Intro to ggplot2

Colleen Nell

5/12/2017

ggplot2

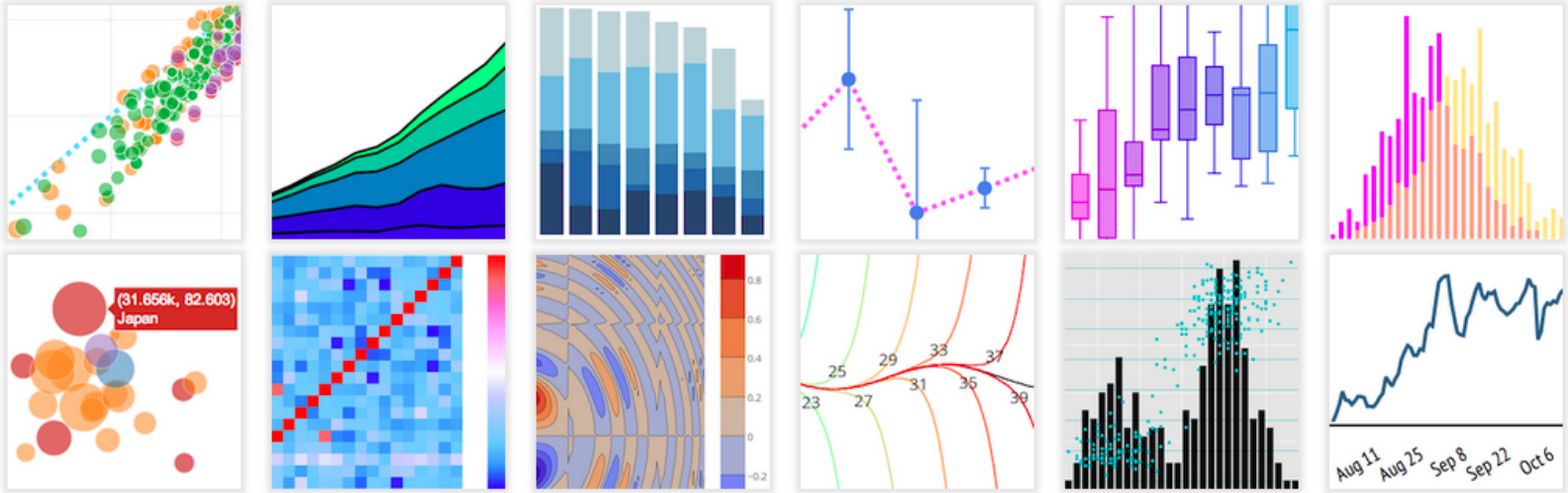
```
install.packages(c('ggplot2','data.table')
library(ggplot2) # plotting
library(data.table) #fread
```

```
install.packages('tidyverse')
library(tidyverse) #includes ggplot2, dplyr, tidy, + more
```

Grammar of ggplot

Geoms

Geometric objects (geoms)



Reference: http://ggplot2.tidyverse.org/reference/

Aesthetics

- Assign coordinates (x,y)
- color, fill, shape, size, alpha ++
- aes() maps data to the geom

```
ggplot()+geom_point(data, aes(x, y))
```

Dataset

Burritos in southern California https://srcole.github.io/100burritos/

Southern California Burritos

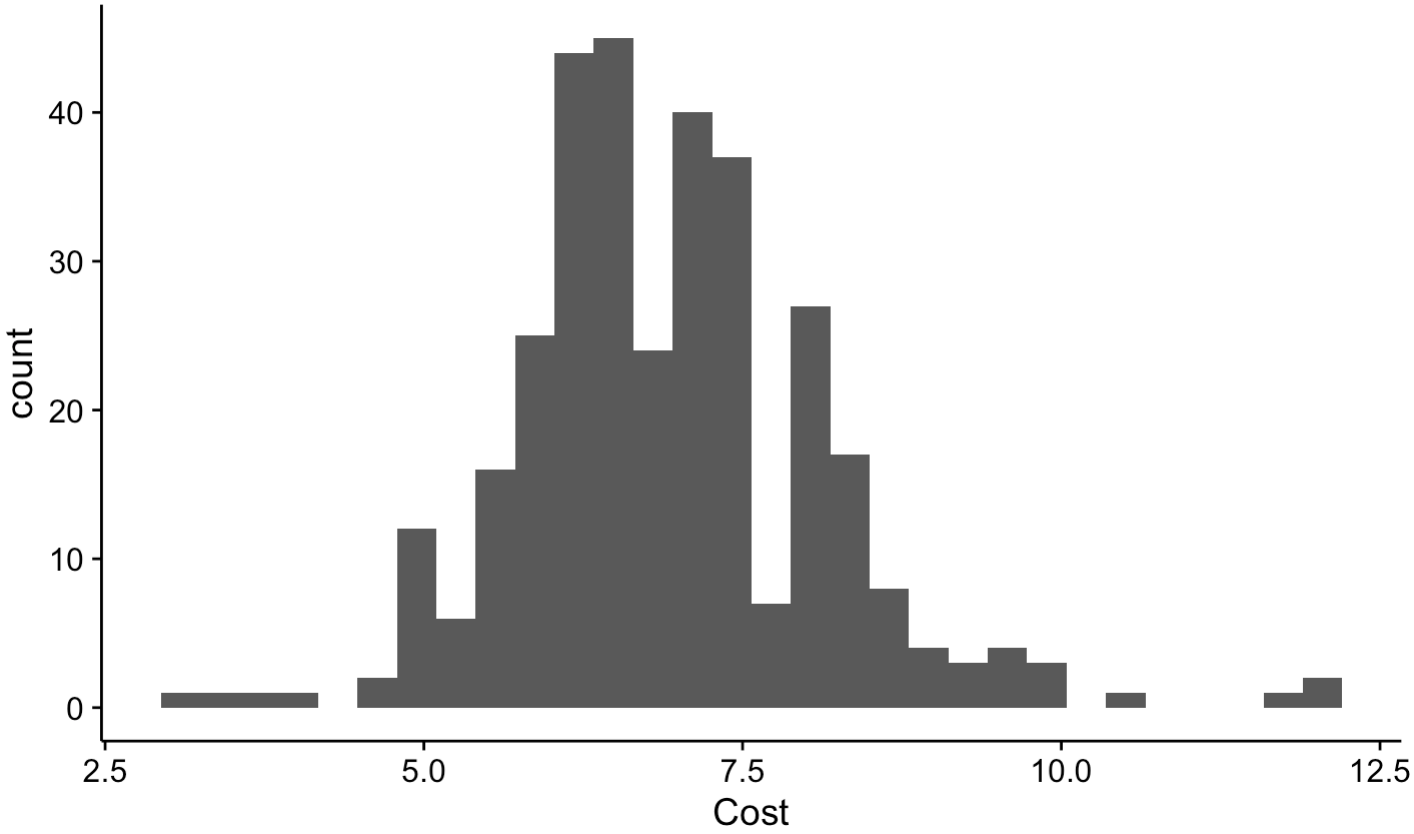
```
url<-'https://raw.githubusercontent.com/collnell/burritos/master/sd_burritos.csv'
ritos <- fread(url)
```

Building a plot

```
ggplot()
```

Mapping data & aes

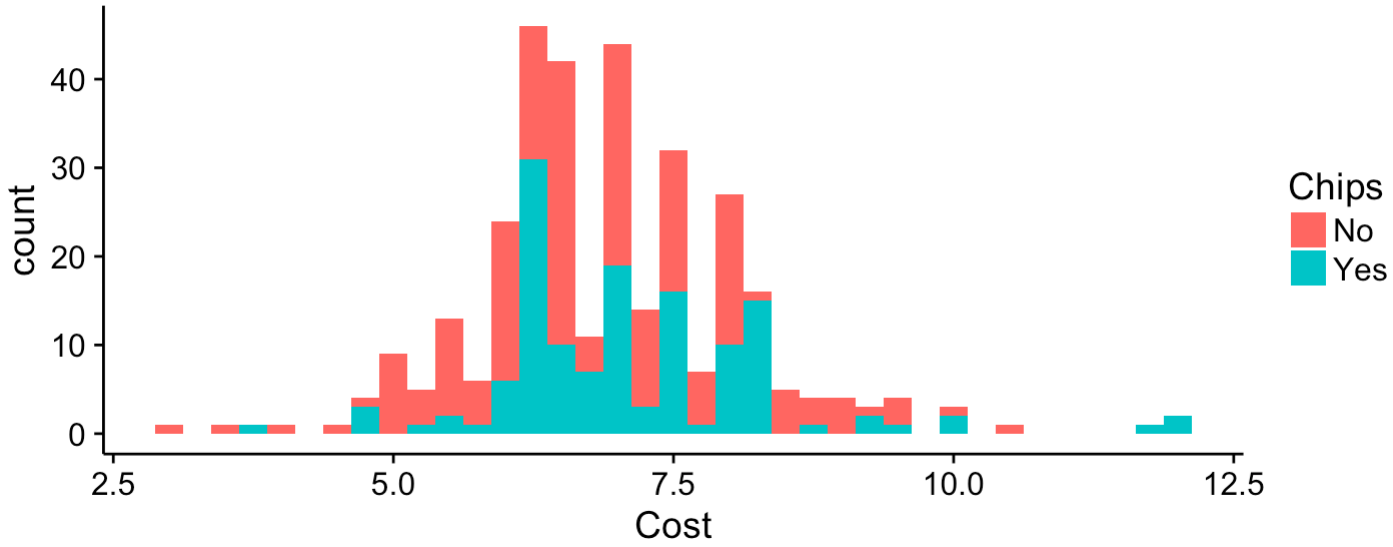
```
ggplot()+
  geom_histogram(data=ritos, aes(x=Cost))
```



Aesthetics

Data & aes() mapping can be applied to each geom:

```
ggplot()+
  geom_histogram(data=ritos, aes(x=Cost, fill=Chips), binwidth = .25)
```

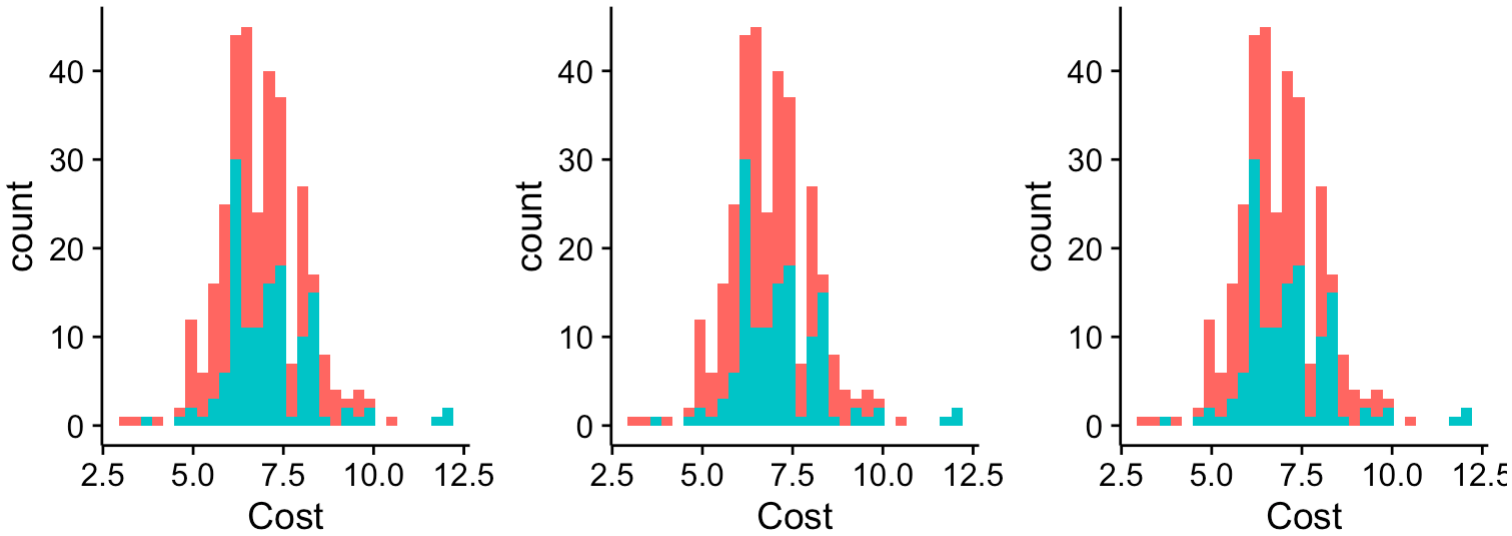


Or to all layers:

```
ggplot(data=ritos, aes(x=Cost, fill=Chips))+
  geom_histogram()
```

