Sample Plots

Taylor Mackay

Contents

Introduction

This file contains examples of basic plots created using the `ggplot2` package in `R` and the corresponding code required to create each plot. All examples below require loading `ggplot2`– any other required packages are noted as needed in the included code.

NOTE: The specific style of the plots below is specified by using `theme_bcg` in addition to the other plot options. This calls the code below in order to specify the plot style, font type and size, and center plot titles.

```
# Setting options for plot formatting, including font type + size, and title
# alignment, using `minimal` theme

theme_bcg <- theme_minimal(base_size = 9, base_family = "Palatino") +
  theme(plot.title = element_text(hjust = 0.5))
```

Data Used in Examples

Most of the datasets used in the first few examples come directly from the sample datasets included with `R`. Many of the later plots, however, use player-level basketball data from the 2015-2016 season from (https://www.basketball-reference.com). This data set can be downloaded from Github using the following code.

```
download.file("github.com/mackaytc/plotting/blob/master/basketball.Rda?raw=true",
              "basketball.Rda")

load("basketball.Rda")

library(xtable)

print(xtable(head(nba.data[,1:10])), include.rownames = F)
```

Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA
1	Quincy Acy	PF	25	SAC	59	29	876	119	214
2	Jordan Adams	SG	21	MEM	2	0	15	2	6
3	Steven Adams	C	22	OKC	80	80	2014	261	426
4	Arron Afflalo	SG	30	NYK	71	57	2371	354	799
5	Alexis Ajinca	C	27	NOP	59	17	861	150	315
6	Cole Aldrich	C	27	LAC	60	5	800	134	225

Useful Resources

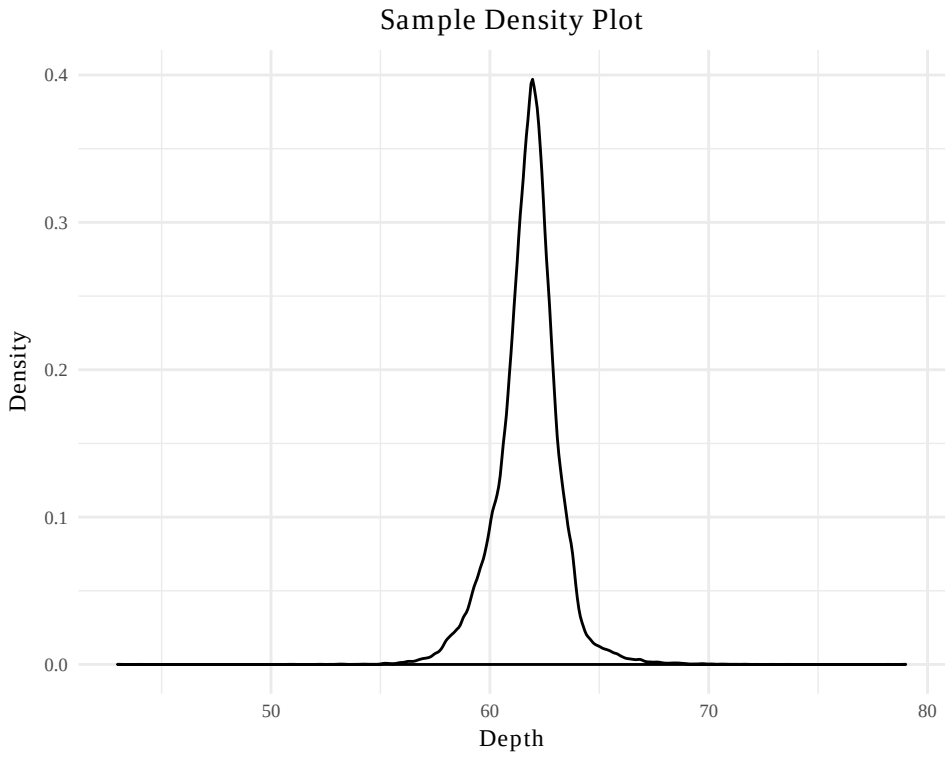
Useful websites with more information on `R` and `ggplot2` (click bulleted items for link to URL).

