Using Legends, Colors, Fonts, and Axes to Improve Visualizations

Dr Austin R Brown

Kennesaw State University

Introduction

- In the last section, we learned to use bar charts and dot charts to communicate the quantities observed between categorical groups.

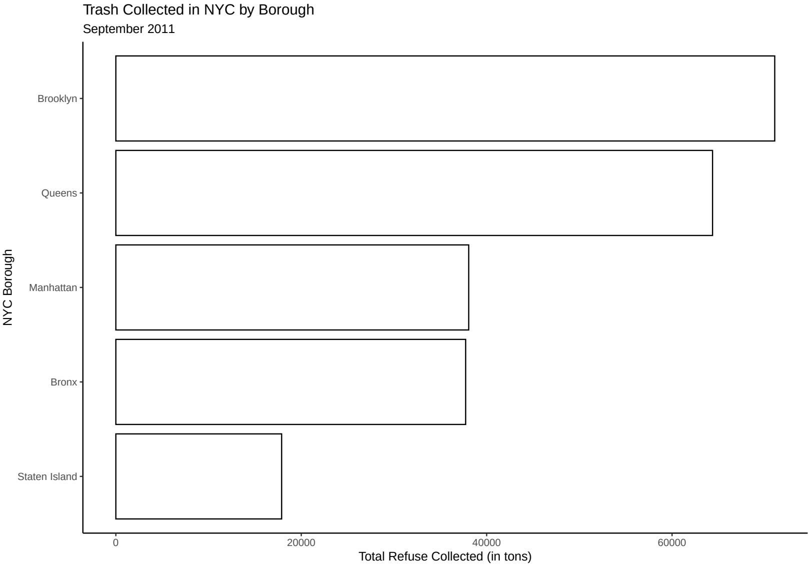
- We saw how small modifications to our `ggplot2` code can significantly improve the interpretability and aesthetic quality of hte visualization using color and plot themes.

- In this section, we will take that further by learning how to leverage `ggplot2` code to create and modify legends, axis elements, colors, color palettes, text, and fonts.

- Let's being by using some text to improve our NYC Garbage visualization.

Annotating Visualizations with Text

- Recall where we left off with our horizontal bar chart:



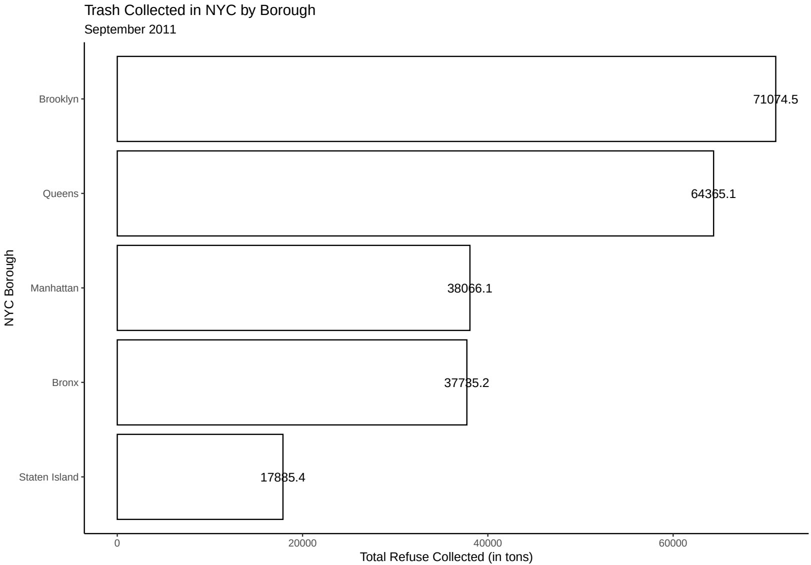
Annotating Visualizations with Text

- We can generally see from this visualization that Staten Island produced the least amount of garbage for the month of September 2011 and Brooklyn produced the most.
- We can also generally determine the amount of garbage collected. For instance, Staten Island was generally around 20K tons, whereas Queens and Brooklyn were somewhat more than 60K tons.
- It might be helpful if we put the actual amount associated with each borough on the bars themselves to increase the amount of information the reader can glean from the visualization.
  - To do this, we can make use of a new geom: `geom_text`

Annotating Visualizations with Text

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=Sum_Trash)) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic()
```

Annotating Visualizations with Text



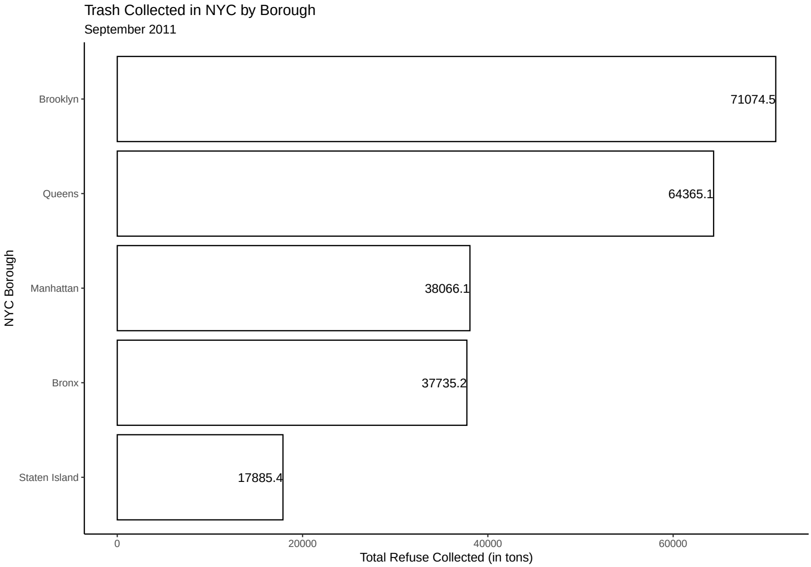
Annotating Visualizations with Text

- Okay cool! But what is the most obvious problem with this visualization?
- The label is centered at the end of the bar, making the text difficult to read.
    - We can change the justification of the text by using the `hjust` argument. This argument allows us to horizontally adjust the alignment of our text labels.
- Positive values of `hjust` move the labels to the right whereas negative values move it to the right.
    - Let's try `hjust=1` to move the labels to the right.

Annotating Visualizations with Text

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=Sum_Trash),hjust = 1) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic()
```

