Data visualization in

Leland Taylor
@lelandtlr

November 21, 2014
www.ebi.ac.uk

EMBL-EBI

Outline

1. ggplot2 for data visualizations
2. Bioconductor for data visualizations
3. Other cool stuff

http://bit.ly/1yWPQA1

Introduction

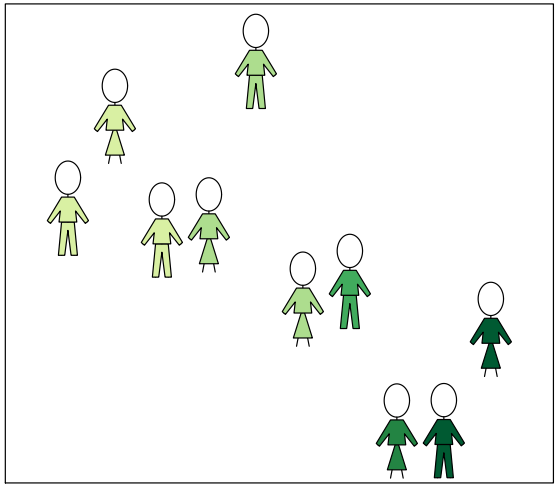
Packages extend R

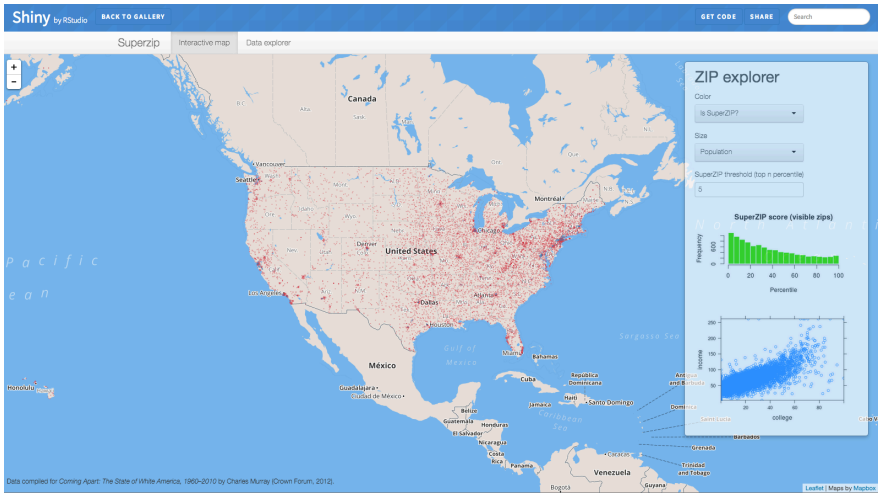
Bundles of functions that extend the core R language

Stored in code repositories (e.g. The Comprehensive R Archive Network – CRAN)

Packages extend R

A package for everything





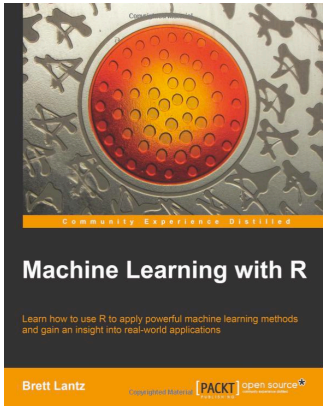
Anders and Huber *Genome Biology* 2010, **11**:R106
http://genomebiology.com/2010/11/10/R106

Genome Biology

METHOD **Open Access**

Differential expression analysis for sequence count data

Simon Anders*, Wolfgang Huber



Packages are easy to install and use

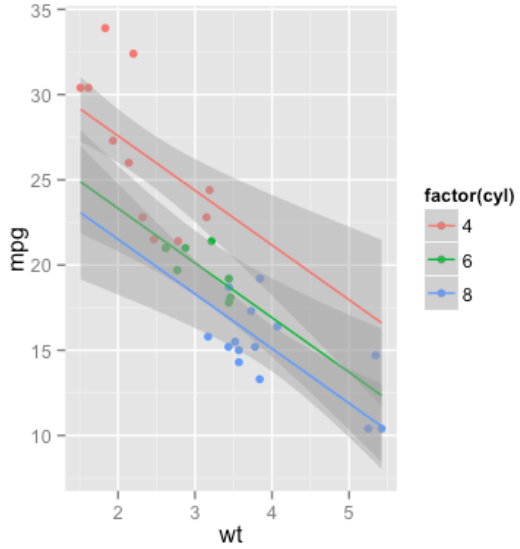
```
# install package
install.packages("ggplot2")


# load package
library("ggplot2")


# supplemental documentation
vignette()
```

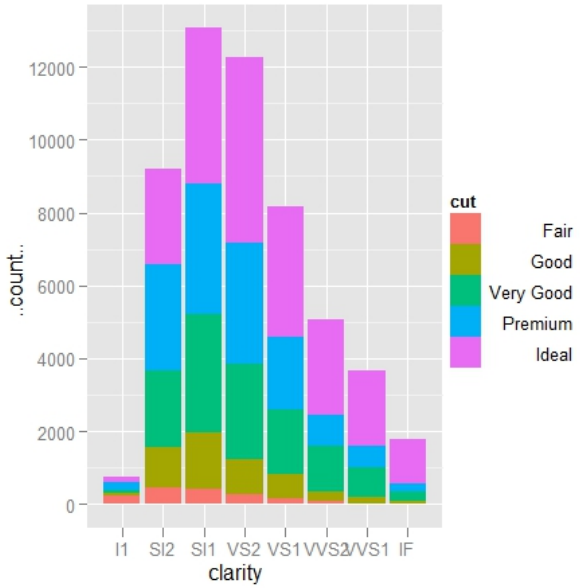
ggplot2 for data visualizations

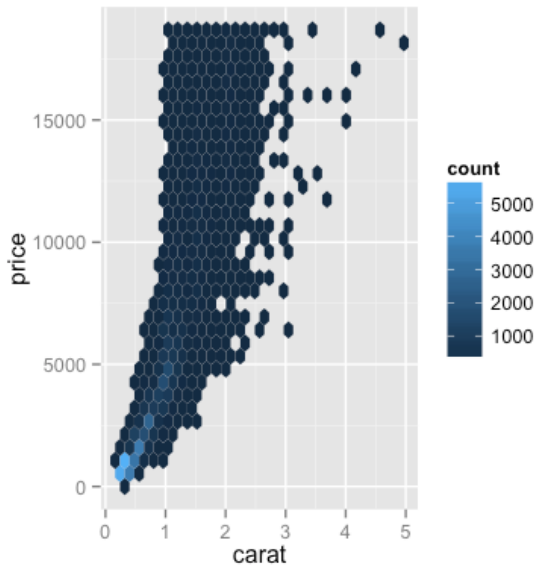
ggplot2

Dataframe based plotting package

http://docs.ggplot2.org







The core function

```
# guesses what plot you want
qplot()

# gives you more control
ggplot()
```