Or between both:

```
ggplot(data=ritos, aes(x=Cost))+
  geom_histogram(aes(fill=Chips))
```

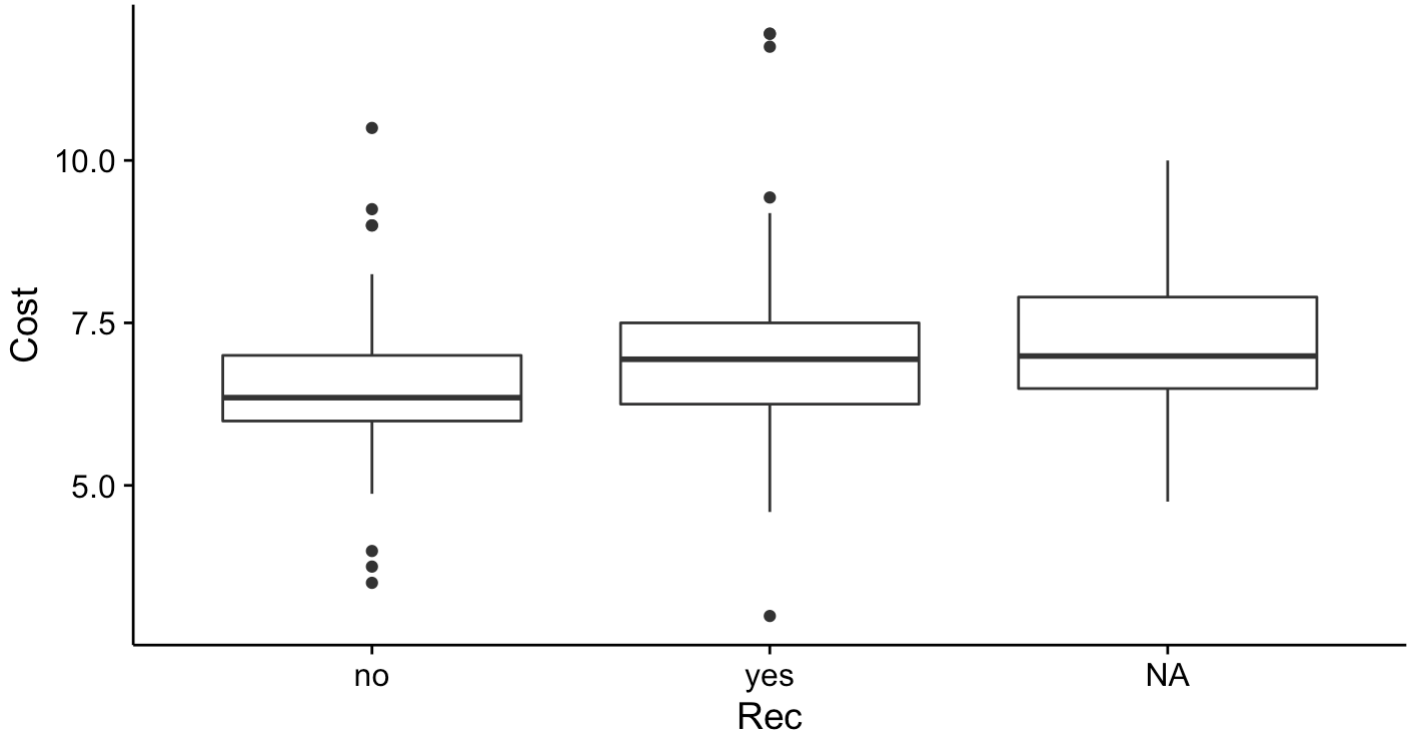


Layering geoms

Multiple geoms can be plotted together using '+'

Use geom_boxplot to examine burrito cost by recommendation (Rec):

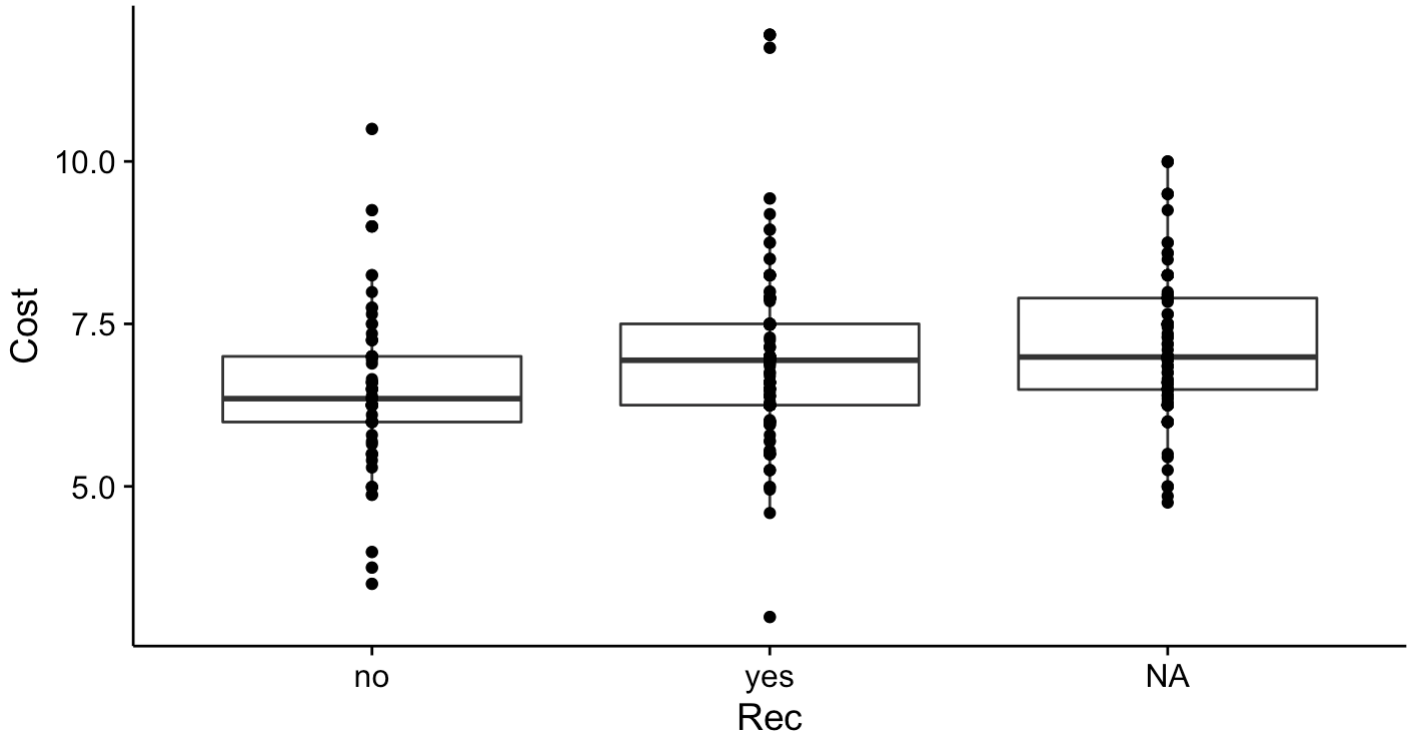
```
ggplot(data=ritos, aes(x=Rec, y=Cost))+
  geom_boxplot()
```



Layering geoms

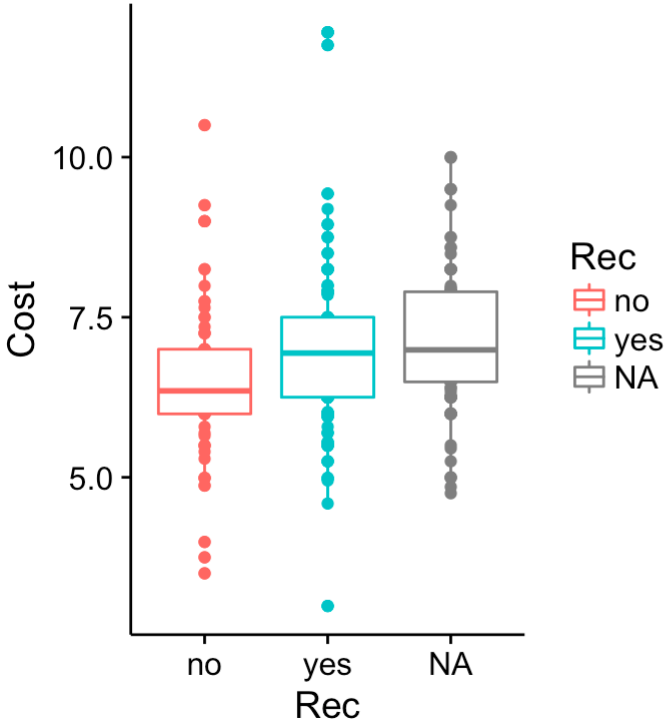
Add layer showing data points with boxplot:

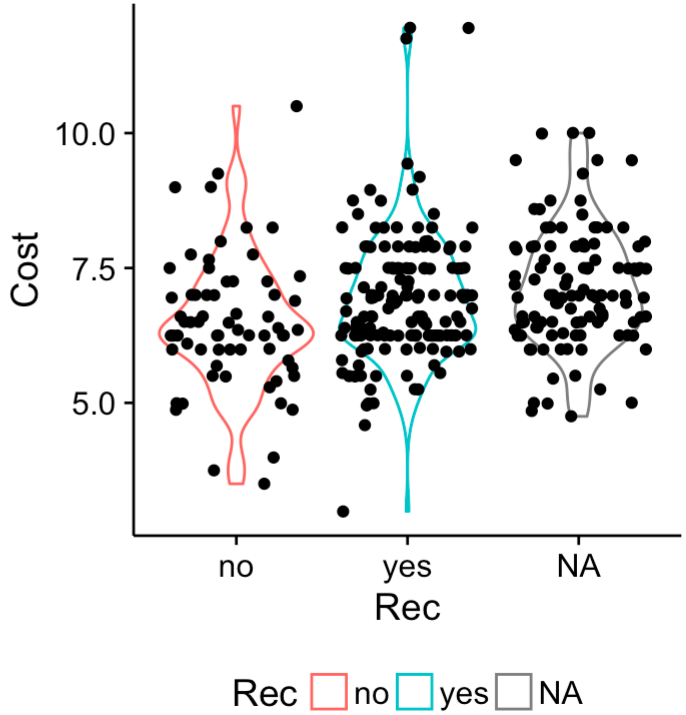
```
ggplot(data=ritos, aes(x=Rec, y=Cost))+
  geom_boxplot()+
  geom_point()
```



Mapping style aesthetics

Color, fill, shape, size, and alpha can be mapped to variables inside 'aes()'.

