

Exercise 1

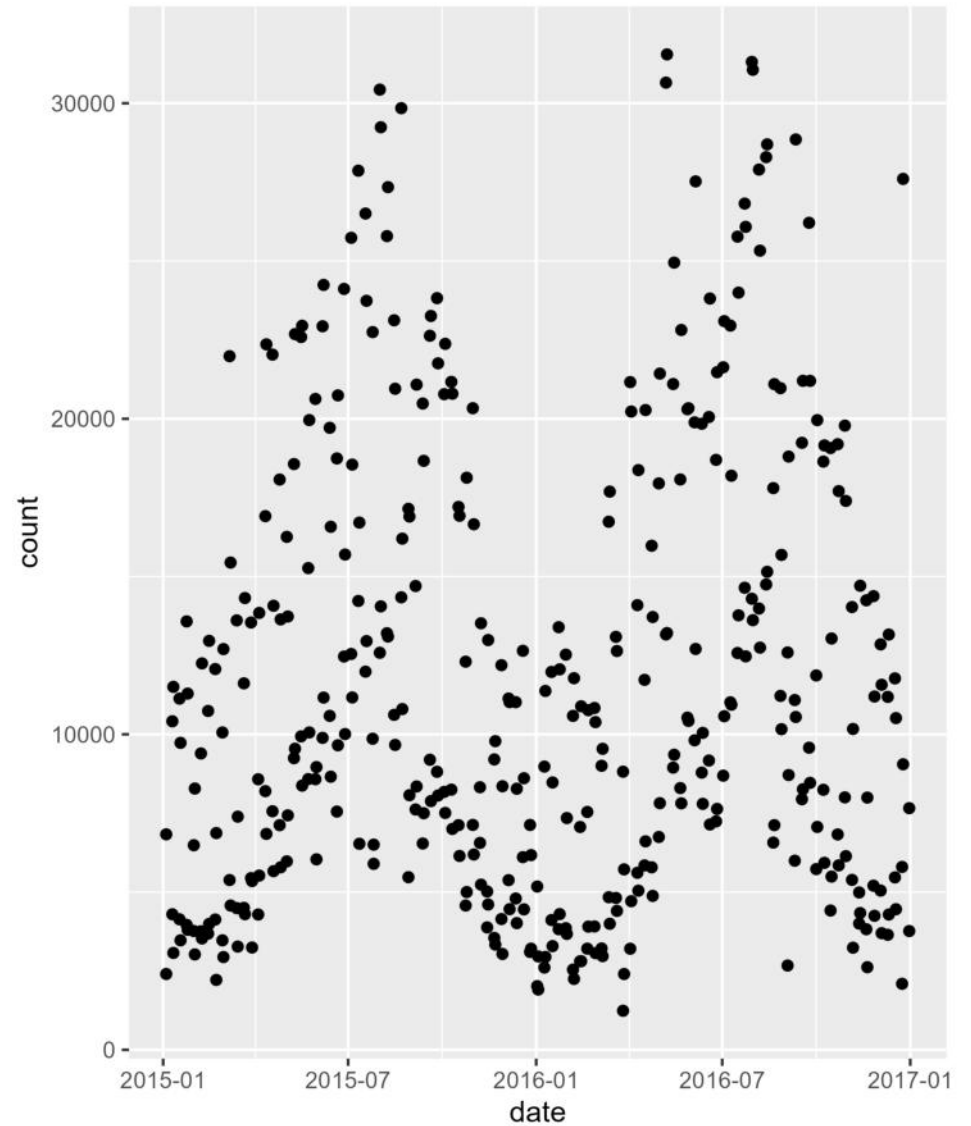
- Explore the TfL bike share data visually:
create a timeseries of reported bike shares on weekend days
 - Highlight day and night encoded by colors and shapes.
 - Connect the points of each period with lines.
 - What is the difference between `geom_line()` and `geom_path()`?
 - Apply your favorite theme to the plot.
 - Add meaningful labels.
 - Bonus: use shape to encode Saturday vs Sunday instead.
- Save the plot as a vector graphic with a decent plot size.

Import the Data Set

```
1 bikes <- readr::read_csv(  
2   here::here("data", "london-bikes-custom.csv"),  
3   col_types = "Dcffffilllddddc"  
4 )  
5  
6 bikes$season <- forcats::fct_inorder(bikes$season)  
7  
8 library(tidyverse)
```

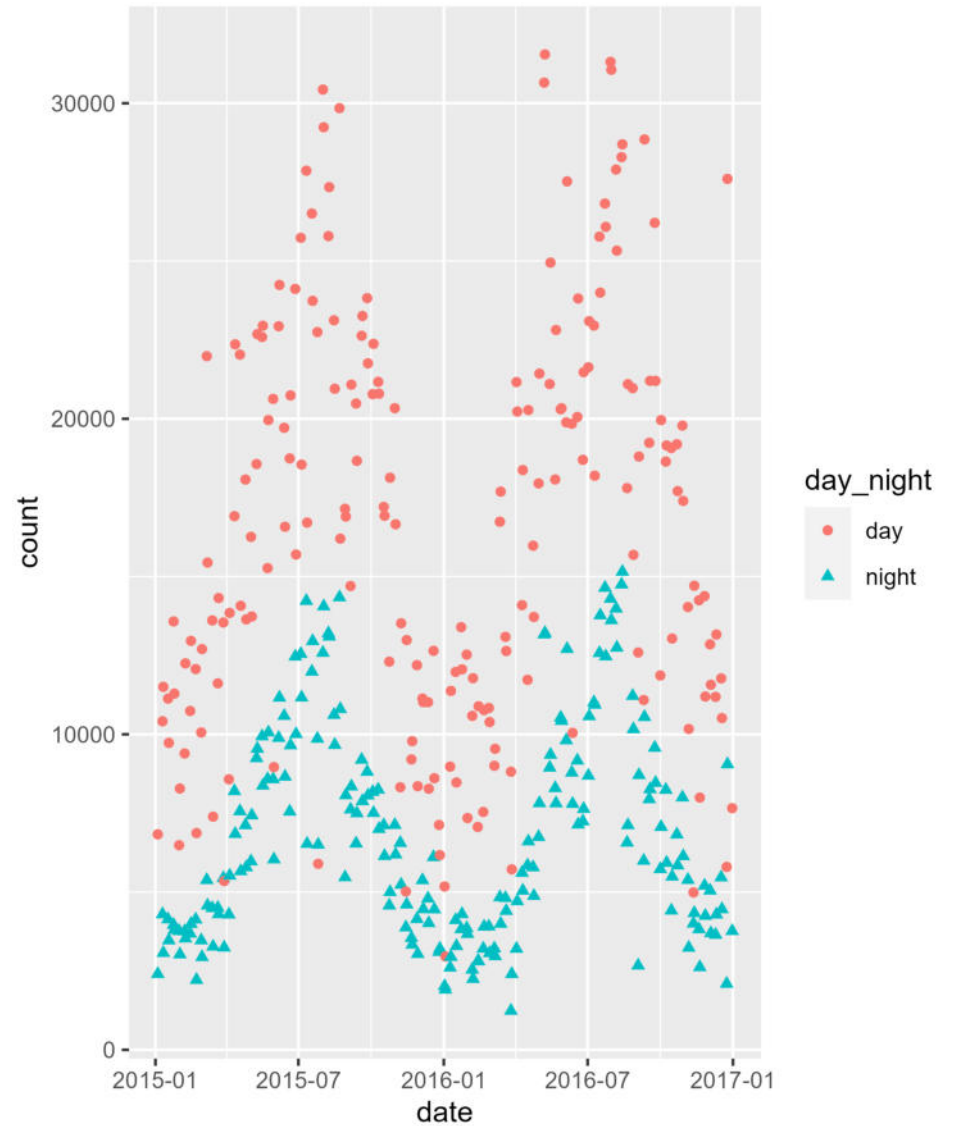
Scatterplot Counts vs. Date

```
1 ggplot(  
2     filter(bikes, is_weekend == TRUE),  
3     aes(x = date, y = count)  
4 ) +  
5     geom_point()
```



