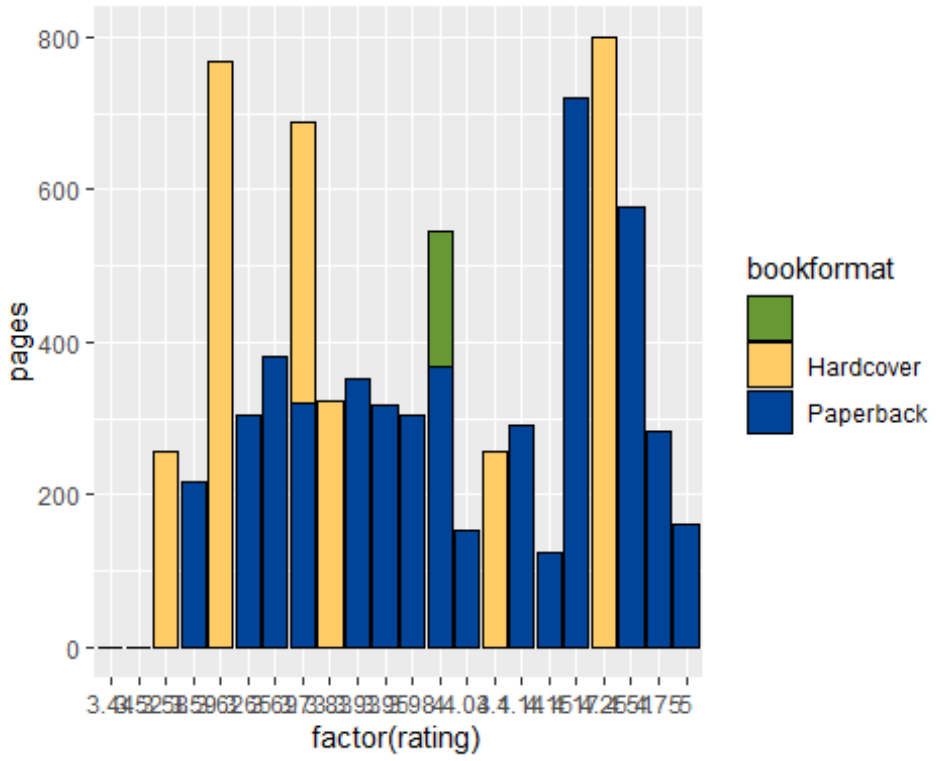
Making a Bar Graph of Counts

```
ggplot(data.new1, aes(x = bookformat)) +
 geom_bar()
```



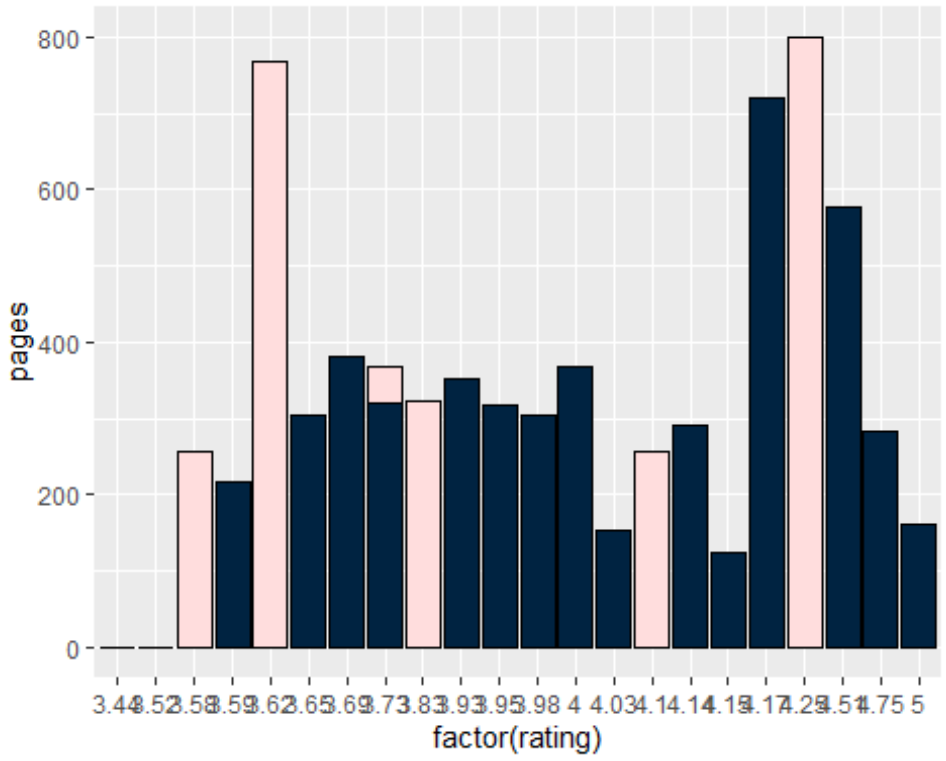
Using Colors in a Bar Graph

```
bar1 + geom_col(colour = "black") +
 scale_fill_manual(values = c("#669933", "#FFCC66","#004499") )
```



