Guidelines for Effective Visualization

1) High data density
   a) Use simple themes
      i) theme_minimal(), theme_classic()...
   b) Remove backgrounds, grids, heavy bars, and shading when not necessary for interpreting the message
2) Meaningful labels
   a) Avoid abbreviations unless universally known
   b) Customize axis labels
      i) ylab(), xlab()
3) Provide useful references
   a) Reference lines to show normal values or average values
      i) 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
      ii) geom_smooth() lines
         (1) Arguments:
            (a) 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".
            (b) span: How wiggly or smooth the smooth line should be (smaller value: more flexible; larger value: more smooth)
            (c) se: TRUE or FALSE, indicating whether to include shading for 95% confidence intervals.
            (d) level: Confidence level for confidence interval (e.g., 0.90 for 90% confidence intervals)
4) Highlight interesting aspects
   a) Highlight key points that people should pay attention to with colors and labels
      i) geom_text()
5) Use small multiples
   a) Small multiples: graphs that use a lot of small plots that show subsets of data to visualize differing relationships and draw comparisons without forfeiting data
      i) facet_grid(), facet_wrap()
6) Order
   a) Make the order meaningful
      i) Ex:
         (1) worldcup %>%
         (2)   group_by(Team) %>%
         (3)   summarize(mean_time = mean(Time)) %>%
         (4)   arrange(mean_time) %>%   # re-order and re-set

(5) mutate(Team = factor(Team, levels = Team)) %>% # *factor levels before plotting*

(6) ggplot(aes(x = mean_time, y = Team)) +

(7) geom_point()

Scales and Color

1) Customize scales and colors of points on plot
    a) scale_x_continuous(), scale_size_continuous