Annotating Visualizations with Text



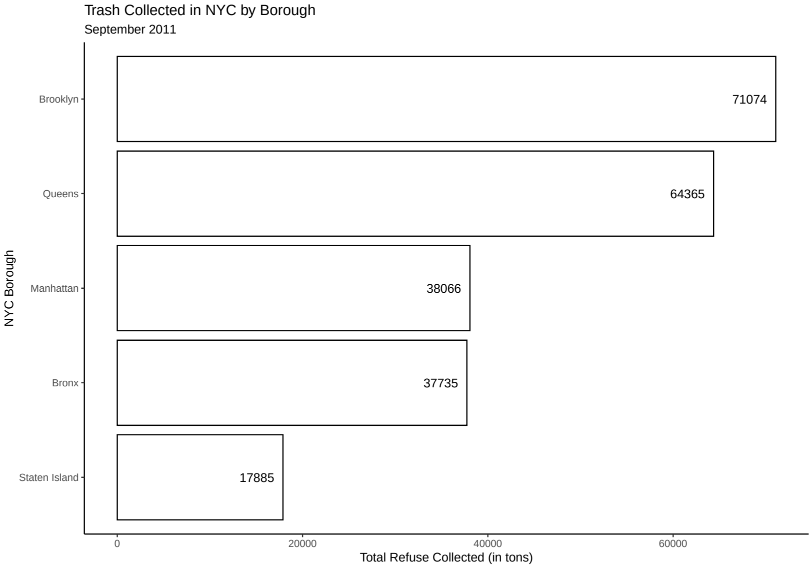
Annotating Visualizations with Text

- Looks better! But now notice that the label itself is rounded to just one decimal place, which seems unusual.
- We can fix this, and also move the label more inside of the bar directly within `geom_text`:

Annotating Visualizations with Text

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),hjust=1.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic()
```

Annotating Visualizations with Text



Modifying Font Characteristics

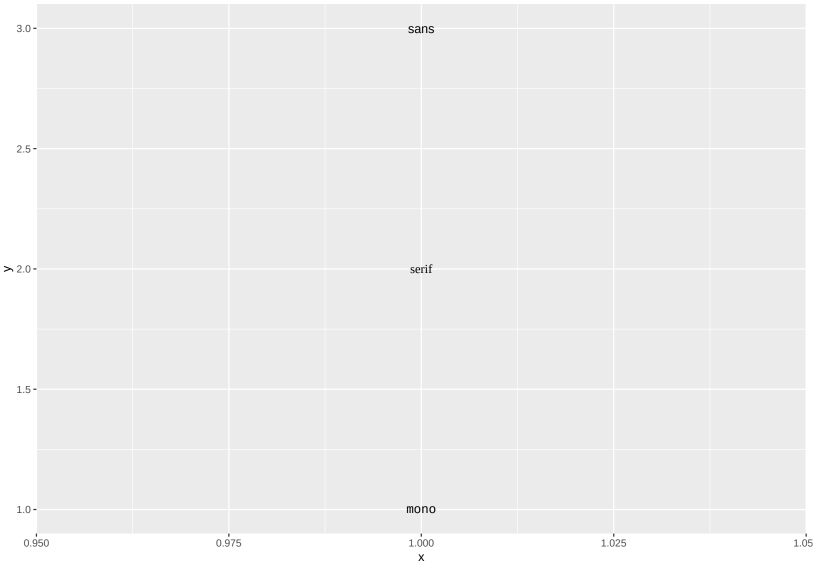
- Our visualization is much improved over what we had originally created!
- But consider font style. Right now, our visualization uses a sans serif style font in a black color by default.
- What if we wanted to change that?

Modifying Font Characteristics: Font Family

- While we can specify a wide variety of fonts, the main three guaranteed to work everywhere in a ggplot2 visualization are sans (default), serif (like Times New Roman), and mono (like typewriter font):

