

# Make Your First Bar Chart

There are plenty of datasets built into R and thousands of others available online. Still, you'll declare your

Here's the dataset you'll use today:

```
library(ggplot2)

data <- data.frame(
  quarter=c("Q1", "Q1", "Q2", "Q2", "Q3", "Q3", "Q4", "Q4"),
  product=c("A", "B", "A", "B", "A", "B", "A", "B"),
  profit=c(10, 14, 12, 11, 13, 15, 16, 18)
)
```

R's standard library for data visualization is `ggplot2`. It's based on the layering principle. For example, you see later how additional layers can make charts more informative and appealing.

To start, you'll make a bar chart that has the column `quarter` on the x-axis and `profit` on the y-axis. The `geom_bar` and `geom_col` layers are used to create bar charts. With the first option, you r

You can create a simple bar chart with this code:

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col()
```

Here's the corresponding visualization:

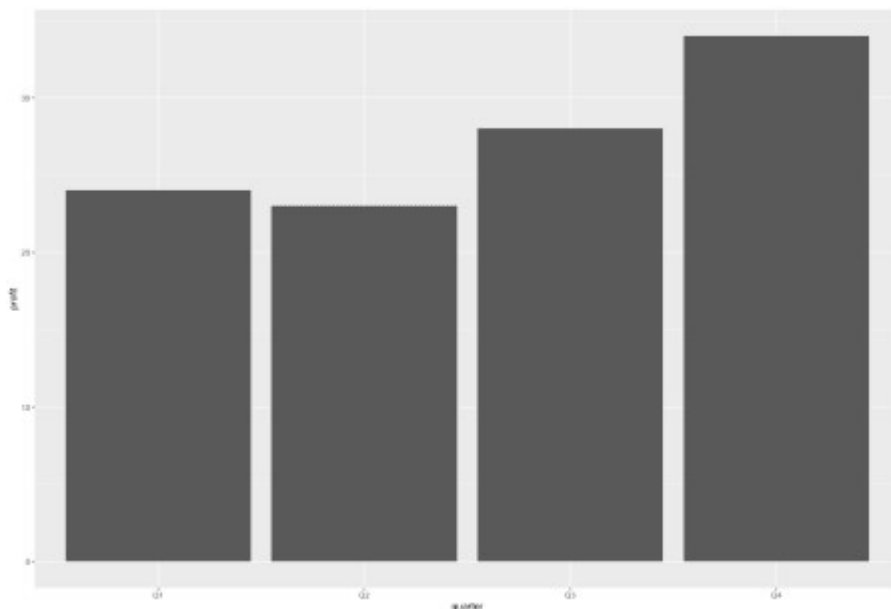


Image 1 – Simple bar chart

This one gets the job done but doesn't look like something you'd want to show to your boss. You'll fix it in

## Colors and Themes

Tweaking colors and themes is the simplest thing you can do to make visualization look better. The `geom`

- `color` – outline color of the bars

- `fill` – fill color of the bars

Here's how to use `fill` to make your chart Appsilon-approved:

```
ggplot(data, aes(x = quarter, y = profit)) +  
  geom_col(fill = "#0099f9")
```

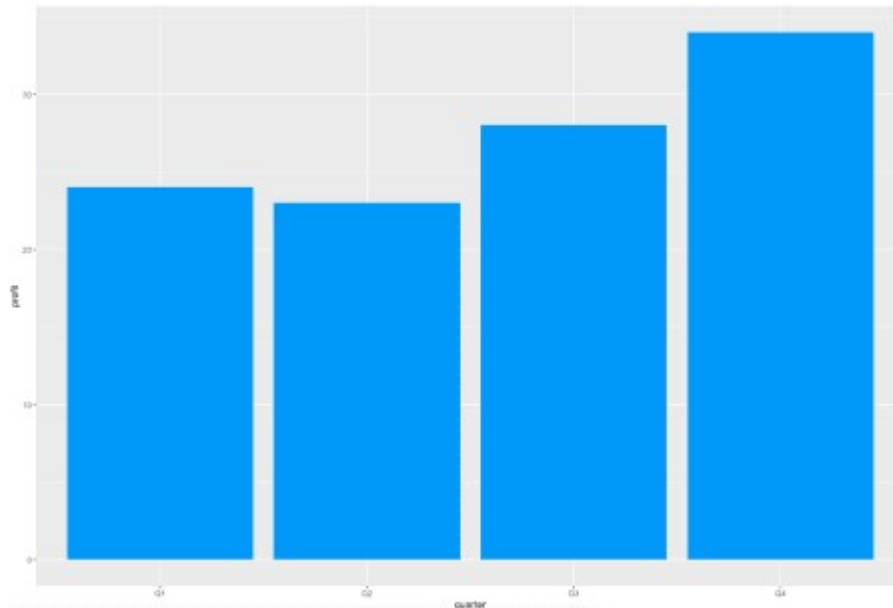


Image 2 – Using `fill` to change the bar color

The `color` parameter changes only the outline. The dataset you're using has two distinct products. R dr: stacked charts later.

Here's what this means in practice. The code snippet below sets the fill color to white and outline color to