- RStudio `ggplot2` Cheatsheet
  - Two page PDF cheat sheet covering the basics of the `ggplot2` package
- Gallery of `ggplot2` Examples
  - 50 different examples of plots, covering a range of plot types and customizations to things like legends and annotations
- `R` Datasets Package
  - A list of the sample datasets available with `R` that are used in this document. Includes a detailed description of all variables in each dataset.

Univariate Plots

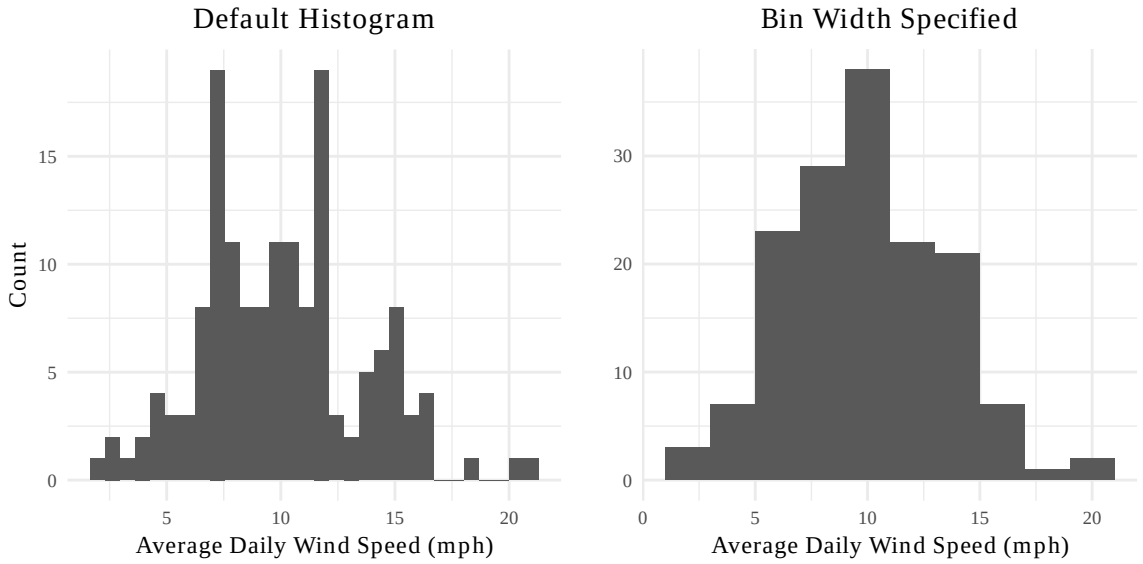
Density Plot

```
# Diamonds samples dataset has prices and other attributes of ~54,000 diamonds.

data(diamonds)

# Generating density plot

ggplot(data = diamonds, aes(x = depth)) + geom_density() +
  labs(title = "Sample Density Plot",
       y = "Density",
       x = "Depth") +
  theme_bcg
```



