Data Visualization in R with ggplot2

Contents

1 Introduction

This course introduces you to data visualization in R using the ggplot2 package. The ggplot2 package implements the grammar of graphics concepts for creating visually appealing and professional looking graphics in R. Basic knowledge of working with datasets in R is essential but experience with plotting functions is not required.

By the end of the course you will be able to:

- Create scatterplots, histograms, line graphs, and boxplots
- Add chart labels, axis labels and legends to the plots
- Apply statistical transformations to the plots
- Change various attributes of plot layers including color, shape, size and scale
- Create mutiple plots in a single figure using facets
- Apply themes to change the appearance of the plots

Last Updated: Oct 19, 2017 12:41 AM

2 Acknowledgments

Content of this workshop is based on the following:

- ggplot2 tutorial from Harvard University
- ggplot2 Workshop (Vanderbilt, 2007)

3 Resources

- Google
- Data Visualization with ggplot2 Cheat Sheet
- ggplot2 Documentation
- R Graphics Cookbook
- ggplot2 tutorial from Harvard University
- ggplot2 Workshop (Vanderbilt, 2007)

4 Getting Started

4.1 Prerequisites

Basic knowledge of working with datasets in R is essential. This course assumes that you're comfortable with reading and manipulating datasets, working with script files, and navigating in RStudio. Experience with plotting functions in R is helpful but not required.

4.2 Software Requirements

4.2.1 R and RStudio

Recent versions of R (version 3.2 or newer) and RStudio (version 0.99 or above) are required.

You can download the latest versions from the links below:

- Download R
- Download RStudio

You can find out the version of R installed by typing `version` at the console:

```
version
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          3
## minor          4.2
## year           2017
## month          01
## day            27
## svn rev        73369
## language       R
## version.string R version 3.4.2 (2017-01-27)
## nickname       Short Summer
```

4.3 Installing ggplot2

If you don't have ggplot2 installed, you can install it using the `install.packages()` function:

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

You can find out the version of ggplot installed using the `packageVersion()` function:

```
packageVersion("ggplot2")
```

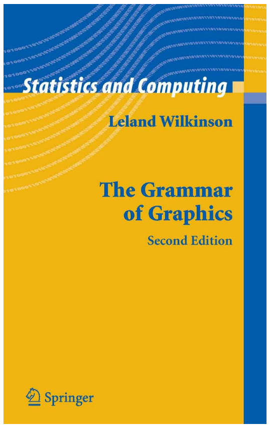
```
## [1] '2.2.1'
```

4.4 Installing ggplot2 Extentions

We need the following ggplot extensions for this tutorial:

```
install.packages("scales")
install.packages("ggrepel")
install.packages("ggthemes")
```

5 Grammer of Graphics



Wilkinson, L. (2006). *The grammar of graphics*. Springer Science & Business Media.

5.1 Building Blocks of a Graph

- Data
- Aesthetic mapping
- Geometric object
- Statistical transformations
- Scales
- Coordinate system
- Position adjustments
- Faceting

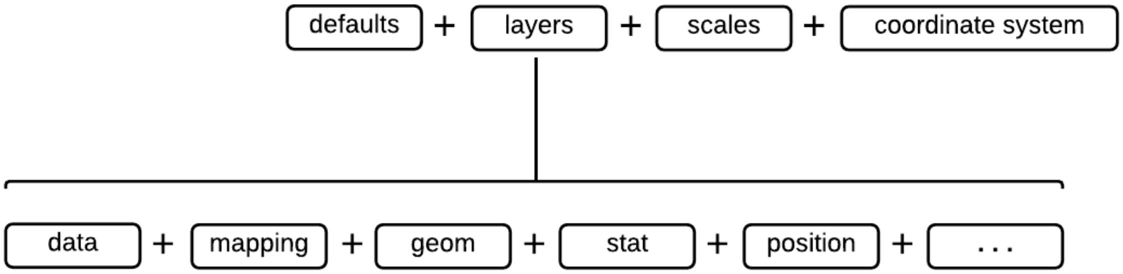


Figure 1:

6 Basic Plots

6.1 Loading ggplot2

Like any other R package, you must load `ggplot2` using the `library` function before you can use any of the functionality that it offers. We will also load the extensions that provide additional features:

```
library(ggplot2)
library(ggrepel)
library(ggthemes)
library(scales)
```

6.2 Dataset

Let's start by loading the housing dataset:

```
housing <- read.csv("https://raw.githubusercontent.com/altaf-ali/ggplot_tutorial/master/data/housing.cs
```

Now, let's see what the dataset looks like:

```
head(housing)
```

```
##   State Region       Date Home.Value Structure.Cost Land.Value
## 1    AK   West 2010-03-01     224952         160599      64352
## 2    AK   West 2010-06-01     225511         160252      65259
## 3    AK   West 2009-09-01     225820         163791      62029
## 4    AK   West 2009-12-01     224994         161787      63207
## 5    AK   West 2007-12-01     234590         155400      79190
## 6    AK   West 2008-03-01     233714         157458      76256
##   Land.Share..Pct. Home.Price.Index Land.Price.Index Year Quarter
## 1             28.6            1.481            1.552 2010       1
## 2             28.9            1.484            1.576 2010       2
## 3             27.5            1.486            1.494 2009       3
## 4             28.1            1.481            1.524 2009       4
## 5             33.8            1.544            1.885 2007       4
## 6             32.6            1.538            1.817 2008       1
```

When dealing with date and time values, it's generally a good idea to convert them to the appropriate data type.

```
housing$Date <- as.Date(housing$Date)
```

Next, we create two subsets of the data, one with housing prices only from New York, and another one with housing prices from 9 states in the North East.

```
newyork <- subset(housing, State == "NY")
northeast <- subset(housing, Region == "N. East")
```

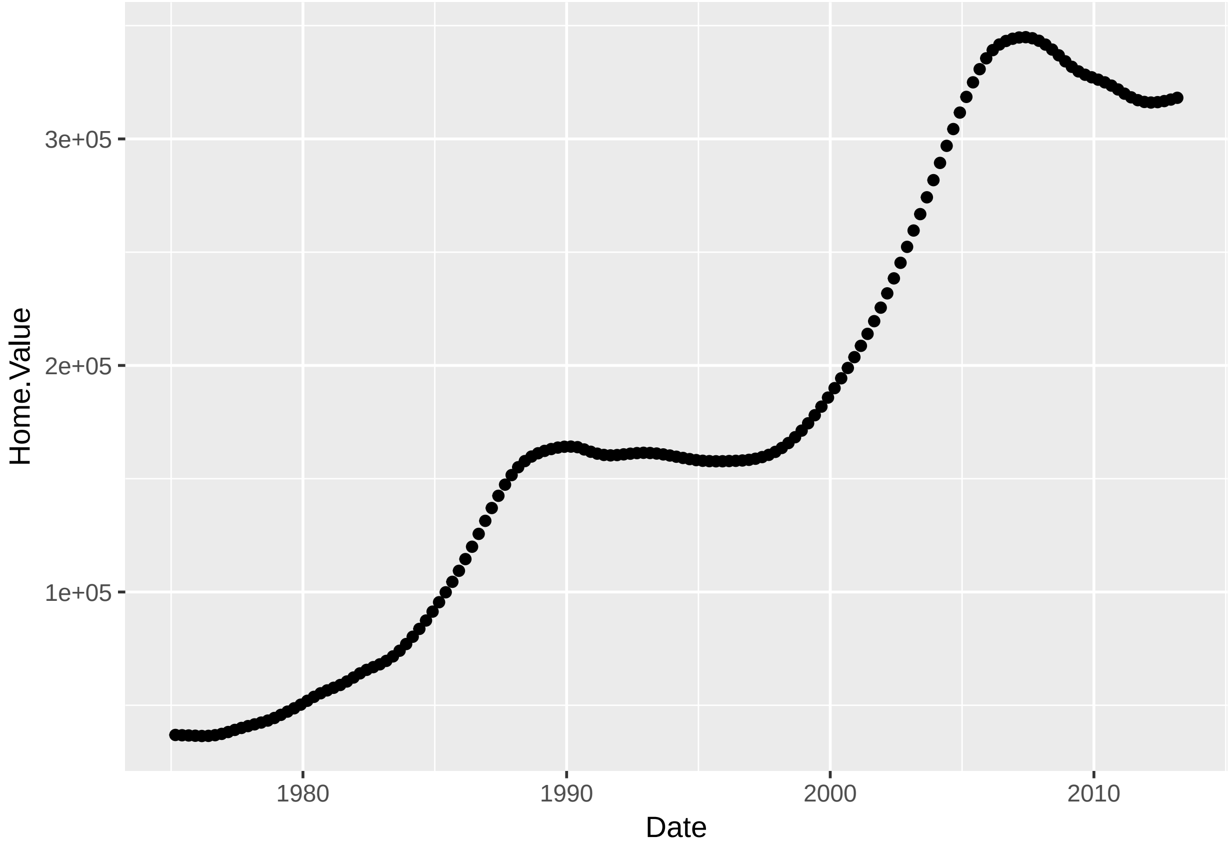
6.3 Scatter Plot

Now we're ready to plot. Everything starts with the `ggplot()` function which creates a plot object. The two arguments passed to `ggplot()` are:

Argument	Description
`data`	Dataset for the plot. It should be a data.frame or something that can be converted to data.frame
`mapping`	Aesthetic mappings for the plot

Using the `newyork` dataset, let's create a scatter plot with `Date` on the x-axis and `Home.Value` on the y-axis.

```
ggplot(newyork, aes(x = Date, y = Home.Value)) +
  geom_point()
```



Now let's see which ggplot building blocks are active in the above example:

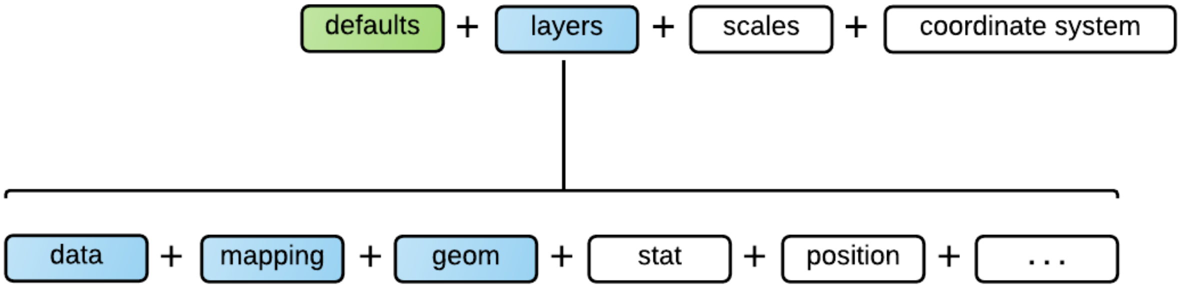


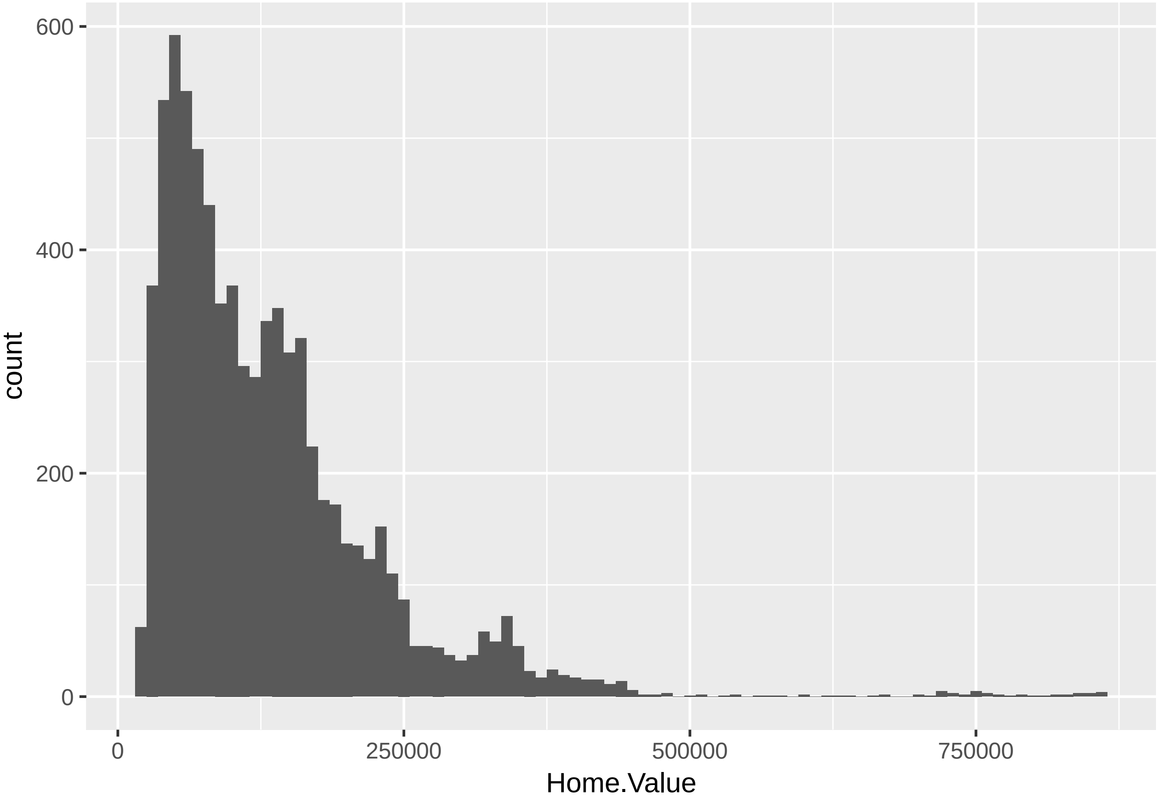
Figure 2:

