ggplot Scatter Plot Grouped or Unique Patterns and Colors

Fan Wang

2021-11-02

Contents

1 ggplot Scatter Plot

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

1.1 Scatter Plot with Unique Shape, Color, and Label for Each

1. y-axis: horsepower
2. x-axis: time for 1/4 Miles (QSEC)
3. filter: select to display six cars as six scattered points

First, select the relevant variables and filter.

```
# Include row-name (car-names) as a variable
tb_carnames <- rownames_to_column(mtcars, var = "car_name")  %>% as_tibble()
# Select only six observations for scatter plot
set.seed(789)
it_cars_select <- 8
tb_carnames_selected <- tb_carnames[sample(dim(tb_carnames)[1], it_cars_select, replace=FALSE), ]
# Select only car name and a few variables
tb_carnames_selected <- tb_carnames_selected %>%
  select(car_name, hp, qsec) %>%
  mutate(car_name = factor(car_name))
```

Second, add styling for each point:

```
# https://www.rgbtohex.net/
# https://fanwangecon.github.io/M4Econ/graph/tools/htmlpdfm/fs_color.html
# ar_st_colours <- c(
#   "#262626", "#922428",
#   "#6b4c9a", "#535154",
#   "#3e9651", "#396ab1",
#   "#cc2529", "#ED8137")
ar_st_colours <- c(
  "#922428", "#922428",
  "#3e9651", "#3e9651",
  "#396ab1", "#396ab1",
```

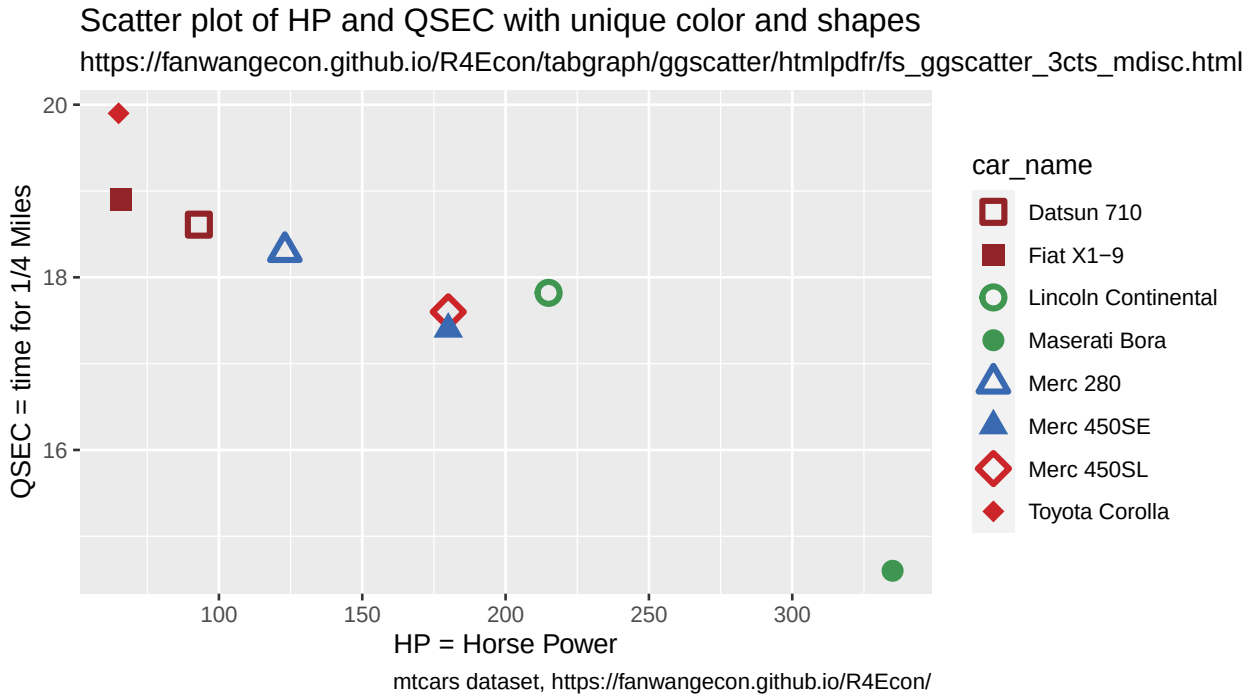
```
  "#cc2529", "#cc2529")
# http://sape.inf.usi.ch/quick-reference/ggplot2/shape
# 32 is invisible shape
# ar_it_shapes <- c(32, 5, 17, 15, 1)
ar_it_shapes <- c(
  0, 15, # square
  1, 16, # circle
  2, 17, # triangle
  5, 18 # diamond
  )
```

