Data Visualization Practice

Answer Key

Overview

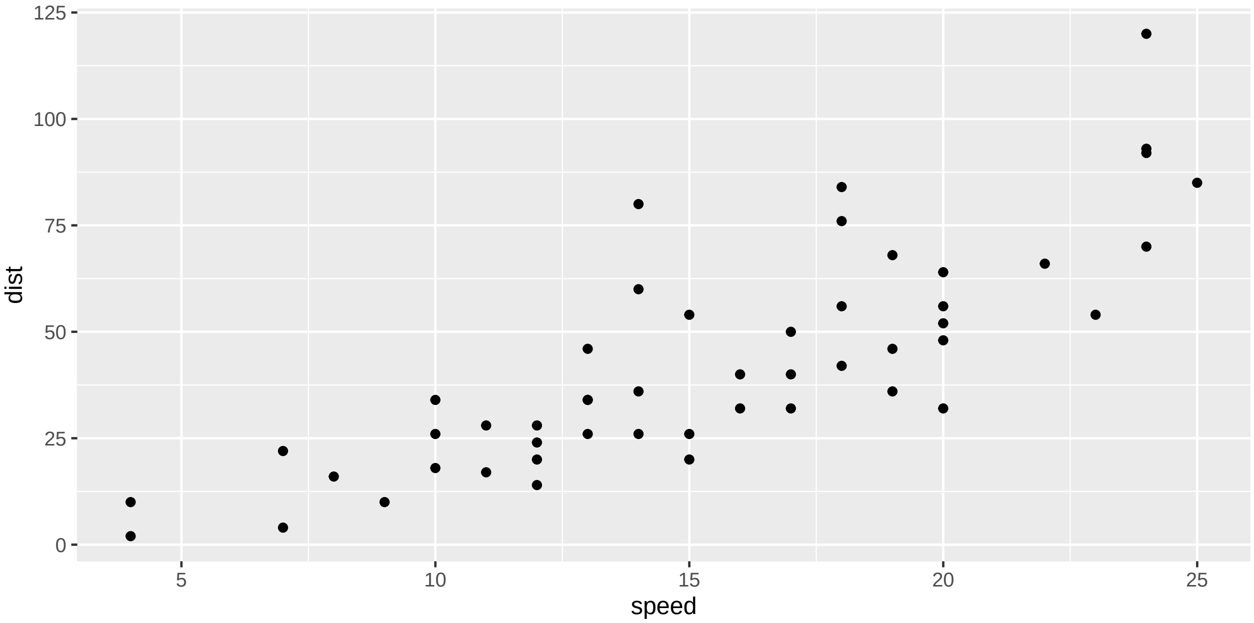
The following practice will focus entirely on plotting. As per usual, we'll use in-house data to plot various event occurrences. We'll draw on multiple datasets rather than just sticking to one, as we've done in the past.

Please use `ggplot2` for *all* plots.

Question 1

Using the `cars` dataset, plot the relationship as points with `speed` on the x-axis and `dist` on the y-axis. Save the plot as an object named `plt1`

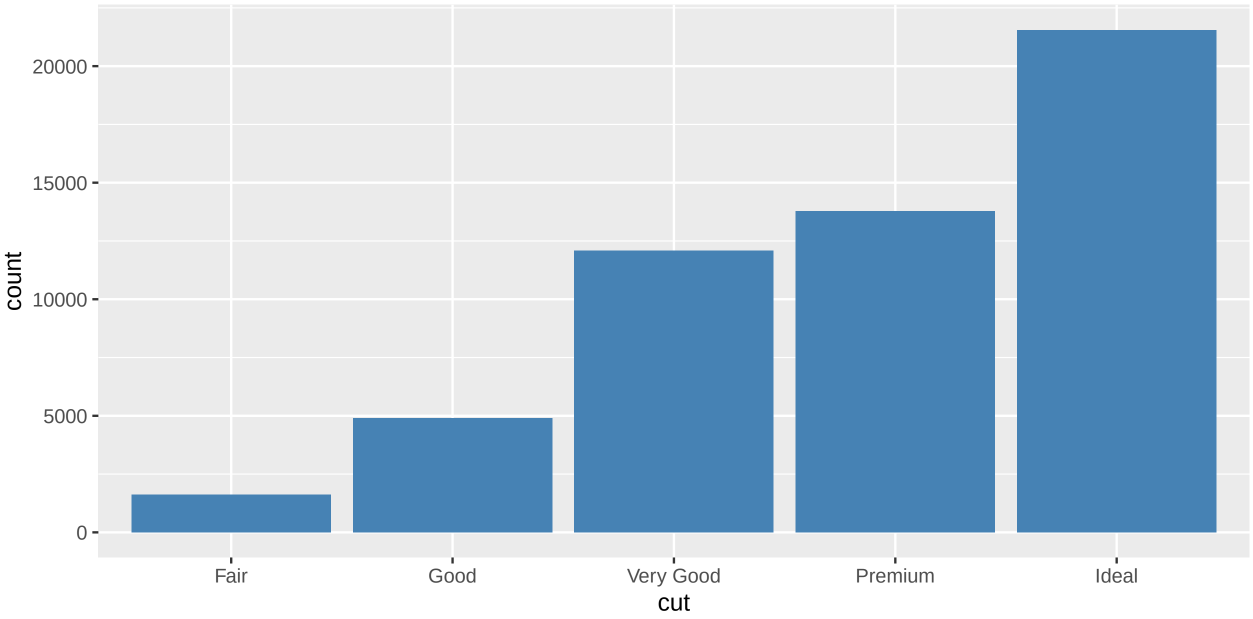
```
plt1 = ggplot(cars,aes(x=speed,y=dist)) + geom_point()
plt1
```