Plot composition

1. A dataset and set of aesthetic mappings
2. Multiple layers
3. A scale for each aesthetic*
4. A coordinate system*

* Optional user input

A dataset and aesthetic mappings

ggplot() initializes a reference to the source data and global aesthetic mappings

```
ggplot(dat = <source_data>,
            aes(<aesthetic_mappings>)
```

A dataset example

```
> head(mpg)
  manufacturer model displ year cyl      trans drv cty hwy fl   class
1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

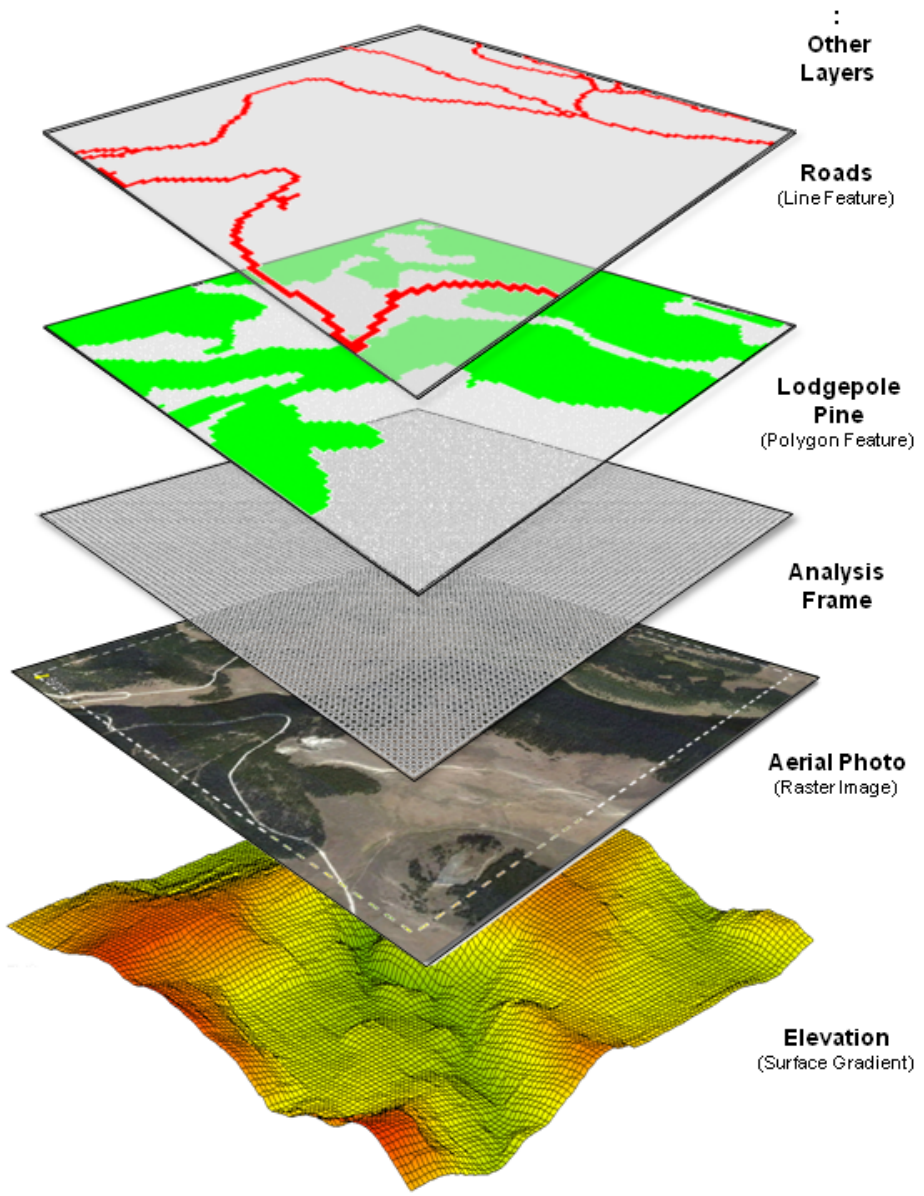
Each row is a point

```
# x and y mappings are displ and hwy for all layers
ggplot(mpg, aes(x=displ, y=hwy))
```

Plots have layers

Layers are the components of the plot

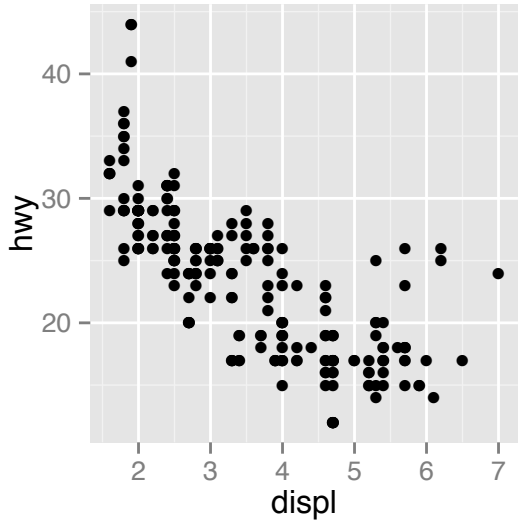
Layers are stacked on top of each other to render the final image



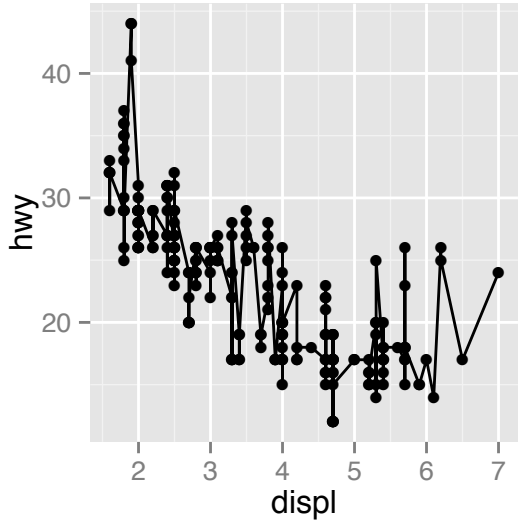
http://www.innovativegis.com/basis/BeyondMappingSeries/BeyondMapping_IV/Topic1/BM_IV_T1_files/image008.png

Adding multiple layers

```
ggplot(mpg, aes(x=displ, y=hwy)) +
      geom_point()
```




```
ggplot(mpg, aes(x=displ, y=hwy)) +
      geom_point() +
      geom_line()
```



Layers have 2 flavors

1. Geometric Objects
   - Plot the raw data
   - **geom_<name>**

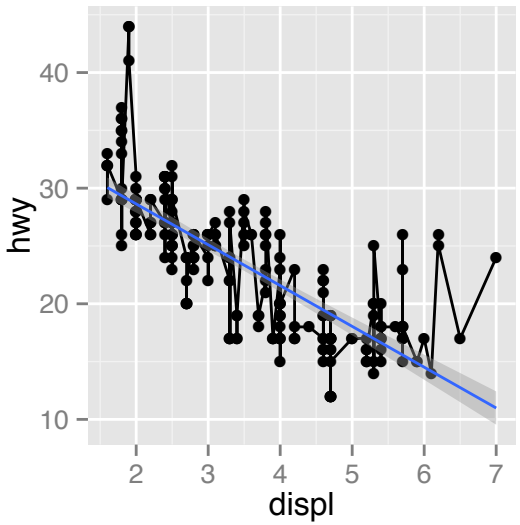
     geom_point(), geom_line(), geom_bar(), geom_boxplot()…

2. Statistical transformations
   - Transform the raw data and then plot it
   - **stat_<name>**

     stat_summary (), stat_smooth(), stat_bin()…

An example of a stat layer

