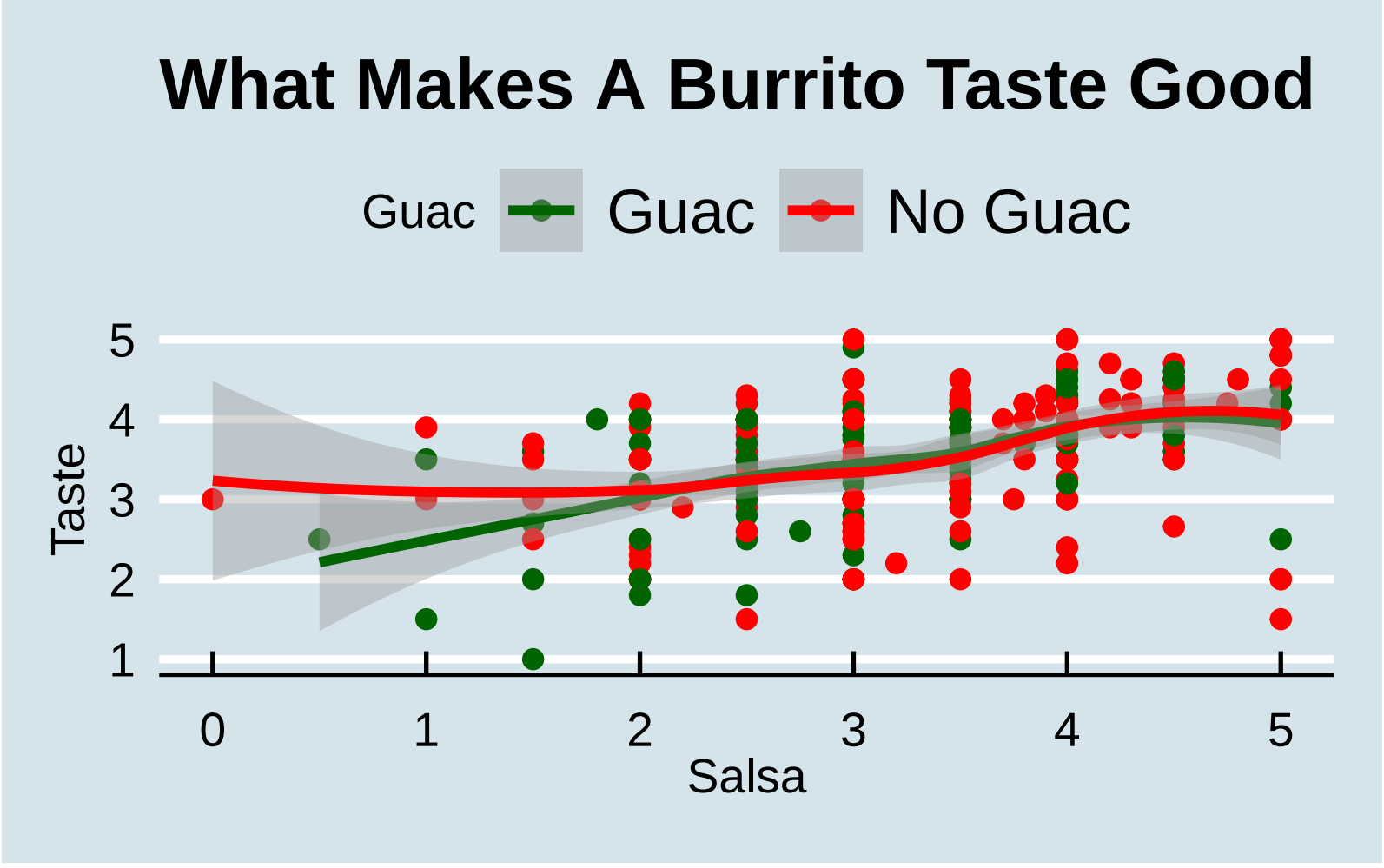
Data Visualization with ggplot

Lingge Li

10/19/2018

Why ggplot

- Beautiful aesthetics
- Flexible and powerful



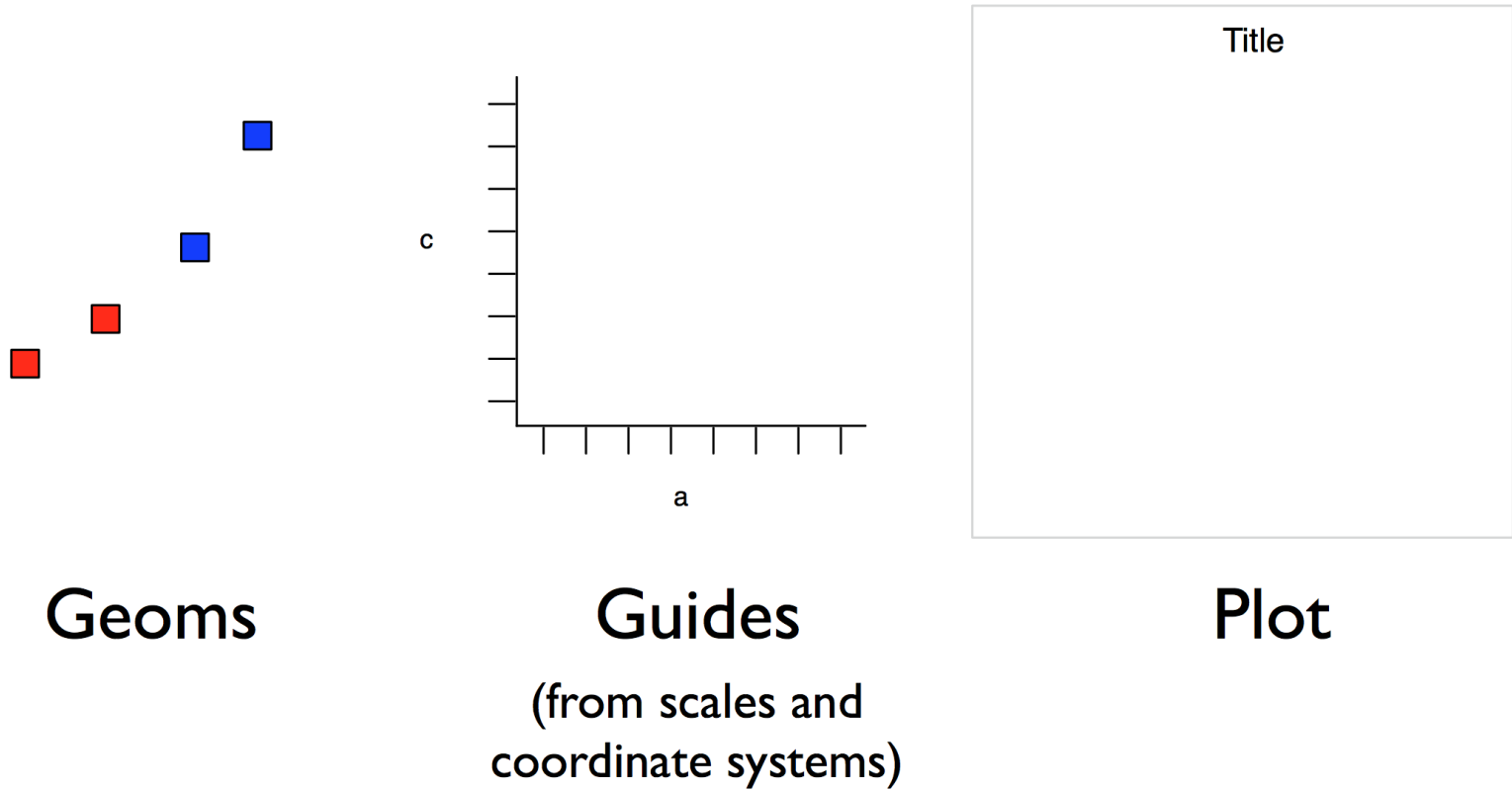
However...

- Syntax slightly compilcated at first glance

```
ggplot(burritos) +
  geom_point(aes(x=Salsa, y=Taste, color=Guac)) +
  geom_smooth(aes(x=Salsa, y=Taste, color=Guac),
              method="loess") +
  scale_colour_manual(values=c("darkgreen", "red")) +
  ggtitle('What Makes A Burrito Taste Good') +
  theme_economist()
```

Layered grammar of graphics

- ggplot2 follows a specific grammar of graphics



Example taken from Hadley Wickham's book

http://vita.had.co.nz/papers/layered-grammar.pdf

How to make a plot

- Geometric objects (geom)
- Aesthetic mapping (aes)
- Statistical transformation (stat)
- Scales and coordinate system

Geoms

- Wide range of geometric objects from points to complex shapes
- geom_point, geom_line, geom_histogram, geom_boxplot...
- Multiple geometric objects on the same plot with +
- https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf

Aesthetics

- Coordinate positions (always needed)
- Colour, fill, shape, size. . .