Question 2

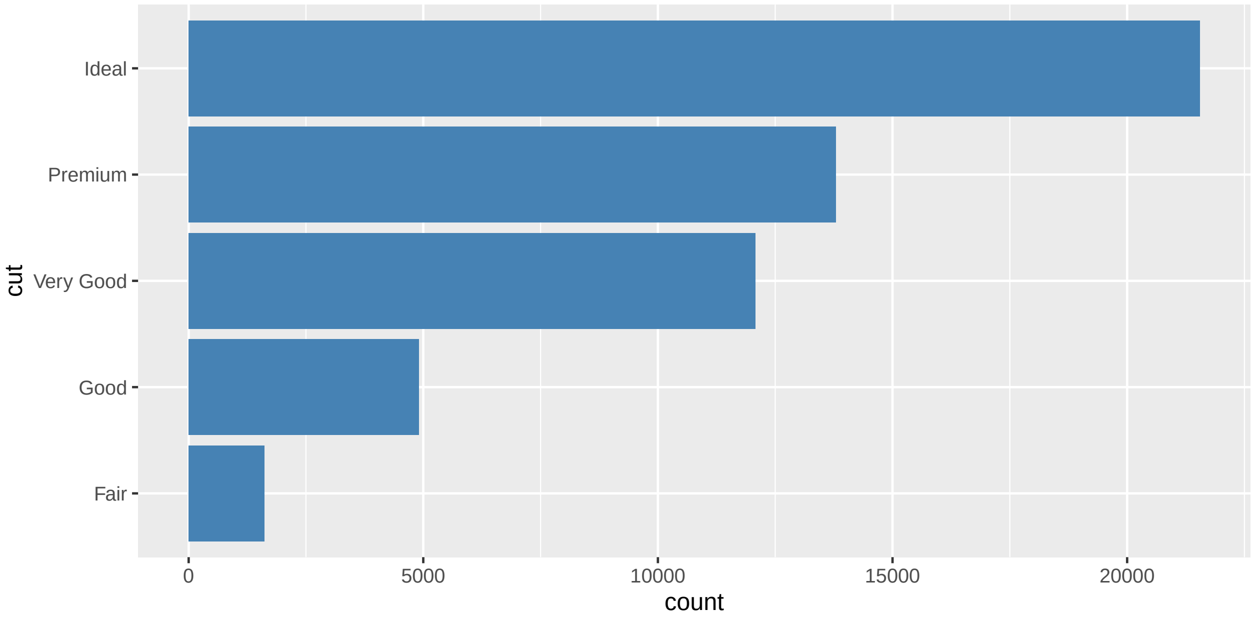
(a) Using the `diamonds` data, plot the distribution of the different `cut` categories. Have the color (i.e. the `fill=`) of the bars be `steelblue`. Save the plot as object `plt2`.

```
plt2 <-
  ggplot(diamonds,aes(x=cut)) +
  geom_bar(fill='steelblue')
plt2
```