```r
df <- data.frame(x = 1, y = 3:1,
                 family = c("sans", "serif", "mono"))
df |>
  ggplot(aes(x, y)) +
  geom_text(aes(label = family,
                family = family))
```

Modifying Font Characteristics: Font Family

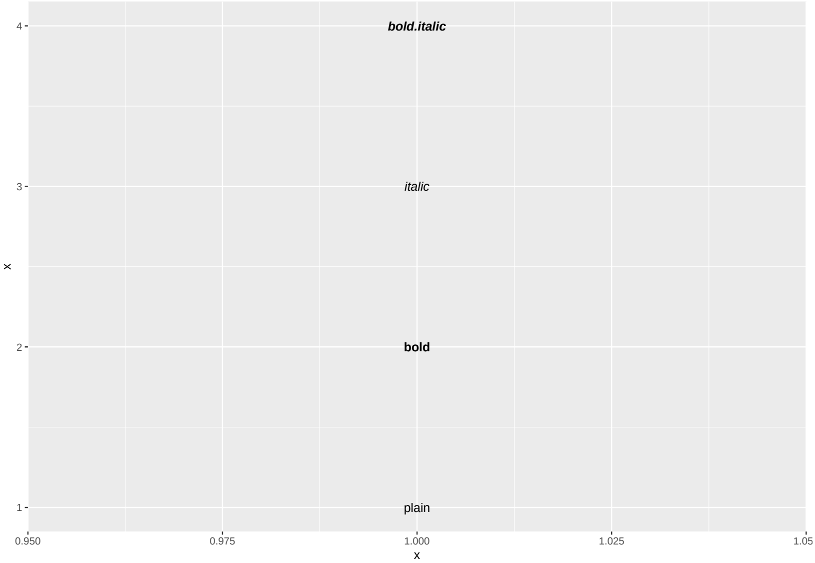


Modifying Font Characteristics: Font Face

- We can also make our fonts **bold**, *italic*, ***bold.italic*** or plain:

```r
df <- data.frame(x = 1:4,
                 fontface = c("plain",
                              "bold",
                              "italic",
                              "bold.italic"))
df |>
  ggplot(aes(1, x)) +
  geom_text(aes(label = fontface,
                fontface = fontface))
```

Modifying Font Characteristics: Font Face

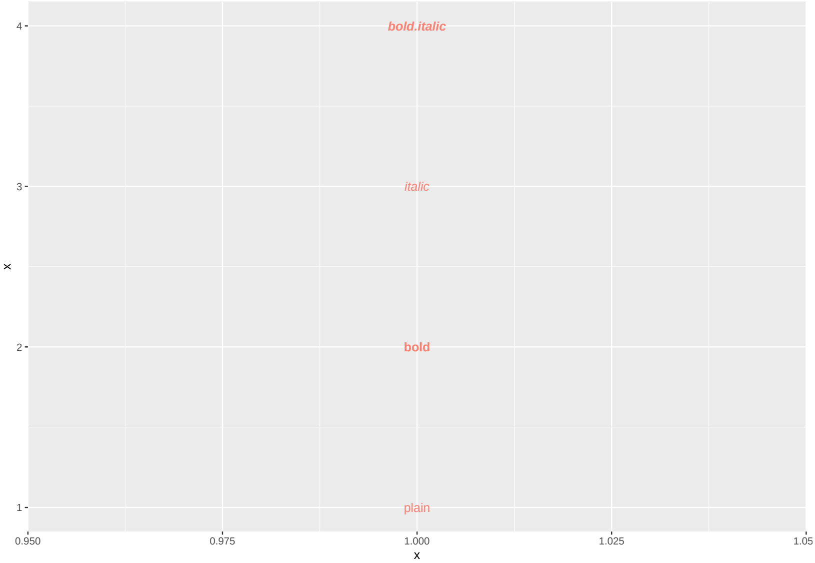


Modifying Font Characteristics: Font Color

- We can modify the color of our text uniformly by using the `color` argument within the `geom_text` function by either using a named color (see the `colors()` function for the full list) or using hex codes:

```
## Salmon Font ##
df |>
  ggplot(aes(1, x)) +
  geom_text(aes(label = fontface,
                fontface = fontface),
            color='salmon')
```

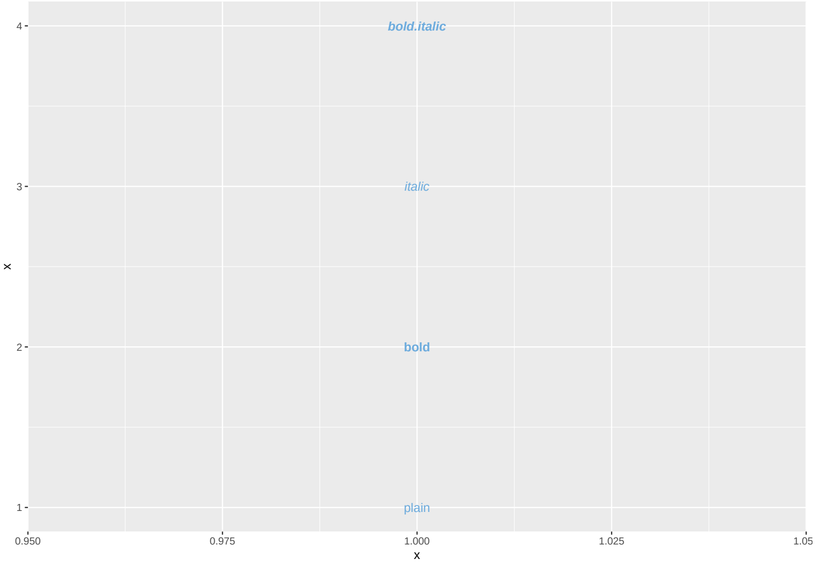
Modifying Font Characteristics: Font Color



Modifying Font Characteristics: Font Color