```
ggplot(data, aes(x = quarter, y = profit)) +  
  geom_col(color = "#0099f9", fill = "#ffffff")
```

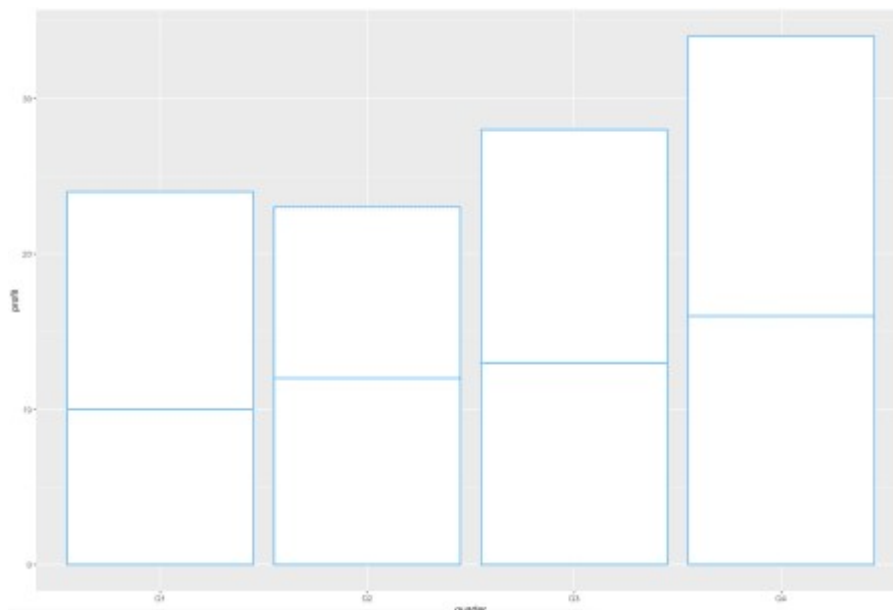


Image 3 – Changing the outline color

In case coloring doesn't do the trick, you can completely change the theme. That's yet another layer to ac

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  theme_classic()
```

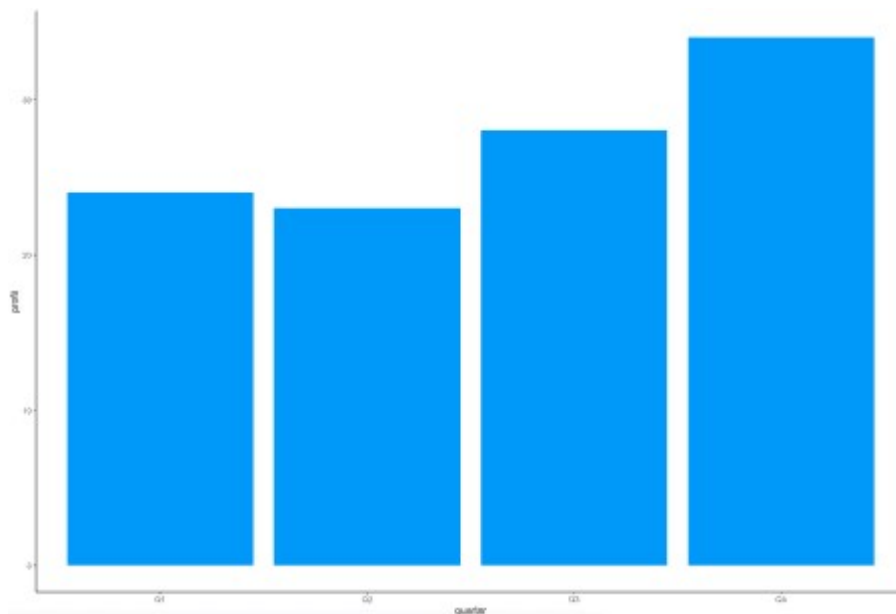


Image 4 – Changing the visualization theme

If this theme isn't your thing, there's plenty more to pick from. You can find the entire list [here](#).

## Titles, Subtitles, and Captions

A visualization without a title is useless. There's no way to know if you're looking at *Election votes* or *2020 Election results*. Captions are useful for placing visualization credits and sources.

The most convenient way to add these is through a `labs()` layer. It takes in values for `title`, `subtitl`

Let's see how to add all three:

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  labs(
    title = "Quarterly Profit (in million U.S. dollars)",
    subtitle = "A simple bar chart",
    caption = "Source: ImaginaryCo"
  )
```

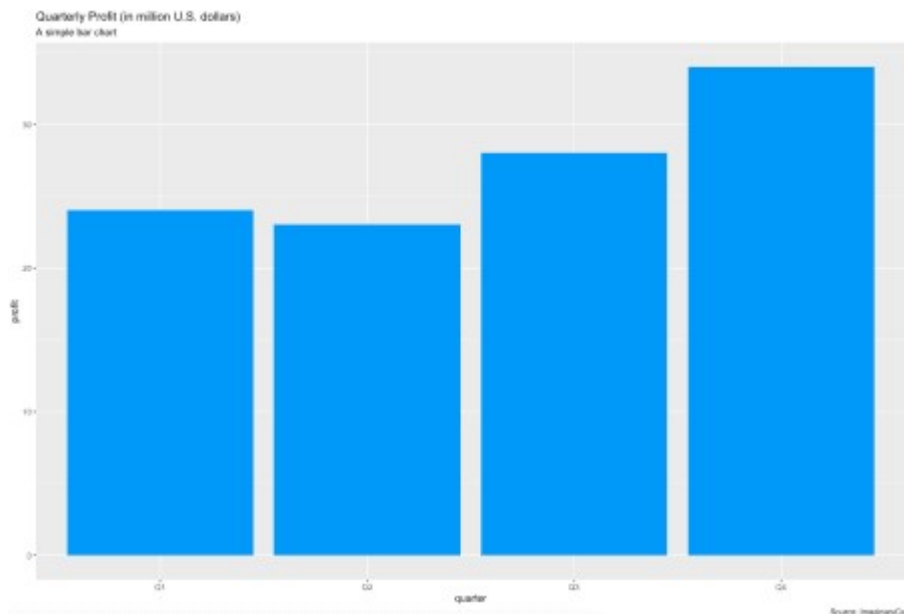


Image 5 – Title, subtitle, and caption with default styles

It's a good start, but what if you want to **add styles**? Let's see how to color the title, bold the subtitle, and