Data	`newyork`
Mapping	`aes(x = Date, y = Home.Value)`
Geom	`geom_point()`

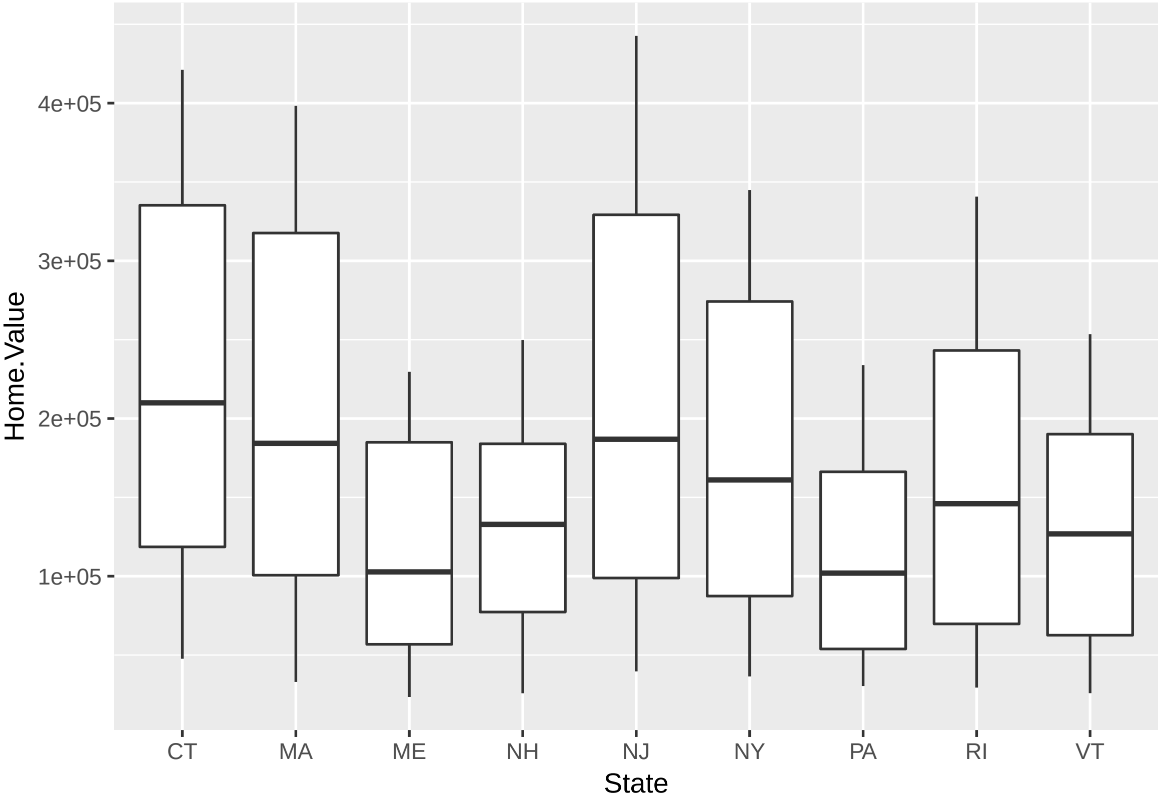
6.4 Exercise

Use the Data Visualization with ggplot2 Cheat Sheet or any other resource to find out how to complete the exercises.

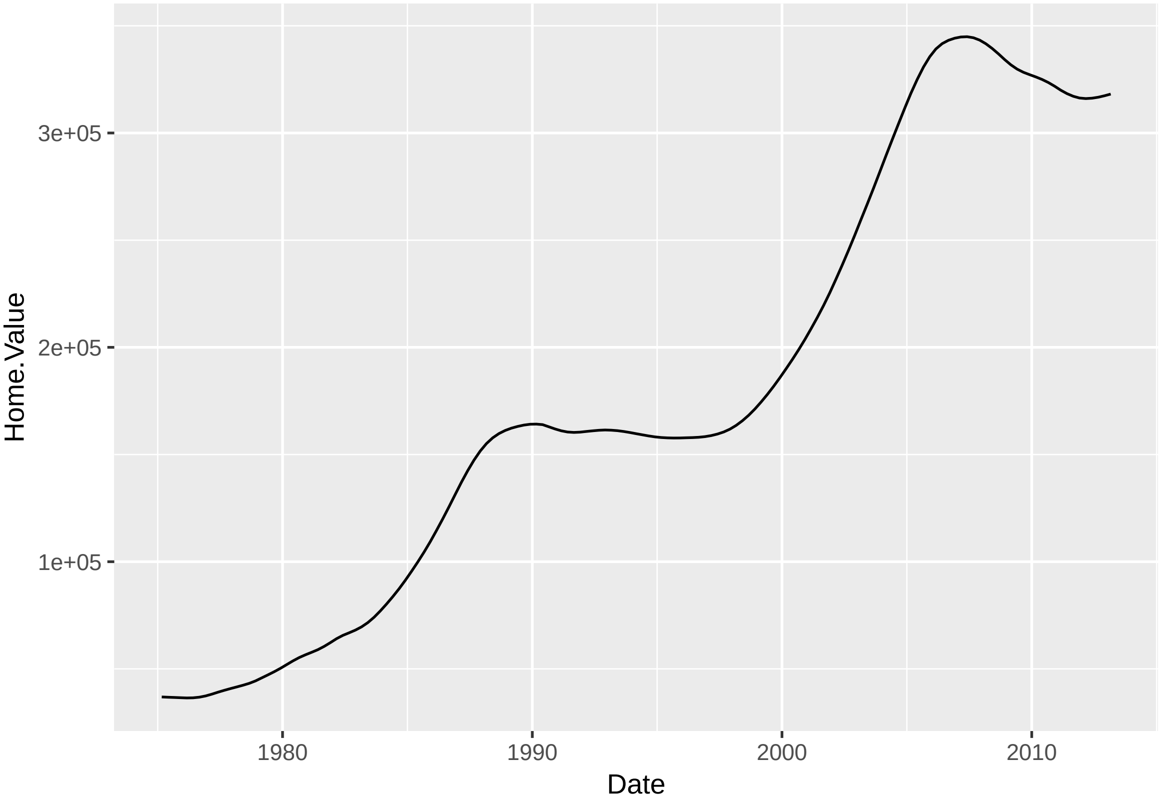
1. Create a histogram of `Home.Value` using the `housing` data.



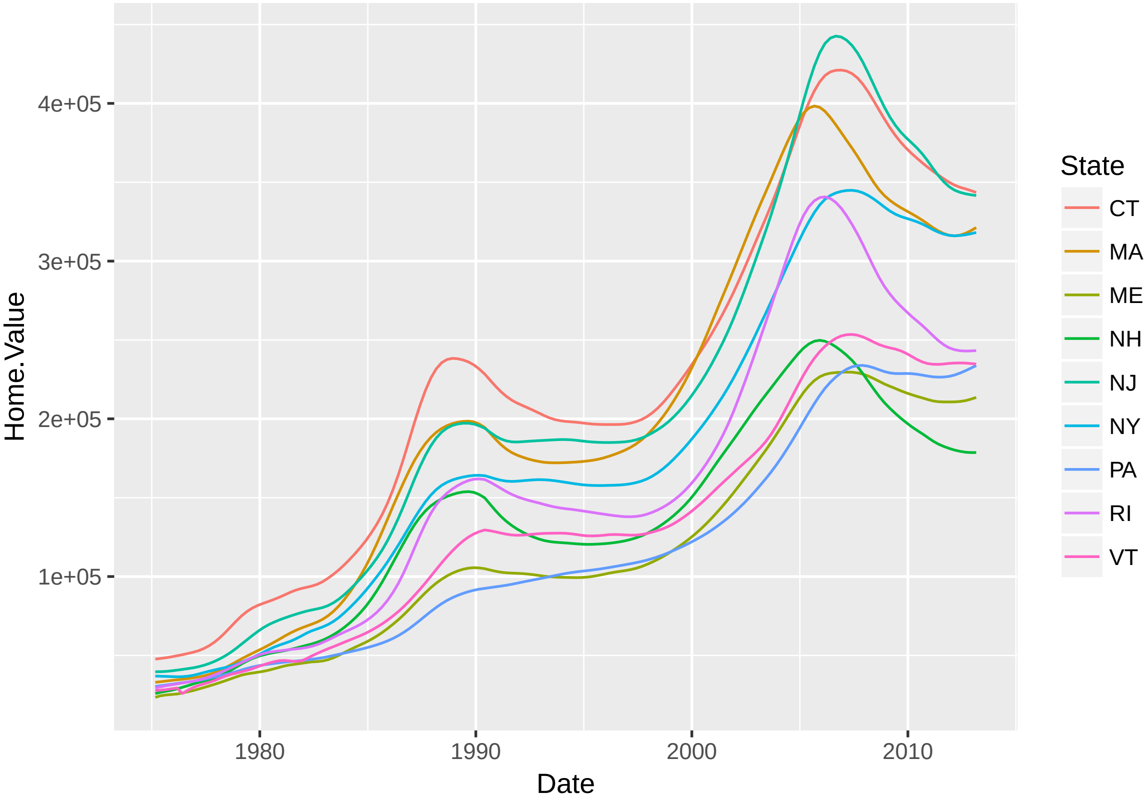
2. Create a box plot of `Home.Value` using `northeast` dataset with `State` on the x-axis



3. Create a line plot using `newyork` dataset with `Date` on the x-axis and `Home.Value` on the y-axis



4. Create a line plot using `northeast` dataset with `Date` on the x-axis and `Home.Value` on the y-axis and use a different color for each state



7 Geoms and Statistics

Geometric objects (geoms) define the basic shape of the elements on the plot.

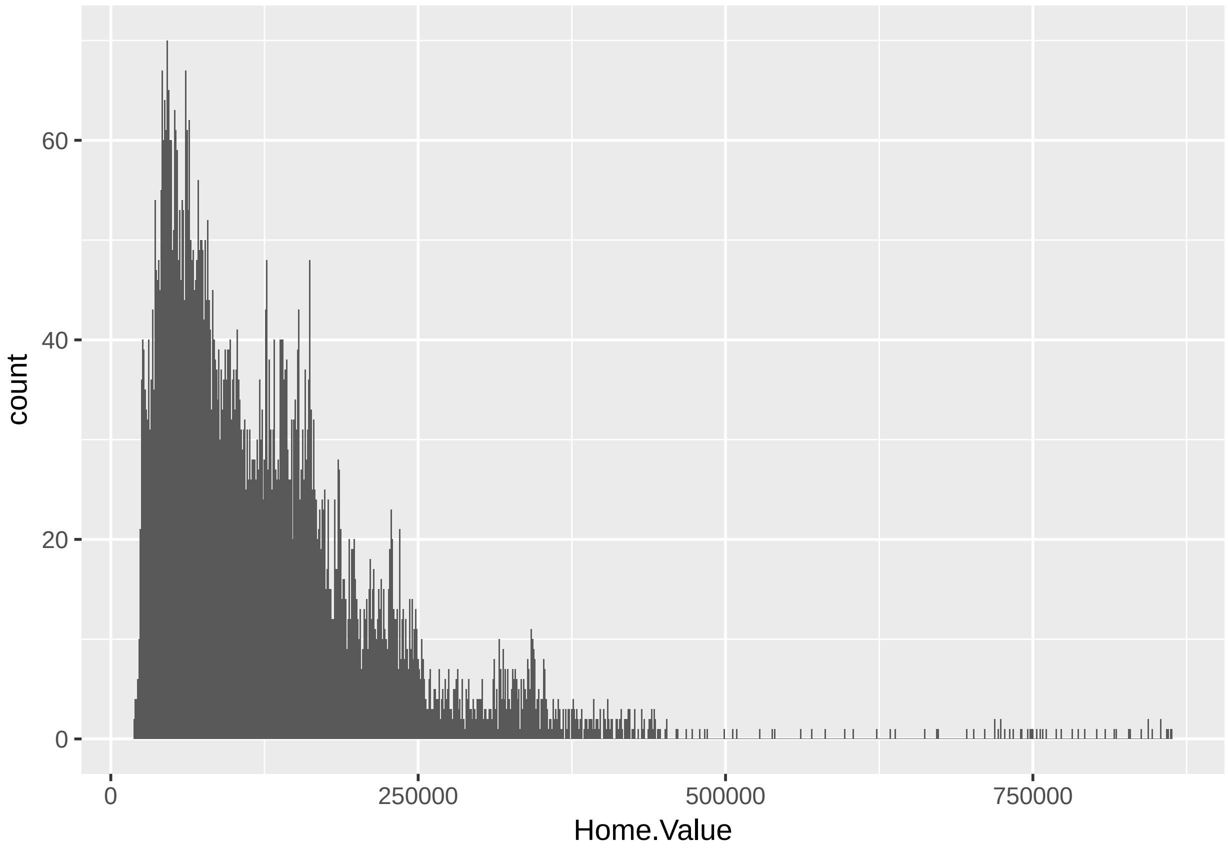
- Every geom has a default statistic
- Every statistic has a default geom

You can get a list of all geoms using the online help in RStudio

```
help.search("geom_", package = "ggplot2")
```