```r
## Manchester City Light Blue ##
df |>
  ggplot(aes(1, x)) +
  geom_text(aes(label = fontface,
                fontface = fontface),
            color = "#6CABDD")
```

Modifying Font Characteristics: Font Color

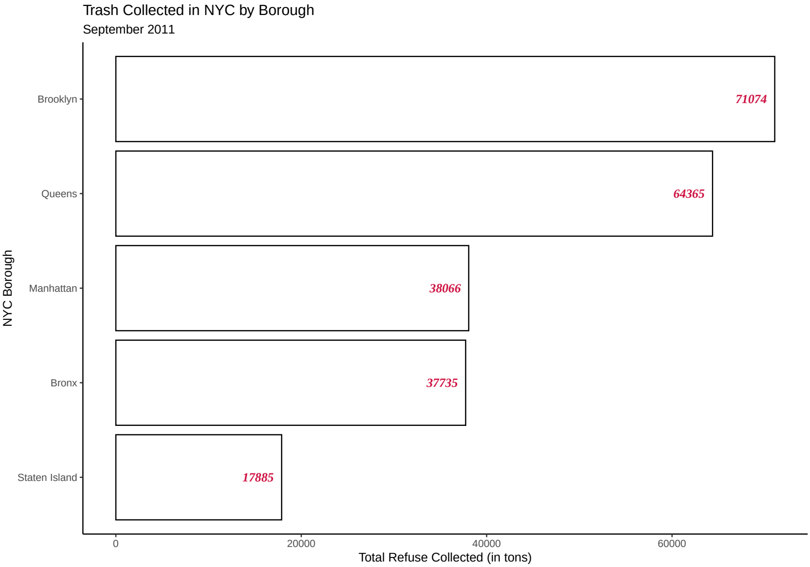


Modifying Font Characteristics: Font Color

- For our NYC Garbage example, suppose I want the text to be serif style with a bold italic font face, and in Atlanta Braves red (hex code #CE1141):

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',hjust=1.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic()
```

Modifying Font Characteristics: Font Color

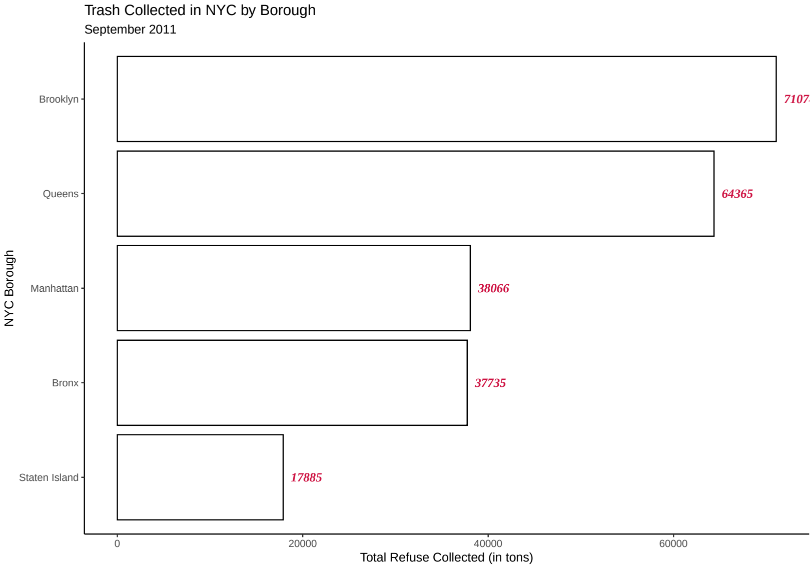


Modifying Axes Elements: Axis Length

- Still using our NYC Garbage example, let's suppose I'd rather have the text labels outside of the bars to the right rather than inside the bars to the left.
- Remember, we can make a simple change to our `hjust` argument to do this:

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',
            hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic()
```

Modifying Axes Elements: Axis Length



Modifying Axes Elements: Axis Length

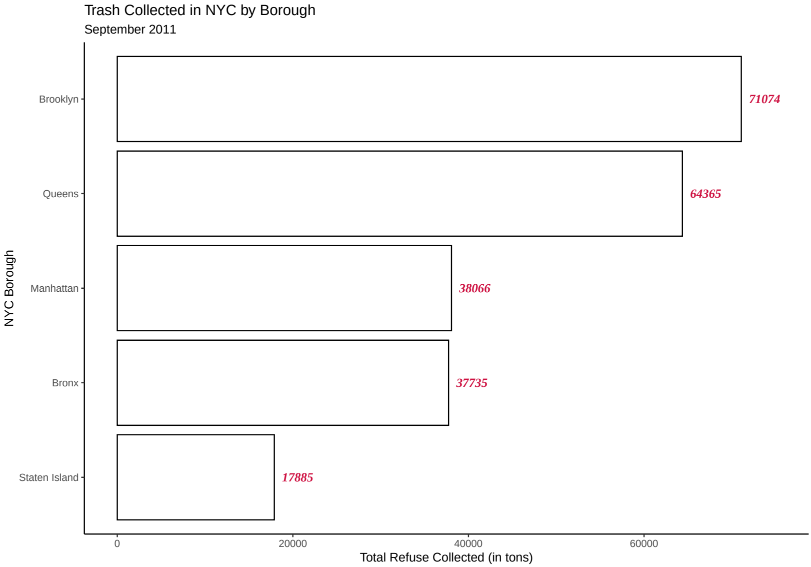
- Whoops! Now I can't see Brooklyn's label! It's being truncated by the size of our viewing window.

- One way this can be modified is by increasing the x-axis length. We can do this by using the `limits` argument within the `scale_x_continuous` function.

Modifying Axes Elements: Axis Length

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',
            hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  scale_x_continuous(limits = c(0,75000))
```

Modifying Axes Elements: Axis Length



Modifying Axes Elements: Tick Marks

- Notice in the prior visualization that our tick marks on the x-axis are in increments of 20,000.

