

# Post02: Ggplot & Grammar of Graphics and advanced package of Plotly

Haibin Lim

11/30/2017

## Introduction

In ggplot package, we learned how to visualize the data. According to the analysis and the purpose of collecting data we can graph it as a ggplot, histogram, boxplot or a line plot. Grammar of Graphics is an implementation in ggplot2 that allows more detailed and varied versions of visualization.

Moreover, in terms of visual graphing, a package in R expedites further data set. It's called plotly.

## Motivation

In reality, ggplot provides everything that you need in terms of visually analyzing data. However, grammar of graphics gives visual cues that facilitates the understanding of the point that you are trying to emphasize. In terms of presenting the data, ggplot and grammar of graphics is the most clear way to deliver the analysis to the audience.

## Code

**First Start By Downloading the Packages** # The data set is of measurement of camping tents that analyzed 90 variables separated by brand. The

```
library(readr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
dat <- read_csv('/Users/haibinlim/stat133/stat133-hws-fall17/post02/data/camping-tents.csv')
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   brand = col_character(),
##   price = col_double(),
##   weight = col_double(),
##   height = col_integer(),
##   bestuse = col_character(),
##   seasons = col_character(),
##   capacity = col_character()
## )
```

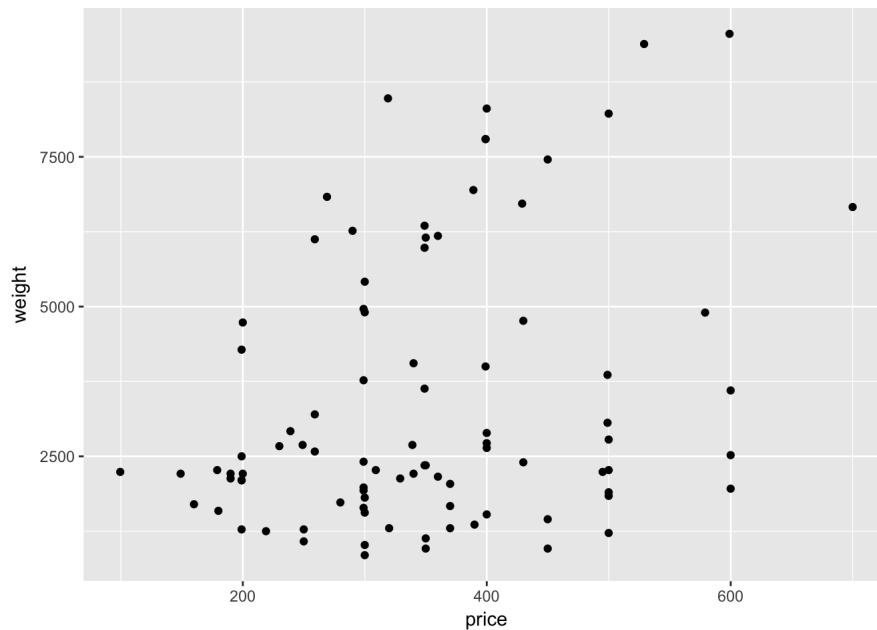
```
head(dat)
```

```
## # A tibble: 6 x 8
##       name      brand price weight height bestuse seasons
##       <chr>    <chr> <dbl> <dbl> <int>   <chr>   <chr>
## 1 fly-creek-ul2 big-agnes 349.95   960    96 Backpacking 3-season
## 2 fly-creek-ul3 big-agnes 449.95  1450   107 Backpacking 3-season
## 3 salida-2      kelty 159.95  1700   102 Backpacking 3-season
## 4 jack-rabbit-sl3 big-agnes 359.95  2160   107 Backpacking 3-season
## 5 passage-2      rei 149.00  2210   107 Backpacking 3-season
## 6 copper-spur-ul2 big-agnes 399.95  1530   107 Backpacking 3-season
## # ... with 1 more variables: capacity <chr>
```

first we can plot the numerical part of the data. The three

numerical categories are price, weight, and height and we can choose any two of the three numerical data set.

