Graphic Design with ggplot2

Concepts of the `{ggplot2}` Package Pt. 2:
Facets, Scales, and Coordinate Systems

Cédric Scherer // rstudio::conf // July 2022

Setup

```
library(tidyverse)

bikes <- readr::read_csv(
  "https://raw.githubusercontent.com/z3tt/graphic-design-ggplot2/main/data/london-bikes-custom.csv",
  col_types = "Dcfffilllddddc"
)

bikes$season <- forcats::fct_inorder(bikes$season)

theme_set(theme_light(base_size = 14, base_family = "Roboto Condensed"))

theme_update(
  panel.grid.minor = element_blank(),
  plot.title = element_text(face = "bold"),
  legend.position = "top",
  plot.title.position = "plot"
)
```

Facets

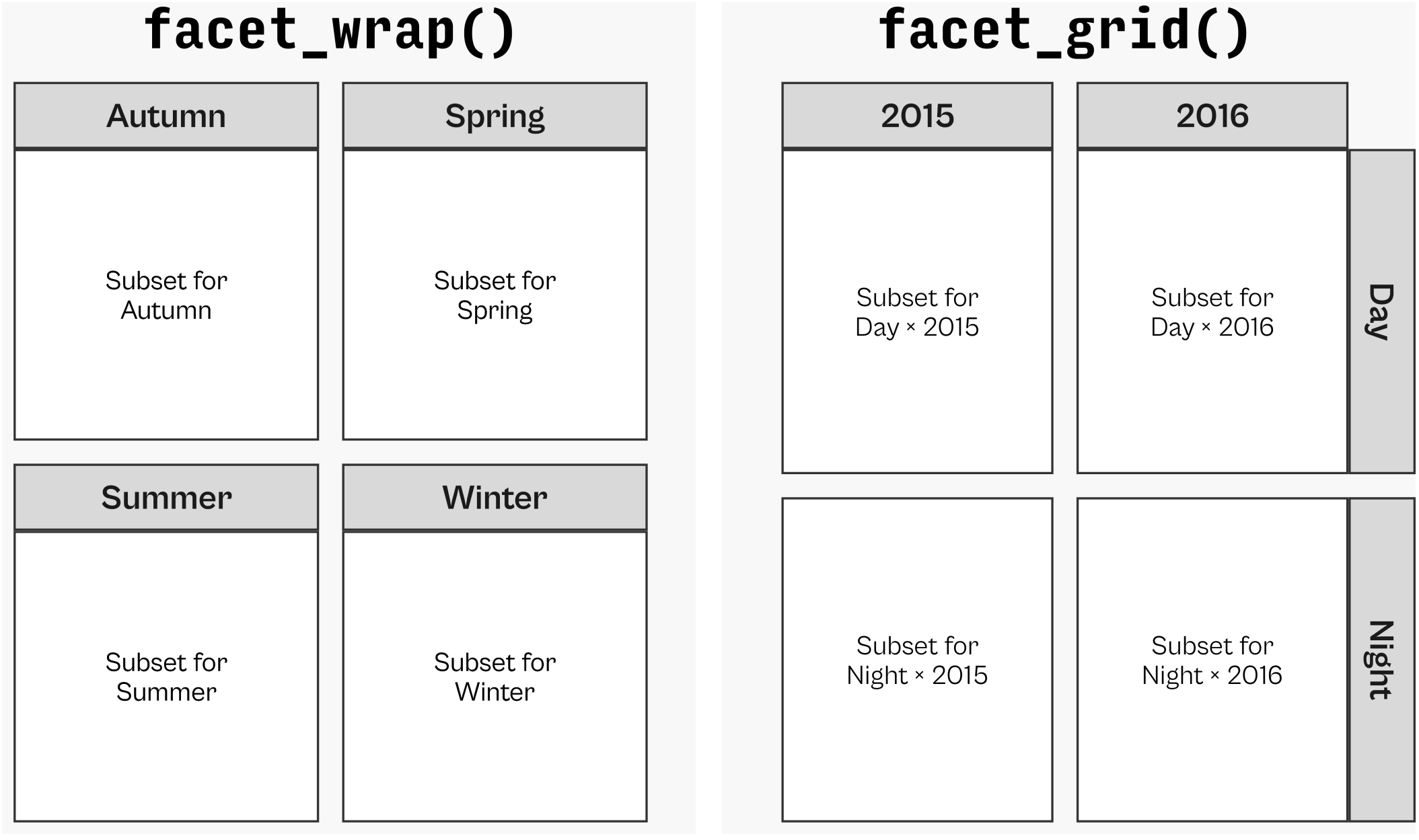
Cédric Scherer // rstudio::conf // July 2022

Facets

= split variables to multiple panels

Facets are also known as:

- small multiples
- trellis graphs
- lattice plots
- conditioning

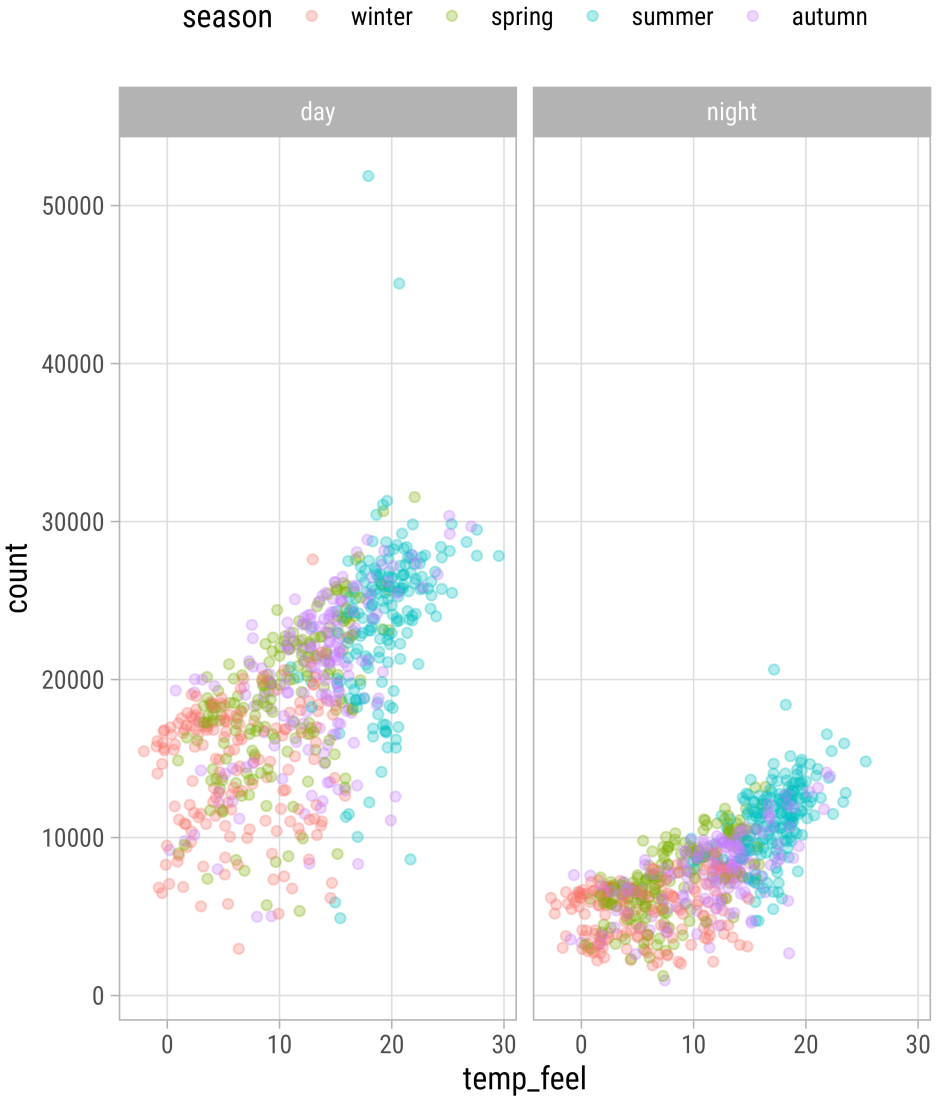

facet_wrap()
Autumn
Spring
Subset for Autumn
Subset for Spring
Summer
Winter
Subset for Summer
Subset for Winter
facet_grid()
2015
2016
Day
Night
Subset for Day × 2015
Subset for Day × 2016
Subset for Night × 2015
Subset for Night × 2016

Setup

```
g <-
  ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = season)
  ) +
  geom_point(
    alpha = .3,
    guide = "none"
  )

g
```

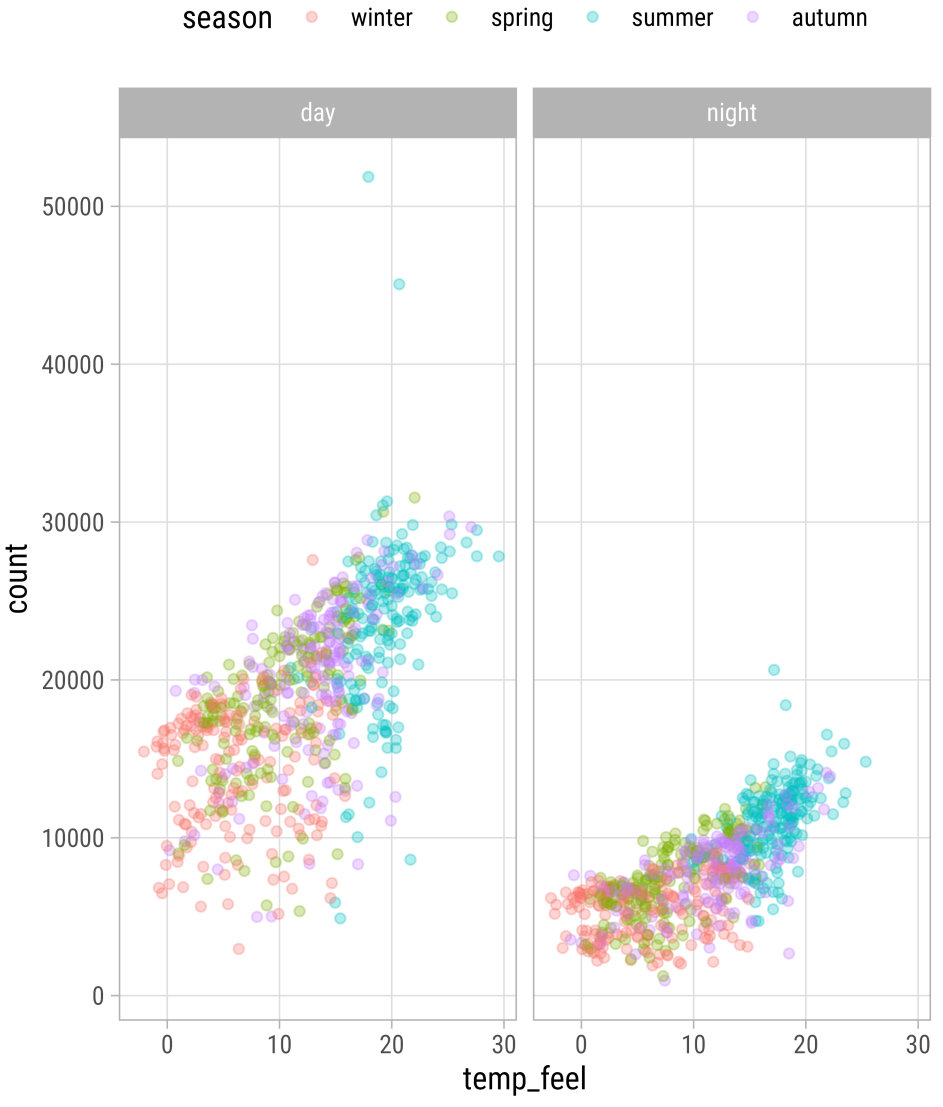
Wrapped Facet

```
g +
  facet_wrap(
    vars(day_night)
  )
```



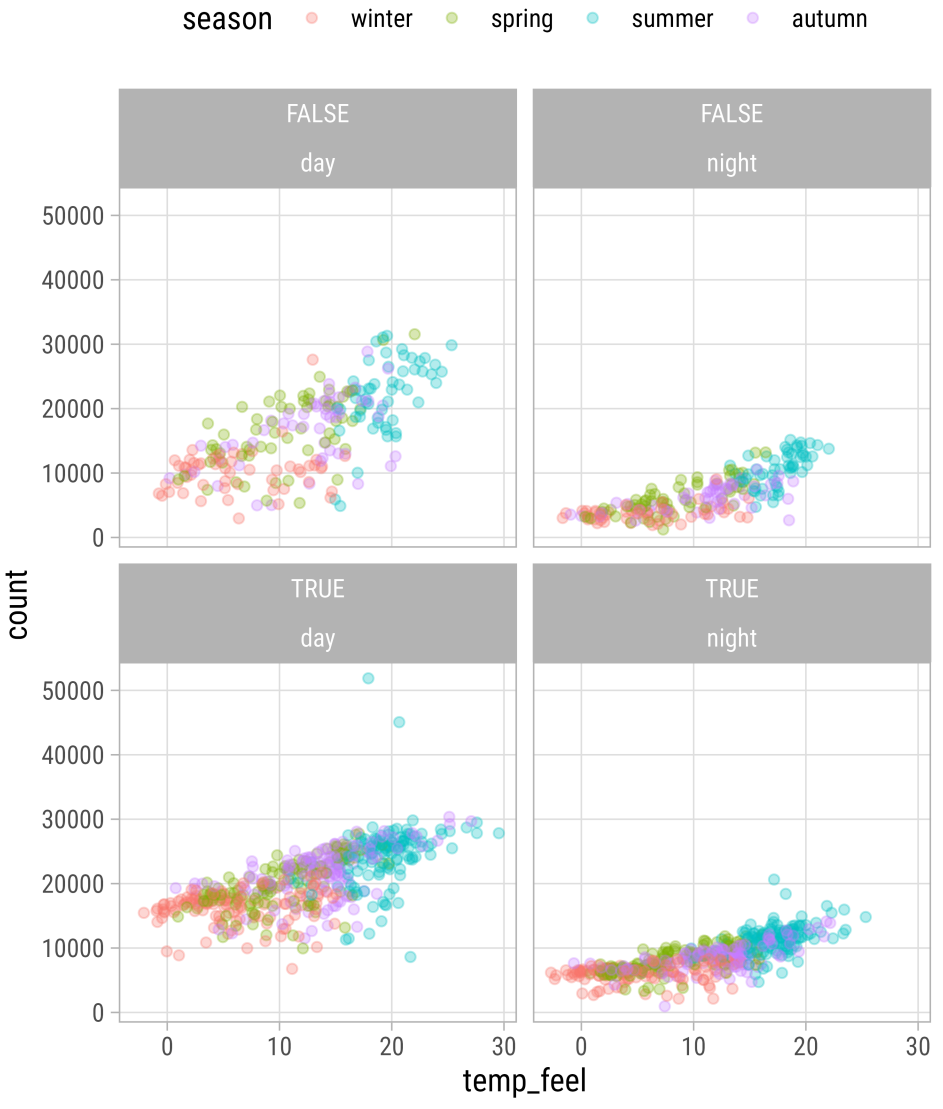
Wrapped Facet

```
1 g +
2   facet_wrap(
3     ~ day_night
4   )
```



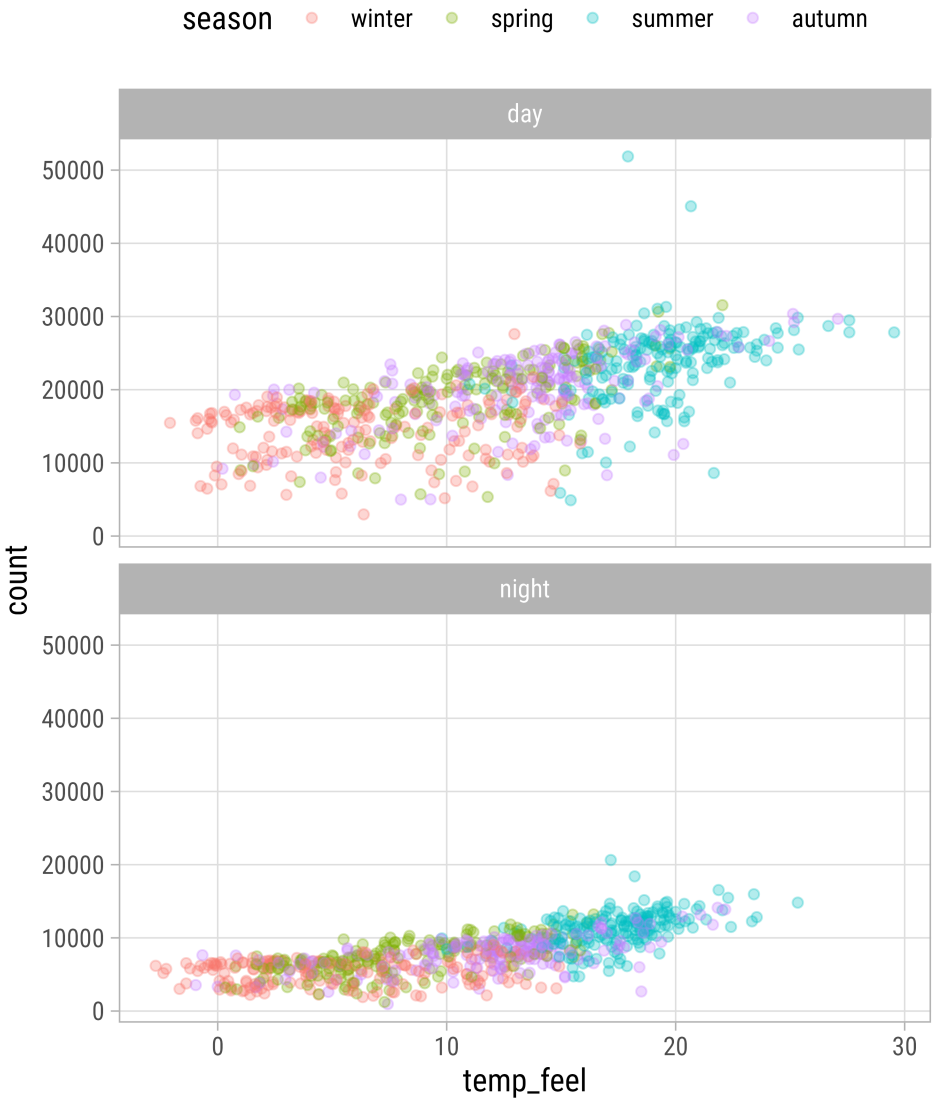
Facet Multiple Variables

```
1 g +
2   facet_wrap(
3     ~ is_workday + day_night
4   )
```



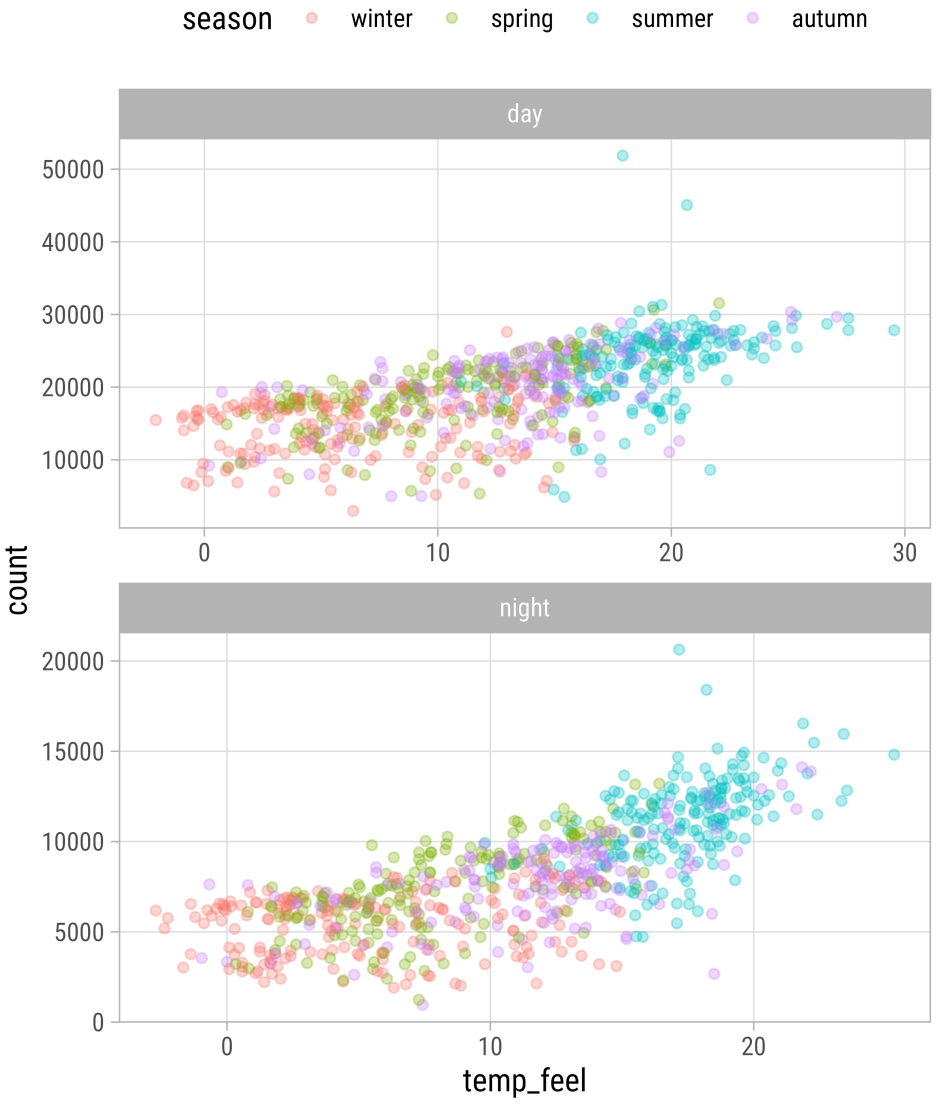
Facet Options: Cols + Rows

```
g +
  facet_wrap(
    ~ day_night,
    ncol = 1
  )
```



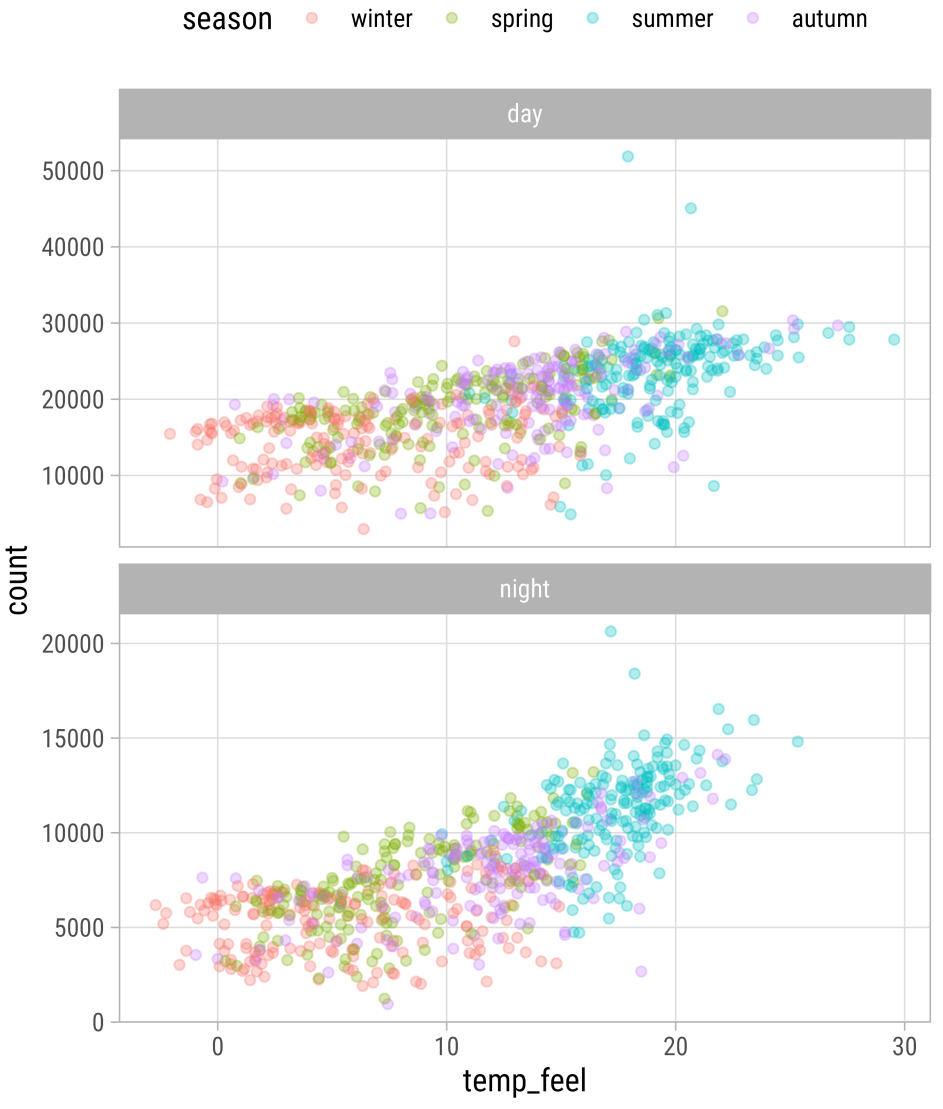
Facet Options: Free Scaling

```
g +
  facet_wrap(
    ~ day_night,
    ncol = 1,
    scales = "free"
  )
```



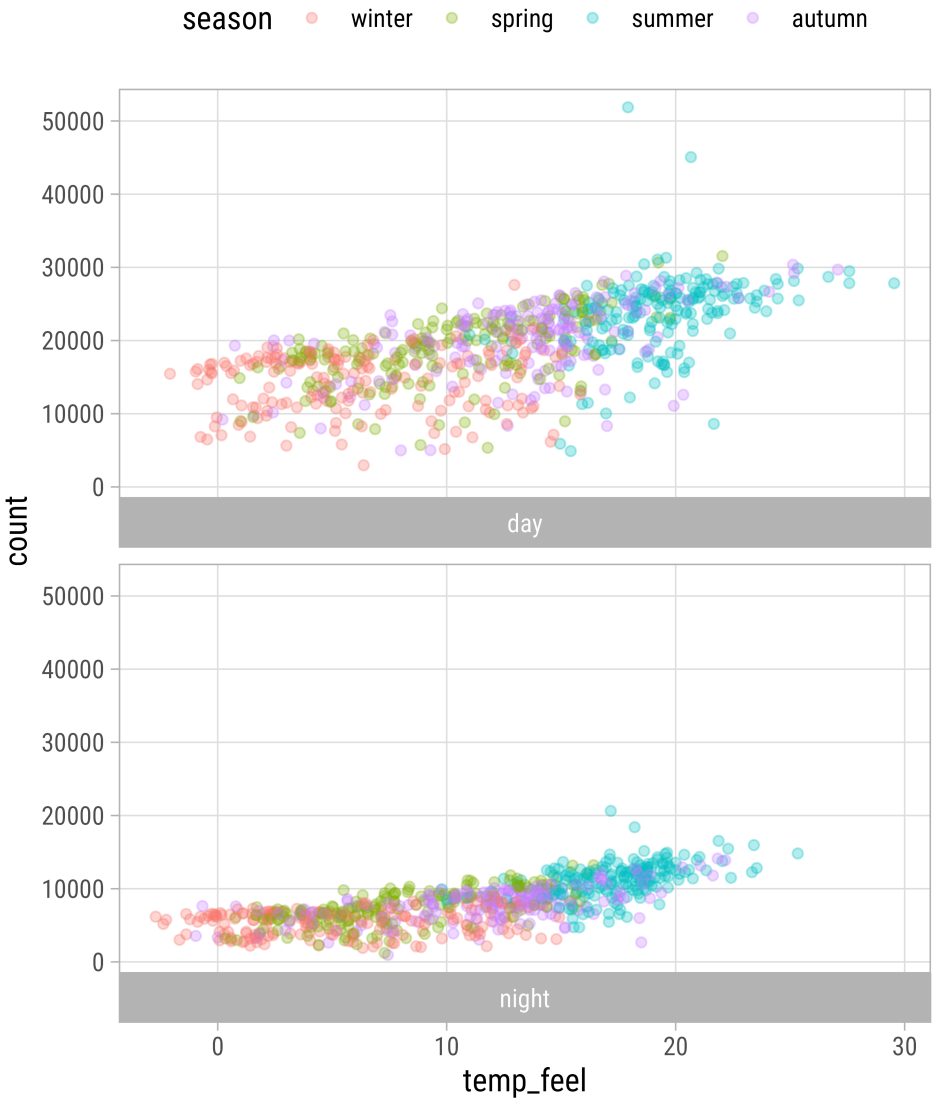
Facet Options: Free Scaling

```
g +
  facet_wrap(
    ~ day_night,
    ncol = 1,
    scales = "free_y"
  )
```



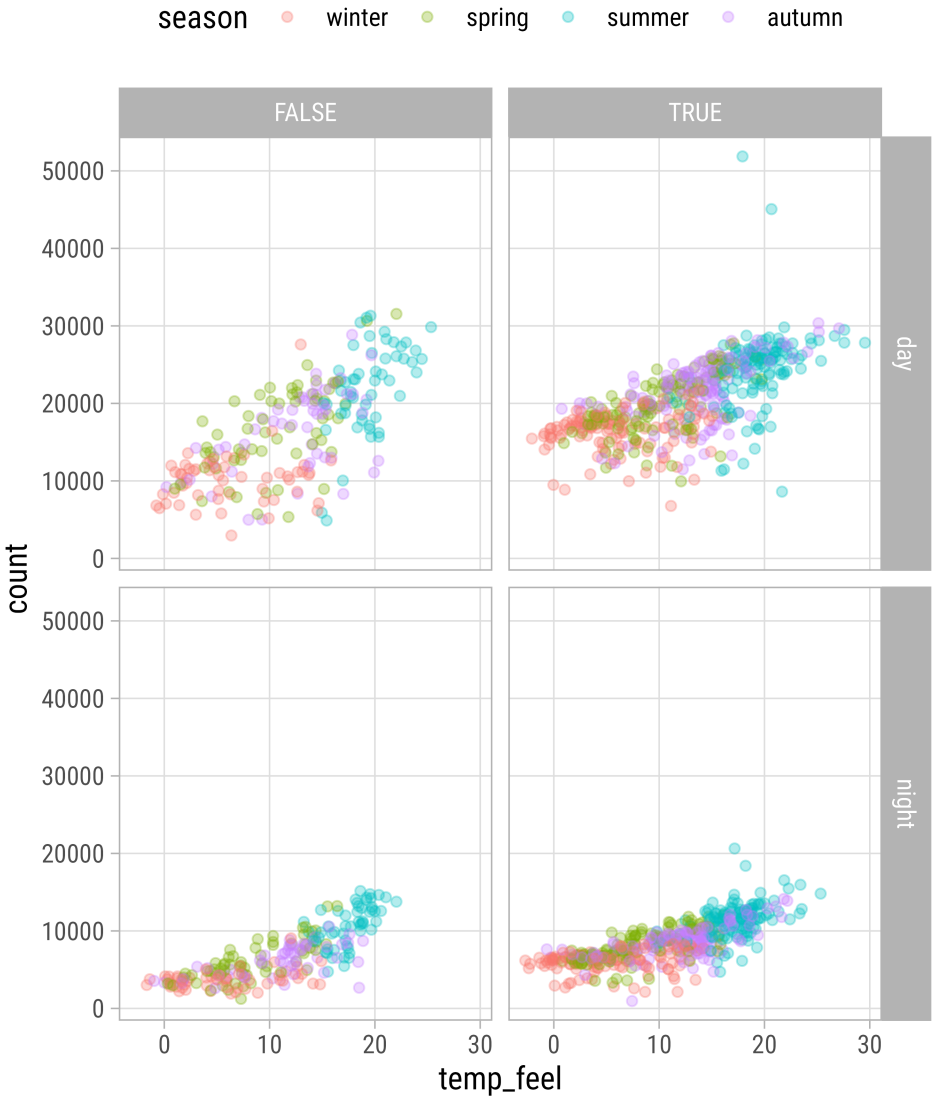
Facet Options: Switch Labels

```
g +
  facet_wrap(
    ~ day_night,
    ncol = 1,
    switch = "x"
  )
```



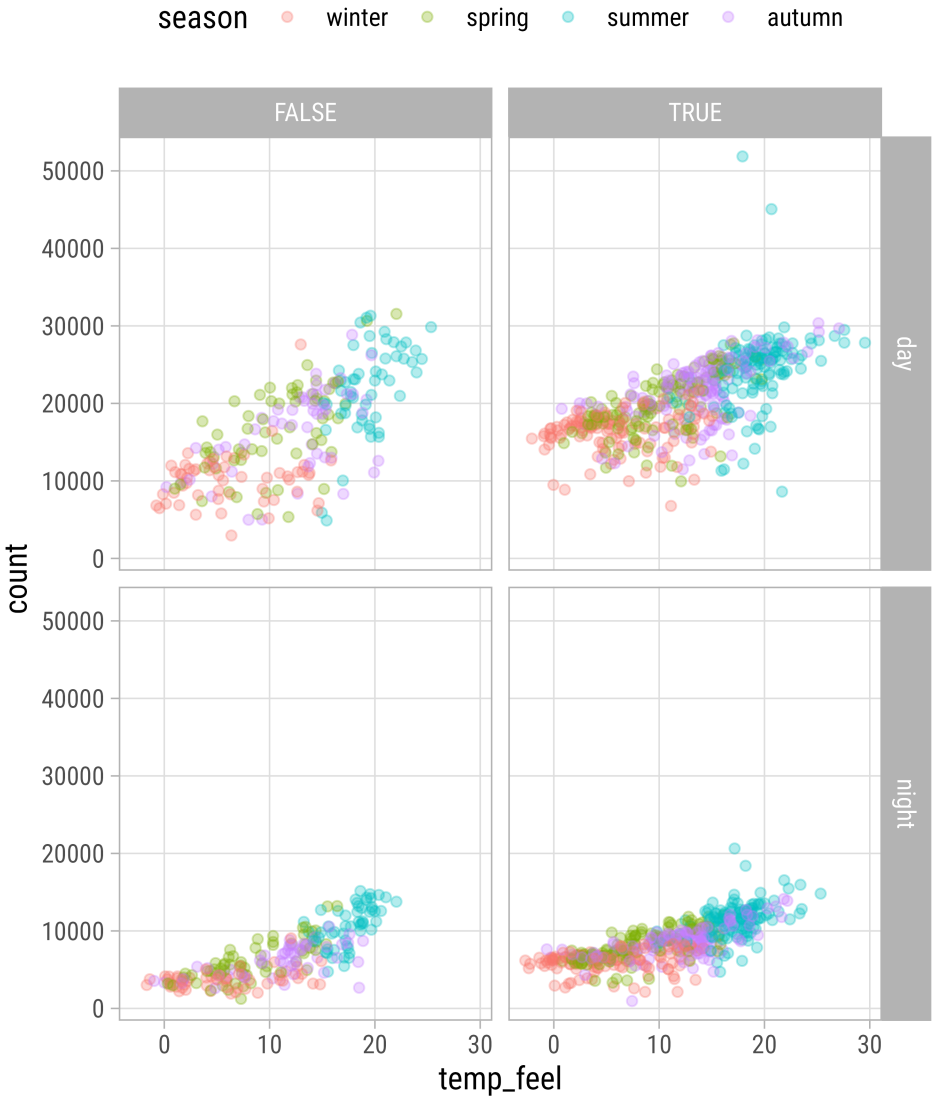
Gridded Facet

```
1 g +
2   facet_grid(
3     rows = vars(day_night),
4     cols = vars(is_workday)
5   )
```



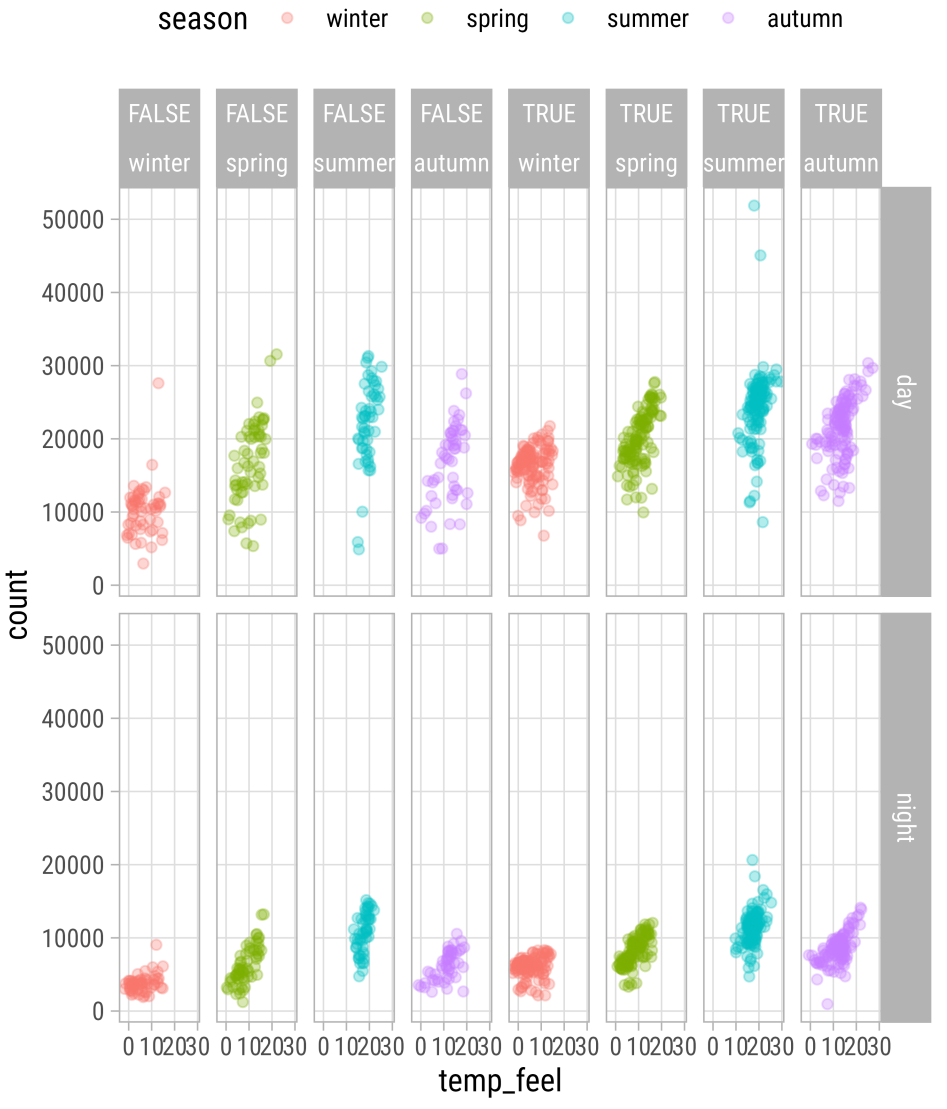
Gridded Facet

```
1 g +
2   facet_grid(
3     day_night ~ is_workday
4   )
```