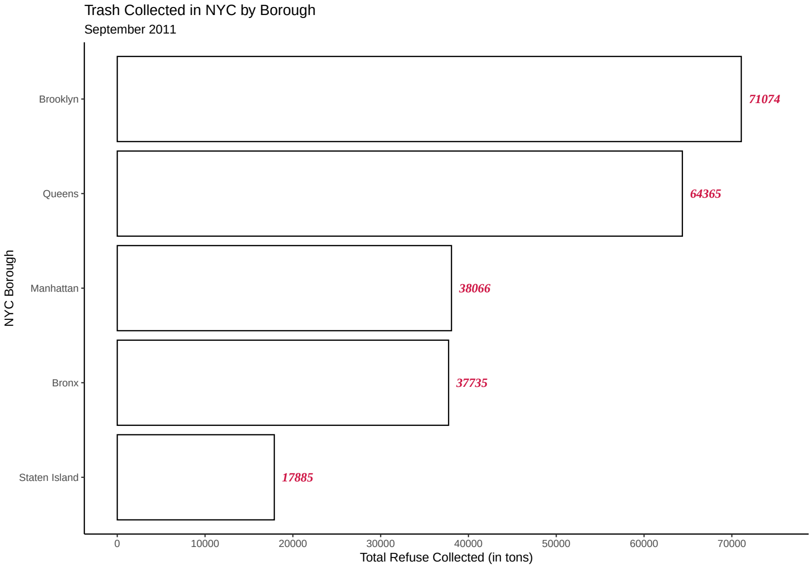
- What if we want to increase the number of tick marks to be in increments of 10,000 instead?

- We can again use `scale_x_continuous` this time making use of the `breaks` argument.

Modifying Axes Elements: Tick Marks

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',
            hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  scale_x_continuous(limits = c(0,75000),
                     breaks = seq(0,80000,
                                  by=10000))
```

Modifying Axes Elements: Tick Marks



Modifying Axes Elements: Tick Marks

- In the above visualization, we note that each tick mark represents a unit measured in the thousands as we can see by the three trailing zeros in each tick mark label.

- We may perhaps wish to represent “thousand” by the common label “K” so that $10000 = 10$K.

- To do this using `ggplot2`, we can once again use the very useful `scale_x_continuous` function, now adding a new element of functionality – the `labels` function:

Modifying Axes Elements: Tick Marks

```
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',
            hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  scale_x_continuous(limits = c(0,75000),
                     breaks = seq(0,80000,
                                  by=10000),
                     labels = scales::label_number(
                       suffix = "K",
                       scale = 1e-3))
```

Modifying Axes Elements: Tick Marks

- We already learned how to modify font styles in the context of `geom_text`, but we can use the exact same logic and syntax to modify font in our axes as well as titles!

- So suppose we want our tick mark labels and axis titles to match the formatting of our data labels.

- We can do this by using the `theme` function in conjunction with the `element_text` function.

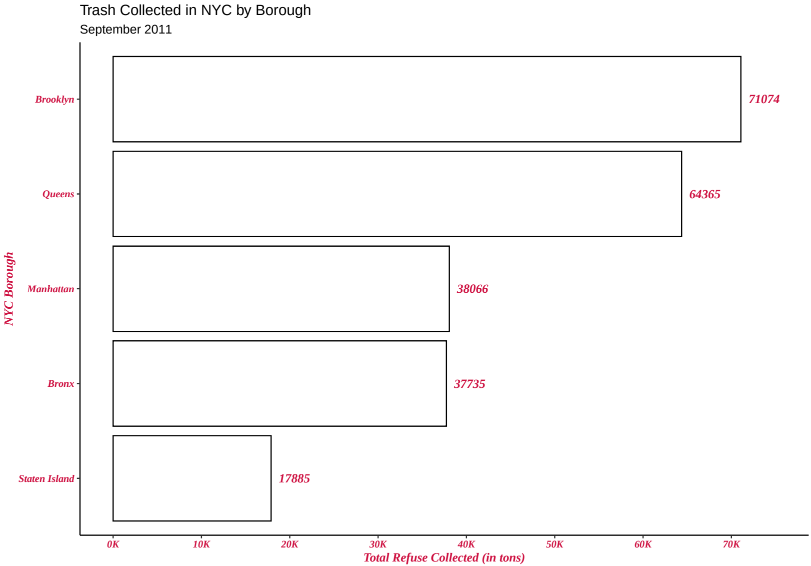
Modifying Axes Elements: Tick Marks

```
## First, Save the Plot to an Object, p ##
p <- trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash))) +
  geom_bar(stat='identity',color='black',fill='white') +
  geom_text(aes(label=round(Sum_Trash)),
            family='serif',
            fontface='bold.italic',
            color='#CE1141',
            hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  scale_x_continuous(limits = c(0,75000),
                     breaks = seq(0,80000,
                                  by=10000),
                     labels = scales::label_number(
                       suffix = "K",
                       scale = 1e-3))
```

Modifying Axes Elements: Tick Marks

