Graphic Design with ggplot2

Working with Colors

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

Setup

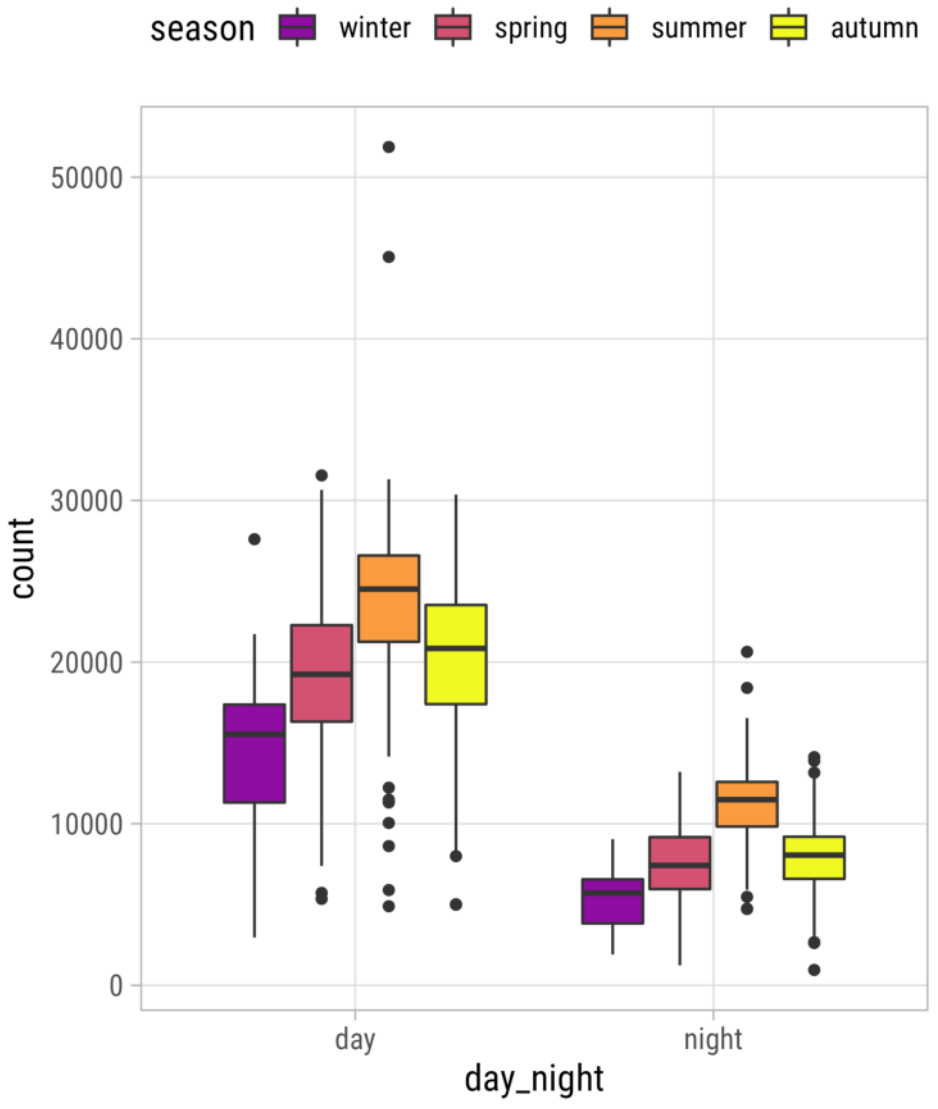
```
library(tidyverse)

library(tidyverse)

bikes <- readr::read_csv(
  here::here("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"
)
```

Color Palettes

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

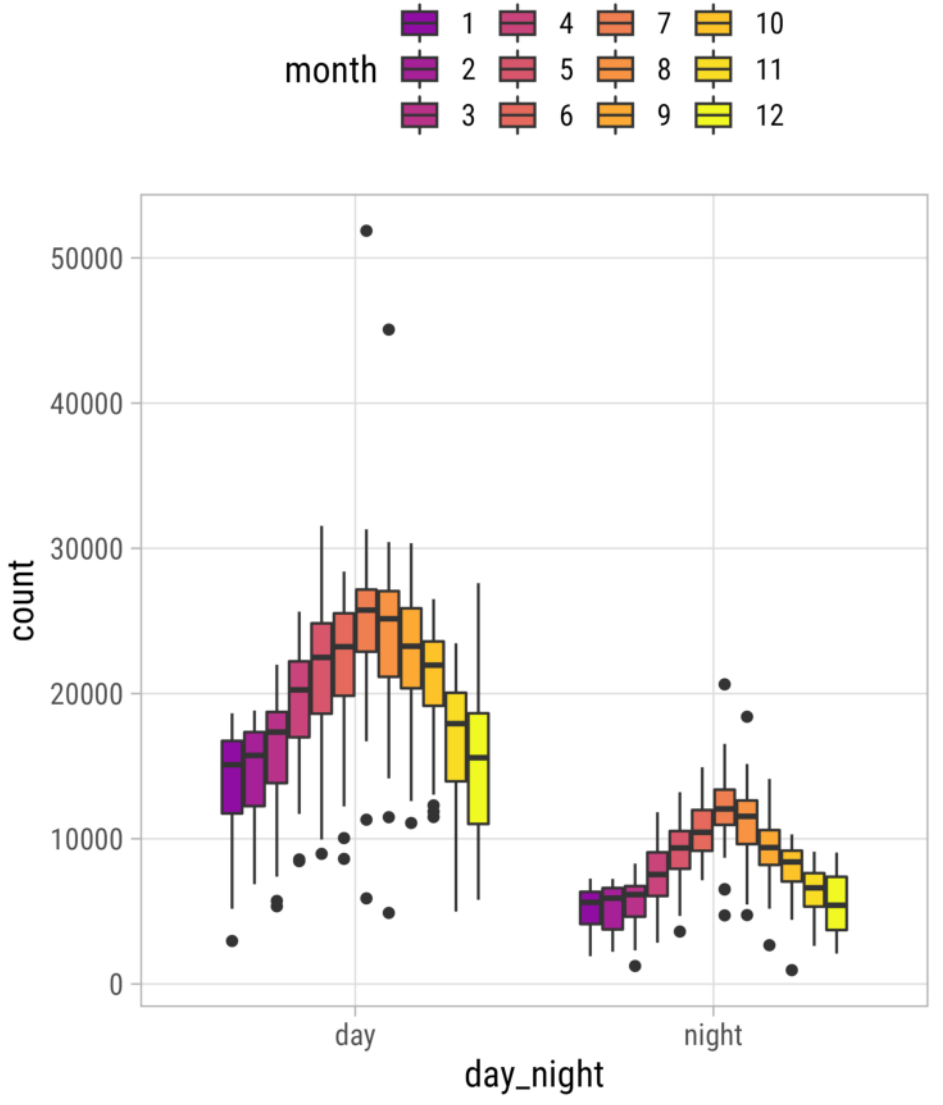
Pre-Defined Color Palettes: Viridis

```
ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scale_fill_viridis_d(
    option = "plasma",
    begin = .3
  )
```



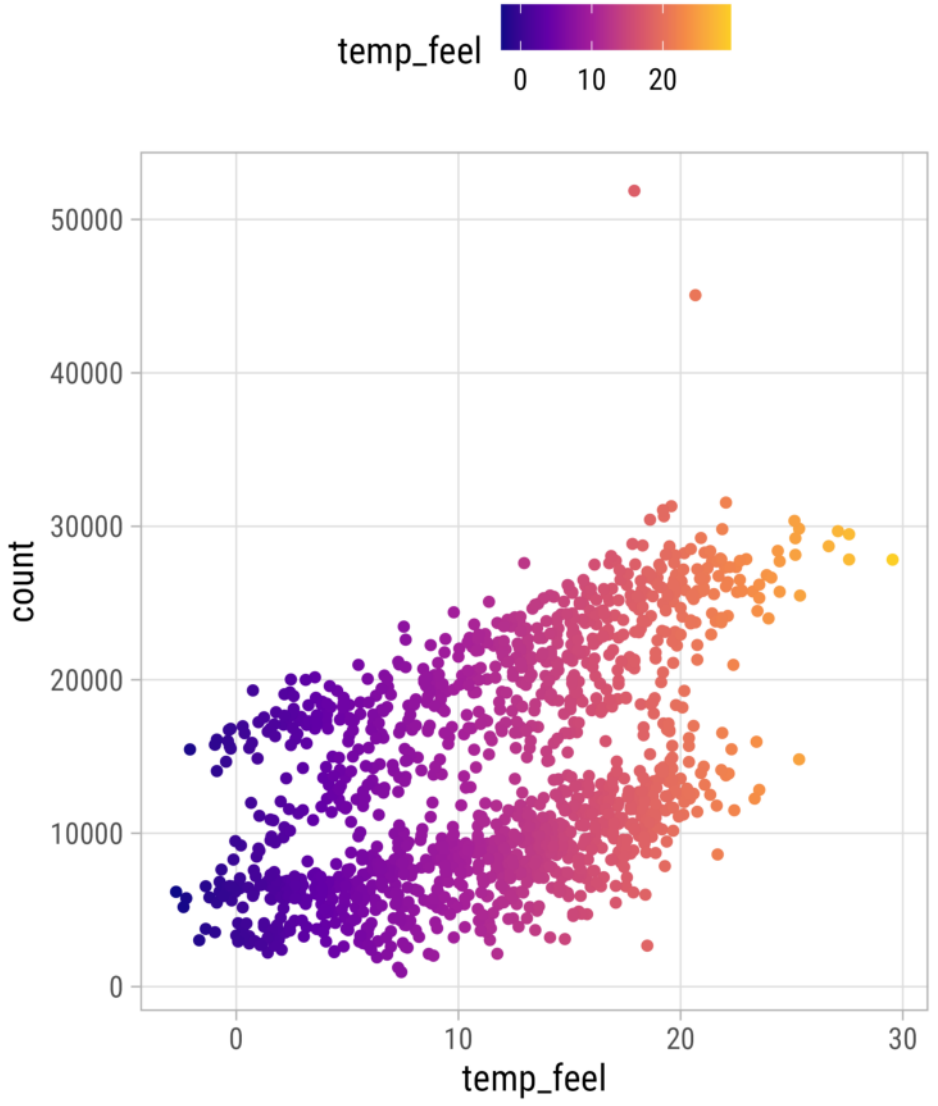
Pre-Defined Color Palettes: Viridis

```
ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = month)
  ) +
  geom_boxplot() +
  scale_fill_viridis_d(
    option = "plasma",
    begin = .3
  )
```



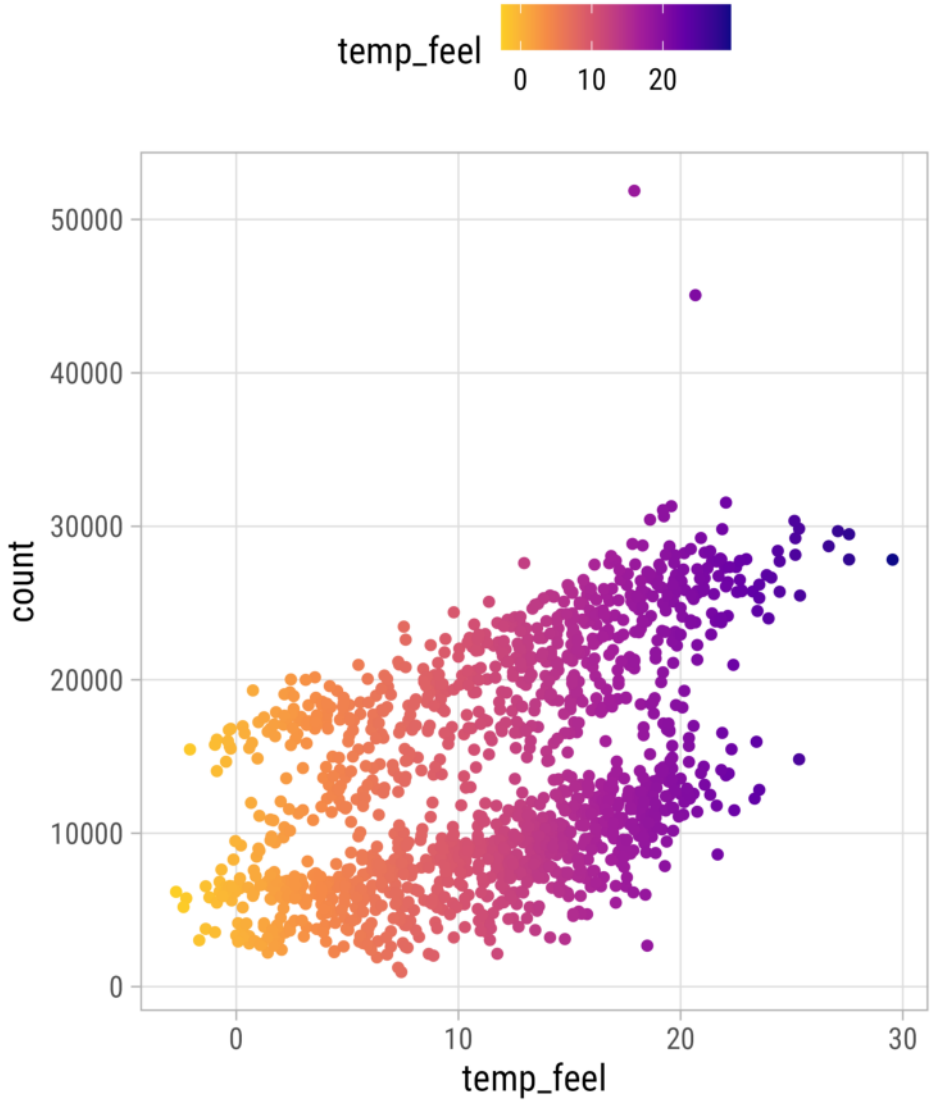
Pre-Defined Color Palettes: Viridis

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_viridis_c(
    option = "plasma",
    end = .9
  )
```



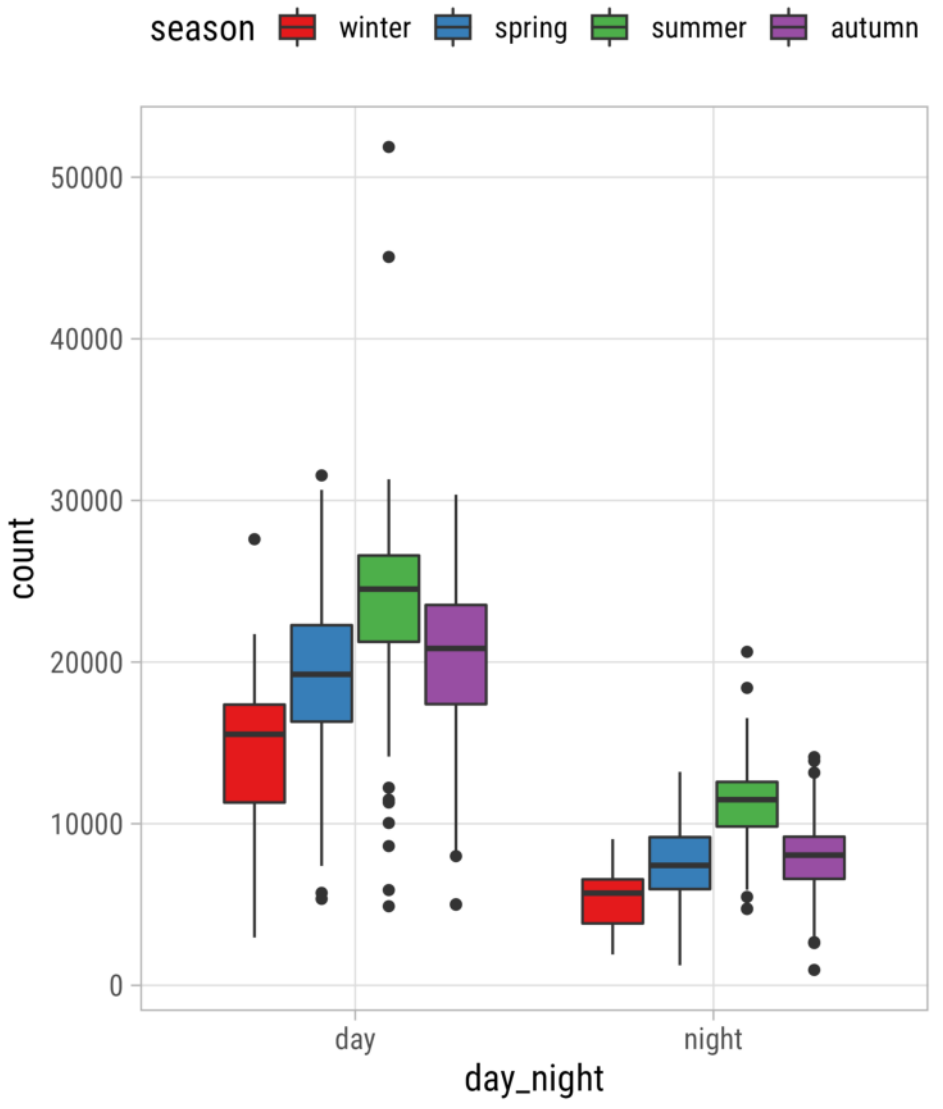
Pre-Defined Color Palettes: Viridis

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_viridis_c(
    option = "plasma",
    end = .9,
    direction = -1
  )
```



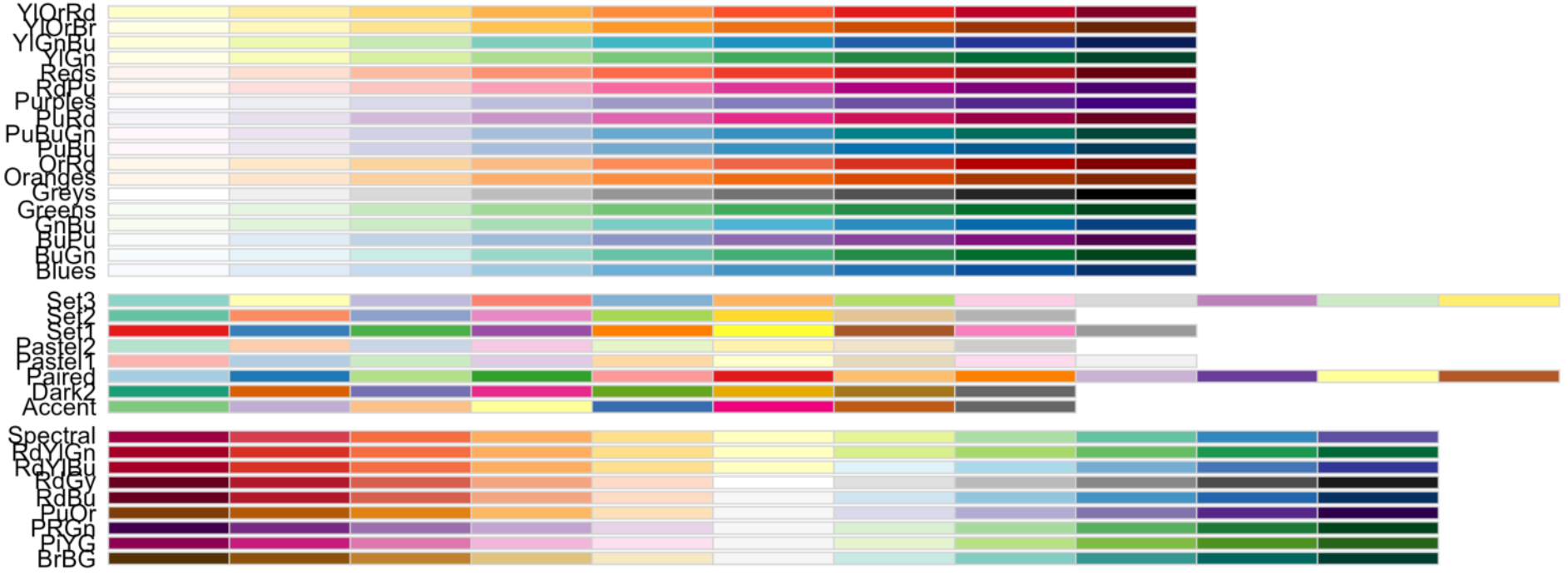
Pre-Defined Color Palettes: Brewer

```
ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scale_fill_brewer(
    palette = "Set1"
  )
```



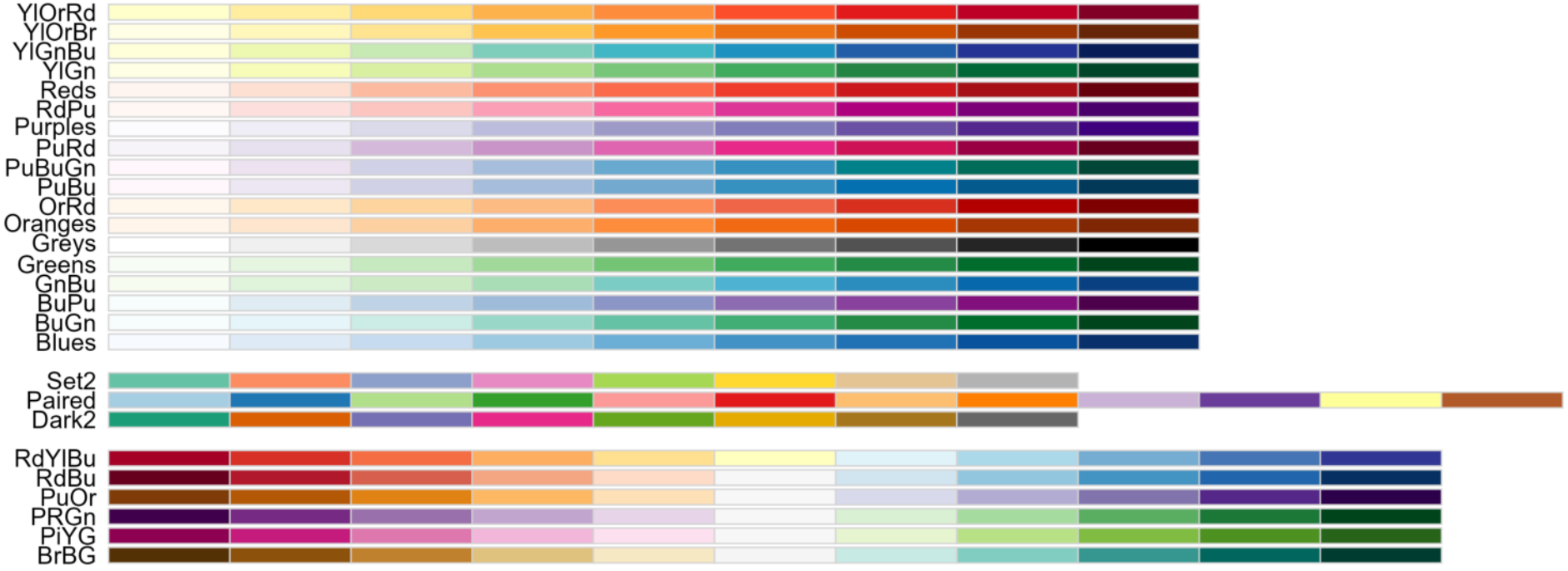
Pre-Defined Color Palettes: Brewer

```
1  RColorBrewer::display.brewer.all()
```