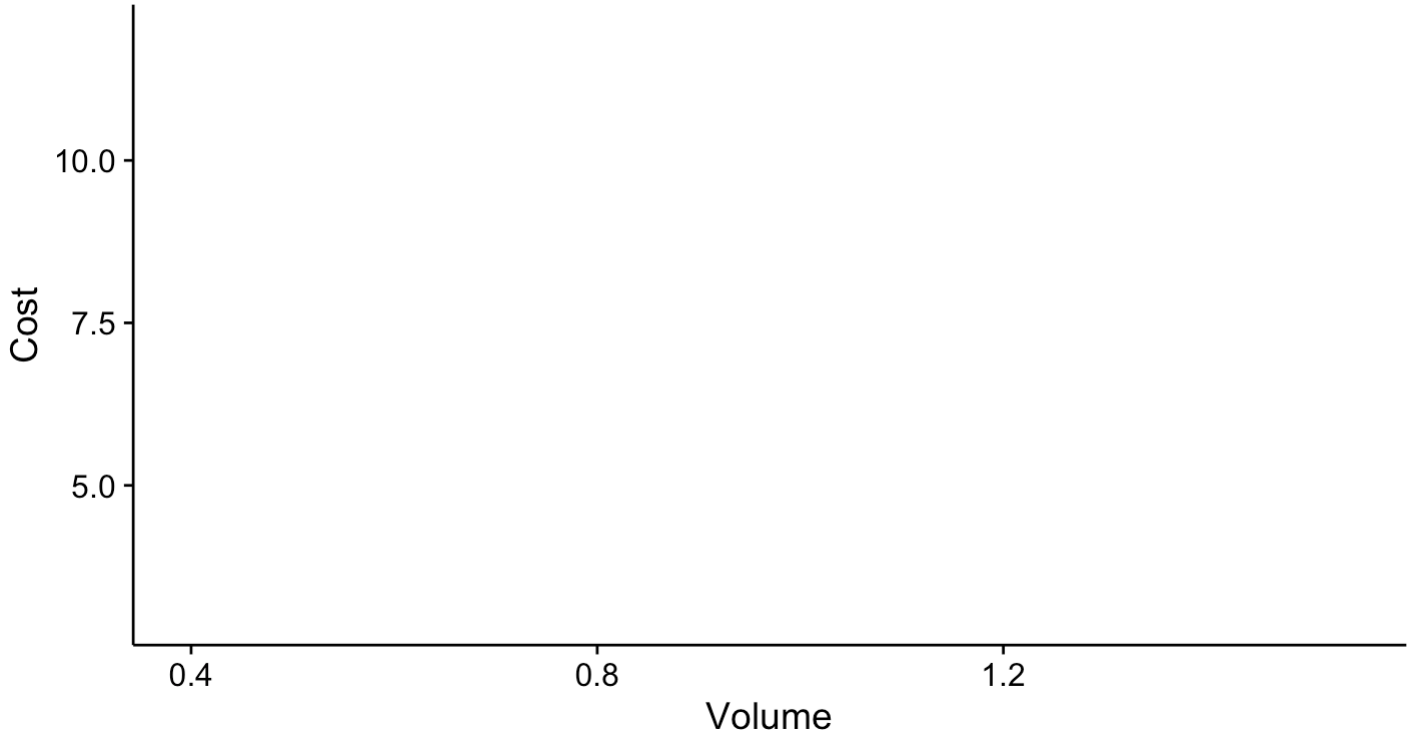




Create a scatterplot with Volume on the x-axis and Cost on y-axis

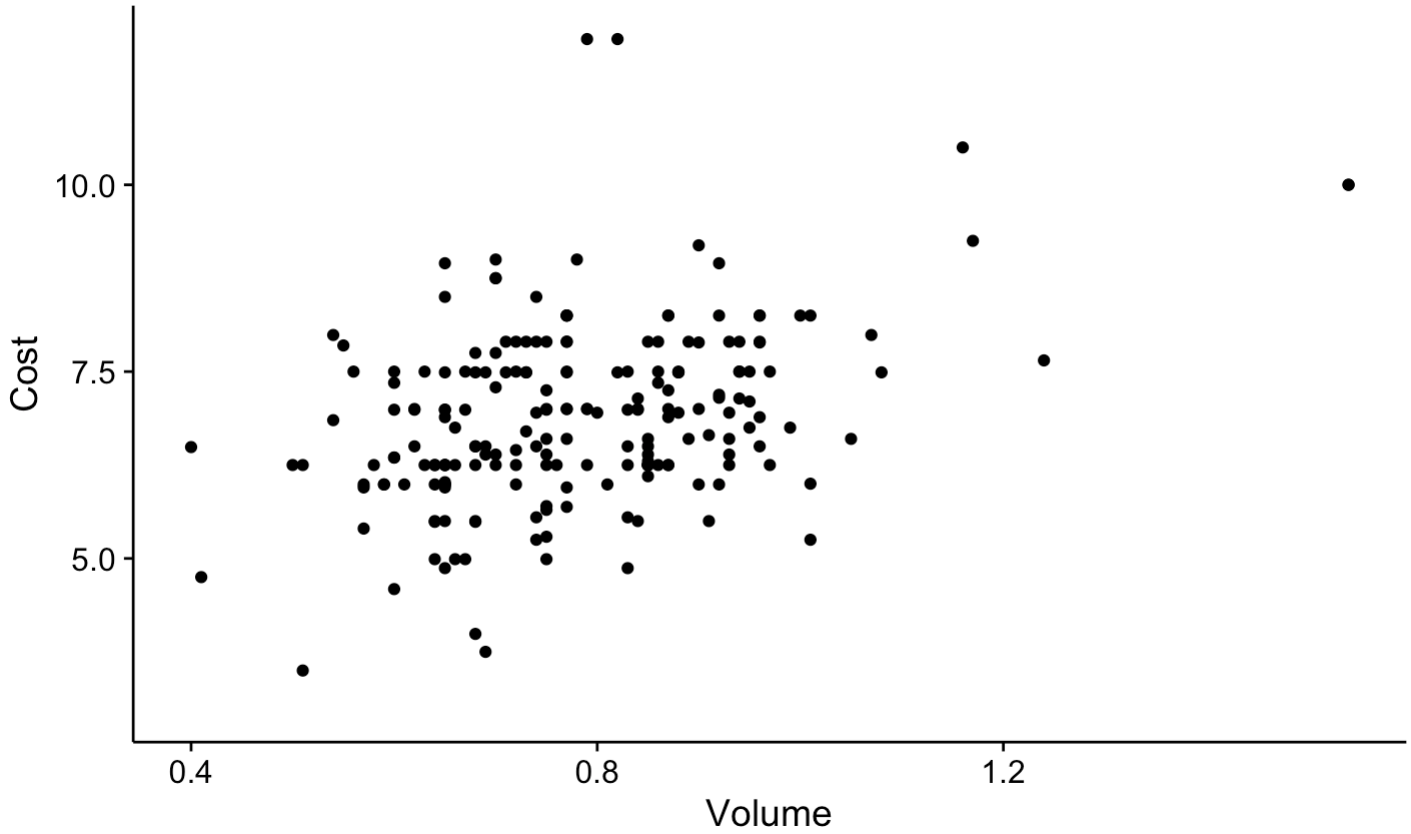
Start with base plot:

```
k<-ggplot(data=ritos, aes(x=Volume, y=Cost))
k
```



Add points for scatterplot:

```
k+geom_point()
```

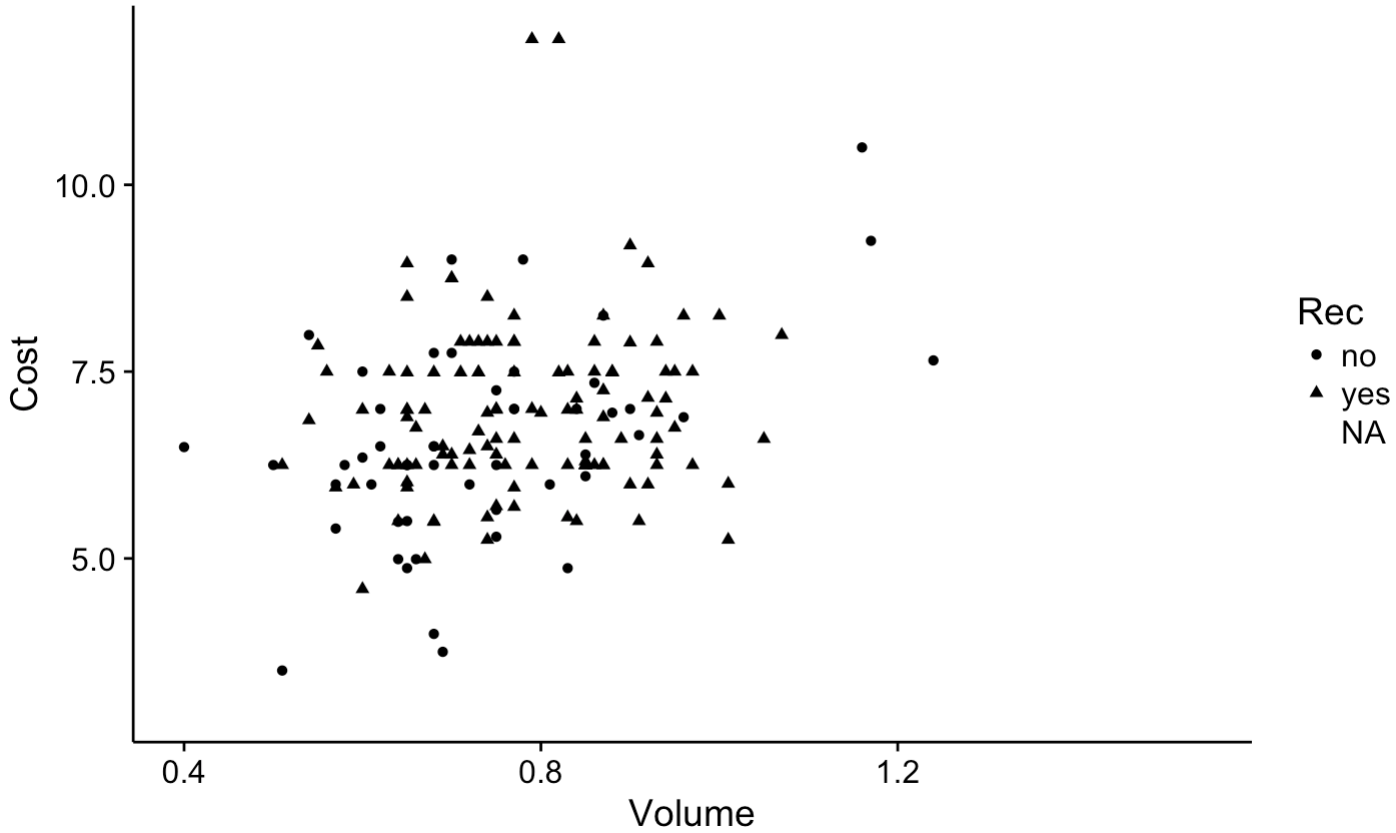


Map levels of a variable using shapes

```
ritos$Rec<-as.factor(ritos$Rec)
levels(ritos$Rec) #aes maps to levels of a variable
```

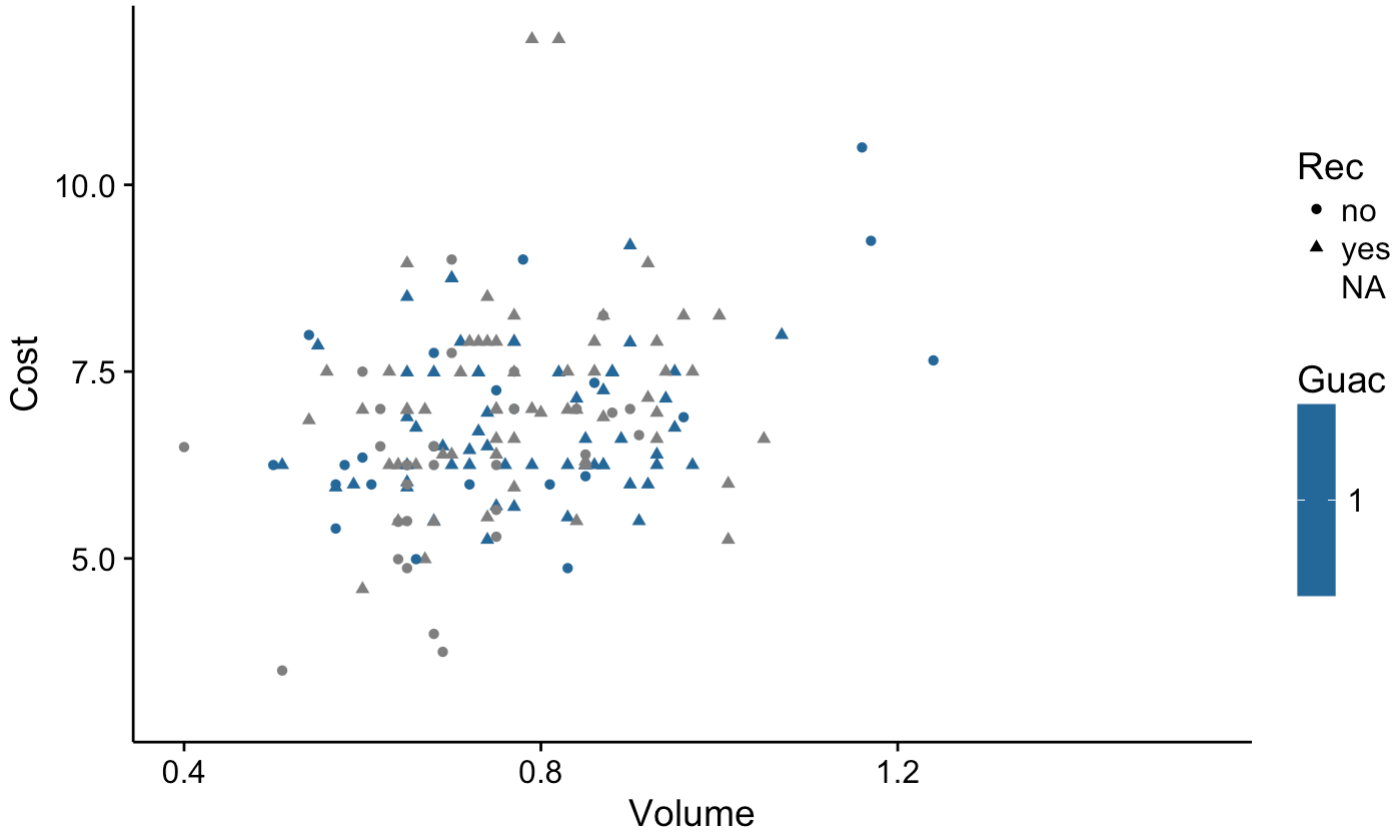
```
[1] "no"  "yes"
```

```
k+geom_point(aes(shape=Rec))
```



