

# Data Visualization in R with **ggplot2**

The Structure of ggplot2 (Part 5)

**Cédric Scherer**

Physalia Courses | November 9-13 2020

Photo by Richard Strozyński

Part 5

# Annotations

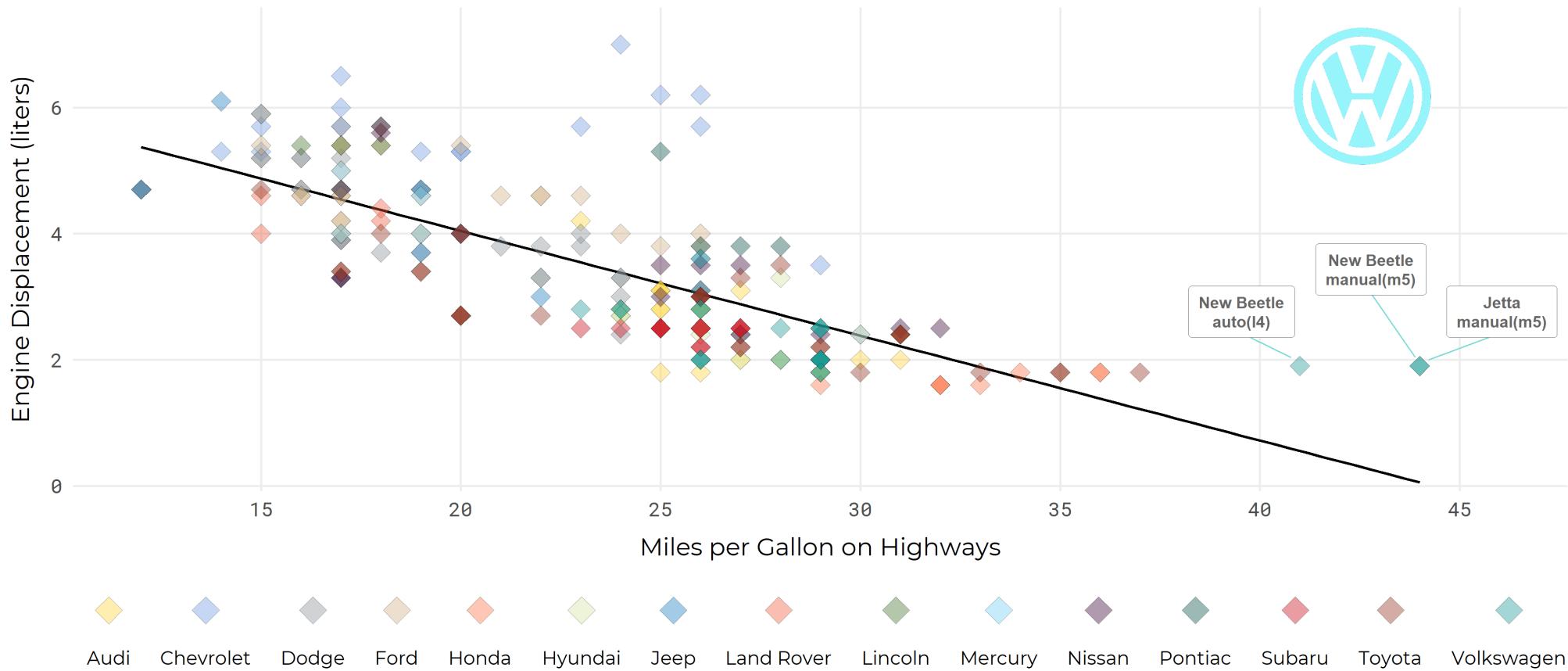
# ggplot2: Build a data MASTERpiece



Illustration by Allison Horst ([github.com/allisonhorst/stats-illustrations](https://github.com/allisonhorst/stats-illustrations))

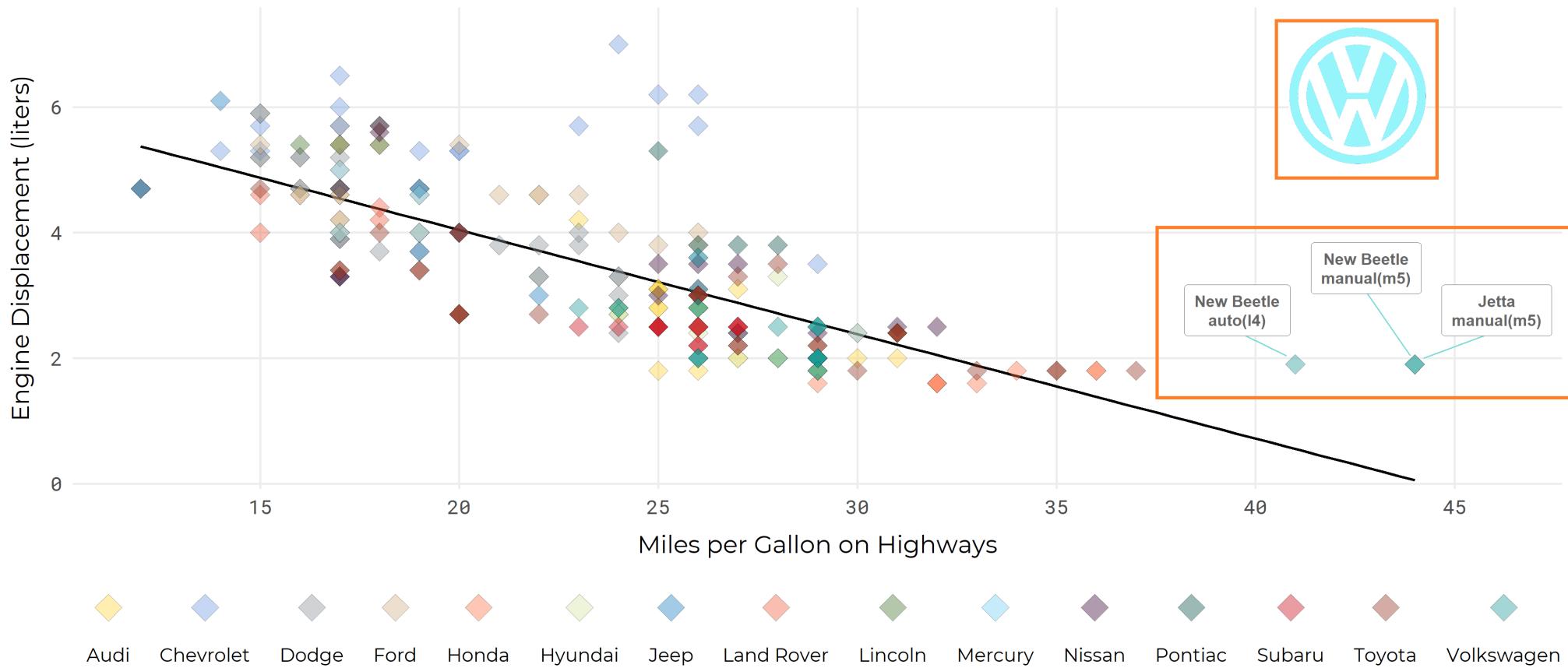
# "New Beetle" and "Jetta" by Volkswagen have the worst car efficiency

Linear regression of engine displacement versus highway miles per gallon (MPG) for 38 popular models of cars from 1999 to 2008.



## "New Beetle" and "Jetta" by Volkswagen have the worst car efficiency

Linear regression of engine displacement versus highway miles per gallon (MPG) for 38 popular models of cars from 1999 to 2008.



# The Structure of `ggplot2`

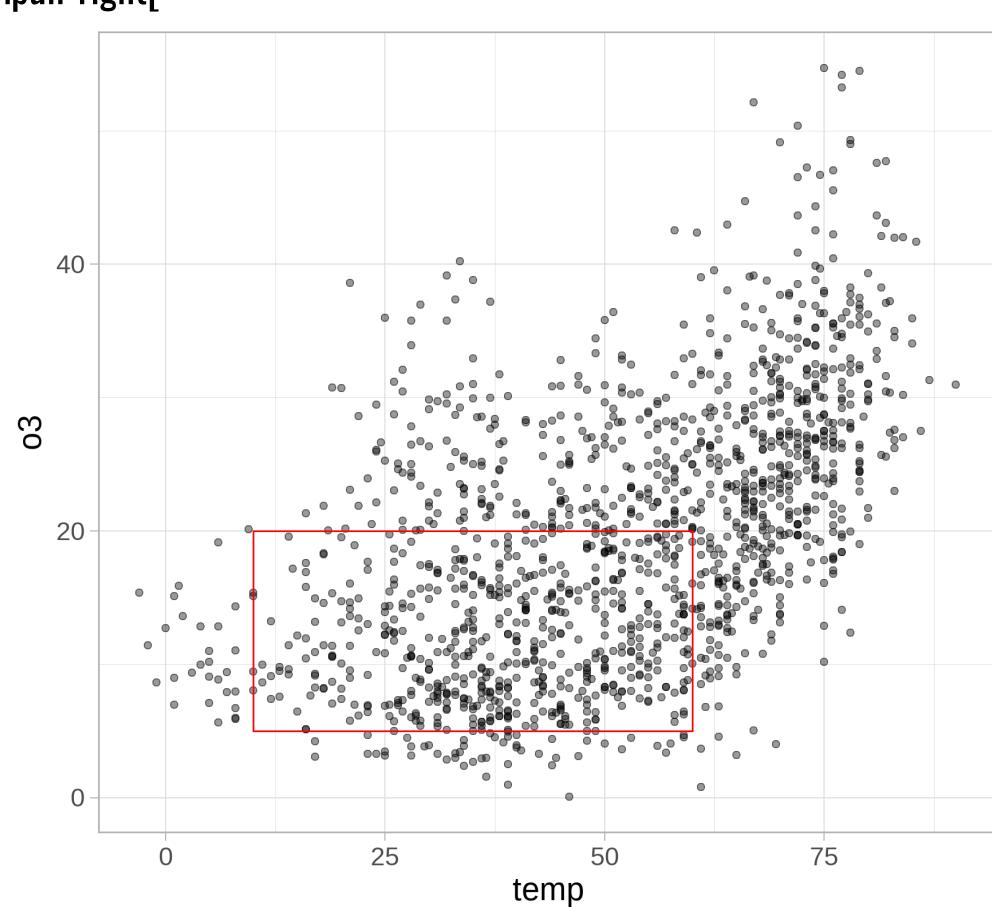
| Layer             | Function  | Explanation   |
|-------------------|---|---|
| Data              | <code>ggplot(data)</code>                         | The raw data that you want to visualise.                              |
| Aesthetics        | <code>aes()</code>                                | Aesthetic mappings of the geometric and statistical objects.          |
| Layers            | <code>geom_*</code> () and <code>stat_*</code> () | The geometric shapes and statistical summaries representing the data. |
| Scales            | <code>scale_*</code> ()                           | Maps between the data and the aesthetic dimensions.                   |
| Coordinate System | <code>coord_*</code> ()                           | Maps data into the plane of the data rectangle.                       |
| Facets            | <code>facet_*</code> ()                           | The arrangement of the data into a grid of plots.                     |
| Visual Themes     | <code>theme()</code> and <code>theme_*</code> ()  | The overall visual defaults of a plot.                                |
| Annotations       | <code>annotate()</code>                           | Add additional labels, geometries or images to a plot.                |

# Annotations

`annotate()` and  
`annotation_custom()`

# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:



# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

```
ggplot(chic, aes(temp, o3)) +  
  geom_point(alpha = .4) +  
  annotate(  
    geom = "rect",  
    xmin = 60,  
    xmax = Inf,  
    ymin = -Inf,  
    ymax = Inf,  
    fill = "red",  
    alpha = .5  
)
```

# annotate()

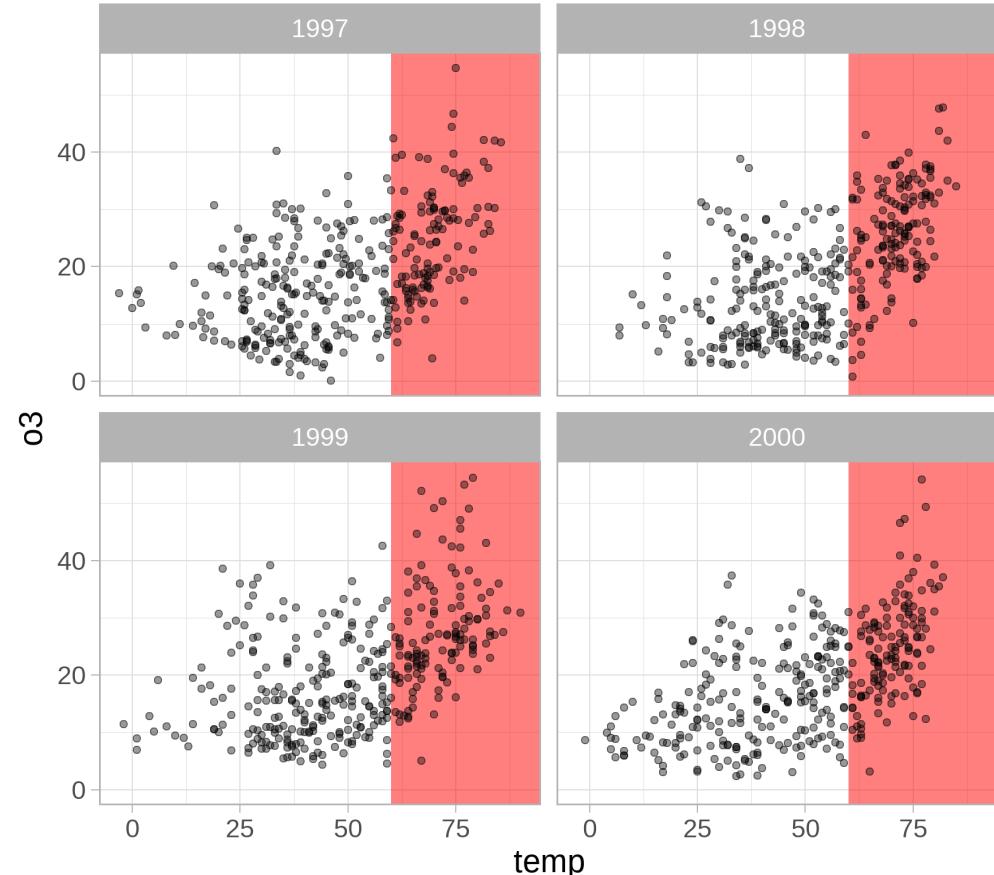
The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

```
ggplot(chic, aes(temp, o3)) +  
  annotate(  
    geom = "rect",  
    xmin = 60,  
    xmax = Inf,  
    ymin = -Inf,  
    ymax = Inf,  
    fill = "red",  
    alpha = .5  
  ) +  
  geom_point(alpha = .4)
```

# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

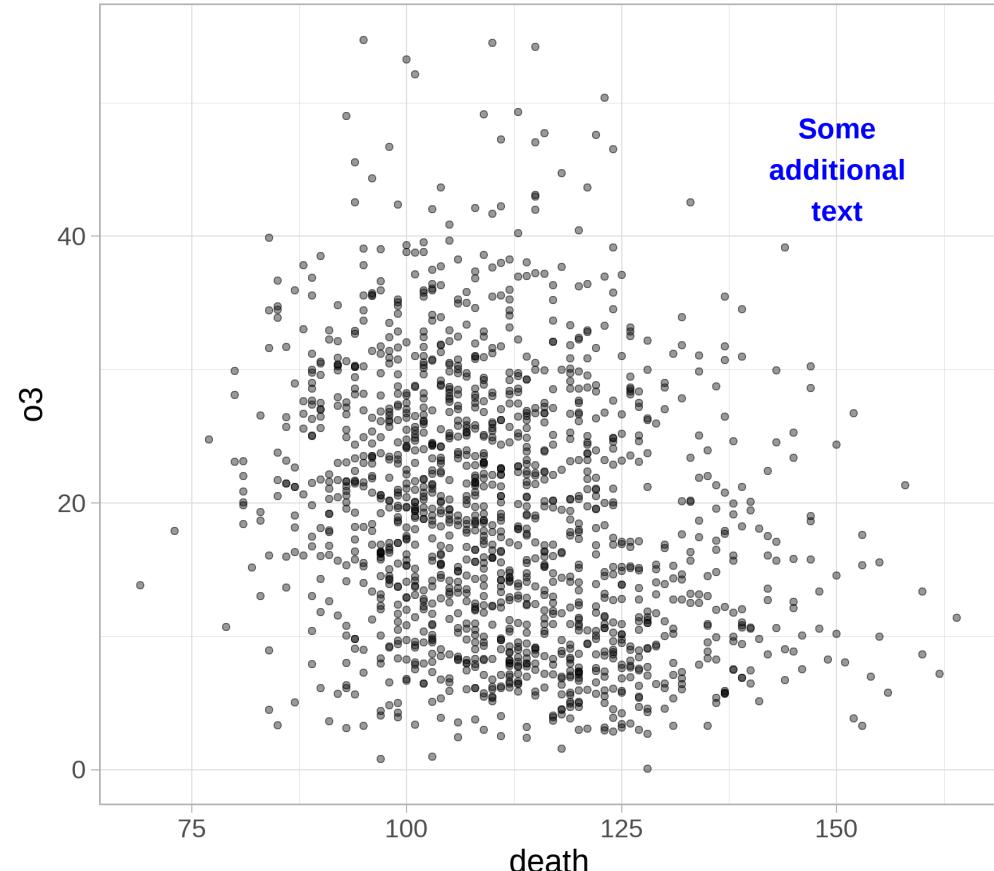
```
ggplot(chic, aes(temp, o3)) +  
  annotate(  
    geom = "rect",  
    xmin = 60,  
    xmax = Inf,  
    ymin = -Inf,  
    ymax = Inf,  
    fill = "red",  
    alpha = .5  
  ) +  
  geom_point(alpha = .4) +  
  facet_wrap(~ year)
```



# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

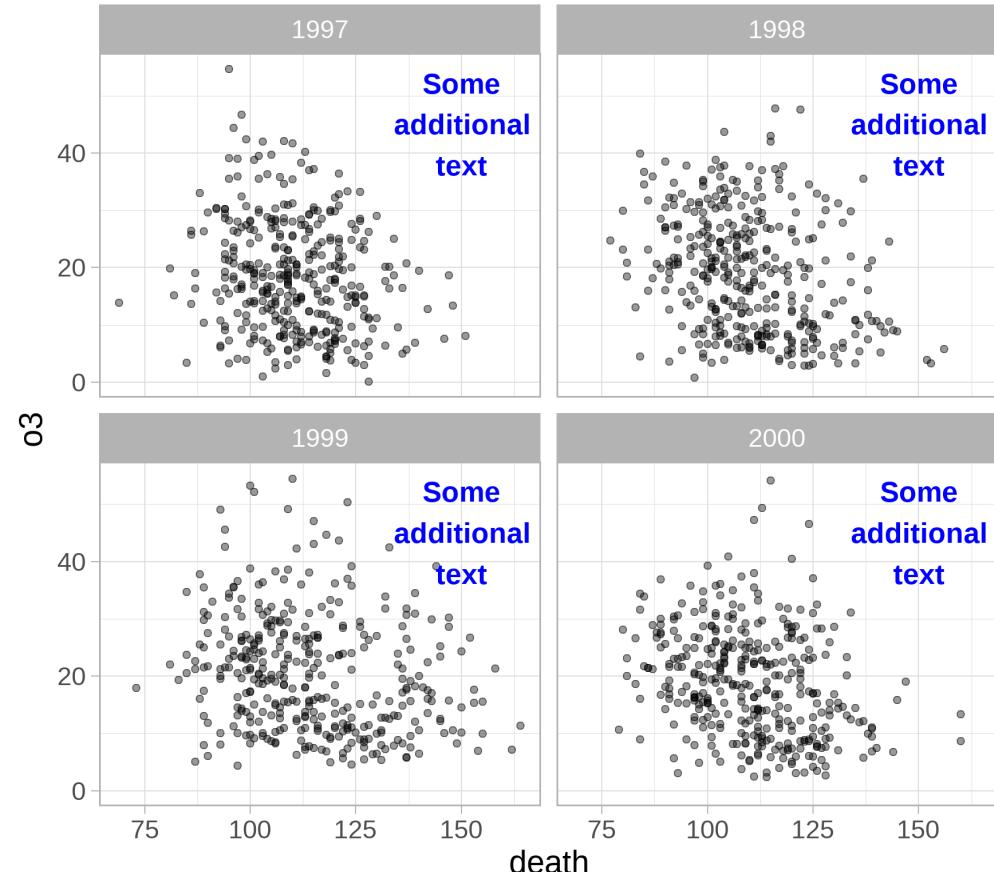
```
ggplot(chic, aes(death, o3)) +  
  geom_point(alpha = .4) +  
  annotate(  
    geom = "text",  
    x = 150,  
    y = 45,  
    label = "Some\nadditional\ntext",  
    size = 5,  
    color = "blue",  
    fontface = "bold"  
)
```



# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

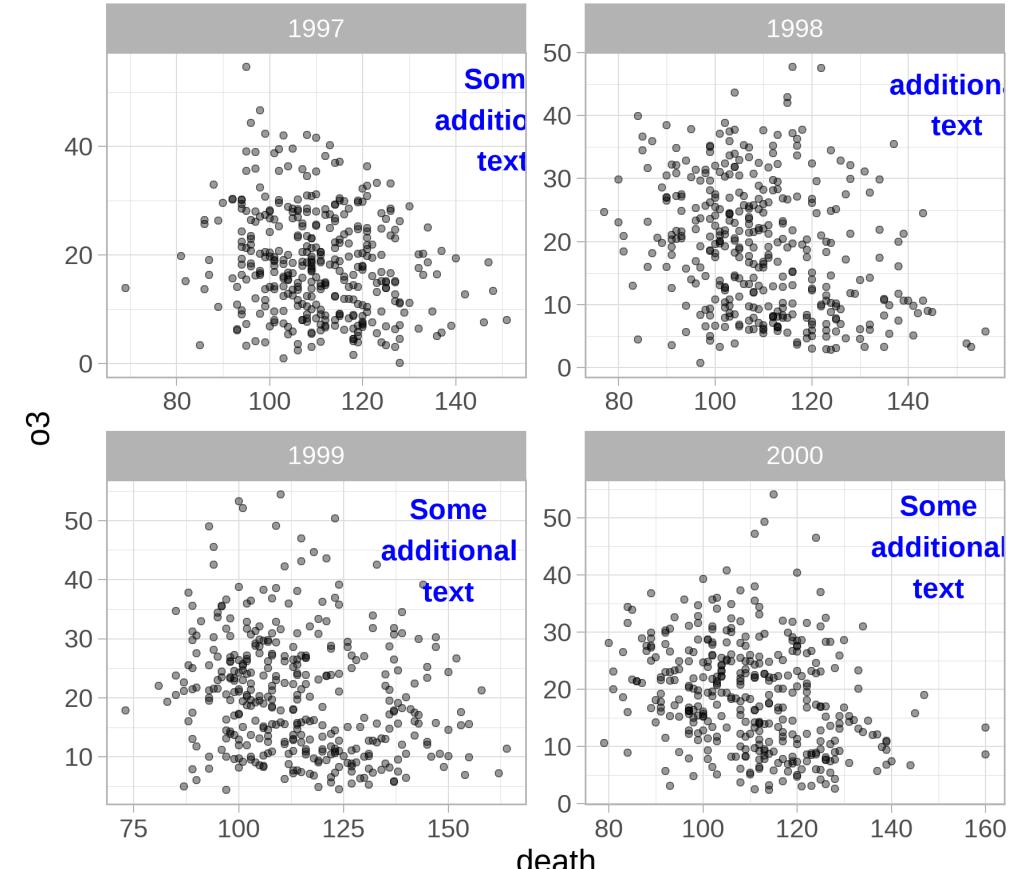
```
ggplot(chic, aes(death, o3)) +  
  geom_point(alpha = .4) +  
  annotate(  
    geom = "text",  
    x = 150,  
    y = 45,  
    label = "Some\nadditional\nntext",  
    size = 5,  
    color = "blue",  
    fontface = "bold"  
  ) +  
  facet_wrap(~ year)
```



# annotate()

The `annotate()` functions allows to add geoms to a plot without mapping variables to aesthetics:

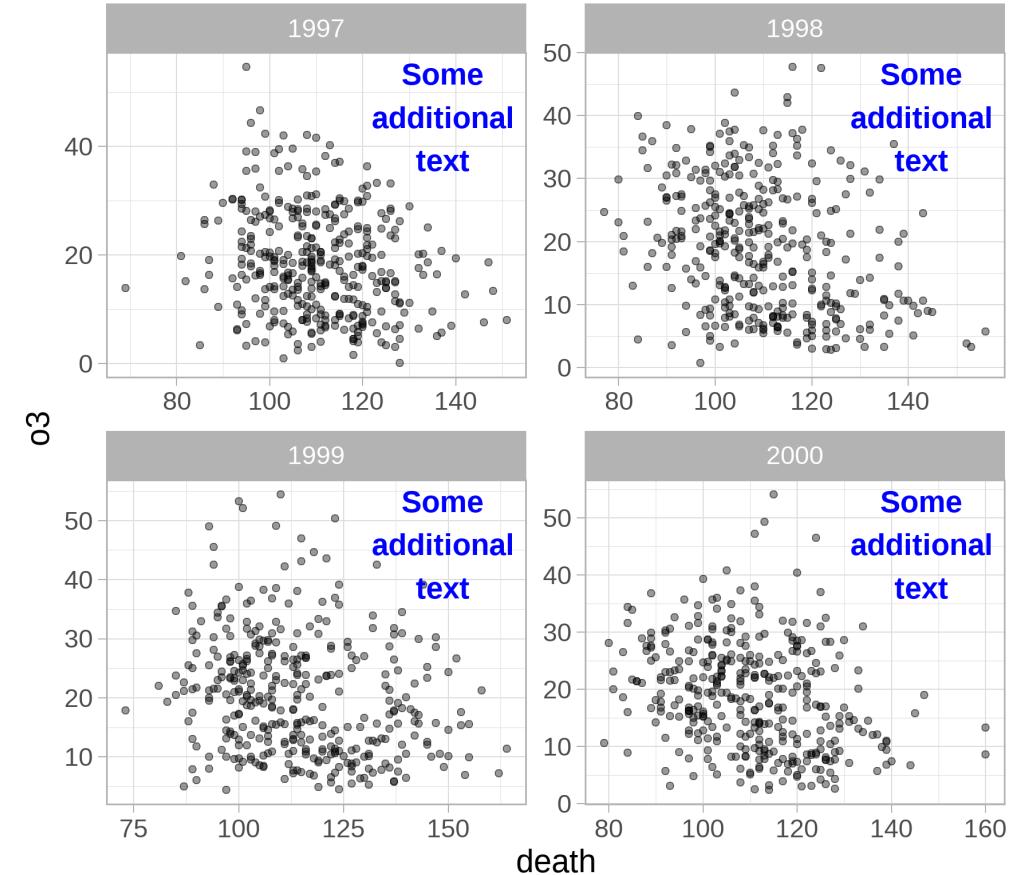
```
ggplot(chic, aes(death, o3)) +  
  geom_point(alpha = .4) +  
  annotate(  
    geom = "text",  
    x = 150,  
    y = 45,  
    label = "Some\nadditional\ntext",  
    size = 5,  
    color = "blue",  
    fontface = "bold"  
  ) +  
  facet_wrap(~ year,  
            scales = "free")
```



# annotation\_custom()

`annotation_custom()` allows to add static annotations which are not affected by scales:

```
ggplot(chic, aes(death, o3)) +  
  geom_point(alpha = .4) +  
  annotation_custom(  
    grid::textGrob(  
      x = 0.8,  
      y = 0.8,  
      label = "Some\\nadditional\\ntext",  
      gp = grid::gpar(  
        col = "blue",  
        fontsize = 15,  
        fontface = "bold"  
      )  
    )  
  ) +  
  facet_wrap(~ year,  
            scales = "free")
```

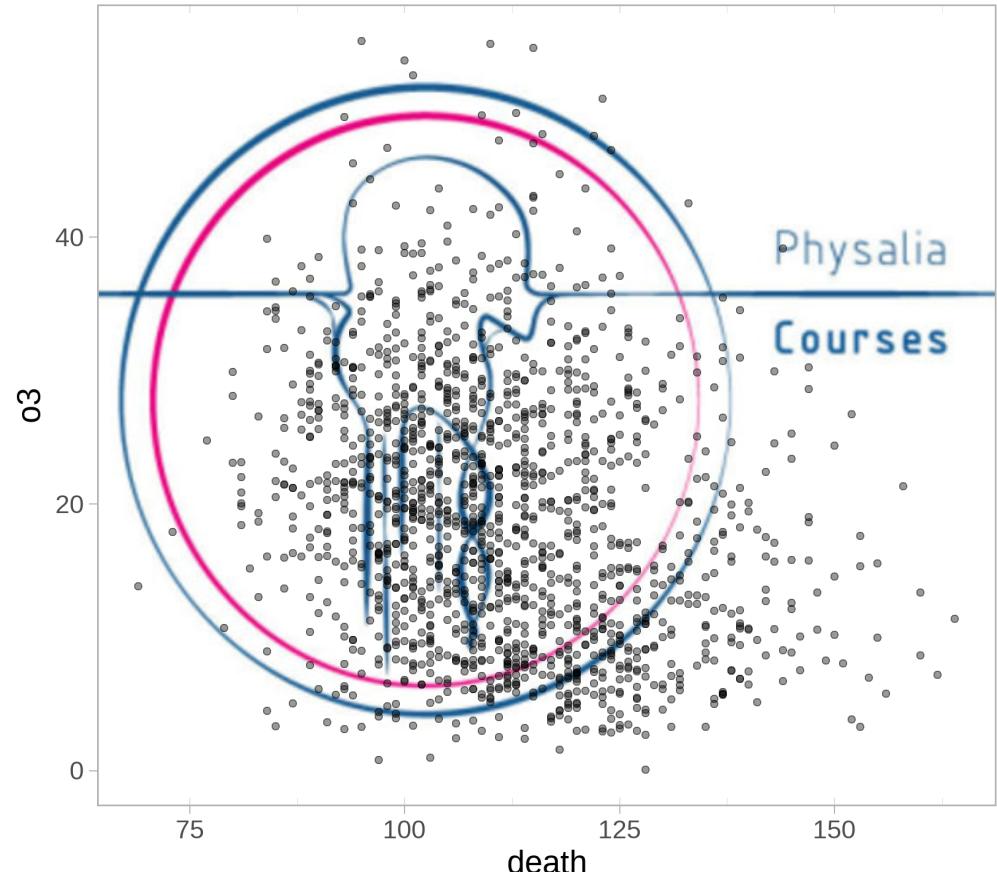


# annotation\_custom()

... and `annotation_custom()` allows also to add images:

```
library(png)

ggplot(chic, aes(death, o3)) +
  annotation_custom(
    grid::rasterGrob(
      image = readPNG(
        "img/physalia.png"
      )
    )
  ) +
  geom_point(alpha = .4)
```