Data + mapping

- aes() maps a dataframe to geom
- Each geom can have its own mapping

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

Stat

- Plotting distributions needs statistical transformation (count for histogram)
- Variables are often transformed to be meaningful (log for concentration)
- Statistical models can highlight data patterns (regression line)

Scales and coordinate system

- Axis ticks and labels can be customized
- Color scale can also be modified
- Polar coordinates are used for pie charts

Burritos

- Dataset of burritos in San Diego

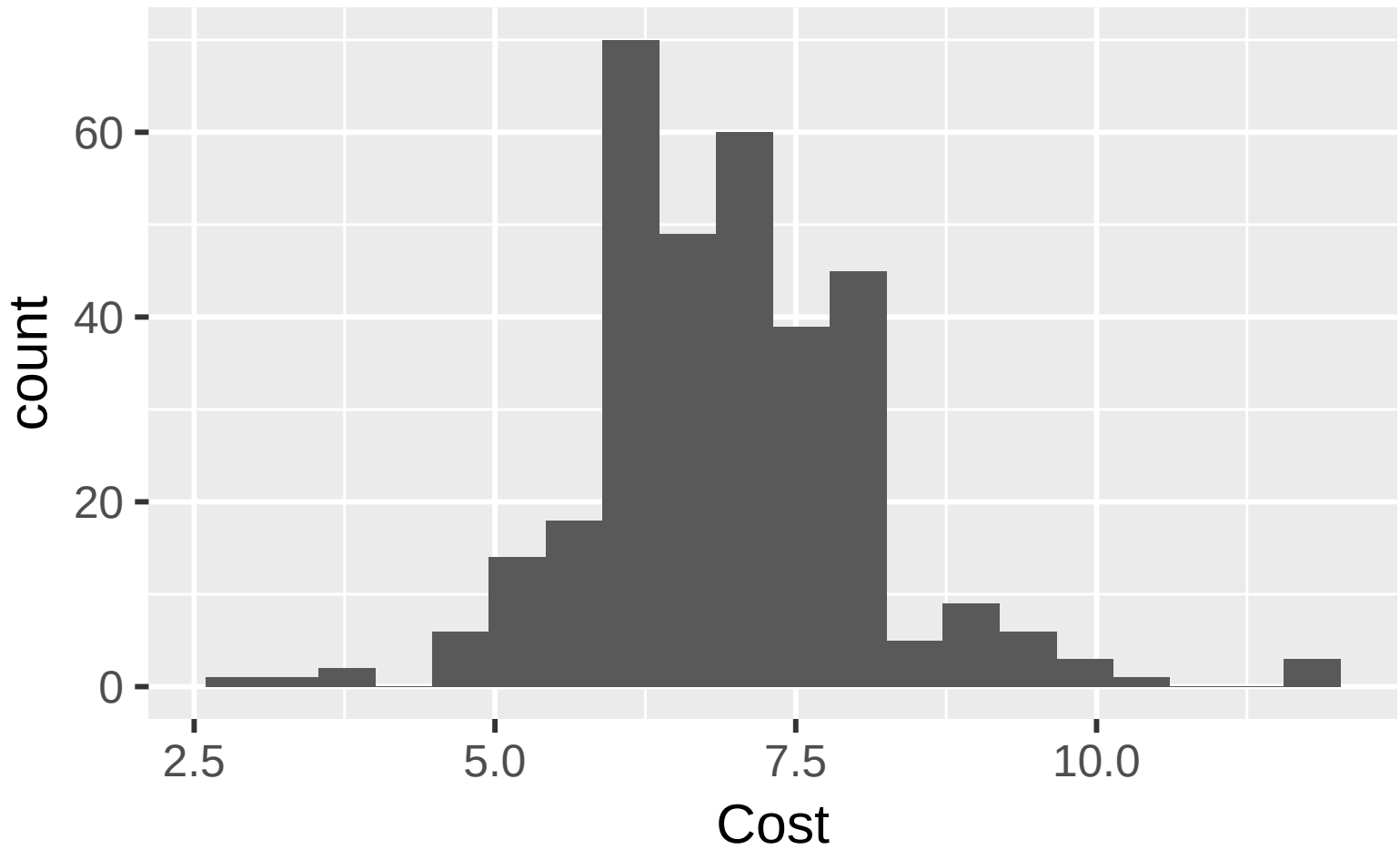
https://srcole.github.io/100burritos/

Read data

```
library(data.table)
library(ggplot2)

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

Univariate plot for continuous variable

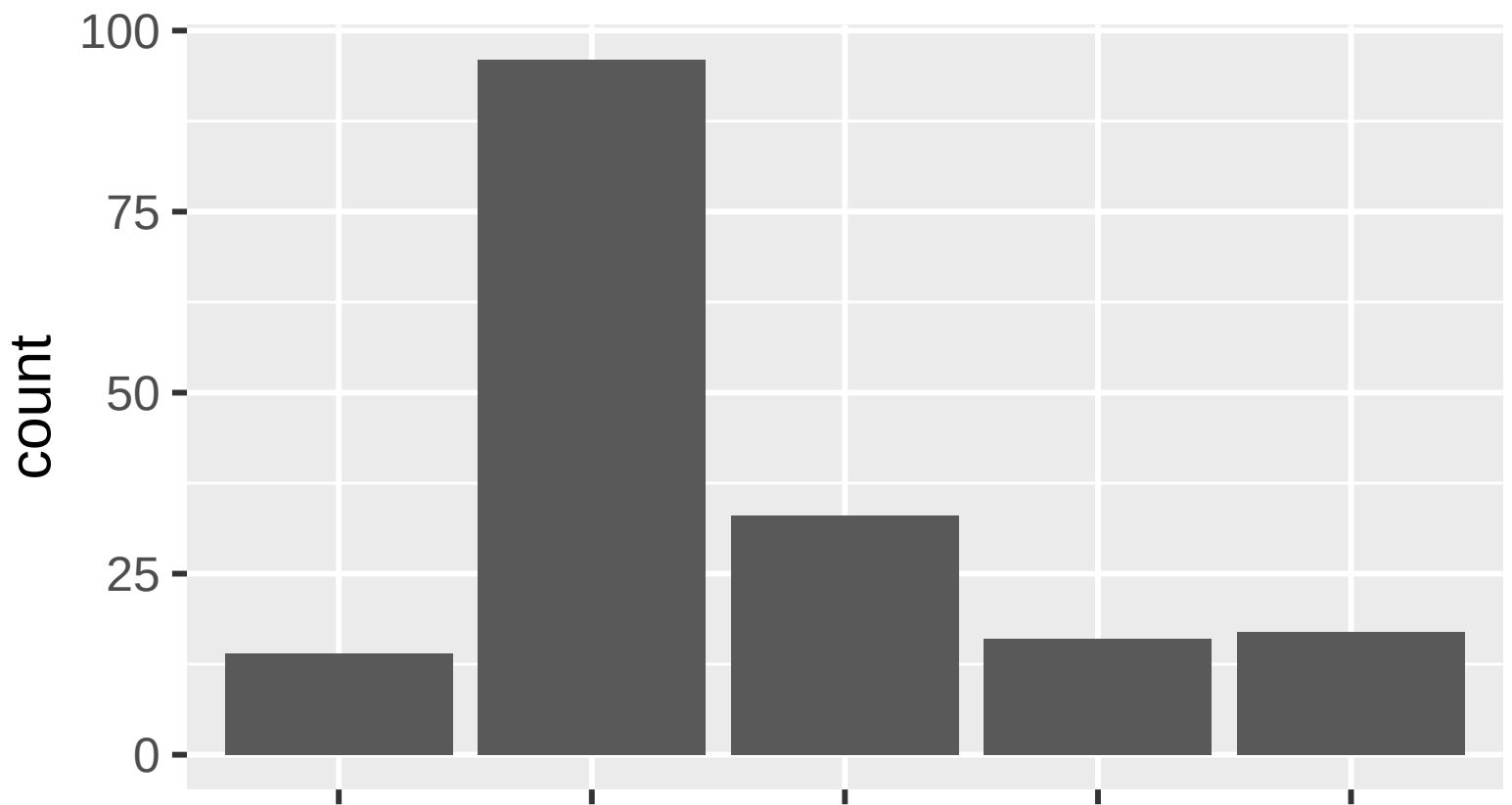
```
ggplot(data=burritos) +
  geom_histogram(aes(x=Cost), bins=20)
```



geom_density

Univariate plot for discrete variable

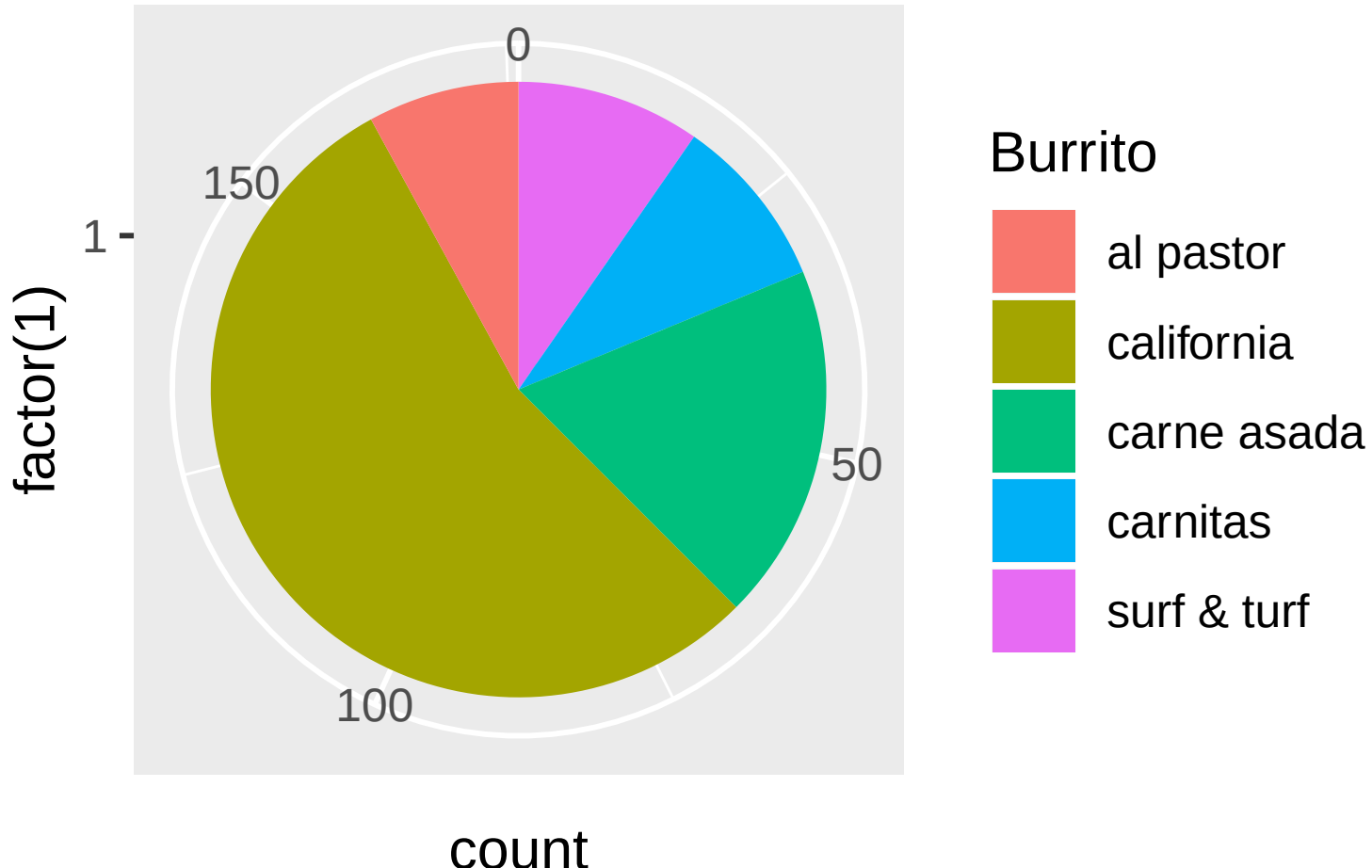
```
burritos$Burrito <- factor(burritos$Burrito)
#summary(burritos$Burrito)
top.burritos <- c('al pastor', 'california', 'carne asada',
temp <- burritos[burritos$Burrito %in% top.burritos, ]

ggplot() + geom_bar(data=temp, aes(x=Burrito))
```



Pie chart

- Pie charts are surprisingly tricky to make
- Stacked bar chart in polar coordinates