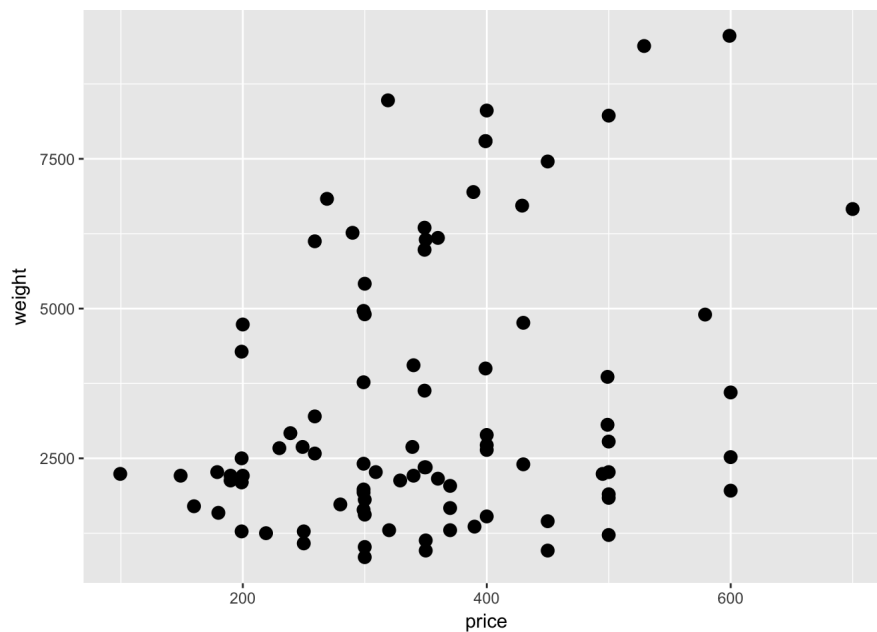
```
ggplot(data = dat, aes(x = price, y = weight )) + geom_point()
```



# You can change the size of the

geom\_point

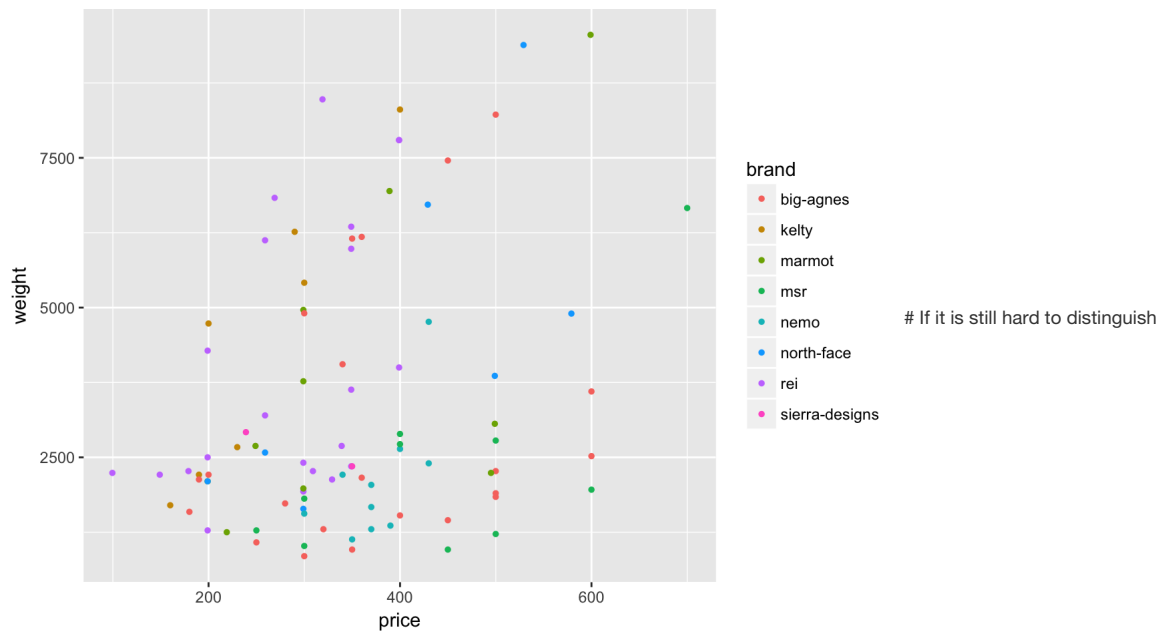
```
ggplot(data = dat, aes(x = price, y = weight )) + geom_point(size = 3)
```