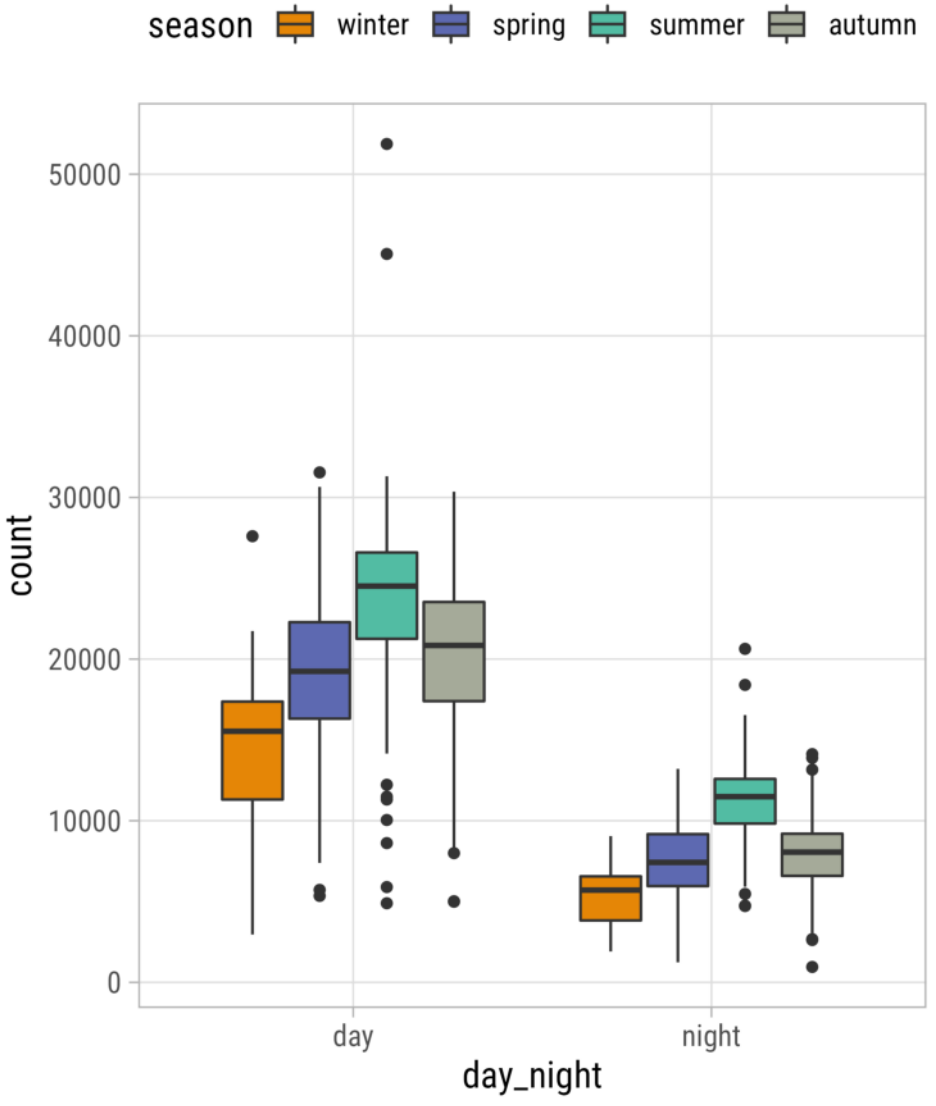
Pre-Defined Color Palettes: Brewer

```
1  RColorBrewer::display.brewer.all(colorblindFriendly = TRUE)
```



{rcartocolor}

```
# install.packages("rcartocolor")

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  rcartocolor::scale_fill_carto_d(
    palette = "Vivid"
  )
```

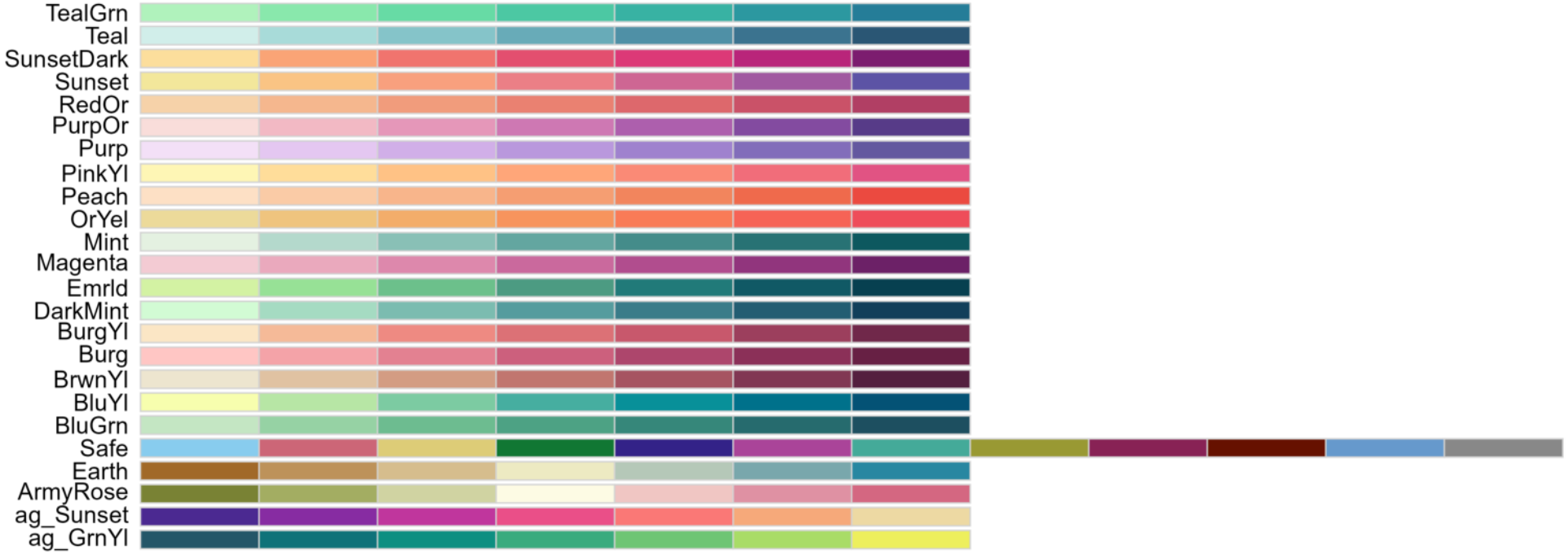


{rcartocolor}

```
1  rcartocolor::display_carto_all()
```

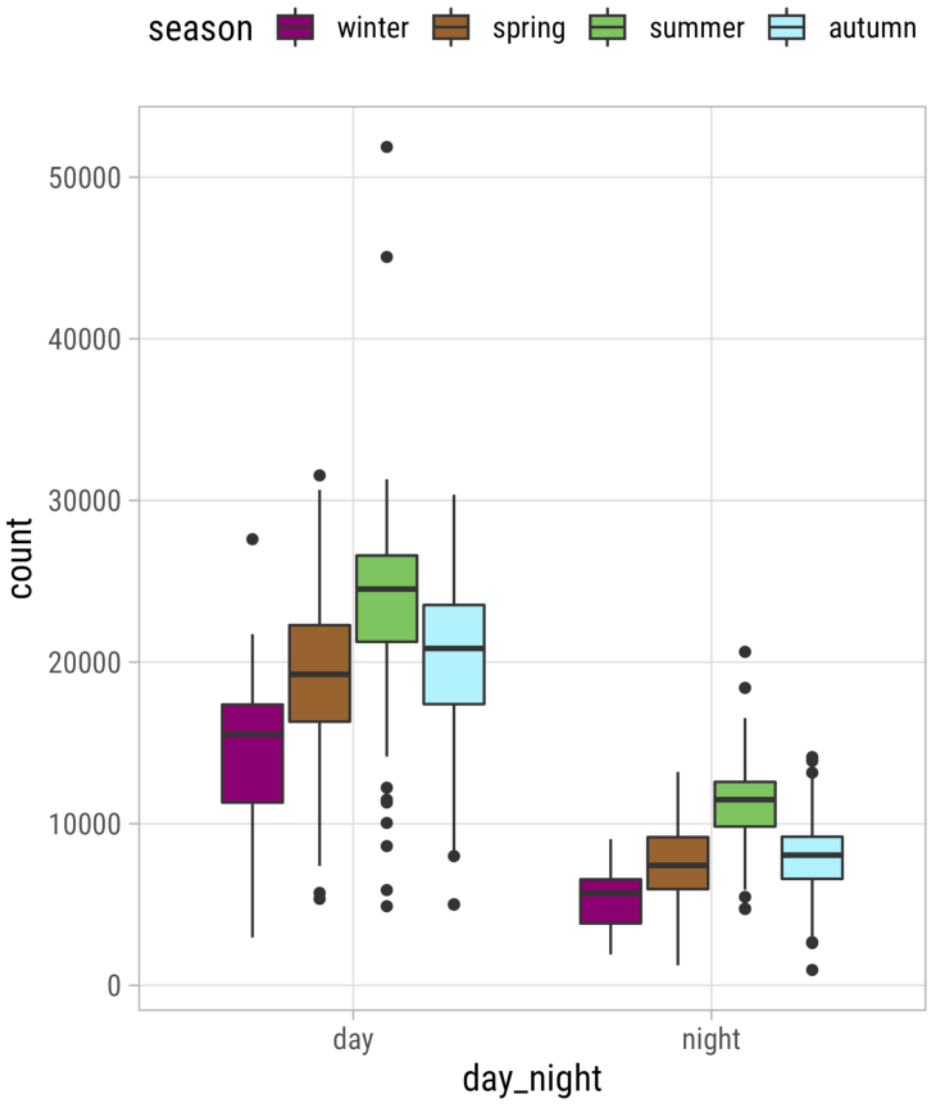
{rcartocolor}

```
1  rcartocolor::display_carto_all(colorblind_friendly = TRUE)
```



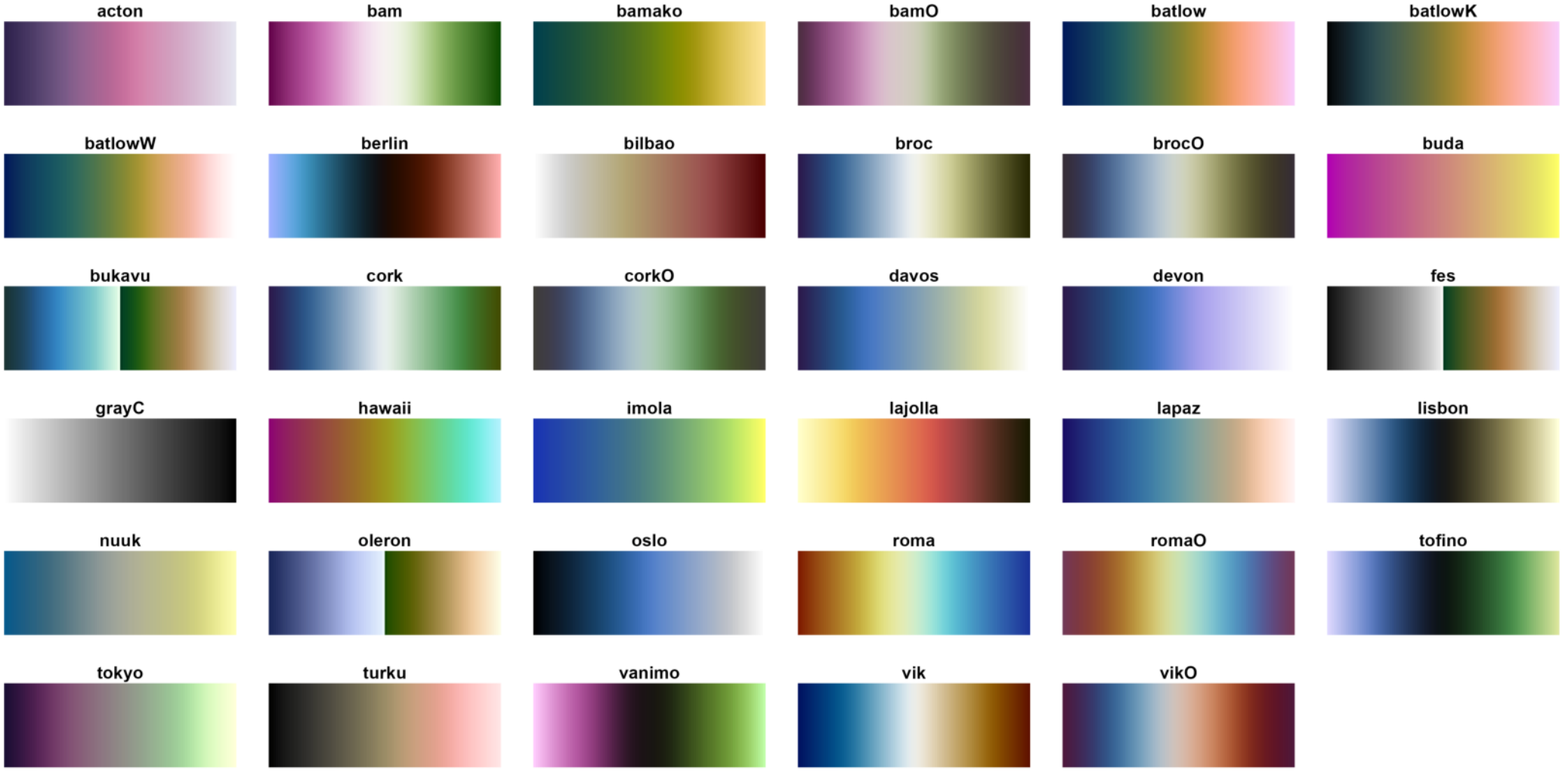
{scico}

```
# install.packages("scico")

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scico::scale_fill_scico_d(
    palette = "hawaii"
  )
```



{scico}

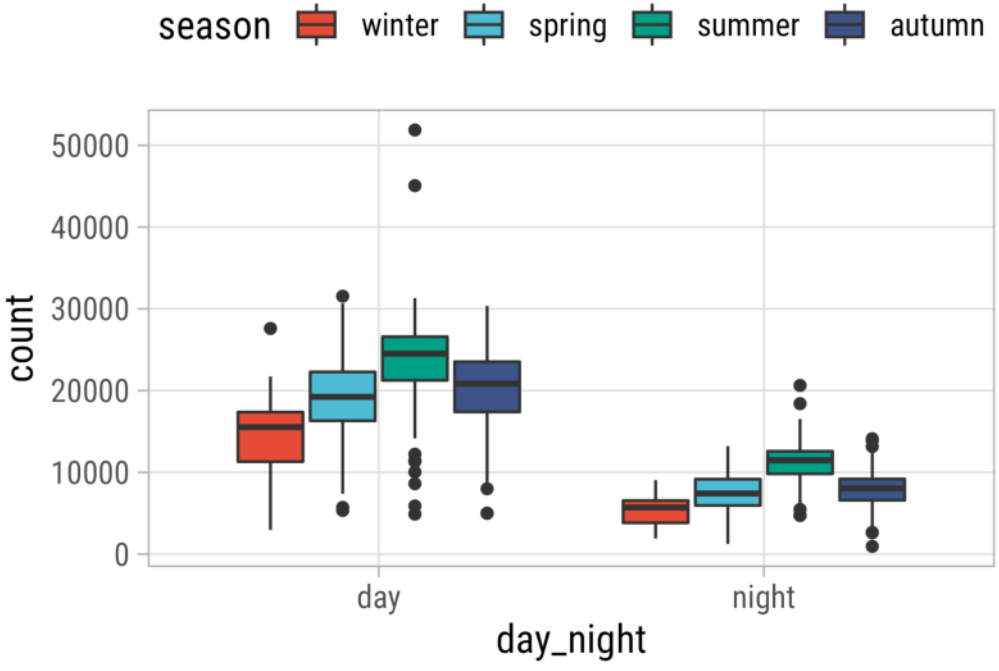
```
1  scico::scico_palette_show()
```

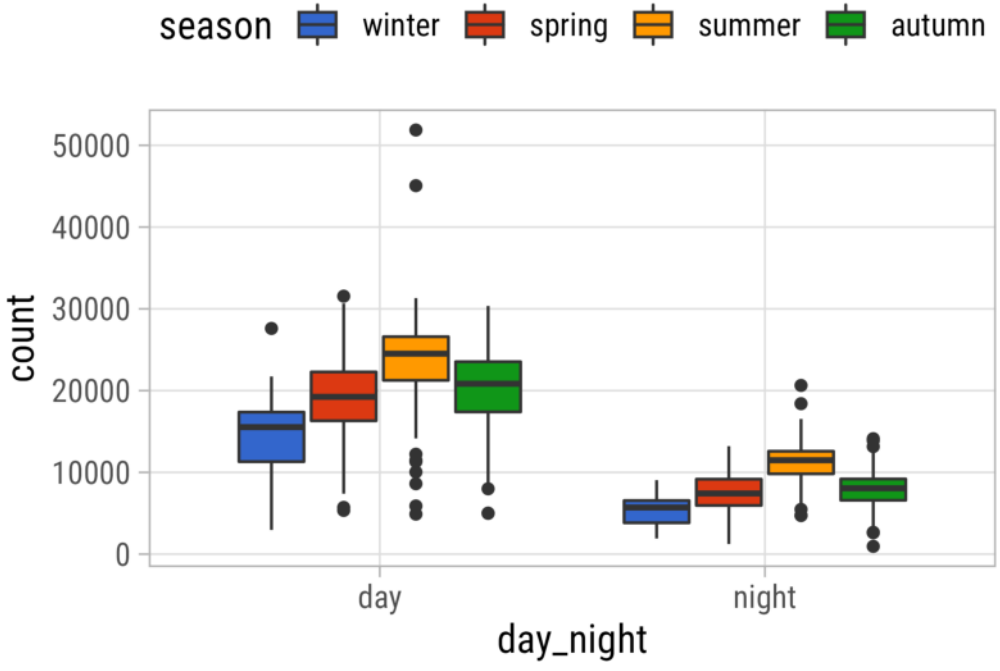


{ggsci} and {ggthemes}

```
1  # install.packages("ggsci")
2  ggplot(
3      bikes,
4      aes(x = day_night, y = count,
5          fill = season)
6    ) +
7    geom_boxplot() +
8    ggsci::scale_fill_npg()
```

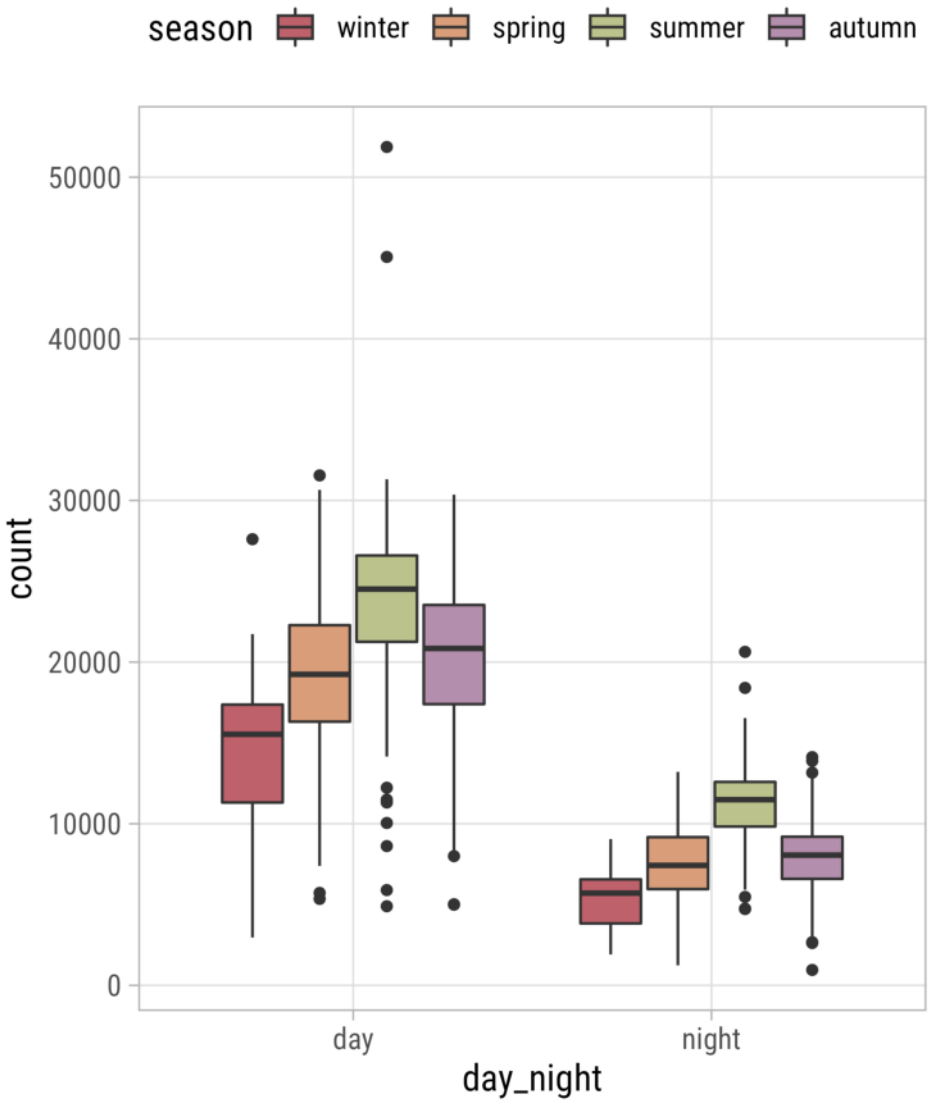
```
1  # install.packages("ggthemes")
2  ggplot(
3      bikes,
4      aes(x = day_night, y = count,
5          fill = season)
6    ) +
7    geom_boxplot() +
8    ggthemes::scale_fill_gdocs()
```





