GGPLOT2

Report on how professional graphics GGPLOT works in R

GGPLOT2 is based on the grammar of graphics, the idea that can build every graph from the same few components: a data set, a set of geoms-visual marks that represent data points, and a coordinate system.

Advantages of ggplot2

1.Consistent underlying grammar of graphics (Wilkinson, 2005)

2.Plot specification at a high level of abstraction

3.Very flexible

4.Theme system for polishing plot appearance

5.Mature and complete graphics system 6.Many users, active mailing list

We willsee how ggplot2 works in R using following examples:

Let's look at housing prices.

Loading the package in R

```
housing <- read.csv("E:/DAV/HW3/dataSets/landdata-states.csv")
head(housing[1:5])
```

```
##   State region    Date Home.Value Structure.Cost
## 1    AK   West 2010.25     224952         160599
## 2    AK   West 2010.50     225511         160252
## 3    AK   West 2009.75     225820         163791
## 4    AK   West 2010.00     224994         161787
## 5    AK   West 2008.00     234590         155400
## 6    AK   West 2008.25     233714         157458
```

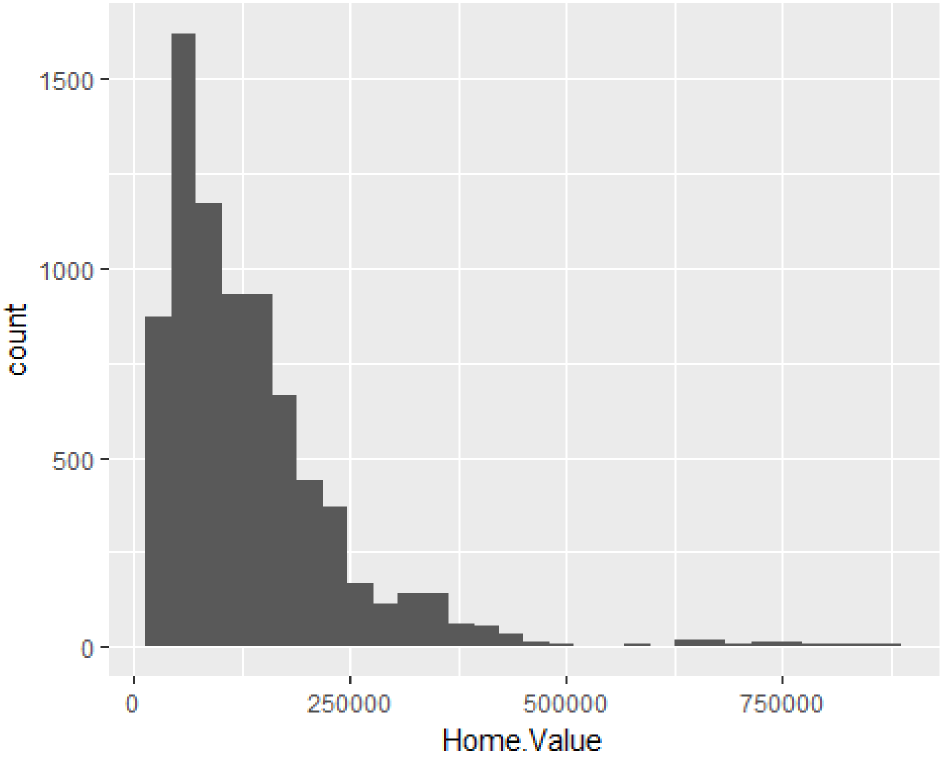
Example of buildinga Histogram using GGPLOT2:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.1
```

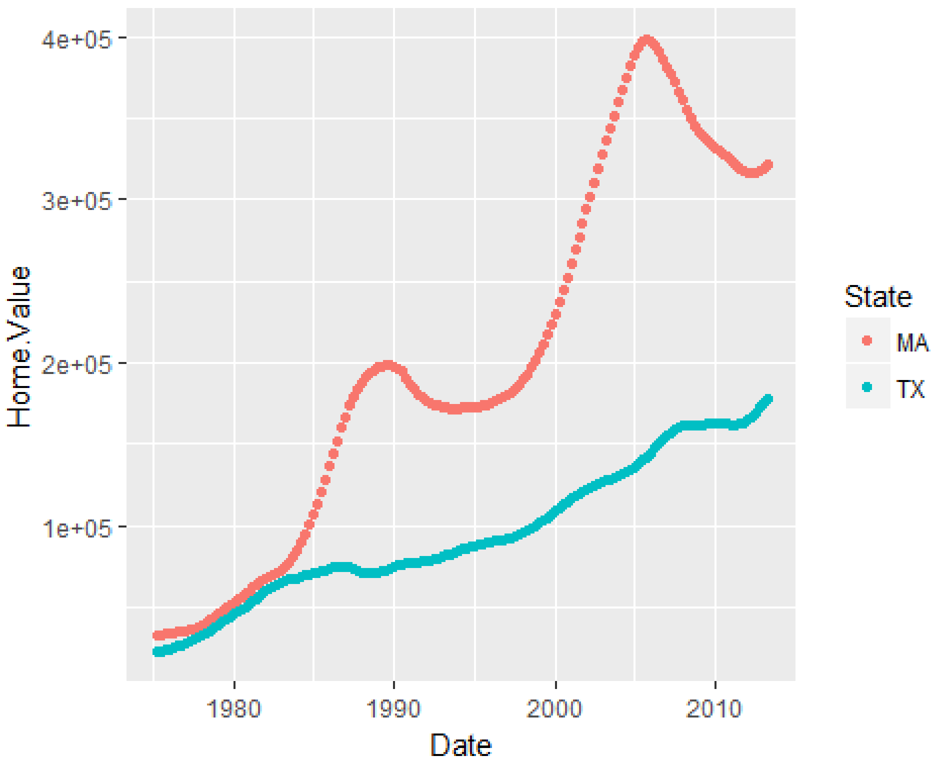
```
ggplot(housing, aes(x = Home.Value)) +
  geom_histogram()
```

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



Now we will use `ggplot2' colored scatter plot example:

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



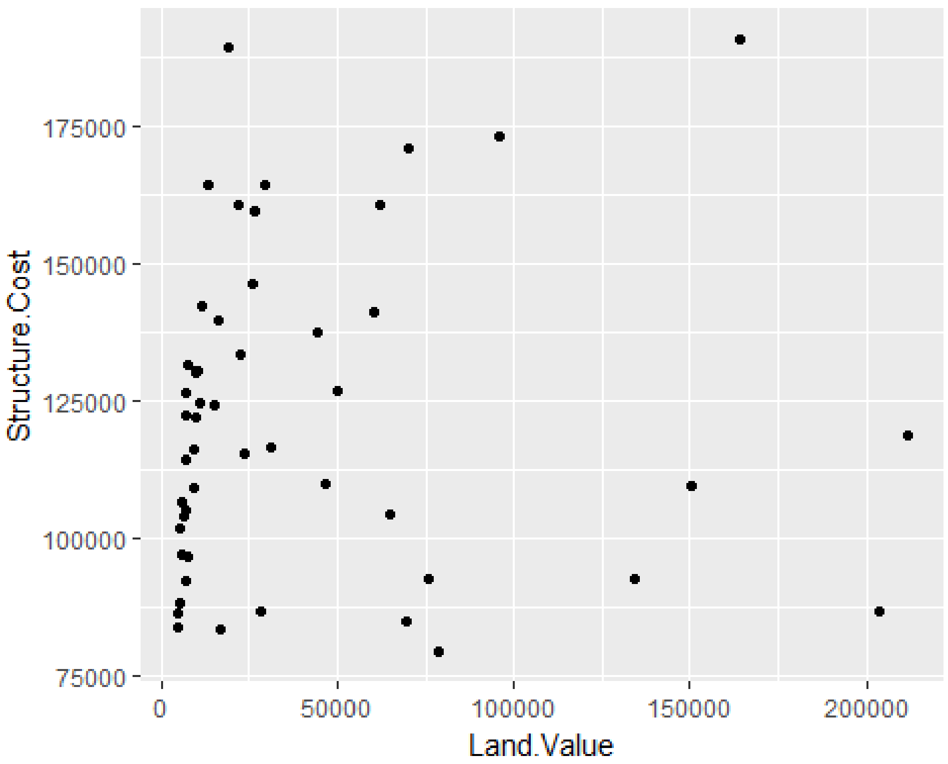
We can see how visually appealing the plot looks and it is very easy to implement.

In ggplot land /aesthetic/ means "something you can see". Examples include:Position (i.e., on the x and y axes), Color ("outside" color),Fill ("inside" color),Shape (of points),Linetype Size.

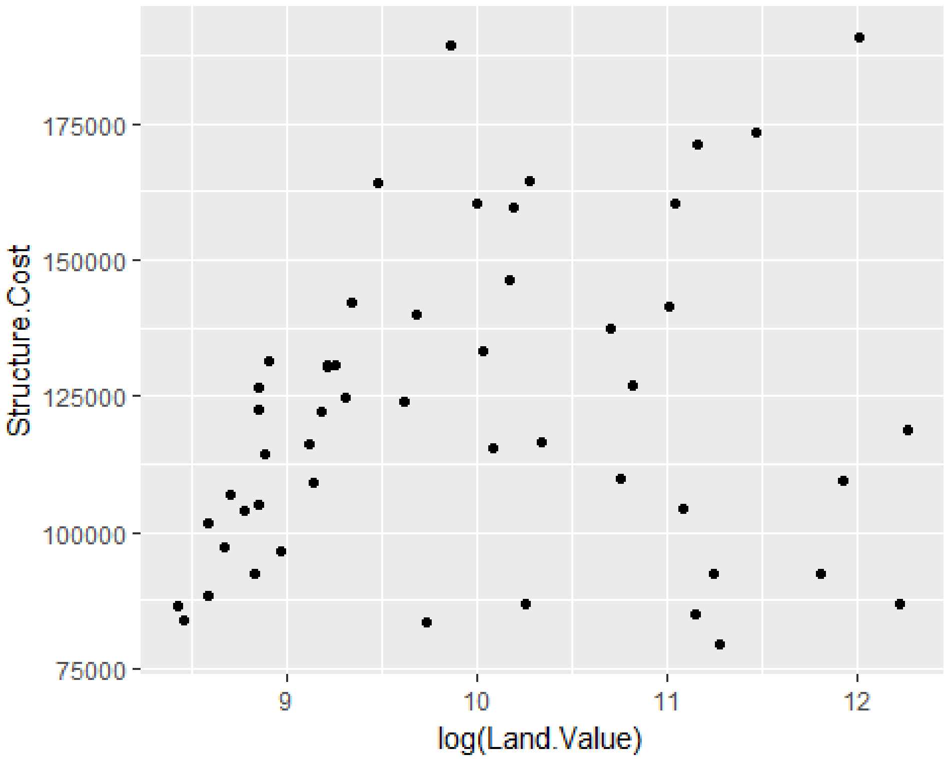
Geometric objects are the actual marks we put on a plot. Examples include: 1.points (`geom_point', for scatter plots, dot plots, etc) 2.lines (`geom_line', for time series, trend lines, etc) 3.boxplot (`geom_boxplot', for, well, boxplots!)

Below is an example where we can see the use of aesthetic and geometric objects

