Exercise 2

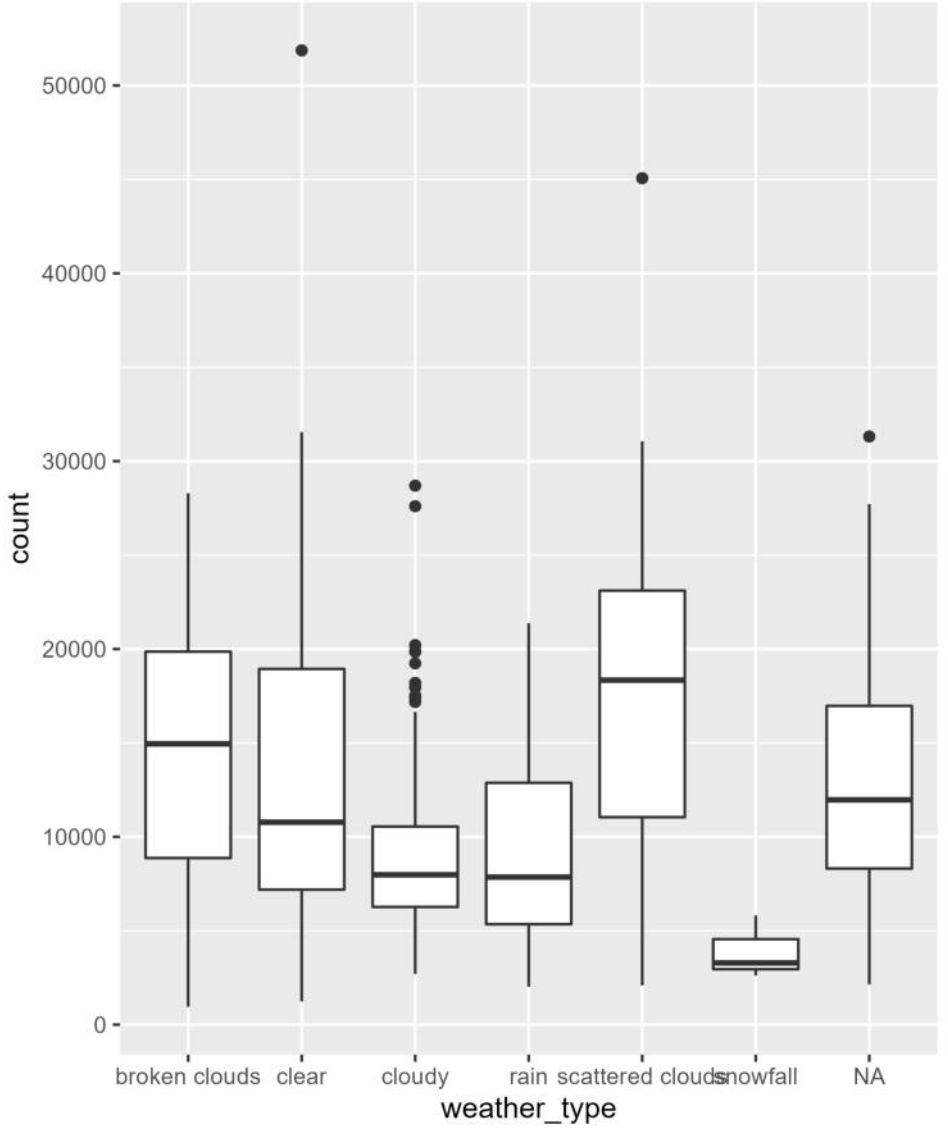
- Explore the TfL bike sharing data visually:
  create a boxplot of counts per weather type
  - Turn the plot into a jitter strips plot (random noise across the x axis)
  - Combine both chart types (jittered points on top of the boxplots)
  - Bonus: Sort the boxplot-jitter hybrid by median counts
  - Apply your favorite theme to the plot.
  - Add meaningful labels.
  - Bonus: Explore other chart types to visualize the distributions.
- Save the plot as a vector graphic with a decent plot size.

Import Data (if not yet)

```
bikes <- readr::read_csv(
  here::here("data", "london-bikes-custom.csv"),
  col_types = "Dcfffilllddddc"
)

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

library(tidyverse)
```

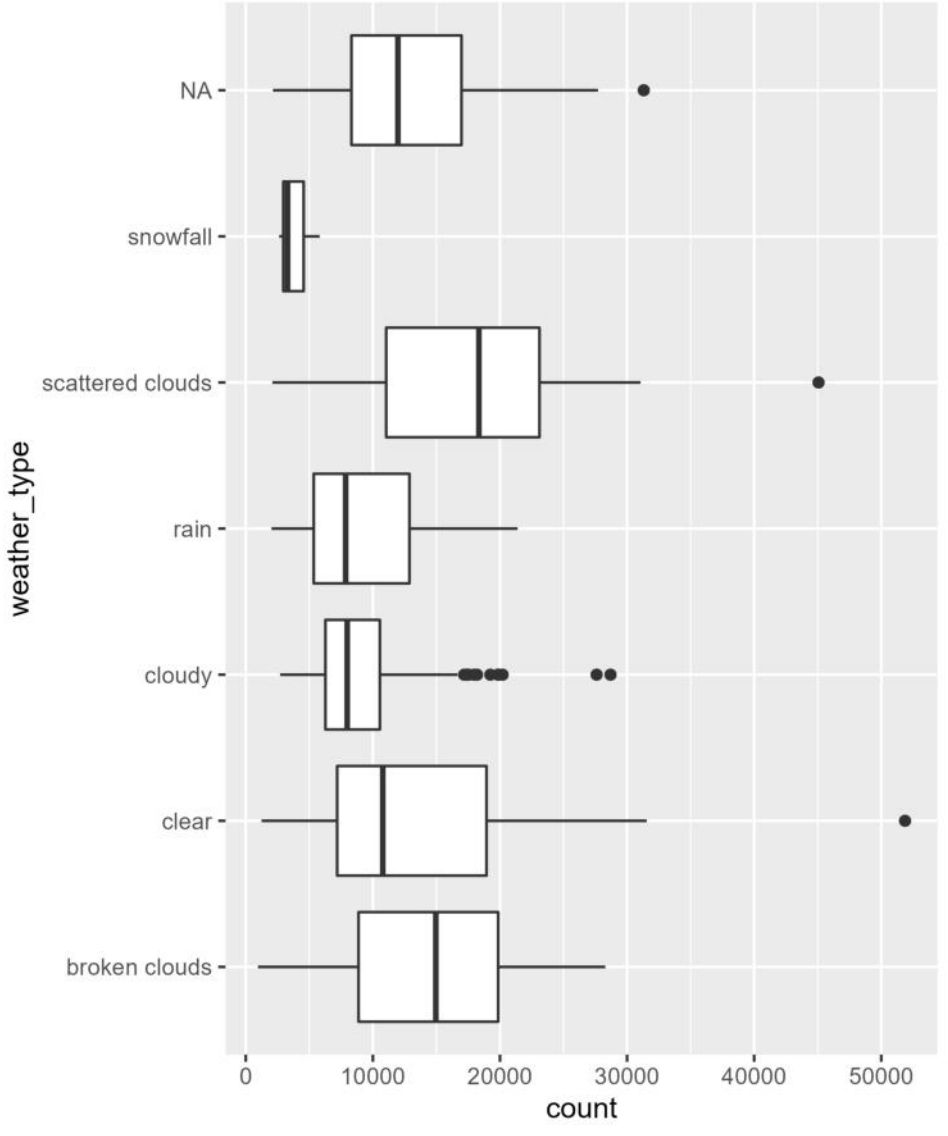
Boxplot of Counts vs. Weather Type

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



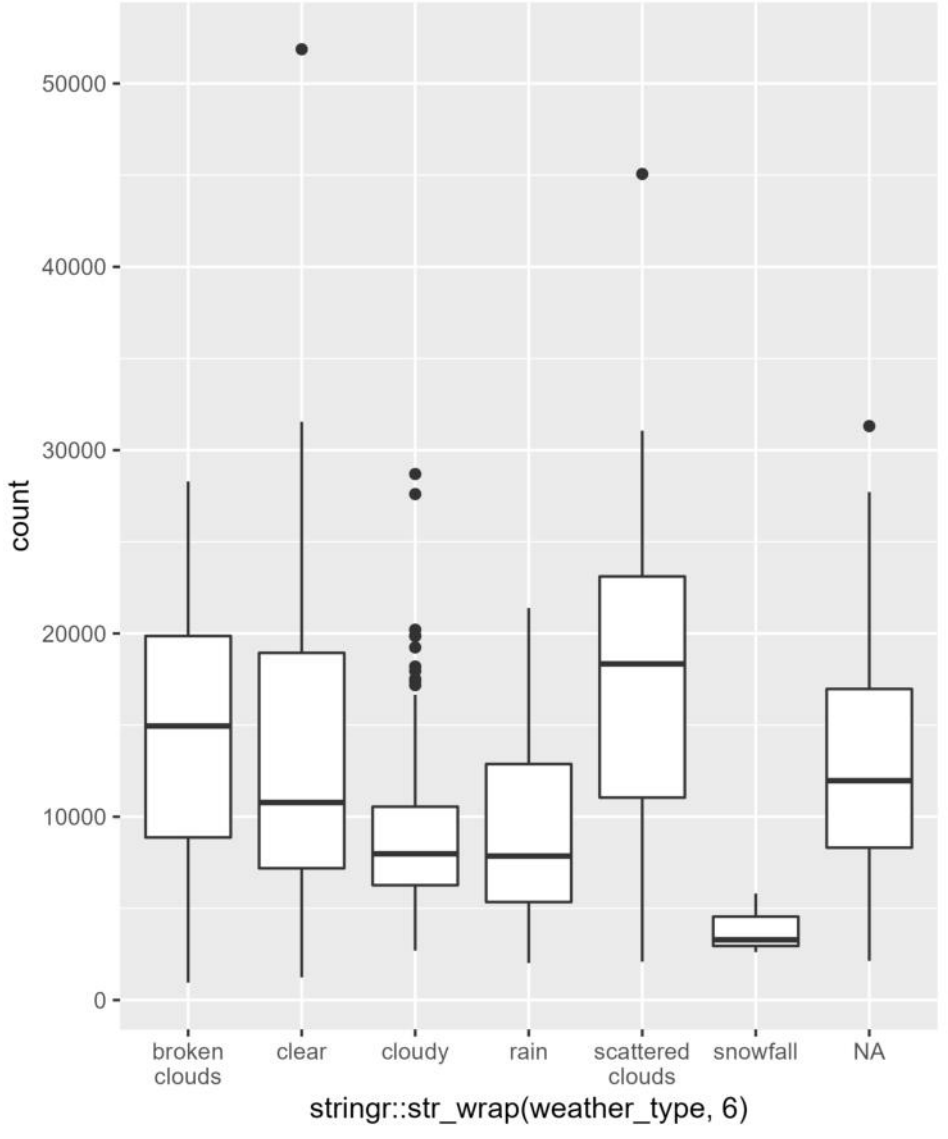
Avoid Overlapping Axis Labels

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



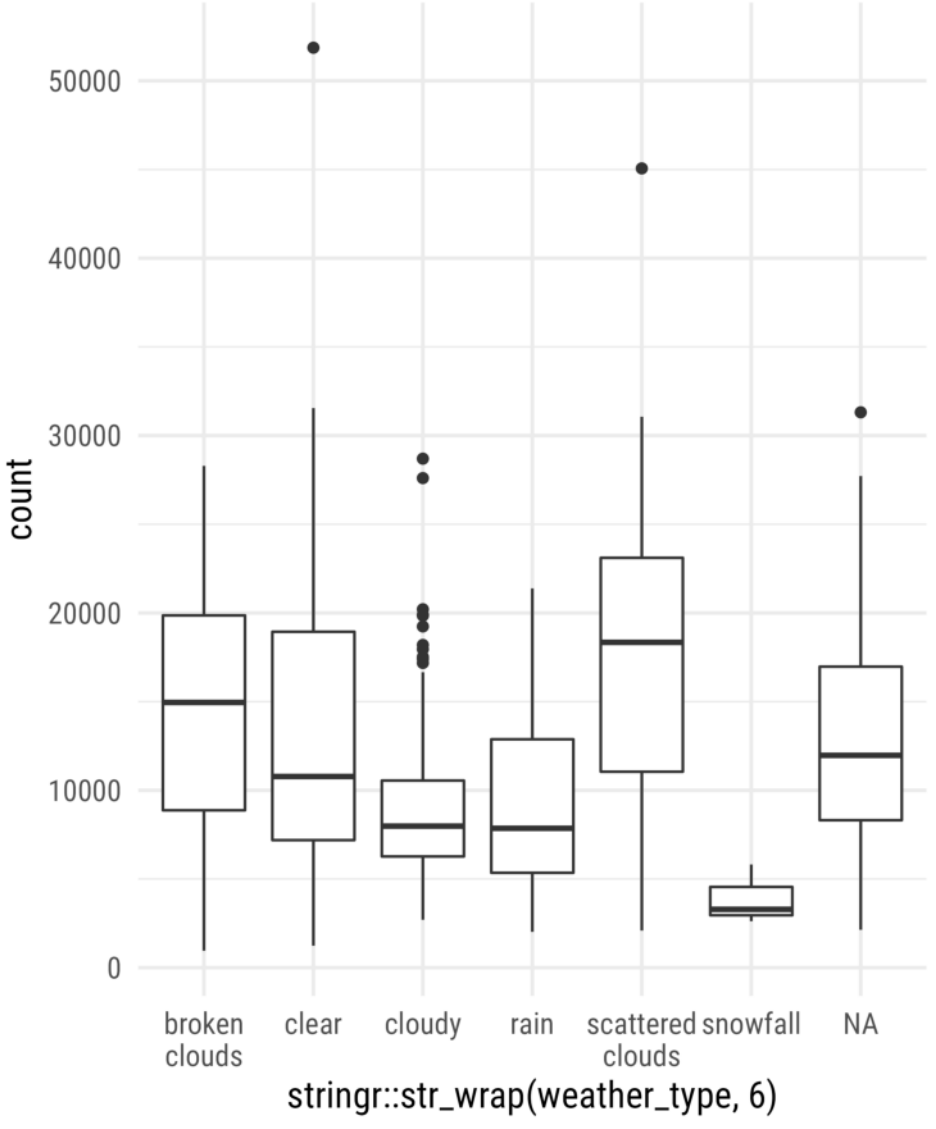
Avoid Overlapping Axis Labels

```
1  ggplot(
2      bikes,
3      aes(x = stringr::str_wrap(weather_type, 6)
4          y = count)
5    ) +
6    geom_boxplot()
```



