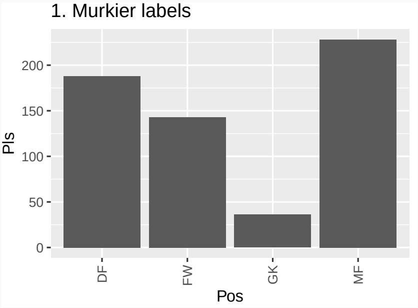
Reporting data results #1

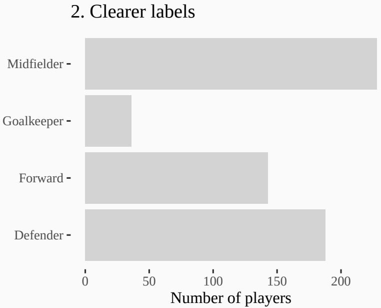
Meaningful labels and highlighting

Meaningful labels

Guideline 2: **Use clear, meaningful labels.**

Graph defaults often use abbreviations for axis labels and other labeling. Further, text labels can sometimes be aligned in a way that makes them hard to read. The plots below give an example of the same information shown without (left) and with (right) clear, meaningful labels.





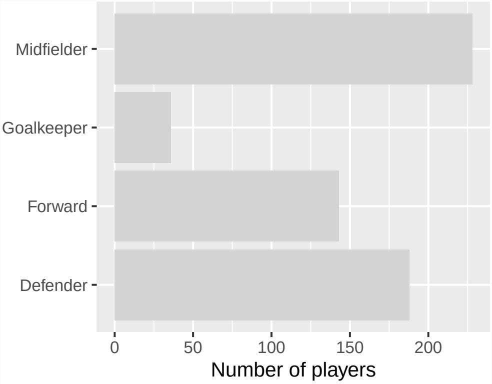
Meaningful labels

There are a few strategies you can use to make labels clearer:

- Add a `labs()` element to the plot, rather than relying on the column names in the original data. This can also be done with `scale` elements (e.g., `scale_x_continuous`), which give you more power to also make other changes to these scales.
- Include units of measurement in axis titles when relevant. If units are dollars or percent, check out the `scales` package, which allows you to add labels directly to axis elements by including arguments like `labels = percent` in `scale` elements.
- If the x-variable requires longer labels (player positions in the example above), consider flipping the coordinates, rather than abbreviating or rotating the labels. You can use `coord_flip` to do this.

Meaningful labels

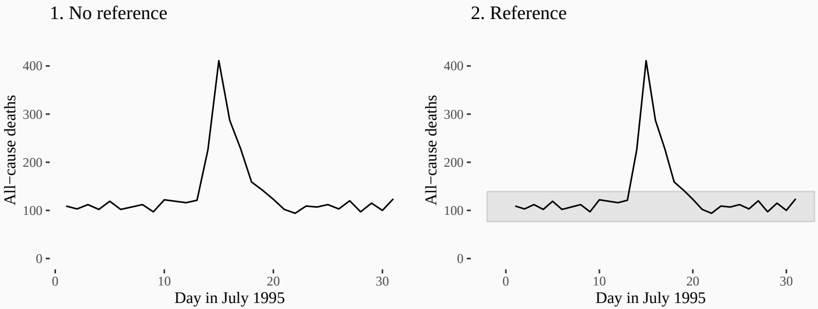
```
ggplot(worldcup, aes(Position)) +
       geom_bar(fill = "lightgray") + coord_flip() +
       labs(x = "", y = "Number of players")
```



References

Guideline 3: **Provide useful references.**

Data is easier to interpret when you add references. For example, if you show what it typical, it helps viewers interpret how unusual outliers are. The graph below on the right has added shading showing the range of daily deaths in July in Chicago for 1990–1994 and 1996–2000, to clarify how unusual July 1995 was.