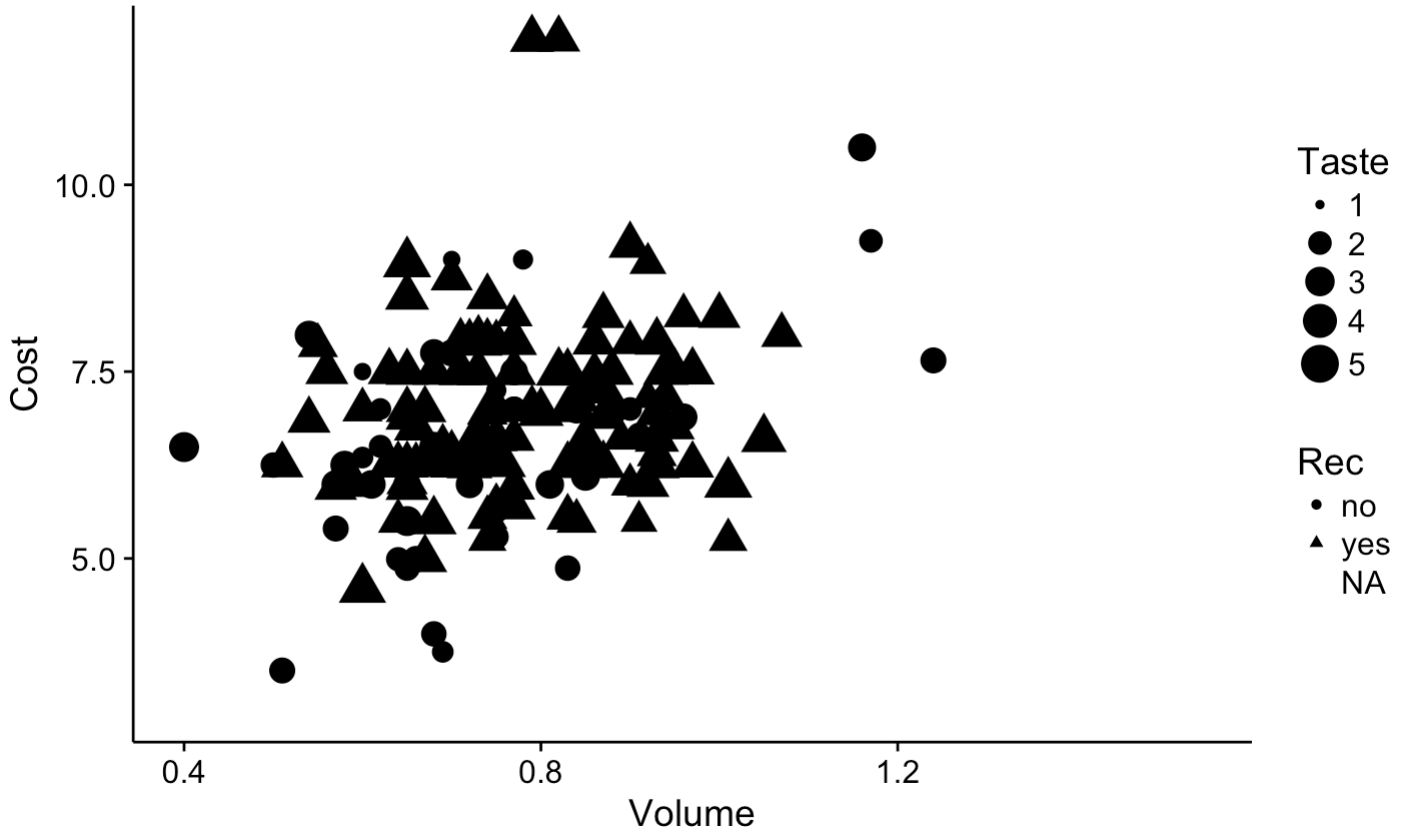
Map geom color to show whether or not the burrito had guacaomle (Guac)

```
k+geom_point(aes(shape=Rec, color=Guac))
```



Size

```
k+geom_point(aes(shape=Rec, size=Taste))
```



##

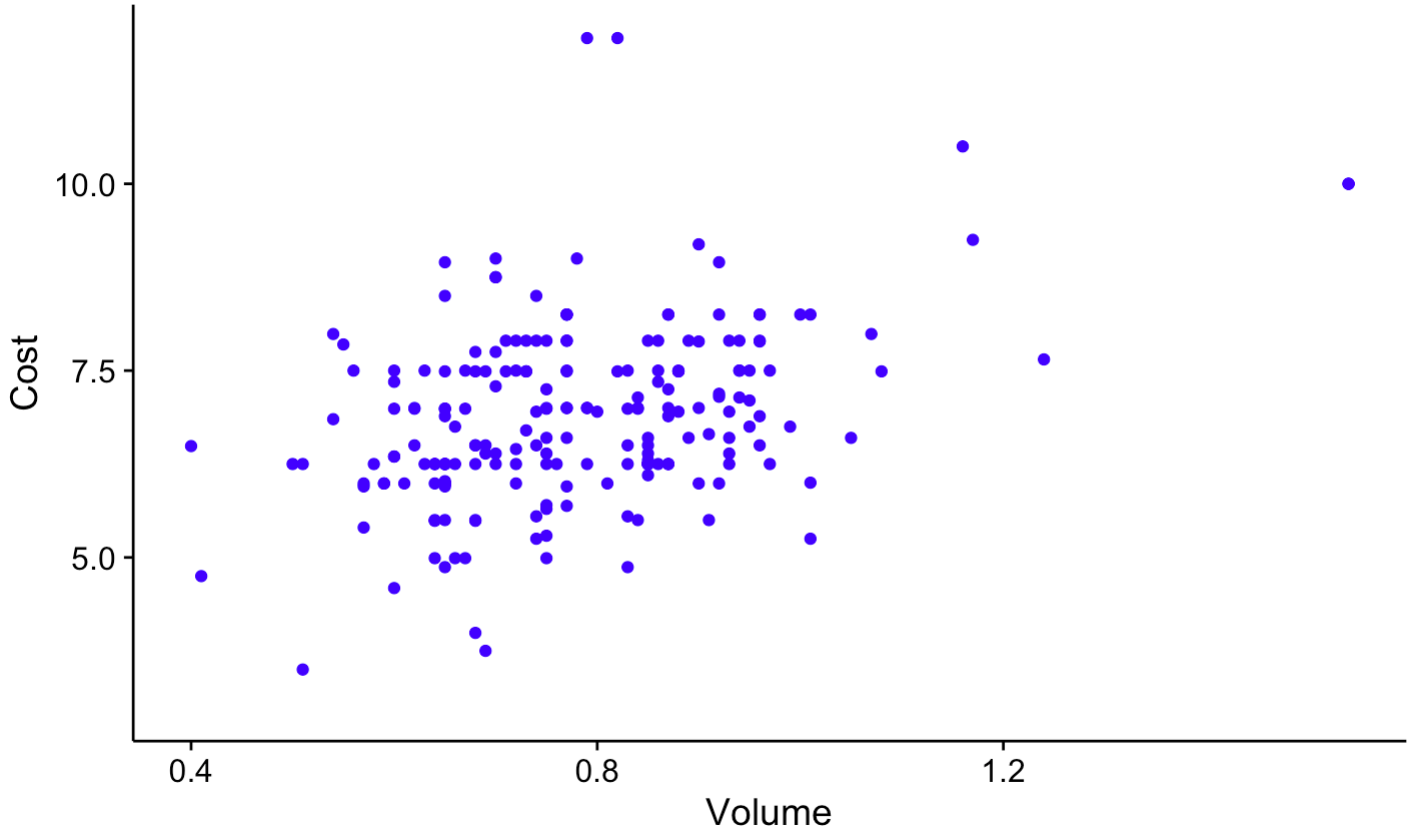
Alpha = transparency

```
k+geom_point(aes(shape=Rec, alpha=Taste))
```

Setting vs. mapping aesthetics

You can also set aesthetic properties manually by assigning them outside of 'aes()'

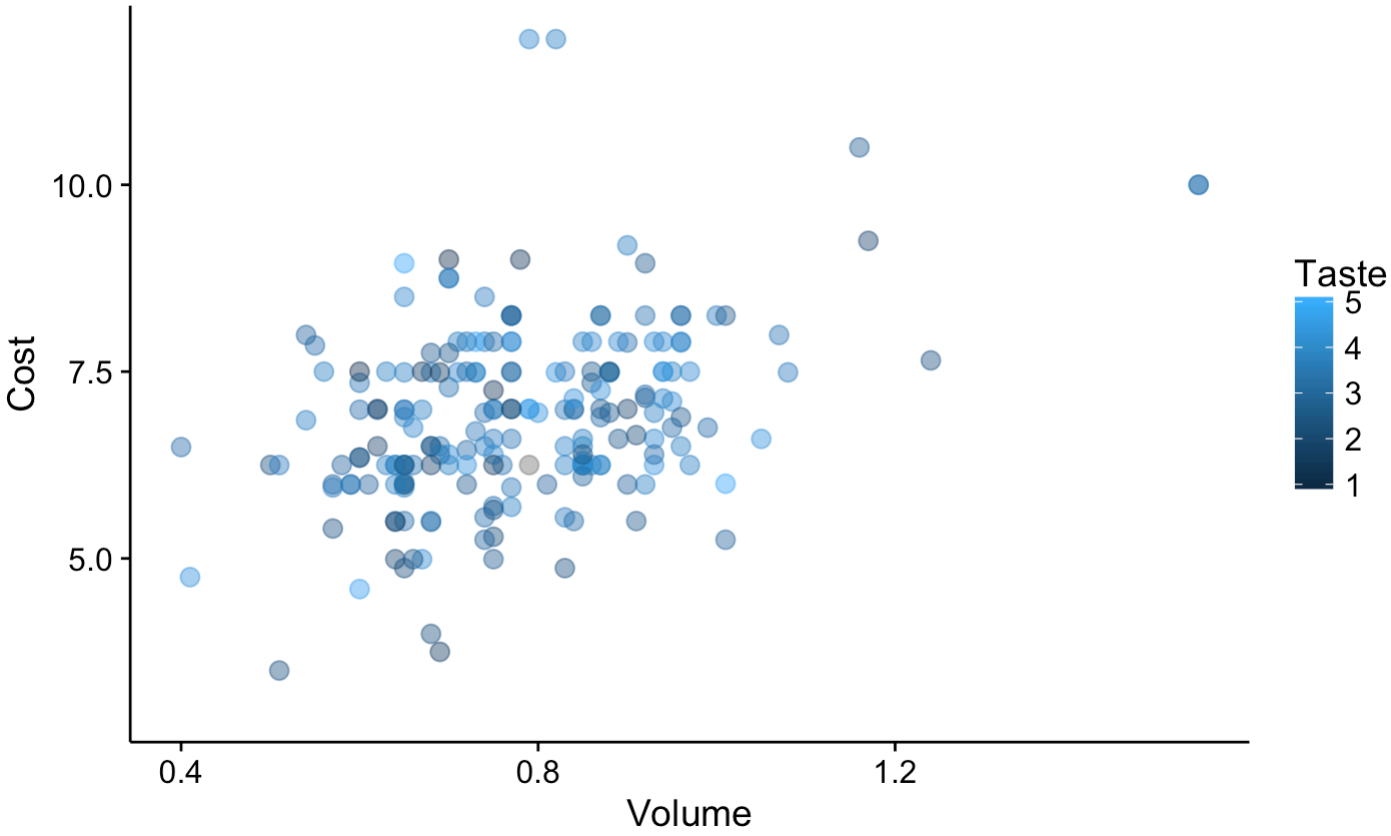
```
k + geom_point(color='blue')
```




```
k + geom_point(aes(color=Taste), shape=21, alpha=.75, size=3)
```