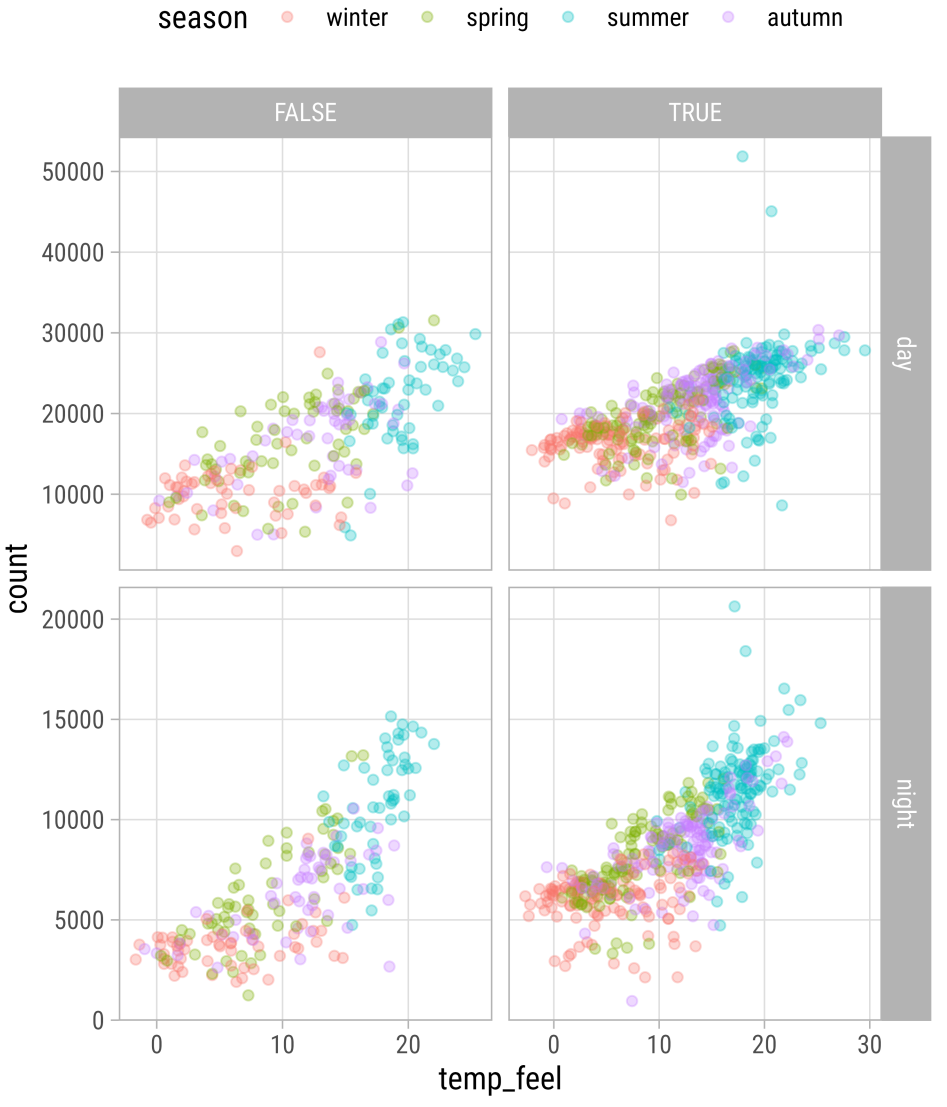
Facet Multiple Variables

```
1 g +
2   facet_grid(
3     day_night ~ is_workday + season
4   )
```



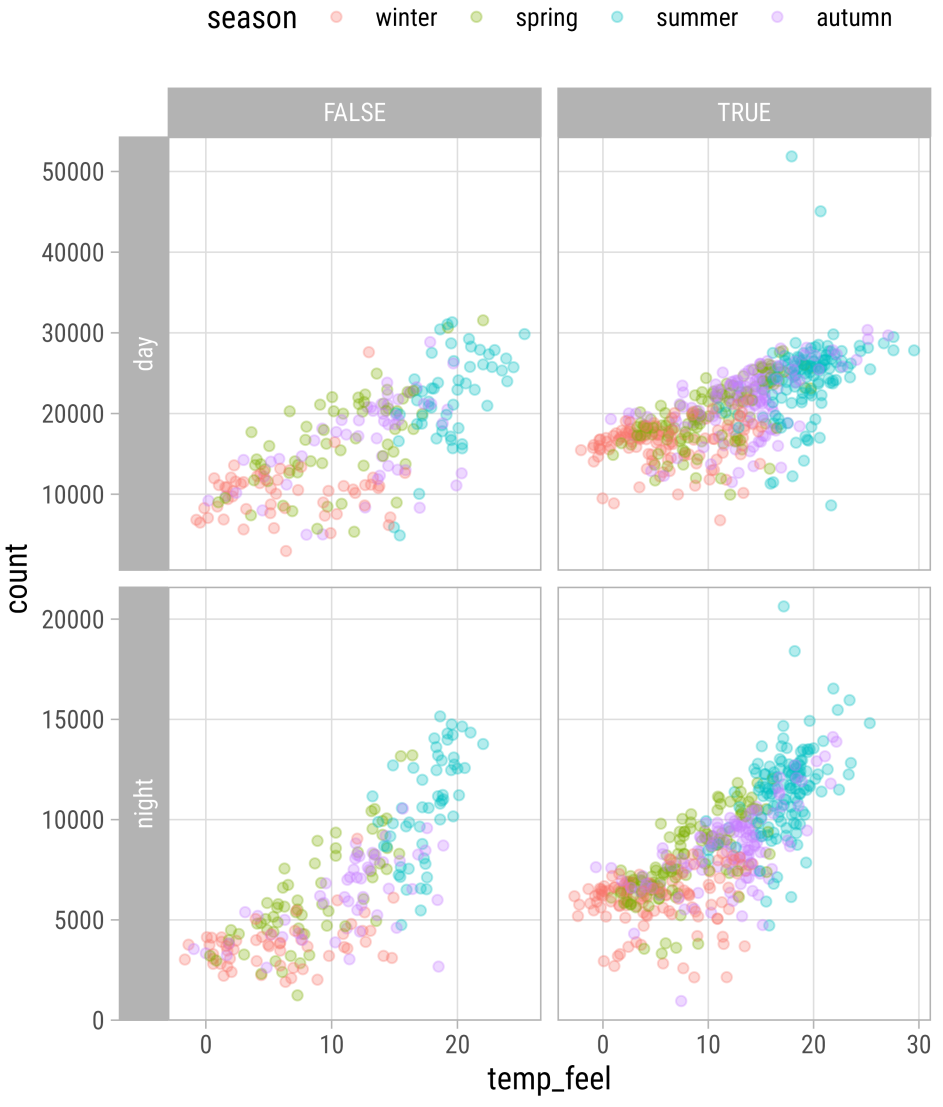
Facet Options: Free Scaling

```
g +
  facet_grid(
    day_night ~ is_workday,
    scales = "free"
  )
```



Facet Options: Switch Labels

```
1 g +
2   facet_grid(
3     day_night ~ is_workday,
4     scales = "free",
5     switch = "y"
6   )
```

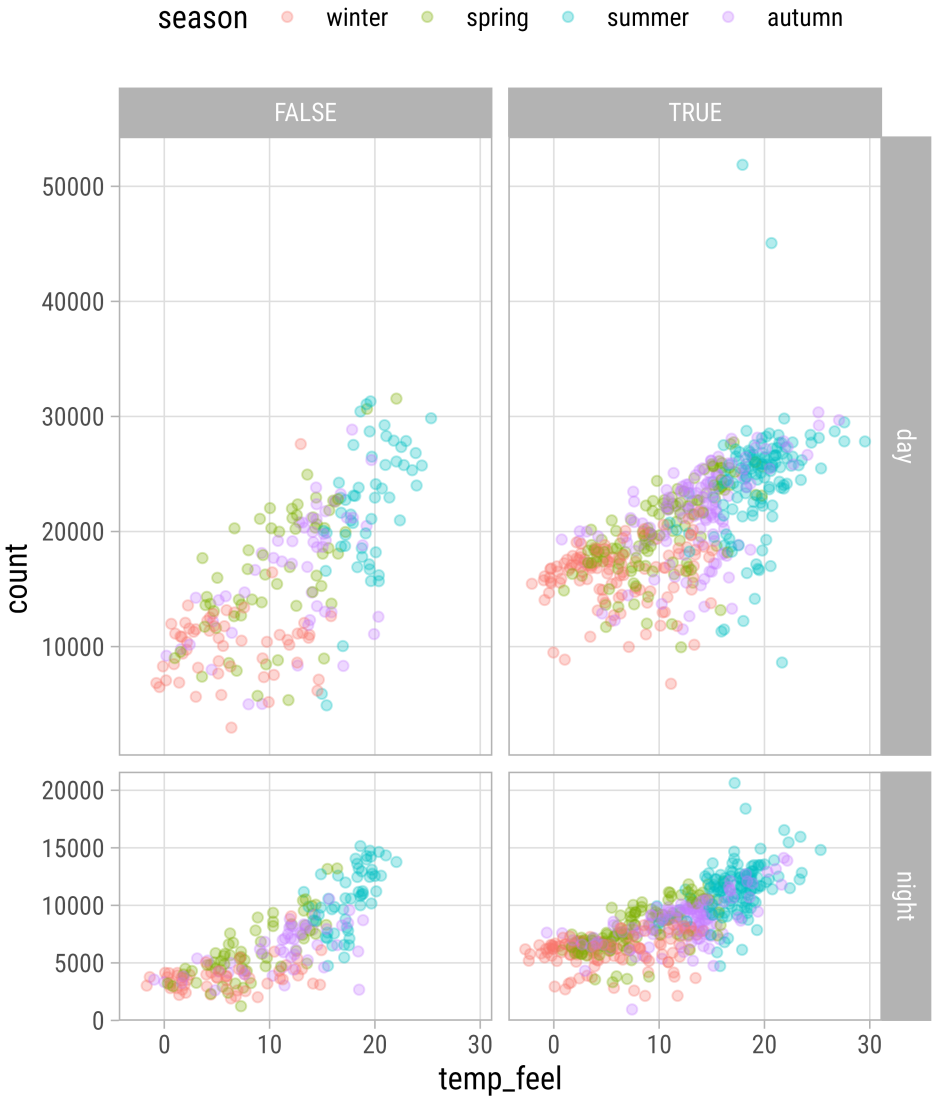


Facet Options: Proportional Spacing

```
g +
  facet_grid(
    day_night ~ is_workday,
    scales = "free",
    space = "free"
  )
```

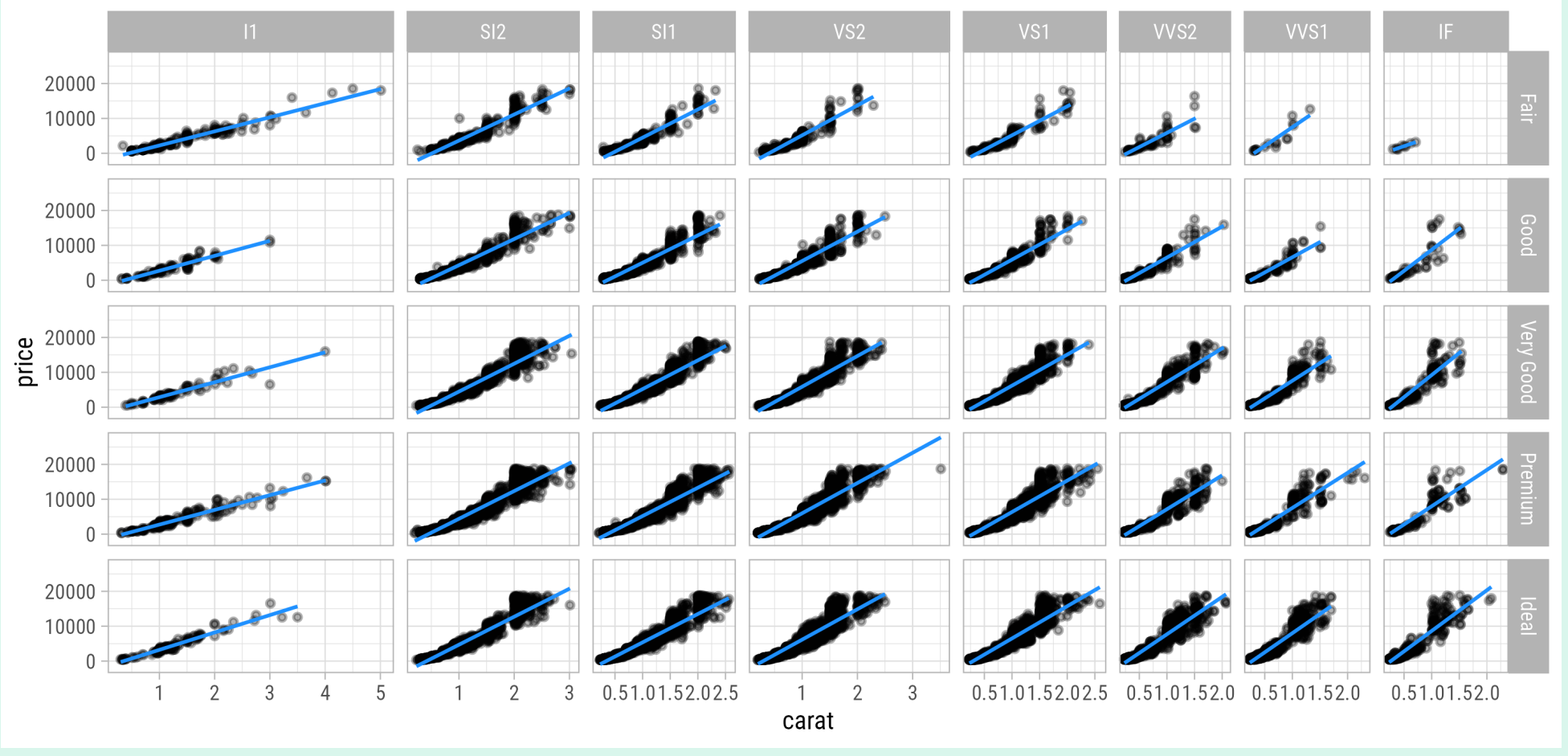
Facet Options: Proportional Spacing

```
g +
  facet_grid(
    day_night ~ is_workday,
    scales = "free_y",
    space = "free_y"
  )
```



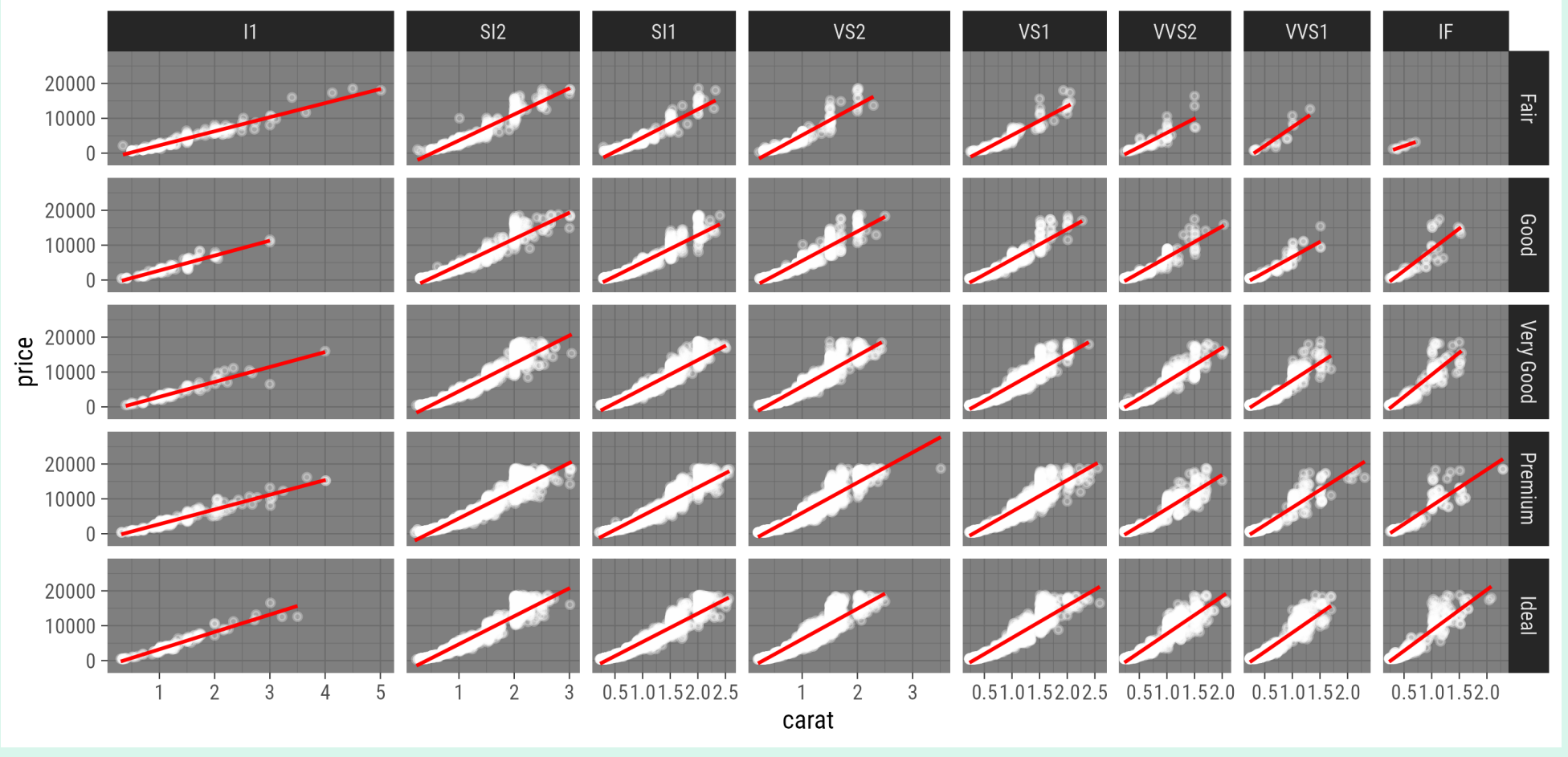
Your Turn!

Create the following facet from the `diamonds` data.



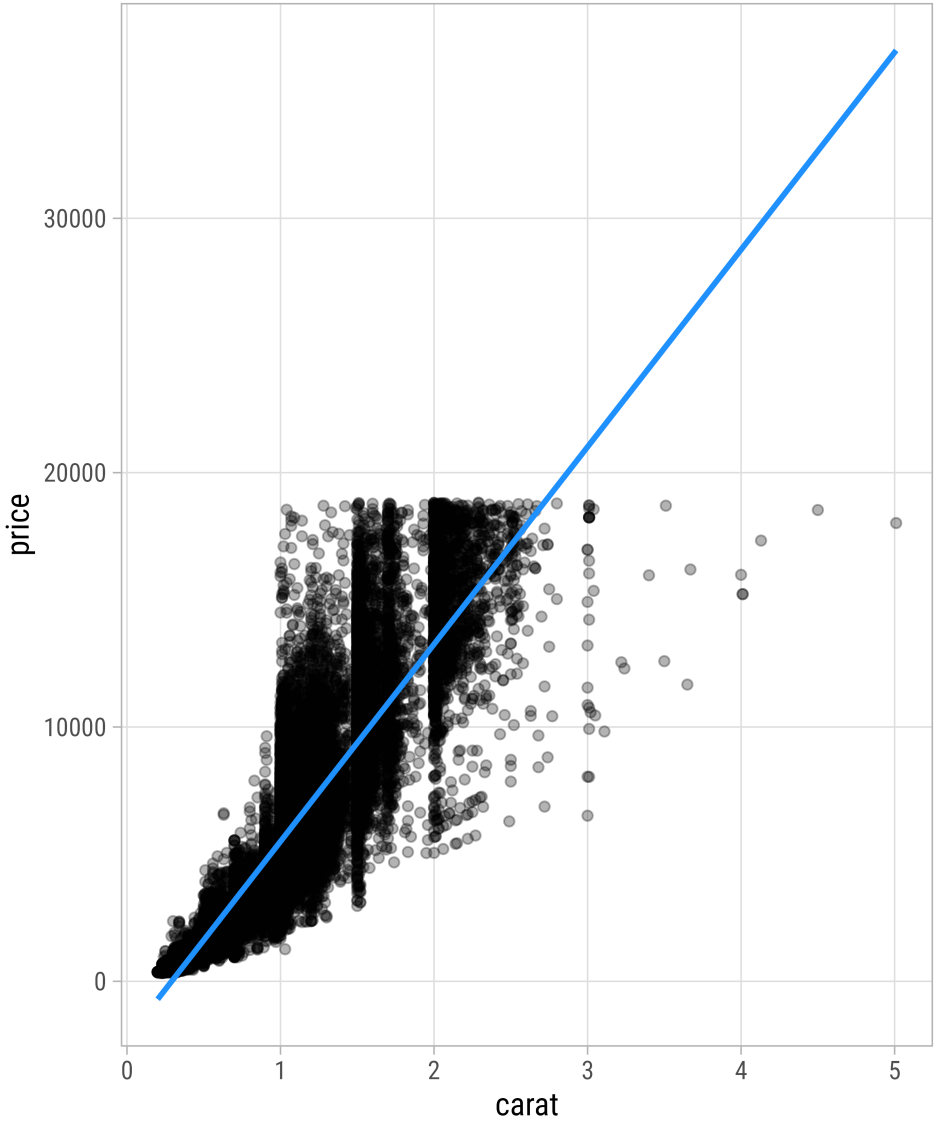
Your Turn!

Bonus: Create this bloody-dark version.



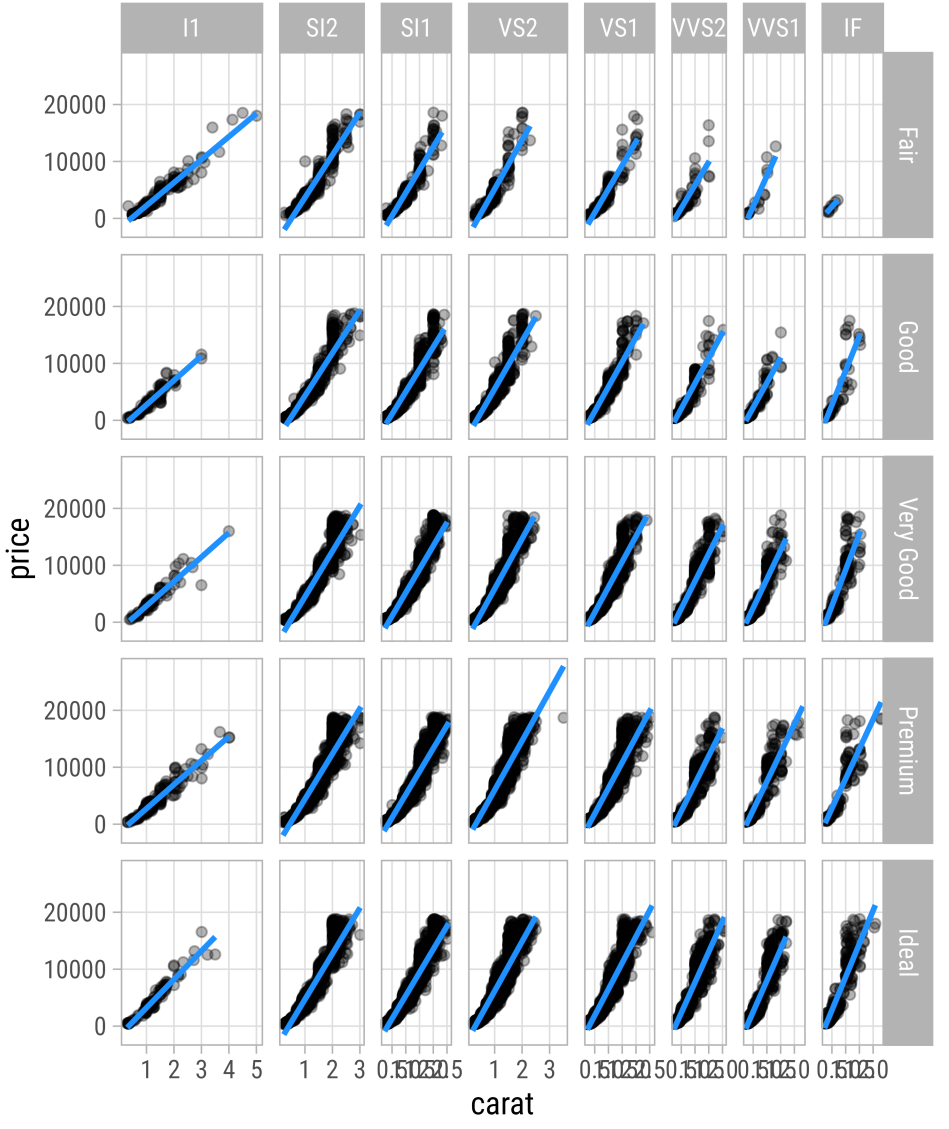
Diamonds Facet

```
ggplot(
    diamonds,
    aes(x = carat, y = price)
  ) +
  geom_point(
    alpha = .3
  ) +
  geom_smooth(
    method = "lm",
    se = FALSE,
    color = "dodgerblue"
  )
```



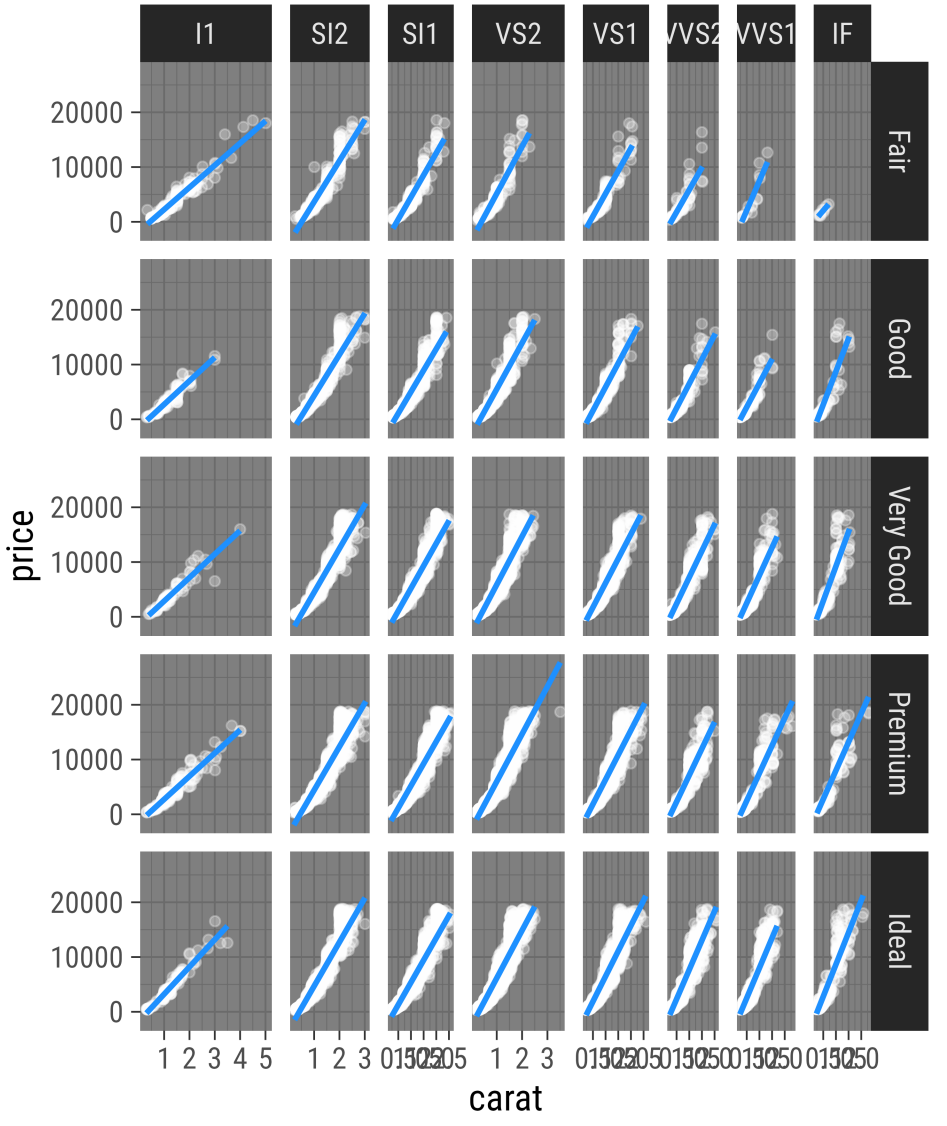
Diamonds Facet

```
ggplot(
    diamonds,
    aes(x = carat, y = price)
  ) +
  geom_point(
    alpha = .3
  ) +
  geom_smooth(
    method = "lm",
    se = FALSE,
    color = "dodgerblue"
  ) +
  facet_grid(
    cut ~ clarity,
    space = "free_x",
    scales = "free_x"
  )
```



Diamonds Facet (Dark Theme Bonus)

```
ggplot(
    diamonds,
    aes(x = carat, y = price)
  ) +
  geom_point(
    alpha = .3,
    color = "white"
  ) +
  geom_smooth(
    method = "lm",
    se = FALSE,
    color = "dodgerblue"
  ) +
  facet_grid(
    cut ~ clarity,
    space = "free_x",
    scales = "free_x"
  ) +
  theme_dark(
    base_size = 14,
    base_family = "Roboto Condensed"
  )
```



Scales

Cédric Scherer // rstudio::conf // July 2022

Scales

= translate between variable ranges and property ranges

- feels-like temperature ⇄ x
- reported bike shares ⇄ y
- season ⇄ color
- year ⇄ shape
- ...

Scales

The `scale_*()` components control the properties of all the **aesthetic dimensions mapped to the data.**

Consequently, there are `scale_*()` functions for all aesthetics such as:

- **positions** via `scale_x_*()` and `scale_y_*()`
- **colors** via `scale_color_*()` and `scale_fill_*()`
- **sizes** via `scale_size_*()` and `scale_radius_*()`
- **shapes** via `scale_shape_*()` and `scale_linetype_*()`
- **transparency** via `scale_alpha_*()`

Scales

The `scale_*()` components control the properties of all the **aesthetic dimensions mapped to the data.**

The extensions (`*`) can be filled by e.g.:

- `continuous()`, `discrete()`, `reverse()`, `log10()`, `sqrt()`, `date()` for positions
- `continuous()`, `discrete()`, `manual()`, `gradient()`, `gradient2()`, `brewer()` for colors
- `continuous()`, `discrete()`, `manual()`, `ordinal()`, `area()`, `date()` for sizes
- `continuous()`, `discrete()`, `manual()`, `ordinal()` for shapes
- `continuous()`, `discrete()`, `manual()`, `ordinal()`, `date()` for transparency

CONTINUOUS

measured data, can have ∞ values within possible range.

I AM 3.1" TALL

I WEIGH 34.16 grams

DISCRETE

OBSERVATIONS CAN ONLY EXIST AT LIMITED VALUES, OFTEN COUNTS.

I HAVE 8 LEGS and 4 SPOTS!

@allison_horst

Illustration by Allison Horst

Continuous vs. Discrete in {ggplot2}

Continuous:
quantitative or numerical data

- height
- weight
- age
- counts

Discrete:
qualitative or categorical data

- species
- sex
- study sites
- age group

Continuous vs. Discrete in {ggplot2}

Continuous:
quantitative or numerical data

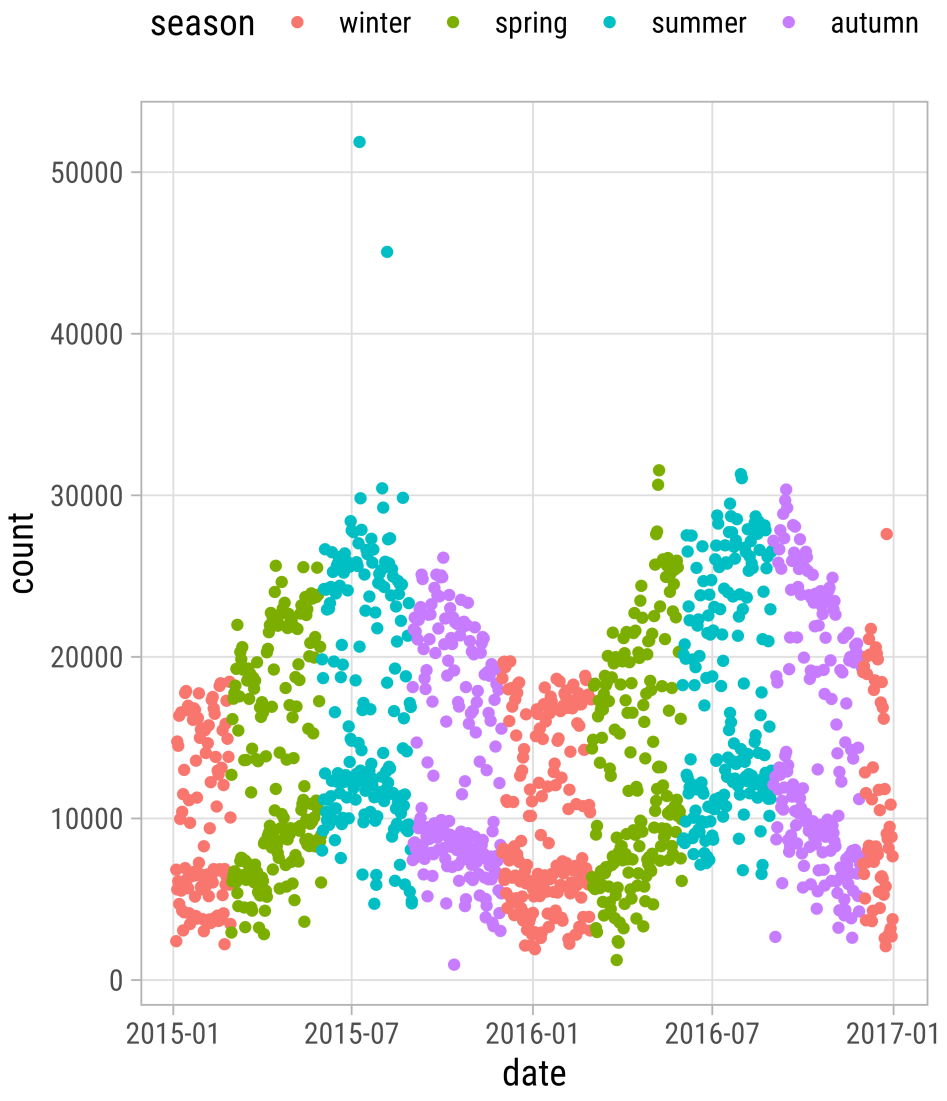
- height (continuous)
- weight (continuous)
- age (continuous or discrete)
- counts (discrete)

Discrete:
qualitative or categorical data

- species (nominal)
- sex (nominal)
- study site (nominal or ordinal)
- age group (ordinal)

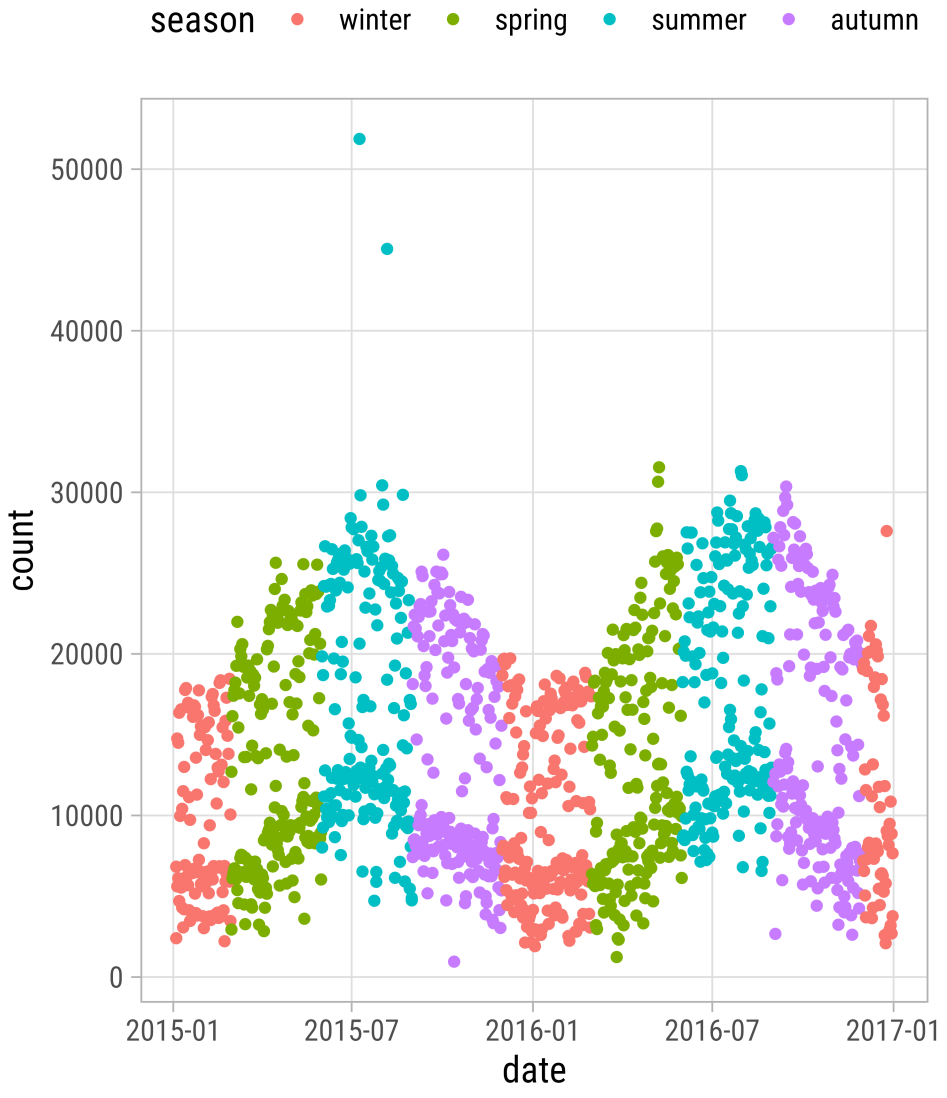
Aesthetics + Scales

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point()
```



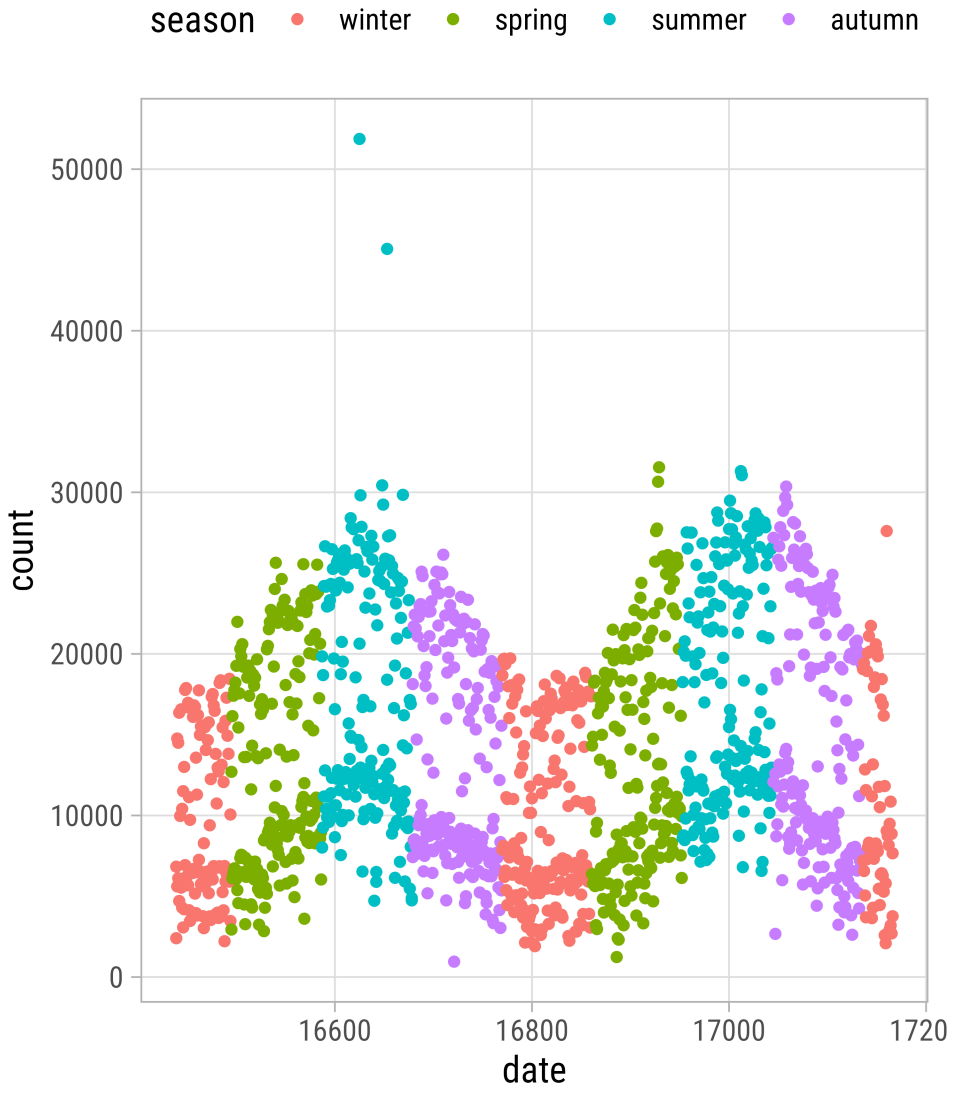
Aesthetics + Scales

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous() +
  scale_color_discrete()
```



