Exploratory data analysis & visualization using R

IBIO 851

Sept 15 2016

Suggested reading:

Chapter 2 in **Ecological Models & Data in R** (Bolker)

Goals for today

- Go through tutorial on how to use graphical tools in R
- Provide overview on useful steps for exploratory data analysis
- Practice these skills with a dataset in R

First steps

- Familiarize yourself with the data at hand before any analysis
- This is called exploratory data analysis
  - Involves graphing variables in distributional displays
  - Plotting relationships between variables

First steps

- Get to know your data
  - Distributions (symmetric, normal, skewed)
  - Data quality problems
  - Outliers
  - Correlations and inter-relationships
  - Subsets of interest
  - Suggest functional relationships

Formatting your data

- You can get your data in a specific format by melting your data before you plot
- melt() function is part of the reshape2 package
  - Takes data in wide format & stacks a set of columns into a single column of data

Formatting your data

```
> melt(dat)
Using FactorA, FactorB as id variables
   FactorA FactorB variable       value
1      Low     Low   Group1 -1.16163338
2   Medium     Low   Group1 -0.59914783
3     High     Low   Group1  0.84207974
4      Low  Medium   Group1  1.62255690
5   Medium  Medium   Group1 -0.34507455
6     High  Medium   Group1  1.60250438


...
36    High    High   Group4  0.23407257
```

Formatting your data

	FactorA	FactorB	Group1	Group2
1	Low	Very Low	6.851828	3.061329
2	Medium	Very Low	7.352169	1.303077
3	High	Very Low	6.918091	2.477875
4	Low	Low	7.402351	2.450527
5	Medium	Low	6.928385	4.334323
6	High	Low	7.400626	3.074158
7	Low	Medium	8.312145	5.725185
8	Medium	Medium	8.251806	4.384492
9	High	Medium	8.339398	3.443789
10	Low	High	5.127386	2.868952
11	Medium	High	8.561181	3.616898
12	High	High	6.993838	3.450634
13	Low	Very High	7.880877	2.950622
14	Medium	Very High	9.439892	3.220295
15	High	Very High	8.799447	3.106060

Formatting your data

```
> melt(dat, id.vars = "FactorB", measure.vars =
c("Group1", "Group2"))
     FactorB variable    value
1   Very Low   Group1 6.851828
2   Very Low   Group1 7.352169
3   Very Low   Group1 6.918091
4        Low   Group1 7.402351
5        Low   Group1 6.928385
6        Low   Group1 7.400626
...
30 Very High   Group2 3.106060
```

Summary statistics

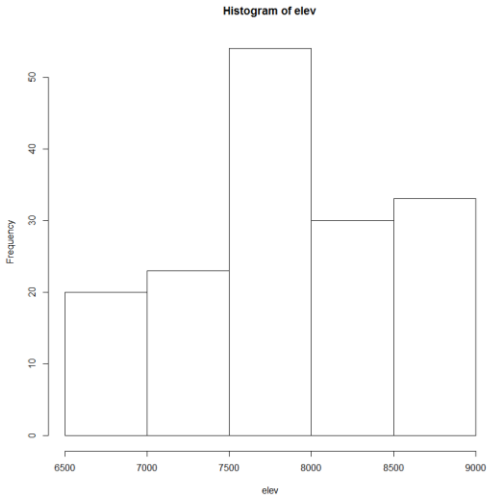
- Non-visual
- Sample statistics of vector X
  - Mean: $\mu = \sum_i X_i / n$
  - Mode: most common value in X
  - Median: $\mathbf{X}$=sort(X), median = $\mathbf{X}_{n/2}$ (half below, half above)
  - Quartiles of sorted $\mathbf{X}$: Q1 value = $\mathbf{X}_{0.25n}$, Q3 value = $\mathbf{X}_{0.75\,n}$
    - Interquartile range: value(Q3) - value(Q1)
    - Range: max(X) - min(X) = $\mathbf{X}_n - \mathbf{X}_1$
  - Variance: $\sigma^2 = \sum_i (X_i - \mu)^2 / n$

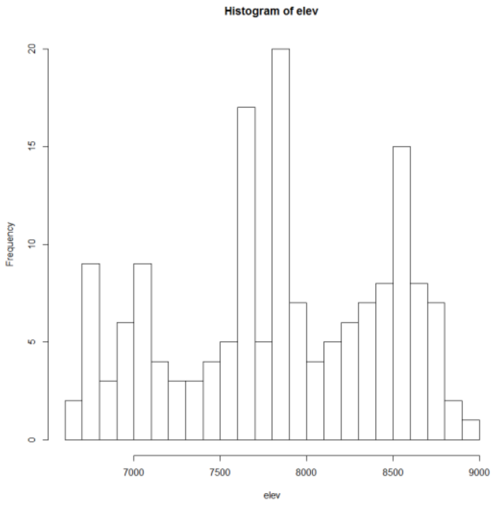
Summary statistics

- Can get most of these stats with one command in R:
  - summary(*variable*)
  - Returns the minimum & maximum values, the 1st & 3rd quartiles, the mean & median of a given vector

Single variable visualization

- Histogram:
  - Shows center, variability, skewness, modality, outliers, or strange patterns
  - Bins matter
  - Beware of real zeros
  - Most common way to examine distribution of a quantitative (continuous) variable
  - hist() command in R





Single variable visualization

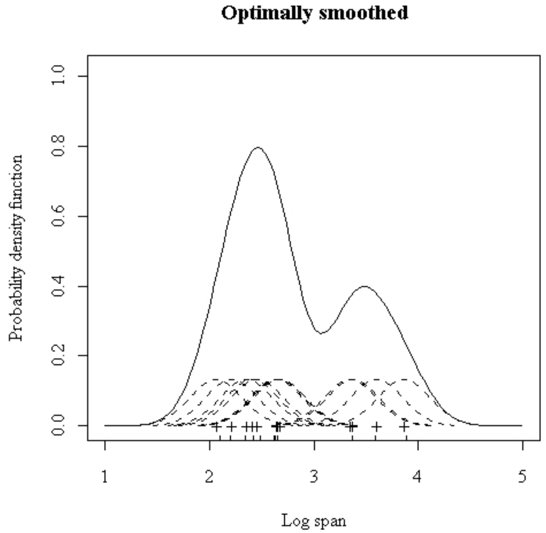
- For small data sets, histograms can be misleading
  - Small changes in the data or bins can deceive
- For large data sets, histograms can be quite effective at illustrating general properties of the distribution
- Histograms effectively only work with 1 variable at a time
  - But 'small multiples' can be effective

Single variable visualization

- Say you have a tree height variable measured in 3 different study populations
- You can specifically plot the distribution according to each population with the command:
  - Hist(height[pop==1] OR
  - Hist(height[pop==“IL”]
  - Double equal sign is used for specifying the population (no space between them)

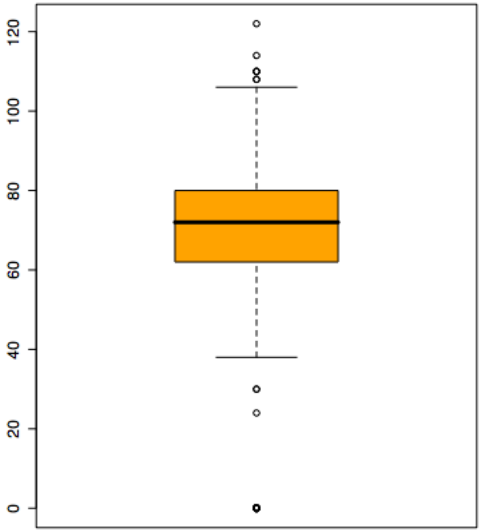
Single variable visualization

- Smoothed histograms
- Appropriate for density estimates
- Kernel estimates smooth out the contribution of each datapoint over a local neighborhood of that point



Single variable visualization

- Boxplots
- Shows a lot of information about a variable in one plot
  - Median
  - IQR
  - Outliers
  - Range
  - Skewness