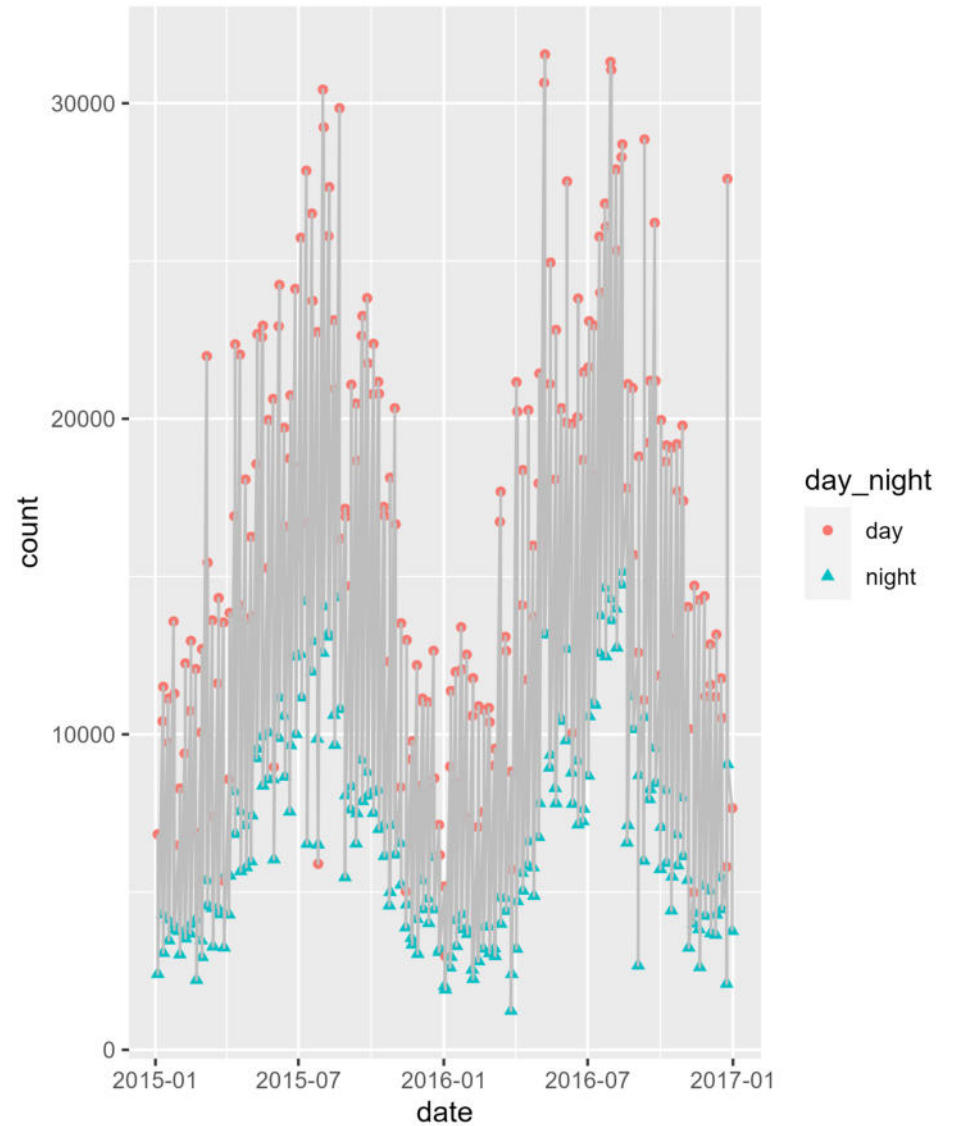
Encode Day Period by Colors and Shapes

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5   geom_point(  
6     aes(color = day_night,  
7         shape = day_night)  
8   )
```



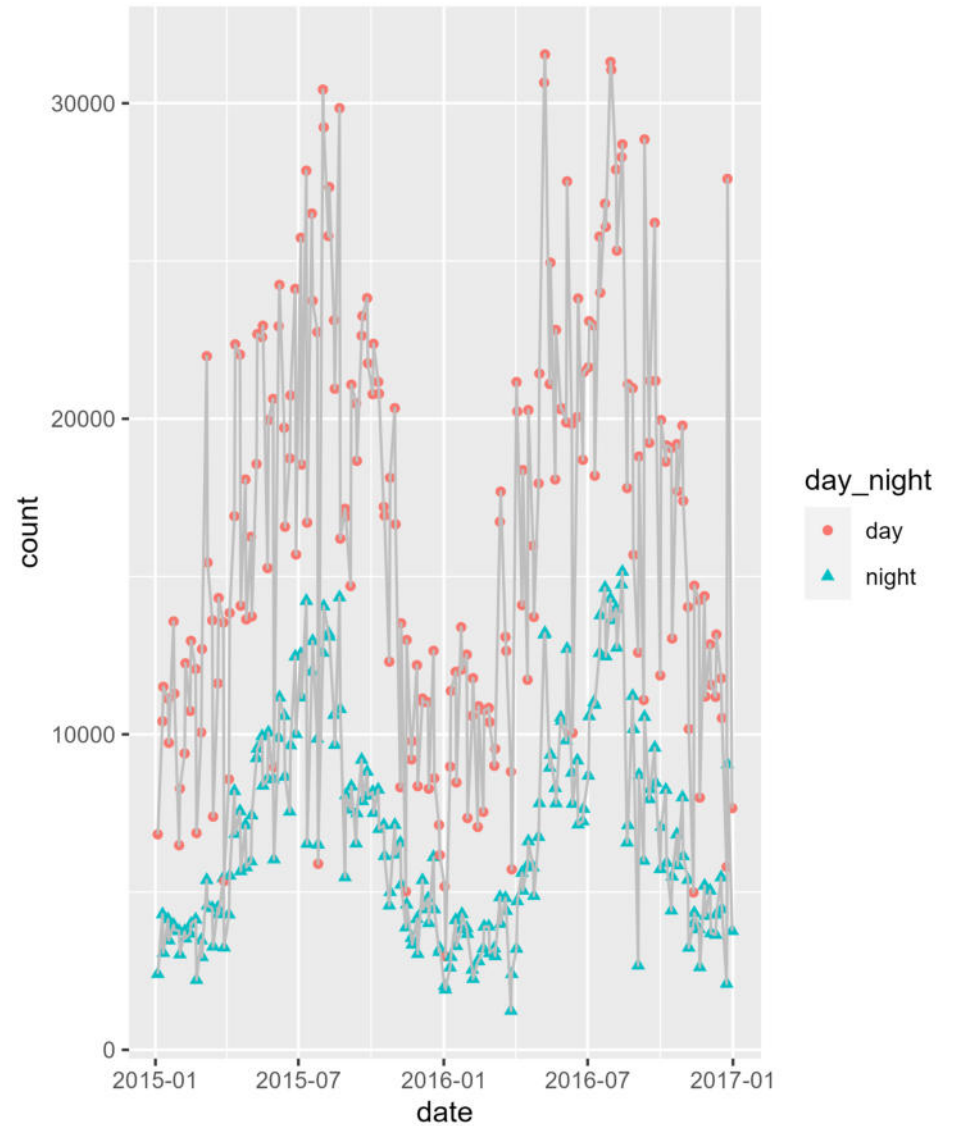
Add Line