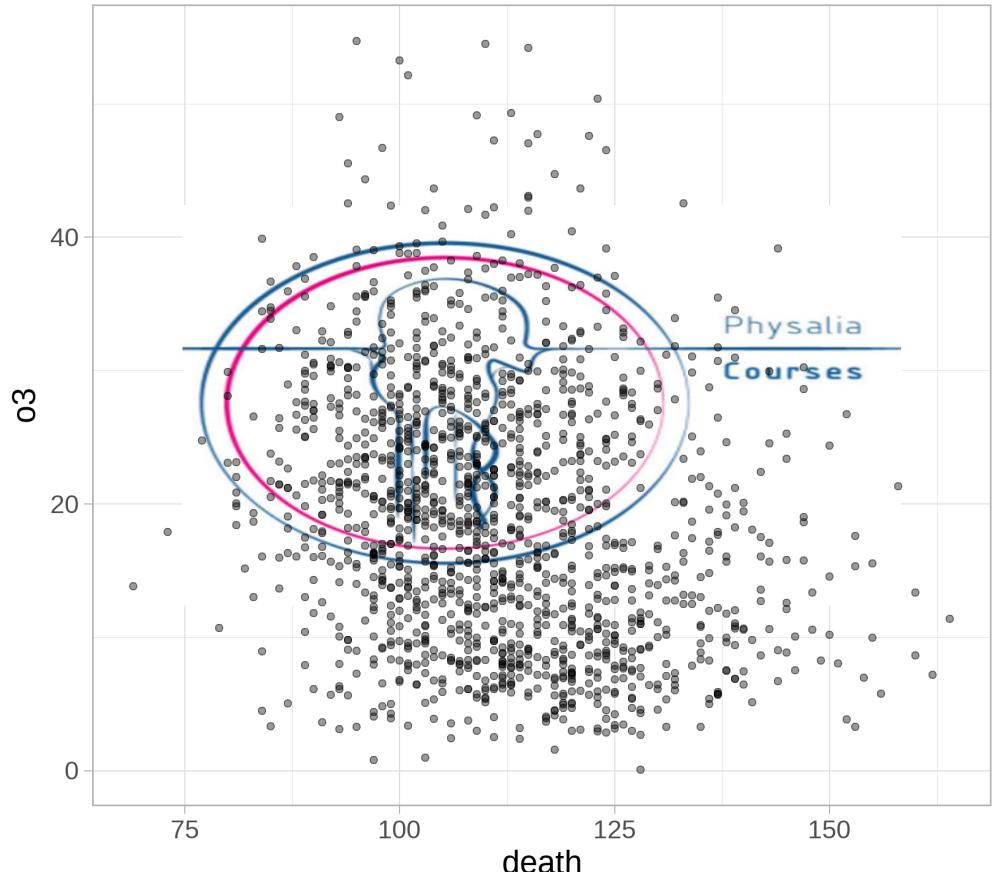


# annotation\_custom()

... and `annotation_custom()` allows also to add images:

```
library(png)

ggplot(chic, aes(death, o3)) +
  annotation_custom(
    grid::rasterGrob(
      image = readPNG(
        "img/physalia.png"
      ),
      width = unit(.8, "npc"),
      height = unit(.5, "npc")
    )
  ) +
  geom_point(alpha = .4)
```

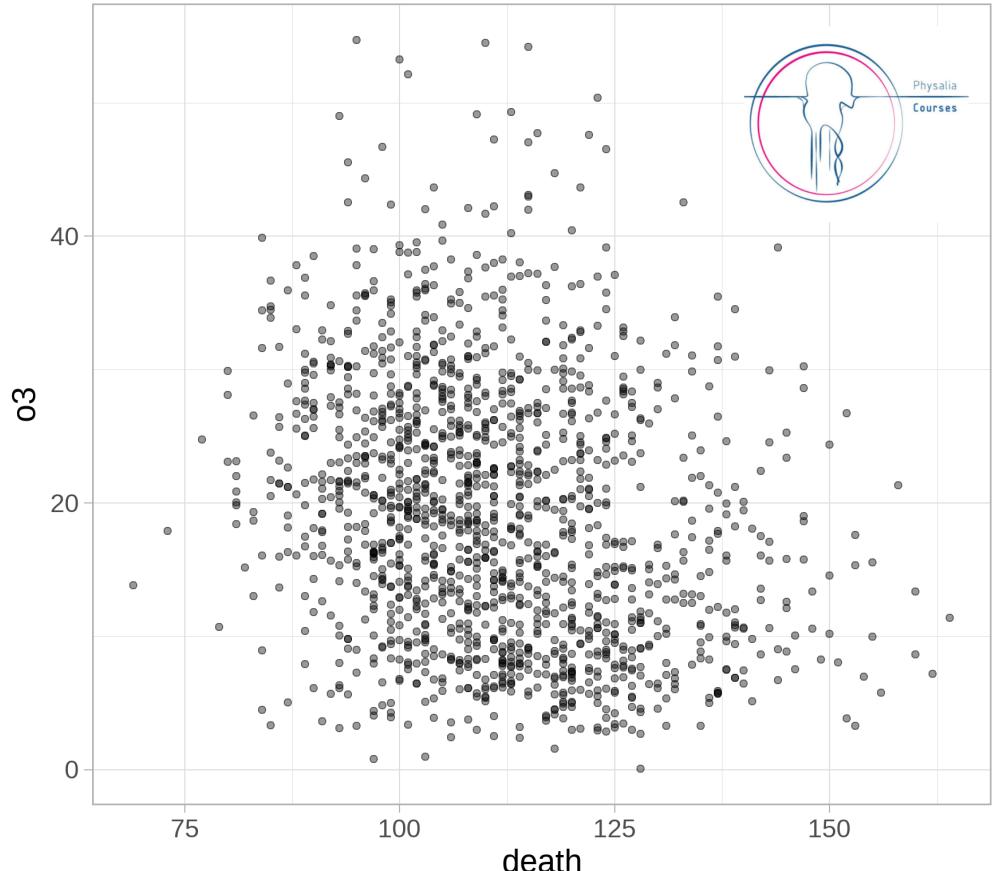


# annotation\_custom()

... and `annotation_custom()` allows also to add images:

```
library(png)

ggplot(chic, aes(death, o3)) +
  annotation_custom(
    grid::rasterGrob(
      image = readPNG(
        "img/physalia.png"
      ),
      x = .85,
      y = .85,
      width = .25
    )
  ) +
  geom_point(alpha = .4)
```



# Annotations

`geom_text()` and `geom_label()`

# geom\_text() and geom\_label()

You can simply add text labels to your plot by using `geom_text()` or `geom_label()`:

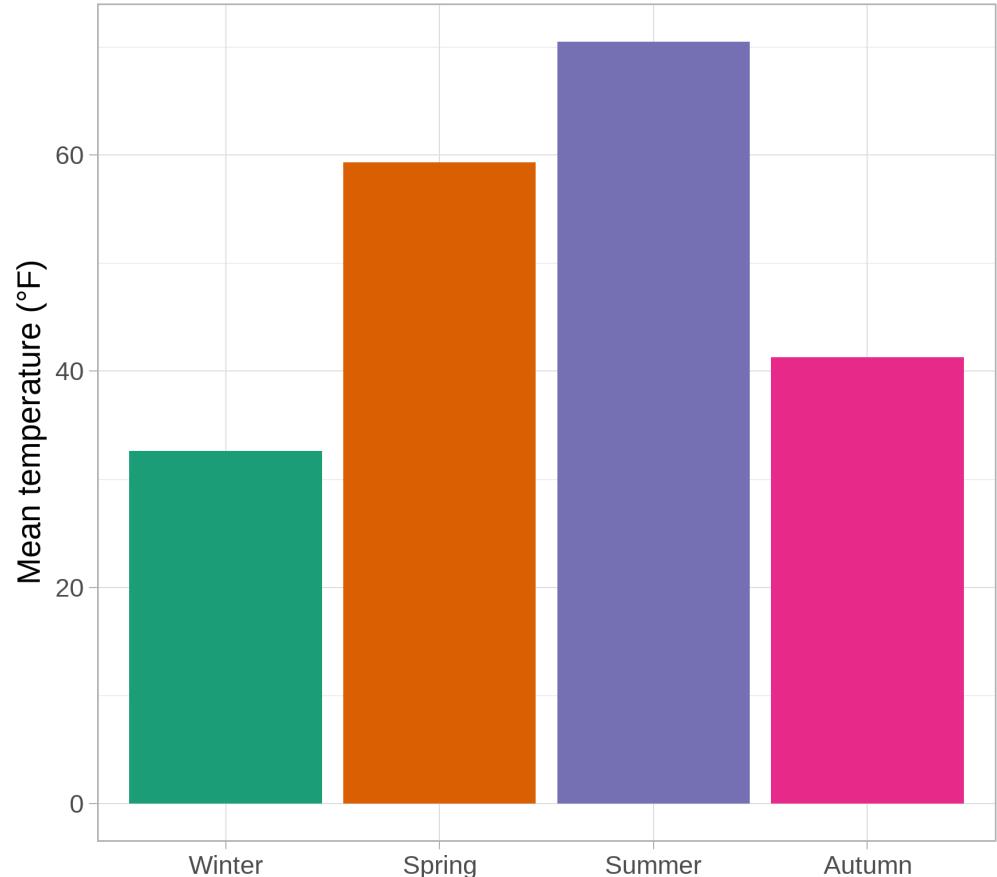
```
chic_sum <-  
  chic %>%  
  group_by(season) %>%  
  summarize(  
    avg = mean(temp, na.rm = TRUE)  
  )  
  
chic_sum
```

```
## # A tibble: 4 x 2  
##   season     avg  
##   <fct>    <dbl>  
## 1 Winter    32.6  
## 2 Spring    59.3  
## 3 Summer    70.5  
## 4 Autumn   41.3
```

# `geom_text()` and `geom_label()`

You can simply add text labels to your plot by using `geom_text()` or `geom_label()`:

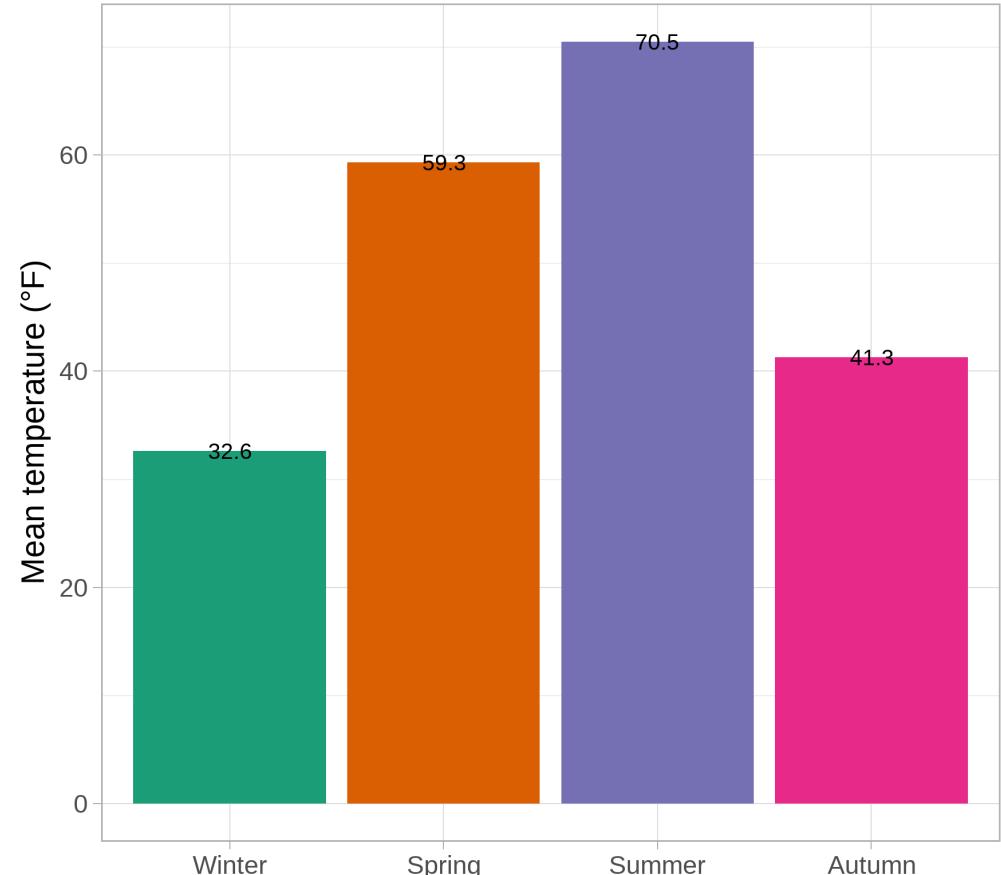
```
(g <-  
  ggplot(chic_sum,  
    aes(  
      season,  
      avg,  
      fill = season  
    )) +  
  geom_col() +  
  scale_fill_brewer(  
    palette = "Dark2",  
    guide = "none"  
  ) +  
  labs(  
    x = NULL,  
    y = "Mean temperature (°F)"  
  )  
)
```



# **geom\_text() and geom\_label()**

You can simply add text labels to your plot by using **geom\_text()** or **geom\_label()**:

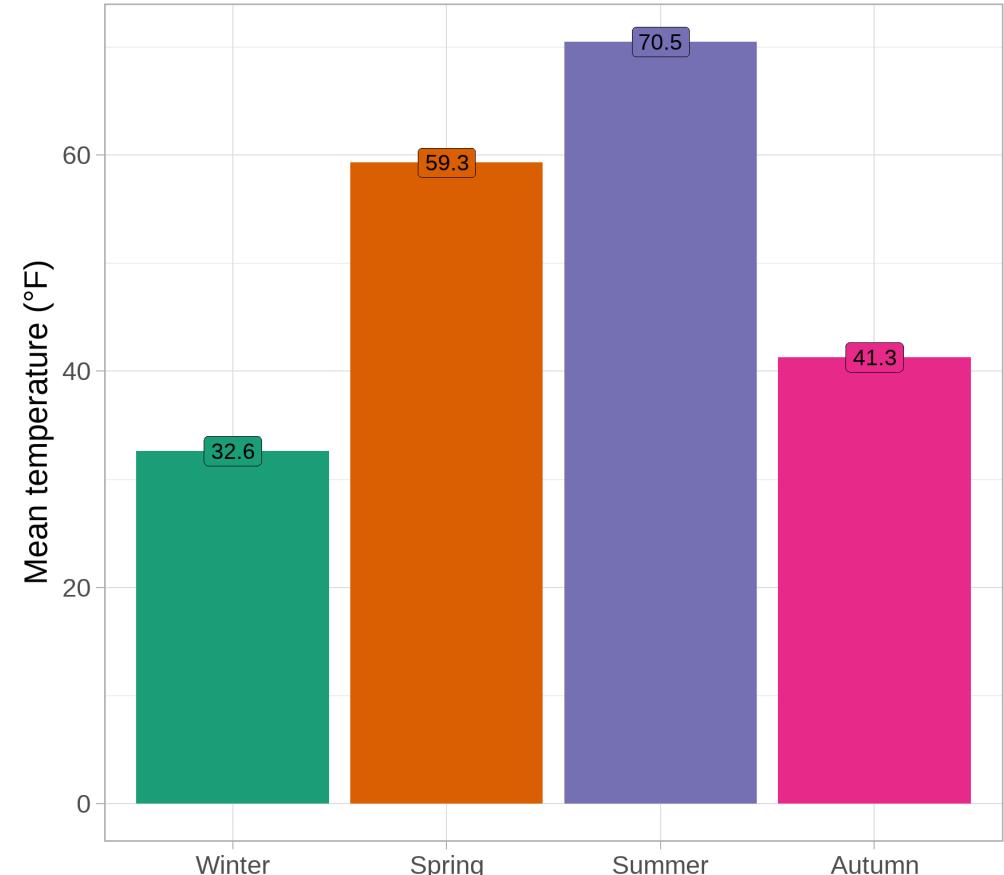
```
g +
  geom_text(
    aes(
      label = round(avg, 1)
    )
  )
```



# **geom\_text() and geom\_label()**

You can simply add text labels to your plot by using **geom\_text()** or **geom\_label()**:

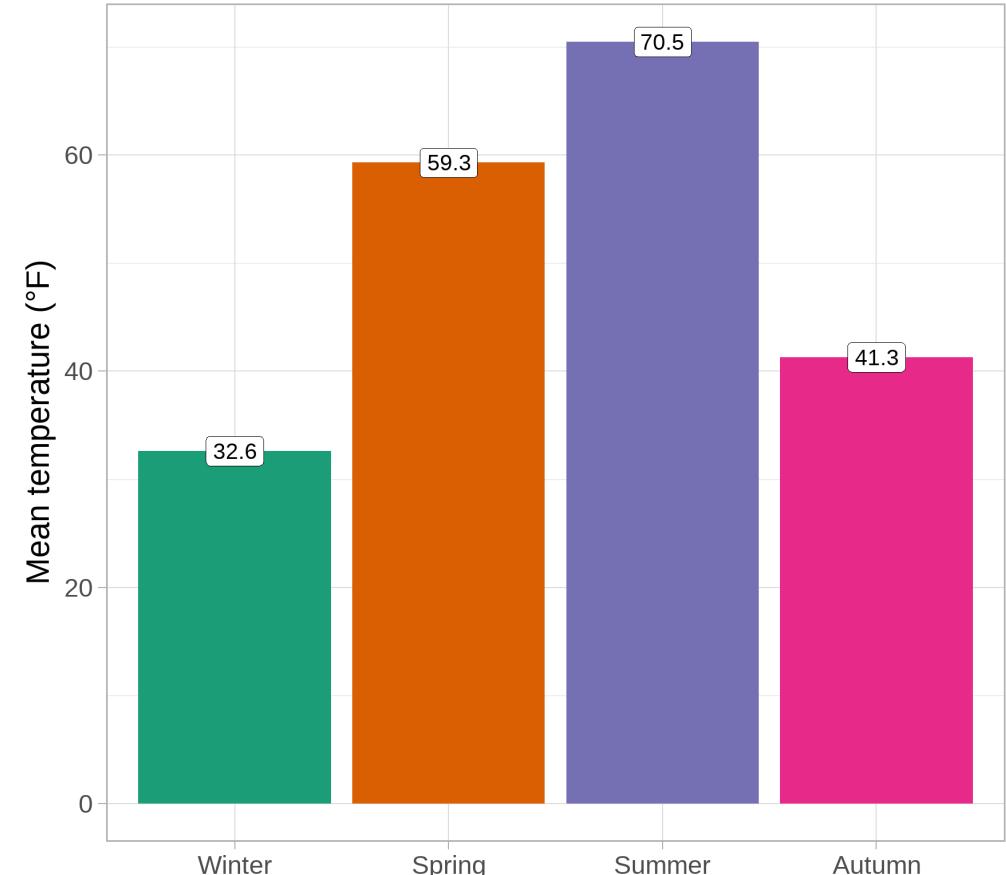
```
g +
  geom_label(
    aes(
      label = round(avg, 1)
    )
  )
```