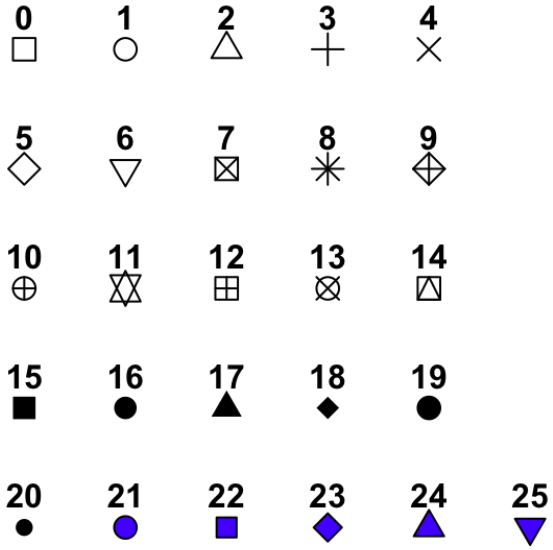
Fill & color assign color to different elements of the geom.

```
k + geom_point(aes(color=Taste), alpha=.5, size=3)
```

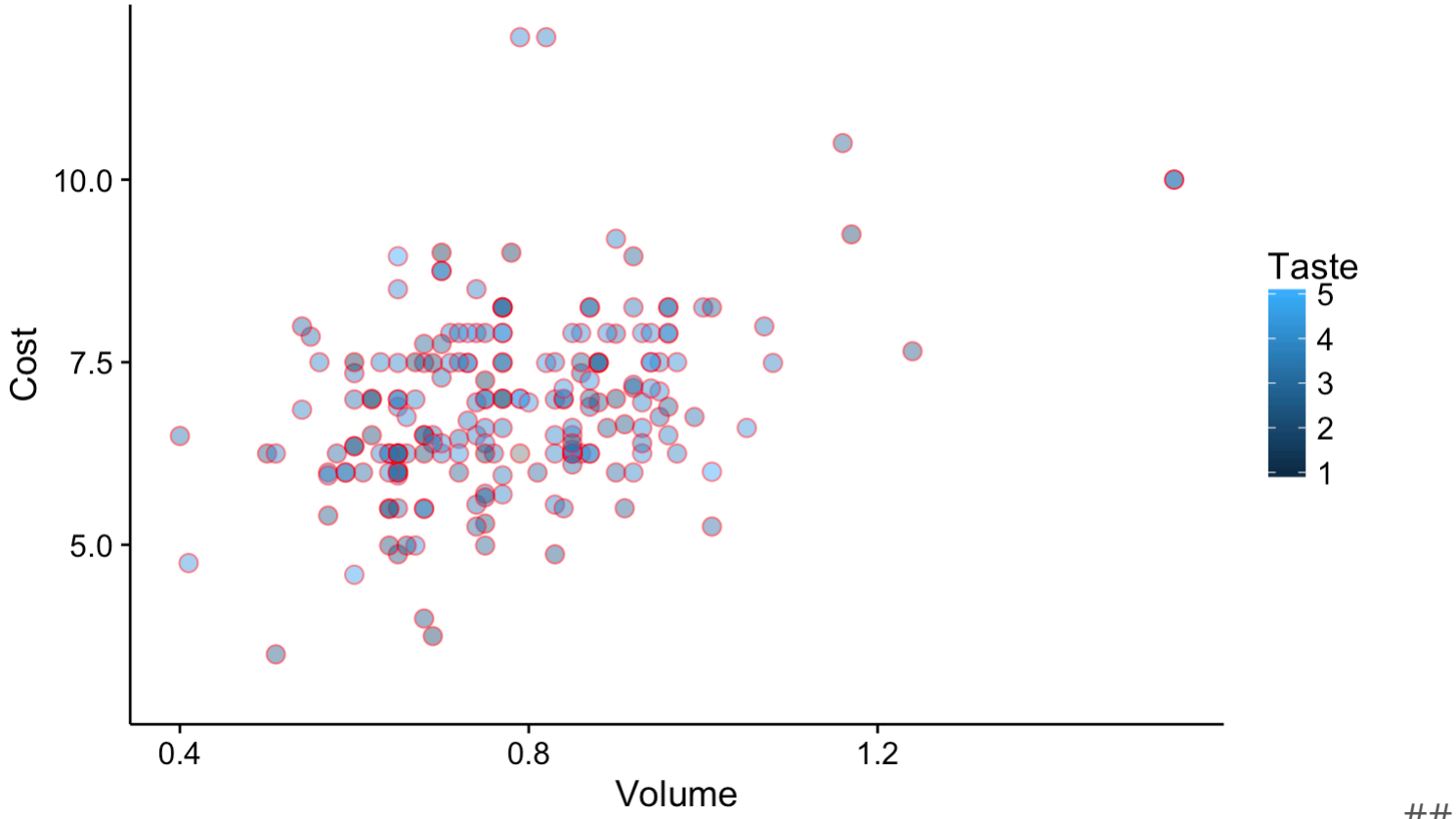



```
#alpha ranges from 0 (transparent) to 1 (opaque)
# size is in mm
```

Shapes:



```
k + geom_point(aes(fill=Taste), color='red',shape=21, alpha=.5, size=3)
```



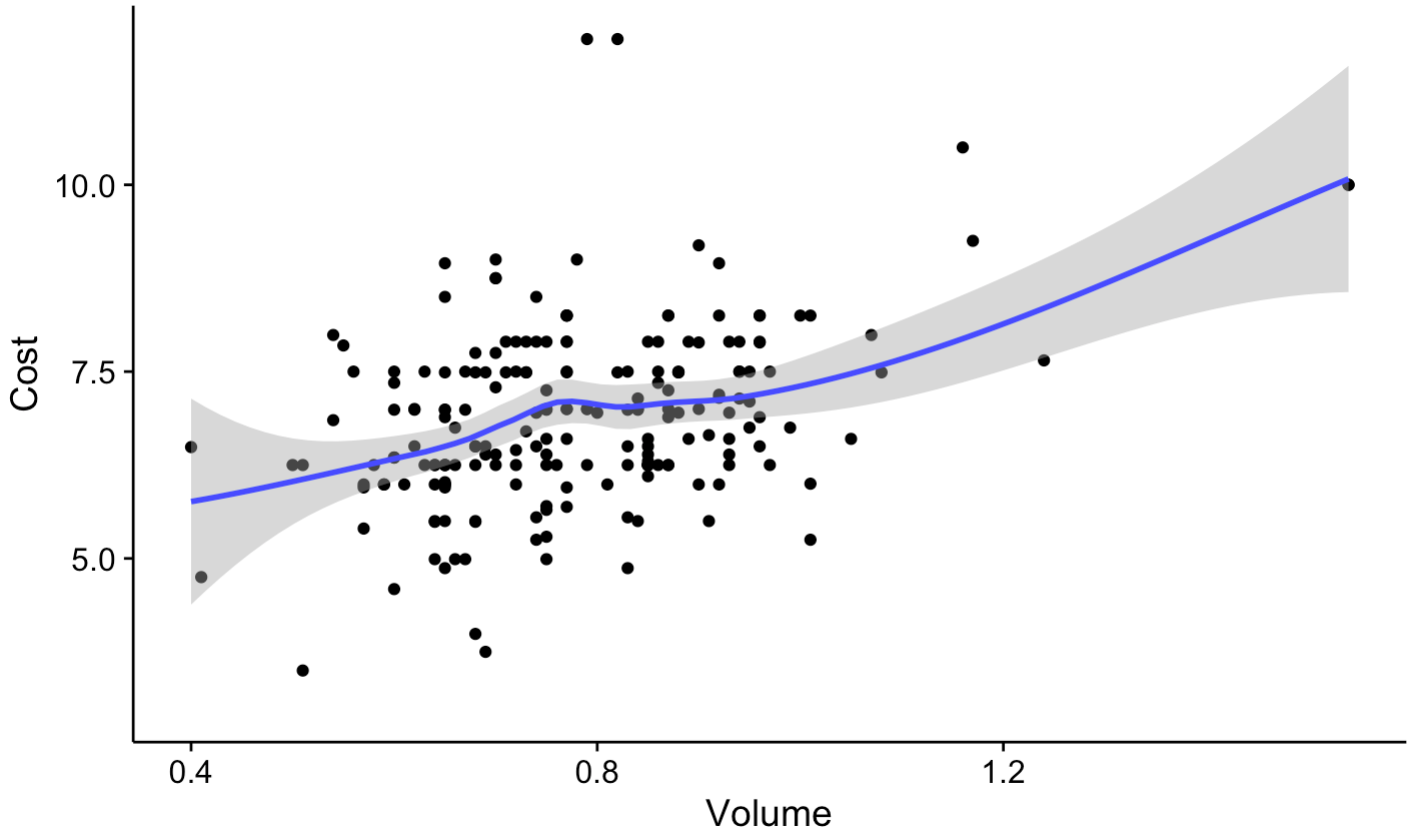
##

```
k + geom_point(aes(fill=Taste), color='red',shape=21, alpha=.5, size=3, stroke = 3)
```

