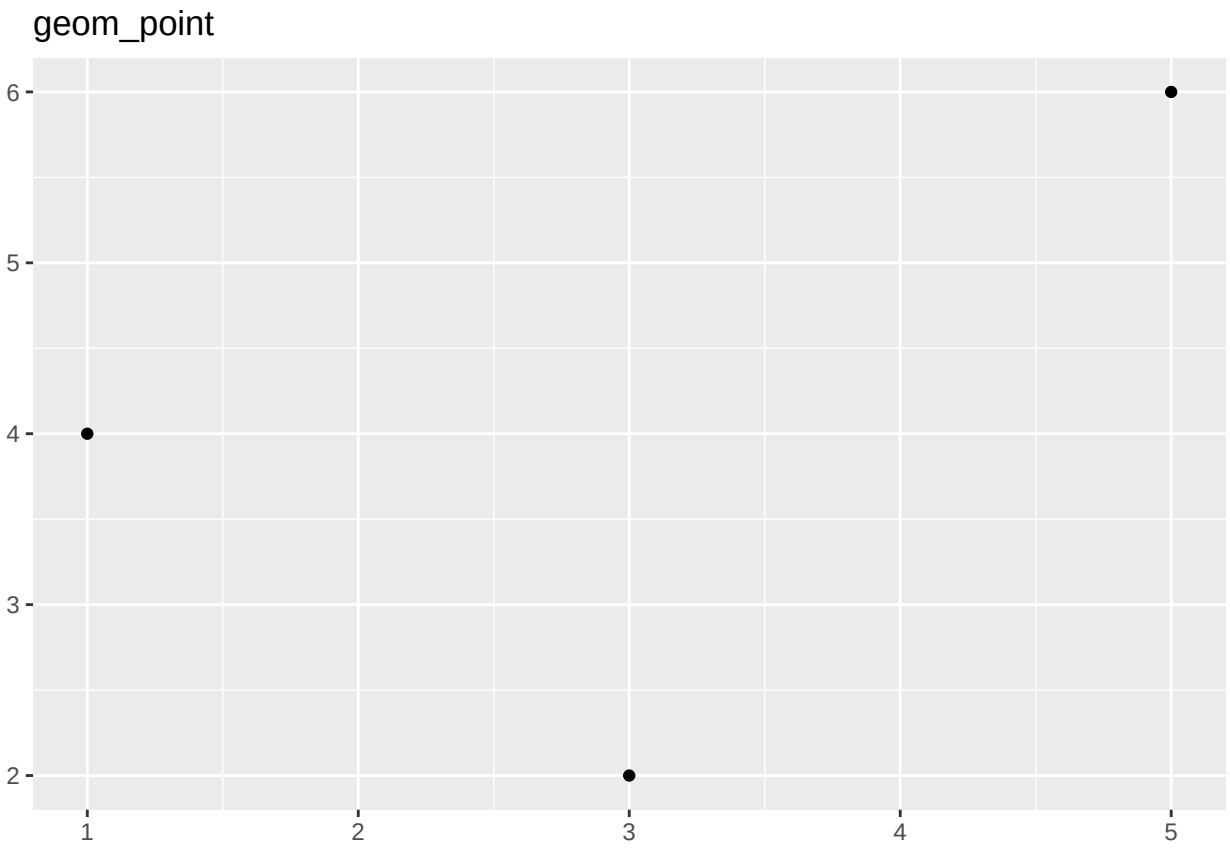
R Notebook

Pradip Basnet

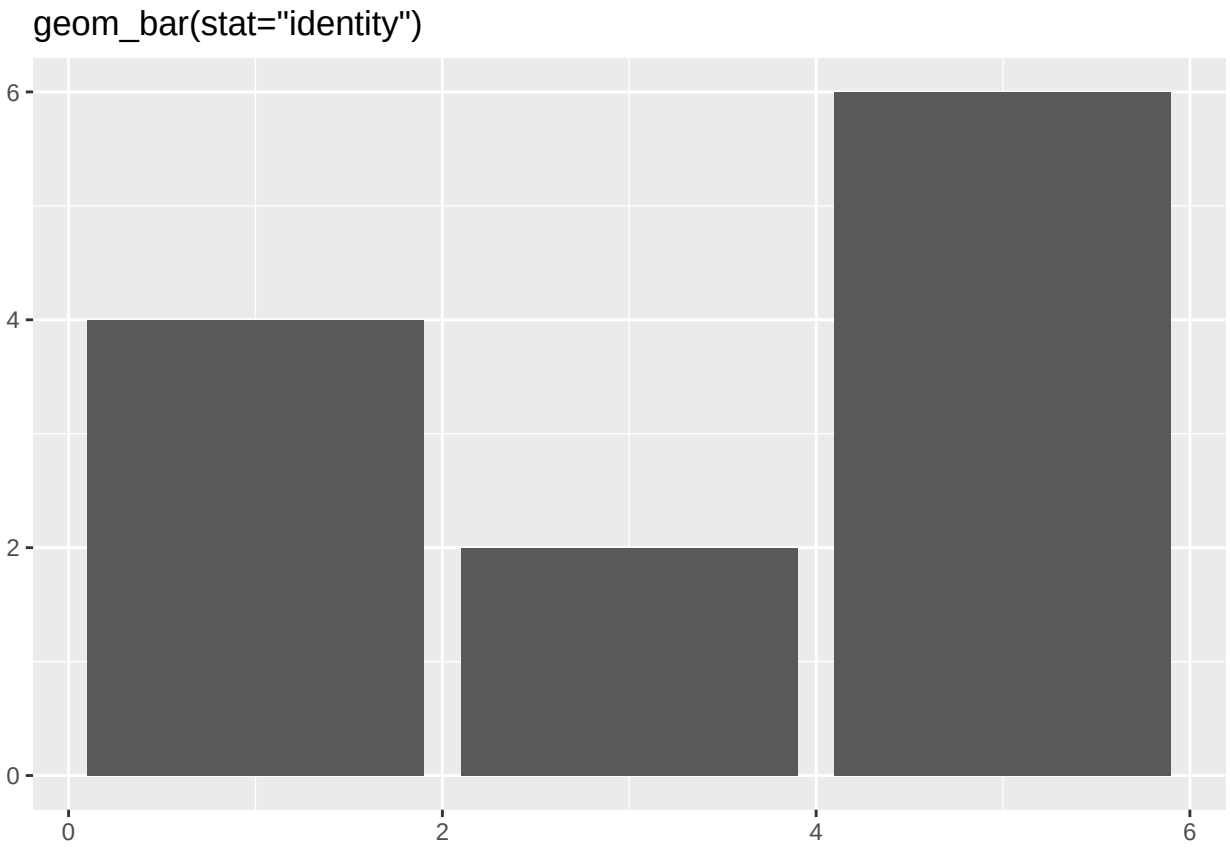
```
library(ggplot2)
```

the code create different types of plots from the same dataset. Each plot is based on the df data frame, which has three columns: x, y, and label. The code creates the plot like scatter plot, bar plot, line plot, area plot, path plot, area plot, and tile plot, overall these plots demonstrate different ways to visualize data using ggplot2, each suited for different types of analysis and presentation.

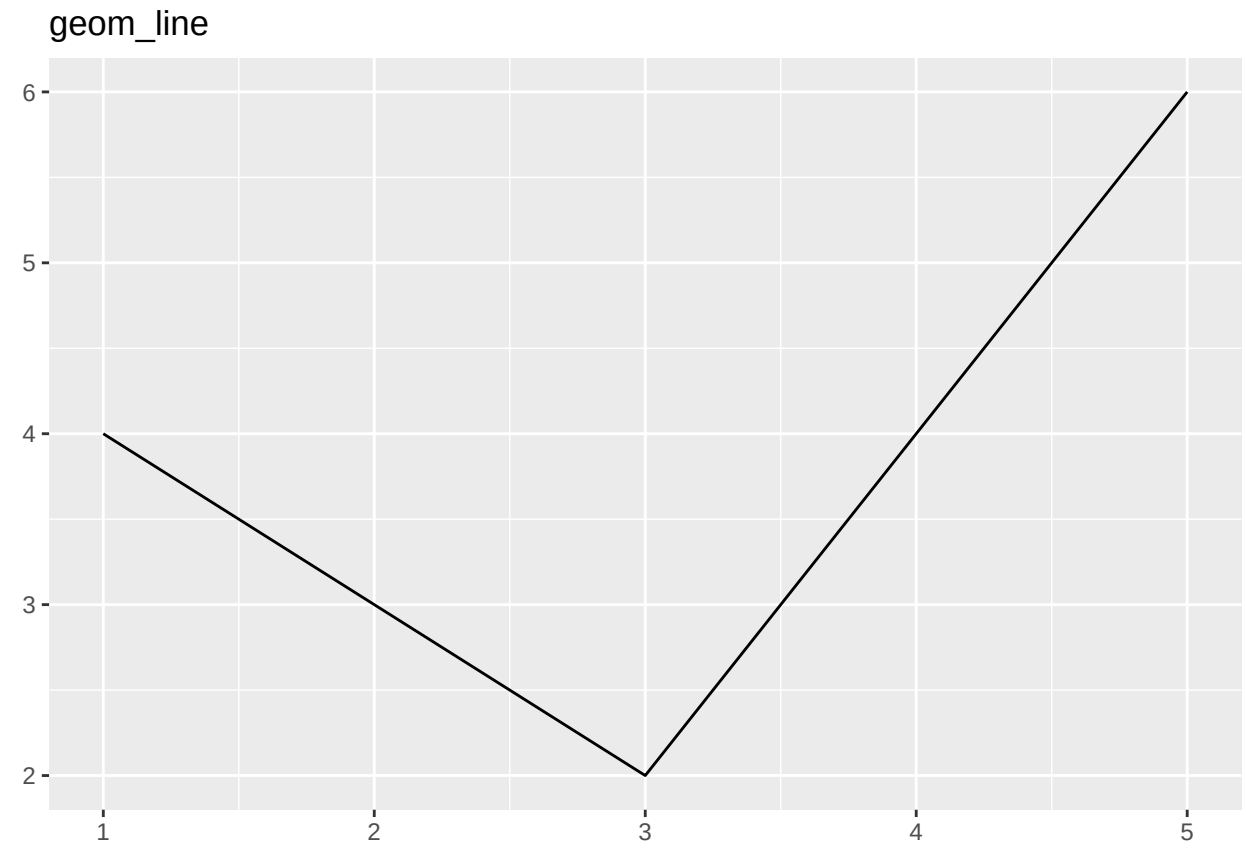
```
df <- data.frame(
  x = c(3, 1, 5),
  y = c(2, 4, 6),
  label = c("a", "b", "c")
)

p <- ggplot(df, aes(x, y, label = label)) +
  xlab(NULL) + ylab(NULL)

# Individual plots
 p + geom_point() + ggtitle("geom_point")
```



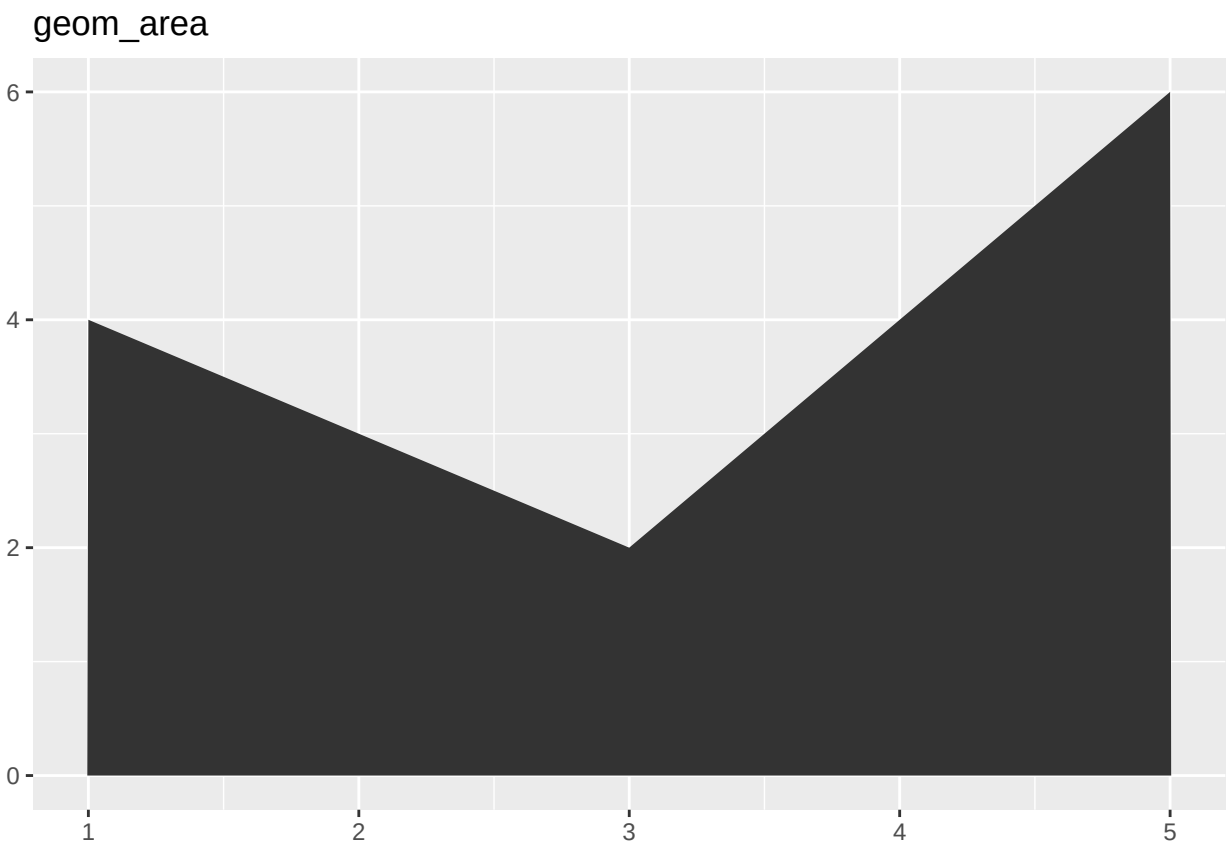

```
p + geom_bar(stat="identity") + ggtitle("geom_bar(stat=\"identity\")")
```



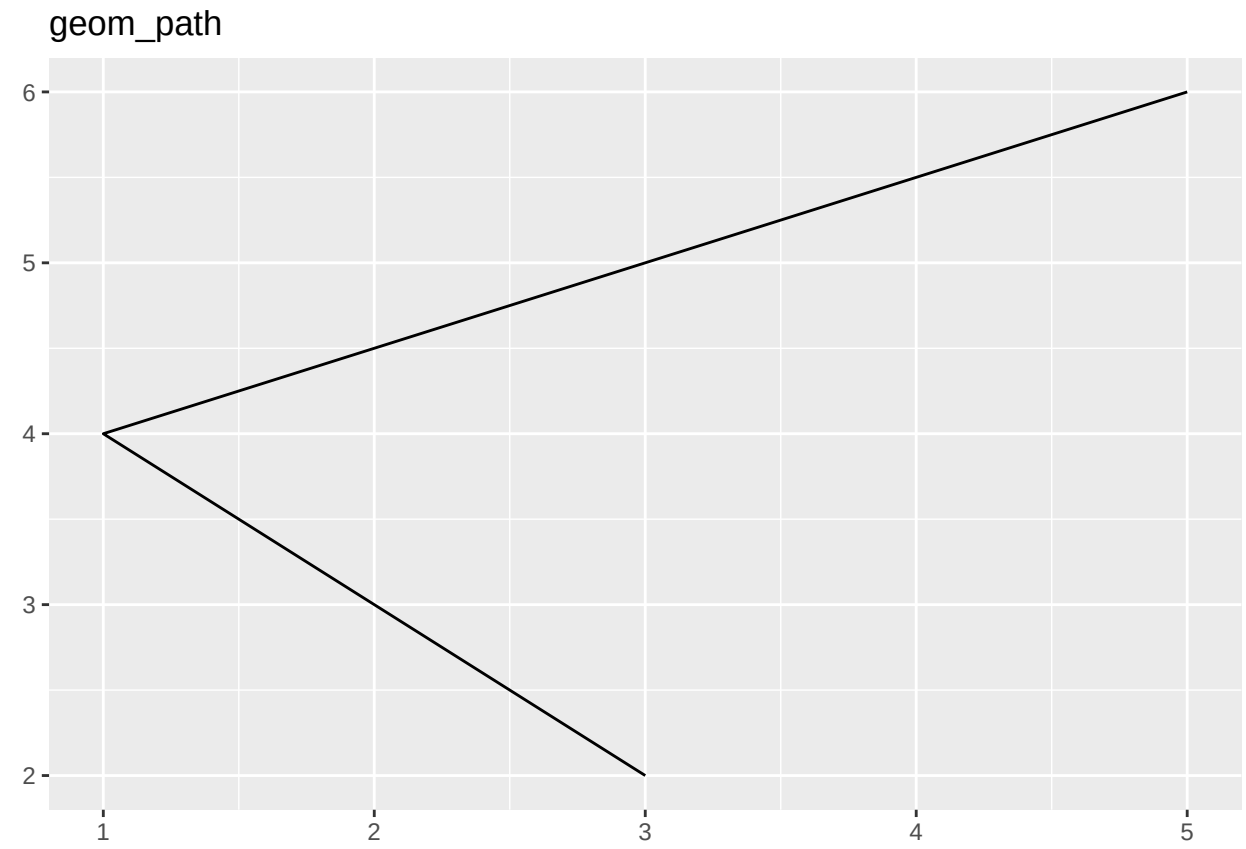

```
p + geom_line() + ggtitle("geom_line")
```