```
ggplot(mpg, aes(x=displ, y=hwy)) +
      geom_point() +
      geom_line() +
      stat_smooth(method="lm")
```



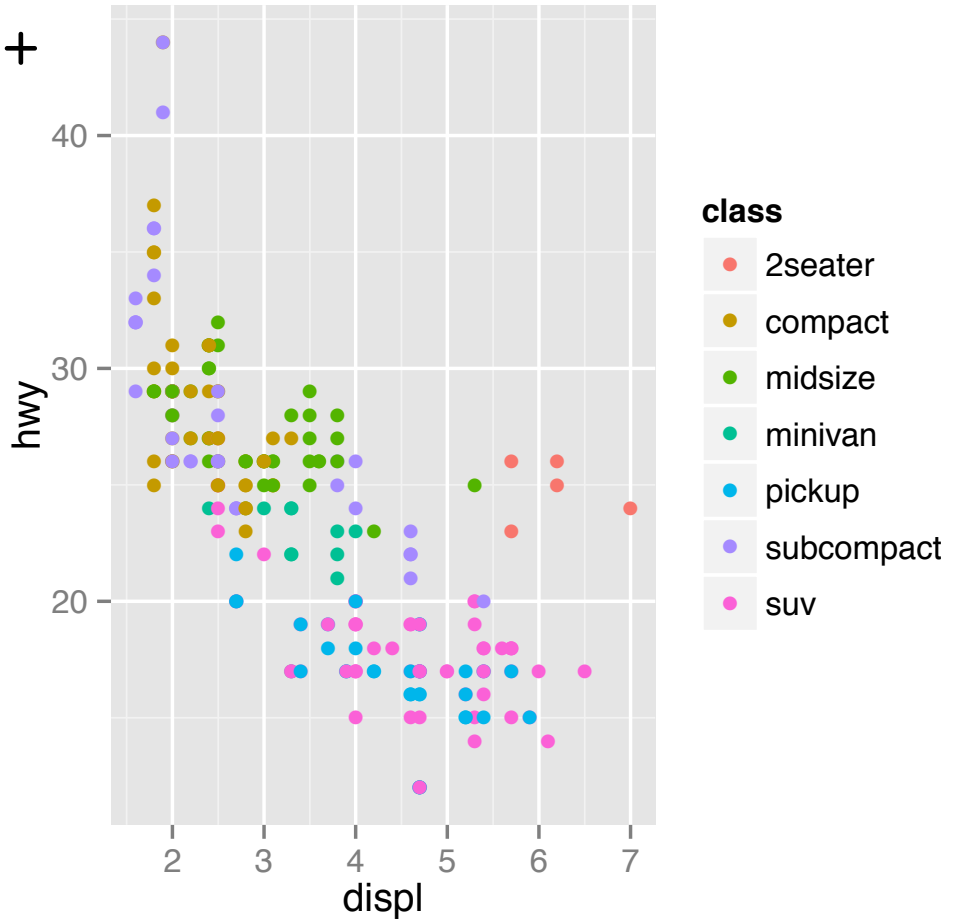
stat_smooth fits and plots a linear regression

We can add many aesthetic mappings

Type	Discrete Data	Continuous Data
Color & Fill	Different color for each (rainbow of colors)	Linear mapping between gradient and value
Size	Discrete size steps for each	Linear mapping between radius and value
Shape	Different shape for each	-
Alpha	Different alpha for each	Linear mapping between alpha and value

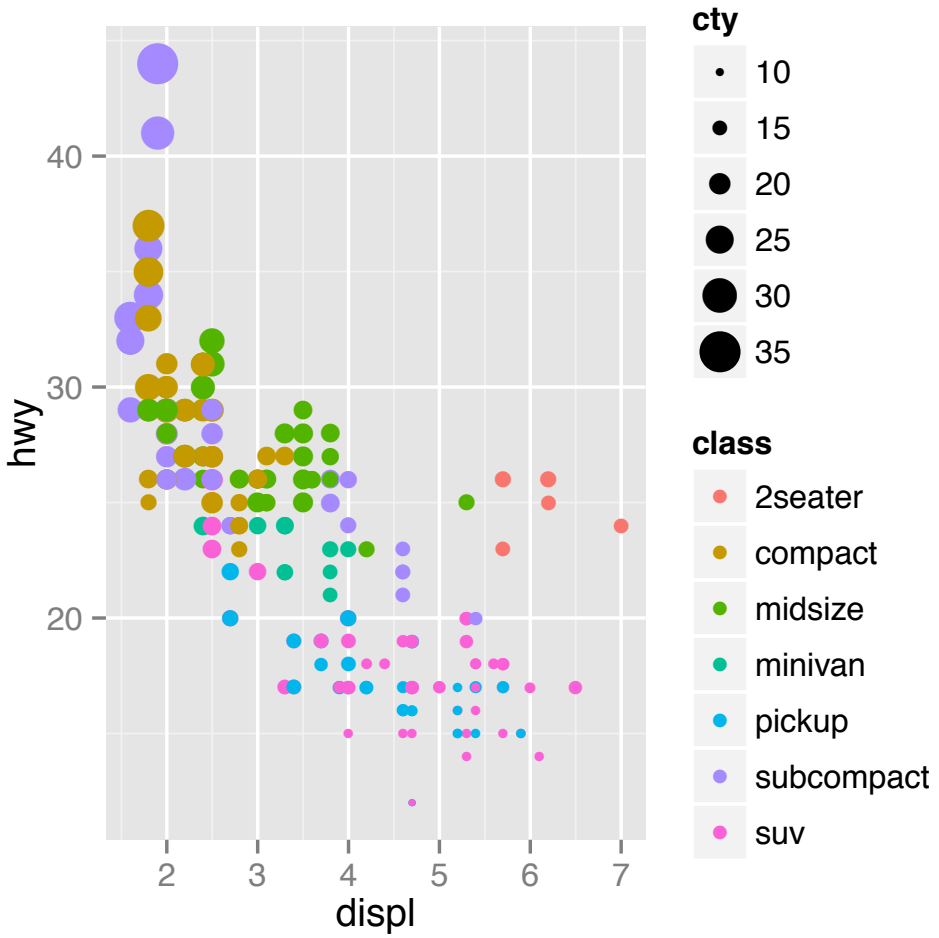
Aesthetic mappings make pretty plots