Histograms with Grid Arrange

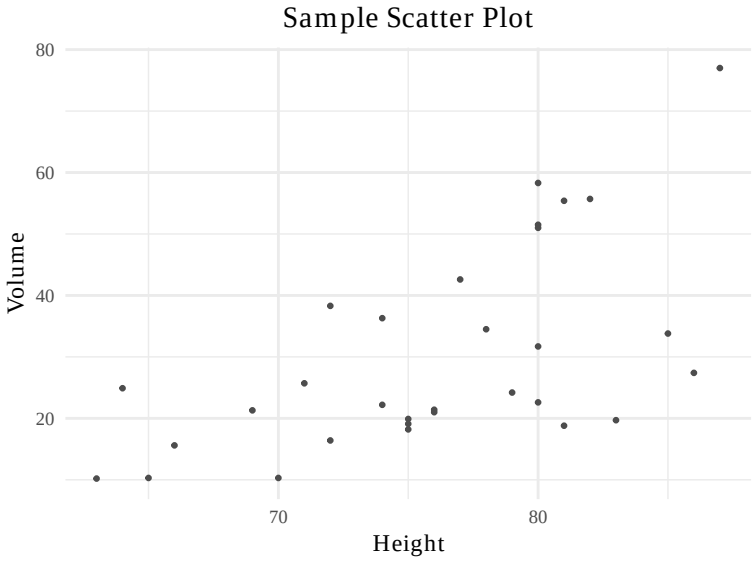
```
# `gridExtra` allows you to print multiple plots together

library(gridExtra)

# Airquality sample dataset has measurements of temperature, windspeed, and
# daily air quality in New York from May to September, 1973.

data("airquality")

# Default Histogram

p.1 <- ggplot(data = airquality) + geom_histogram(aes(x = Wind)) +
  labs(title = "Default Histogram",
       y = "Count",
       x = "Average Daily Wind Speed (mph)") +
  theme_bcg

p.2 <- ggplot(data = airquality) + geom_histogram(aes(x = Wind), binwidth = 2) +
  labs(title = "Bin Width Specified",
       y = "",
       x = "Average Daily Wind Speed (mph)") +
  theme_bcg

# Using `grid.arrange` to print both plots side by side (by setting nrow = 1)

grid.arrange(p.1, p.2, nrow = 1)
```



Two-Way Plots

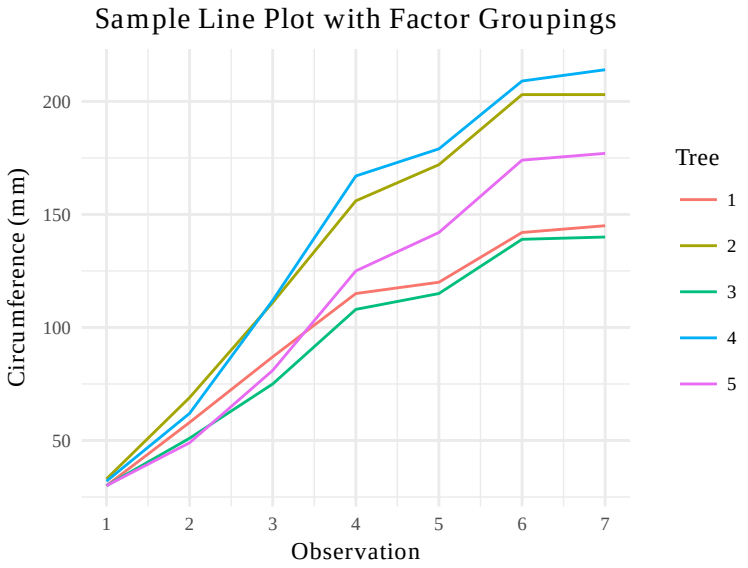
Scatter Plot

```
# Trees sample dataset has measurements of the height, weight, and length of
# 31 observations.

data(trees)

# Scatter Plot with size and color of points specified

ggplot(data = trees) + geom_point(aes(x = Height, y = Volume),
                                  size = 0.5, color = "grey30") +
  labs(title = "Sample Scatter Plot",
       y = "Volume",
       x = "Height") +
  theme_bcg
```