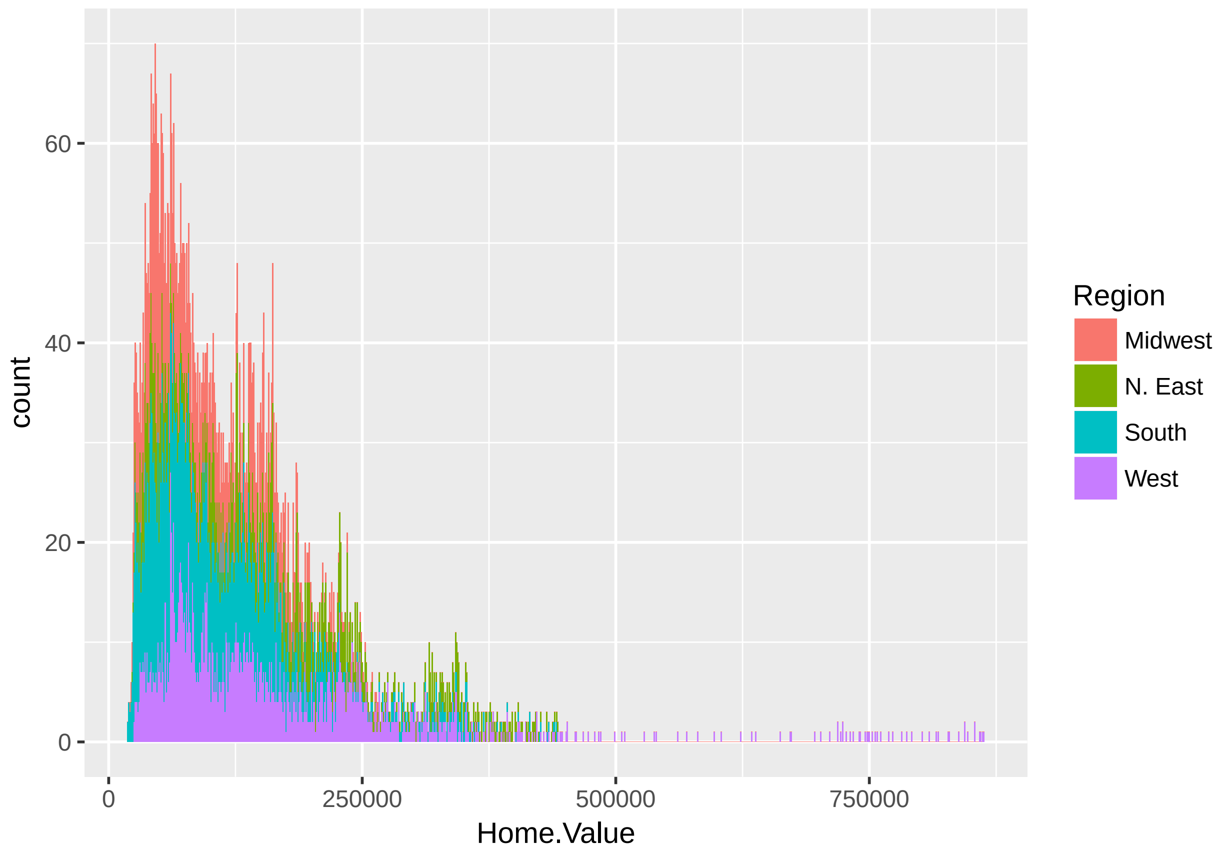
Change the size of each bin:

```
ggplot(housing, aes(x = Home.Value)) +
  geom_histogram(binwidth = 1000)
```



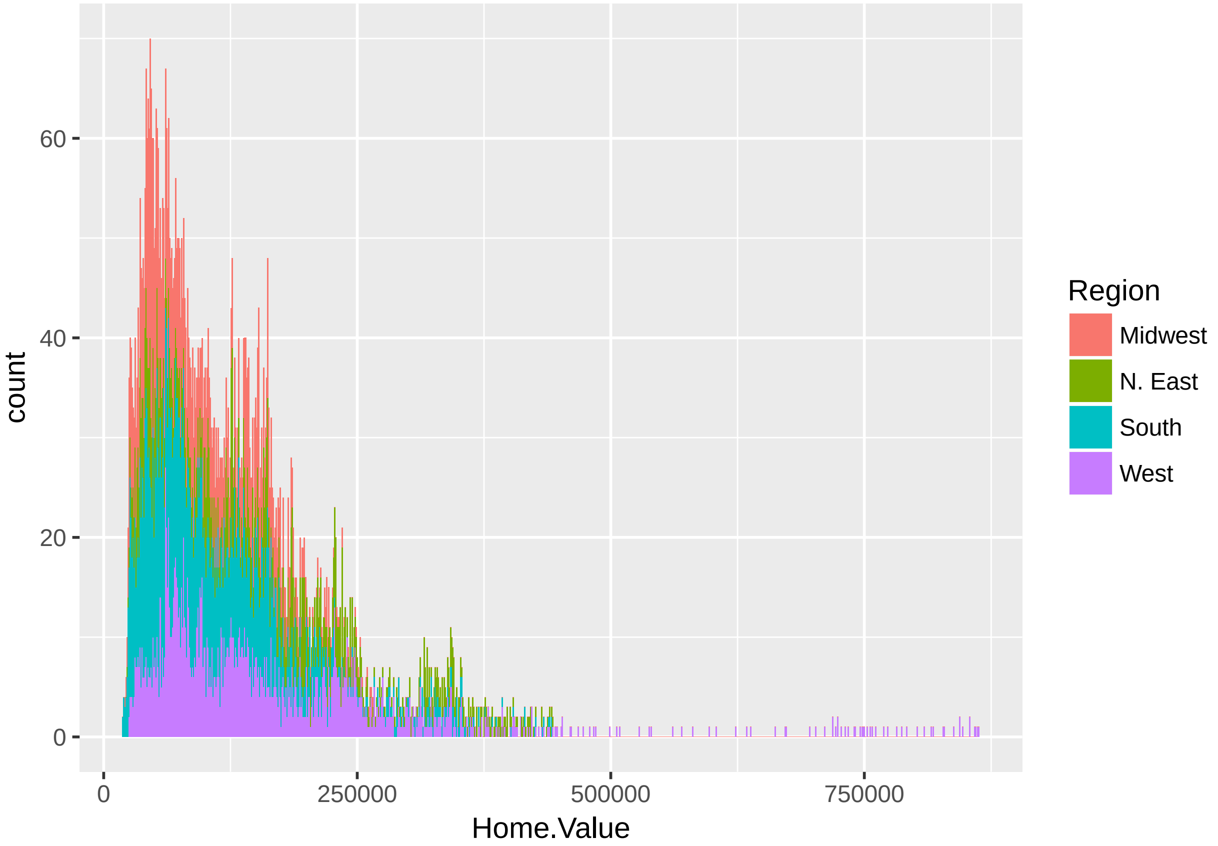
Add a mapping for the fill color:

```
ggplot(housing, aes(x = Home.Value, fill = Region)) +
  geom_histogram(binwidth = 1000)
```



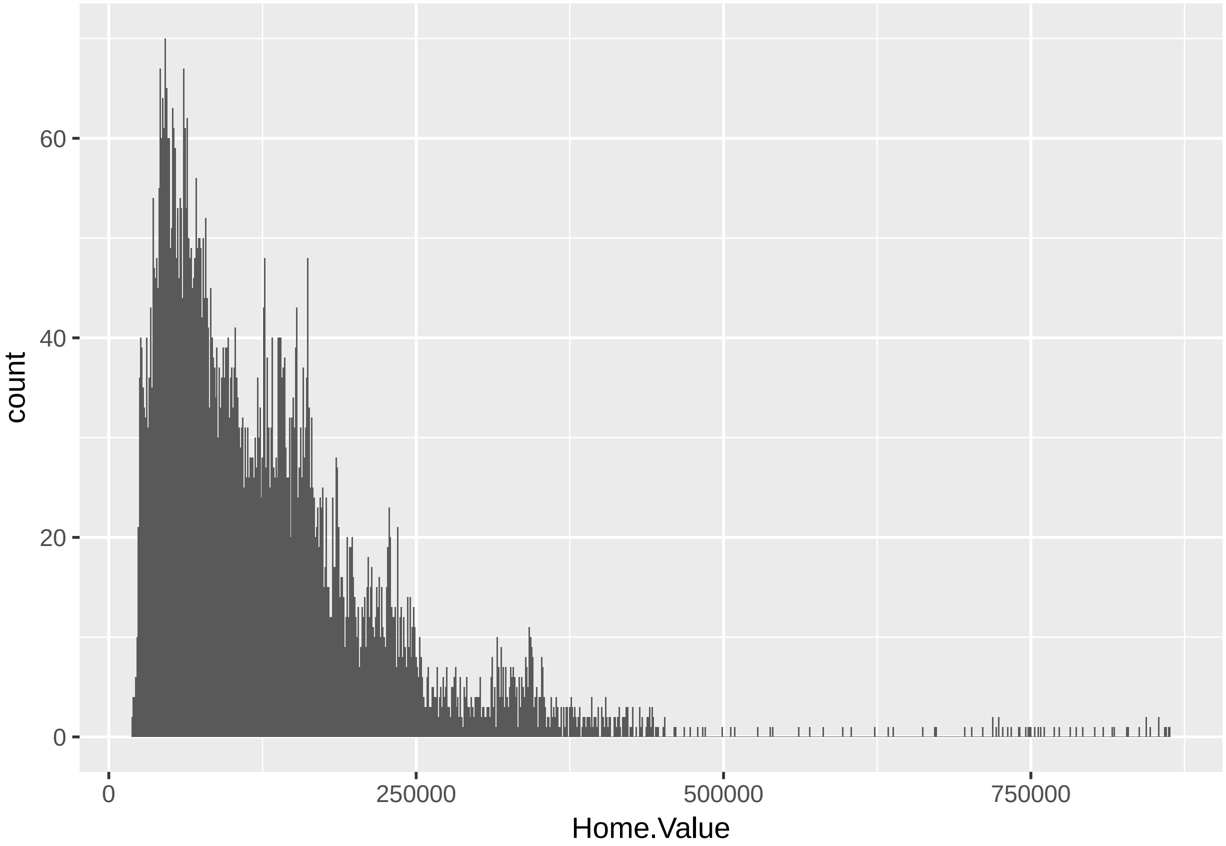
Mapping can also be specified in the geom:

```
ggplot(housing, aes(x = Home.Value)) +
  geom_histogram(aes(fill = Region), binwidth = 1000)
```



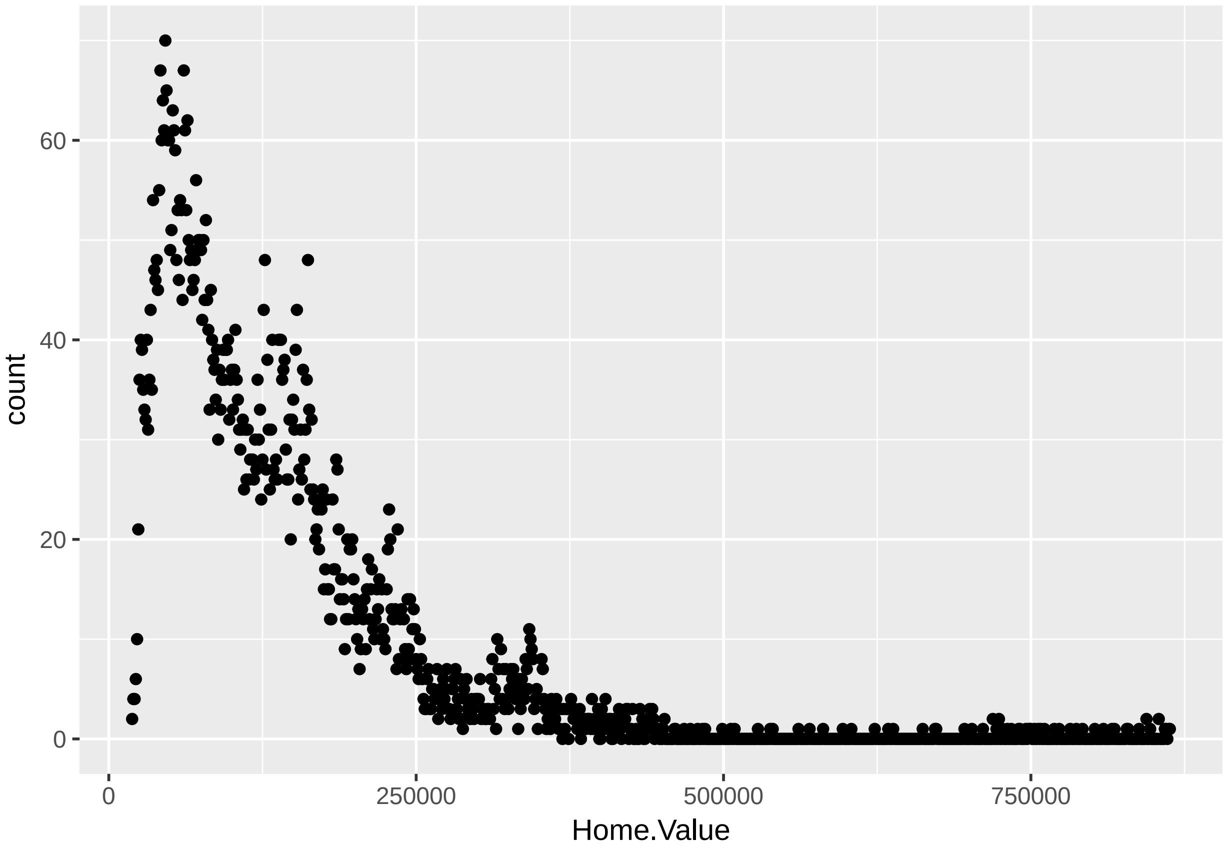
Same plot can also be created using `stat_bin` transformation. The default geom for stat_bin is "area"

```
ggplot(housing, aes(x = Home.Value)) +
  stat_bin(binwidth = 1000)
```