```
ggplot(temp, aes(x=factor(1), fill=Burrito)) +
  geom_bar(width=1) + coord_polar(theta='y')
```

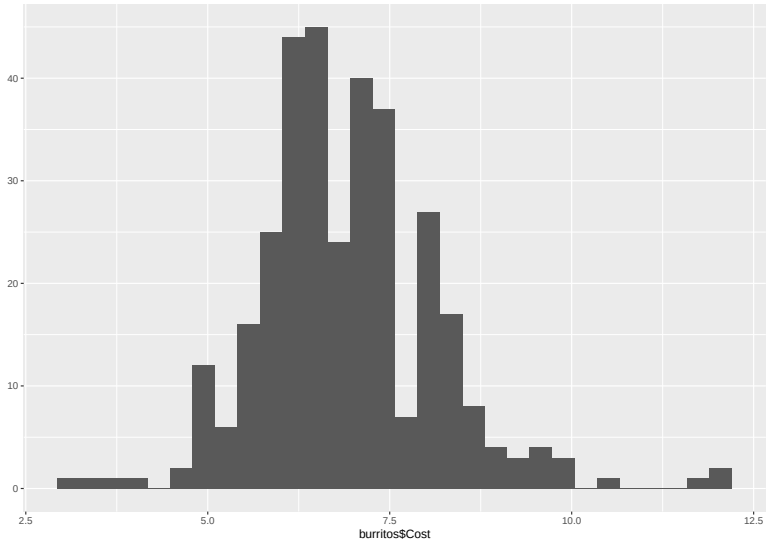


Quick plot

- qplot similar to base plot
- http://docs.ggplot2.org/current/qplot.html

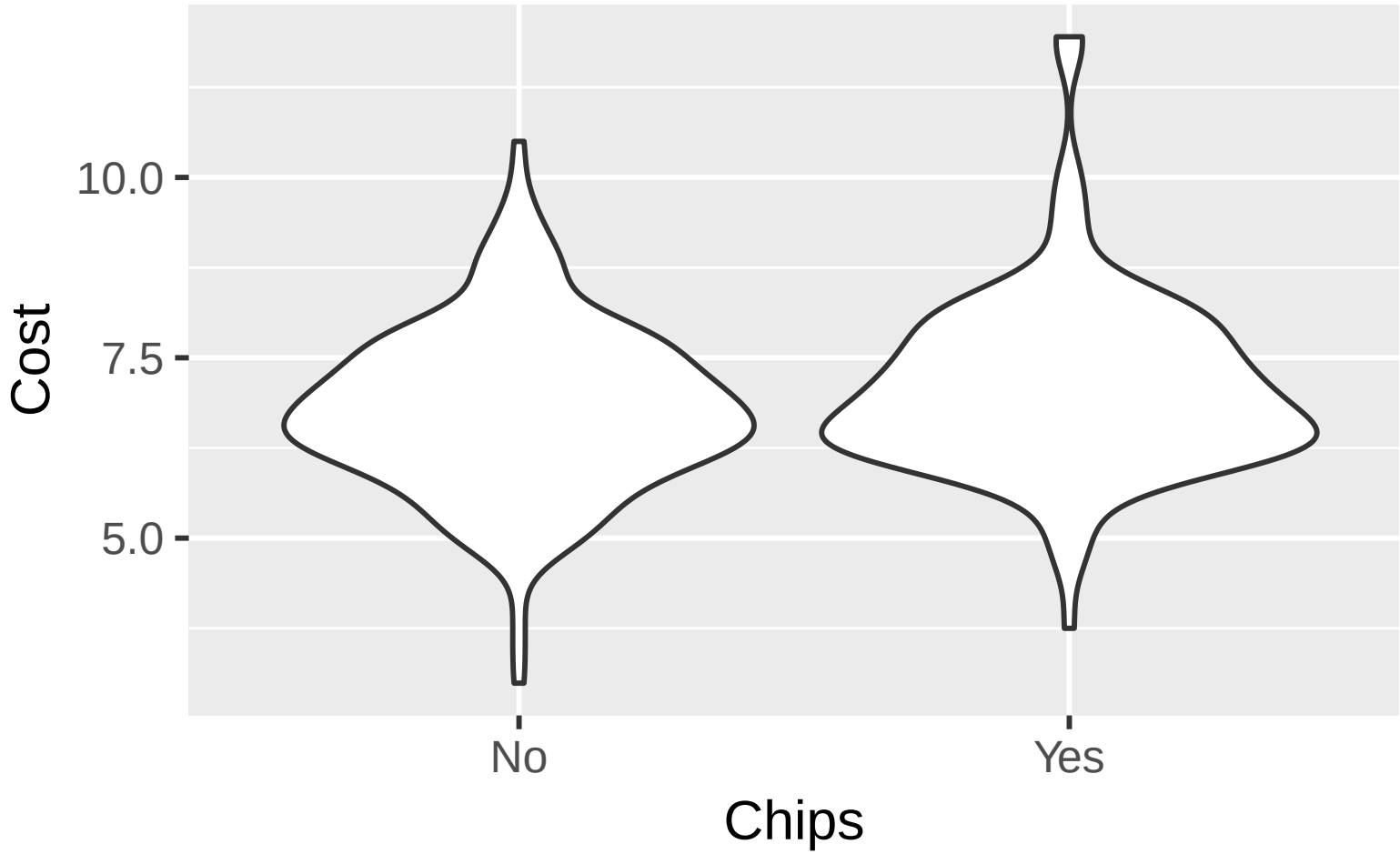
```
x <- rnorm(500)
qplot(burritos$Cost, geom='histogram')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `
```



Bivariate plot

```
ggplot() + geom_violin(data=burritos, aes(x=Chips, y=Cost))
```




```
#ggplot() + geom_boxplot(data=burritos, aes(x=Chips, y=Cost
```

Mapping vs setting

- What is the difference

```
ggplot() +
  geom_violin(data=burritos, aes(x=Chips, y=Cost), fill='re
ggplot() +
  geom_violin(data=burritos, aes(x=Chips, y=Cost, fill=Chi
```

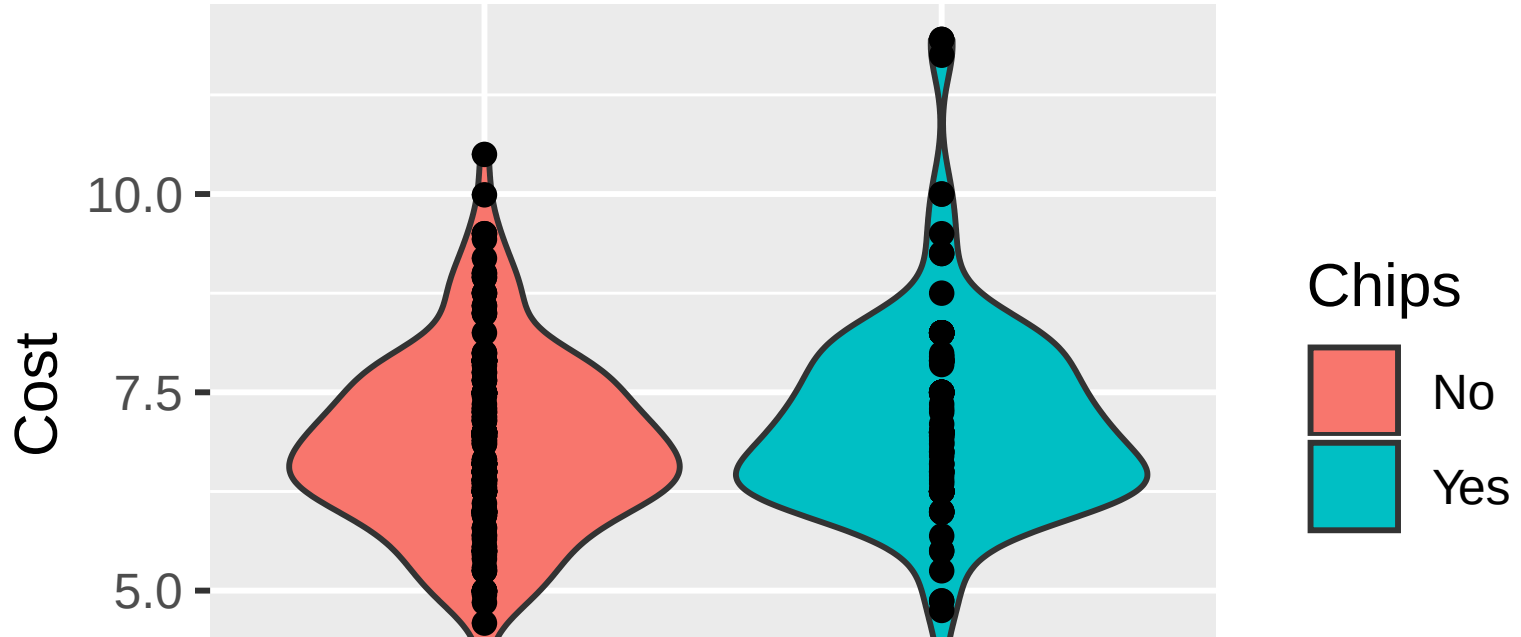
- Try this with histogram

Violin plot with points

- geom_violin + geom_point
- geom_jitter