```
1 ggplot(  
2     filter(bikes, is_weekend == TRUE),  
3     aes(x = date, y = count)  
4 ) +  
5     geom_point(  
6     aes(color = day_night,  
7         shape = day_night)  
8 ) +  
9     geom_line(  
10     color = "grey"  
11 )
```



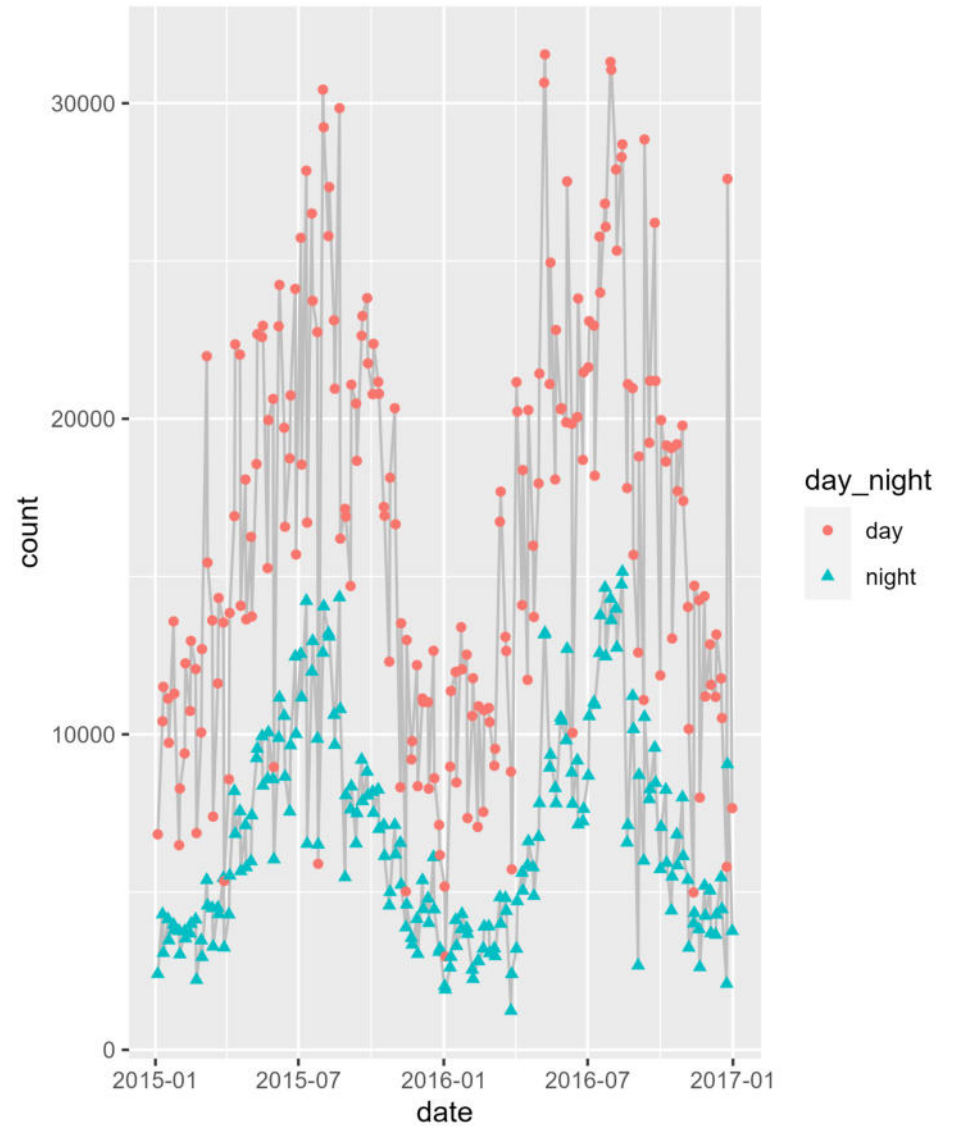
Group Lines by Day Period

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5   geom_point(  
6     aes(color = day_night,  
7         shape = day_night)  
8   ) +  
9   geom_line(  
10    aes(group = day_night),  
11    color = "grey"  
12  )
```