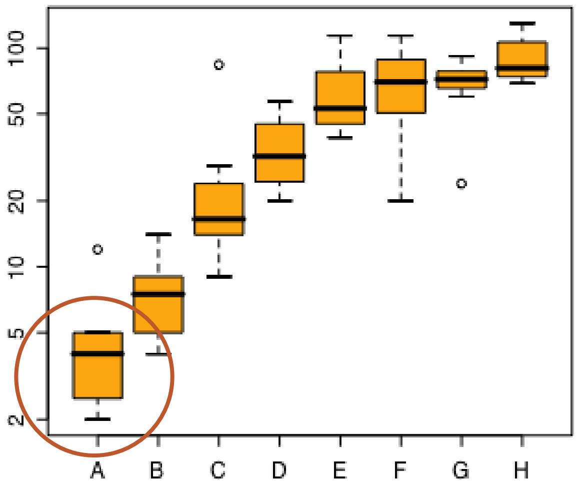


- Drawbacks
  - Overplotting
  - Hard to tell distributional shape

Two variables: 1 categorical

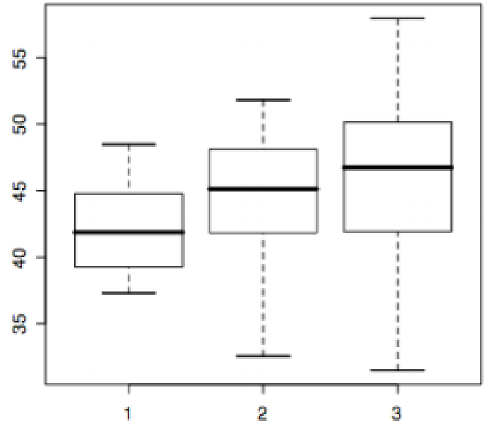
- Side by side boxplots are effective in showing differences in a quantitative variable across factor levels
  - E.g. measuring potency of various orchard sprays in repelling insect pests

Most effective spray



Side-by-side boxplots

- boxplot(*variable*) command in R
  - Boxplot(height[pop==1], height[pop==2], height[pop==3])



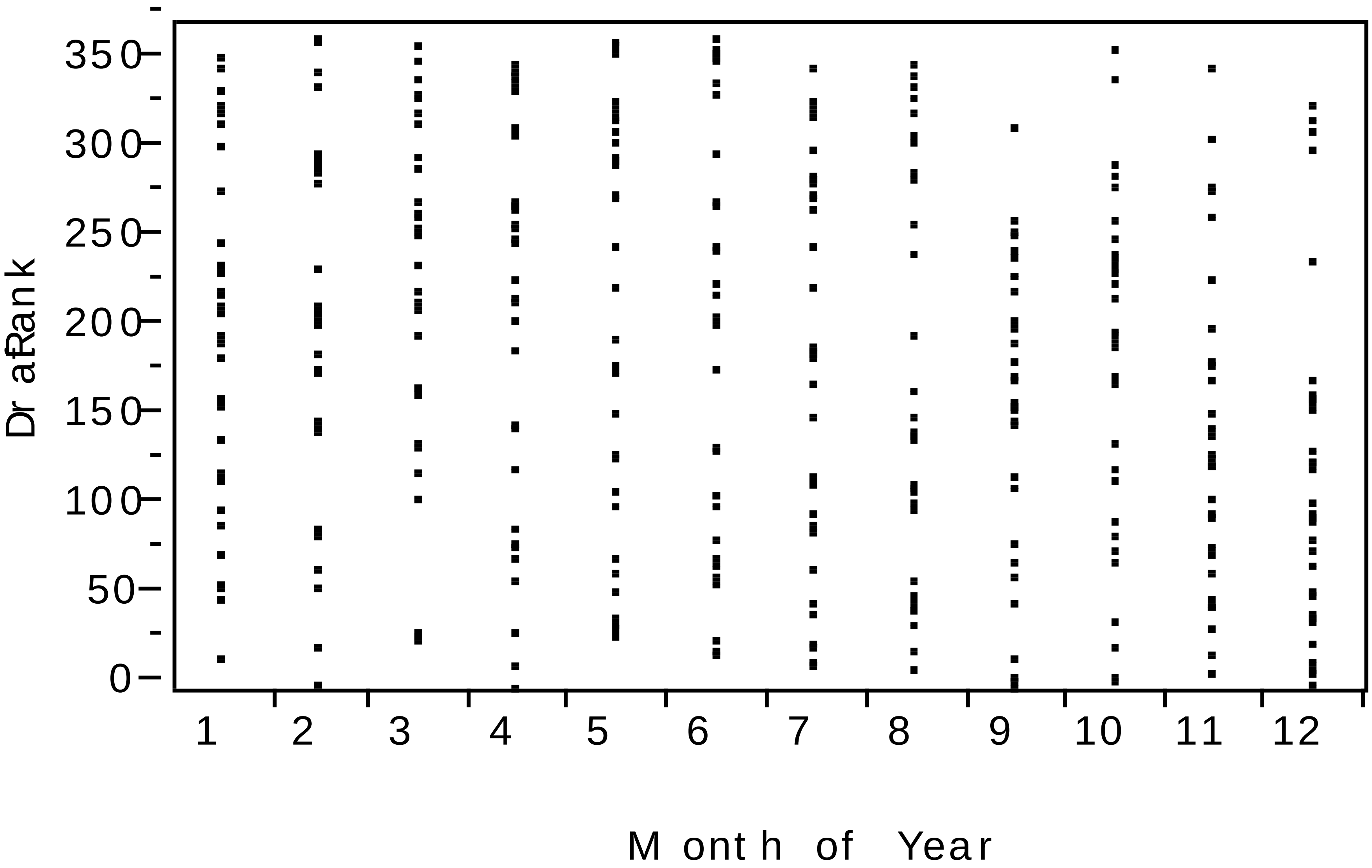
Cool short-cut: you can also see all boxplots for the subclasses of a variable by doing: boxplot(height~pop)