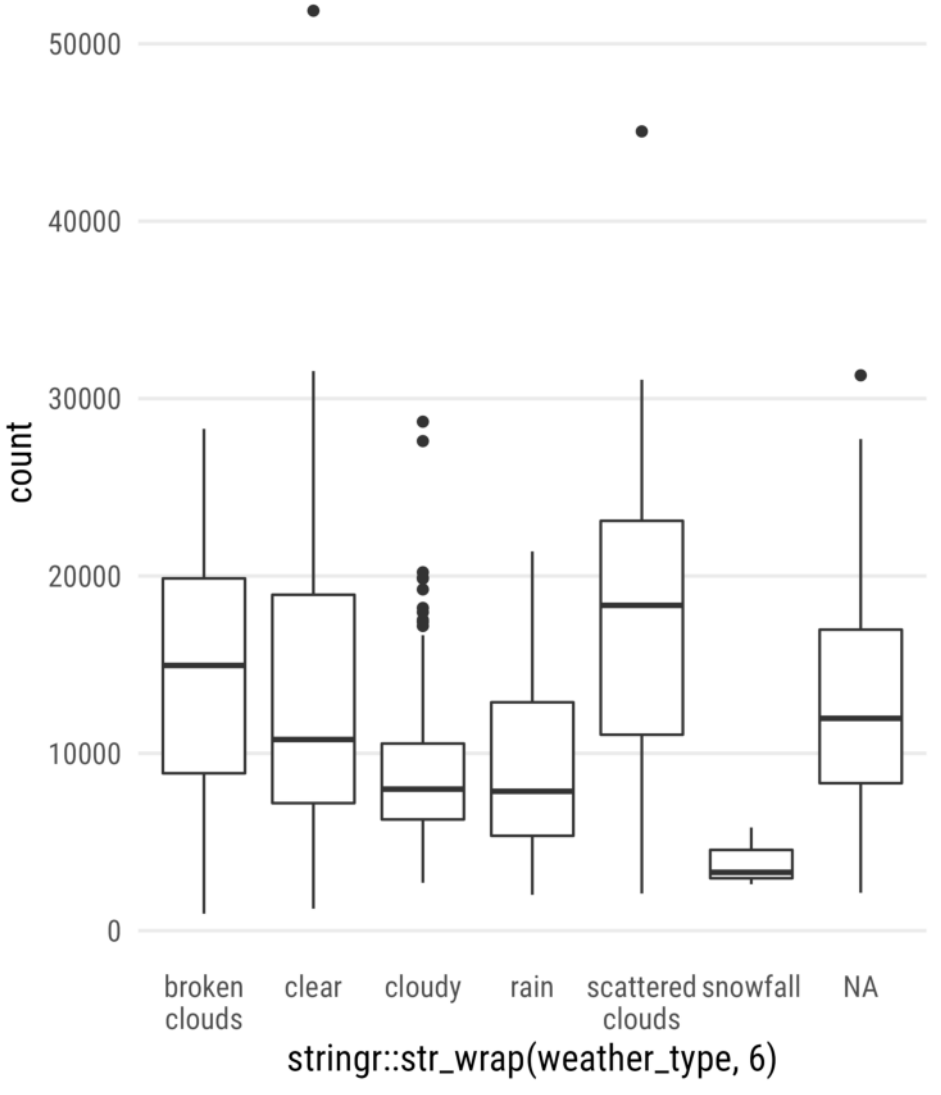
Apply a Theme

```
theme_set(theme_minimal(
  base_size = 14,
  base_family = "Roboto Condensed"
))

ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_boxplot()
```



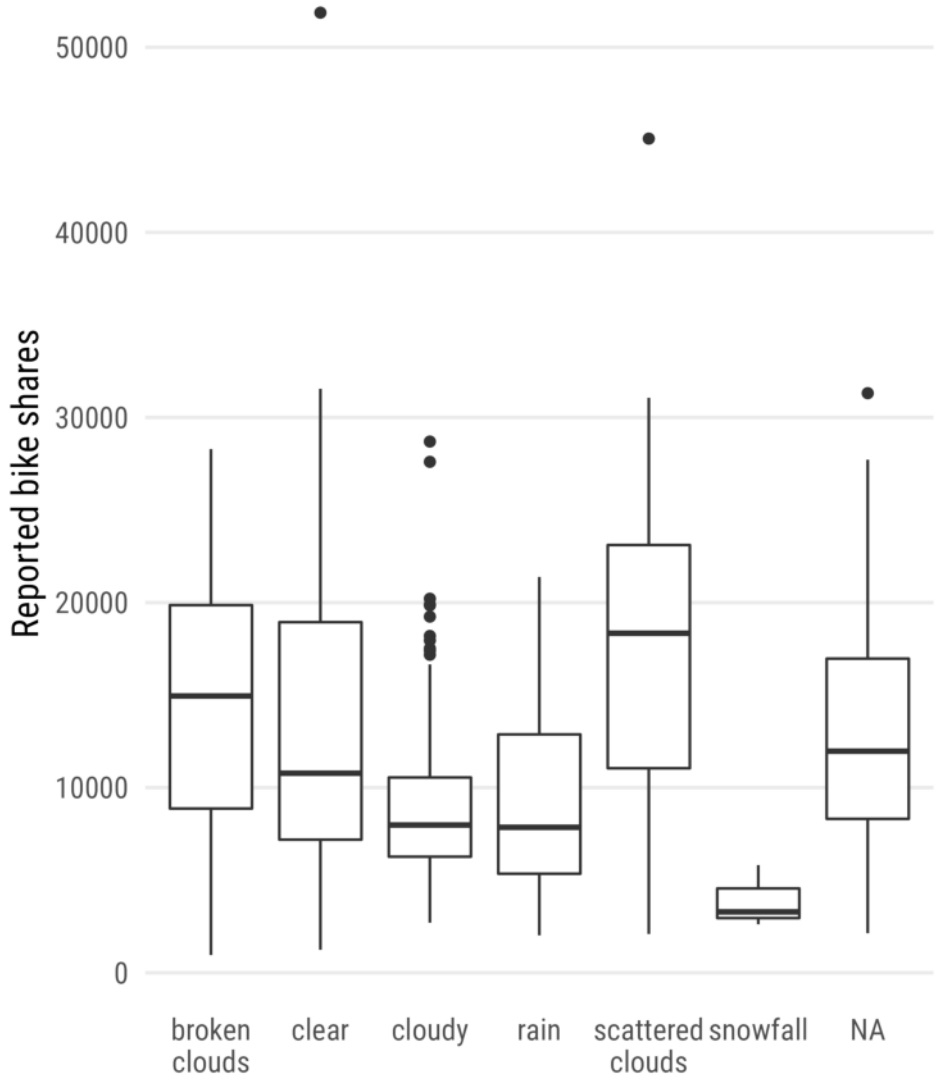
Customize the Theme

```
theme_set(theme_minimal(
  base_size = 14,
  base_family = "Roboto Condensed"
))

theme_update(
  panel.grid.major.x = element_blank(),
  panel.grid.minor = element_blank()
)

ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_boxplot()
```



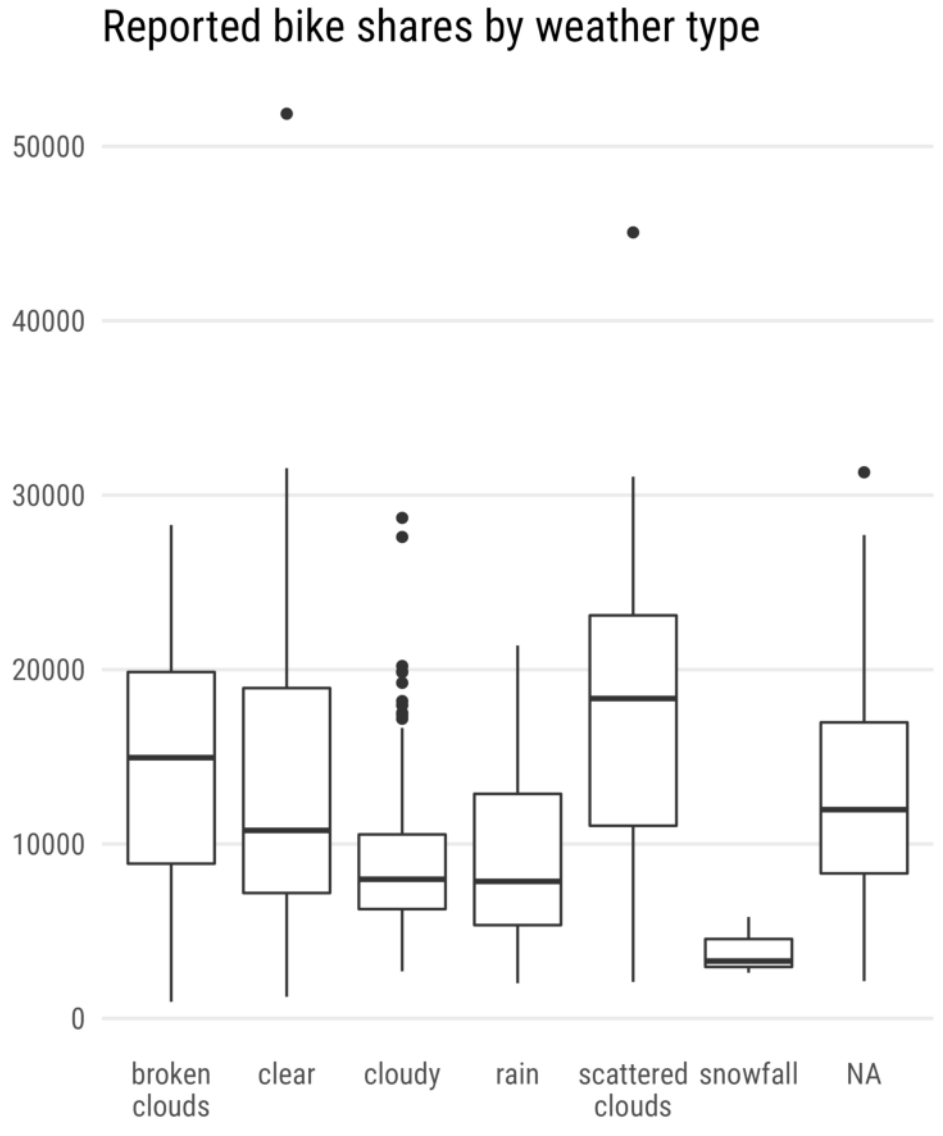
Add Meaningful Labels

```
ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_boxplot() +
  labs(x = NULL, y = "Reported bike shares")
```