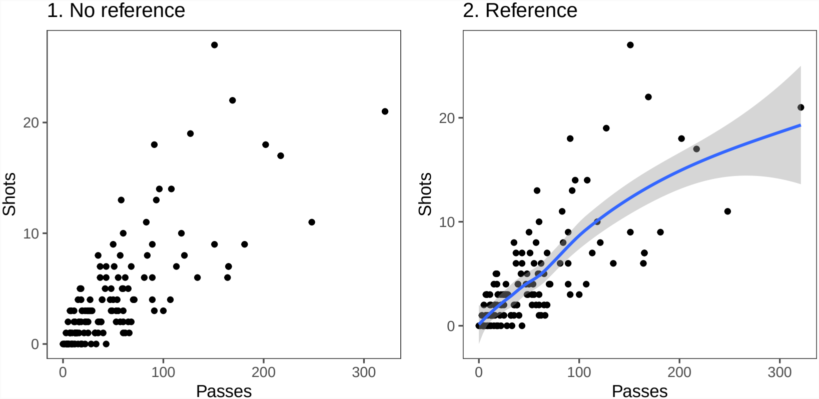
1. No reference

2. Reference

References

Guideline 3: **Provide useful references.**

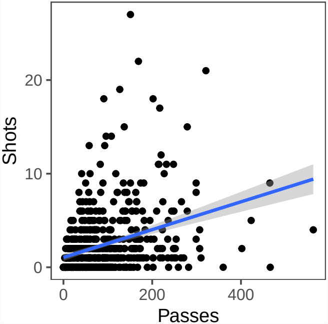
Another useful way to add references is to add a linear or smooth fit to the data, to help clarify trends in the data.



References

You can use the function geom_smooth to add a smooth or linear reference line:

```
ggplot(worldcup, aes(x = Passes, y = Shots)) +
  geom_point() + theme_few() +
  geom_smooth(method = "lm")
```



References

The most useful `geom_smooth` parameters to know are:

- `method`: The default is to add a loess curve if the data includes less than 1000 points and a generalized additive model for 1000 points or more. However, you can change to show the fitted line from a linear model using `method = "lm"` or from a generalized linear model using `method = "glm"`.
- `span`: How wiggly or smooth the smooth line should be (smaller value: more wiggly; larger value: more smooth)
- `se`: TRUE or FALSE, indicating whether to include shading for 95% confidence intervals.
- `level`: Confidence level for confidence interval (e.g., `0.90` for 90% confidence intervals)

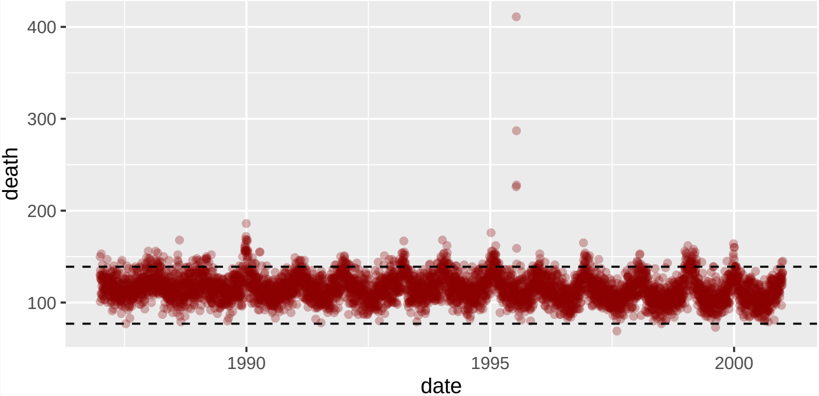
References

Lines and polygons can also be useful for adding references. Useful geoms include:

- `geom_hline`, `geom_vline`: Add a horizontal or vertical line
- `geom_abline`: Add a line with an intercept and slope
- `geom_polygon`: Add a filled polygon
- `geom_path`: Add an unfilled polygon

References

```
ggplot(data = chic, aes(x = date, y = death)) +
  geom_point(color = "darkred", alpha = 0.3) +
  geom_hline(yintercept = c(77, 139),
             linetype = 2)
```



References

When adding these references:

- If you want them to go behind the main points, add reference elements first, so they will be plotted under the data, instead of on top of it.
- Use `alpha` to add transparency to these elements.
- Use colors that are unobtrusive (e.g., grays)
- For lines, consider using non-solid line types (e.g., `linetype = 3`)