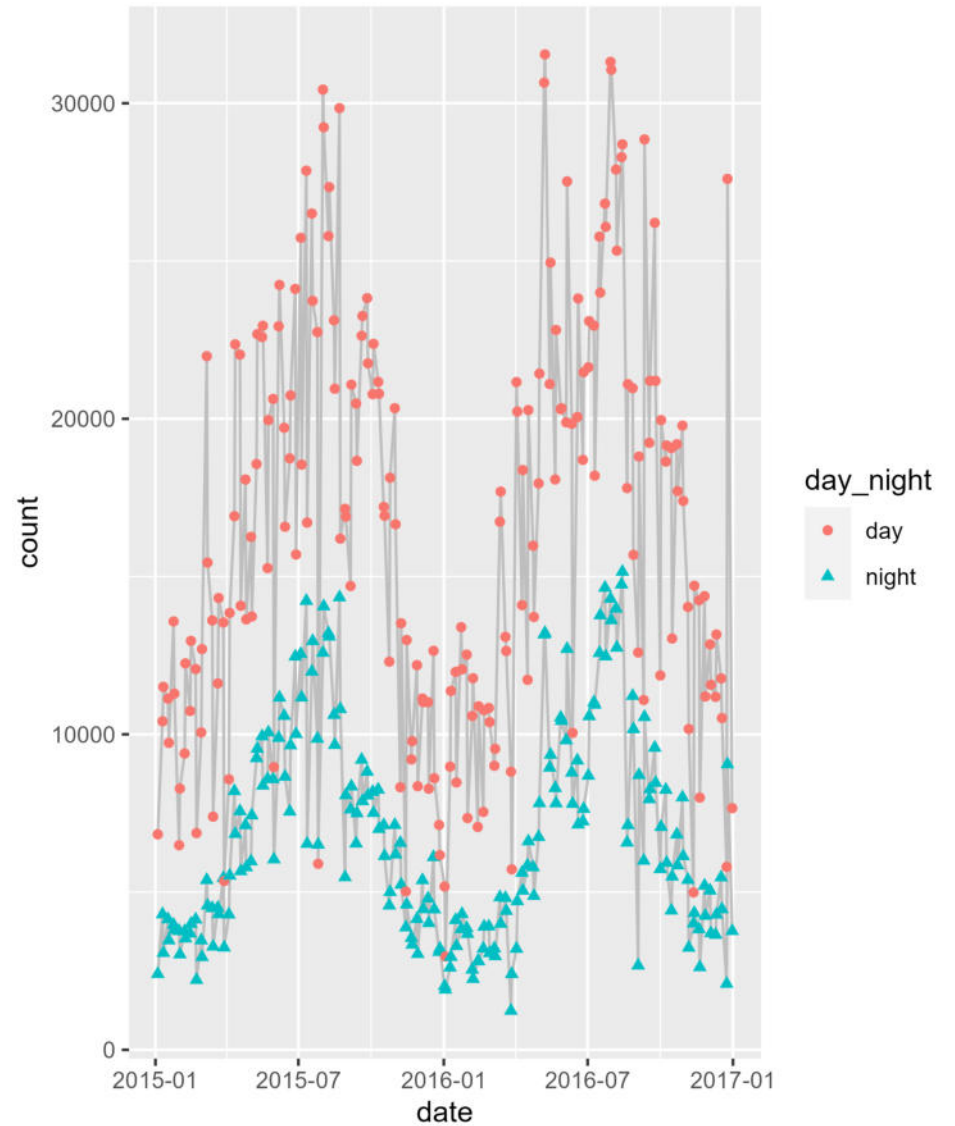
Order Layers

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5   geom_line(  
6     aes(group = day_night),  
7     color = "grey"  
8 ) +  
9   geom_point(  
10    aes(color = day_night,  
11         shape = day_night)  
12 )
```



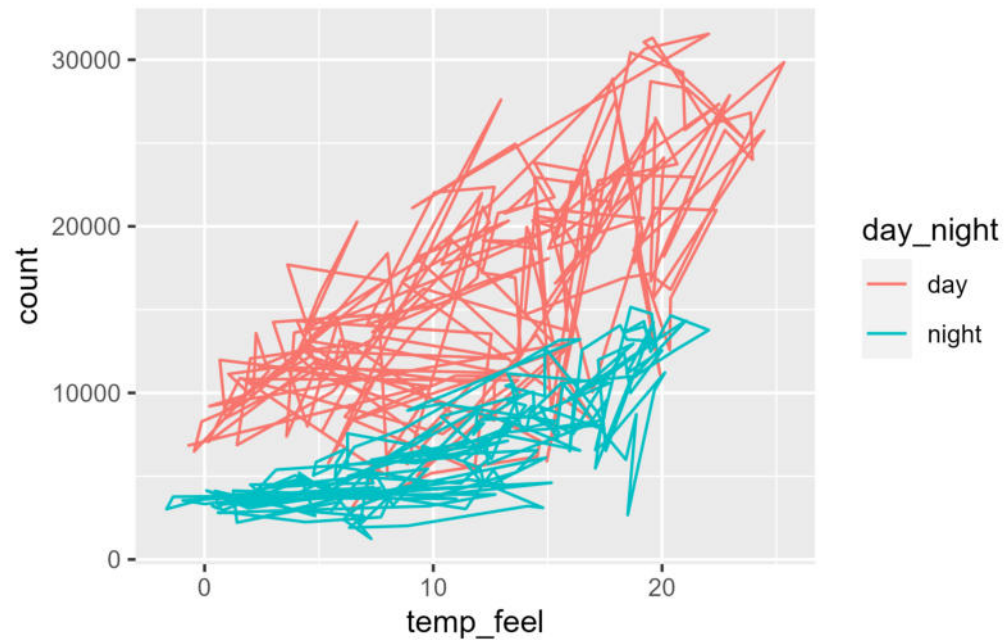
Use `geom_path()` instead

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5   geom_path(  
6     aes(group = day_night),  
7     color = "grey"  
8 ) +  
9   geom_point(  
10    aes(color = day_night,  
11        shape = day_night)  
12 )
```

