Data Visualisation with R using ggplot

Chipo Zidana

Department of Mathematics and Statistical Sciences, **BIUST**

What is Data visualisation?

- **Data visualisation** practice of translating information into a visual context
- a step in data science process
- It uses visual elements like **charts, graphs, dashboards, info graphics and maps**
- **Goal**:identify patterns, trends and outlier in data
- **Big data is here!!!** call for better tools

Tell your story right

- Marry the data and the visuals
- delicate balancing act between form and function
- its an art of great analysis and story telling

Who data Visualisation?

ME,ALL  **AND YOU**

Data Talks

Why Data Visualisation

- "A good sketch is better than a long speech."
- Sketching out our data by visualizing is more impactful than simply describing the patterns and trends we find.
- "The simple graph has brought more information to the data analyst's mind than any other device." — **John Tukey**
- it's easier to learn from something that we can see rather than read
- **We need a powerful too!!**In R data visualisation is a snap using

1. Base R
2. Lattice
3. Grid
4. **ggplot2**

Why ggplot2

- Uses the grammar of graphics - Easy and layers tell the complete story
- "It can do quick-and-dirty and complex"-Mendy Mejia
- Easy superposition, facetting, Automatic legends, colors, etc.ie Multivariate exploration is greatly simplified.
- Store any ggplot2 object for modification or future recall. Super useful for packages.
- Lots of users (less bugs, more help on Stack Overflow).
- Lots of extensions.ggtheme, ggrepel etc
- flexible- Nice saving option
- Plots can evolve and devolve with minimal changes

HOW in R?

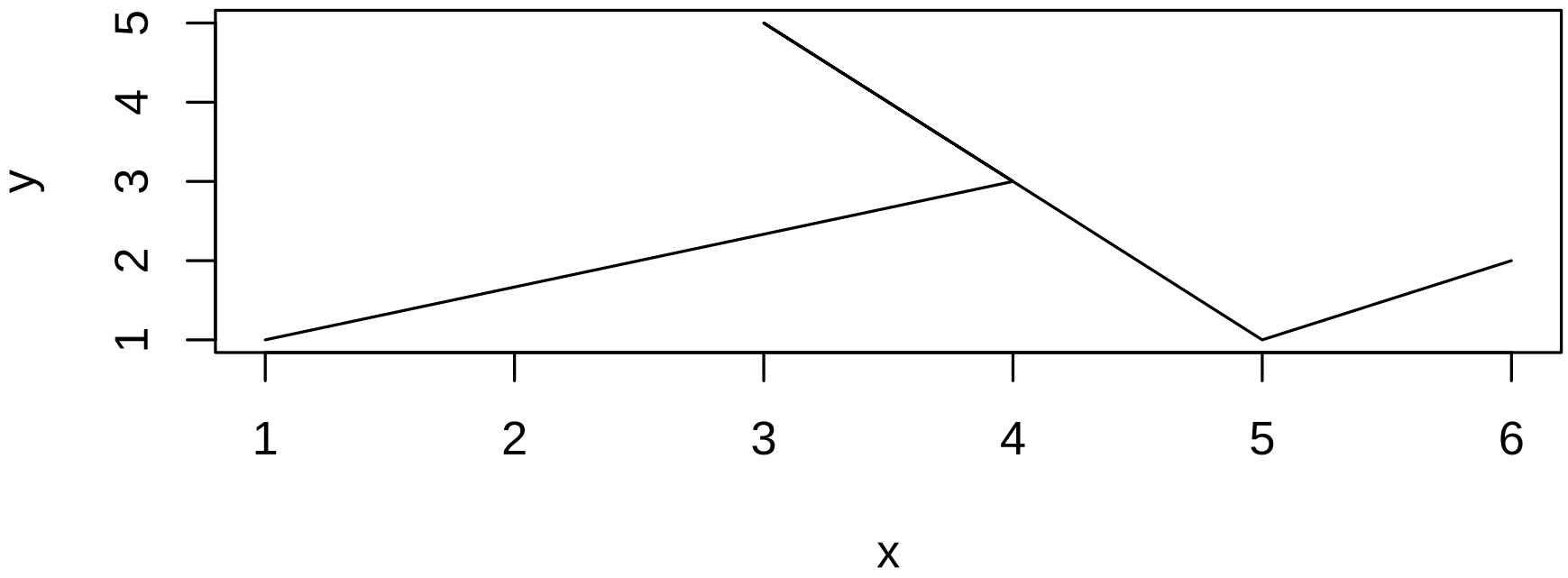
```
#install.packages("tidyverse")
library(ggplot2)
```

What does ggplot stand for?

- A Grammar of graphics
- A grammar of graphics defines the rules of structuring mathematics and aesthetic elements into a meaningful graph.
- Leland Wilkinson (2005) designed the grammar upon which ggplot2 is based.
- **ggplot2** package follows and describes data in the graph aspects

General aspects/What is in Graph?

- there are many type of visualisations but we can generalise as follows



GGplot2 Grammar aspects/Layers

1. **Data**: variables mapped to aesthetic features of the graph.
2. **Geoms**: objects/shapes on the graph.
3. **Stats**: statistical transformations that summarize data,(e.g mean, confidence intervals).
4. **Scales**: mappings of aesthetic values to data values. Legends and axes visualize scales.
5. **Coordinate systems**: the plane on which data are mapped on the graphic.
6. **Faceting**: splitting the data into subsets to create multiple variations of the same graph (paneling).

- **ANY Plot can be described uniquely as a combination of these aspects parameters with layers**.

ggplot2 Aspects: Plot = data + Aesthetics + Geometry

- *Data* -data frame/table
- *Aesthetics* - indicate x and y variables, control colour, size and shape
- *geometry* - defines the type of graphics (histogram, box plot, line plot, density plot, dot plot, ...)

$$
\begin{aligned}
-\mathbf{ggplot}(data =< \mathbf{DATA} >) + \\
< \mathbf{GEOM\ FUNCTION} >(\\
mapping = aes(< MAPPINGS >), \\
stat =< STAT >, \\
position =< POSITION >) + \\
< \mathbf{COORDINATE\ FUNCTION} > + \\
< \mathbf{FACET\ FUNCTION} >
\end{aligned}
$$

DATA

- Data should be tidy/ clean (**Use dplyr to re arrange data tidy -disadvantage**)
- Tidy data is easier to work and work with ie:

1. Each variables are represent a column
2. each row present an observation
3. each value has a unique column and row

- import and tidy your data before visualising;
- **DATA CLEANING IS VITAL**

TIDY DATA

```
## # A tibble: 437 x 28
##      PID county  state  area poptotal popdensity popwhite p
##    <int> <chr>   <chr> <dbl>    <int>      <dbl>    <int>
##  1   561 ADAMS   IL    0.052    66090      1271.    63917
##  2   562 ALEXAN~ IL    0.014    10626       759      7054
##  3   563 BOND    IL    0.022    14991       681.    14477
##  4   564 BOONE   IL    0.017    30806      1812.    29344
##  5   565 BROWN   IL    0.018     5836       324.     5264
##  6   566 BUREAU  IL    0.05     35688       714.    35157
##  7   567 CALHOUN IL    0.017     5322       313.     5298
##  8   568 CARROLL IL    0.027    16805       622.    16519
##  9   569 CASS    IL    0.024    13437       560.    13384
## 10   570 CHAMPA~ IL    0.058   173025      2983.   146506
## # ... with 427 more rows, and 19 more variables: popasian <
## #   popother <int>, percwhite <dbl>, percblack <dbl>, perca
## #   percasian <dbl>, percother <dbl>, popadults <int>, perc
```

The ggplot() structure

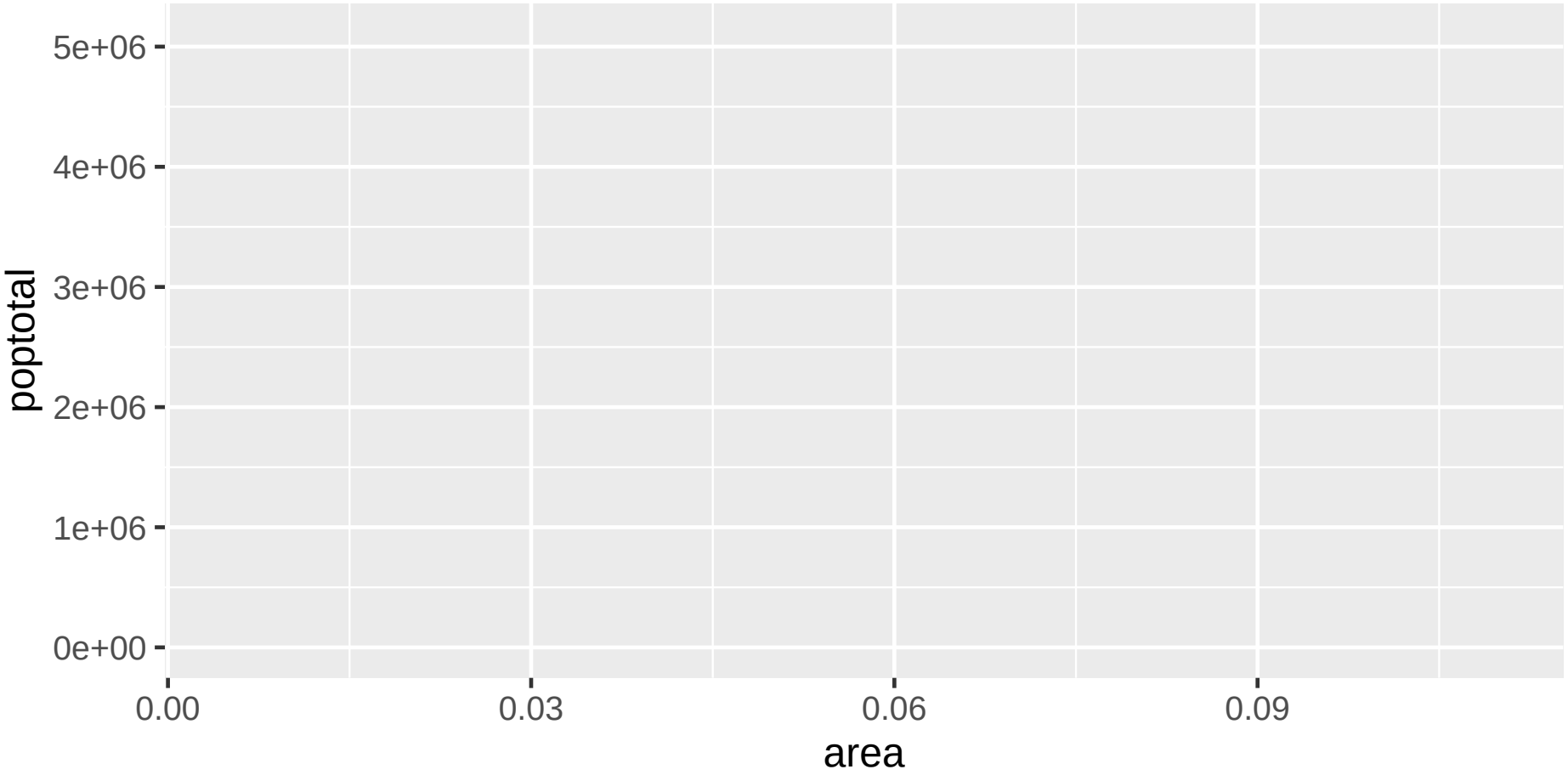
- In the ggplot() function - specify the data set that holds the variables we will be mapping to aesthetics, the visual properties of the graph.
- The data set must be a data.frame object.
- **ggplot(data, aes(x=xvar, y=yvar))**

1. data- data frame
2. x and y: aesthetics that position objects on the graph
3. xvar and yvar: names of variables in data mapped to x and y

- **NB:** aes are specified inside aes(), which is itself nested inside of ggplot().
- aesthetics specified inside of ggplot() are inherited by subsequent layer