# **geom\_text() and geom\_label()**

You can simply add text labels to your plot by using **geom\_text()** or **geom\_label()**:

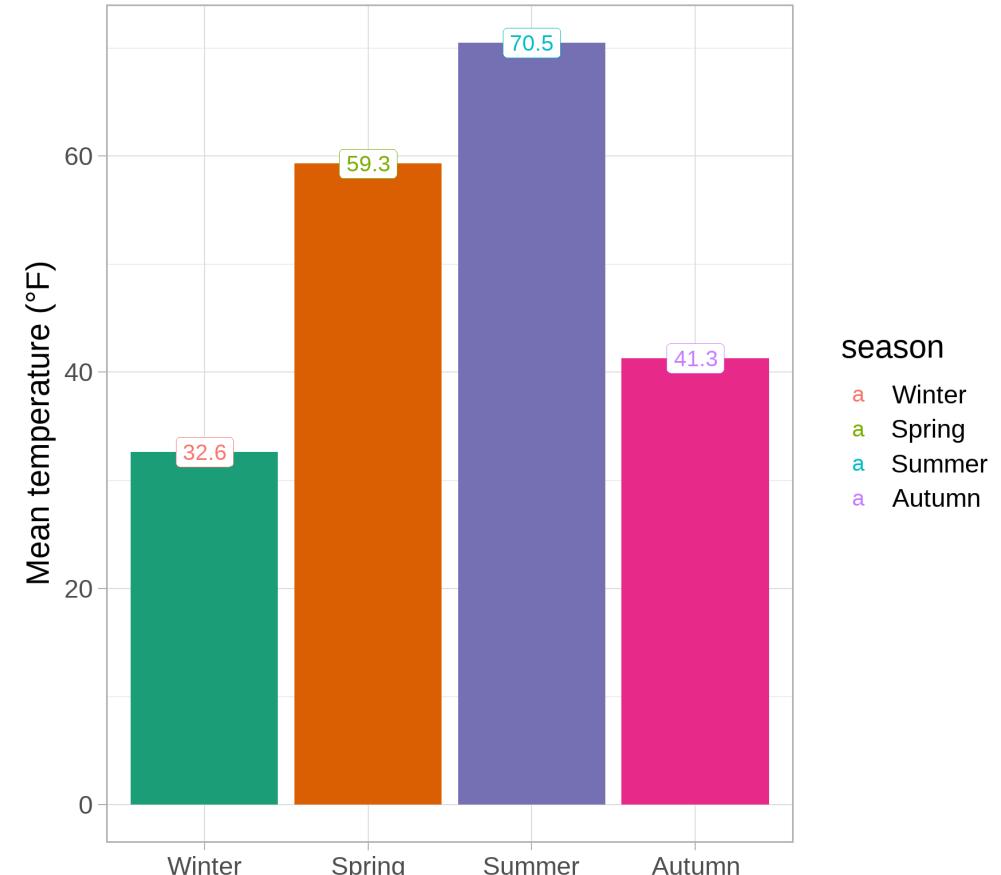
```
g +
  geom_label(
    aes(
      label = round(avg, 1)
    ),
    fill = "white"
  )
```



# `geom_text()` and `geom_label()`

You can simply add text labels to your plot by using `geom_text()` or `geom_label()`:

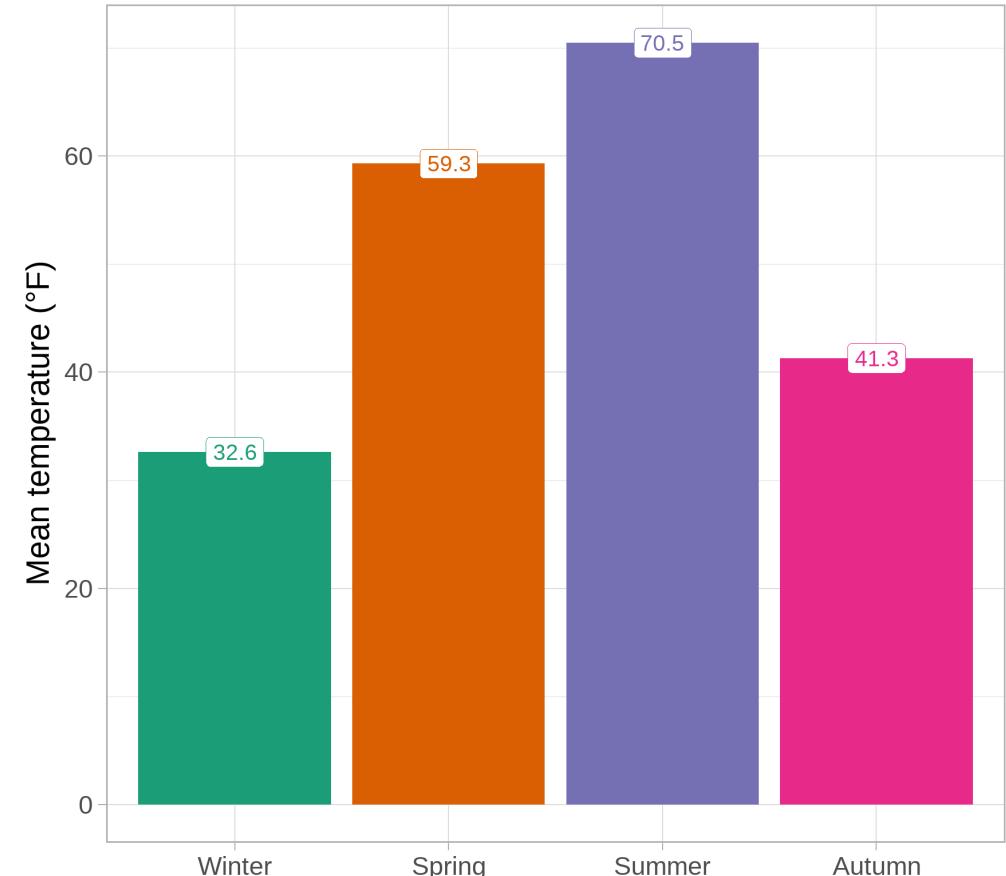
```
g +
  geom_label(
    aes(
      label = round(avg, 1),
      color = season
    ),
    fill = "white"
  )
```



# `geom_text()` and `geom_label()`

You can simply add text labels to your plot by using `geom_text()` or `geom_label()`:

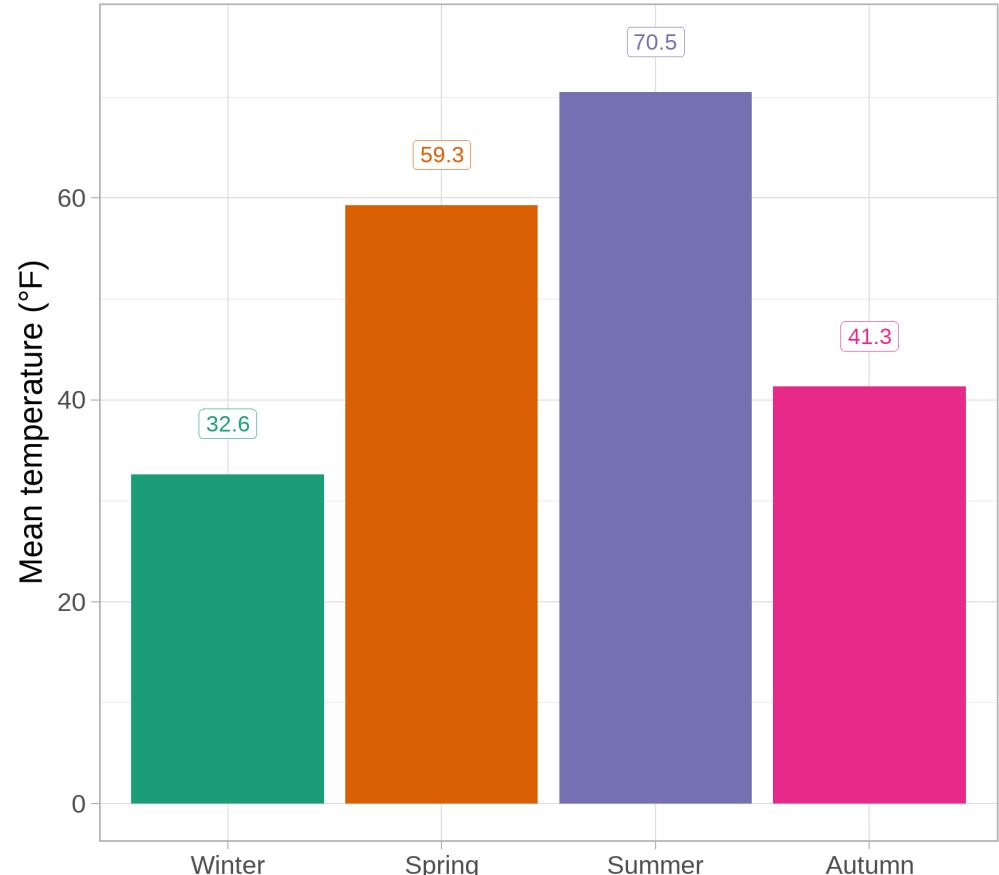
```
g <-  
  g +  
    scale_color_brewer(  
      palette = "Dark2",  
      guide = "none"  
    )  
  
g +  
  geom_label(  
    aes(  
      label = round(avg, 1),  
      color = season  
    ),  
    fill = "white"  
  )
```



# `geom_text()` and `geom_label()`

You can move the text labels by passing the argument `nudge_x` or `nudge_y`:

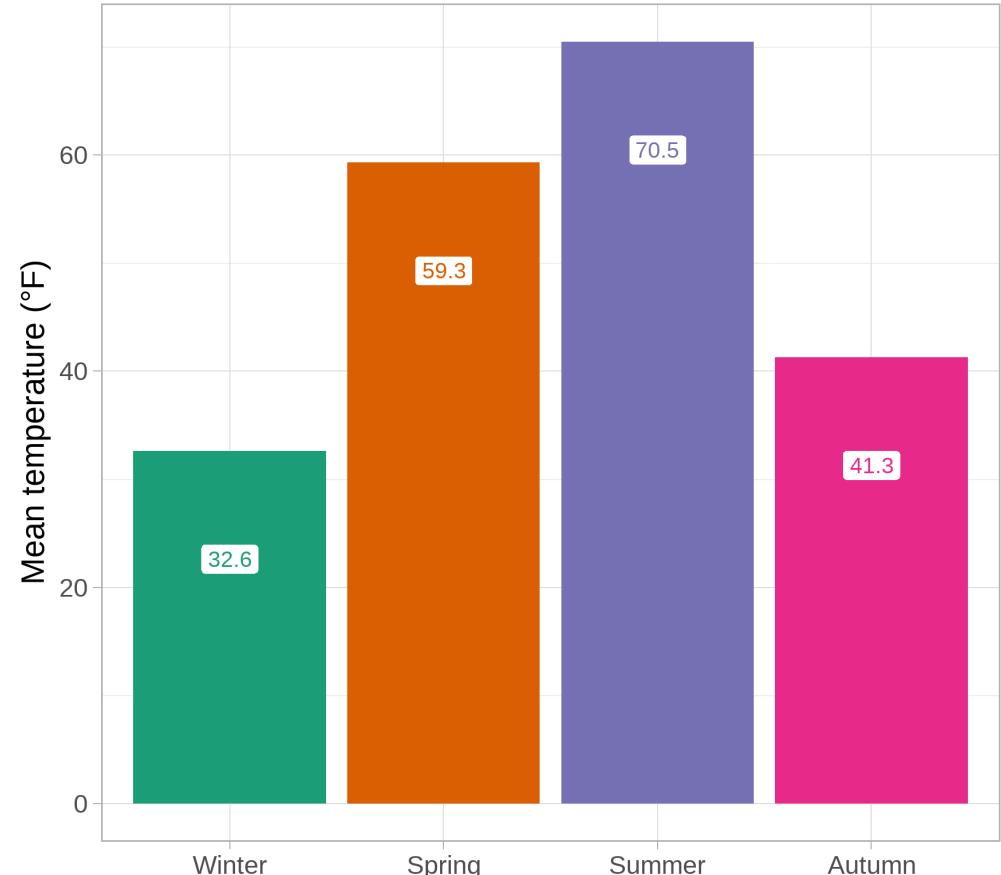
```
g +
  geom_label(
    aes(
      label = round(avg, 1),
      color = season
    ),
    fill = "white",
    nudge_y = 5
  )
```



# geom\_text() and geom\_label()

You can move the text labels by passing the argument `nudge_x` or `nudge_y`:

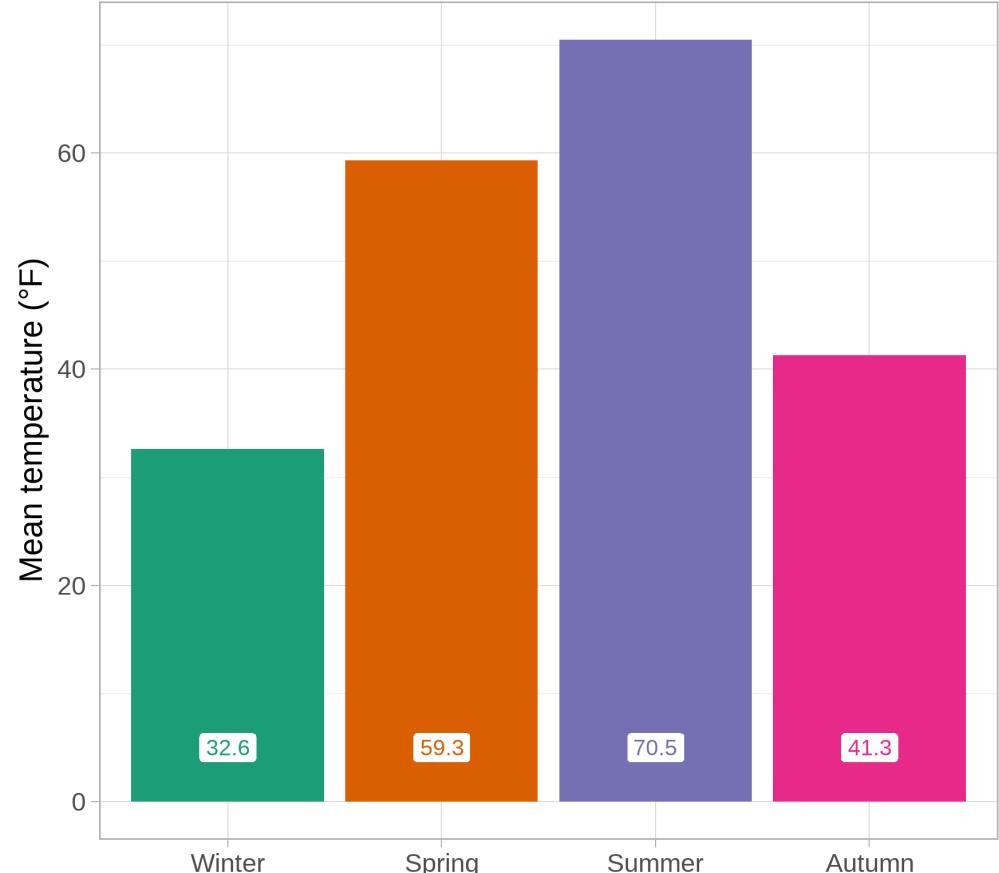
```
g +
  geom_label(
    aes(
      label = round(avg, 1),
      color = season
    ),
    fill = "white",
    nudge_y = -10
  )
```



# geom\_text() and geom\_label()

But you can, of course, also change the position aesthetics!

