# Data Visualization in R

Clay Ford

Spring 2020

# About ggplot2

- ▶ Developed by Hadley Wickham in 2005.
- ▶ Implements the graphics scheme described in the book *The Grammar of Graphics* by Leland Wilkinson.
- ▶ Uses a standardized system of syntax that makes it easy(-ish) to learn
- ▶ It takes care of a lot fiddly details such as colors, scales, and legend placement
- ▶ It does not do 3D or interactive graphics

# The Grammar of Graphics

The *Grammar of Graphics* boiled down to 5 bullets, courtesy of Wickham (2016, p. 4):

- a statistical graphic is a mapping from data to **aes**thetic attributes (location, color, shape, size) of **geom**etric objects (points, lines, bars).

- the geometric objects are drawn in a specific **coord**inate system.

- **scale**s control the mapping from data to aesthetics and provide tools to read the plot (ie, axes and legends).

- the plot may also contain **stat**istical transformations of the data (means, medians, bins of data, trend lines).

- **facet**ing can be used to generate the same plot for different subsets of the data.

# Basic ggplot2 syntax

**Specify data, aesthetics and geometric shapes**

```
ggplot(data, aes(x=, y=, color=, shape=, size=)) +
geom_point(), or geom_histogram(), or geom_boxplot(), etc.
```

- ▶ This combination is very effective for exploratory graphs.

- ▶ The data must be a data frame.

- ▶ The `aes()` function maps columns of the data frame to aesthetic properties of geometric shapes to be plotted.

- ▶ `ggplot()` defines the plot; the geoms show the data; each component is added with +

- ▶ Some examples should make this clear

# The Albemarle county homes data

We'll demonstrate `ggplot2` using the Albemarle County real estate data, which was downloaded from Office of Geographic Data Services.

Some variables of interest:

- total value of home (totalvalue)
- finished square feet (finsqft)
- number of bedrooms (bedroom)
- city where house is located (city)
- whether or not house has a fireplace (fp)

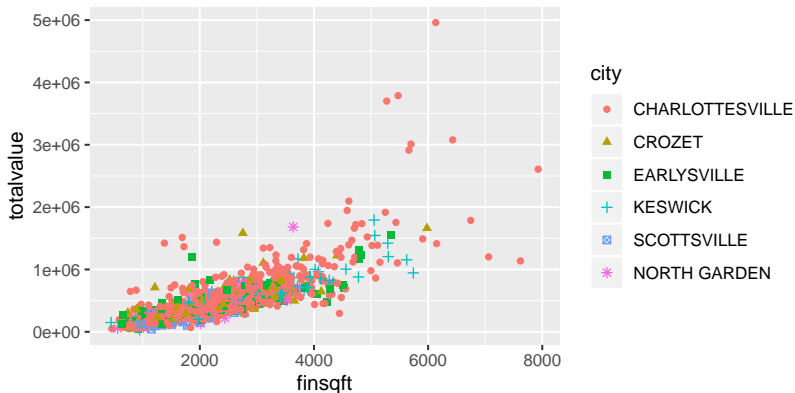Note: the following examples use a sample of the homes data.