```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class)) +
       geom_point()
```



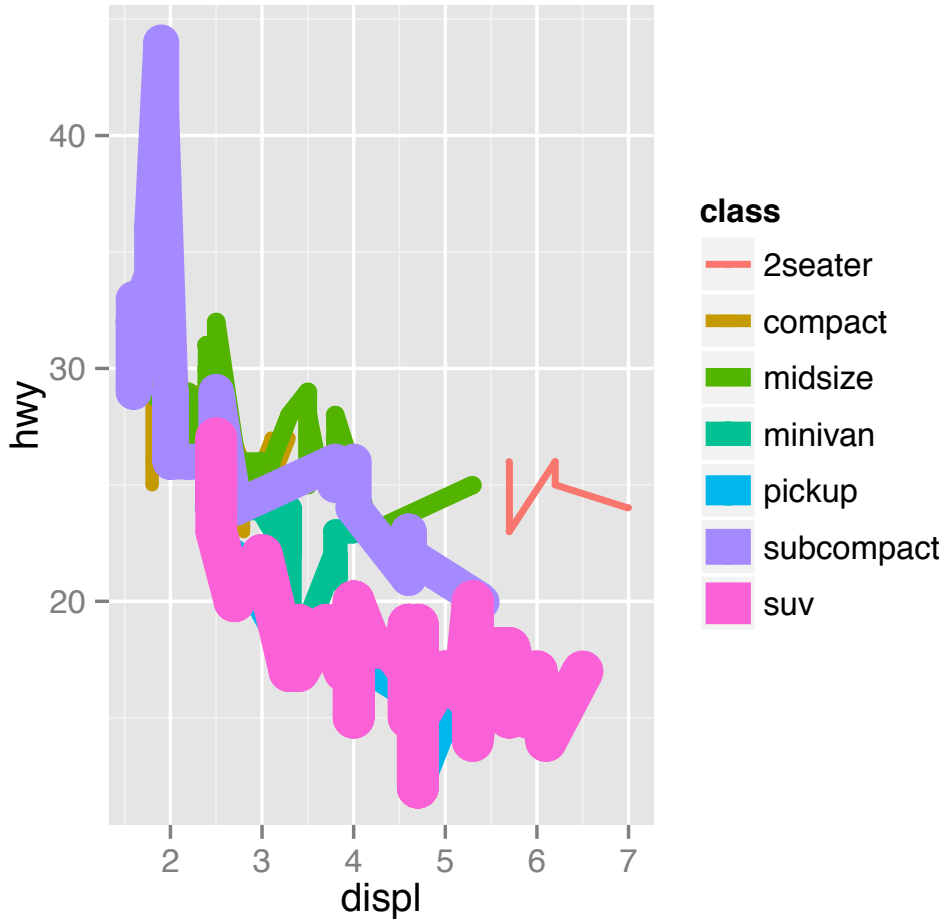
Aesthetic mappings make pretty plots

```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class,
       size=cty)) +
       geom_point()
```



Aesthetic mappings can be global or local

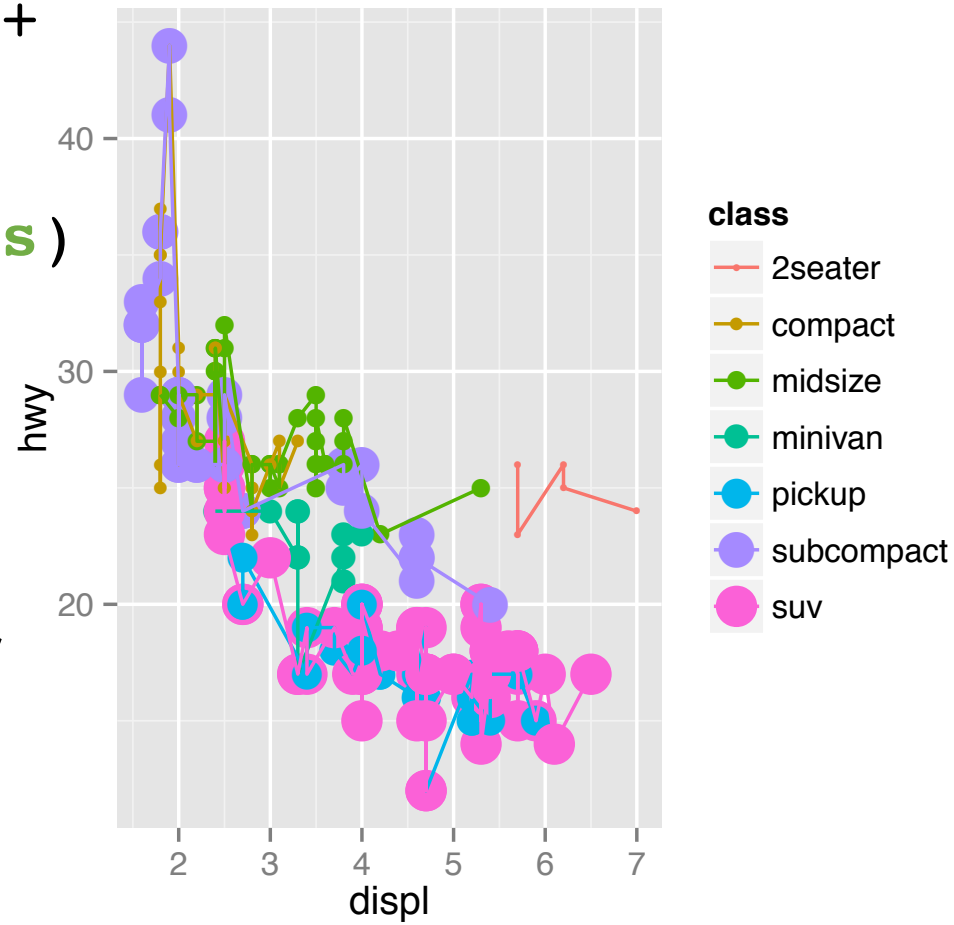
```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class,
       size=class)) +
       geom_point() +
       geom_line()
```



size applies to <u>all layers</u>

Aesthetic mappings can be global or local