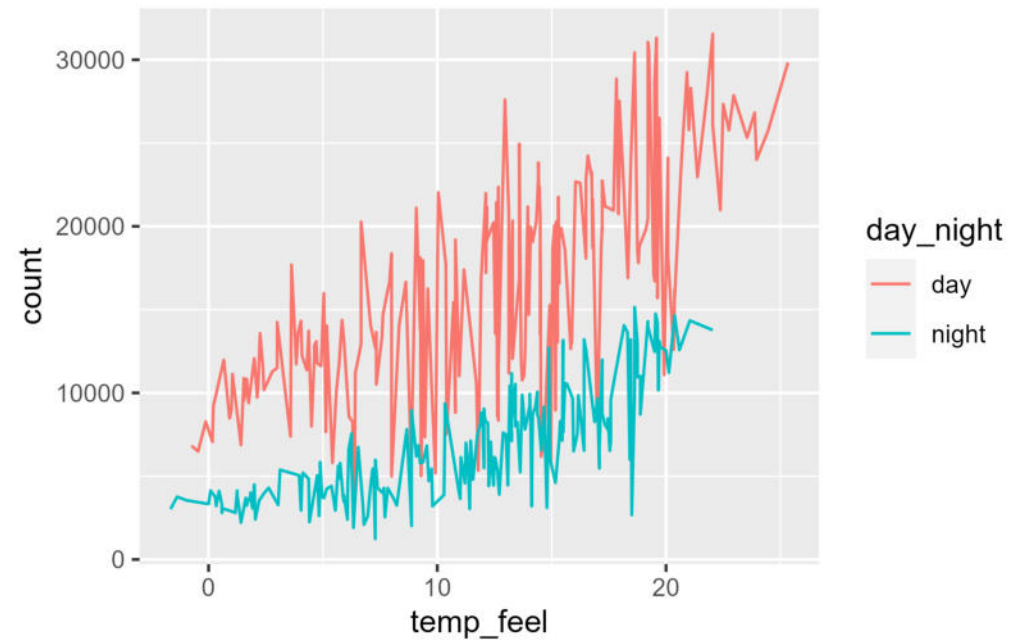


`geom_line()` vs. `geom_path()`

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = temp_feel, y = count)  
4 ) +  
5   geom_path(  
6     aes(color = day_night)  
7   )
```

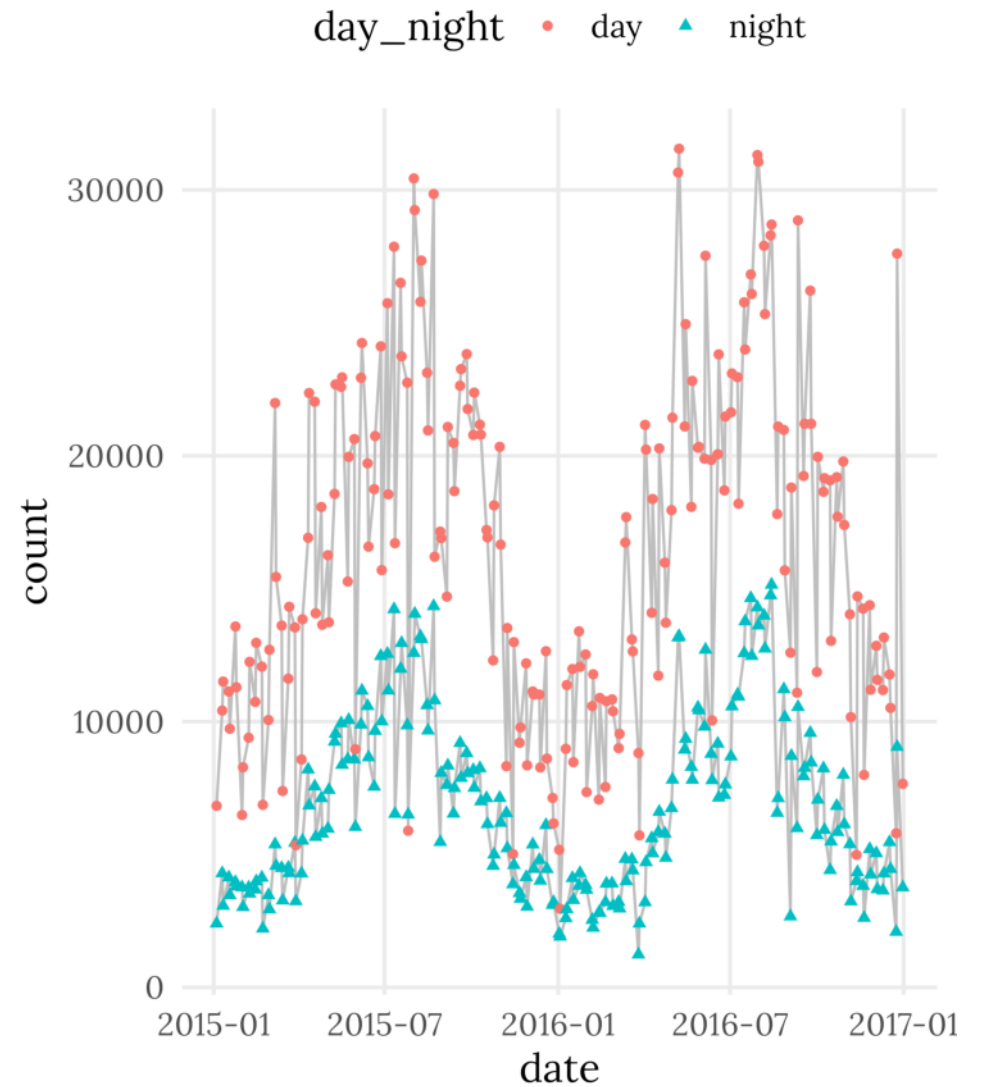


```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = temp_feel, y = count)  
4 ) +  
5   geom_line(  
6     aes(color = day_night)  
7   )
```



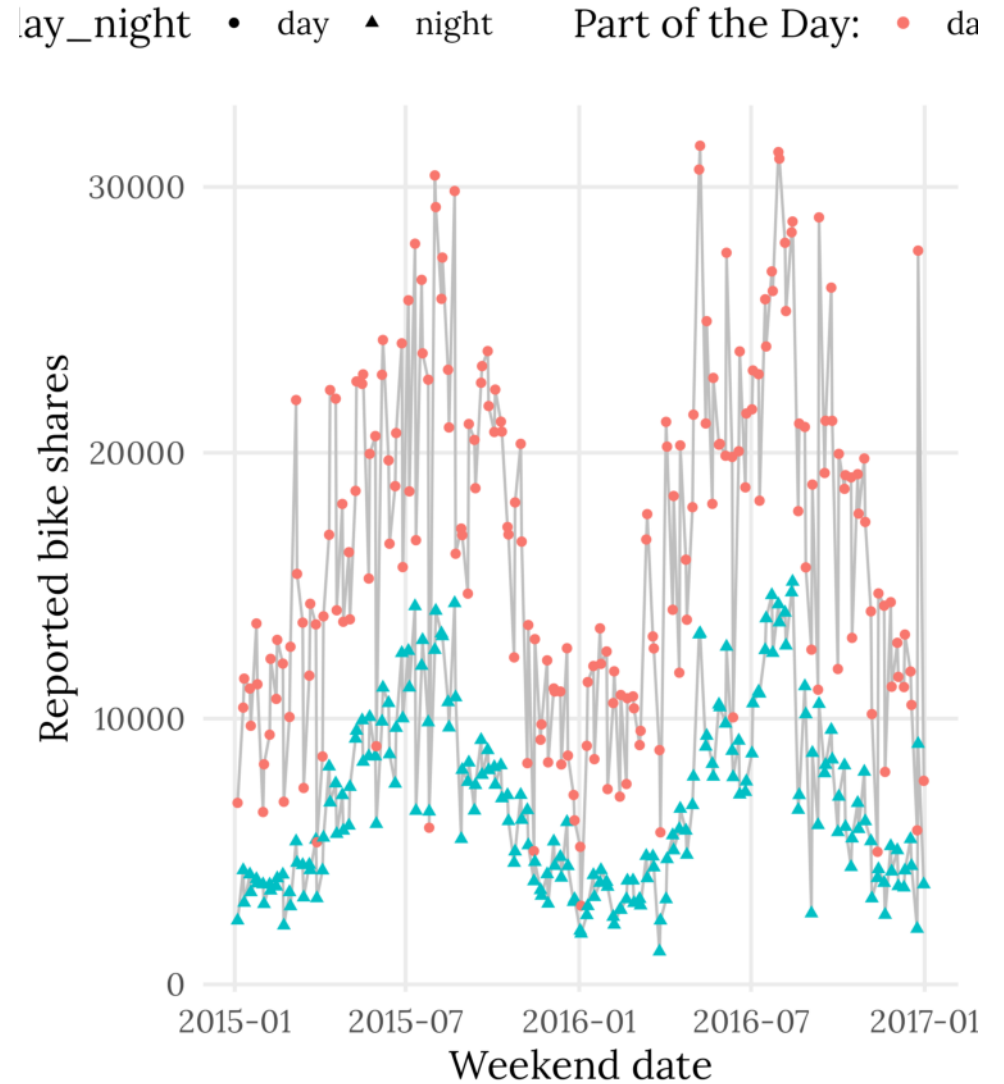
Apply a Theme