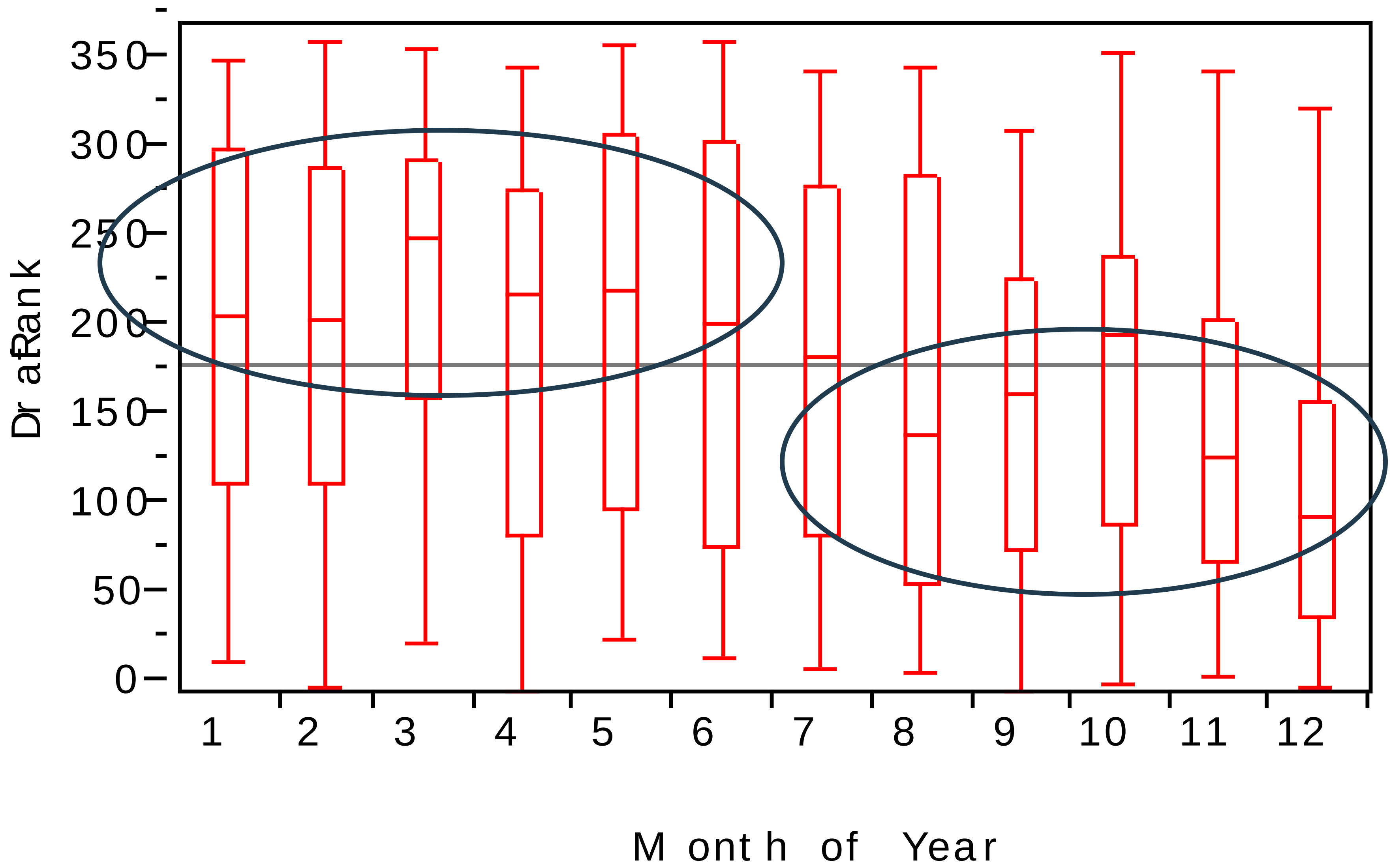
Side-by-side boxplots

- Example: Vietnam draft lottery
  - In 1970, US govt drafted men for service via a random lottery
  - Paper slips containing all dates in January placed in a box & mixed
  - This was repeated for all months until 366 dates were mixed in
  - Dates were successfully drawn without replacement
  - First date drawn was ranked 1 (i.e. first called to service), etc.

Side-by-side boxplots

- People began to complain that the randomization system was not fair
- Birth dates later in the year were believed to have lower lottery numbers
- What do the data say?


Draft Rank
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
Month of Year

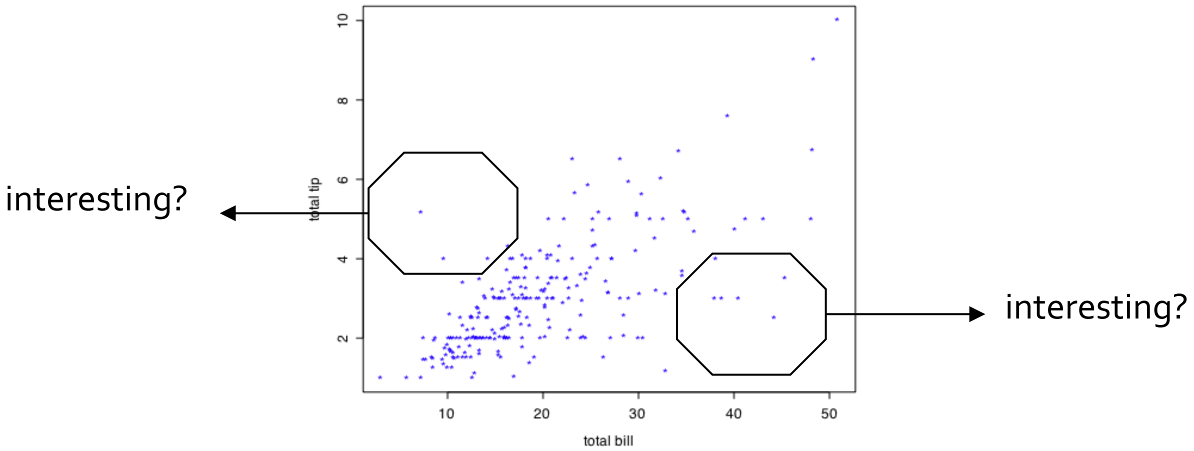

Draft Rank
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
Month of Year

Two continuous variables

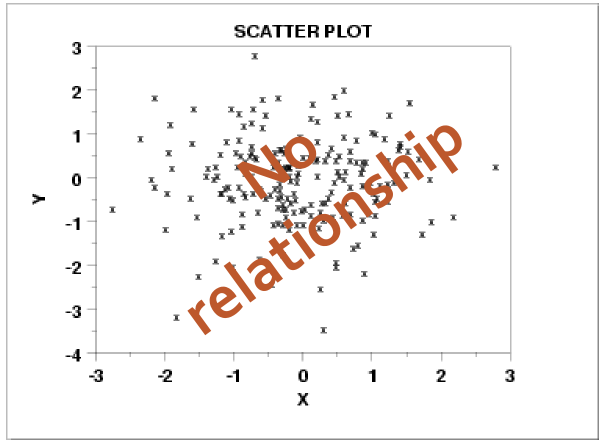
- Scatterplots are the standard graph for visualizing relationship between 2 quantitative variables
- plot(x,y) command in R
  - 2 numeric vectors required as arguments x & y

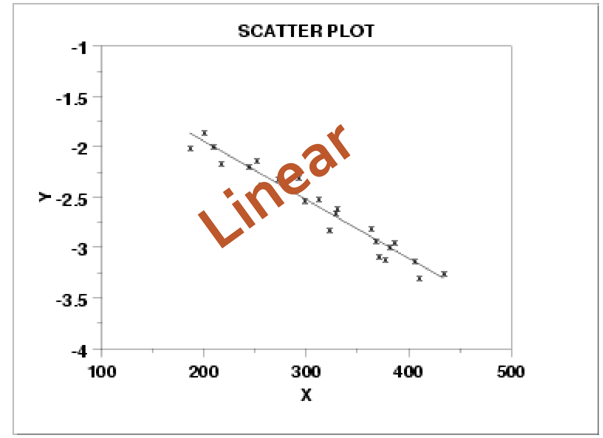
Two continuous variables

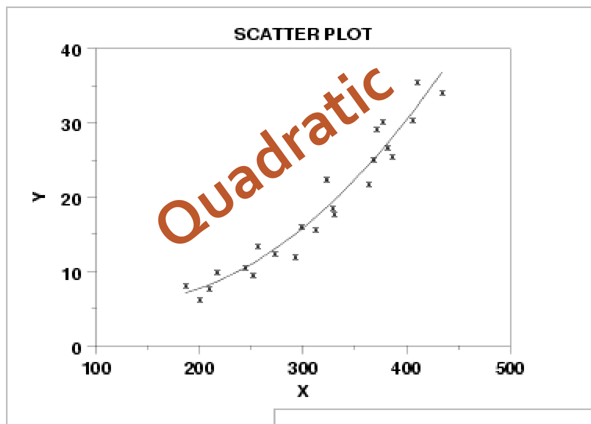
- Scatterplots are useful to answer:
  - Are x & y related?
  - Is the relationship linear, quadratic, other?
  - Does the variance of y depend on x?
  - Outliers present?