```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class)) +
       geom_point(
        aes(size=class)
       ) +
       geom_line()
```



size applies to point layer

Your turn

http://docs.ggplot2.org

Anscombe_summarize.csv

1. Summarize all variables
   - Median, quartiles, outliers, distribution of the data

Anscombe_regression.csv

1. Fit a linear regression to the **entire** dataset
   - Education should be the dependent variable
2. Fit a linear regression to **each** variable

Plot scales

Scale = mappings of data to aesthetic properties

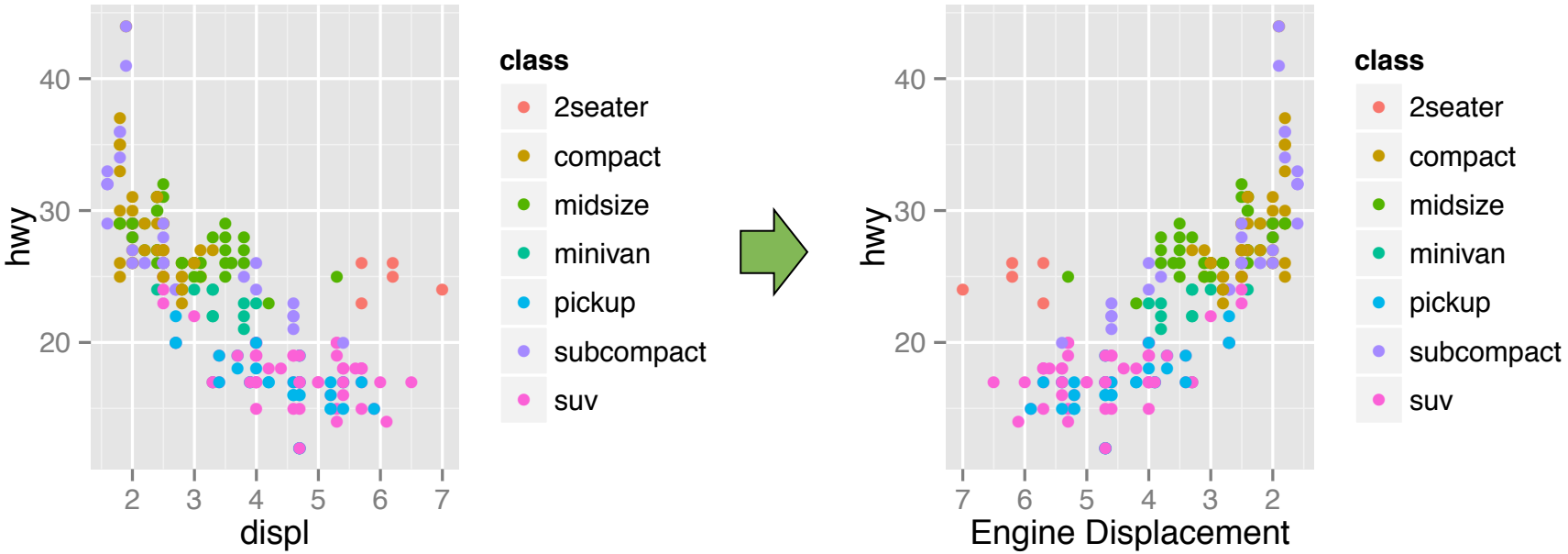
Convey mappings via guides (axes and legends)

Manually adjusting scales:

- **scale_<aesthetic_name>**
- All default scales are continuous or discrete

Manually adjusting scales

```
ggplot(mpg, aes(x=displ, y=hwy, color=class)) +
      geom_point() +
      scale_x_reverse("Engine Displacement")
```



Plot coordinate system

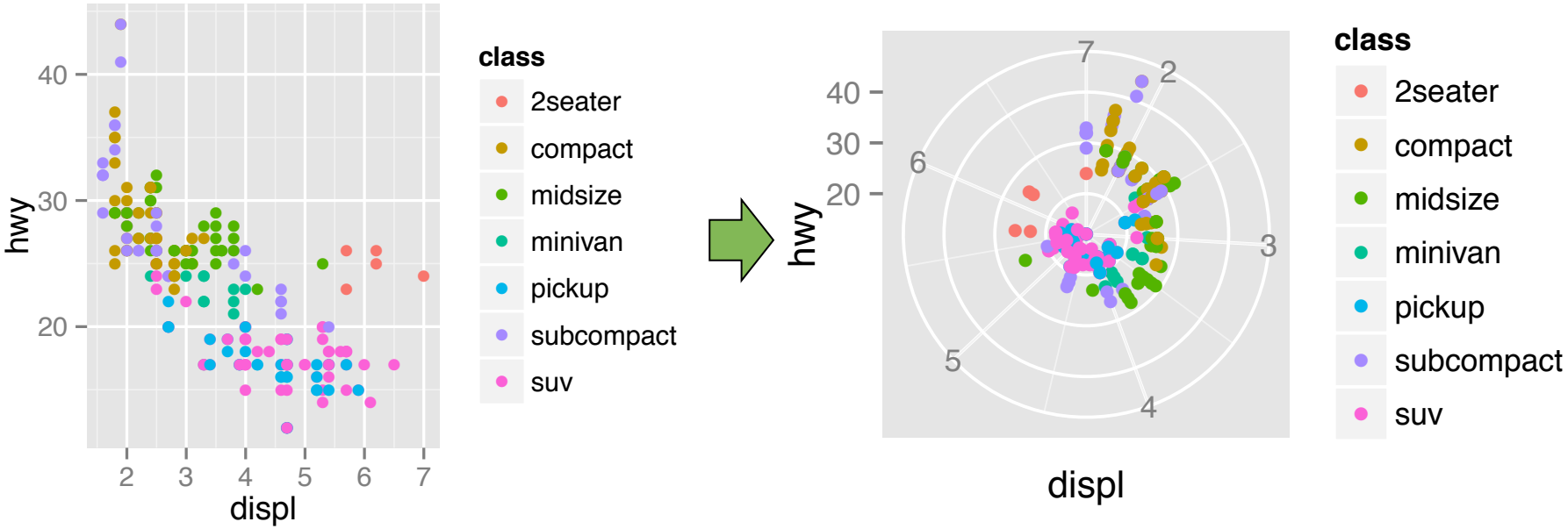
Coordinate systems = mapping from plot coordinates to the 2d plane of the computer screen.

Manually adjusting coordinates:

- **coord_<coordinate_system>**
- Can transform or zoom in on coordinates

Manually adjusting coordinate system