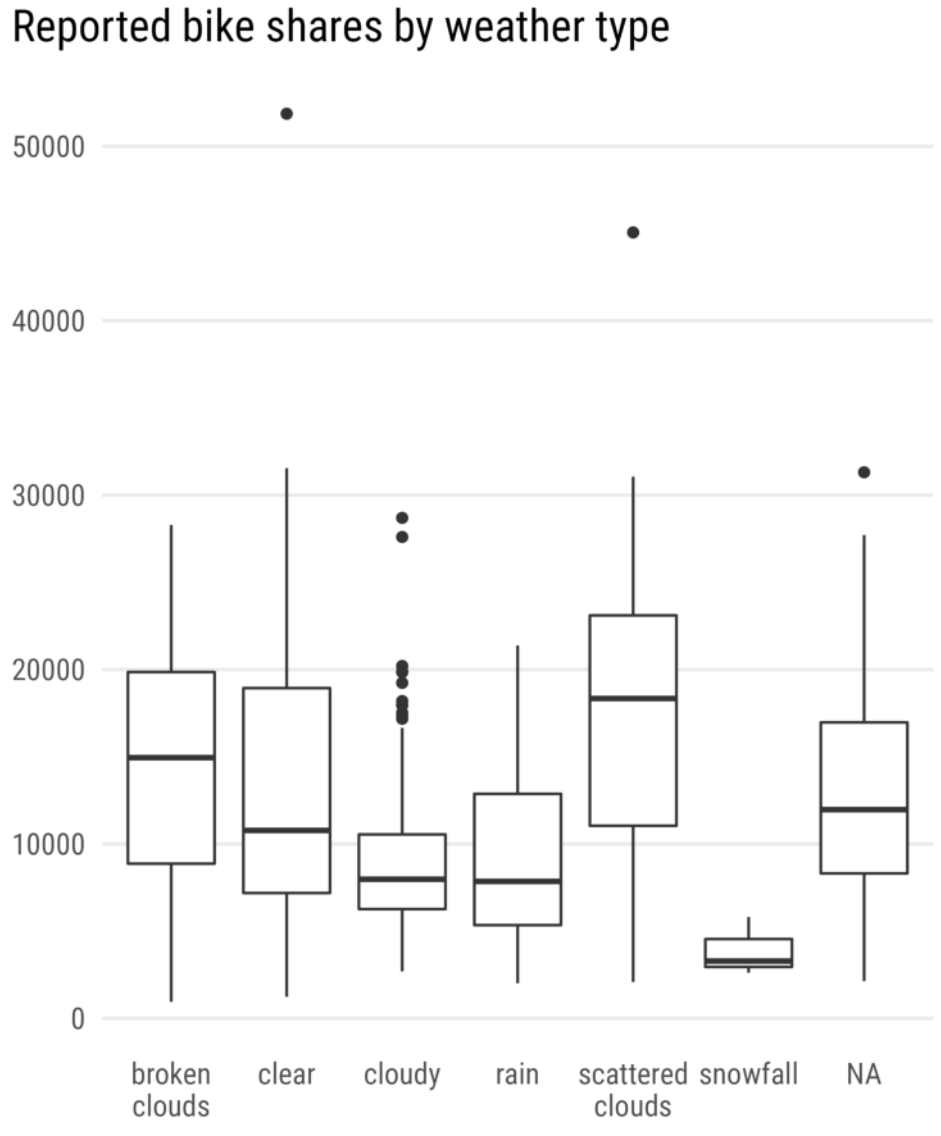
Add Meaningful Labels

```
ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_boxplot() +
  labs(
    x = NULL, y = NULL,
    title = "Reported bike shares by weather t
  )
```



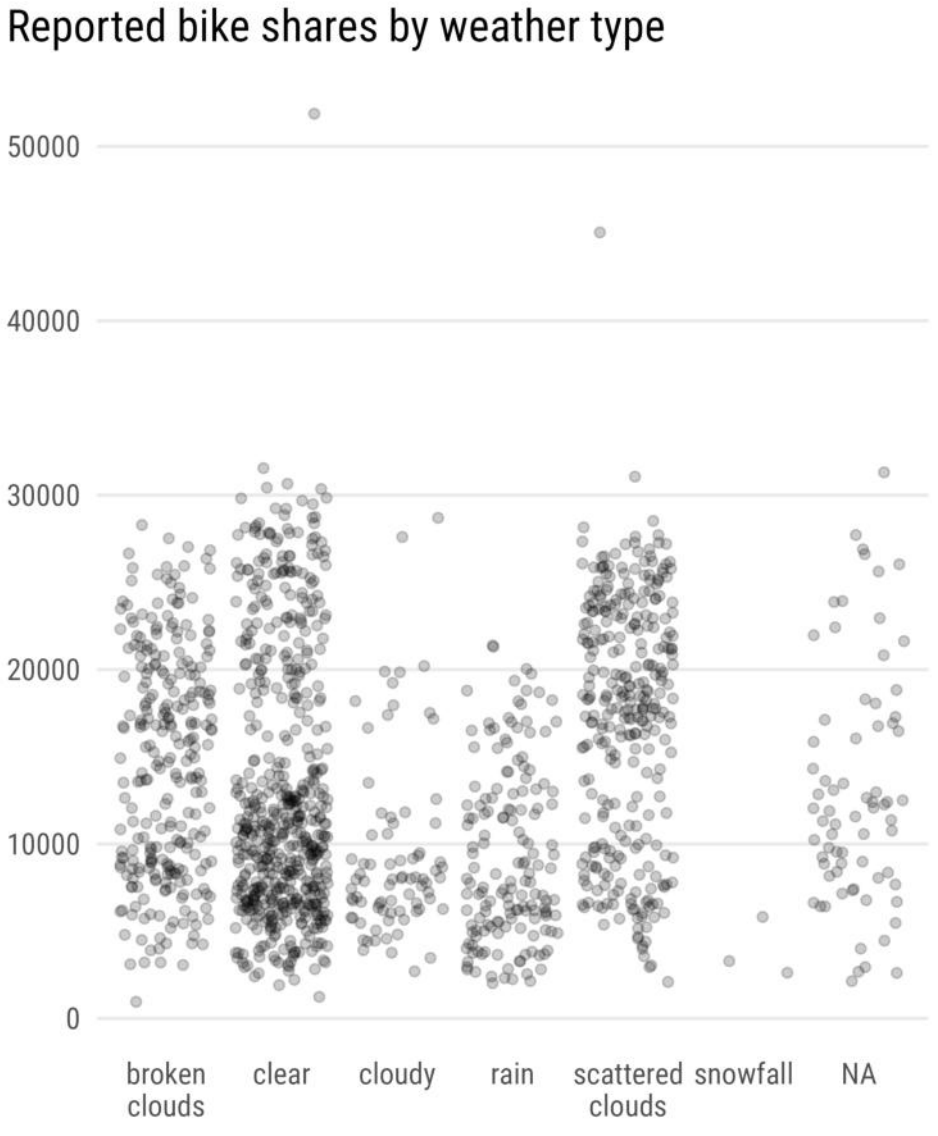
Add Meaningful Labels

```
theme_update(
  plot.title.position = "plot",
  axis.title = element_blank()
)

ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_boxplot() +
  ggtitle("Reported bike shares by weather typ
```



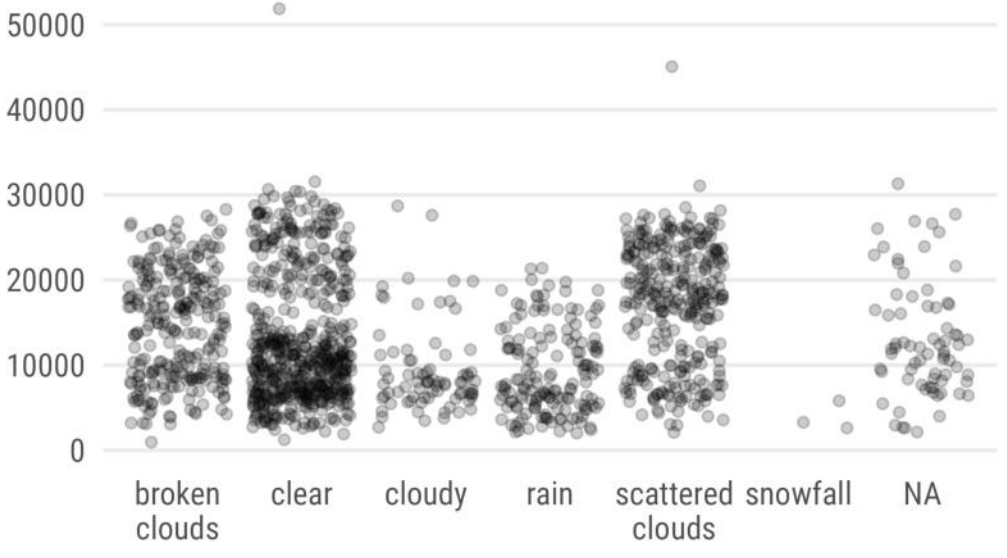
Jitter Strips of Counts per Weather Type

```
ggplot(
    bikes,
    aes(x = stringr::str_wrap(weather_type, 6)
        y = count)
  ) +
  geom_jitter(
    alpha = .2
  ) +
  ggtitle("Reported bike shares by weather typ
```

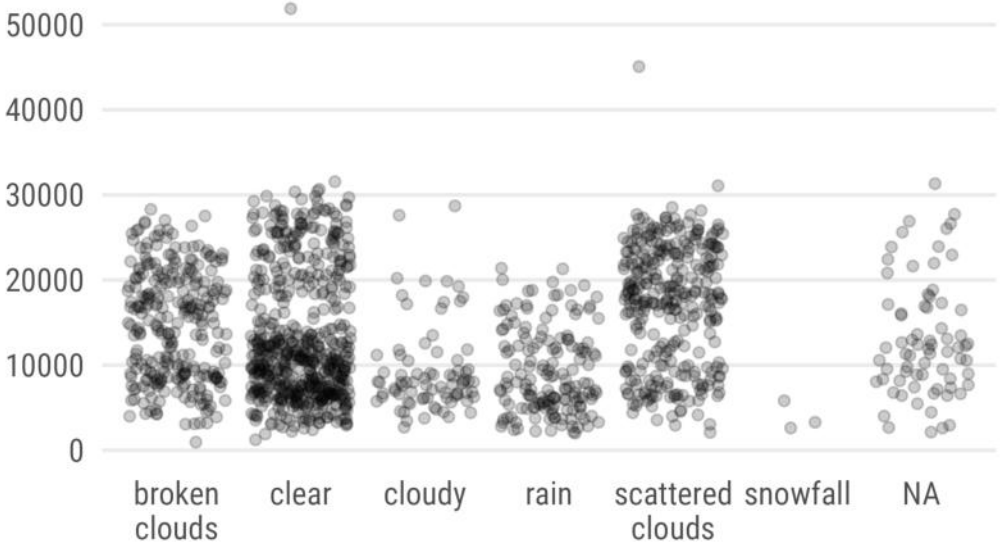


Jitter Strips of Counts per Weather Type

```r
ggplot(
    bikes,
    aes(x = str_wrap(weather_type, 6),
        y = count)
  ) +
  geom_point(
    position = "jitter",
    alpha = .2
  )
```

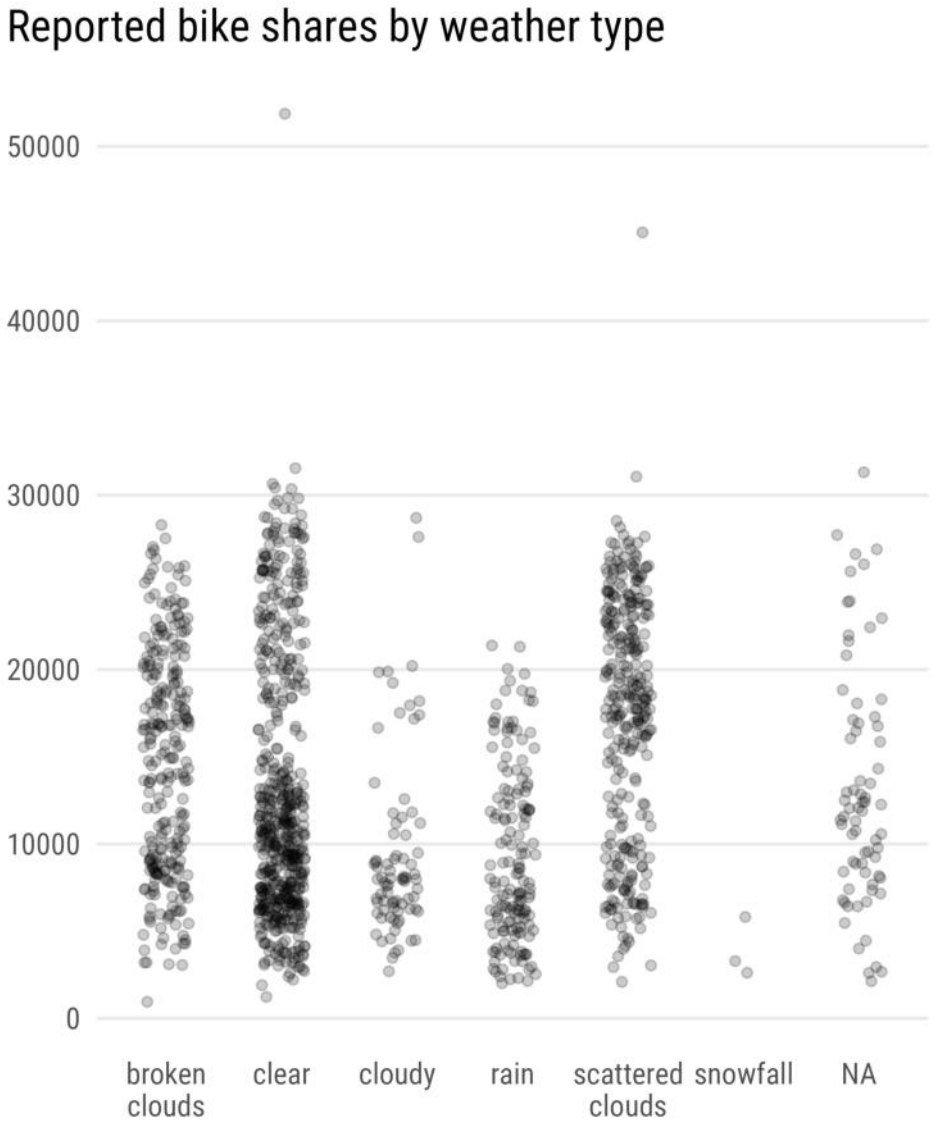



```r
ggplot(
    bikes,
    aes(x = str_wrap(weather_type, 6),
        y = count)
  ) +
  geom_point(
    position = position_jitter(),
    alpha = .2
  )
```