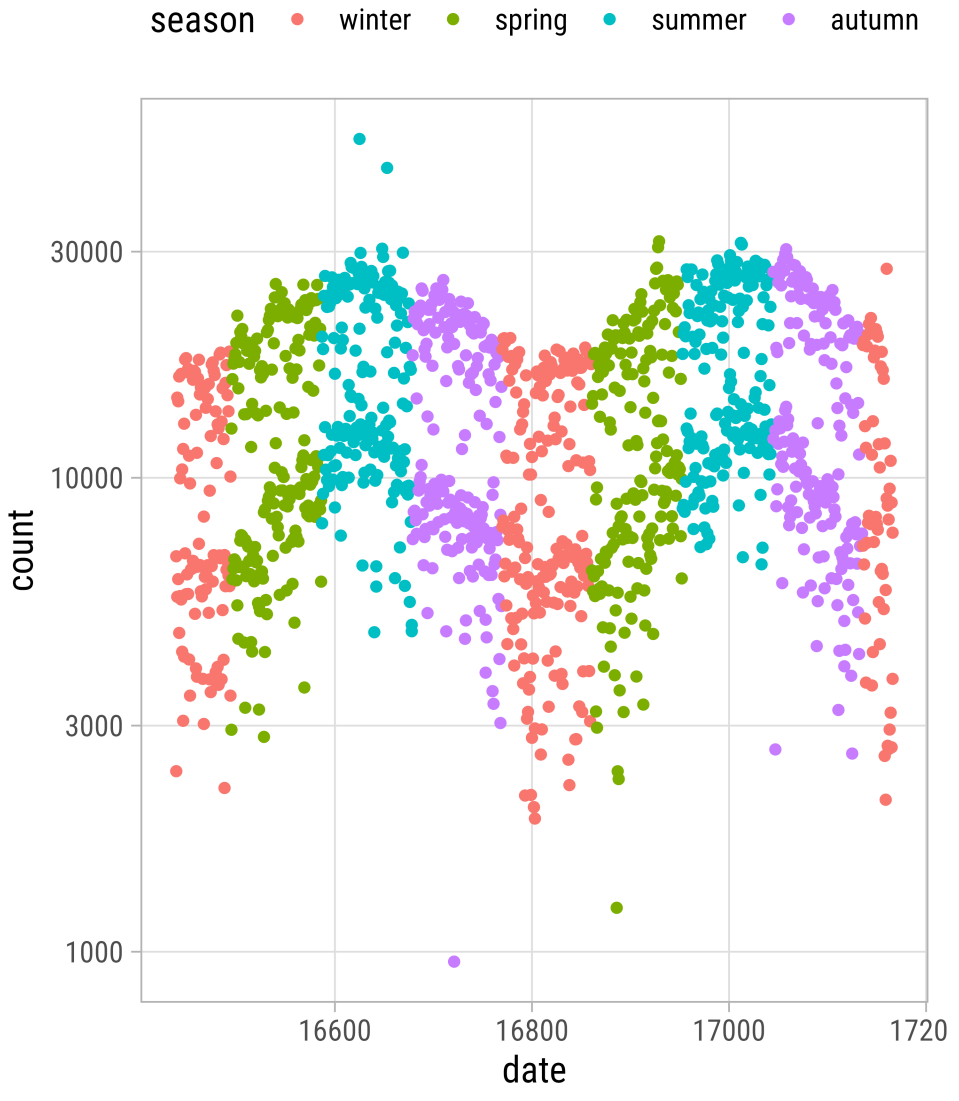
Scales

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_continuous() +
  scale_y_continuous() +
  scale_color_discrete()
```



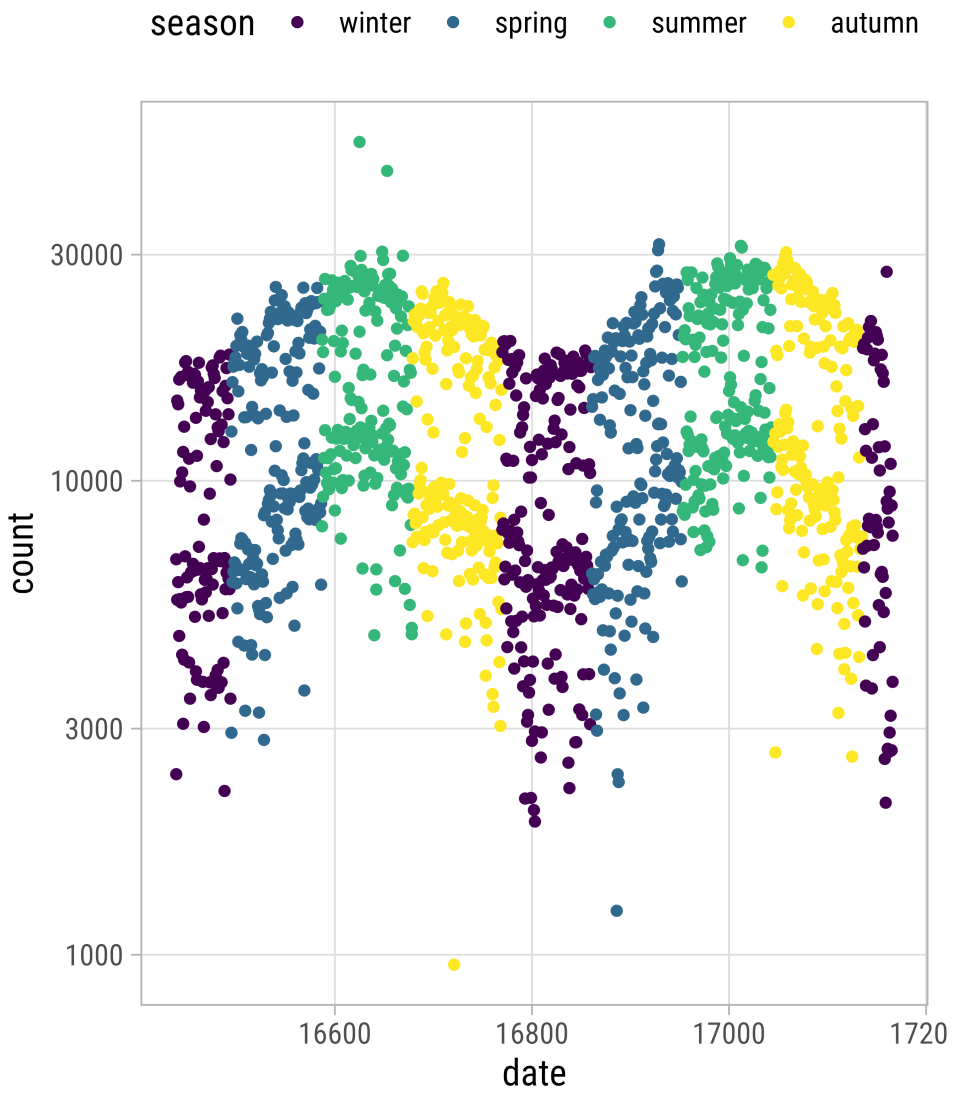
Scales

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_continuous() +
  scale_y_log10() +
  scale_color_discrete()
```



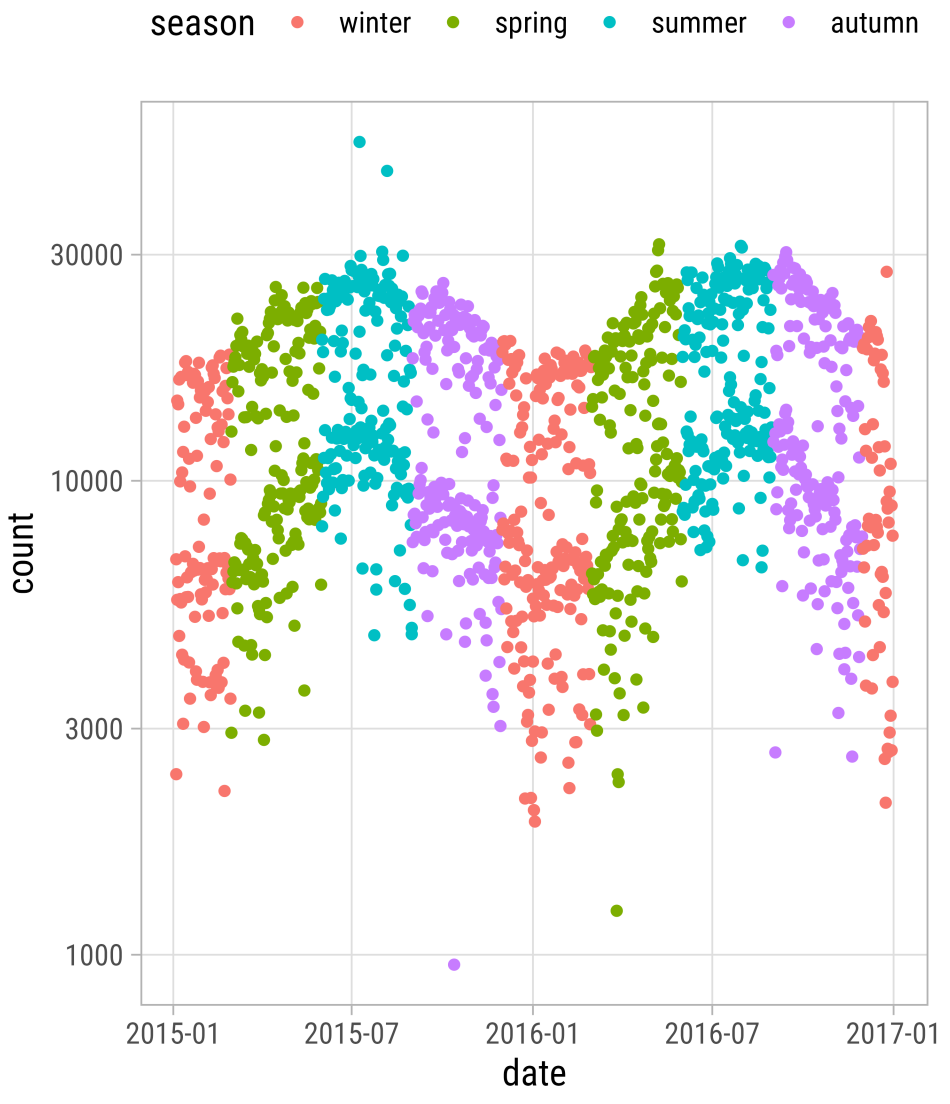
Scales

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_continuous() +
  scale_y_log10() +
  scale_color_viridis_d()
```



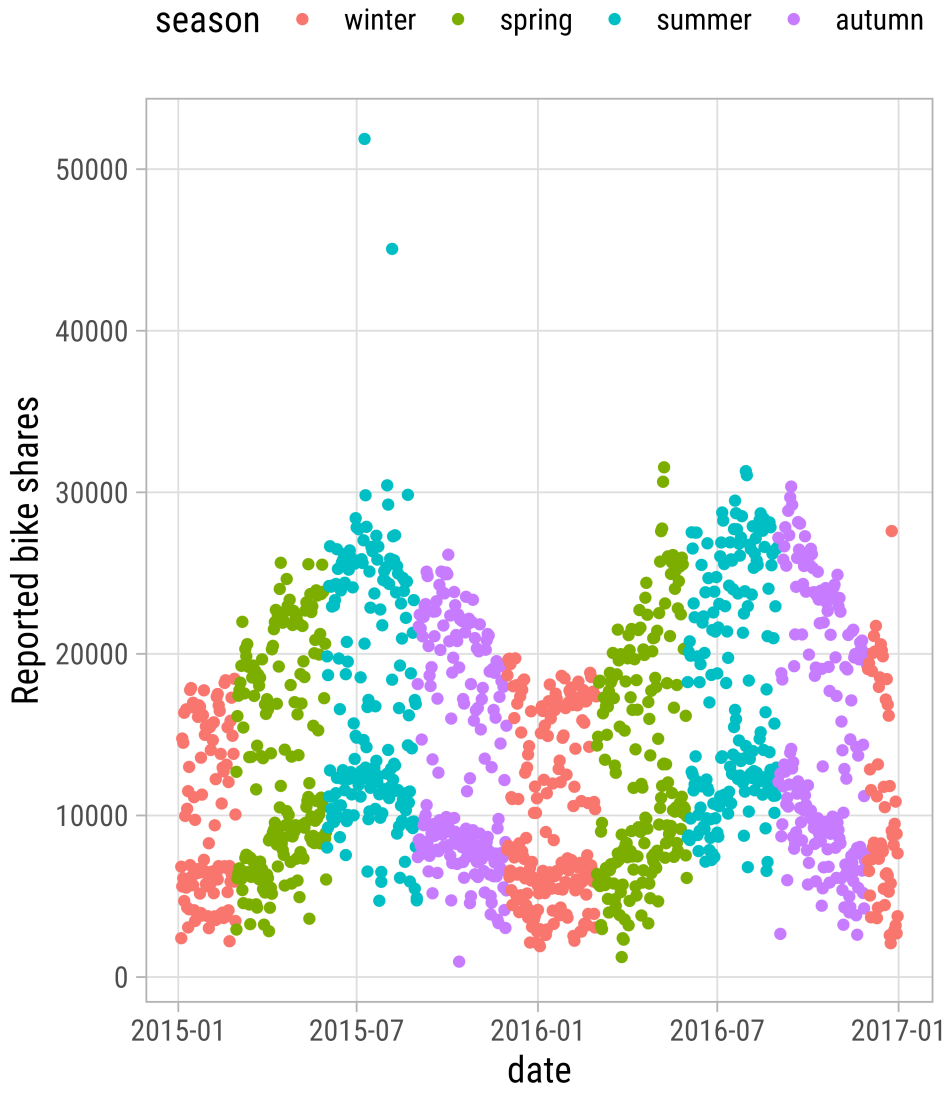
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    trans = "log10"
  ) +
  scale_color_discrete()
```



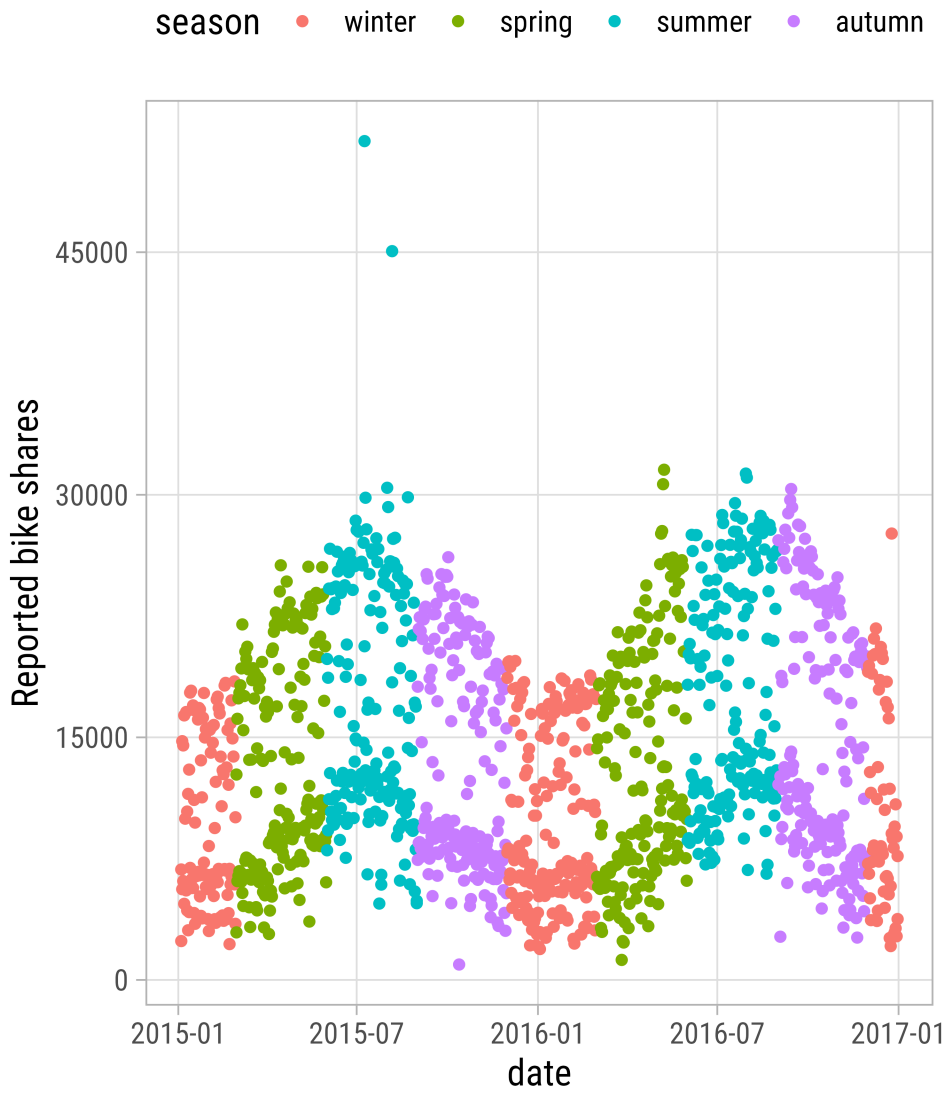
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares"
  ) +
  scale_color_discrete()
```



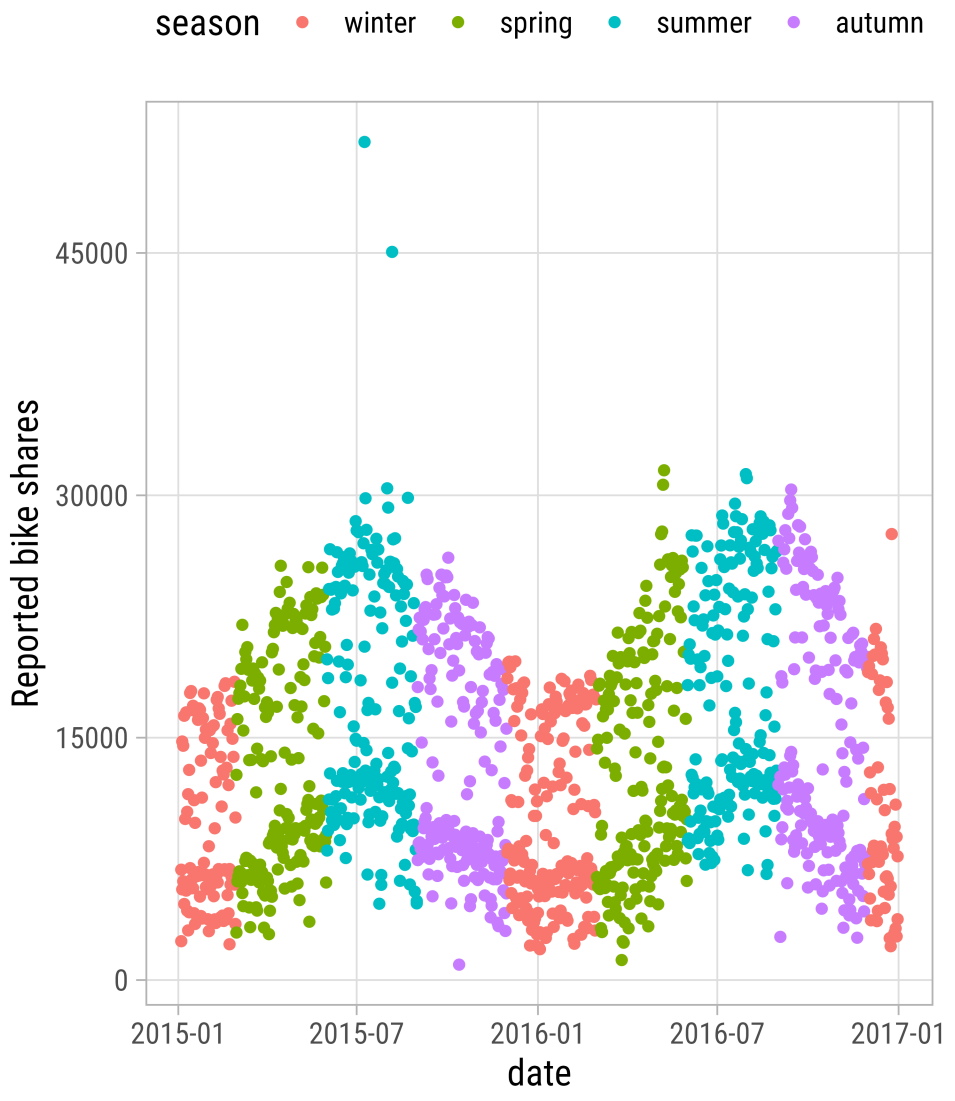
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = seq(0, 60000, by = 15000)
  ) +
  scale_color_discrete()
```



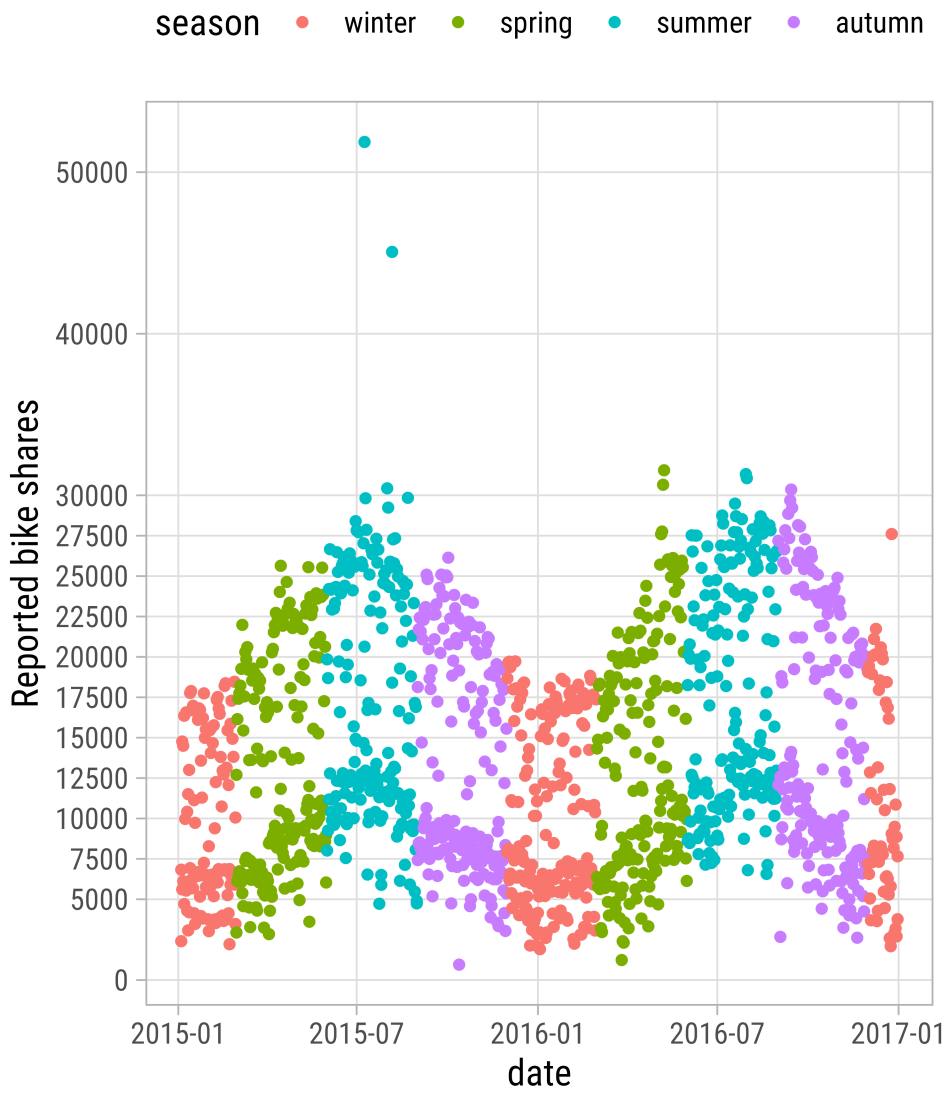
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = 0:4*15000
  ) +
  scale_color_discrete()
```



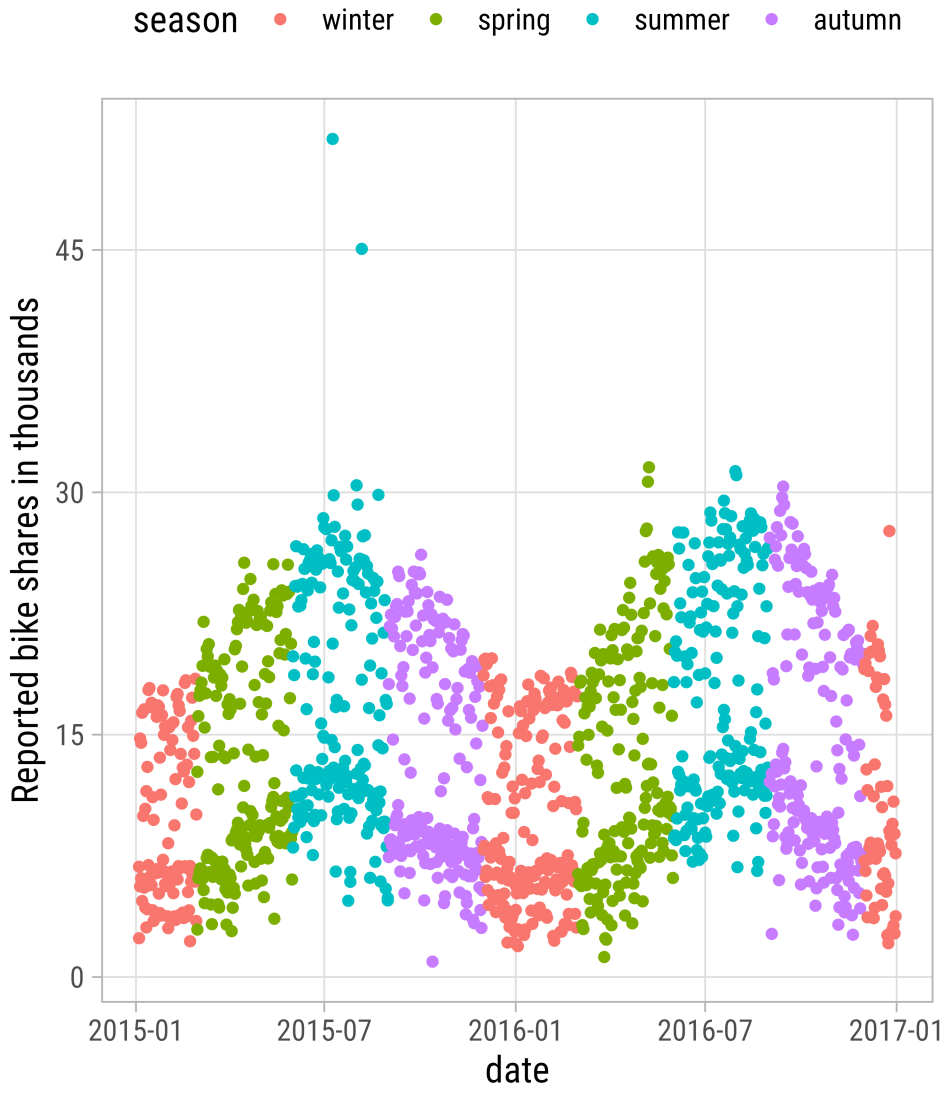
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = c(0, seq(5000, 30000, by = 2500),
  ) +
  scale_color_discrete()
```



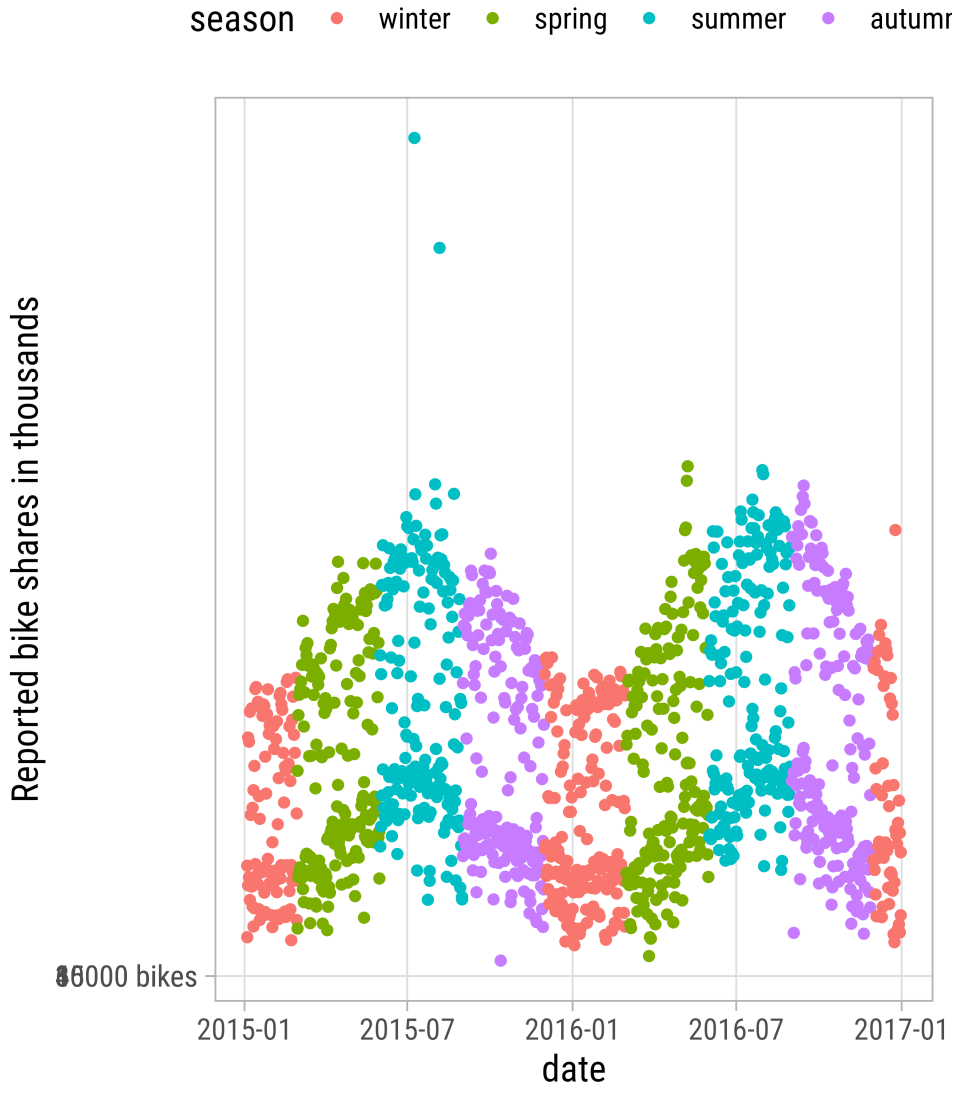
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares in thousands"
    breaks = 0:4*15000,
    labels = 0:4*15
  ) +
  scale_color_discrete()
```



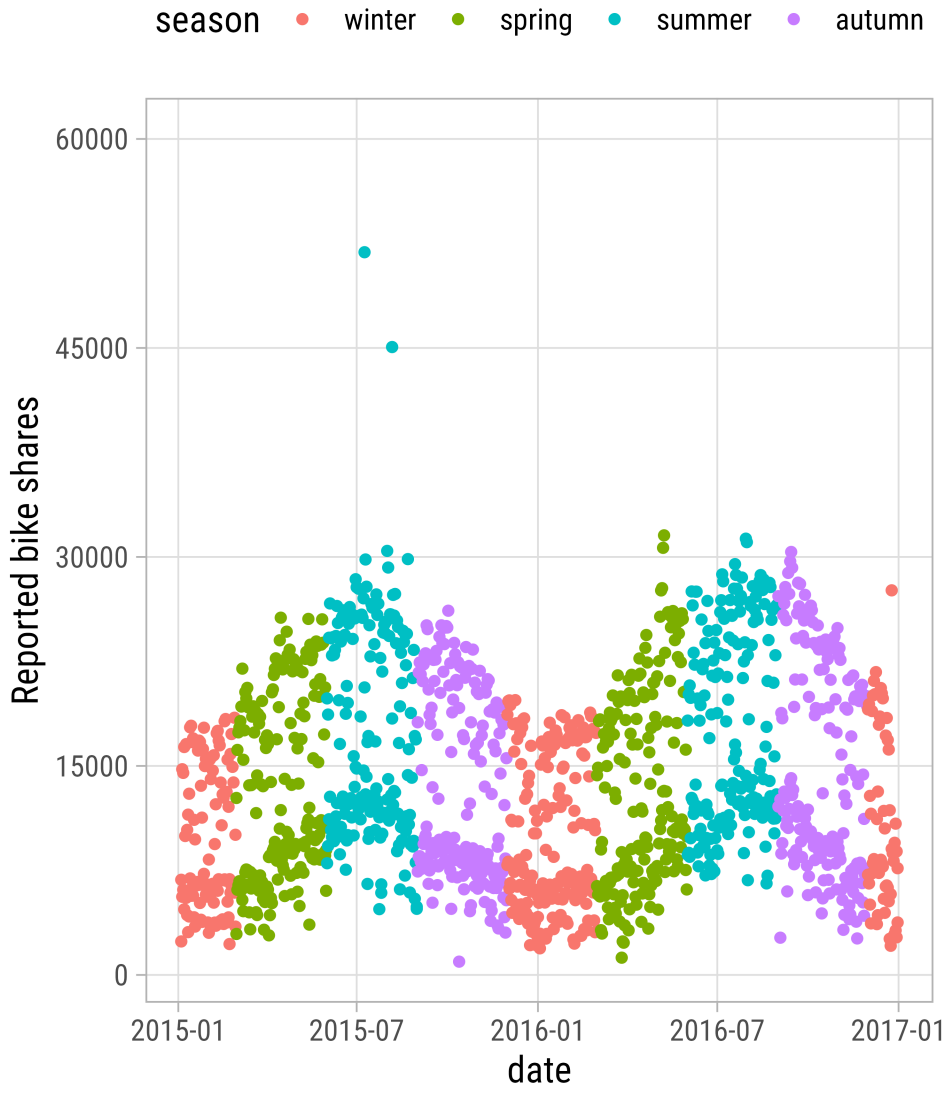
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares in thousands"
    breaks = 0:4,
    labels = paste(0:4*15000, "bikes")
  ) +
  scale_color_discrete()
```



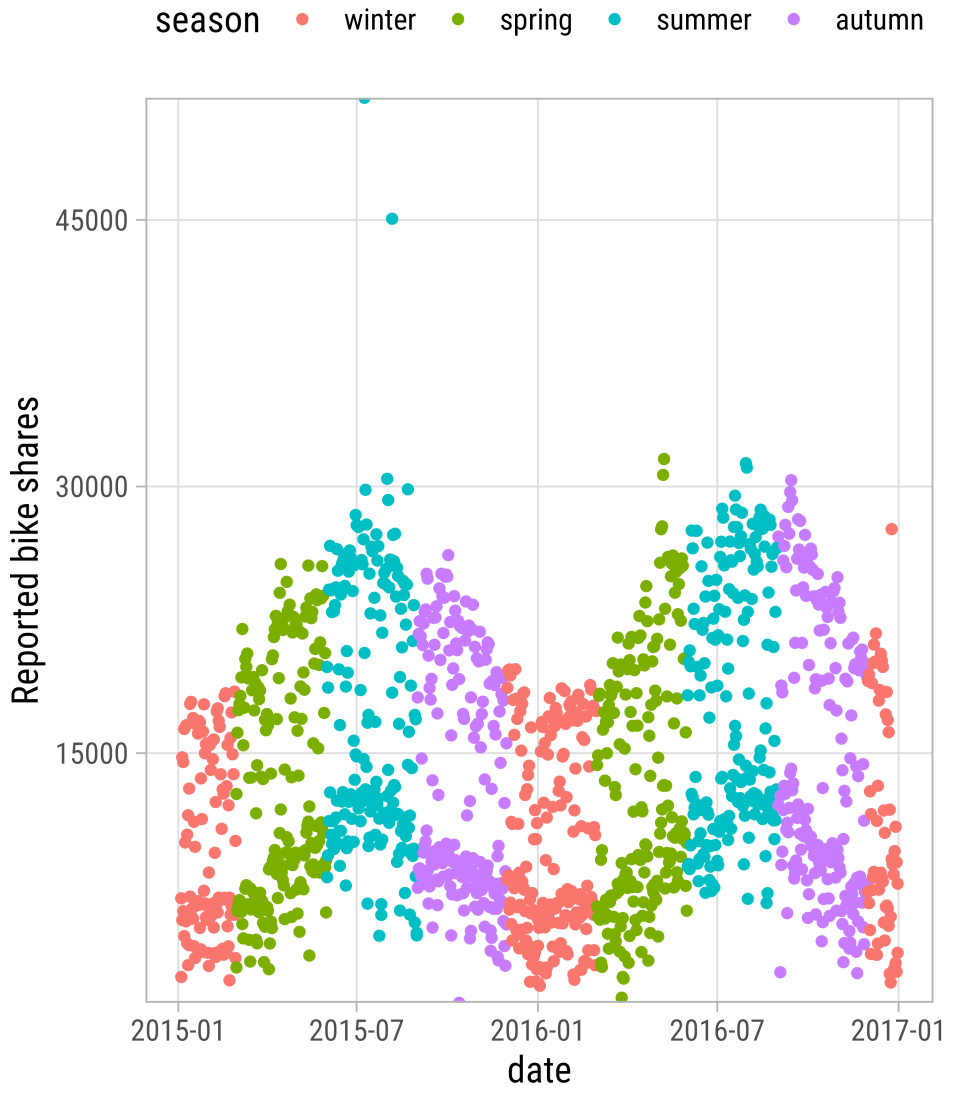
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = 0:4*15000,
    limits = c(NA, 60000)
  ) +
  scale_color_discrete()
```



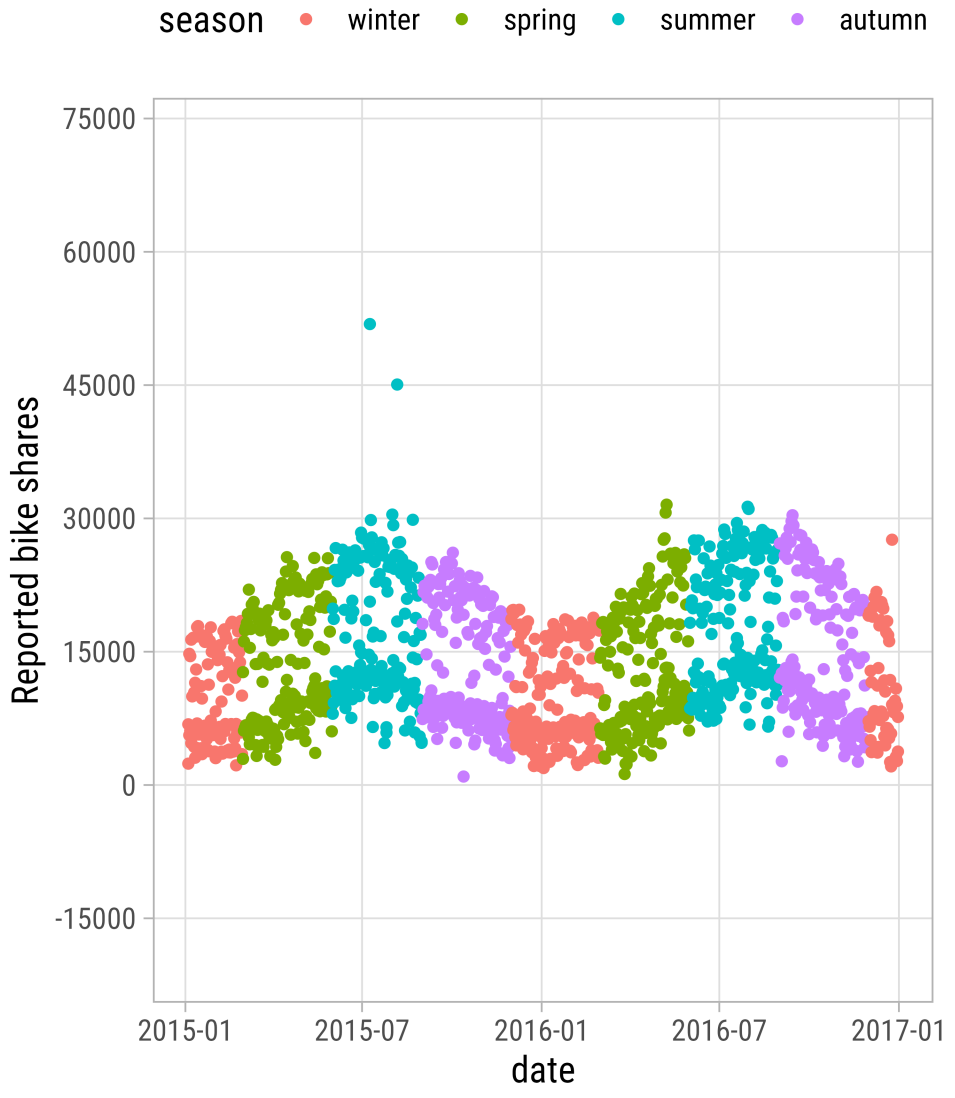
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = 0:4*15000,
    expand = c(0, 0)
  ) +
  scale_color_discrete()
```



