

Base R Graphics Cheat Sheet

David Gerard

2019-01-22

Abstract:

I reproduce some of the plots from Rstudio's [ggplot2](#) cheat sheet using Base R graphics. I didn't try to pretty up these plots, but **you should**.

I use this dataset

```
data(mpg, package = "ggplot2")
```

General Considerations

The main functions that I generally use for plotting are

- Plotting Functions
 - `plot`: Makes scatterplots, line plots, among other plots.
 - `lines`: Adds lines to an already-made plot.
 - `par`: Change plotting options.
 - `hist`: Makes a histogram.
 - `boxplot`: Makes a boxplot.
 - `text`: Adds text to an already-made plot.
 - `legend`: Adds a legend to an already-made plot.
 - `mosaicplot`: Makes a mosaic plot.
 - `barplot`: Makes a bar plot.
 - `jitter`: Adds a small value to data (so points don't overlap on a plot).
 - `rug`: Adds a rugplot to an already-made plot.
 - `polygon`: Adds a shape to an already-made plot.
 - `points`: Adds a scatterplot to an already-made plot.
 - `mtext`: Adds text on the edges of an already-made plot.
- Sometimes needed to transform data (or make new data) to make appropriate plots:
 - `table`: Builds frequency and two-way tables.
 - `density`: Calculates the density.
 - `loess`: Calculates a smooth line.
 - `predict`: Predicts new values based on a model.

All of the plotting functions have arguments that control the way the plot looks. You should read about these arguments. In particular, read carefully the help page `?plot.default`. Useful ones are:

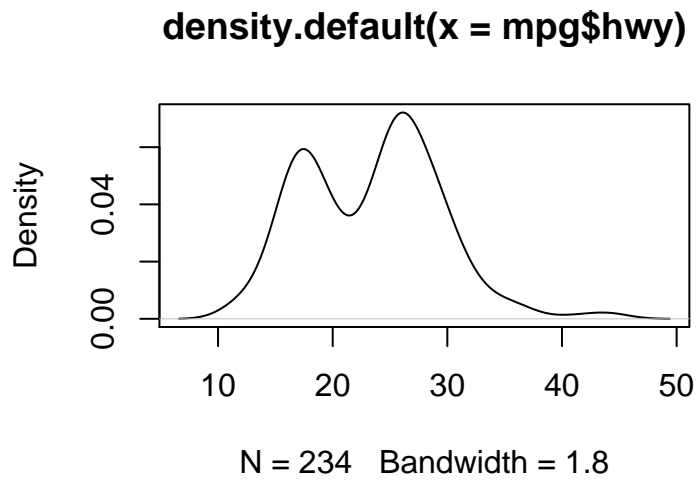
- `main`: This controls the title.
- `xlab`, `ylab`: These control the x and y axis labels.
- `col`: This will control the color of the lines/points/areas.
- `cex`: This will control the size of points.
- `pch`: The type of point (circle, dot, triangle, etc...)
- `lwd`: Line width.
- `lty`: Line type (solid, dashed, dotted, etc...).

One Variable

Continuous

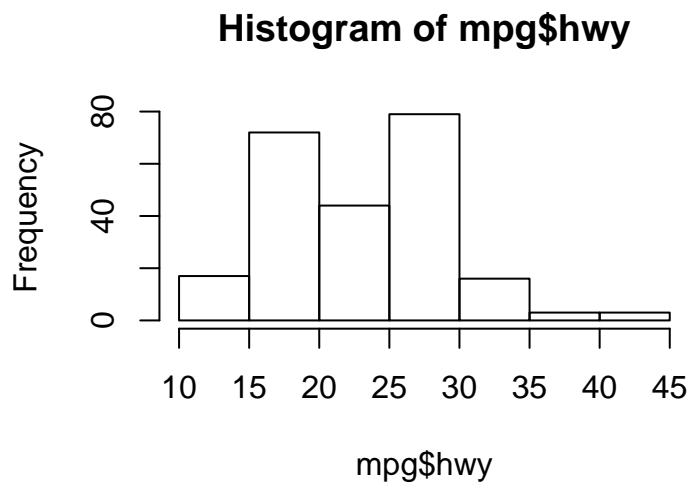
Density plot

```
density_object <- density(mpg$hwy)  
plot(density_object)
```