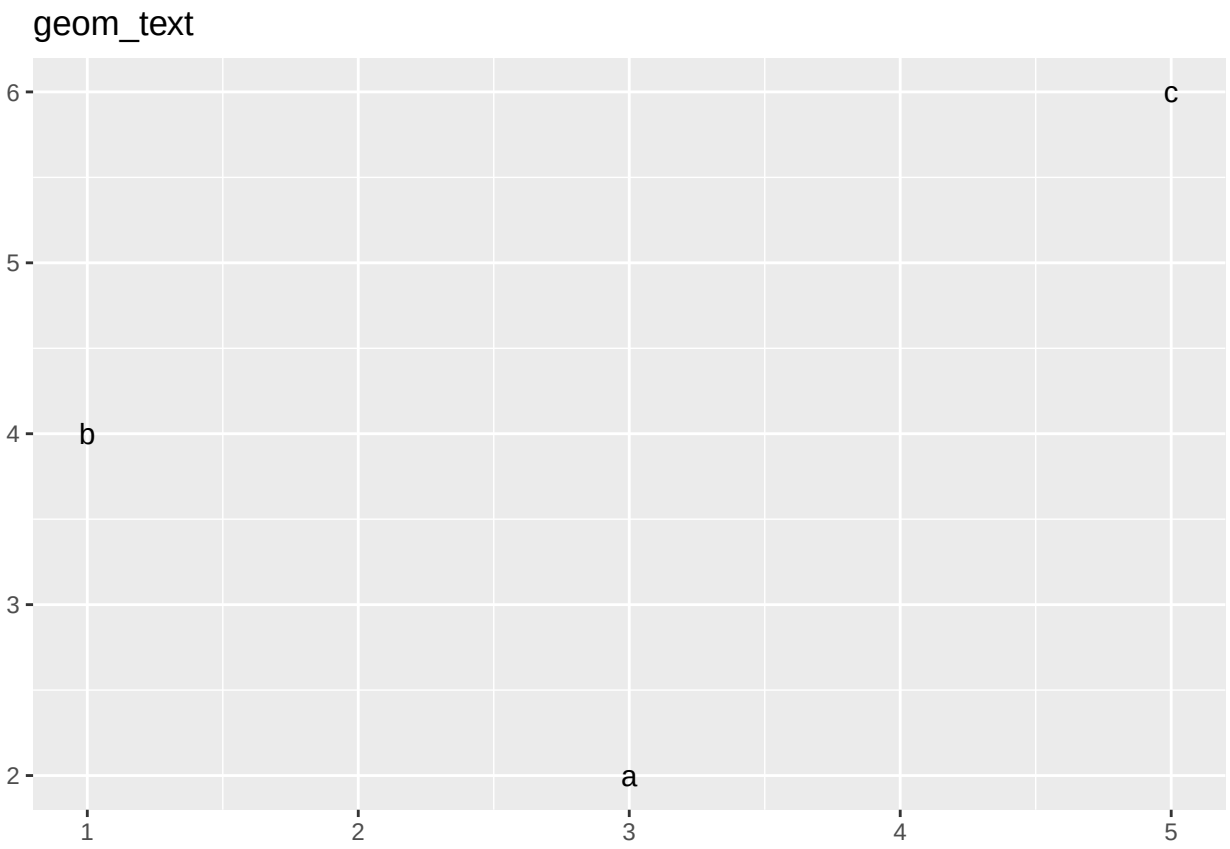

```
p + geom_area() + ggtitle("geom_area")
```



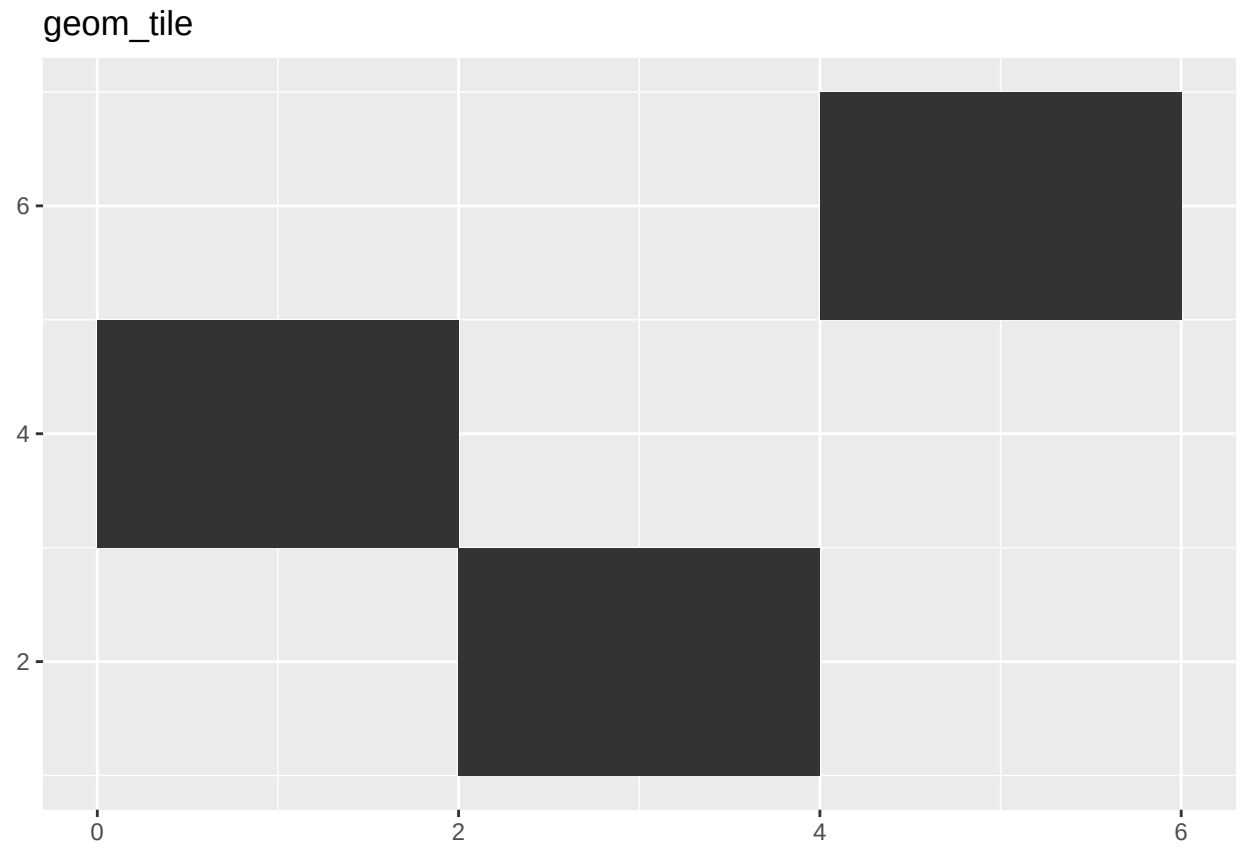

```
p + geom_path() + ggtitle("geom_path")
```




```
p + geom_text() + ggtitle("geom_text")
```




```
p + geom_tile() + ggtitle("geom_tile")
```



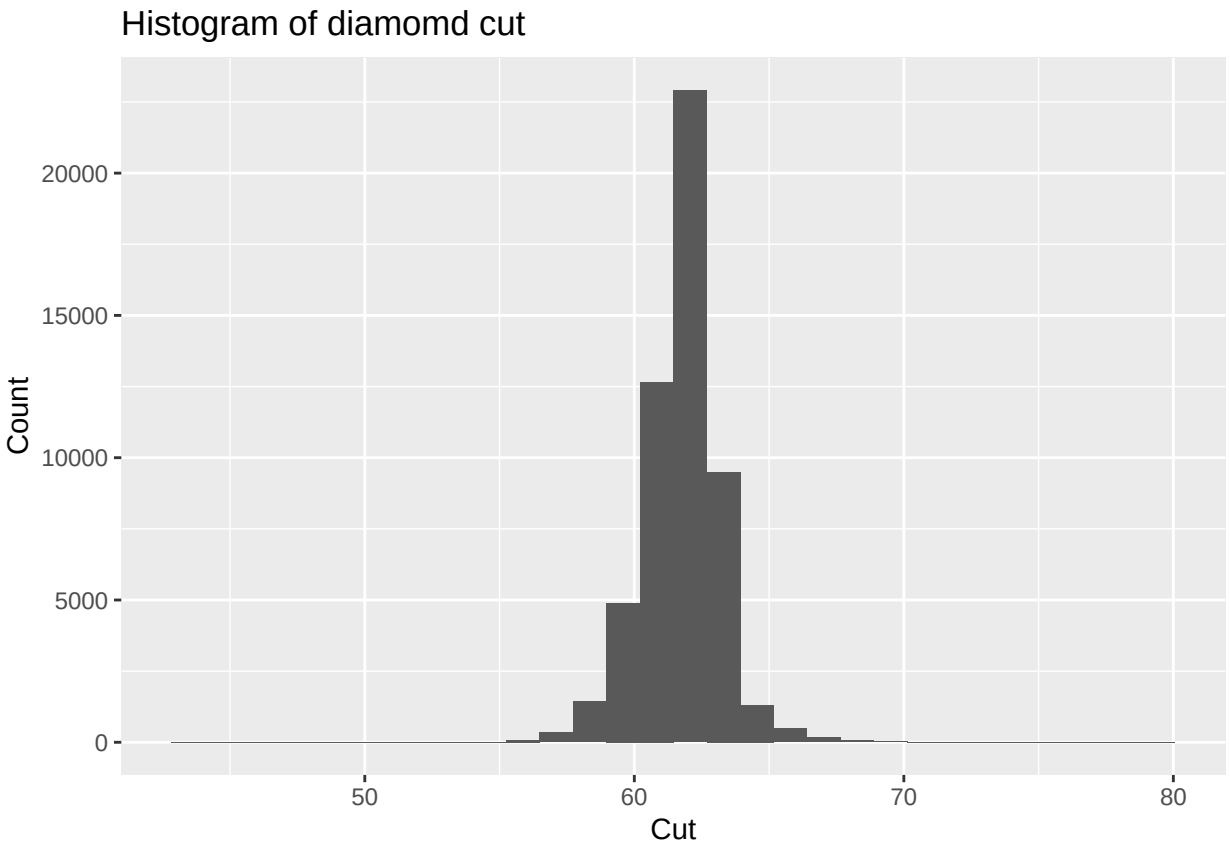
importing the required dataset

```
data("diamonds")
```

The histogram illustrates a right-skewed distribution of diamond cut quality. This indicates that a majority of the diamonds in the dataset possess a higher cut quality, with fewer diamonds having lower quality cuts. The distribution peaks around the 60-65 mark on the “Cut” axis, suggesting that this range represents the most common cut quality in the dataset.There is a noticeable spread in the data, indicating variation in the cut quality among the diamonds. The presence of a few outliers on the lower end of the “Cut” axis suggests the existence of some diamonds with significantly lower cut quality compared to the majority.

```
ggplot(diamonds,aes(x=depth))+
  geom_histogram()+
  xlab("Cut")+
  ylab("Count")+
  ggtitle("Histogram of diamomd cut")
```

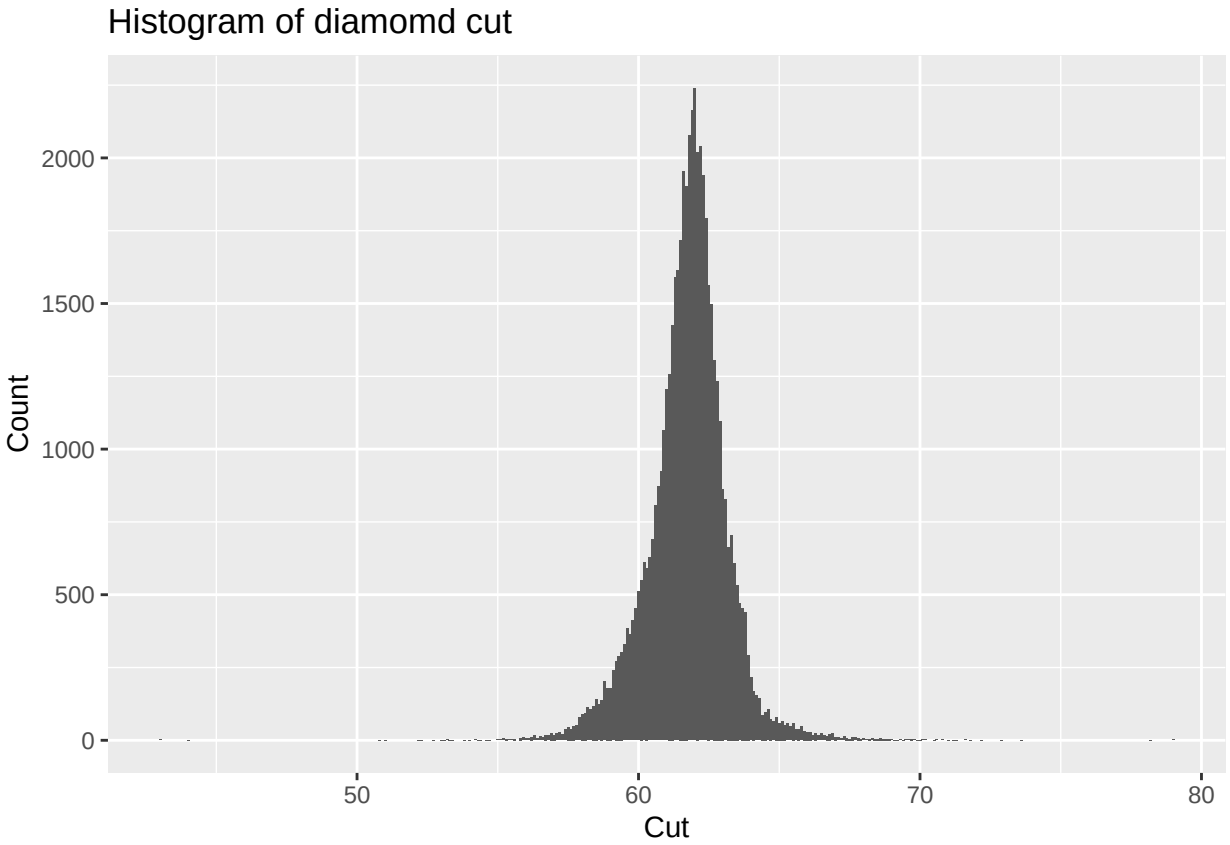
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



the following plot is same as the above plot with different binwidth.he histogram illustrates that the cut quality of diamonds in this dataset is generally good, with a significant proportion falling within the higher range. While there is some variability in cut quality, the majority of diamonds exhibit cuts that are considered to be of high quality. The presence of a few outliers with lower cut quality suggests that there might be some variation in the cutting process or the quality of the raw diamonds used.

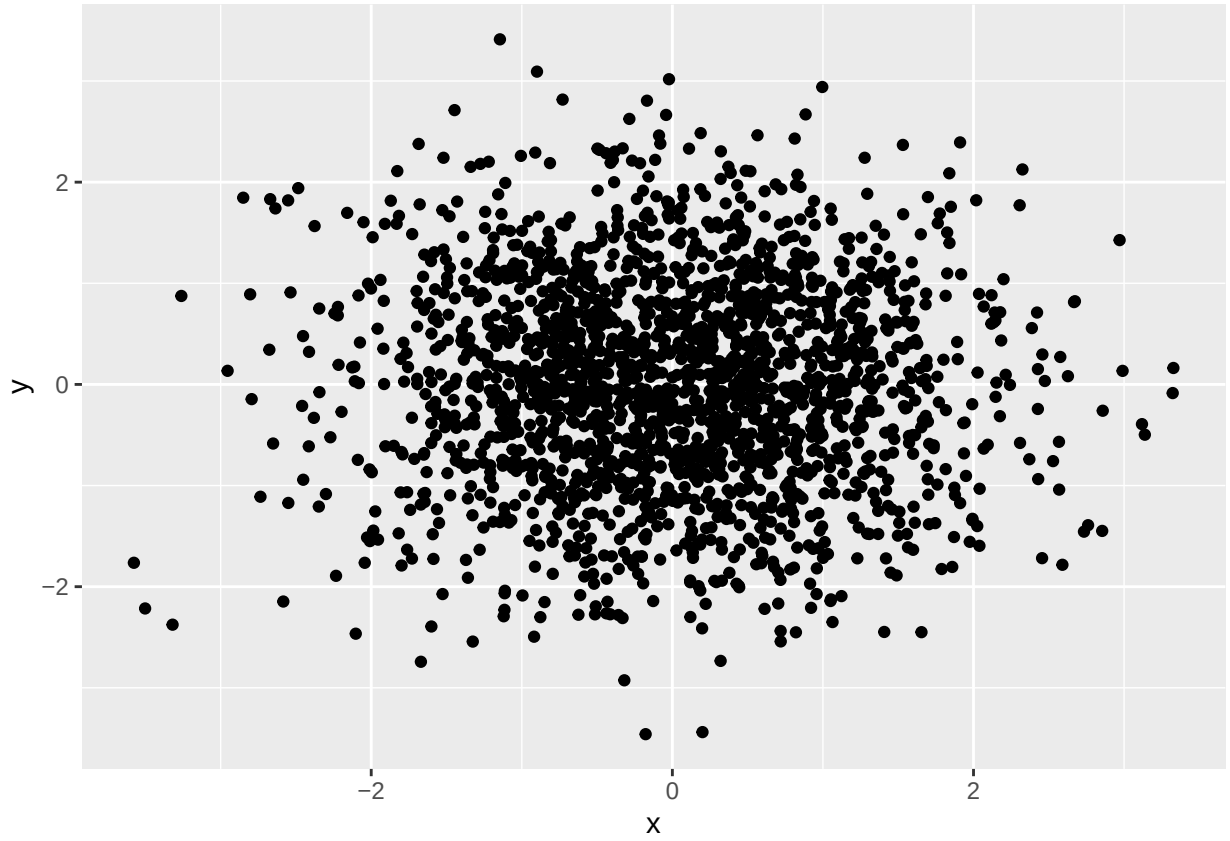
```
ggplot(diamonds,aes(x=depth))+
  geom_bar(binwidth=0.1)+
  xlab("Cut")+
  ylab("Count")+
  ggtitle("Histogram of diamomd cut")
```

```
## Warning in geom_bar(binwidth = 0.1): Ignoring unknown parameters: ‘binwidth‘
```