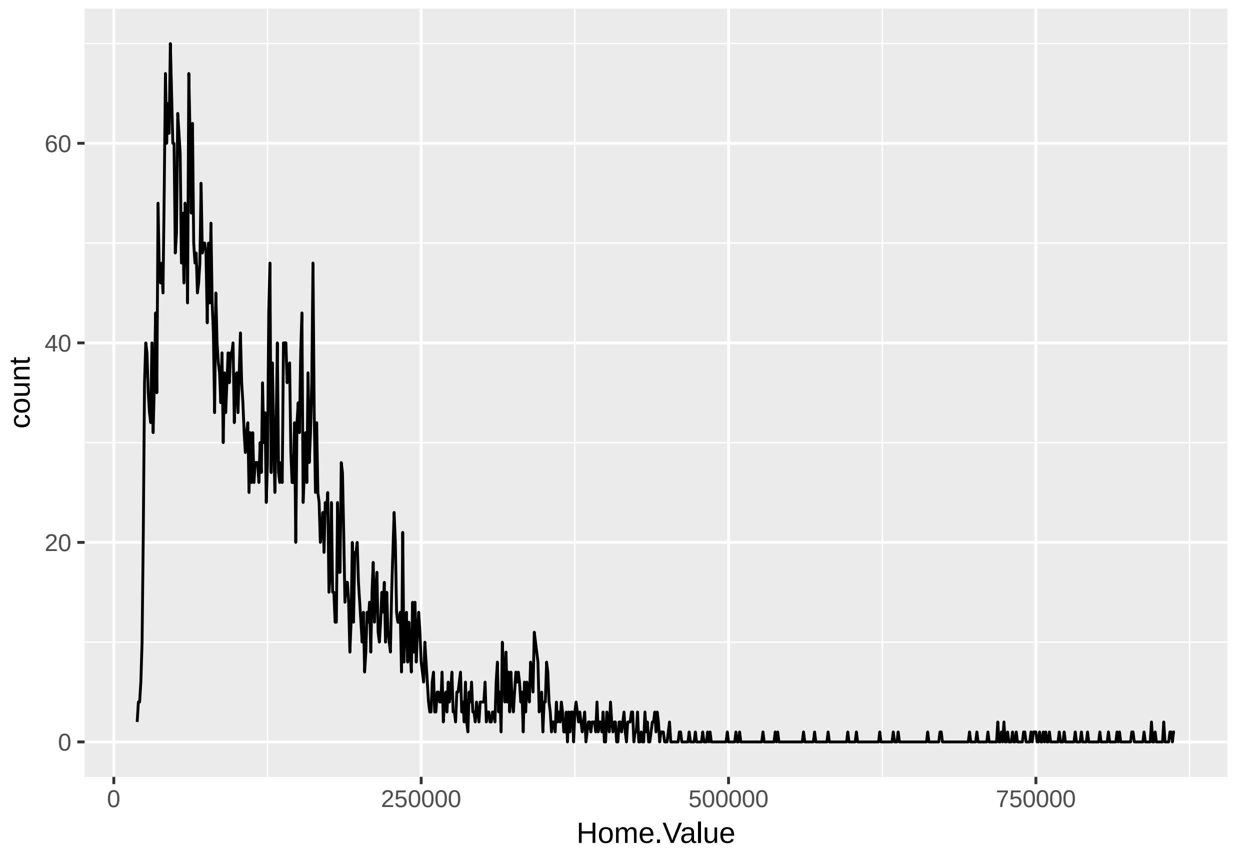
Change the default geom to "point"

```
ggplot(housing, aes(x = Home.Value)) +
  stat_bin(geom = "point", binwidth = 1000)
```



Change the default geom to "line"

```
ggplot(housing, aes(x = Home.Value)) +
  stat_bin(geom = "line", binwidth = 1000)
```

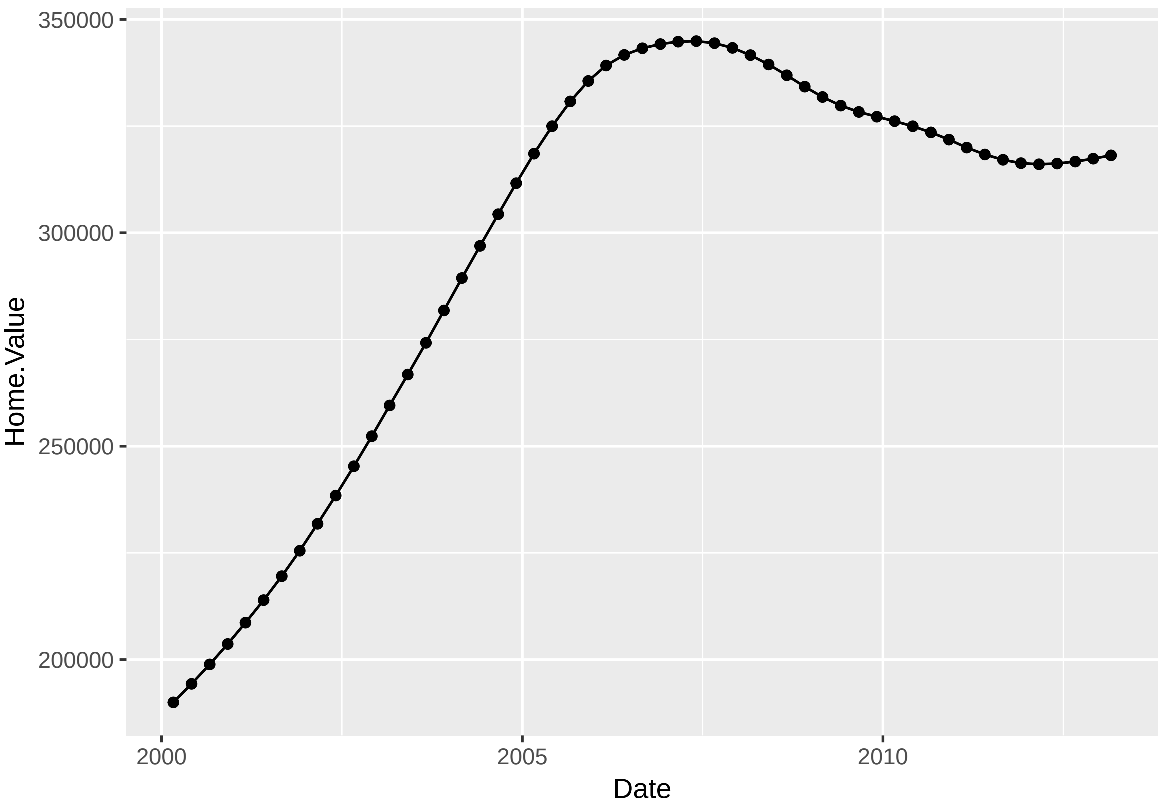


7.1 Exercise

Create a subset of housing data from New York since 2000

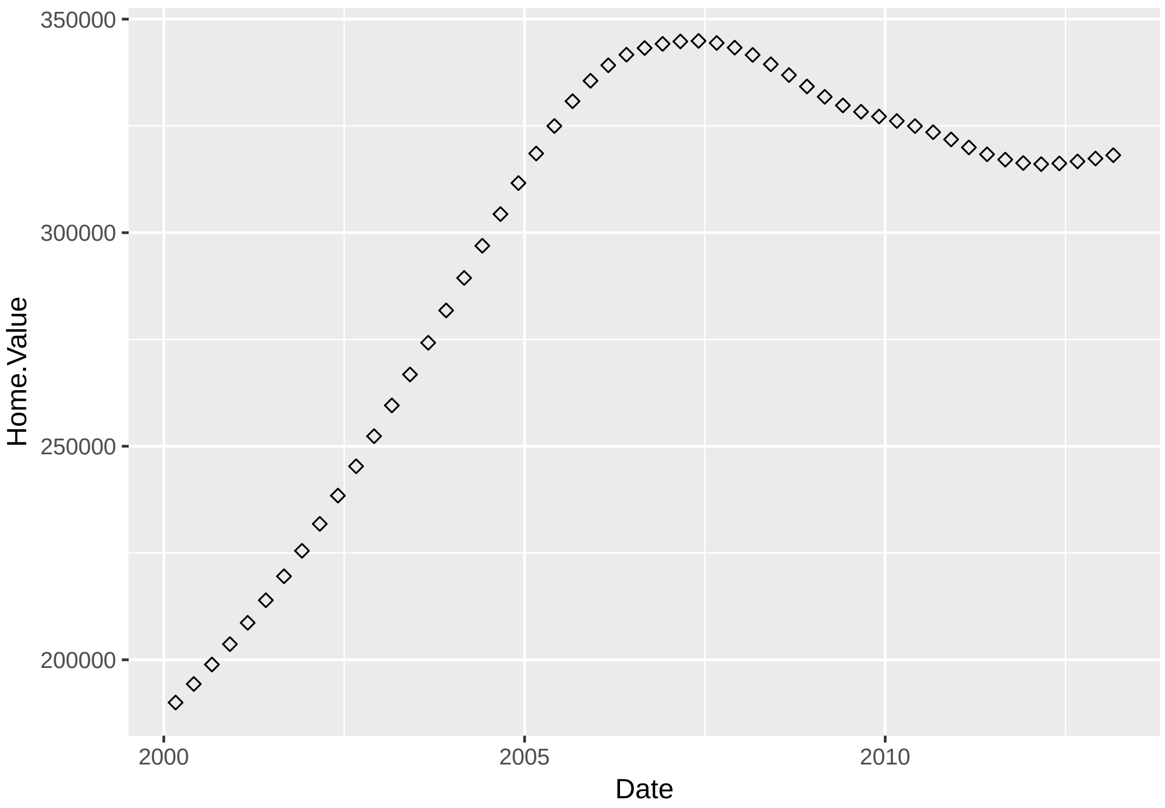
```
newyork2k <- subset(newyork, Year >= 2000)
```

1. Create a plot that includes multiple geometric objects, for example, lines and points.

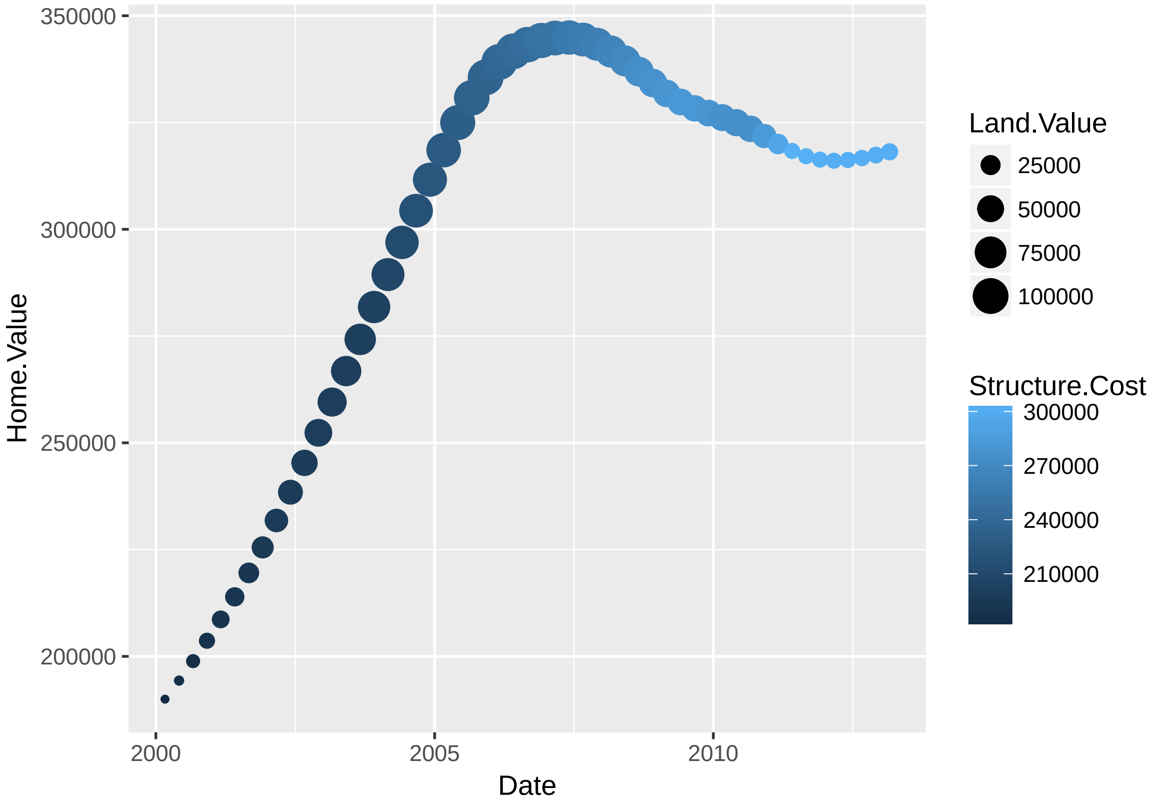


2. Change the shape to be hollow diamond

   HINT: Take a look at **Shape Scales** in the Data Visualization with ggplot2 Cheat Sheet



3. Change the size of the point based on `Land.Value` and color based on `Structure.Cost`



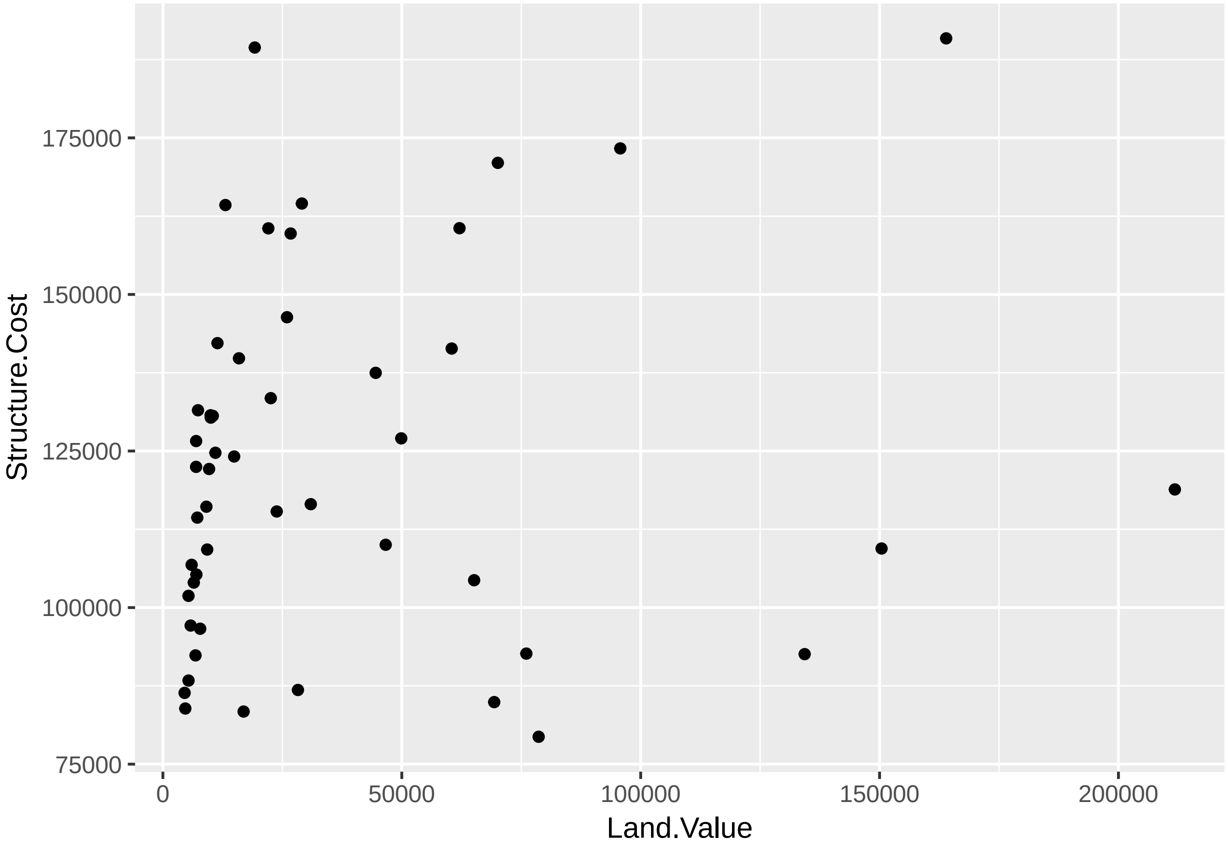
8 Scales

Let's create another subset that includes only the data from the first quarter of 2001.

```
housing2001q1 <- subset(housing, Year == 2001 & Quarter == 1)
```