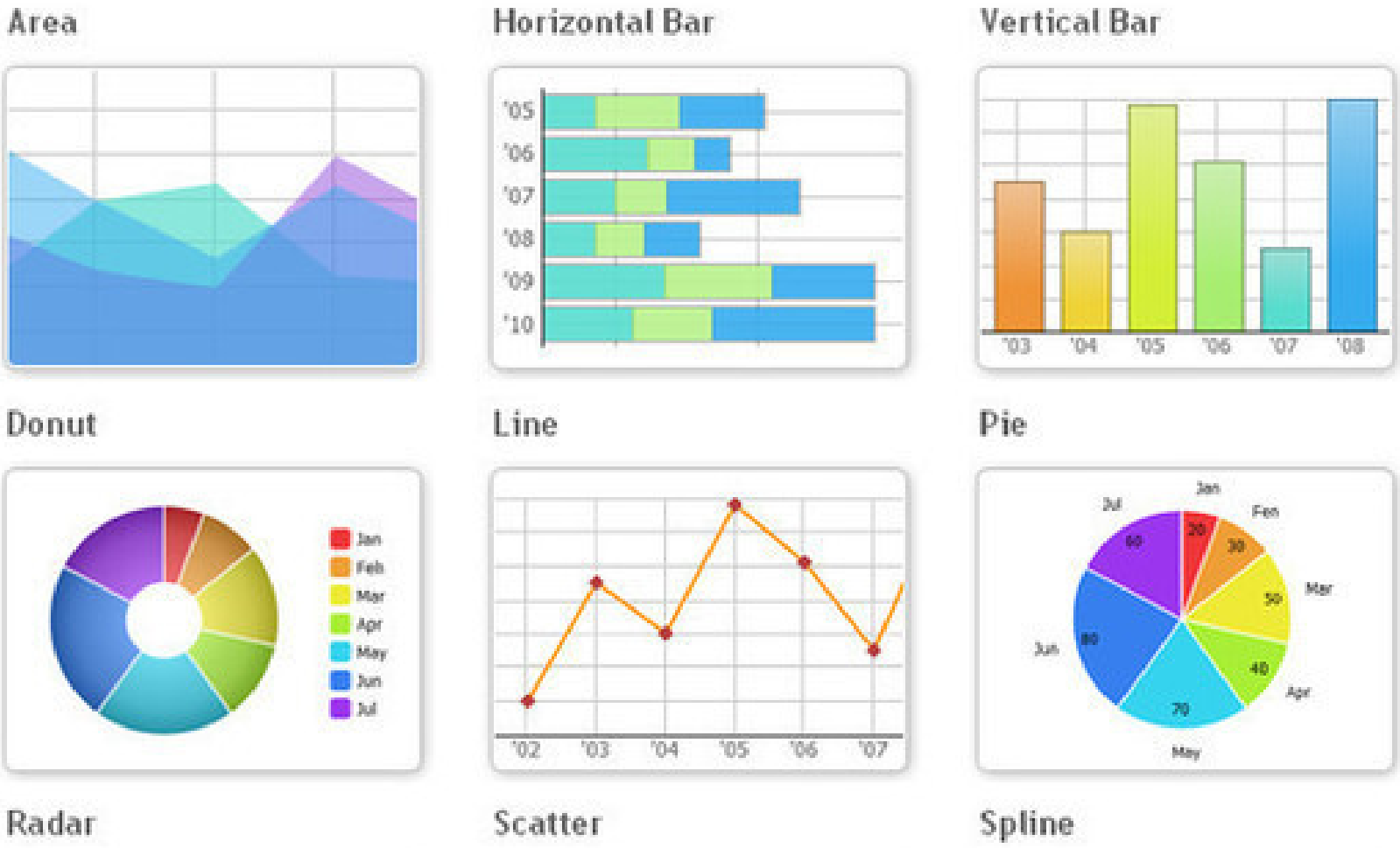
Global GGplot

```
ggplot(data = midwest, aes(x = area, y = poptotal))
```



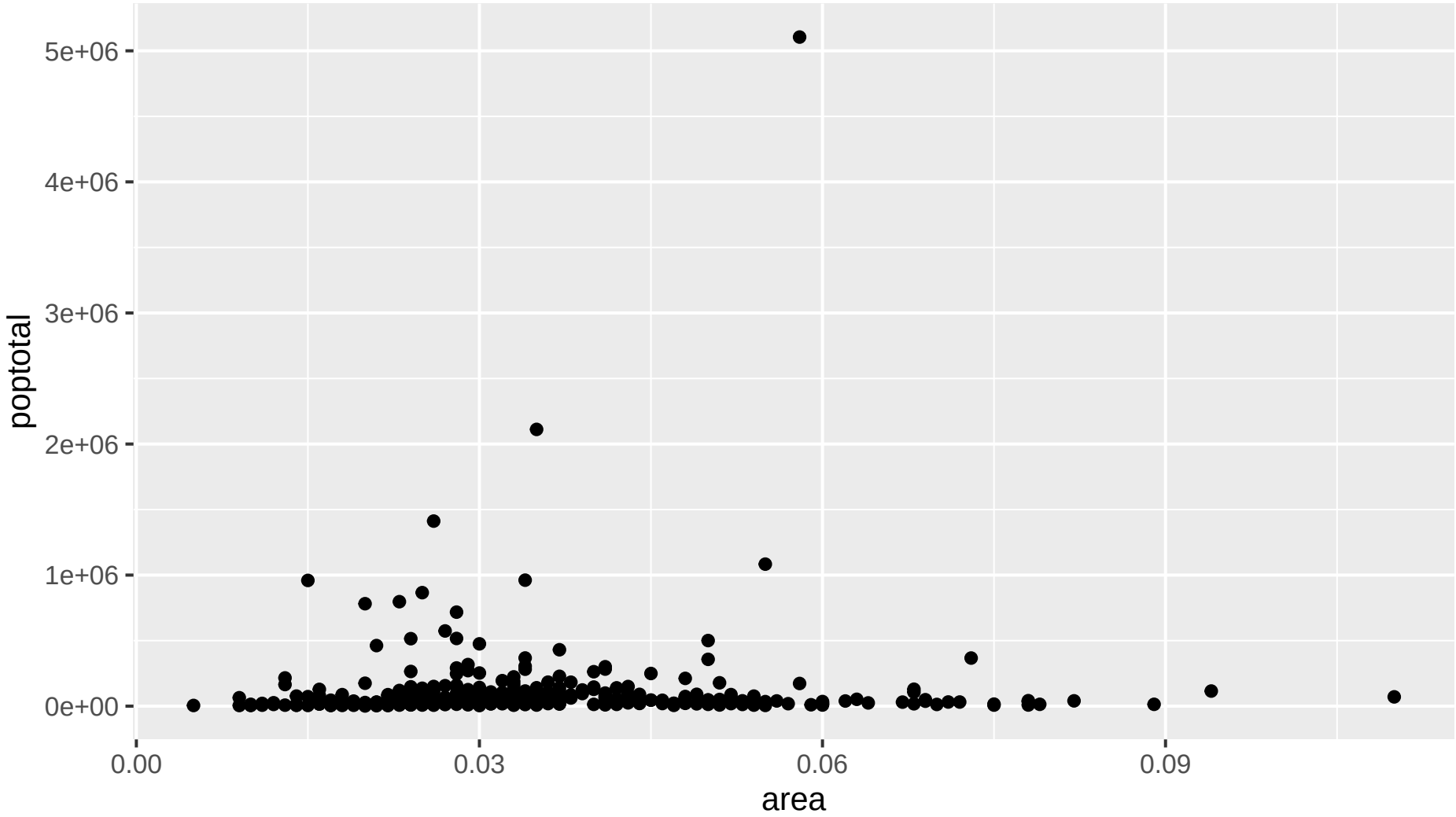
Add Geom function

- define geometries, - compute summary statistics, -
- define what scales to use, - or even change styles.



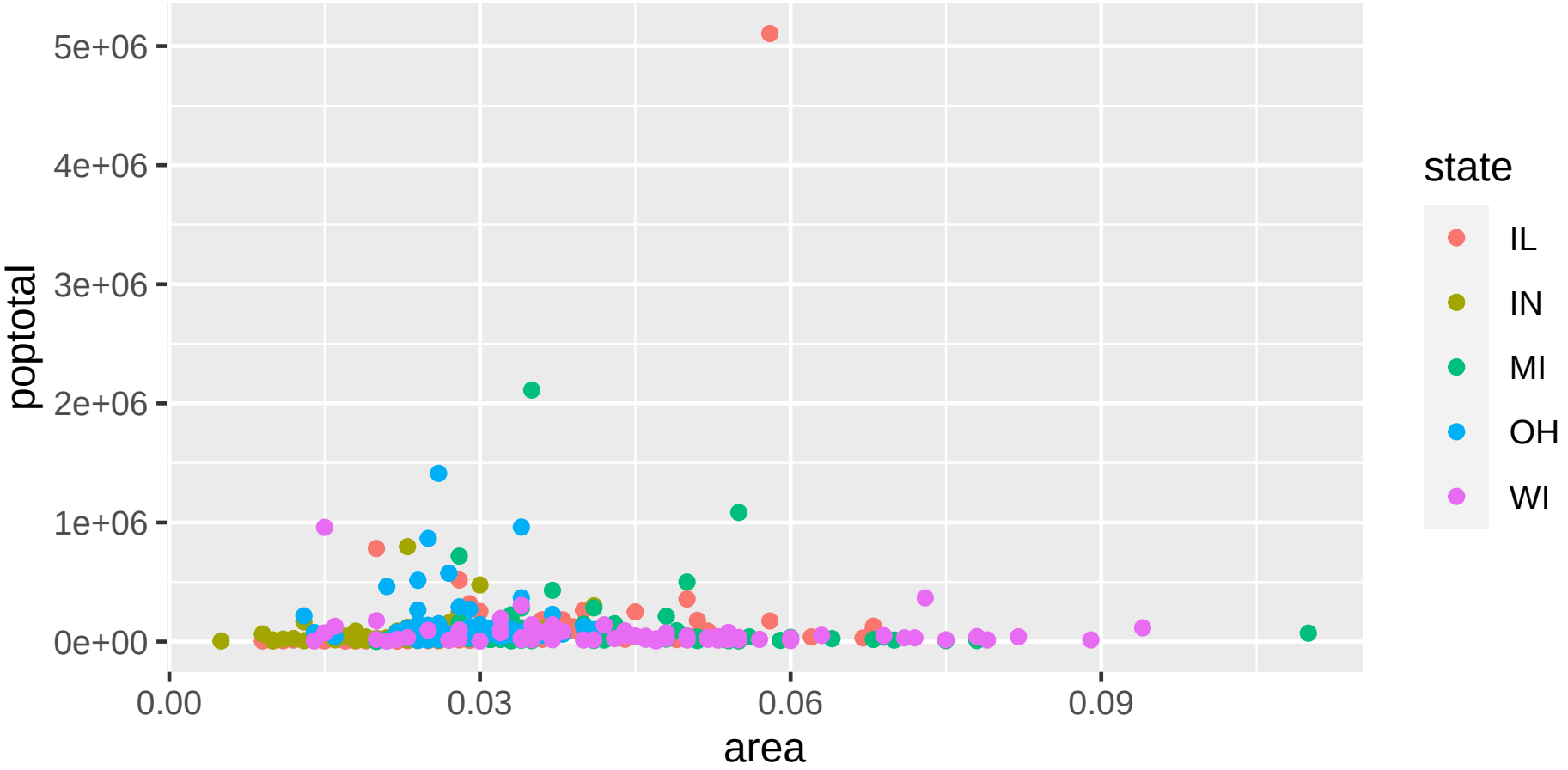
1. Scatter Plot: geom_point()

```
ggplot(data = midwest, aes(x = area, y = poptotal))+
  geom_point()
```



Add colour

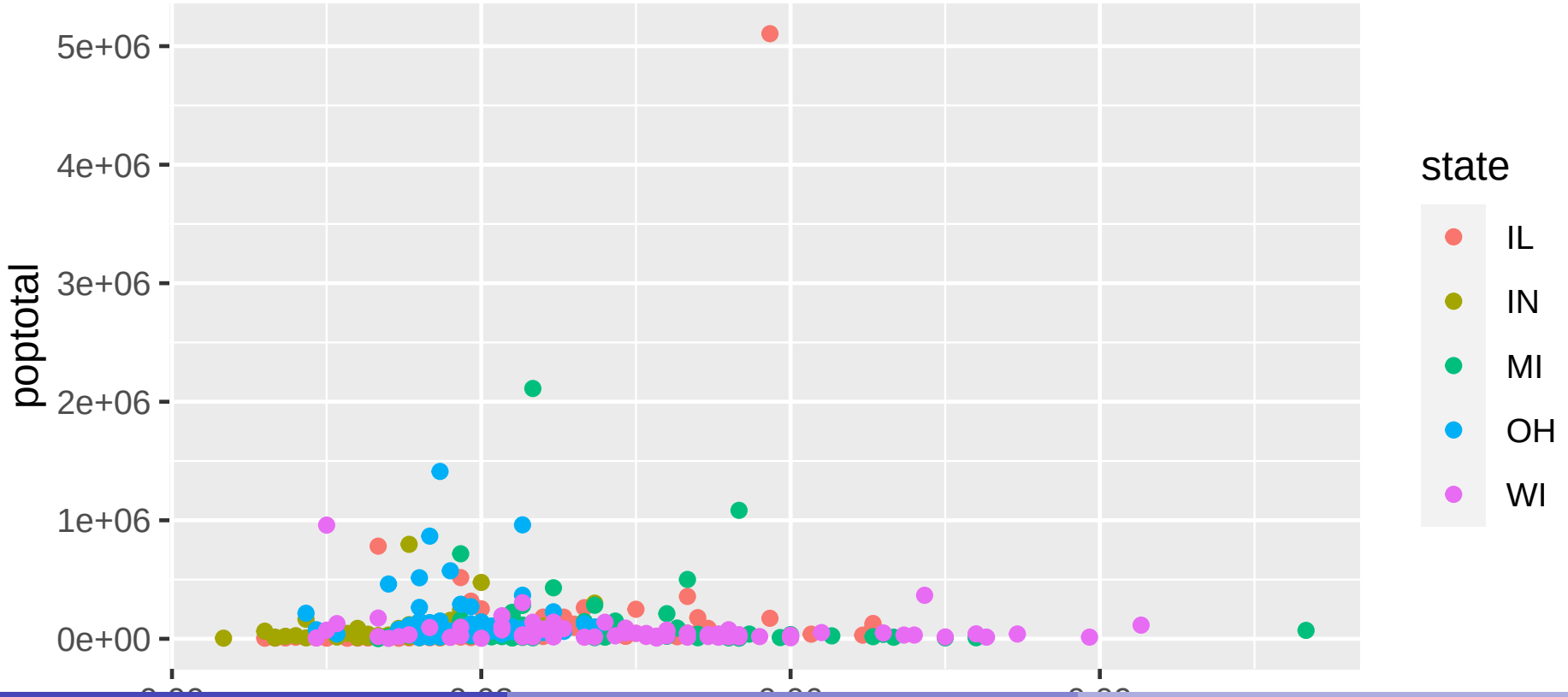
```
ggplot(data = midwest, aes(x = area, y = poptotal,
       color= state)) + geom_point()
```