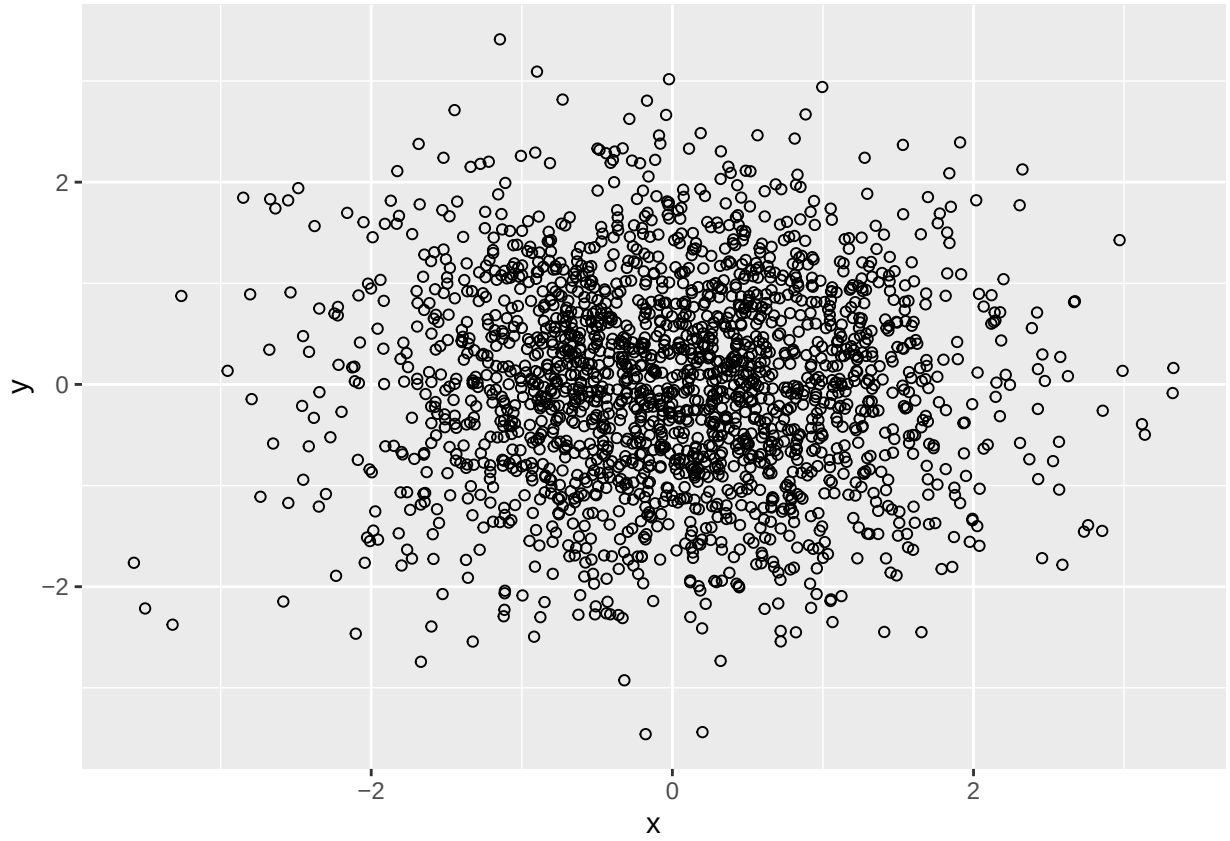


create scatter plots of x versus y from a data frame. The initial plot shows points with the default shape (typically filled circles), while shape = 1 changes the points to open circles. The shape = "." option renders the points as tiny dots, which is useful for visualizing dense data where points overlap. Each variant adjusts the visual representation of the data to highlight different aspects of the plot.

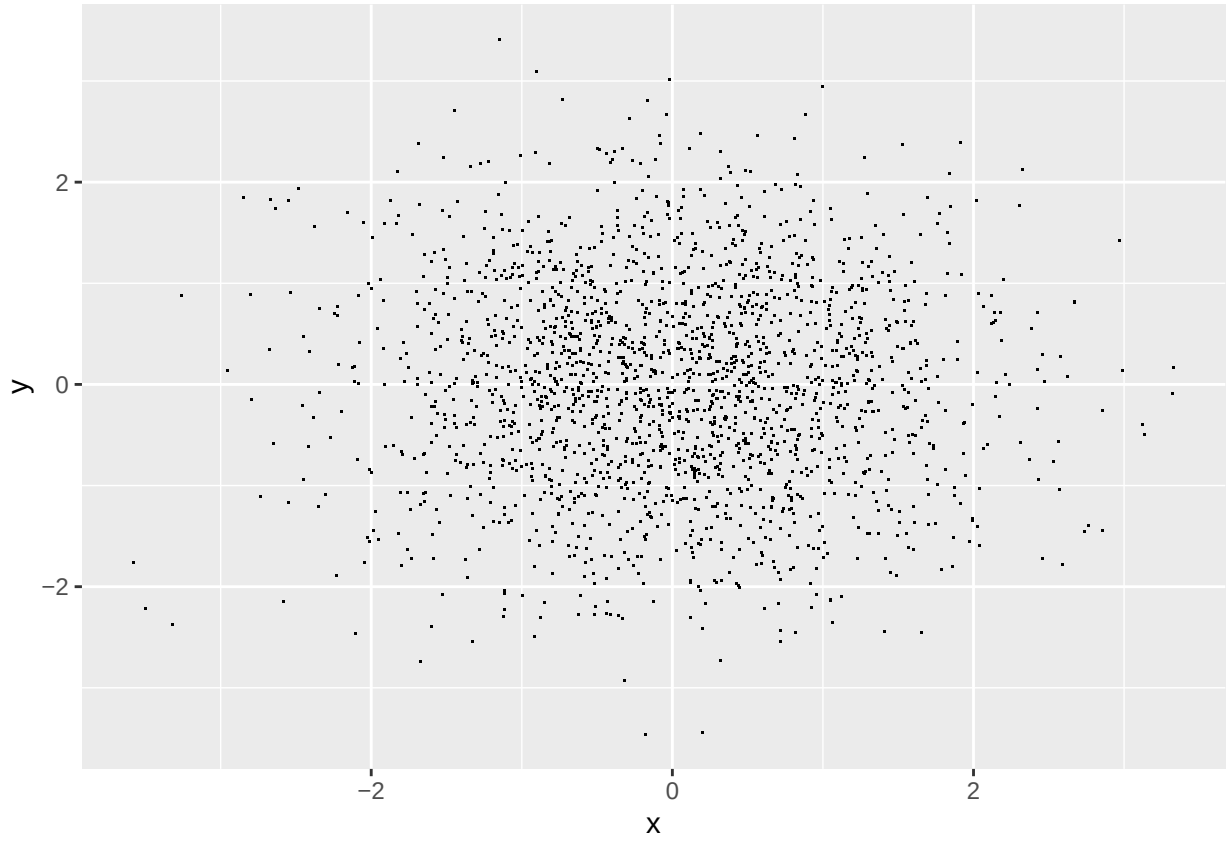
```
df <- data.frame(x = rnorm(2000), y = rnorm(2000))

norm <- ggplot(df, aes(x, y))

norm + geom_point()
```




```
norm + geom_point(shape = 1)
```

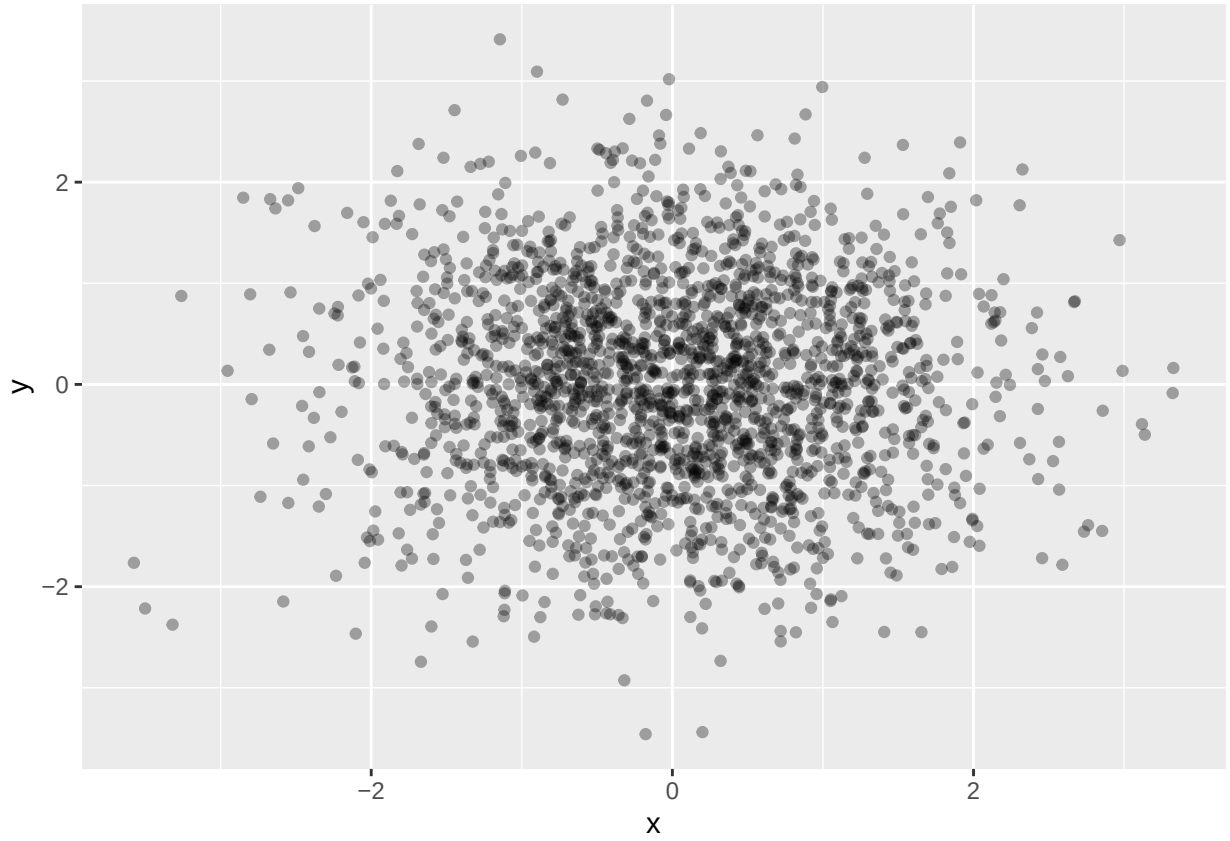



```
# Plot with shape = "." (pixel-sized points)
norm + geom_point(shape = ".")
```

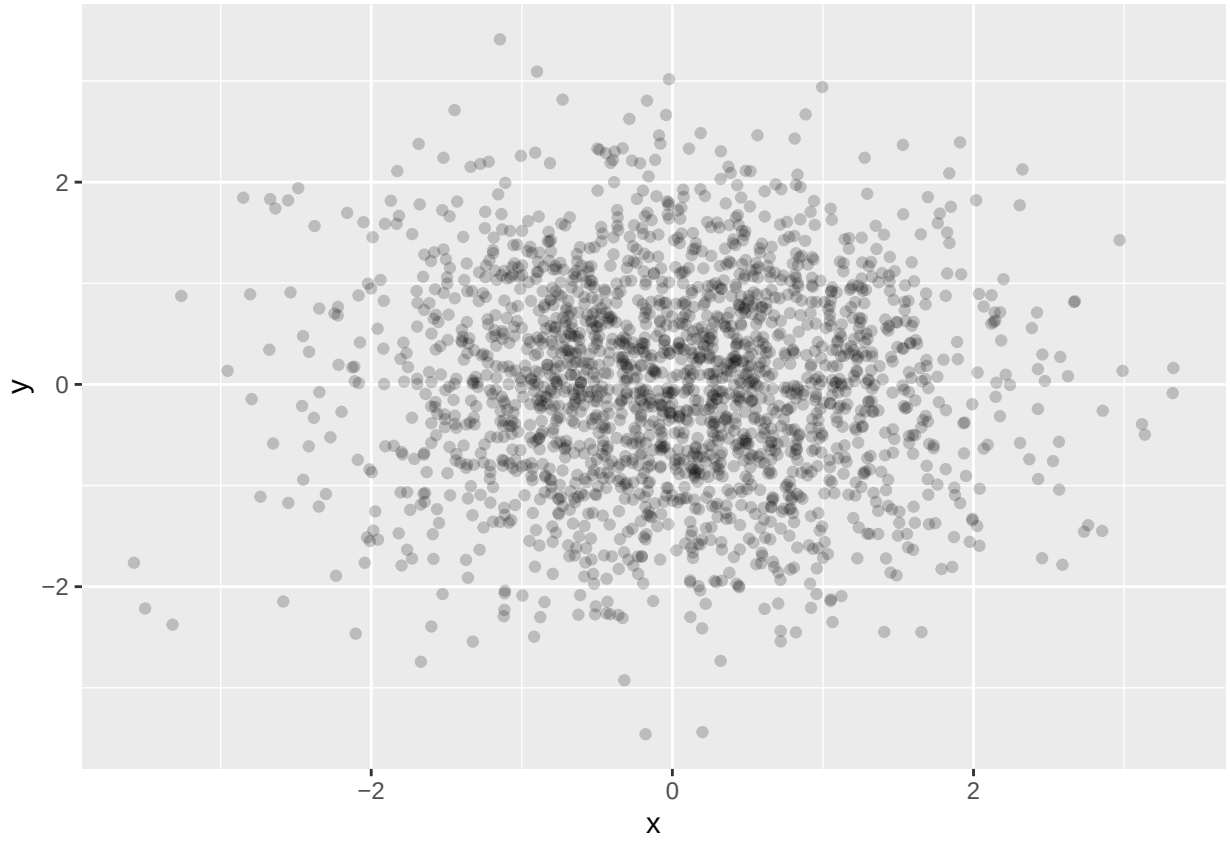


The code adjusts the transparency of points in a scatter plot created with ggplot2 using different alpha values. Setting alpha("black", 1/3) makes the points about 33% opaque, alpha("black", 1/5) makes them 20% opaque, and alpha("black", 1/10) makes them 10% opaque. These variations help manage point overlap and enhance plot clarity, especially in dense datasets

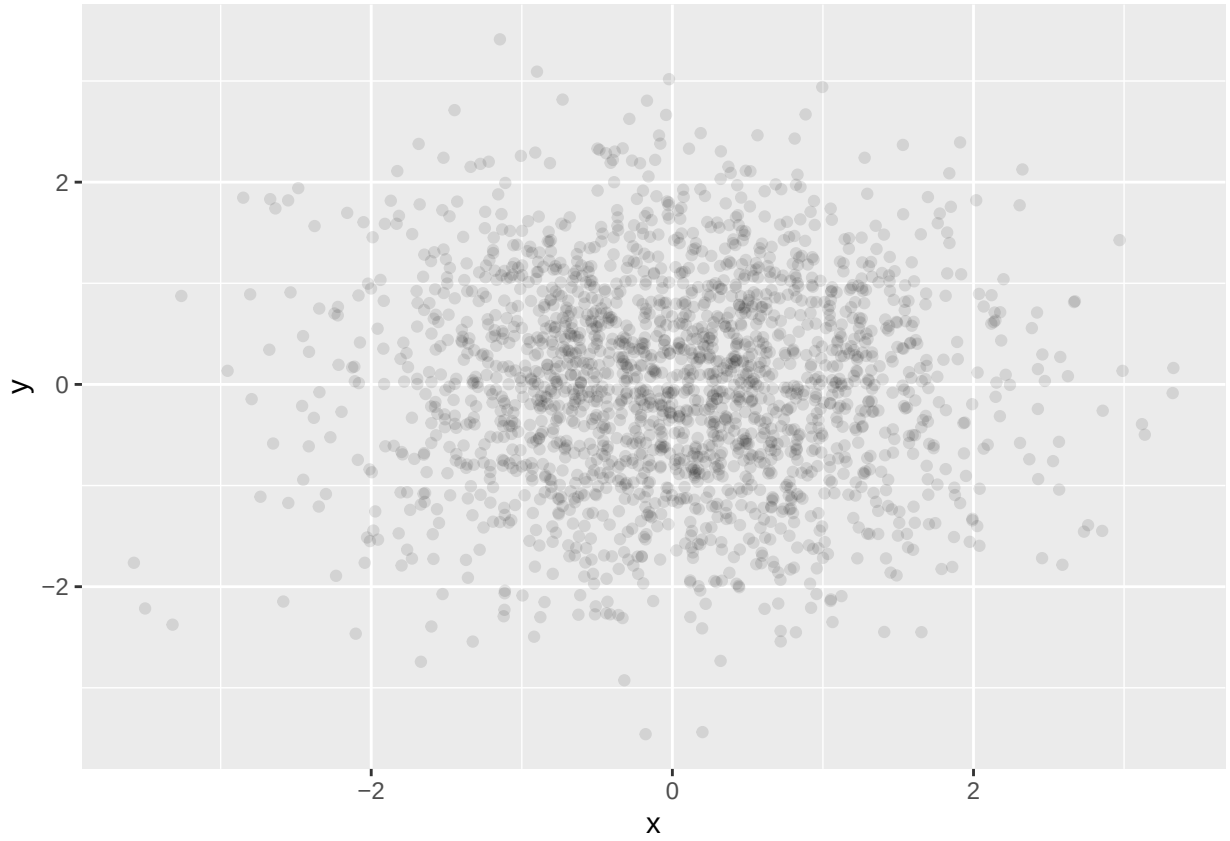
```
norm+geom_point(colour=alpha("black",1/3))
```




```
norm+geom_point(colour=alpha("black",1/5))
```