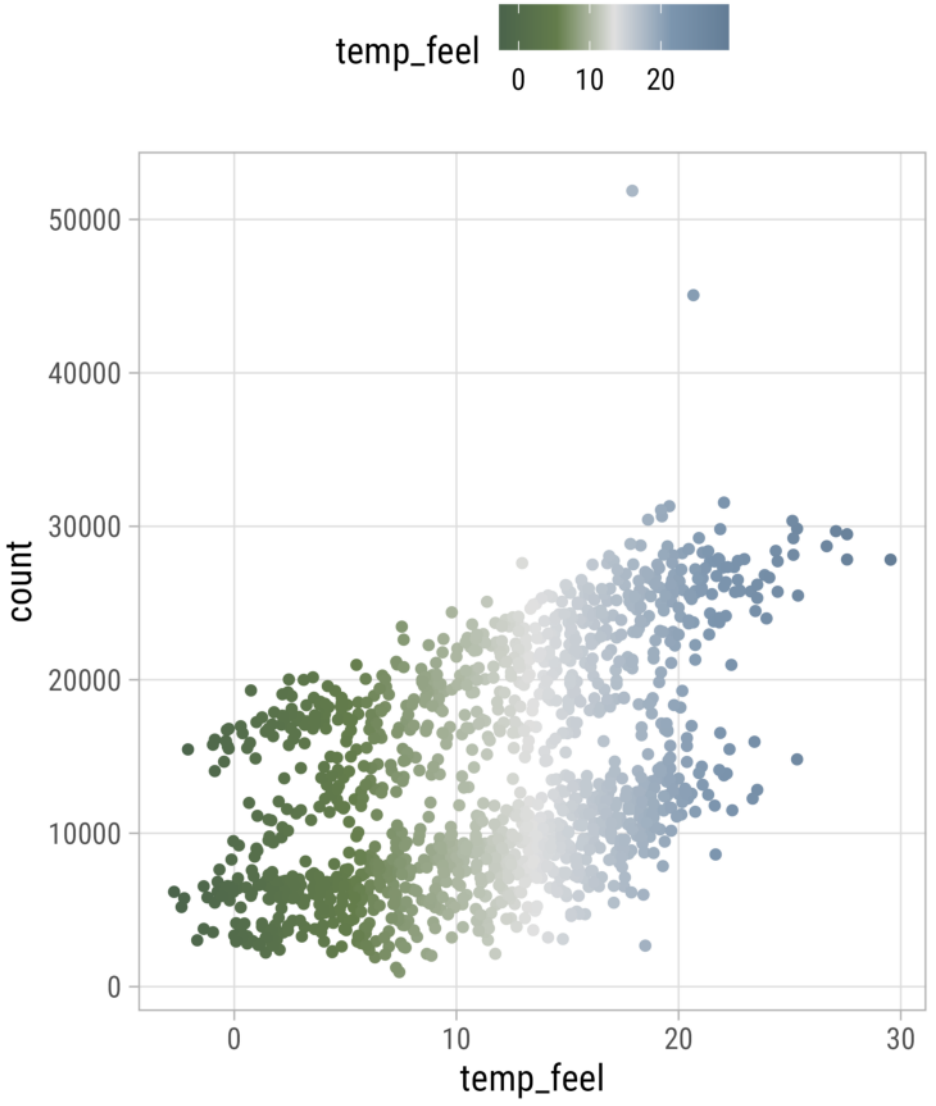
{nord}

```
# install.packages("nord")

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  nord::scale_fill_nord(
    palette = "aurora"
  )
```



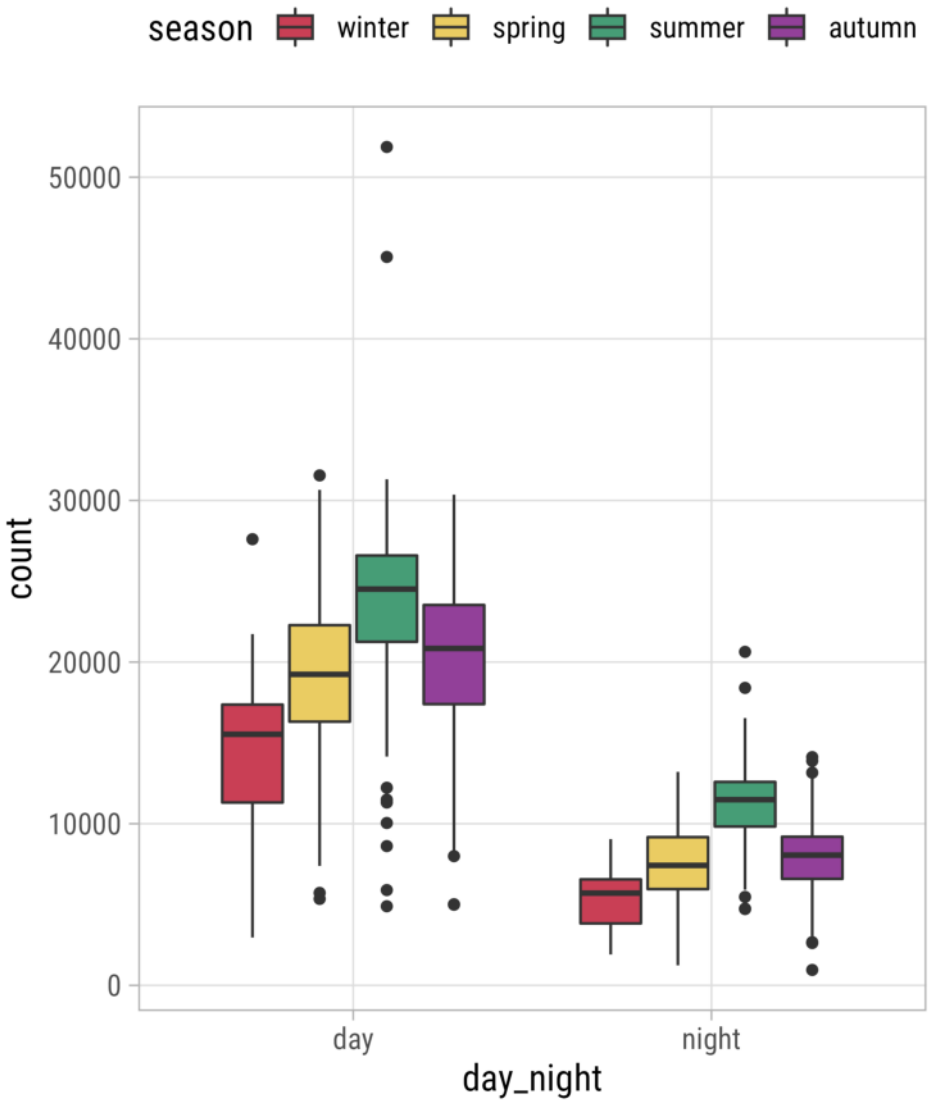
{nord}

```
# install.packages("nord")

ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  nord::scale_color_nord(
    palette = "silver_mine",
    discrete = FALSE
  )
```



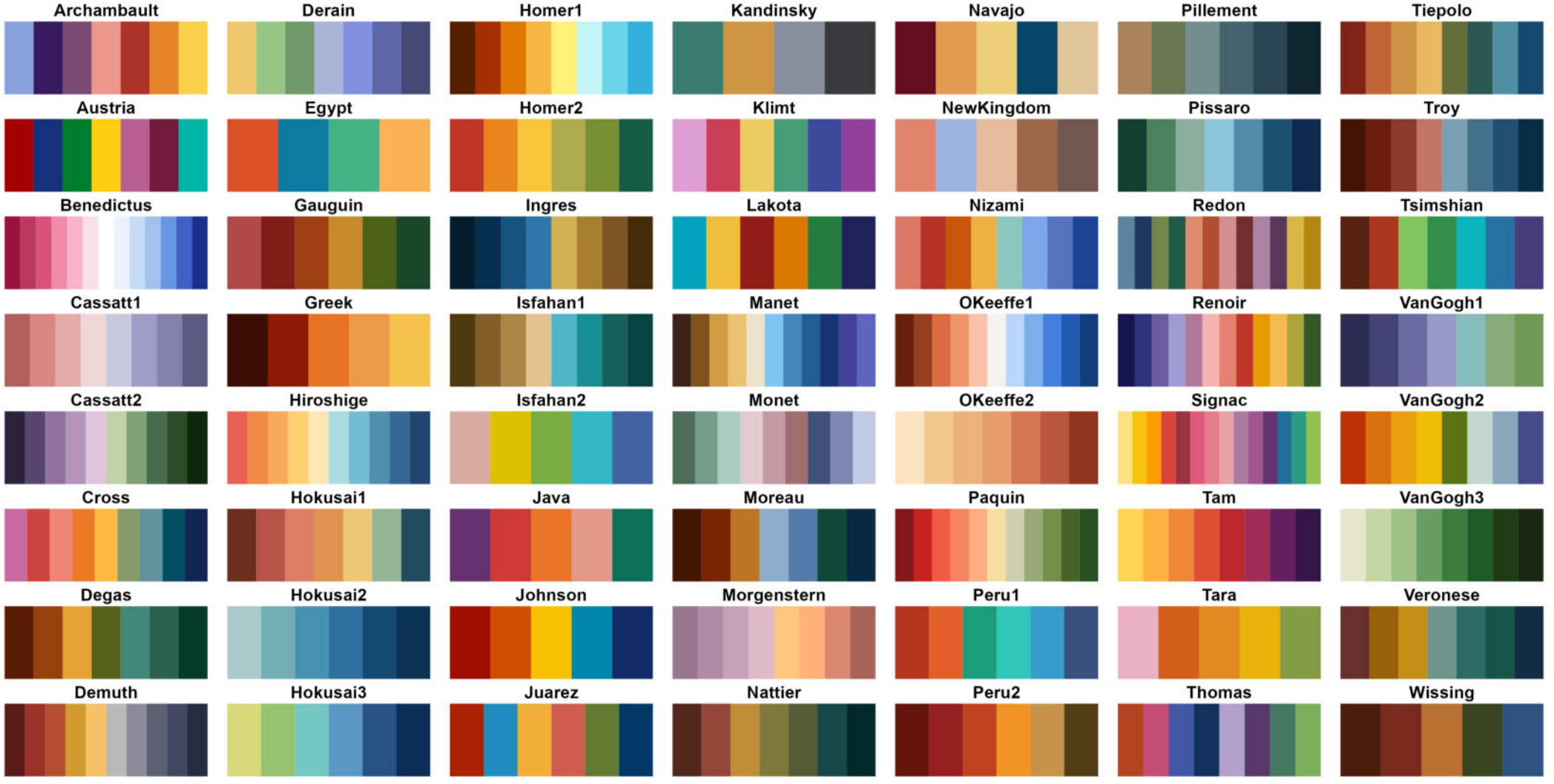
{MetBrewer}

```r
# install.packages("MetBrewer")

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  MetBrewer::scale_fill_met_d(
    name = "Klimt"
  )
```



{MetBrewer}

```
1  MetBrewer::display_all()
```

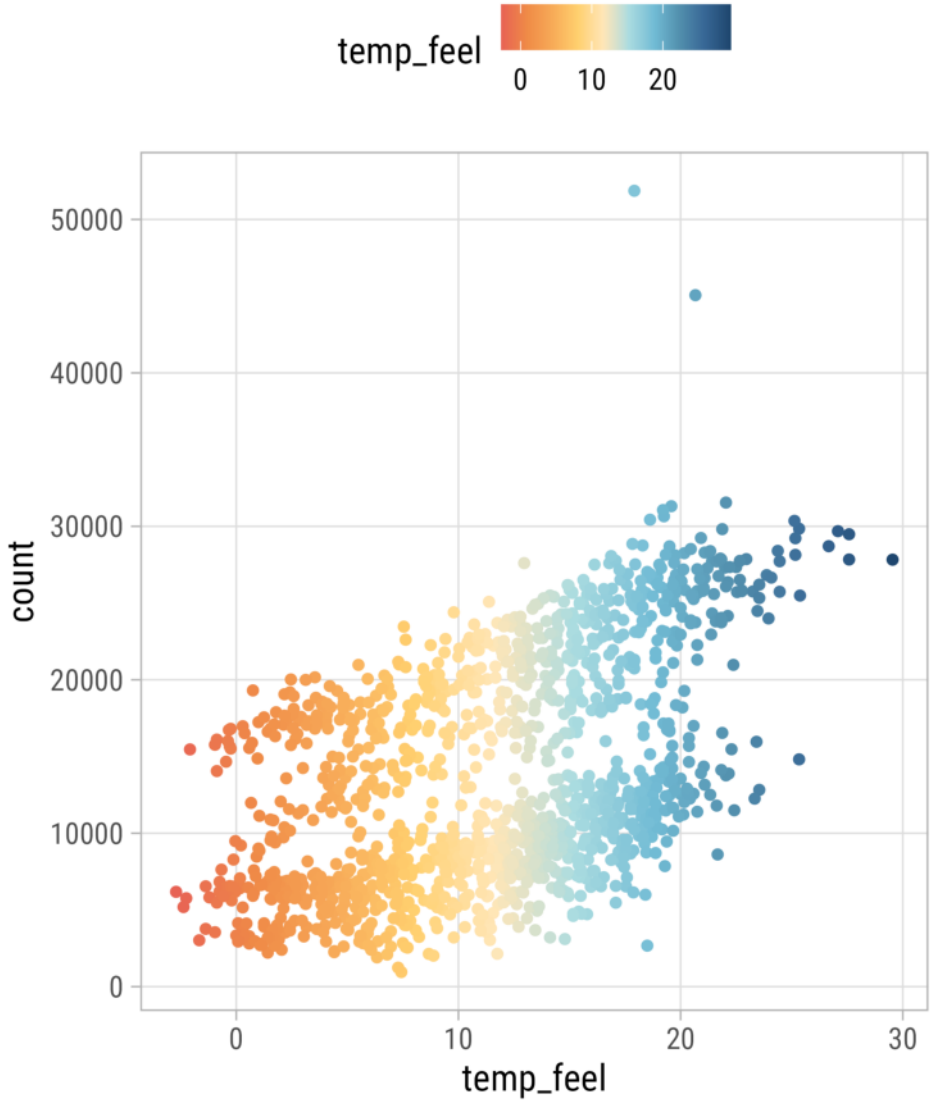


{MetBrewer}

```
1  MetBrewer::display_all(colorblind_only = TRUE)
```

{MetBrewer}

```r
# install.packages("MetBrewer")

ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  MetBrewer::scale_color_met_c(
    name = "Hiroshige"
  )
```

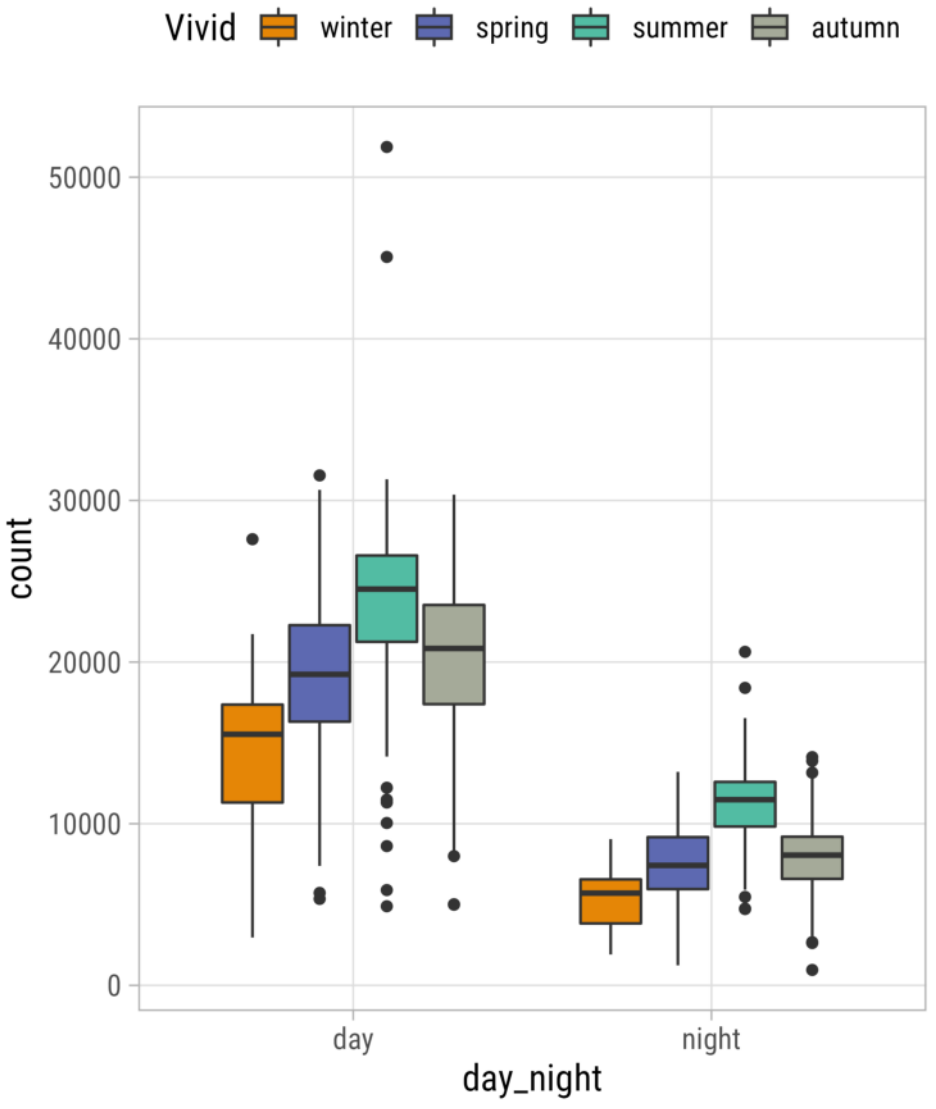


Customize Palettes

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

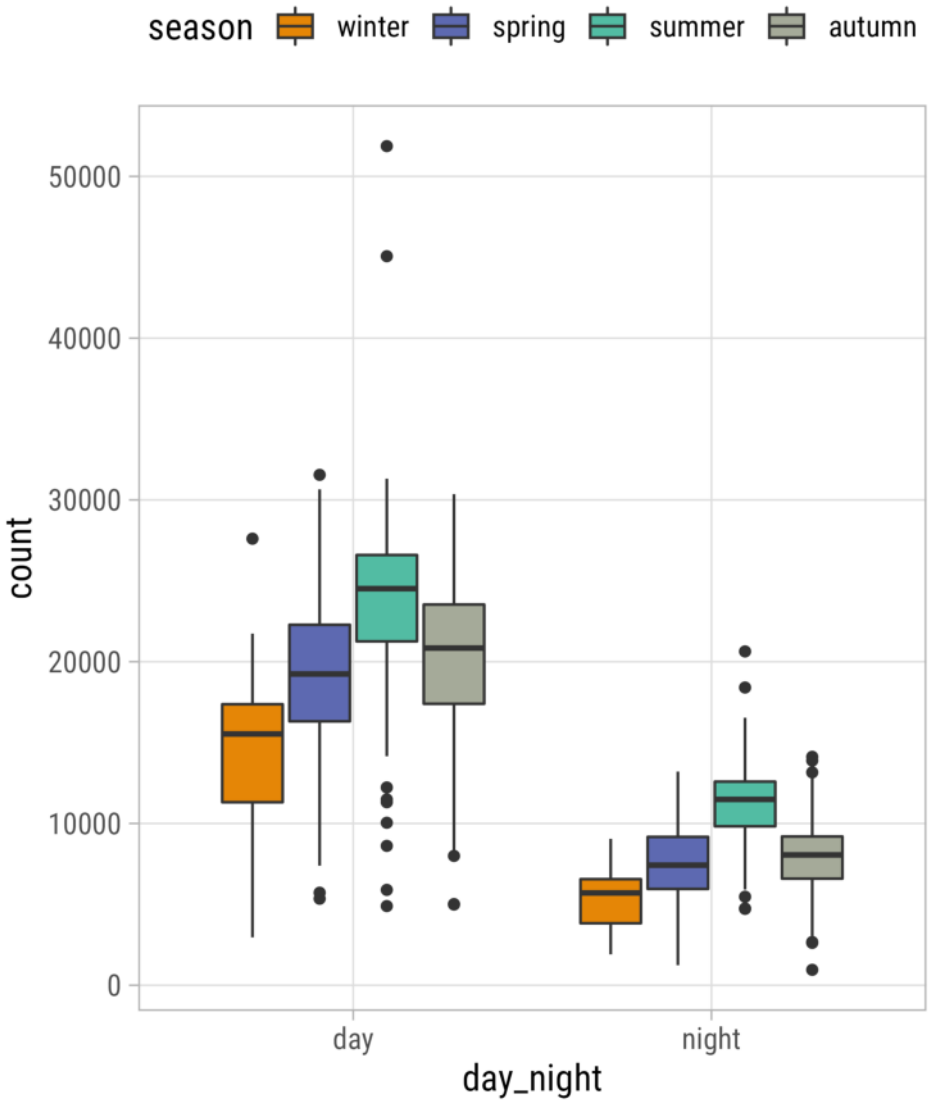
Customize Existing Palettes

```
1  library("rcartocolor")
2
3  ggplot(
4      bikes,
5      aes(x = day_night, y = count,
6          fill = season)
7    ) +
8    geom_boxplot() +
9    rcartocolor::scale_fill_carto_d(
10     name = "Vivid"
11   )
```



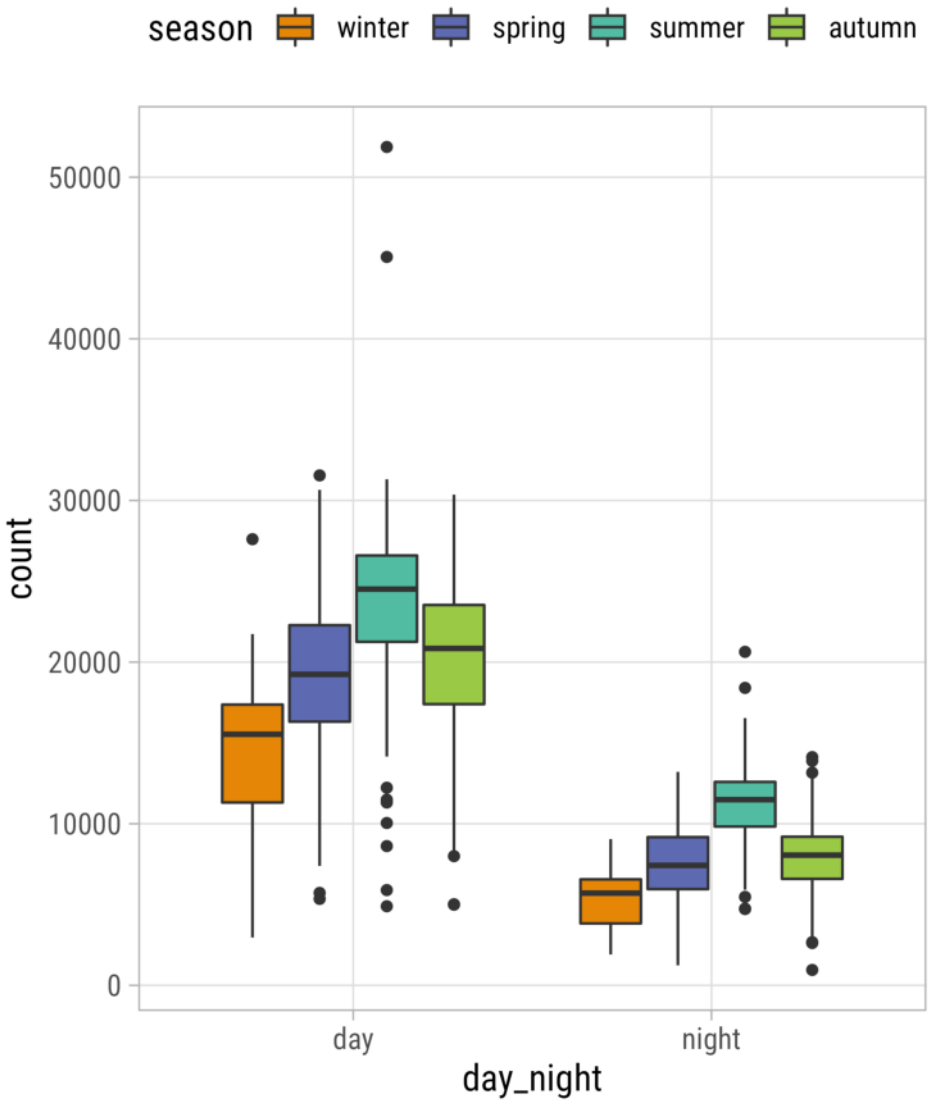
Customize Existing Palettes

```
1  library("rcartocolor")
2
3  ggplot(
4      bikes,
5      aes(x = day_night, y = count,
6          fill = season)
7    ) +
8    geom_boxplot() +
9    scale_fill_manual(
10     values = carto_pal(
11       name = "Vivid", n = 4
12     )
13   )
```