Custom the graph

- Name the ggplot plot and custom it

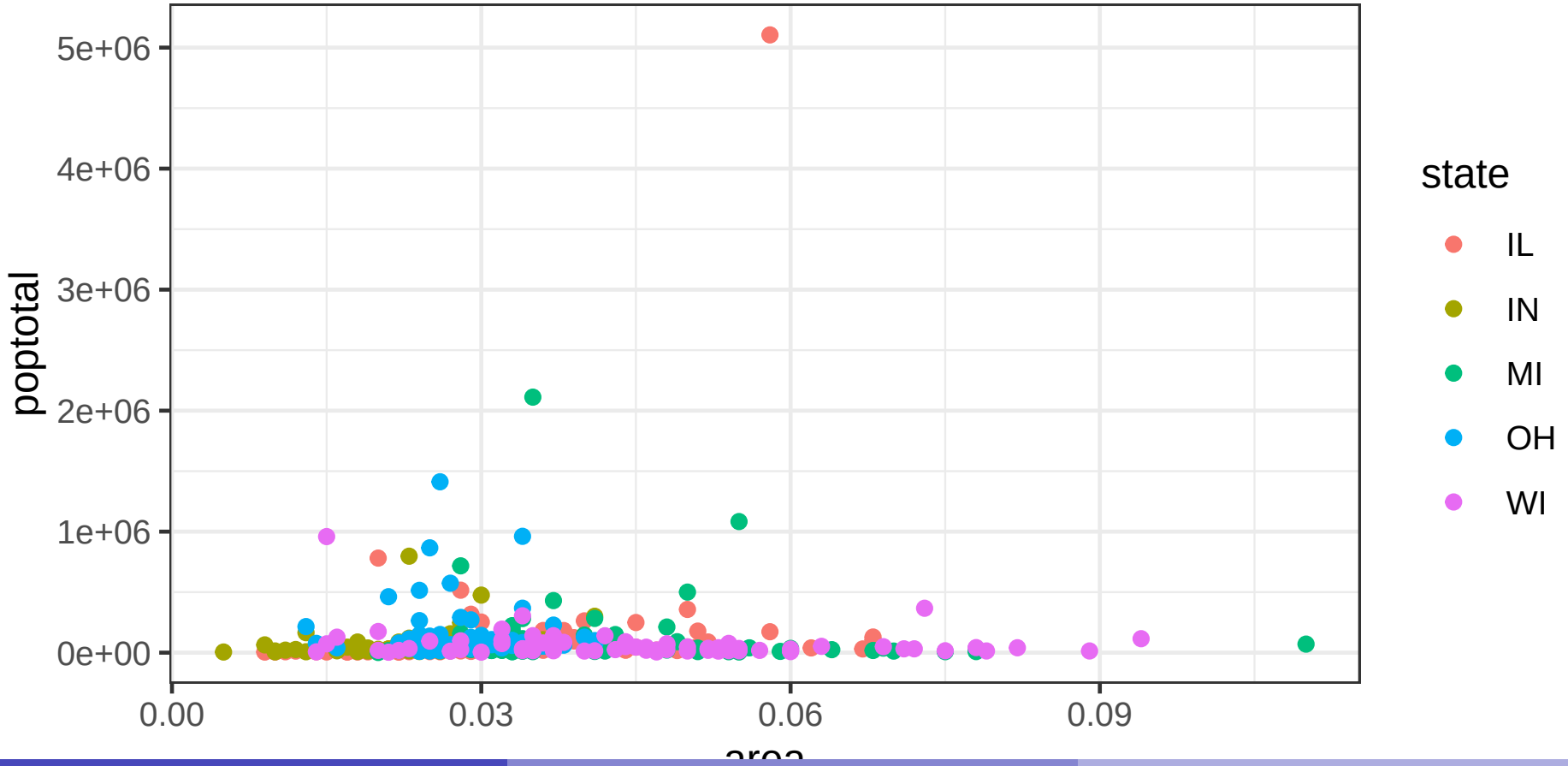
```
g = ggplot(data = midwest, aes(x = area, y = poptotal, color=
  geom_point()
plot(g)
```



Change background

- backgraound themes can be changed to different ways:
- **eg:** theme_bw() / theme_classic()