```
ggplot(data, aes(x = quarter, y = profit)) +  
  geom_col(fill = "#0099f9") +  
  labs(  
    title = "Quarterly Profit (in million U.S. dollars)",  
    subtitle = "A simple bar chart",  
    caption = "Source: ImaginaryCo"  
  ) +  
  theme(  
    plot.title = element_text(color = "#0099f9", size = 20),  
    plot.subtitle = element_text(face = "bold"),  
    plot.caption = element_text(face = "italic")  
  )
```

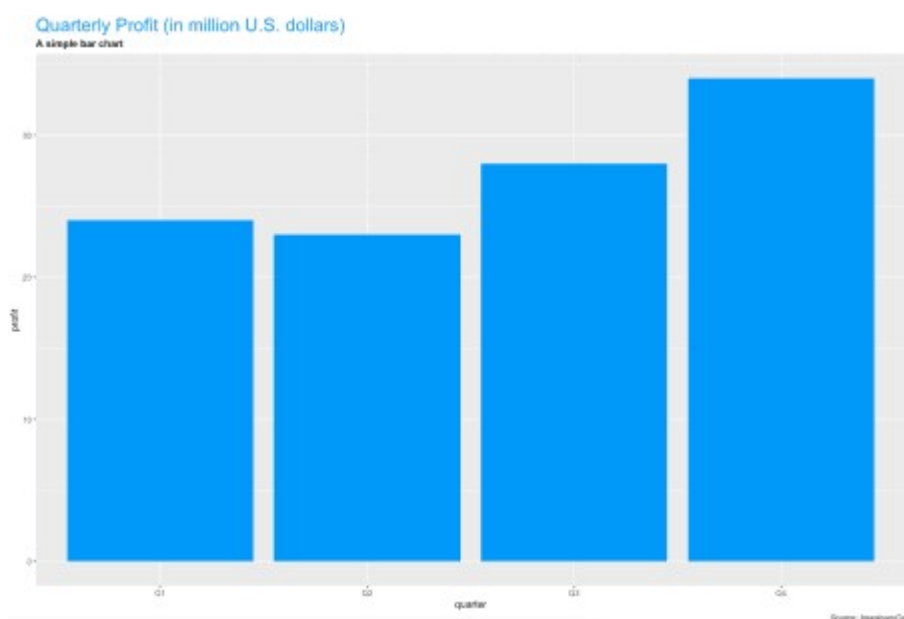


Image 6 – Styling title, subtitle, and caption

Let's take this a step further. Here's how to align the title to the middle, subtitle to the right, and caption to

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  labs(
    title = "Quarterly Profit (in million U.S. dollars)",
    subtitle = "A simple bar chart",
    caption = "Source: ImaginaryCo"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 1),
    plot.caption = element_text(hjust = 0)
  )
```

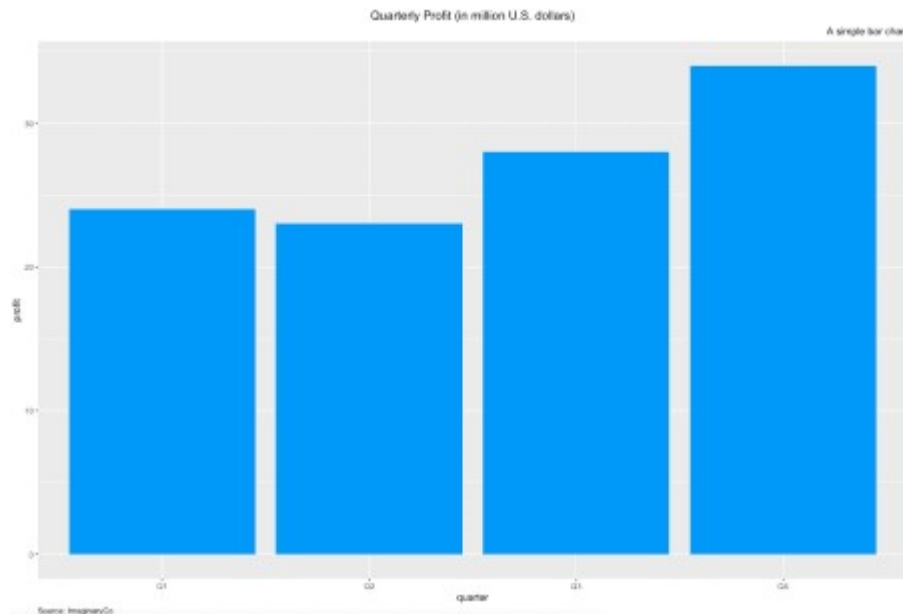


Image 7 – Aligning title, subtitle, and caption

You’ve learned how to add a nicely-formatted title, but the default axis labels still holds your visualization

## Axis Labels

Long story short – it works identically as with titles and subtitles. The `labs()` layer takes in values for bo

Here’s how to change the text:

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  labs(
    x = "Quarter of 2020",
    y = "Profit in 2020"
  )
```

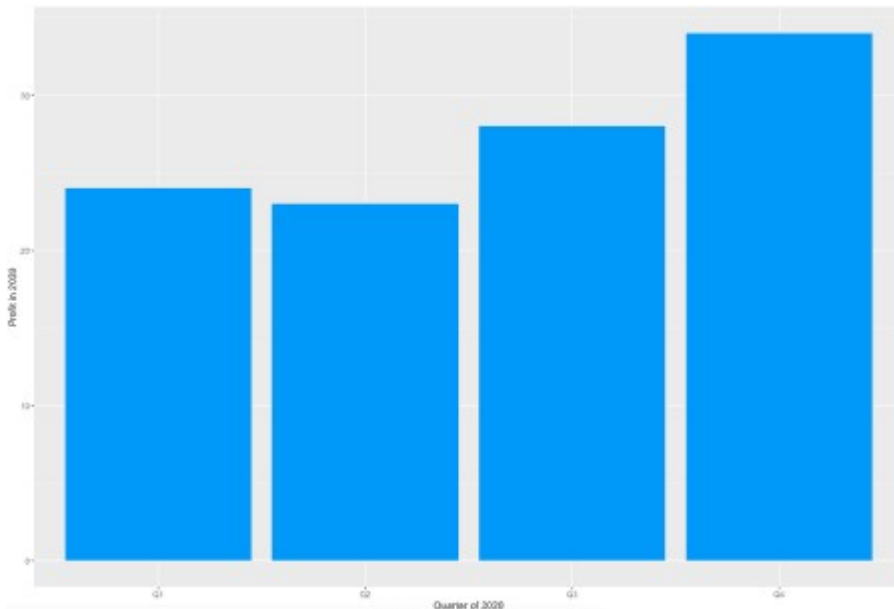


Image 8 – Changing X and Y axis labels

You can change the styles the same way you did with titles, subtitles, and captions. The following code snippet

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  labs(
    x = "Quarter of 2020",
    y = "Profit in 2020"
  ) +
  theme(
    axis.title.x = element_text(color = "#0099f9", size = 15, face = "bold"),
    axis.title.y = element_text(size = 15, face = "italic")
  )
```

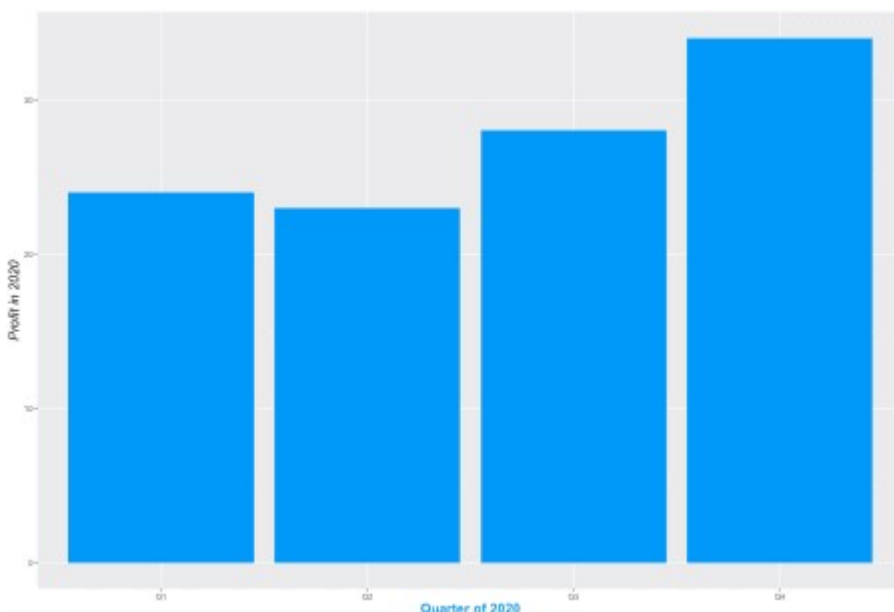


Image 9 – Changing stylings of X and Y axis labels

And that does it for changing the basic visuals. You'll learn how to work with different bar charts next – stacked

## Stacked, Grouped, and Horizontal Bar Charts

The `ggplot2` package uses stacked bar charts by default. Stacked bar charts are best used when all po

To change the coloring, you only need to change the `fill` value in the data layer. Here's an example:

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +  
  geom_col()
```

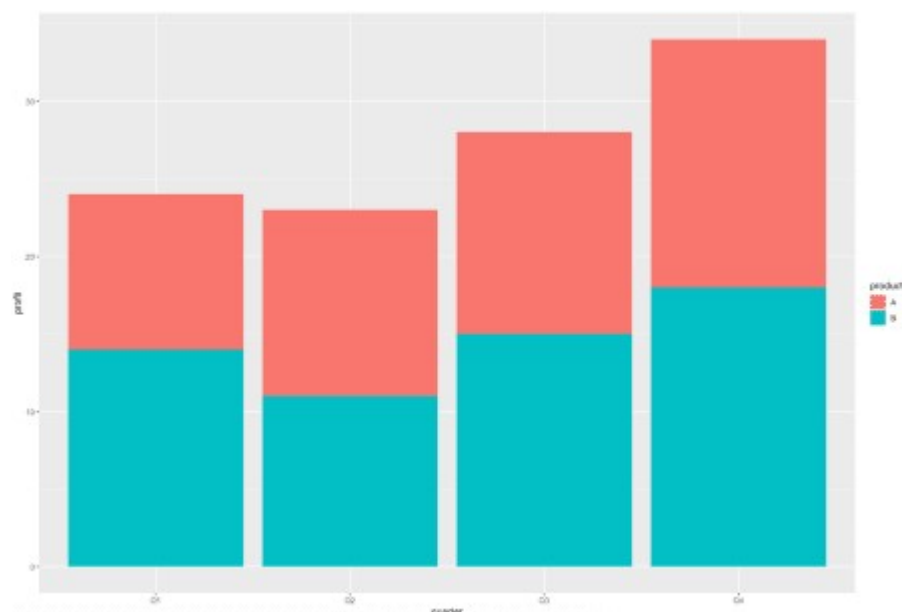


Image 10 – Default stacked bar chart

There's a visible distinction between products, and you can now see how much profit each product made

There are two ways to change each portion's color:

- **Manually** – by specifying a vector of color names or color hex codes
- **With palettes** – by using built-in color palettes

Let's cover the manual approach first. You have to add a layer with `scale_fill_manual`:

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +  
  geom_col() +  
  scale_fill_manual(values = c("#69c6ff", "#0099f9"))
```