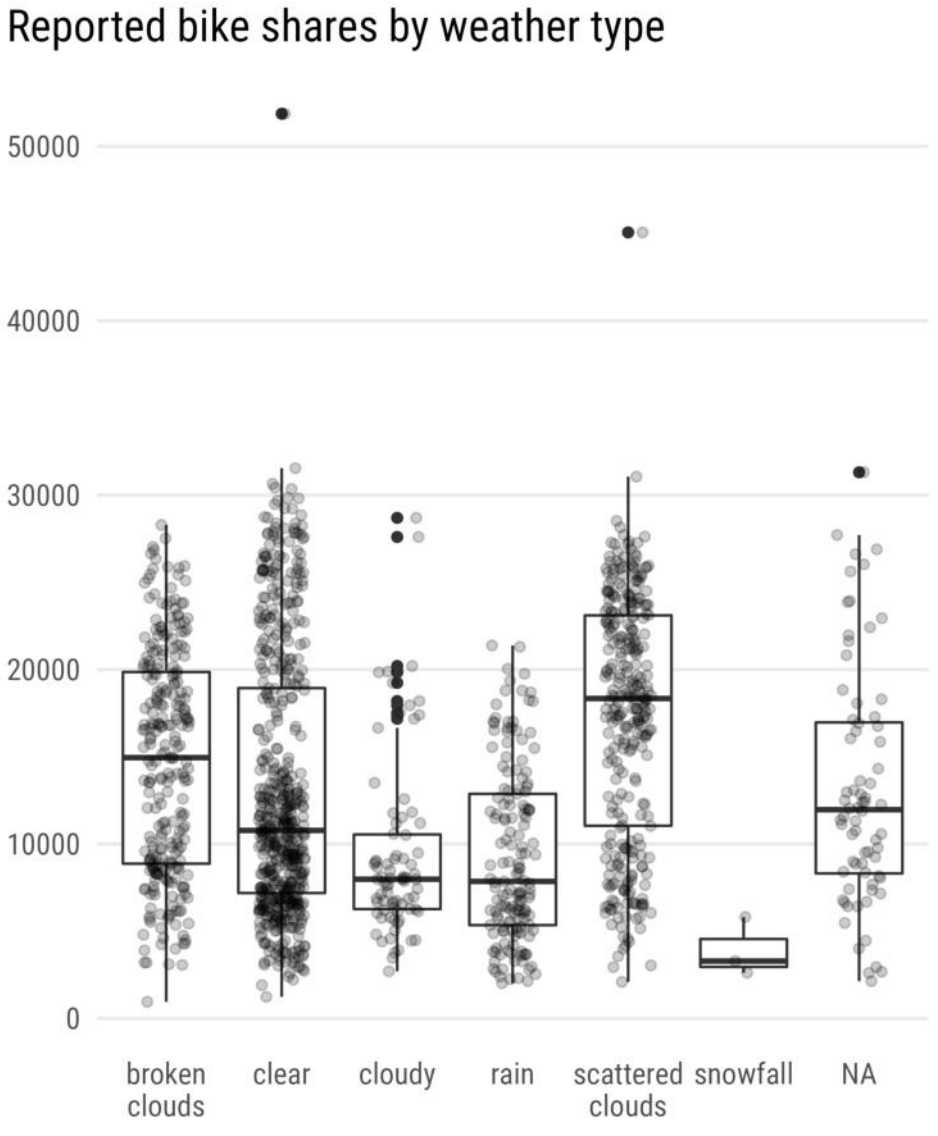
Jitter Strips of Counts vs. Weather Type

```
ggplot(
    bikes,
    aes(x = str_wrap(weather_type, 6),
        y = count)
  ) +
  geom_point(
    position = position_jitter(
      seed = 2022,
      width = .2,
      height = 0
    ),
    alpha = .2
  ) +
  ggtitle("Reported bike shares by weather typ
```



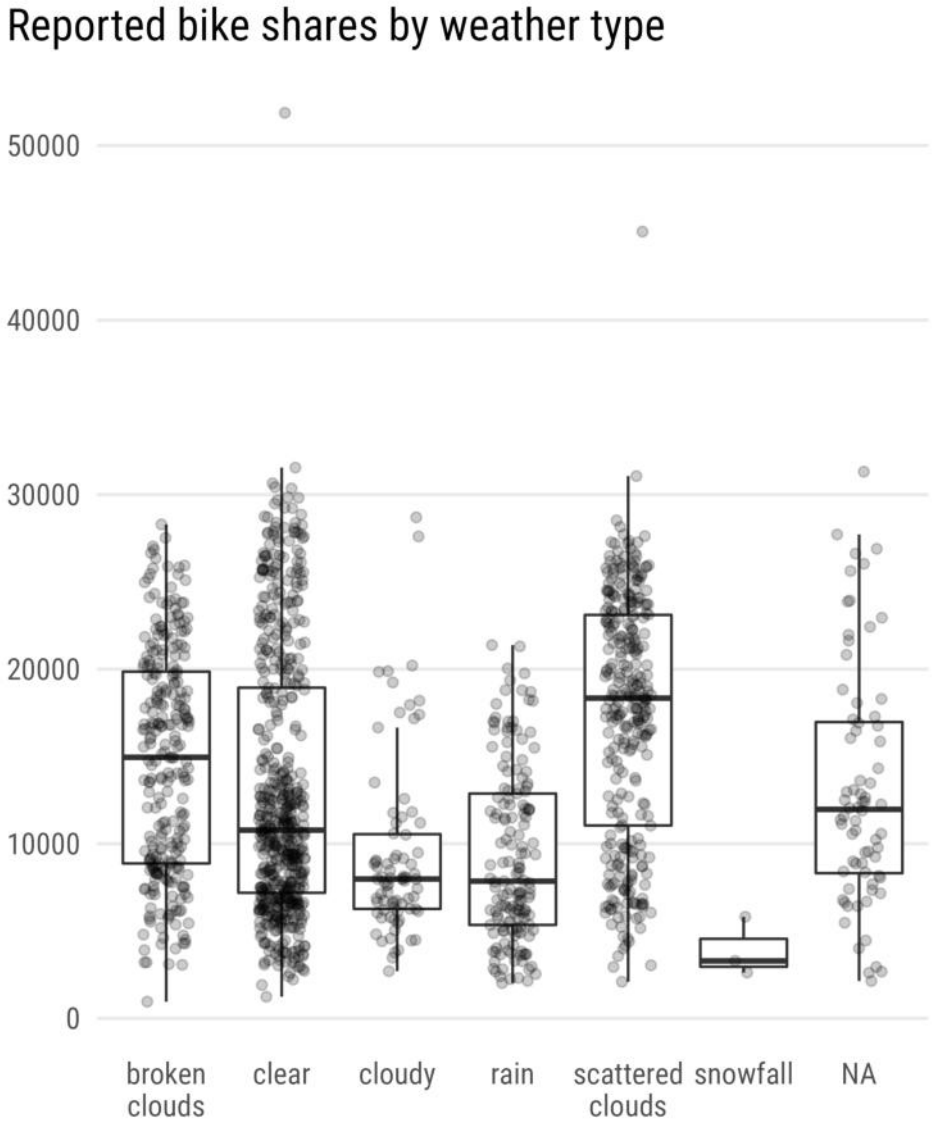
Boxplot + Jitter Strip Hybrid

```
ggplot(
    bikes,
    aes(x = str_wrap(weather_type, 6),
        y = count)
  ) +
  geom_boxplot() +
  geom_point(
    position = position_jitter(
      seed = 2022,
      width = .2,
      height = 0
    ),
    alpha = .2
  ) +
  ggtitle("Reported bike shares by weather typ
```



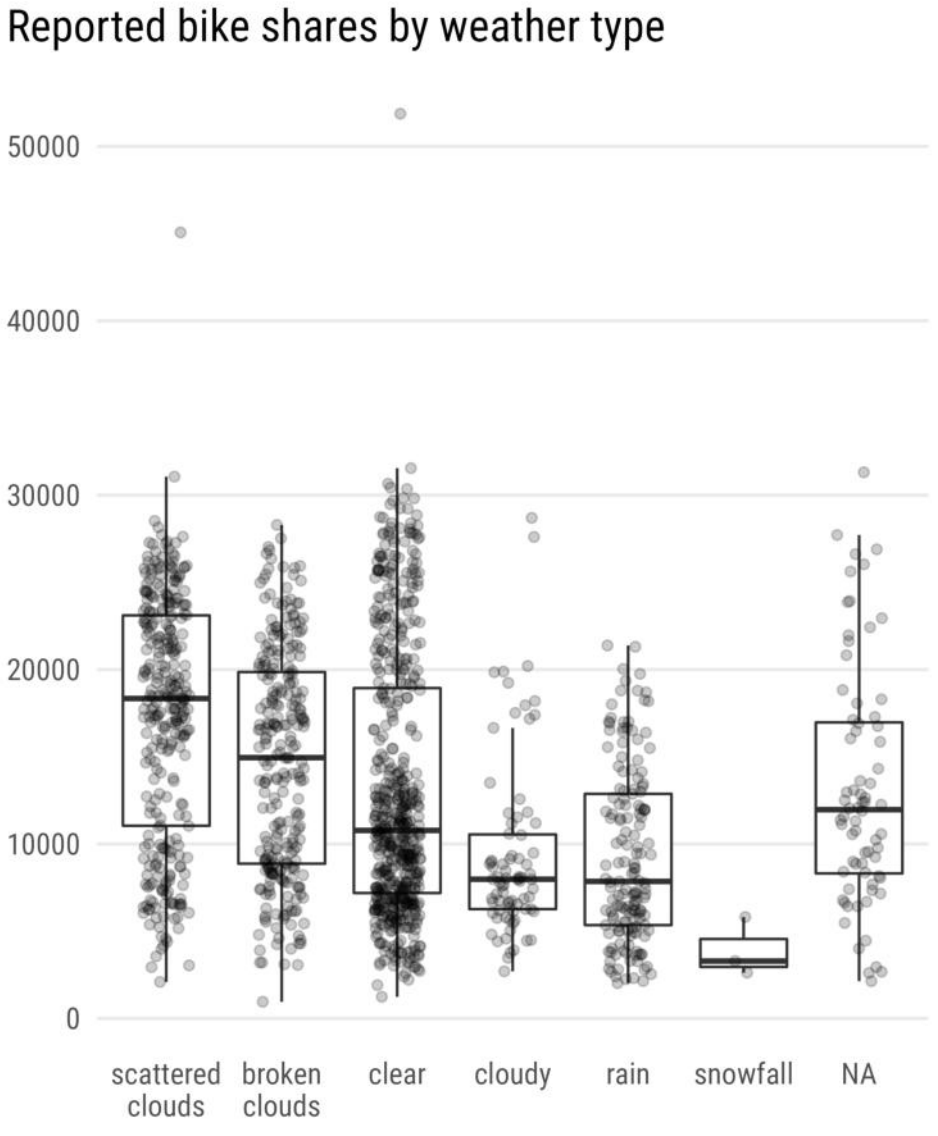
Boxplot + Jitter Strip Hybrid

```
ggplot(
    bikes,
    aes(x = str_wrap(weather_type, 6),
        y = count)
  ) +
  geom_boxplot(
     outlier.shape = NA
     # outlier.color = "transparent"
     # outlier.alpha = 0
  ) +
  geom_point(
    position = position_jitter(
      seed = 2022,
      width = .2,
      height = 0
    ),
    alpha = .2
  ) +
  ggtitle("Reported bike shares by weather typ
```



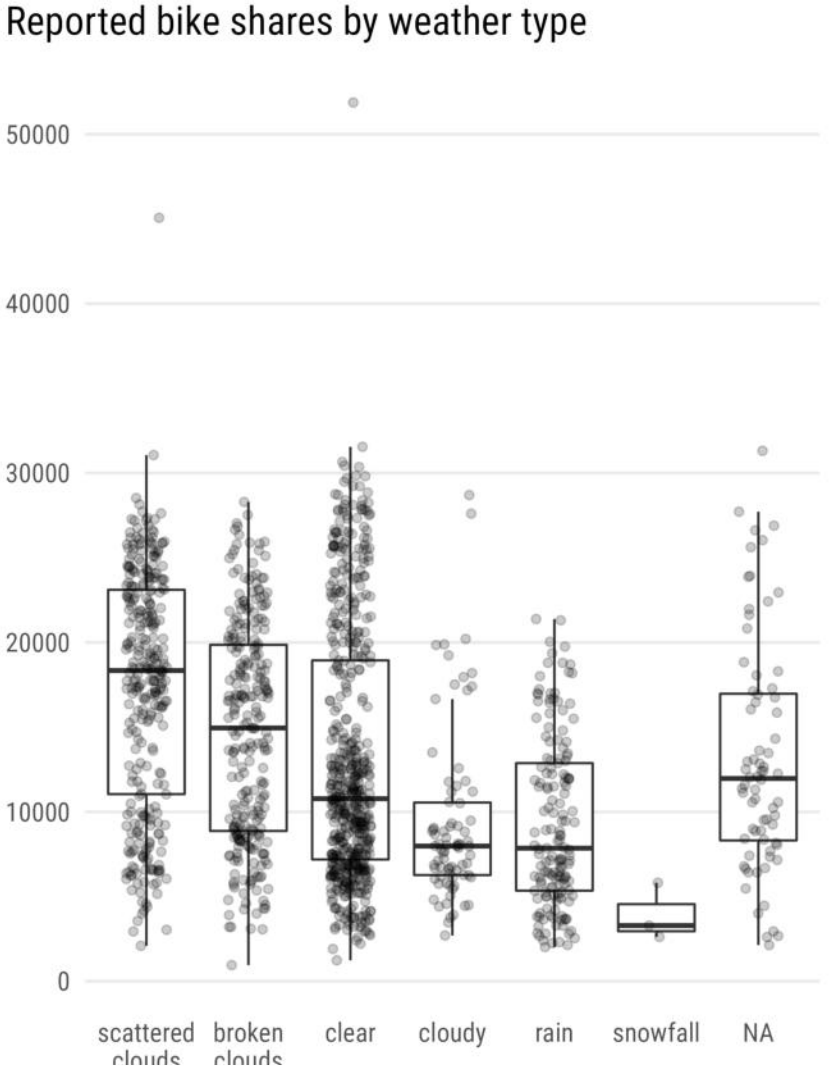
Bonus: Sort Weather Types

```
ggplot(
    bikes,
    aes(
      x = forcats::fct_reorder(
        str_wrap(weather_type, 6), -count
      ),
      y = count)
  ) +
  geom_boxplot(
     outlier.shape = NA
     # outlier.color = "transparent"
     # outlier.alpha = 0
  ) +
  geom_point(
    position = position_jitter(
      seed = 2022,
      width = .2,
      height = 0
    ),
    alpha = .2
  ) +
  ggtitle("Reported bike shares by weather typ
```