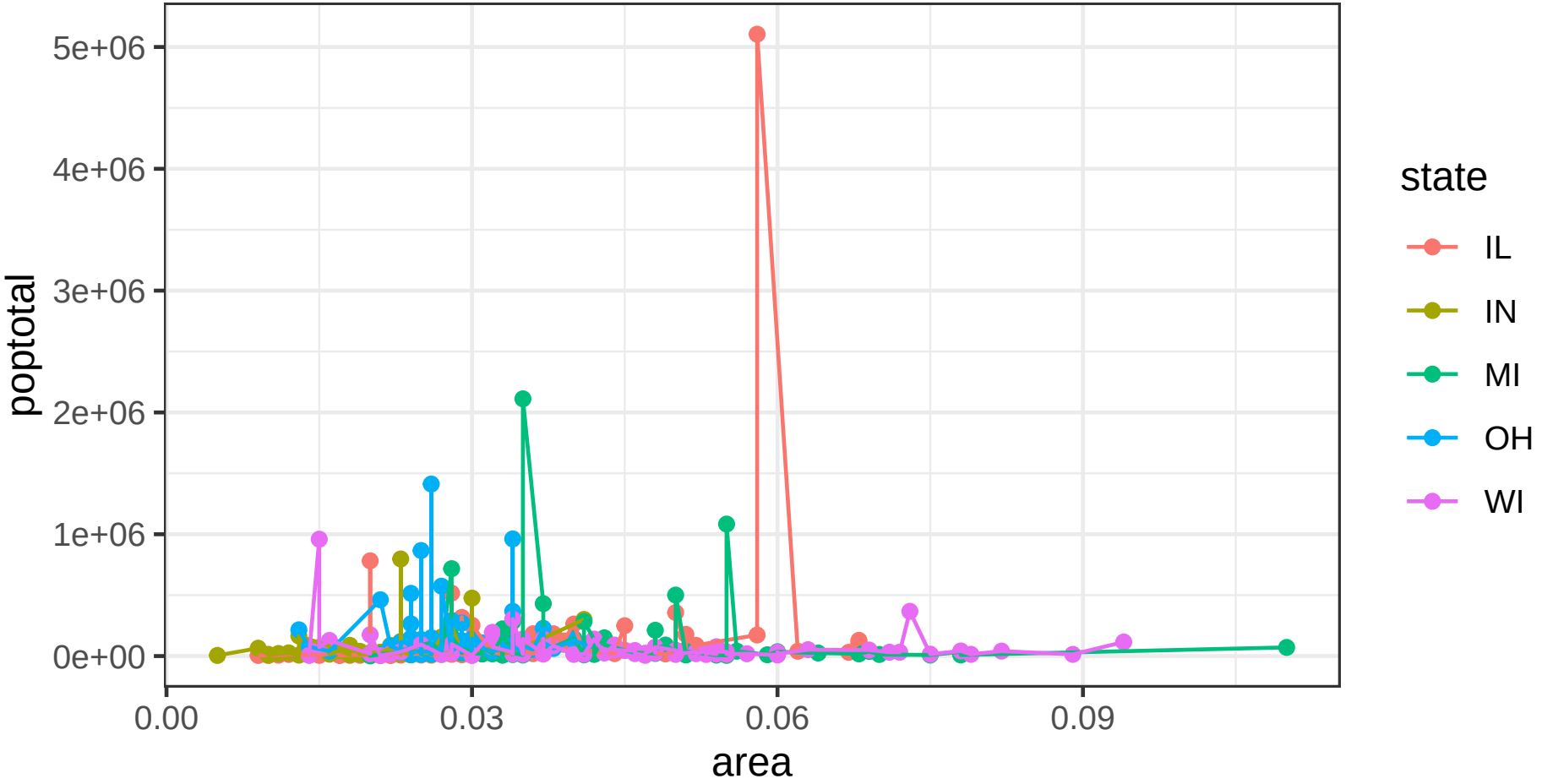
```
g+theme_bw()
```



Add Another Geom function

2. Line graph: geom_line()

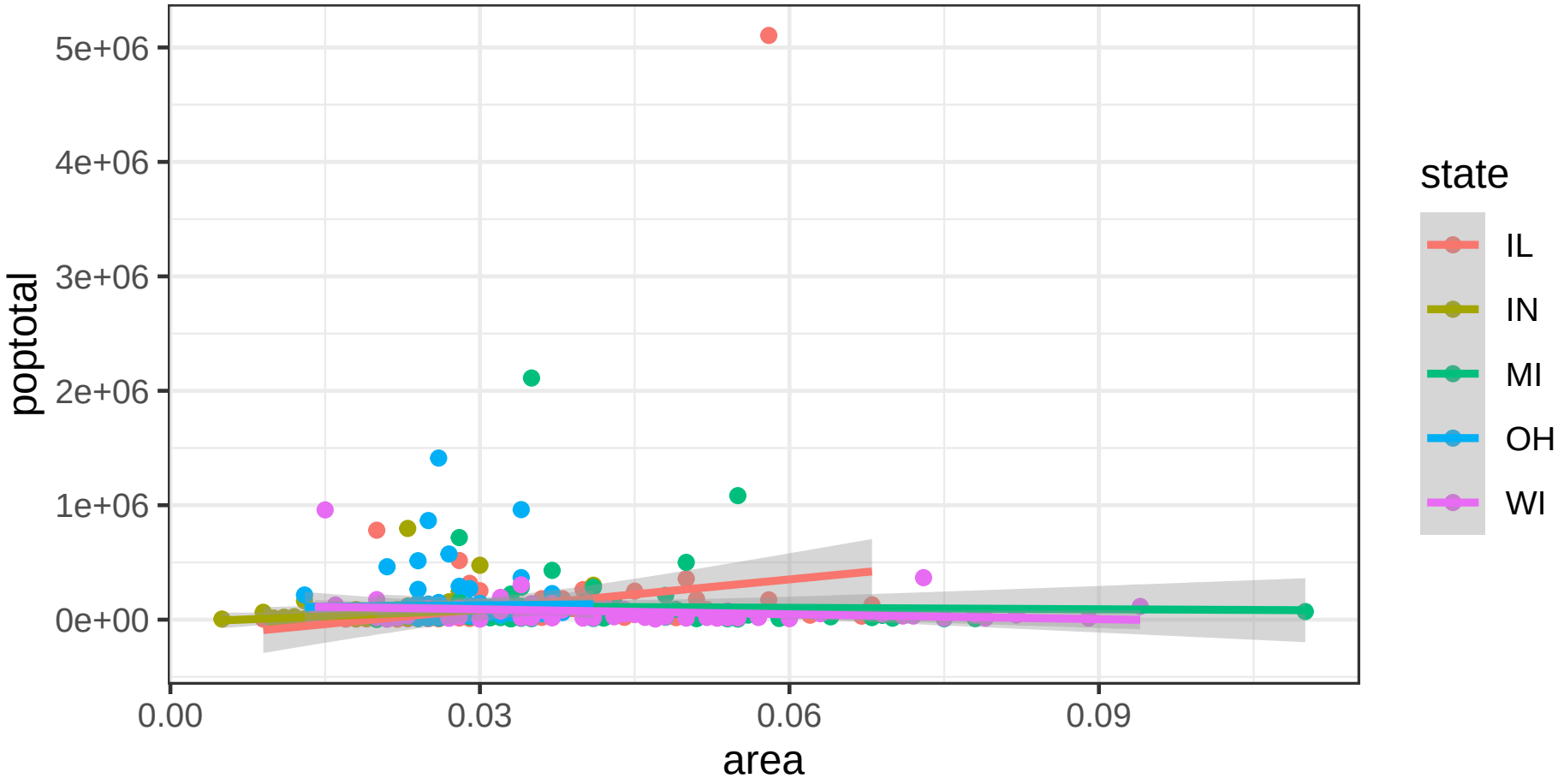
```
g+theme_bw()+geom_line()
```



Add Geom Smooth Layer

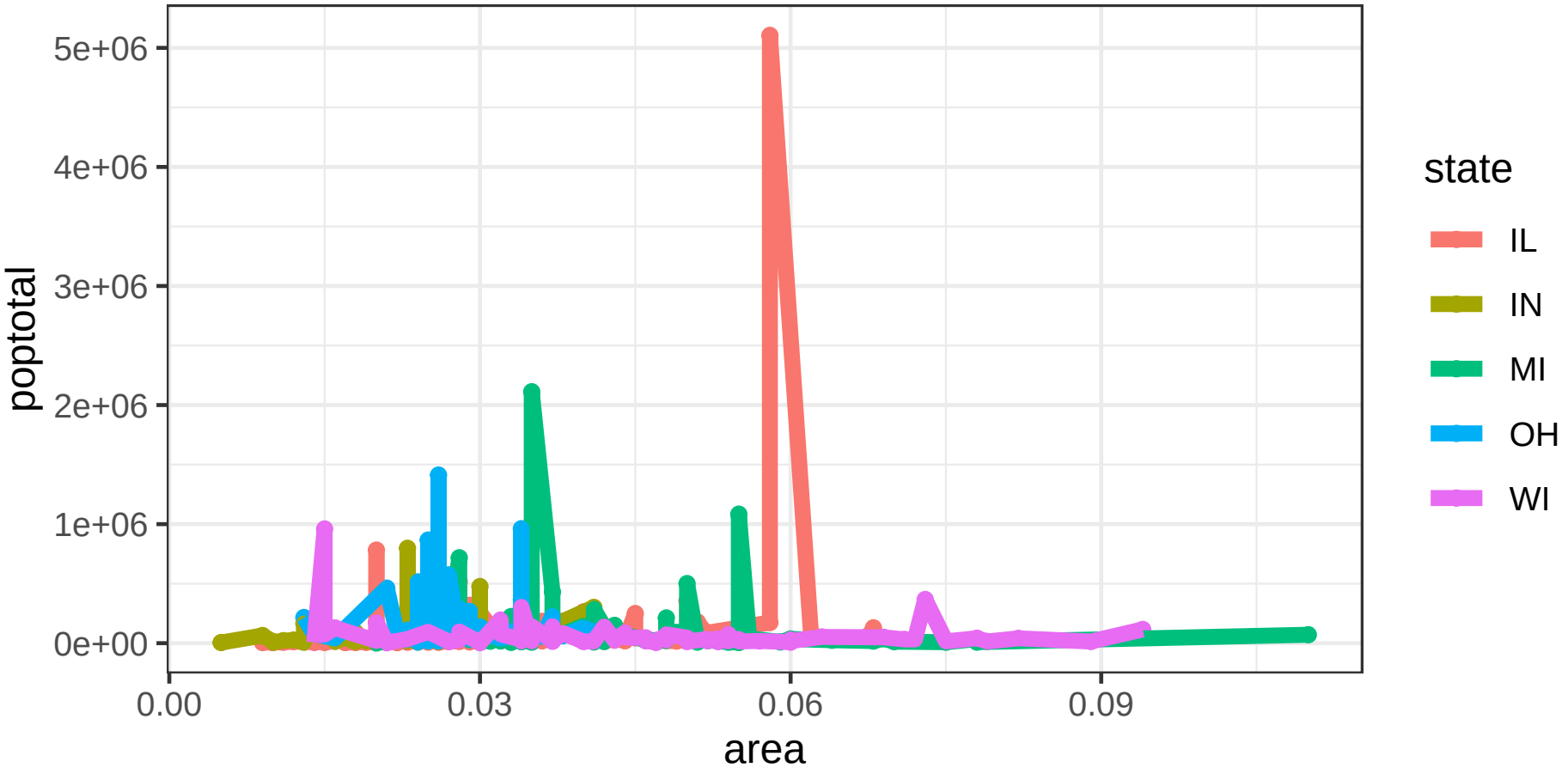
```
g+theme_bw()+geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



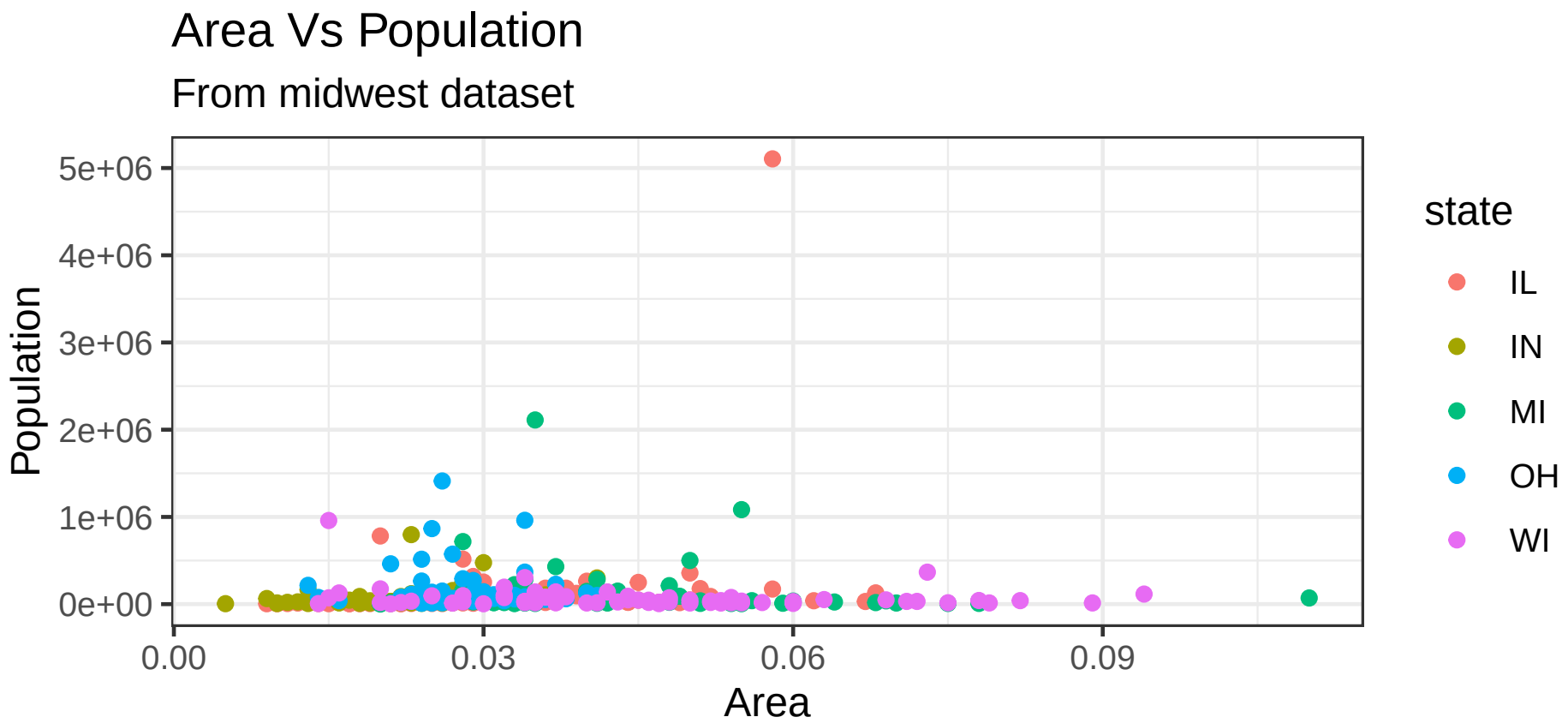
Change size of points and lines

```
g+theme_bw()+geom_line(size =2)
```



Add Title and Axis labels

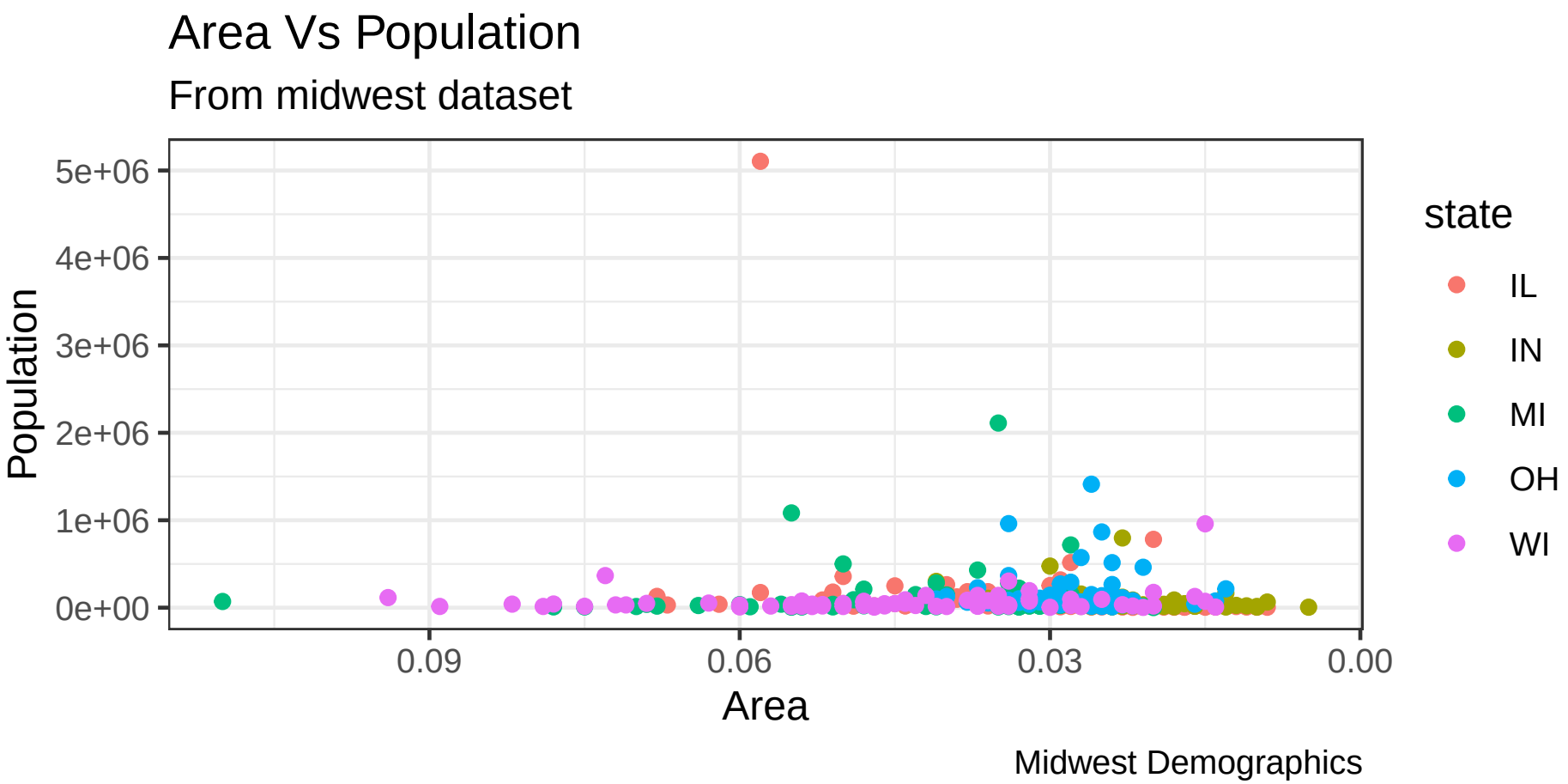
```
gg=g+theme_bw()+ labs(title="Area Vs Population",
    subtitle="From midwest dataset",y="Population",
    x="Area", caption="Midwest Demographics")
gg
```



Scale reservesing

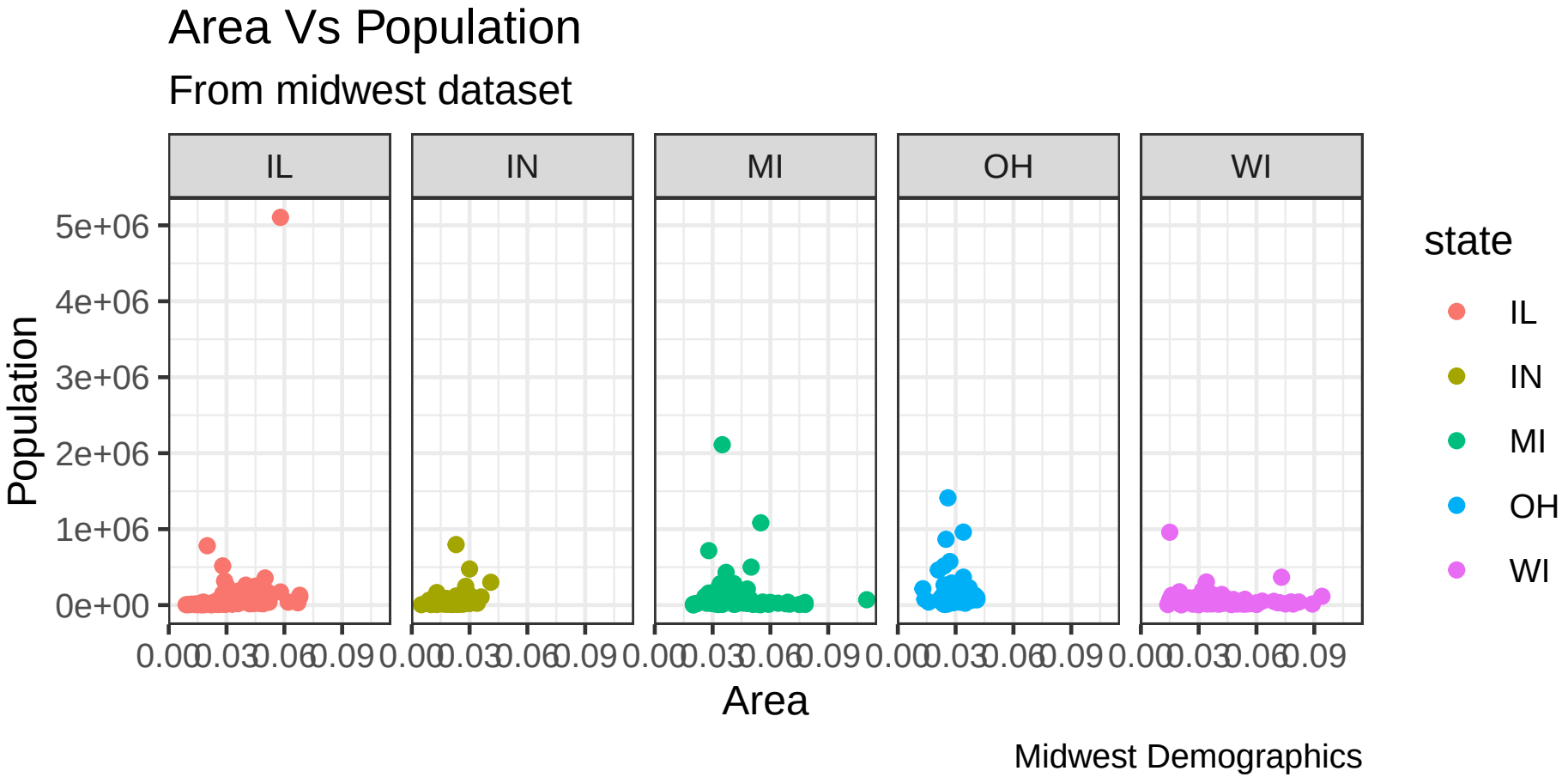
- If you need to reverse the scale, use scale_x_reverse().