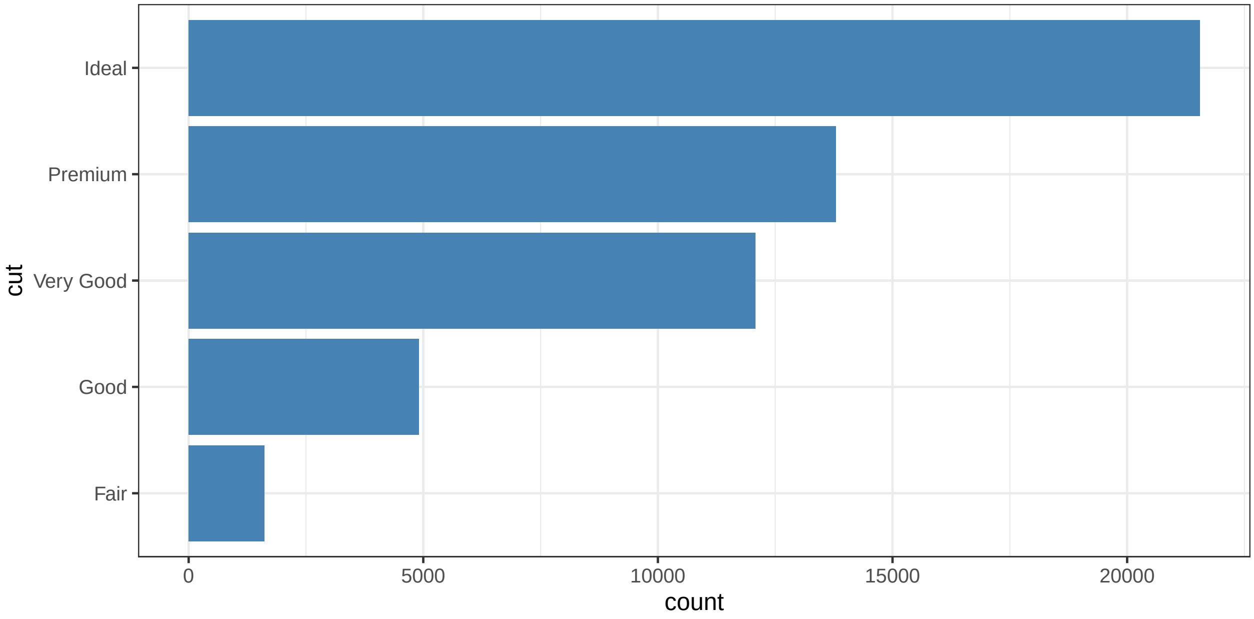
(b) Using `plt2`, flip the coordinates so that the x-axis is on the y-axis. *Don't render a new plot.* Overwrite the `plt2` with the new plot.

```
plt2 <- plt2 + coord_flip()
plt2
```



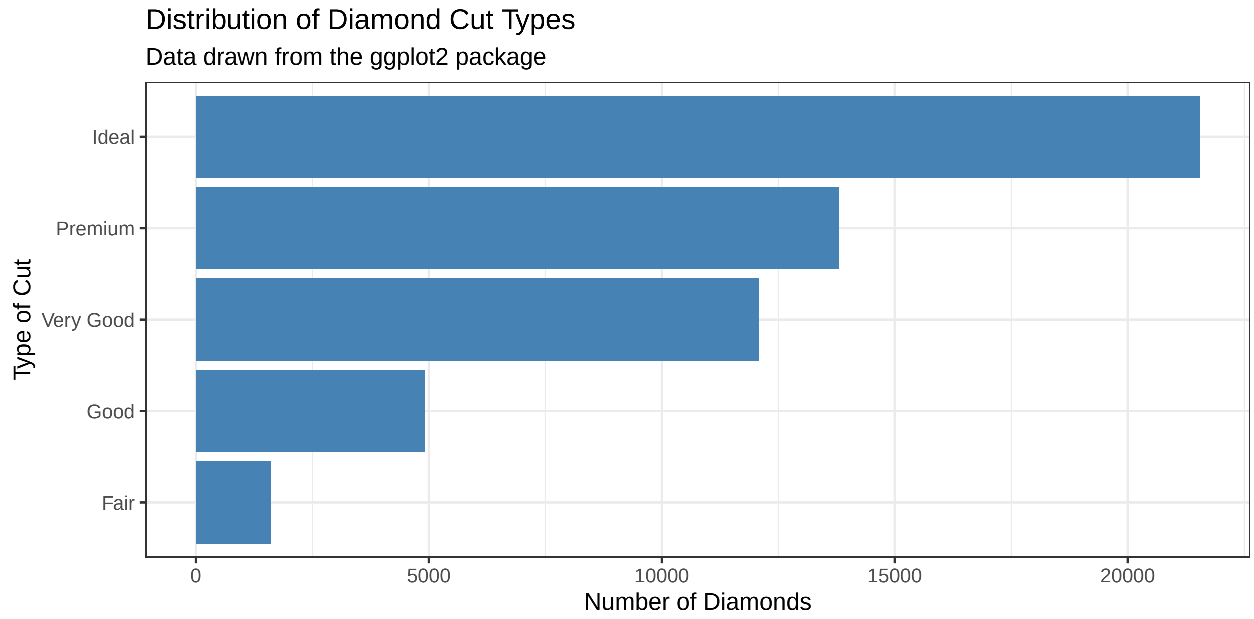
(c) Change the theme of the plot (i.e. the background color) to be black and white instead of grey using the `plt2` object. Overwrite the `plt2` with the new plot.

```
plt2 <- plt2 + theme_bw()
plt2
```