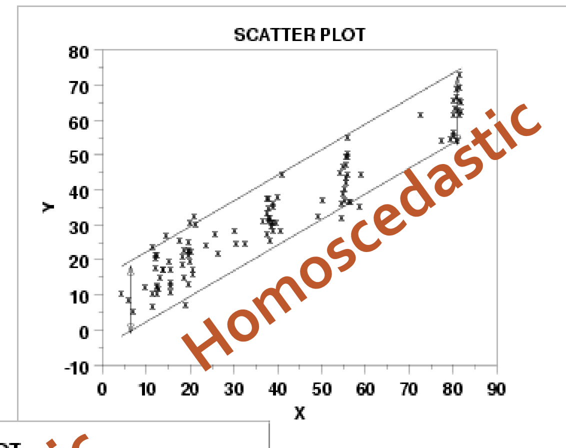


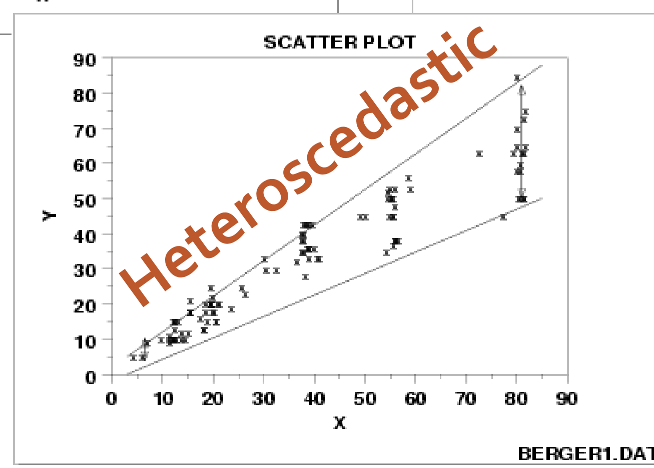
Two continuous variables











Two continuous variables

1. Why is whether homoscedasticity is present in your data important to determine in classical statistical modeling?

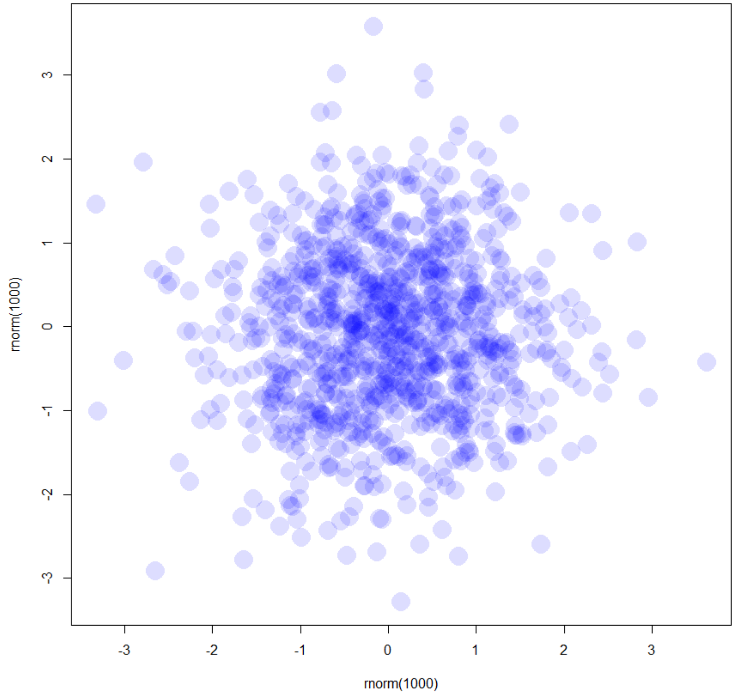
2. What is heteroscedasticity?

1. Homoscedasticity (i.e. same variance of residuals) is central assumption of linear regression models.

2. Variation in Y differs depending on the value of X
e.g. Y=annual income; X=age

Two continuous variables

- Scatterplots are not useful when there are lots of data
- But you can use transparent plotting to help with this issue
  - plot(rnorm(1000), rnorm(1000), col="blue", pch=16,cex=3)



Correlation coefficient

- How should we assess strength of association between 2 variables?
- Need a value that:
  - Doesn't change when units change
  - Makes no distinction between response & explanatory variables

Correlation coefficient

- **Correlation coefficient**: a quantity used to measure the direction & strength of a linear relationship between 2 quantitative variables
  - Denoted as 'r'
- Let x, y be any 2 quantitative variables for *n* individuals:

$$r = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu_x}{\sigma_x}\right)\left(\frac{y_i - \mu_y}{\sigma_y}\right)$$

Where μ_x and μ_y are the means and σ_x and σ_y are the SDs of the variables x & y

Correlation coefficient

$$r = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu_x}{\sigma_x}\right)\left(\frac{y_i - \mu_y}{\sigma_y}\right)$$

- Remember $\frac{x_i - \mu_x}{\sigma_x}$ and $\frac{y_i - \mu_y}{\sigma_y}$ are standardized values of variables x & y
- The correlation r is an average of the products of the standardized values of the 2 variables x & y for the *n* observations
- Cor(x,y) in R

Correlation coefficient

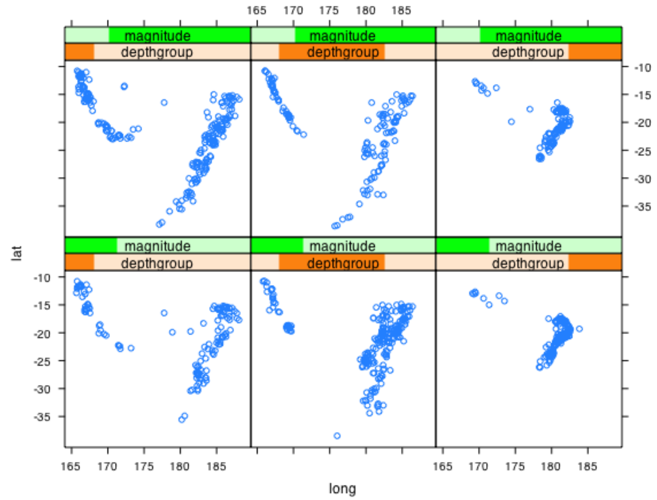
- True or False?
- Let X be weight in grams of piping plovers and Y be tarsus length in mm. Changing Y to cm changes the value of the correlation

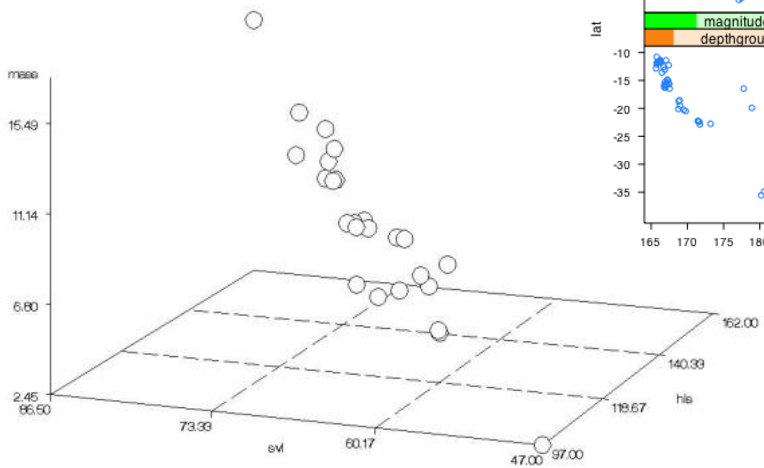
Correlation coefficient

- Properties of r
  - Makes no distinction between explanatory & response variables
  - Both variables must be quantitative
  - Is invariant to change of units
  - Between -1 & 1
  - Is affected by outliers
  - Measures strength of association ONLY for linear relationships

Multivariate data

- Get creative!
- Lots of different possible visualizations
  - Lattice plots
  - 3D plots