```
hp2001Q1 <- subset(housing, Date == 2001.25)
ggplot(hp2001Q1,
       aes(y = Structure.Cost, x = Land.Value)) +
  geom_point()
```



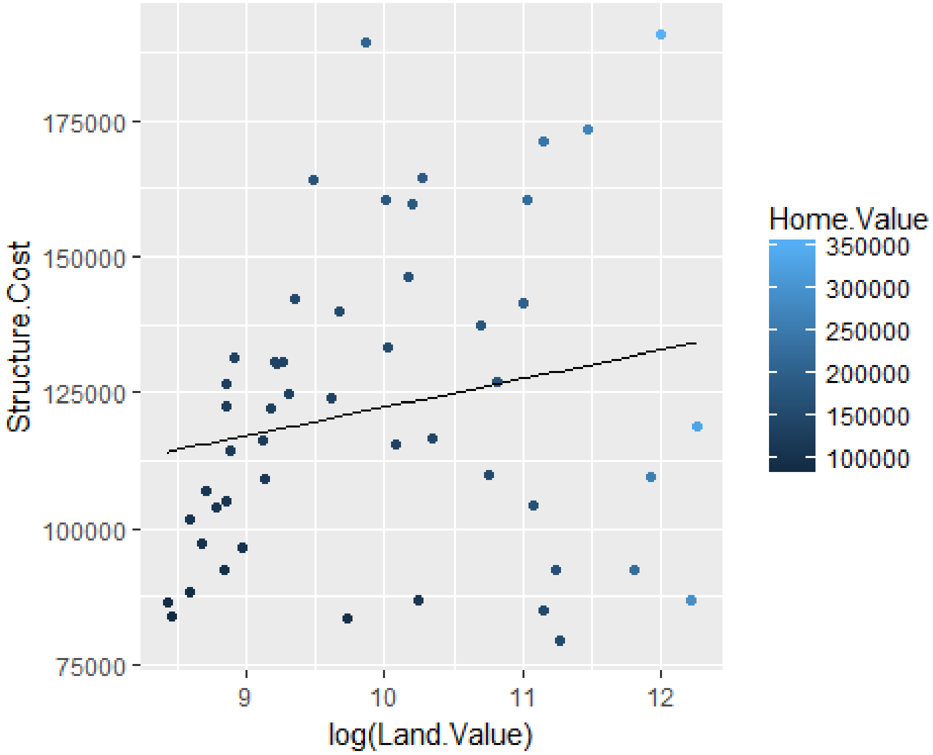
```
ggplot(hp2001Q1,
       aes(y = Structure.Cost, x = log(Land.Value))) +
  geom_point()
```



Lines (Prediction Line)

A plot constructed with ggplot' can have more than one geom. In that case the mappings established in theggplot()' call are plot defaults that can be added to or overridden. Our plot could use a regression line:

```
hp2001Q1$pred.SC <- predict(lm(Structure.Cost ~ log(Land.Value), data =
hp2001Q1))

p1 <- ggplot(hp2001Q1, aes(x = log(Land.Value), y = Structure.Cost))

p1 + geom_point(aes(color = Home.Value)) +
  geom_line(aes(y = pred.SC))
```

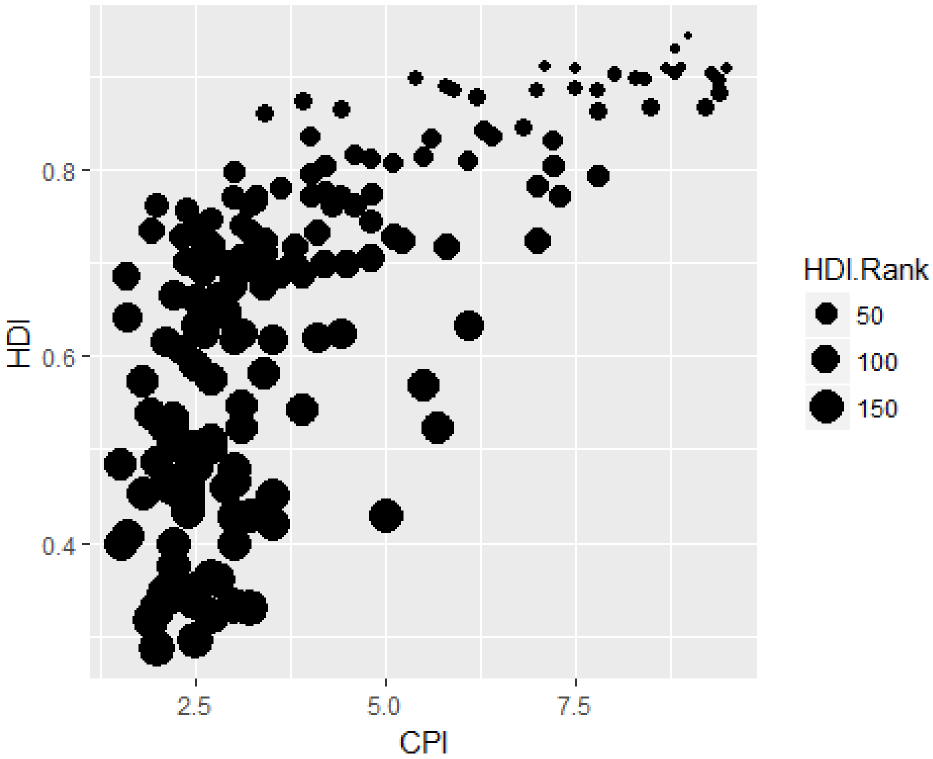


`dataSets/EconomistData.csv' file. Read it in with:

```
dat <- read.csv("E:/DAV/HW3/dataSets/EconomistData.csv")
head(dat)

##   X     Country HDI.Rank   HDI CPI            Region
## 1 1 Afghanistan      172 0.398 1.5      Asia Pacific
## 2 2     Albania       70 0.739 3.1 East EU Cemt Asia
## 3 3     Algeria       96 0.698 2.9              MENA
## 4 4      Angola      148 0.486 2.0               SSA
## 5 5   Argentina       45 0.797 3.0          Americas
## 6 6     Armenia       86 0.716 2.6 East EU Cemt Asia
```

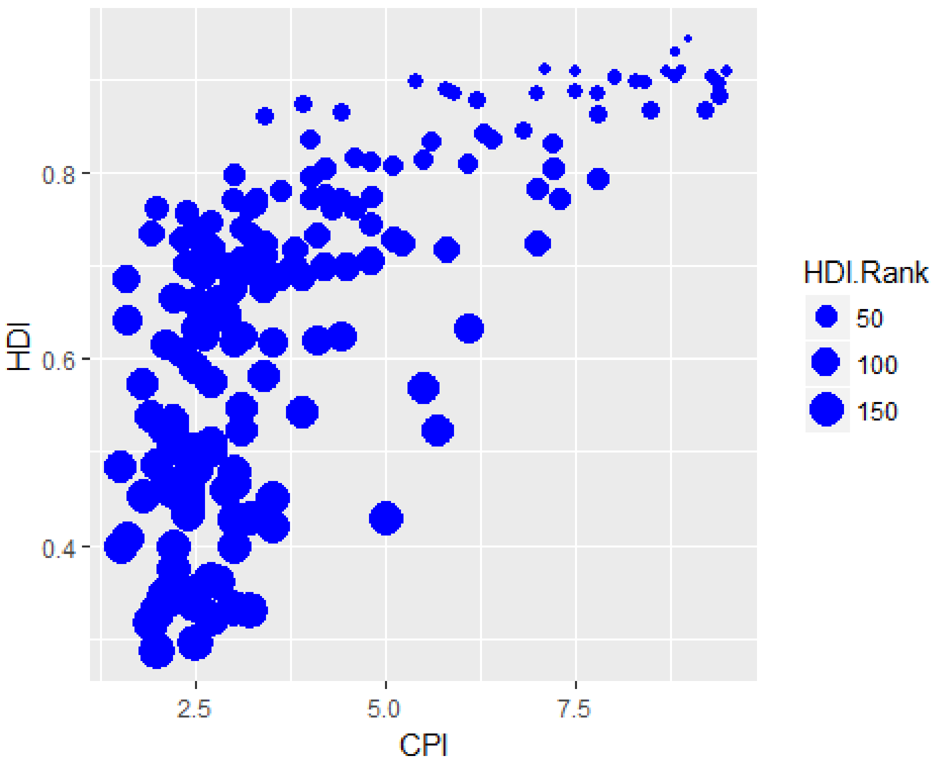
1. Scatter plot with CPI on x-aixs and HDI on y axis

```
p<-ggplot(dat, aes(x = CPI, y = HDI, size = HDI.Rank)) + geom_point()
p
```



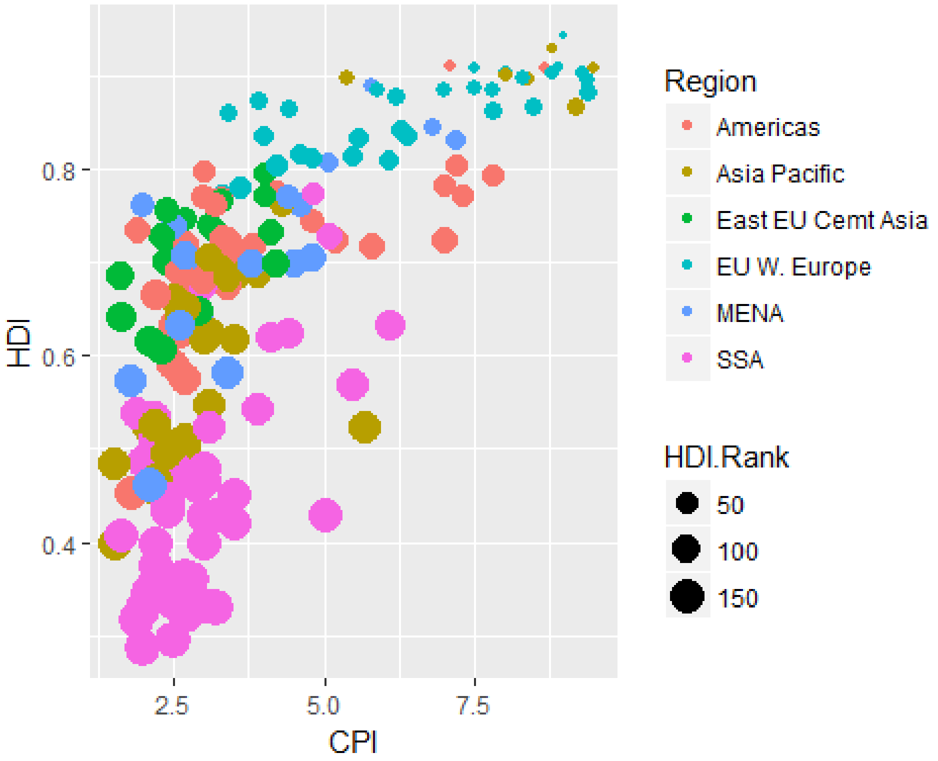
2.Color the points blue.