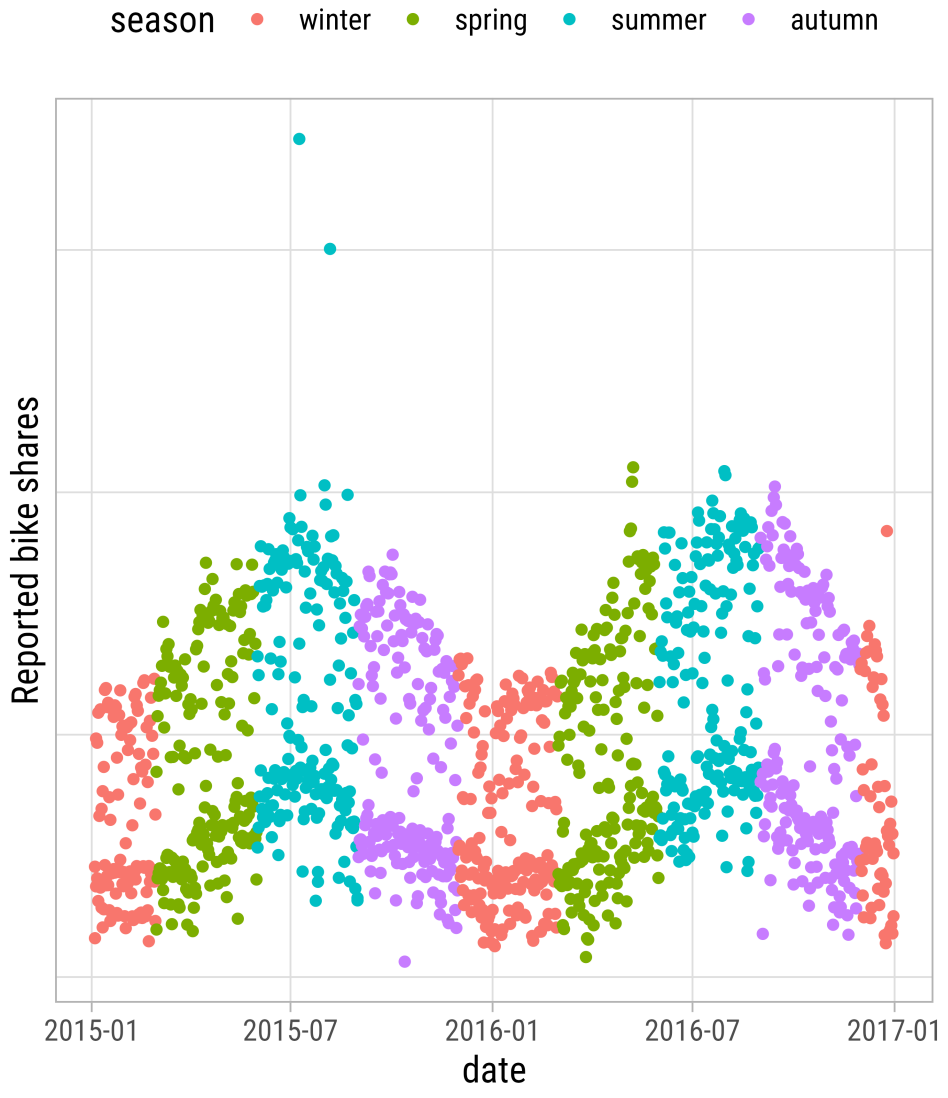
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = -1:5*15000,
    expand = c(.5, .5)
  ) +
  scale_color_discrete()
```



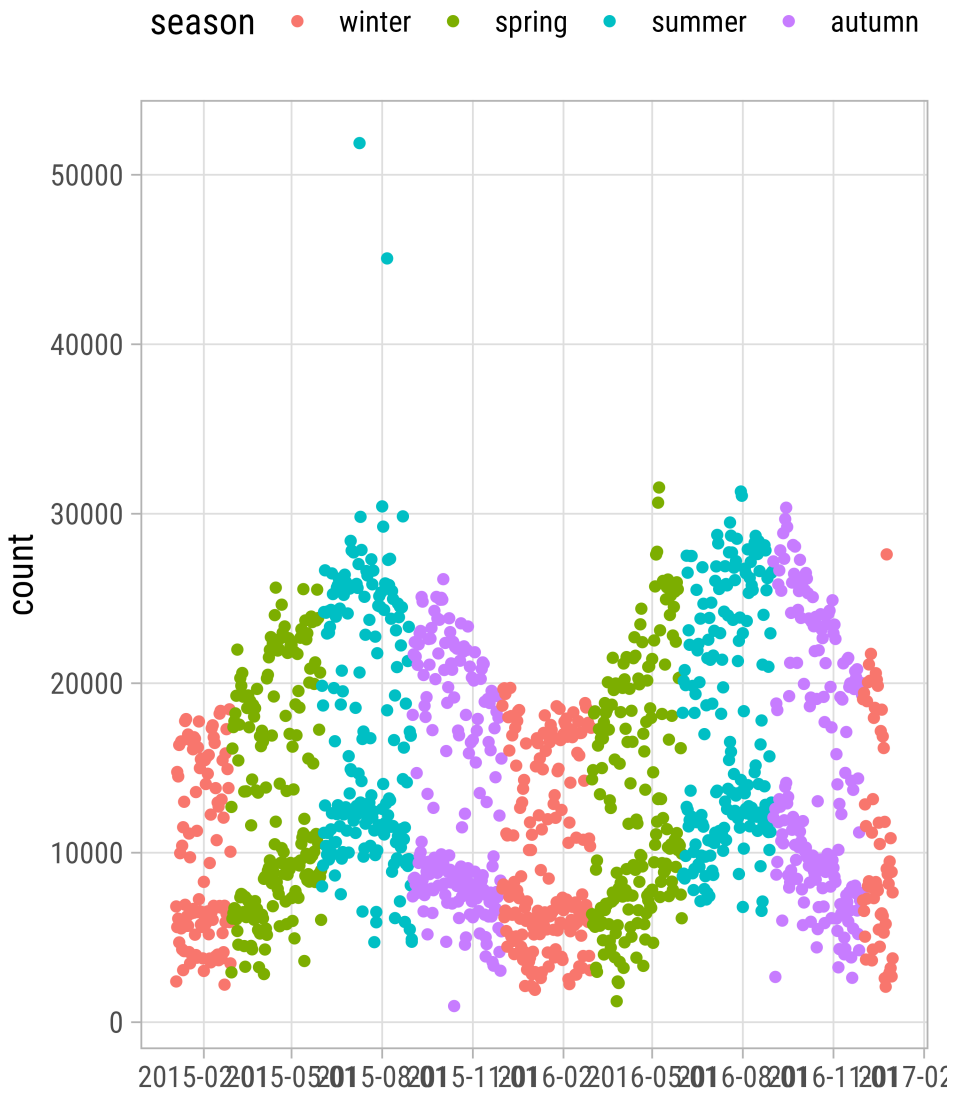
`scale_x|y_continuous`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_y_continuous(
    name = "Reported bike shares",
    breaks = 0:4*15000,
    guide = "none"
  )
```



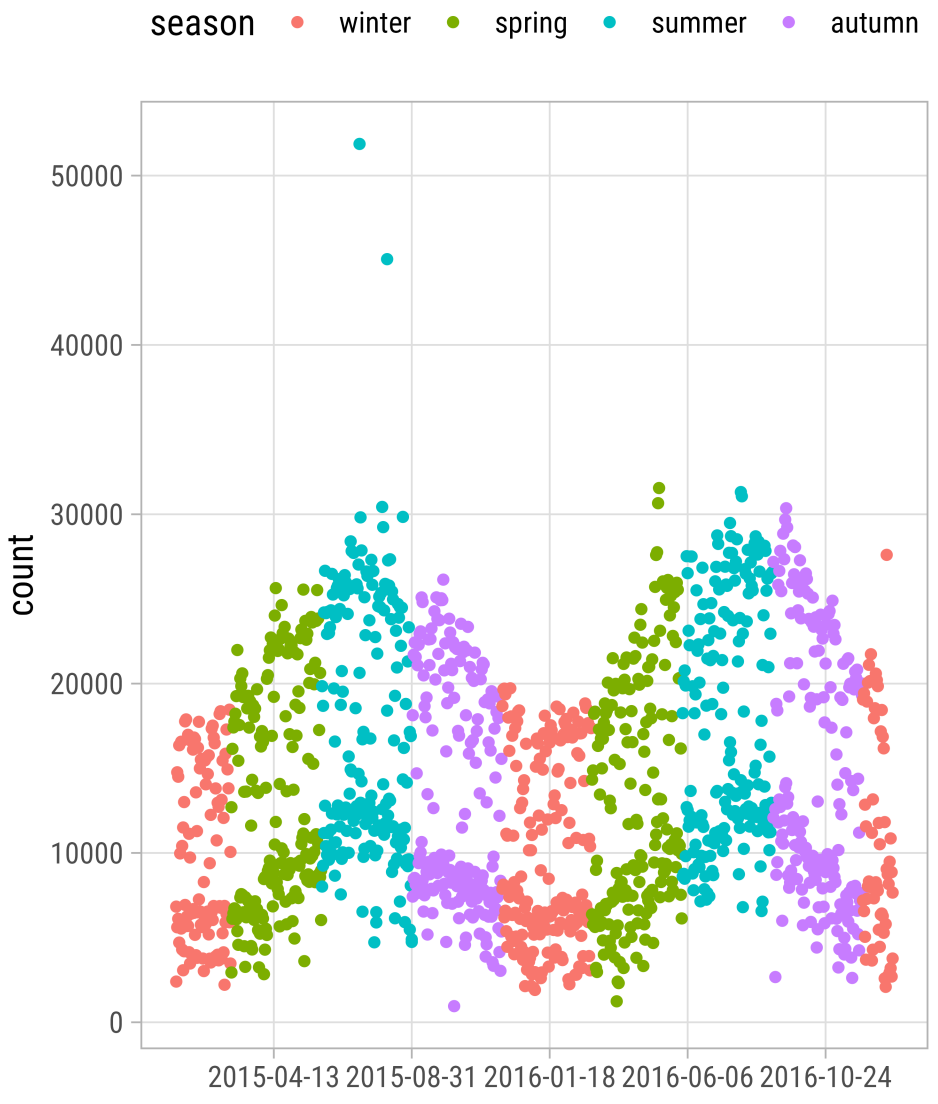
`scale_x|y_date`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date(
    name = NULL,
    date_breaks = "3 months"
  )
```



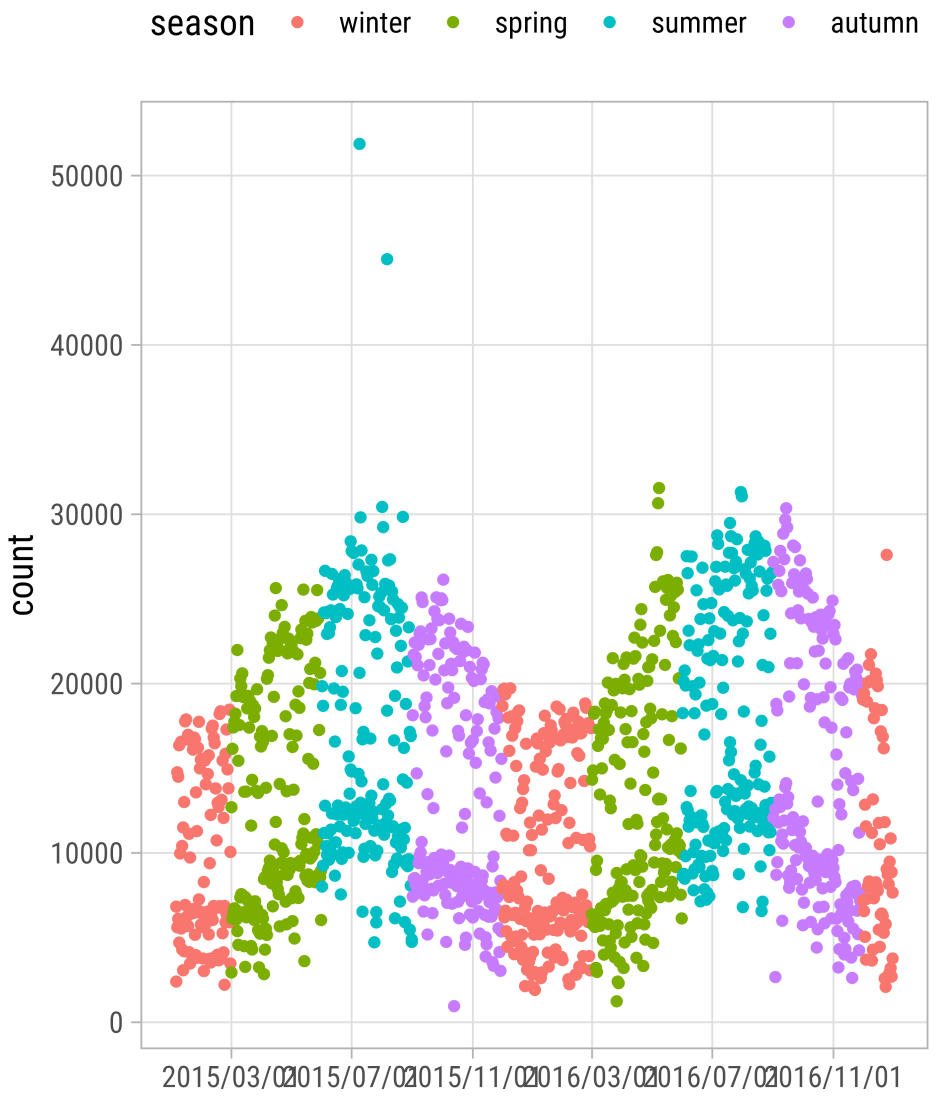
`scale_x|y_date`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date(
    name = NULL,
    date_breaks = "20 weeks"
  )
```



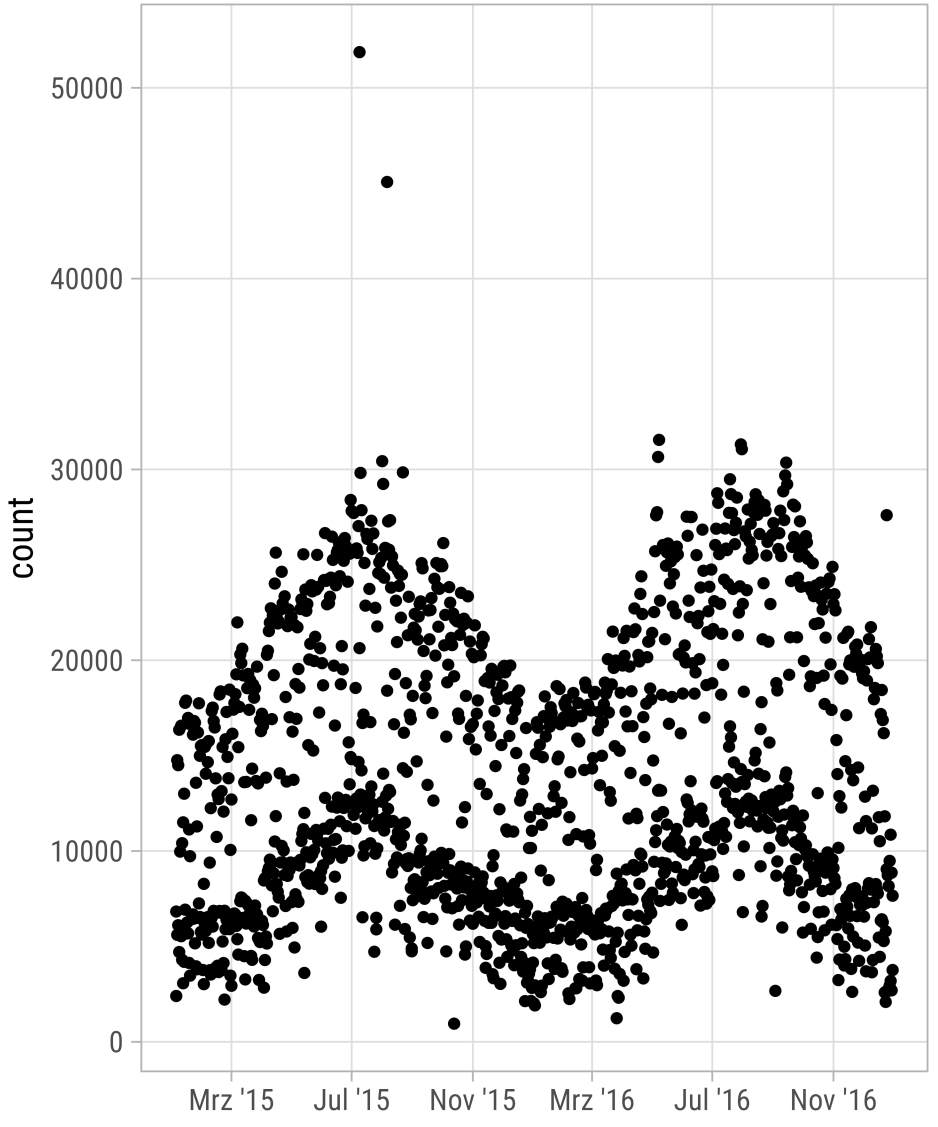
`scale_x|y_date` with `strftime()`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_x_date(
    name = NULL,
    date_breaks = "4 months",
    date_labels = "%Y/%m/%d"
  )
```



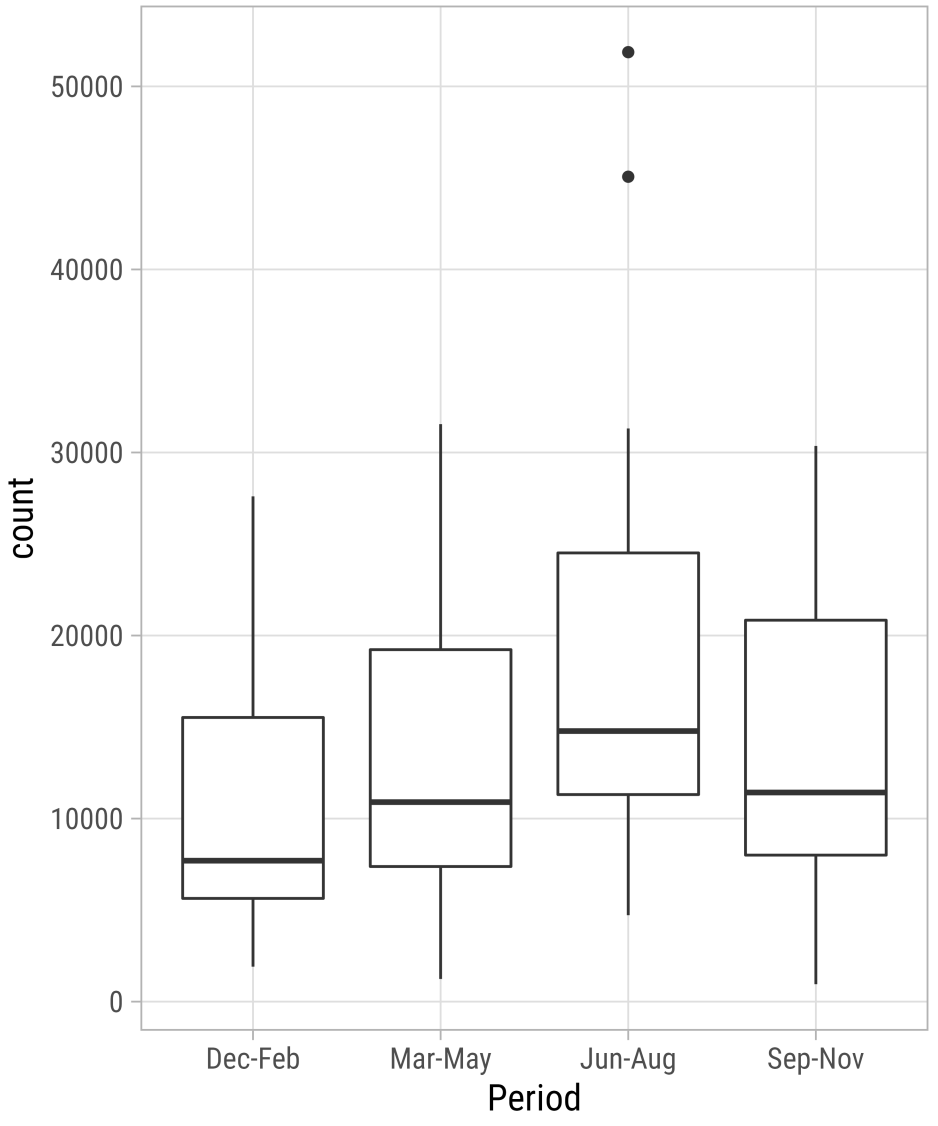
`scale_x|y_date` with `strftime()`

```
ggplot(
    bikes,
    aes(x = date, y = count)
  ) +
  geom_point() +
  scale_x_date(
    name = NULL,
    date_breaks = "4 months",
    date_labels = "%b '%y"
  )
```



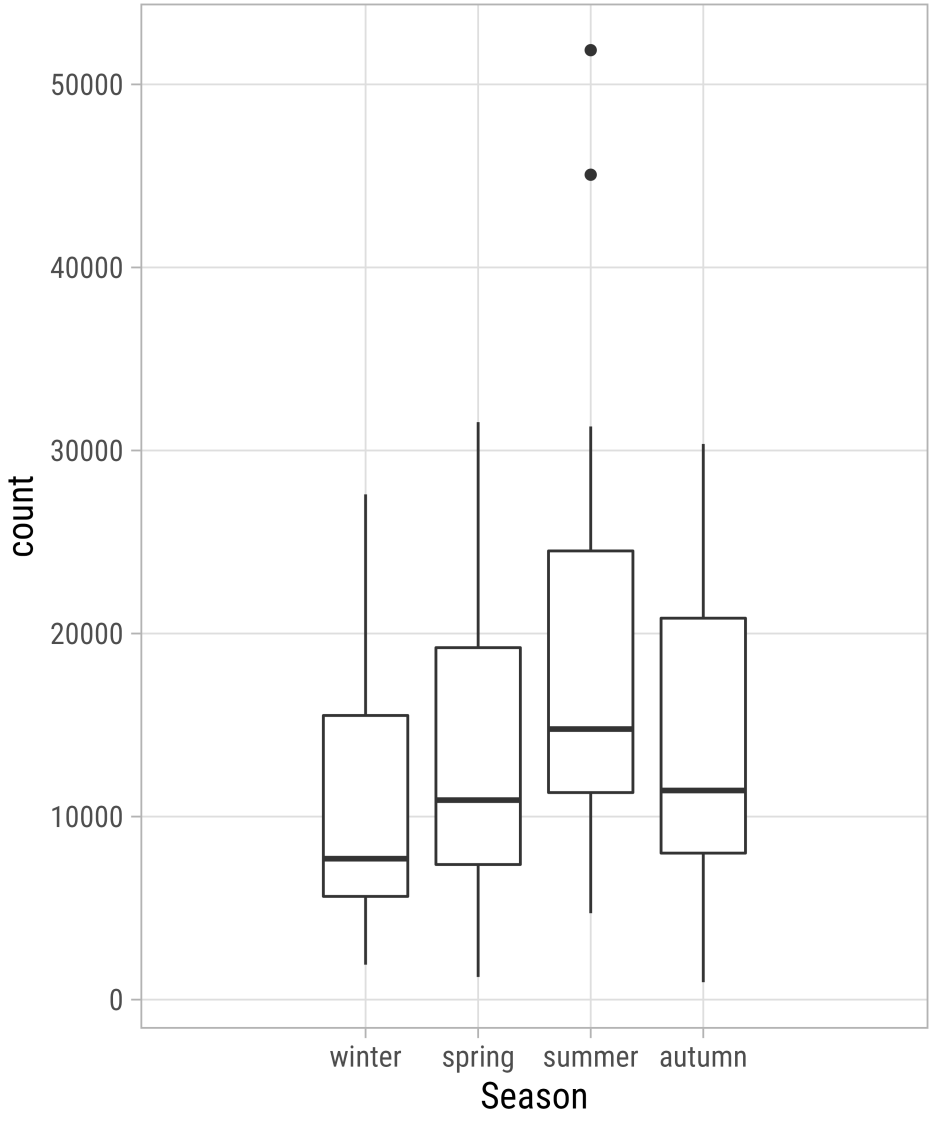
`scale_x|y_discrete`

```
1  ggplot(
2      bikes,
3      aes(x = season, y = count)
4    ) +
5    geom_boxplot() +
6    scale_x_discrete(
7      name = "Period",
8      labels = c("Dec-Feb", "Mar-May", "Jun-Aug"
9    )
```



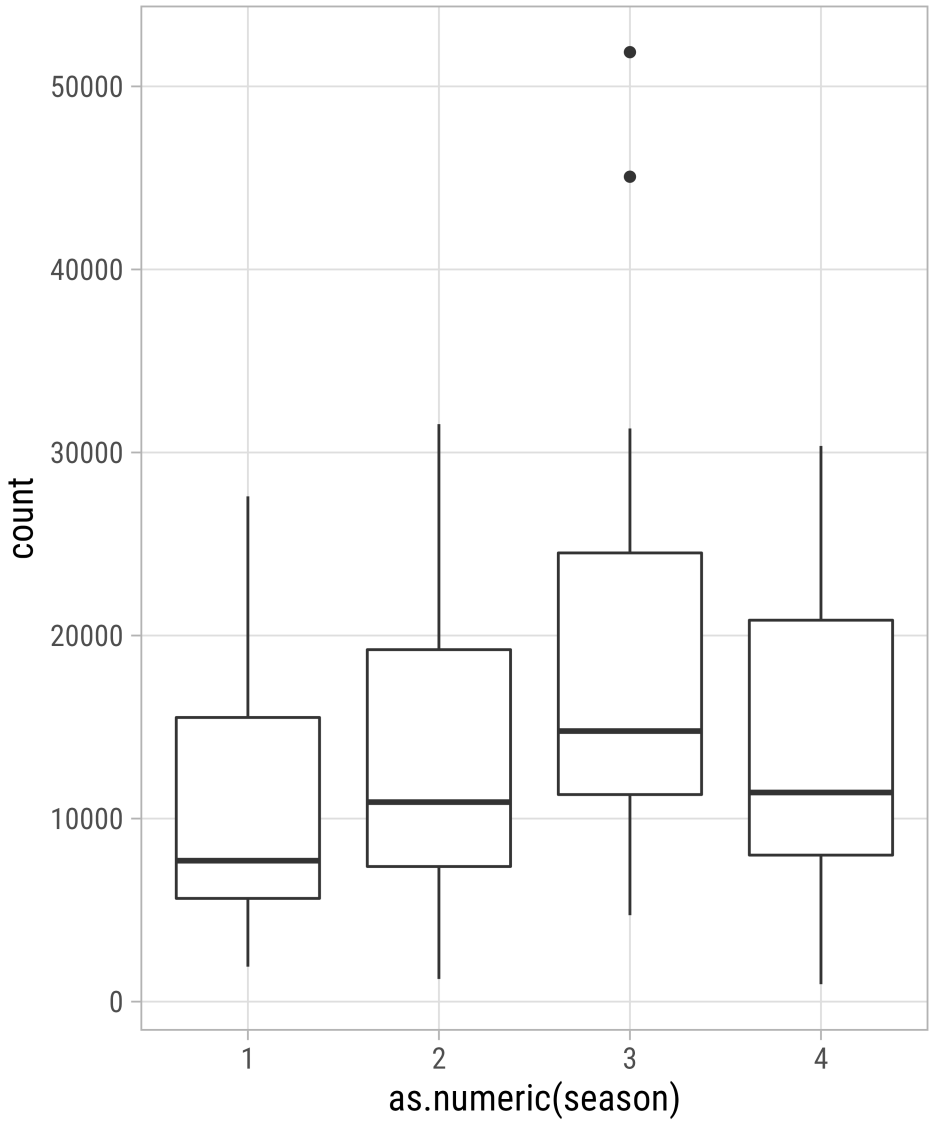
`scale_x|y_discrete`

```
1  ggplot(
2      bikes,
3      aes(x = season, y = count)
4    ) +
5    geom_boxplot() +
6    scale_x_discrete(
7      name = "Season",
8      expand = c(.5, .5)
9    )
```



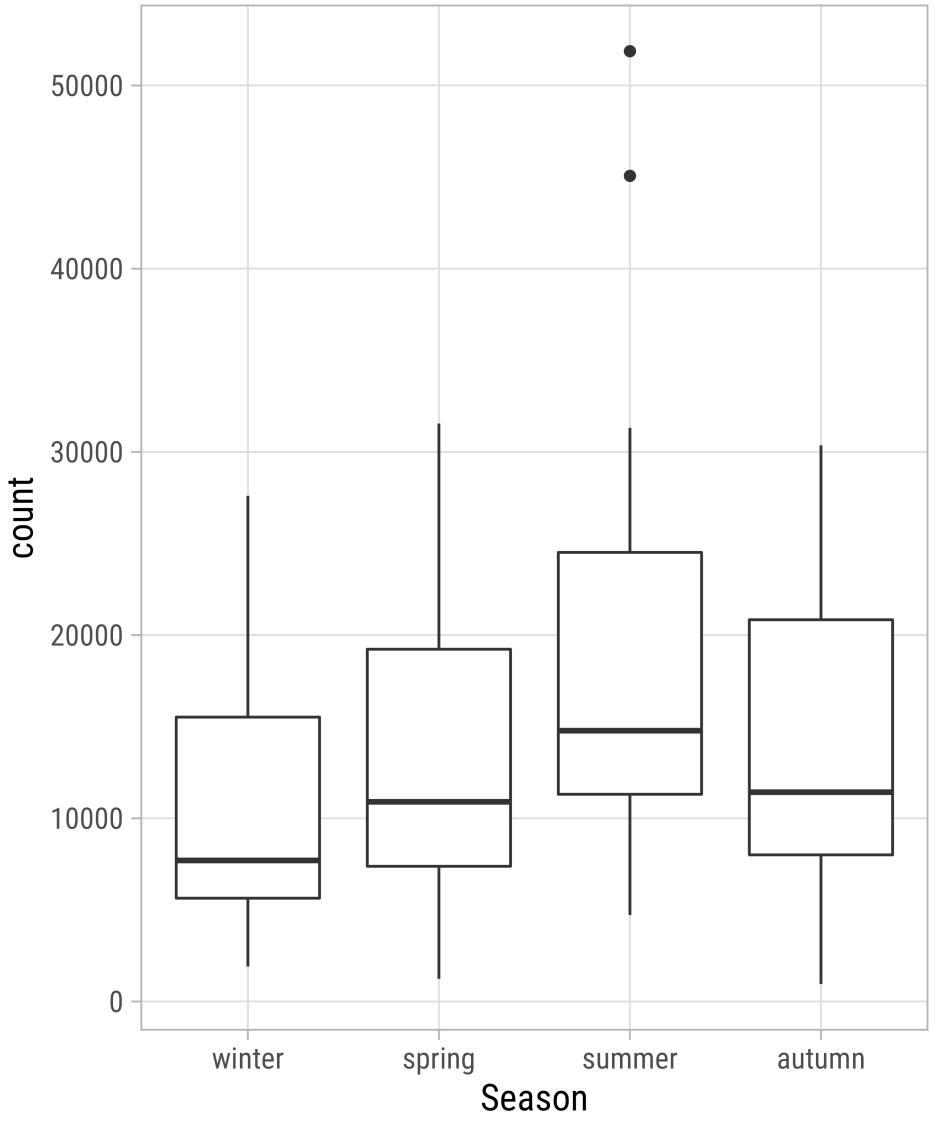
Discrete or Continuous?

```
1  ggplot(
2      bikes,
3      aes(x = as.numeric(season), y = count)
4    ) +
5    geom_boxplot(
6      aes(group = season)
7    )
```



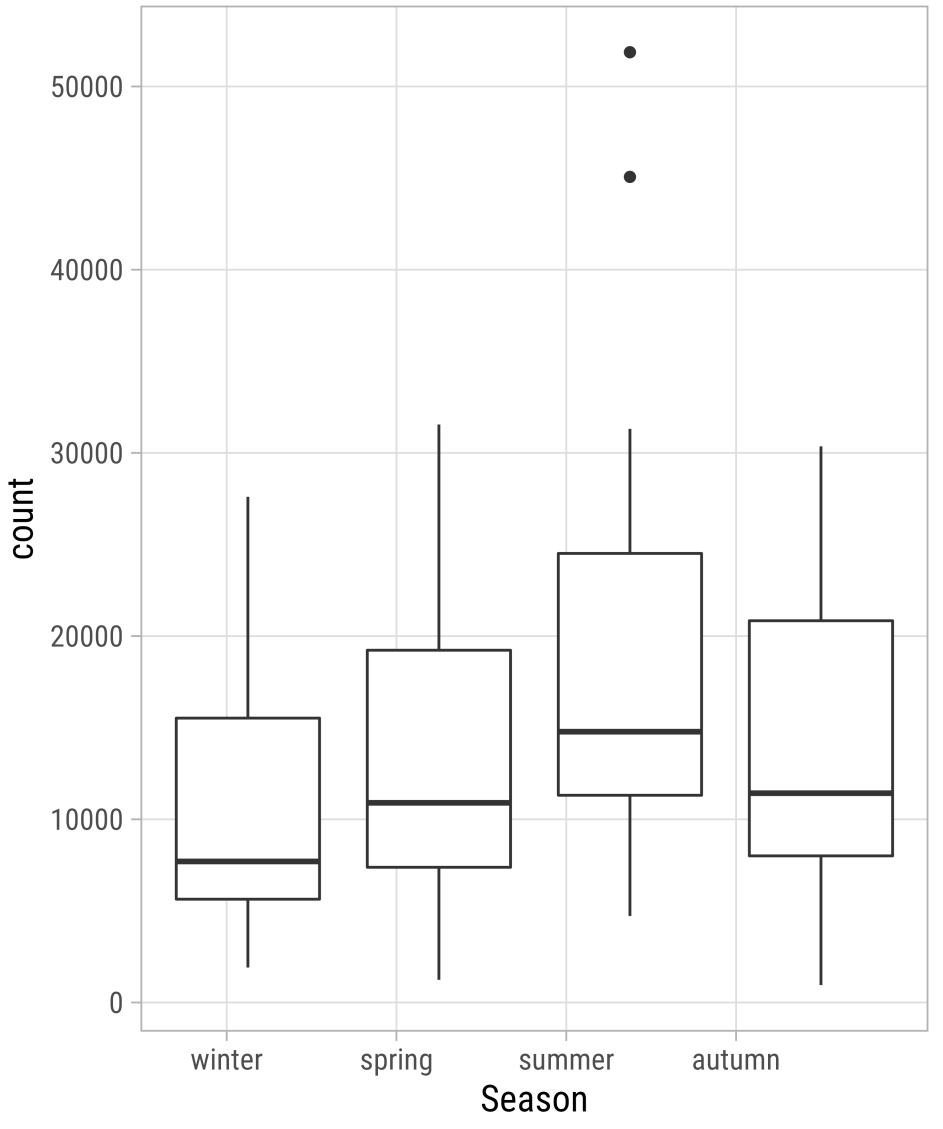
Discrete or Continuous?

```
ggplot(
    bikes,
    aes(x = as.numeric(season),
        y = count)
  ) +
  geom_boxplot(
    aes(group = season)
  ) +
  scale_x_continuous(
    name = "Season",
    breaks = 1:4,
    labels = levels(bikes$season)
  )
```



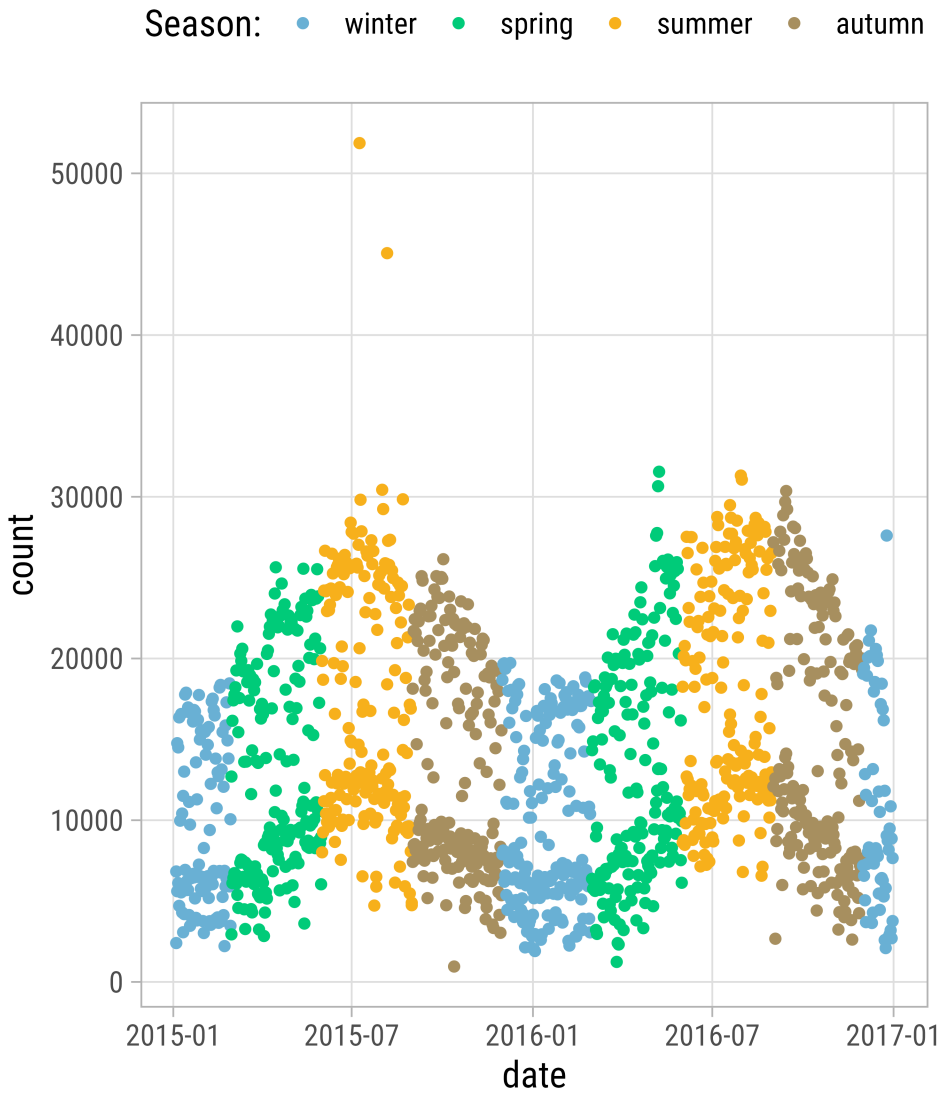
Discrete or Continuous?