(d) Using `plt2` change the labels of the plot. Label the axis with the "count" to "Number of Diamonds", the axis with "cut" to "Type of Cut", and add the title "Distribution of Diamond Cut Types" with the subtitle "Data drawn from the ggplot2 package". Overwrite the `plt2` with the new plot.

```
plt2 <-
  plt2 +
  labs(y="Number of Diamonds",x="Type of Cut",
       title="Distribution of Diamond Cut Types",
       subtitle="Data drawn from the ggplot2 package")
plt2
```



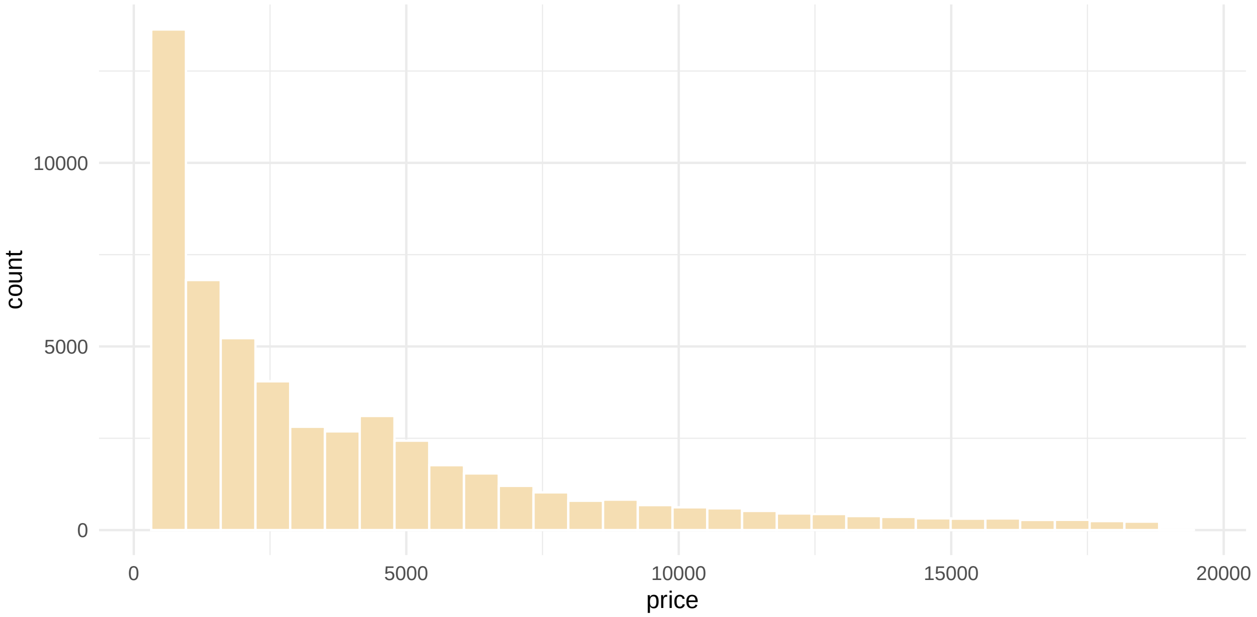
(e) Save plot to your desktop as a `.pdf` using `ggsave()`. Make sure the height is 5 inches and width is 10 inches.

```
ggsave(filename = "my_plot.pdf",device = "pdf",plot = plt2,
       width = 10,height = 5,units = "in")
```