```
1 g <- ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5 geom_line(  
6   aes(group = day_night),  
7   color = "grey"  
8 ) +  
9 geom_point(  
10  aes(color = day_night,  
11      shape = day_night)  
12 )  
13  
14 g +  
15 theme_minimal(  
16   base_size = 15,  
17   base_family = "Lora"  
18 ) +  
19 theme(  
20   legend.position = "top",  
21   panel.grid.minor = element_blank()  
22 )
```



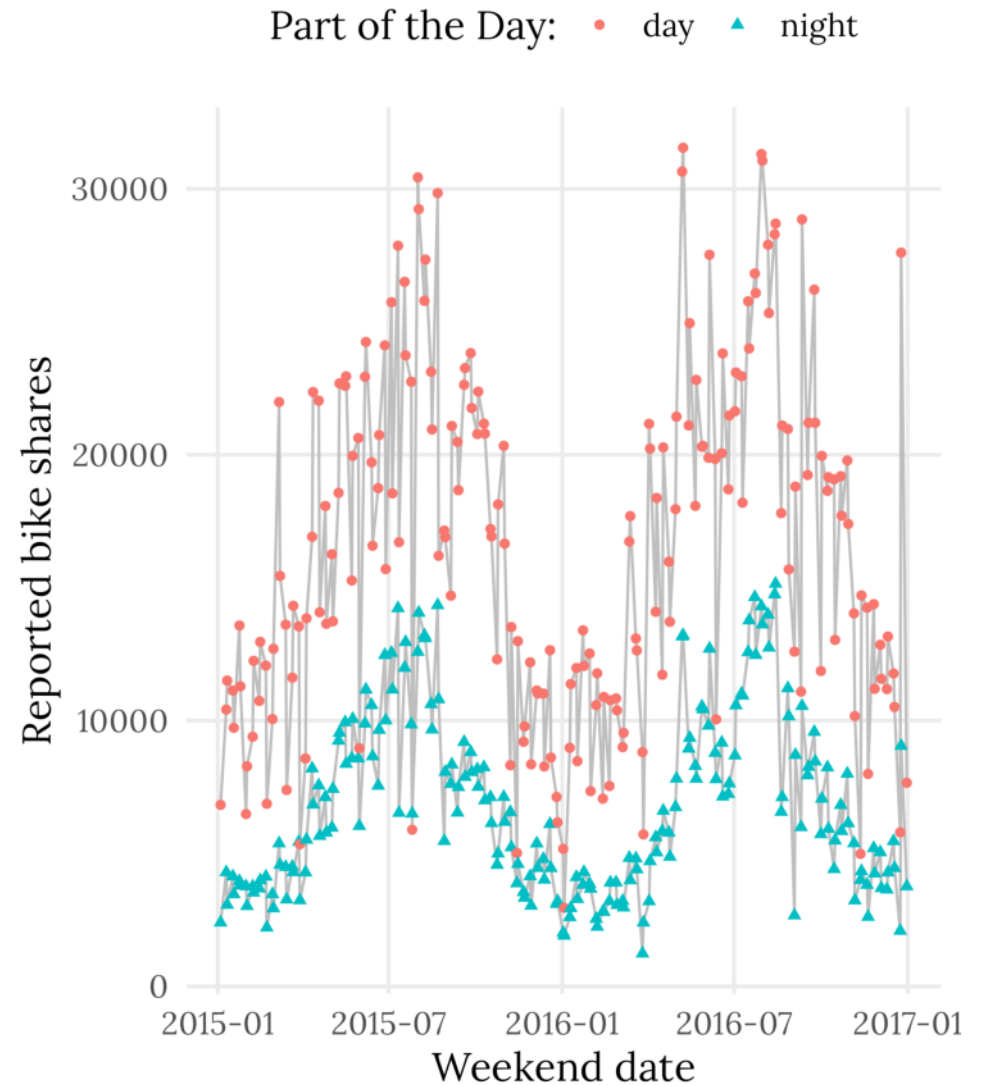
Add Meaningful Labels

```
1 g +
2   labs(
3     x = "Weekend date",
4     y = "Reported bike shares",
5     color = "Part of the Day:"
6   ) +
7   theme_minimal(
8     base_size = 15,
9     base_family = "Lora"
10  ) +
11  theme(
12    legend.position = "top",
13    panel.grid.minor = element_blank()
14  )
```



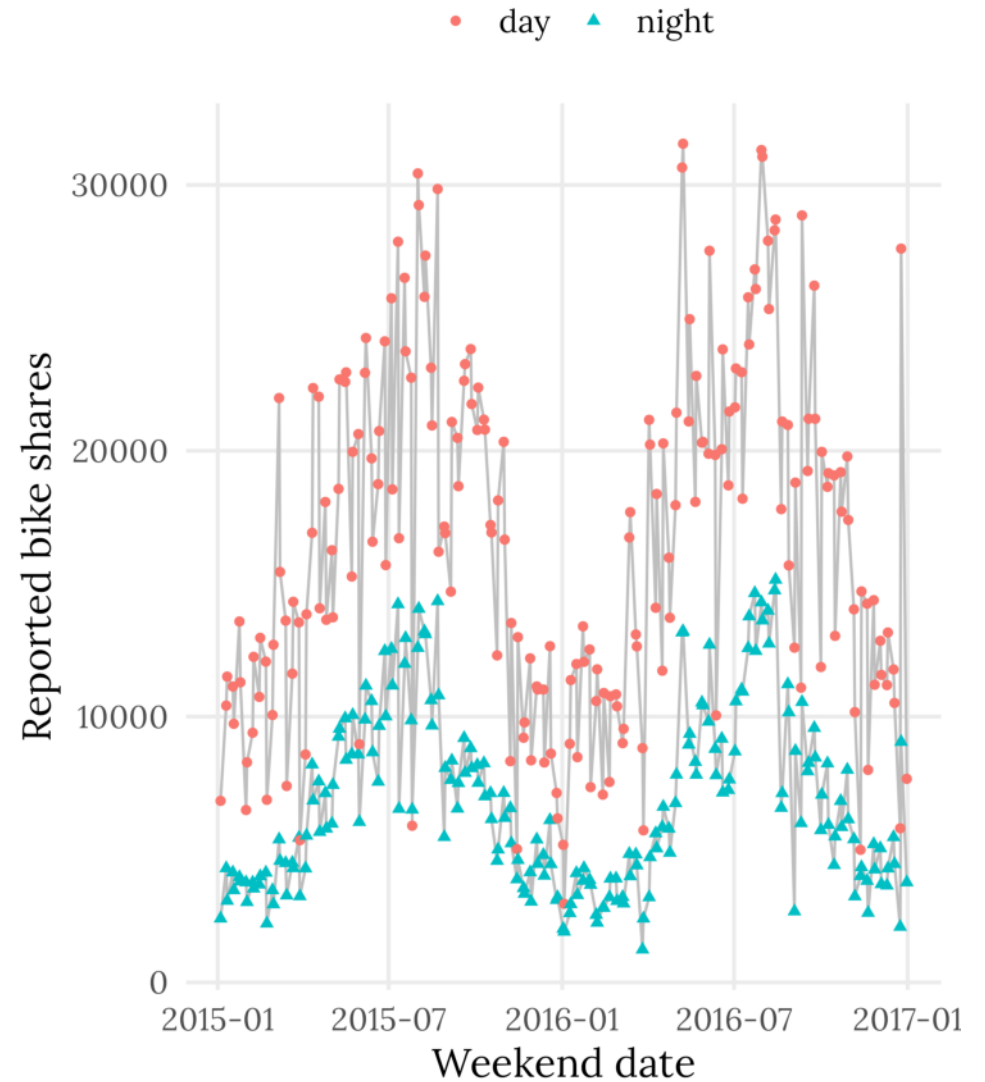
Add Meaningful Labels