```
ggplot(mpg, aes(x=displ, y=hwy, color=class)) +
      geom_point() +
      coord_polar()
```



Your turn

http://docs.ggplot2.org

Anscombe_summarize.csv

1. Summarize all variables
   - Median, quartiles, outliers, distribution of the data
   - A linear and log10 scale
   - Plots should have a title and labels on **all aesthetics**

Facets

Facets display subsets of the dataset in different panels.

facet_grid()

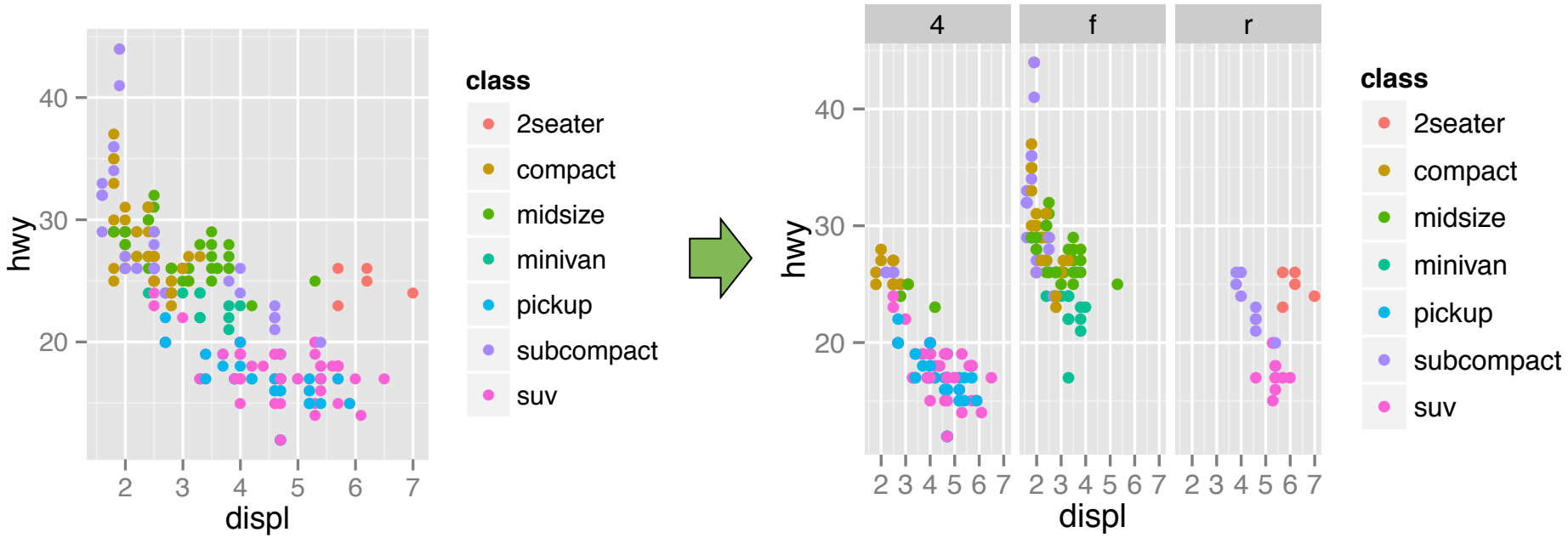
- Lay out panels in a grid

facet_wrap()

- Wrap a ribbon of panels

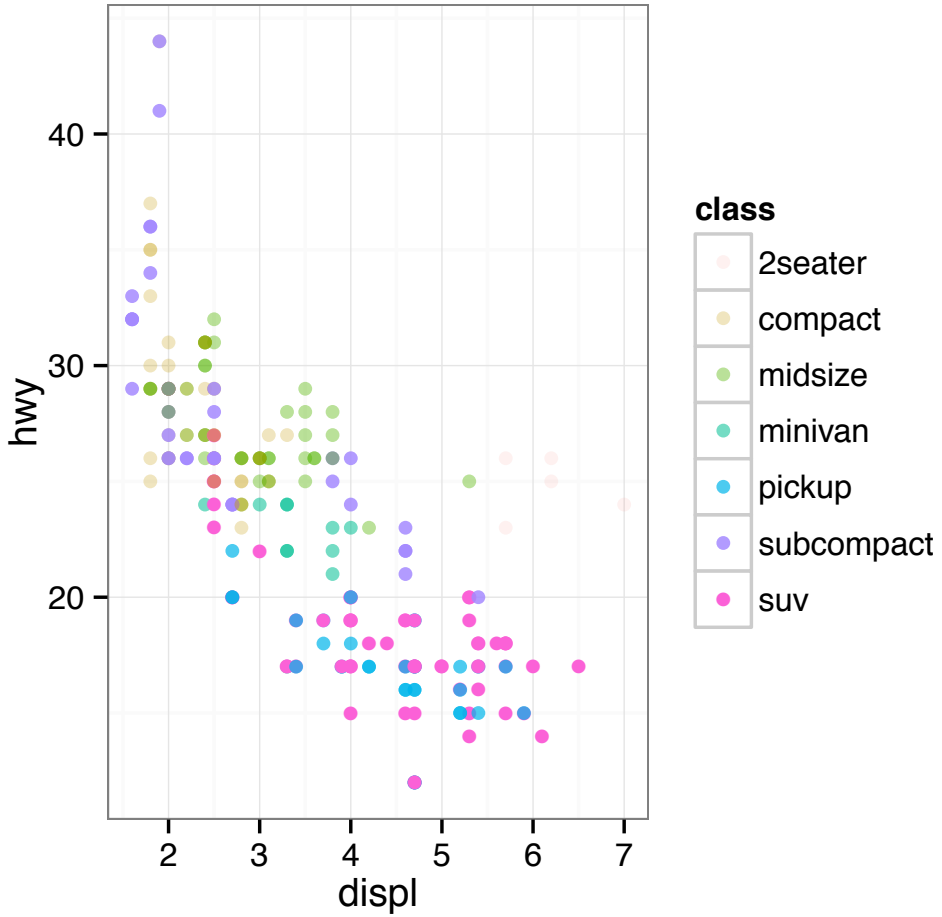
Facets subset data

```
ggplot(mpg, aes(x=displ, y=hwy, color=class)) +
      geom_point() +
      facet_grid( ~ drv)
```



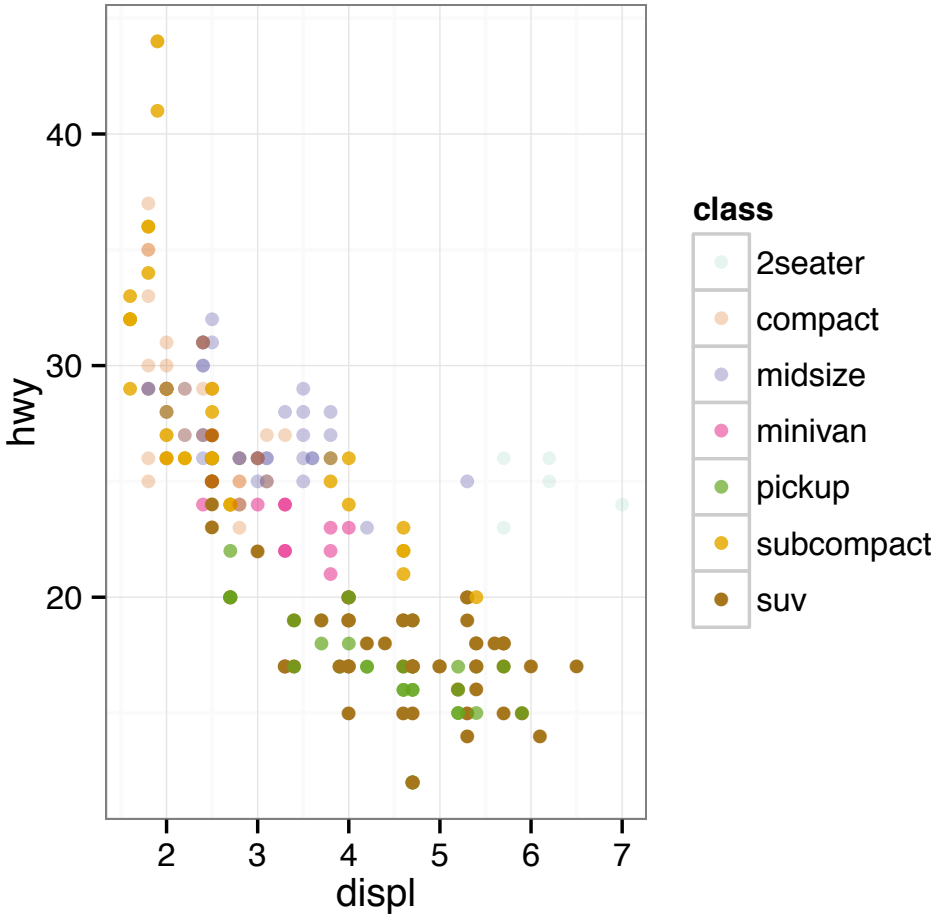
Preset themes for your plot