# scatter plot colored by city

```r
library(ggplot2) # or library(tidyverse)
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color=city)) + geom_point()
```
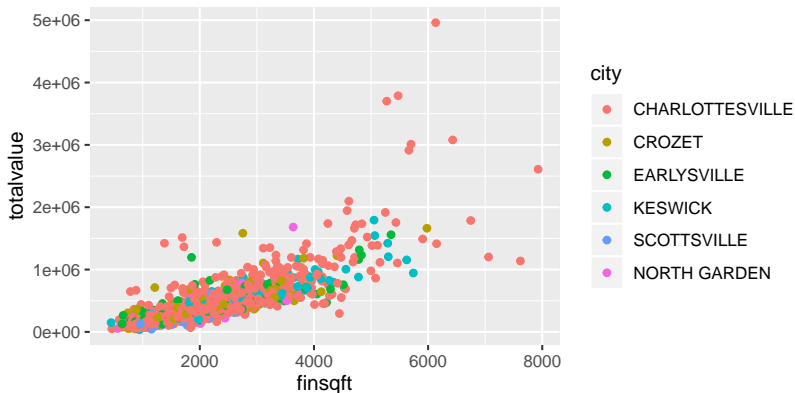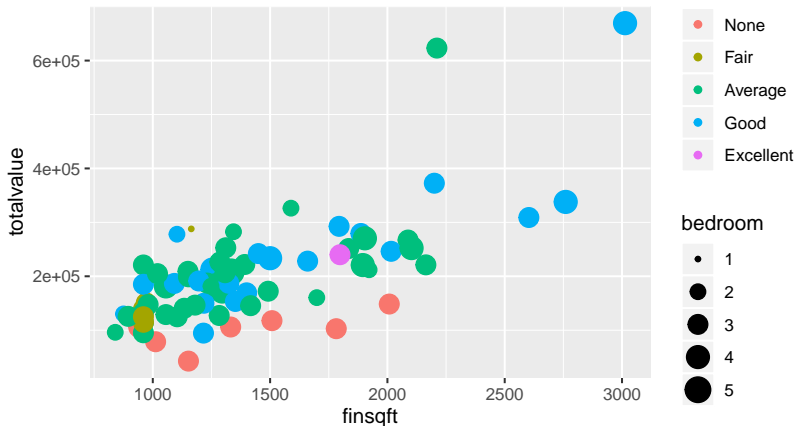
# scatter plot colored and shaped by city

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color=city, shape=city)) +
  geom_point()
```
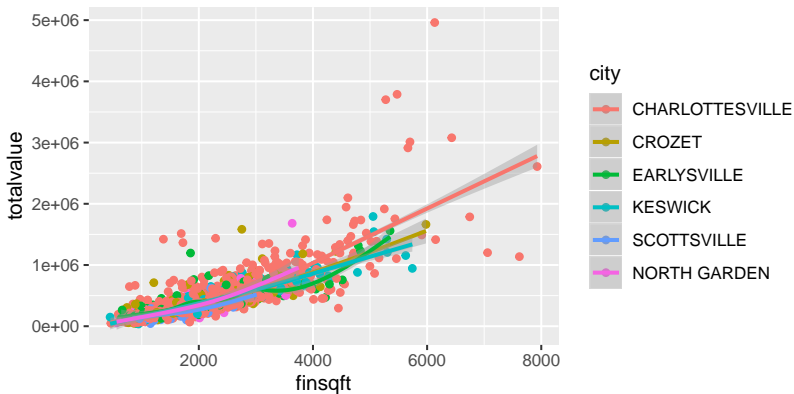
# scatter plot colored by condition, sized by Bedroom

```
ggplot(filter(homes, city == "SCOTTSVILLE"),
       aes(x=finsqft, y=totalvalue,
           color=condition, size=bedroom)) +
  geom_point()
```
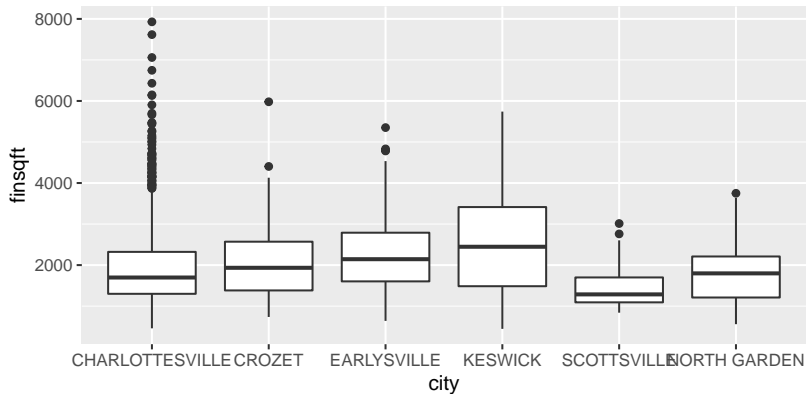
# add multiple geoms (points and smooth line)

```
ggplot(homes, aes(x=finsqft, y=totalvalue, color=city)) +
  geom_point() +
  geom_smooth()
```

# boxplot (statistical transformation)