Save the Plot

```
1 ggsave(here::here("exercises", "plots", "02_concepts_pt1_ex2.pdf"),
2        width = 5, height = 6.5, device = cairo_pdf)
```



Alternative Chart Types to Visualize Distributions

Let's Update our Data Set

```
 1 bikes <-
 2   bikes %>%
 3   mutate(
 4     weather_type_fct = stringr::str_wrap(
 5       weather_type, 6
 6     ),
 7     weather_type_fct = forcats::fct_reorder(
 8       weather_type_fct, -count
 9     )
10   )
11
12 levels(bikes$weather_type_fct)
```

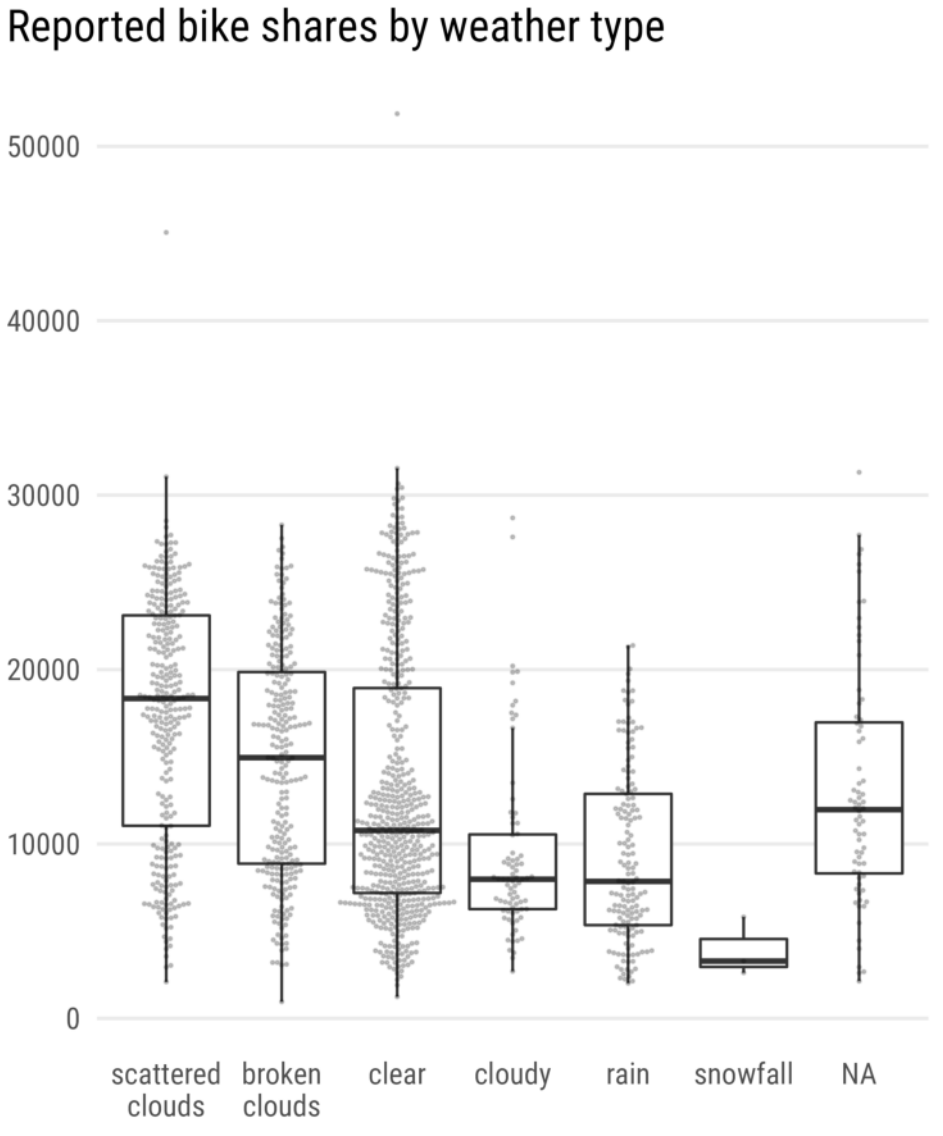
```
[1] "scattered\nclouds" "broken\nclouds"    "clear"
[4] "cloudy"            "rain"              "snowfall"
```

Let's Store Our ggplot Setup

```
g <-
  ggplot(
    bikes,
    aes(x = weather_type_fct,
        y = count)
  ) +
  ggtitle(
    "Reported bike shares by weather type"
  )
```

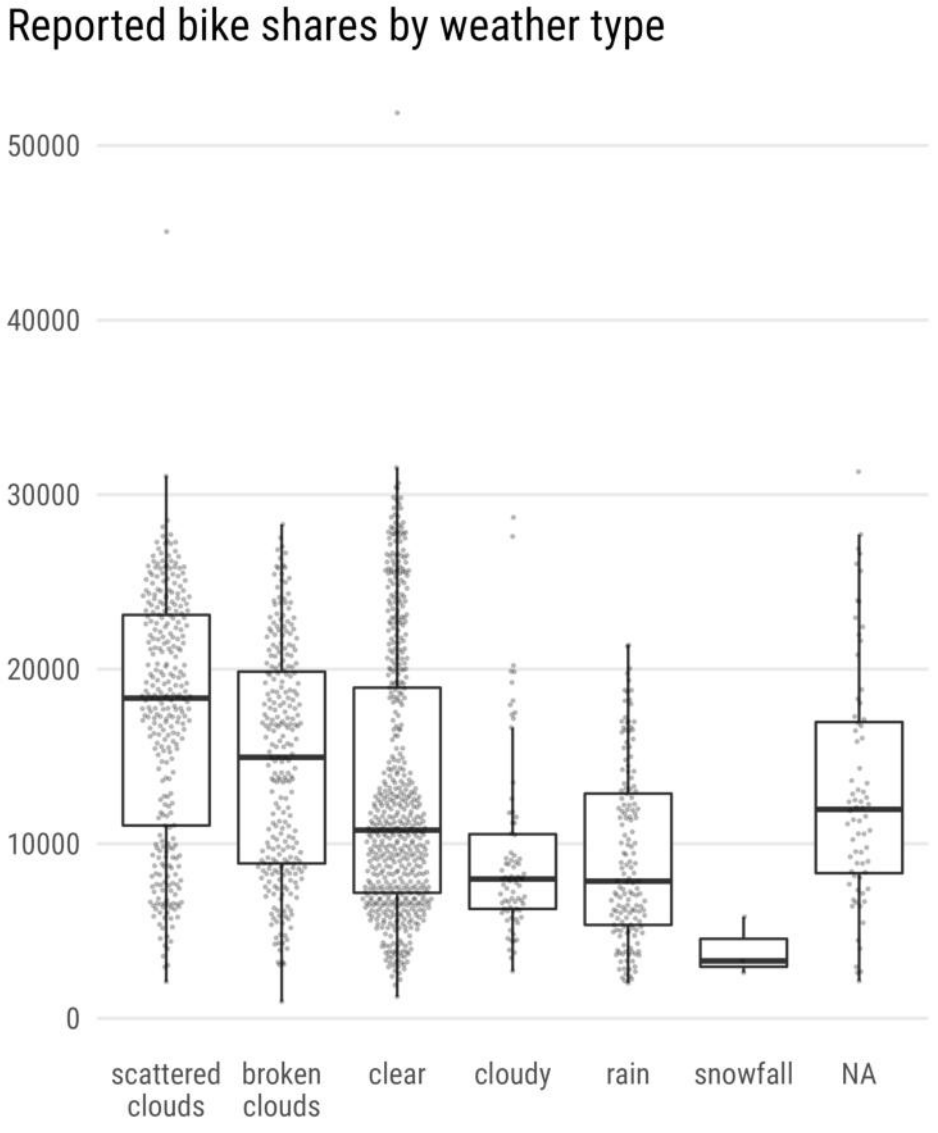
Beeswarm Plots with {ggbeeswarm}

```
g +
  geom_boxplot(
     outlier.shape = NA
  ) +
  ggbeeswarm::geom_beeswarm(
    size = .3,
    alpha = .2,
    cex = .6
  )
```



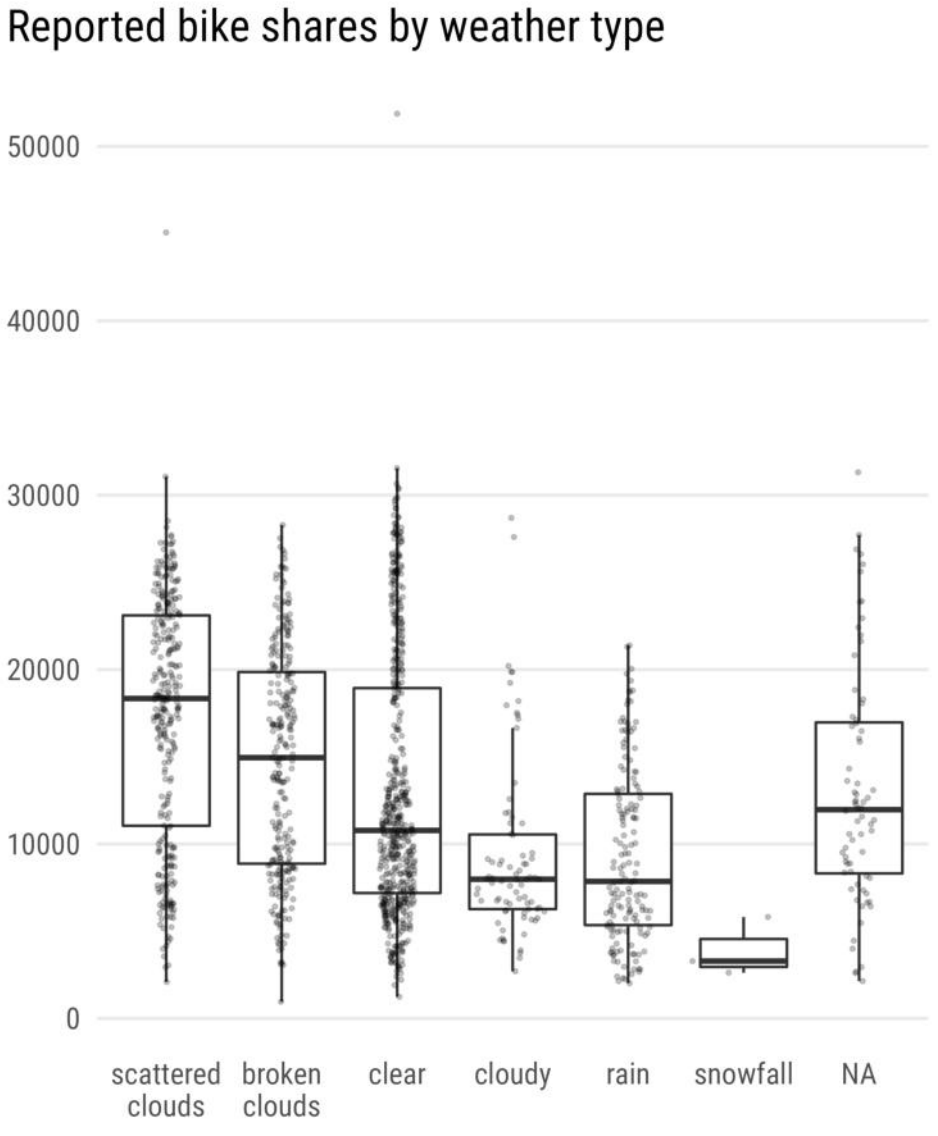
Beeswarm Plots with {ggbeeswarm}

```
g +
  geom_boxplot(
     outlier.shape = NA
  ) +
  ggbeeswarm::geom_quasirandom(
    size = .3,
    alpha = .2,
    width = .3,
    varwidth = TRUE
  )
```