```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class,
       alpha=class)) +
       geom_point() +
       theme_bw()
```



Preset color schemes for your plot

```
ggplot(mpg, aes(x=displ, y=hwy,
       color=class,
       alpha=class)) +
       geom_point() +
       theme_bw() +
scale_colour_brewer(
       palette="Dark2"
)
```



Saving plots

```
# saves your last plot, guesses settings
ggsave("cars.pdf")


# pdf
pdf(file="cars01.pdf", height=5, width=6)
      <ggplot_call_here>
dev.off()


# png = good for many data points
png(file="cars.png", width=1000, height=700,
      units='px', res=120)
      <ggplot_call_here>
dev.off()
```

Your turn

http://docs.ggplot2.org

Anscombe_regression.csv

1. Fit a linear regression to **each** variable
   - Education should be the dependent variable

- Plots should have a title and labels on **all aesthetics**
- Change something about the theme
- Set the scale to the EBI color scheme
  - "#71B360", "#48877C", "#CC6D78"
- Facet by variable and adjust the scales so you can see the regression
- Save the plot to a pdf file

Bioconductor for data visualizations

Bioconductor

Collection of packages for high-throughput genomics

http://www.bioconductor.org/

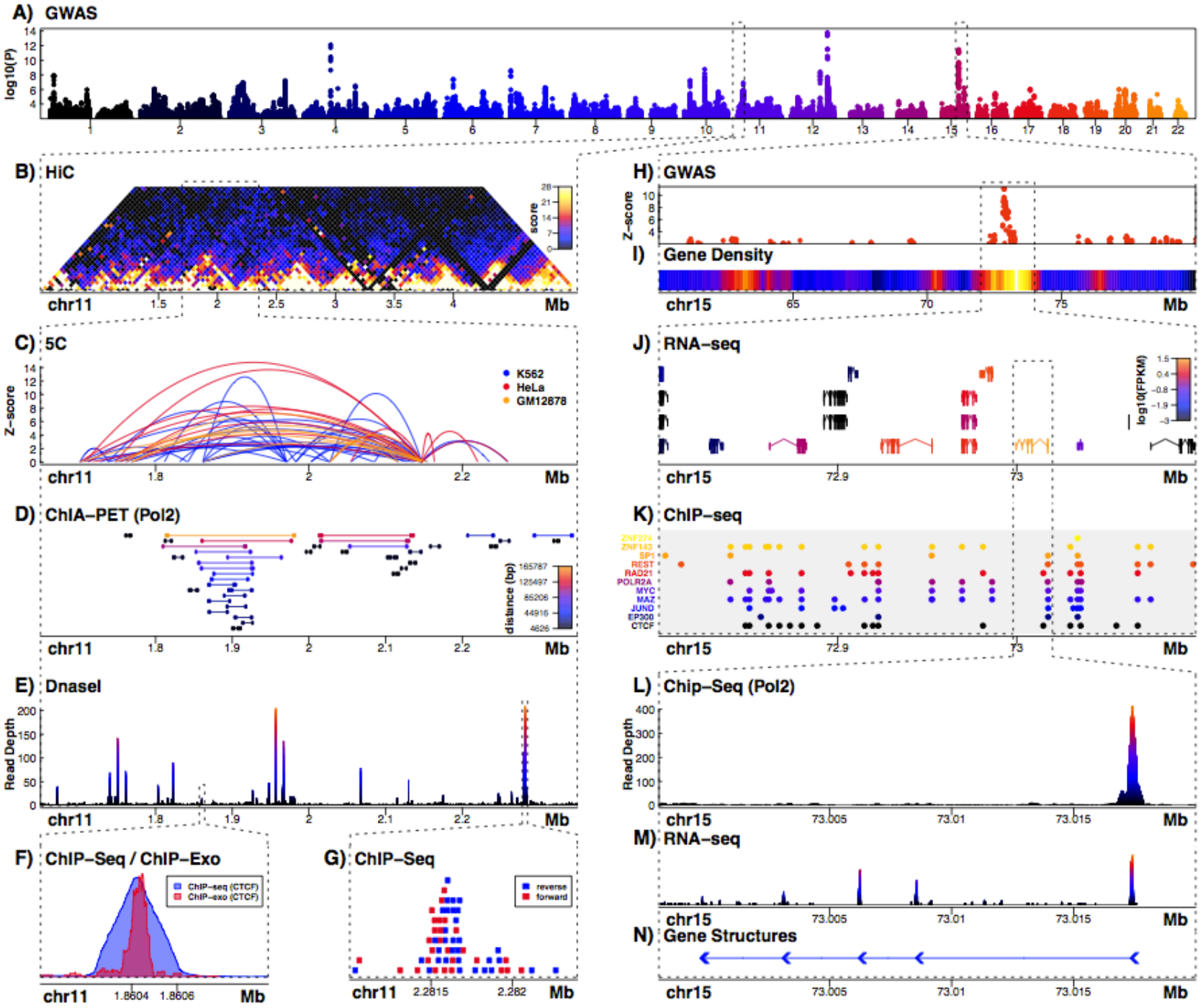
R + Bioconductor Applications

- Visualize high-throughput genomics data
- Differential gene expression analysis
- Gene ontology enrichment
- Workflows for analyzing variants, epigenomics data (e.g. ChIP-seq), transcriptome data
- Applying machine learning techniques to biological data

http://www.bioconductor.org/help/workflows/

Installing Bioconductor

```
source("http://bioconductor.org/biocLite.R")
biocLite()