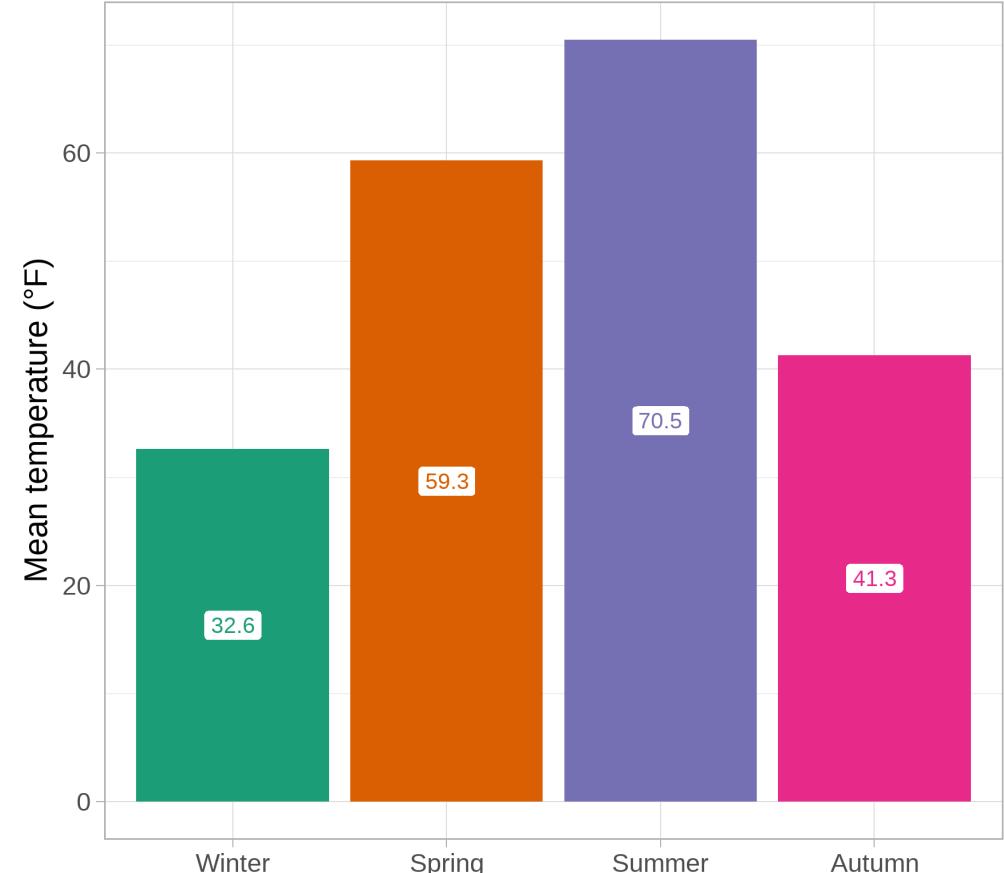
```
g +
  geom_label(
    aes(
      y = 5,
      label = round(avg, 1),
      color = season
    ),
    fill = "white"
  )
```



# `geom_text()` and `geom_label()`

But you can, of course, also change the position aesthetics!

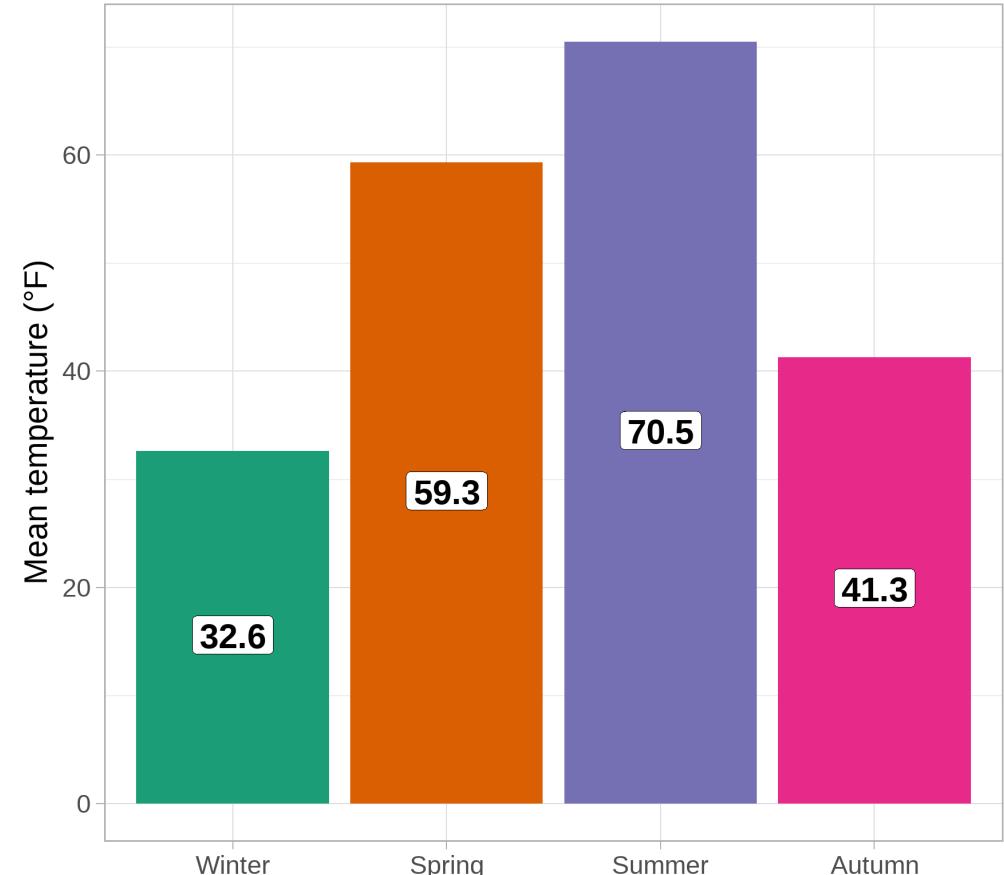
```
g +
  geom_label(
    aes(
      y = avg / 2,
      label = round(avg, 1),
      color = season
    ),
    fill = "white"
  )
```



# geom\_text() and geom\_label()

The appearance of the text can be adjusted:

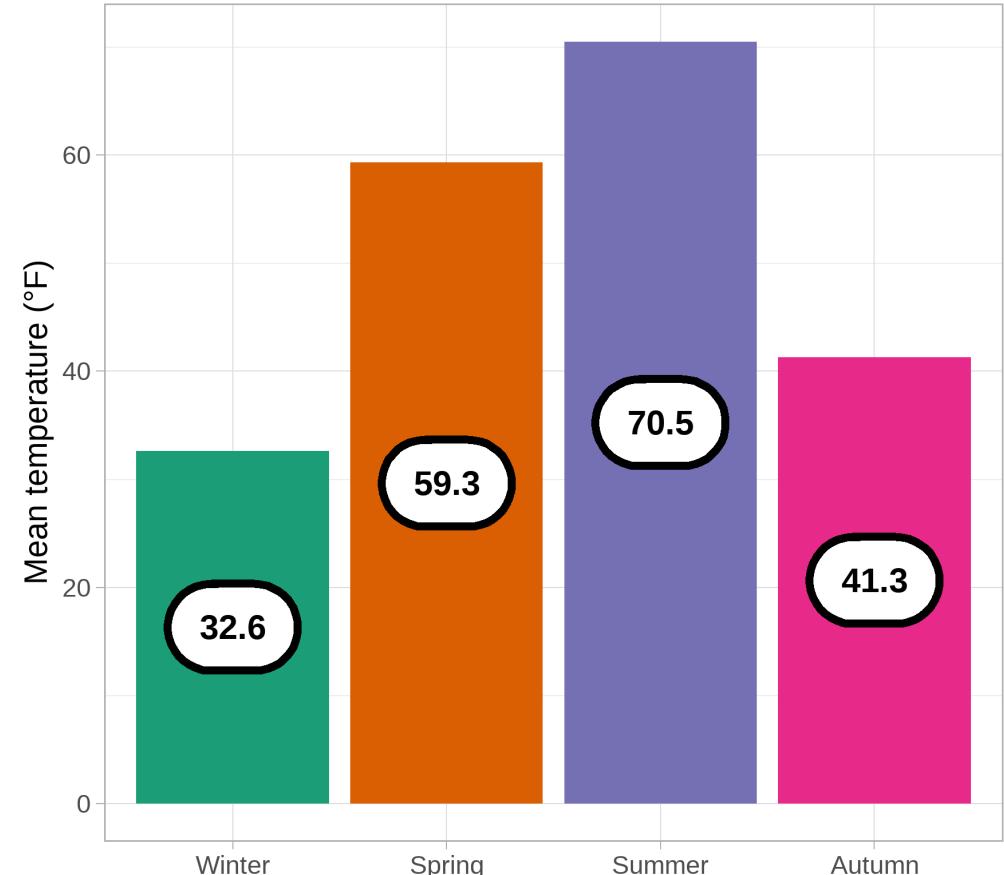
```
g +
  geom_label(
    aes(
      y = avg / 2,
      label = round(avg, 1)
    ),
    fill = "white",
    size = 6,
    family = "Roboto Mono",
    fontface = "bold",
    vjust = .7
  )
```



# geom\_text() and geom\_label()

... and the appearance of the box can be adjusted as well:

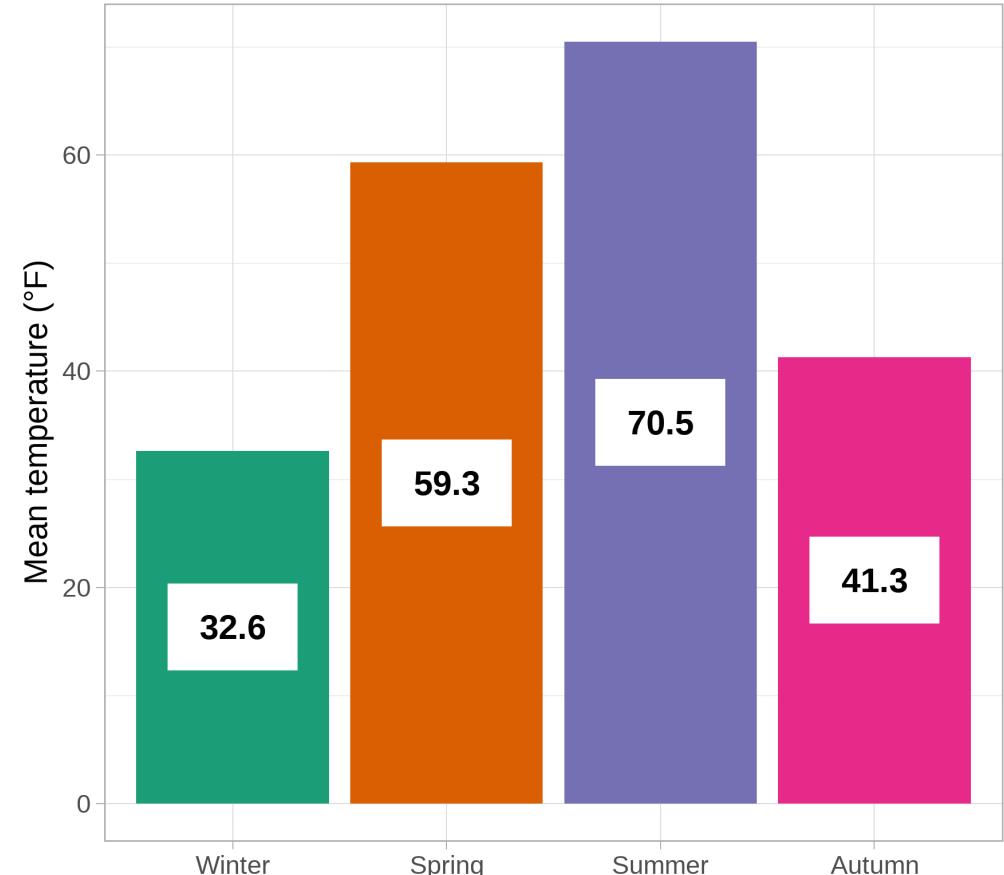
```
g +
  geom_label(
    aes(
      y = avg / 2,
      label = round(avg, 1)
    ),
    fill = "white",
    size = 6,
    family = "Roboto Mono",
    fontface = "bold",
    label.padding = unit(1.1, "lines"),
    label.r = unit(1.5, "lines"),
    label.size = 2
  )
```



# geom\_text() and geom\_label()

... and the appearance of the box can be adjusted as well:

```
g +
  geom_label(
    aes(
      y = avg / 2,
      label = round(avg, 1)
    ),
    fill = "white",
    size = 6,
    family = "Roboto Mono",
    fontface = "bold",
    label.padding = unit(1.1, "lines"),
    label.r = unit(0, "lines"),
    label.size = 0
  )
```



# Annotations

## The `ggrepel` Package

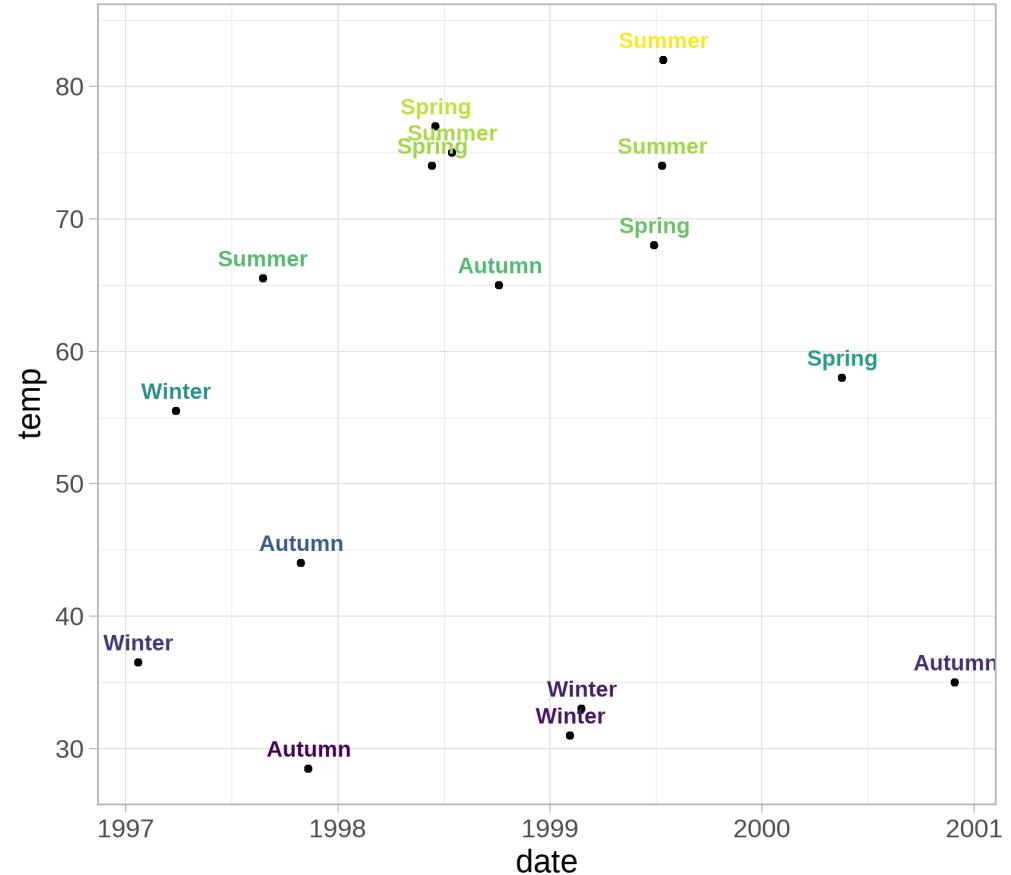
# The `ggrepel` Package

The `ggrepel` package provides the geoms `geom_text_repel()` and `geom_label_repel()` to repel overlapping text labels:

```
set.seed(2020)

sample <- chic %>%
  group_by(season) %>%
  sample_frac(0.01)

ggplot(sample, aes(date, temp)) +
  geom_point() +
  geom_text(
    aes(
      color = temp,
      label = season),
    fontface = "bold",
    nudge_y = 1.5) +
  scale_color_viridis_c(guide = "none")
```