geom_smooth

Add trend line:

```
k+geom_point()+
  geom_smooth()
```

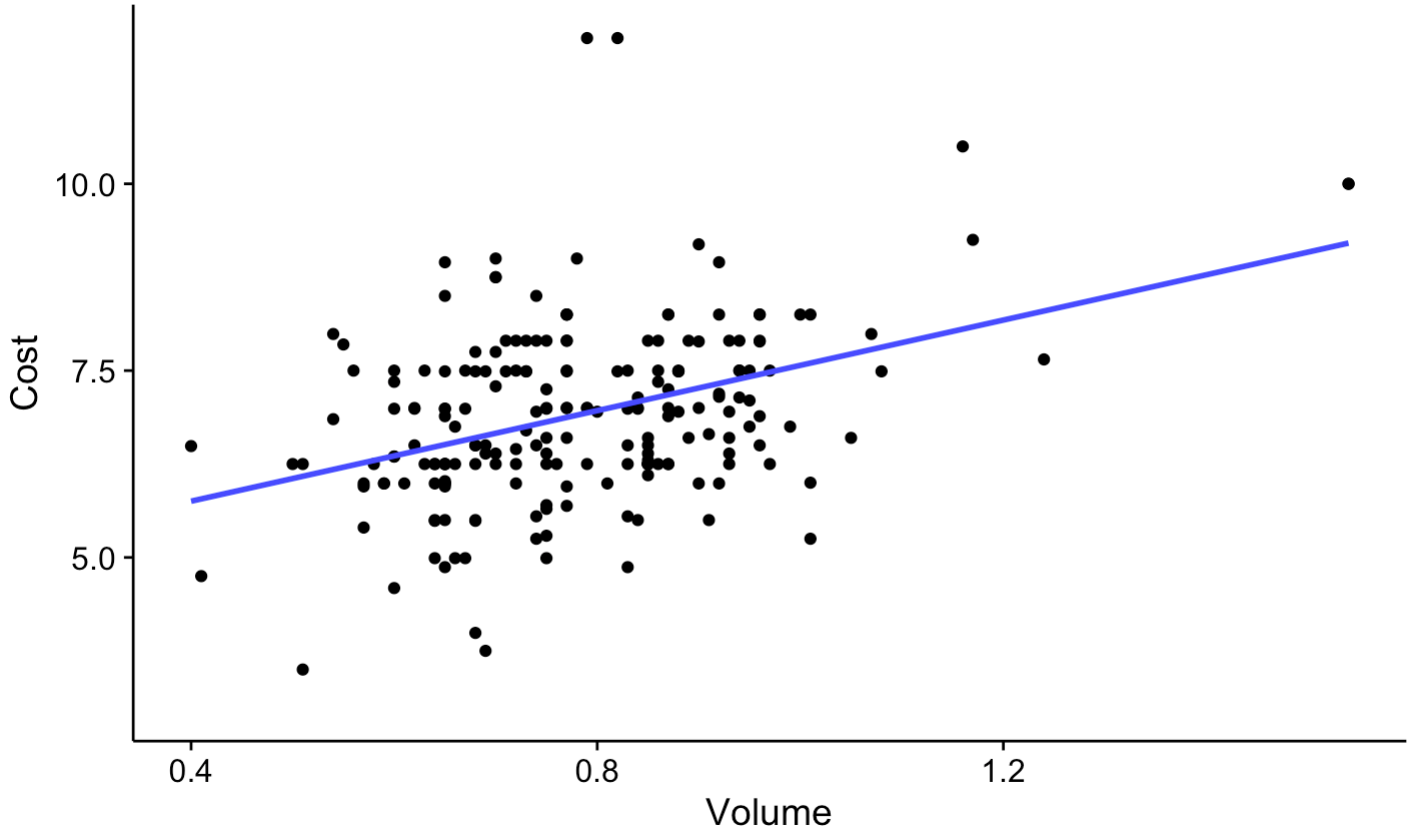


geom_smooth

method - lm, glm, loess

geom_smooth

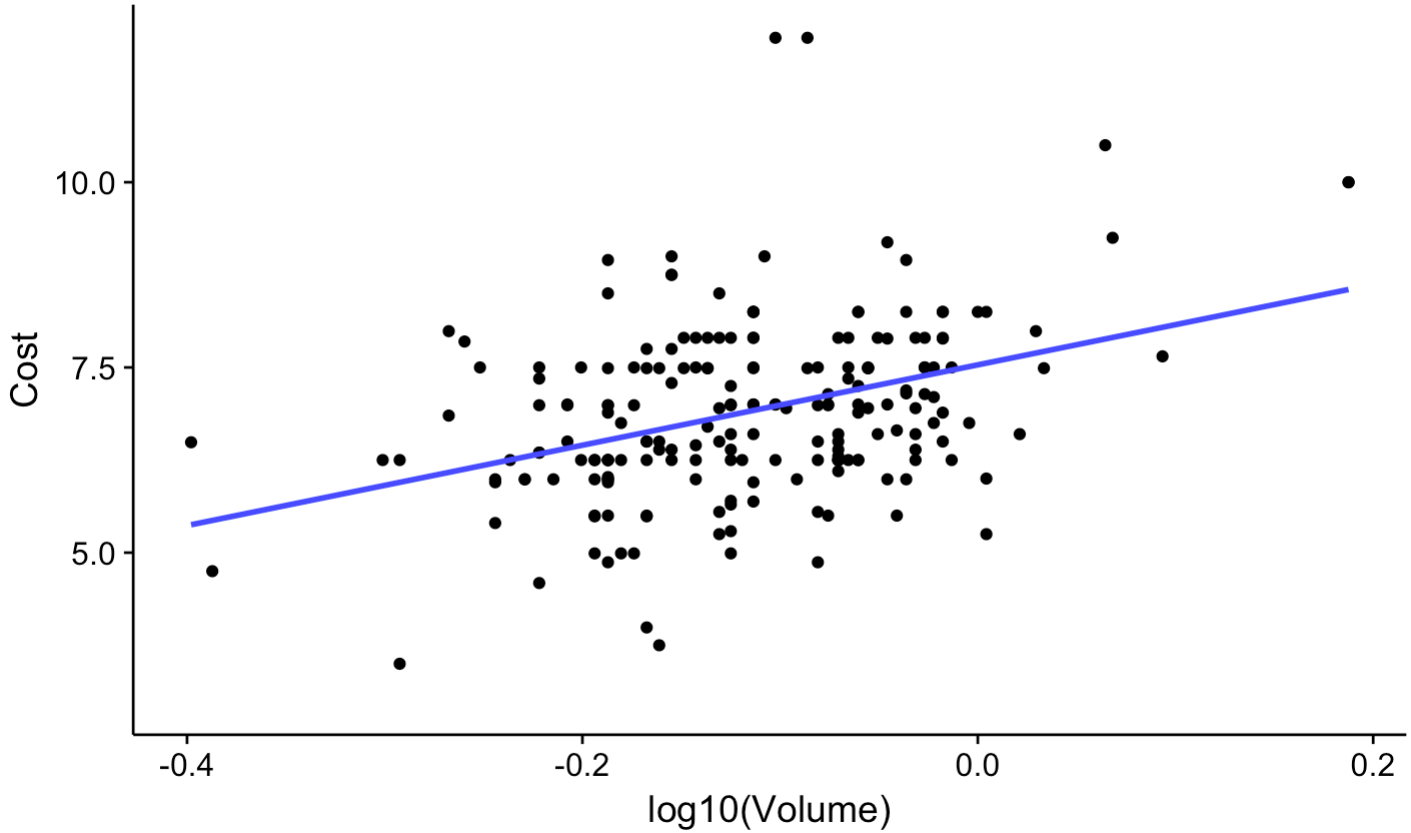
```
k+geom_point()+
  geom_smooth(method='lm', se=F)
```



Scales

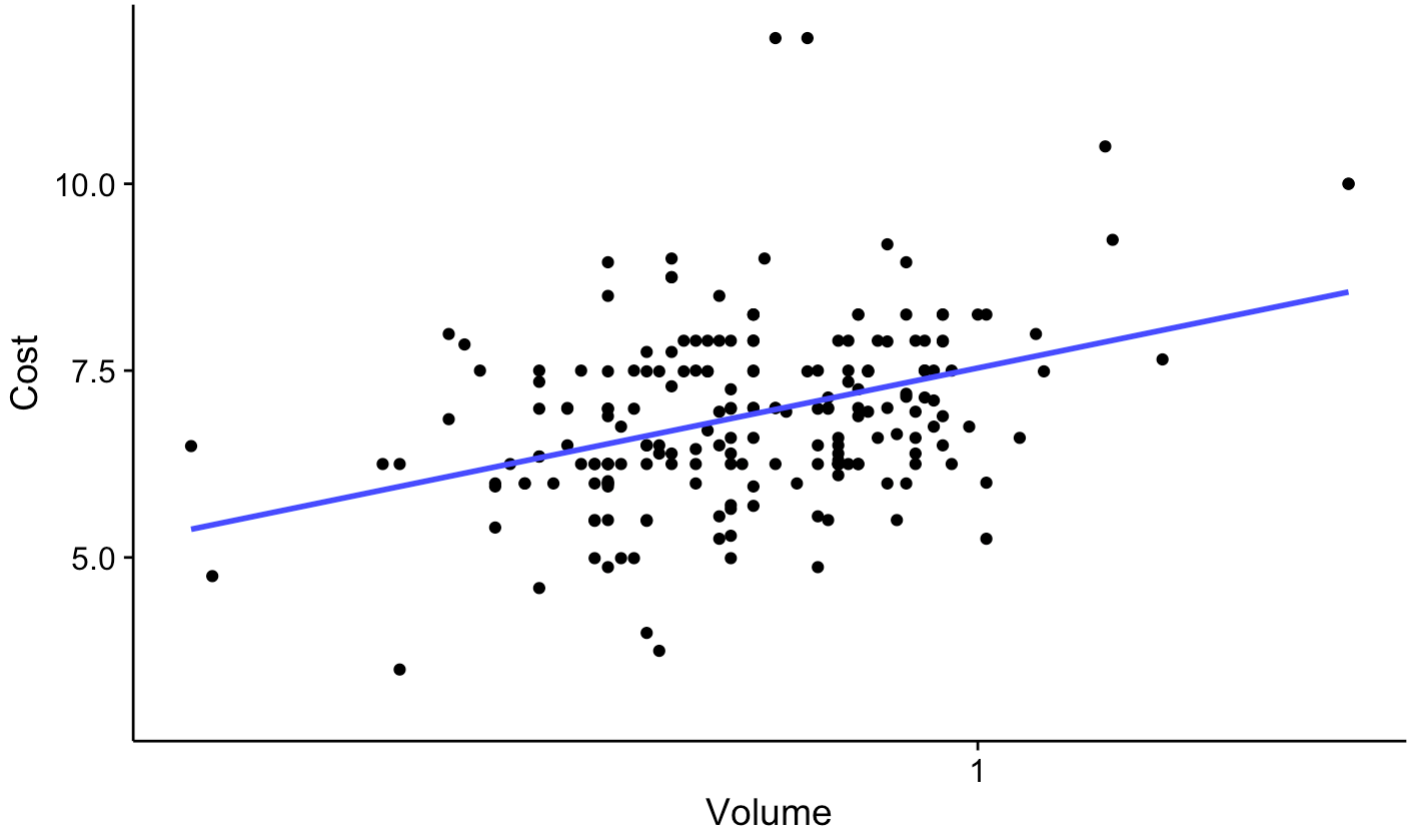
Control the mapping of data and aesthetics
scale_x_reverse()
scale_y_reverse()
scale_x_log10()

```
ggplot(ritos, aes(log10(Volume), Cost))+
  geom_point()+
  geom_smooth(method='lm', se=F)
```



Transform the scale instead of the data.
How is this different than before?

```
ggplot(ritos, aes(Volume, Cost))+
  geom_point()+
  geom_smooth(method='lm', se=F)+
  scale_x_log10()
```



http://ggplot2.tidyverse.org/reference/index.html

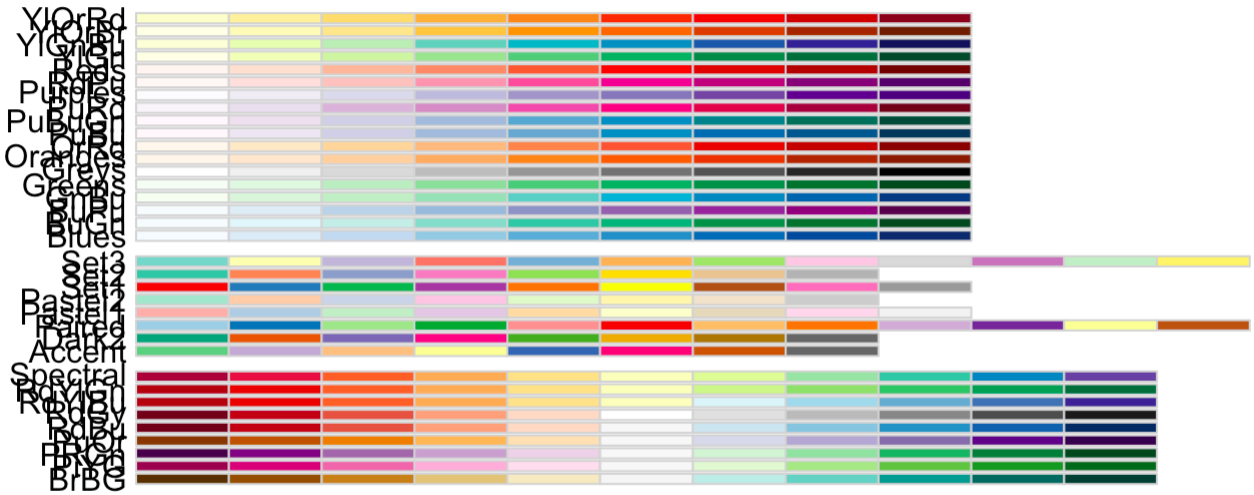
Color scales

Wes Anderson - https://github.com/karthik/wesanderson
virdis - https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html
RColorBrewer - http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3
Color cheatsheet - https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/colorPaletteCheatsheet.pdf