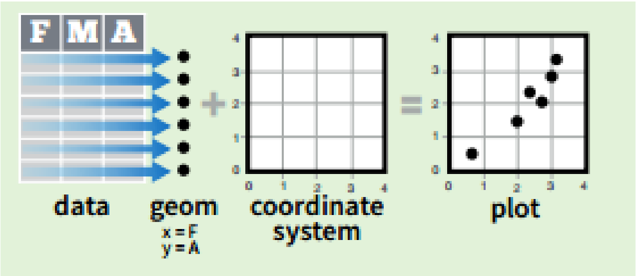
Exercises in R

- Let’s do some basic EDA & data visualization in RStudio

ggplot2: advanced visualization in R

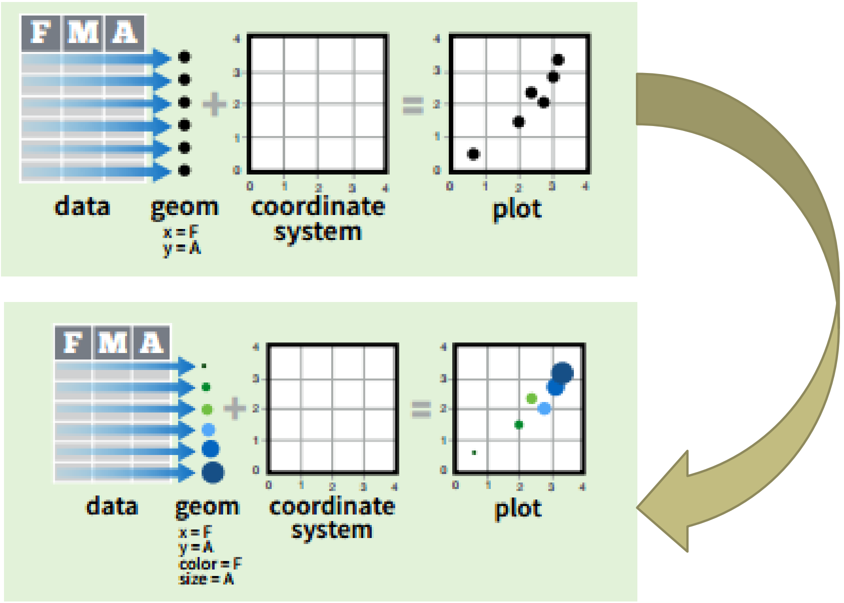
ggplot2

- ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same few components:
  - A data set
  - A set of geoms—visual marks that represent data points
  - A coordinate system



ggplot2

- To display data values, map variables in the data set to aesthetic properties of the geom like size, color & x/y locations

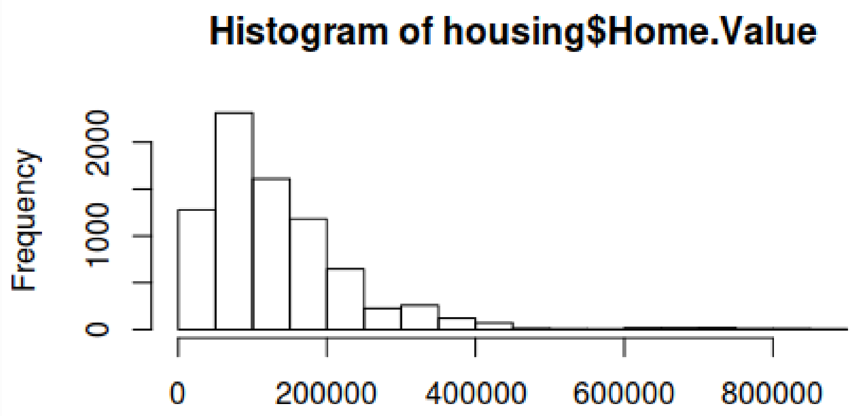


ggplot2

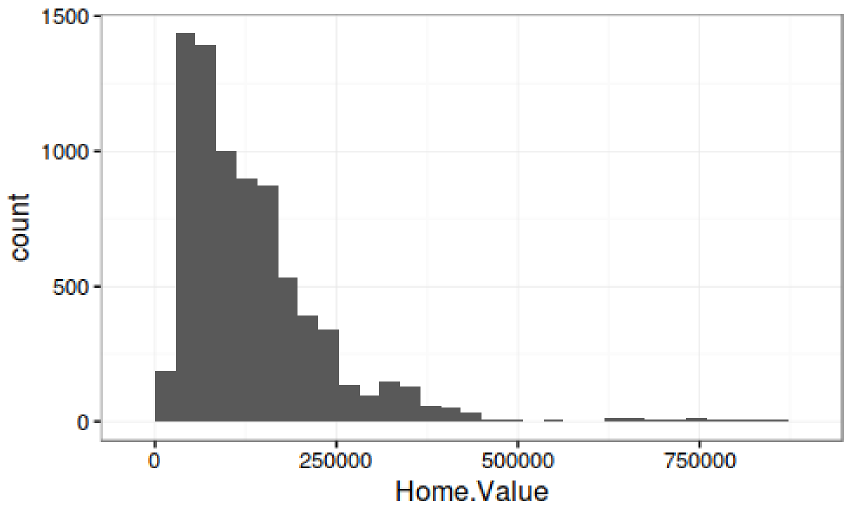
- Compared to base graphics, ggplot2
  - Is more verbose for simple graphics
  - Is less verbose for complex graphics
  - Is not method-specific (data always provided in a data frame)
  - Uses a different system for adding plot elements

Base graphics vs. ggplot2

```
hist(housing$Home.Value)
```

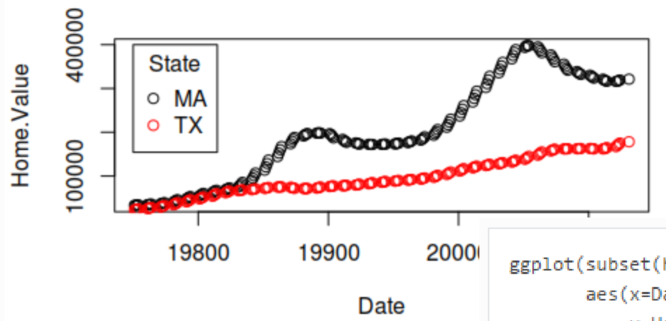



```
library(ggplot2)
ggplot(housing, aes(x = Home.Value)) +
  geom_histogram()
```

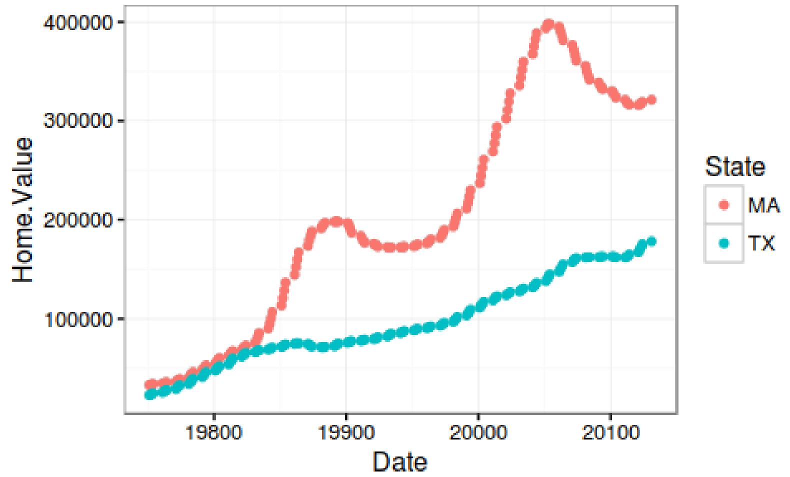


Base graphics vs. ggplot2

```
plot(Home.Value ~ Date,
     data=subset(housing, State == "MA"))
points(Home.Value ~ Date, col="red",
       data=subset(housing, State == "TX"))
legend(19750, 400000,
       c("MA", "TX"), title="State",
       col=c("black", "red"),
       pch=c(1, 1))
```