```
p1<-p+geom_point(colour='blue')
p1
```



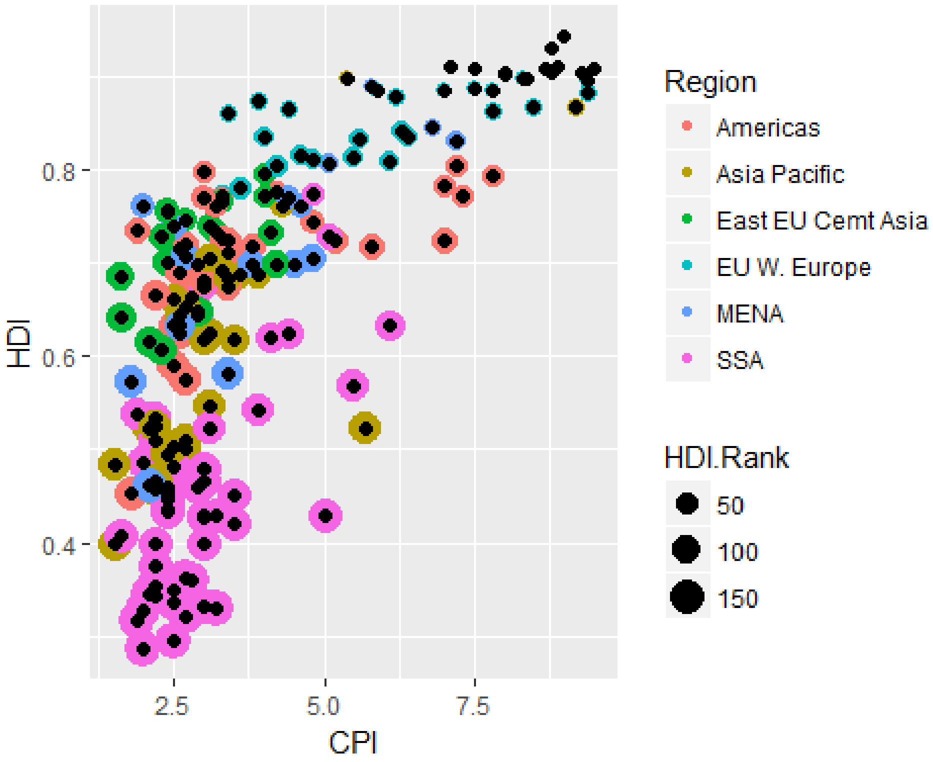
3. Map the color of the points to Region.

```
p2<-p1+geom_point(aes(colour=Region))
p2
```



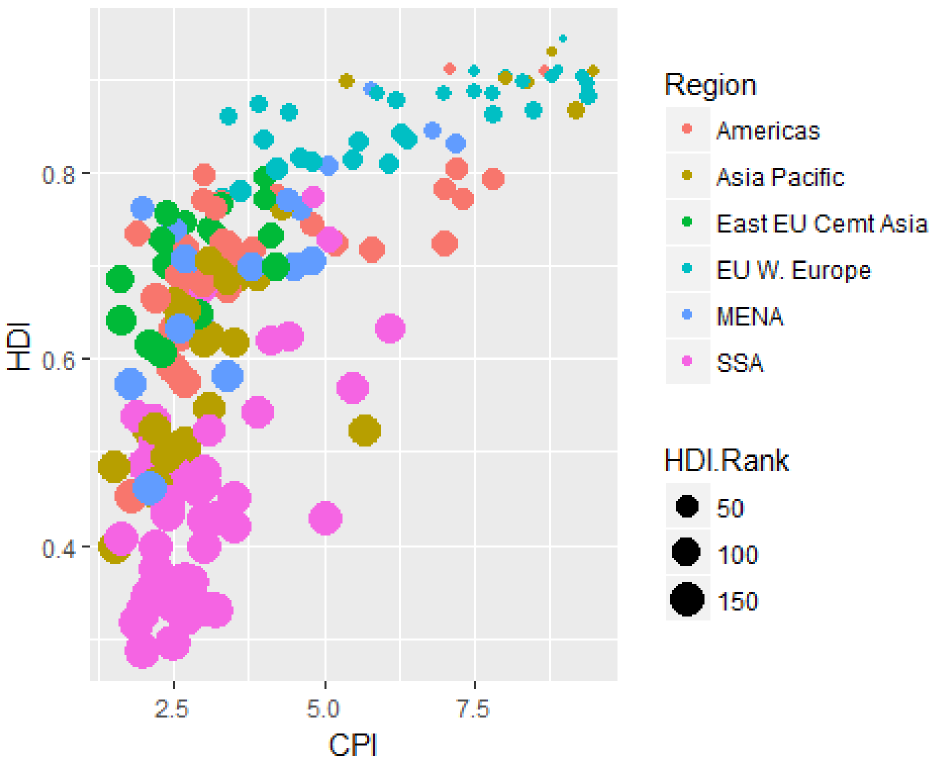
4. Make the points bigger by setting size to 2