# You can also add colors according

to each subset of the data; the points could be color coordinated by brand in our example.

```
ggplot(dat, aes(price, weight, color = brand)) + geom_point(size = 1)
```

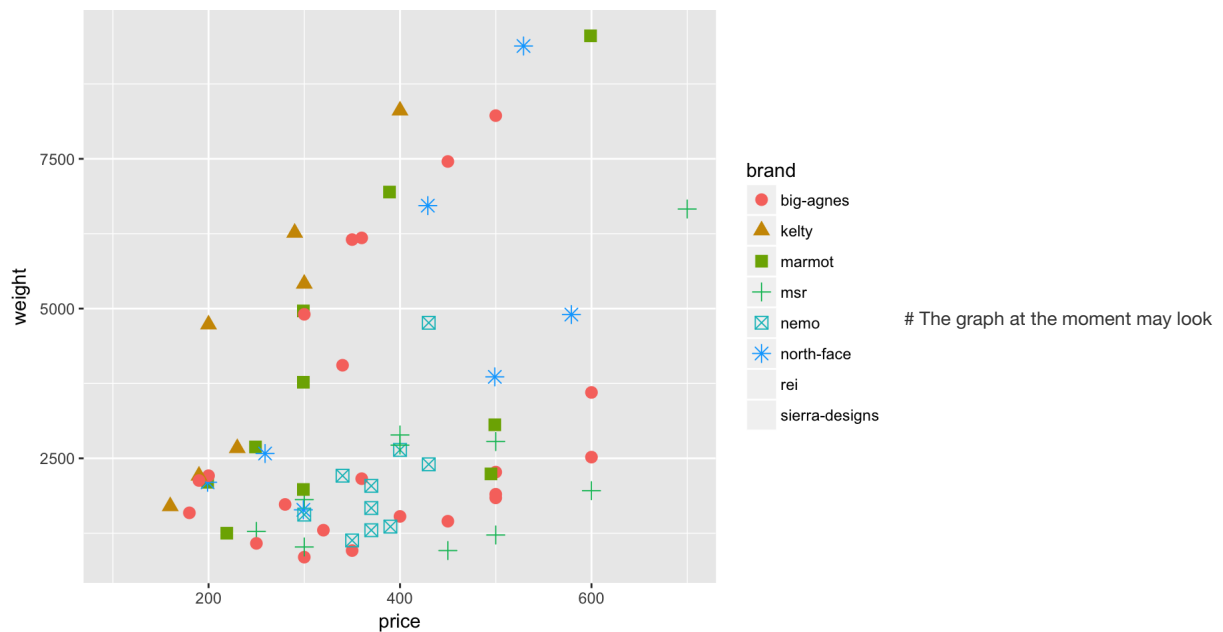


between the points you add on to the color with different shapes.

```
ggplot(dat, aes(price, weight, color = brand)) + geom_point(aes(shape = brand), size = 3)
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 8.
## Consider specifying shapes manually if you must have them.
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```



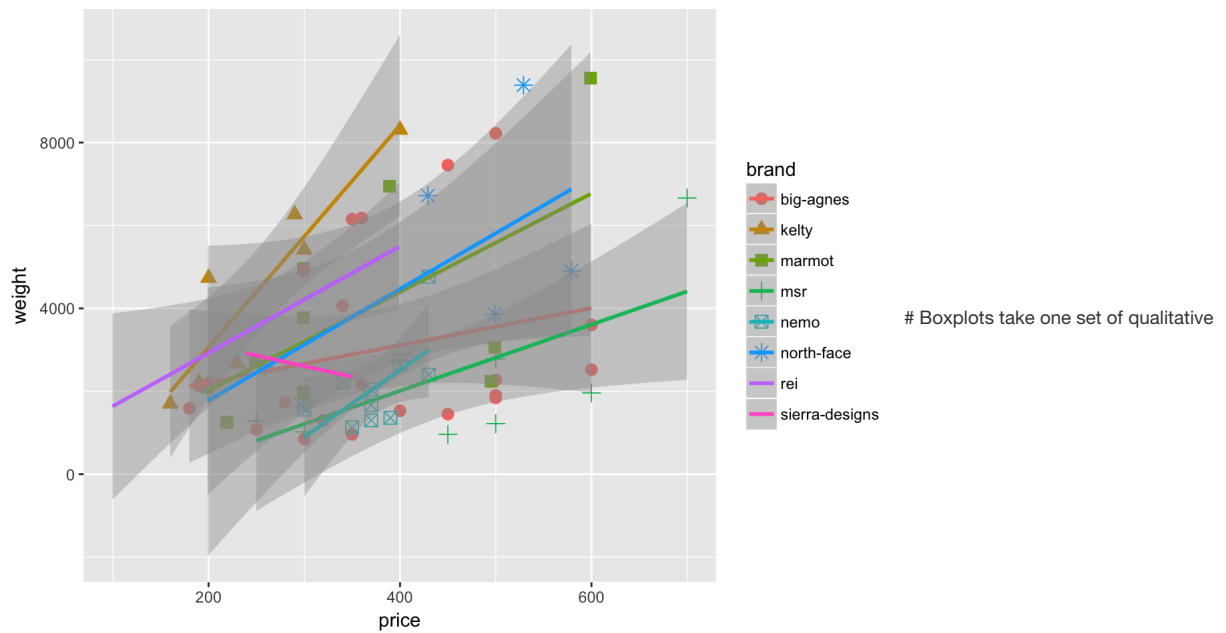
a bit messy. In order to understand the graph according to the variable, create the lm line.