```
ggplot(data=burritos) + geom_violin(aes(x=Chips, y=Cost, f
  geom_point(aes(x=Chips, y=Cost))
```

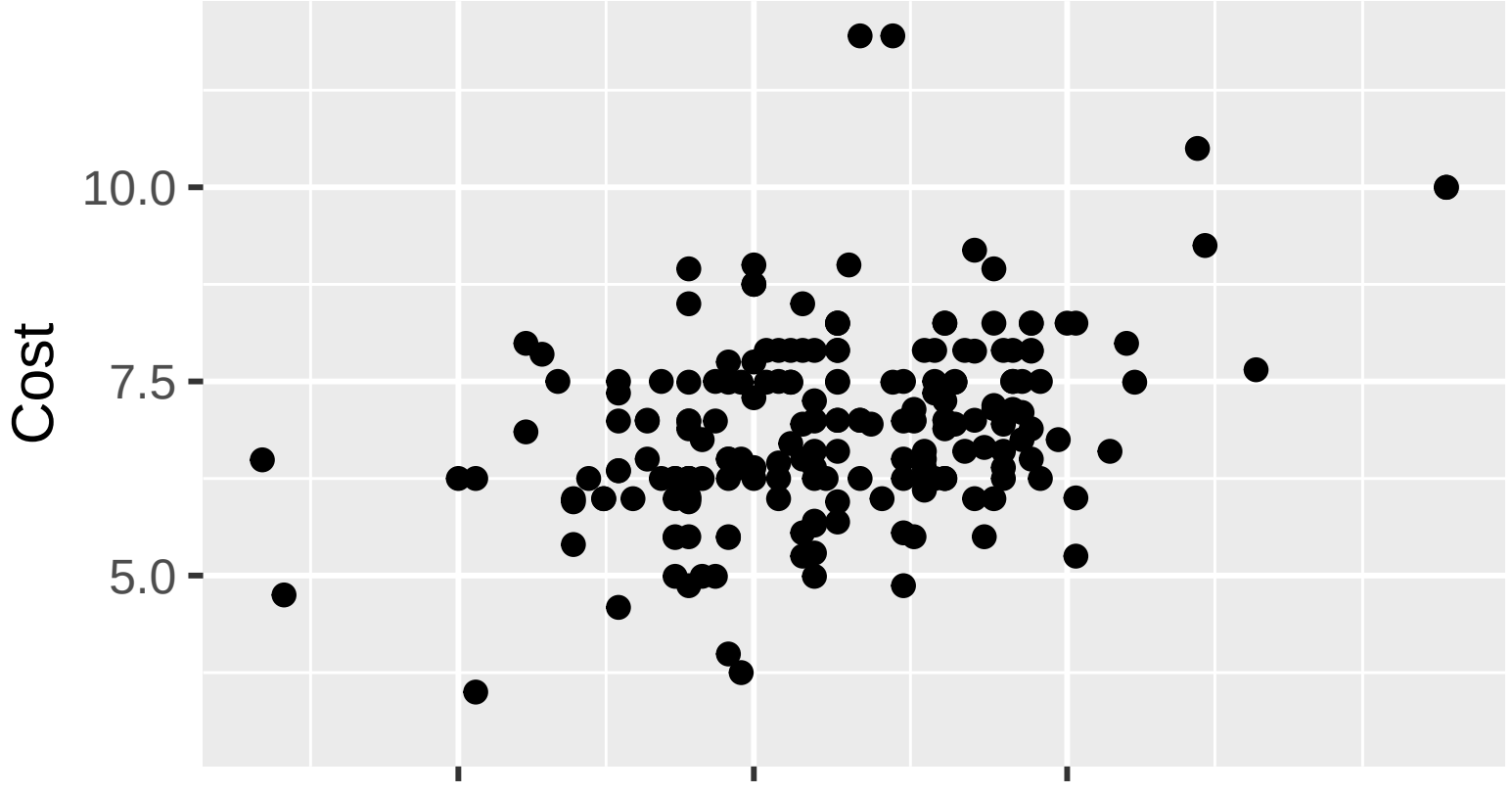
```
## Warning: Removed 7 rows containing non-finite values (st

## Warning: Removed 7 rows containing missing values (geom_
```



Changing the scale

```
#ggplot(burritos) + geom_point(aes(x=Volume, y=Cost))
#ggplot(burritos) + geom_point(aes(x=log10(Volume), y=Cost)
ggplot(burritos) + geom_point(aes(x=Volume, y=Cost)) +
  scale_x_log10()
```

```
## Warning: Removed 135 rows containing missing values (geo
```



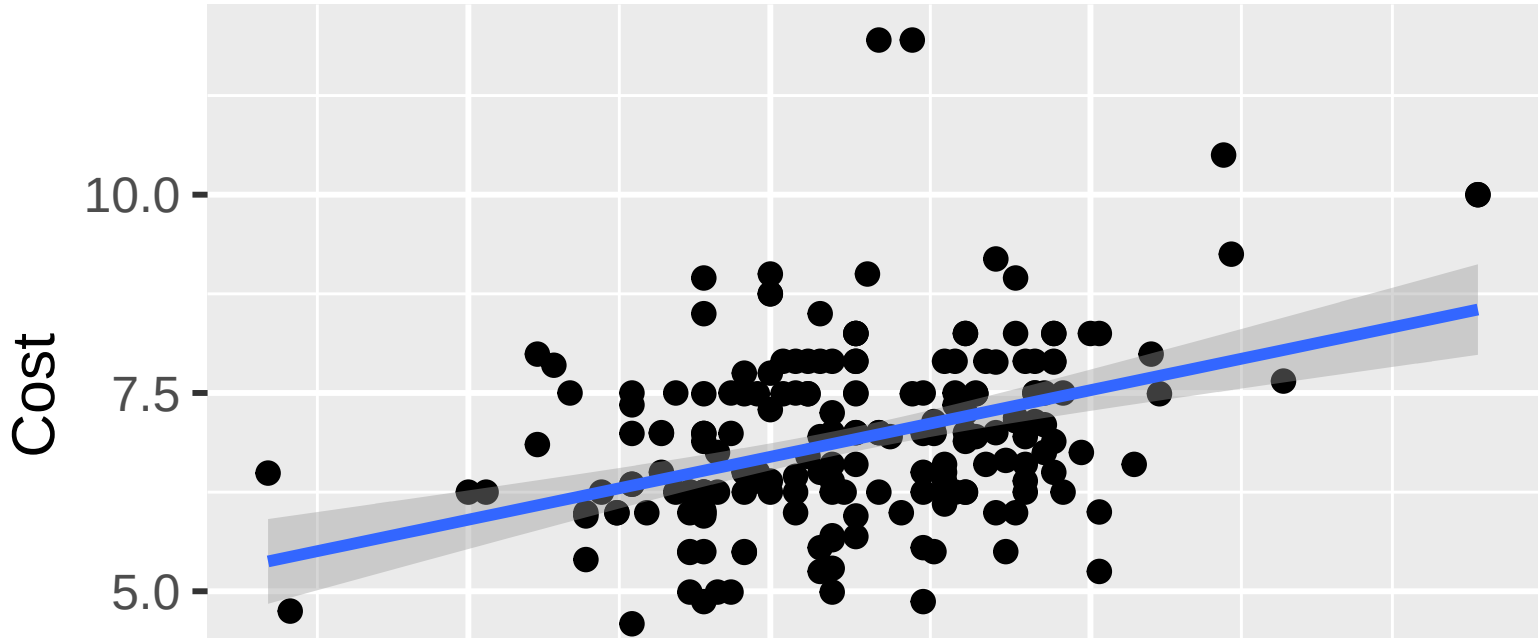
Adding trend

- Linear regression or kernel smoother

```
ggplot(burritos) + geom_point(aes(x=Volume, y=Cost)) +
  geom_smooth(method="lm", aes(x=Volume, y=Cost)) +
  scale_x_log10()
```

```
## Warning: Removed 135 rows containing non-finite values (
```

```
## Warning: Removed 135 rows containing missing values (geo
```



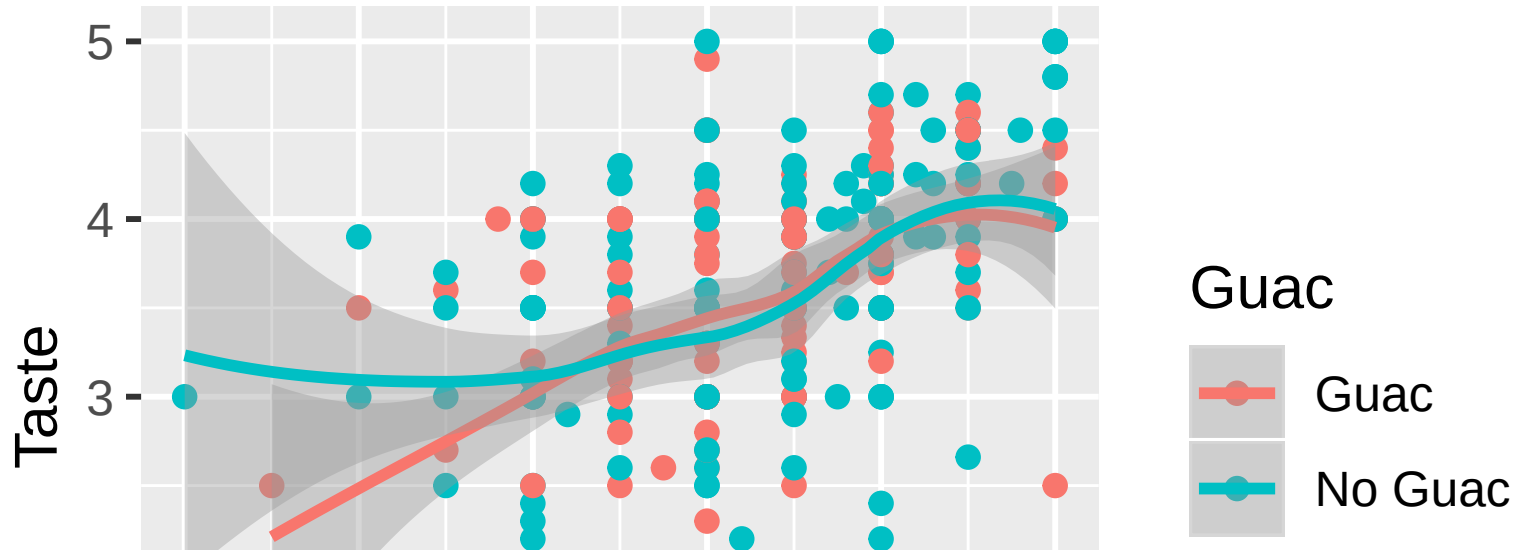
Visualize three variables

- Use color or shape to indicate a third variable