Histogram

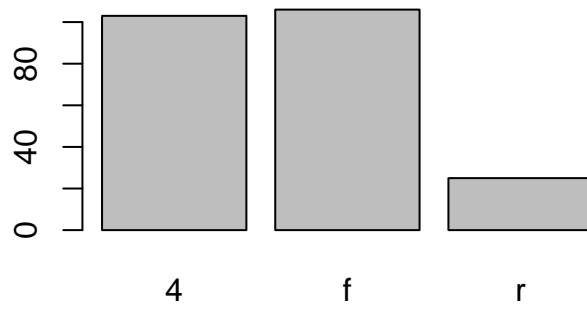
```
hist(mpg$hwy, breaks = 10)
```



Discrete

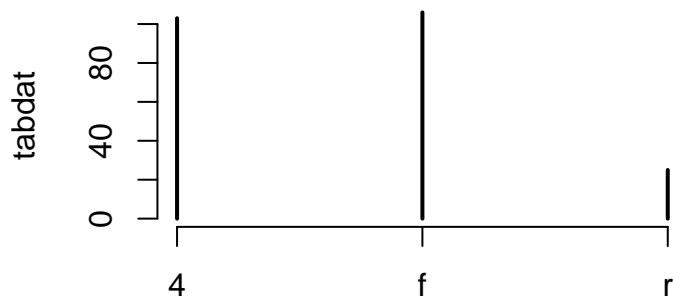
Barplot

```
tabdat <- table(mpg$drv)  
barplot(tabdat)
```



Different type of bar plot

```
plot(tabdat)
```

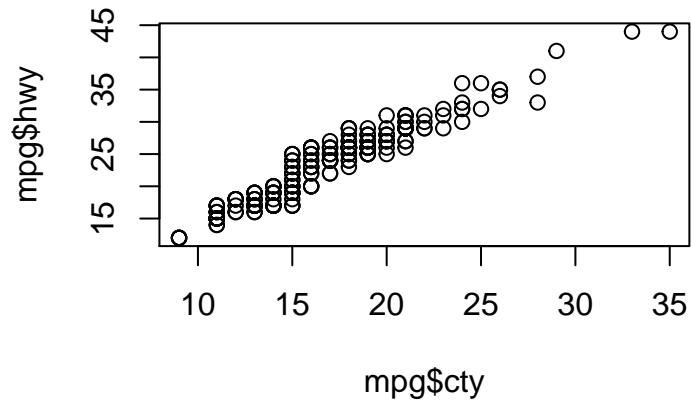


Two Variables

Continuous X, Continuous Y

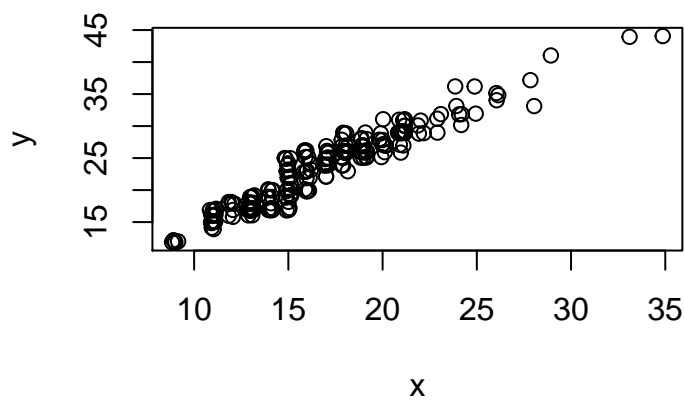
Scatterplot

```
plot(mpg$cty, mpg$hwy)
```