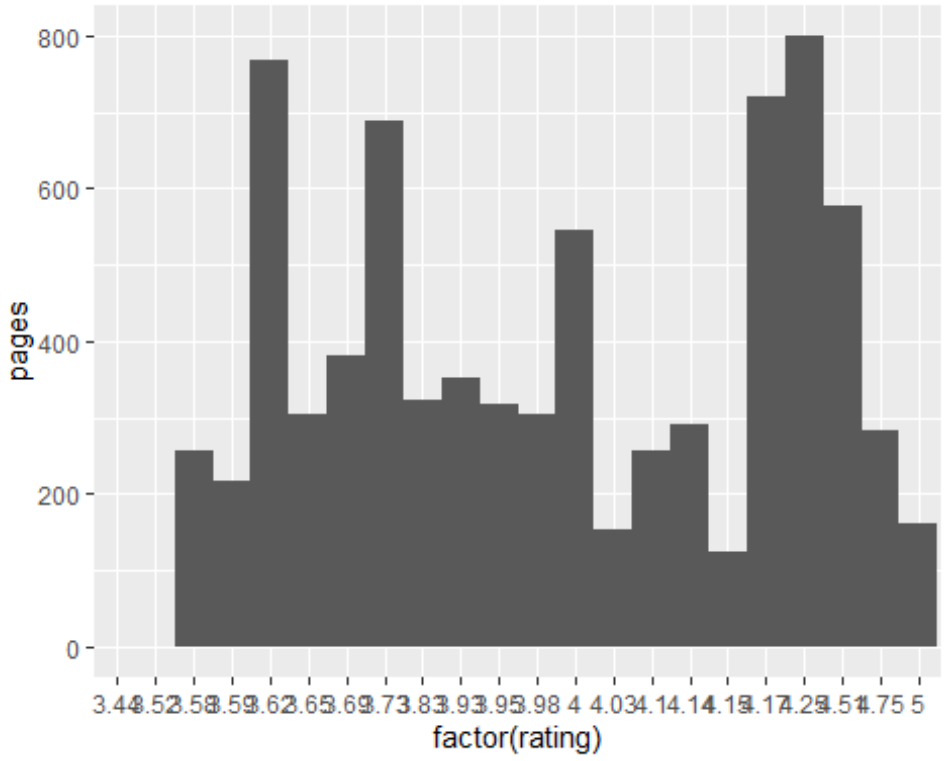
Making bar graph with customized colors and no legend

```
bar1 +  geom_col(position = "identity", colour = "black", size = 0.25) +
  scale_fill_manual(values = c("#CCEEFF", "#FFDDDD","#002341"), guide =
FALSE)

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide.
Please
## use `guide = "none"` instead.
```



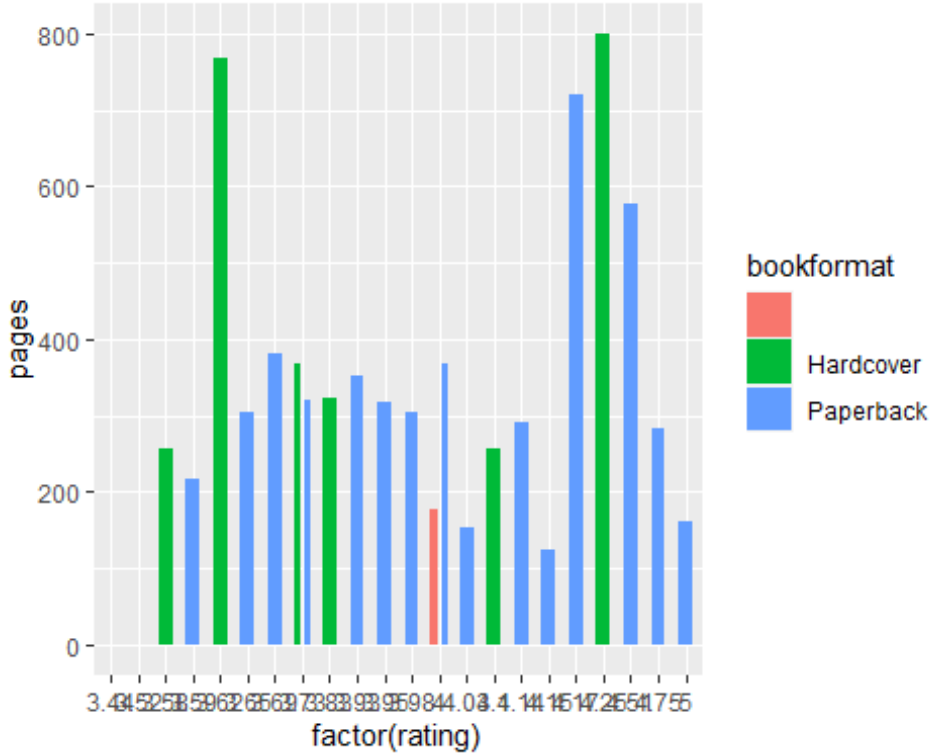
Adjusting Bar Width

```
ggplot(data.new2, aes(x = factor(rating), y = pages)) + geom_col(width = 1)
```