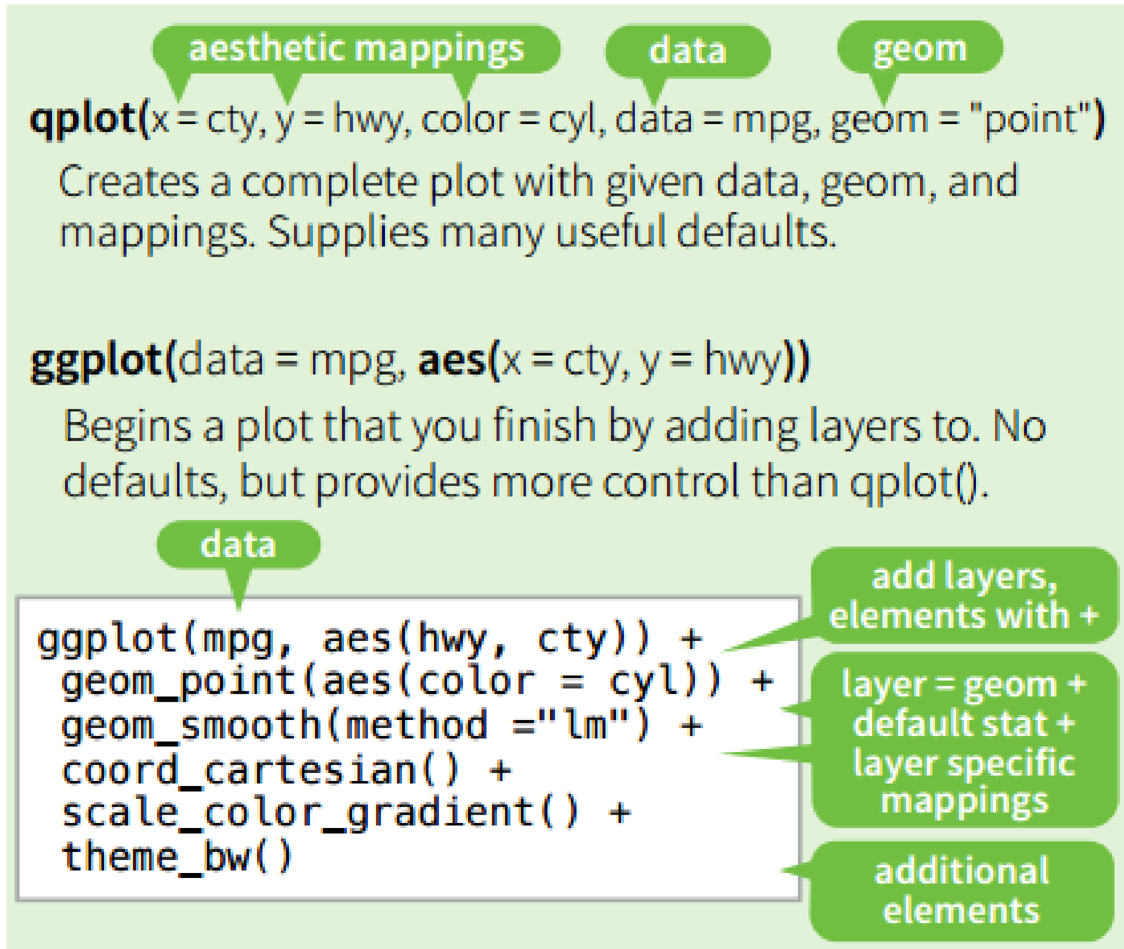



```
ggplot(subset(housing, State %in% c("MA", "TX")),
       aes(x=Date,
           y=Home.Value,
           color=State))+
  geom_point()
```



ggplot2

- Build a graph with commands qplot() or ggplot()



ggplot2

- Data: must be stored as a data frame
- Coordinate system: describes 2-d space that data is projected onto
  - E.g. Cartesian coordinates, polar coordinates, map projections
- Geoms: describe type of geometric objects that represent data
  - E.g. points, lines, polygons
- Aesthetics: describe visual characteristics that represent data
  - E.g. position, size, color, shape
  - Each type of geom accepts only a certain subset of all aesthetics

ggplot2

- Scales: for each aesthetic, describe how visual characteristic is converted to display values
  - E.g. log scales, color scales
- Stats: describe statistical transformations that typically summarize data
  - E.g. counts, means, medians

ggplot2

- Pretend we have a data set on world population attributes
  - tfr: total fertility rate
  - le: life expectancy at birth
  - area: Africa, America, Asias, etc.

country	pop2012	tfr	le	area
Algeria	37.4	2.9	73	Africa
Egypt	82.3	2.9	72	Africa
Libya	6.5	2.6	75	Africa
Morocco	32.6	2.3	72	Africa
South Sudan	9.4	5.4	52	Africa
Sudan	33.5	4.2	60	Africa
Tunisia	10.8	2.1	75	Africa
Benin	9.4	5.4	56	Africa
Burkina Faso	17.5	6.0	55	Africa
Cote d'Ivoire	20.6	4.6	55	Africa
Gambia	1.8	4.9	58	Africa
Ghana	25.5	4.2	64	Africa
⋮	⋮	⋮	⋮	⋮

ggplot2

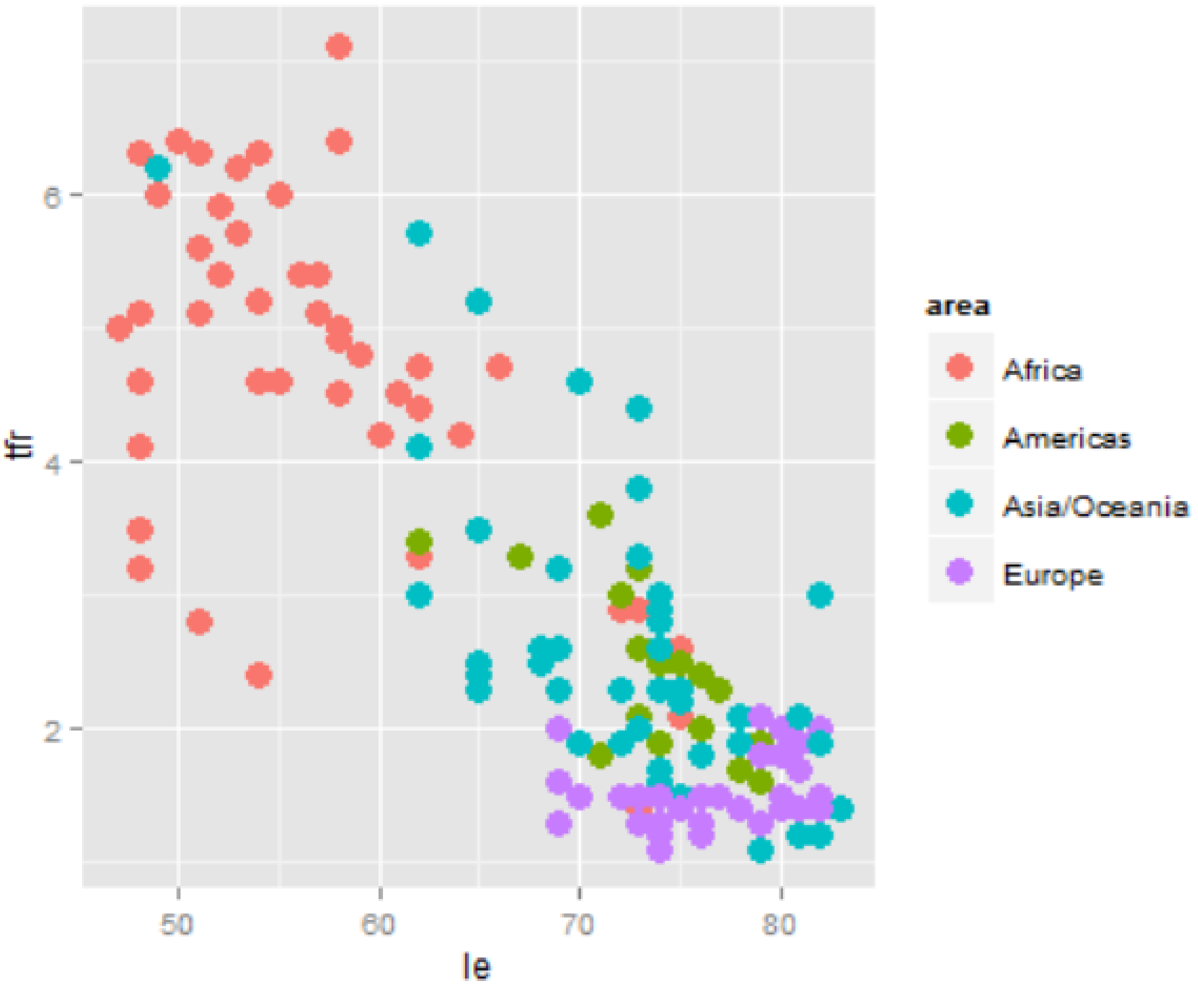
P <- ggplot(data=w, aes(x=le, y=tfr, color=area))

- le value is indicated by x position
- tfr value is indicated by y position
- area value indicated by color

- BUT object P cannot be displayed without adding another layer—there's nothing to see yet! You need at least 1 geom layer in all plots