```
p3<-p2+geom_point(size=2)
p3
```



5. Map the size of the points to HDI.Rank

```
p3<-p2+geom_point(aes(colour=Region,size=HDI.Rank))
p3
```

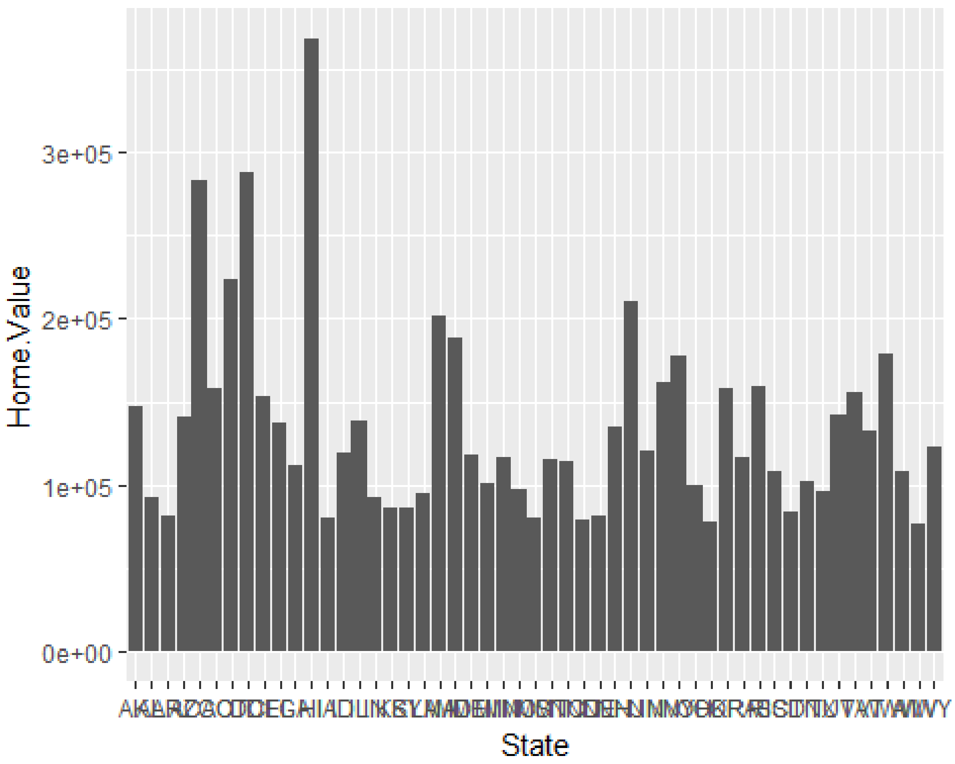


Changing The Statistical Transformation

```
housing.sum <- aggregate(housing["Home.Value"], housing["State"], FUN=mean)
rbind(head(housing.sum), tail(housing.sum))
```

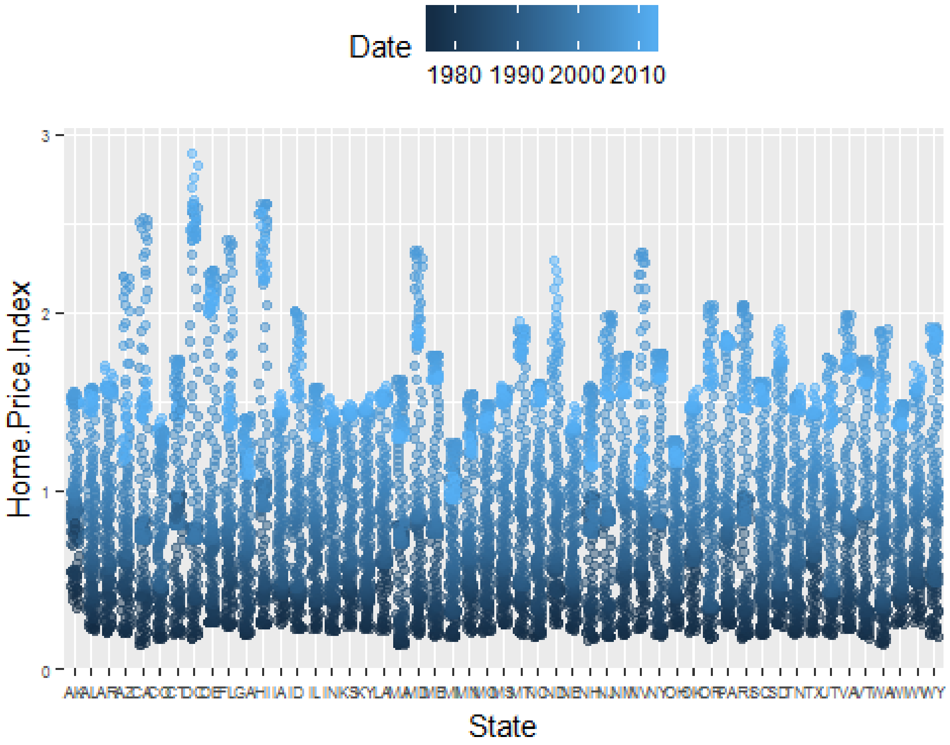
```
##    State Home.Value
## 1     AK  147385.14
## 2     AL   92545.22
## 3     AR   82076.84
## 4     AZ  140755.59
## 5     CA  282808.08
## 6     CO  158175.99
## 46    VA  155391.44
## 47    VT  132394.60
## 48    WA  178522.58
## 49    WI  108359.45
## 50    WV   77161.71
## 51    WY  122897.25
```