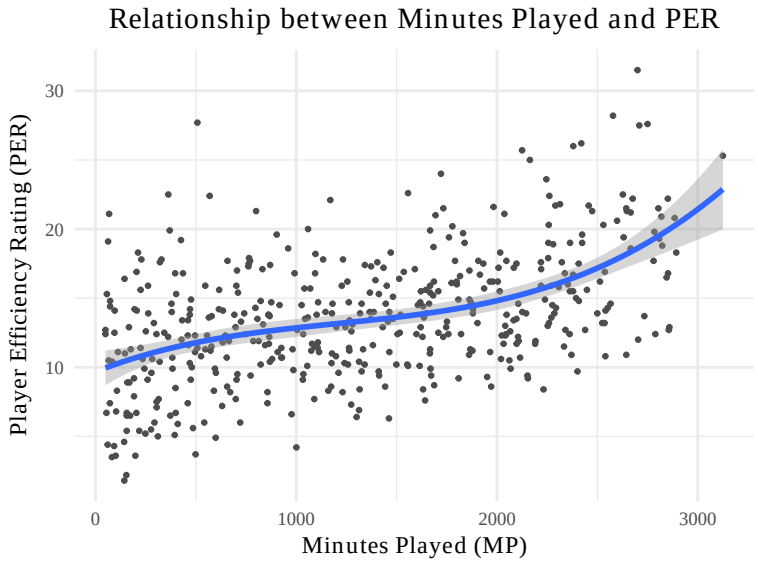
Line Plot with Outcome Grouped by Factor Variable

```
# Orange sample data set has 7 measurements of age and circumference for 5
# different oranges (total of 35 observations)

data(Orange)

# Start by creating a observation count by ID variable using `dplyr`. Note that
# data needs to be in *long* form.

library(dplyr)

df <- group_by(Orange, Tree) %>%
  mutate(count = row_number())

# Creating re-ordered `tree` factor variable

df$Tree <- factor(df$Tree, levels = c(1,2,3,4,5))

# Line Plot-- notice options for setting x-axis ticks + legend label

ggplot(data = df) + geom_line(aes(x = count, y = circumference,
                                  color = Tree)) +
  labs(title = "Sample Line Plot with Factor Groupings",
       y = "Circumference (mm)", x = "Observation",
       color = "Tree") +
  scale_x_continuous(breaks=seq(1, 7, 1)) +
  theme_bcg
```



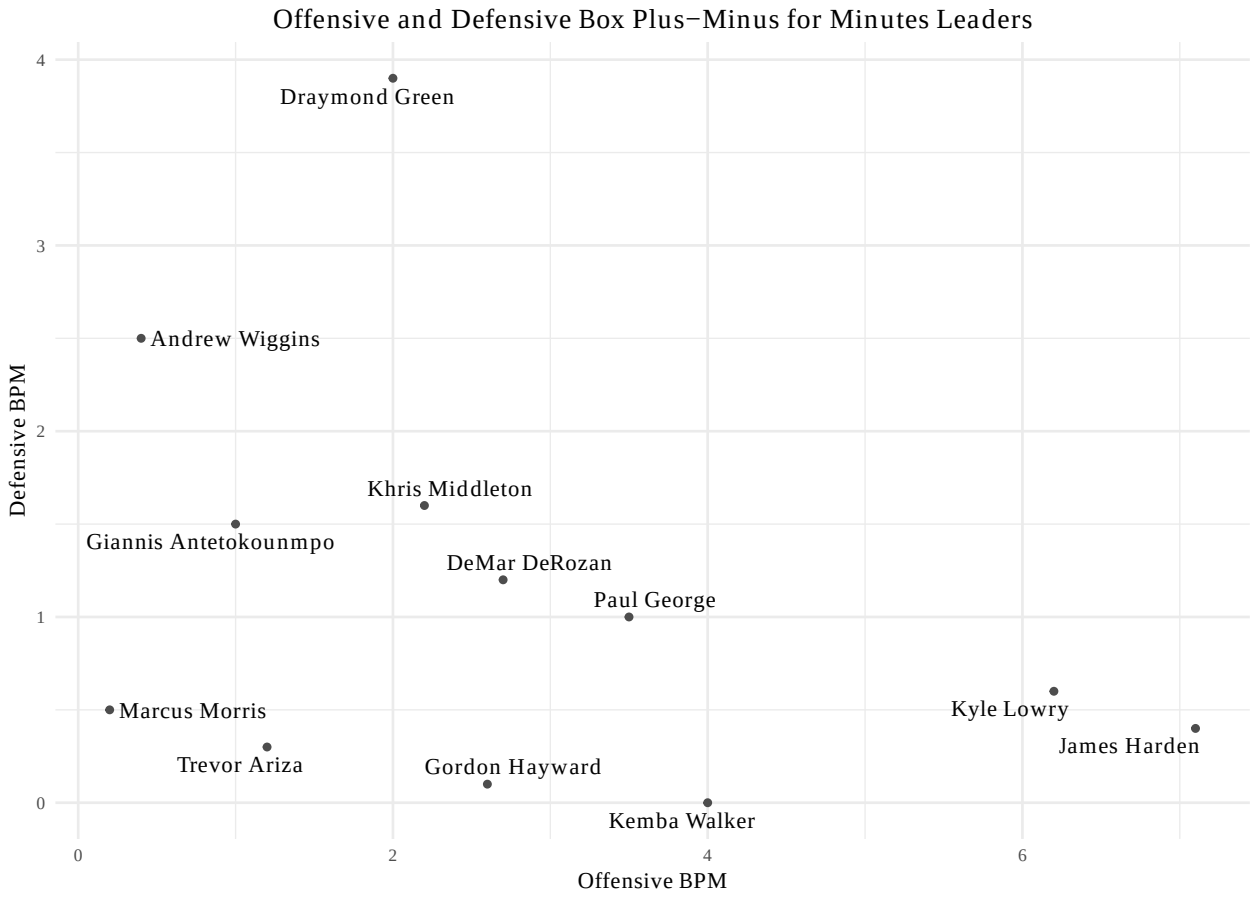
Scatter Plot with Fitted Line

```
# Load sample dataset with 2016 player statistics for all players in NBA

load("basketball.Rda")

# Scatter Plot with Minutes Played and Player Efficiency Rating (PER)

# Data is filtered to only players with at least one full game (48 minutes)
# worth of playing time during the season. `geom_smooth` options set to display
# a 3rd-degree polynomial fitted line with SE bands displayed

library(dplyr)

ggplot(data = filter(nba.data, MP > 48), aes(x = MP, y = PER)) +
  geom_point(size = 0.5, color = "grey30") +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2) + I(x^3), se = TRUE) +
  labs(title = "Relationship between Minutes Played and PER",
       y = "Player Efficiency Rating (PER)",
       x = "Minutes Played (MP)") +
  theme_bcg
```