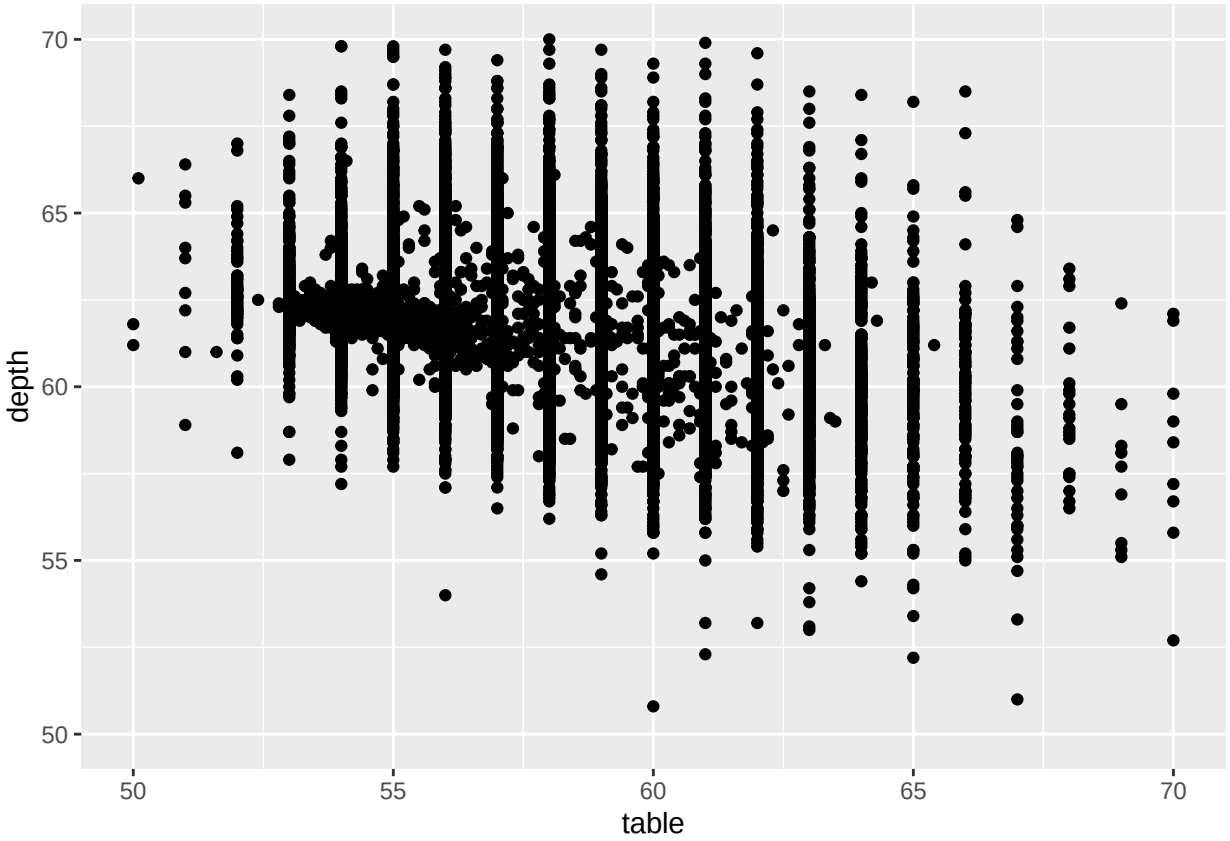

```
norm+geom_point(colour=alpha("black",1/10))
```



The code creates a scatter plot of table versus depth from the diamonds dataset, first displaying it with default points, then applying jittering to reduce overlap. It further adjusts the plot by varying the transparency of the points, from 10% to 0.5% opacity, to make overlapping points more distinguishable and reveal the underlying data patterns.

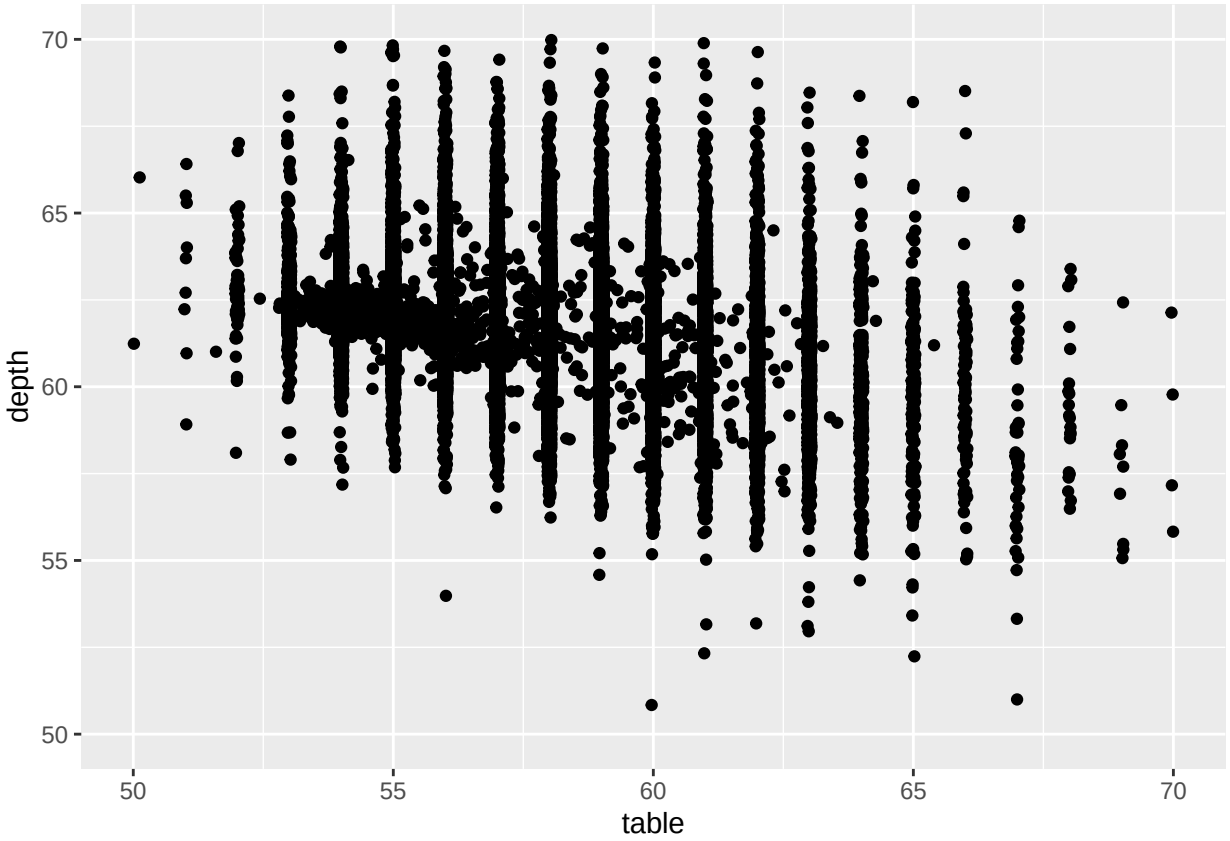
```
td <- ggplot(diamonds, aes(table, depth)) +
xlim(50, 70) + ylim(50, 70)
td + geom_point()
```

```
## Warning: Removed 36 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



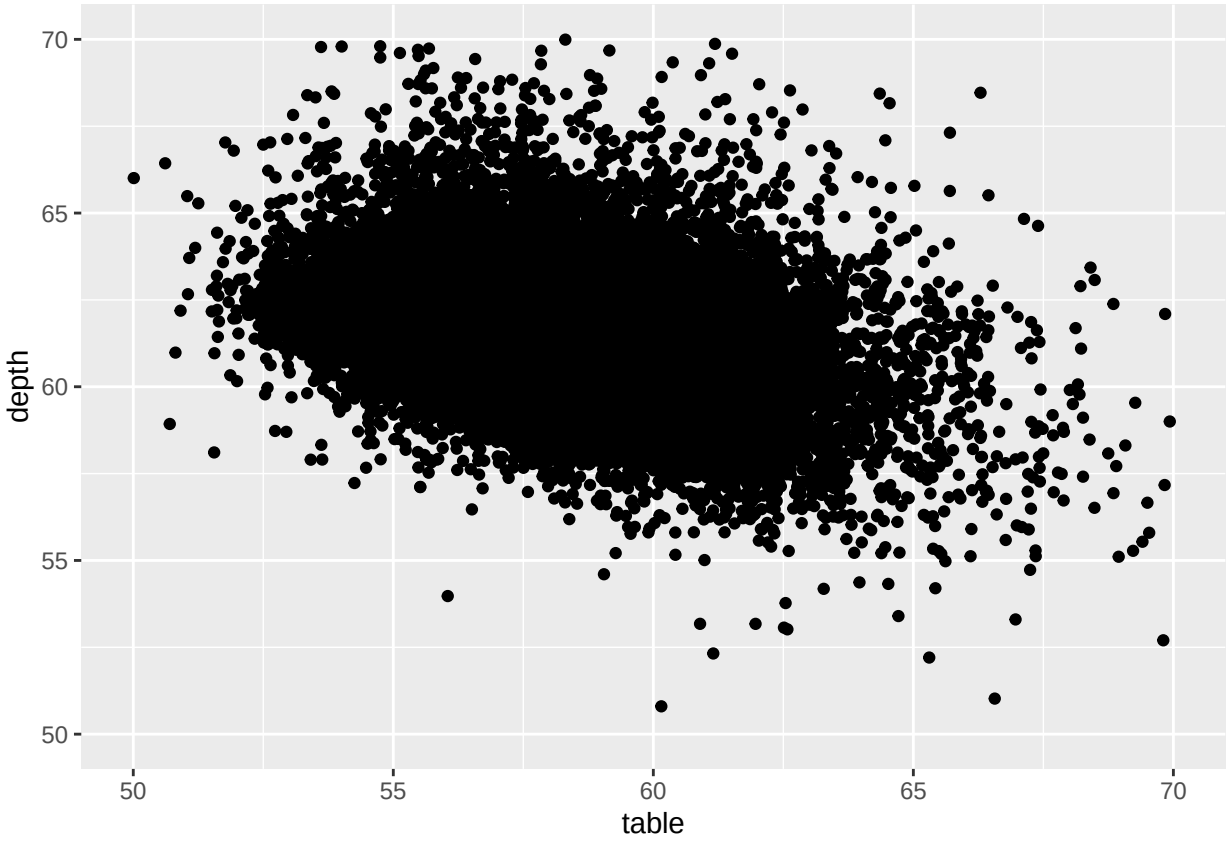

```
td + geom_jitter()
```

```
## Warning: Removed 42 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



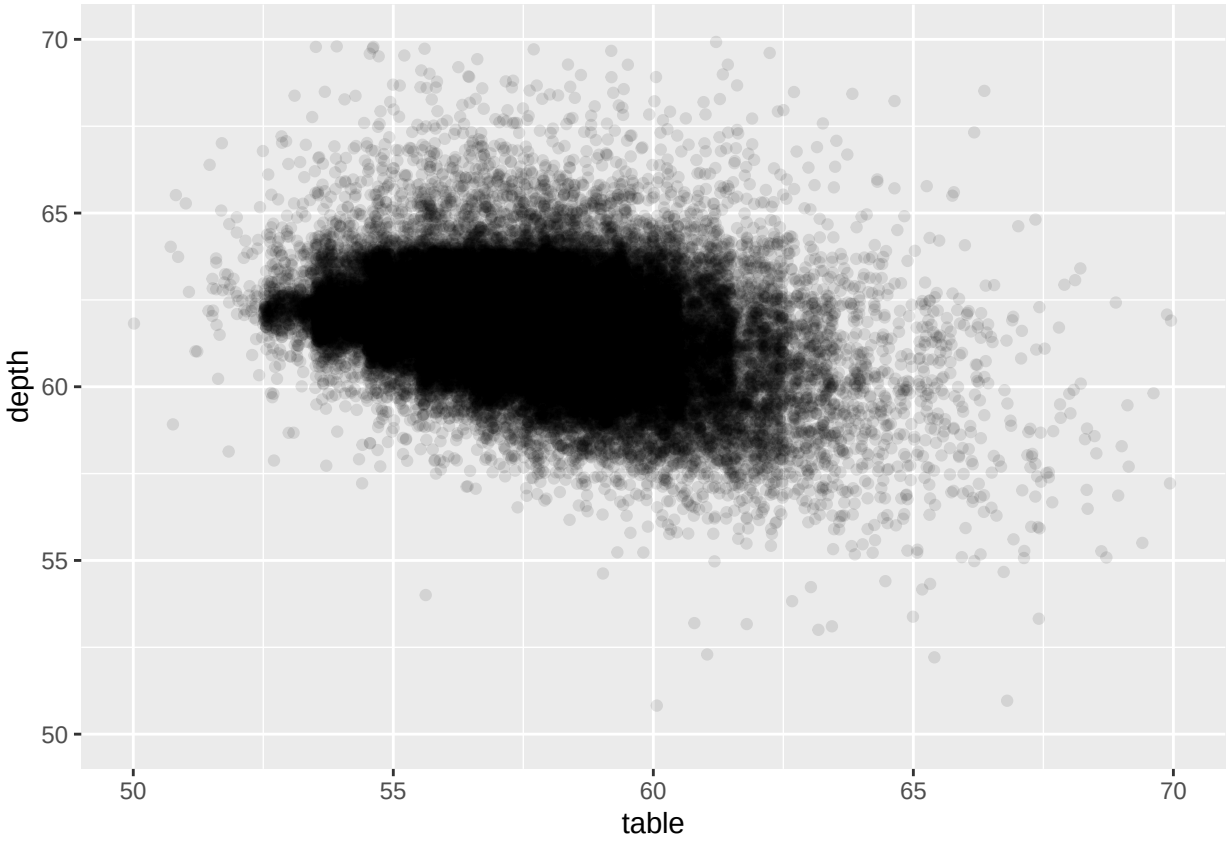

```
jit <- position_jitter(width = 0.5)
td + geom_jitter(position = jit)
```

```
## Warning: Removed 41 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



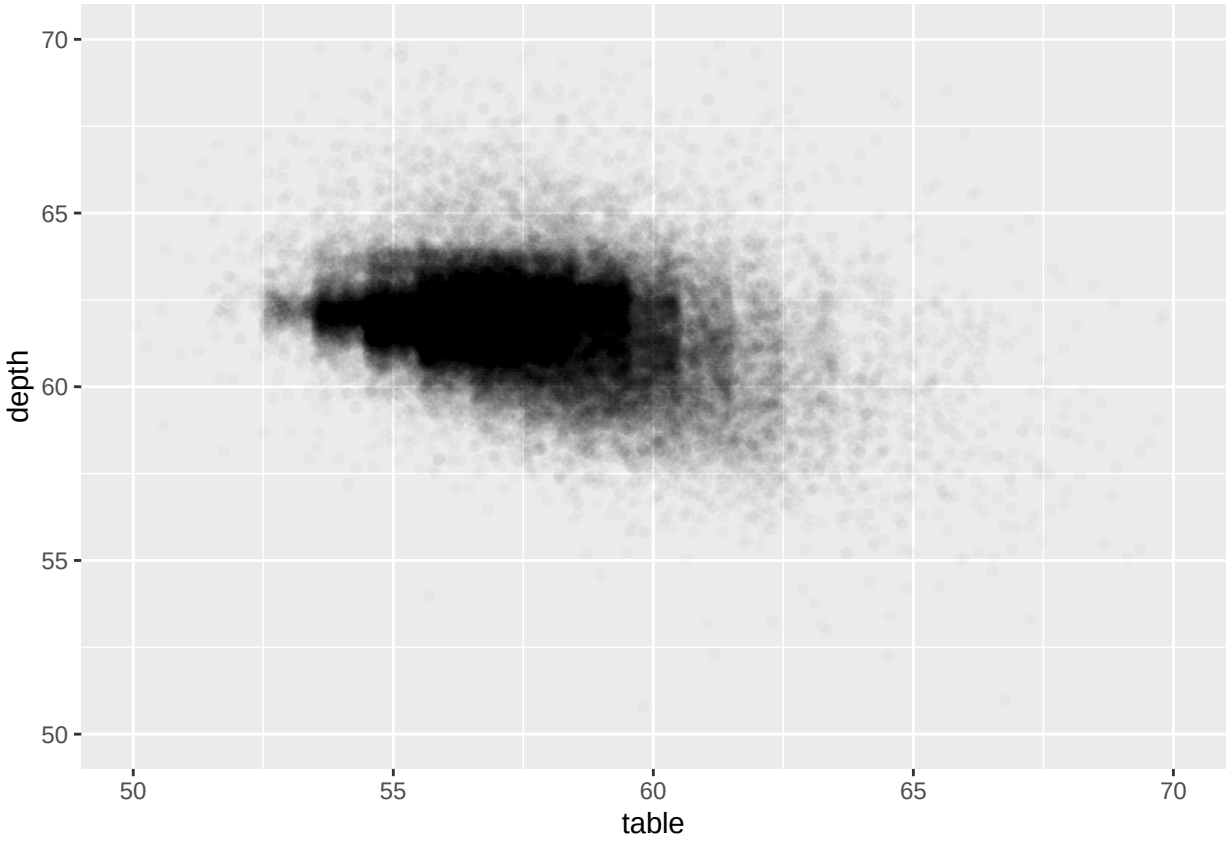