```
#burritos$Guac[is.na(burritos$Guac)] <- 0
#burritos$Guac <- ifelse(burritos$Guac == 1, 'Guac', 'No G
```

```
ggplot(burritos) +
  geom_point(aes(x=Salsa, y=Taste, color=Guac)) +
  geom_smooth(aes(x=Salsa, y=Taste, color=Guac))
```

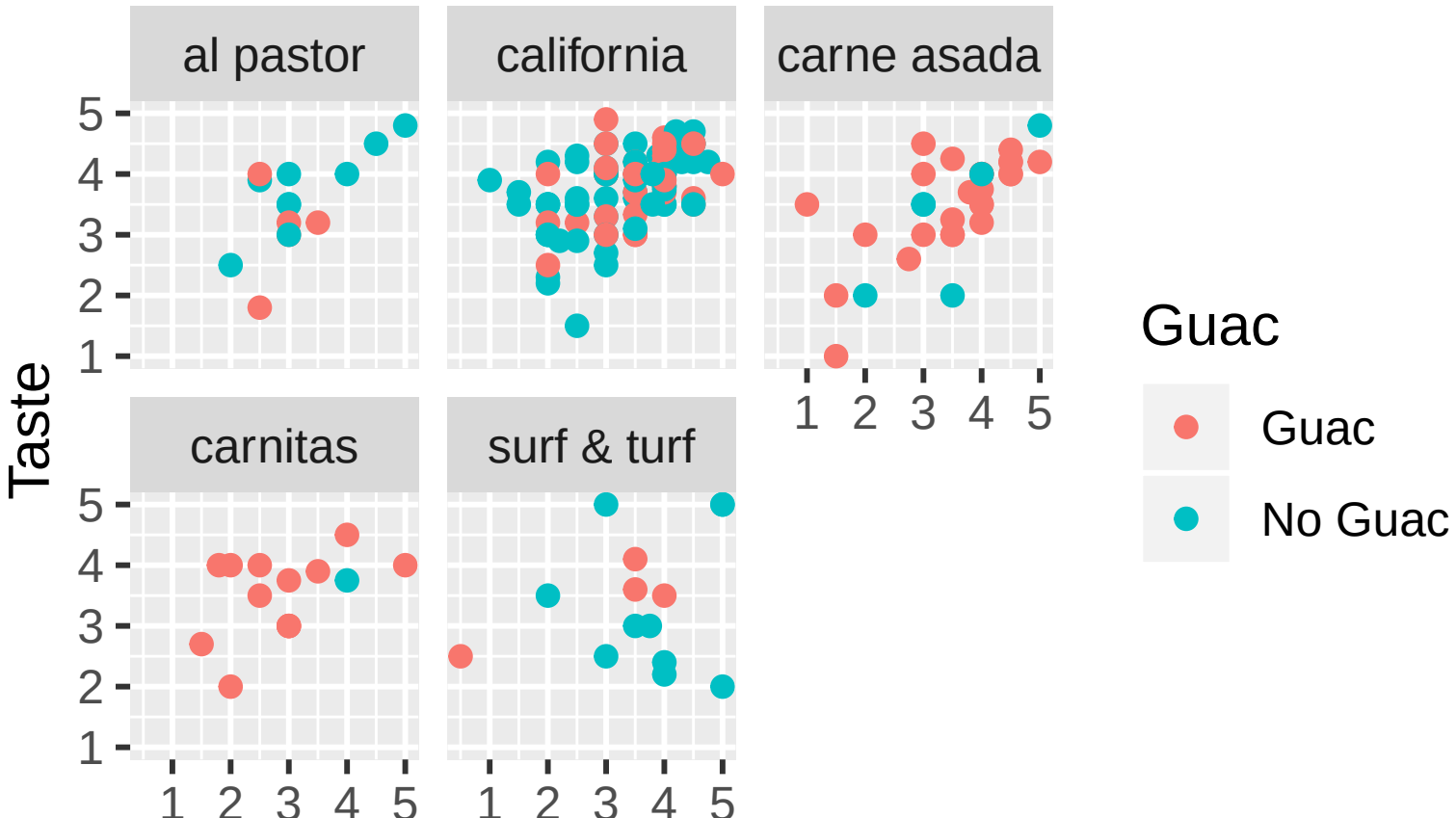
```
## `geom_smooth()` using method = 'loess' and formula 'y ~
```



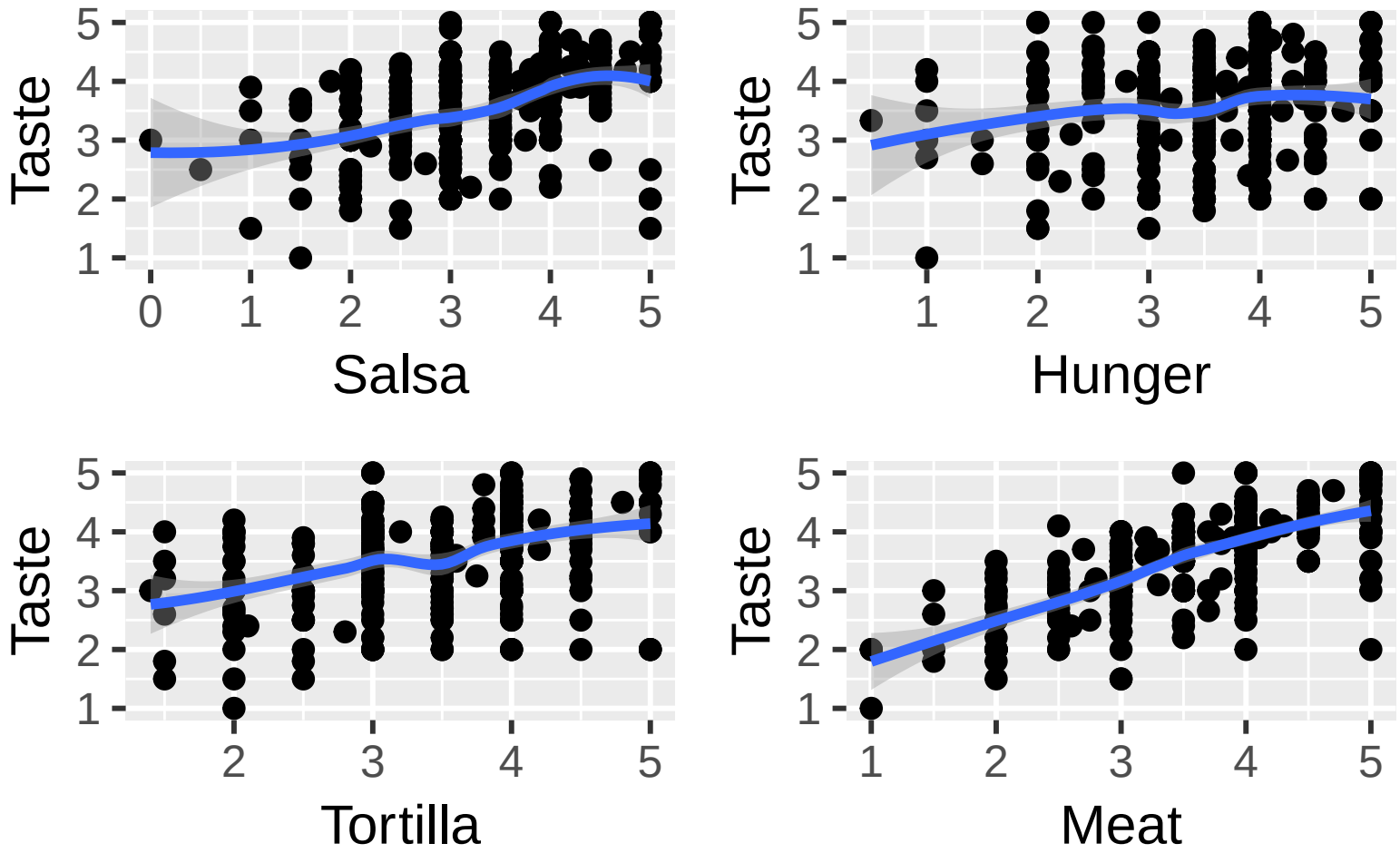
More than three variables

- Similar plots for different categories with facet
- http://www.cookbook-r.com/Graphs/Facets_(ggplot2)/