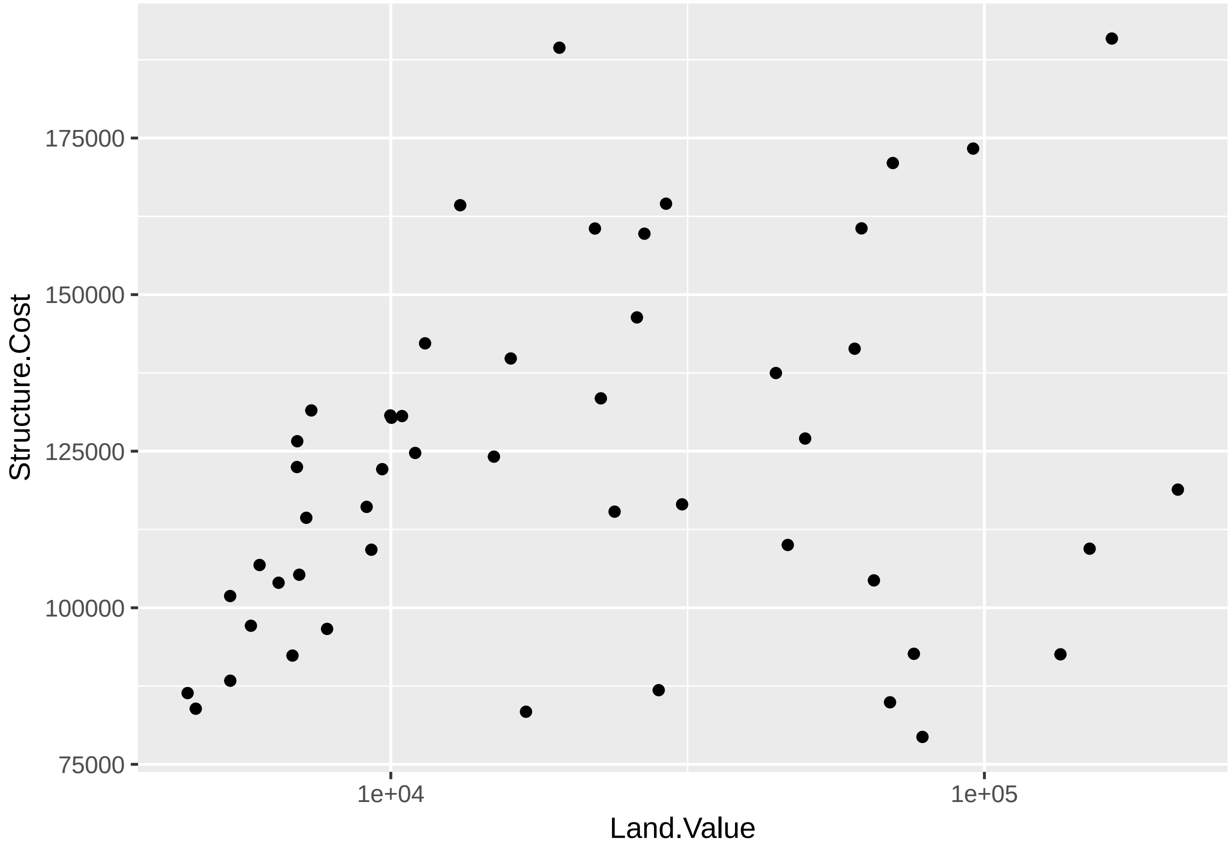
And now we create a scatter plot with this dataset

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point()
```



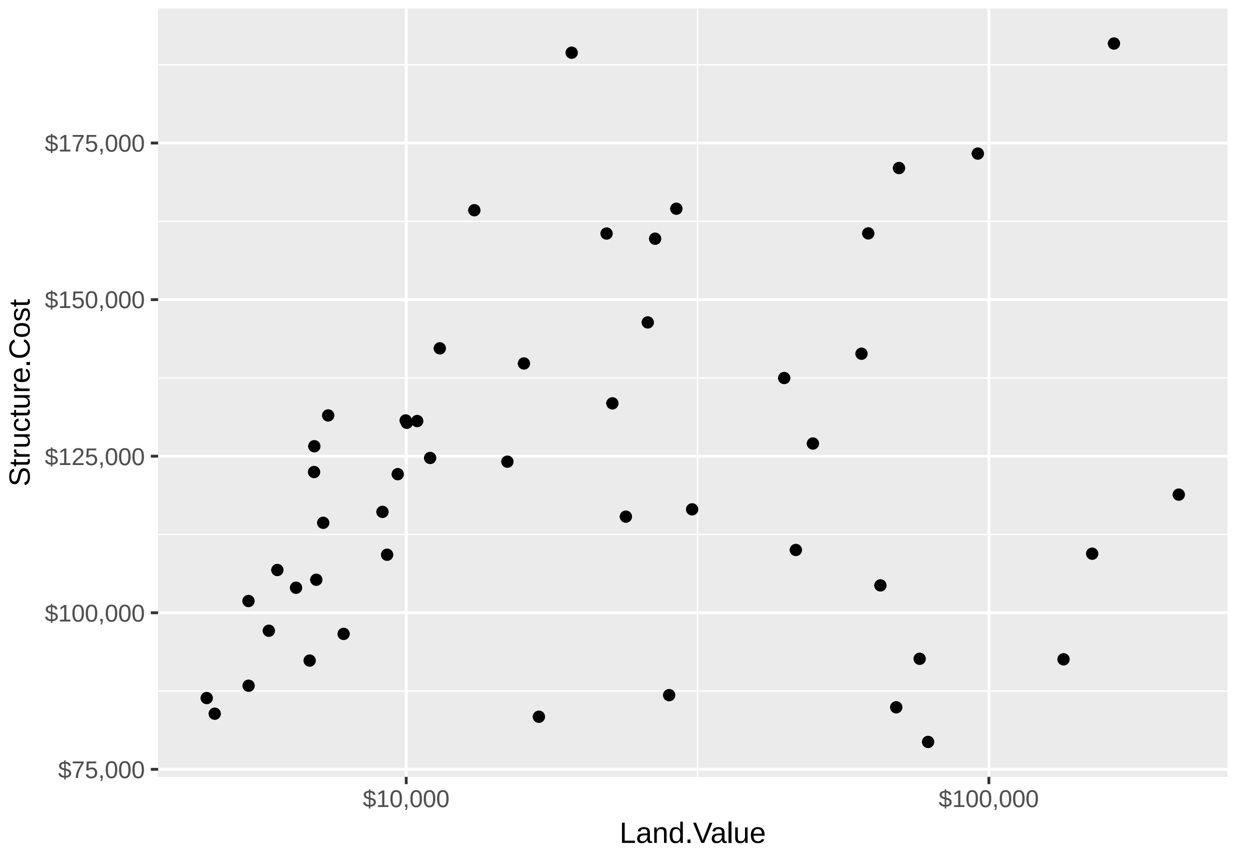
Our dataset is skewed so in order to help with interpretation, let's change the x-axis to log scale

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10()
```



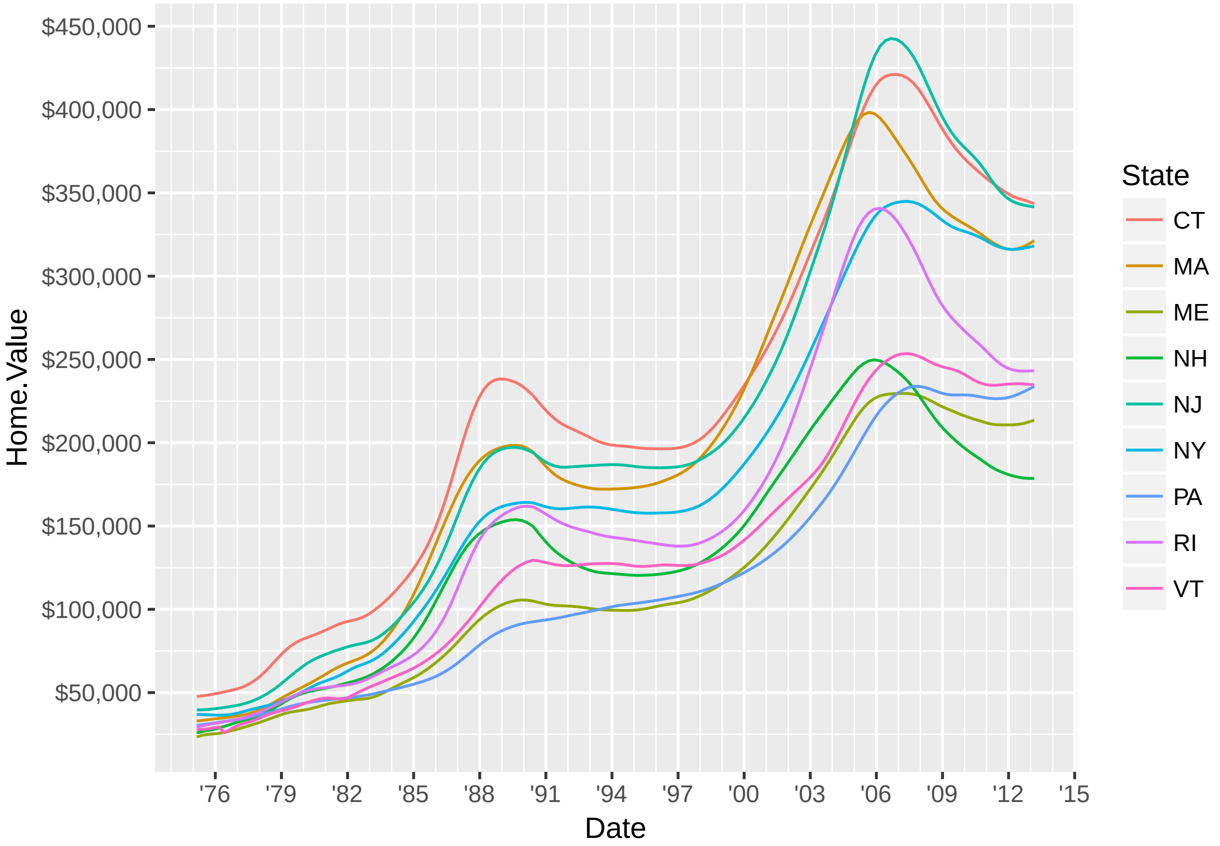
Now add a dollar sign in front of our axis labels