Question 3

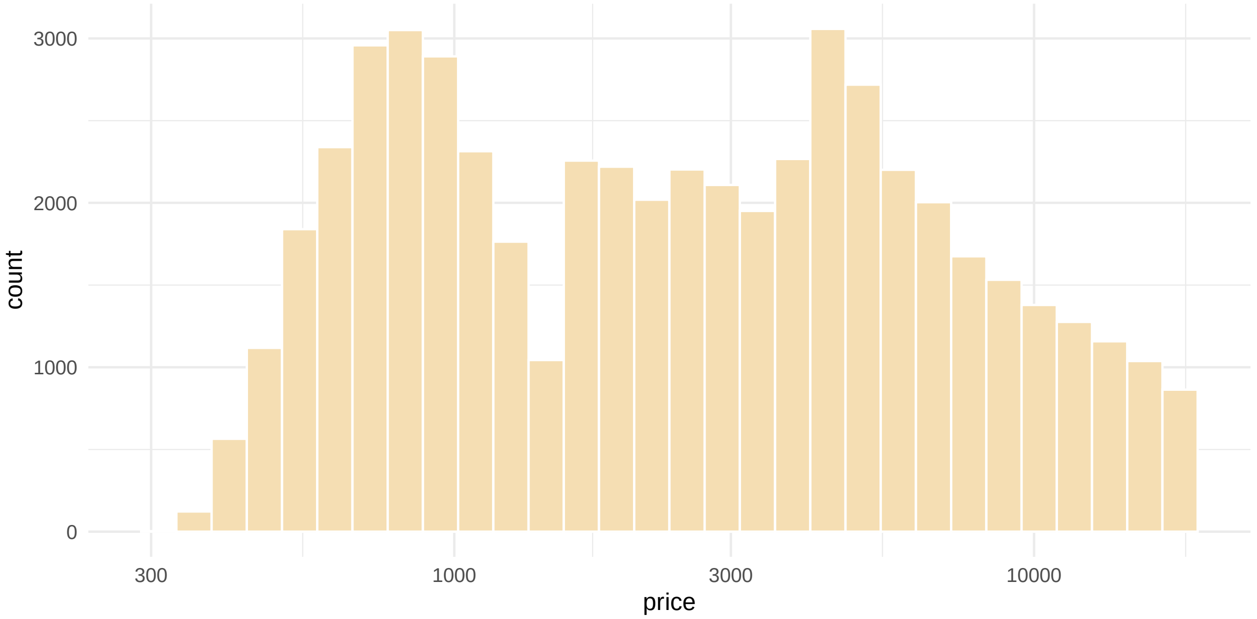
(a) using the `diamonds` data, plot distribution of `price` using a histogram. Set the color to `white` and the fill to `wheat`, and the theme to `minimal`. Save as object `plt3`

```
plt3 <-
  ggplot(diamonds,aes(x=price)) +
  geom_histogram(color="white",fill="wheat") +
  theme_minimal()
plt3
```



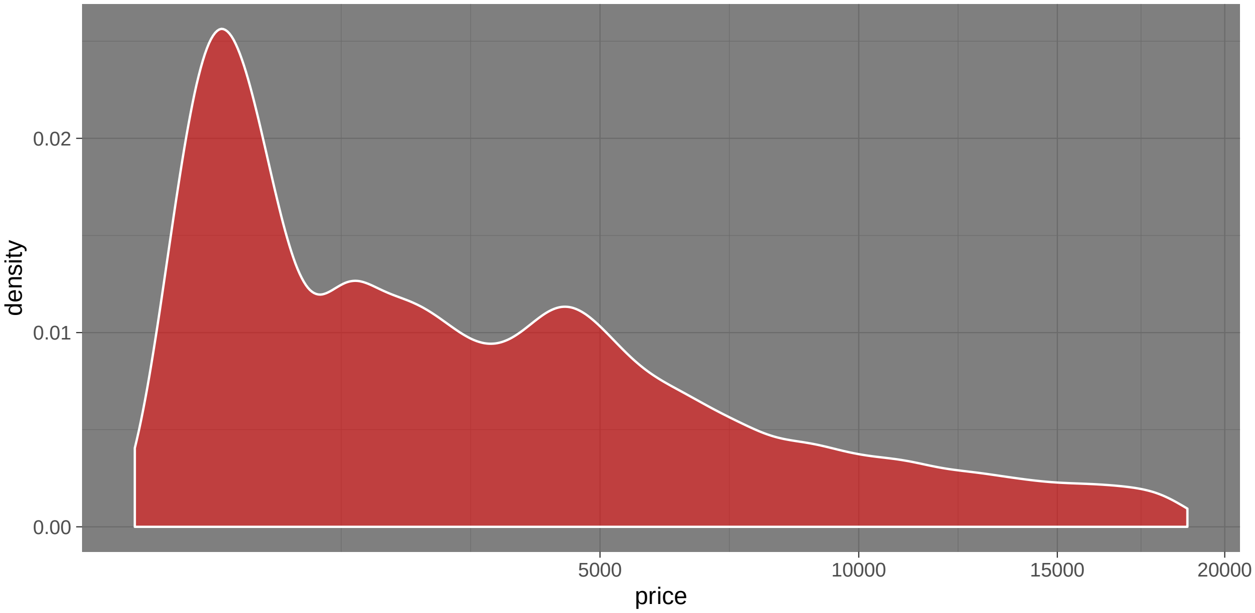
(b) using the plt3, change the distribution to a log_10 distribution.

```
plt3 <- plt3 + scale_x_log10()
plt3
```