# install the Sushi
biocLite("Sushi")
```


A) GWAS
log10(P)
B) HiC
score
chr11
Mb
C) 5C
Z-score
K562
HeLa
GM12878
D) ChIA-PET (Pol2)
distance (bp)
E) DnaseI
Read Depth
F) ChIP-Seq / ChIP-Exo
ChIP-seq (CTCF)
ChIP-exo (CTCF)
G) ChIP-Seq
reverse
forward
H) GWAS
Z-score
I) Gene Density
chr15
J) RNA-seq
log10(FPKM)
K) ChIP-seq
ZNF274
ZNF143
SP1
REST
RAD21
POLR2A
MYC
MAZ
JUND
EP300
CTCF
L) Chip-Seq (Pol2)
M) RNA-seq
N) Gene Structures

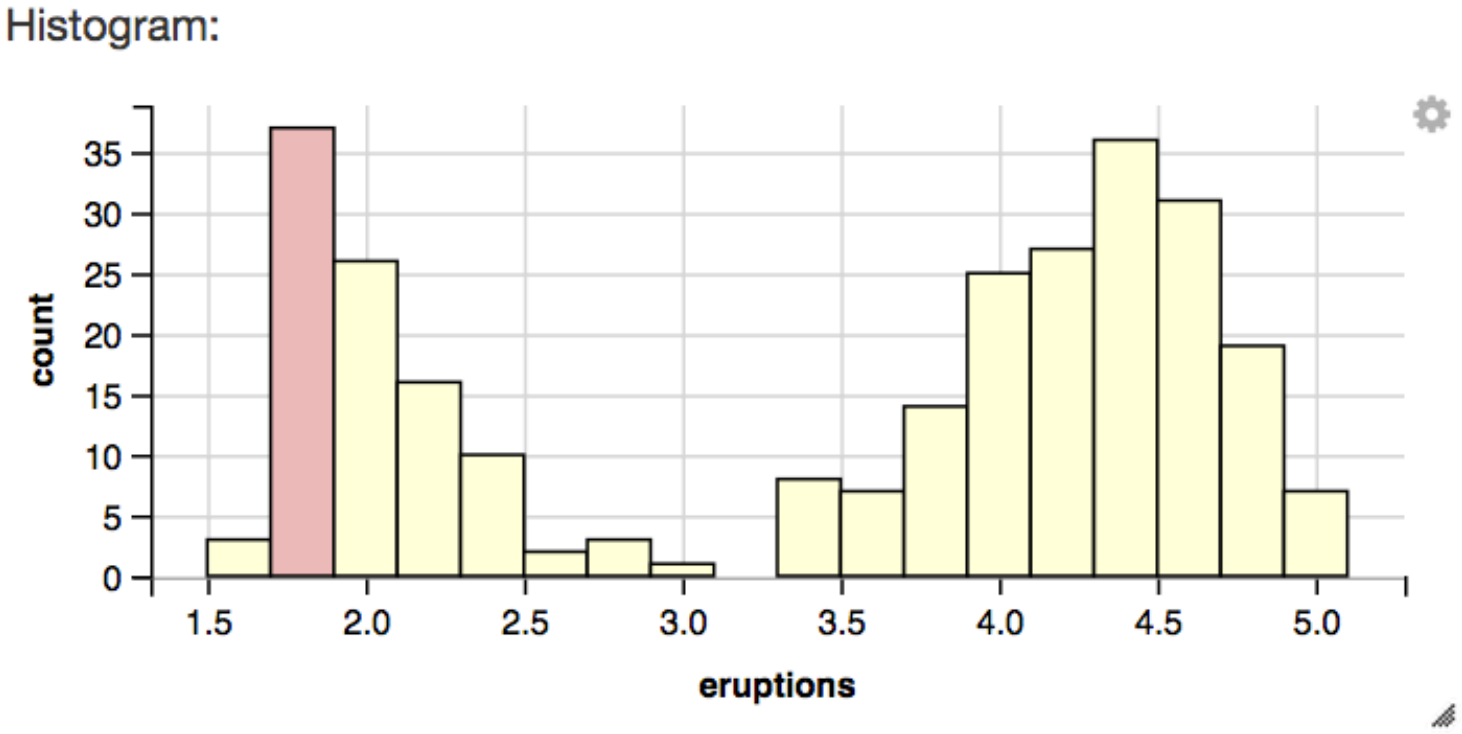
Other cool stuff

*wow your boss

ggvis

Create interactive graphics

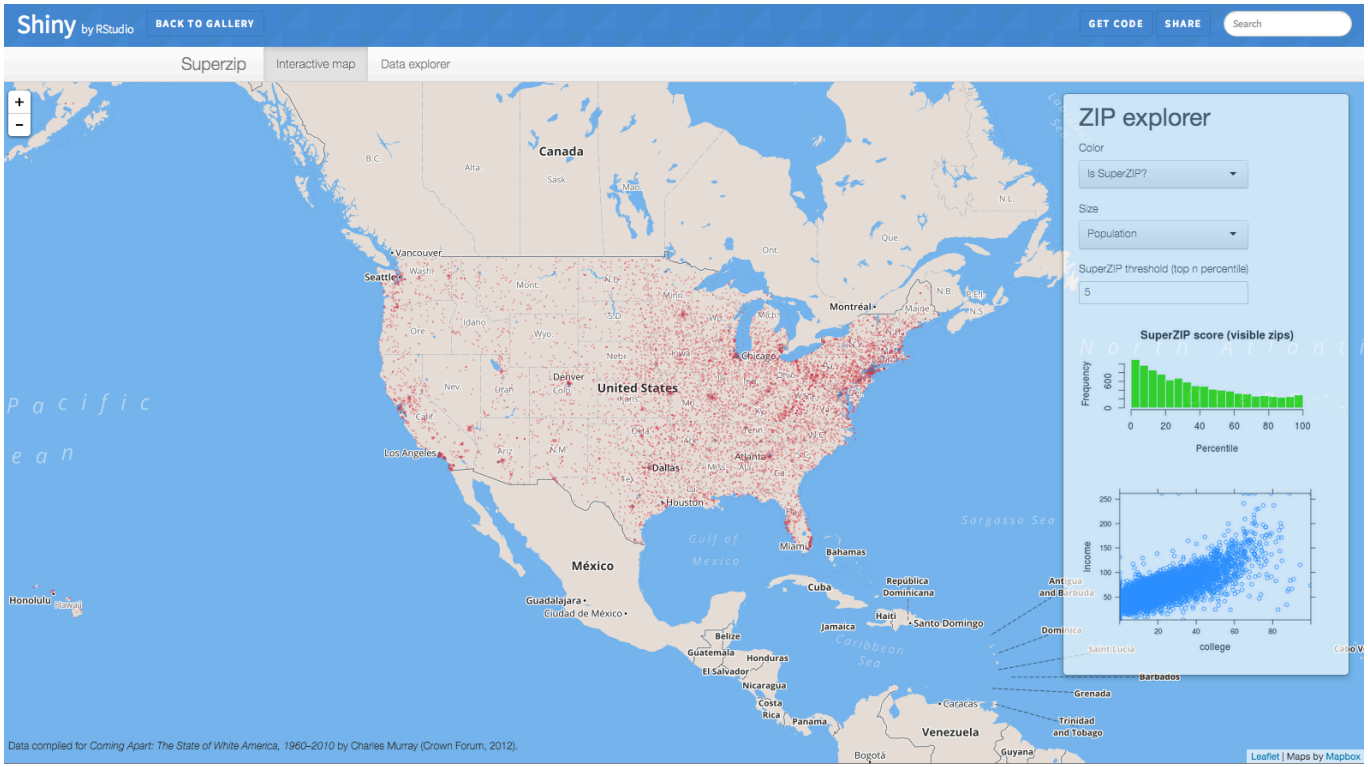
http://ggvis.rstudio.com/



Shiny

Create interactive applications

http://shiny.rstudio.com/



Acknowledgements

This module is derived from resources by

- **Hadley Wickham** (under the under the Creative Commons Attribution-Noncommercial 3.0 United States License)