```
ggplot(homes, aes(x=city, y=finsqft)) +
  geom_boxplot()
```

# Moving beyond ggplot + geoms

- ▶ A natural next step in exploratory graphing is to create plots of subsets of data. These are called facets in ggplot2.

- ▶ Use `facet_wrap()` if you want to facet by one variable and have ggplot2 control the layout. Example:

    - ▶ `+ facet_wrap( ~ var)`

- ▶ Use `facet_grid()` if you want to facet by one and/or two variables and control layout yourself.
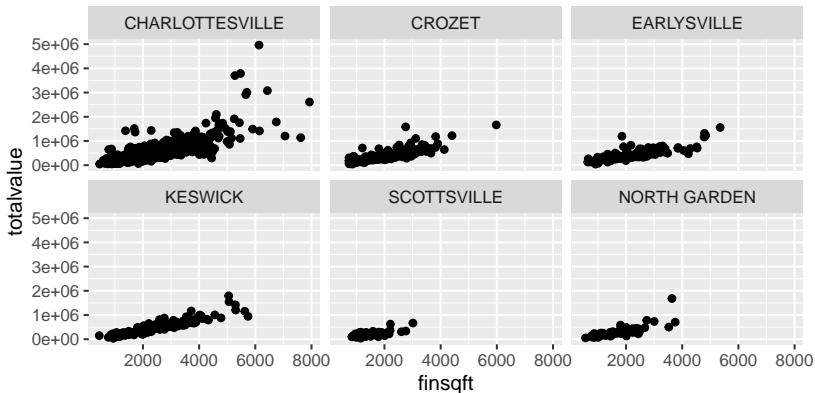
Examples:
`+ facet_grid(. ~ var1)` - facets in columns
`+ facet_grid(var1 ~ .)` - facets in rows
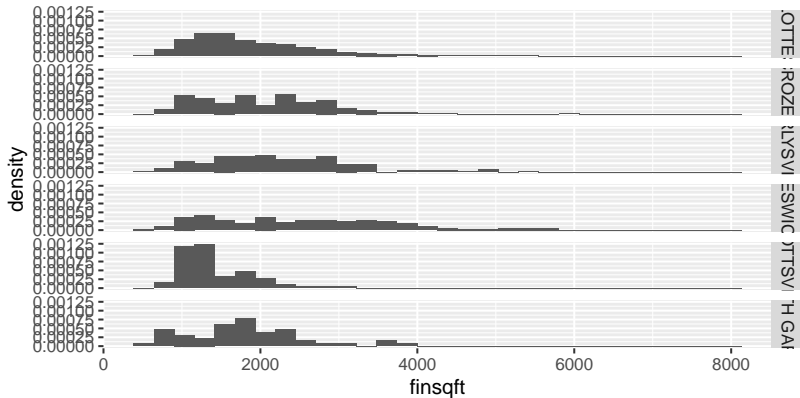`+ facet_grid(var1 ~ var2)` - facets in rows and columns

# facet_wrap

```
ggplot(homes, aes(x=finsqft, y=totalvalue)) +
  geom_point() + facet_wrap(~ city)
```

# facet_grid (histograms)

```
ggplot(homes, aes(x=finsqft, y = stat(density))) +
  geom_histogram() + facet_grid(city ~ .)
```
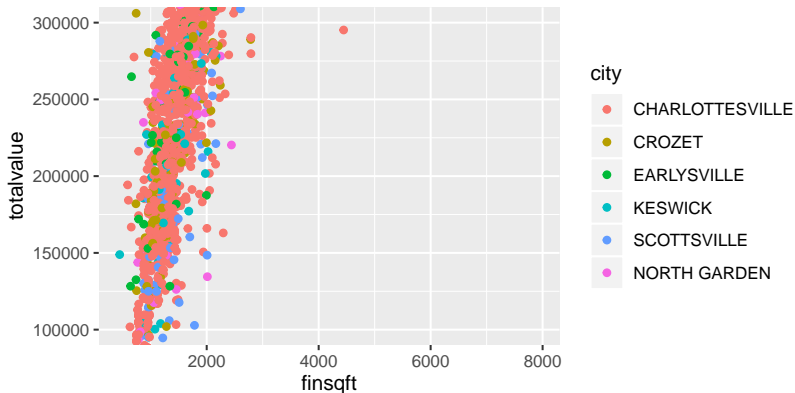
# Modifying the coordinate system

- `coord_cartesian` allows us to zoom in on a plot, as if using magnifying glass
- `coord_fixed` allows us to control "aspect ratio"
- `coord_flip` allows us to flip the x and y axis
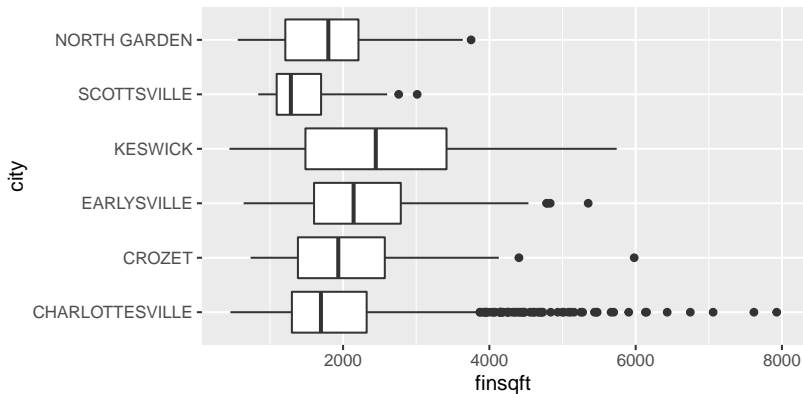
# zoom in on plot

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color=city)) + geom_point() +
  coord_cartesian(ylim = c(1e5, 3e5))
```