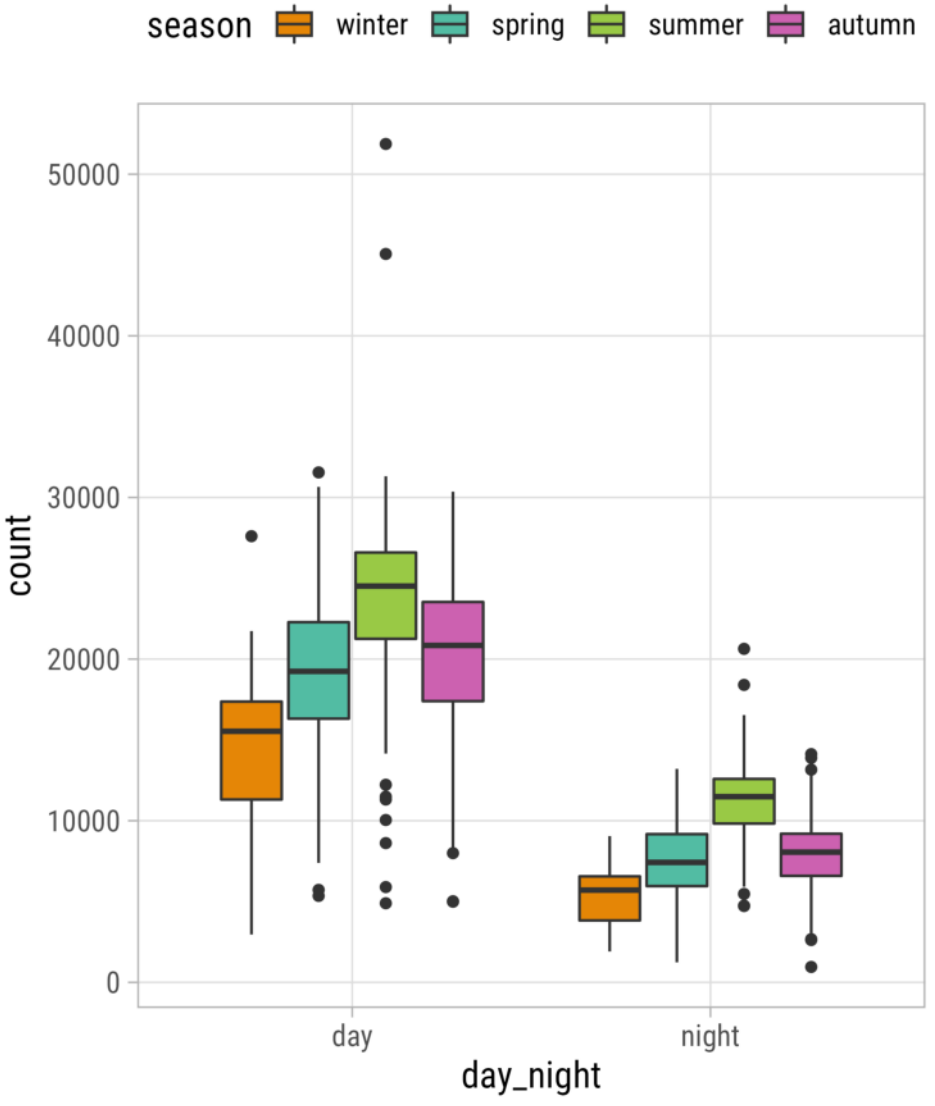
Customize Existing Palettes

```
1  library("rcartocolor")
2
3  ggplot(
4      bikes,
5      aes(x = day_night, y = count,
6          fill = season)
7    ) +
8    geom_boxplot() +
9    scale_fill_manual(
10     values = carto_pal(
11       name = "Vivid", n = 5
12     )[1:4]
13   )
```



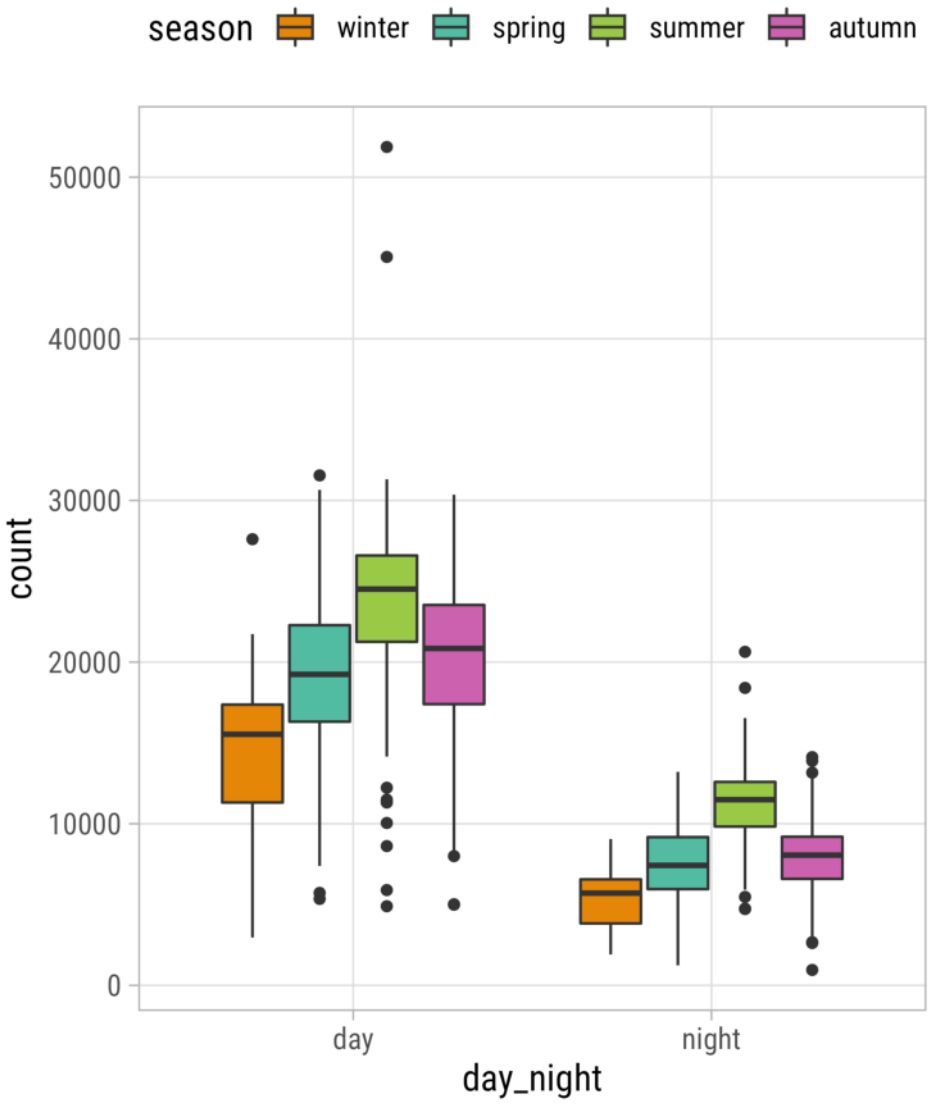
Customize Existing Palettes

```
 1  library("rcartocolor")
 2
 3  ggplot(
 4      bikes,
 5      aes(x = day_night, y = count,
 6          fill = season)
 7    ) +
 8    geom_boxplot() +
 9    scale_fill_manual(
10      values = carto_pal(
11        name = "Vivid", n = 6
12      )[c(1, 3:5)]
13    )
```



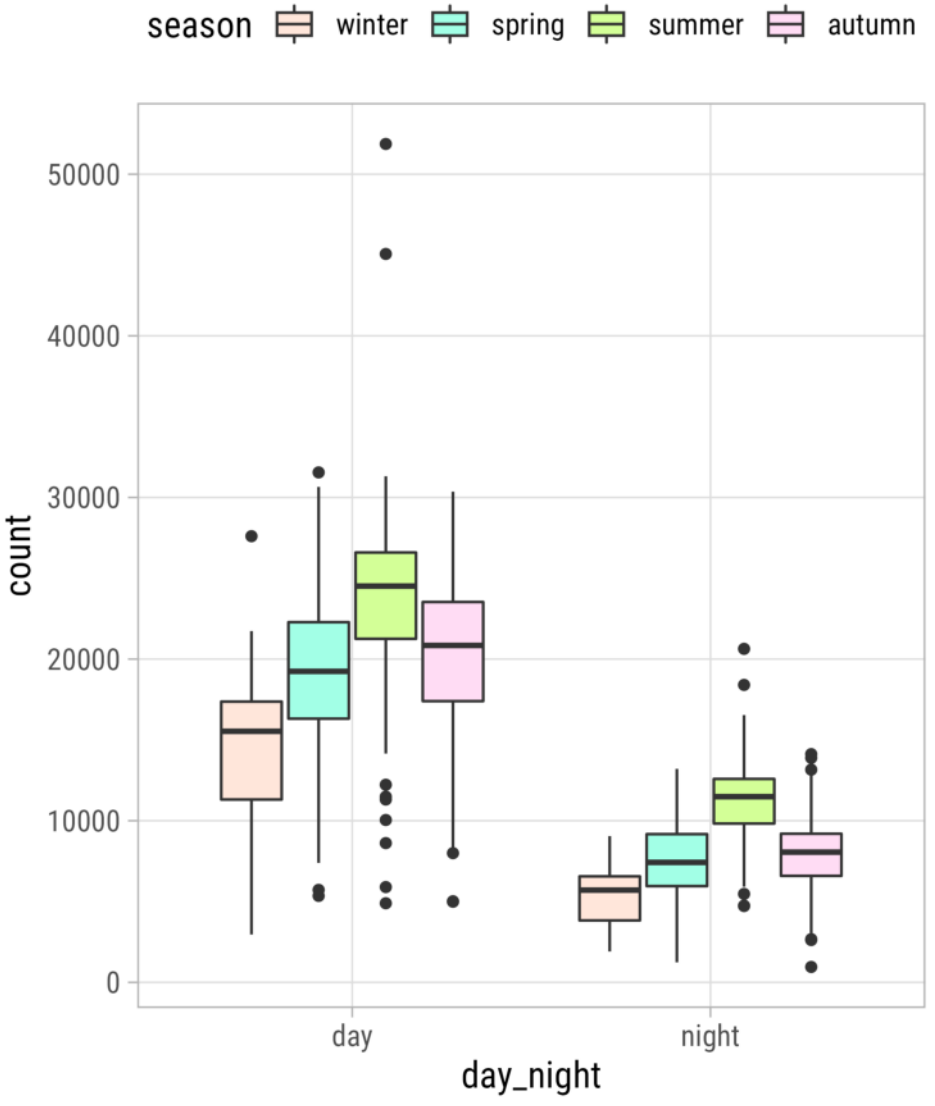
Customize Existing Palettes

```
carto_custom <-
  carto_pal(
    name = "Vivid", n = 6
  )[c(1, 3:5)]

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scale_fill_manual(
    values = carto_custom
  )
```



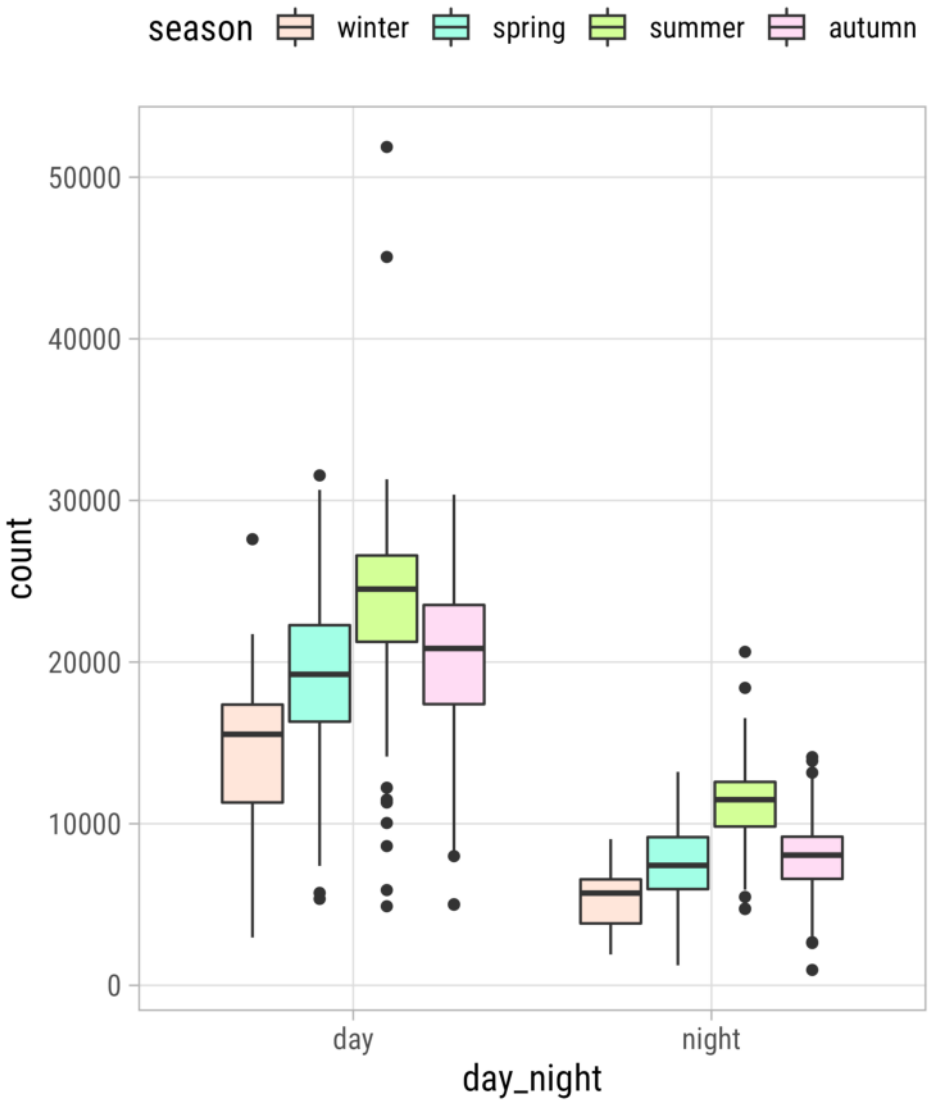
Customize Existing Palettes

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

carto_light <- lighten(carto_custom, .8)

ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scale_fill_manual(
    values = carto_light
  )
```



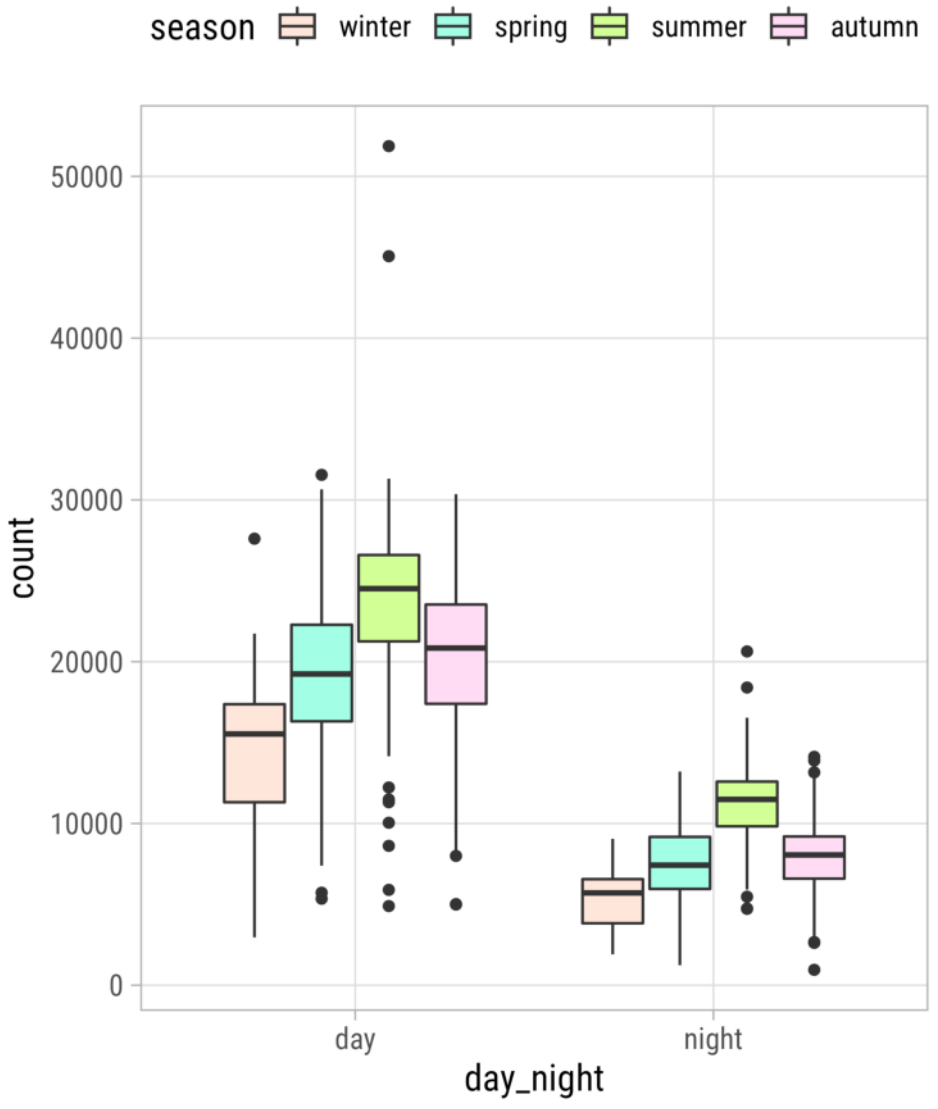
Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(fill = season,
        fill = after_scale(
          lighten(fill, .8)
    ))
  ) +
  scale_fill_manual(
    values = carto_custom
  )
```



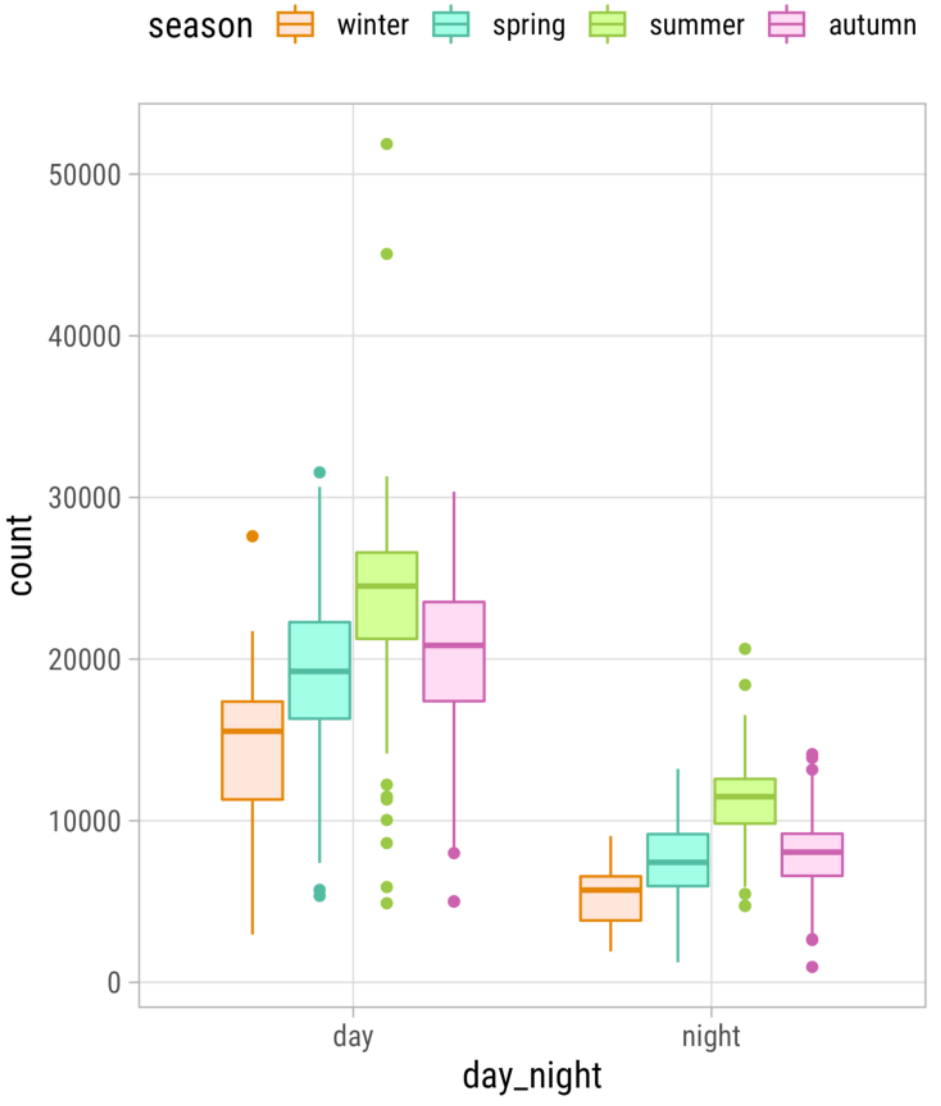
Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(
      fill = stage(
        season,
        after_scale =
          lighten(fill, .8)
      )
    )
  ) +
  scale_fill_manual(
    values = carto_custom
  )
```



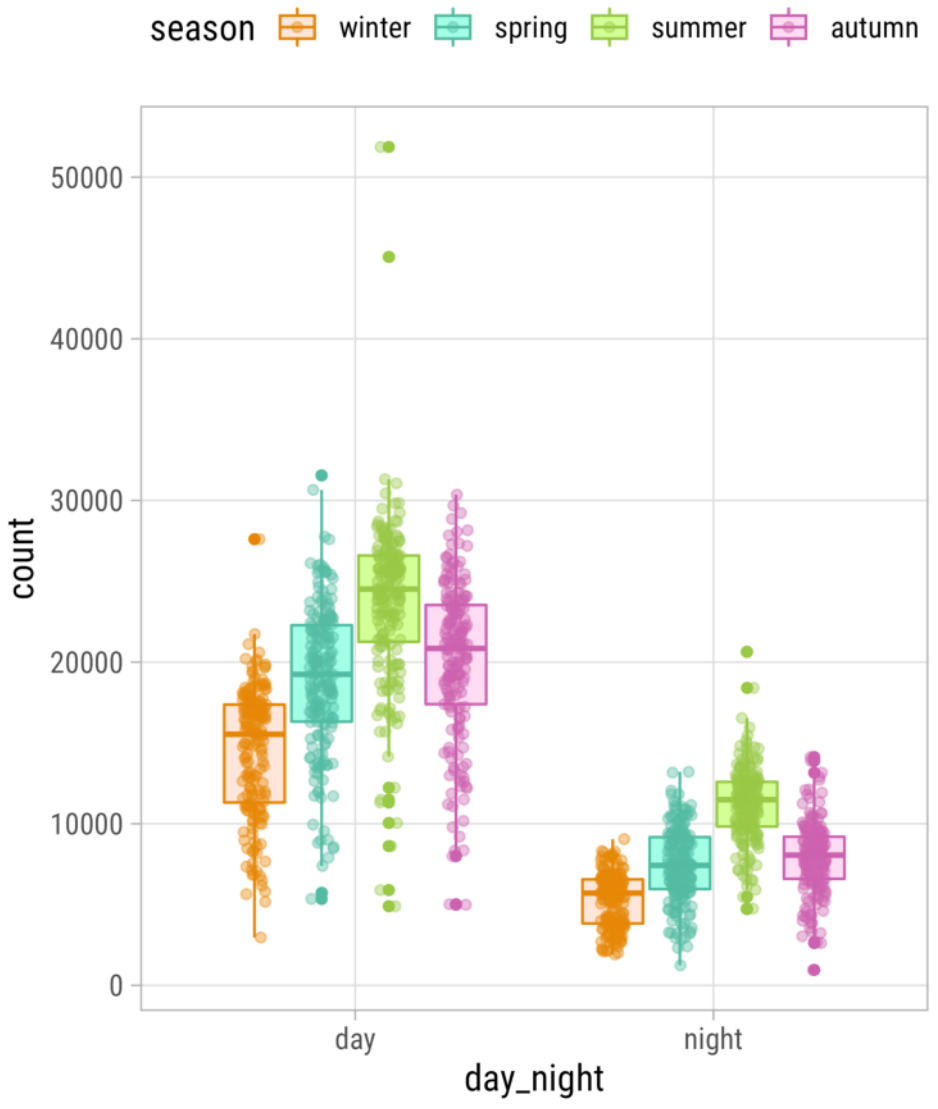
Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(color = season,
        fill = after_scale(
          lighten(color, .8)
    ))
  ) +
  scale_color_manual(
    values = carto_custom
  )
```