```
ggplot(
    bikes,
    aes(x = as.numeric(season) + as.numeric(se
        y = count)
  ) +
  geom_boxplot(
    aes(group = season)
  ) +
  scale_x_continuous(
    name = "Season",
    breaks = 1:4,
    labels = levels(bikes$season)
  )
```



`scale_color|fill_discrete`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_color_discrete(
    name = "Season:",
    type = c("#69b0d4", "#00CB79", "#F7B01B",
  )
```



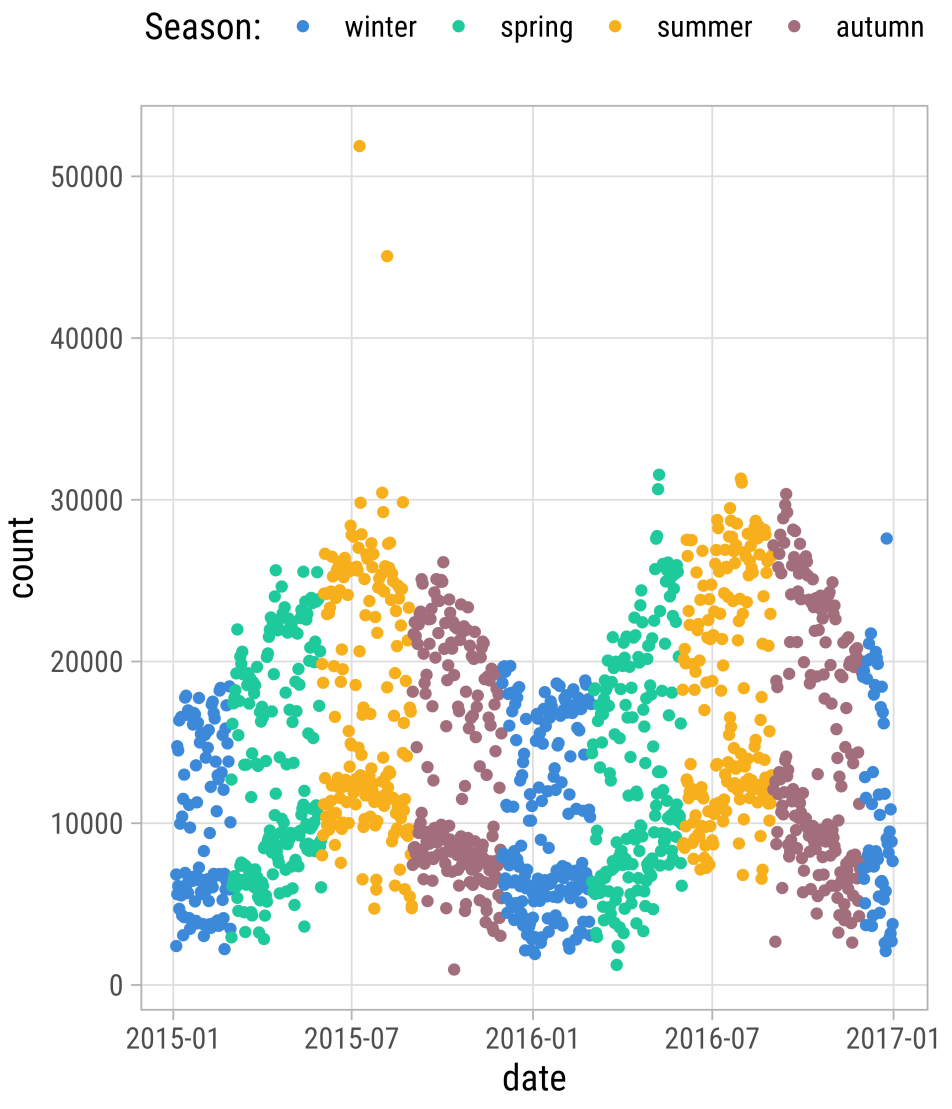
Inspect Assigned Colors

```
 1  g <- ggplot(
 2      bikes,
 3      aes(x = date, y = count,
 4          color = season)
 5    ) +
 6    geom_point() +
 7    scale_color_discrete(
 8      name = "Season:",
 9      type = c("#3ca7d9", "#1ec99b", "#F7B01B"
10    )
11
12  gb <- ggplot_build(g)
13
14  gb$data[[1]][1:10,]
```

```
    colour     x     y PANEL group
shape size fill alpha stroke
1   #3ca7d9 16439  6830     1     1
19   1.5   NA    NA    0.5
2   #3ca7d9 16439  2404     1     1
19   1.5   NA    NA    0.5
3   #3ca7d9 16440 14763     1     1
19   1.5   NA    NA    0.5
4   #3ca7d9 16440  5609     1     1
19   1.5   NA    NA    0.5
5   #3ca7d9 16441 14501     1     1
19   1.5   NA    NA    0.5
6   #3ca7d9 16441  6112     1     1
19   1.5   NA    NA    0.5
7   #3ca7d9 16442 16358     1     1
```

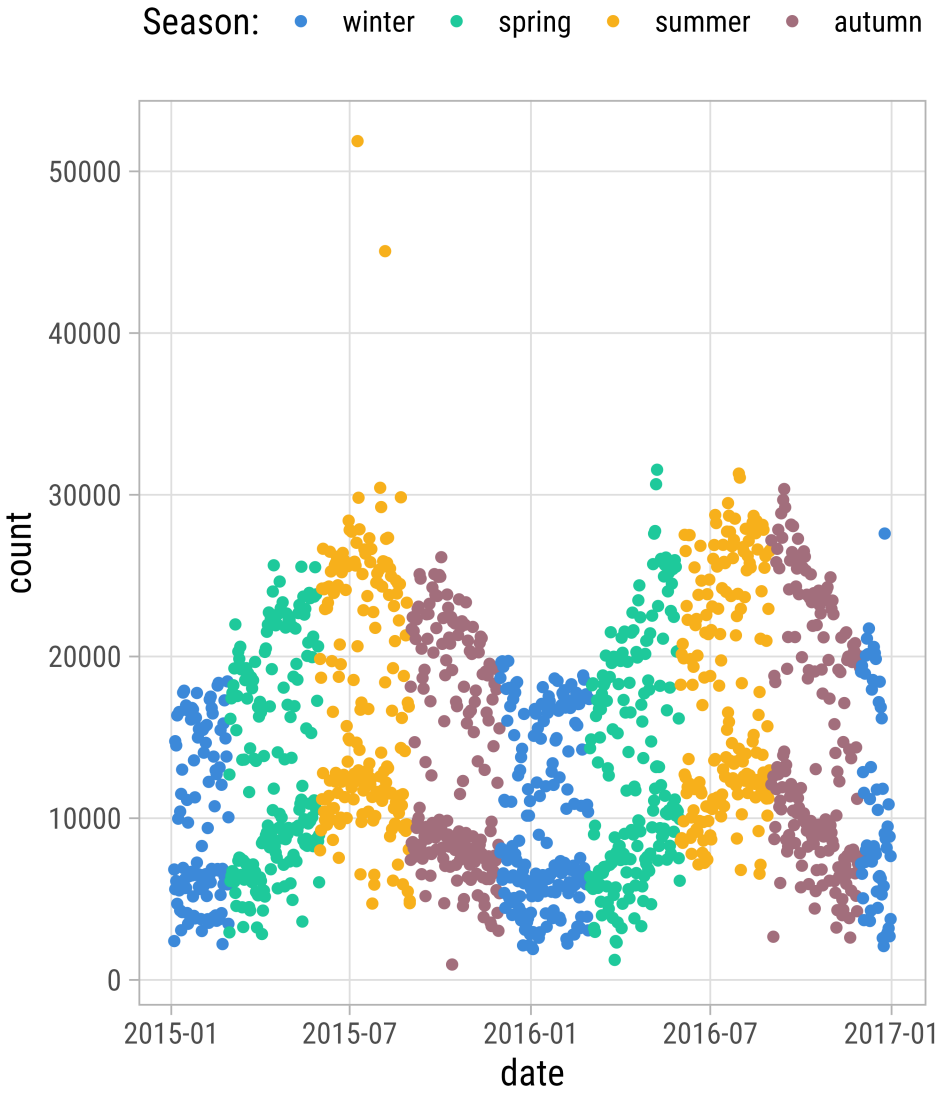
`scale_color|fill_discrete`

```
my_colors <- c(
  'winter' = "#3c89d9",
  'spring' = "#1ec99b",
  'summer' = "#F7B01B",
  'autumn' = "#a26e7c"
)

ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_color_discrete(
    name = "Season:",
    type = my_colors
  )
```



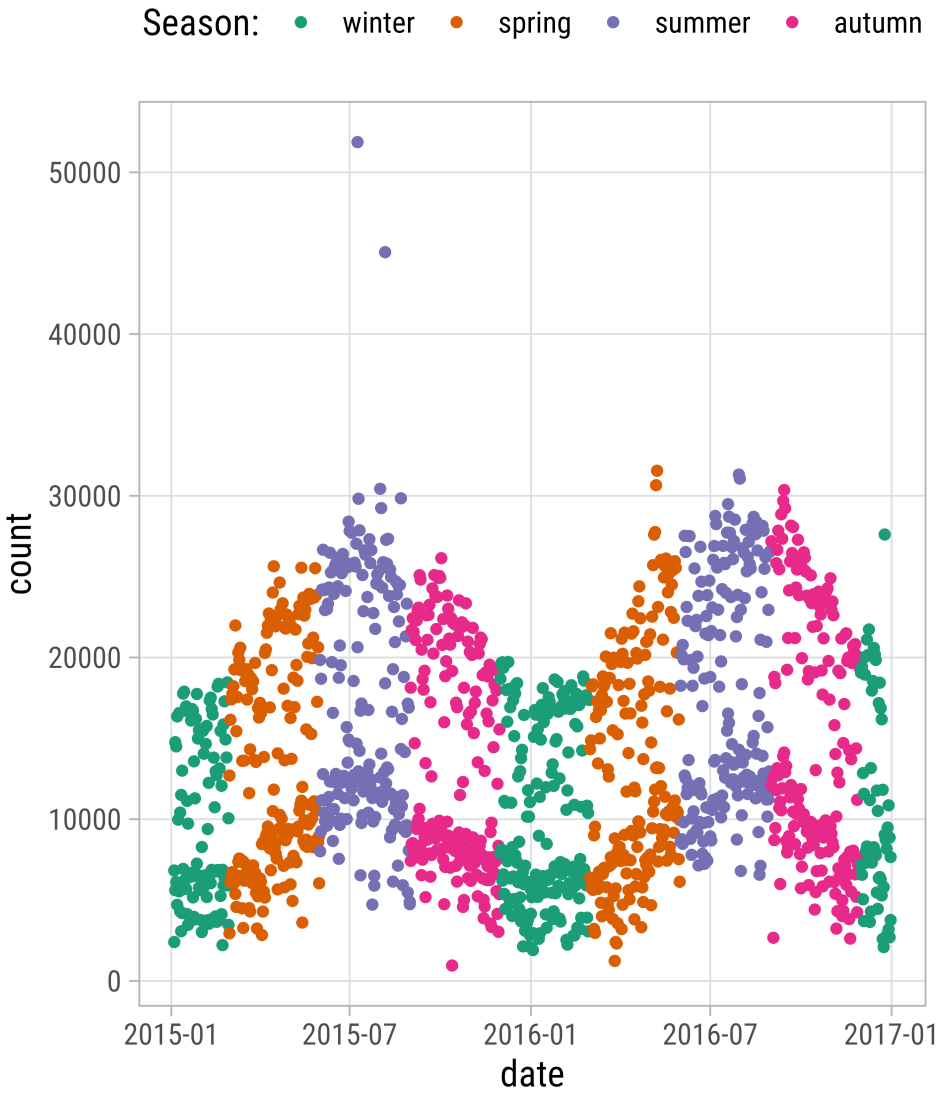
`scale_color|fill_discrete`

```
my_colors_alphabetical <- c(
  'autumn' = "#a26e7c",
  'spring' = "#1ec99b",
  'summer' = "#F7B01B",
  'winter' = "#3c89d9"
)

ggplot(
    bikes,
    aes(x = date, y = count,
        color = season)
  ) +
  geom_point() +
  scale_color_discrete(
    name = "Season:",
    type = my_colors_alphabetical
  )
```



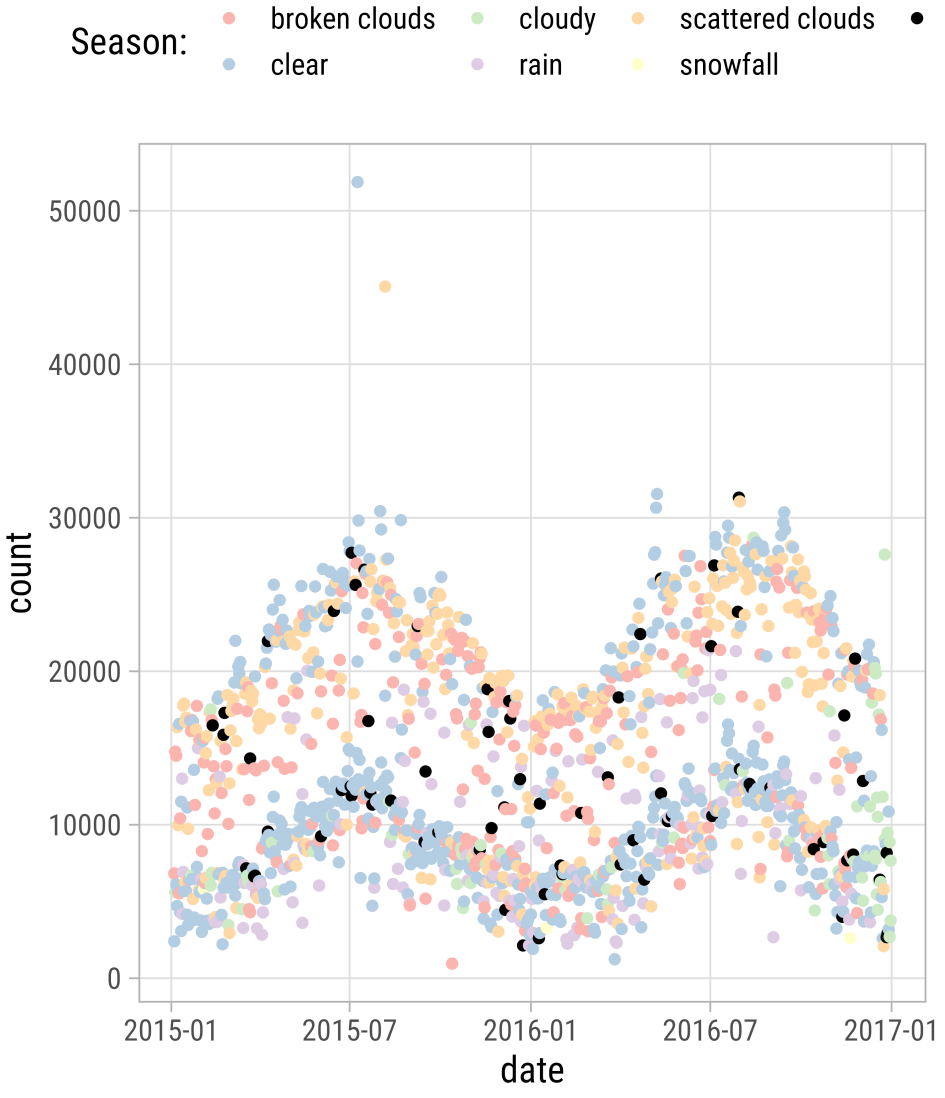
`scale_color|fill_discrete`

```
 1  library(RColorBrewer)
 2
 3  ggplot(
 4      bikes,
 5      aes(x = date, y = count,
 6          color = season)
 7    ) +
 8    geom_point() +
 9    scale_color_discrete(
10      name = "Season:",
11      type = brewer.pal(
12        n = 4, name = "Dark2"
13      )
14    )
```



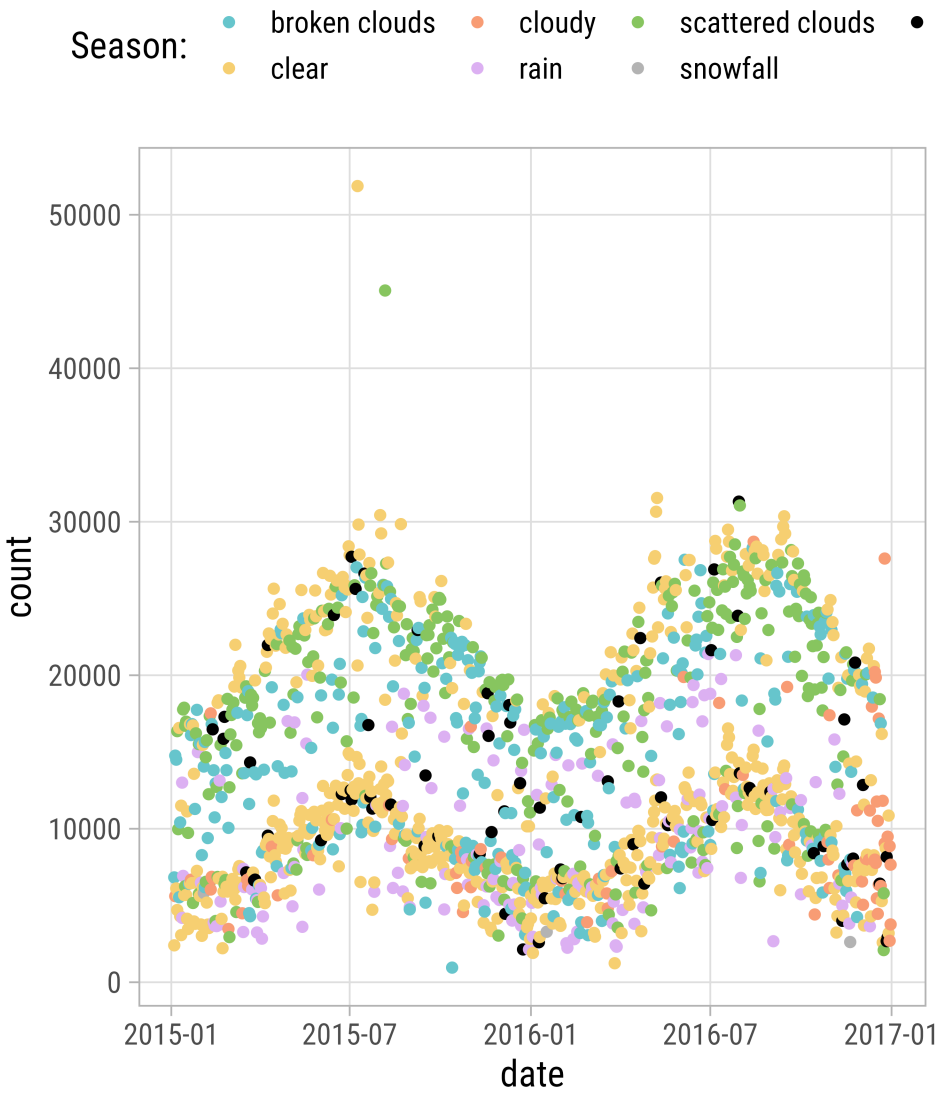
`scale_color|fill_manual`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = weather_type)
  ) +
  geom_point() +
  scale_color_manual(
    name = "Season:",
    values = brewer.pal(n = 6, name = "Pastel1
    na.value = "black"
  )
```



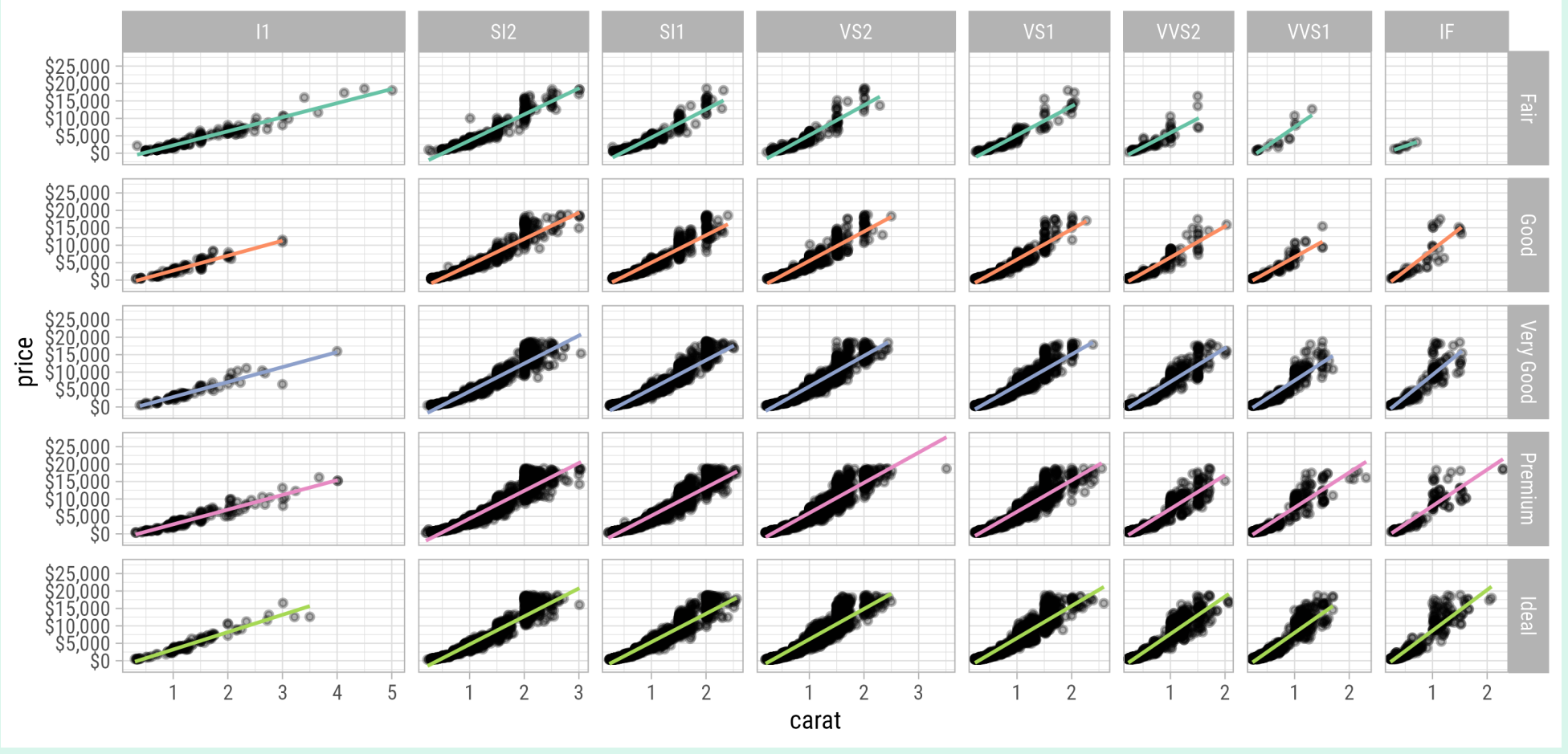
`scale_color|fill_carto_d`

```
ggplot(
    bikes,
    aes(x = date, y = count,
        color = weather_type)
  ) +
  geom_point() +
  rcartocolor::scale_color_carto_d(
    name = "Season:",
    palette = "Pastel",
    na.value = "black"
  )
```



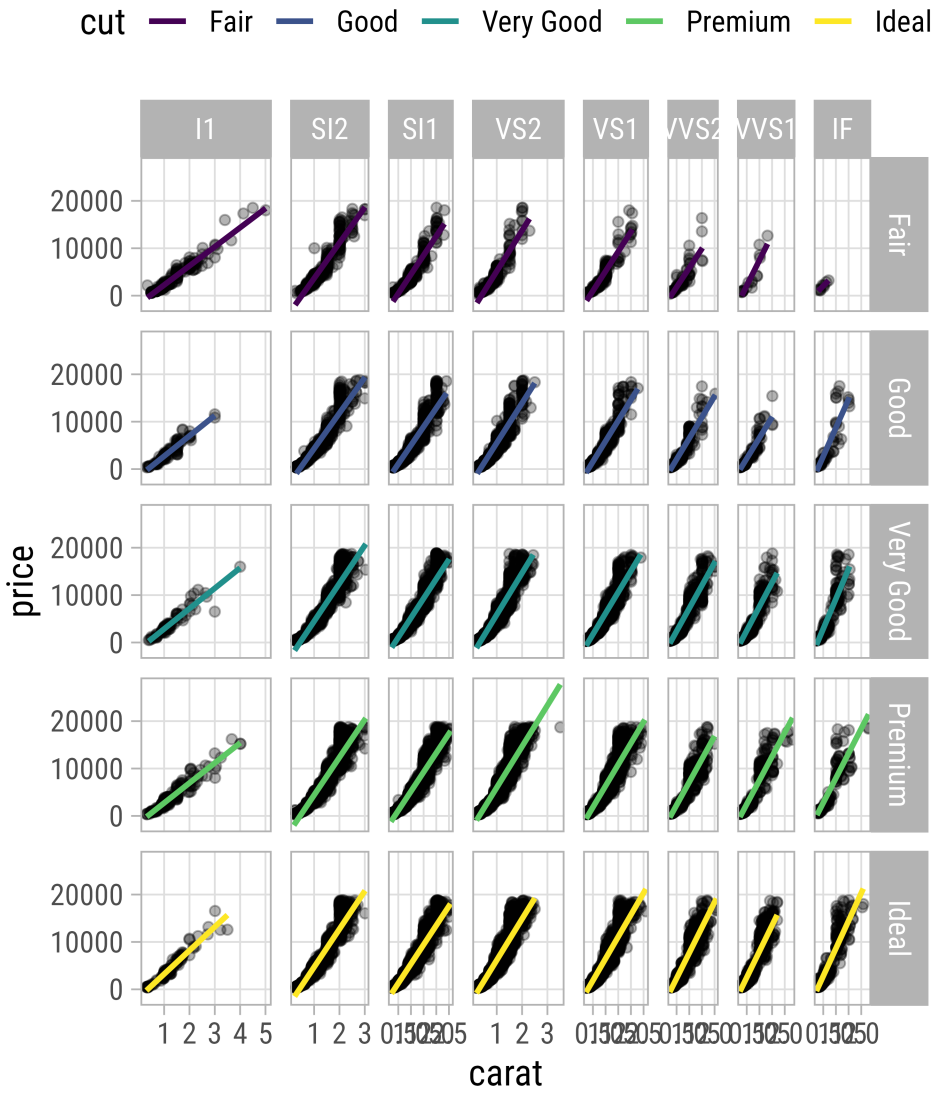
Your Turn!

**Modify our dianmonds facet like this:



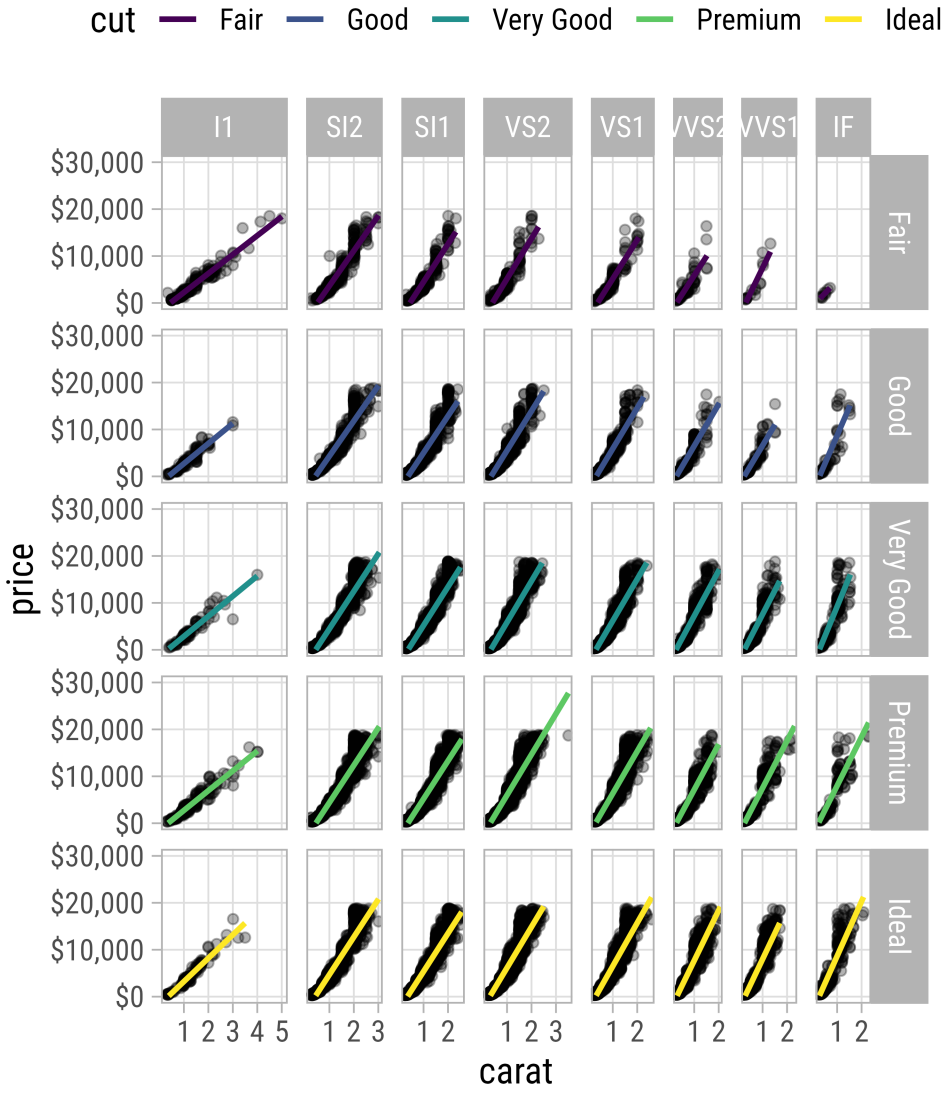
Diamonds Facet

```
facet <-
  ggplot(
    diamonds,
    aes(x = carat, y = price)
  ) +
  geom_point(
    alpha = .3
  ) +
  geom_smooth(
    aes(color = cut),
    method = "lm",
    se = FALSE
  ) +
  facet_grid(
    cut ~ clarity,
    space = "free_x",
    scales = "free_x"
  )

facet
```



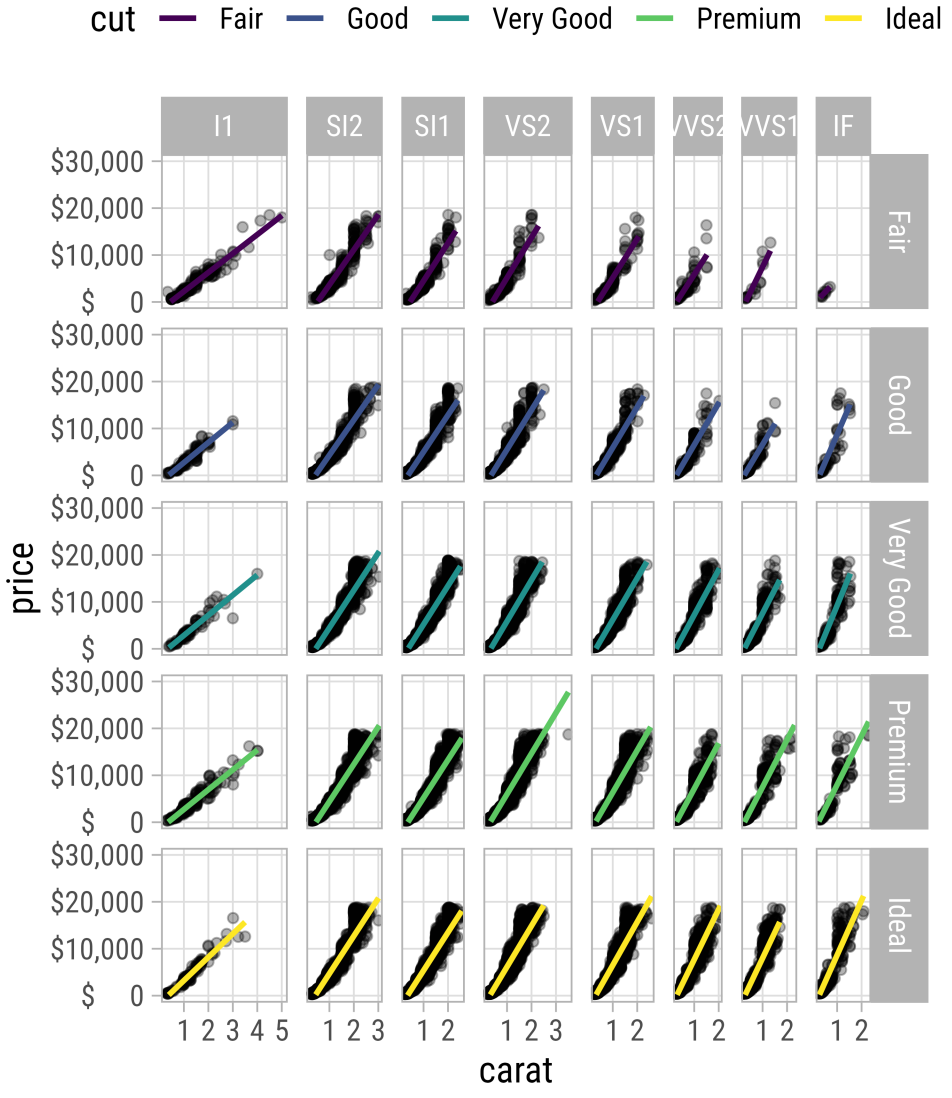
Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = c("$0", "$10,000", "$20,000", "$3
  )
```



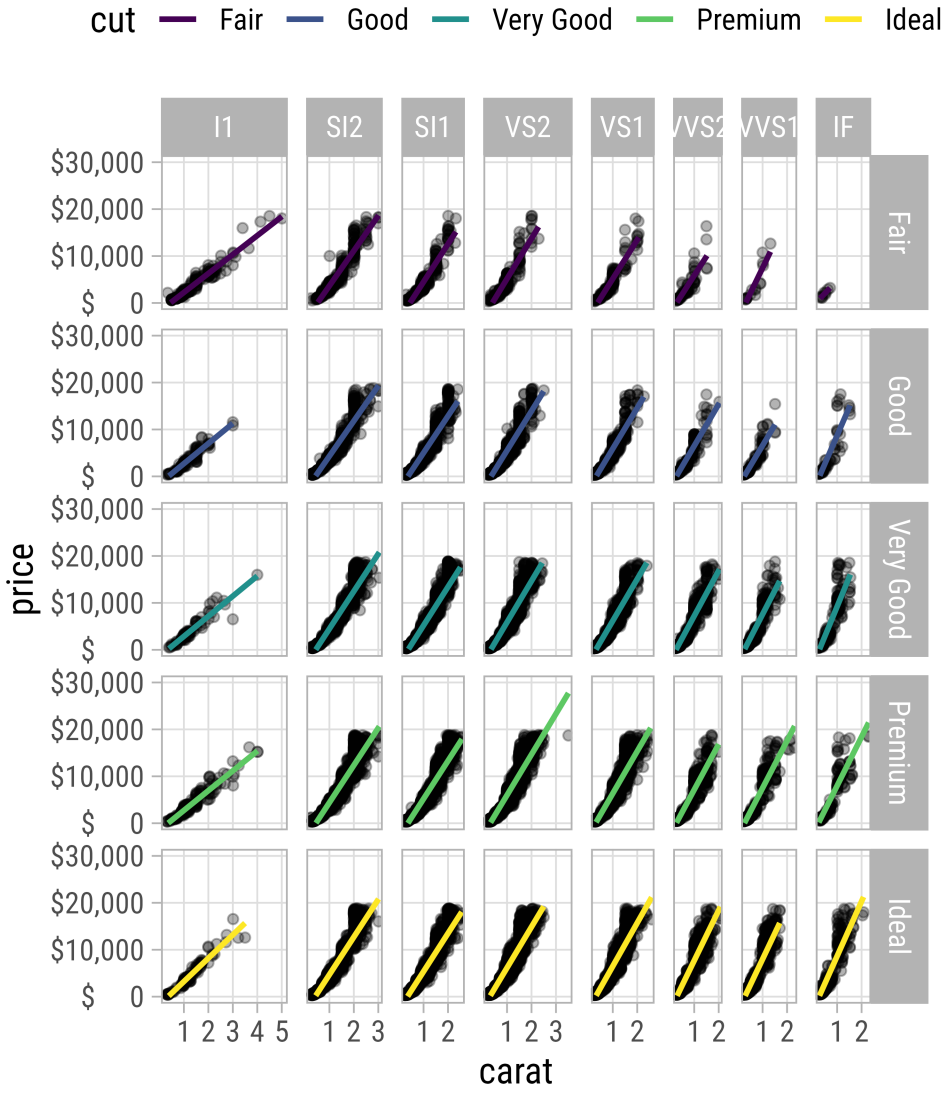
Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = paste0(
      "$", format(0:3*10000, big.mark = ",")
    )
  )
```



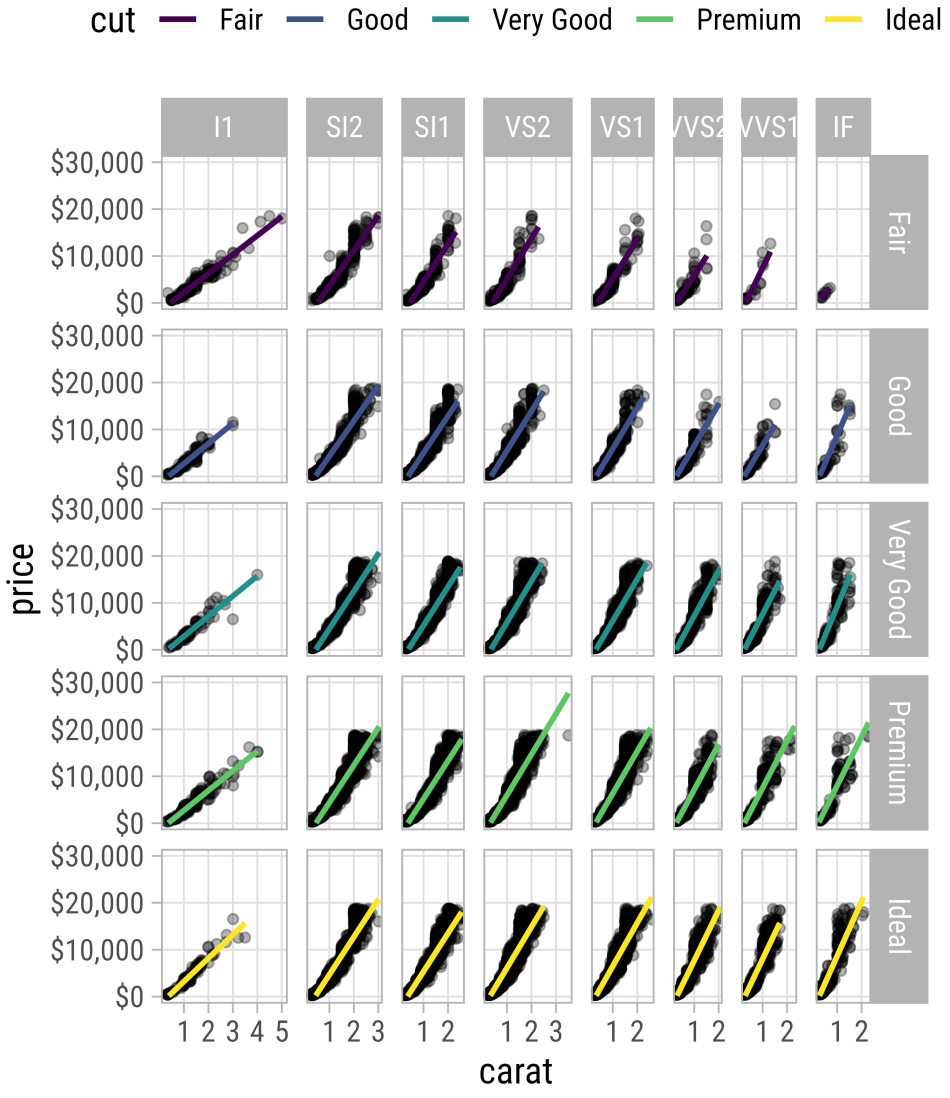
Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = function(y) paste0(
      "$", format(y, big.mark=",")
    )
  )
```



Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = scales::dollar_format()
  )
```



Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = scales::dollar_format()
  ) +
  scale_color_brewer(
    palette = "Set2",
    guide = "none"
  )
```

Diamonds Facet

```
facet +
  scale_x_continuous(
    breaks = 0:5
  ) +
  scale_y_continuous(
    limits = c(0, 30000),
    breaks = 0:3*10000,
    labels = scales::dollar_format()
  ) +
  scale_color_brewer(
    palette = "Set2"
  ) +
  theme(
    legend.position = "none"
  )
```