# flip coordinate axes

```
ggplot(homes, aes(x=city, y=finsqft)) +
  geom_boxplot() +
  coord_flip()
```

# Customizing scales

▶ Scales control the mapping from data to aesthetics and provide tools to read the plot (ie, axes and legends).

▶ Every aesthetic has a default scale. To modify a scale, use a `scale` function.

▶ All scale functions have a common naming scheme: scale _ name of aesthetic _ name of scale

▶ Examples: `scale_y_continuous`, `scale_color_discrete`, `scale_fill_manual`

▶ Heads up: The documentation for `ggplot2` scale functions will frequently use functions from the `scales` package (also by Wickham)!
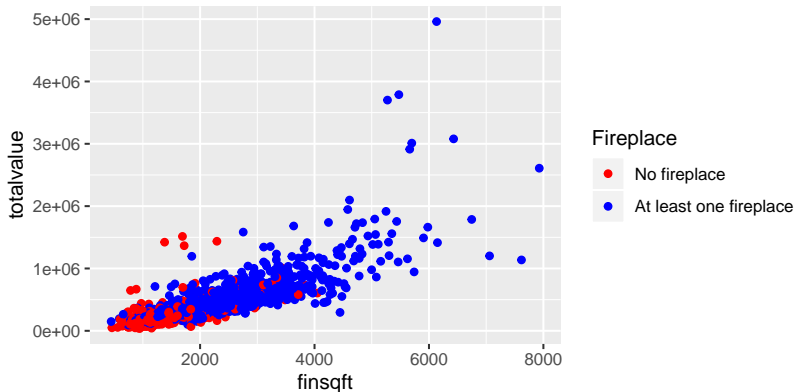
# update scales for x- and y-axis

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color=city)) + geom_point() +
  scale_y_continuous(labels = scales::dollar) +
  scale_x_continuous(labels = scales::comma)
```
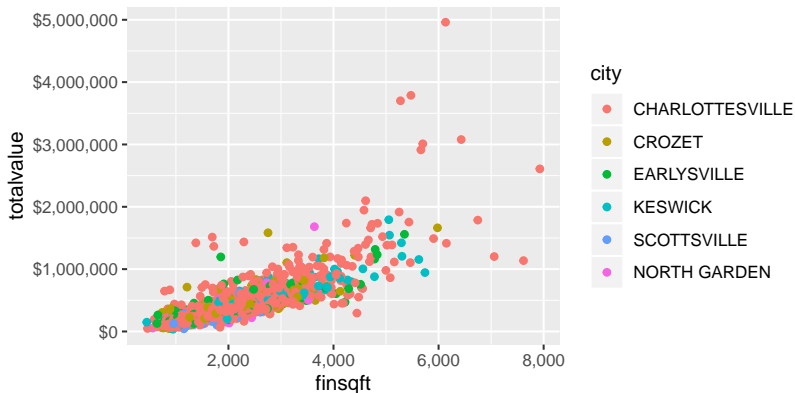
## update scale for color

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color=factor(fp))) + geom_point() +
  scale_color_manual(name="Fireplace",
                     labels = c("No fireplace", "At least o
                     values=c("red","blue"))
```