```
gg+  scale_x_reverse()
```



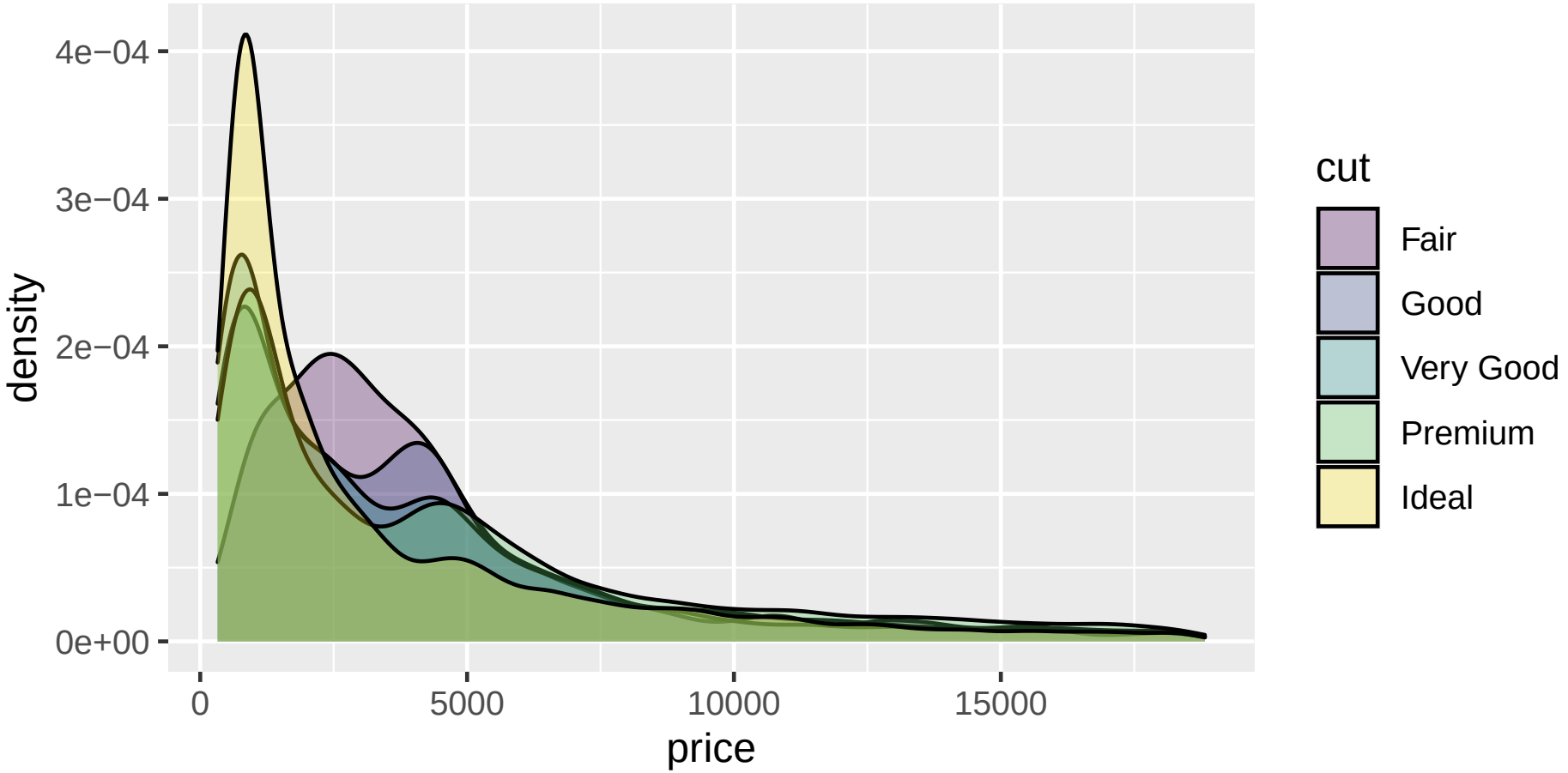
Facets

```
gg+facet_grid(~state)
```



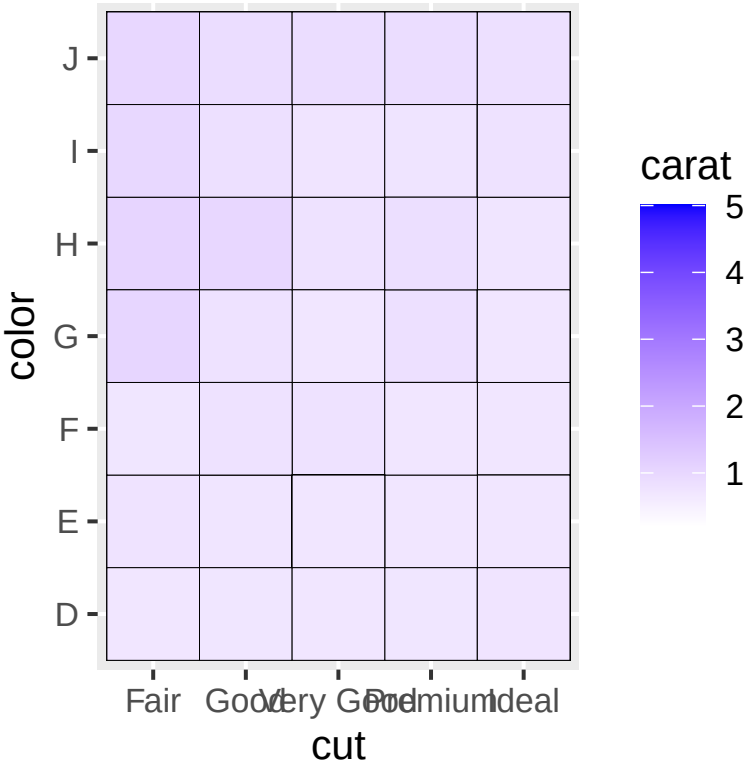
Flexibility

```
data(diamonds)
ggplot(data=diamonds, aes(x=price, fill=cut)) +
  geom_density(alpha=0.3)
```



Heat Map

```
ggplot(diamonds,aes(x = cut, y = color, fill = carat))+
     geom_tile(color = "black") +
    scale_fill_gradient(low = "white",  high = "blue") +
    coord_fixed()
```

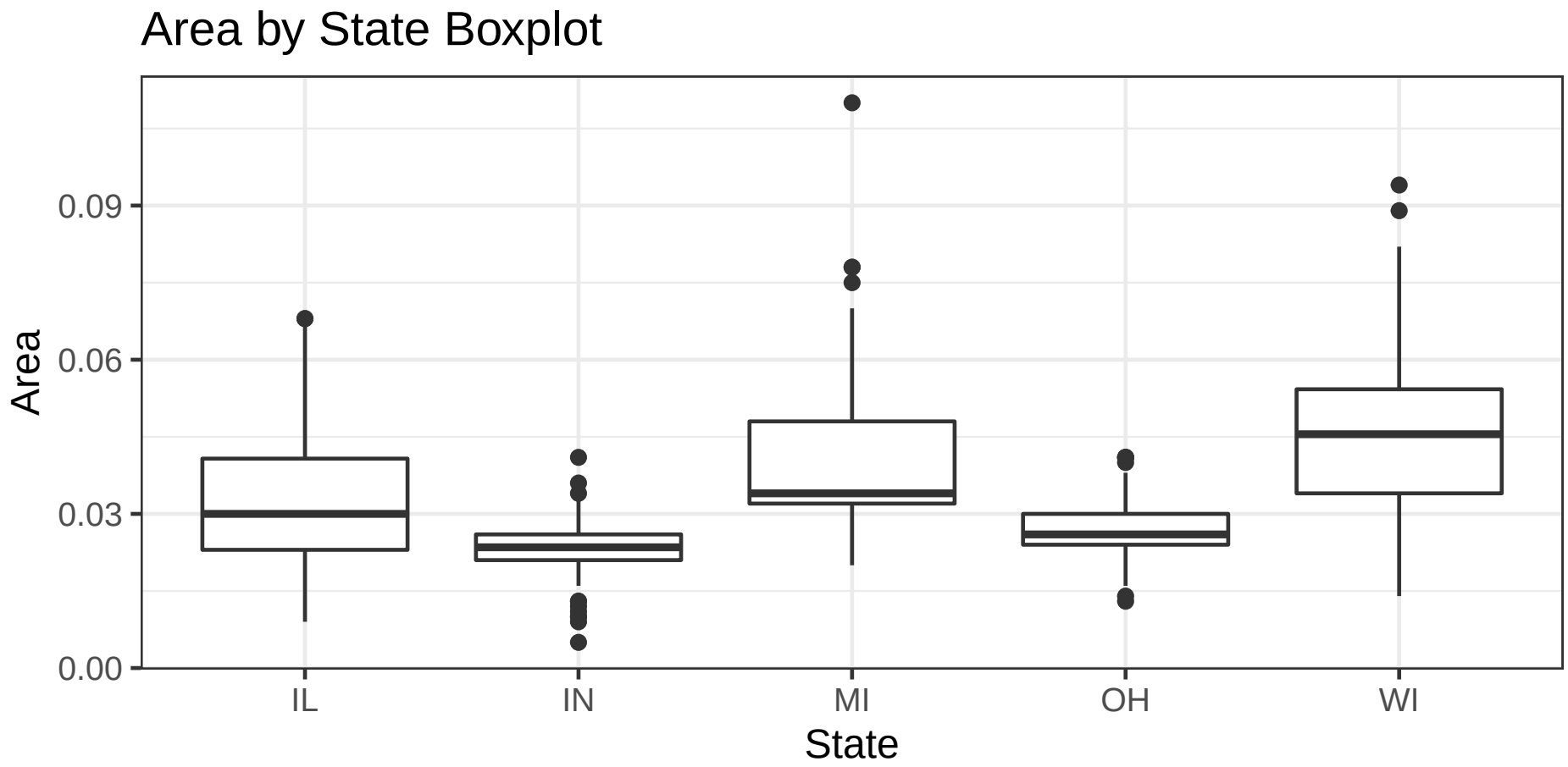


Section 1

Box Plot

geom_boxplot()