```
ggplot(housing.sum, aes(x=State, y=Home.Value)) +
  geom_bar(stat="identity")
```



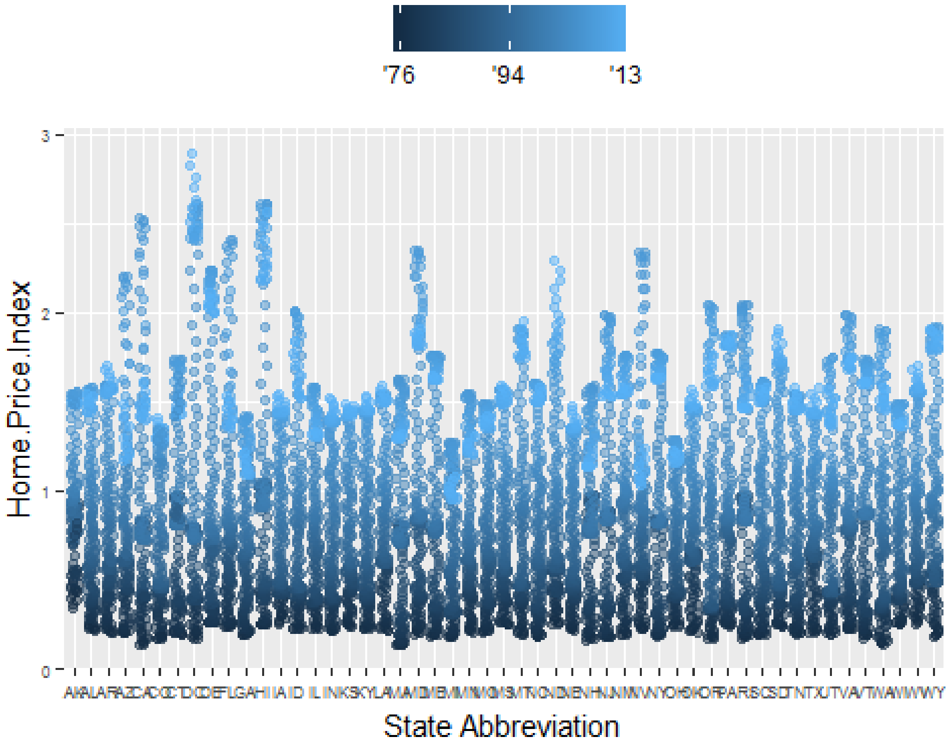
Constructing a dotplot showing the distribution of home values by Date and State.

```
p5 <- ggplot(housing,
             aes(x = State,
                 y = Home.Price.Index)) +
  theme(legend.position="top",
        axis.text=element_text(size = 6))
(p6 <- p5 + geom_point(aes(color = Date),
                       alpha = 0.5,
                       size = 1.5,
                       position = position_jitter(width = 0.25, height = 0)))
```



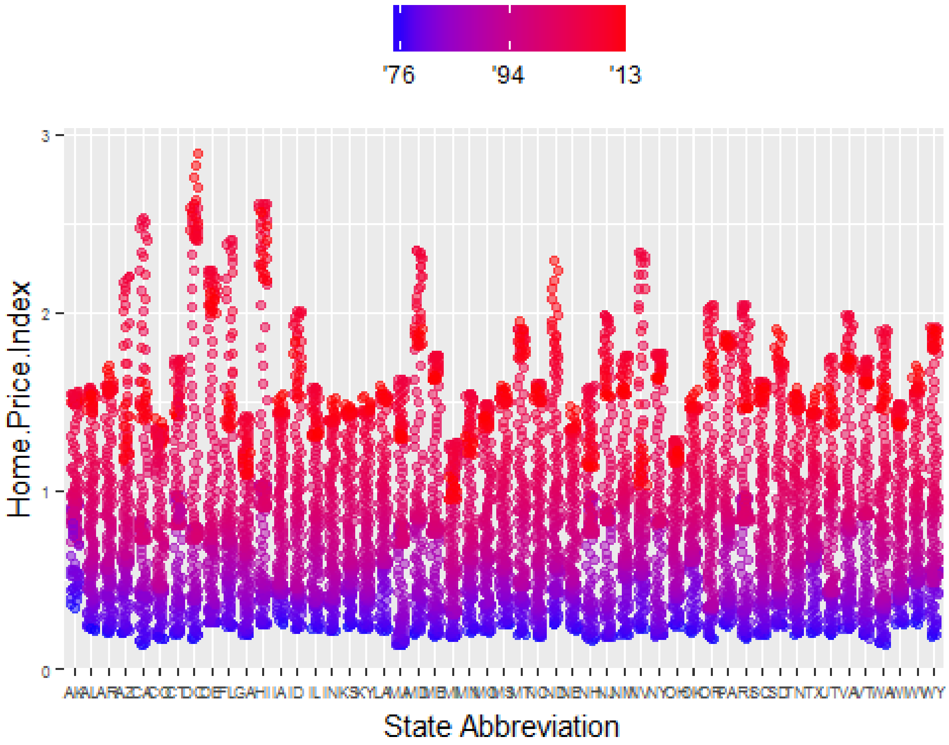
Modifying the breaks for the x axis and color scales