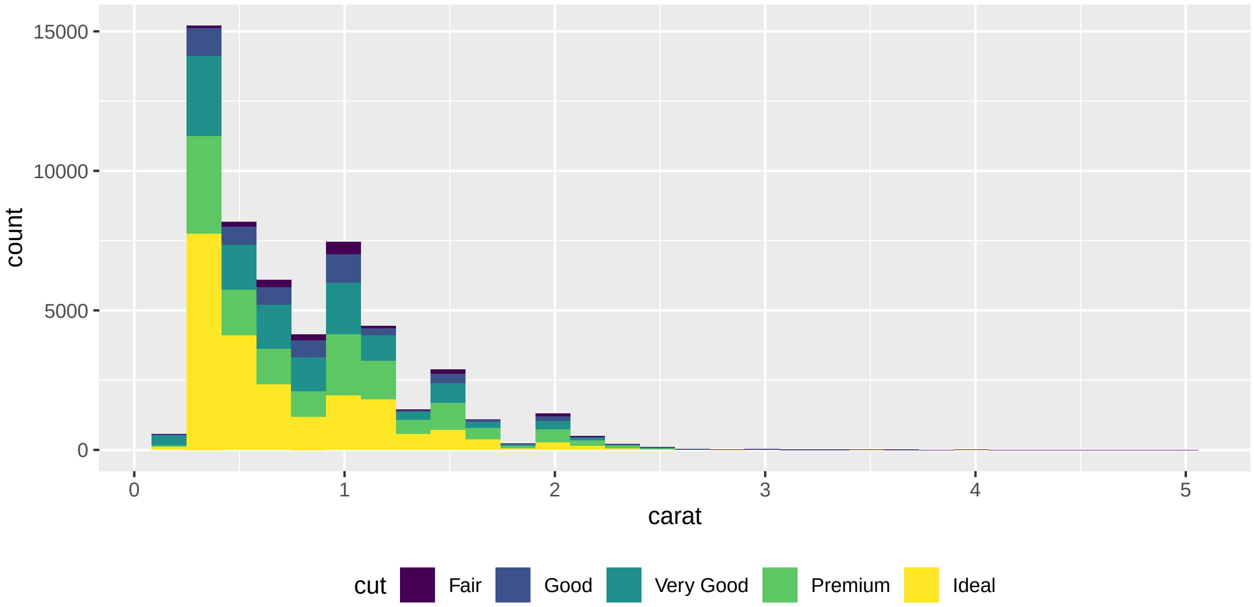
(c) using the diamonds data, plot distribution of price using a density plot. Change the scale to a square root scale. Fill the density plot with a "red", set the alpha to .5 (so it's translucent), and set the color to "white". Use a dark theme.

```
ggplot(diamonds,aes(x=price)) +
  scale_x_sqrt() +
  geom_density(color="white",fill="red",alpha=.5) +
  theme_dark()
```



(d) using the diamonds data, plot a histogram by carat and fill by cut type. Set the theme to black and white. Change the plot so that the legend is on the bottom.

```
ggplot(diamonds,aes(x=carat,fill=cut)) +
  geom_histogram() +
  theme(legend.position = "bottom")
```



Question 4

For this question, I'm going to simulate some data. Note that to generate this data, I'm going to draw from a multivariate normal distribution. This uses the `MASS` package.

```
fake_data <- MASS::mvrnorm(1000,mu = c(0,1),Sigma = matrix(c(1,.25,.57,1),nrow = 2)) %>%
  as.tibble()
```