Coordinate Systems

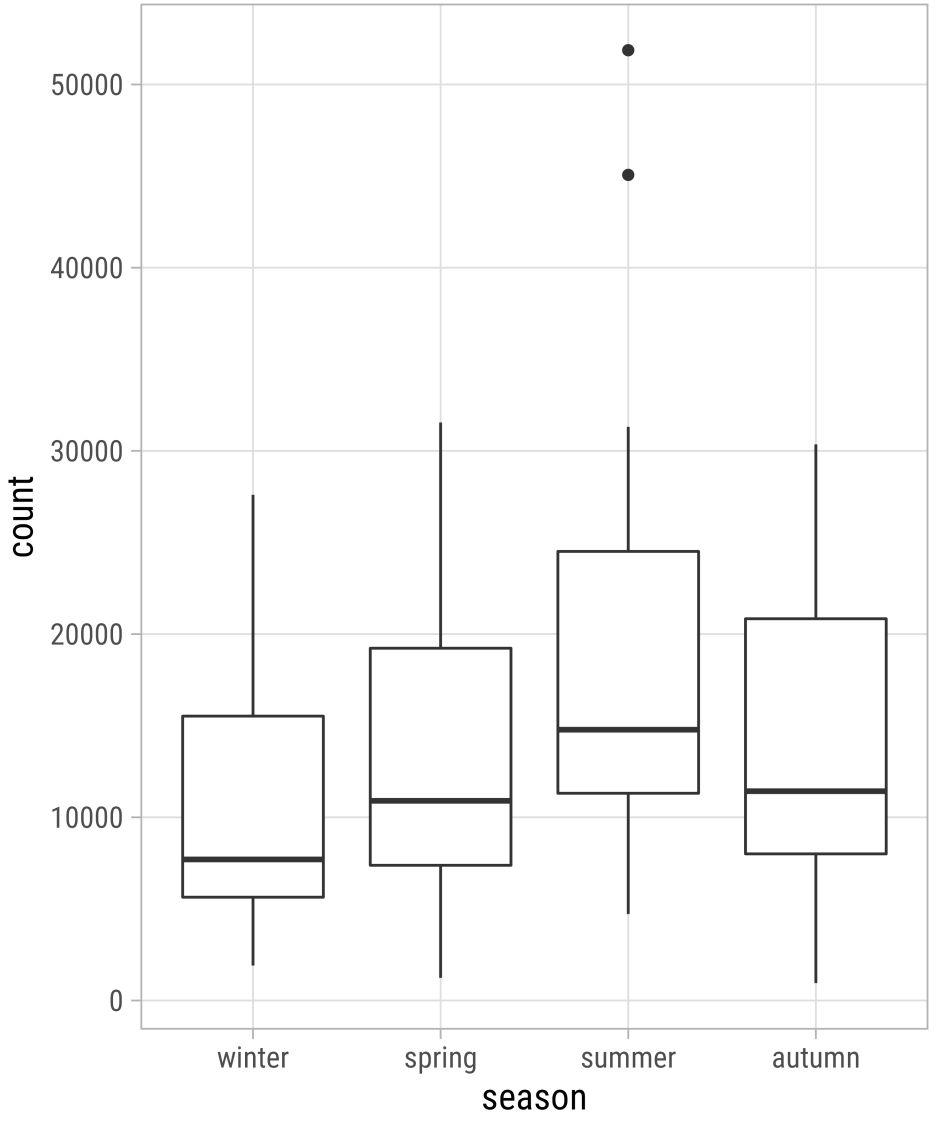
Cédric Scherer // rstudio::conf // July 2022

Coordinate Systems

= interpret the position aesthetics

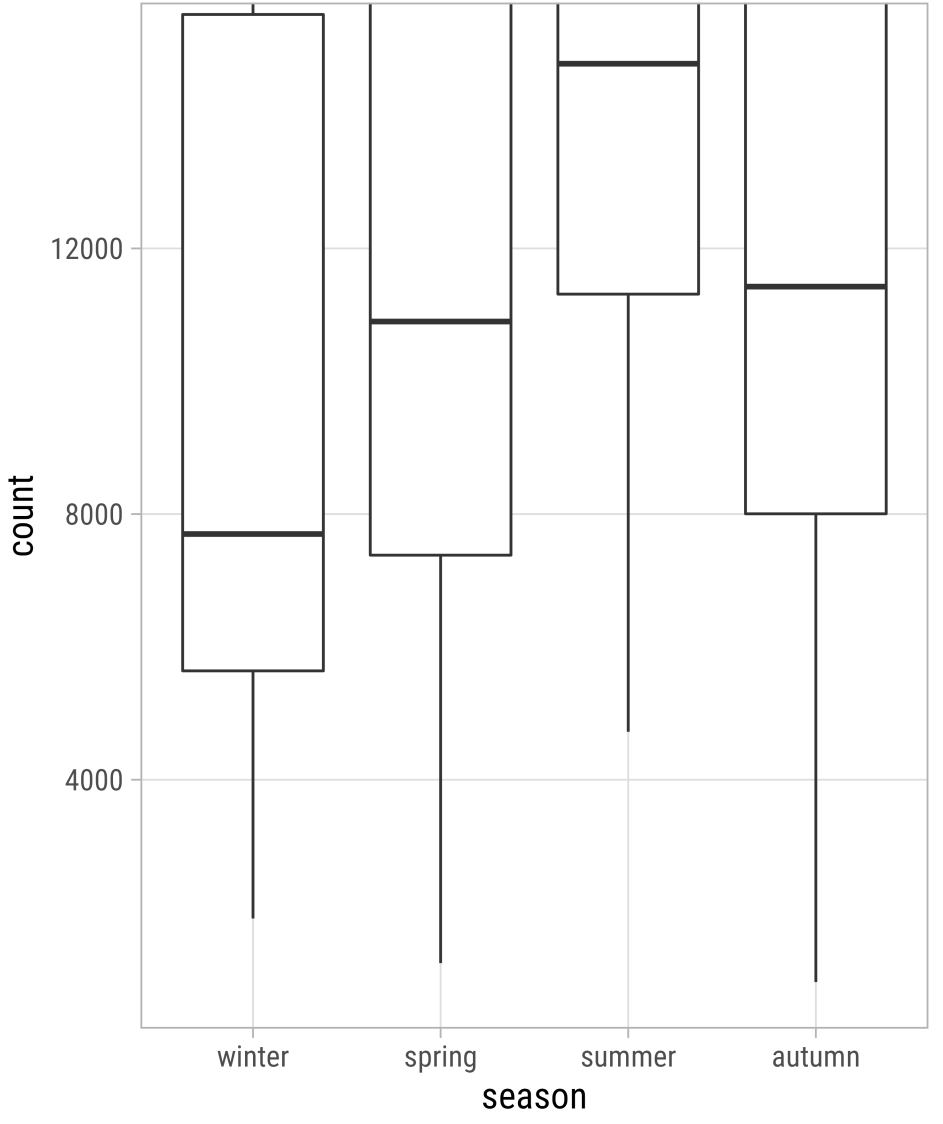
Cartesian Coordinate System

```
1 ggplot(
2     bikes,
3     aes(x = season, y = count)
4   ) +
5   geom_boxplot() +
6   coord_cartesian()
```



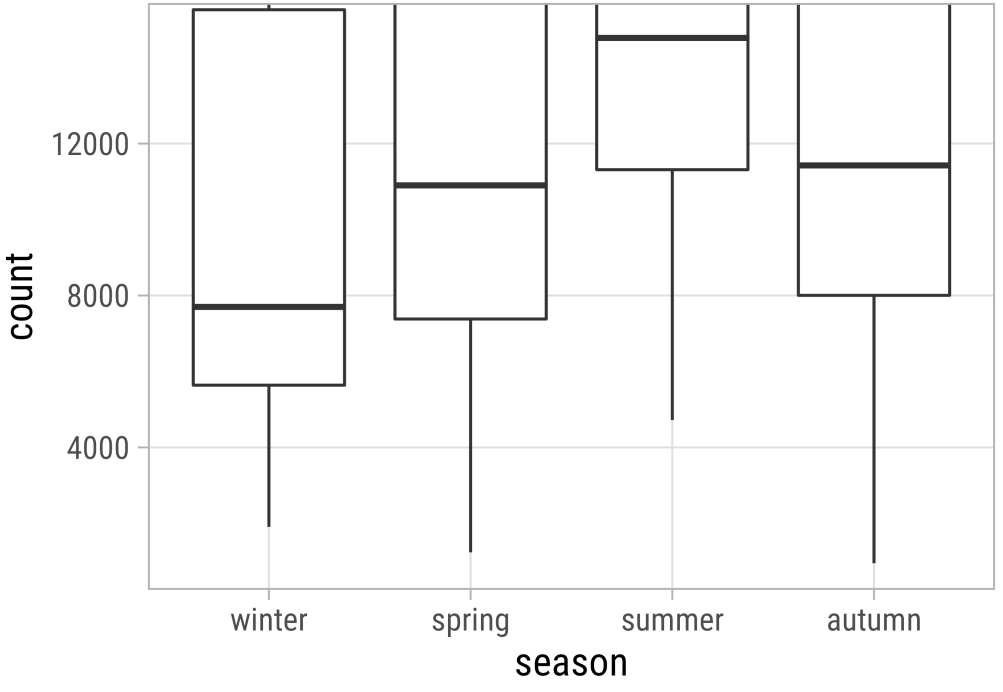
Cartesian Coordinate System

```
ggplot(
    bikes,
    aes(x = season, y = count)
  ) +
  geom_boxplot() +
  coord_cartesian(
    ylim = c(NA, 15000)
  )
```

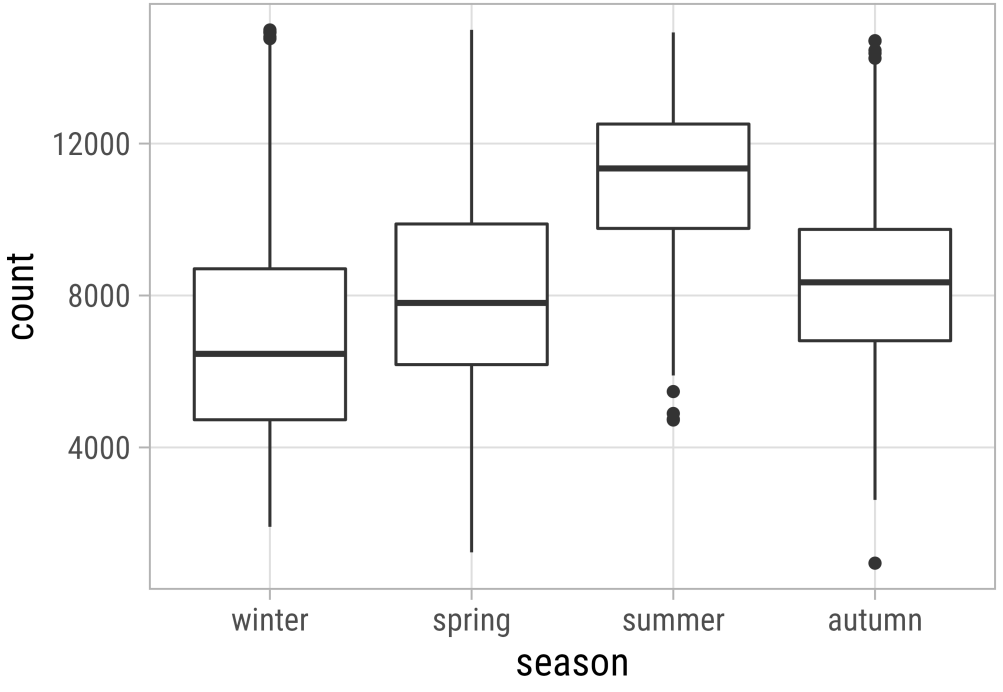


Changing Limits

```
ggplot(
    bikes,
    aes(x = season, y = count)
  ) +
  geom_boxplot() +
  coord_cartesian(
    ylim = c(NA, 15000)
  )
```

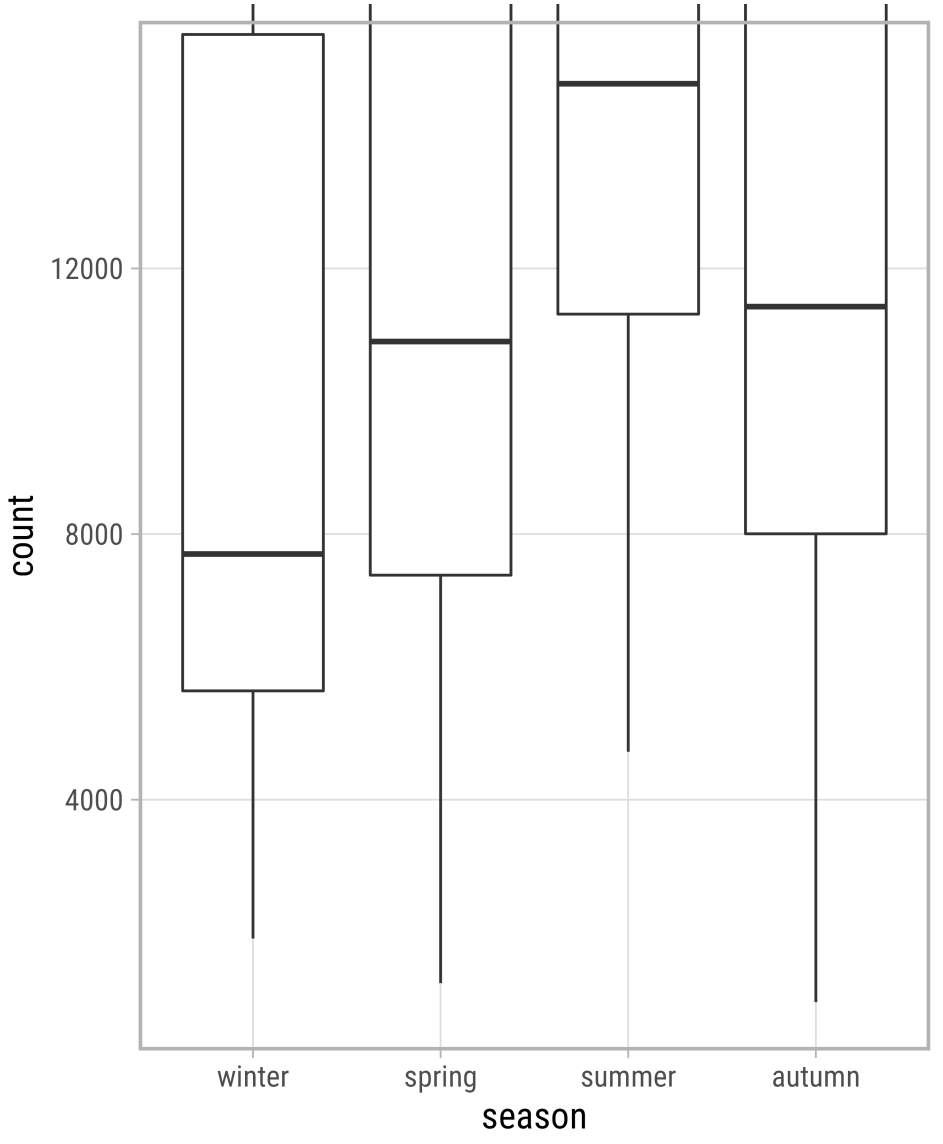



```
ggplot(
    bikes,
    aes(x = season, y = count)
  ) +
  geom_boxplot() +
  scale_y_continuous(
    limits = c(NA, 15000)
  )
```



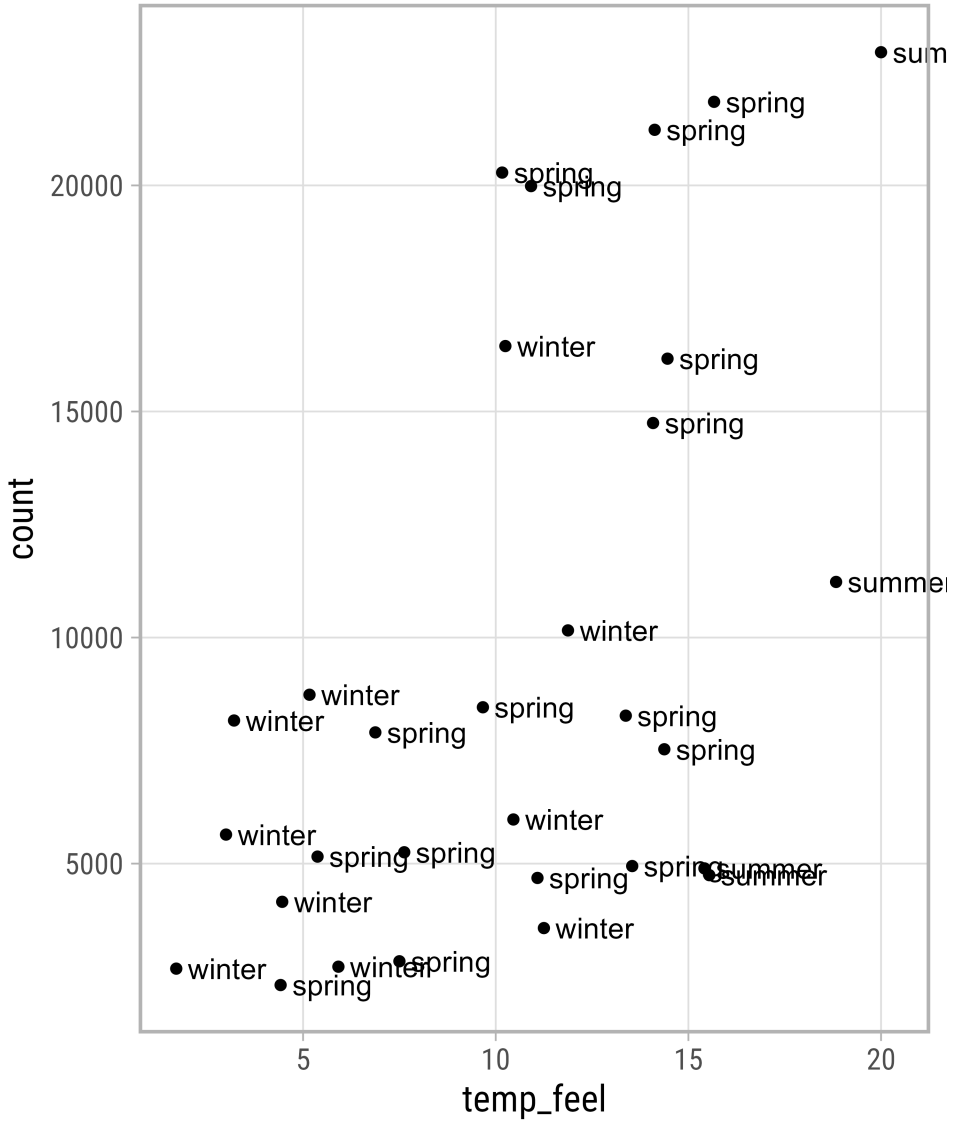
Clipping

```
ggplot(
    bikes,
    aes(x = season, y = count)
  ) +
  geom_boxplot() +
  coord_cartesian(
    ylim = c(NA, 15000),
    clip = "off"
  )
```



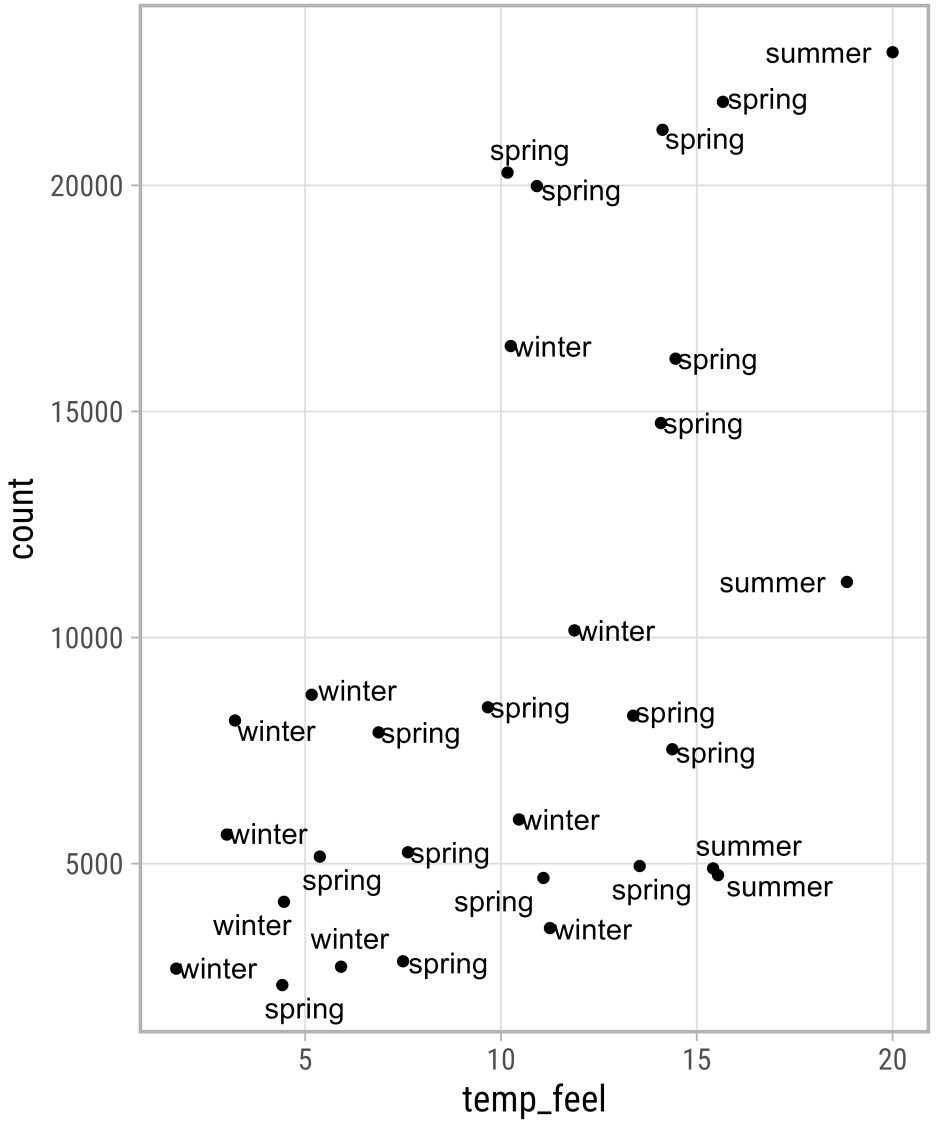
Clipping

```
ggplot(
    filter(bikes, is_holiday == TRUE),
    aes(x = temp_feel, y = count)
  ) +
  geom_point() +
  geom_text(
    aes(label = season),
    nudge_x = .3,
    hjust = 0
  ) +
  coord_cartesian(
    clip = "off"
  )
```



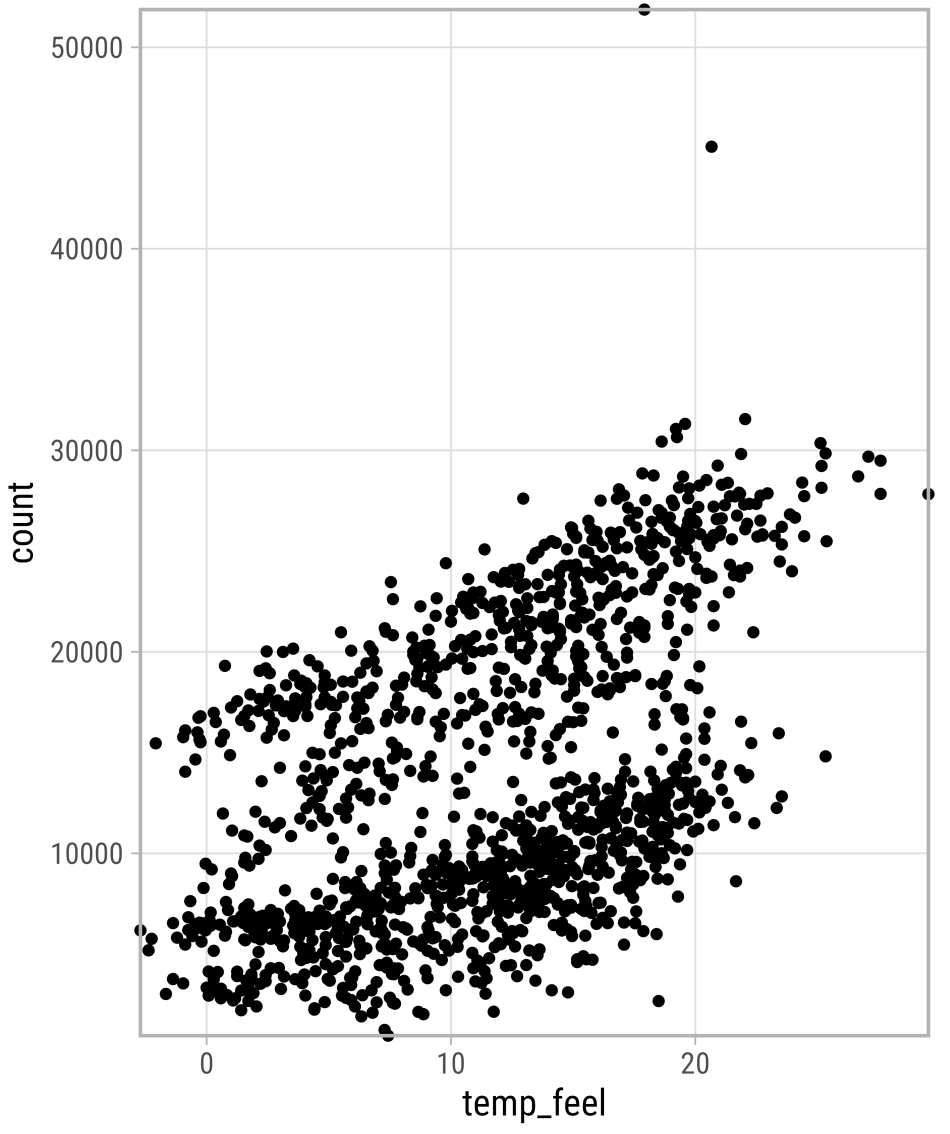
… or better use {ggrepel}

```
ggplot(
    filter(bikes, is_holiday == TRUE),
    aes(x = temp_feel, y = count)
  ) +
  geom_point() +
  ggrepel::geom_text_repel(
    aes(label = season),
    nudge_x = .3,
    hjust = 0
  ) +
  coord_cartesian(
    clip = "off"
  )
```



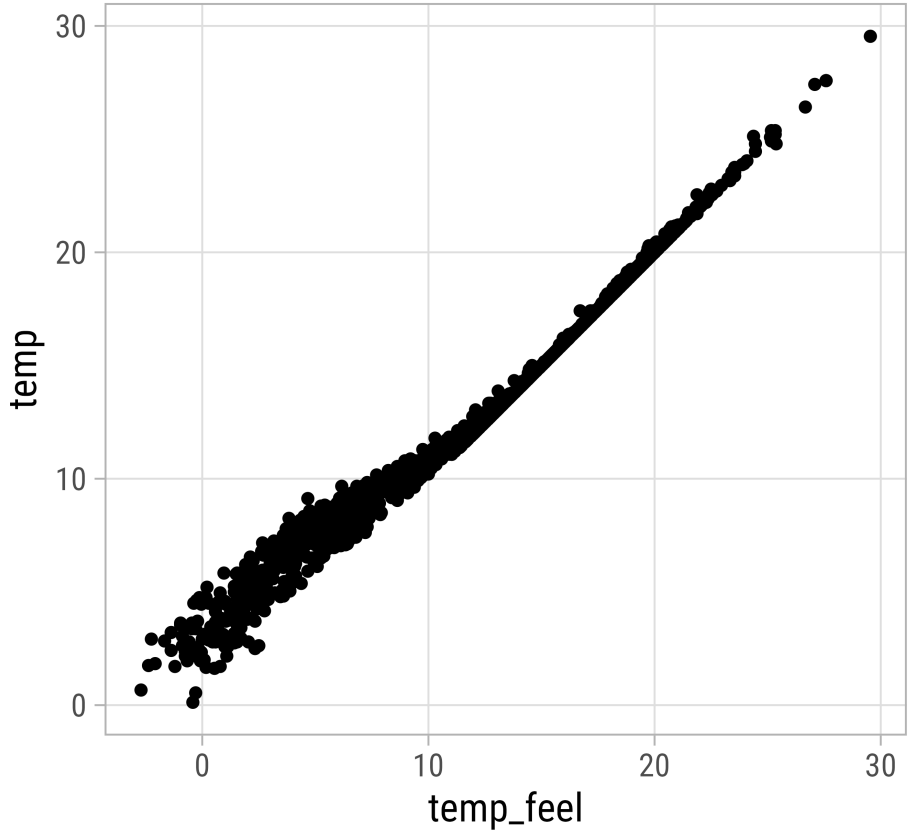
Remove All Padding

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count)
  ) +
  geom_point() +
  coord_cartesian(
    expand = FALSE,
    clip = "off"
  )
```

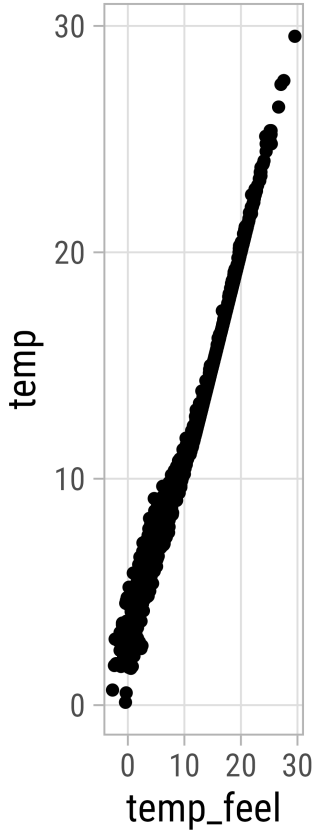


Fixed Coordinate System

```
1  ggplot(
2      bikes,
3      aes(x = temp_feel, y = temp)
4    ) +
5    geom_point() +
6    coord_fixed()
```



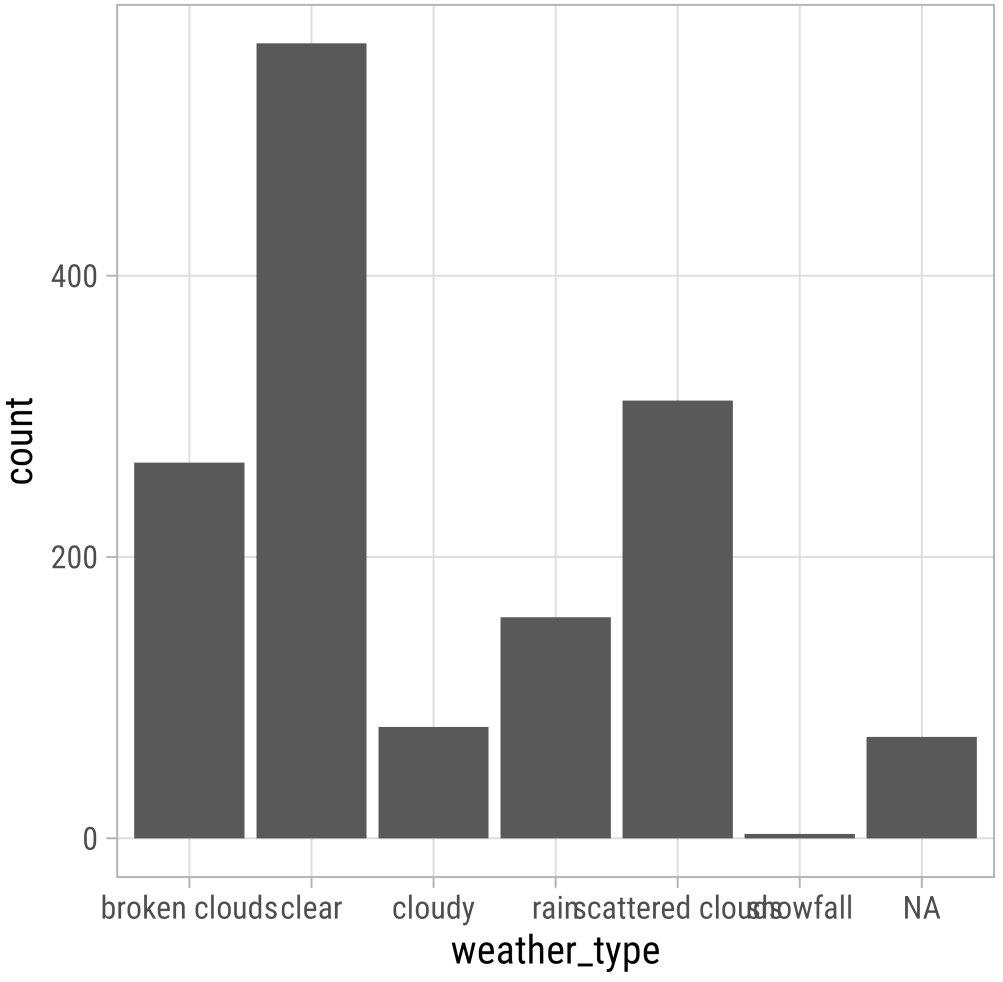

```
1  ggplot(
2      bikes,
3      aes(x = temp_feel, y = temp)
4    ) +
5    geom_point() +
6    coord_fixed(ratio = 4)
```

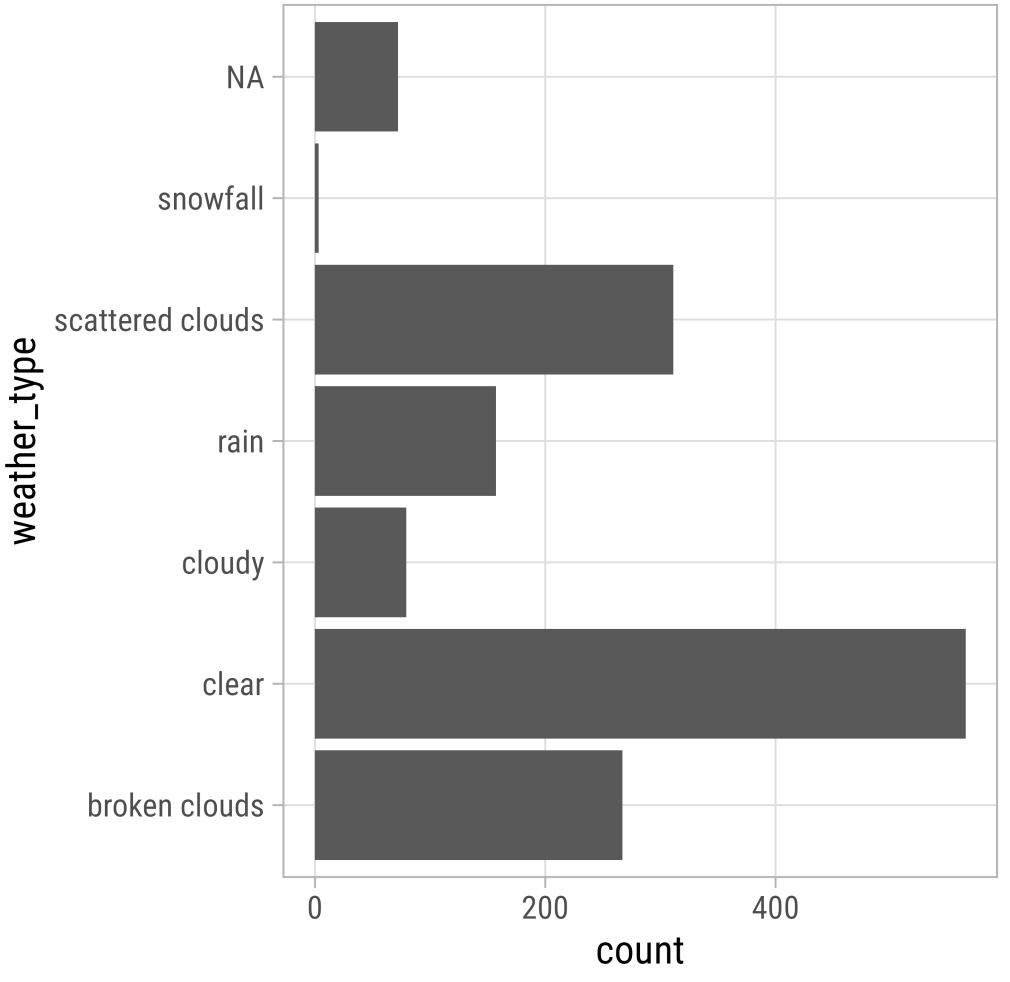


Flipped Coordinate System

```
ggplot(
    bikes,
    aes(x = weather_type)
  ) +
  geom_bar() +
  coord_cartesian()
```

```
ggplot(
    bikes,
    aes(x = weather_type)
  ) +
  geom_bar() +
  coord_flip()
```

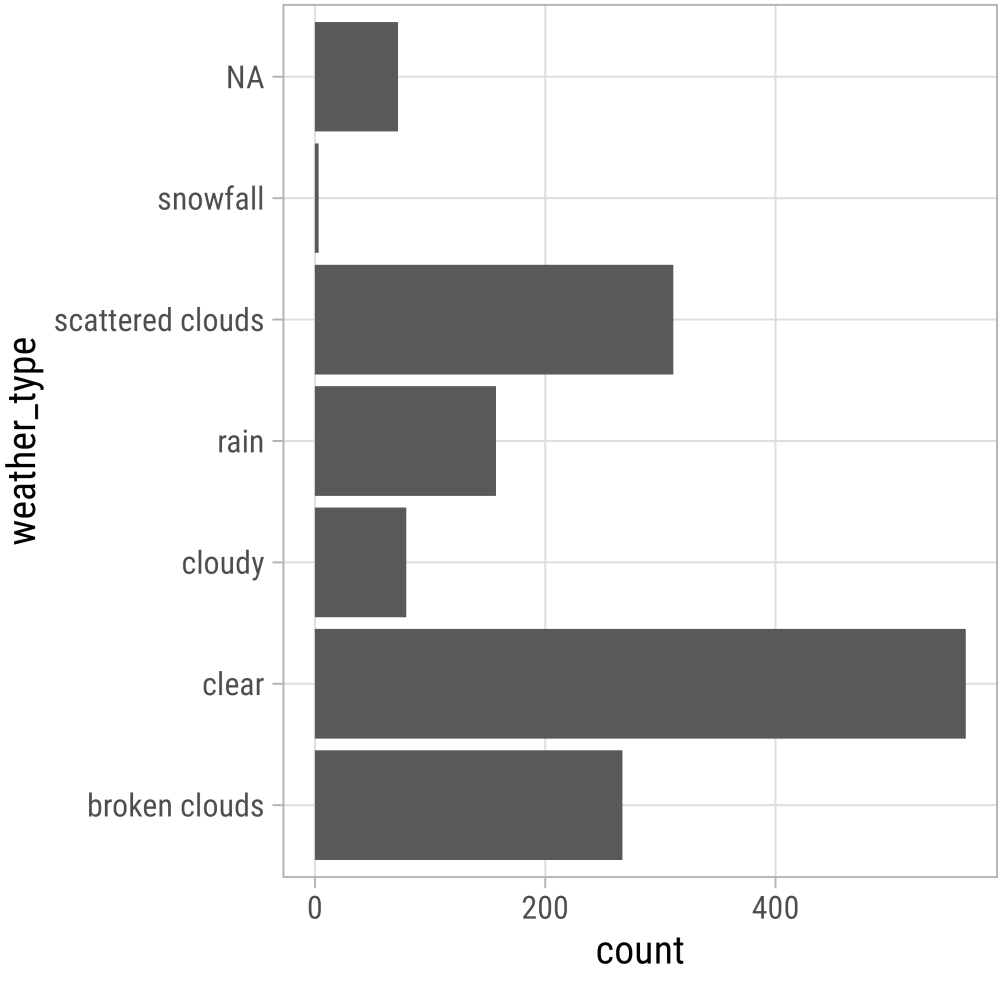

count
400
200
0
broken clouds
clear
cloudy
rain
scattered clouds
snowfall
NA
weather_type

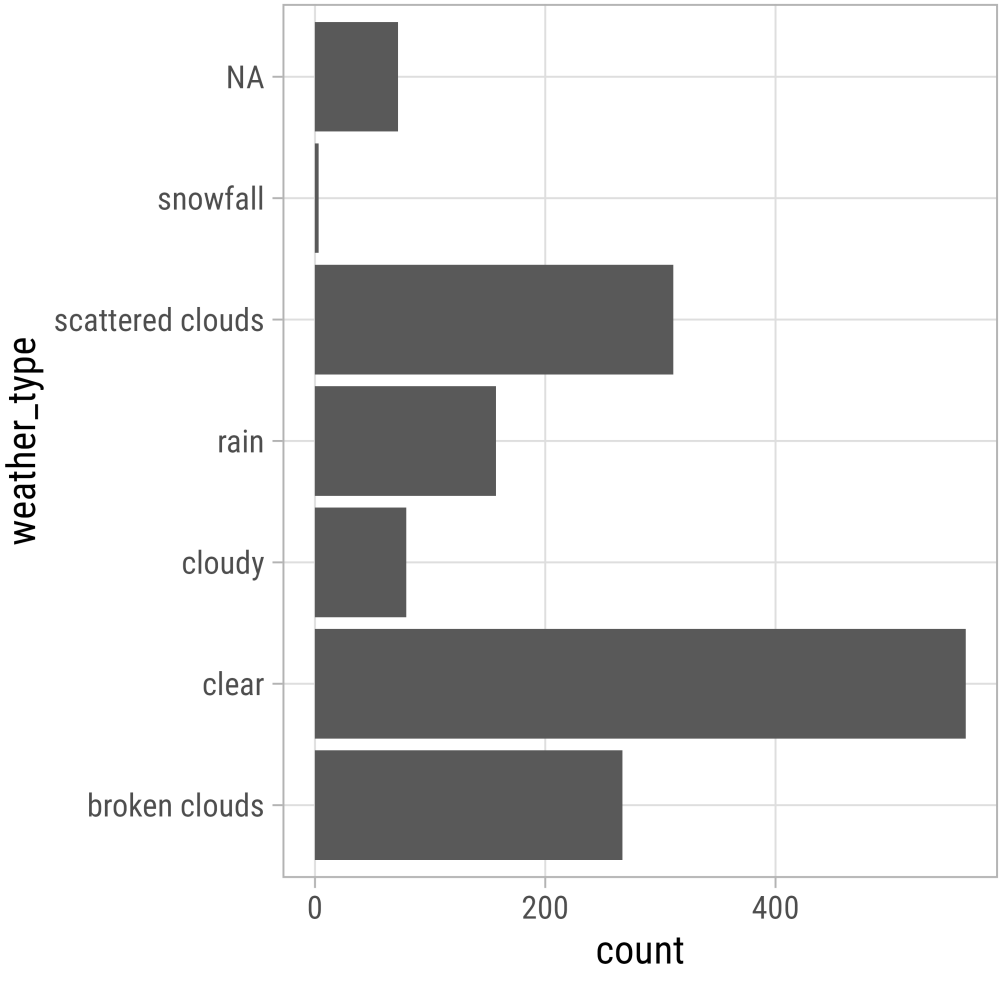

weather_type
NA
snowfall
scattered clouds
rain
cloudy
clear
broken clouds
0
200
400
count

Flipped Coordinate System