Third, draw a scatter plot, with defaults.

```
# Labeling
st_title <- paste0('Scatter plot of HP and QSEC with unique color and shapes')
st_subtitle <- paste0('https://fanwangecon.github.io/',
                      'R4Econ/tabgraph/ggscatter/htmlpdfr/fs_ggscatter_3cts_mdisc.html')
st_caption <- paste0('mtcars dataset, ',
                     'https://fanwangecon.github.io/R4Econ/')
st_x_label <- 'HP = Horse Power'
st_y_label <- 'QSEC = time for 1/4 Miles'
# Graphing
plt_mtcars_scatter <- tb_carnames_selected %>%
  ggplot(aes(x=hp, y=qsec,
             colour = car_name, shape = car_name,
             label=car_name)) +
  geom_point(size=3, stroke = 1.75) +
  labs(title = st_title, subtitle = st_subtitle,
       x = st_x_label, y = st_y_label, caption = st_caption)
  # geom_text(color='black', size = 3.5, check_overlap = TRUE)
# Display preliminary
# print(plt_mtcars_scatter)
```

Fourth, add in color and shape for each point based on our specifications.

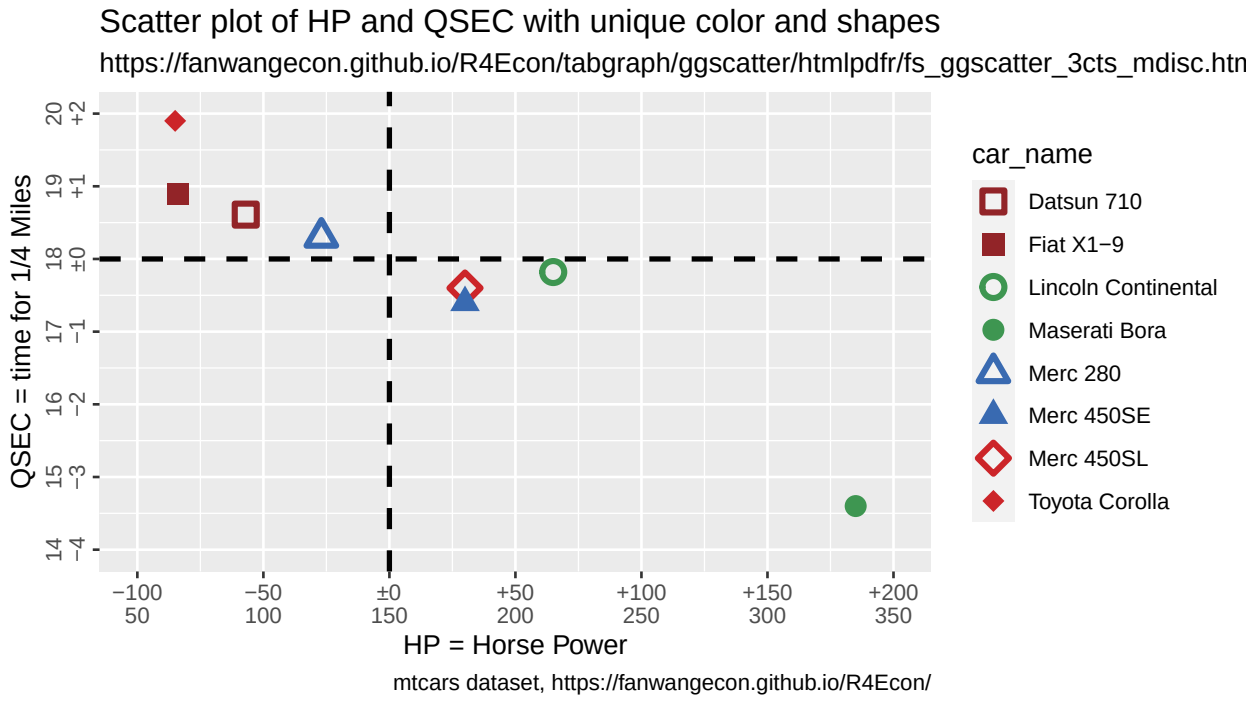
```
plt_mtcars_scatter <- plt_mtcars_scatter +
  scale_colour_manual(values = ar_st_colours) +
  scale_shape_manual(values = ar_it_shapes)
# Display preliminary
print(plt_mtcars_scatter)
```