```
## Now, Modify the Font of the Axes and Titles ##
p +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic',
                                   color = '#CE1141'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic',
                                    color = '#CE1141'),
        axis.text.y = element_text(family = 'serif',
                                   face = 'bold.italic',
                                   color = '#CE1141'),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic',
                                    color = '#CE1141'))
```

Modifying Axes Elements: Tick Marks



Legends

- Rather than having the bars all be a uniform color (white in this case), suppose I want to have the colors of the bars differ by the particular borough they're representing.

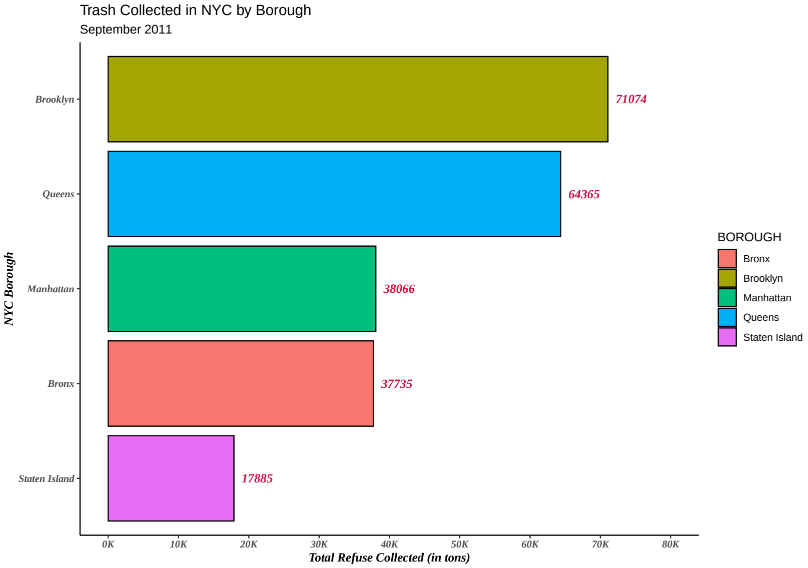
- We can do so with a very slight modification to the existing code.

- In the global `ggplot` function, let's add `fill=BOROUGH`:

Legends

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),
             fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic')) +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3))
```

Legends



Trash Collected in NYC by Borough
September 2011
Brooklyn
Queens
Manhattan
Bronx
Staten Island
71074
64365
38066
37735
17885
NYC Borough
0K
10K
20K
30K
40K
50K
60K
70K
80K
Total Refuse Collected (in tons)
BOROUGH
Bronx
Brooklyn
Manhattan
Queens
Staten Island

Legends

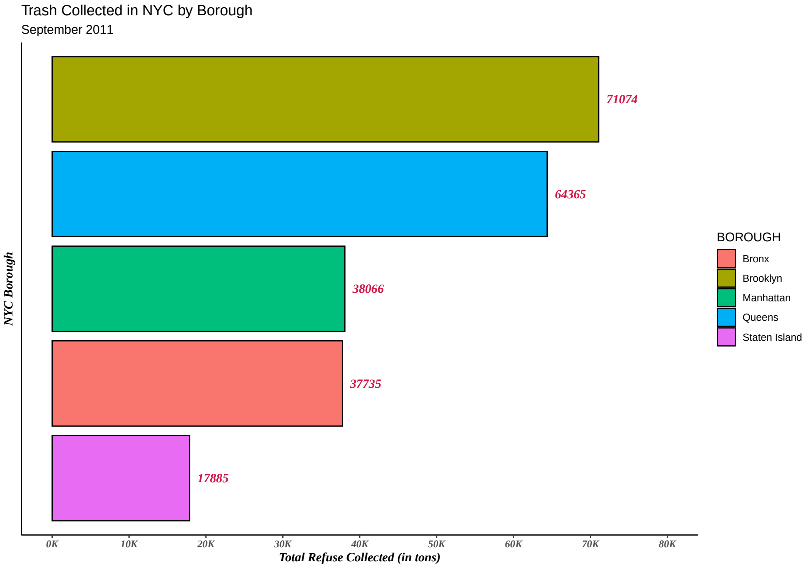
- Cool right? But now, we don't really have a need for the y-axis labels.

- We can supress those and the tick marks using the `theme` function:

Legends

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),
             fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank()) +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3))
```

Legends


Trash Collected in NYC by Borough
September 2011
NYC Borough
71074
64365
38066
37735
17885
0K
10K
20K
30K
40K
50K
60K
70K
80K
Total Refuse Collected (in tons)
BOROUGH
Bronx
Brooklyn
Manhattan
Queens
Staten Island

Legends

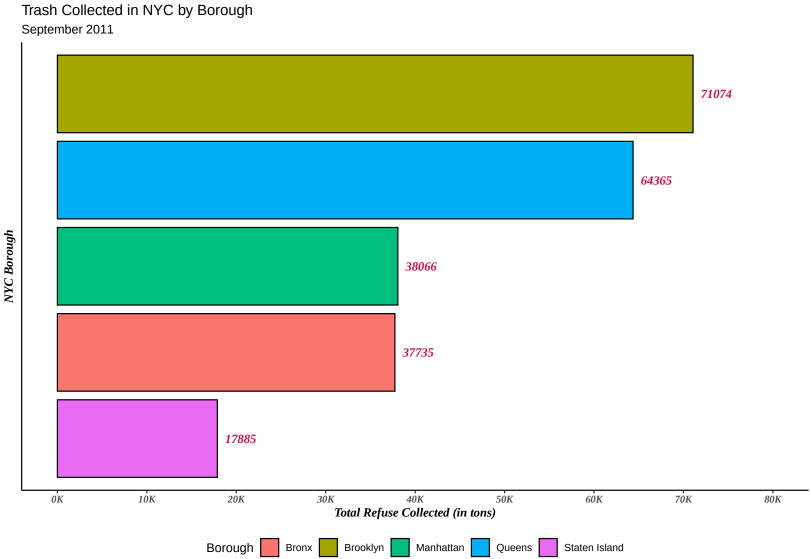
- We can also modify the legend title using the `labs` function.

- Also notice that the legend is positioned to the right of the plot by default. We can change this using the `legend.position` argument within `theme` function.
    - We can place it at the `top`, `bottom`, `left`, or `right` of the plot, or we can suppress it entirely by using `legend.position = "none"`.
    - Let's try placing the legend at the bottom of the plot.

Legends