```
ggplot(temp) +
  geom_point(aes(x=Salsa, y=Taste, color=Guac)) +
  facet_wrap(~Burrito)
```



Multiple plots



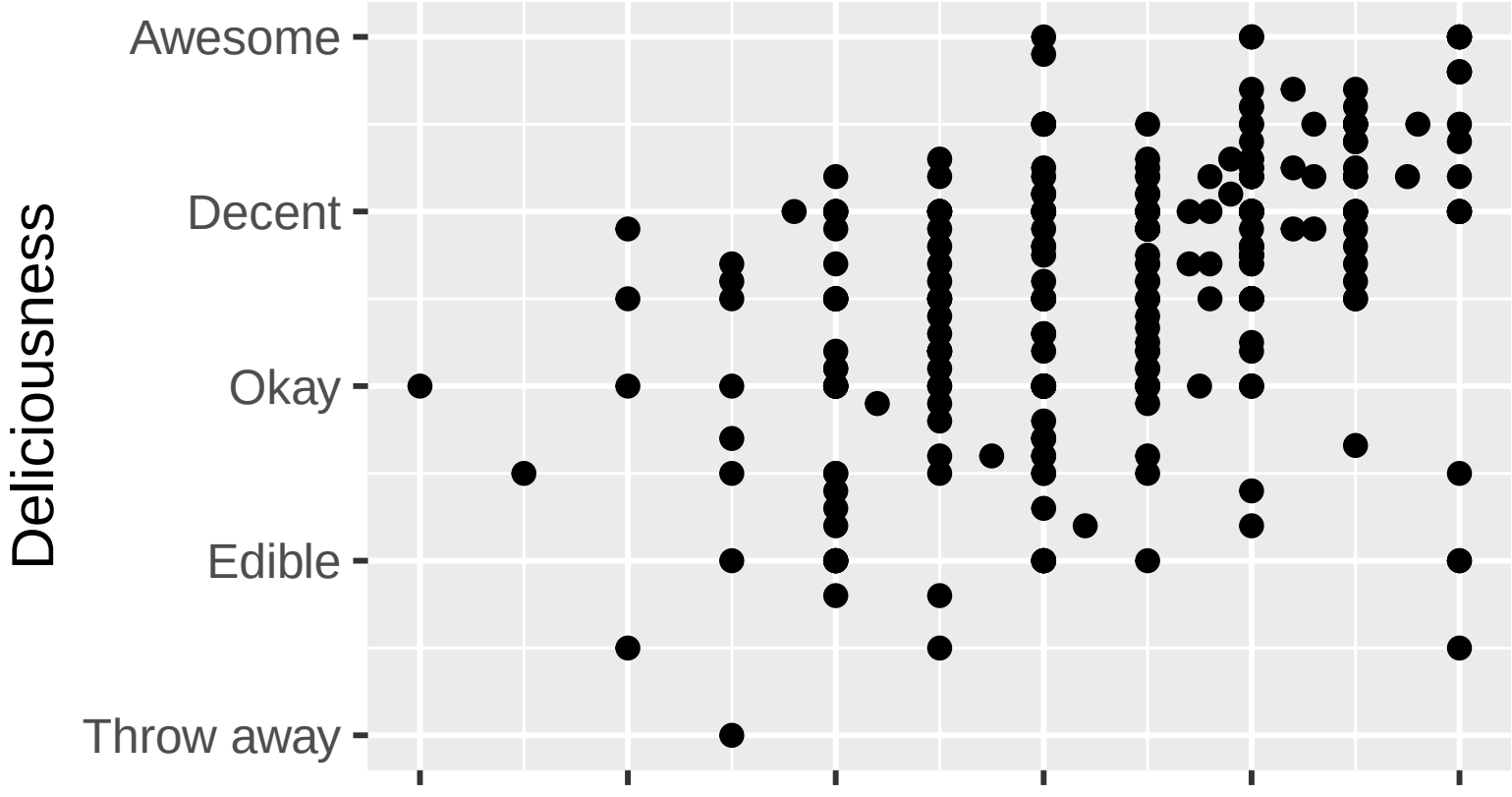
gridExtra

- https://cran.r-project.org/web/packages/gridExtra/vignettes/arrangeGrob.html
- Each plot is an object

```
library(gridExtra)

p1 <- ggplot(burritos) + geom_point(aes(x=Salsa, y=Taste))
  geom_smooth(aes(x=Salsa, y=Taste), method="loess")
p2 <- ggplot(burritos) + geom_point(aes(x=Hunger, y=Taste))
  geom_smooth(aes(x=Hunger, y=Taste), method="loess")
p3 <- ggplot(burritos) + geom_point(aes(x=Tortilla, y=Taste
  geom_smooth(aes(x=Tortilla, y=Taste), method="loess")
p4 <- ggplot(burritos) + geom_point(aes(x=Meat, y=Taste)) +
  geom_smooth(aes(x=Meat, y=Taste), method="loess")

grid.arrange(p1, p2, p3, p4, ncol=2)
```

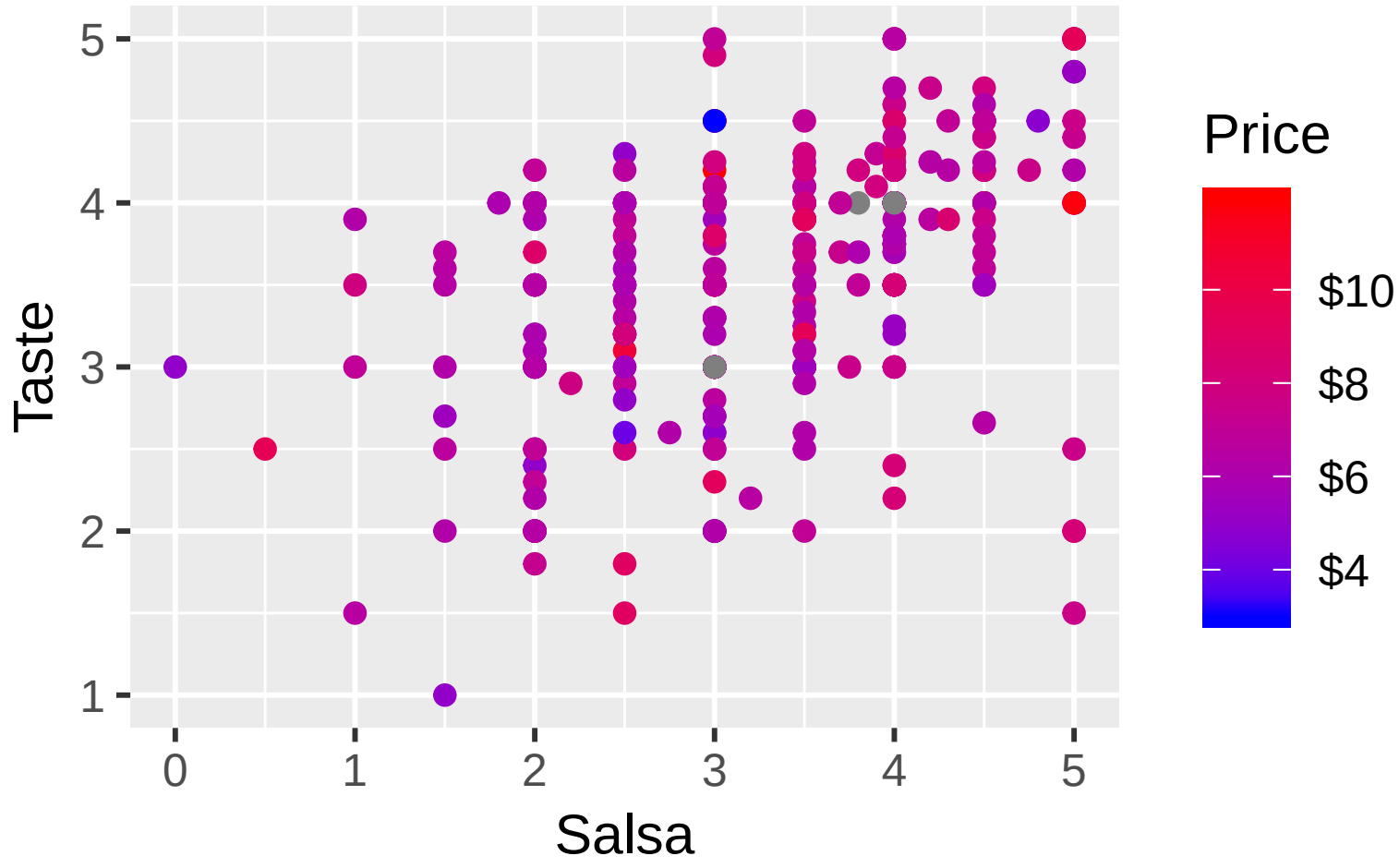
Custom tick marks and labels

```
ggplot(burritos) +
  geom_point(aes(x=Salsa, y=Taste)) +
  scale_y_continuous(breaks=c(1, 2, 3, 4, 5),
                     labels=c('Throw away', 'Edible',
                              'Okay', 'Decent', 'Awesome')) +
  ylab('Deliciousness')
```



Color scale

- scale_color_continuous
- scale_colour_manual

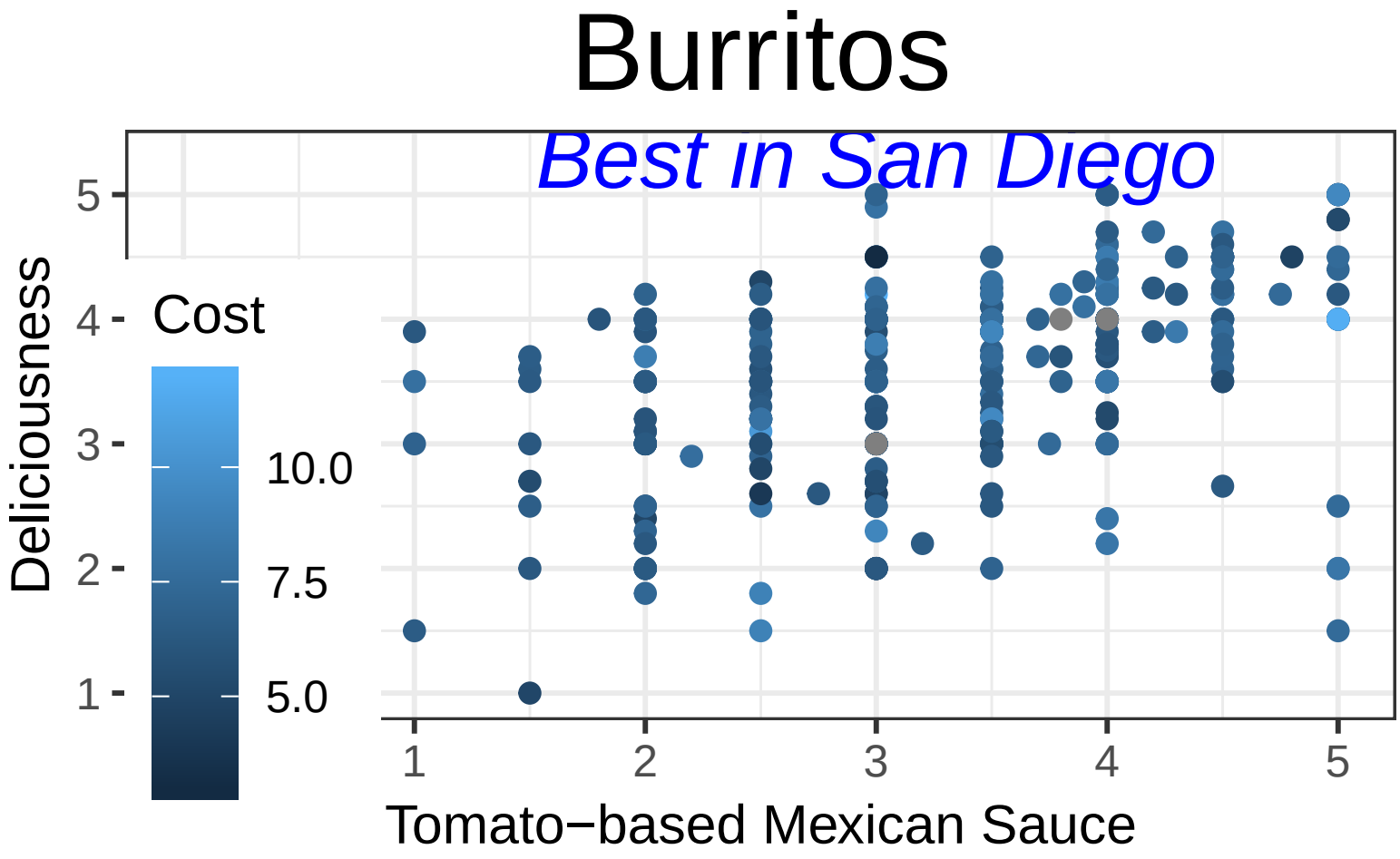


Other details

- annotate
- theme