```
bx1 = ggplot(data = midwest, aes(x = state, y =area)) +
  geom_boxplot()+theme_bw()+ labs(title = "Area by State Boxpl
bx1
```



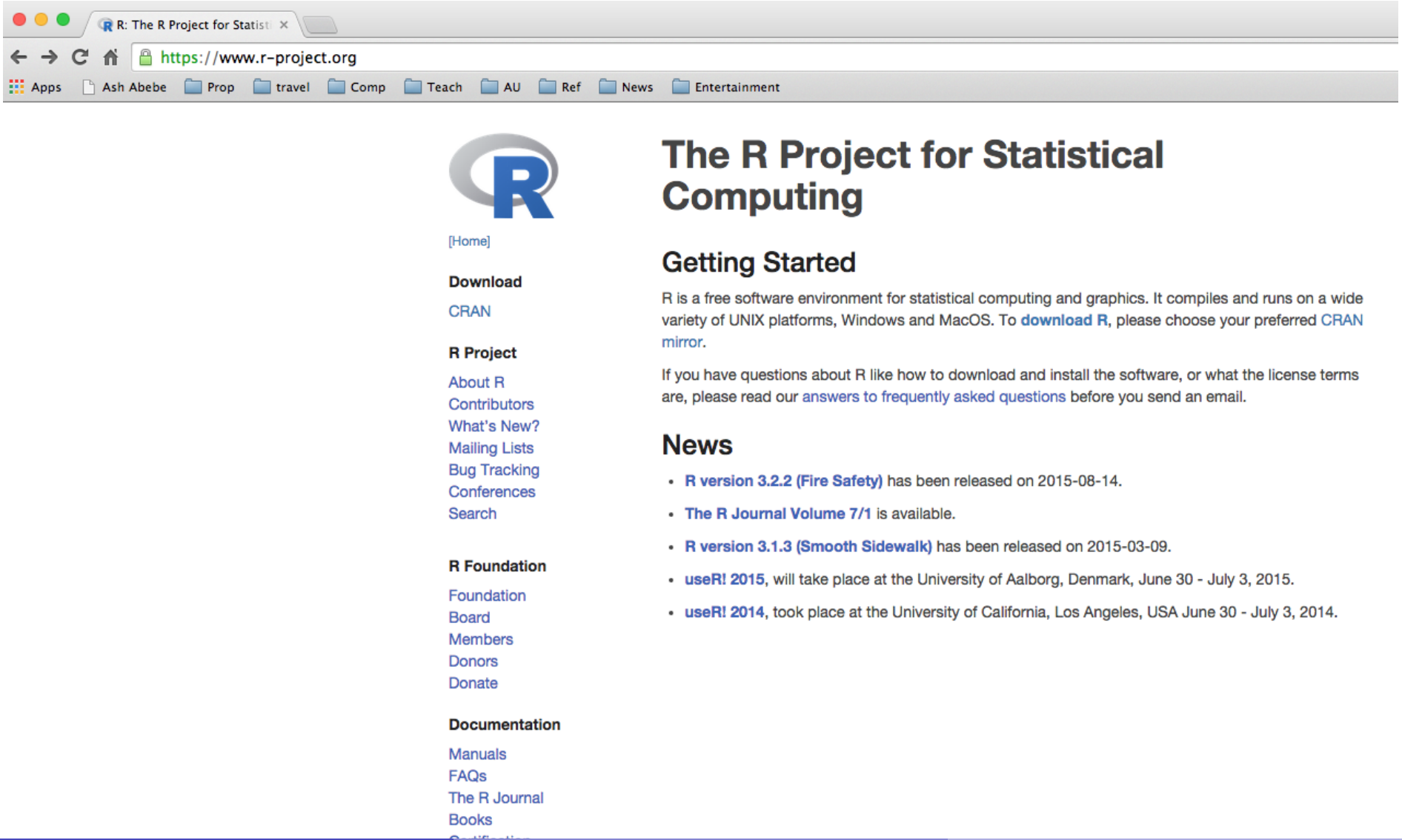
Resources

1. Free R
2. Free R studio
3. Many websites to

Learn and Learn and Learn More

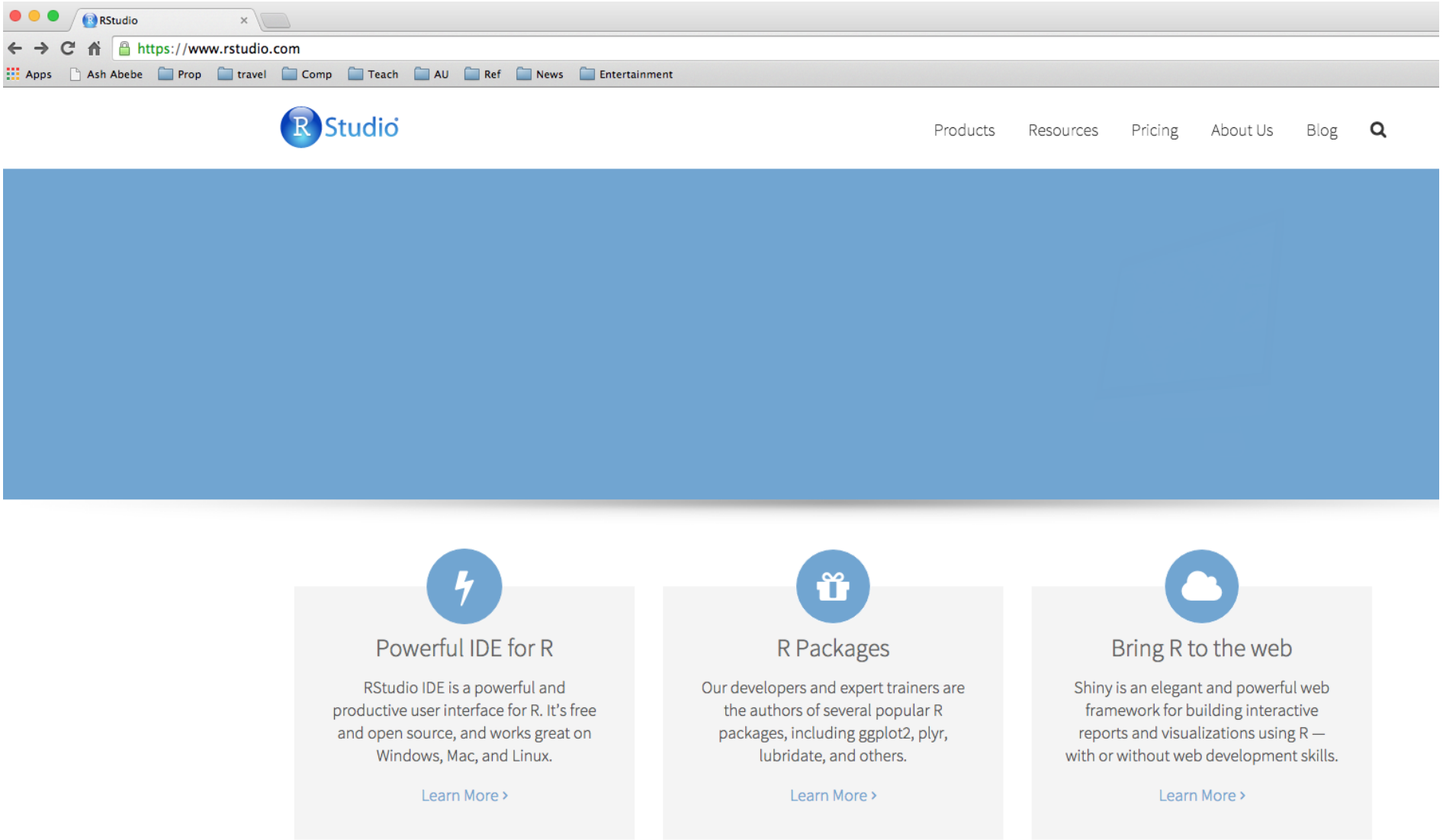
Install R

https://www.r-project.org/

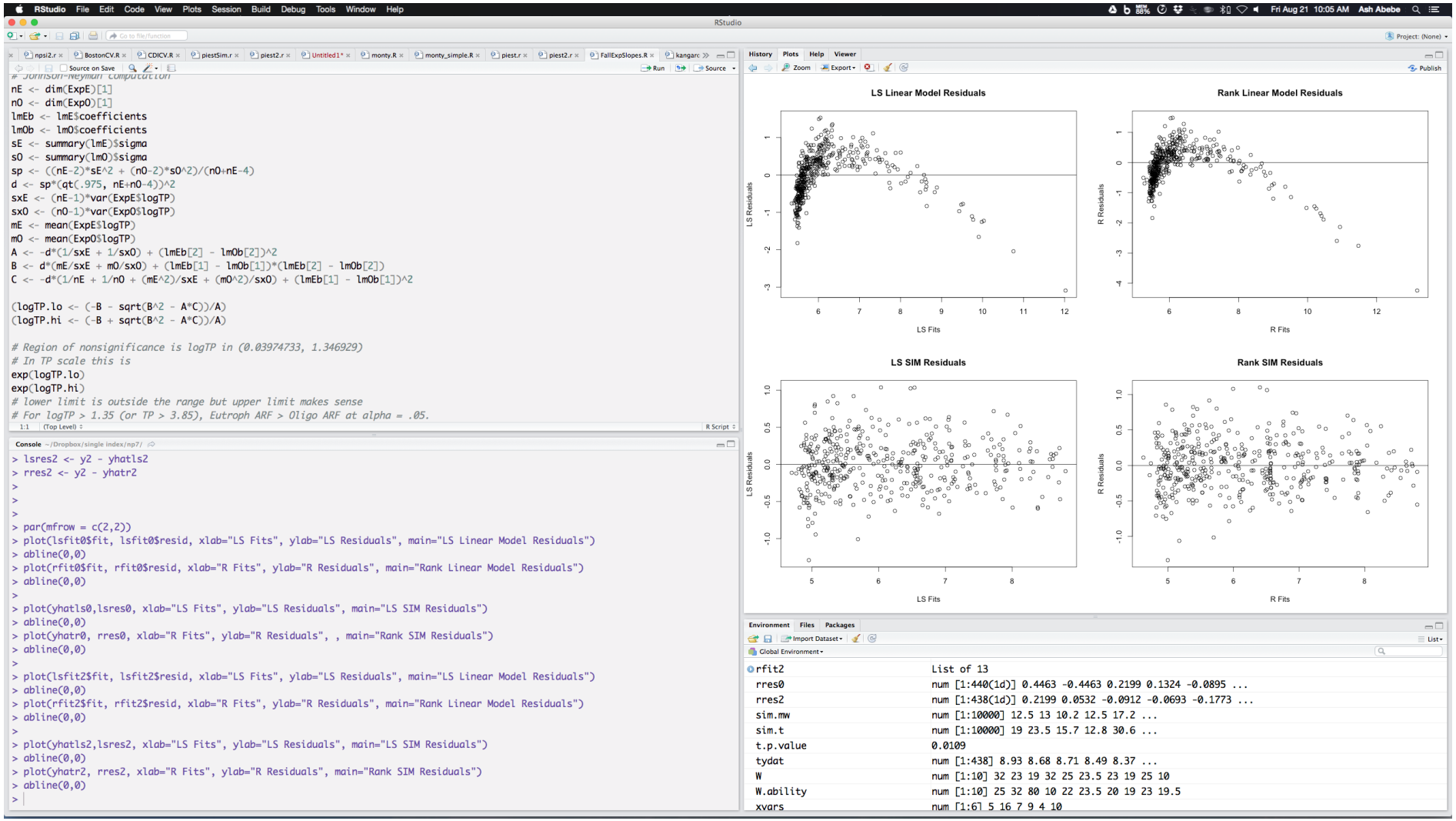


Install R-Studio

https://www.rstudio.com/

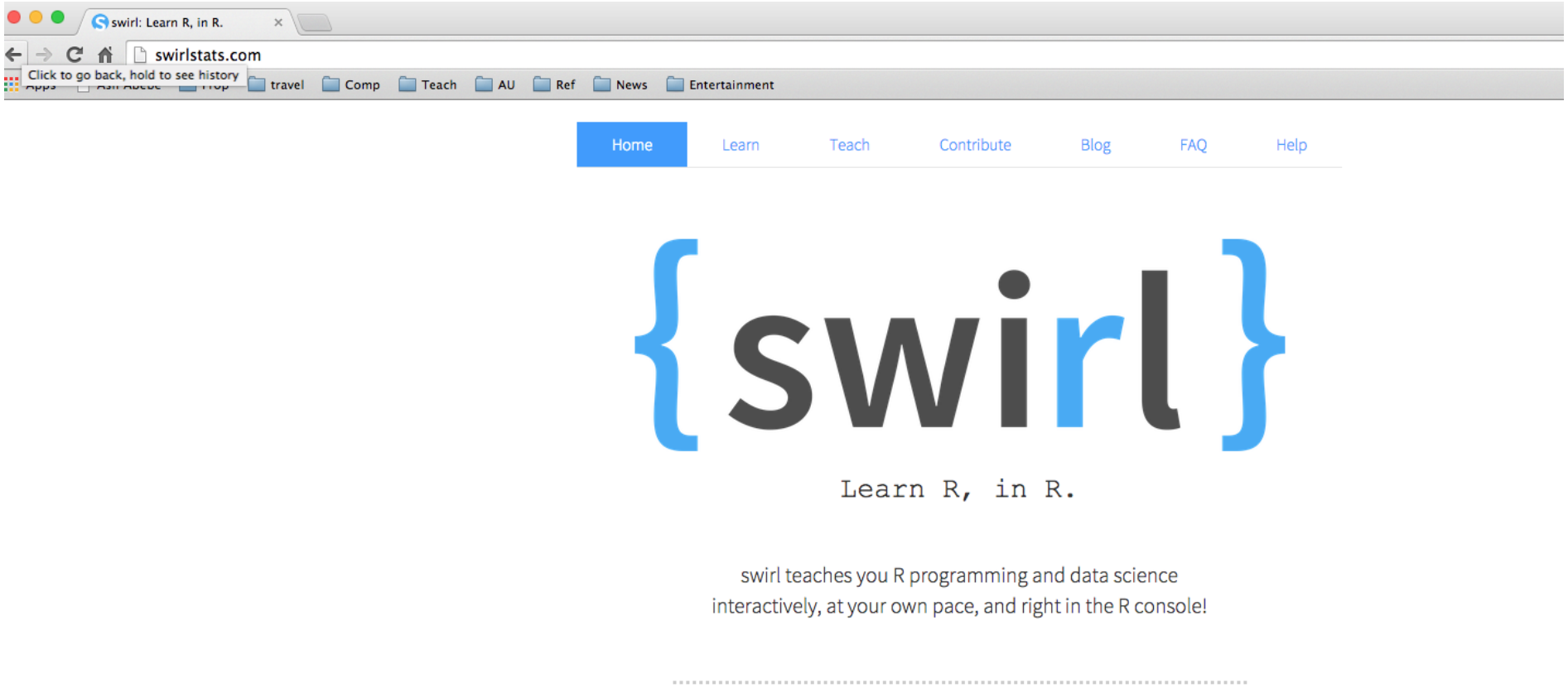


My R-Studio


RStudio File Edit Code View Plots Session Build Debug Tools Window Help
Fri Aug 21 10:05 AM Ash Abebe
RStudio
npsi2.r BostonCV.R CDICV.R piestSim.r piest2.r Untitled1* monty.R monty_simple.R piest.r piest2.r FallExpSlopes.R kangarc
Source on Save Run Source
# Johnson-Neyman computation
nE <- dim(ExpE)[1]
nO <- dim(ExpO)[1]
lmEb <- lmE$coefficients
lmOb <- lmO$coefficients
sE <- summary(lmE)$sigma
sO <- summary(lmO)$sigma
sp <- ((nE-2)*sE^2 + (nO-2)*sO^2)/(nO+nE-4)
d <- sp*(qt(.975, nE+nO-4))^2
sxE <- (nE-1)*var(ExpE$logTP)
sxO <- (nO-1)*var(ExpO$logTP)
mE <- mean(ExpE$logTP)
mO <- mean(ExpO$logTP)
A <- -d*(1/sxE + 1/sxO) + (lmEb[2] - lmOb[2])^2
B <- d*(mE/sxE + mO/sxO) + (lmEb[1] - lmOb[1])*(lmEb[2] - lmOb[2])
C <- -d*(1/nE + 1/nO + (mE^2)/sxE + (mO^2)/sxO) + (lmEb[1] - lmOb[1])^2
(logTP.lo <- (-B - sqrt(B^2 - A*C))/A)
(logTP.hi <- (-B + sqrt(B^2 - A*C))/A)
# Region of nonsignificance is logTP in (0.03974733, 1.346929)
# In TP scale this is
exp(logTP.lo)
exp(logTP.hi)
# lower limit is outside the range but upper limit makes sense
# For logTP > 1.35 (or TP > 3.85), Eutroph ARF > Oligo ARF at alpha = .05.
1:1 (Top Level) R Script
Console ~/Dropbox/single index/np7/
> lsres2 <- y2 - yhatls2
> rres2 <- y2 - yhatr2
>
>
>
> par(mfrow = c(2,2))
> plot(lsfit0$fit, lsfit0$resid, xlab="LS Fits", ylab="LS Residuals", main="LS Linear Model Residuals")
> abline(0,0)
> plot(rfit0$fit, rfit0$resid, xlab="R Fits", ylab="R Residuals", main="Rank Linear Model Residuals")
> abline(0,0)
>
> plot(yhatls0,lsres0, xlab="LS Fits", ylab="LS Residuals", main="LS SIM Residuals")
> abline(0,0)
> plot(yhatr0, rres0, xlab="R Fits", ylab="R Residuals", , main="Rank SIM Residuals")
> abline(0,0)
>