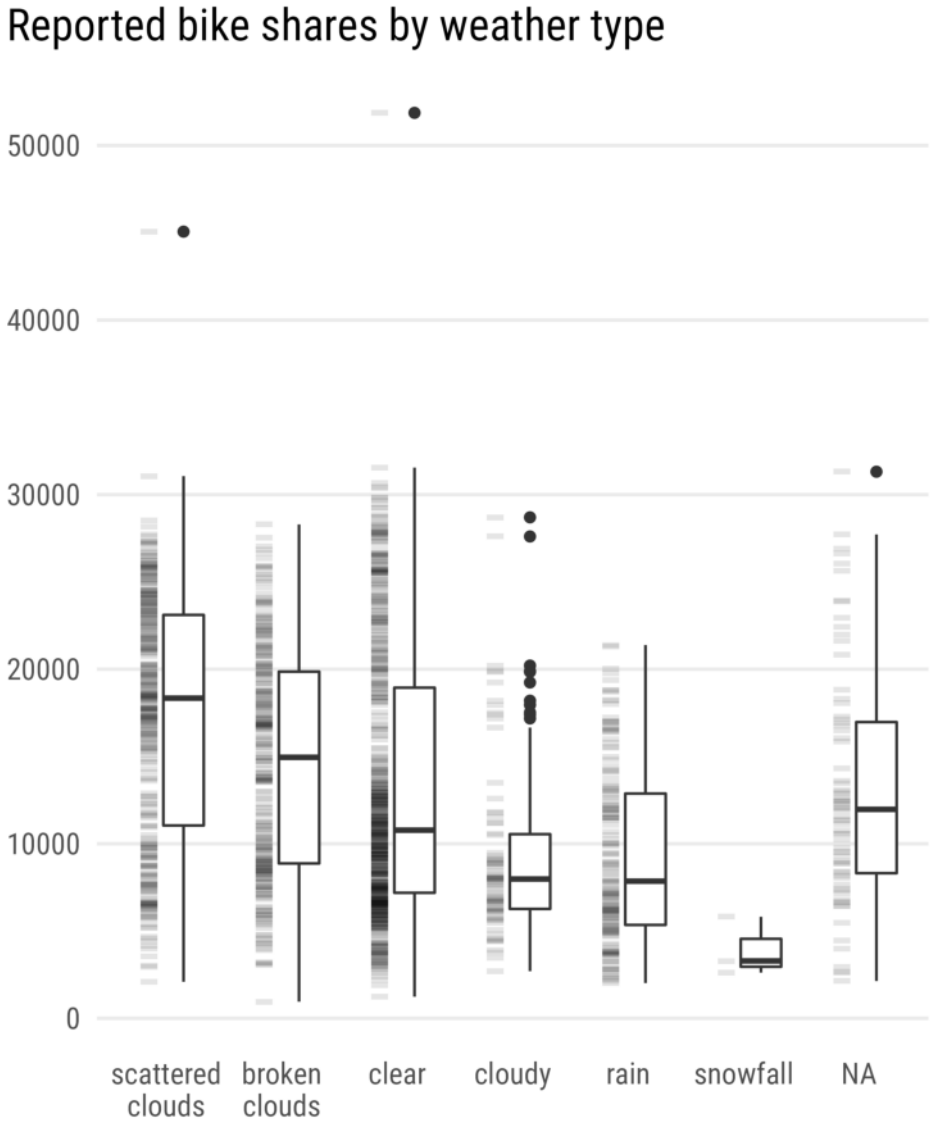
Sina Plots with {ggforce}

```
g +
  geom_boxplot(
     outlier.shape = NA
  ) +
  ggforce::geom_sina(
    size = .5,
    alpha = .2,
    maxwidth = 1.2
  )
```



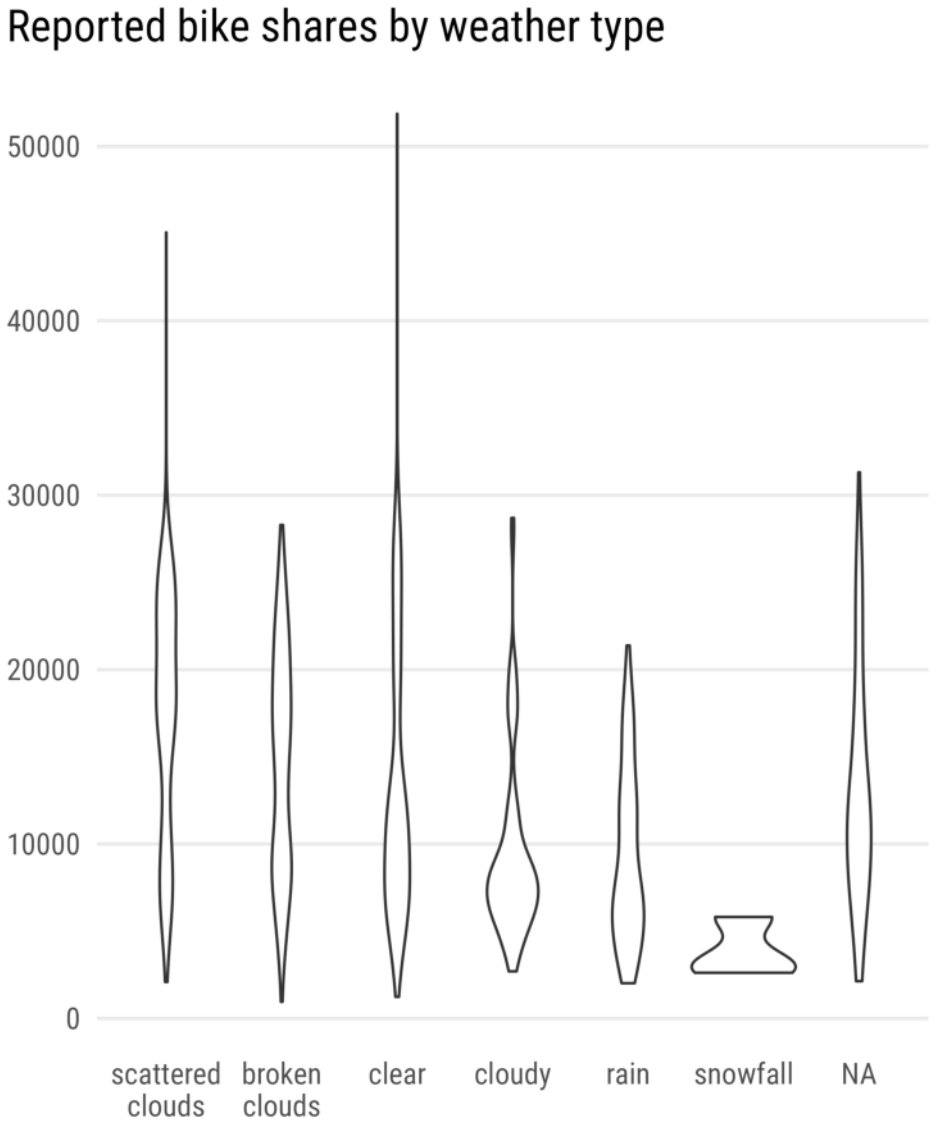
Barcode Strips

```
g +
  geom_boxplot(
     position = position_nudge(x = .15),
     width = .35
  ) +
  geom_point(
    shape = "-",
    size = 8,
    alpha = .1,
    position = position_nudge(x = -.15)
  )
```



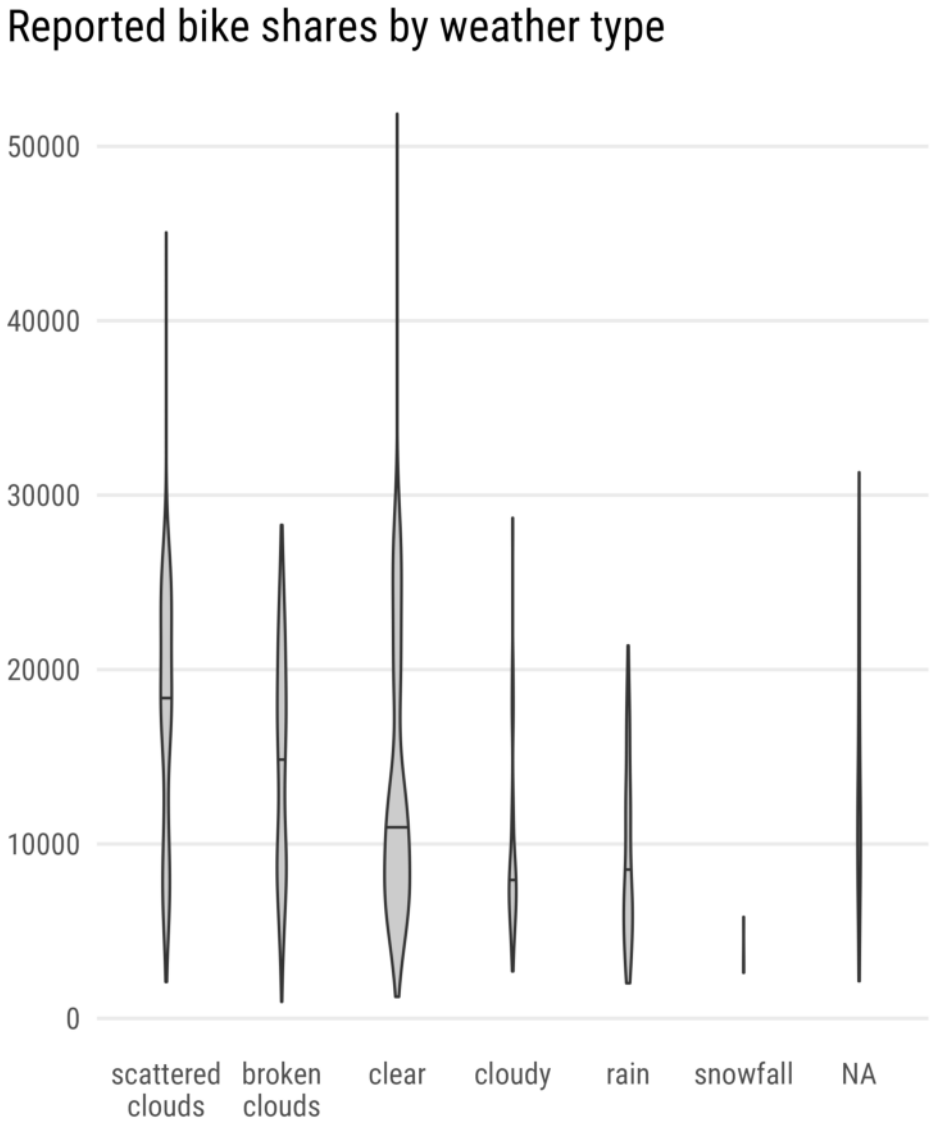
Violin Plots

```
1 g +
2   geom_violin()
```



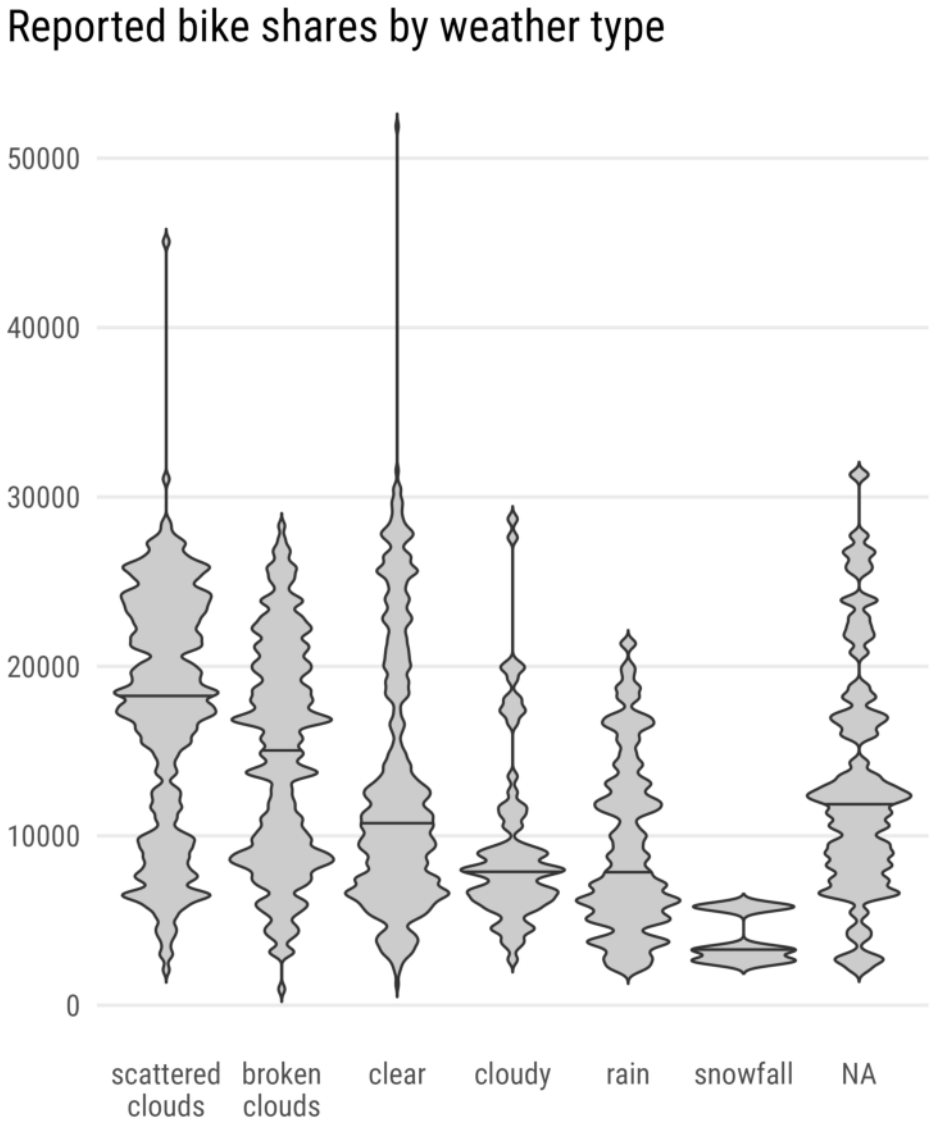
Violin Plots

```
g +
  geom_violin(
    scale = "count",
    draw_quantiles = c(.5),
    fill = "grey80"
  )
```