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10(labels = dollar) +
  scale_y_continuous(labels = dollar)
```



Next we change the scale for the x-axis which is in a Date format and control the breaks for y-axis which is a continuous variable.

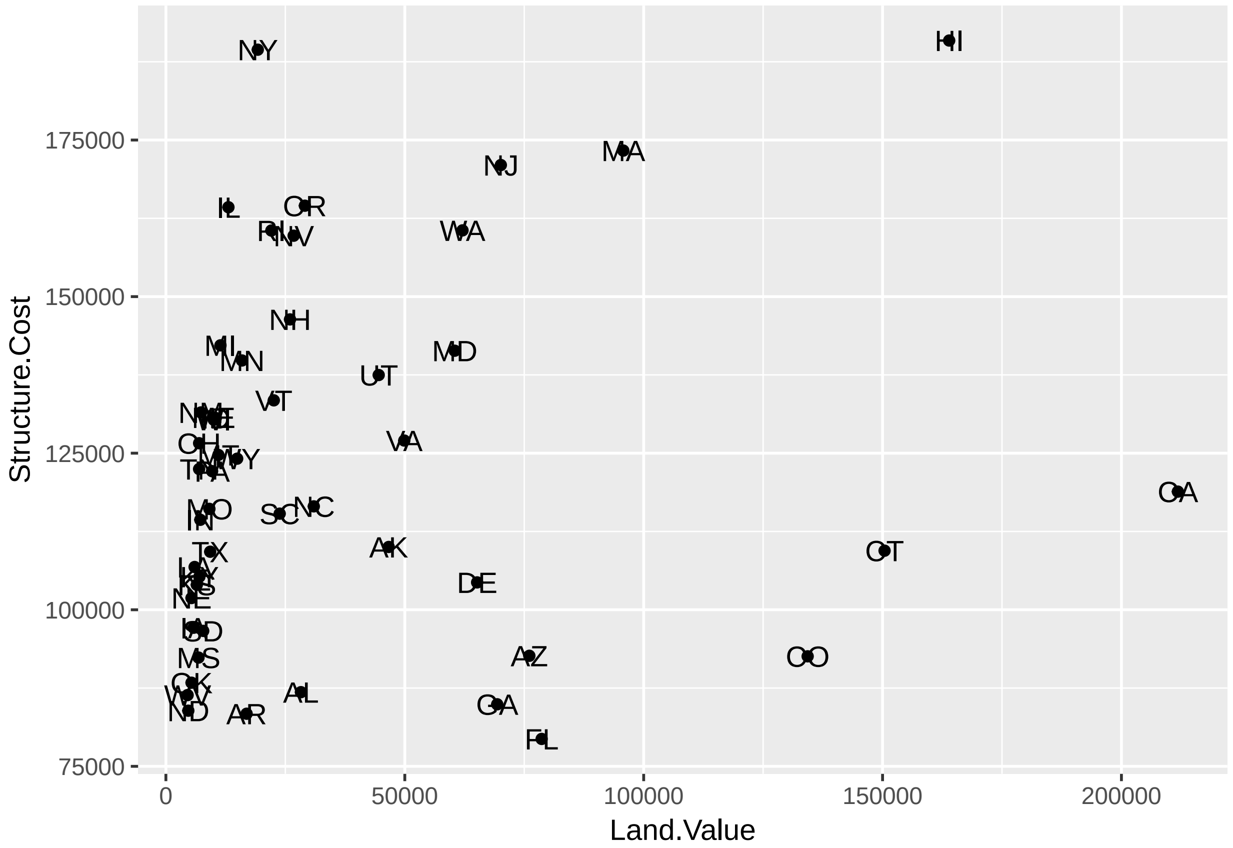
```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  scale_x_date(date_breaks ="3 year", date_minor_breaks ="1 year", date_labels = "'%y") +
  scale_y_continuous(breaks = seq(0, 500000, 50000), labels = dollar)
```



9 Text and Labels

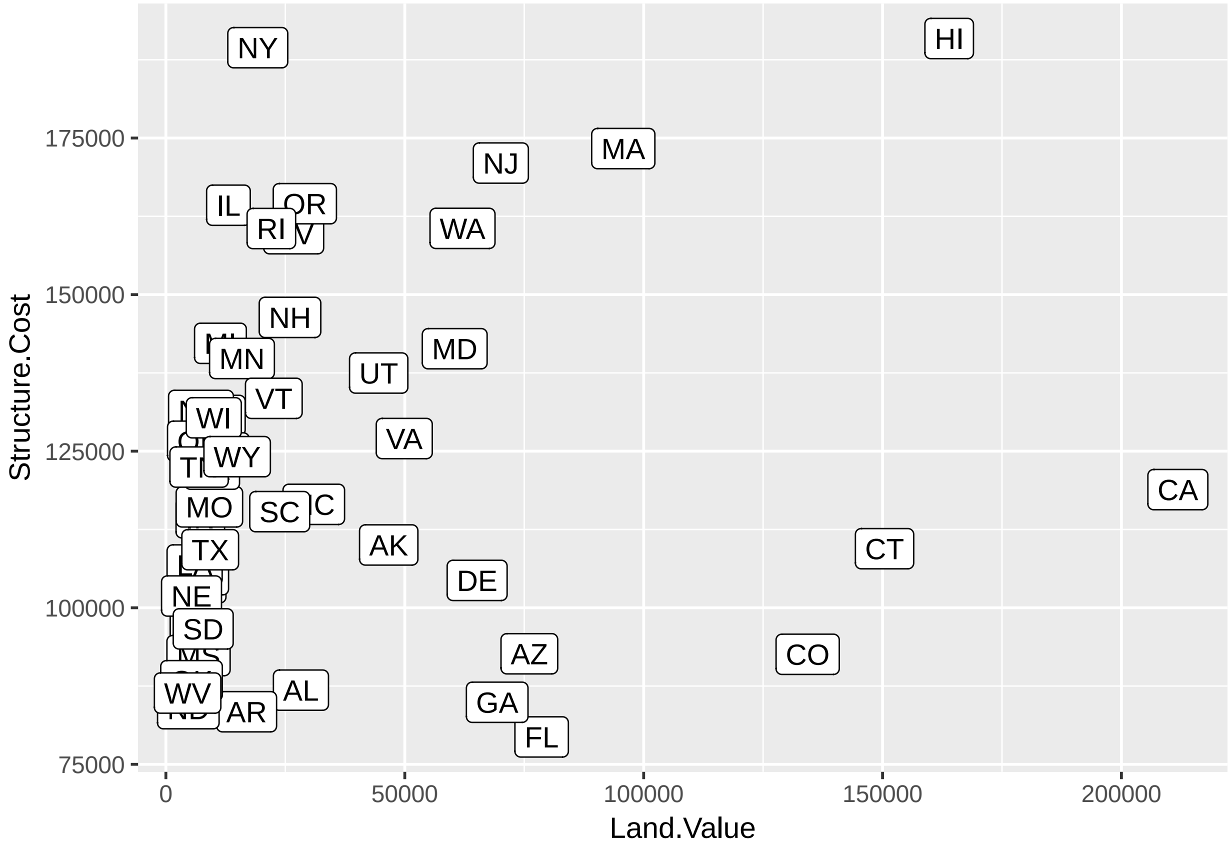
Let's continue with the subset of the data from the previous section and add text to the scatterplot.

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  geom_text(aes(label = State))
```



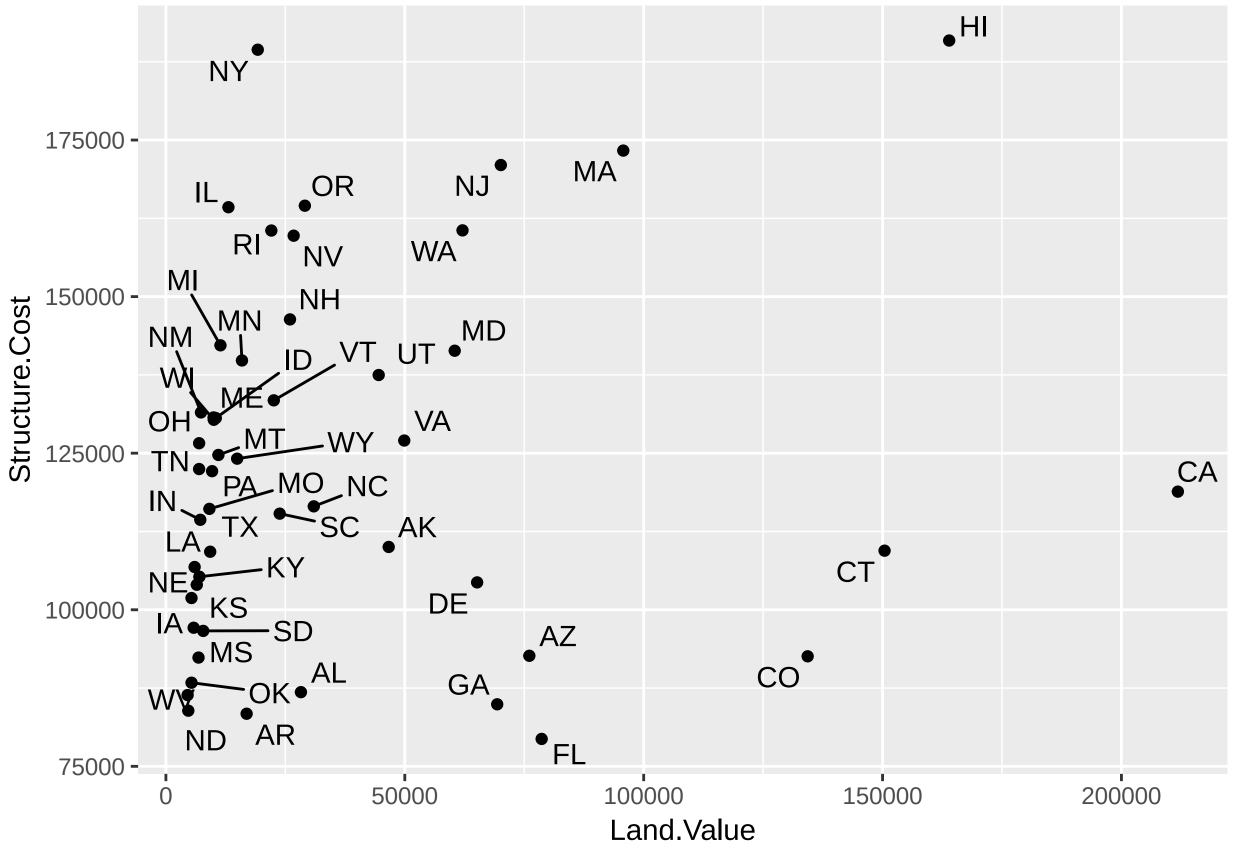
The result isn't very nice as the labels overlap each other. Let's try the same with `geom_label()` instead which draws the text with a border around it.

```r
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  geom_label(aes(label = State))
```



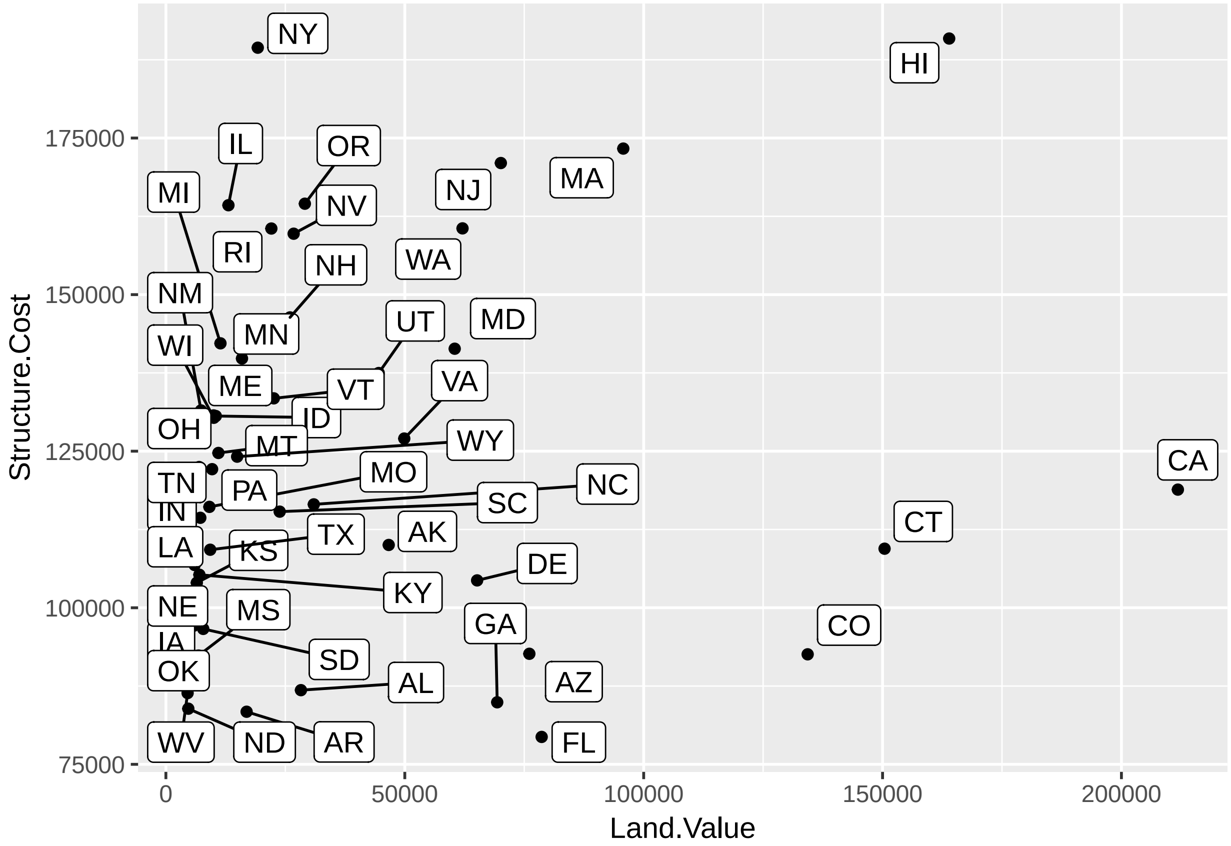
The `ggrepel` extension we loaded earlier can also help fix this problem.

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  geom_text_repel(aes(label = State))
```



And we can repel the labels with a border using `geom_label_repel()`.