```
td + geom_jitter(position = jit, colour = alpha("black", 1/10))
```

```
## Warning: Removed 44 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



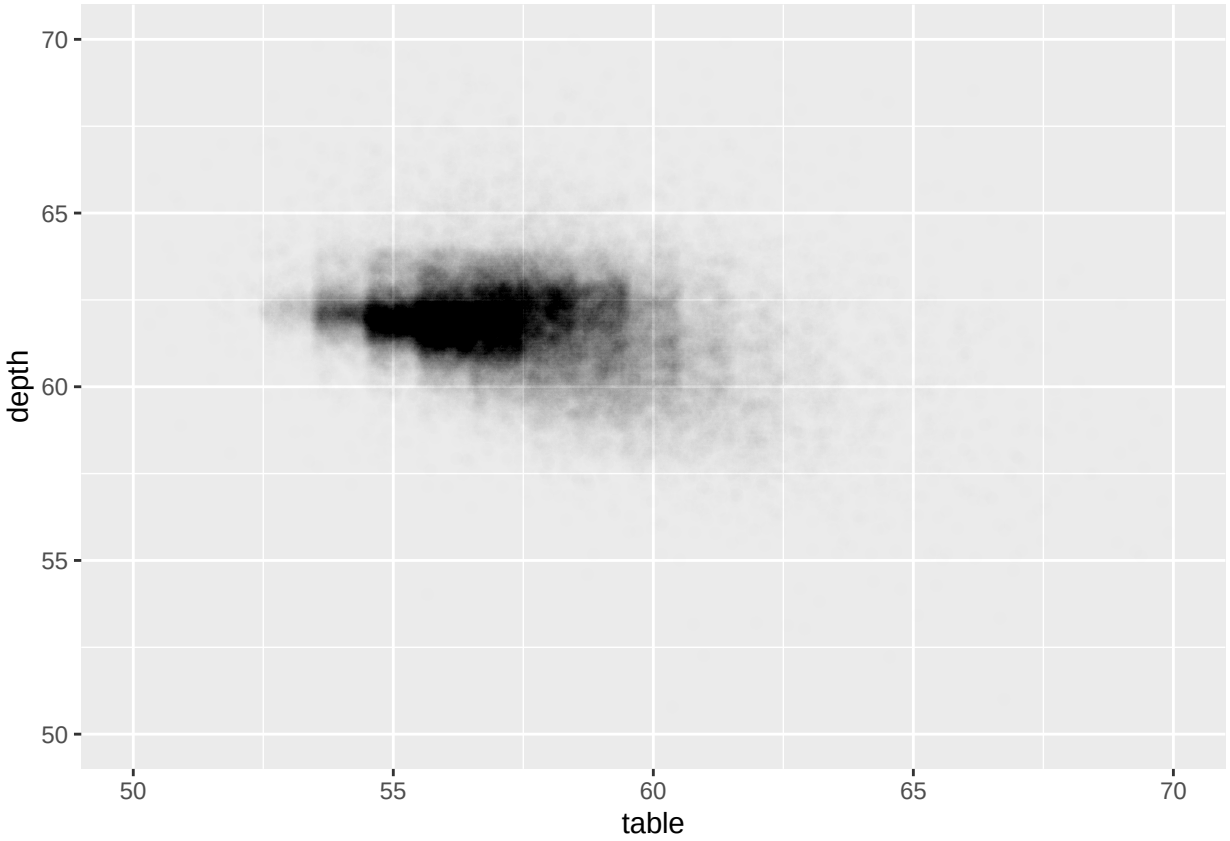

```
td + geom_jitter(position = jit, colour = alpha("black", 1/50))
```

```
## Warning: Removed 42 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
td + geom_jitter(position = jit, colour = alpha("black", 1/200))
```

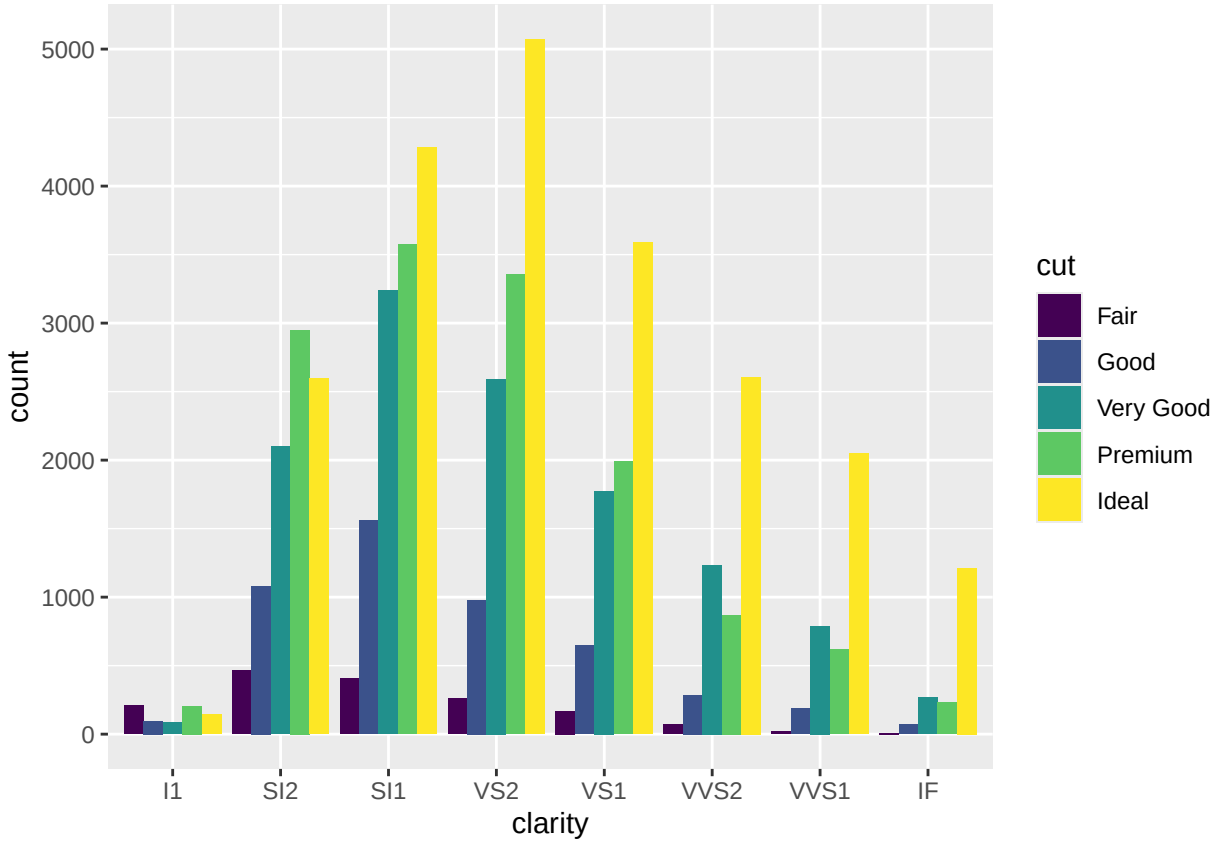
```
## Warning: Removed 43 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



#Position adjustments

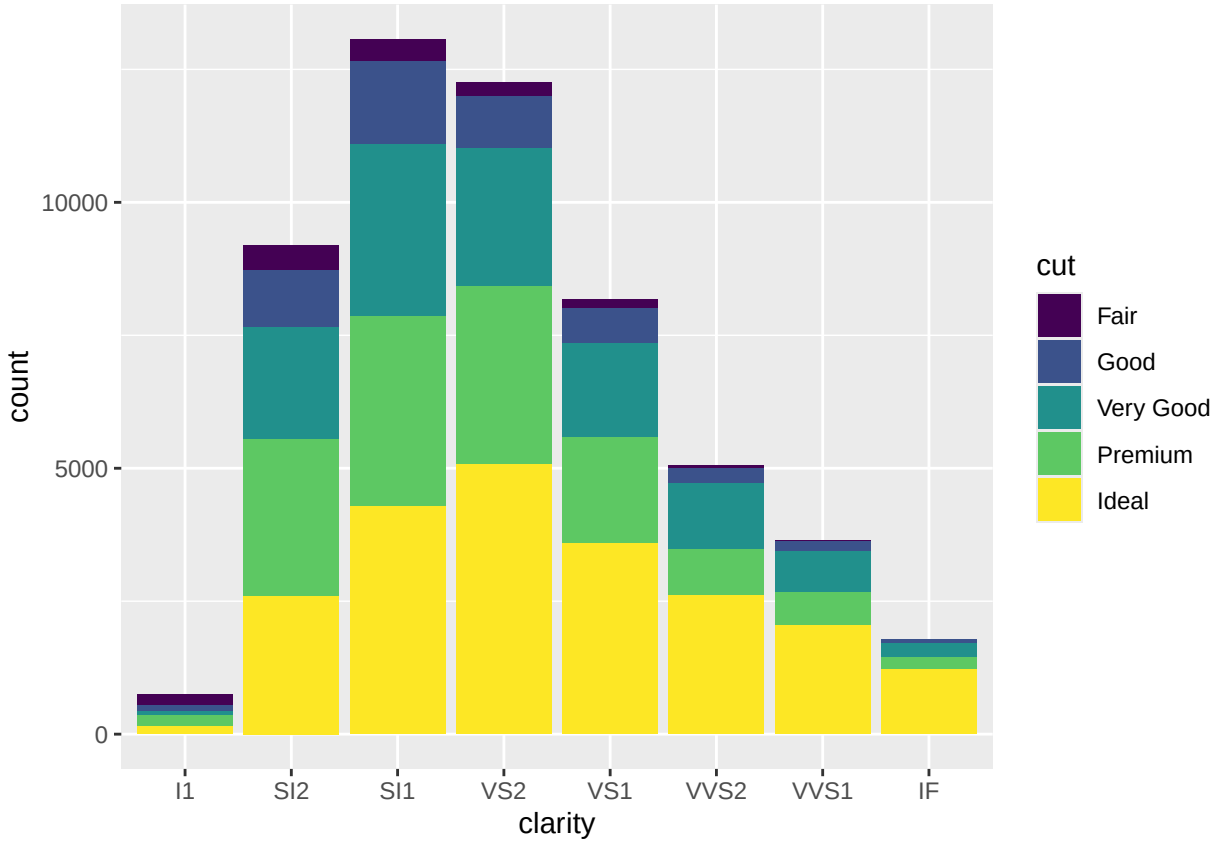
the plot reveals that Most diamonds have a high cut quality, with a smaller portion having lower quality cuts.A few diamonds have significantly lower cut quality compared to the majority. it is clear that Higher diamond clarity is associated with higher cut quality.Higher diamond clarity is associated with higher cut quality. As clarity increases, the proportion of higher cut quality diamonds also increases.

```
depth_dist <- ggplot(diamonds, aes(clarity))
depth_dist + geom_bar(aes(fill = cut), position = "dodge")
```



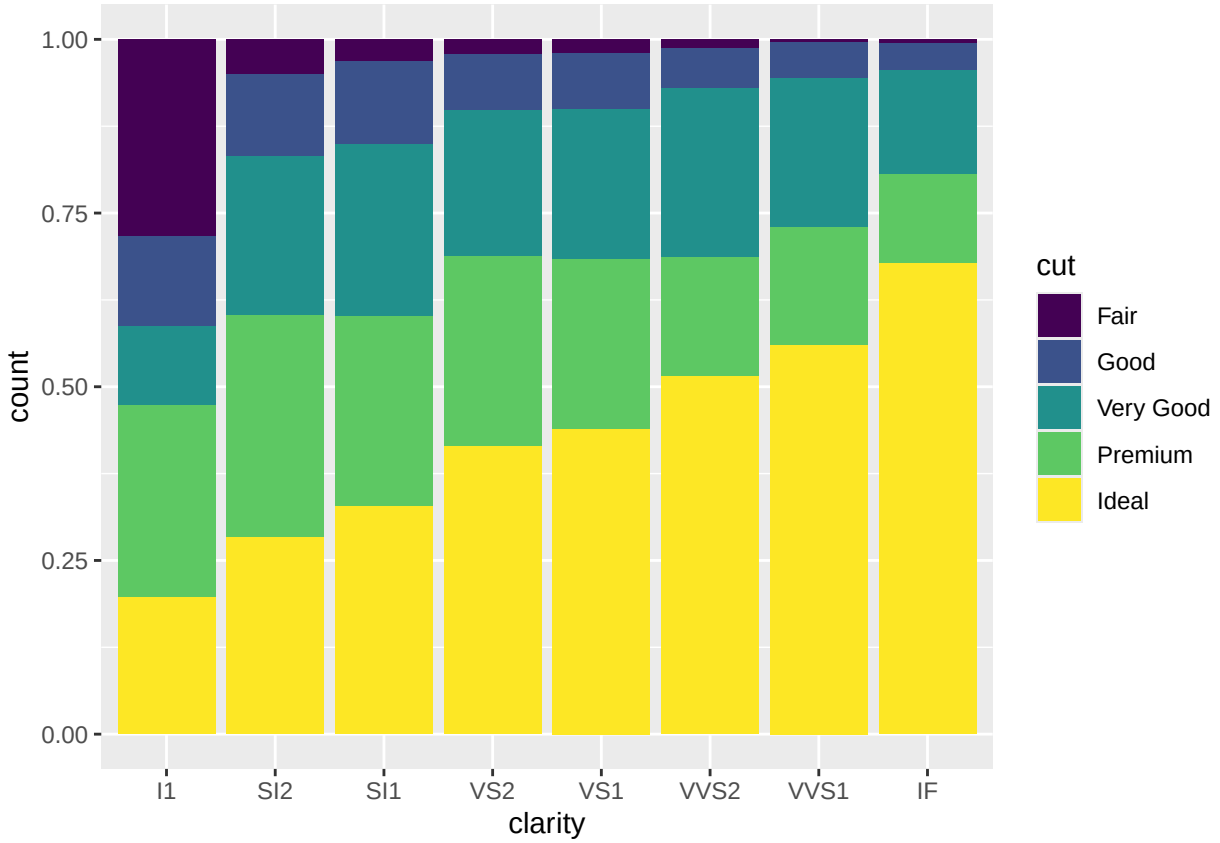
The bar chart represents the relationship between diamond clarity and cut quality. The chart reveals a strong correlation between these two factors, with higher clarity diamonds generally associated with higher cut quality categories. This suggests that diamonds with fewer inclusions are more likely to achieve superior cuts. However, the data also indicates a class imbalance, with certain clarity levels being more frequent than others, which might influence further analysis.

```
depth_dist <- ggplot(diamonds, aes(clarity))
depth_dist + geom_bar(aes(fill = cut), position = "stack")
```



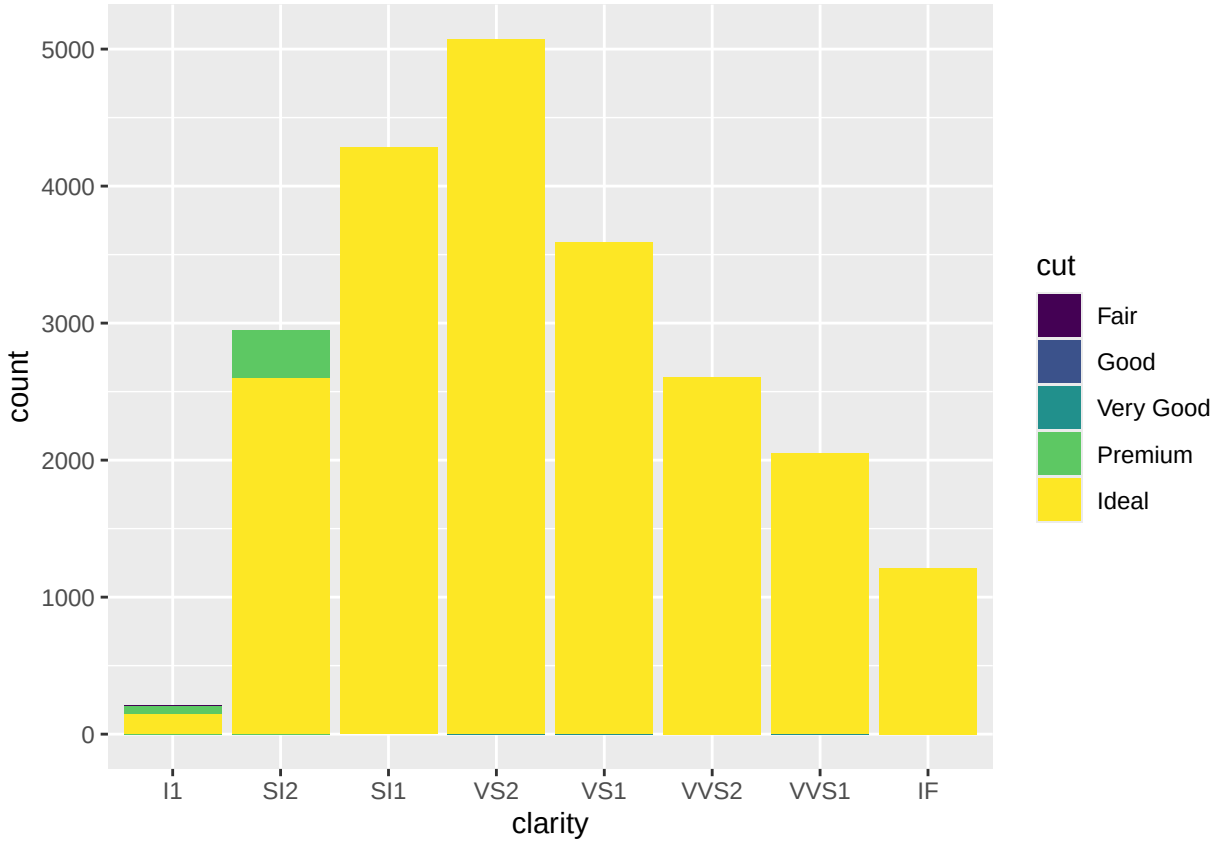
the bar chart reveals that the relationship between diamond clarity and cut quality. The chart reveals a strong correlation between these two factors, with higher clarity diamonds generally associated with higher cut quality categories. This suggests that diamonds with fewer inclusions are more likely to achieve superior cuts. However, the data also indicates a class imbalance, with certain clarity levels being more frequent than others, which might influence further analysis.