```
## Bottom ##
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011",
       fill = "Borough") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank(),
        legend.position = "bottom") +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3))
```

Legends


Trash Collected in NYC by Borough
September 2011
NYC Borough
71074
64365
38066
37735
17885
0K
10K
20K
30K
40K
50K
60K
70K
80K
Total Refuse Collected (in tons)
Borough
Bronx
Brooklyn
Manhattan
Queens
Staten Island

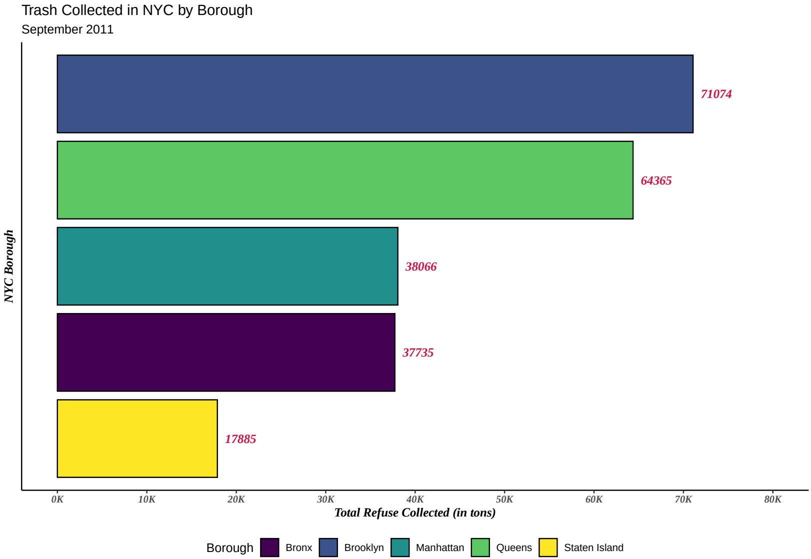
Legends: Changing Color Palettes

- In the above plot, the generated colors are the defaults.
- We can change the palette we use eithr manually or by using palettes within packages such as `viridis` (which uses colorblind-friendly palettes) or `RColorBrewer`.
- Let's check out the `viridis` package:

Legends: Changing Color Palettes

```
library(viridis)
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011",
       fill = "Borough") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank(),
        legend.position = "bottom") +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3)) +
  scale_fill_viridis(discrete = T)
```

Legends: Changing Color Palettes

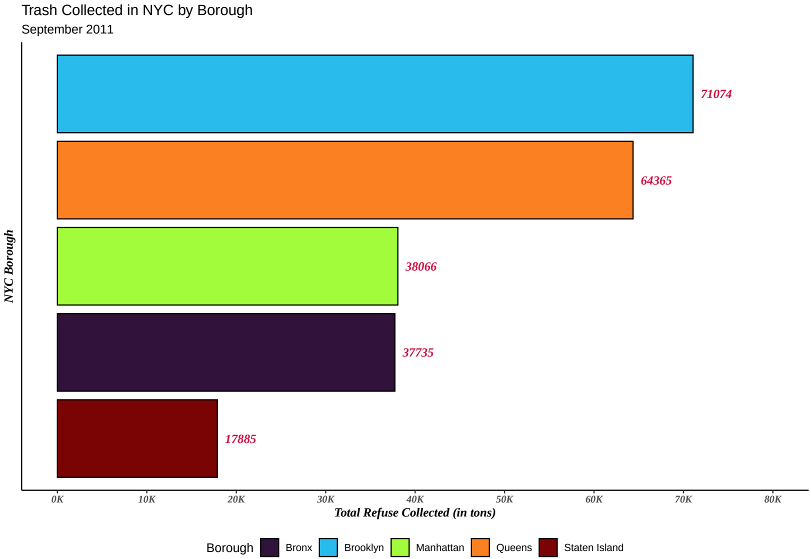


Legends: Changing Color Palettes

- Within `scale_fill_viridis`, we have eight different palettes we can specify (A - H). So for example, if I want to employ the “turbo” option (“H”):

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011",
       fill = "Borough") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank(),
        legend.position = "bottom") +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3)) +
  scale_fill_viridis(discrete = T,
                     option = "H")
```

Legends: Changing Color Palettes



Legends: Changing Color Palettes

- The RColorBrewer package also provides a nice set of palettes we can use to customize our visualizations.

- We can take a look at the possibilities by:

```
library(RColorBrewer)
print(brewer.pal.info)
```

Legends: Changing Color Palettes

- The first set of palettes (labeled “div”) are best for quantitative data.