> plot(lsfit2$fit, lsfit2$resid, xlab="LS Fits", ylab="LS Residuals", main="LS Linear Model Residuals")
> abline(0,0)
> plot(rfit2$fit, rfit2$resid, xlab="R Fits", ylab="R Residuals", main="Rank Linear Model Residuals")
> abline(0,0)
>
> plot(yhatls2,lsres2, xlab="LS Fits", ylab="LS Residuals", main="LS SIM Residuals")
> abline(0,0)
> plot(yhatr2, rres2, xlab="R Fits", ylab="R Residuals", main="Rank SIM Residuals")
> abline(0,0)
>
History Plots Help Viewer
Zoom Export Publish
LS Linear Model Residuals
Rank Linear Model Residuals
LS SIM Residuals
Rank SIM Residuals
LS Residuals
R Residuals
LS Fits
R Fits
Environment Files Packages
Import Dataset List
Global Environment
rfit2 List of 13
rres0 num [1:440(1d)] 0.4463 -0.4463 0.2199 0.1324 -0.0895 ...
rres2 num [1:438(1d)] 0.2199 0.0532 -0.0912 -0.0693 -0.1773 ...
sim.mw num [1:10000] 12.5 13 10.2 12.5 17.2 ...
sim.t num [1:10000] 19 23.5 15.7 12.8 30.6 ...
t.p.value 0.0109
tydat num [1:438] 8.93 8.68 8.71 8.49 8.37 ...
W num [1:10] 32 23 19 32 25 23.5 23 19 25 10
W.ability num [1:10] 25 32 80 10 22 23.5 20 19 23 19.5
xvars num [1:6] 5 16 7 9 4 10

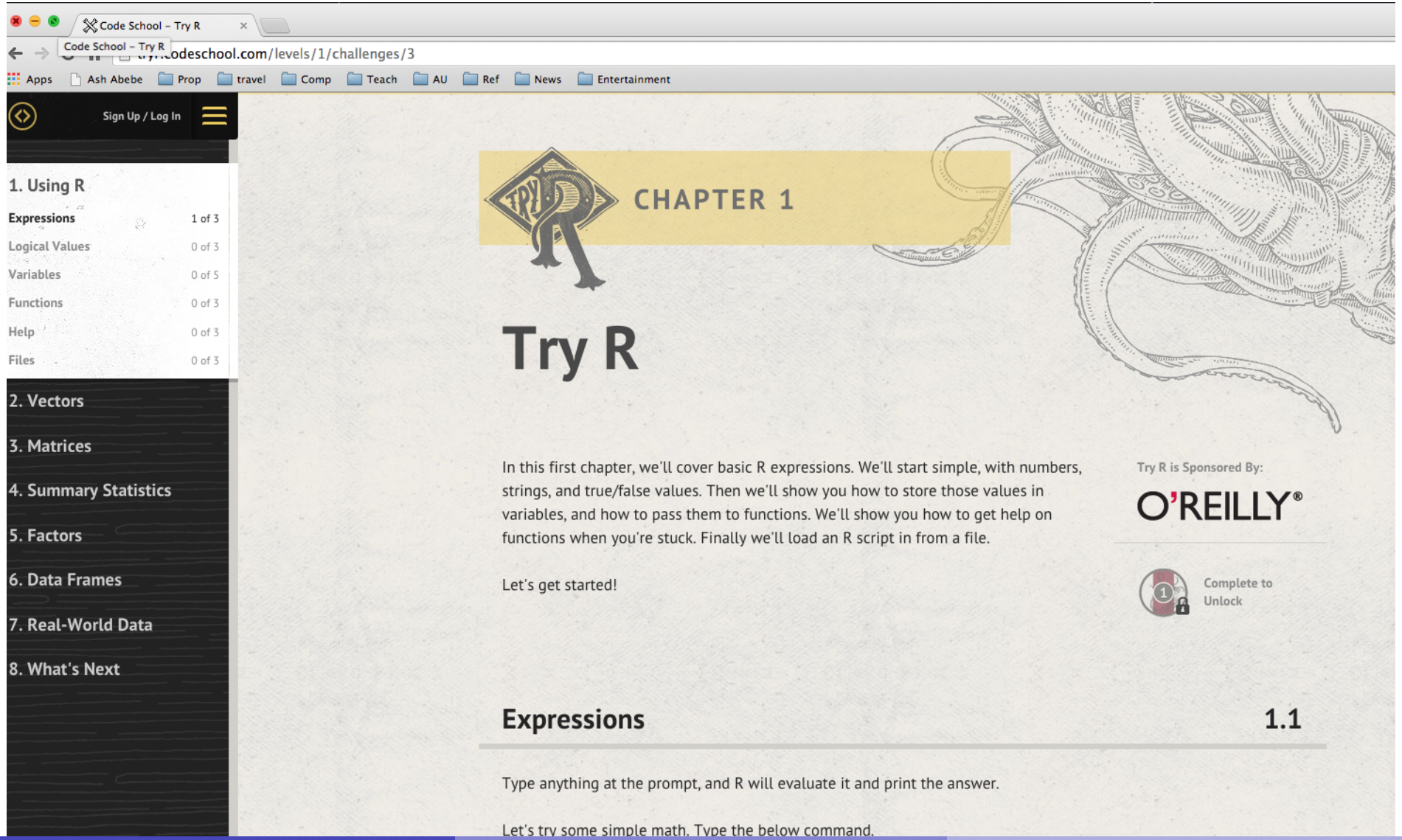
Learn R

http://swirlstats.com/



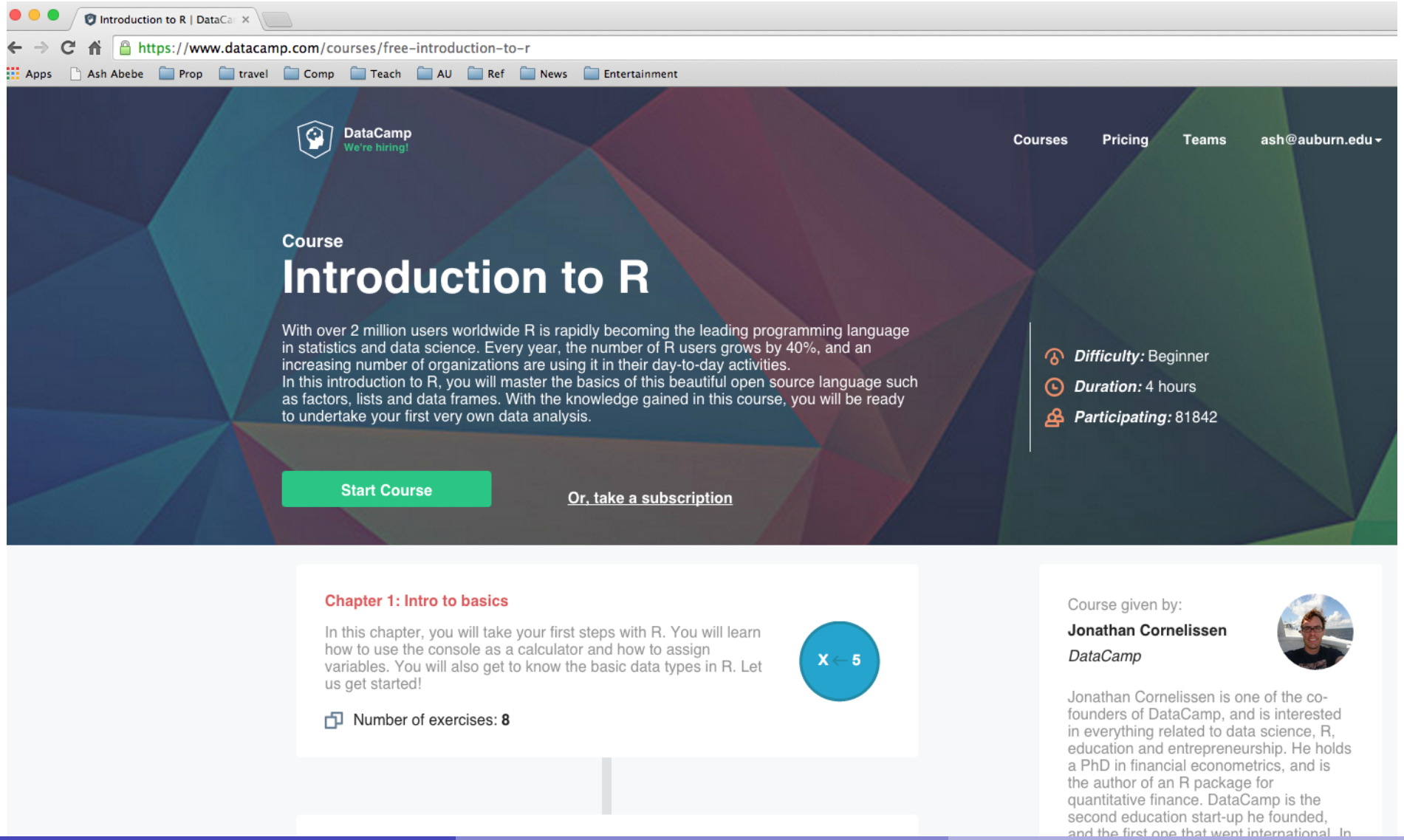
Learn R

http://tryr.codeschool.com/



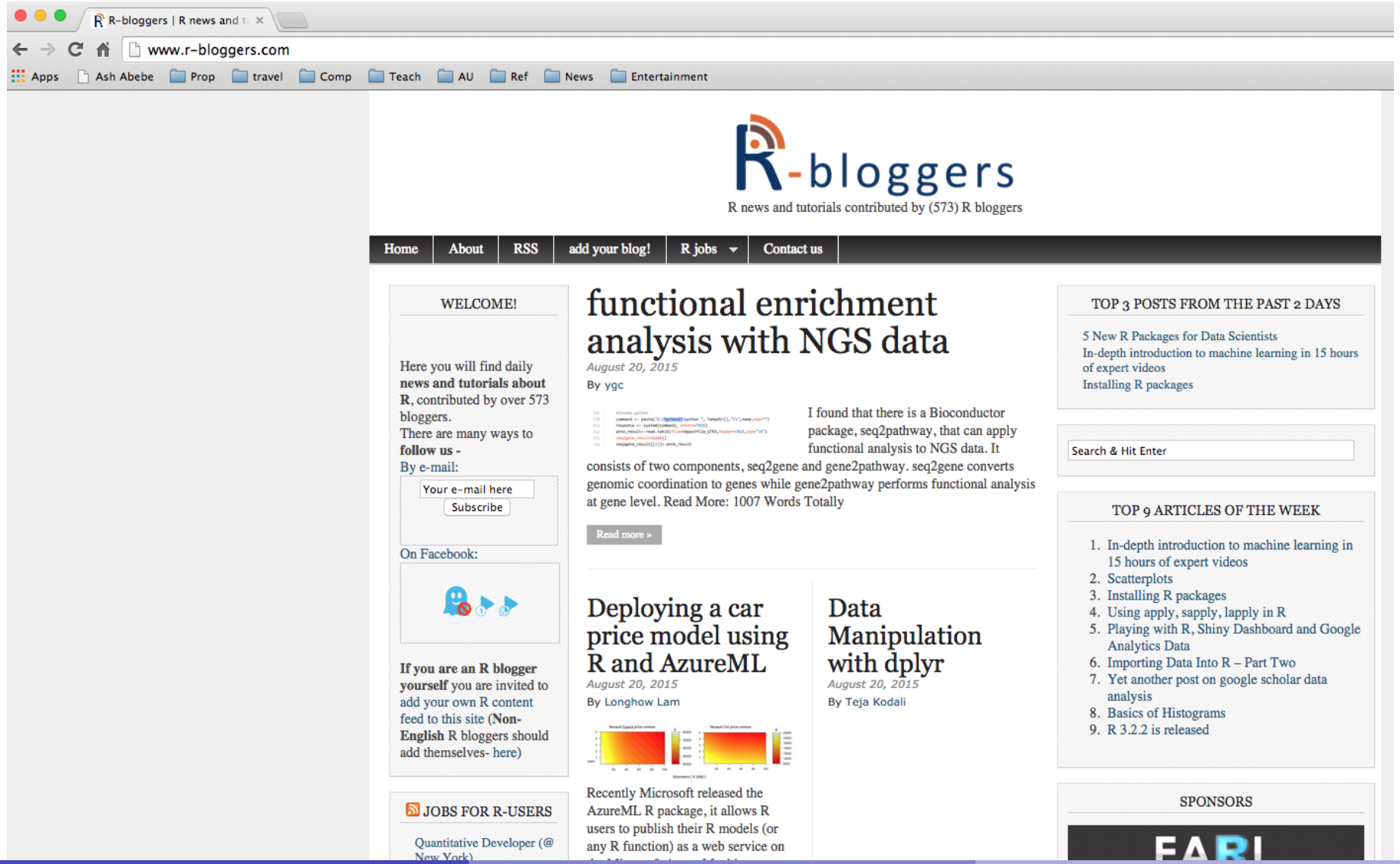
Learn R

https://www.datacamp.com/



Obsess about R

http://www.r-bloggers.com/



Ask questions about R

https://stat.ethz.ch/mailman/listinfo/r-help

R-help Info Page

https://stat.ethz.ch/mailman/listinfo/r-help

Apps | Ash Abebe | Prop | travel | Comp | Teach | AU | Ref | News | Entertainment

R-help -- Main R Mailing List: Primary help

About R-help

The main R mailing list, for announcements about the development of R and the availability of new code, questions and answers about problems and solutions using R, enhancements and patches with S and S-plus, and for the posting of nice examples and benchmarks. Please read the *General Instructions* on the R Mailing Lists page **and** follow the ***posting guide***!

This has become quite an active list with often dozens of messages per day.
It might make sense therefore to check the *digest* box in your subscription page.