```
ggplot(
    bikes,
    aes(y = weather_type)
  ) +
  geom_bar() +
  coord_cartesian()
```

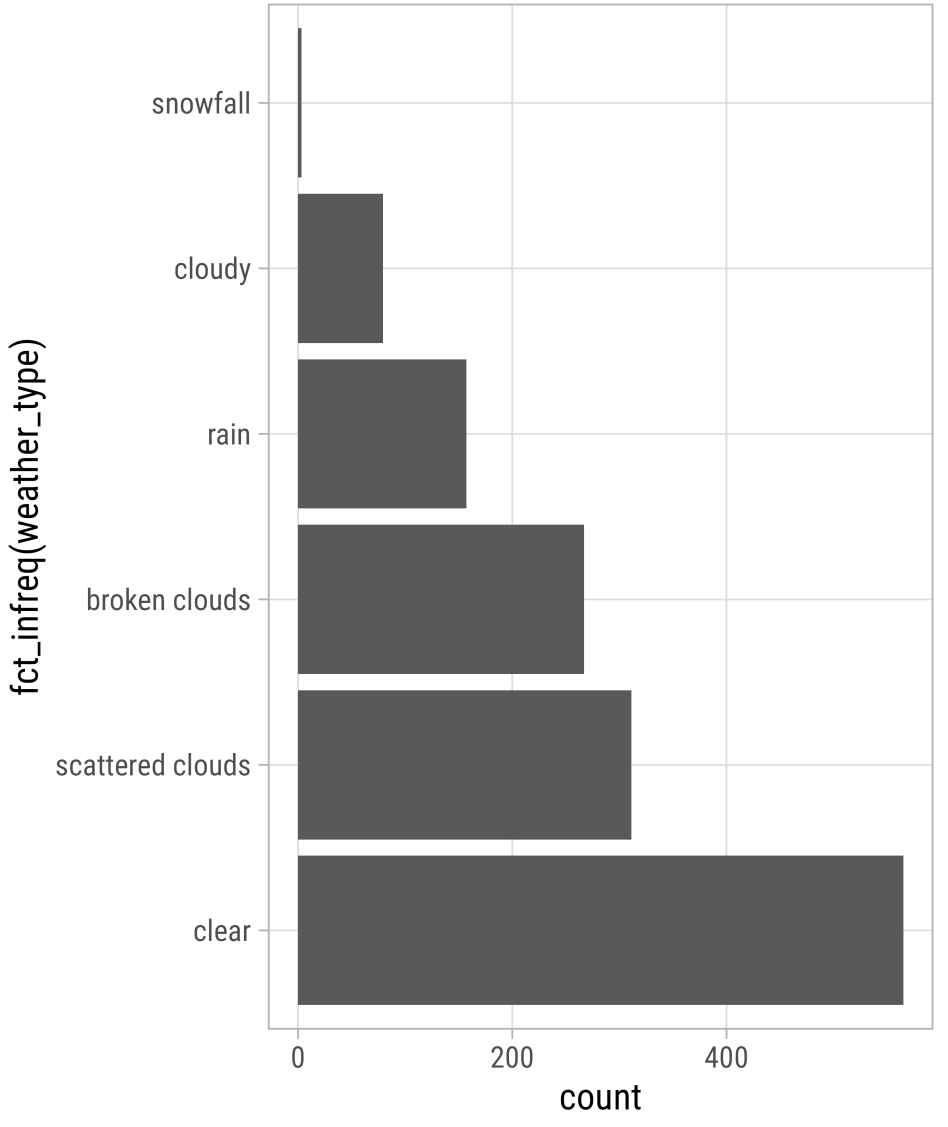
```
ggplot(
    bikes,
    aes(x = weather_type)
  ) +
  geom_bar() +
  coord_flip()
```


weather_type
NA
snowfall
scattered clouds
rain
cloudy
clear
broken clouds
0
200
400
count


weather_type
NA
snowfall
scattered clouds
rain
cloudy
clear
broken clouds
0
200
400
count

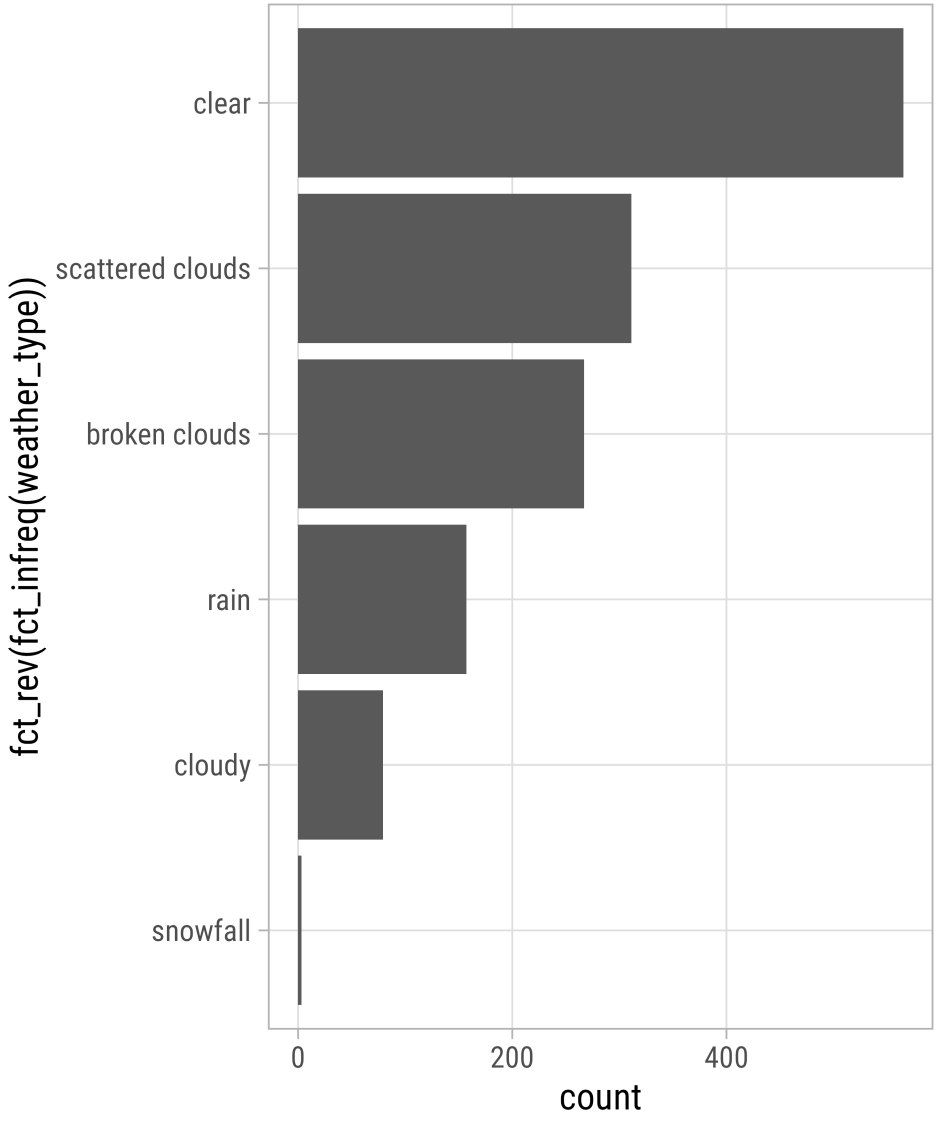
Reminder: Sort Your Bars!

```
1  ggplot(
2      filter(bikes, !is.na(weather_type)),
3      aes(y = fct_infreq(weather_type))
4    ) +
5    geom_bar()
```



Reminder: Sort Your Bars!

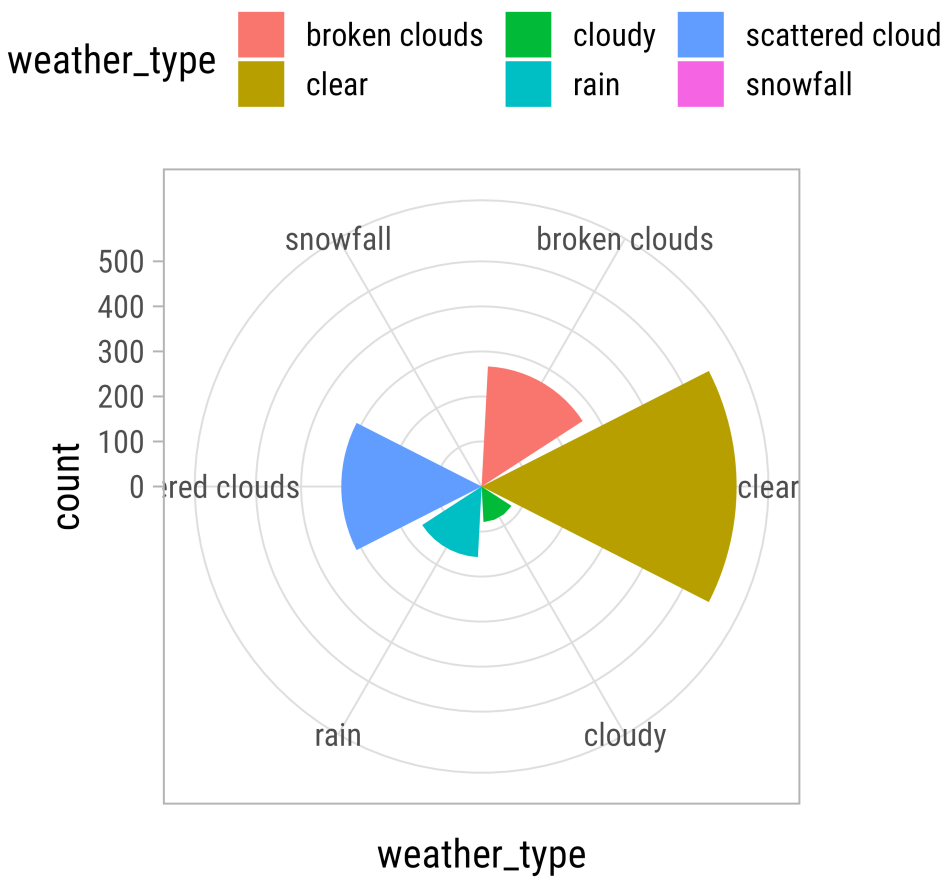
```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(y = fct_rev(
      fct_infreq(weather_type)
    ))
  ) +
  geom_bar()
```

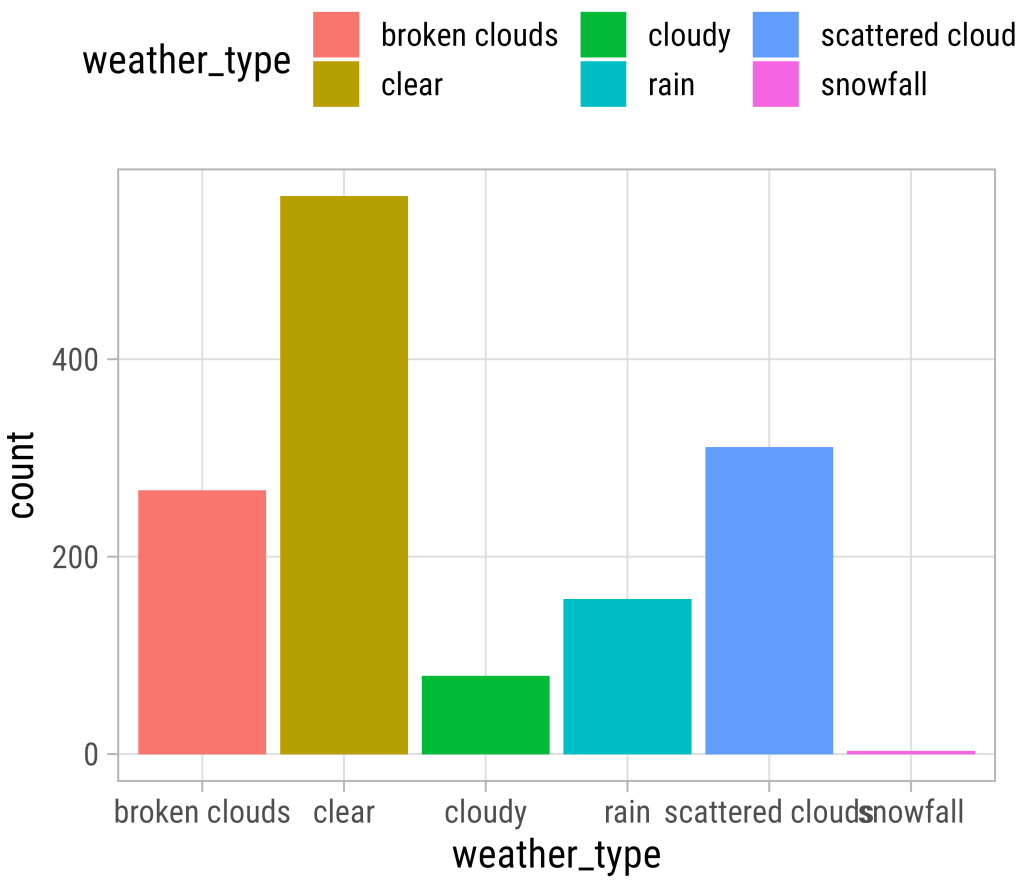


Circular Corrdinate System

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = weather_type,
        fill = weather_type)
  ) +
  geom_bar() +
  coord_polar()
```

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = weather_type,
        fill = weather_type)
  ) +
  geom_bar() +
  coord_cartesian()
```

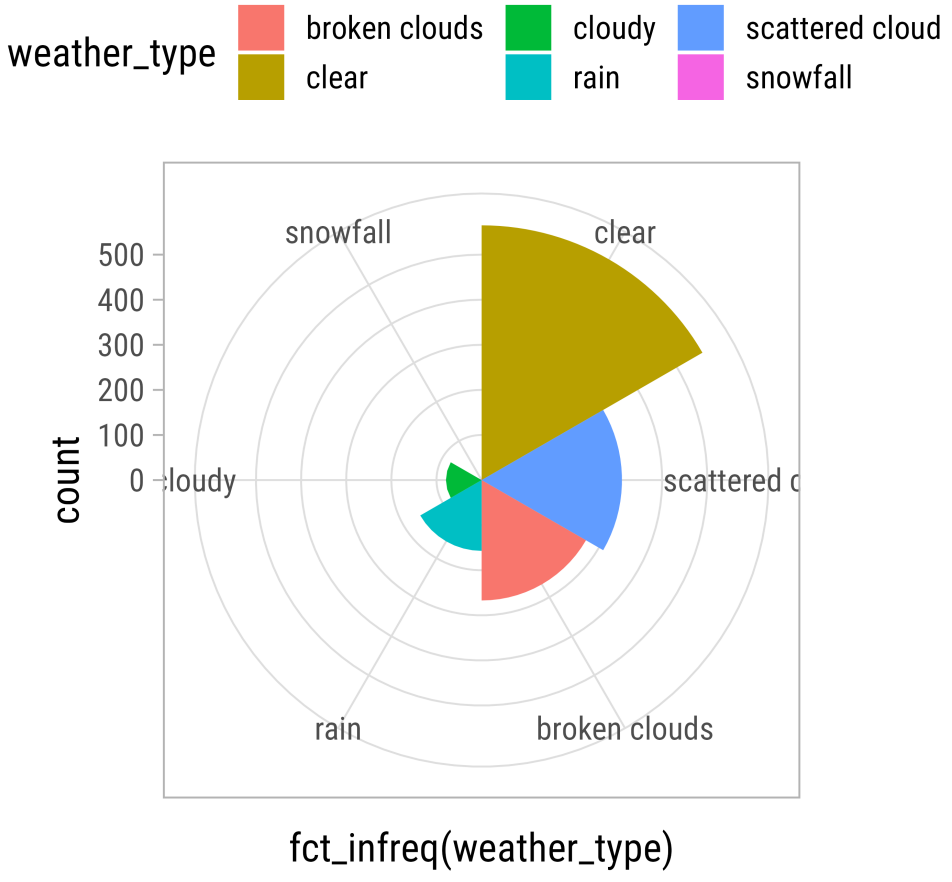

weather_type
broken clouds
cloudy
scattered cloud
clear
rain
snowfall
snowfall
broken clouds
500
400
300
200
100
0
red clouds
clear
rain
cloudy
count
weather_type

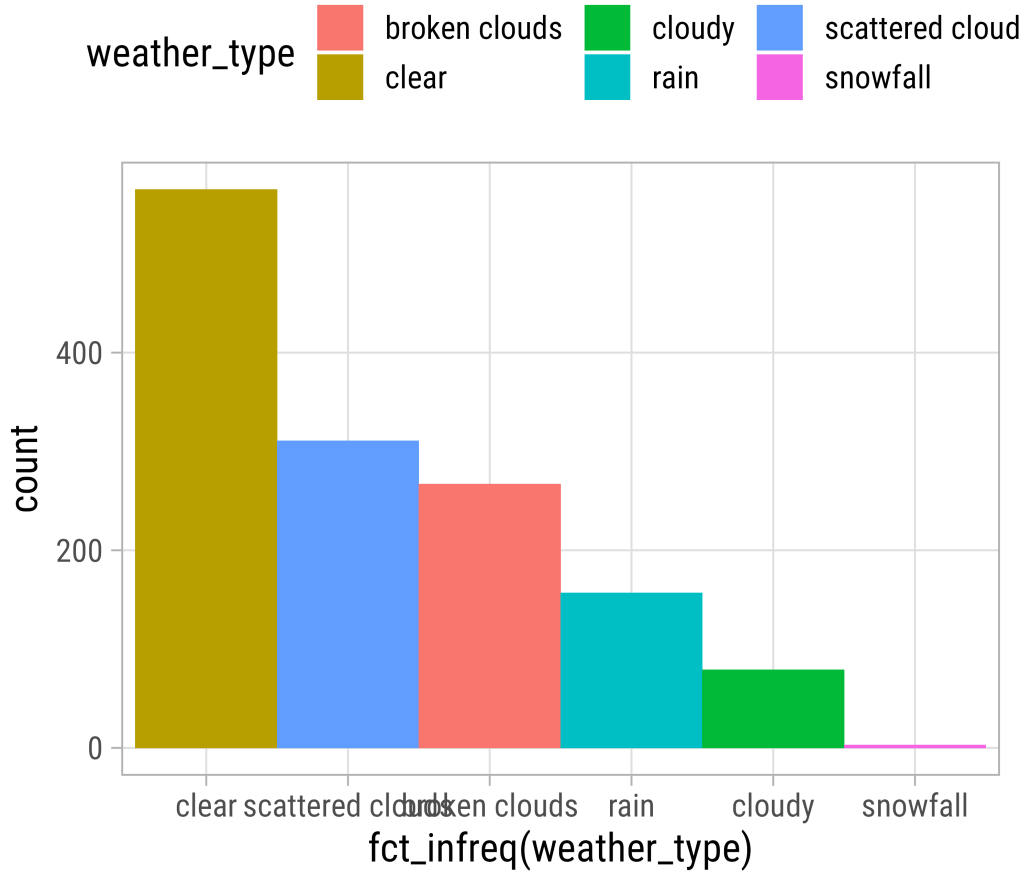

weather_type
broken clouds
cloudy
scattered cloud
clear
rain
snowfall
400
200
0
count
broken clouds
clear
cloudy
rain
scattered clouds
snowfall
weather_type

Circular Corrdinate System

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = fct_infreq(weather_type),
        fill = weather_type)
  ) +
  geom_bar(width = 1) +
  coord_polar()
```

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = fct_infreq(weather_type),
        fill = weather_type)
  ) +
  geom_bar(width = 1) +
  coord_cartesian()
```

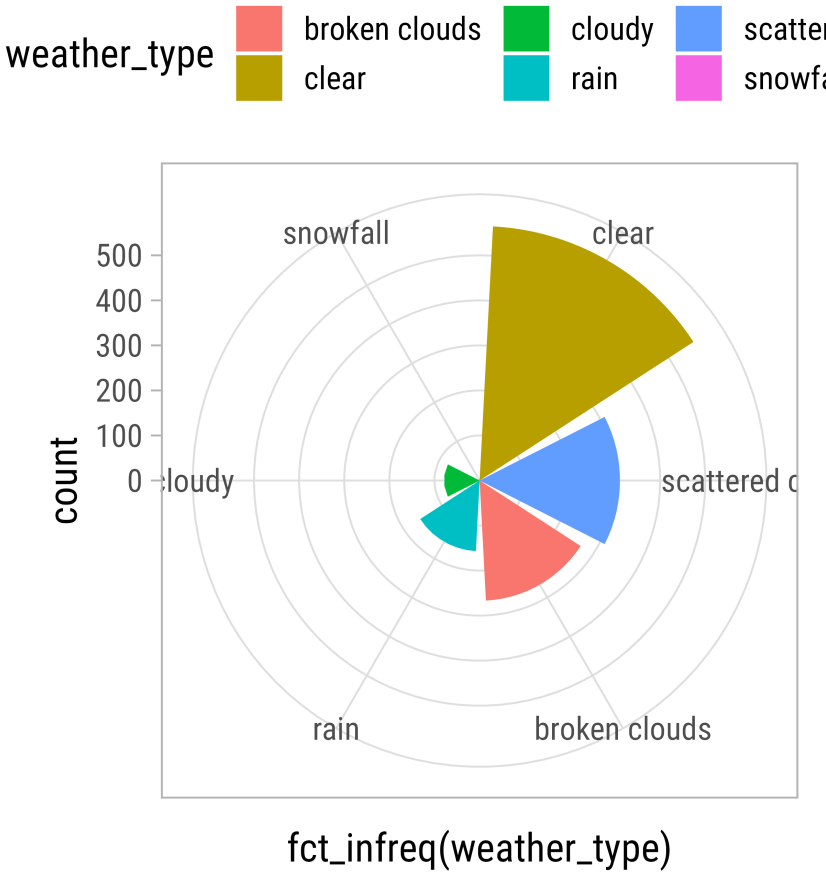

weather_type
broken clouds
cloudy
scattered cloud
clear
rain
snowfall
500
400
300
200
100
0
count
snowfall
clear
loudy
scattered c
rain
broken clouds
fct_infreq(weather_type)

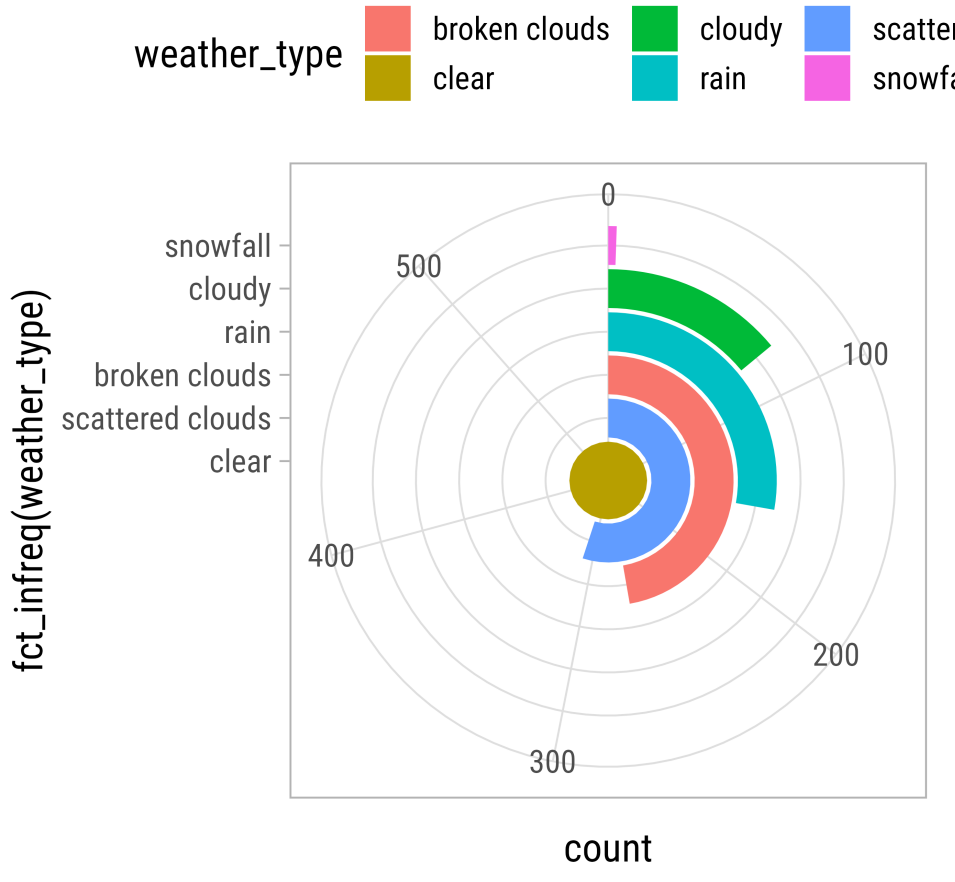

weather_type
broken clouds
cloudy
scattered cloud
clear
rain
snowfall
400
200
0
count
clear
scattered clouds
broken clouds
rain
cloudy
snowfall
fct_infreq(weather_type)

Circular Corrdinate System

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = fct_infreq(weather_type),
        fill = weather_type)
  ) +
  geom_bar() +
  coord_polar(theta = "x")
```

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = fct_infreq(weather_type),
        fill = weather_type)
  ) +
  geom_bar() +
  coord_polar(theta = "y")
```

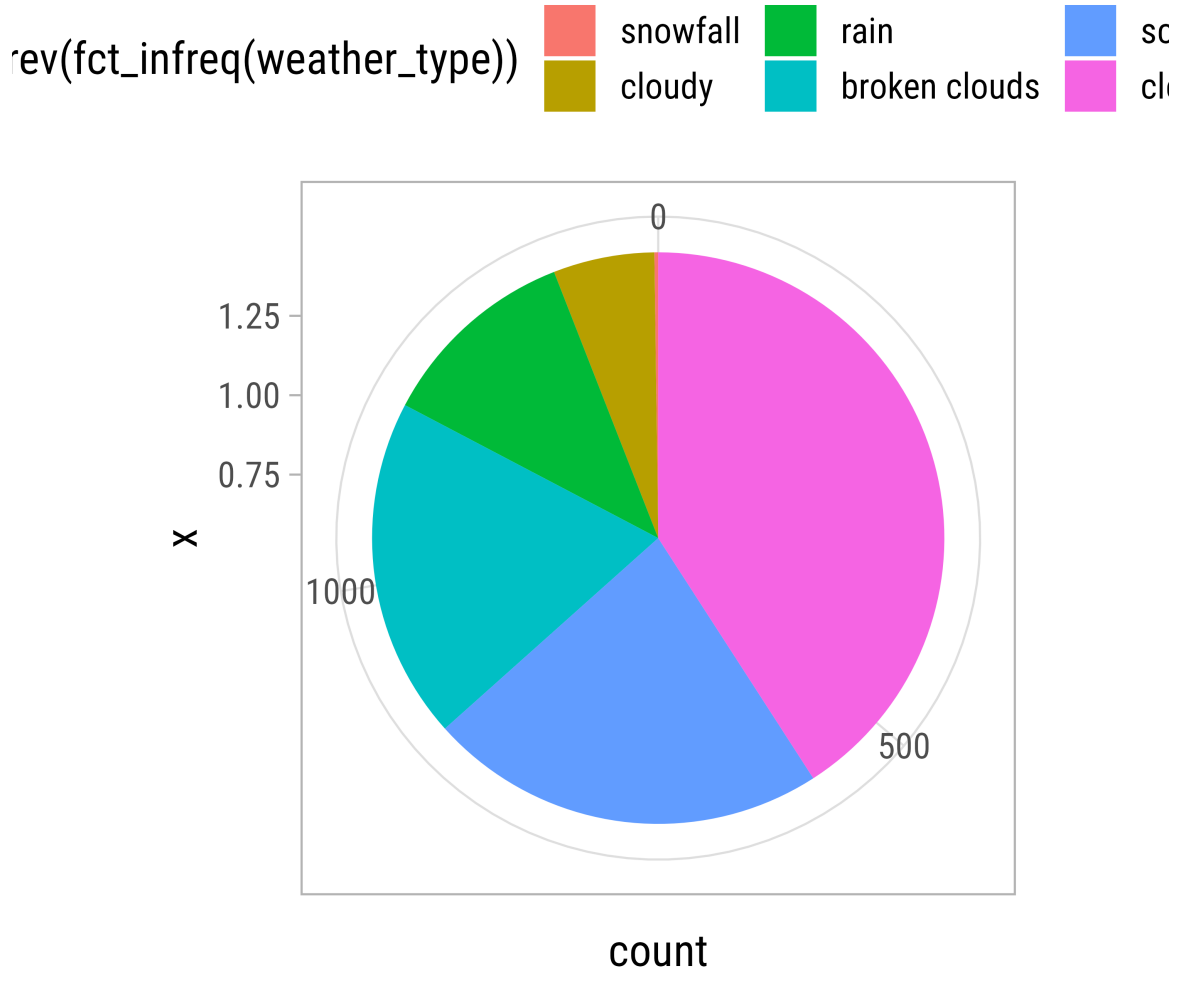

weather_type
broken clouds
cloudy
scattered cloud
clear
rain
snowfall
count
500
400
300
200
100
0
snowfall
clear
loudy
scattered c
rain
broken clouds
fct_infreq(weather_type)

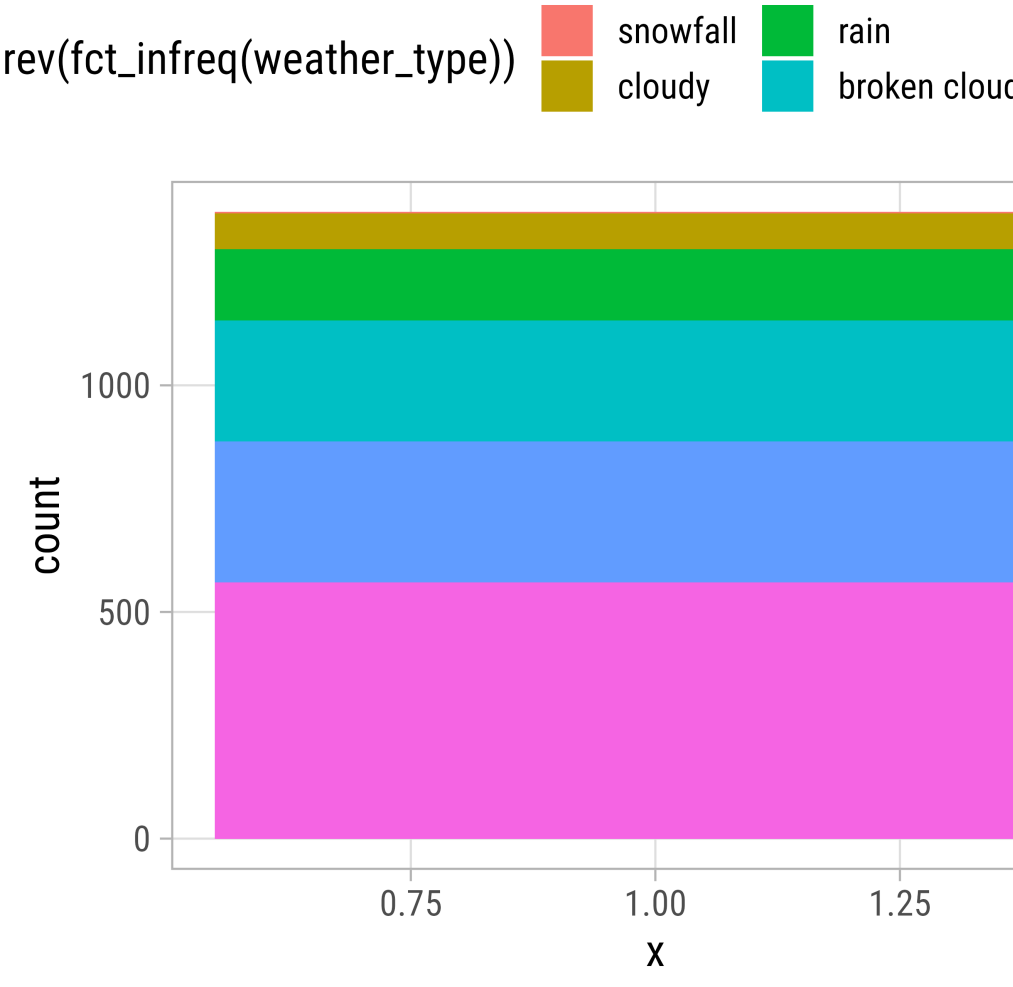

weather_type
broken clouds
cloudy
scatterec
clear
rain
snowfall
fct_infreq(weather_type)
snowfall
cloudy
rain
broken clouds
scattered clouds
clear
0
100
200
300
400
500
count

Circular Corrdinate System

```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = 1,
        fill = fct_rev(fct_infreq(weather_type)))
  ) +
  geom_bar(position = "stack") +
  coord_polar(theta = "y")
```

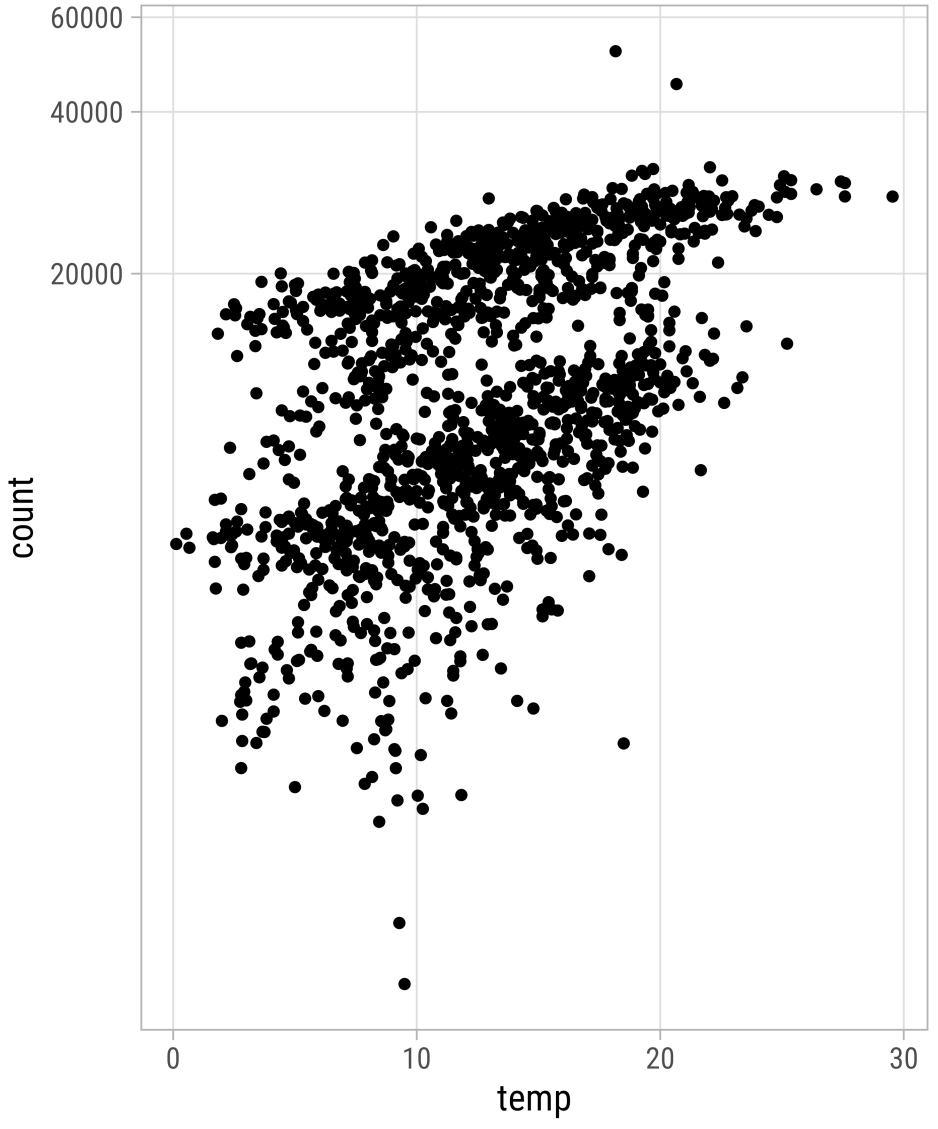
```
ggplot(
    filter(bikes, !is.na(weather_type)),
    aes(x = 1,
        fill = fct_rev(fct_infreq(weather_type)
  ) +
  geom_bar(position = "stack") +
  coord_cartesian()
```


rev(fct_infreq(weather_type))
snowfall
cloudy
rain
broken clouds
0
1.25
1.00
0.75
x
1000
500
count


rev(fct_infreq(weather_type))
snowfall
cloudy
rain
broken cloud
count
1000
500
0
0.75
1.00
1.25
x

Transform a Coordinate System

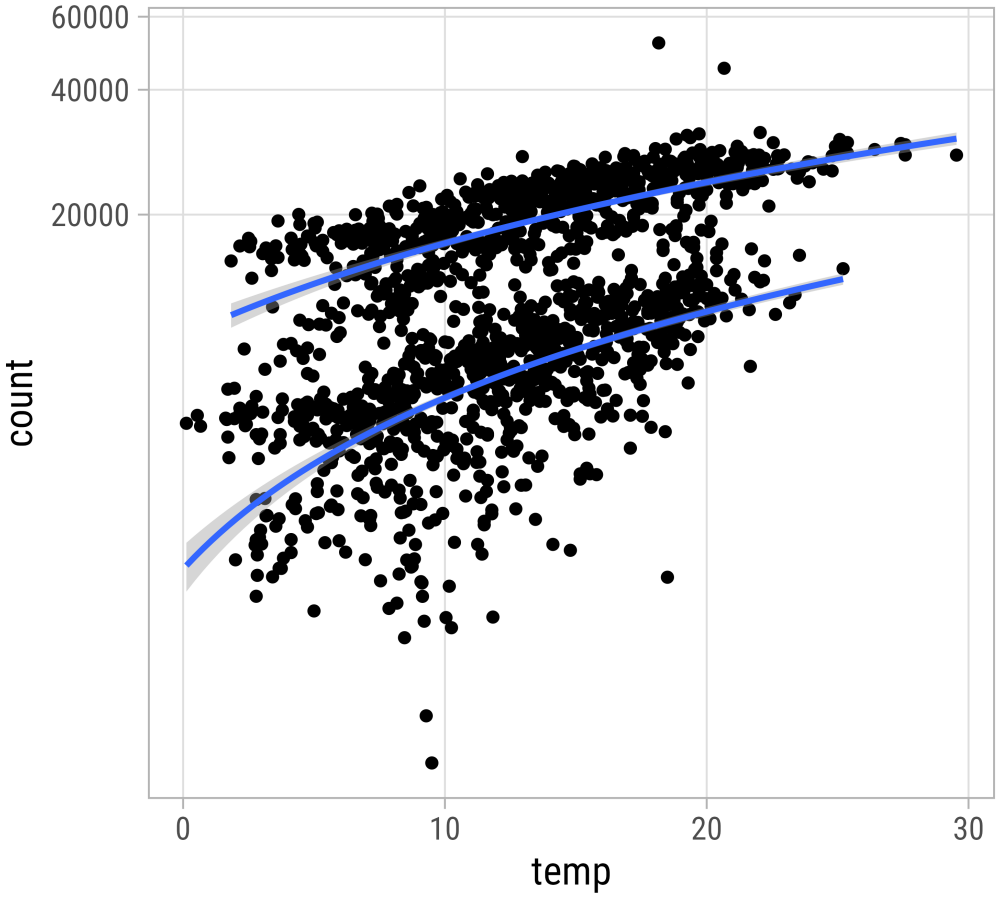
```
1  ggplot(
2      bikes,
3      aes(x = temp, y = count)
4    ) +
5    geom_point() +
6    coord_trans(y = "log10")
```

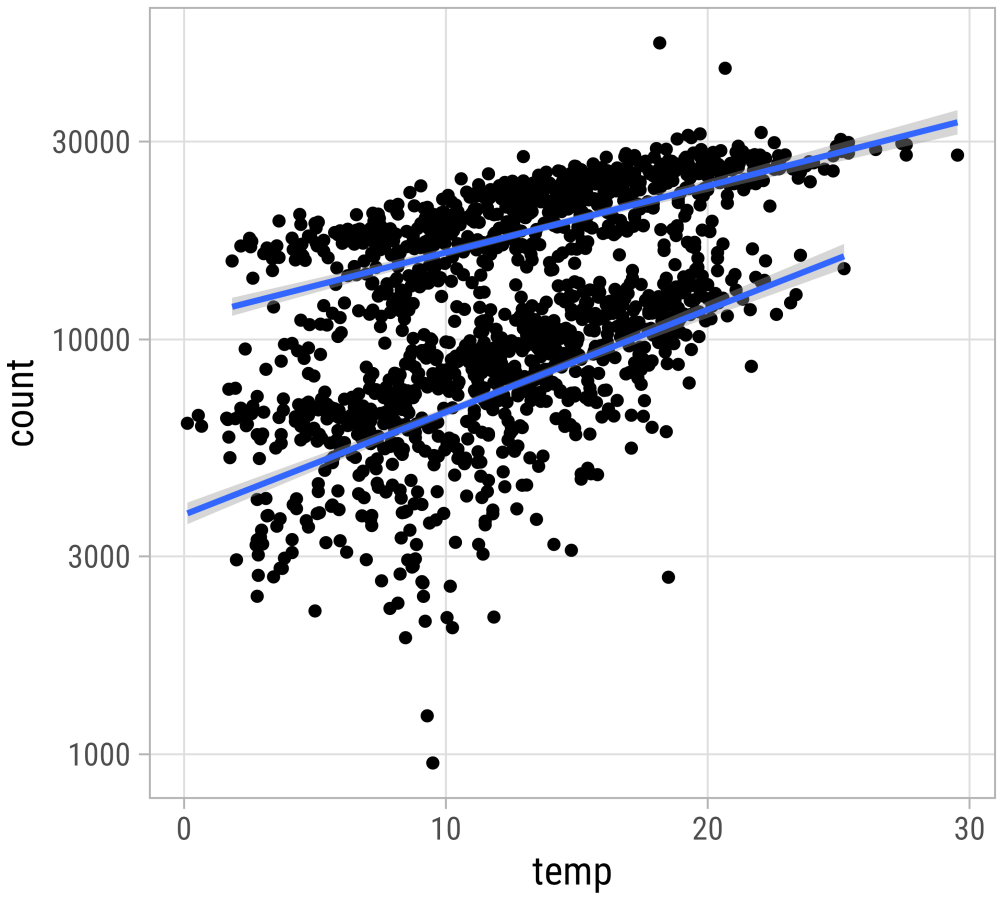


Transform a Coordinate System