P + layer(geom="point", geom_params=list(size=4))

ggplot2



6
4
2
tfr
50
60
70
80
le
area
Africa
Americas
Asia/Oceania
Europe

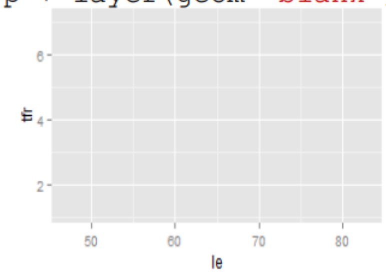
ggplot2

- Full specification of a layer:

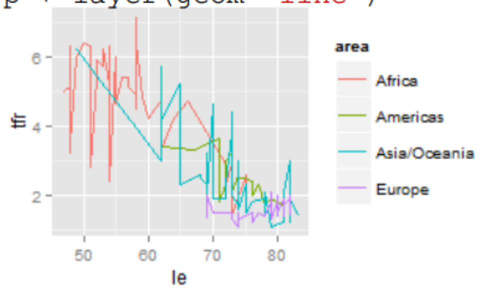
layer(geom, geom_params, stat, stat_params, data, mapping, position)

- Every layer specifies a geom, stat or both
- Add a geom layer examples:

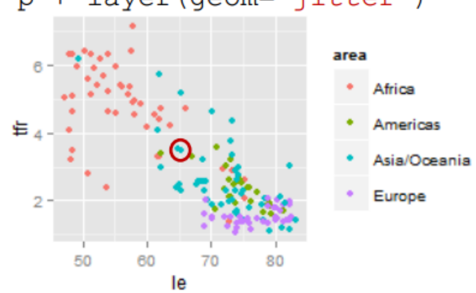
p + layer(geom="blank")



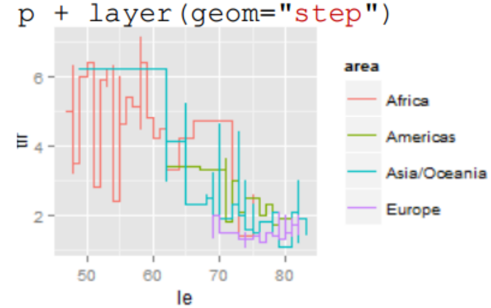
p + layer(geom="line")



p + layer(geom="jitter")



p + layer(geom="step")

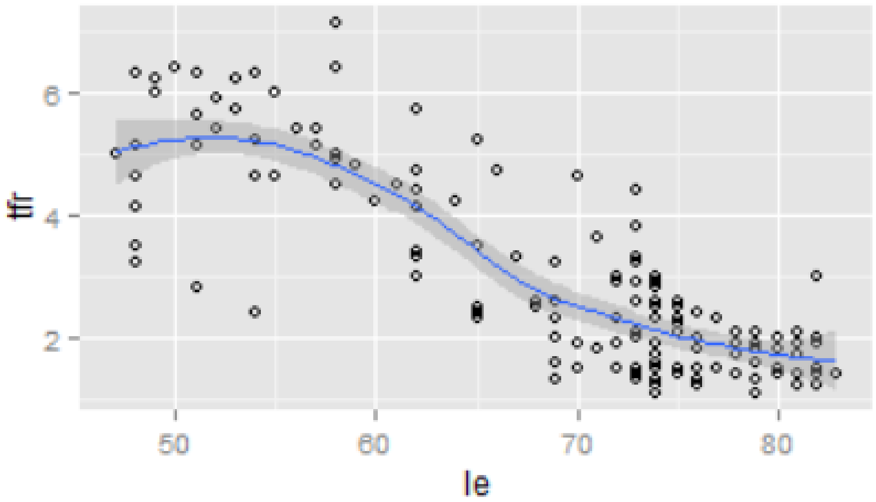


ggplot2

- Add a stat layer:

P +
layer(geom="point",geom_params=list(shape=1)) +

layer(stat="smooth")



ggplot2

- Can use geom_xxx and stat_xxx shortcut functions so you don't have to keep typing 'layer…'

```
P + geom_point(shape=1) +
stat_smooth()
```

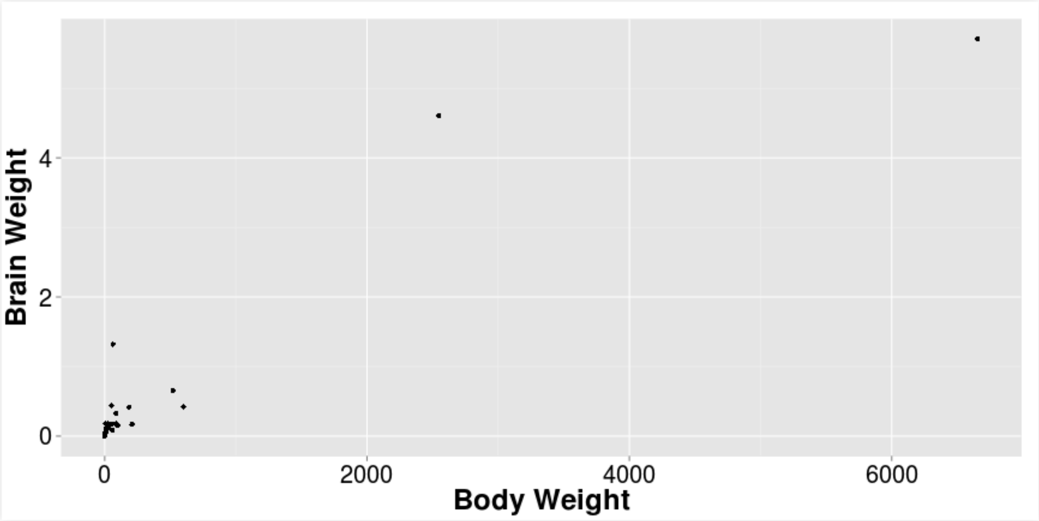
- This will be the convention of writing ggplot2 code in labs/exercises

ggplot2

- Some more examples before trying it in R
  - msleep is the data set to graph
  - geom_point says to make it a scatterplot
  - aes (or aesthetics) tells R what to put on x & y axes

```
alt <- ggplot(msleep) + geom_point(aes(x=bodywt, y=brainwt))
```