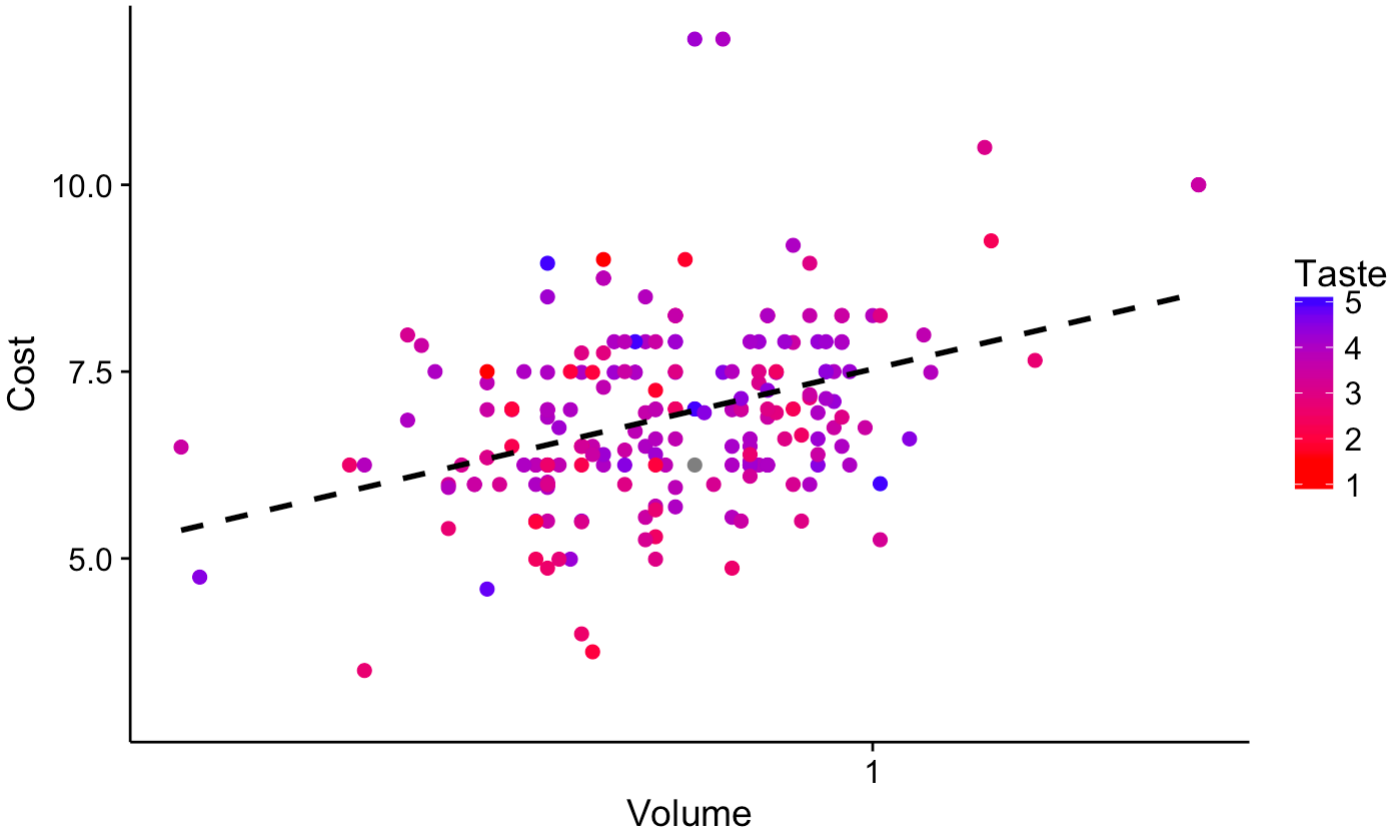
Color scales

```
#install.packages('RColorBrewer')
library(RColorBrewer)

display.brewer.all()
```



```
ggplot(ritos, aes(Volume, Cost))+
  geom_point(aes(color = Taste), size=2)+
  geom_smooth(method='lm', se=F, color='black', lty='dashed')+
  scale_x_log10()+
  scale_color_gradient(low='red', high='blue')
```

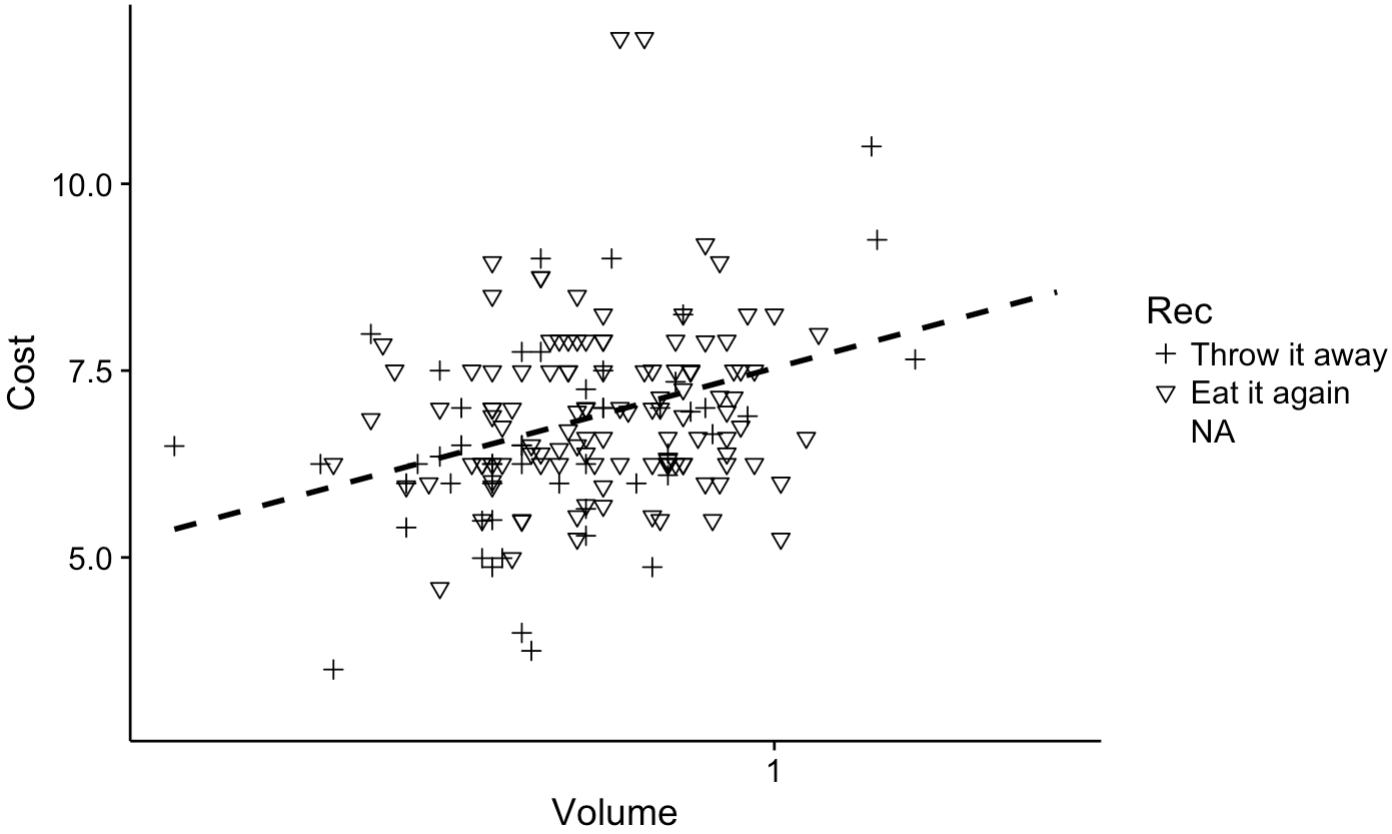


Scales

scale_color_gradient - sequential color scale
scale_color_gradient2 - diverging color scale
scale_fill_gradient
scale_shape_discrete
scale_shape_manual - supply own values

scale_ + color/fill/size/shape/linetype/alpha_ + gradient/discrete/manual

```
ggplot(ritos, aes(Volume, Cost))+
  geom_point(aes(shape = Rec), size=2)+
  geom_smooth(method='lm', se=F, color='black', lty='dashed')+
  scale_x_log10()+
  scale_shape_manual(values = c(3,6), labels = c('Throw it away','Eat it again'))
```



Coordinates

Occur after statistics, affect geom appearance

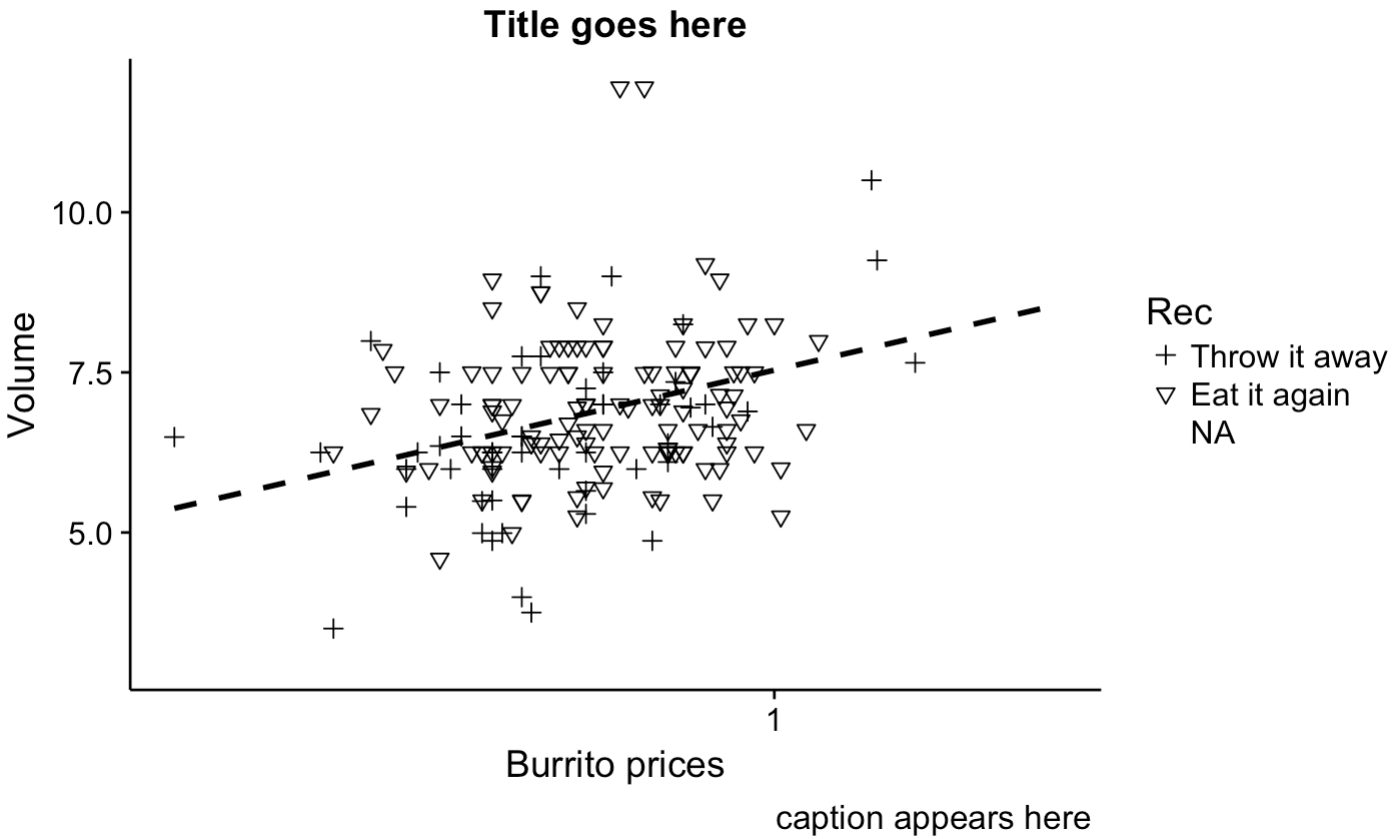
To make a pie chart:

```
?coord_polar
```

```
d<-last_plot()
d+coord_flip()
```