```r
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  geom_label_repel(aes(label = State))
```

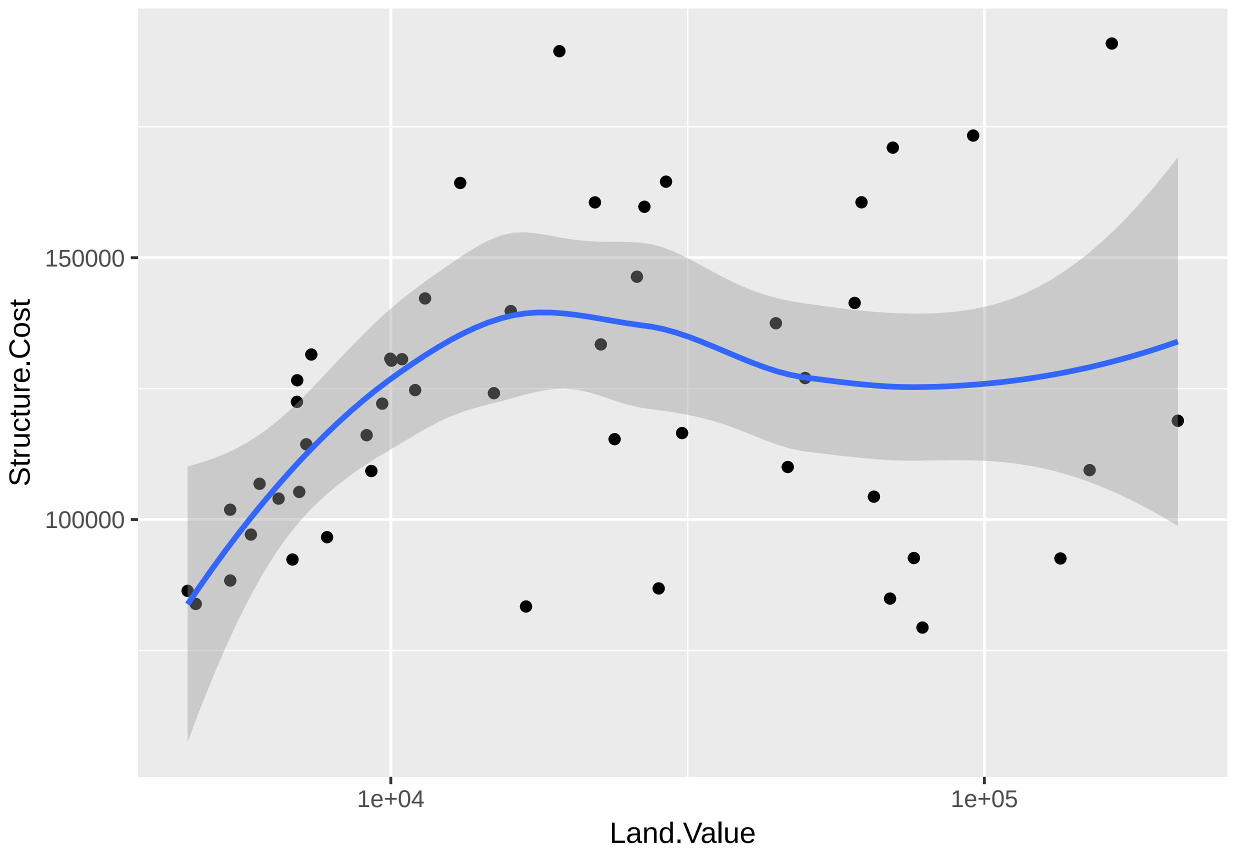


10 Smoother

Let's continue with the 2001 first quarter dataset and add a smoother.

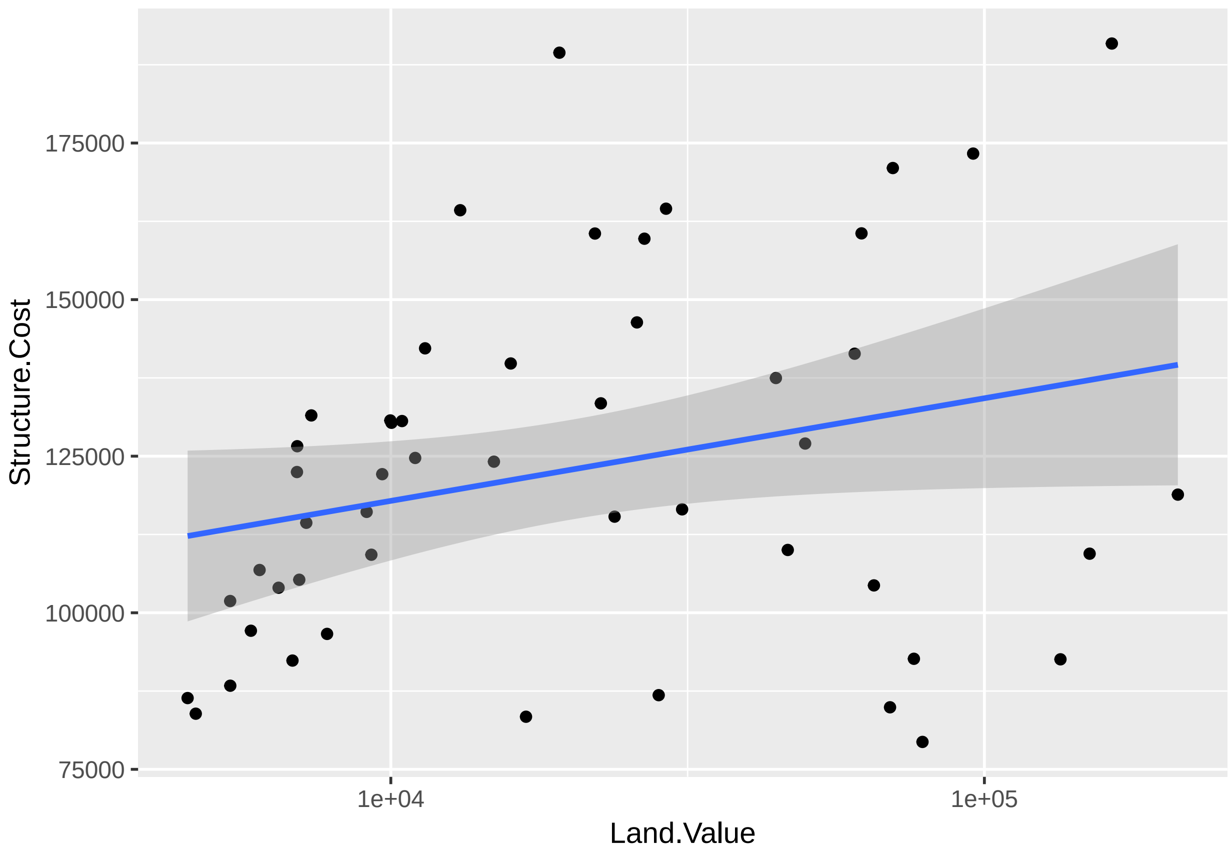
```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10() +
  stat_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



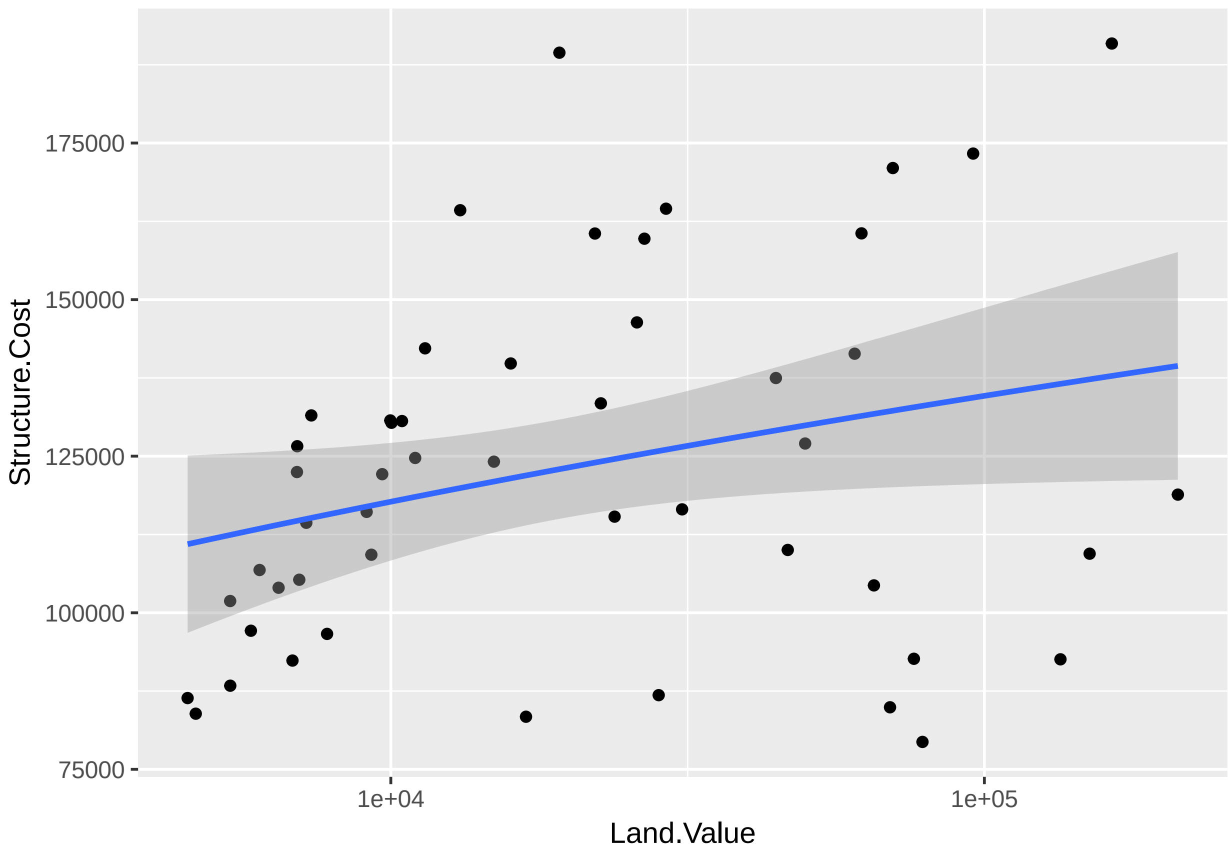
We can fit a linear model to our dataset

```r
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10() +
  stat_smooth(method = "lm")
```



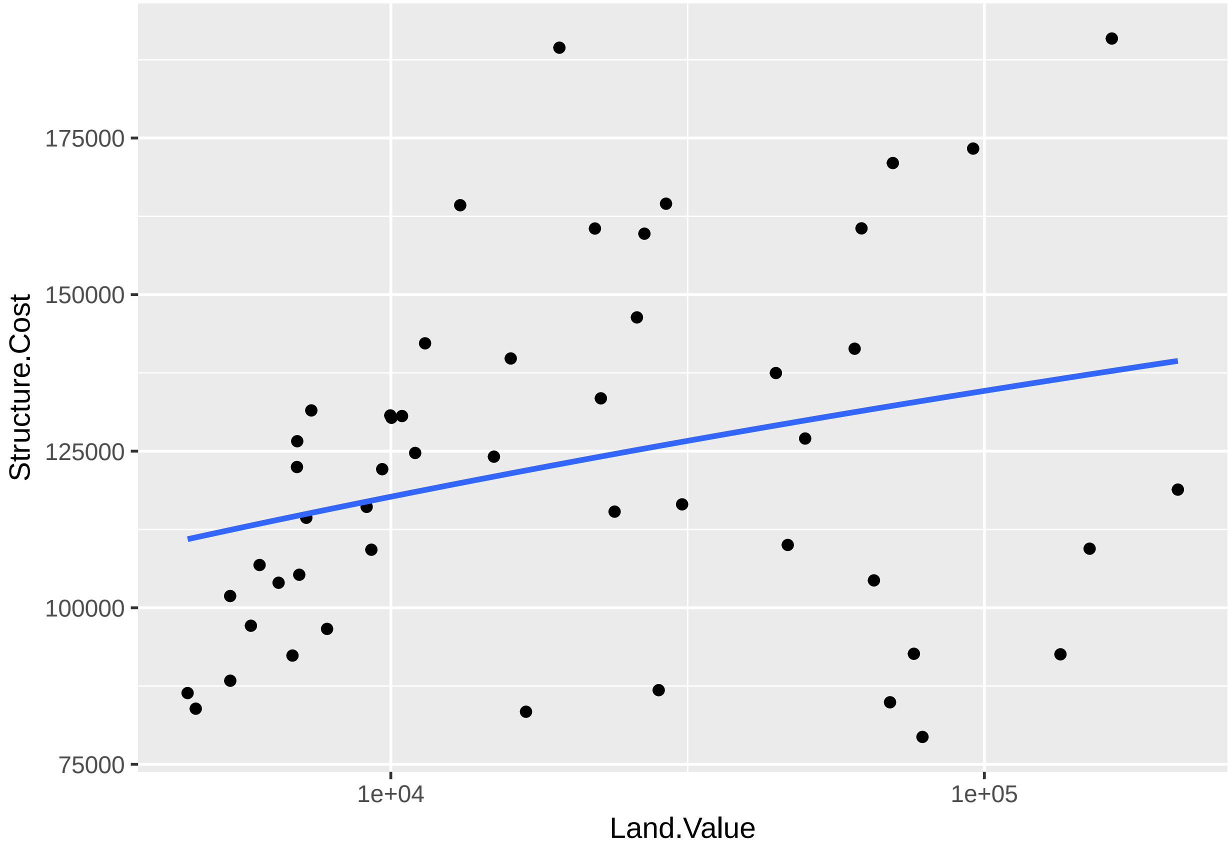
We can also specify the formula for the model

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10() +
  stat_smooth(method = "lm", formula = y ~ log(x))
```



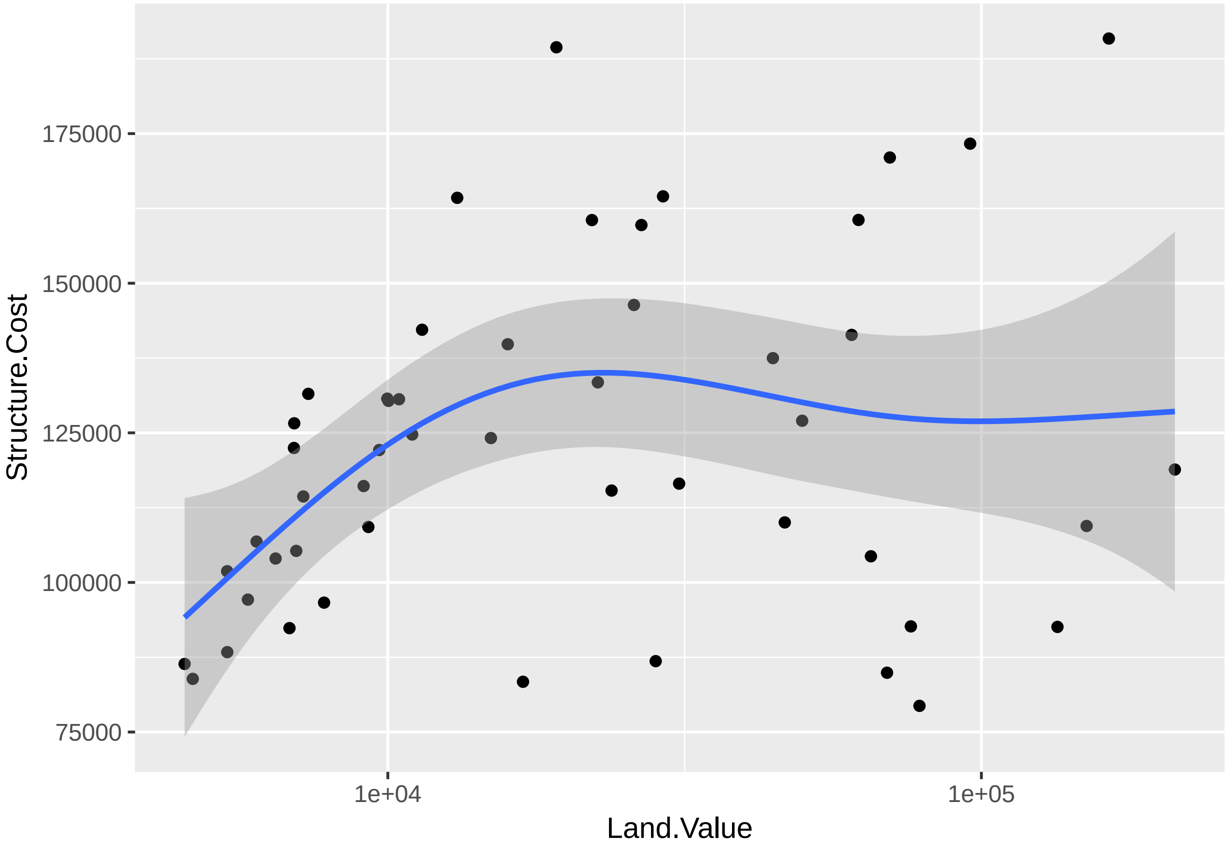
We can turn the turn off the confidence interval