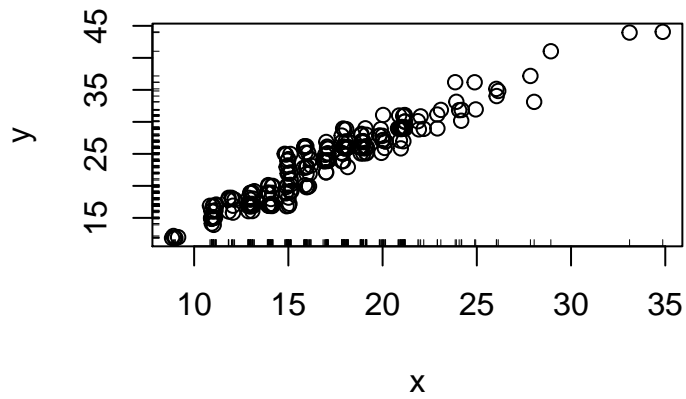
Jitter points to account for overlaying points.

```
x <- jitter(mpg$cty)  
y <- jitter(mpg$hwy)  
plot(x, y)
```



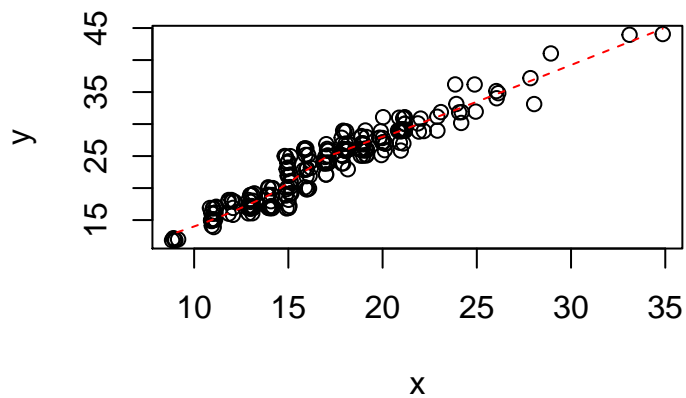
Add a rug plot

```
plot(x, y)
rug(x, side = 1)
rug(y, side = 2)
```



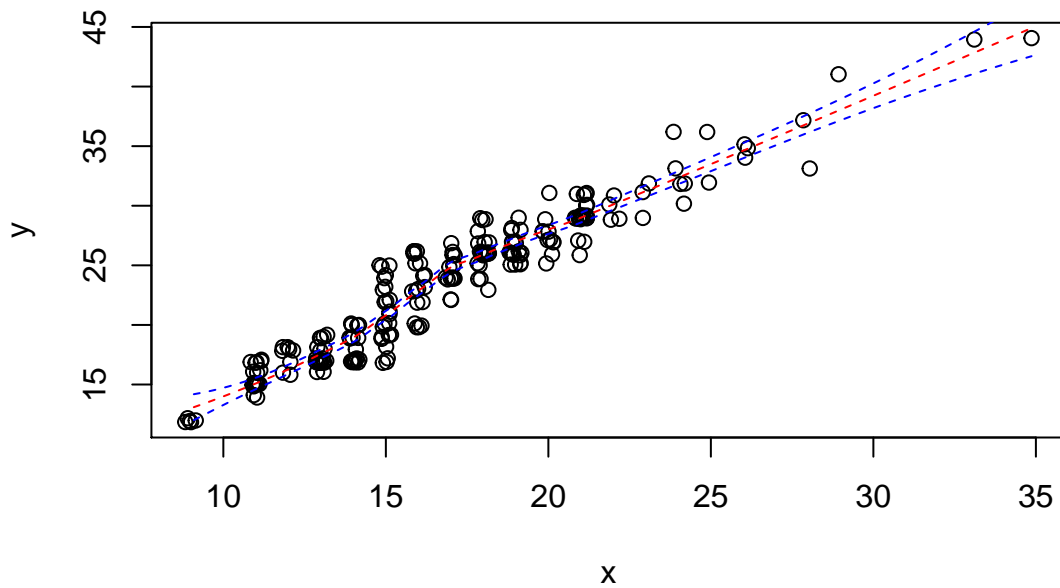
Add a Loess Smoother

```
loess_fit <- loess(hwy ~ cty, data = mpg)
xnew <- seq(min(x), max(x), length = 100)
ynew <- predict(object = loess_fit, newdata = data.frame(cty = xnew))
plot(x, y)
lines(xnew, ynew, col = 2, lty = 2)
```