For discussion about new features or R's future development, use the **R-devel** mailing list. See the R Project site for more information about R. Before asking, please read the FAQ about R.

To see the collection of prior postings to the list, visit the R-help Archives --- or use the *searchable* archives provided by Robert King and U. Newcastle.AU or the *full R site search* provided by Jc
interfaces.
Posters should be aware that the R lists are *public* discussion lists and anything you post will be **archived and accessible** via several websites for many years.

Using R-help

Since Jan.31, 2008, to be allowed to freely post messages, you (i.e., your sending e-mail address) **must be subscribed** to the list.

Thanks to a dozen of **volunteer** moderators, posting of non-subscribers is still possible -- with a delay because of the moderator approval needed. The volunteers, generously devoting part of their
Robin Hankin, Klaus Nordhausen, Anne York, Adrian Dusa, Kevin Thorpe, Peter Alspach and David Winsemius (in addition to the R-help list maintainers).

To post a message to all the list members, send email to r-help@r-project.org.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to R-help

Subscribe to R-help by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a hidden list, which means that tl

Your email address:	
Your name (optional):	

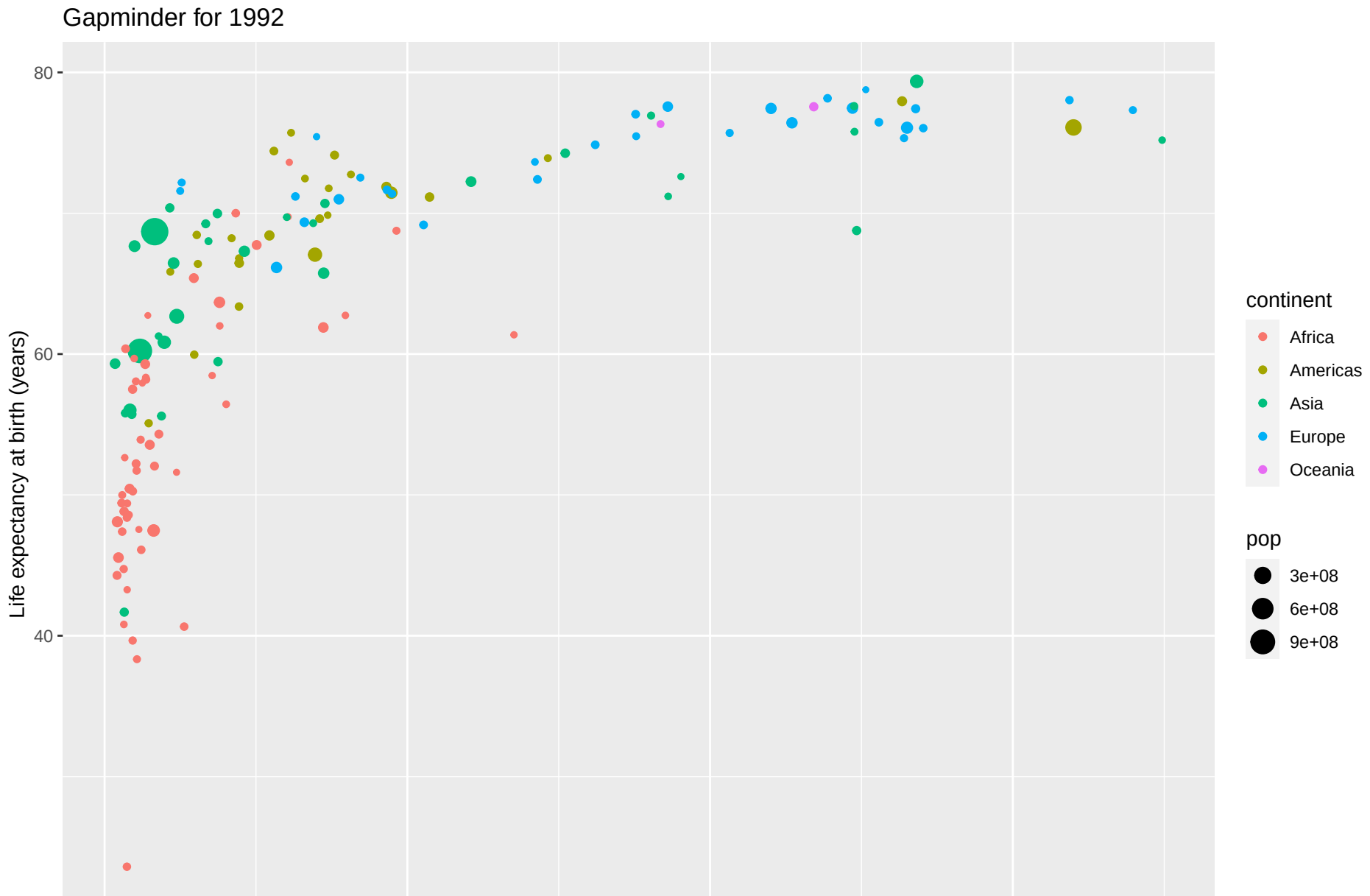
You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

Your Turn :Use Gapminder data

```
library(gapminder)
library(dplyr)
 df <- gapminder::gapminder
 df1= df%>%filter(year ==1992)
 df1
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp       pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Afghanistan Asia       1992    41.7  16317921      649.
##  2 Albania     Europe     1992    71.6   3326498     2497.
##  3 Algeria     Africa     1992    67.7  26298373     5023.
##  4 Angola      Africa     1992    40.6   8735988     2628.
##  5 Argentina   Americas   1992    71.9  33958947     9308.
##  6 Australia   Oceania    1992    77.6  17481977    23425.
##  7 Austria     Europe     1992    76.0   7914969    27042.
##  8 Bahrain     Asia       1992    72.6    529491    19036.
```

Produce this Plot



THANK YOU

- **Thank You ALL**