```
1 g +
2   labs(
3     x = "Weekend date",
4     y = "Reported bike shares",
5     color = "Part of the Day:",
6     shape = "Part of the Day:"
7   ) +
8   theme_minimal(
9     base_size = 15,
10    base_family = "Lora"
11  ) +
12  theme(
13    legend.position = "top",
14    panel.grid.minor = element_blank()
15  )
```



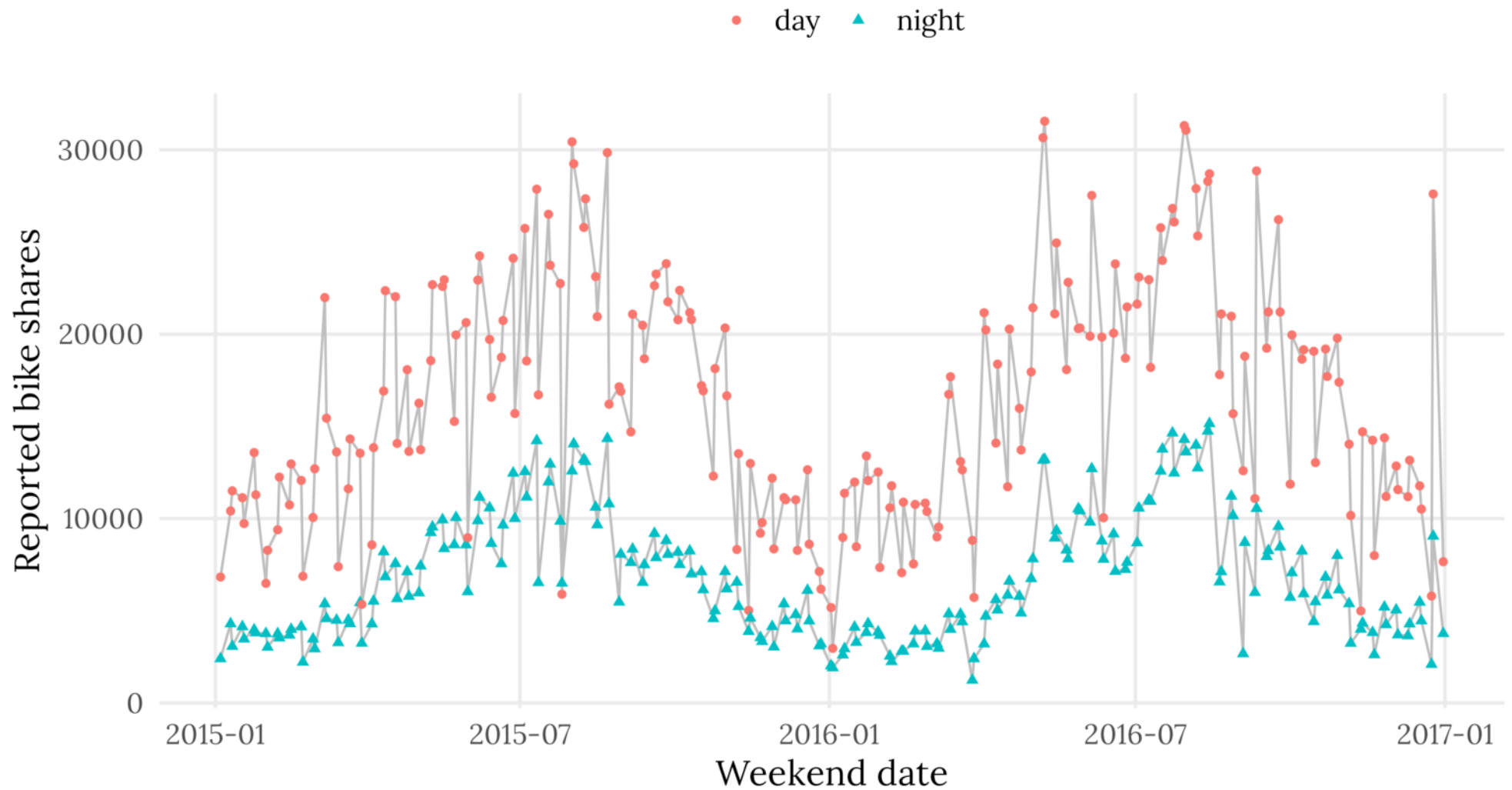
Add Meaningful Labels

```
1 g +
2   labs(
3     x = "Weekend date",
4     y = "Reported bike shares",
5     color = NULL,
6     shape = NULL
7   ) +
8   theme_minimal(
9     base_size = 15,
10    base_family = "Lora"
11  ) +
12  theme(
13    legend.position = "top",
14    panel.grid.minor = element_blank()
15  )
```



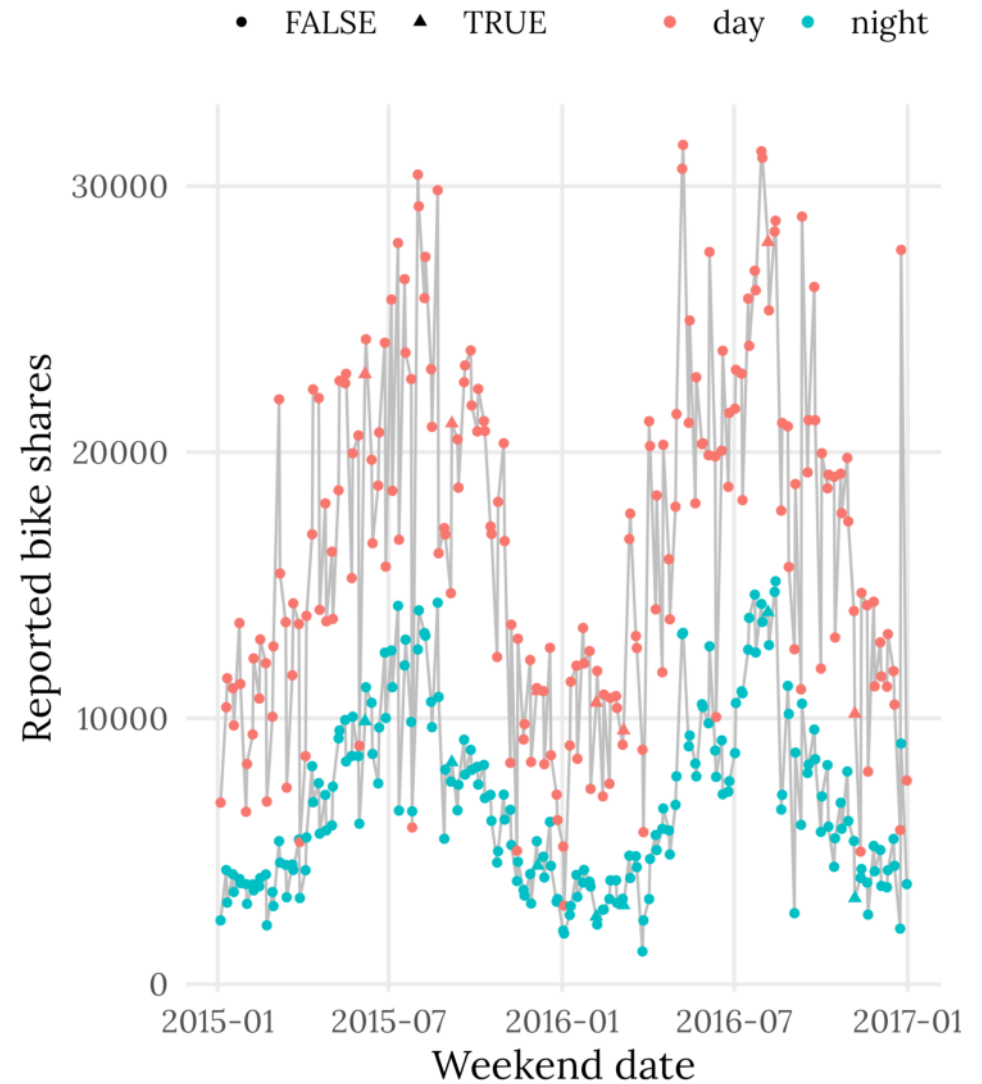
Save the Plot