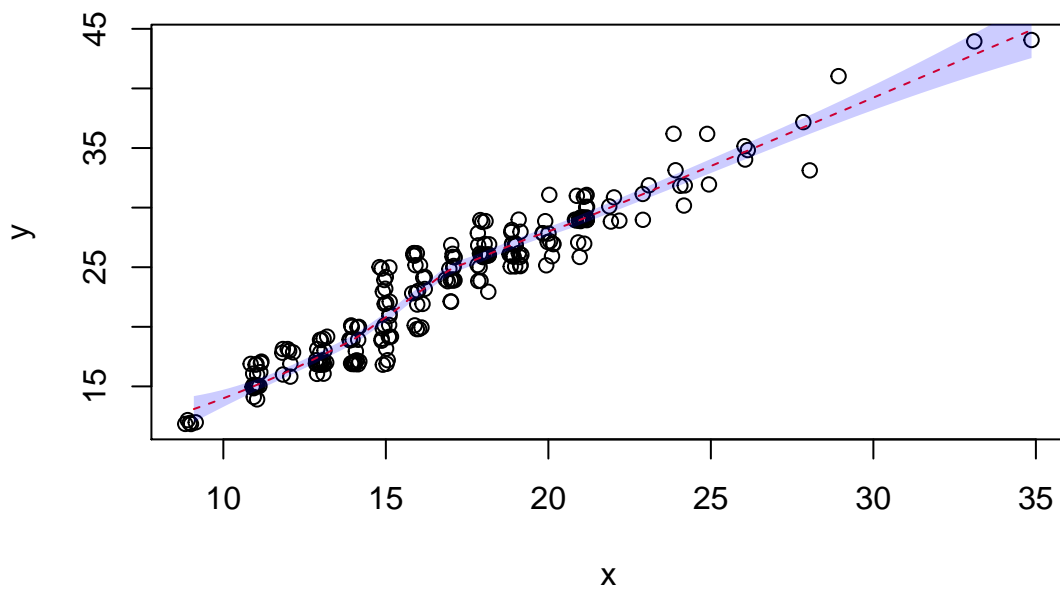
Loess smoother with upper and lower 95% confidence bands

```
loess_fit <- loess(hwy ~ cty, data = mpg)
xnew <- seq(min(x), max(x), length = 100)
pfit <- predict(object = loess_fit, newdata = data.frame(cty = xnew), se = TRUE)
ynew <- pfit$fit
upper_bound <- pfit$fit + qnorm(0.975) * pfit$se.fit
lower_bound <- pfit$fit - qnorm(0.975) * pfit$se.fit
plot(x, y)
lines(xnew, ynew, col = 2, lty = 2)
lines(xnew, upper_bound, col = 4, lty = 2)
lines(xnew, lower_bound, col = 4, lty = 2)
```