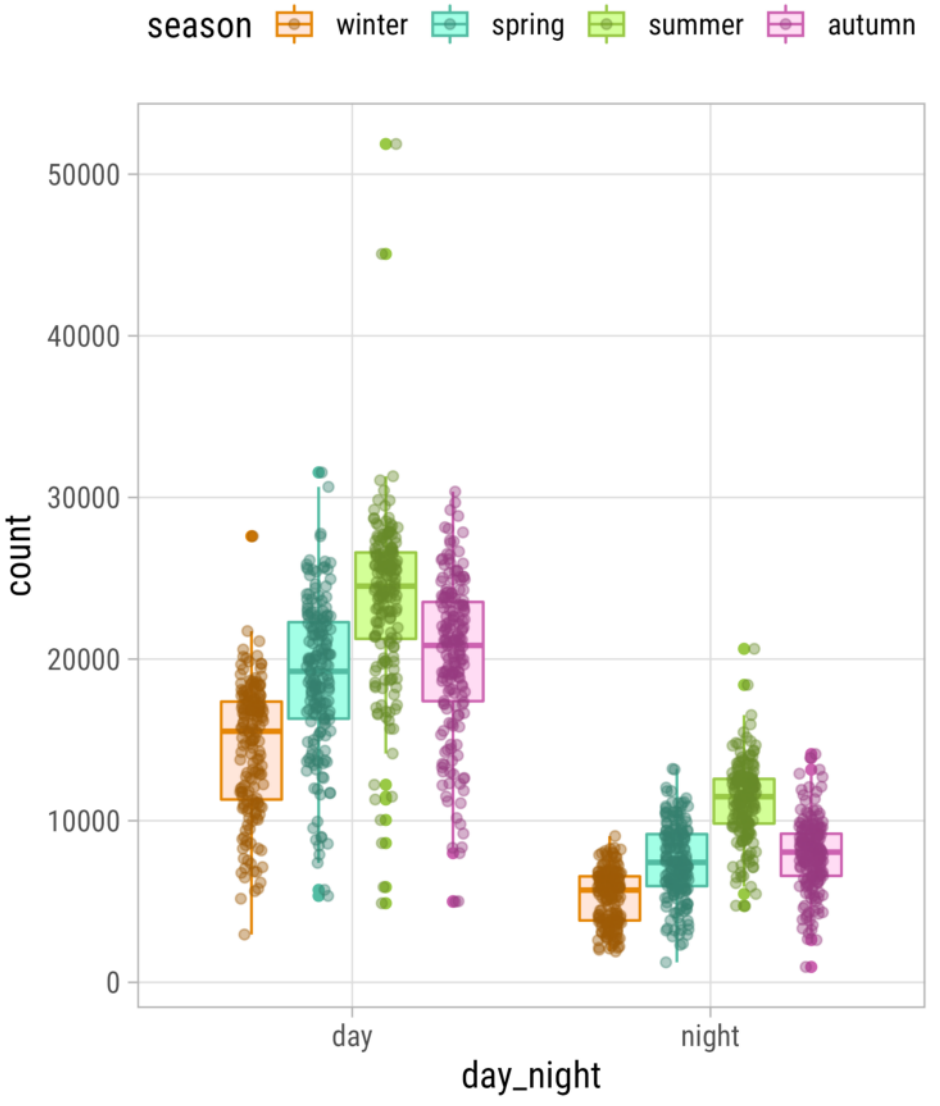
Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(color = season,
        fill = after_scale(
          lighten(color, .8)
    ))
  ) +
  geom_jitter(
    aes(color = season),
    position = position_jitterdodge(
      dodge.width = .75,
      jitter.width = .2
    ),
    alpha = .4
  ) +
  scale_color_manual(
    values = carto_custom
  )
```



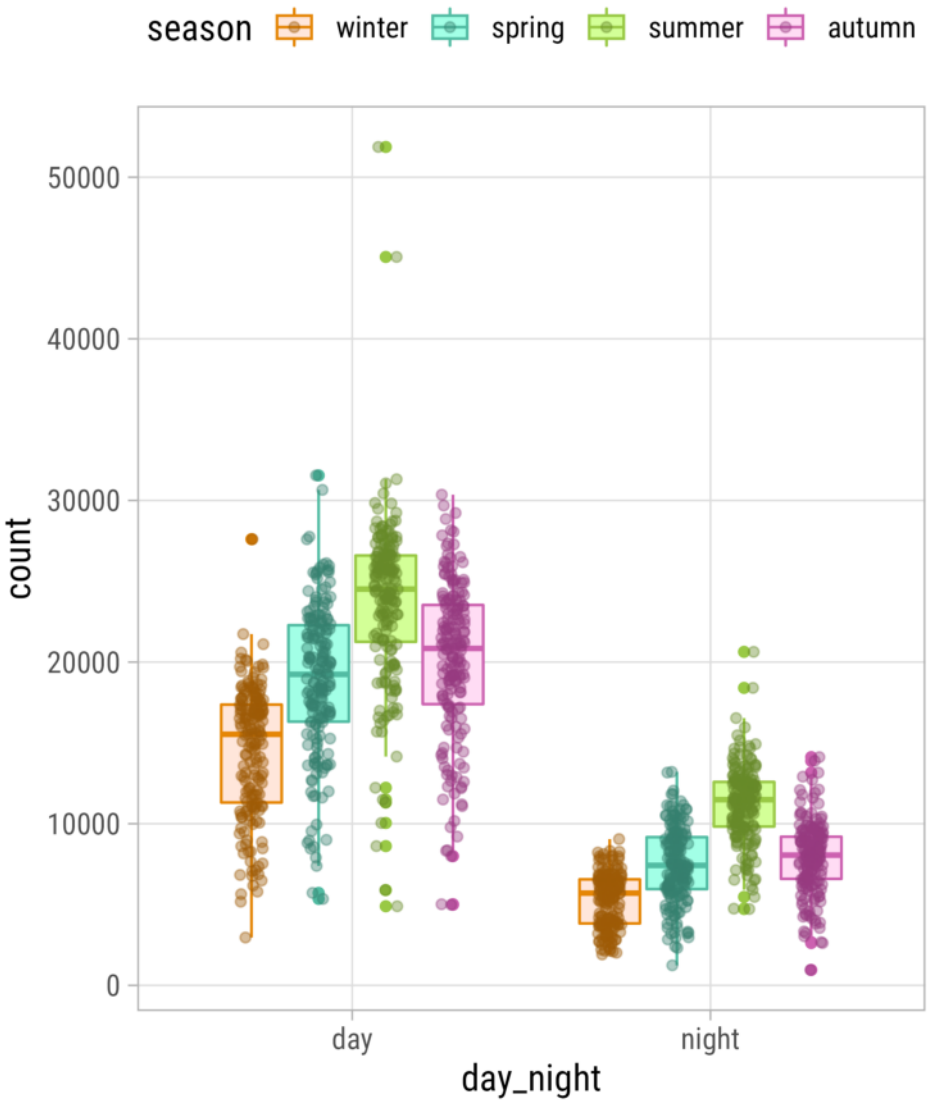
Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(color = season,
        fill = after_scale(
          lighten(color, .8)
        ))
  ) +
  geom_jitter(
    aes(color = season,
        color = after_scale(
          darken(color, .3)
    )),
    position = position_jitterdodge(
      dodge.width = .75,
      jitter.width = .2
    ),
    alpha = .4
  ) +
  scale_color_manual(
    values = carto_custom
```



Customize Existing Palettes

```
ggplot(
    bikes,
    aes(x = day_night, y = count)
  ) +
  geom_boxplot(
    aes(color = season,
        fill = after_scale(
          lighten(color, .8)
        ))
  ) +
  geom_jitter(
    aes(color = season,
        color = after_scale(
          darken(color, .3)
        )),
    position = position_jitterdodge(
      dodge.width = .75,
      jitter.width = .2
    ),
    alpha = .4
  ) +
  scale_color_manual(
    values = carto_custom
```

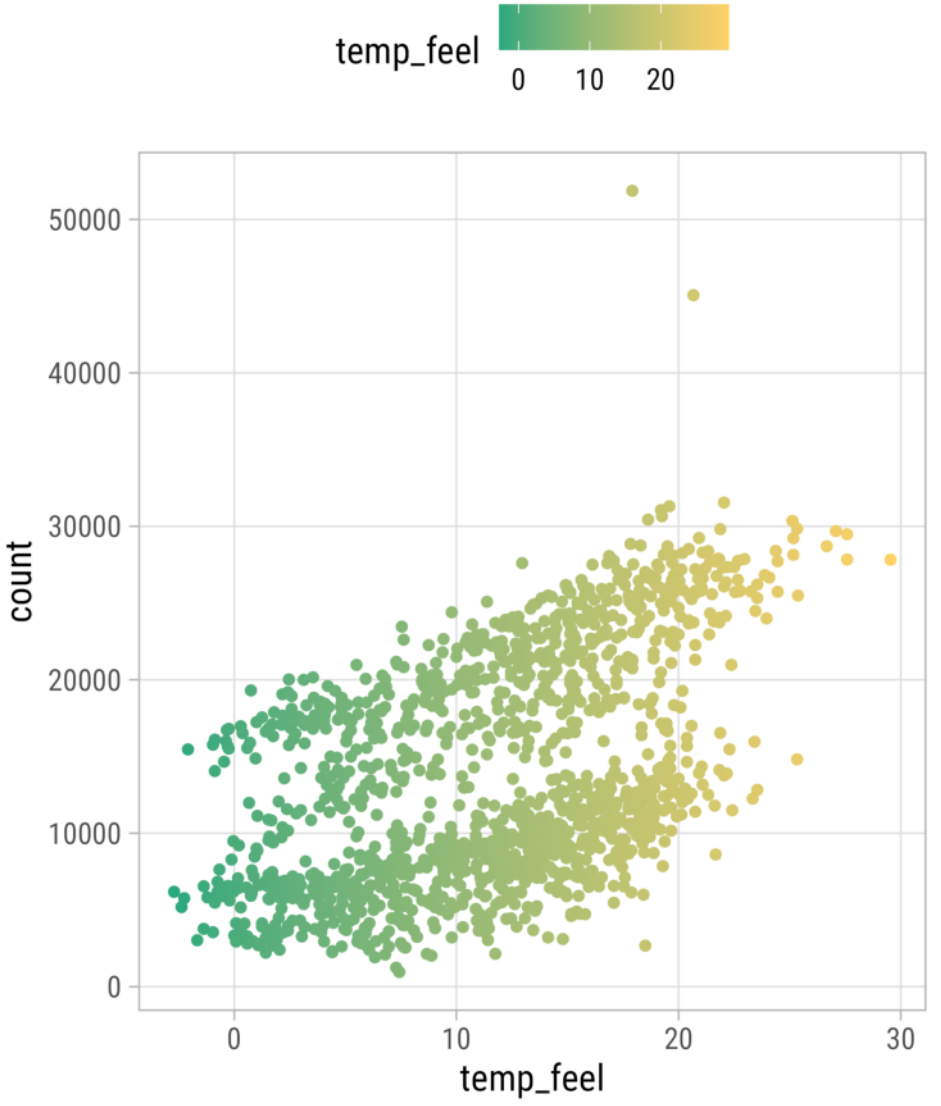


Create New Palettes

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

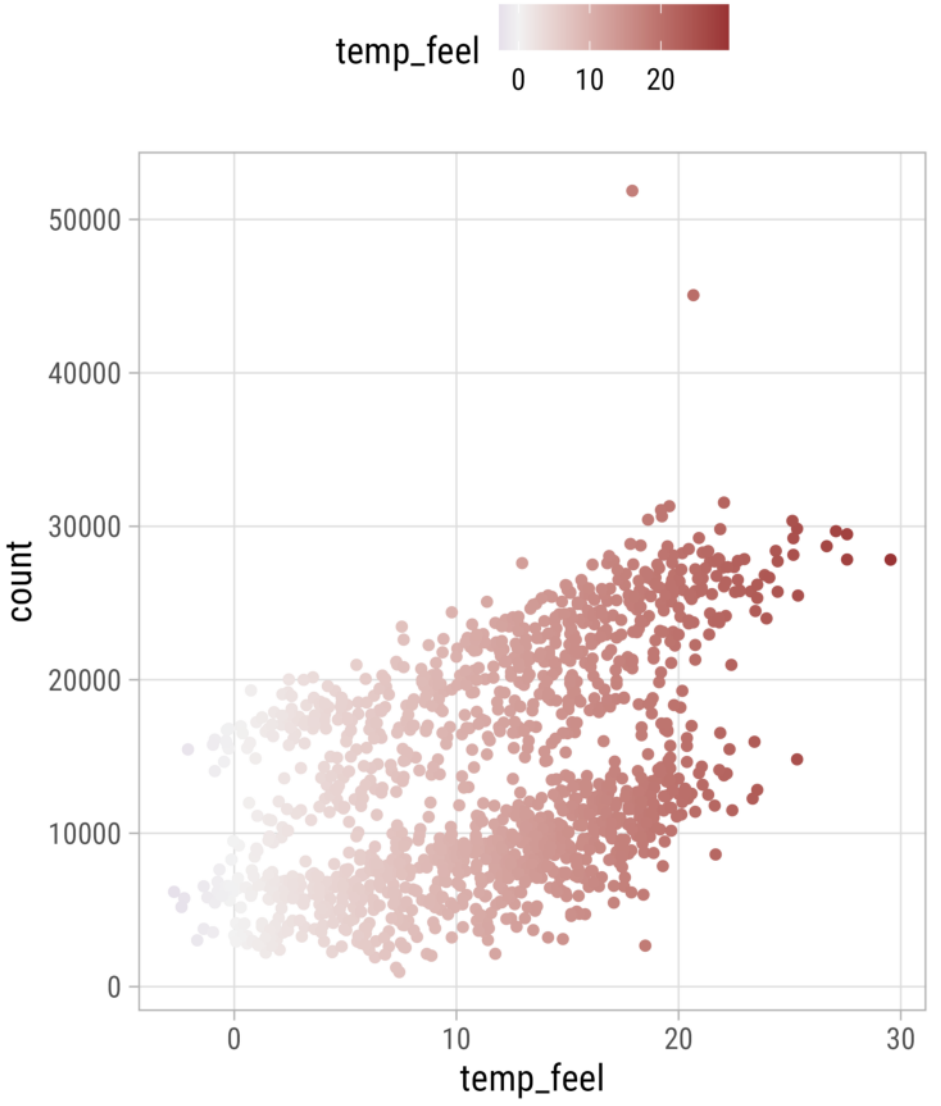
Create Sequential Palettes

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradient(
    low = "#28A87D",
    high = "#FFD166"
  )
```



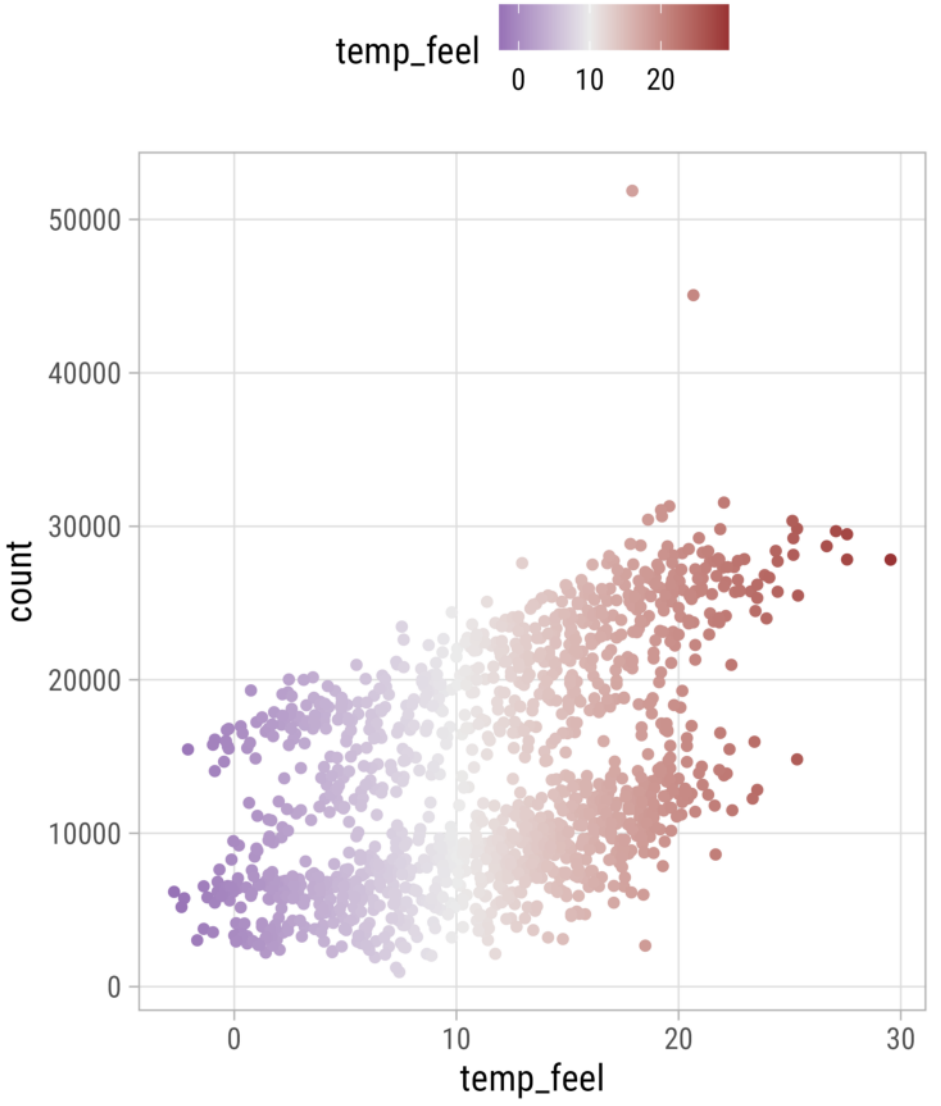
Create Diverging Palettes

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradient2(
    low = "#663399",
    high = "#993334",
    mid = "grey95"
  )
```



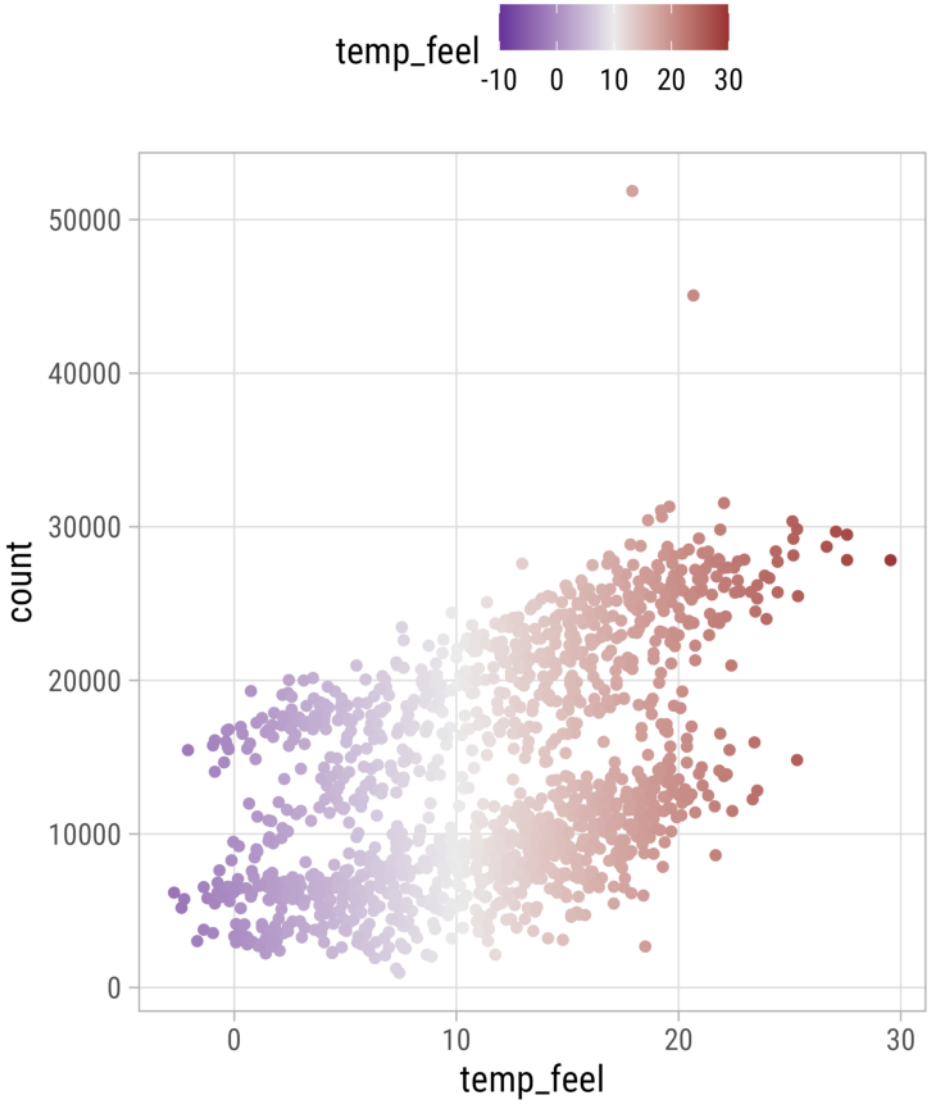
Create Diverging Palettes

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradient2(
    low = "#663399",
    high = "#993334",
    mid = "grey92",
    midpoint = 10
  )
```



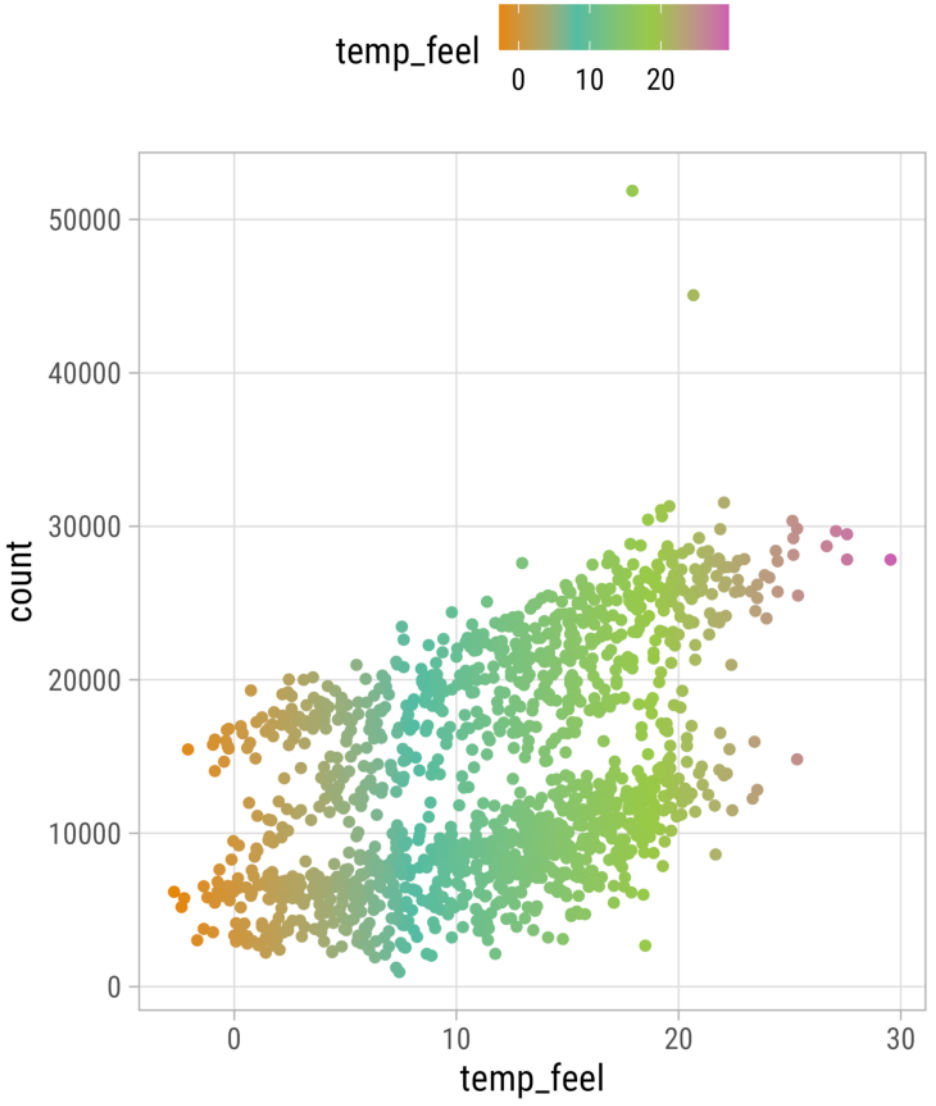
Create Diverging Palettes

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradient2(
    low = "#663399",
    high = "#993334",
    mid = "grey92",
    midpoint = 10,
    limits = c(-10, 30)
  )
```