Adjusting Bar Spacing

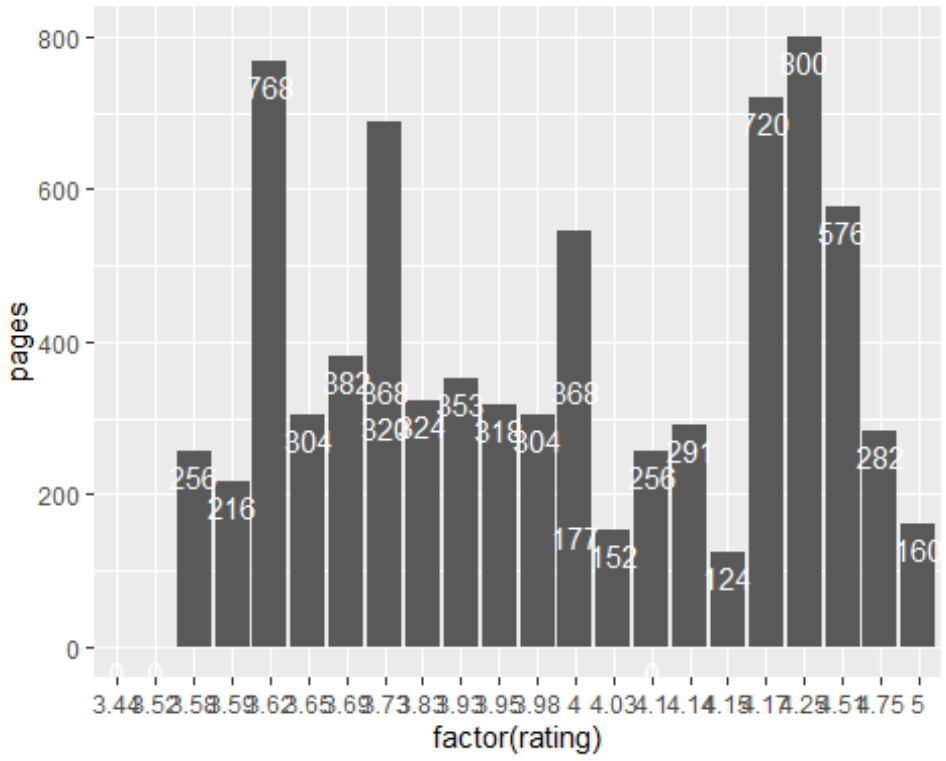
```
bar1 +  geom_col(width = 0.5, position = position_dodge(0.8))
```