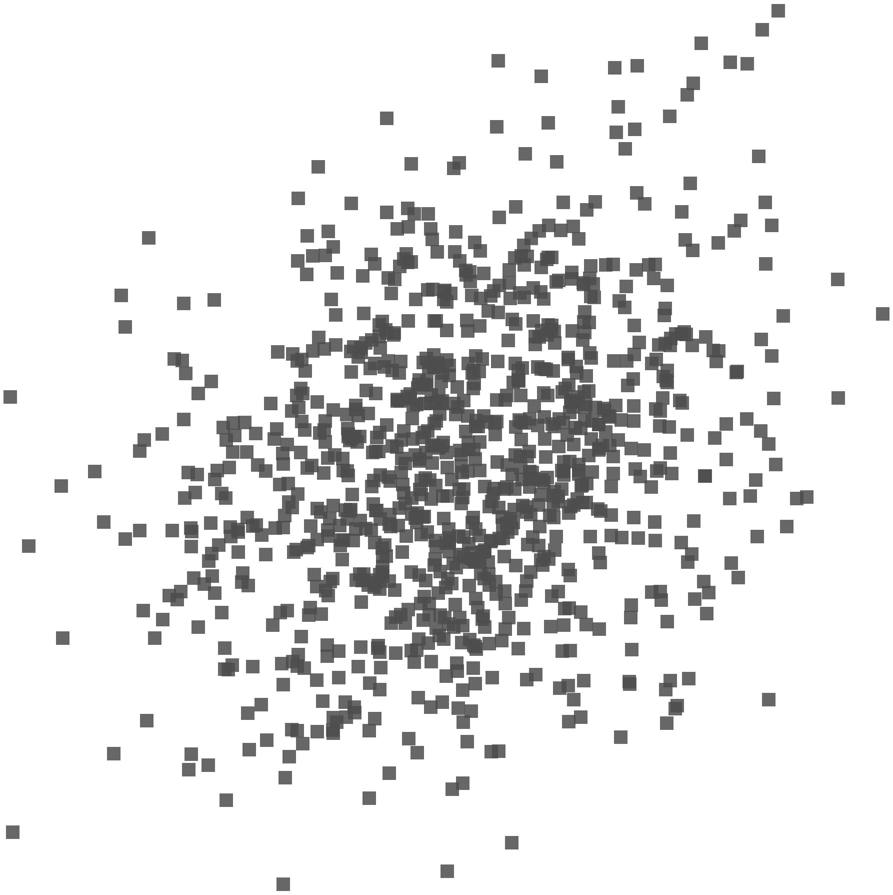
```
FALSE Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semantics).
FALSE This warning is displayed once per session.
```

```
head(fake_data)
```

```
FALSE # A tibble: 6 x 2
FALSE       V1       V2
FALSE    <dbl>    <dbl>
FALSE 1  0.153  0.191
FALSE 2 -2.01   0.00847
FALSE 3  0.607  1.64
FALSE 4  0.181 -0.189
FALSE 5 -0.450 -0.117
FALSE 6  0.145  0.920
```

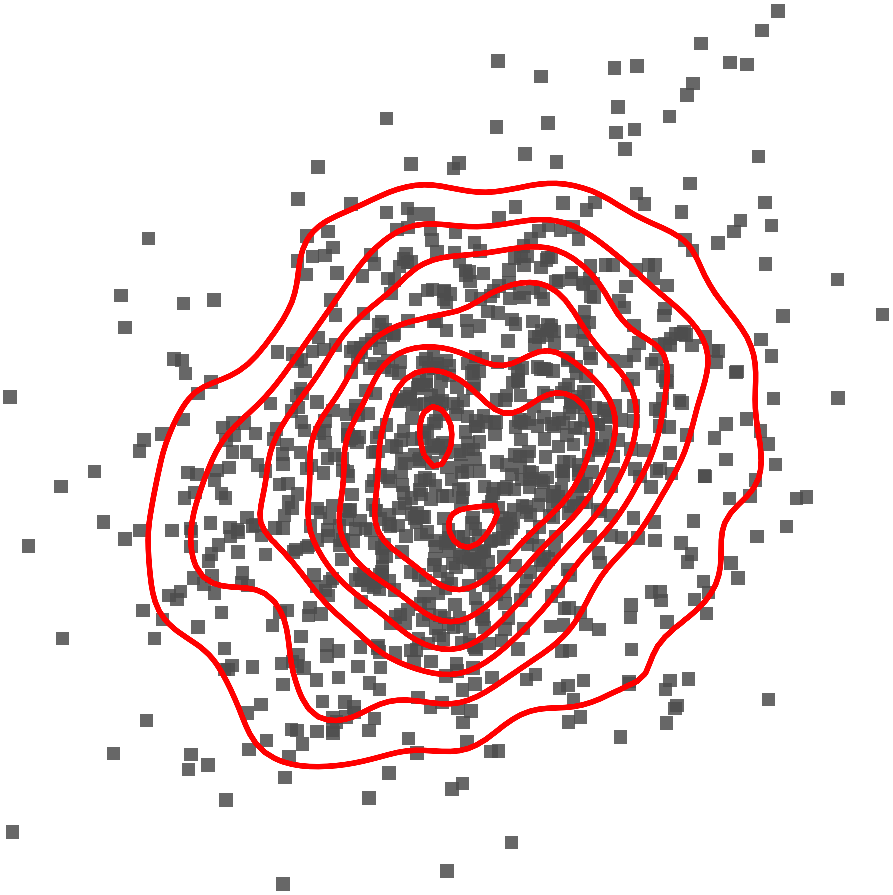
(a) Using this `fake_data`, plot a scatter plot with a `light` theme a title of some kind, and changing the shape of the points to something else (like a square). Set the color of the points to "grey30" and increase the point size to 2, and the alpha at .85 (again, to male the points slightly transparent). Set the theme to "void". Save as object `plt4`

```
plt4 <-
  ggplot(fake_data,aes(V1,V2)) +
  geom_point(pch=15,size=2,color="grey30",alpha=.85) +
  theme_void()
plt4
```