```
ggplot(dat, aes(price, weight, color = brand)) + geom_point(aes(shape = brand), size = 3) +
  geom_smooth(method = "lm")
```

```
## Warning in qt((1 - level)/2, df): NaNs produced
```

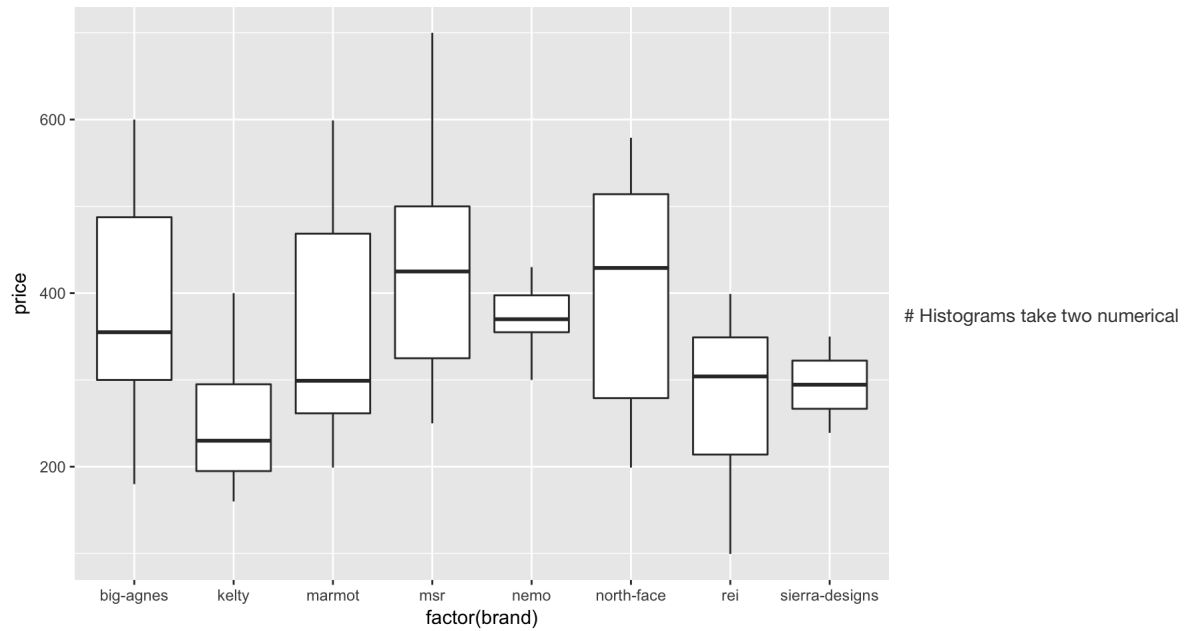
```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 8.
## Consider specifying shapes manually if you must have them.
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```