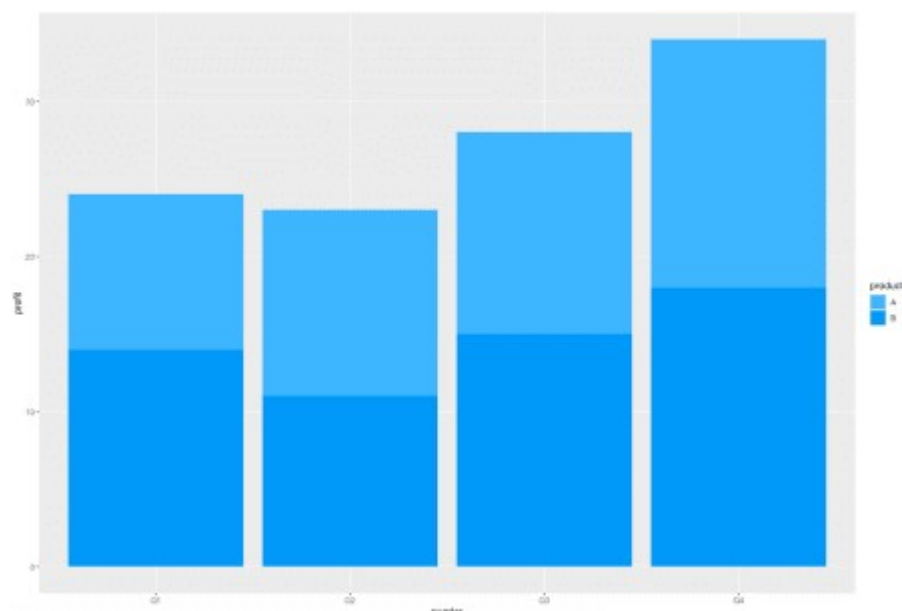


Image 11 – Stacked bar chart with custom colors

Palettes are a bit easier because you don't need to know exact color values. For the same reason, it can work with palettes:

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +  
  geom_col() +  
  scale_fill_brewer(palette = "Set1")
```

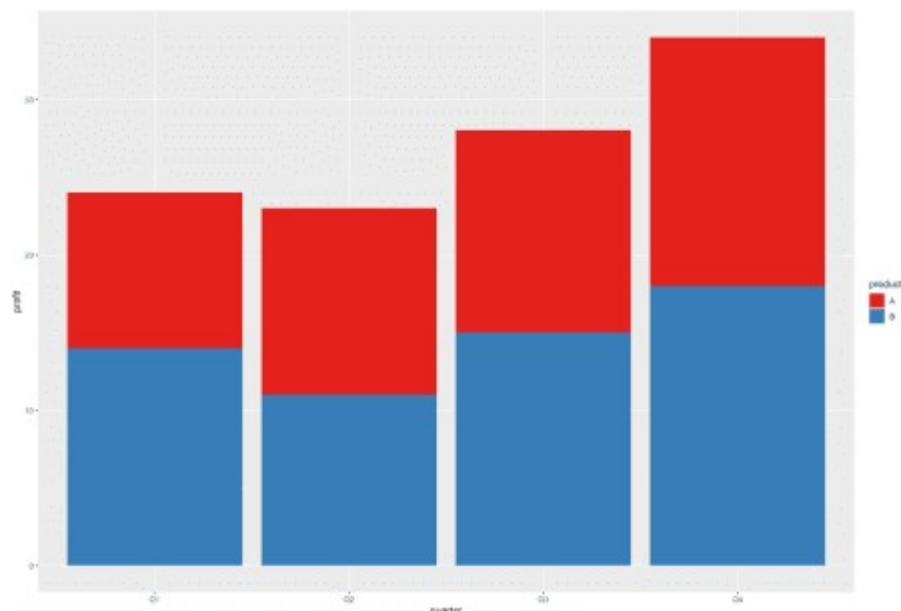


Image 12 – Stacked bar chart colored with a built-in palette

Onto the **grouped bar charts** now. They display bars corresponding to a group next to each other instead of stacked. Use the `geom_bar` layer:

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +  
  geom_bar(position = position_dodge())
```