Adding Labels to a Bar Graph

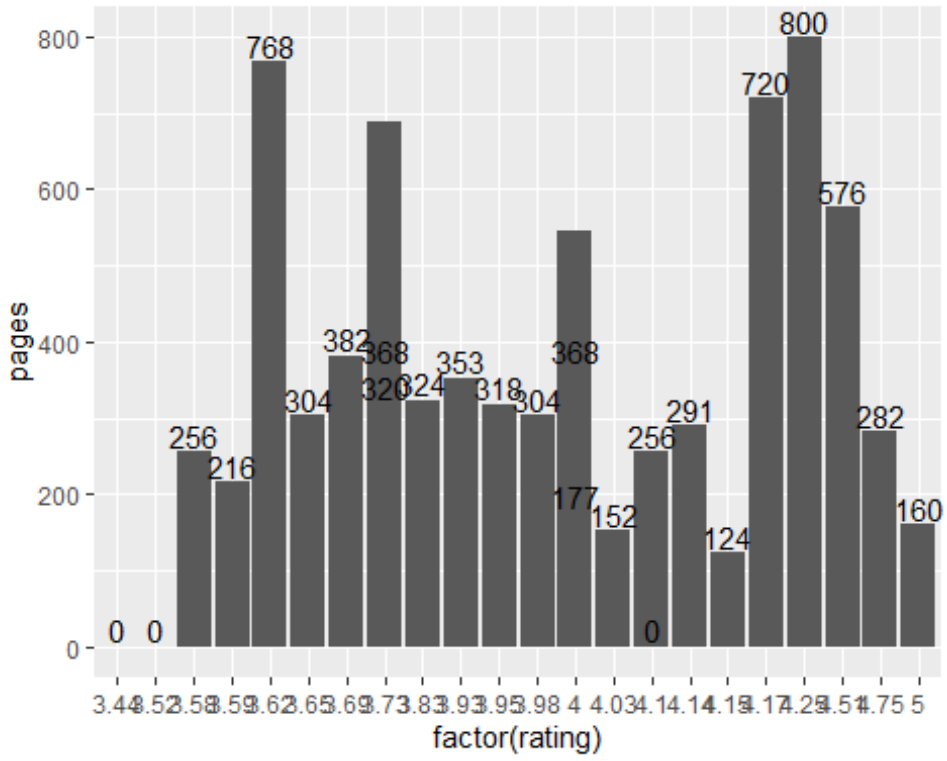
Below the top

```
ggplot(data.new2, aes(x = factor(rating), y = pages)) + geom_col() +
geom_text(aes(label = pages), vjust = 1.5, colour = "white")
```



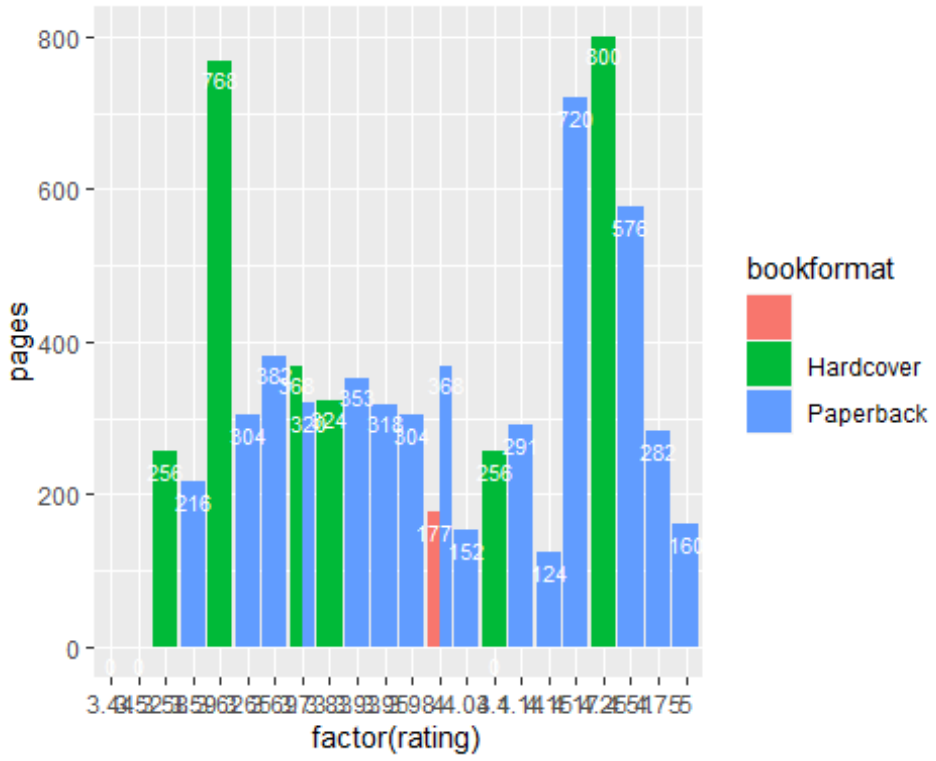
Above the top

```
ggplot(data.new2, aes(x = factor(rating), y = pages)) + geom_col() +
 geom_text(aes(label = pages), vjust = -0.2)
```



Adding labels on grouped bars

```
bar1 +   geom_col(position = "dodge") +
geom_text(
 aes(label = pages),
 colour = "white", size = 3,
 vjust = 1.5, position = position_dodge(.9)
 )
```


800
600
400
200
0
pages
factor(rating)
bookformat
Hardcover
Paperback