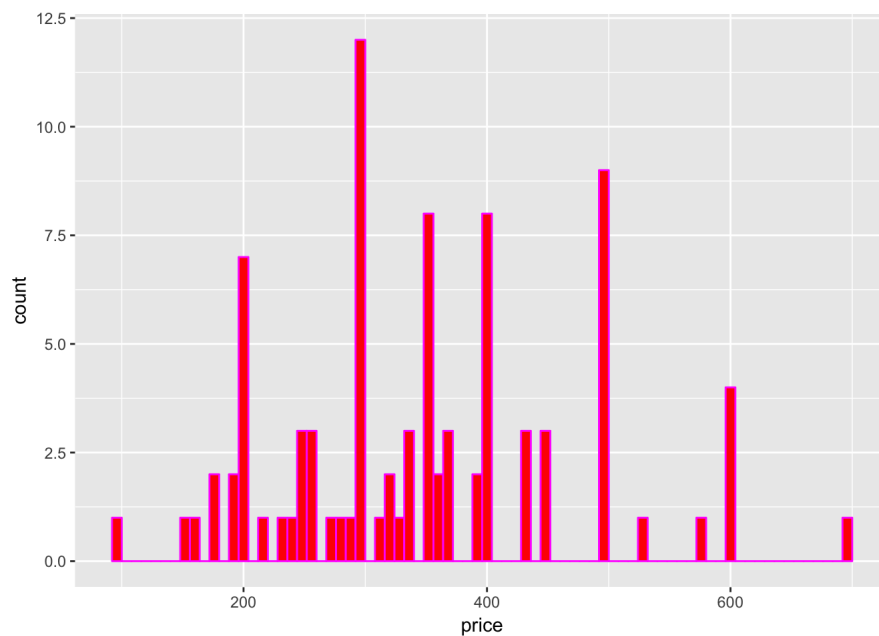
data with numerical data and plots.

```
ggplot(dat, aes(factor(brand), price)) + geom_boxplot()
```



data set and graph the price with the count up to 12.5.

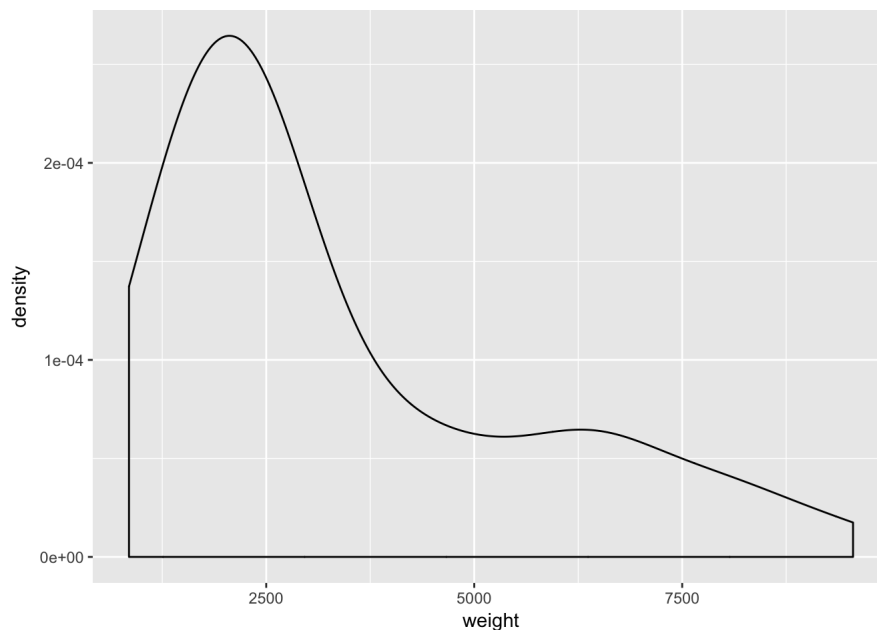
```
hist <- ggplot(dat, aes(x = price))
hist + geom_histogram(binwidth = 8, fill = "red", colour = "magenta")
```