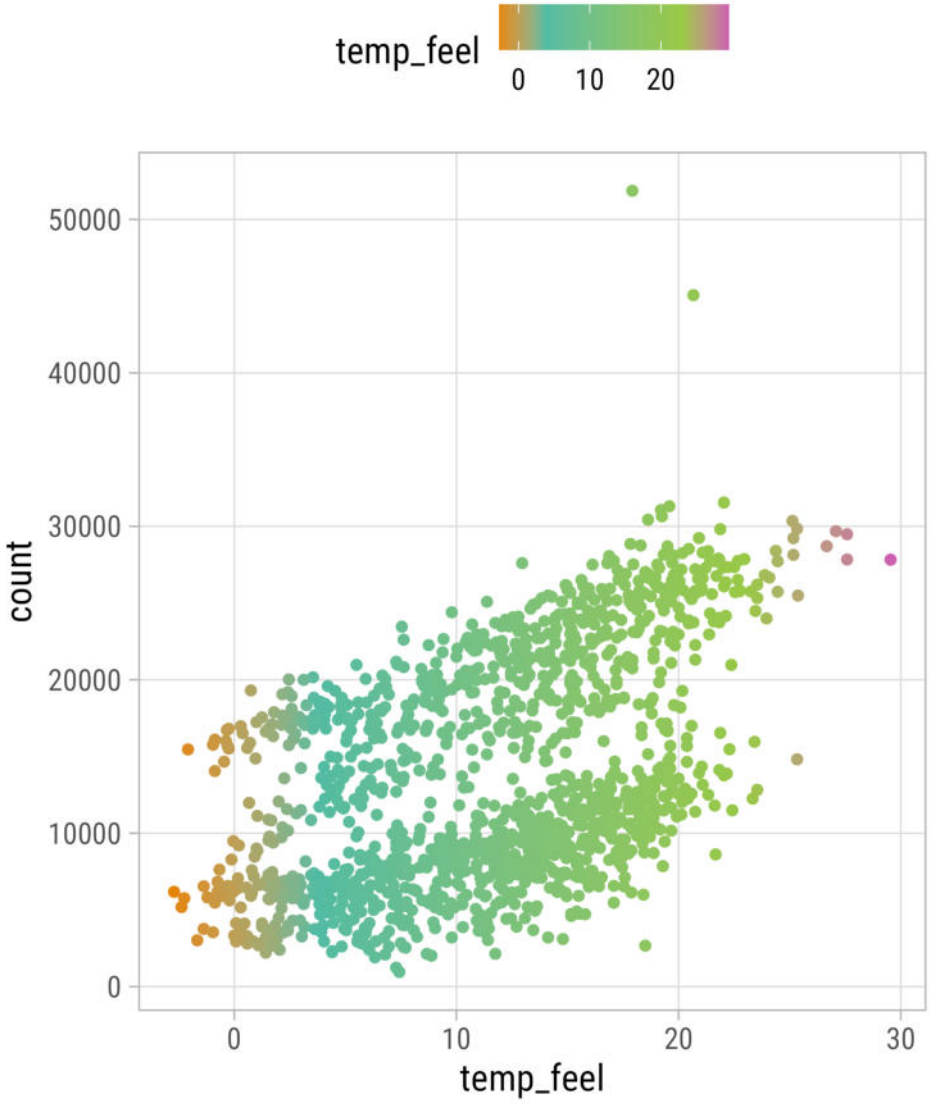
Create Any Palette

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradientn(
    colors = carto_custom
  )
```



Create Any Palette

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradientn(
    colors = carto_custom,
    values = c(0, .2, .8, 1)
  )
```



Build Your Own `scale_color|fill_*()`

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

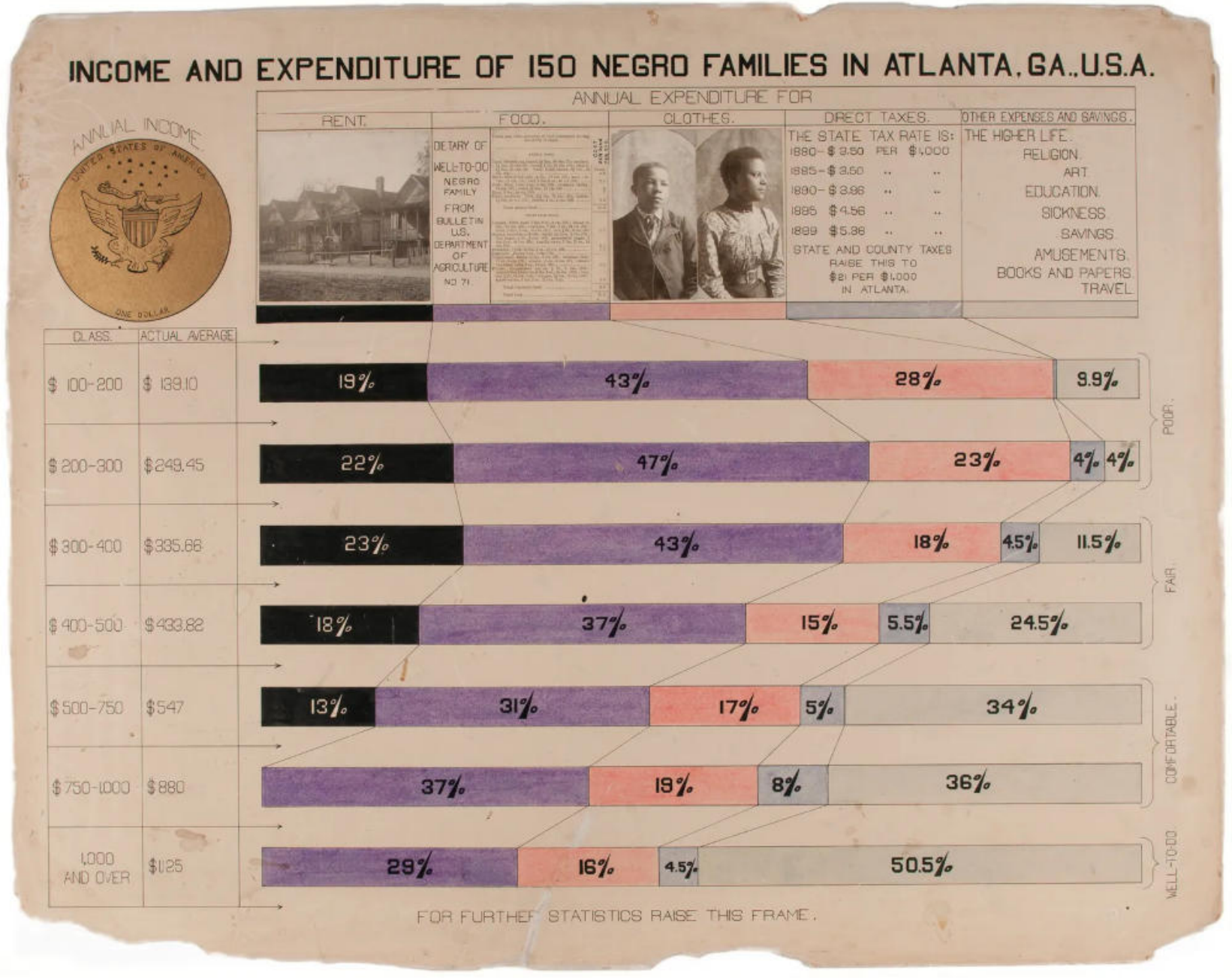
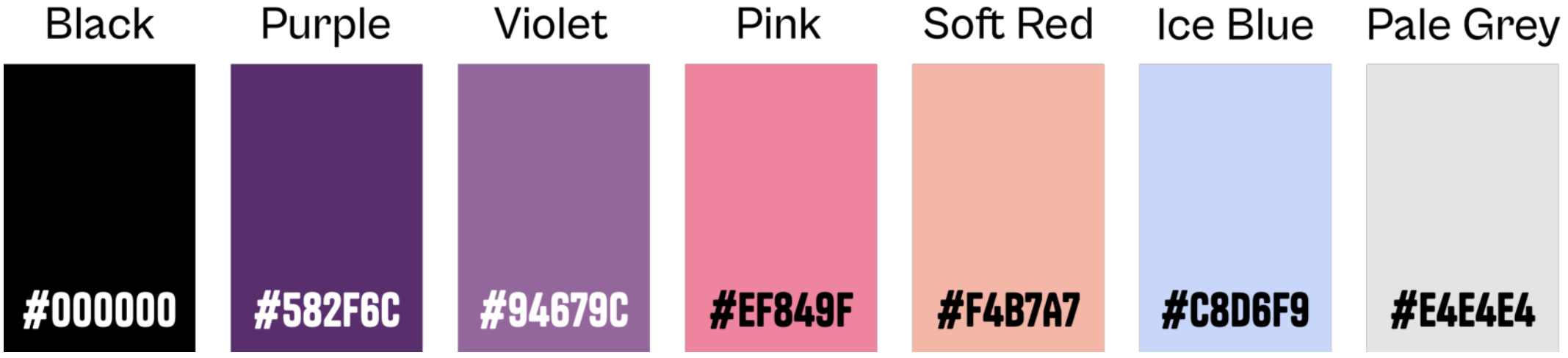


Illustration by W. E. B. Du Bois, Courtesy Library of Congress


Black
#000000
Purple
#582F6C
Violet
#94679C
Pink
#EF849F
Soft Red
#F4B7A7
Ice Blue
#C8D6F9
Pale Grey
#E4E4E4

Build Your Own `scale_color|fill_*()`

```
dubois_colors <- function(...) {
  dubois_cols <- c(
    `black`    = "#000000",
    `purple`   = "#582f6c",
    `violet`   = "#94679C",
    `pink`     = "#ef849f",
    `softred`  = "#f4b7a7",
    `iceblue`  = "#bccbf3",
    `palegrey` = "#e4e4e4"
  )

  cols <- c(...)

  if (is.null(cols))
    return (dubois_cols)

  dubois_cols[cols]
}

```

```
    black      pink   softred   iceblue
"#000000" "#ef849f" "#f4b7a7" "#bccbf3"
```

Build Your Own `scale_color|fill_*_d()`

```
 1 dubois_pal_d <- function(palette = "default", reverse = FALSE) {
 2   function(n) {
 3     if(n > 5) stop('Palettes only contains 5 colors')
 4
 5     if (palette == "default") { pal <- dubois_colors("black", "violet", "softred", "iceblue", "paleg
 6     if (palette == "dark") { pal <- dubois_colors(1:5)[1:n] }
 7     if (palette == "light") { pal <- dubois_colors(3:7)[1:n] }
 8
 9     pal <- unname(pal)
10
11     if (reverse) rev(pal) else pal
12   }
13 }
14
15 dubois_pal_d()(3)
```

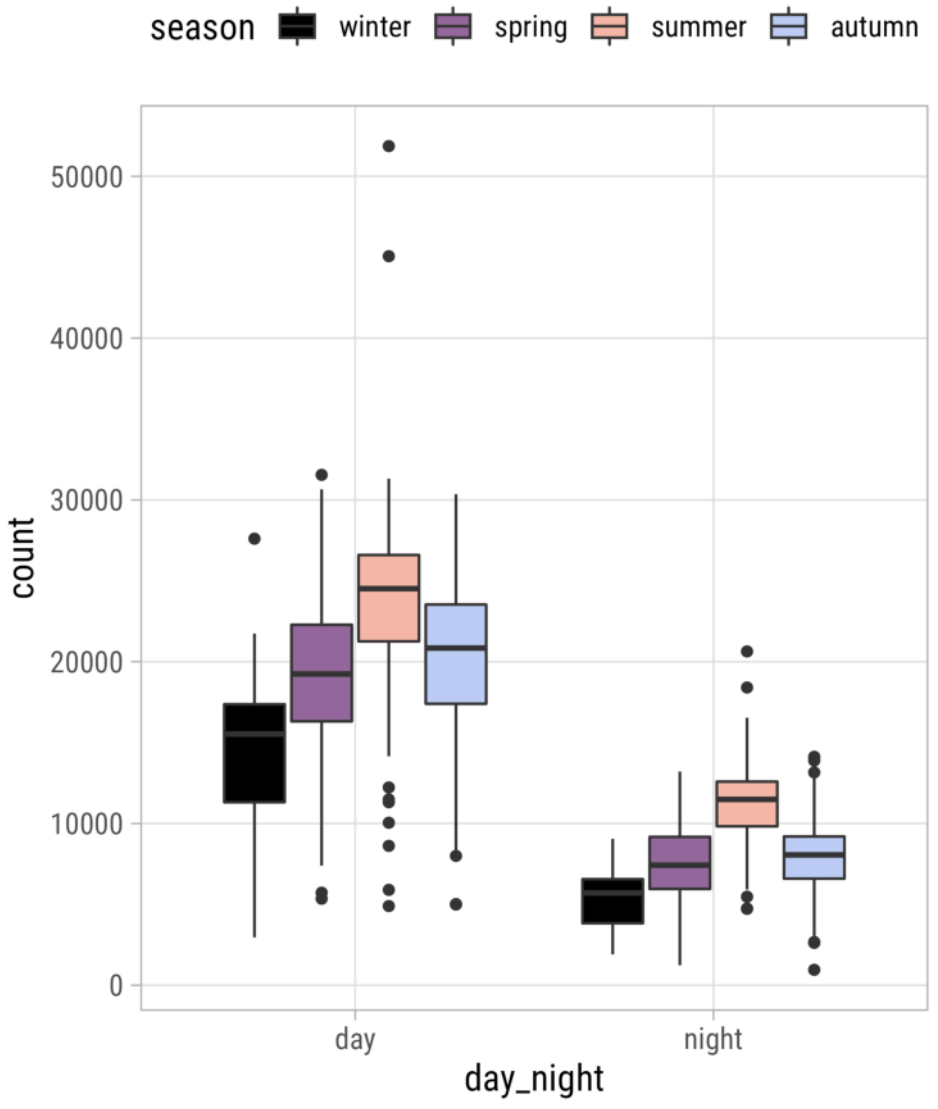
```
[1] "#000000" "#94679C" "#f4b7a7"
```

Build Your Own `scale_fill|color_*_d()`

```
1  scale_color_dubois_d <- function(palette = "default", reverse = FALSE, ...) {
2    if (!palette %in% c("default", "dark", "light")) stop('Palette should be "default", "dark" or "lig
3
4    pal <- dubois_pal_d(palette = palette, reverse = reverse)
5
6    ggplot2::discrete_scale("colour", paste0("dubois_", palette), palette = pal, ...)
7  }
8
9  scale_fill_dubois_d <- function(palette = "default", reverse = FALSE, ...) {
10   if (!palette %in% c("default", "dark", "light")) stop('Palette should be "default", "dark" or "lig
11
12   pal <- dubois_pal_d(palette = palette, reverse = reverse)
13
14   ggplot2::discrete_scale("fill", paste0("dubois_", palette), palette = pal, ...)
15 }
```

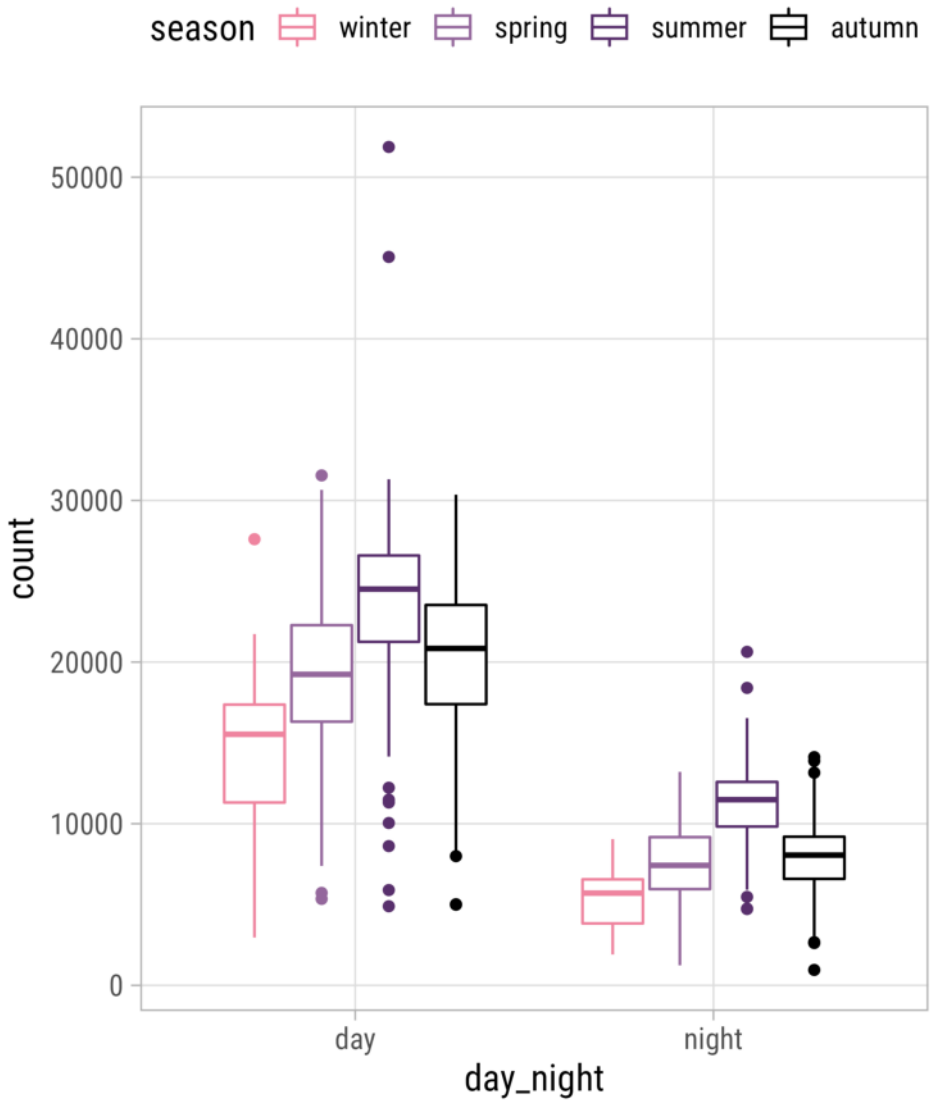
Use Your Own `scale_fill_*_d()`

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



Use Your Own `scale_color_*_d()`

```
ggplot(
    bikes,
    aes(x = day_night, y = count,
        color = season)
  ) +
  geom_boxplot() +
  scale_color_dubois_d(
    palette = "dark",
    reverse = TRUE
  )
```