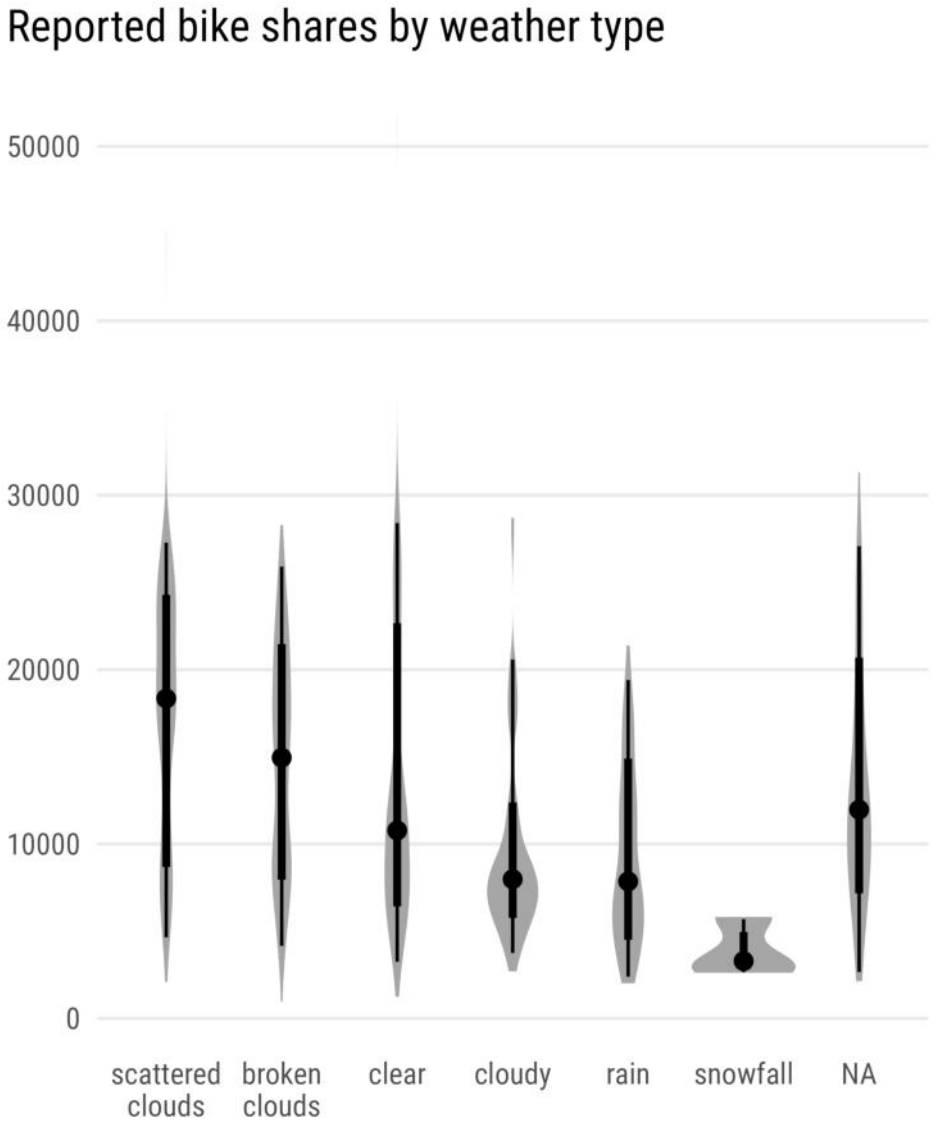
Violin Plots

```
g +
  geom_violin(
    scale = "width",
    draw_quantiles = c(.5),
    trim = FALSE,
    bw = 250,
    fill = "grey80"
  )
```



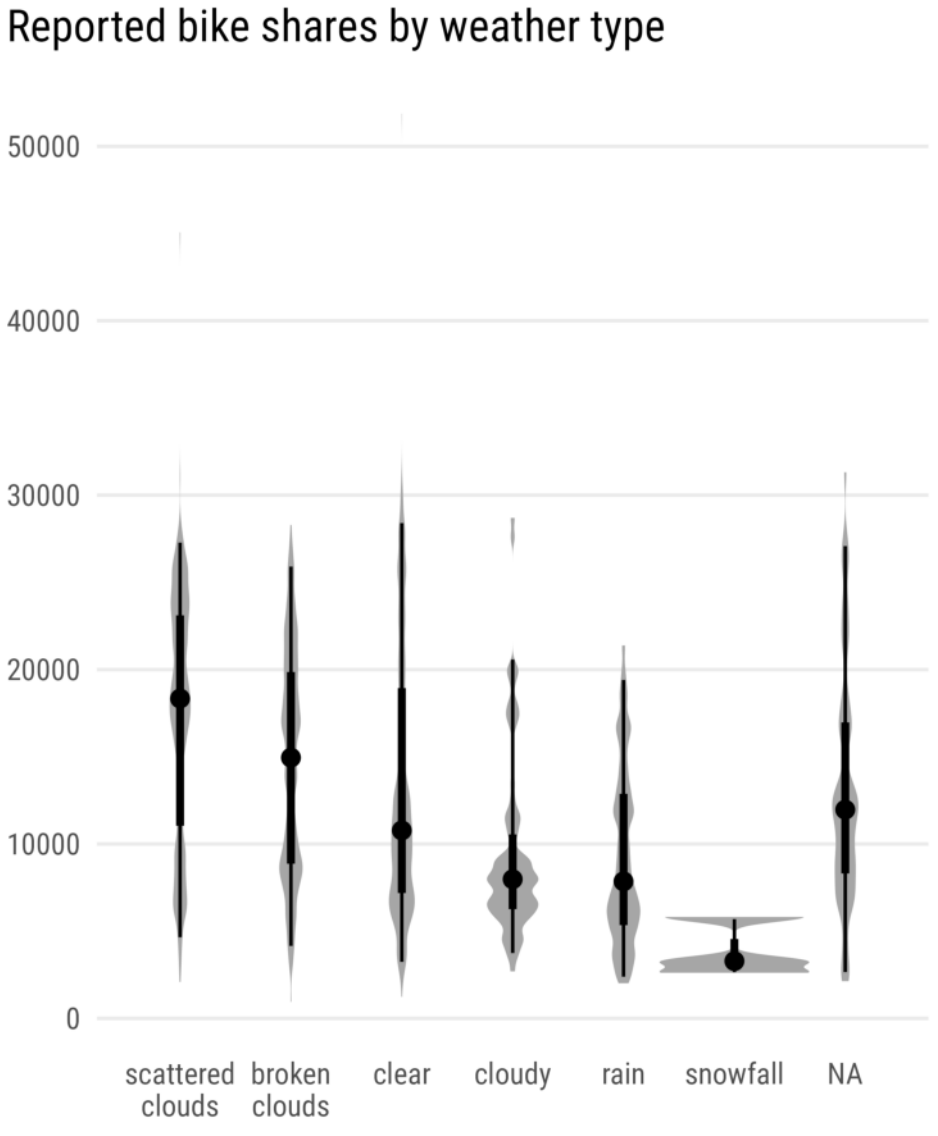
Violin Plots with {ggdist}

```
1 g +
2   ggdist::stat_eye()
```



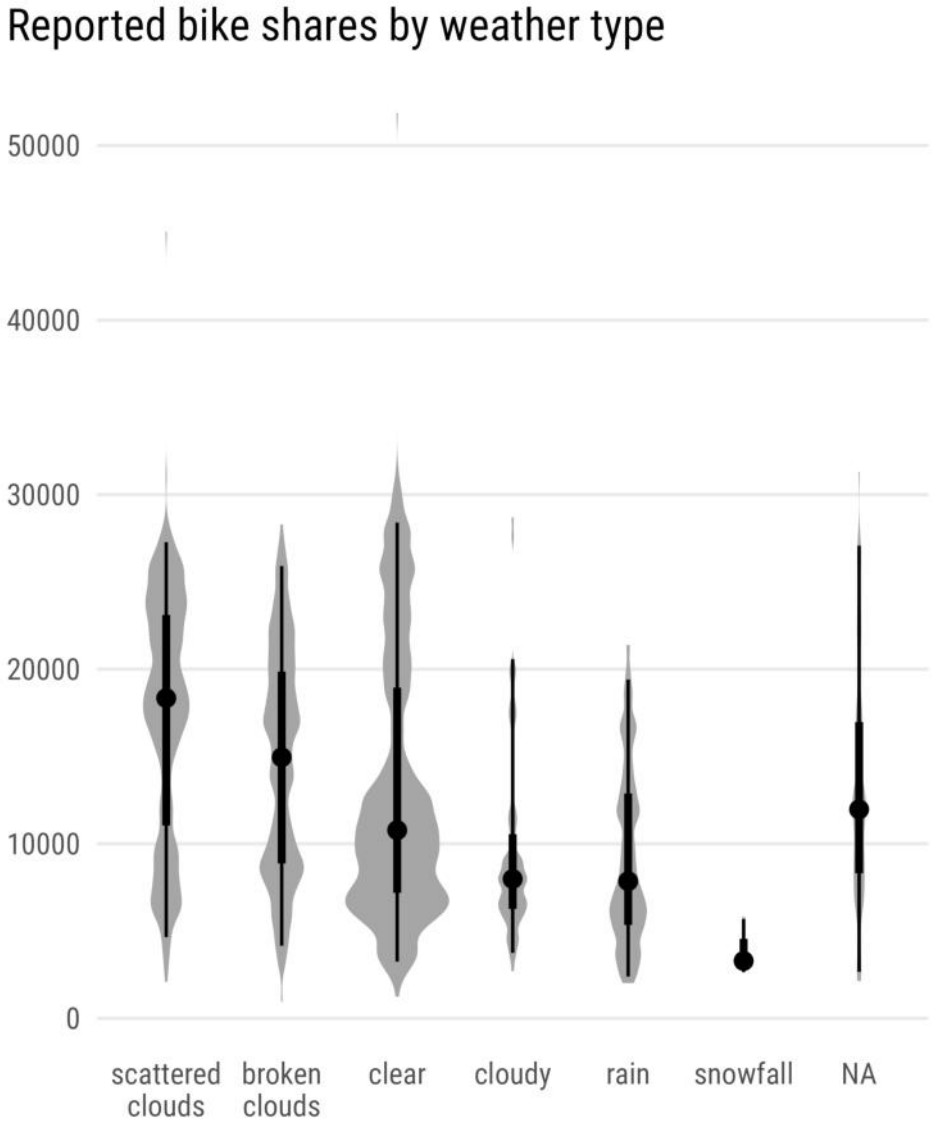
Violin Plots with {ggdist}

```
g +
  ggdist::stat_eye(
    .width = c(0.5, 0.95),
    width = 1.5,
    adjust = .33
  )
```



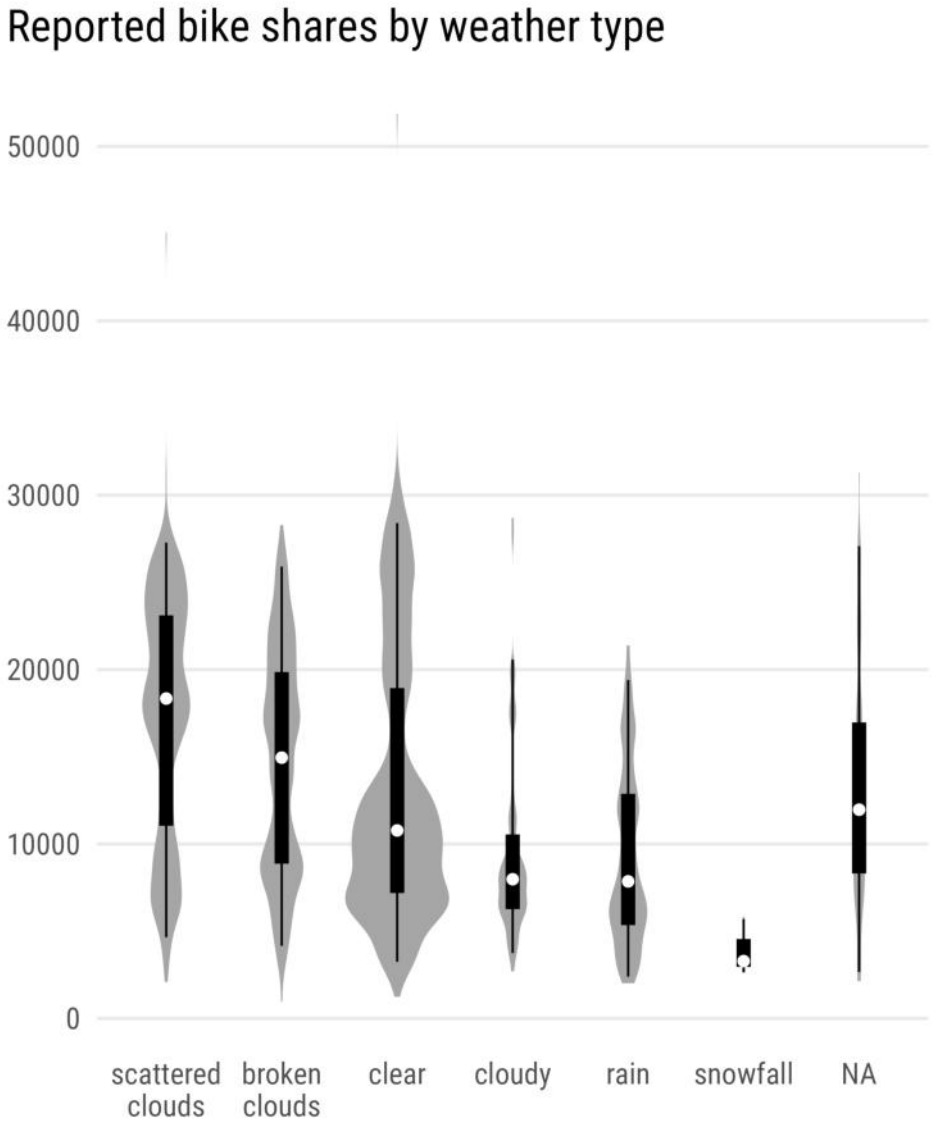
Violin Plots with {ggdist}

```
g +
  ggdist::stat_eye(
    aes(thickness = stat(f*n)),
    .width = c(0.5, 0.95),
    adjust = .33
  )
```