# Line plots take the density and plot

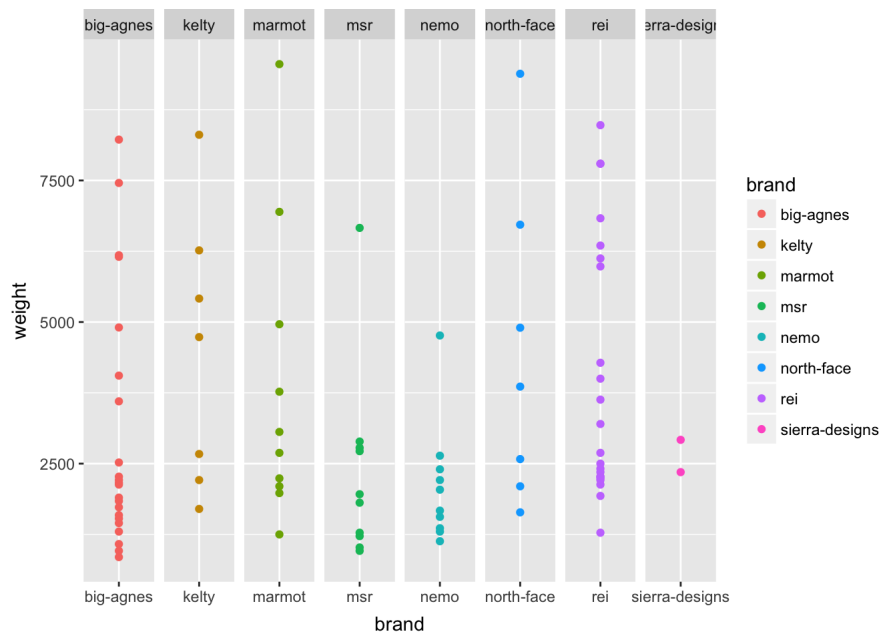
the lowest line of the data (in this case weight).

```
ggplot(dat, aes(weight)) + geom_density()
```



Mapping Variables to colors. Facet\_grid function is something new. It creates a matrix of panels (works with two discrete variables). In this example,

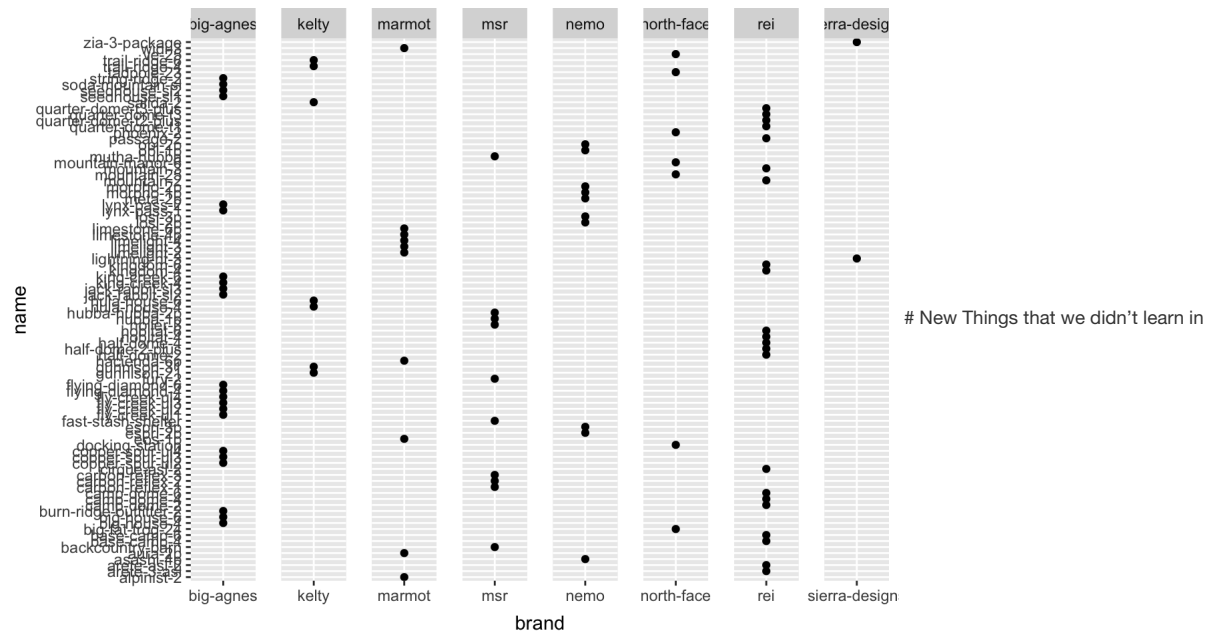
```
ggplot(dat, aes(brand, weight, color = brand)) +  
  geom_point() +  
  facet_grid(. ~ brand, scales = "free", space = "free", shrink = TRUE)
```



```
#scale_color_manual(values = c("orange", "sky blue", "violet"))
```