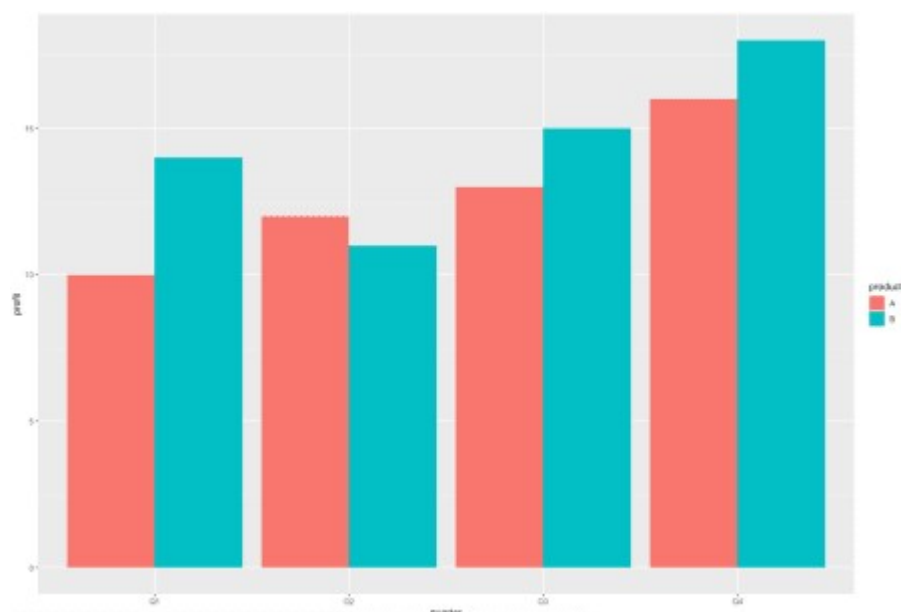


Image 13 – Grouped bar chart (default)

You can change the coloring the same way you did with stacked bar charts – through the `scale_fill_r`



```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +
  geom_col(position = position_dodge()) +
  scale_fill_manual(values = c("#3db5ff", "#0099f9"))
```

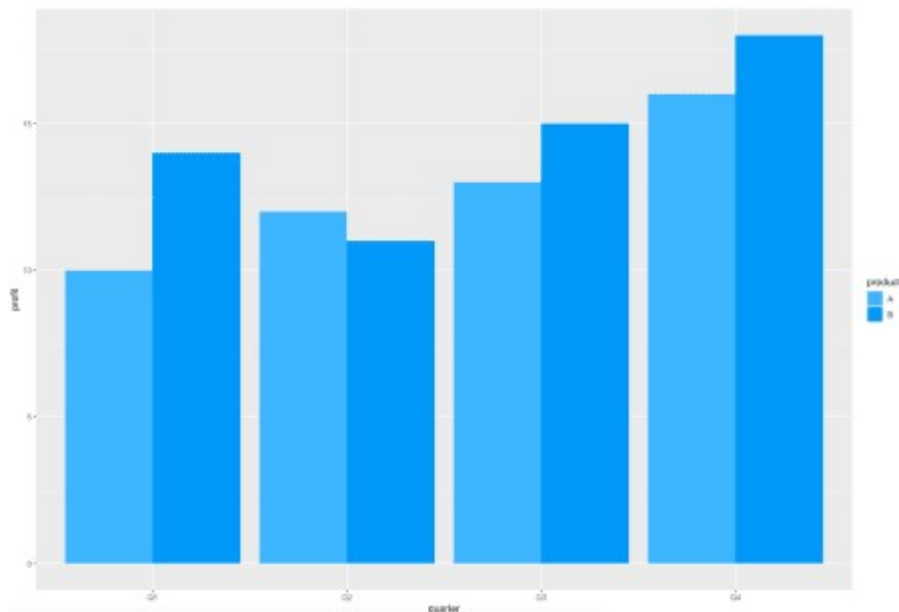


Image 14 – Grouped bar chart with custom colors

Finally, let's cover **horizontal bar charts**. They are useful when there are many categories on the x-axis

```
ggplot(data, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  coord_flip()
```

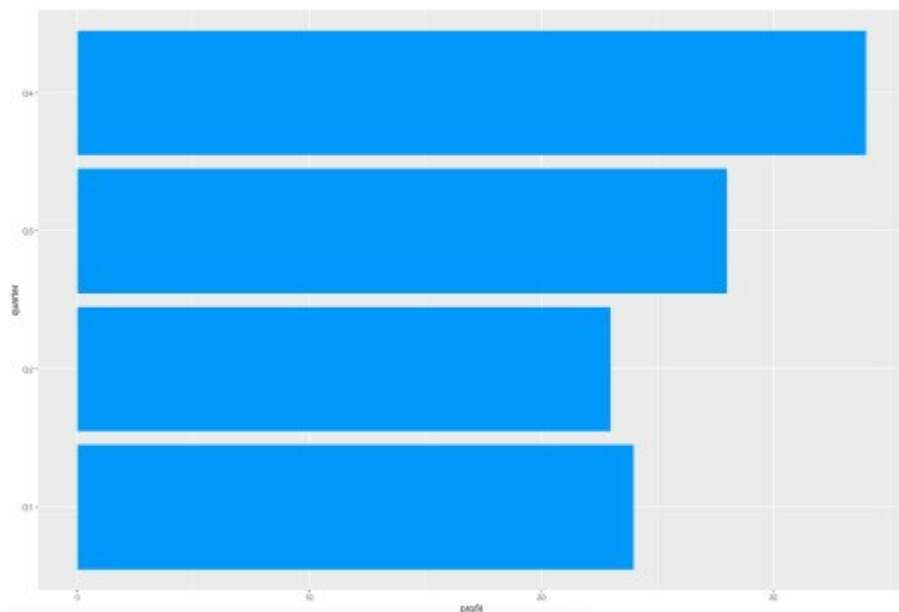


Image 15 – Horizontal bar chart (default)

You can use the `scale_fill_manual` or `scale_fill_brewer` layers to change the color. Here's an example