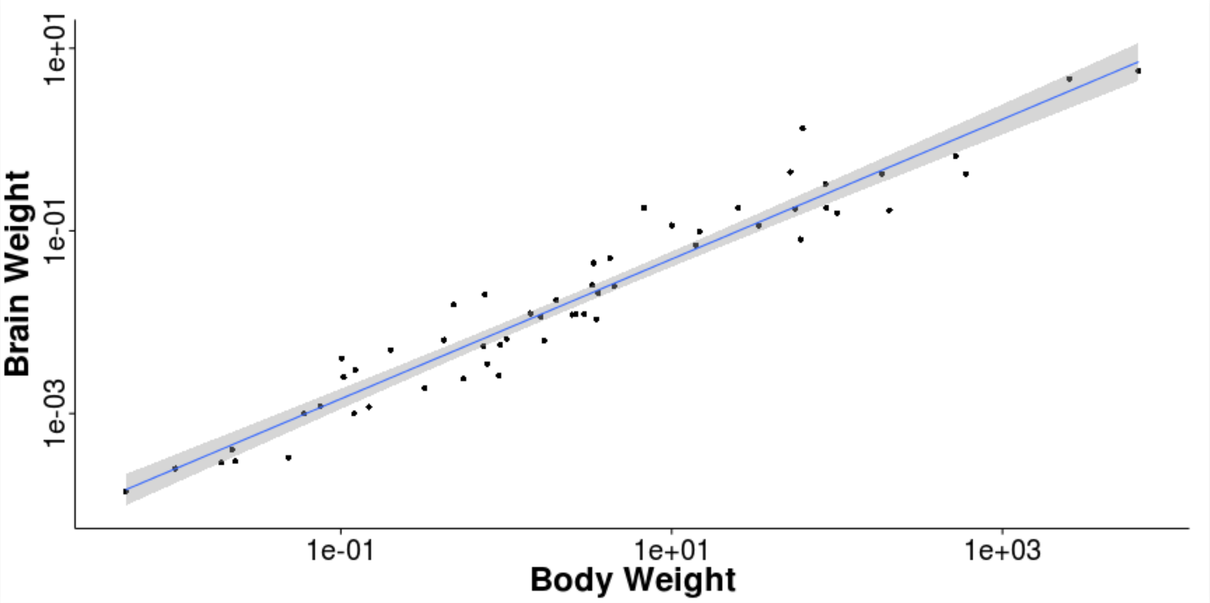
```
alt
```



ggplot2

- Add OLS regression line

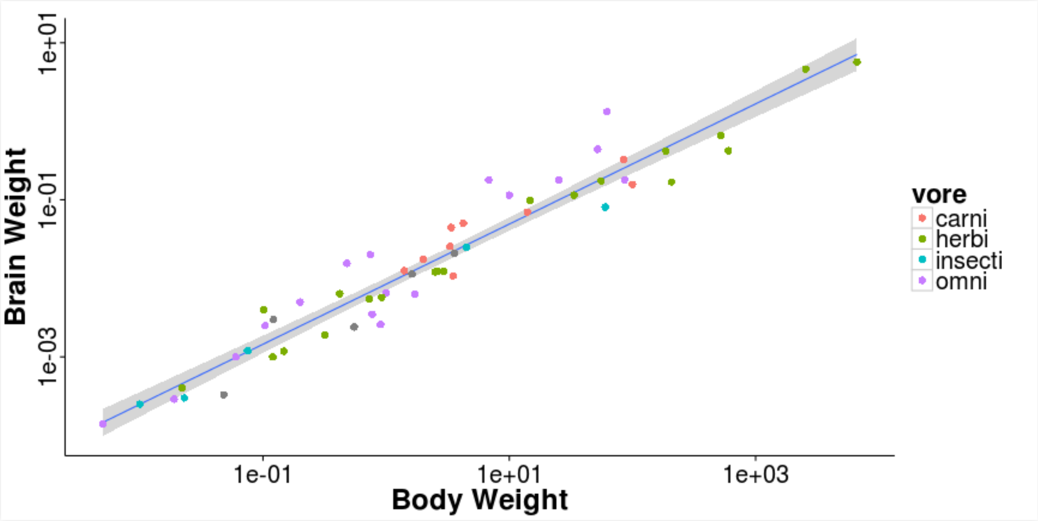
```
p <- p + stat_smooth(method="lm")
p
```



ggplot2

- Specify color according to diet group

```
p + geom_point(aes(color=vore), size=3)
```

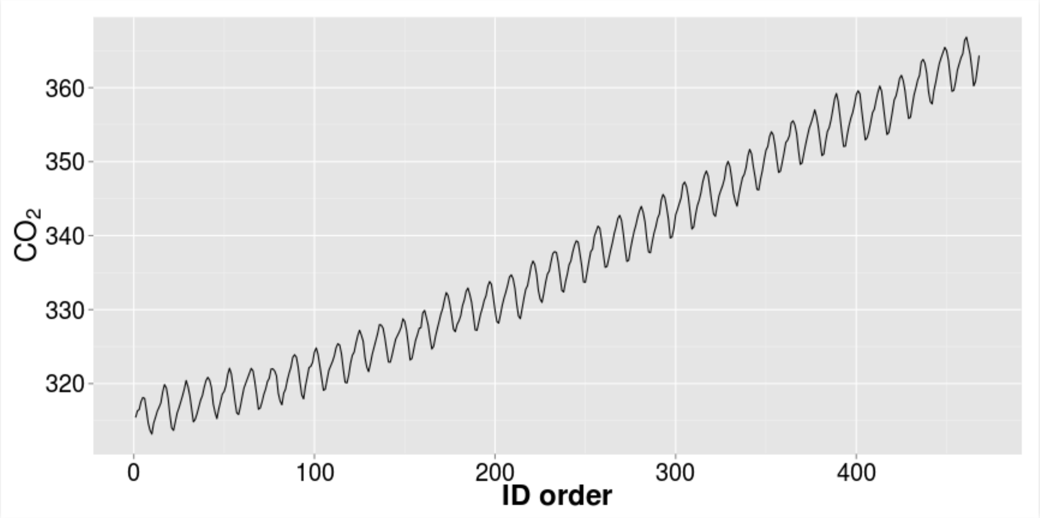


ggplot2

- Line plots

```
data(co2)
lp <- ggplot(data.frame(co2), aes(x=1:length(co2), y=co2)) + geom_line()
```

```
lp
```

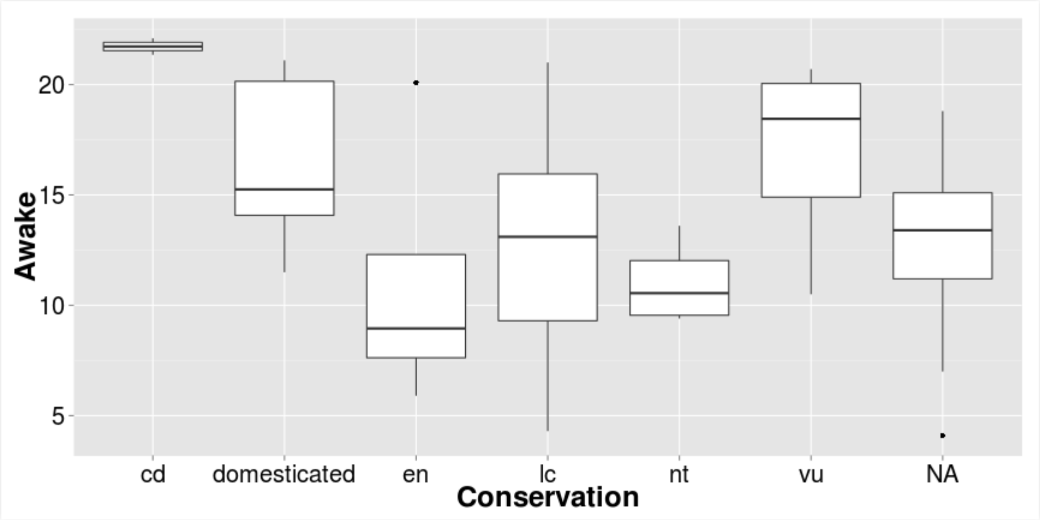


ggplot2

- Boxplots

```
bp <- ggplot(msleep, aes(x=conservation, y=awake)) + geom_boxplot()
```

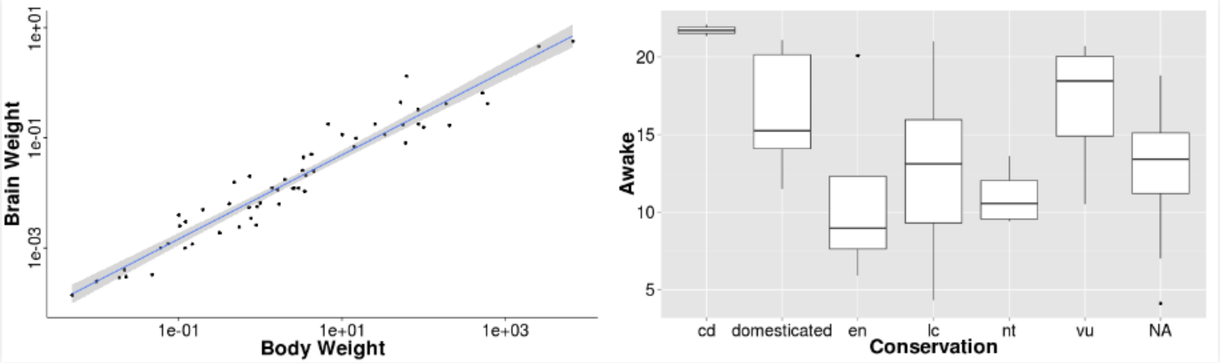
```
bp
```



ggplot2

- Combining plot types

```
require(gridExtra)
grid.arrange(p, bp, ncol=2)
```



ggplot2 practice

- Let's go to RStudio to go through a ggplot2 tutorial
- Assignment #2 due 9/22 @ midnight