```
depth_dist <- ggplot(diamonds,aes(clarity))
depth_dist + geom_bar(aes(fill=cut),position = "fill")
```



the stacked bar chart effectively displays the relationship between two categorical variables: diamond clarity and cut quality. the chart illustrates a strong correlation between diamond clarity and cut quality. Diamonds with higher clarity levels tend to have a higher proportion of superior cuts (Ideal, Premium). While Ideal cut is the most common across all clarity levels, its dominance is more pronounced in higher clarity categories. However, the data also shows an imbalance in the number of diamonds per clarity level, which might impact overall analysis.

```
depth_dist <- ggplot(diamonds, aes(clarity))
depth_dist + geom_bar(aes(fill = cut), position = "identity")
```

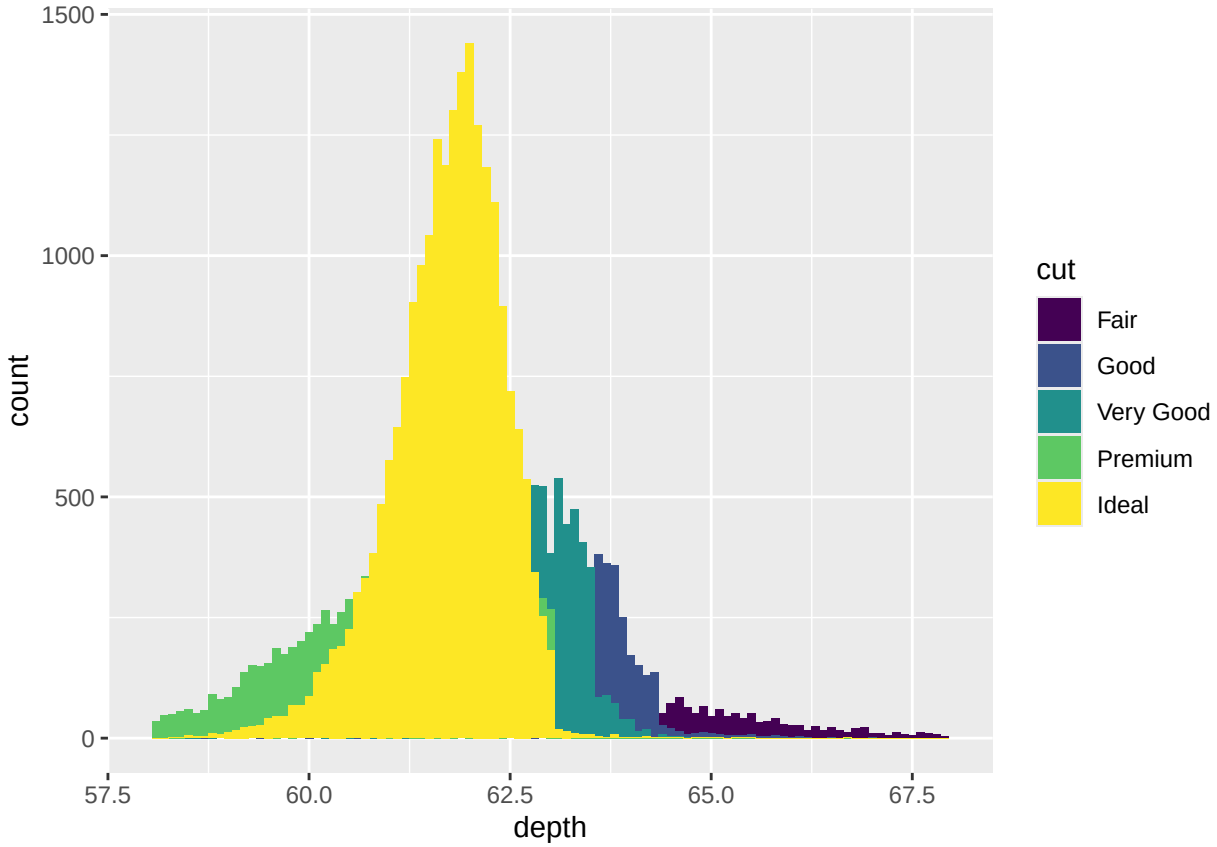


this code creates a histogram of the depth variable in the diamonds dataset, with the bars filled according to the cut category and grouped into bins of width 0.1 within the x-axis range of 58 to 68. histogram reveals a strong correlation between clarity and cut quality, with higher clarity diamonds exhibiting superior cuts. However, diamond depth, while showing some variation across cut categories, is not a reliable predictor of cut quality on its own.

```
depth_dist <- ggplot(diamonds, aes(depth)) + xlim(58, 68)
depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,
position = "identity")
```

```
## Warning: Removed 669 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```

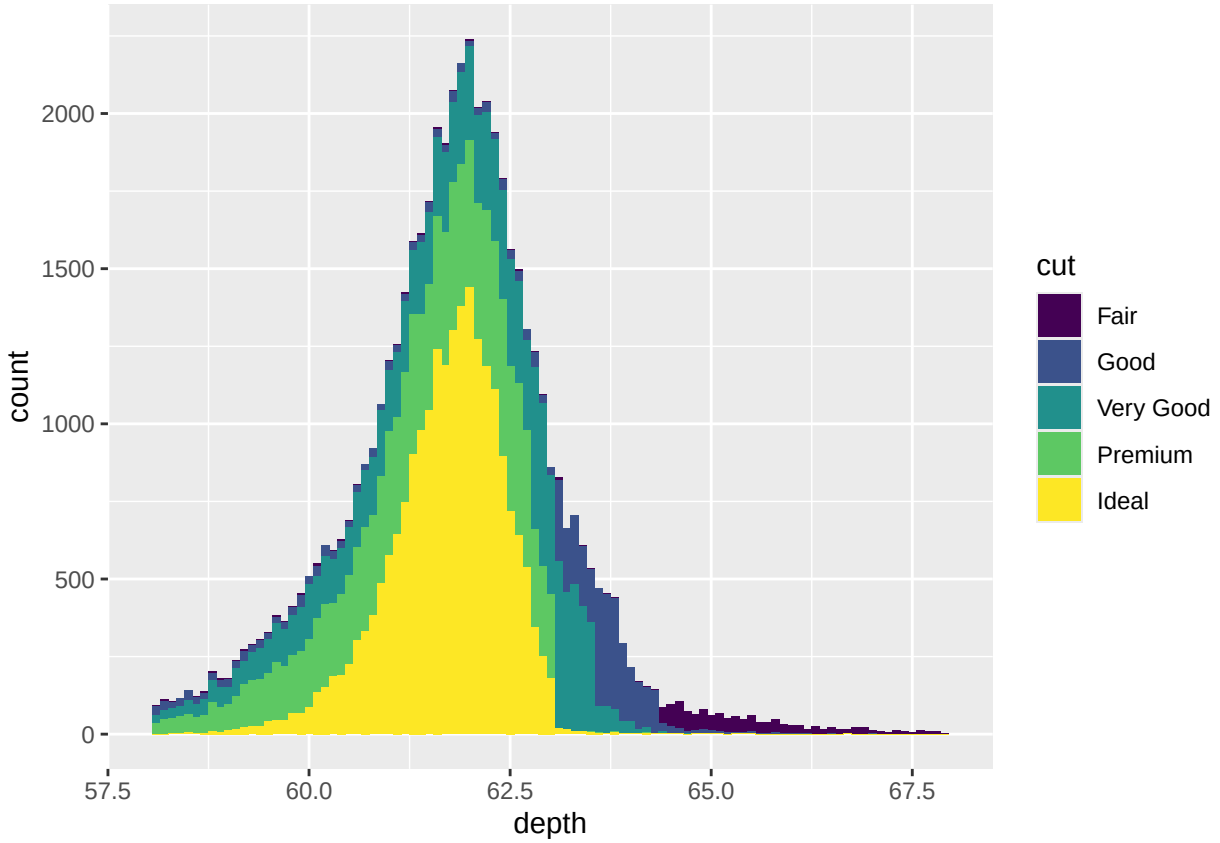


the code creates a stacked histogram of the depth variable in the diamonds dataset, with the bars filled according to the cut category. The bars are grouped into bins of width 0.1 and stacked on top of each other within the x-axis range of 58 to 68. This visualization shows the distribution of depth values while also displaying the proportion of each cut category within each bin.

```
depth_dist <- ggplot(diamonds, aes(depth)) + xlim(58, 68)
depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,
position = "stack")
```

```
## Warning: Removed 669 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```

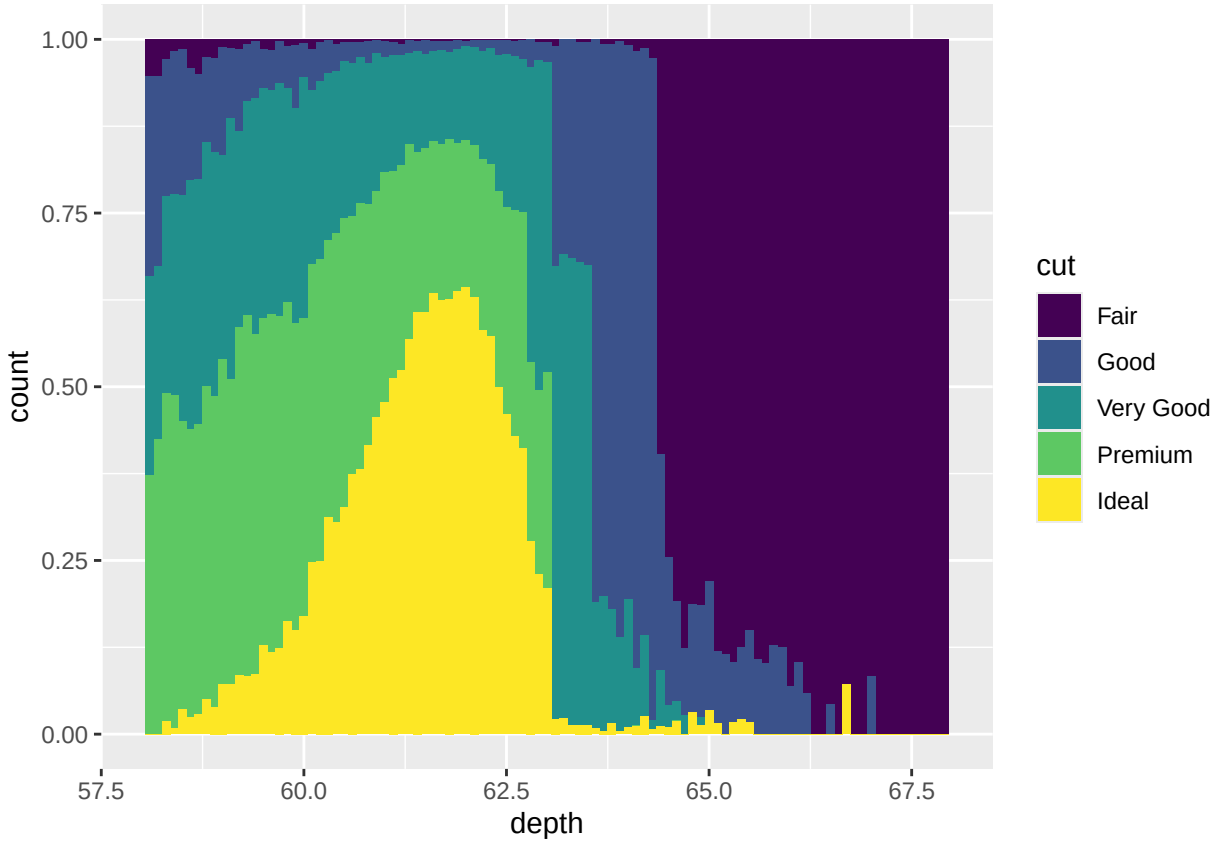


the code creates a filled histogram of the depth variable in the diamonds dataset, with the bars filled according to the cut category. The bars are grouped into bins of width 0.1 and stacked on top of each other, normalized to show the proportion of each cut category within each bin. The x-axis is limited to the range 58 to 68. This visualization highlights the relative distribution of cut categories within each depth bin, allowing for easy comparison of proportions across bins.

```
depth_dist <- ggplot(diamonds, aes(depth)) + xlim(58, 68)
depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,
position = "fill")
```

```
## Warning: Removed 669 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```