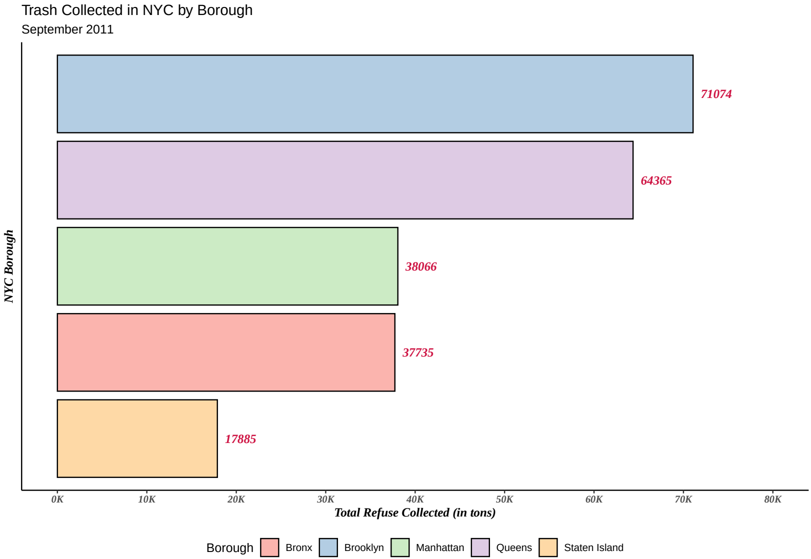
- The third set (labeled “seq”) are best for quantitative data with clear extremes.

- The middle set (labeled “qual”) are best for qualitative data.
  - Let’s try `Pastel1` from the “qual” set.

Legends: Changing Color Palettes

```
library(RColorBrewer)
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011",
       fill = "Borough") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank(),
        legend.position = "bottom") +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3)) +
  scale_fill_brewer(palette = "Pastel1")
```

Legends: Changing Color Palettes



Legends: Custom Color Palettes

- Being able to use existing color palettes in R packages like `viridis` and `RColorBrewer` is great, but what if we want to create our own custom color palette?
  - For example, we need to use our company's colors.

- To do this, we can use the `scale_fill_manual` function and specify the colors we want to use in a vector.
  - Note, we can also use hex colors here rather than named colors.

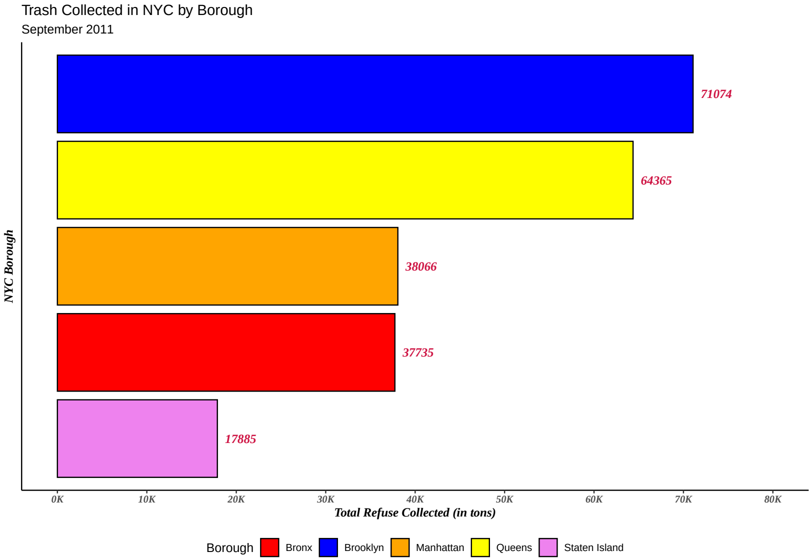
Legends: Custom Color Palettes

```r
## Specify Vector of Colors ##
borough_colors <- c("Bronx" = 'red',
                    "Brooklyn" = 'blue',
                    "Manhattan" = "orange",
                    "Queens" = "yellow",
                    "Staten Island" = 'violet')
```

Legends: Custom Color Palettes

```r
trash_tot |>
  ggplot(aes(x=Sum_Trash,y=reorder(BOROUGH,Sum_Trash),fill=BOROUGH)) +
  geom_bar(stat='identity',color='black') +
  geom_text(aes(label=round(Sum_Trash)),family='serif',
            fontface='bold.italic',color='#CE1141',hjust=-0.25) +
  labs(y = "NYC Borough",
       x = "Total Refuse Collected (in tons)",
       title = "Trash Collected in NYC by Borough",
       subtitle = "September 2011",
       fill = "Borough") +
  theme_classic() +
  theme(axis.text.x = element_text(family = 'serif',
                                   face = 'bold.italic'),
        axis.title.x = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.text.y = element_blank(),
        axis.title.y = element_text(family = 'serif',
                                    face = 'bold.italic'),
        axis.ticks.y = element_blank(),
        legend.position = "bottom") +
  scale_x_continuous(limits = c(0,80000),
                     breaks = seq(0,80000,by=10000),
                     labels = scales::label_number(suffix = "K",
                                                   scale = 1e-3)) +
  scale_fill_manual(values = borough_colors)
```

Legends: Custom Color Palettes


Trash Collected in NYC by Borough
September 2011
NYC Borough
0K
10K
20K
30K
40K
50K
60K
70K
80K
Total Refuse Collected (in tons)
Borough
Bronx
Brooklyn
Manhattan
Queens
Staten Island