```
1 ggsave(here::here("exercises", "plots", "02_concepts_pt1_ex1.pdf"),  
2       width = 9, height = 5, device = cairo_pdf)
```



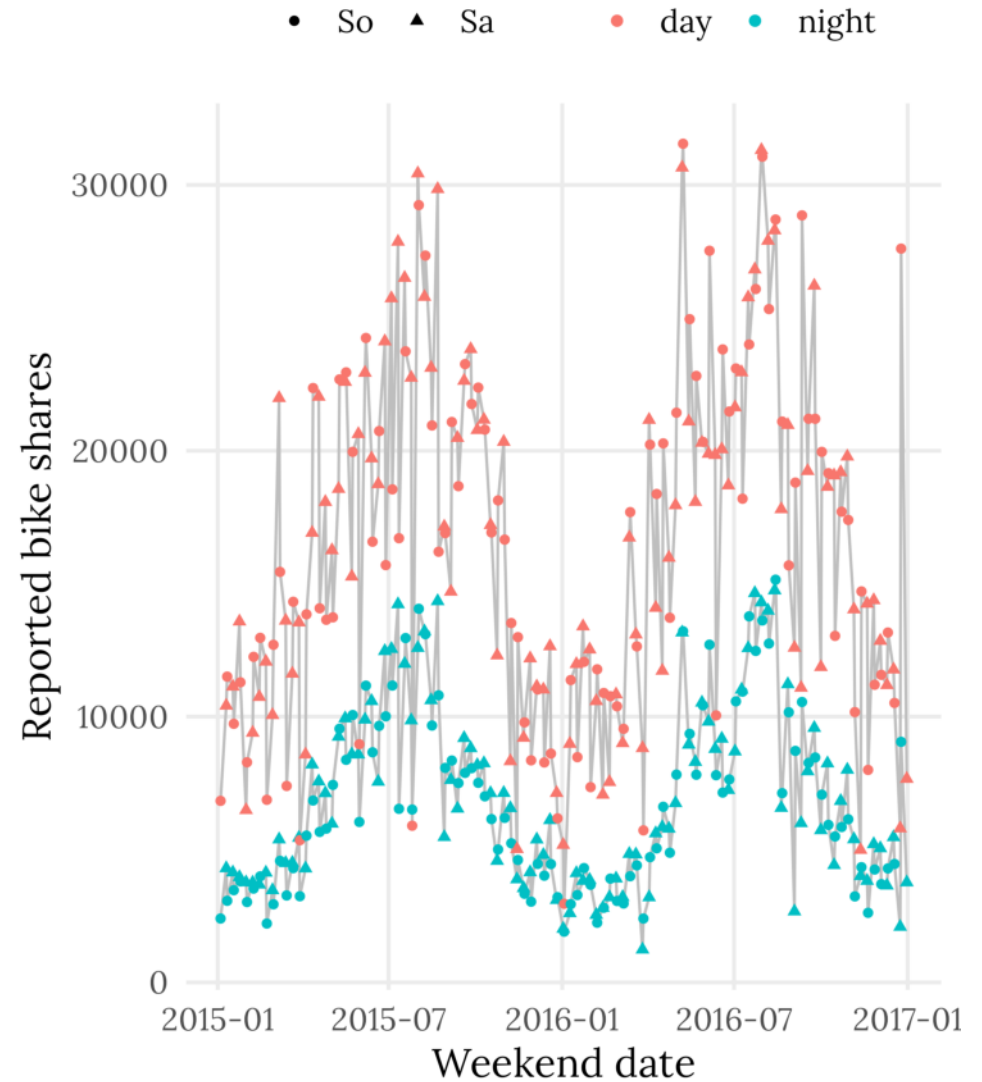
Bonus: Use Shape to Encode Sat vs Sun

```
1 ggplot(  
2   filter(bikes, is_weekend == TRUE),  
3   aes(x = date, y = count)  
4 ) +  
5 geom_line(  
6   aes(group = day_night),  
7   color = "grey"  
8 ) +  
9 geom_point(  
10  aes(color = day_night,  
11     shape = lubridate::day(date) == 6)  
12 ) +  
13 labs(  
14   x = "Weekend date",  
15   y = "Reported bike shares",  
16   color = NULL,  
17   shape = NULL  
18 ) +  
19 theme_minimal(  
20   base_size = 15,  
21   base_family = "Lora"  
22 ) +  
23 theme(  
24   # Additional theme customizations  
25 )
```



Bonus: Use Shape to Encode Sat vs Sun

```
1 ggplot(  
2     filter(bikes, is_weekend == TRUE),  
3     aes(x = date, y = count)  
4 ) +  
5 geom_line(  
6     aes(group = day_night),  
7     color = "grey"  
8 ) +  
9 geom_point(  
10    aes(color = day_night,  
11         shape = lubridate::wday(date, label =  
12    ) +  
13    labs(  
14        x = "Weekend date",  
15        y = "Reported bike shares",  
16        color = NULL,  
17        shape = NULL  
18    ) +  
19    theme_minimal(  
20        base_size = 15,  
21        base_family = "Lora"  
22    ) +  
23    theme(  
24
```



Bonus: Use Shape to Encode Sat vs Sun

```
1 invisible(  
2   Sys.setlocale("LC_TIME", "C")  
3 )  
4  
5 ggplot(  
6   filter(bikes, is_weekend == TRUE),  
7   aes(x = date, y = count)  
8 ) +  
9   geom_line(  
10    aes(group = day_night),  
11    color = "grey"  
12 ) +  
13   geom_point(  
14     aes(color = day_night,  
15         shape = lubridate::wday(date, label =  
16 ) +  
17   labs(  
18     x = "Weekend date",  
19     y = "Reported bike shares",  
20     color = NULL,  
21     shape = NULL  
22 ) +  
23   theme_minimal()
```