```
ggplot(burritos) +
  geom_point(aes(x=Salsa, y=Taste, color=Cost)) +
  annotate('text', x=3, y=5.3, label='Best in San Diego',
           fontface='italic', size=6, colour='blue') +
  theme_bw() +
  labs(title='Burritos', x='Tomato-based Mexican Sauce', y=
  theme(plot.title=element_text(size=rel(2), hjust = 0.5))
  theme(legend.position=c(0.1, 0.3))
```

Other details



ggthemes and other extensions

- The Economist and FiveThirtyEight themes
- http://www.ggplot2-exts.org/ggthemes.html

Resources

- Documentation

http://docs.ggplot2.org/current/

- R Cookbook

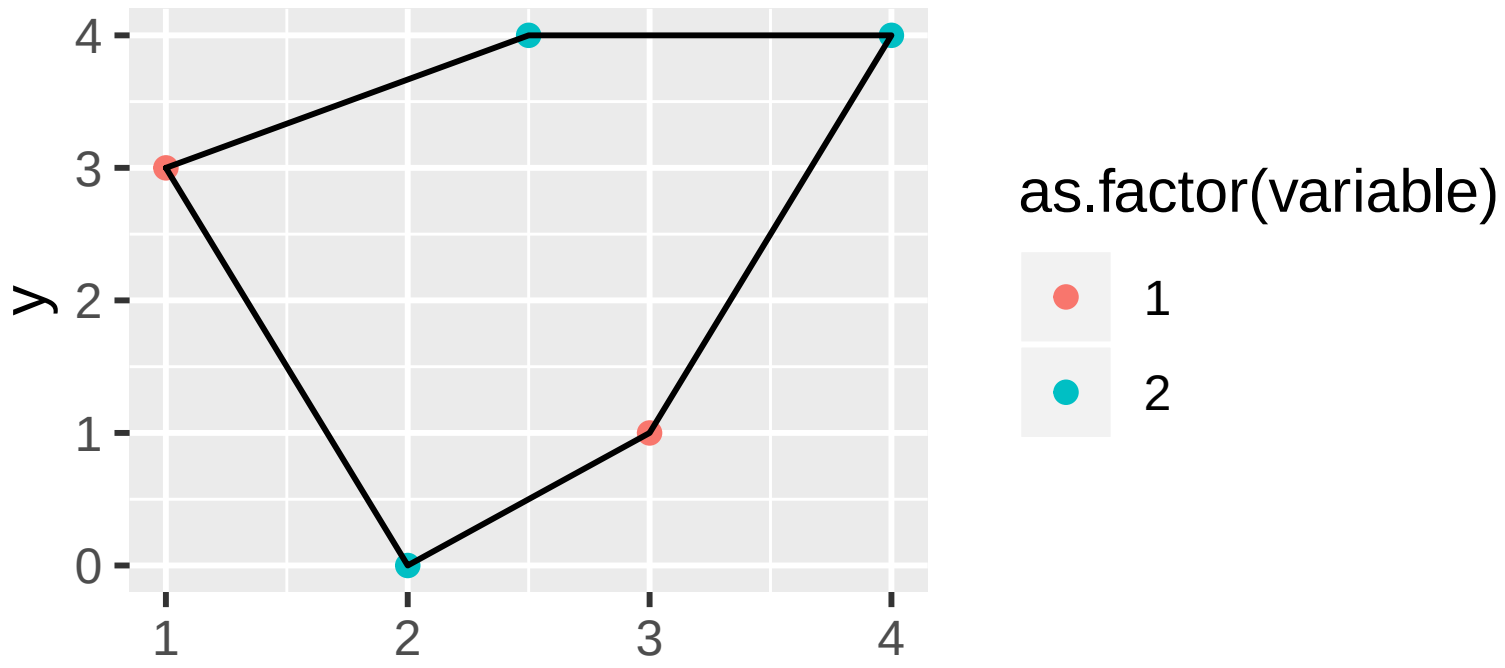
http://www.cookbook-r.com/Graphs/

- Stackoverflow

http://stackoverflow.com/questions/tagged/ggplot2

Polygon

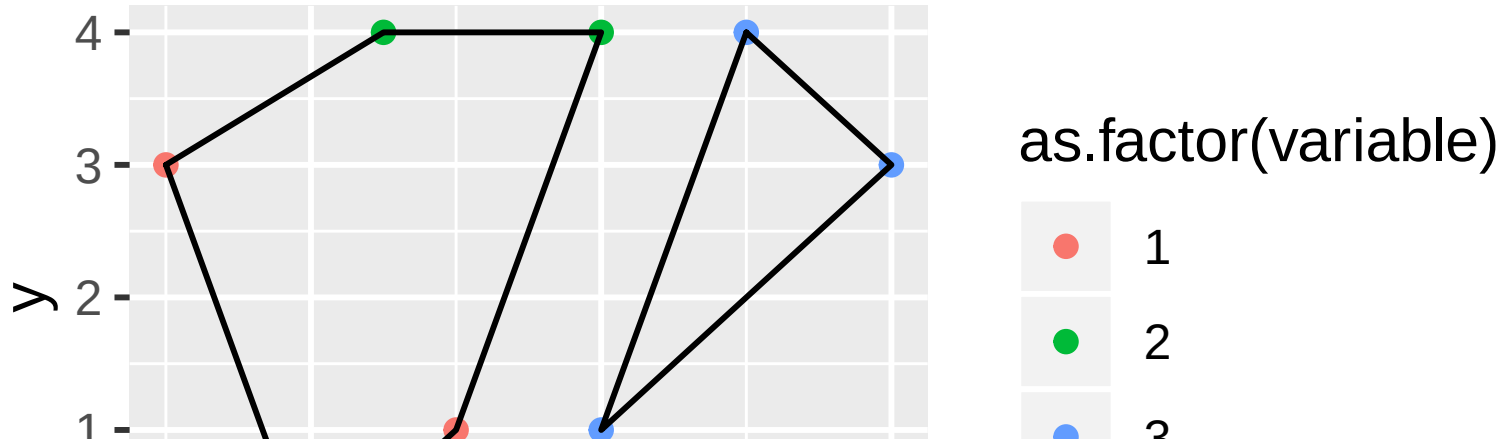
```
x <- c(1, 2, 3, 4, 2.5)
y <- c(3, 0, 1, 4, 4)
variable <- c(1, 2, 1, 2, 2)
example <- data.frame(x, y, variable)

ggplot(data=example) +
  geom_point(aes(x=x, y=y, colour=as.factor(variable))) +
  geom_polygon(aes(x=x, y=y), colour='black', fill=NA)
```



Polygons

```
x <- c(5, 6, 4)
y <- c(4, 3, 1)
variable <- c(3, 3, 3)
triangle <- data.frame(x, y, variable)
```

```
example$group <- 1
triangle$group <- 2
both <- rbind(example, triangle)
ggplot(data=both) +
  geom_point(aes(x=x, y=y, colour=as.factor(variable))) +
  geom_polygon(aes(x=x, y=y, group=group), colour='black',
```



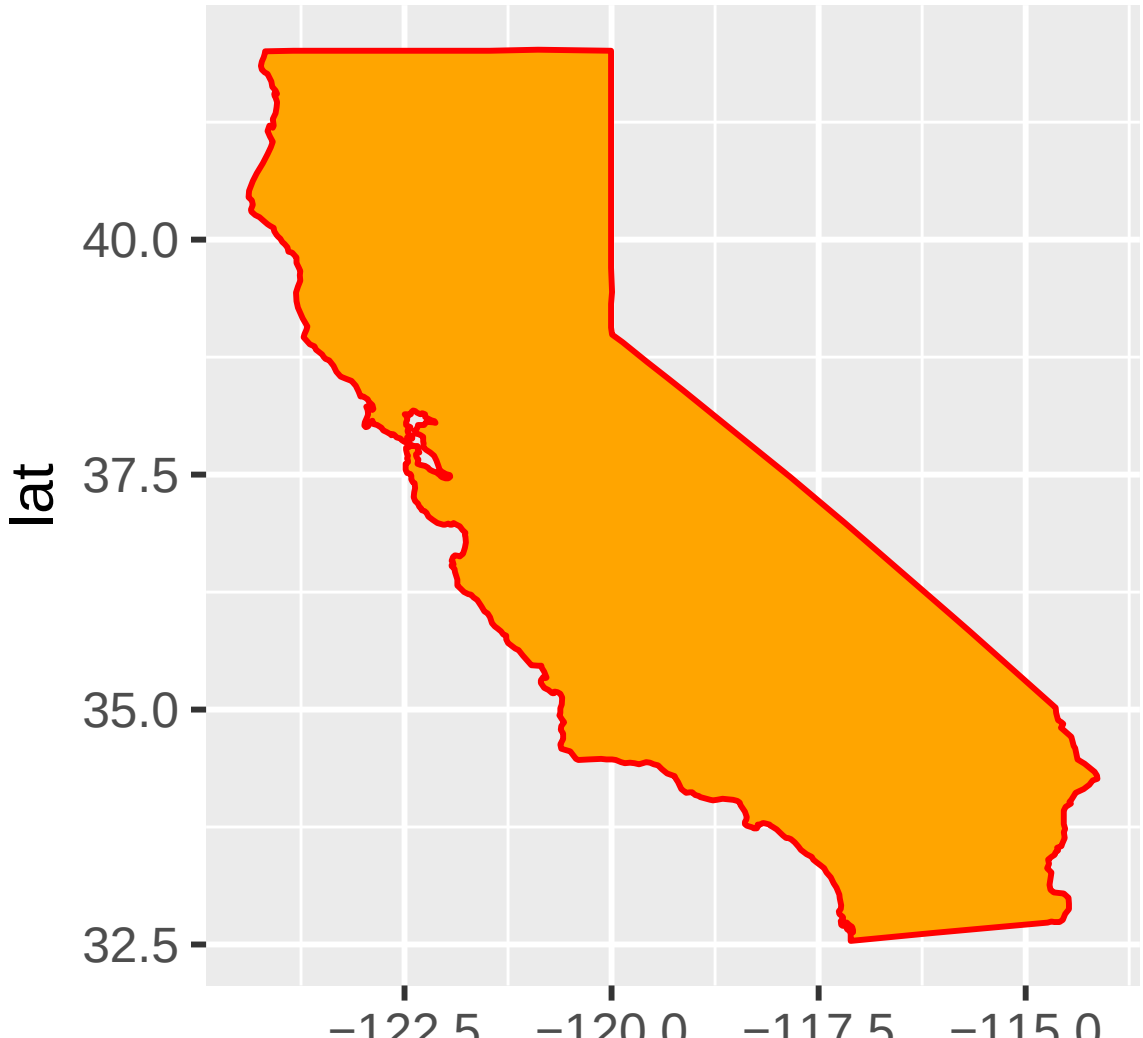
State dataframe