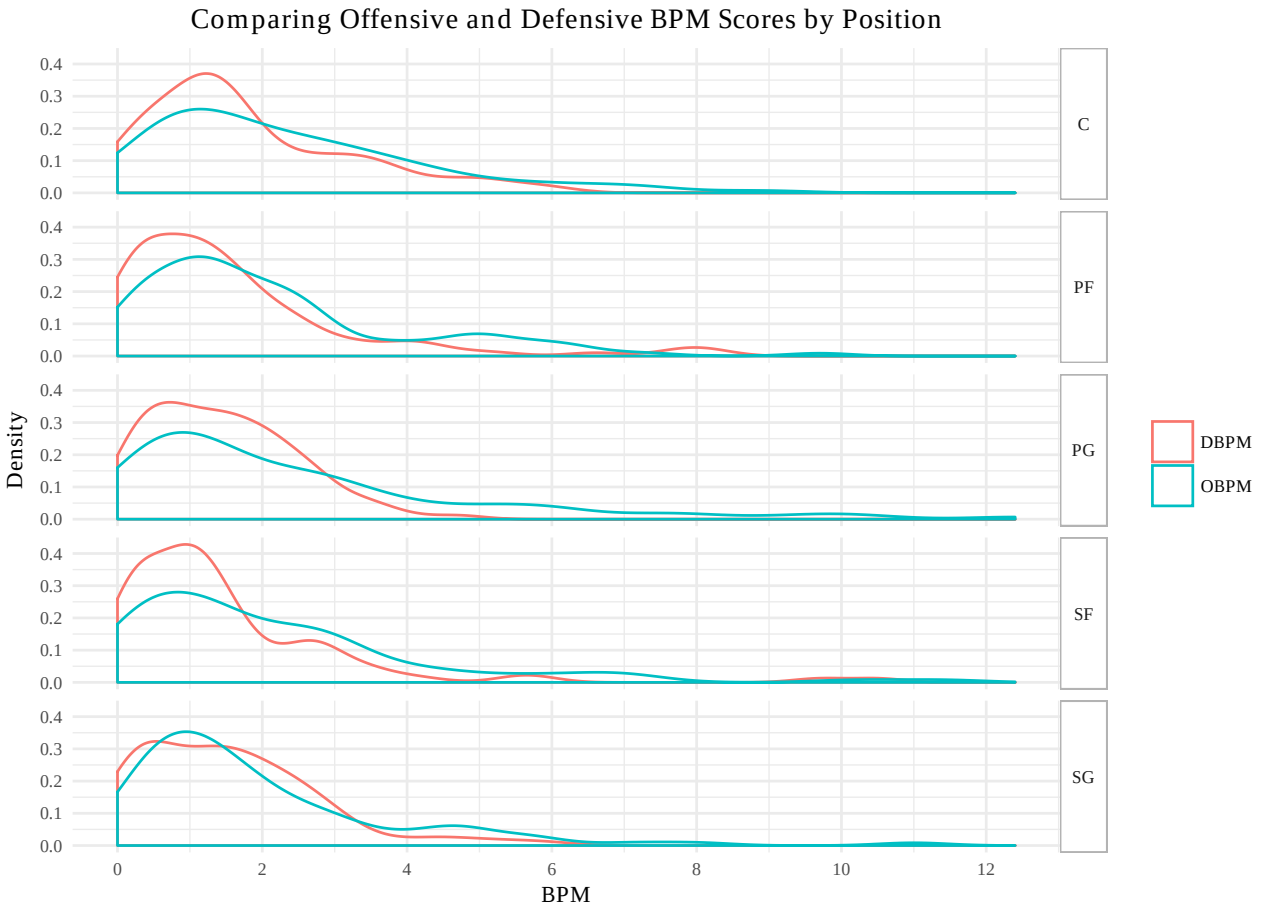
Scatter Plot with (Neatly) Labeled Points

```
# Load sample dataset with 2016 player statistics for all players in NBA

load("basketball.Rda")

# Plot Relationship between Offensive and Defensive Box Plus-Minus (BPM)

# We can use the `ggrepel` package for neatly formatted point labelling

library(ggrepel)

# Use `dplyr` to select + sort the top 10 players by total minutes played

arrange(filter(nba.data, MP > 2000),-MP)[1:12,] %>%
  ggplot(aes(x = OBPM, y = DBPM)) +
  geom_point(size = 1, color = "grey30") +
  geom_text_repel(aes(label = Player), family = "Palatino", size = 3,
                  segment.color = NA) +
  labs(title = "Offensive and Defensive Box Plus-Minus for Minutes Leaders",
         y = "Defensive BPM", x = "Offensive BPM") + theme_bcg
```