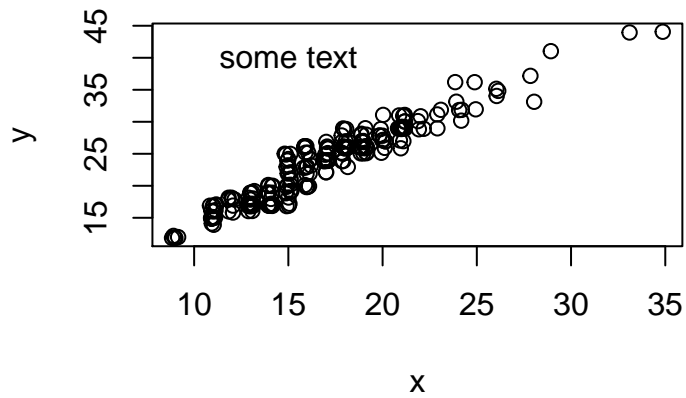
Loess smoother with upper and lower 95% confidence bands and that fancy shading from ggplot2.

```
loess_fit <- loess(hwy ~ cty, data = mpg)
xnew <- seq(min(x), max(x), length = 100)
pfit <- predict(object = loess_fit, newdata = data.frame(cty = xnew), se = TRUE)
ynew <- pfit$fit
upper_bound <- pfit$fit + qnorm(0.975) * pfit$se.fit
lower_bound <- pfit$fit - qnorm(0.975) * pfit$se.fit
xshade <- c(xnew, xnew[length(xnew):1])
yshade <- c(upper_bound, lower_bound[length(lower_bound):1])
plot(x, y)
lines(xnew, ynew, col = 2, lty = 2)
polygon(xshade, yshade, col = "#0000FF33", border = FALSE)
```



Add text to a plot

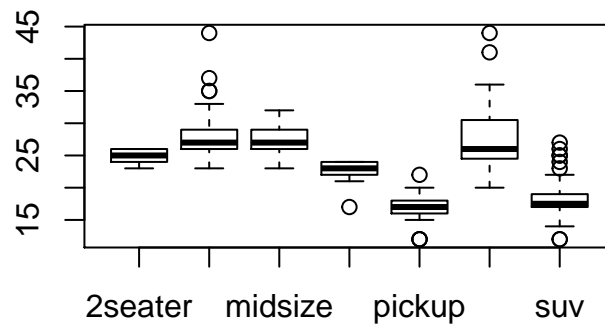
```
plot(x, y)  
text(15, 40, "some text")
```



Discrete X, Continuous Y

Boxplot

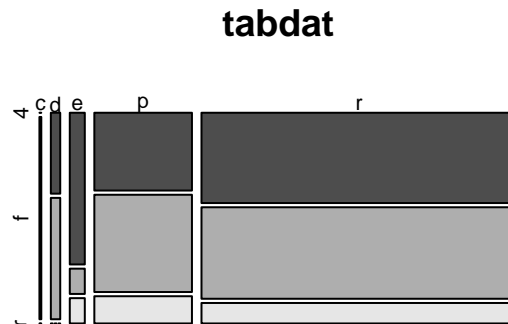
```
boxplot(hwy ~ class, data = mpg)
```



Discrete X, Discrete Y

Mosaic Plot

```
tabdat <- table(mpg$f1, mpg$drv)
mosaicplot(tabdat, color = TRUE)
```