Plot and axis titles

```
d+labs(title='Title goes here', y='Volume',
       x='Burrito prices', caption='caption appears here ')
```



Multiple plots

```
install.packages('cowplot')
library(cowplot)

?plot_grid
```

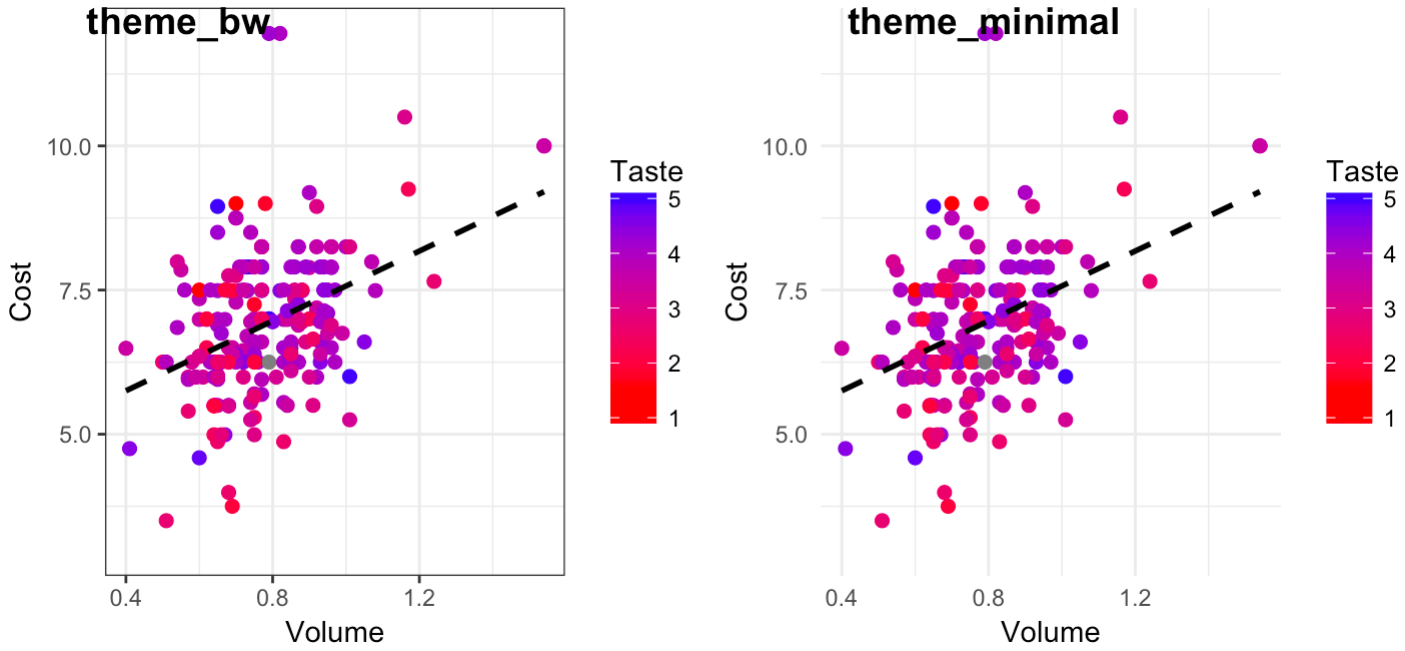
Theme

```
g<-ggplot(ritos, aes(Volume, Cost))+
  geom_point(aes(color = Taste), size=2)+
  geom_smooth(method='lm', se=F, color='black', lty='dashed')+
  scale_color_gradient(low='red', high='blue')

g_bw<-g+theme_bw()

g_min<-g+theme_minimal()

plot_grid(g_bw, g_min, nrow=1, ncol=2, labels=c('theme_bw','theme_minimal'))
```

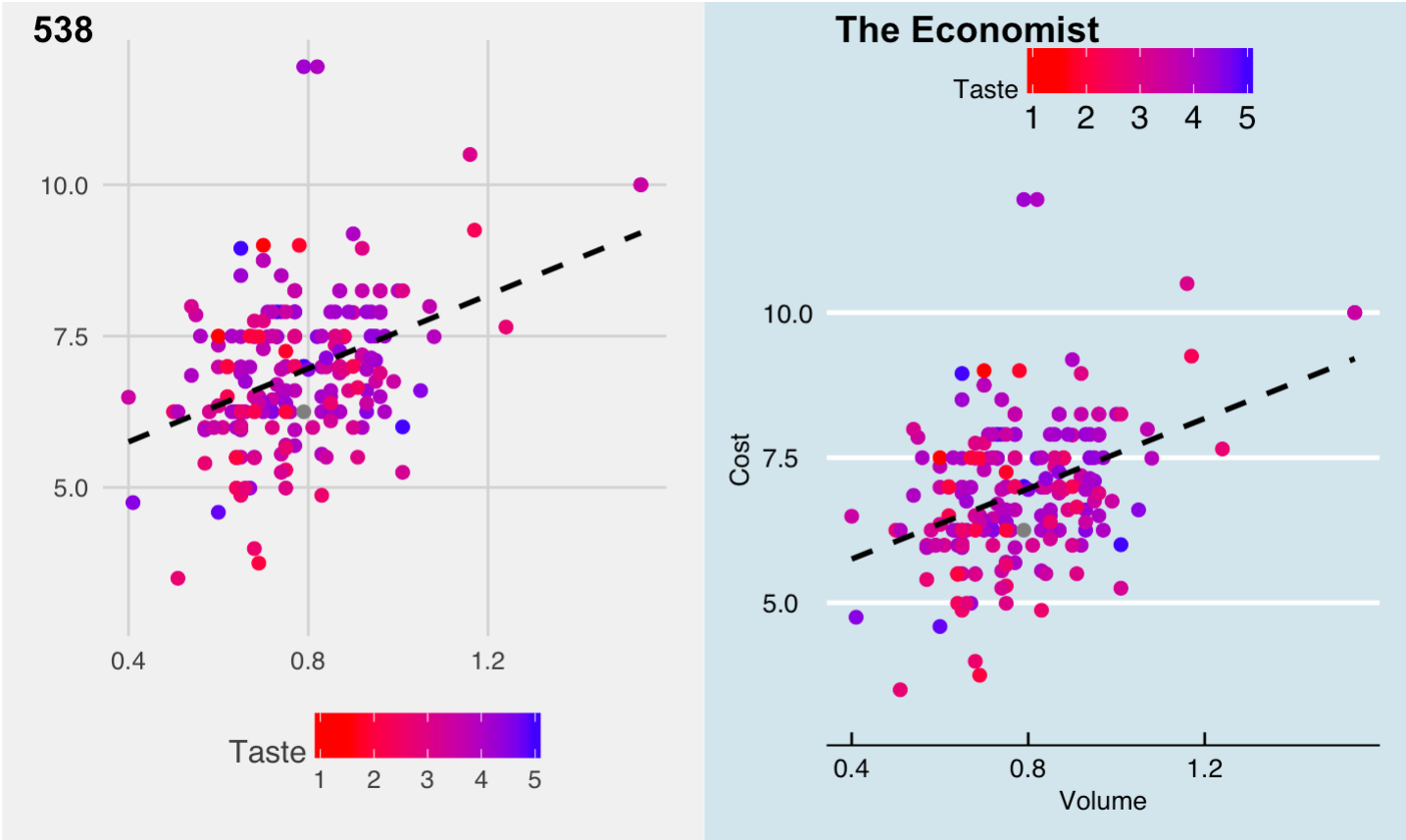


```
library(ggthemes)

g_538<-g+theme_fivethirtyeight()

g_econ<-g+theme_economist()

plot_grid(g_538, g_econ, nrow=1, ncol=2, labels=c('538','The Economist'))
```

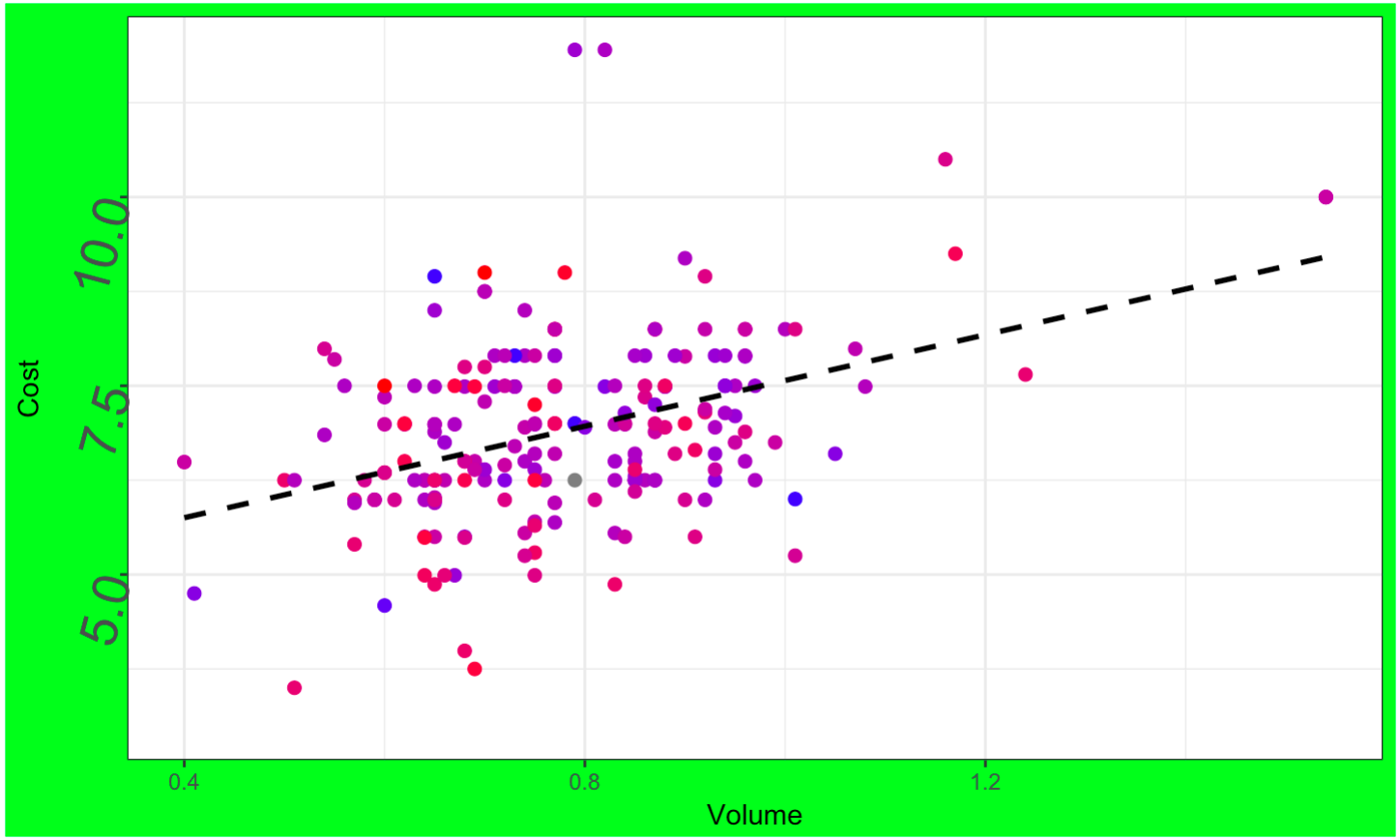


https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html

Themes

Or independent elements can be set manually

```
g_bw + theme(legend.position = 'none', plot.background = element_rect(fill='green'),axis.text.y=element_text(angle=75,
```




```
g_bw+theme_mooney()
```

Resources

ggplot2 cheatsheet - https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Symbols and color palettes - http://vis.supstat.com/2013/04/plotting-symbols-and-color-palettes/

Color cheatsheet - https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/colorPaletteCheatsheet.pdf

ggthemes - https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html