Line Plots with Facets to Create Subplots

```
# Load sample dataset with 2016 player statistics for all players in NBA

load("basketball.Rda")

# We'll `tidyr` to reshape the data from `wide` to `long` format using the
# `gather` command and create a new dataset where each player in the data set
# has two rows-- one corresponding to their defensive BPM and one corresponding
# to their offensive BPM.

library(dplyr)
library(tidyr)

facet.data <- select(nba.data, Player, Pos, OBPM, DBPM) %>%
  gather(key = c(Player, Pos), value = BPM, OBPM:DBPM) %>%
  rename(stat = `c(Player, Pos)`) %>%
  arrange(Player)

# Facet Plot-- note the formatting options at the bottom to specify facet
# formatting

ggplot(facet.data) + facet_grid(Pos ~ .) +
  geom_density(aes(x = BPM, color = stat)) +
  scale_x_continuous(breaks = seq(0, 12, 2)) +
  labs(title = "Comparing Offensive and Defensive BPM Scores by Position",
       x = "BPM", y = "Density", color = "") +
  theme_bcg + theme(strip.text.y = element_text(angle = 0),
                    strip.background = element_rect(color = "grey70",
                                                    size = 0.5))
```


Comparing Offensive and Defensive BPM Scores by Position
Density
BPM
C
PF
PG
SF
SG
DBPM
OBPM
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10
12