```
library(maps)

states <- map_data('state')
head(states)
```

```
##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

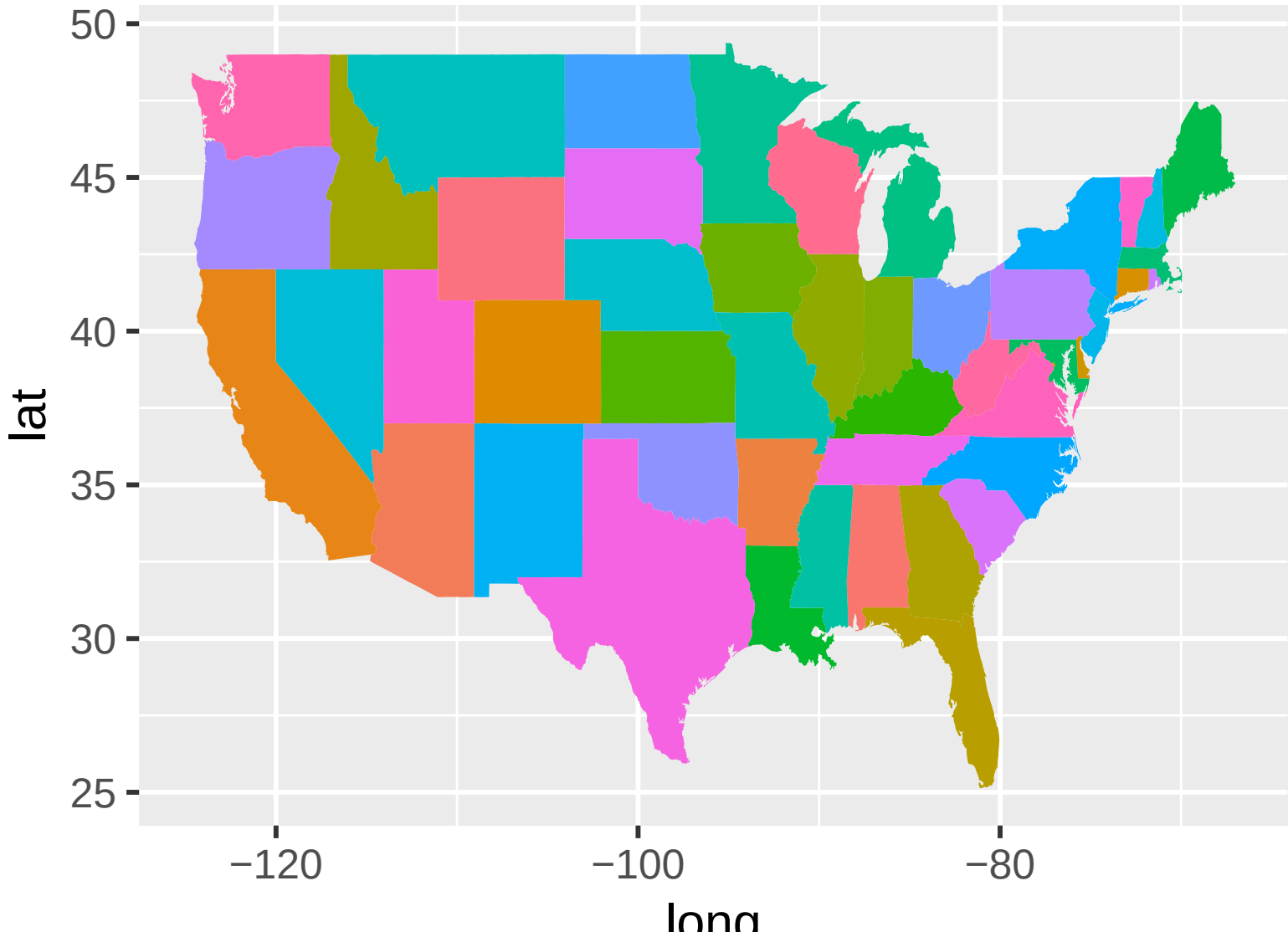
California

```
california <- states[states$region == 'california', ]
ggplot(data=california) +
  geom_polygon(aes(x=long, y=lat), fill='orange', color='re
```



USA

```
ggplot(data=states) +
  geom_polygon(aes(x=long, y=lat, group=group, fill=region)
```



Shapefile

- Packages in R can read and process common shapefiles

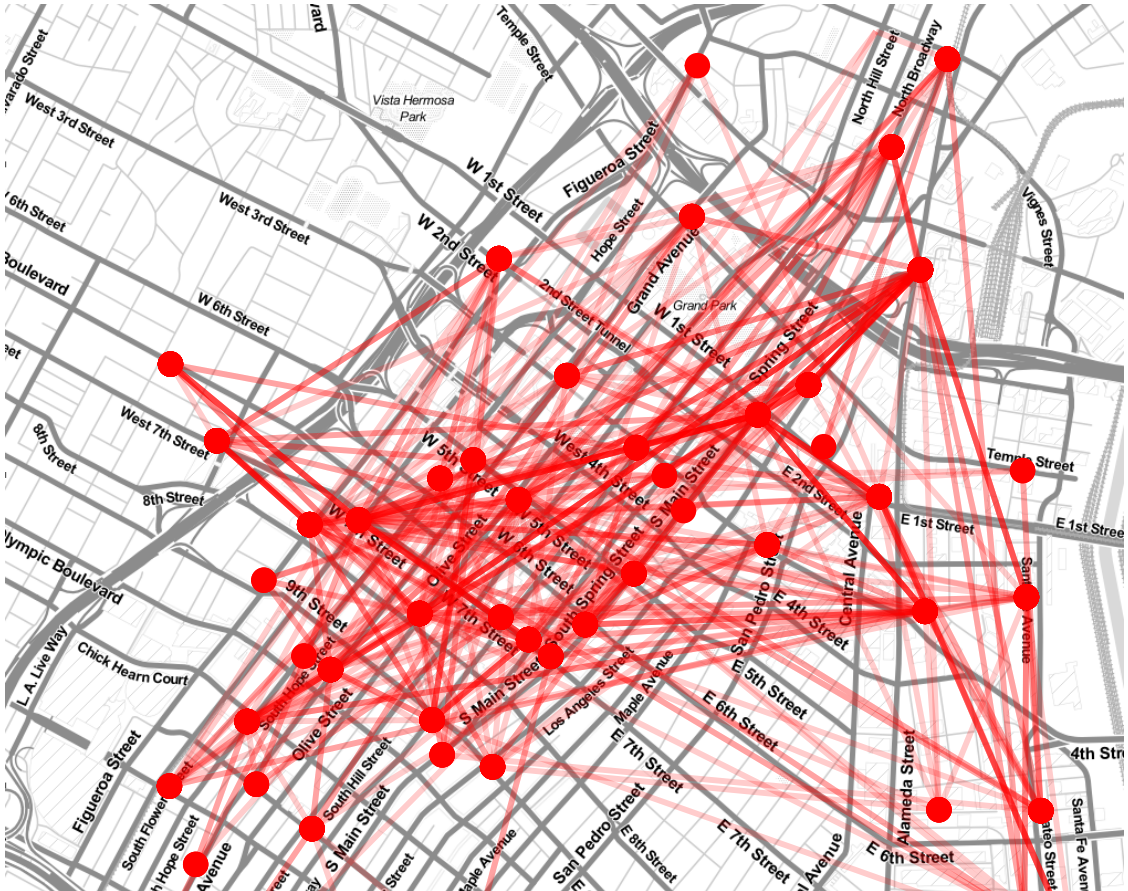
```
CAPD <- readOGR('.', 'CPAD_2016b1_SuperUnits')
CAPD <- spTransform(CAPD,
CRS("+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +uni
proj4string(CAPD) <- CRS("+proj=longlat +ellps=WGS84 +towgs
```

ggmap and Los Angeles Metro Bike Share Trip Data

- https://github.com/dkahle/ggmap
- https://www.kaggle.com/cityofLA/los-angeles-metro-bike-share-trip-data/version/25#metro-bike-share-trip-data.csv

Plot trips as lines

```
qmplot(Starting.Station.Longitude, Starting.Station.Latitud
       data=metro.bike.share.trip.data[1:500, ], maptype="t
  geom_segment(aes(x=Starting.Station.Longitude, xend=Endi
                   y=Starting.Station.Latitude, yend=Ending
                   color='red', alpha=0.2)
```



The end

- https://rstudio.github.io/leaflet/
- https://shiny.rstudio.com/gallery/
- https://plot.ly/r/
- https://rstudio.github.io/r2d3/articles/introduction.html
- Thank you!