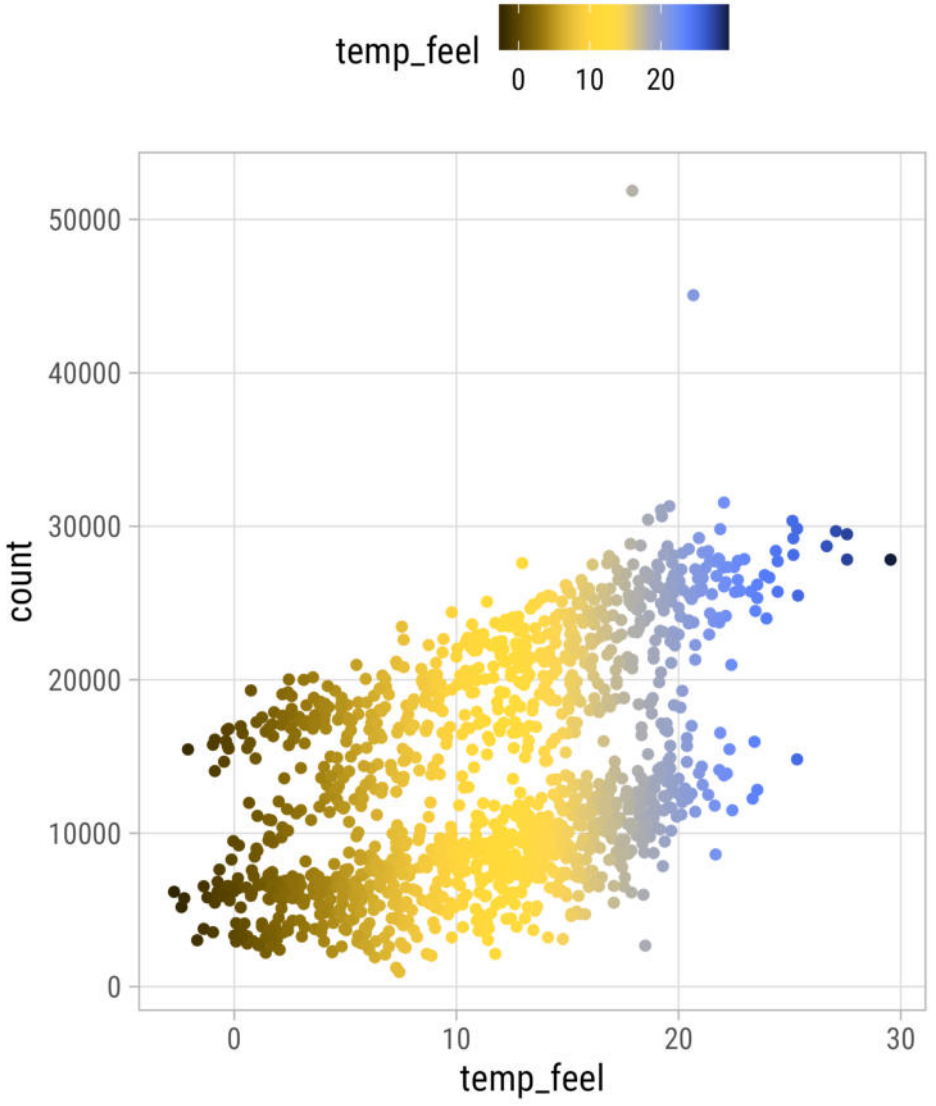


Test Your Palettes

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

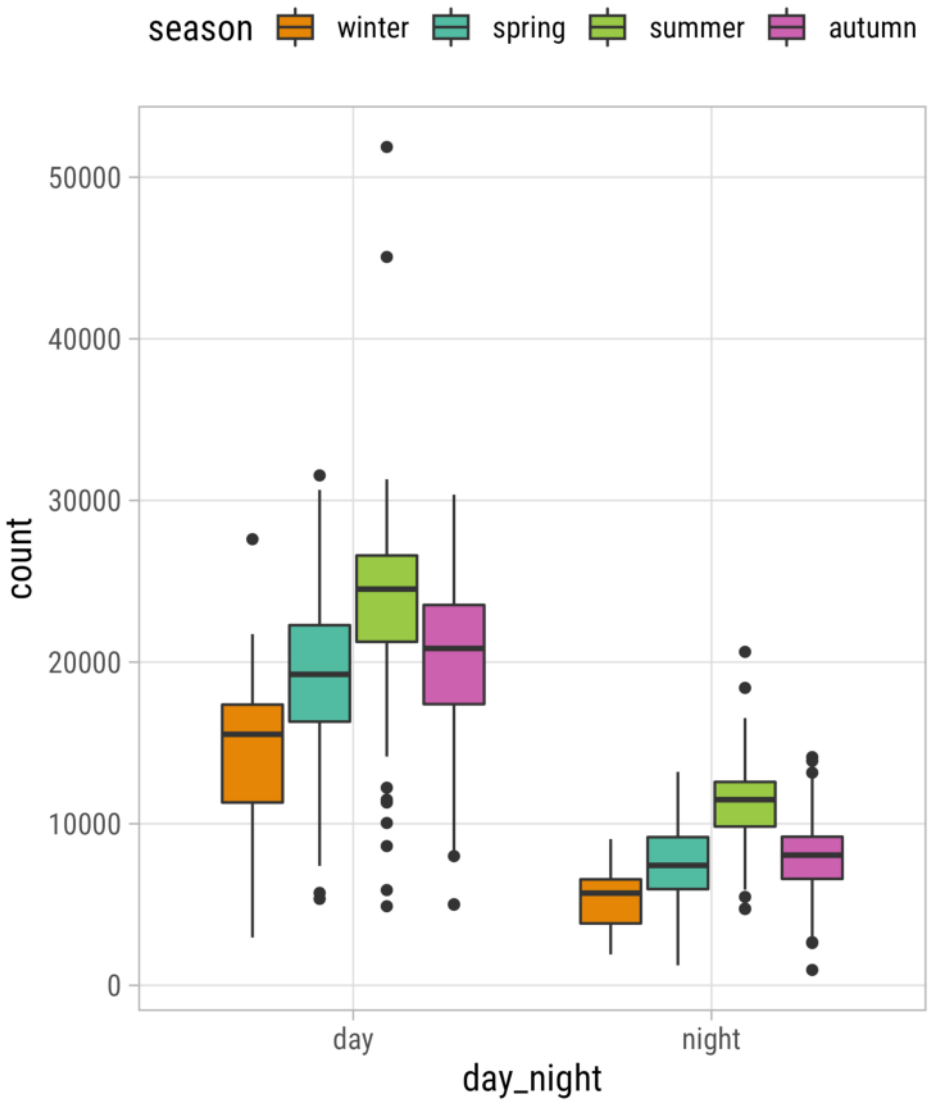
Emulate CVD

```
deut <-
  colorspace::deutan(
    viridis::turbo(
      n = 100, direction = -1
    )
  )

ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_gradientn(
    colors = deut
  )
```



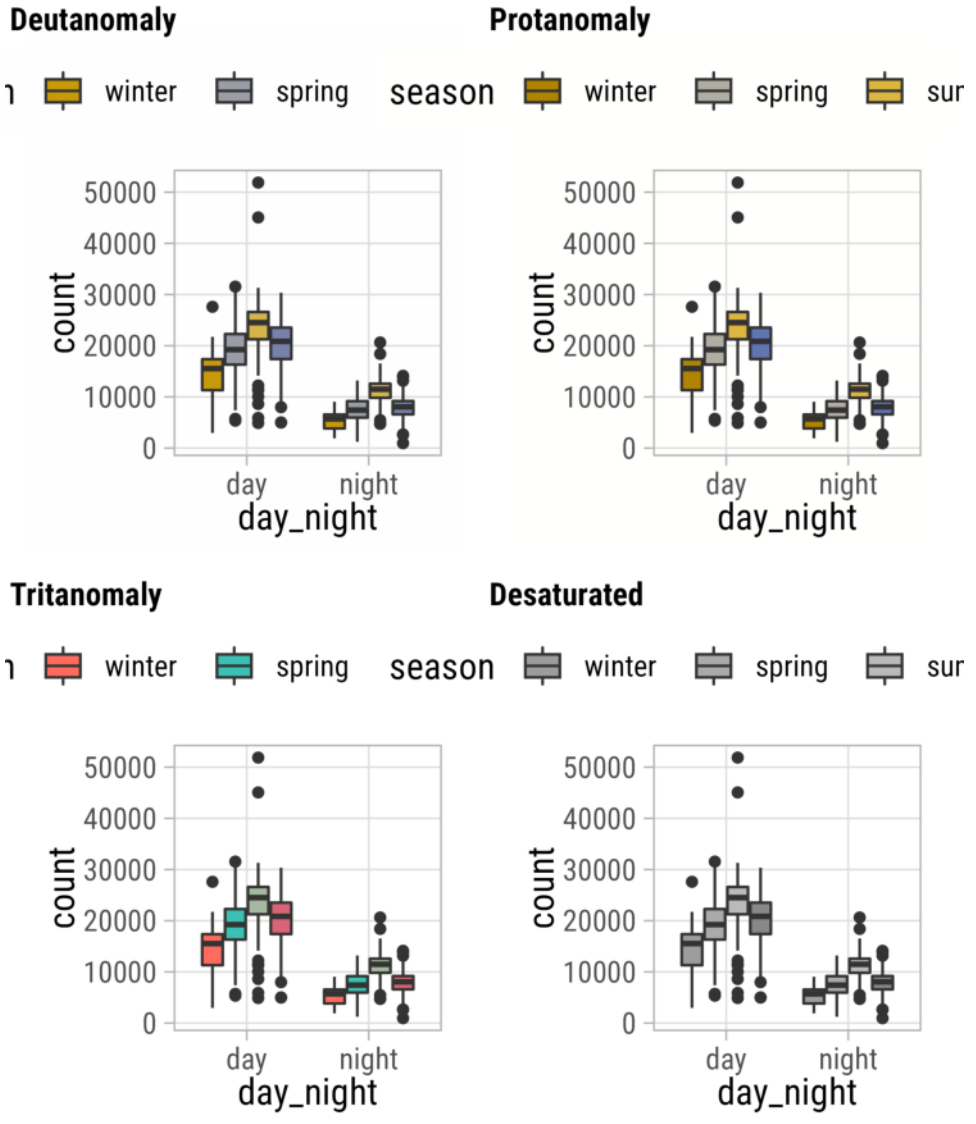
Emulate CVD

```
g <-
  ggplot(
    bikes,
    aes(x = day_night, y = count,
        fill = season)
  ) +
  geom_boxplot() +
  scale_fill_manual(
    values = carto_custom
  )

g
```



Emulate CVD

```
# devtools::install_github(
#   "clauswilke/colorblindr"
# )

colorblindr::cvd_grid(g)
```

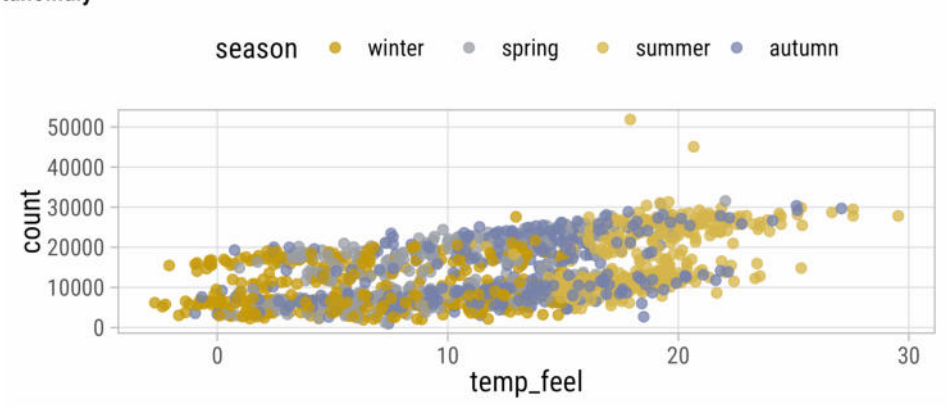


Emulate CVD

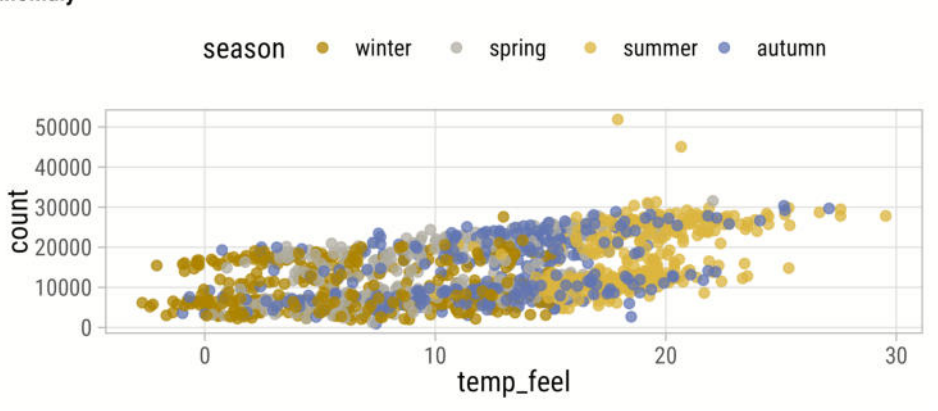
Emulate CVD

```
# devtools::install_github("clauswilke/colorblindr")

colorblindr::cvd_grid(g2)
```

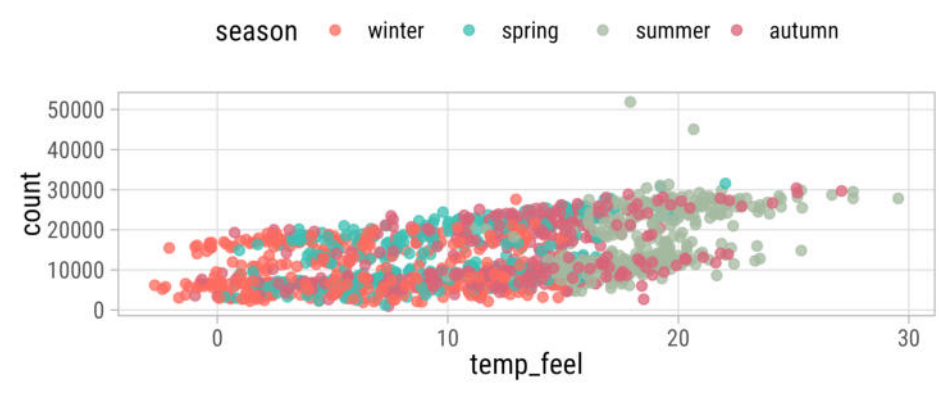
Deutanomaly



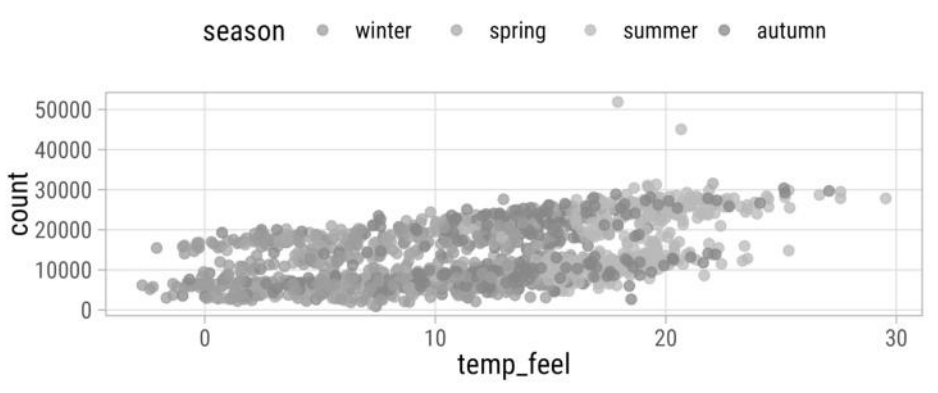
Protanomaly



Tritanomaly



Desaturated

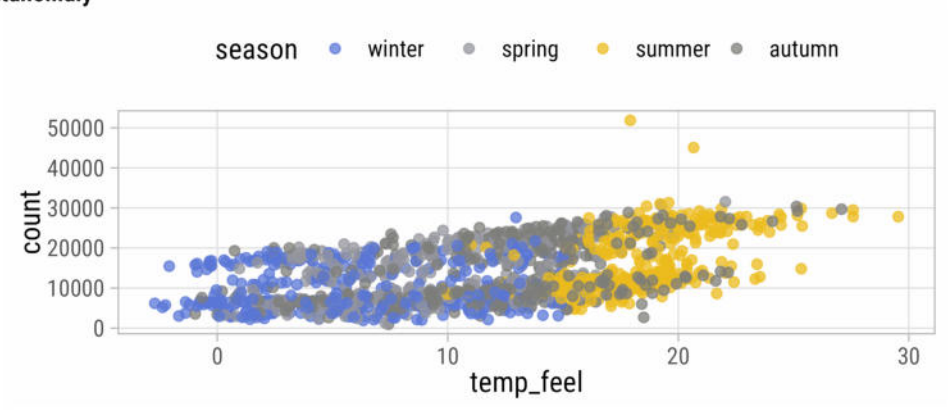


Emulate CVD

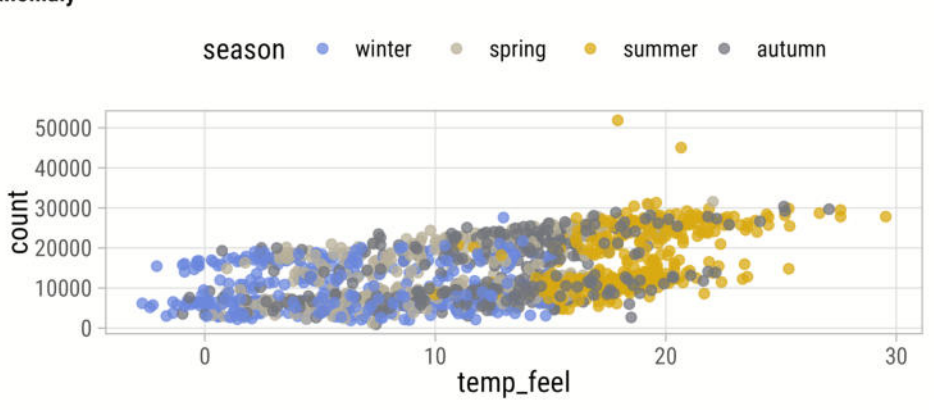
Emulate CVD

```
# devtools::install_github("clauswilke/colorblindr")

colorblindr::cvd_grid(g3)
```

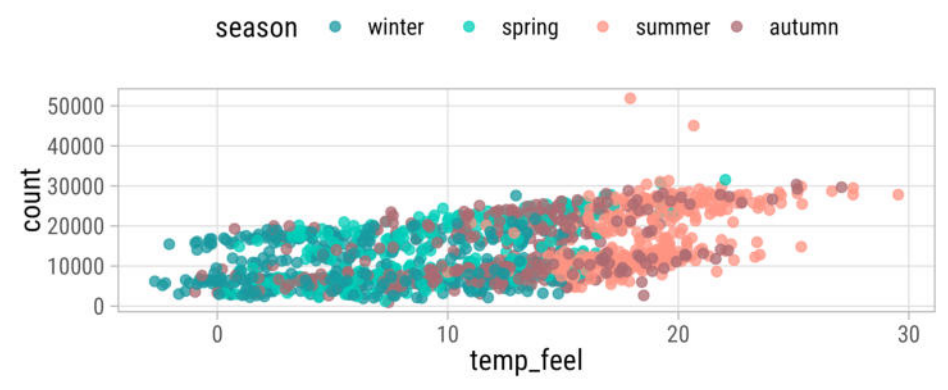
Deutanomaly



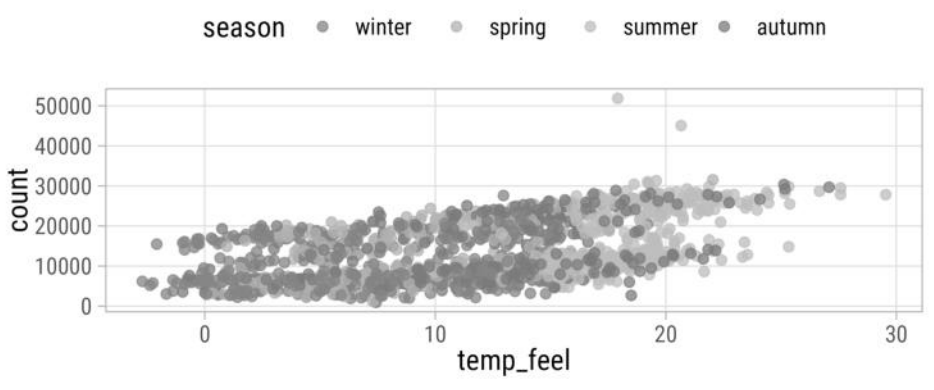
Protanomaly



Tritanomaly



Desaturated



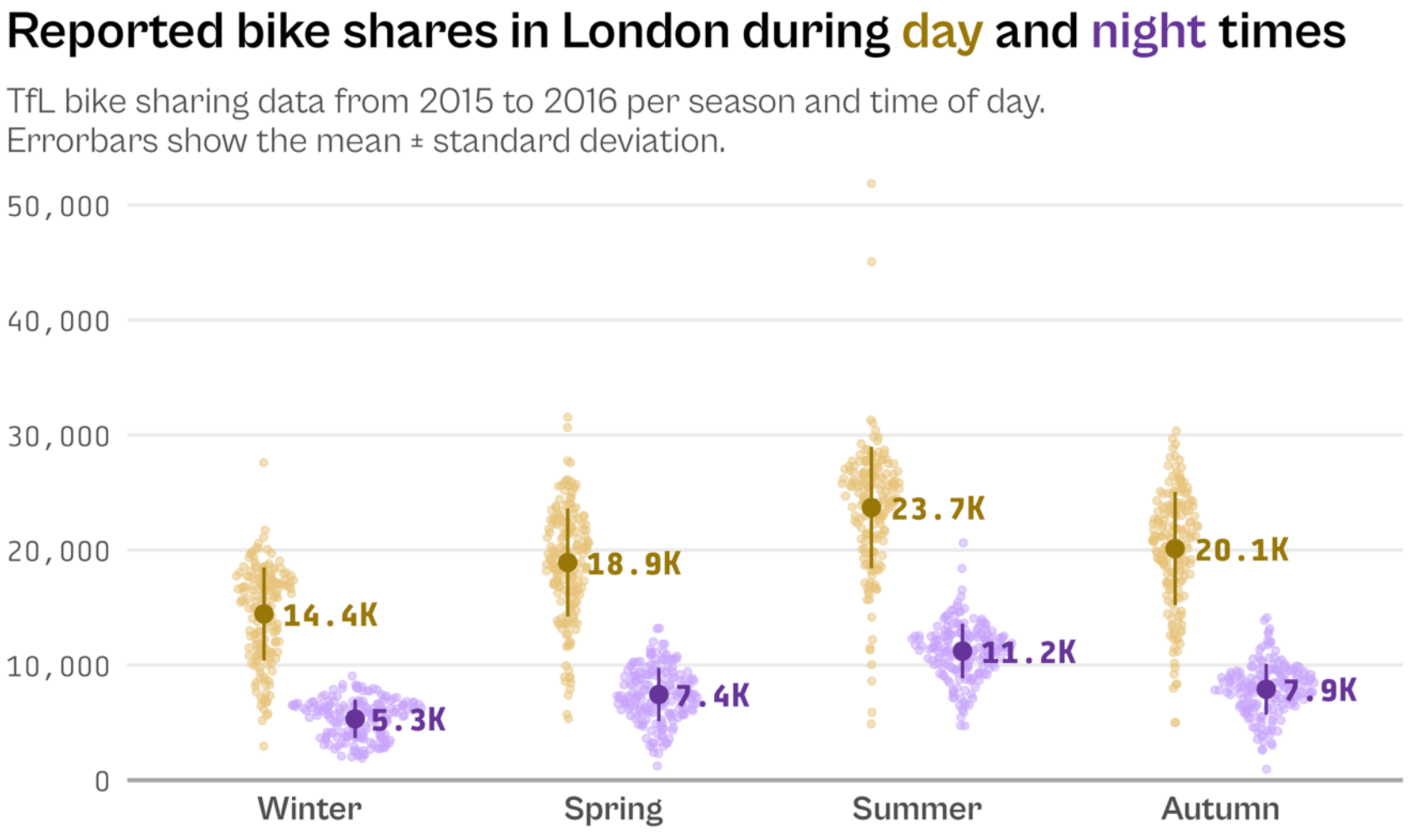
Recap

- use **categorical** palettes for **qualitative** data
  - e.g. `scale_*_discrete()` and `scale_*_manual()` for custom options
  - e.g. `scale_*_viridis_d` and `scale_*_brewer()` for pre-defined options
- use **sequential or diverging** palettes for **quantitative** data
  - e.g. `scale_*_gradient()` or `scale_*_gradient2()` for custom options
  - e.g. `scale_*_viridis_c` and `scale_*_distiller()` for pre-defined options
- various packages provide palettes incl. `scale_*` components
  - e.g. `{rcartocolors}`, `{scico}`, `{ggsci}`, `{ggthemes}`, `{nord}`
- those and even more packages return palettes as vectors
  - modify and supply thehm to `scale_*_manual()` and `scale_*_gradientn()`
- use **`after_scale`** to modify and recycle color scales