```
p6 + scale_x_discrete(name="State Abbreviation") +
  scale_color_continuous(name="",
                         breaks = c(1976, 1994, 2013),
                         labels = c("'76", "'94", "'13"))
```

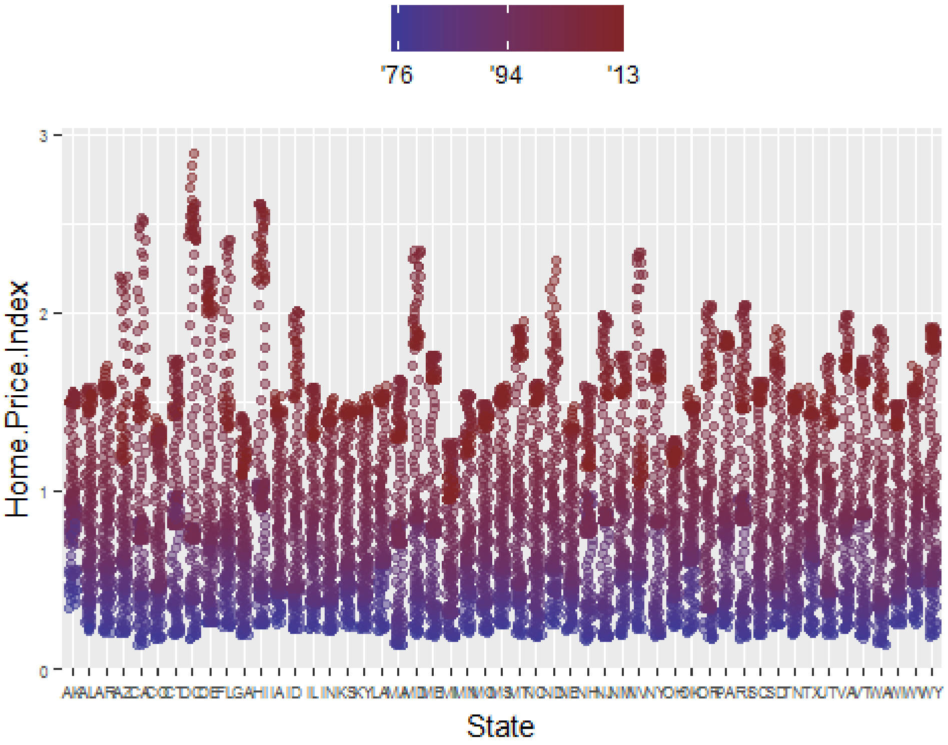


Changing the low and high values to blue and red:

```
p6+
  scale_x_discrete(name="State Abbreviation") +
  scale_color_continuous(name="",
                         breaks = c(1976, 1994, 2013),
                         labels = c("'76", "'94", "'13"),
                         low = "blue", high = "red")
```

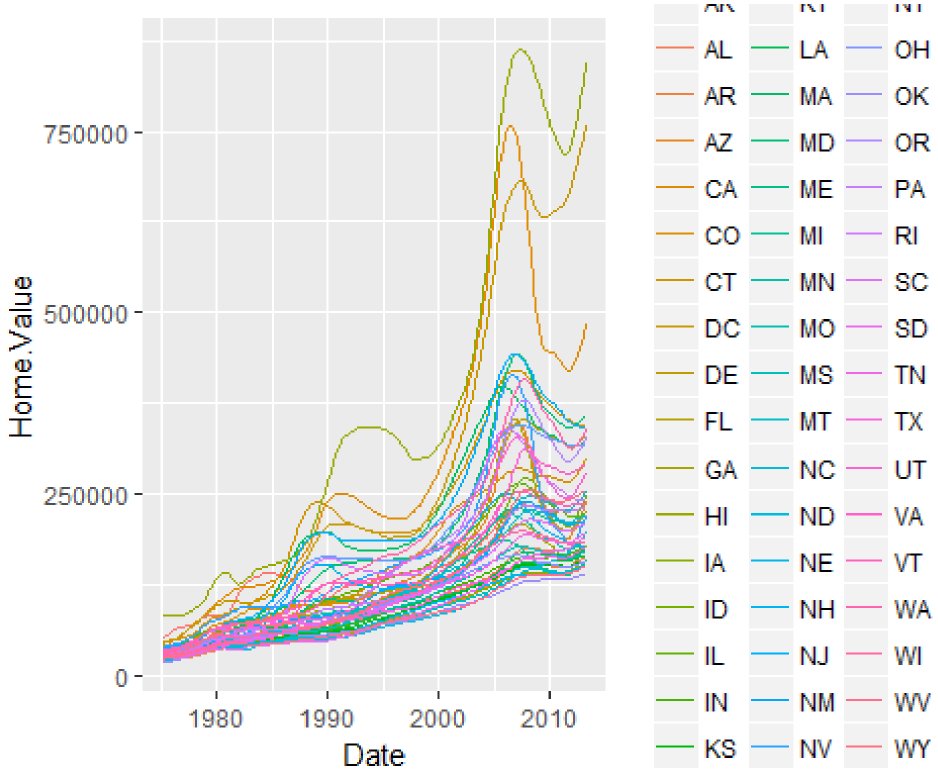


```
library(scales)

## Warning: package 'scales' was built under R version 3.4.1

p6 +
  scale_color_continuous(name="",
                         breaks = c(1976, 1994, 2013),
                         labels = c("'76", "'94", "'13"),
                         low = muted("blue"), high = muted("red"))
```



Mapping State to color:

```r
p7 <- ggplot(housing, aes(x = Date, y = Home.Value))
p7 + geom_line(aes(color = State))
```



Reasons to use ggplot2

1.Easy to make pretty and elaborate graphs

2.Easy facetting

3.Easy legend

4.Approaces the graph from a visual perspective rather than a programming perspective.