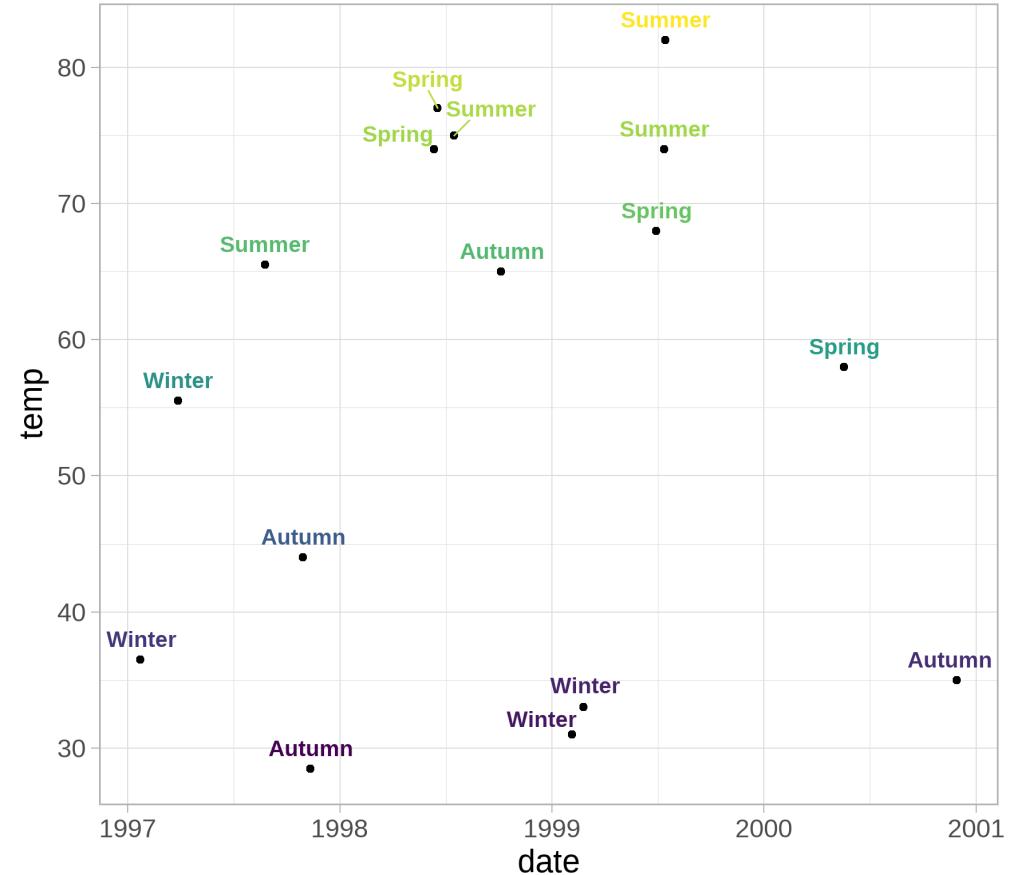


# The `ggrepel` Package

The `ggrepel` package provides the geoms `geom_text_repel()` and `geom_label_repel()` to repel overlapping text labels:

```
install.packages("ggrepel")
library(ggrepel)

ggplot(sample, aes(date, temp)) +
  geom_point() +
  geom_text_repel(
    aes(
      color = temp,
      label = season),
    fontface = "bold",
    nudge_y = 1.5) +
  scale_color_viridis_c(guide = "none")
```

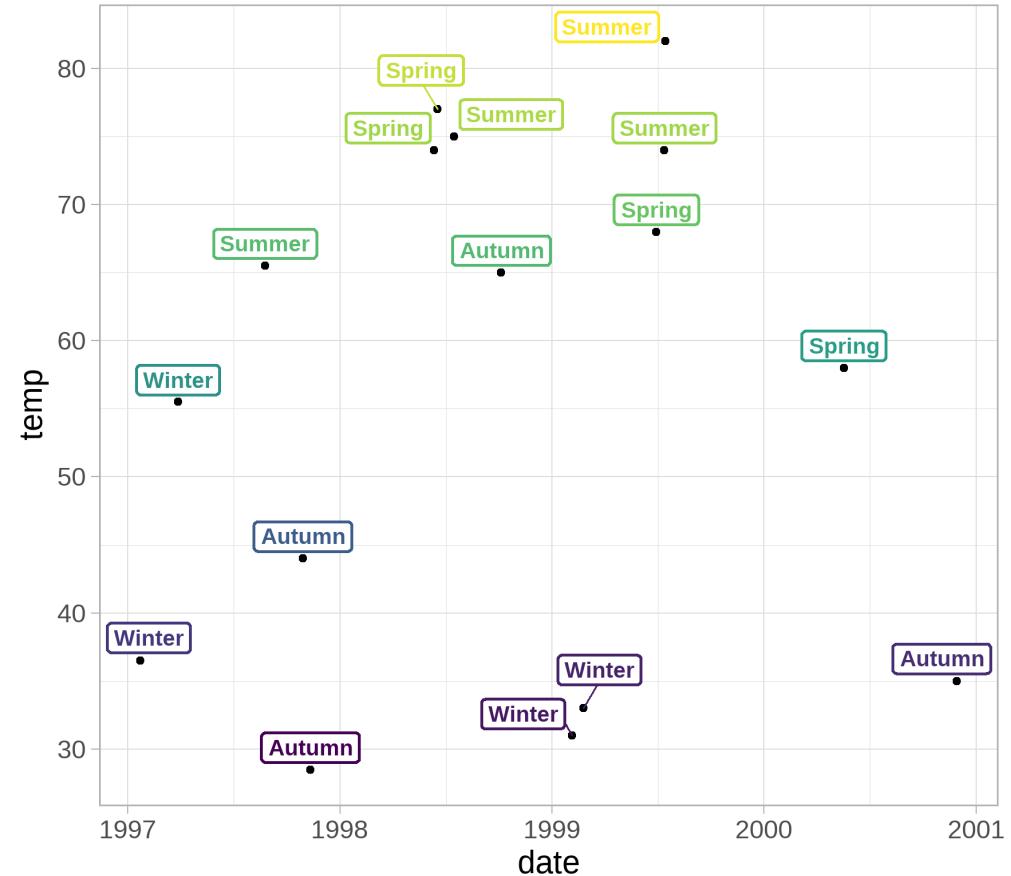


# The `ggrepel` Package

The `ggrepel` package provides the geoms `geom_text_repel()` and `geom_label_repel()` to repel overlapping text labels:

```
install.packages("ggrepel")
library(ggrepel)

ggplot(sample, aes(date, temp)) +
  geom_point() +
  geom_label_repel(
    aes(
      color = temp,
      label = season),
    fontface = "bold",
    nudge_y = 1.5,
    label.size = .8
  ) +
  scale_color_viridis_c(guide = "none")
```



# Annotations

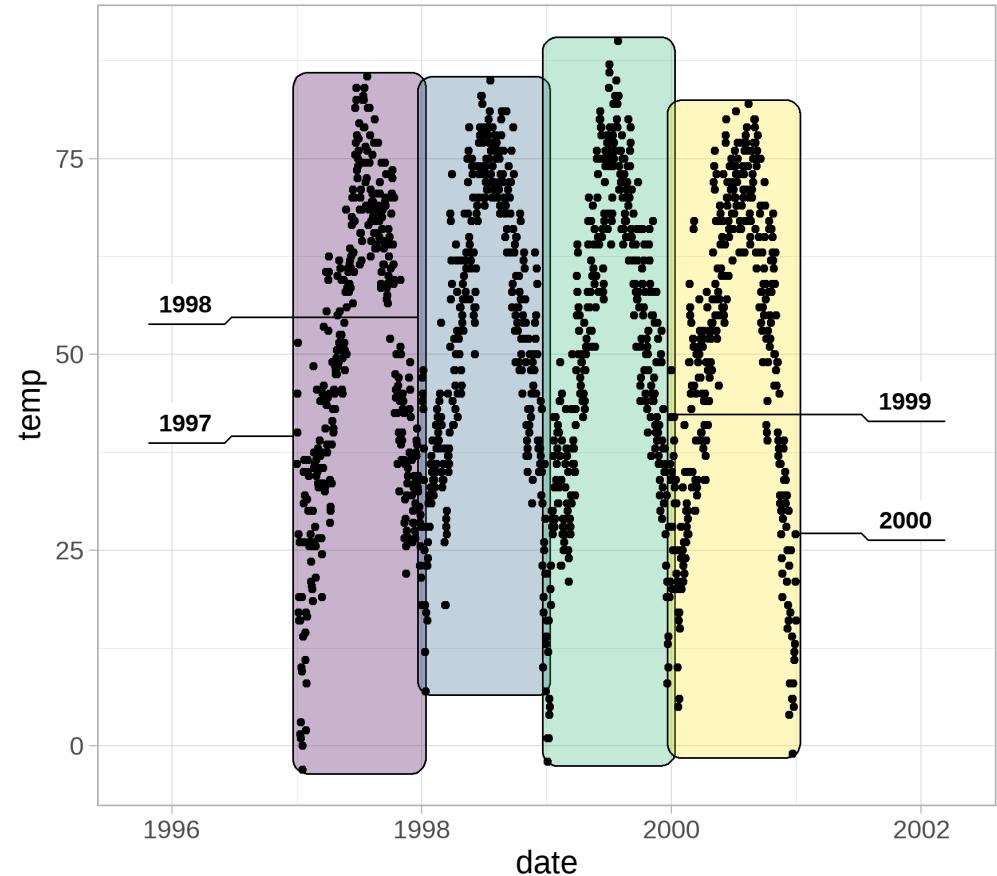
## The `ggforce` Package

# The **ggforce** Package: **geom\_mark\_\***()

The **ggforce** package comes with several **geom\_mark\_\***() functions to add nice designed labels:

```
install.packages("ggforce")
library(ggforce)

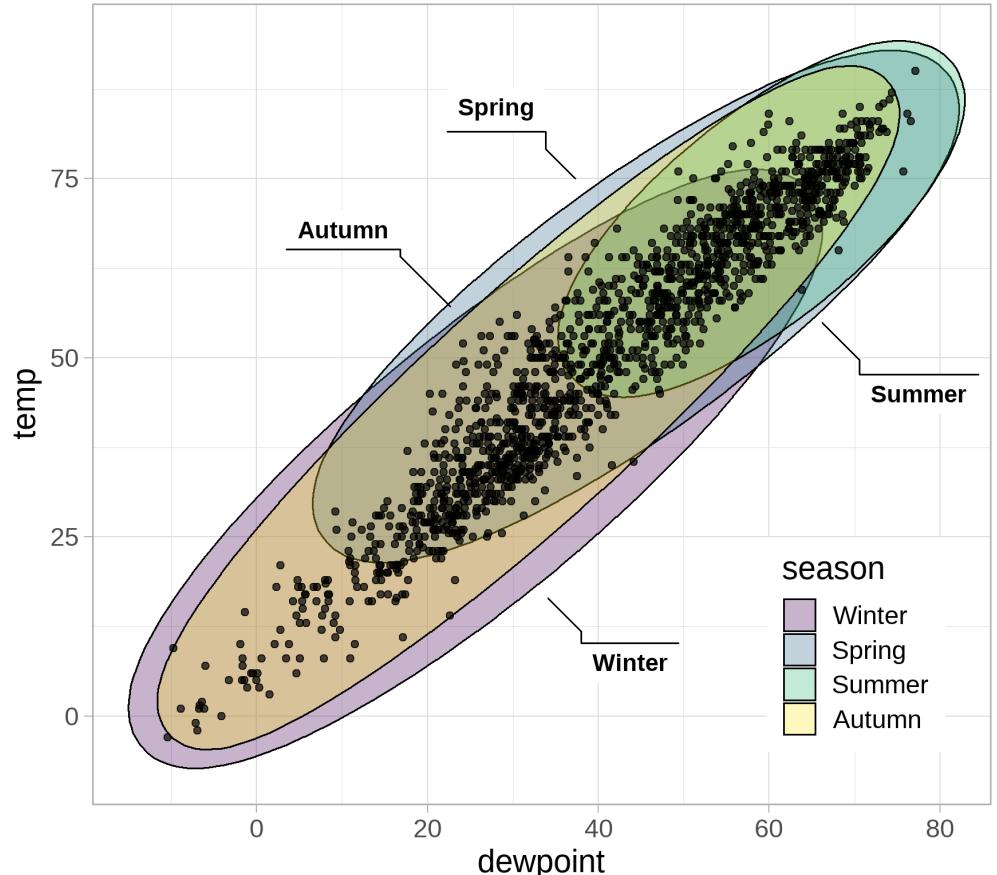
ggplot(chic, aes(date, temp)) +
  geom_mark_rect(
    aes(
      label = year,
      fill = year
    ),
    con.cap = unit(0, "pt"),
    expand = unit(2, "pt")
  ) +
  geom_point() +
  scale_fill_viridis_d(
    guide = "none"
  ) +
  scale_x_date(
    expand = c(.4, .4)
  )
```



# The **ggforce** Package: **geom\_mark\_\***()

The **ggforce** package comes with several **geom\_mark\_\***() functions to add nice designed labels:

```
ggplot(chic, aes(dewpoint, temp)) +  
  geom_mark_ellipse(  
    aes(  
      label = season,  
      fill = season  
    )  
  ) +  
  geom_point(alpha = .7) +  
  scale_fill_viridis_d() +  
  scale_x_continuous(  
    expand = c(.1, .1)  
  ) +  
  scale_y_continuous(  
    expand = c(.1, .1)  
  ) +  
  theme(  
    legend.position = c(.85, .2)  
  )
```



# Annotations

## The **ggtext** Package

# The **ggttext** Package

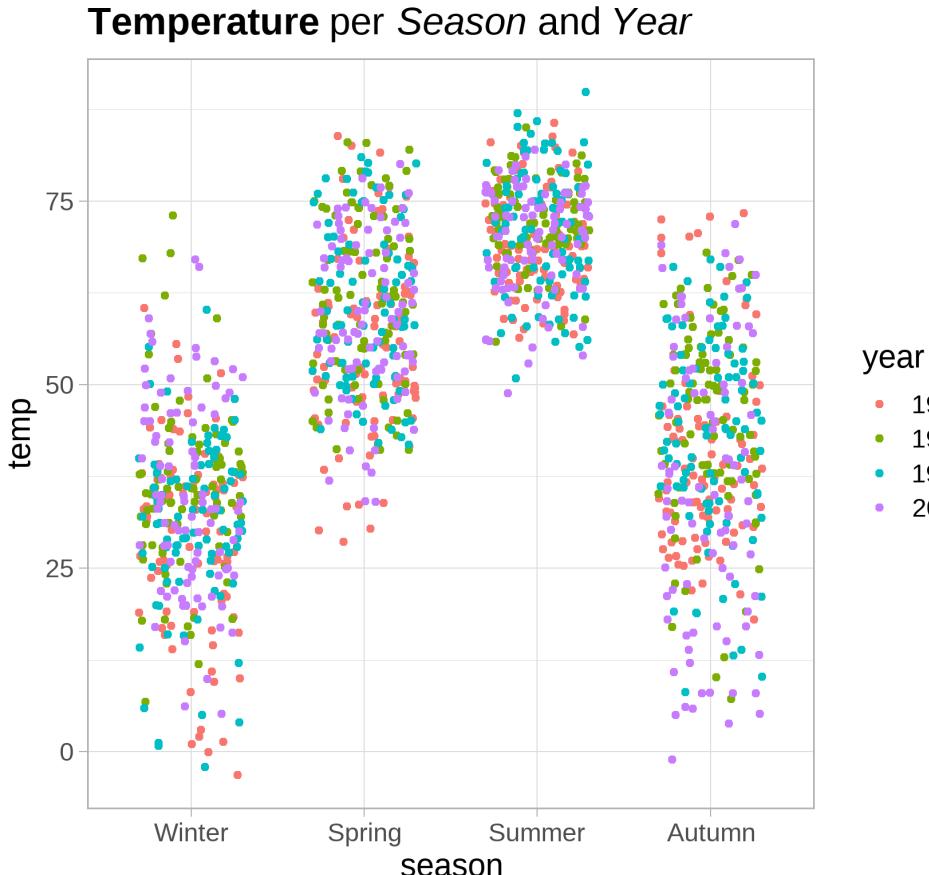
The **ggttext** package provides rich-text (basic HTML and Markdown) support for titles, subtitles, captions, axis labels, legends and labels.

```
remotes::install_github("wilkelab/ggttext"
library(ggttext)

ggplot(chic, aes(season, temp)) +
  geom_jitter(
    aes(color = year),
    width = 0.3
  ) +
  ggttitle("**Temperature** per *Season* and *Year*")
  theme(plot.title = element_markdown())
```