In the case of categorical axes (which in this example is the brand and name of the tents), we can still graph it. This example has way too many name variables so it looks a little messy but when the data set is fixed in a smaller quantity it comes in useful.

```
ggplot(dat, aes(brand, name)) +
  geom_point() +
  facet_grid(. ~ brand, scales = "free", space = "free") +
  theme(strip.text.y = element_text(angle = 90), panel.spacing.x = unit(1, "lines"))
```



class.

```
"plotly" is a package in r that makes all kinds of graphs that exceeds beyond ggplot.
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
## last_plot
```

```
## The following object is masked from 'package:stats':
##
##   filter
```

```
## The following object is masked from 'package:graphics':
##
##   layout
```

```
x <- dat$weight
y <- dat$height

data <- data.frame(x, y)

p <- plot_ly(data, x = ~x, y = ~y, name = "<b>weight", type = 'scatter', mode = 'lines') %>%
  add_trace(y = ~y - 5, name = "<b>height ", connectgaps = TRUE)
p
```

```
dat2 <- head(dat)
dat2
```

```
## # A tibble: 6 x 8
##       name      brand price weight height bestuse seasons
##       <chr>    <chr> <dbl> <dbl> <int>   <chr>   <chr>
## 1 fly-creek-ul2 big-agnes 349.95   960    96 Backpacking 3-season
## 2 fly-creek-ul3 big-agnes 449.95  1450   107 Backpacking 3-season
## 3 salida-2      kelty 159.95  1700   102 Backpacking 3-season
## 4 jack-rabbit-sl3 big-agnes 359.95  2160   107 Backpacking 3-season
## 5 passage-2      rei 149.00  2210   107 Backpacking 3-season
## 6 copper-spur-ul2 big-agnes 399.95  1530   107 Backpacking 3-season
## # ... with 1 more variables: capacity <chr>
```

```
plot_bb <- plot_ly(dat2, x = ~weight, y = ~height, text = ~name, type = 'scatter', mode = 'markers',
  marker = list(size = dat$price), opacity = 0.5) %>%
  layout(title = "count of weight and height",
    xaxis = list(showgrid = FALSE),
    yaxis = list(showgrid = FALSE))
```

## Making donut charts using plotly.

Making different charts are easy using plotly. Donut chart is something that we didn't cover in class but it comes in handy for more graphics than pie chart.

```
plot_ct1 <- dat %>%
  group_by(brand) %>%
  summarize(count = n()) %>%
  plot_ly(labels = ~brand, values = ~count) %>%
  add_pie(hole = 0.6) %>%
  layout(title = "Donut chart of brands", showlegend = F, xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
    yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
plot_ct1
```

In the end, the grammar of graphics is just an easier way to present analysis of the dataset. However, the graphs that can be made within ggplot packages are helpful in facilitating the understanding of each data set. In order to understand the dataset, every graph needs to be clear. Ggplot and grammar of graphics is not merely for the aesthetic but more for technical reasons.

## Reference

1. <https://ramnathv.github.io/pycon2014-r/visualize/ggplot2.html>
2. <https://github.com/ucb-stat133/stat133-fall-2017/blob/master/slides/13-grammar-graphics.pdf>
3. <http://ggplot2.org/resources/2007-vanderbilt.pdf>
4. [http://ggplot2.tidyverse.org/reference/facet\\_grid.html](http://ggplot2.tidyverse.org/reference/facet_grid.html)
5. <http://ggplot2.tidyverse.org/reference/theme.html>
6. <https://stackoverflow.com/questions/16140582/r-draw-abline-lm-line-of-best-fit-through-arbitrary-point>
7. [https://www3.nd.edu/~steve/computing\\_with\\_data/13\\_Facets/facets.html](https://www3.nd.edu/~steve/computing_with_data/13_Facets/facets.html)