# Update themes and labels

- ▶ The default ggplot2 theme is excellent. It follows the advice of several landmark papers regarding statistics and visual perception. (Wickham 2016, p. 176)

- ▶ However you can change the theme using ggplot2's themeing system. To date, there are seven built-in themes: `theme_gray` (*default*), `theme_bw`, `theme_linedraw`, `theme_light`, `theme_dark`, `theme_minimal`, `theme_classic`

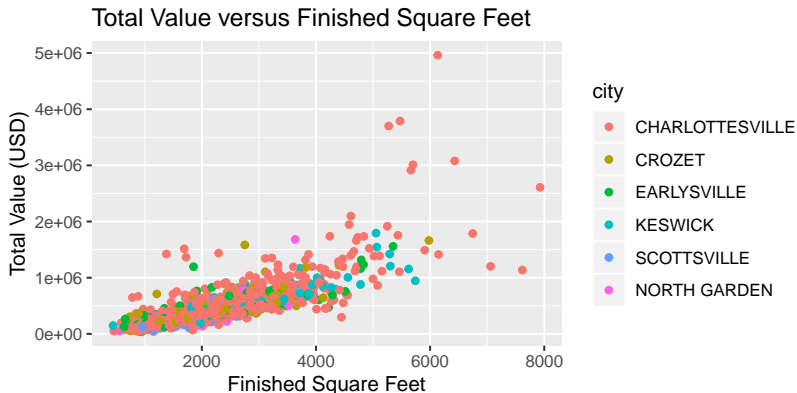- ▶ You can also update axis labels and titles using the `labs` function.

# change theme

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color = city)) + geom_point() +
  theme_minimal()
```
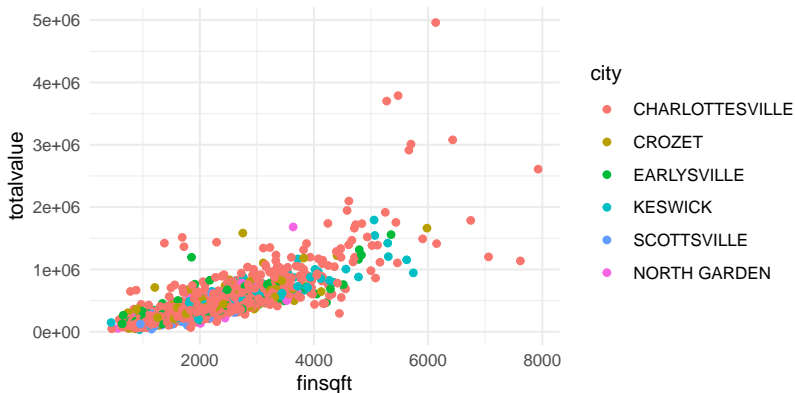
# update labels

```
ggplot(homes, aes(x=finsqft, y=totalvalue,
                  color = city)) + geom_point() +
  labs(title="Total Value versus Finished Square Feet",
       x="Finished Square Feet", y="Total Value (USD)")
```

# ggplot2 - some tips

- ▶ Can do a lot with `ggplot(data, aes())` + geom!
- ▶ Data must be a data frame (not a matrix or collection of vectors)
- ▶ The `ggplot2` documentation has many good examples
- ▶ Prepare to invest some time if you want master ggplot2; the RStudio ggplot2 cheat sheet can help.

Let's go to R!

# References and further study

- Chang, W. (2013), *R Graphics Cookbook*, O'Reilly.

- Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis* (2nd ed), Springer.

- Wickham, H. and Grolemund G. (2017), *R for Data Science*. O'Reilly. http://r4ds.had.co.nz/

**ggplot2 cheat sheet**
https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf

**Cookbook for R - Graphs**
http://www.cookbook-r.com/Graphs/

**Official ggplot2 web site**
https://ggplot2.tidyverse.org/

# Thanks for coming

- ▶ For statistical consulting: statlab@virginia.edu

- ▶ Sign up for more workshops or see past workshops:
  http://data.library.virginia.edu/training/

- ▶ Register for the Research Data Services newsletter to be
  notified of new workshops:
  http://data.library.virginia.edu/newsletters/