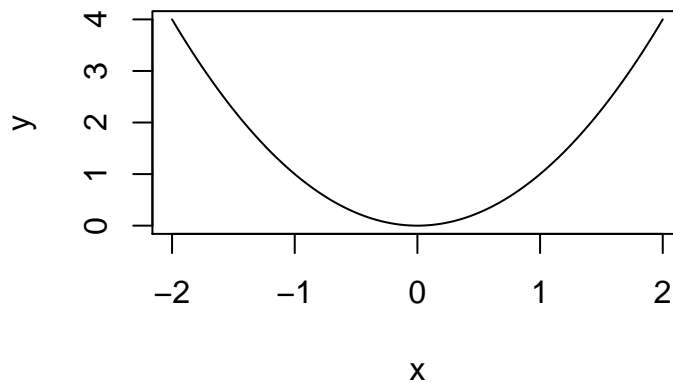


Continuous Function

Line plot

```
f <- function(x) {
  return(x ^ 2)
}

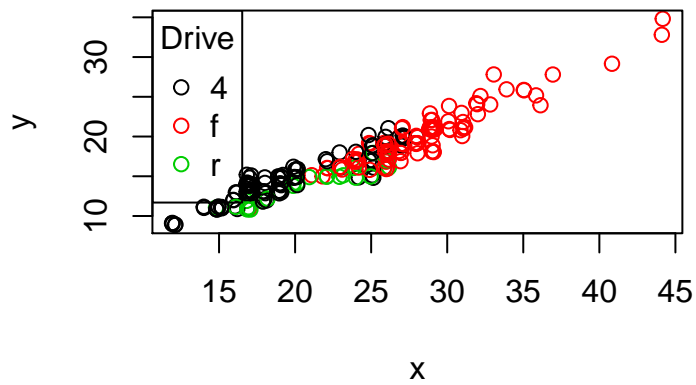
x <- seq(-2, 2, length = 100)
y <- f(x)
plot(x, y, type = "l")
```



Color Coding and Legends

Color code a scatterplot by a categorical variable and add a legend.

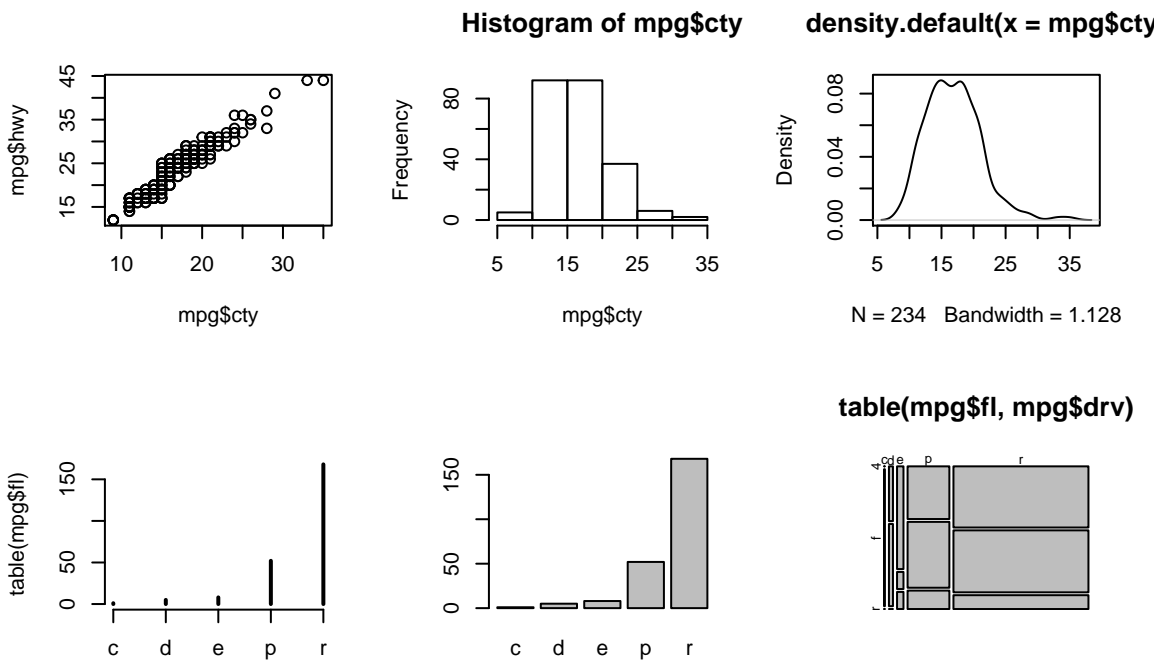
```
x <- jitter(mpg$hwy)
y <- jitter(mpg$cty)
z <- factor(mpg$drv)
plot(x, y, col = z)
legend("topleft", legend = levels(z), col = 1:nlevels(z), pch = 1, title = "Drive")
```



Faceting

`par` sets the graphics options, where `mfrow` is the parameter controlling the facets.

```
old_options <- par(mfrow = c(2, 3))
plot(mpg$cty, mpg$hwy)
hist(mpg$cty)
plot(density(mpg$cty))
plot(table(mpg$f1))
barplot(table(mpg$f1))
plot(table(mpg$f1, mpg$drv))
```



```
par(old_options)
```

The first line sets the new options and saves the old options in the list `old_options`. The last line reinstates the old options.