**Full markdown Code:**

**\*\*Temperature\*\* per \*Season\* and \*Year\***



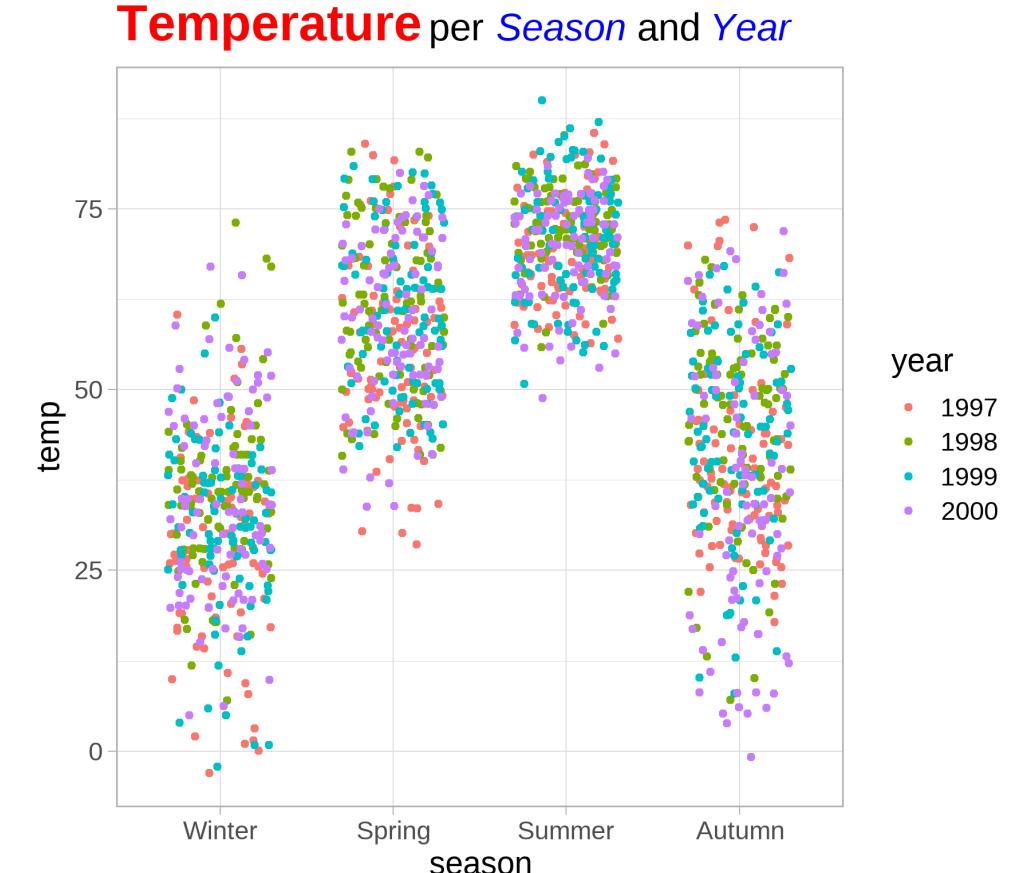
# The **ggttext** Package

The **ggttext** package provides rich-text (basic HTML and Markdown) support for titles, subtitles, captions, axis labels, legends and labels.

```
ggplot(chic, aes(season, temp)) +  
  geom_jitter(  
    aes(color = year),  
    width = 0.3  
  ) +  
  ggtitle("<b style='color:red;font-size:25pt;'>  
Temperature</b> per <i style='color:  
blue;'>Season</i> and <i style='color:  
blue;'>Year</i>")  
  theme(plot.title = element_markdown())
```

**Full html Code:**

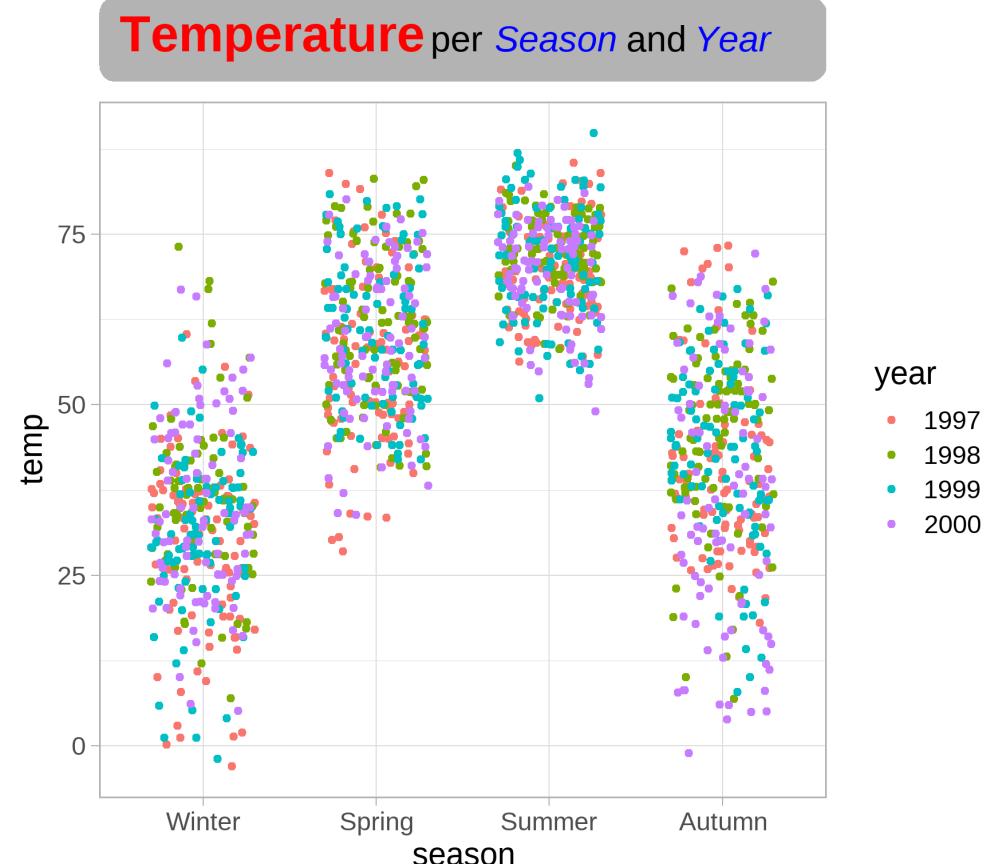
```
<b style='color:red;font-size:25pt;'>  
Temperature</b> per <i style='color:  
blue;'>Season</i> and <i style='color:  
blue;'>Year</i>
```



# The **ggttext** Package

The **ggttext** package provides rich-text (basic HTML and Markdown) support for titles, subtitles, captions, axis labels, legends and labels.

```
ggplot(chic, aes(season, temp)) +  
  geom_jitter(  
    aes(color = year),  
    width = 0.3  
  ) +  
  ggtitle("<b style='color:red;font-size:18px;white-space:pre;'" +  
    "text-align:center;">Temperature per Season and Year  
  ) +  
  theme(  
    plot.title = element_textbox_simple(  
      size = 18,  
      fill = "grey70",  
      hjust = .5,  
      padding = margin(10, 10, 10, 10),  
      margin = margin(0, 0, 10, 0),  
      r = unit(8, "pt")  
    )  
  )
```



# The `ggttext` Package

The `ggttext` package provides rich-text (basic HTML and Markdown) support for titles, subtitles, captions, axis labels, legends and labels.

```
(labels <-
  tibble(
    season = factor(c("Spring", "Summer"), levels = levels(chic$season)),
    temp = c(5, 70),
    label = c(
      "<b style='color:red'>An important note!</b><br>Please report <b>R<sup>2</sup></b>
      '**Nice Viz!**<br>*Well done my friend.*"
    ),
    angle = c(40, -10),
    fill = c("grey40", "goldenrod"),
    hjust = c(0, .5)
  )
)
## # A tibble: 2 x 6
##   season  temp label                               angle fill     hjust
##   <fct>   <dbl> <chr>                            <dbl> <chr>     <dbl>
## 1 Spring     5 <b style='color:red'>An important note!</b><~      40 grey40     0
## 2 Summer    70 **Nice Viz!**<br>*Well done my friend.*      -10 golden~   0.5
```

# The `ggttext` Package

The `ggttext` package provides rich-text (basic HTML and Markdown) support for titles, subtitles, captions, axis labels, legends and labels.

```
ggplot(chic, aes(season, temp)) +  
  geom_jitter(  
    aes(color = year),  
    width = 0.3  
  ) +  
  geom_richtext(  
    data = labels,  
    aes(  
      label = label,  
      fill = fill,  
      angle = angle,  
      hjust = hjust  
    ),  
    size = 5,  
    color = "white"  
  ) +  
  scale_fill_identity()
```



# The `ggttext` Package

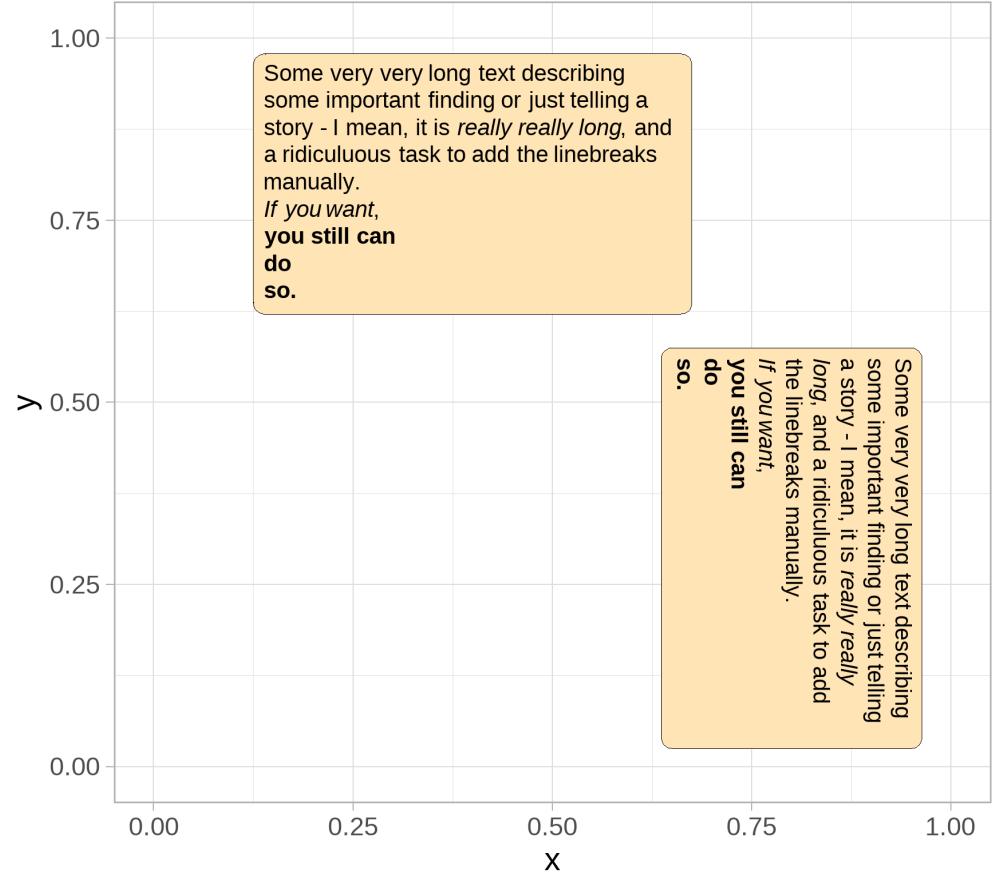
`element_textbox()` offers support for rendering larger amounts of text that require word wrapping (but only at fixed orientations corresponding to 0, 90, 180, and 270 degrees).

```
(labels <-
  tibble(
    label = rep("Some very very long text describing some important finding or just te
    x = c(.4, .8),
    y = c(.8, .3),
    orientation = c("upright", "right-rotated")
  )
)
## # A tibble: 2 x 4
##   label                                x     y orientation
##   <chr>                               <dbl> <dbl> <chr>
## 1 Some very very long text describing some important f~  0.4   0.8 upright
## 2 Some very very long text describing some important f~  0.8   0.3 right-rotat~
```

# The `ggttext` Package

`element_textbox()` offers support for rendering larger amounts of text that require word wrapping for fixed orientations.

```
ggplot(  
  labels,  
  aes(  
    x, y,  
    label = label,  
    orientation = orientation  
) +  
  geom_textbox(  
    size = 4,  
    fill = "moccasin",  
    width = unit(0.5, "npc")  
) +  
  xlim(0, 1) + ylim(0, 1)
```



# Annotations

## The **cowplot** Package

(once again)

# The **ggttext** Package

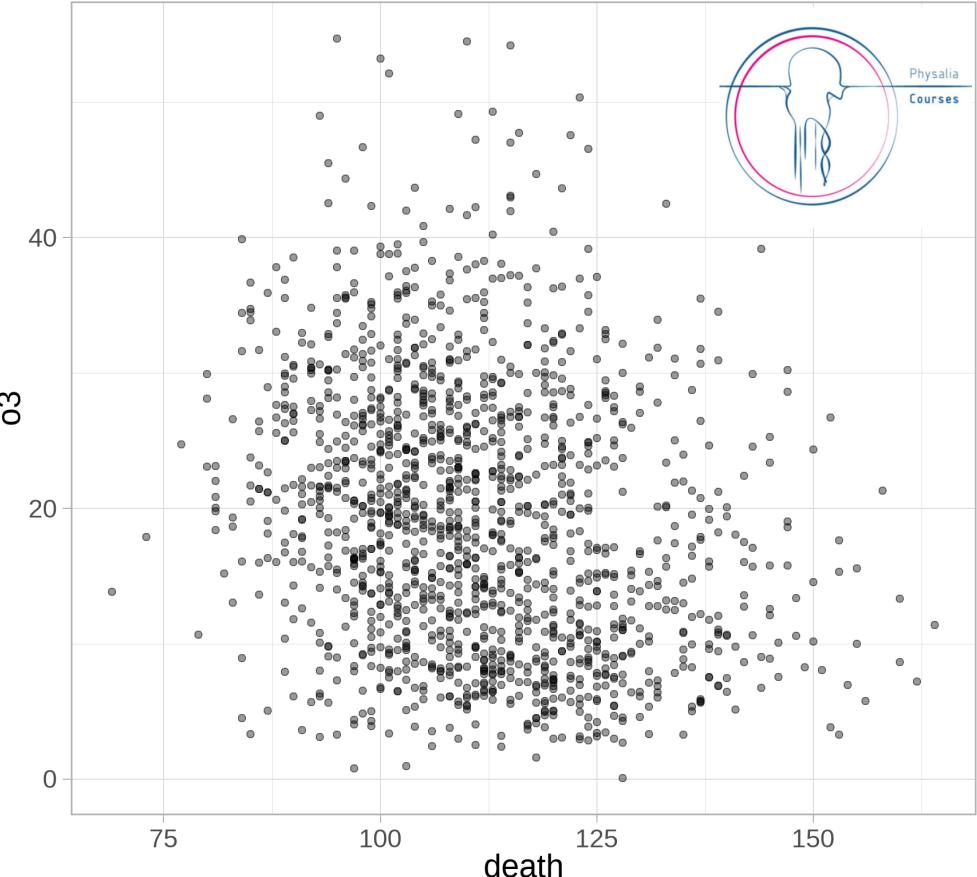
The **cowplot** package can not only add plots to other plots but also images:

```
library(cowplot)

g <- ggplot(chic, aes(death, o3)) +
  geom_point(alpha = .4)

logo <- png:::readPNG("img/physalia.png")

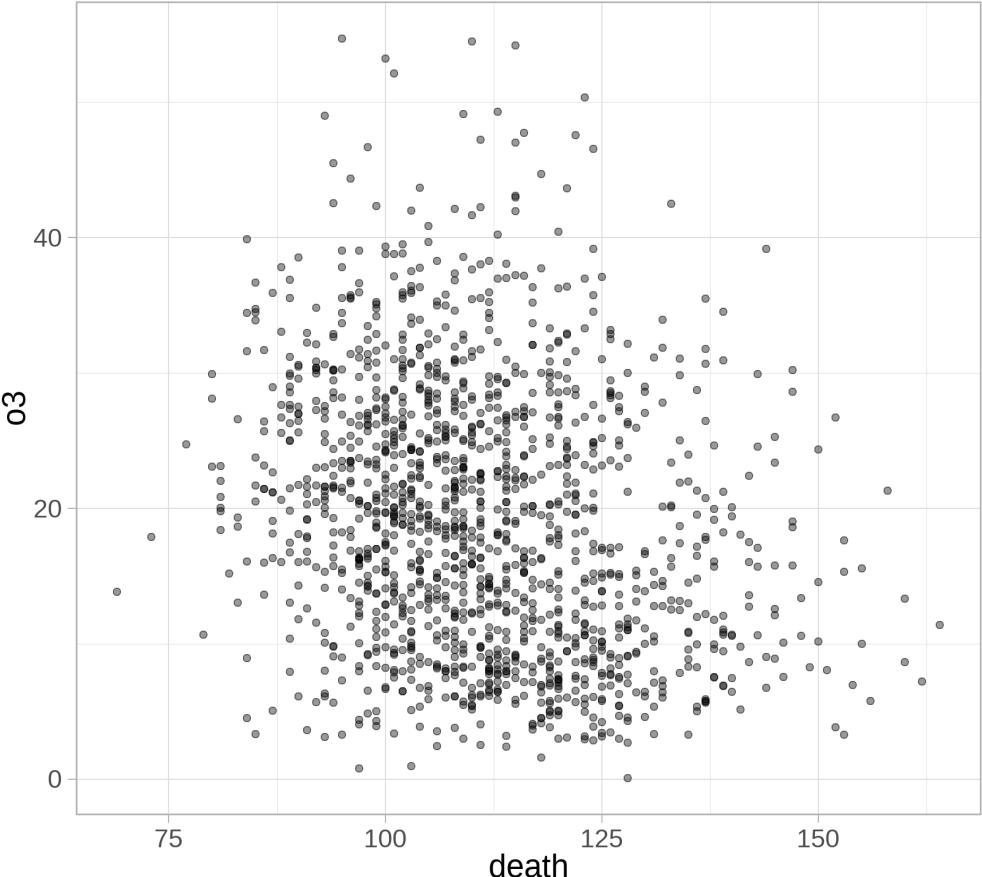
ggdraw(g) +
  draw_image(
    logo,
    x = .98,
    y = .98,
    hjust = 1,
    vjust = 1,
    width = .25,
    height = .25
)
```