Exercise

Exercise

- Create a similar visualization as close as possible:



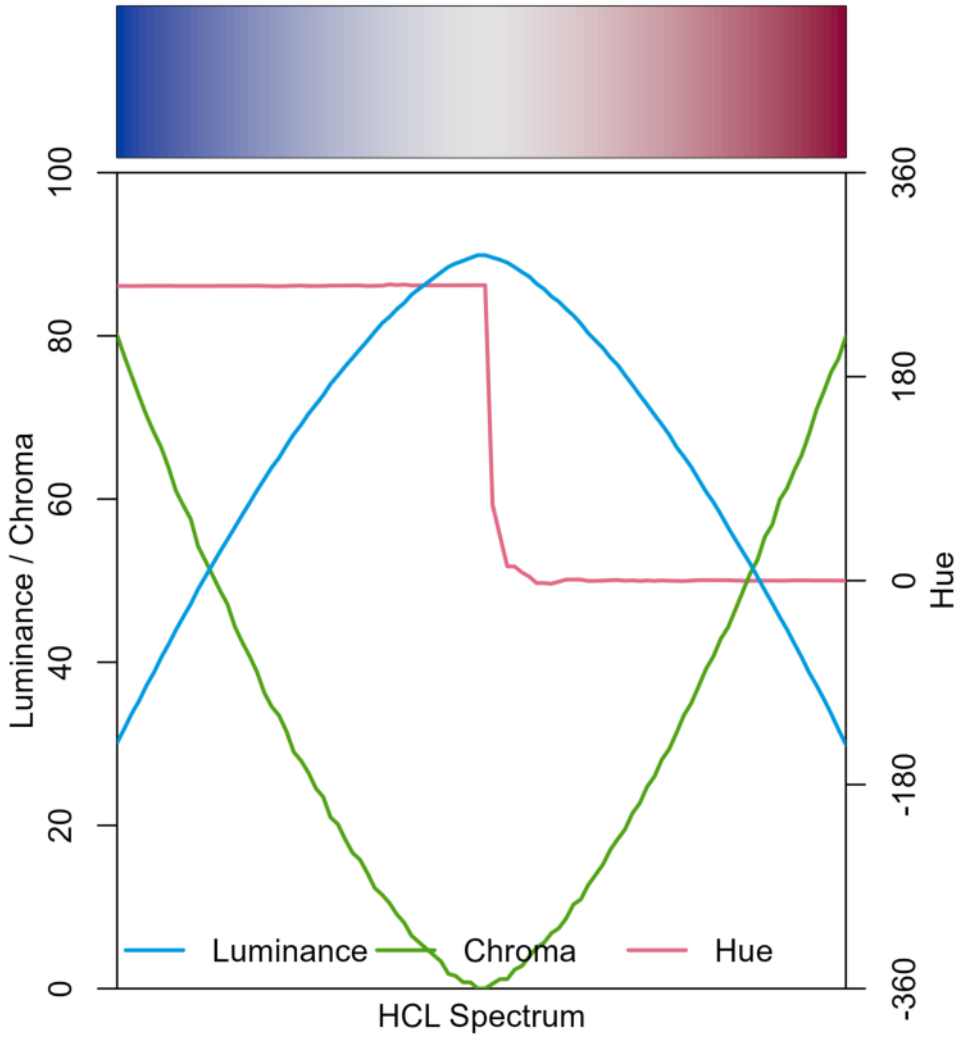
Appendix

HCL Spectrum

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

Evaluate HCL Spectrum

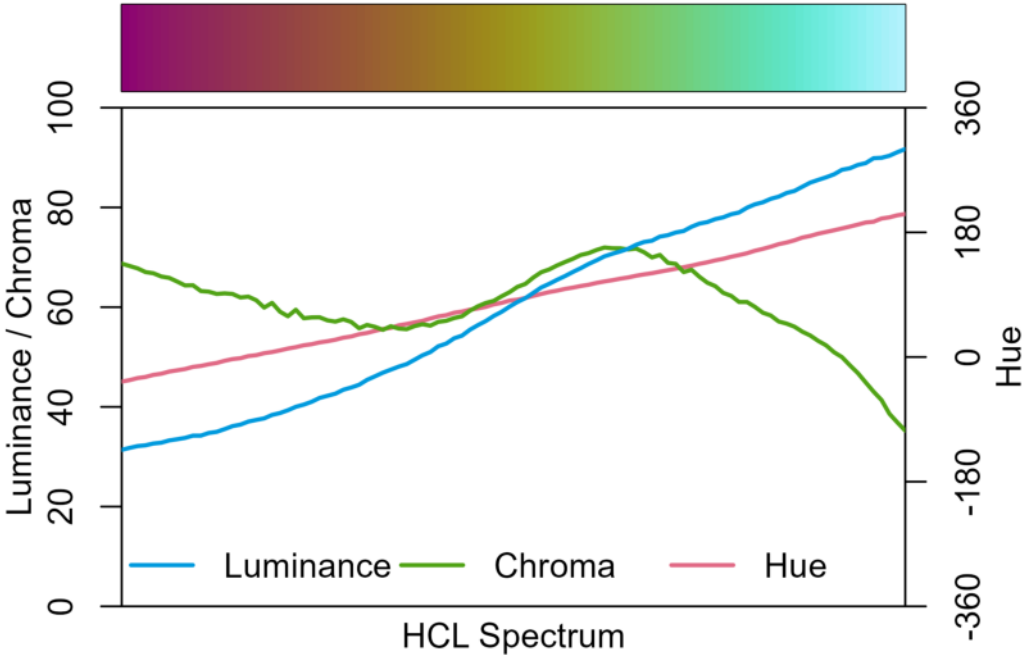
```
colorspace::specplot(
  colorspace::diverging_hcl(
    n = 100, palette = "Blue-Red"
  )
)
```

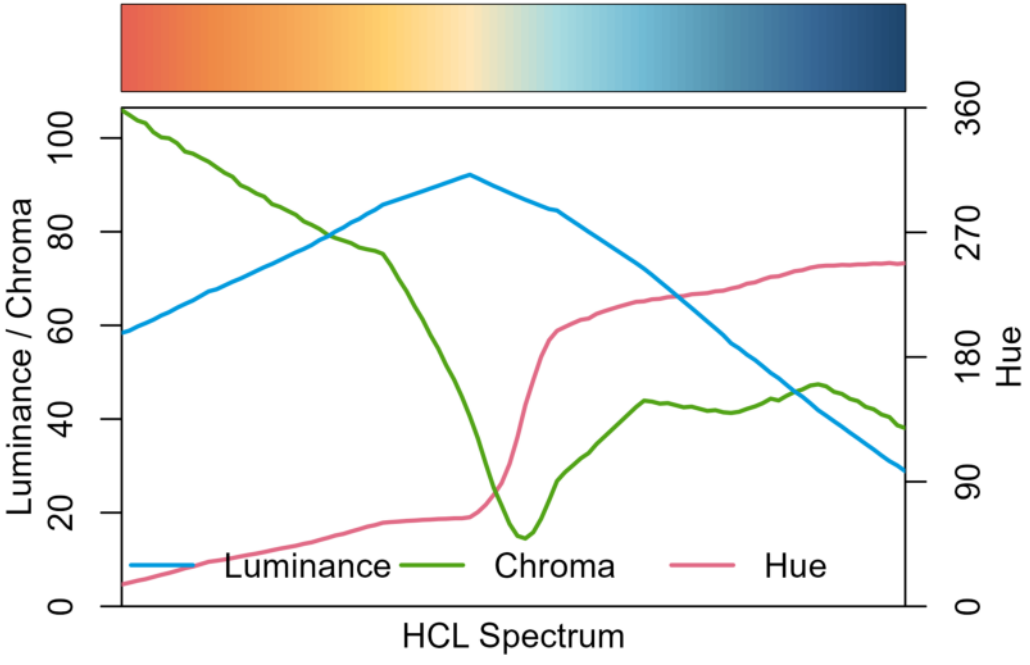


Evaluate HCL Spectrum

```
1  colorspace::specplot(
2    scico::scico(
3      n = 100, palette = "hawaii"
4    )
5  )
```

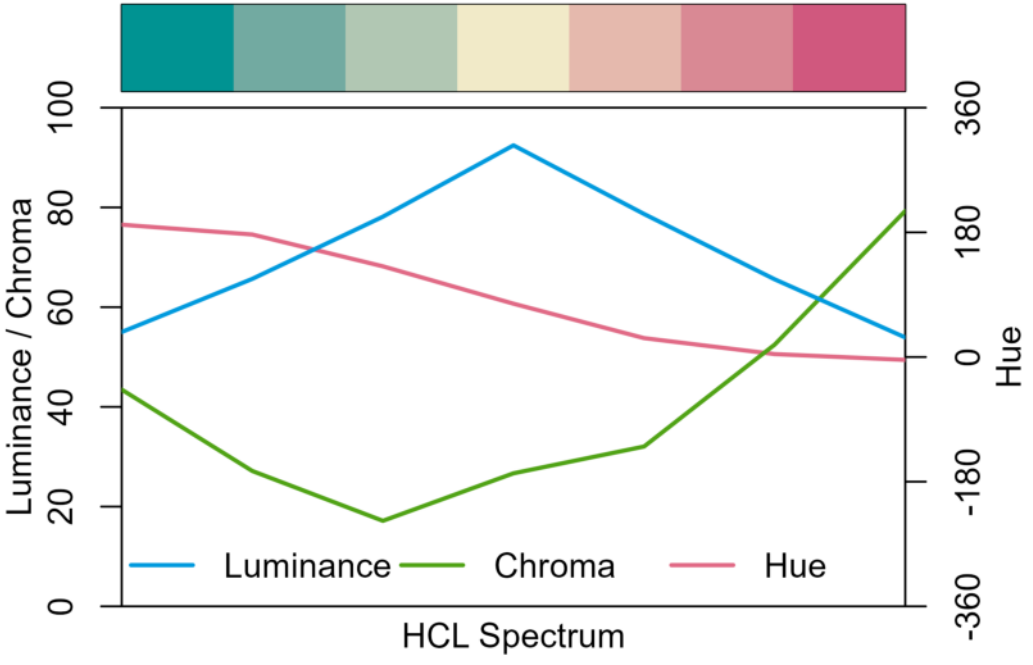
```
1  colorspace::specplot(
2    MetBrewer::met.brewer(
3       n = 100, name = "Hiroshige"
4    )
5  )
```



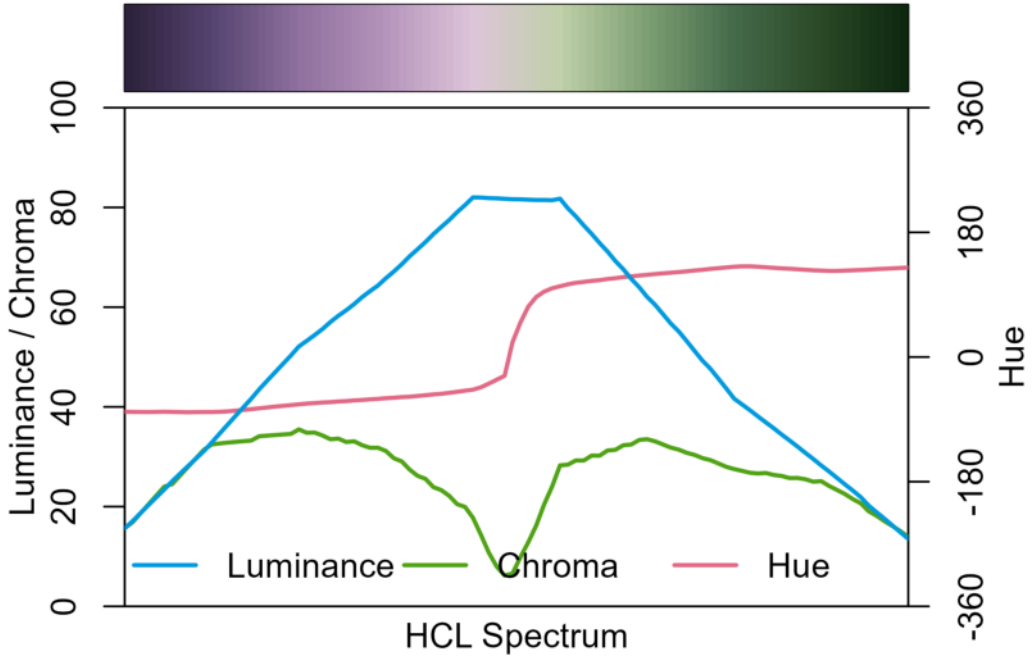


Evaluate HCL Spectrum

```
colorspace::specplot(
  rcartocolor::carto_pal(
    n = 7, name = "TealRose"
  )
)
```

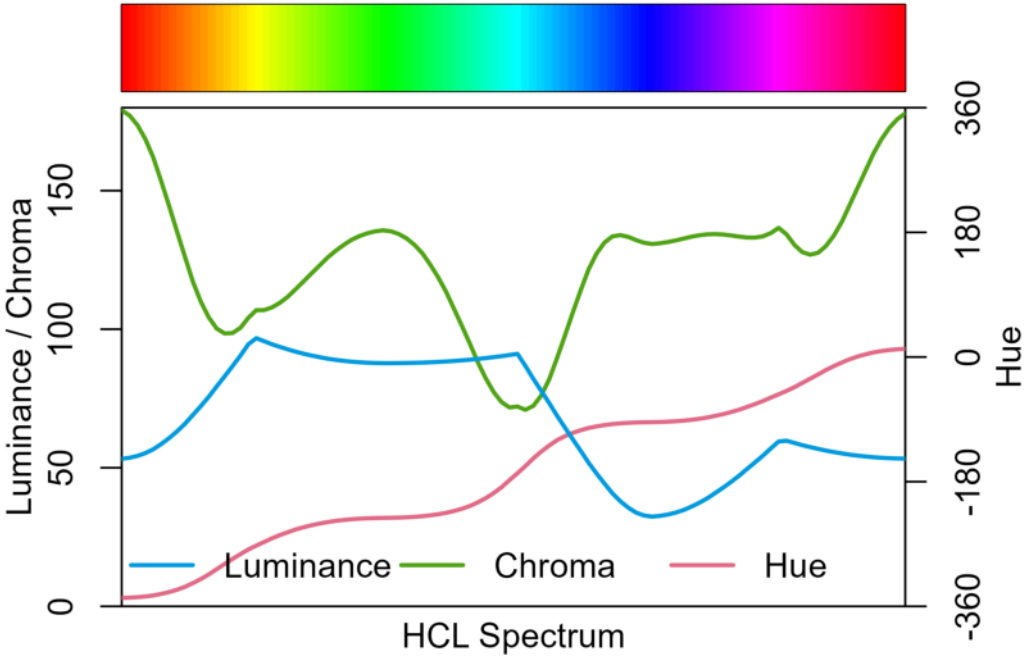



```
colorspace::specplot(
  MetBrewer::met.brewer(
    n = 100, name = "Cassatt2"
  )
)
```

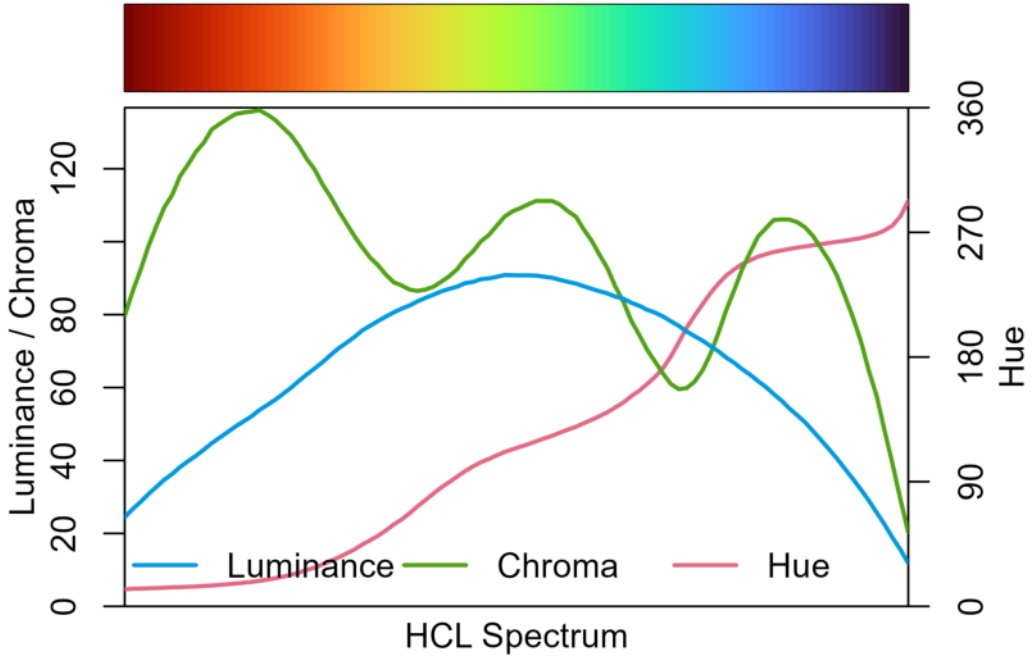


Evaluate HCL Spectrum

```
colorspace::specplot(
  rainbow(
    n = 100
  )
)
```




```
colorspace::specplot(
  viridis::turbo(
     n = 100, direction = -1
  )
)
```



Build Your Own `scale_color|fill_*_c()`

```
dubois_pal_c <- function(palette = "dark", reverse = FALSE, ...) {
  dubois_palettes <- list(
    `dark`    = dubois_colors("black", "purple", "violet", "pink"),
    `light`   = dubois_colors("purple", "violet", "pink", "palered")
  )

  pal <- dubois_palettes[[palette]]
  pal <- unname(pal)

  if (reverse) pal <- rev(pal)

  grDevices::colorRampPalette(pal, ...)
}

dubois_pal_c(palette = "light", reverse = TRUE)(3)
```

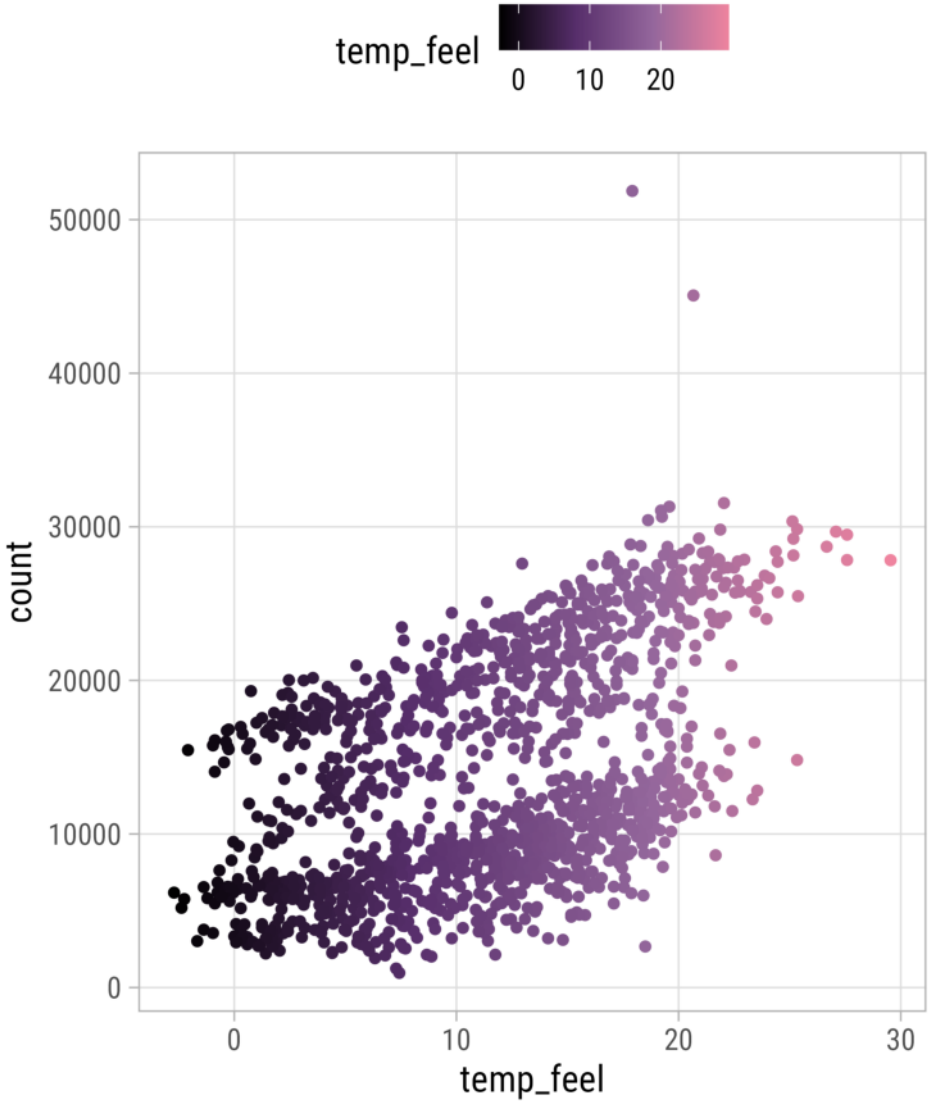
```
[1] "#FFFFFF" "#C1759D" "#582F6C"
```

Build Your Own `scale_color|fill_*_c()`

```
1  scale_fill_dubois_c <- function(palette = "dark", reverse = FALSE, ...) {
2    if (!palette %in% c("dark", "light")) stop('Palette should be "dark" or "light".')
3
4    pal <- dubois_pal_c(palette = palette, reverse = reverse)
5
6    ggplot2::scale_fill_gradientn(colours = pal(256), ...)
7  }
8
9  scale_color_dubois_c <- function(palette = "dark", reverse = FALSE, ...) {
10   if (!palette %in% c("dark", "light")) stop('Palette should be "dark" or "light".')
11
12   pal <- dubois_pal_c(palette = palette, reverse = reverse)
13
14   ggplot2::scale_color_gradientn(colours = pal(256), ...)
15 }
```

Use Your Own `scale_color|fill_*_c()`

```
1  ggplot(
2      bikes,
3      aes(x = temp_feel, y = count,
4          color = temp_feel)
5    ) +
6    geom_point() +
7    scale_color_dubois_c()
```



Use Your Own `scale_color|fill_*_c()`

```
ggplot(
    bikes,
    aes(x = temp_feel, y = count,
        color = temp_feel)
  ) +
  geom_point() +
  scale_color_dubois_c(
    palette = "light",
    reverse = TRUE
  )
```