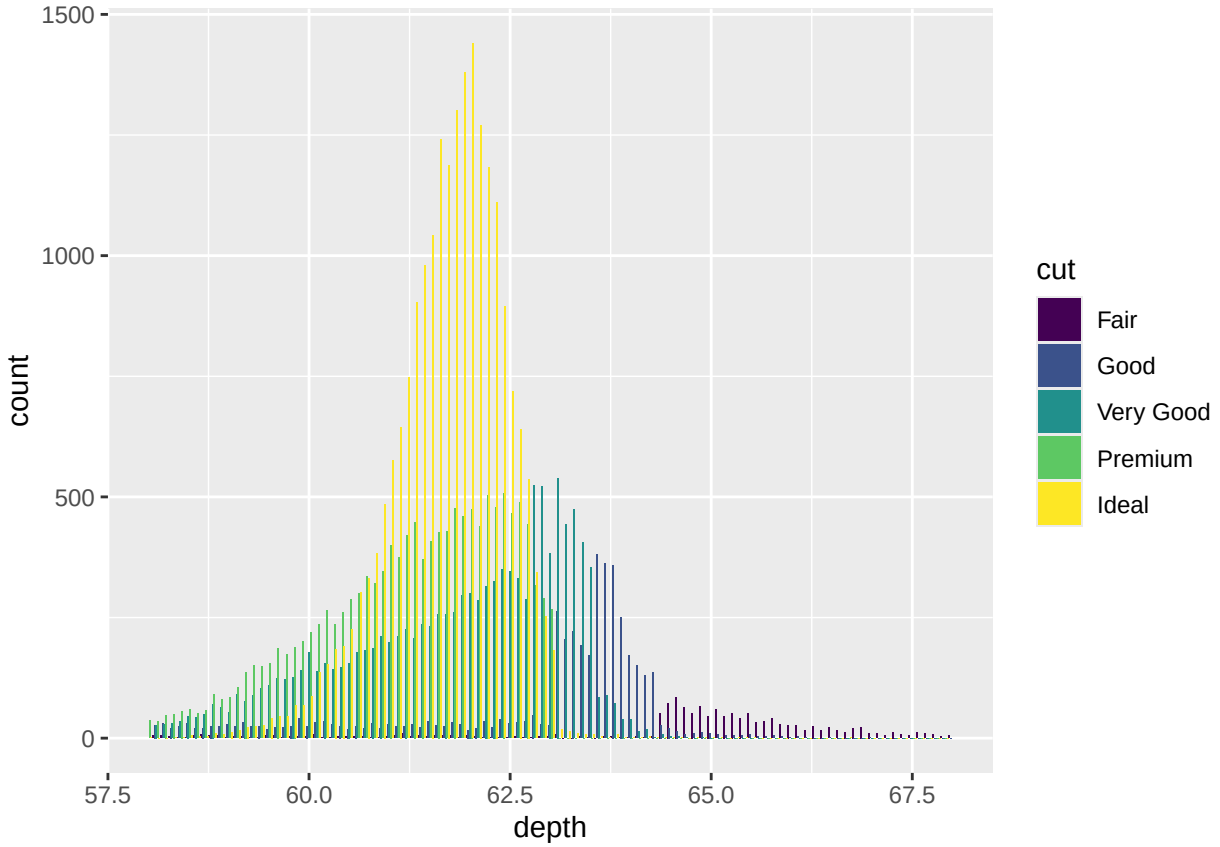


the code creates a dodged histogram of the depth variable in the diamonds dataset, with bars filled according to the cut category. The bars are grouped into bins of width 0.1 and displayed side by side for each cut category within the x-axis range of 58 to 68. This visualization allows for a direct comparison of the frequency of different cut categories within each depth bin.

```
depth_dist <- ggplot(diamonds, aes(depth)) + xlim(58, 68)
depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,
position = "dodge")
```

```
## Warning: Removed 669 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

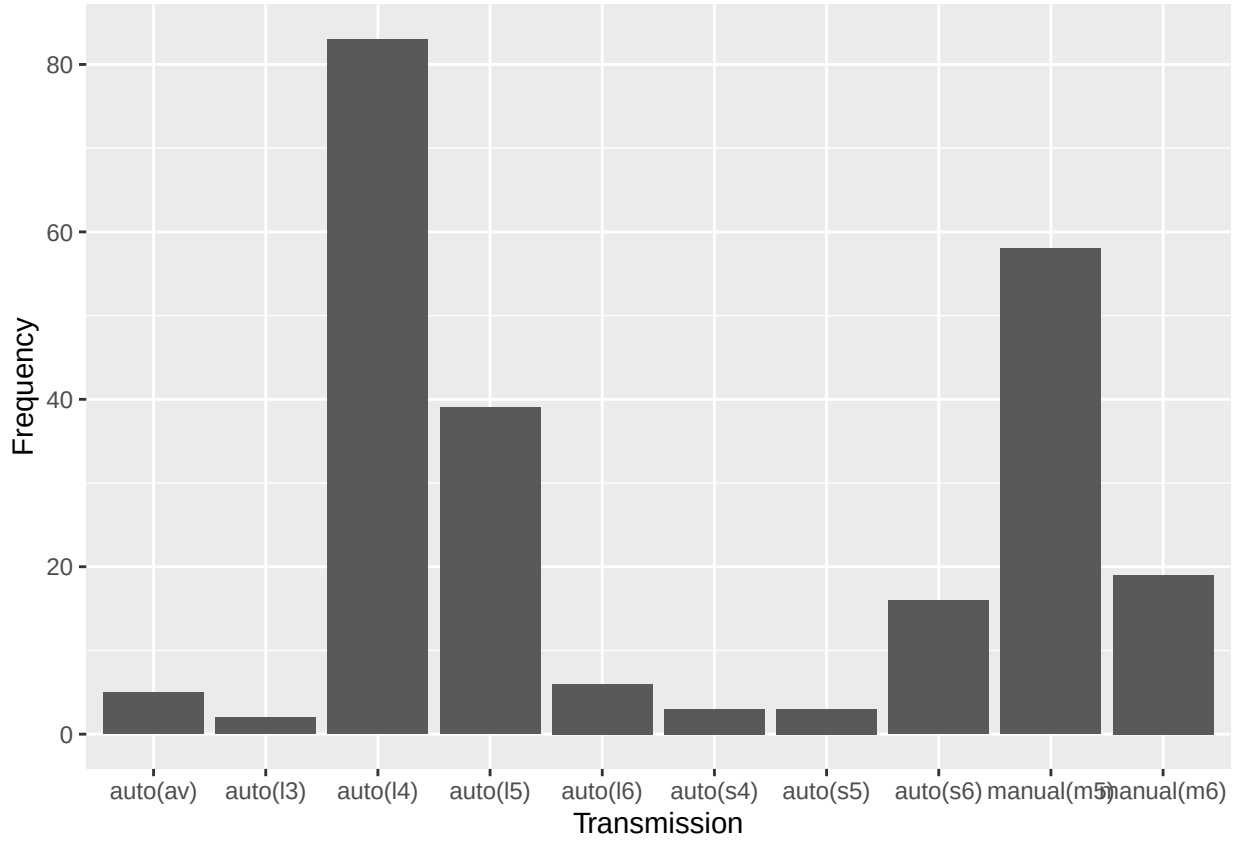
```
## Warning: Removed 6 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



#Overlapping labels

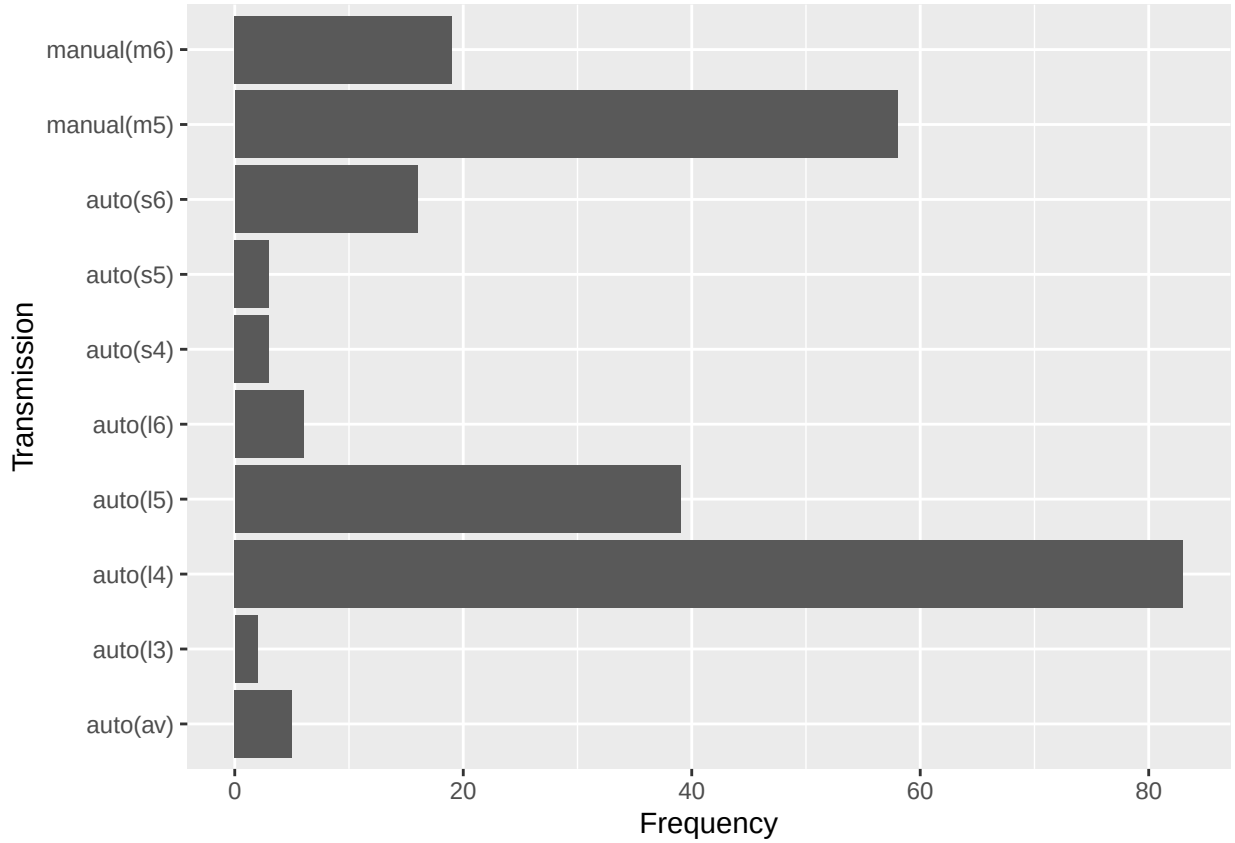
code creates a bar plot that shows the frequency of each transmission type in the mpg dataset, with the x-axis labeled “Transmission” and the y-axis labeled “Frequency

```
ggplot(mpg, aes(x = trans)) +
  geom_bar() +
  labs(x = "Transmission", y = "Frequency")
```



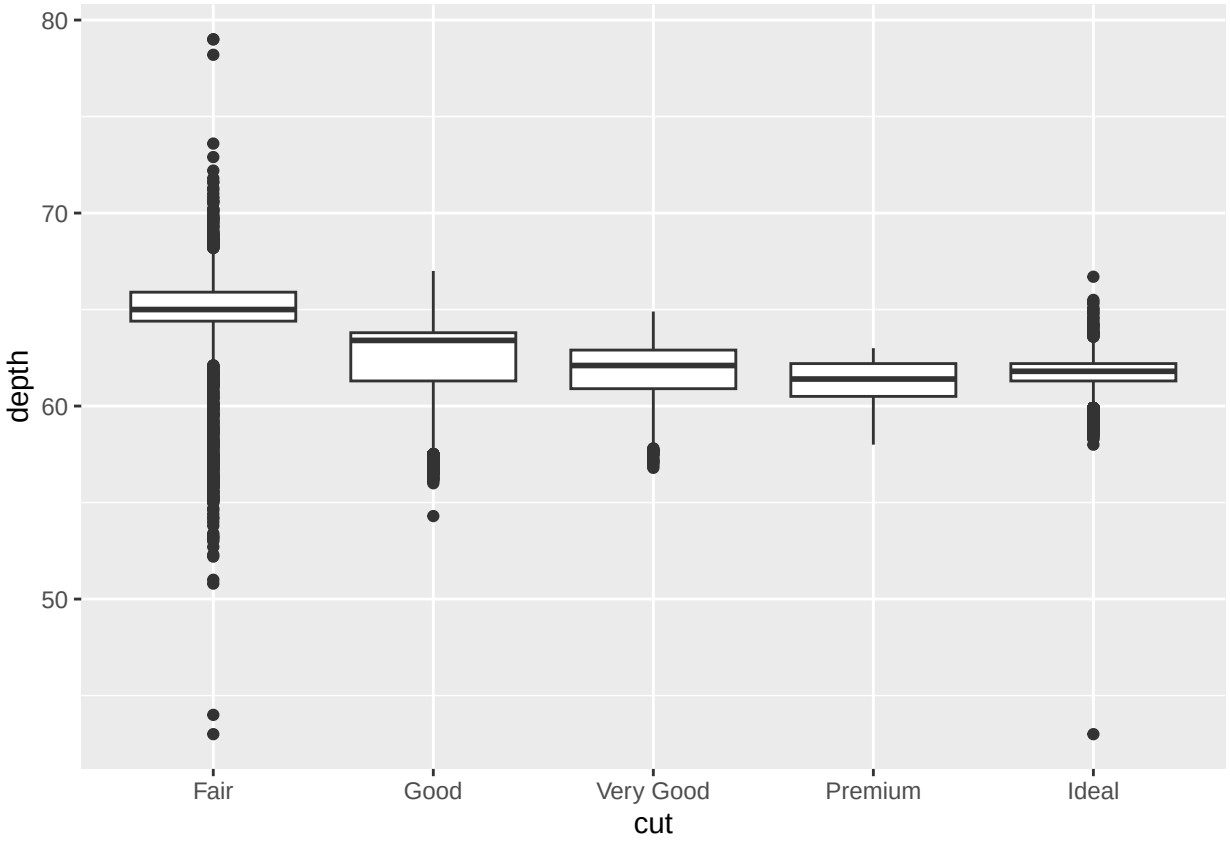
the code creates a horizontal bar plot that shows the frequency of each transmission type in the mpg dataset. The x-axis is labeled "Transmission" and the y-axis is labeled "Frequency", but due to coord_flip(), these labels will appear as "Frequency" on the horizontal axis and "Transmission" on the vertical axis. This format can make it easier to read the labels if they are long or numerous.

```
ggplot(mpg, aes(x = trans)) +
geom_bar() +
labs(x = "Transmission",
y = "Frequency") +
coord_flip()
```



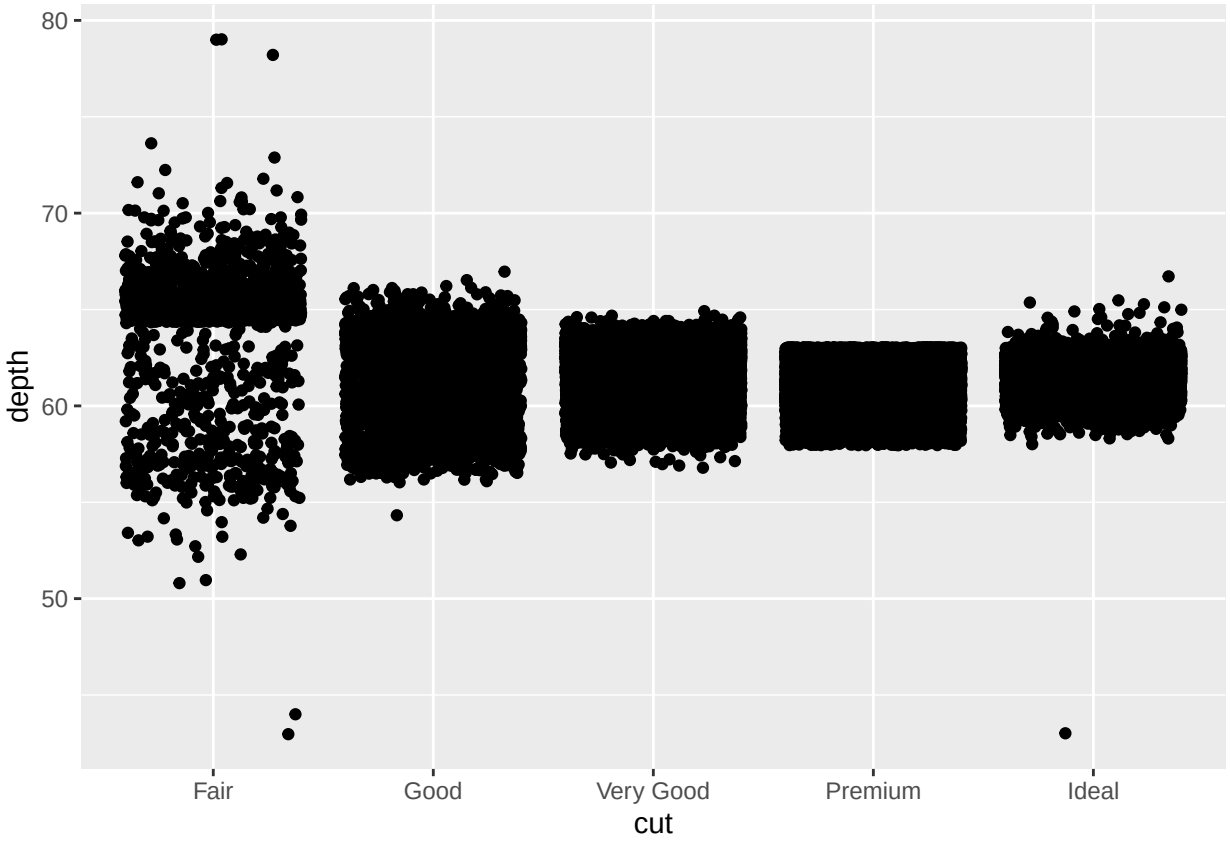
The boxplot illustrates how diamond depth varies across different cut qualities.The median depth (represented by the horizontal line within each box) generally decreases as cut quality improves, from Fair to Ideal.The spread of depth, as indicated by the box and whiskers, is relatively consistent across cut categories, with some variation.There are numerous outliers present in all cut categories, suggesting the presence of diamonds with unusually high or low depths for their respective cuts. Overall, while there is a slight trend of decreasing median depth with improving cut quality, the overlap in depth distributions between different cut categories is substantial. This suggests that depth alone is not a strong indicator of diamond cut quality.

```
ggplot(diamonds,
aes(x = cut,
y = depth)) +
geom_boxplot()
```



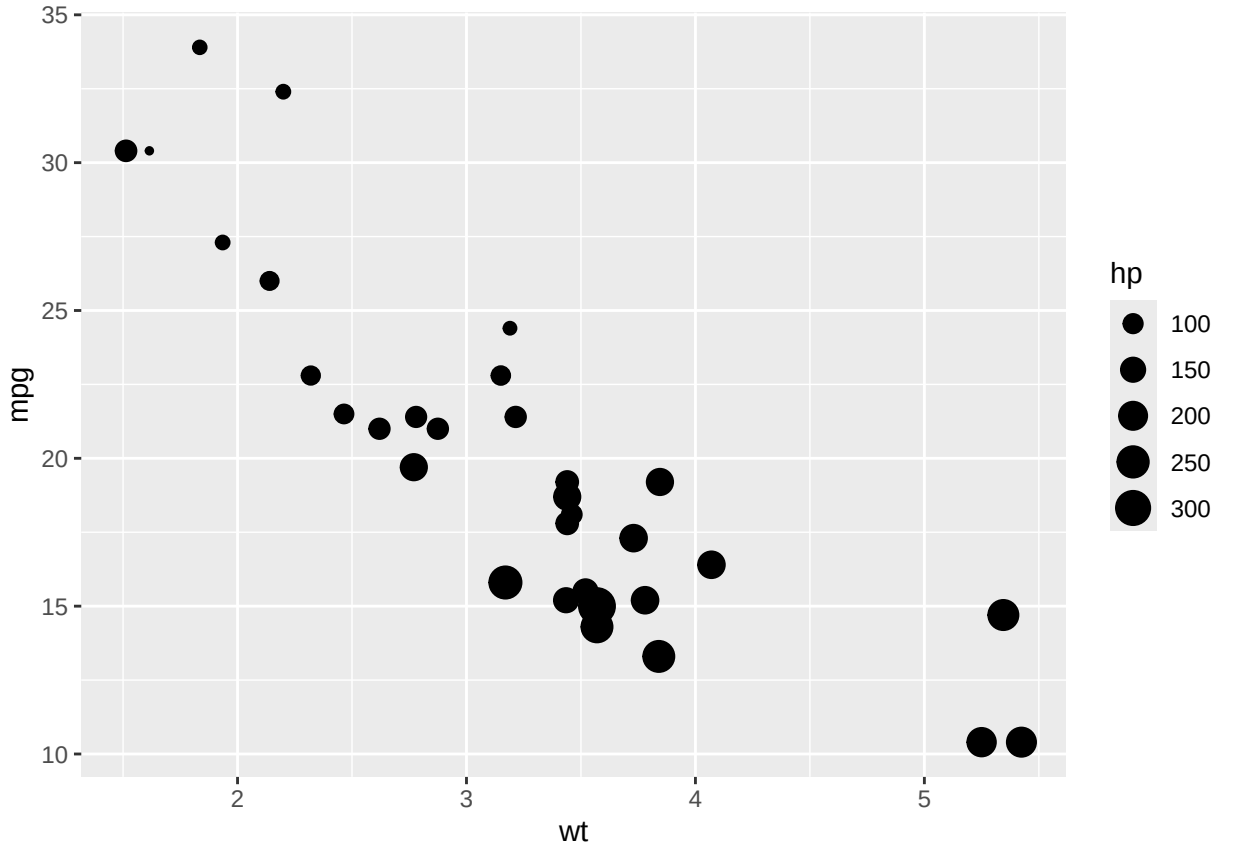
the code creates a jitter plot that shows the relationship between the diamond cut quality (on the x-axis) and the depth percentage (on the y-axis). The jittering helps to spread out the points horizontally and vertically, reducing overlap and making the distribution of data points more visible.

```
ggplot(diamonds,
aes(x = cut,
y = depth)) +
geom_jitter()
```



code creates a scatter plot showing the relationship between car weight (wt) and fuel efficiency (mpg) in the mtcars dataset. The size of each point represents the car's horsepower (hp), providing a third dimension of information in the plot. This allows you to see how car weight and fuel efficiency are related, and how horsepower varies among the cars the plot reveals a general trend of decreasing miles per gallon (mpg) as weight (wt) increases. This indicates that heavier cars tend to have lower fuel efficiency. The size of the points, representing horsepower (hp), suggests that cars with higher horsepower generally weigh more and have lower fuel efficiency. However, there is considerable variation in the data, with some cars deviating from this general trend.