```
ggplot(
    bikes,
    aes(x = temp, y = count,
        group = day_night)
  ) +
  geom_point() +
  geom_smooth(method = "lm") +
  coord_trans(y = "log10")
```

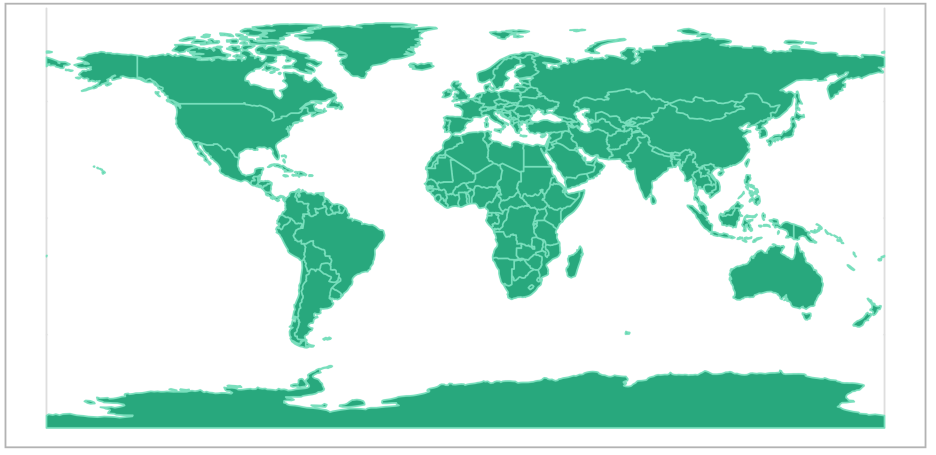
```
ggplot(
    bikes,
    aes(x = temp, y = count,
        group = day_night)
  ) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_y_log10()
```


60000
40000
20000
count
0
10
20
30
temp


30000
10000
3000
1000
count
0
10
20
30
temp

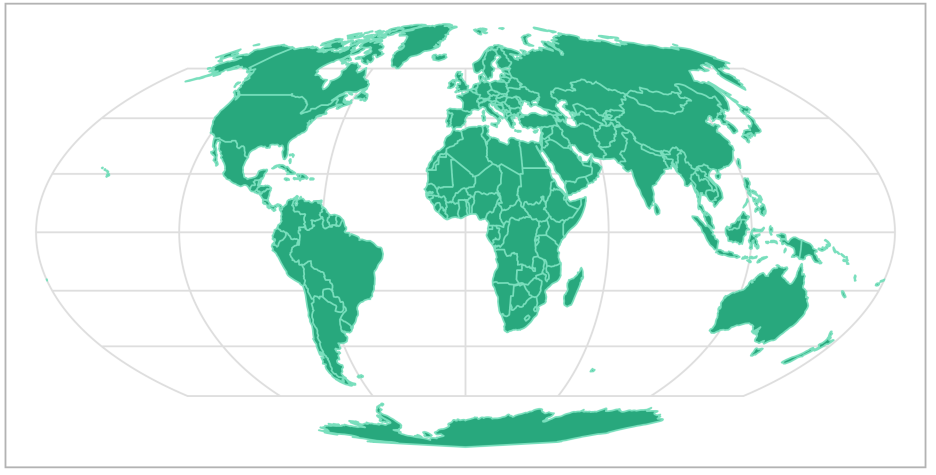
Spatial Coordinate (Reference) Systems

```r
countries <- rnaturalearth::ne_countries(
  returnclass = "sf"
)

ggplot() +
  geom_sf(
    data = countries,
    color = "#79dfbd",
    fill = "#28a87d",
    size = .3
  )
```



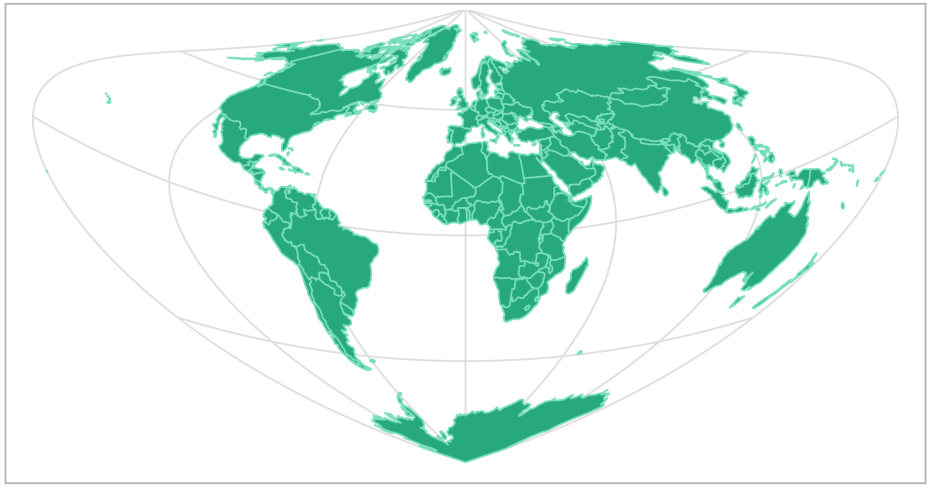
Spatial Coordinate (Reference) Systems

```
ggplot() +
  geom_sf(
    data = countries,
    color = "#79dfbd",
    fill = "#28a87d",
    size = .3
  ) +
  coord_sf(
    crs = "+proj=moll"
  )
```



Spatial Coordinate (Reference) Systems

```
ggplot() +
  geom_sf(
    data = countries,
    color = "#79dfbd",
    fill = "#28a87d",
    size = .3
  ) +
  coord_sf(
    crs = "+proj=bonne +lat_1=10"
  )
```

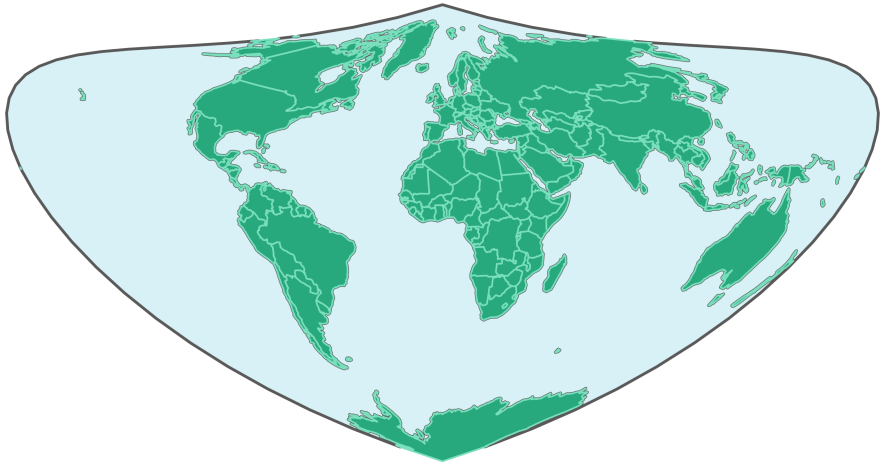


Spatial Coordinate (Reference) Systems

```
oceans <- rnaturalearth::ne_download(
  category = "physical", type = "ocean", retu
)
```

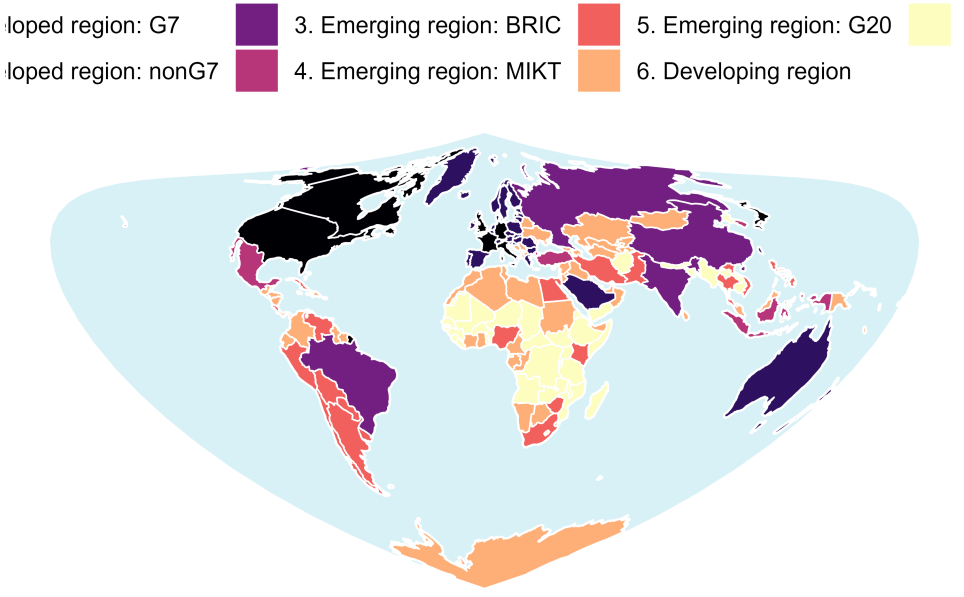
```
OGR data source with driver: ESRI
Shapefile
Source:
"C:\Users\DataVizard\AppData\Local\Tem
 layer: "ne_110m_ocean"
with 2 features
It has 3 fields
```

```
ggplot() +
  geom_sf(
    data = oceans,
    fill = "#d8f1f6"
  ) +
  geom_sf(
    data = countries,
    color = "#79dfbd",
    fill = "#28a87d",
    size = .3
  ) +
  coord_sf(
    crs = "+proj=bonne +lat_1=10"
  ) +
  theme_void()
```



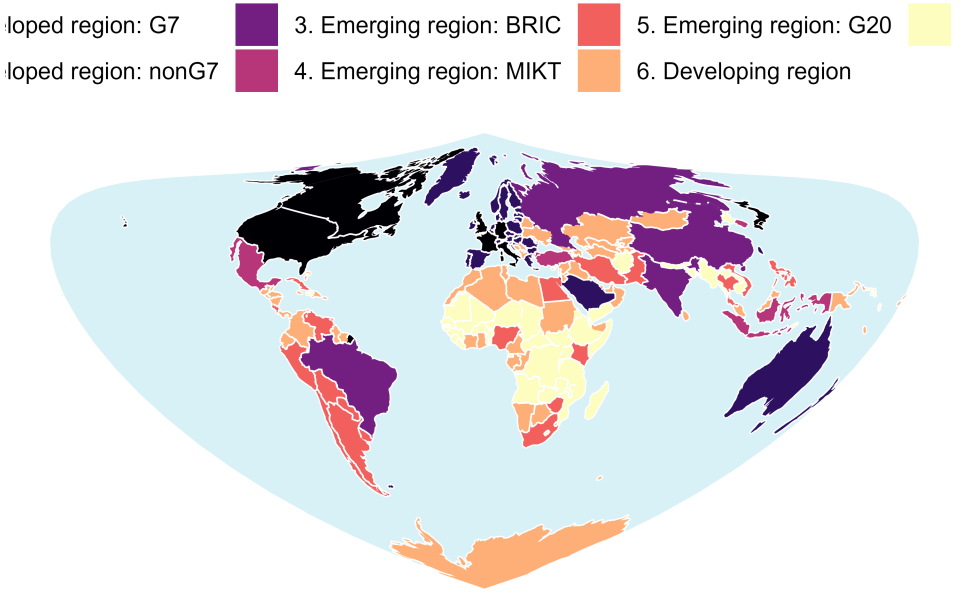
Mapping of Visual Properties

```
ggplot() +
  geom_sf(
    data = oceans,
    fill = "#d8f1f6",
    color = "white"
  ) +
  geom_sf(
    data = countries,
    aes(fill = economy),
    color = "white",
    size = .3
  ) +
  coord_sf(
    crs = "+proj=bonne +lat_1=10"
  ) +
  scale_fill_viridis_d(option = "magma") +
  theme_void() +
  theme(legend.position = "top")
```



Better Borders

```
borders <- rmapshaper::ms_innerlines(countries

ggplot() +
  geom_sf(
    data = oceans,
    fill = "#d8f1f6",
    color = "white"
  ) +
  geom_sf(
    data = countries,
    aes(fill = economy),
    color = "transparent"
  ) +
  geom_sf(
    data = borders,
    fill = "transparent",
    color = "white",
    size = .3
  ) +
  coord_sf(
    crs = "+proj=bonne +lat_1=10"
  ) +
  scale_fill_viridis_d(option = "magma") +
```



Recap

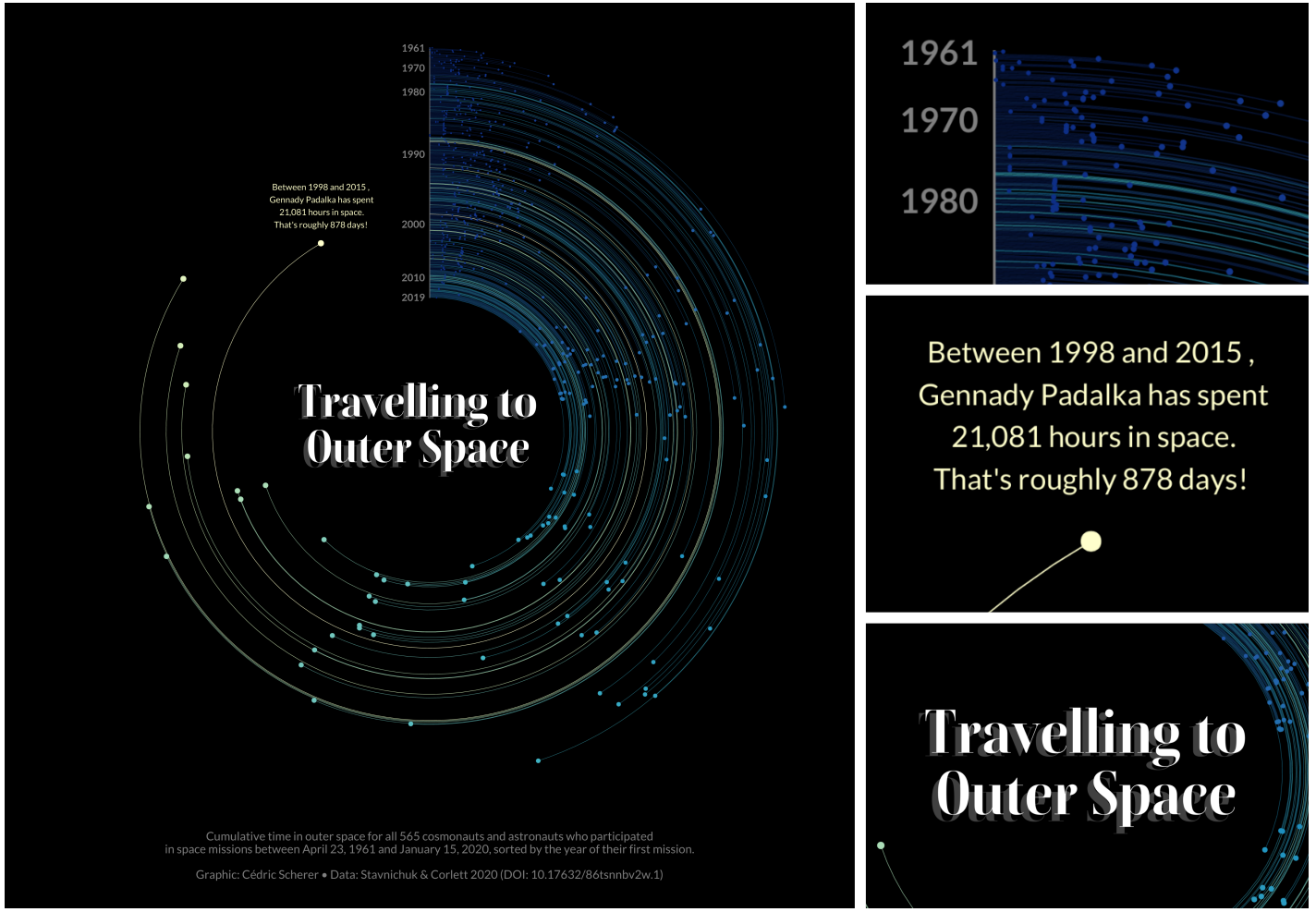
- `facet_*()` functions allow to create small multiples
- `scale_*()` functions translate between `aes`thetics and properties
- use `*_continuous()` for numerical data
- … and `*_discrete()` for categorical data
- use `scale_color|fill_*()` to customize data-related colors
- `coord_*()` functions interpret the positional aesthetics
- be careful when adjusting axis limits:
  - `scale_*_continuous(limits)` removes the data outside the range
  - `coord_*(*lim)` keeps the data and zooms into the range
- `{ggplot2}` + `{sf}` are a powerful duo to create maps in R

Exercises

Exercise 1

- Have a look at the following visualization of the cumulative time that cosmo- and astronauts have spent in outer space. The data also contains information on the year of their first and last travel, respectively.
- Together with your group, discuss which layers and modifications are needed to create such a chart with `{ggplot2}`.
- Note down the aesthetics, geometries, and scales used for each element of this graphic.
- What is the coordinate system? Have any adjustments been made?
- Which theme was used and how was it modified?

10:00


1961
1970
1980
1990
2000
2010
2019
Between 1998 and 2015 , Gennady Padalka has spent 21,081 hours in space. That's roughly 878 days!
Travelling to Outer Space
Cumulative time in outer space for all 565 cosmonauts and astronauts who participated in space missions between April 23, 1961 and January 15, 2020, sorted by the year of their first mission.
Graphic: Cédric Scherer • Data: Stavnichuk & Corlett 2020 (DOI: 10.17632/86tsnnbv2w.1)

I am done with the exercise!
What now?

↓

Put on a **green** sticky note.

↓

Is my neighbor done
with the exercise as well?

Yes | **No**

Compare and discuss your solutions.

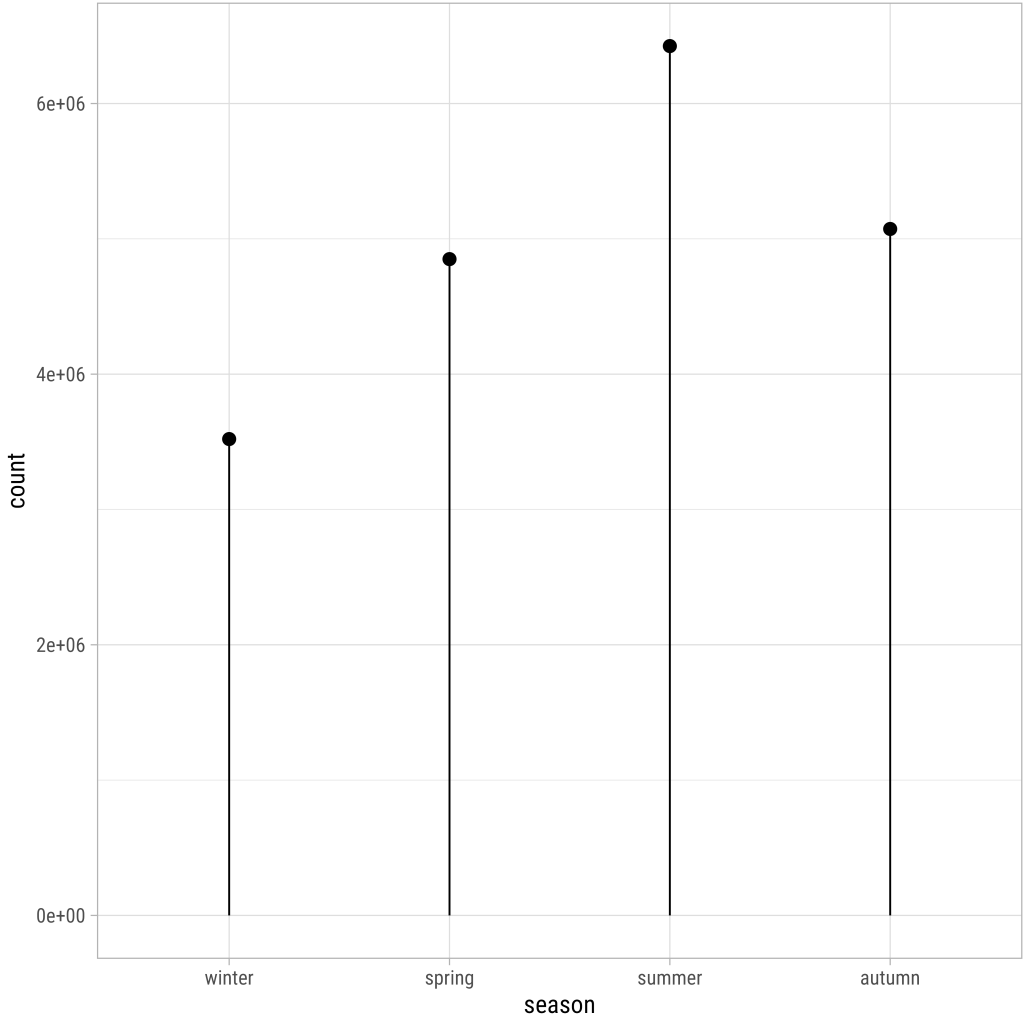
Ask if you can help out.

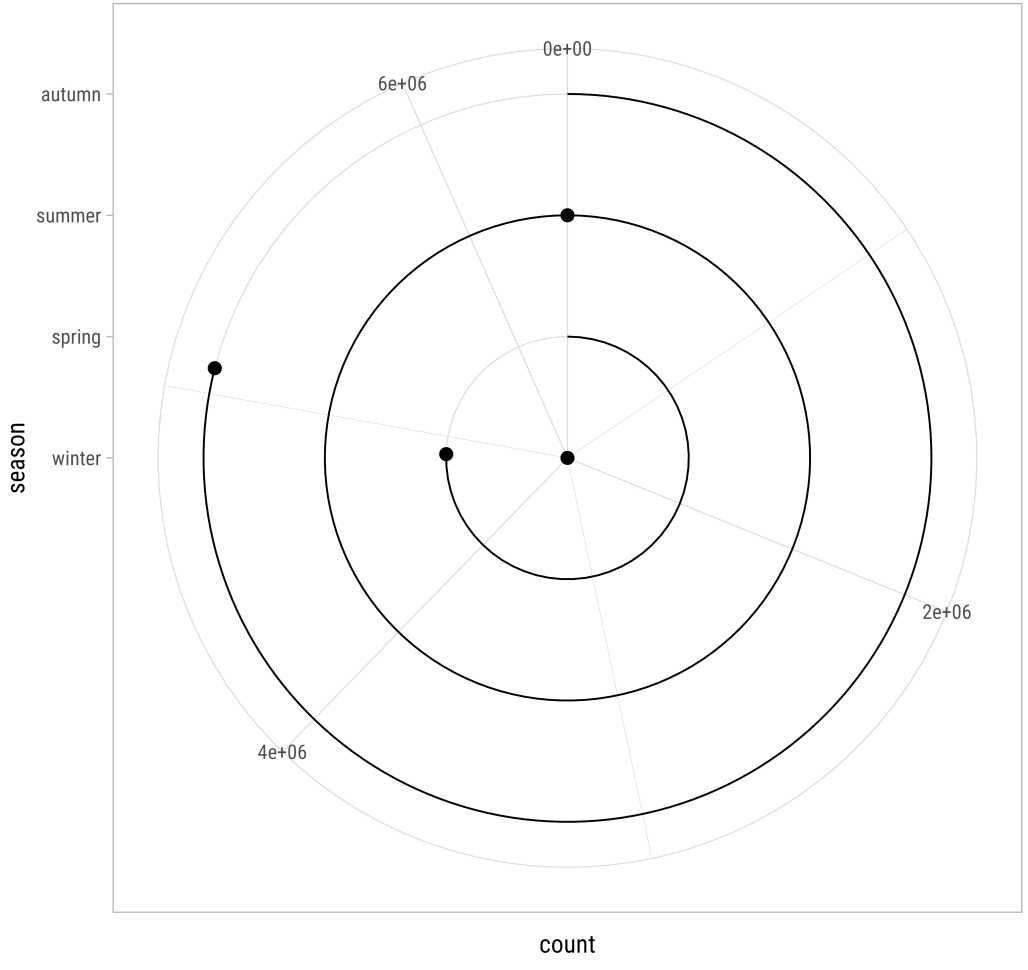
... or take a short break.

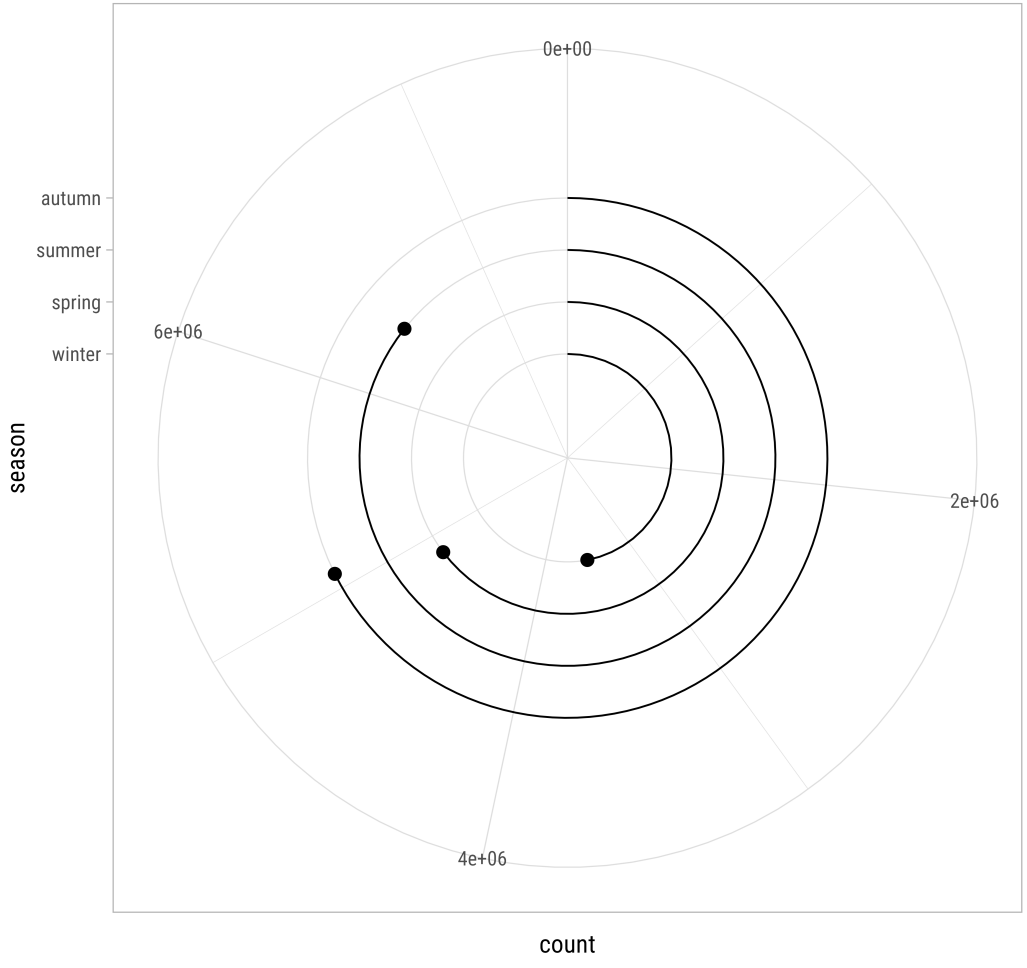
Exercise 2

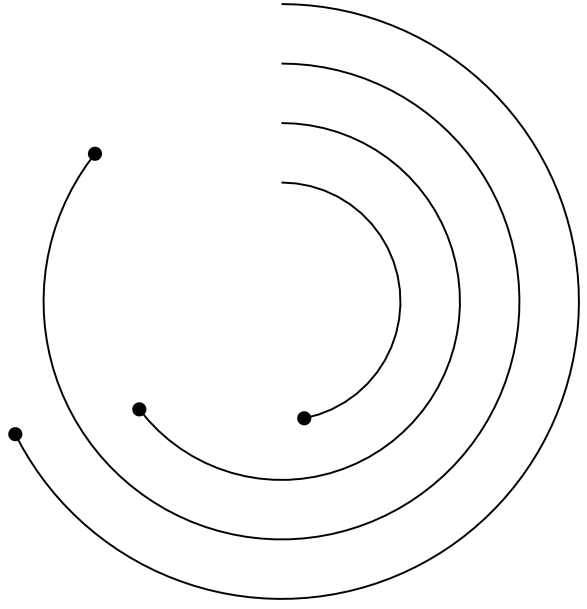
- Open the script `exercises/03_concepts_pt2_ex2.qmd`.
- Create a circular lollipop plot of reported bike shares per season.
- The data is not in the right format as there are no totals.
  How can you solve it?
- Remove all legend elements (with a single line of code).
- How can you add the labels next to the starting point of each lollipop?
- How could you add a baseline?

30:00

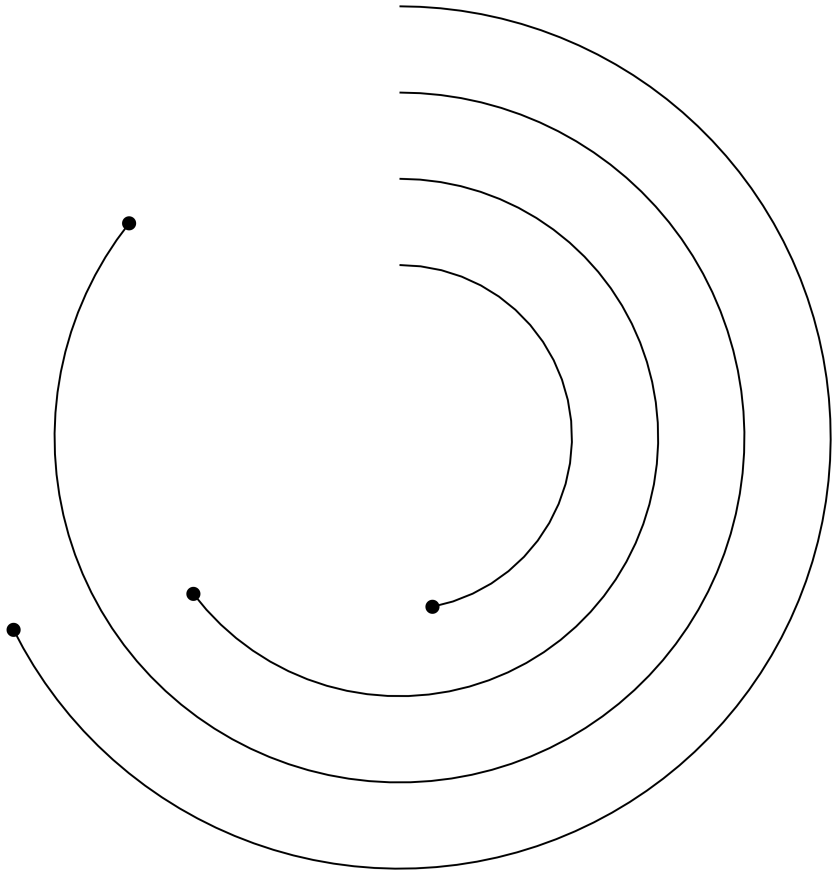

6e+06
4e+06
2e+06
0e+00
count
winter
spring
summer
autumn
season

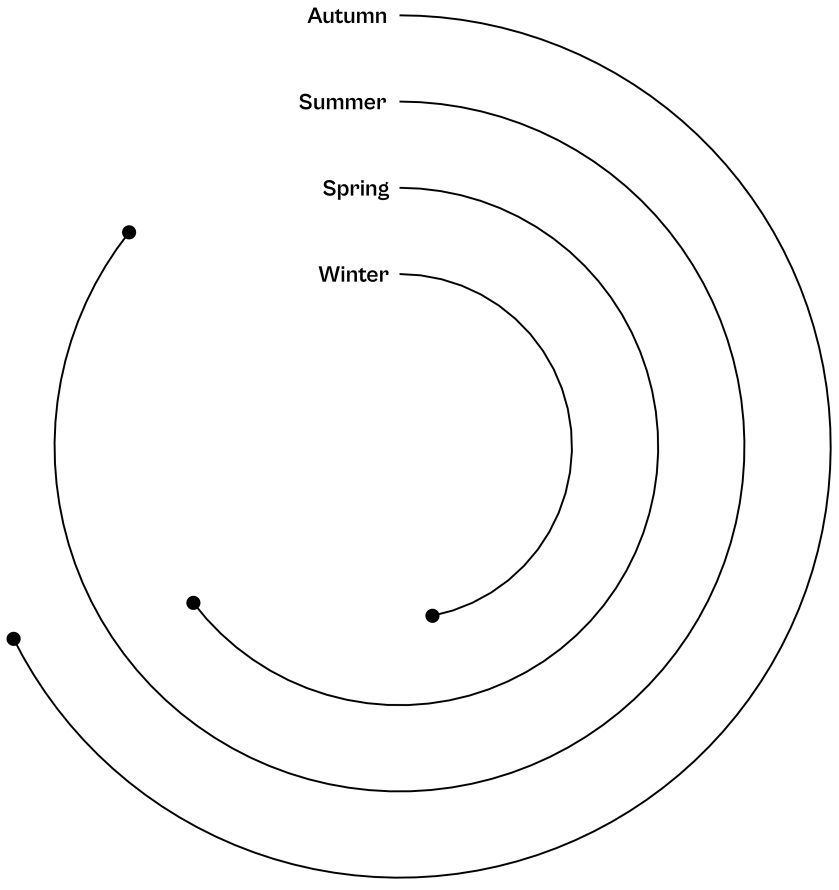

0e+00
6e+06
2e+06
4e+06
autumn
summer
spring
winter
season
count

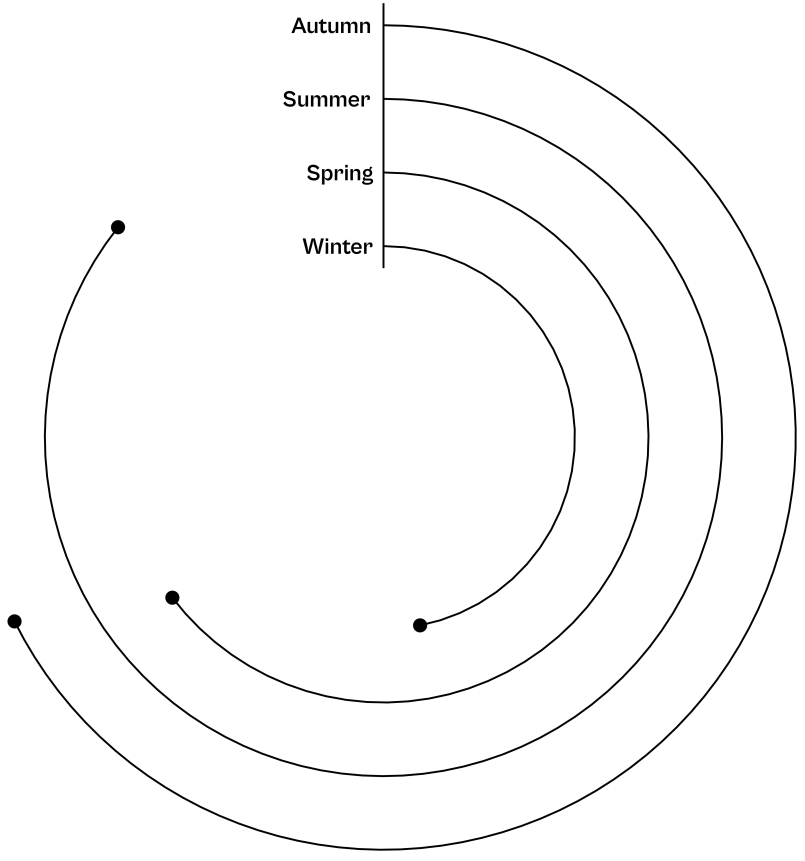

0e+00
2e+06
4e+06
6e+06
autumn
summer
spring
winter
season
count



Cédric Scherer // rstudio::conf // July 2022




Autumn
Summer
Spring
Winter


Autumn
Summer
Spring
Winter

I am done with the exercise!
What now?

↓

Put on a **green** sticky note.

↓

Is my neighbor done
with the exercise as well?

Yes | **No**

Compare and discuss your solutions.

Ask if you can help out.

... or take a short break.