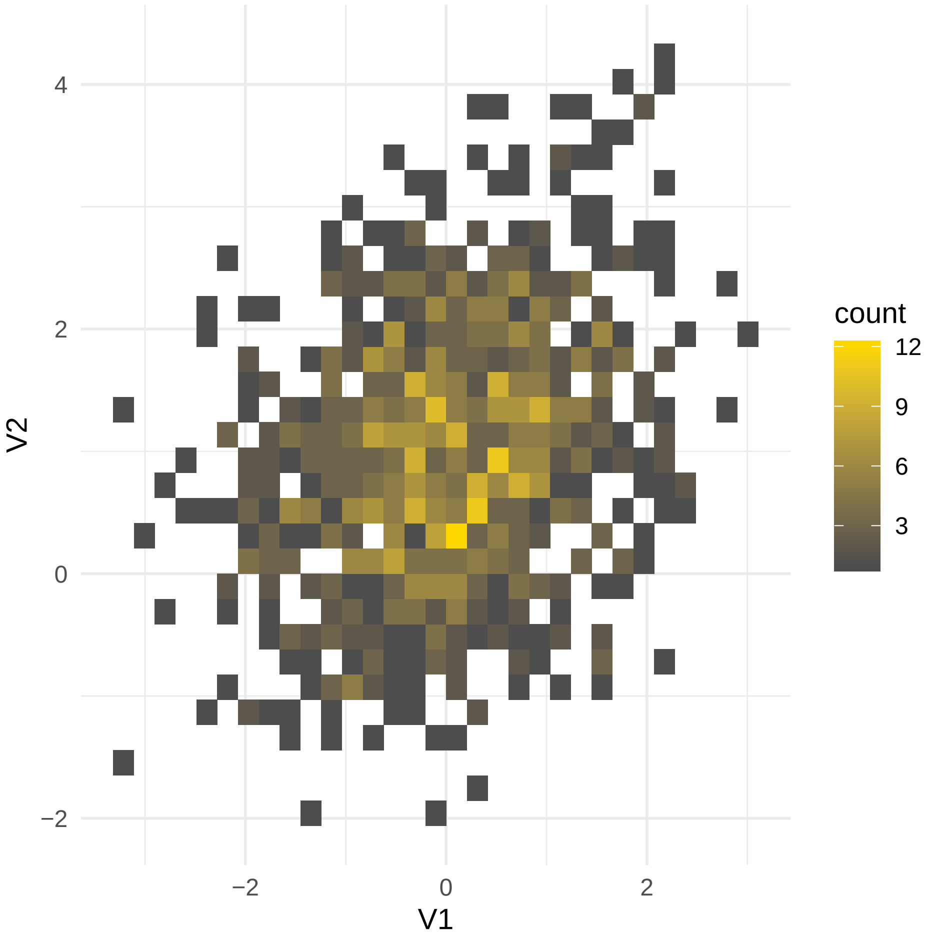
(b) Using object `plt4`, plot a 2d density plot to the points. Set the color to "red" and the linewidth to 1. What does this plot tell us?

```
plt4 +
  geom_density2d(color="red",lwd=1)
```



Using `fake_data`, build a heatmap using `geom_bin2d()`. Change the color scale to move from "grey20" (low) to "gold" (high). Set theme to minimal.

```r
ggplot(fake_data,aes(V1,V2)) +
  geom_bin2d() +
  scale_fill_gradient(low="grey30", high="gold") +
  theme_minimal()
```


count
12
9
6
3
V2
4
2
0
−2
V1
−2
0
2