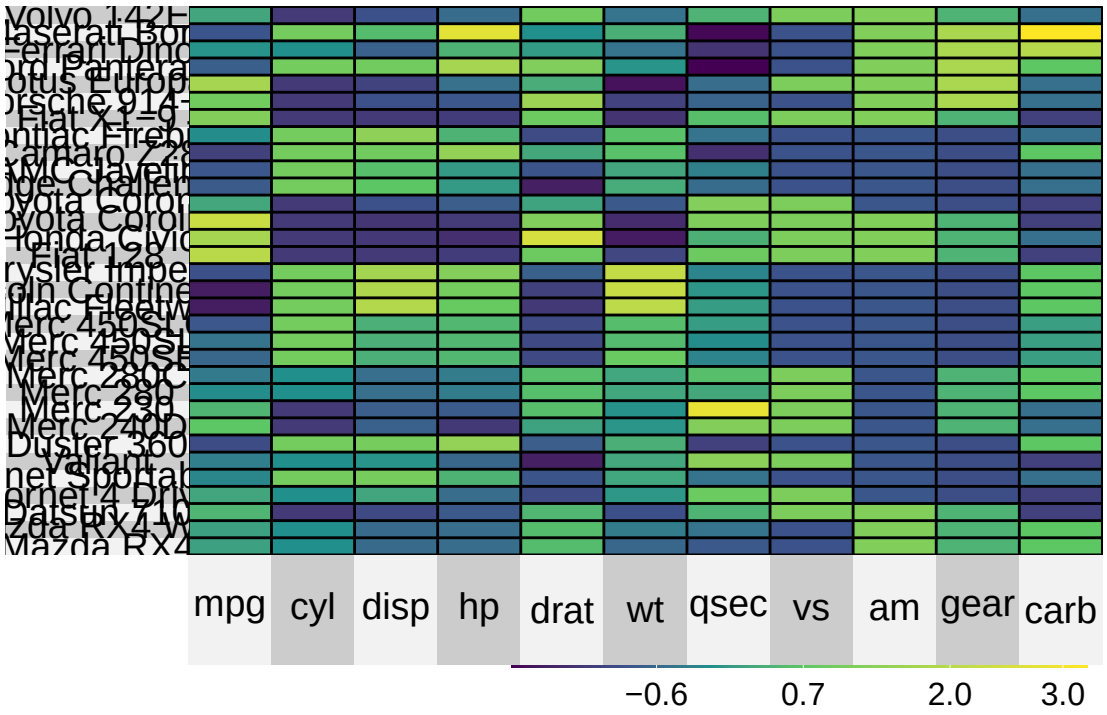
```
data(mtcars)
ggplot(mtcars,
aes(x = wt, y = mpg, size = hp)) +
geom_point()
```




```
#install.packages("superheat")
library(superheat)
```

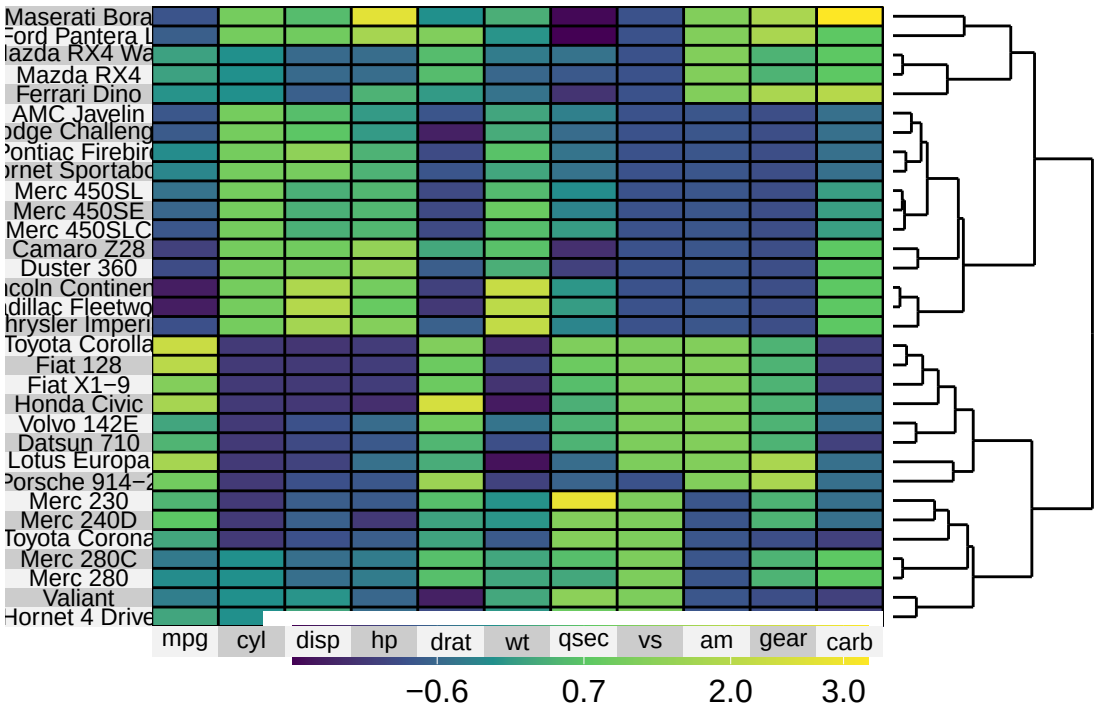
the code generates a heatmap of the mtcars dataset, with standardized data, and includes customized label sizes for both rows and columns. It also includes a dendrogram for rows to show clustering, which helps in understanding the similarities among the rows based on their scaled values.

```
superheat(mtcars, scale = TRUE)
```



The provided visualization is a heatmap with dendrograms on both the rows and columns. This type of plot is often used to visualize relationships between multiple variables. The dendrograms on the sides attempt to group similar cars based on their attribute values.Some attributes appear to be correlated, as evidenced by similar color patterns across rows or columns. For instance, there might be a correlation between horsepower (hp) and weight (wt), as they show similar color patterns.Some cars have distinct color patterns, suggesting they have unique combinations of attributes.

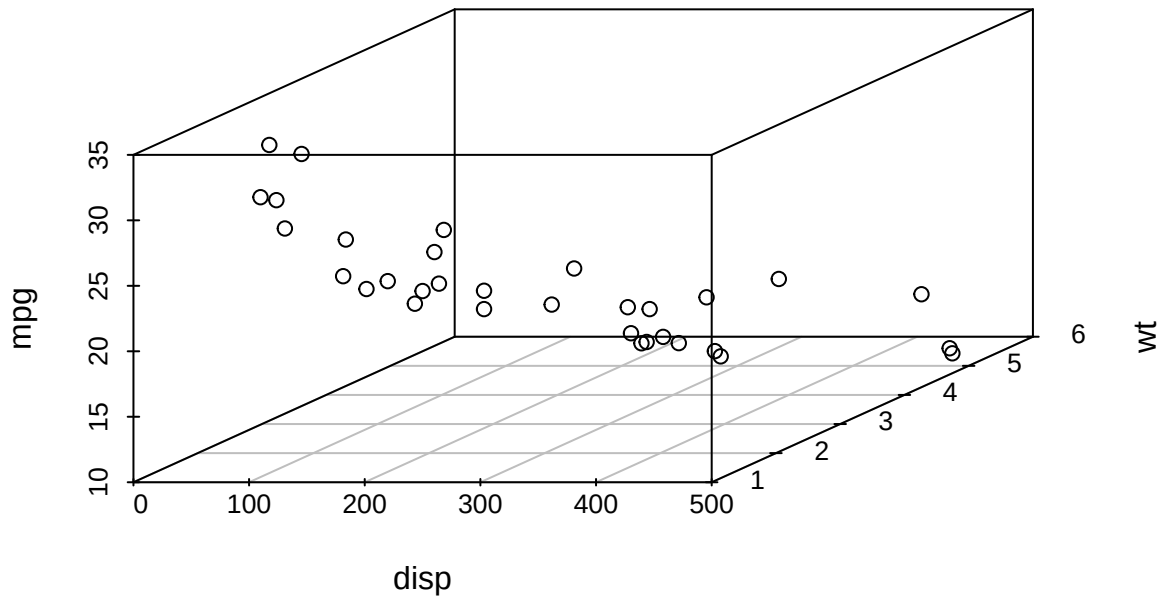
```
superheat(mtcars,
scale = TRUE,
left.label.text.size=3,
bottom.label.text.size=3,
bottom.label.size = .05,
row.dendrogram = TRUE )
```



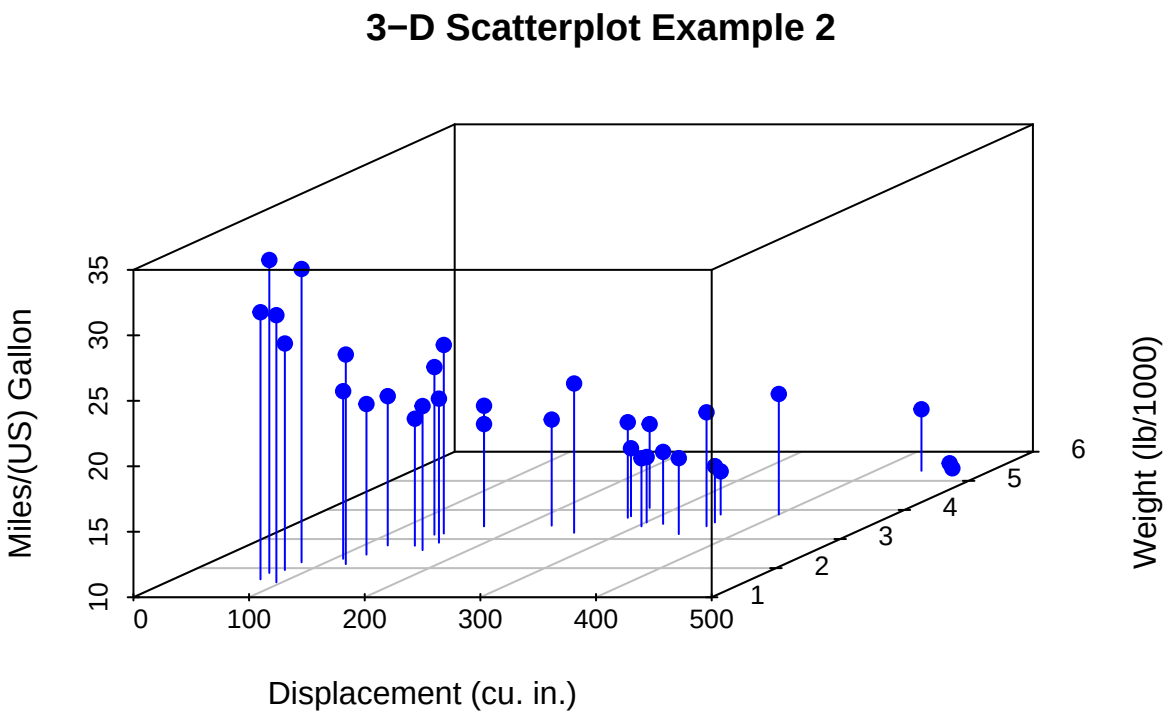

```
#install.packages("scatterplot3d")
library(scatterplot3d)
```

the code generates a 3D scatter plot with disp on the x-axis, wt on the y-axis, and mpg on the z-axis. The plot provides a visual representation of the relationship between these three variables in the mtcars dataset, with a title "3-D Scatterplot Example 1"

```
with(mtcars, {
scatterplot3d(x = disp,
y = wt,
z = mpg,
main="3-D Scatterplot Example 1")
})
```




```
with(mtcars, {
scatterplot3d(x = disp,
y = wt,
z = mpg,
# filled blue circles
color="blue",
pch=19,
# lines to the horizontal plane
type = "h",
main = "3-D Scatterplot Example 2",
xlab = "Displacement (cu. in.)",
ylab = "Weight (lb/1000)",
zlab = "Miles/(US) Gallon")
})
```



#Combining geoms and stats

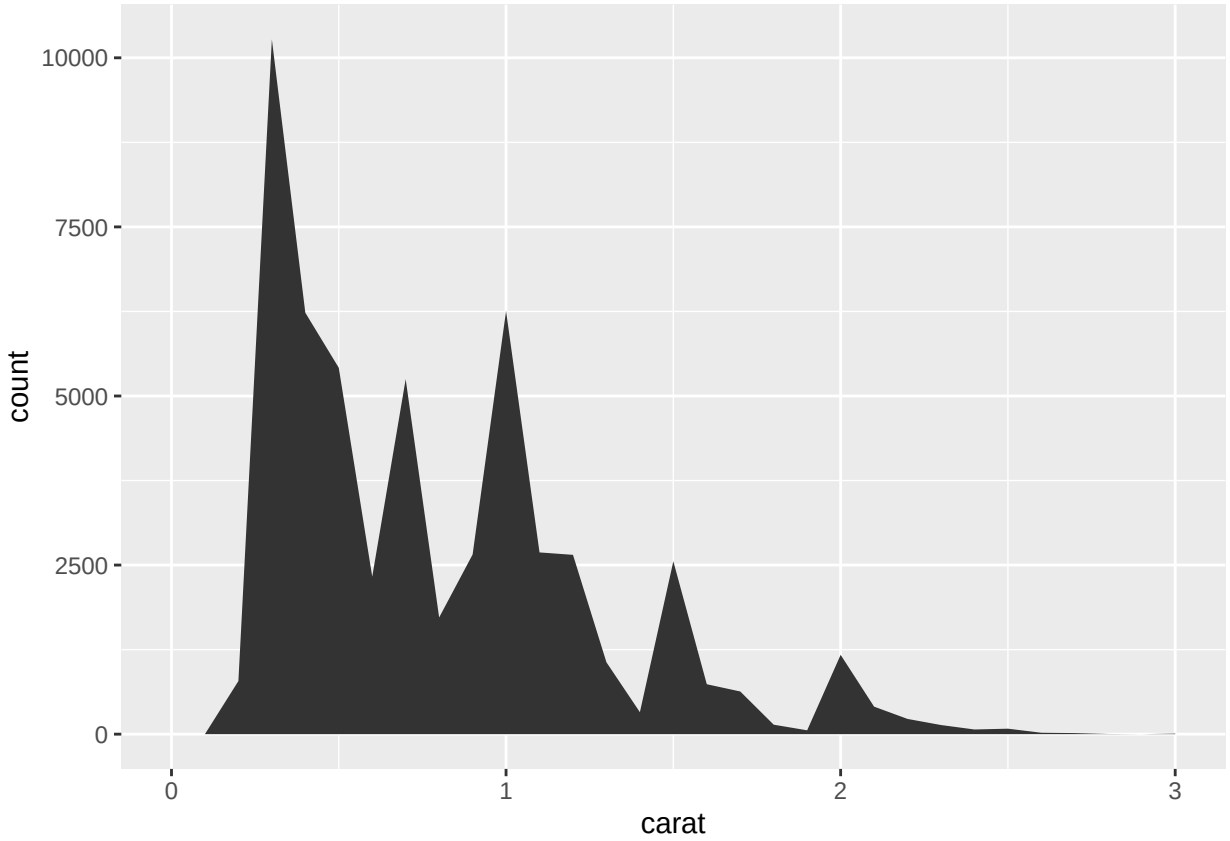
the code creates an area plot that shows the distribution of the carat variable for diamonds. The plot bins the data into intervals of 0.1 carat and fills the area under the histogram bars to represent the count of diamonds in each bin. The x-axis is limited to a range of 0 to 3 carats.

```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(ymax = ..count..), binwidth = 0.1, geom = "area")
```

```
## Warning in stat_bin(aes(ymax = ..count..), binwidth = 0.1, geom = "area"):
## Ignoring unknown aesthetics: ymax
```

```
## Warning: The dot-dot notation (‘..count..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(count)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: Removed 32 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```


10000
7500
5000
2500
0
count
0
1
2
3
carat