Fifth, axis control, add-in mid-lines and additional layer of axis to show differences from added mid-lines. Add two layers of y and x labels, so that we have levels as well as deviations from the horizontal and vertical lines. Have two layers of labels so that have levels and deviations from levels.

```
# A. Y-line and X-line
fl_y_line_val <- 18
fl_x_line_val <- 150

# B. X labels
x_breaks <- c(50, 100, 150, 200, 250, 300, 350)
# x labels layer 2
x_breaks_devi <- x_breaks - fl_x_line_val
st_x_breaks_devi <- paste0(x_breaks_devi)
st_x_breaks_devi[x_breaks_devi>0] <- paste0("+", st_x_breaks_devi[x_breaks_devi>0])
st_x_breaks_devi[x_breaks_devi==0] <- paste0("±", st_x_breaks_devi[x_breaks_devi==0])
# x labels layer 1 and 2 joined
x_labels <- paste0(st_x_breaks_devi[1:length(x_breaks)], '\n', x_breaks[1:length(x_breaks)])
# x-bounds
x_min <- 50
x_max <- 350

# C. Y labels layer 1
y_breaks <- seq(14, 20, by=1)
# Y labels layer 2
y_breaks_devi <- y_breaks - fl_y_line_val
st_y_breaks_devi <- paste0(y_breaks_devi)
st_y_breaks_devi[y_breaks_devi>0] <- paste0("+", st_y_breaks_devi[y_breaks_devi>0])
st_y_breaks_devi[y_breaks_devi==0] <- paste0("±", st_y_breaks_devi[y_breaks_devi==0])
# Y labels layer 1 and 2 joined
y_labels <- paste0(y_breaks[1:length(y_breaks)], '\n', st_y_breaks_devi[1:length(y_breaks)])
# y-bounds
```

```r
y_min <- 14
y_max <- 20

# D. Add custom axis
plt_mtcars_scatter <- plt_mtcars_scatter +
  geom_hline(yintercept=fl_y_line_val, linetype="dashed", color="black", size=1) +
  geom_vline(xintercept=fl_x_line_val, linetype="dashed", color="black", size=1) +
  scale_x_continuous(
    labels = x_labels, breaks = x_breaks,
    limits = c(x_min, x_max)
  ) +
  scale_y_continuous(
    labels = y_labels, breaks = y_breaks,
    limits = c(y_min, y_max)
  )

# E. Rotate Text
plt_mtcars_scatter <- plt_mtcars_scatter +
  theme(axis.text.y = element_text(angle = 90, hjust = 0.5, vjust = 0.5))

# F. print
print(plt_mtcars_scatter)
```



1.2 Three Continuous Variables and Two Categorical Variables

We will generate a graph that is very similar to the graph shown for fs_tib_factors, with the addition that scatter color and shape will be for two separate variables, and with the addition that scatter size will be for an additional continuous variable.

We have three continuous variables:

1. y-axis: time for 1/4 Miles (QSEC)
2. x-axis: horsepower
3. scatter-size: miles per gallon (mpg)

We have two categorical ariables:

1. color: vs engine shape (vshaped or straight)
2. shape: am shift type (auto or manual)

First, Load in the mtcars dataset and convert to categorical variables to factor with labels.

```
# First make sure these are factors
tb_mtcars <- as_tibble(mtcars) %>%
  mutate(vs = as_factor(vs), am = as_factor(am))
# Second Label the Factors
am_levels <- c(auto_shift = "0", manual_shift = "1")
vs_levels <- c(vshaped_engine = "0", straight_engine = "1")
tb_mtcars <- tb_mtcars %>%
  mutate(vs = fct_recode(vs, !!!vs_levels),
         am = fct_recode(am, !!!am_levels))
```

Second, generate the core graph, a scatterplot with a nonlinear trendline. Note that in the example below color and shpae only apply to the jitter scatter, but not the trendline graph.

```
# Graphing
plt_mtcars_scatter <-
  ggplot(tb_mtcars, aes(x=hp, y=qsec)) +
  geom_jitter(aes(size=mpg, colour=vs, shape=am), width = 0.15) +
  geom_smooth(span = 0.50, se=FALSE) +
  theme_bw()
```

Third, control Color and Shape Information. There will be two colors and two shapes. See all shape listing.

```
# Color controls
ar_st_colors <- c("#33cc33", "#F8766D")
ar_st_colors_label <- c("v-shaped", "straight")
fl_legend_color_symbol_size <- 5
st_leg_color_lab <- "Engine-Shape"
# Shape controls
ar_it_shapes <- c(9, 15)
ar_st_shapes_label <- c("auto", "manuel")
fl_legend_shape_symbol_size <- 5
st_leg_shape_lab <- "Transmission"
```

Fourth, control the size of the scatter, which will be the MPG variable.

```
# Control scatter point size
fl_min_size <- 3
fl_max_size <- 6
ar_size_range <- c(fl_min_size, fl_max_size)
st_leg_size_lab <- "MPG"
```

Fifth, control graph strings.

```
# Labeling
st_title <- paste0('Distribution of HP and QSEC from mtcars')
st_subtitle <- paste0('https://fanwangecon.github.io/',
                      'R4Econ/tabgraph/ggscatter/htmlpdfr/fs_ggscatter_3cts_mdisc.html')
st_caption <- paste0('mtcars dataset, ',
```

```
                      'https://fanwangecon.github.io/R4Econ/')
st_x_label <- 'HP = Horse Power'
st_y_label <- 'QSEC = time for 1/4 Miles'
```

Sixth, combine graphical components.