# The **ggttext** Package

The **cowplot** package can not only add plots to other plots but also images:

```
g <- ggplot(chic, aes(death, o3)) +  
  geom_point(alpha = .4) +  
  theme(  
    panel.background = element_blank(),  
    panel.border = element_rect(  
      fill = NA  
    )  
  )  
  
logo <- png:::readPNG("img/physalia.png")  
  
cowplot::ggdraw() +  
  cowplot::draw_image(  
    logo,  
    scale = .5  
  ) +  
  cowplot::draw_plot(g)
```



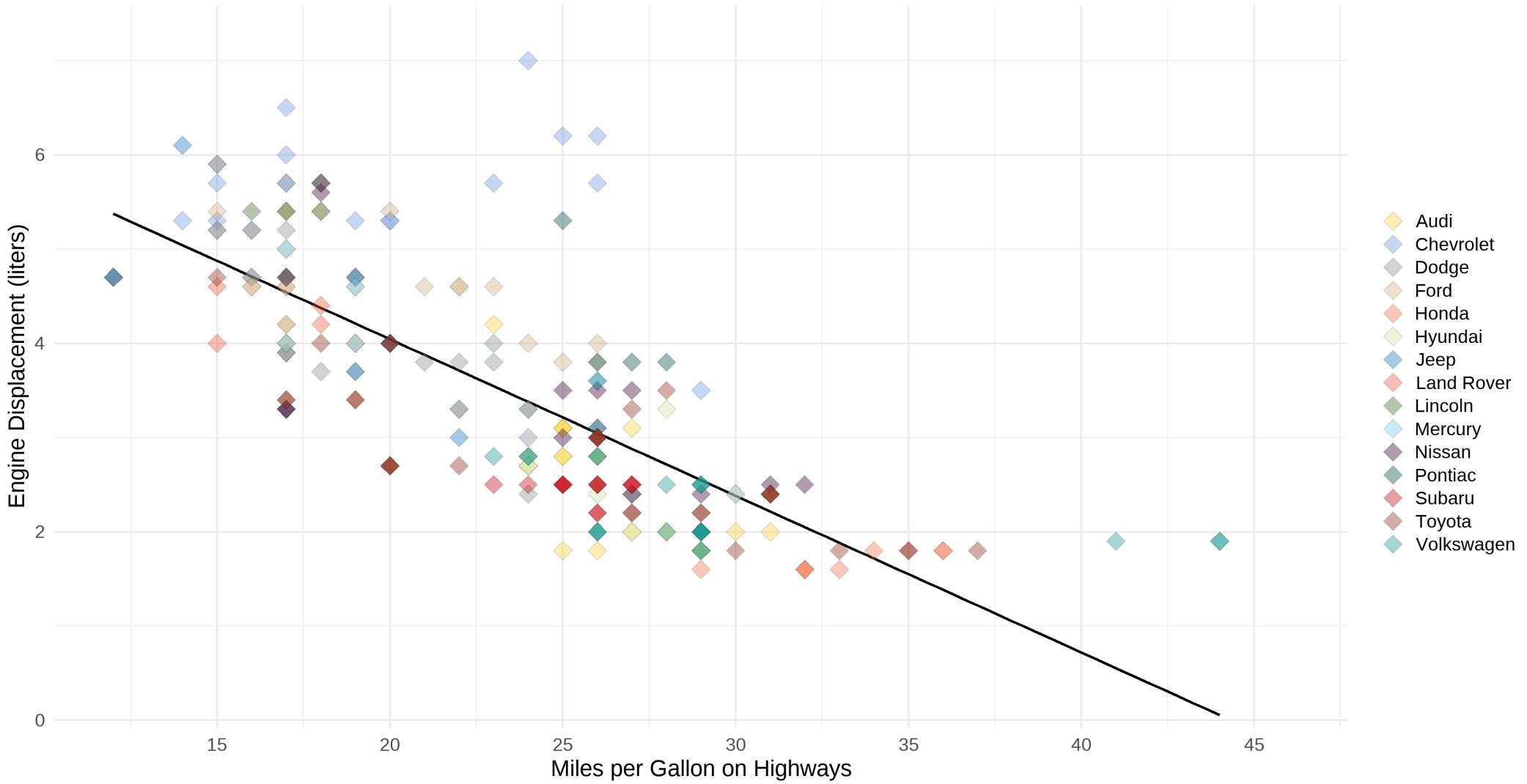
# Wrap-Up

## The Structure of `ggplot2`

# The Structure of ggplot2

```
(g <- mpg %>%
  mutate(manufacturer = str_to_title(manufacturer)) %>%
  ggplot(aes(hwy, displ)) +
  stat_smooth(method = "lm",
              color = "black",
              se = FALSE) +
  geom_point(aes(fill = manufacturer),
             shape = 23,
             size = 5,
             alpha = .4,
             color = "black",
             stroke = .3) +
  scale_x_continuous(breaks = seq(15, 45, by = 5),
                     limits = c(NA, 46)) +
  scale_y_continuous(expand = c(.01, .01),
                     limits = c(0, 7.5)) +
  ggsci::scale_fill_simpsons(name = "") +
  labs(x = "Miles per Gallon on Highways",
       y = "Engine Displacement (liters)") +
  theme_minimal(base_size = 17, base_family = "Montserrat"))
```

# The Structure of `ggplot2`



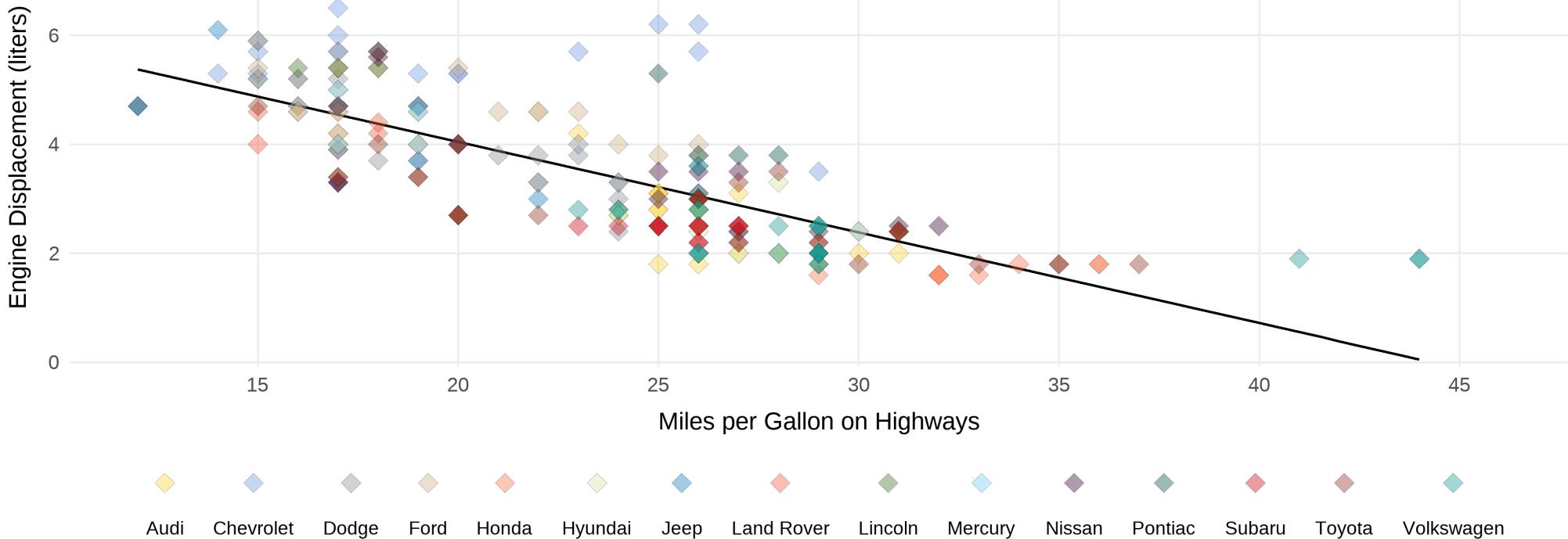
# The Structure of ggplot2

```
(g <- g +  
  labs(title = '<i>New Beetle</i> and <i>Jetta</i> by <b>S  
    subtitle = "Linear regression of engine displacement versus highway miles per gallon"  
    caption = "\nVisualization: Cédric Scherer • Data: EPA (www.fueleconomy.gov)"  
  guides(fill = guide_legend(nrow = 1, label.position = "bottom")) +  
  theme(plot.title.position = "plot",  
        plot.title = element_markdown(face = "bold", size = 24,  
                                       margin = margin(5, 0, 5, 0)),  
        plot.subtitle = element_text(size = 16, color = "grey40",  
                                      margin = margin(5, 0, 35, 0)),  
        plot.caption = element_text(color = "grey40", face = "bold", size = 11),  
        plot.margin = margin(30, 30, 30, 30),  
        plot.background = element_rect(color = "black", size = 1.2),  
        panel.grid.major = element_line(color = "grey93"),  
        panel.grid.minor = element_blank(),  
        legend.position = "bottom",  
        legend.text = element_text(size = 13, margin = margin(10, 6.5, 0, 6.5)),  
        axis.text = element_text(family = "Roboto Mono", size = 14),  
        axis.title.x = element_text(margin = margin(t = 12)),  
        axis.title.y = element_text(margin = margin(r = 12))))
```

# The Structure of `ggplot2`

**"New Beetle" and "Jetta" by Volkswagen have the worst car efficiency**

Linear regression of engine displacement versus highway miles per gallon (MPG) for 38 popular models of cars from 1999 to 2008.



Visualization: Cédric Scherer • Data: EPA ([www.fueleconomy.gov](http://www.fueleconomy.gov))

# The Structure of `ggplot2`

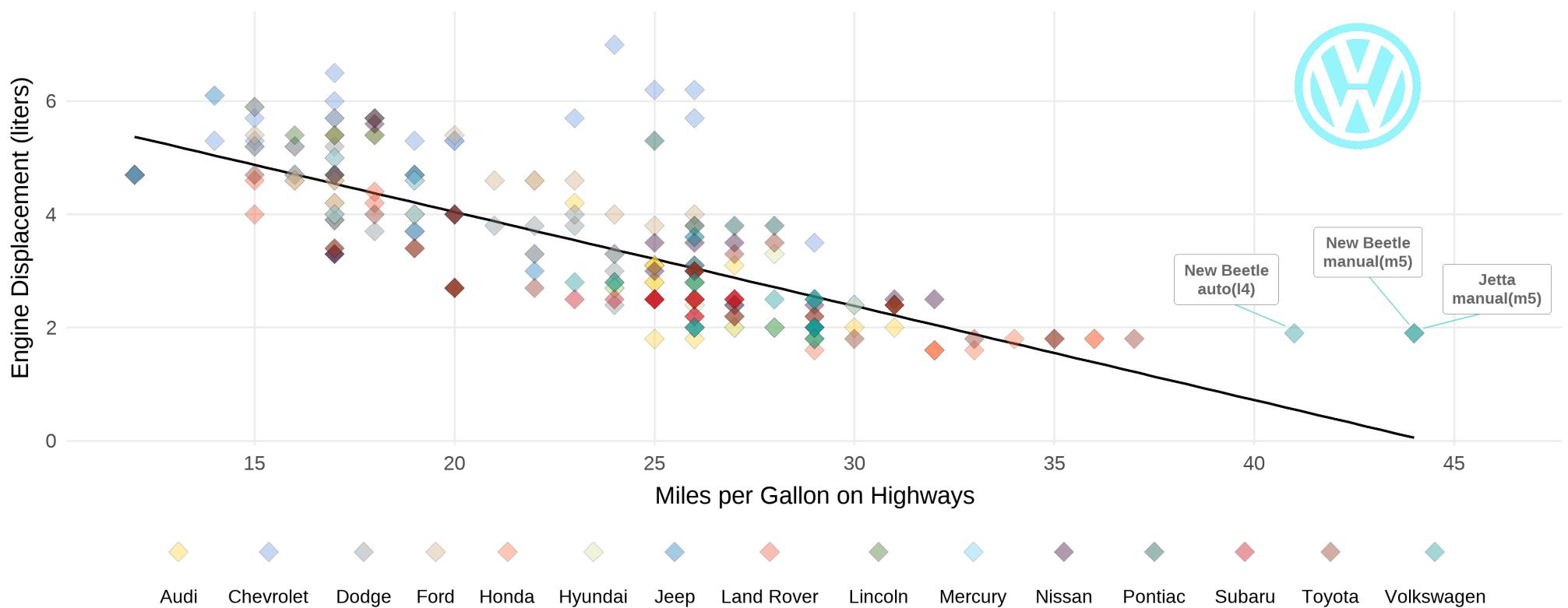
```
g <- g +
  ggrepel::geom_label_repel(data = filter(mpg, hwy > 40),
                             aes(hwy, displ,
                                 group = model,
                                 label = glue::glue("{str_to_title(model)}\n{trans}")),
                             nudge_x = .15,
                             nudge_y = 1,
                             color = "grey40",
                             fontface = "bold",
                             lineheight = 1,
                             box.padding = .5,
                             label.padding = .5,
                             point.padding = .9,
                             segment.color = "#7adb6")

cowplot::ggdraw(g) +
  cowplot::draw_image(png::readPNG("img/vw-logo-blue.png"),
                     x = .924, y = .8,
                     hjust = 1, vjust = 1,
                     width = .15, height = .15)
```

# The Structure of ggplot2

**"NewBeetle" and "Jetta" by Volkswager have the worst car efficiency**

Linear regression of engine displacement versus highway miles per gallon (MPG) for 38 popular models of cars from 1999 to 2008.



Visualization: Cédric Scherer • Data: EPA ([www.fueleconomy.gov](http://www.fueleconomy.gov))