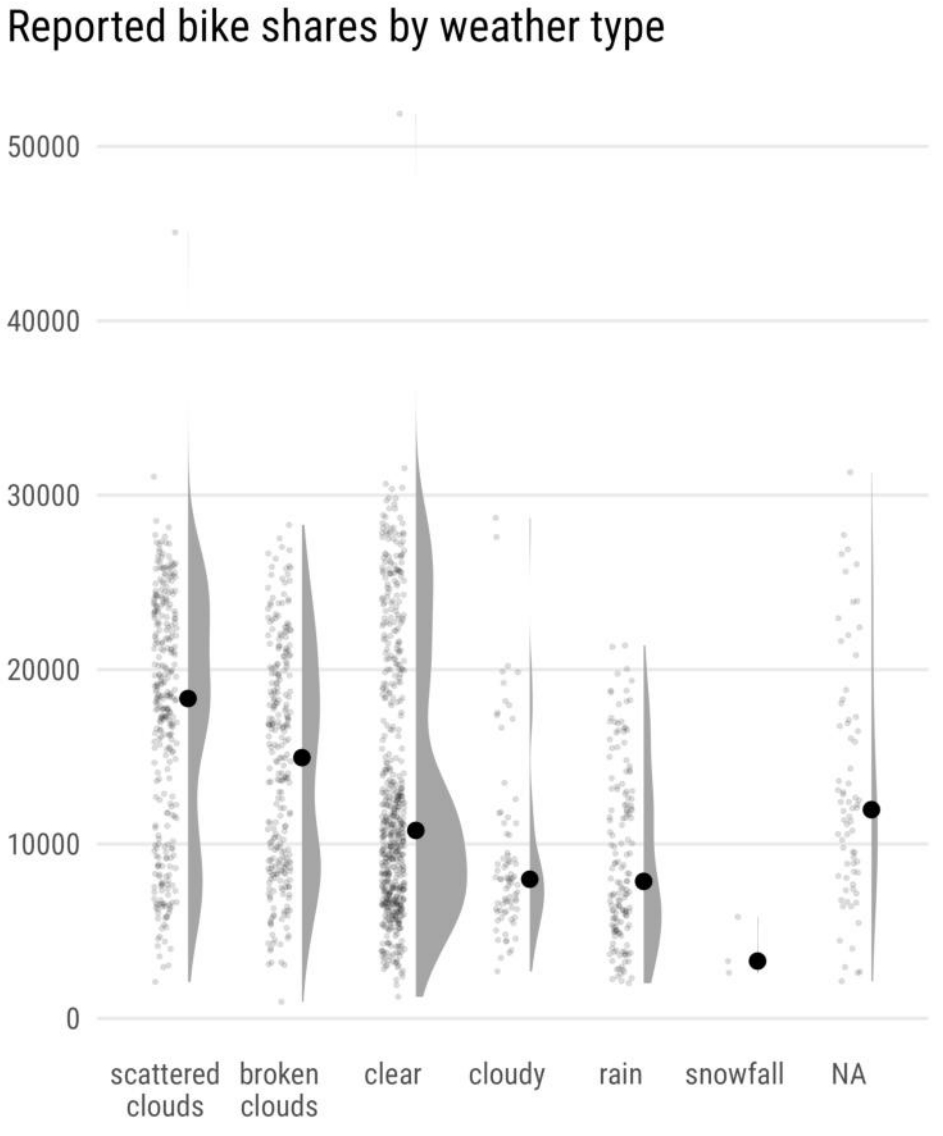
Violin Plots with {ggdist}

```
g +
  ggdist::stat_eye(
    aes(thickness = stat(f*n)),
    .width = c(0.5, 0.95),
    adjust = .5,
    interval_size_range = c(.4, 2.5),
    point_size = 1.2,
    point_color = "white"
  )
```



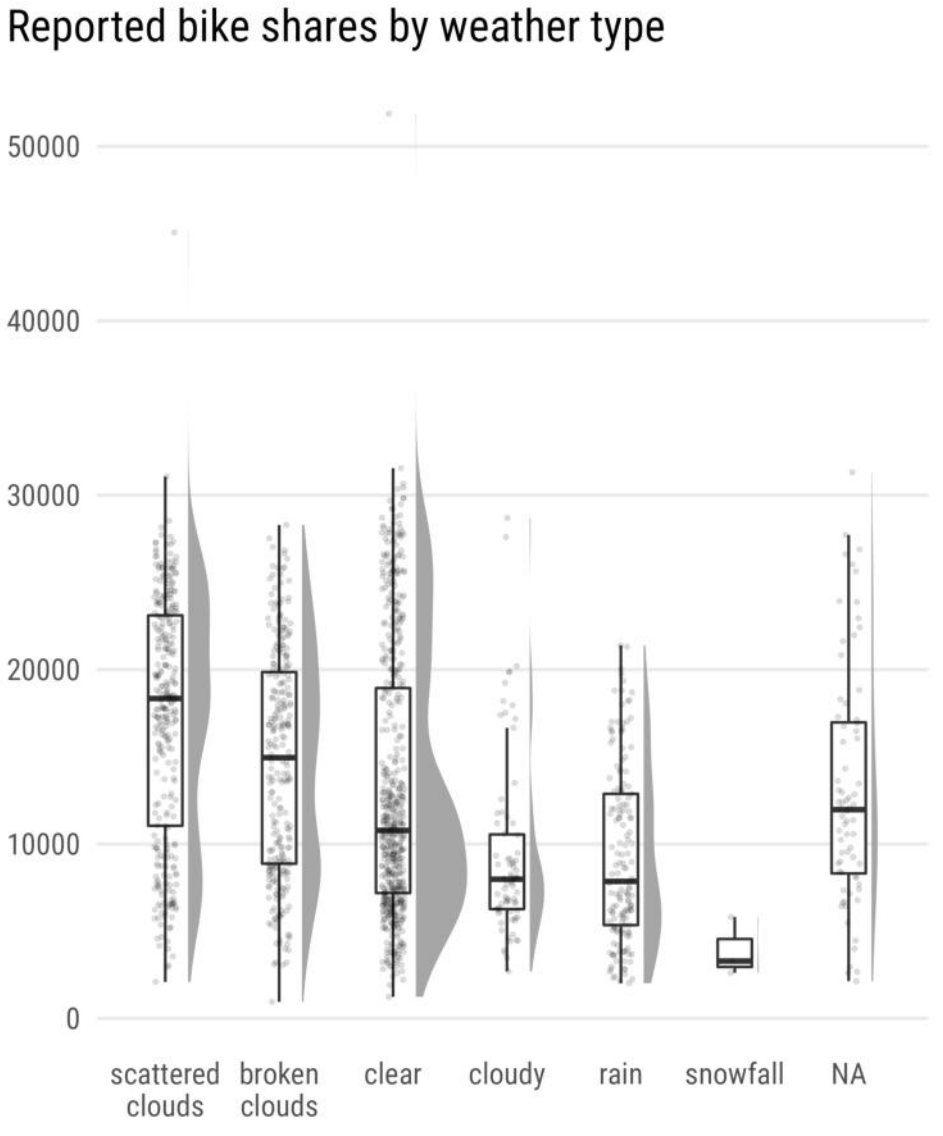
Raincloud Plots with {ggdist}

```
g +
  ggdist::stat_halfeye(
    aes(thickness = stat(f*n)),
    .width = 0,
    width = .5,
    position = position_nudge(x = .2)
  ) +
  geom_jitter(
    width = .1,
    size = .5,
    alpha = .1
  )
```



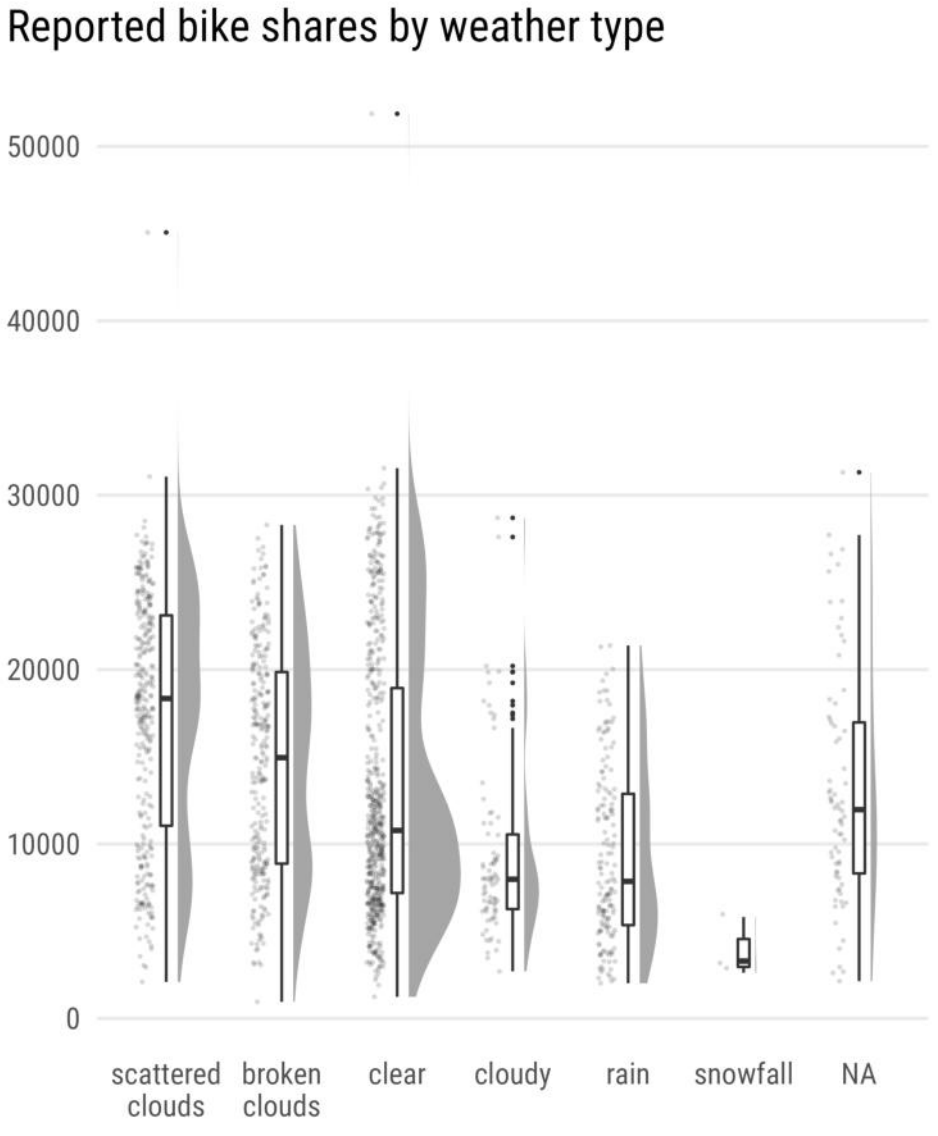
Raincloud Plots with {ggdist}

```
g +
  ggdist::stat_halfeye(
    aes(thickness = stat(f*n)),
    color = NA,
    width = .5,
    position = position_nudge(x = .2)
  ) +
  geom_boxplot(
    width = .3,
    outlier.color = NA
  ) +
  geom_jitter(
    width = .1,
    size = .5,
    alpha = .1
  )
```



Raincloud Plots with {gghalves}

```
g +
  ggdist::stat_halfeye(
    aes(thickness = stat(f*n)),
    color = NA,
    width = .5,
    position = position_nudge(x = .1)
  ) +
  geom_boxplot(
    width = .1,
    outlier.size = .2
  ) +
  gghalves::geom_half_point(
    side = "l",
    range_scale = .4,
    size = .3,
    alpha = .1
  )
```



Raincloud Plots with {gghalves}

```
g +
  ggdist::stat_halfeye(
    aes(thickness = stat(f*n)),
    color = NA,
    width = .5,
    position = position_nudge(x = .1)
  ) +
  geom_boxplot(
    width = .1,
    outlier.size = .2
  ) +
  gghalves::geom_half_point(
    side = "l",
    range_scale = .4,
    size = .3,
    alpha = .1
  ) +
  coord_flip()
```