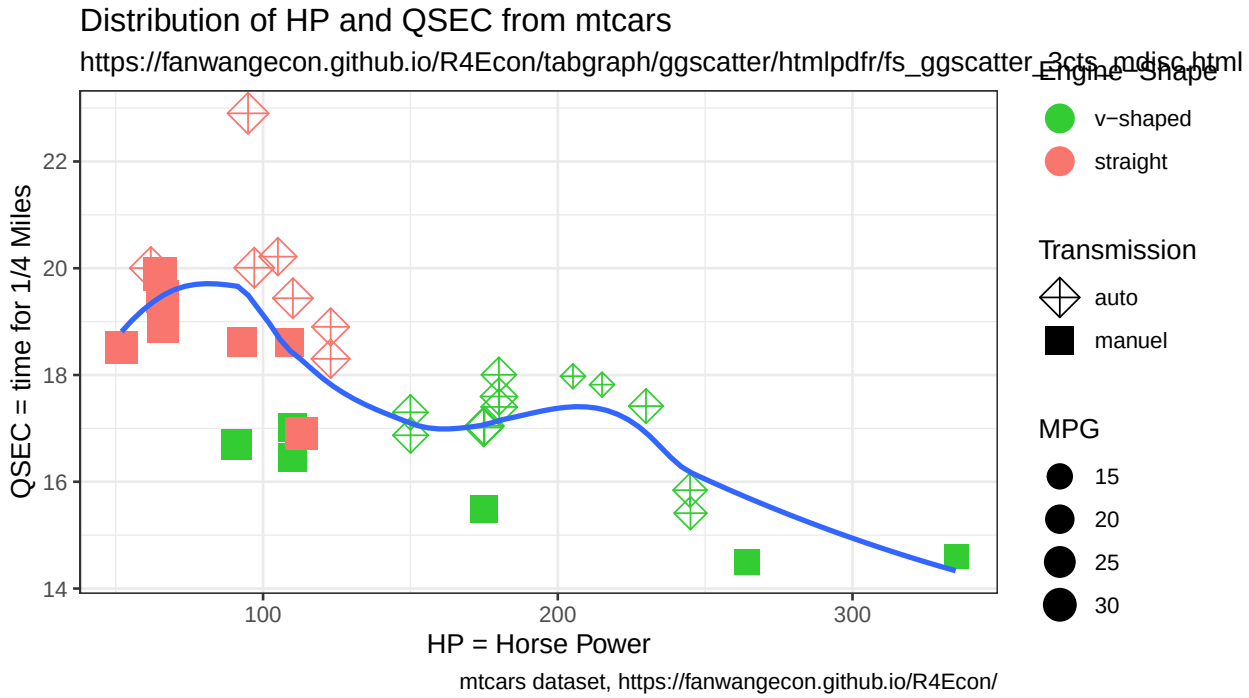
```
# Add titles and labels
plt_mtcars_scatter <- plt_mtcars_scatter +
  labs(title = st_title, subtitle = st_subtitle,
       x = st_x_label, y = st_y_label, caption = st_caption)
# Color, shape and size controls
plt_mtcars_scatter <- plt_mtcars_scatter +
  scale_colour_manual(values=ar_st_colors, labels=ar_st_colors_label) +
  scale_shape_manual(values=ar_it_shapes, labels=ar_st_shapes_label) +
  scale_size_continuous(range = ar_size_range)
```

Eigth, replace default legends titles for color, shape and size.

```
# replace the default labels for each legend segment
plt_mtcars_scatter <- plt_mtcars_scatter +
  labs(colour = st_leg_color_lab,
       shape = st_leg_shape_lab,
       size = st_leg_size_lab)
```

Ninth, additional controls for the graph.

```
# Control the order of legend display
# Show color, show shape, then show size.
plt_mtcars_scatter <- plt_mtcars_scatter + guides(
  colour = guide_legend(order = 1, override.aes = list(size = fl_legend_color_symbol_size)),
  shape = guide_legend(order = 2, override.aes = list(size = fl_legend_shape_symbol_size)),
  size = guide_legend(order = 3))
# show
print(plt_mtcars_scatter)
```


Distribution of HP and QSEC from mtcars
https://fanwangecon.github.io/R4Econ/tabgraph/ggscatter/htmlpdfr/fs_ggscatter_3cts_mdisc.html
QSEC = time for 1/4 Miles
22
20
18
16
14
100
200
300
HP = Horse Power
mtcars dataset, https://fanwangecon.github.io/R4Econ/
Engine Shape
v−shaped
straight
Transmission
auto
manuel
MPG
15
20
25
30