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +
  geom_col(position = position_dodge()) +
  scale_fill_manual(values = c("#3db5ff", "#0099f9")) +
  coord_flip()
```

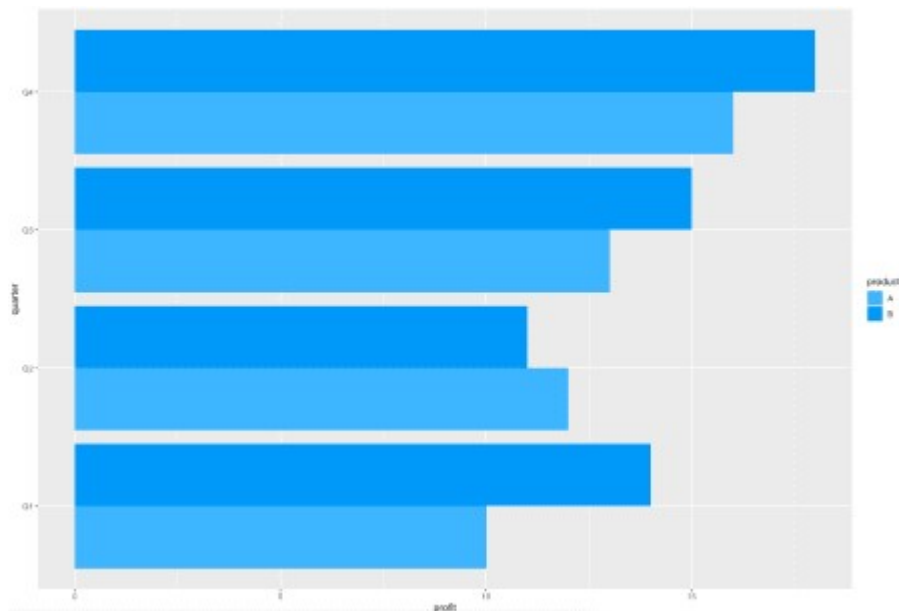


Image 16 – Horizontal bar chart with custom colors

Now you know how to make every type of bar chart – but there’s still one thing you can improve. Let’s see

## Labels

Bar charts can be hard to look at. Knowing the exact value is often a requirement. If the y-axis is on a scale

You can put text somewhere near the top of each bar to show the exact value. That solves the problem of y-axis constantly.

You’ll learn how to put labels on top of bars. For the first example, you’ll need to filter the dataset so only products in the stack for each quarter. You’ll learn how to add labels for multiple stacks later, but let’s start

Here’s the code:

```
library(dplyr)

data_a <- data %>%
  filter(product == "A")

ggplot(data_a, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  geom_text(aes(label = profit), vjust = -0.5, size = 5)
```

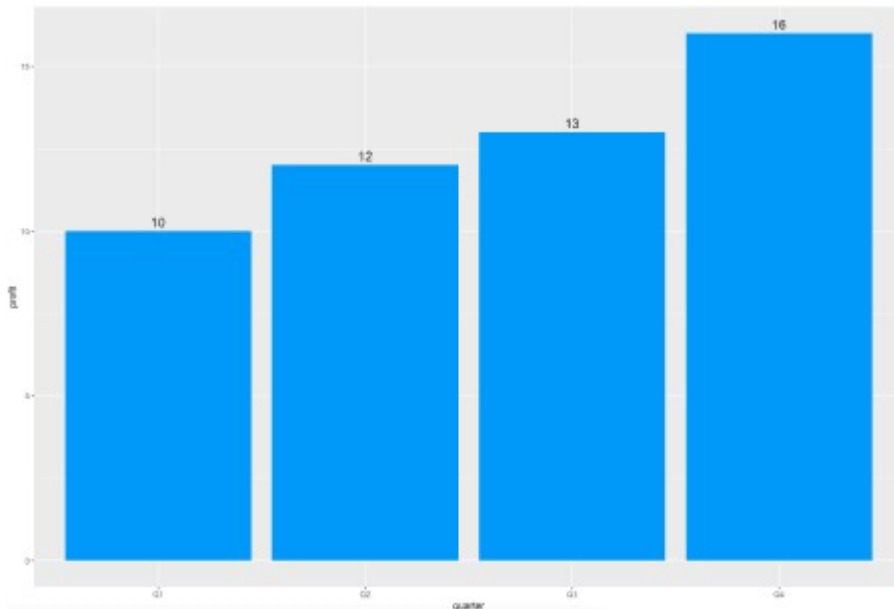


Image 17 – Labels on top of bars

But what if you want to **put the labels inside**? Just play with `vjust` a bit. Setting it to 2 does the trick:

```
ggplot(data_a, aes(x = quarter, y = profit)) +
  geom_col(fill = "#0099f9") +
  geom_text(aes(label = profit), vjust = 2, size = 5, color = "#ffffff")
```

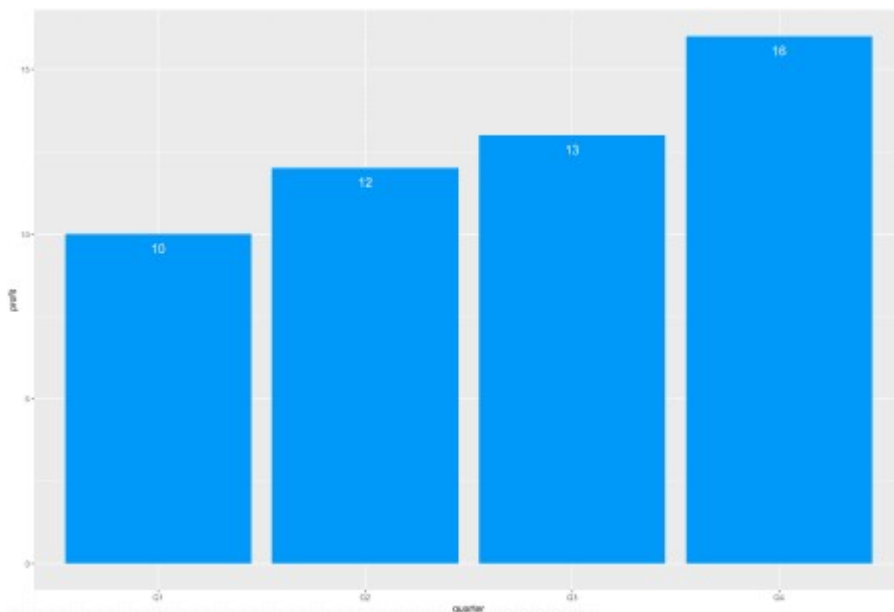


Image 18 – Labels inside the bars

Things get a bit trickier if you need labels for multiple stacks. You have to specify `position = position_stack(vjust = 0.5)`

```
ggplot(data, aes(x = quarter, y = profit, fill = product, label = profit)) +
  geom_col() +
  scale_fill_manual(values = c("#3db5ff", "#0099f9")) +
  geom_text(position = position_stack(vjust = 0.5), size = 4, color = "#fffff")
```

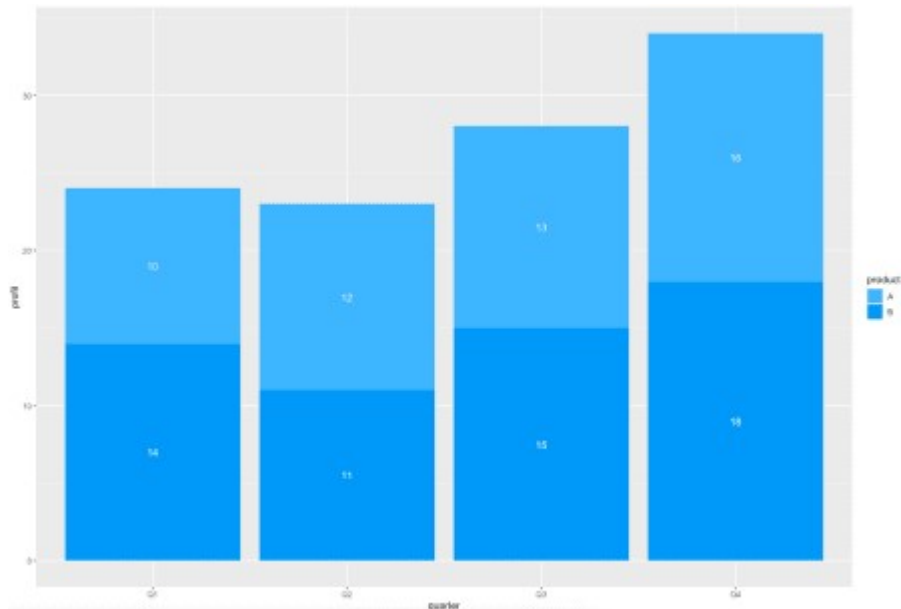


Image 19 – Labels inside the stacked bar chart

There's an alternative for a grouped bar chart. You'll have to specify `position = position_dodge()`

```
ggplot(data, aes(x = quarter, y = profit, fill = product)) +
  geom_col(position = position_dodge()) +
  scale_fill_manual(values = c("#3db5ff", "#0099f9")) +
  geom_text(aes(label = profit), position = position_dodge(0.9), vjust = 2, size = 10, color = "white")
```

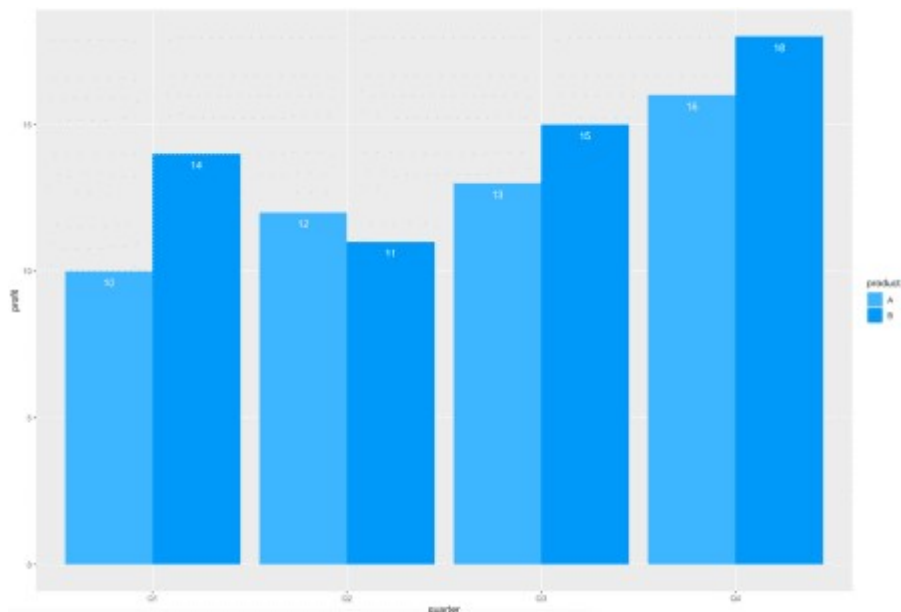


Image 20 – Labels centered inside the grouped bar chart