```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10() +
  stat_smooth(method = "lm", formula = y ~ log(x), se = FALSE)
```



Formula is specific to the type of model used. Here we're using a General Additive Model (GAM).

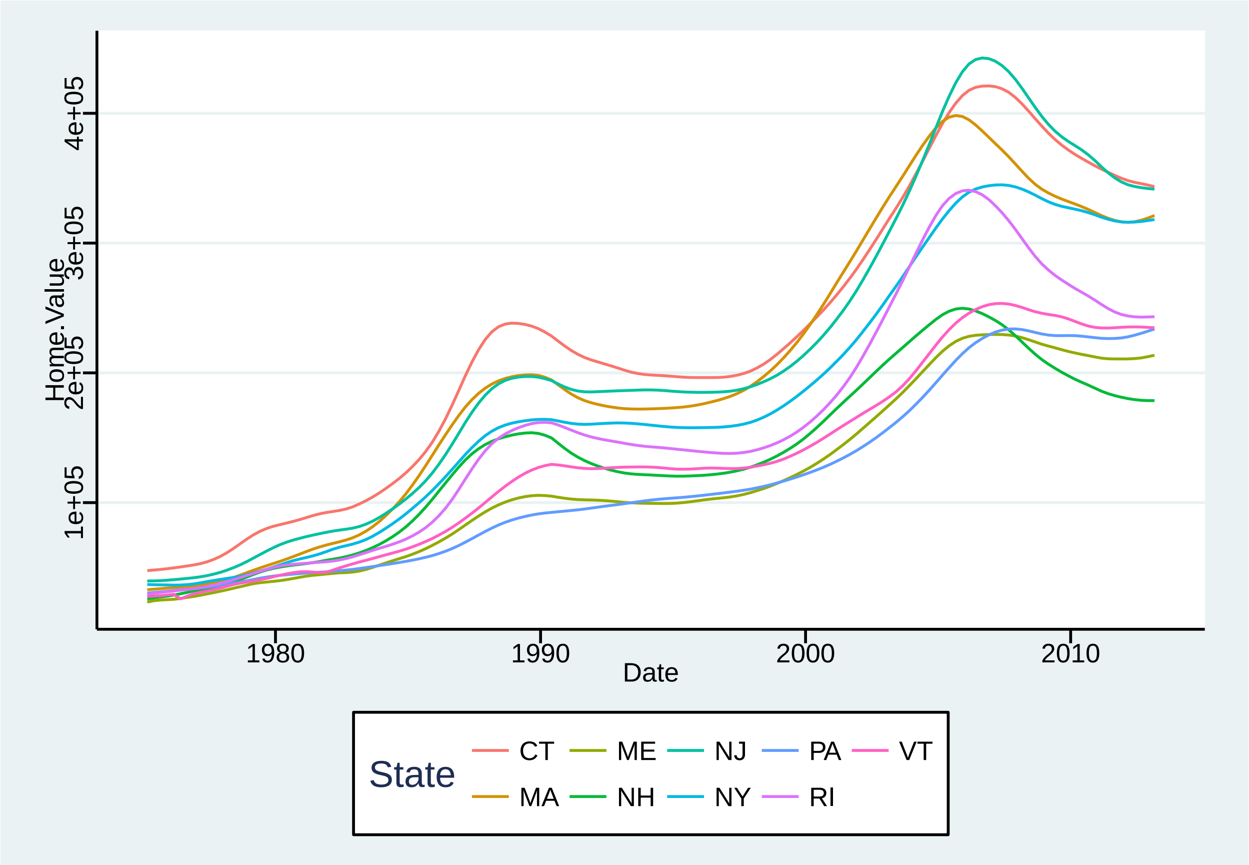
```
ggplot(housing2001q1, aes(x = Land.Value, y = Structure.Cost)) +
  geom_point() +
  scale_x_log10() +
  stat_smooth(method = "gam", formula = y ~ s(x,k=10))
```



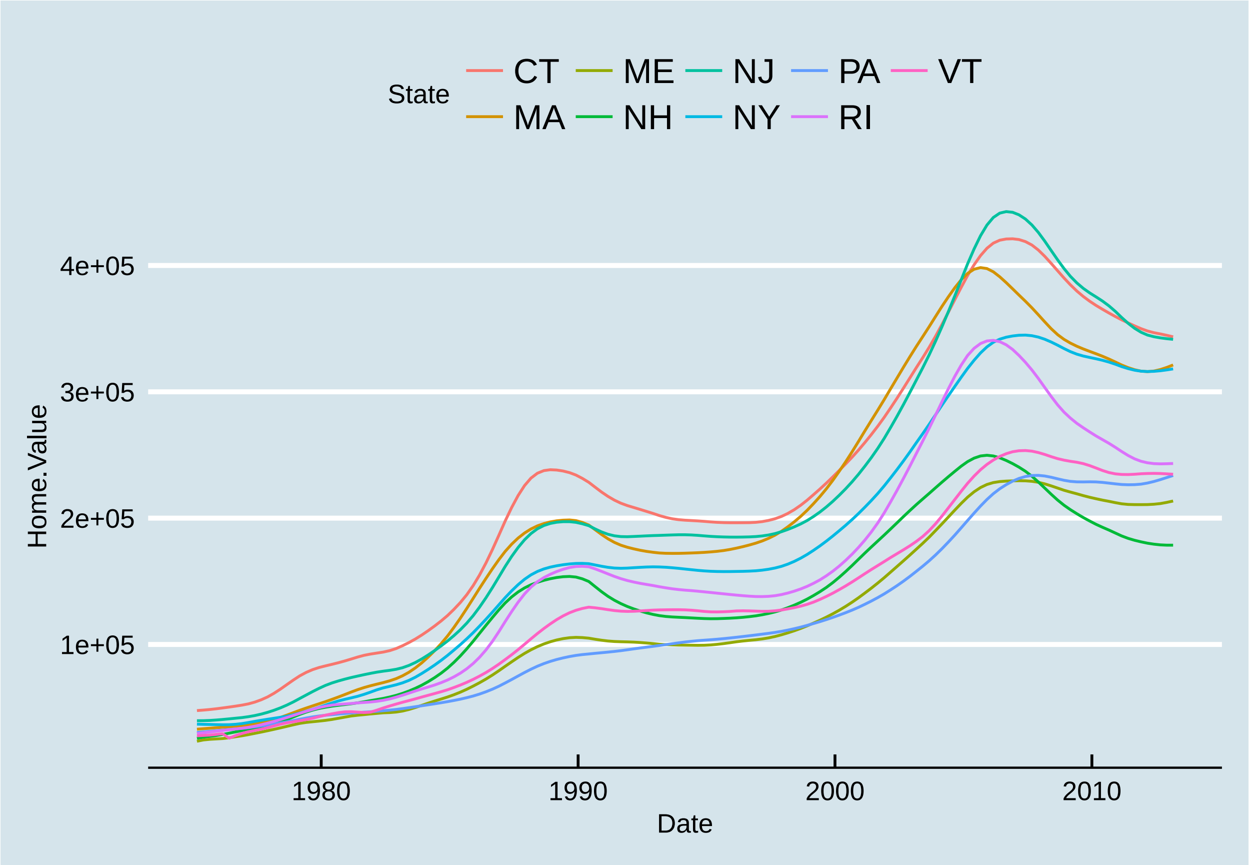
11 Theme and Title

First, let’s try some of the themes from the `ggthemes` package

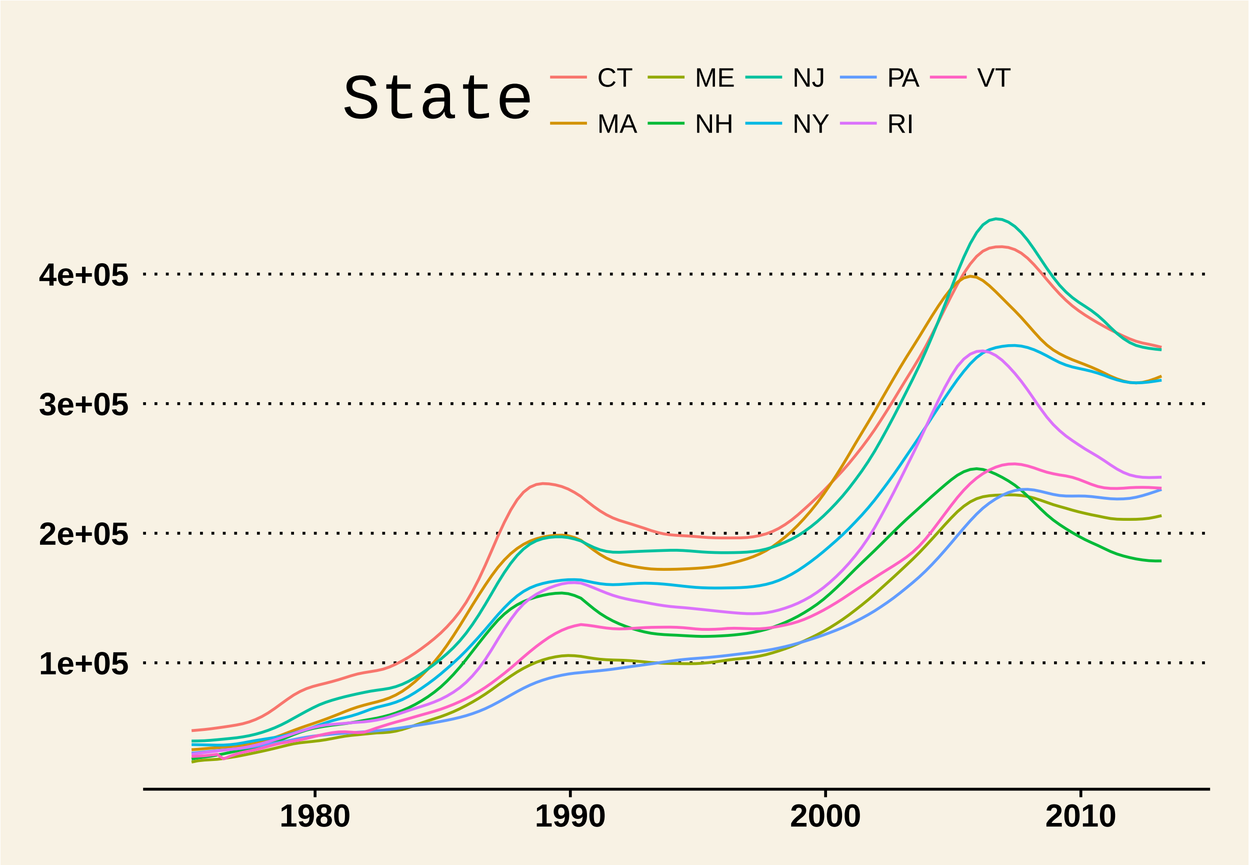
```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_stata()
```



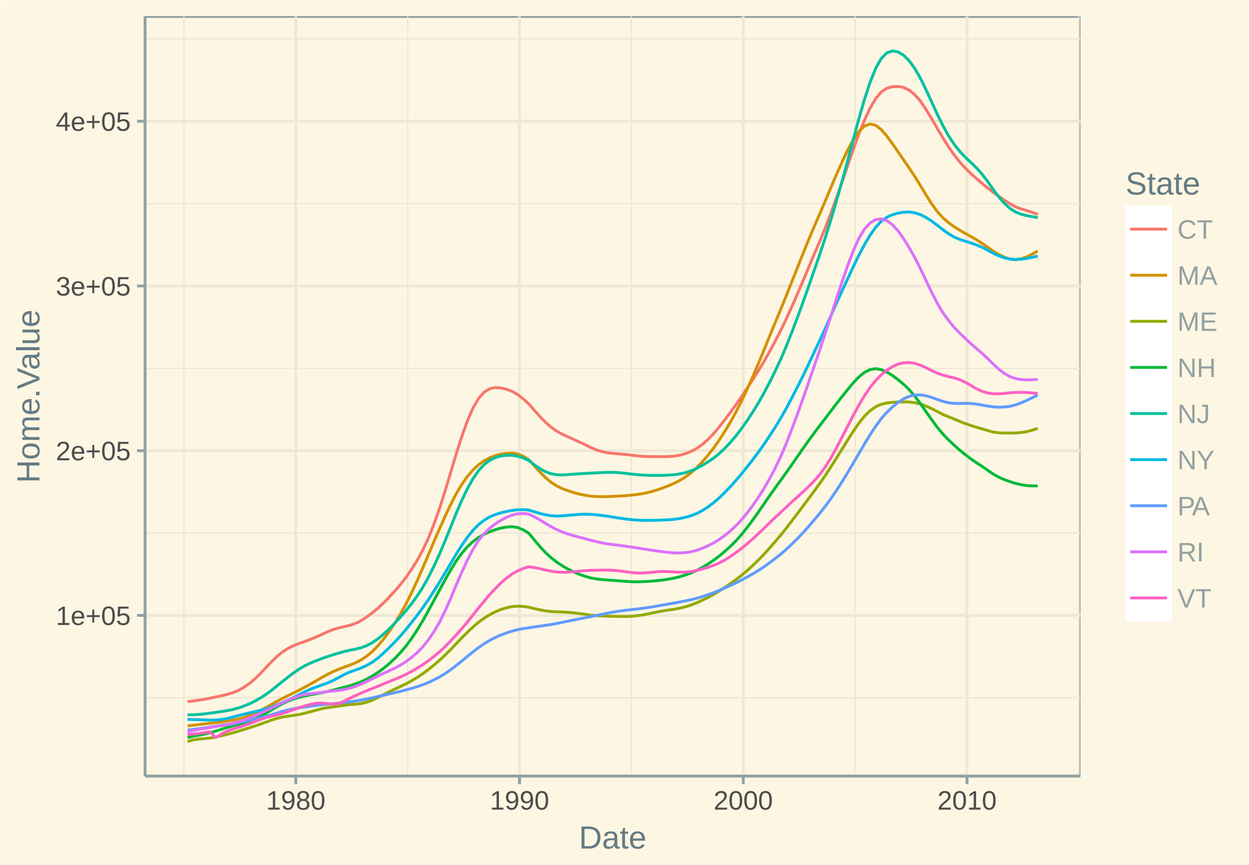
```r
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_economist()
```



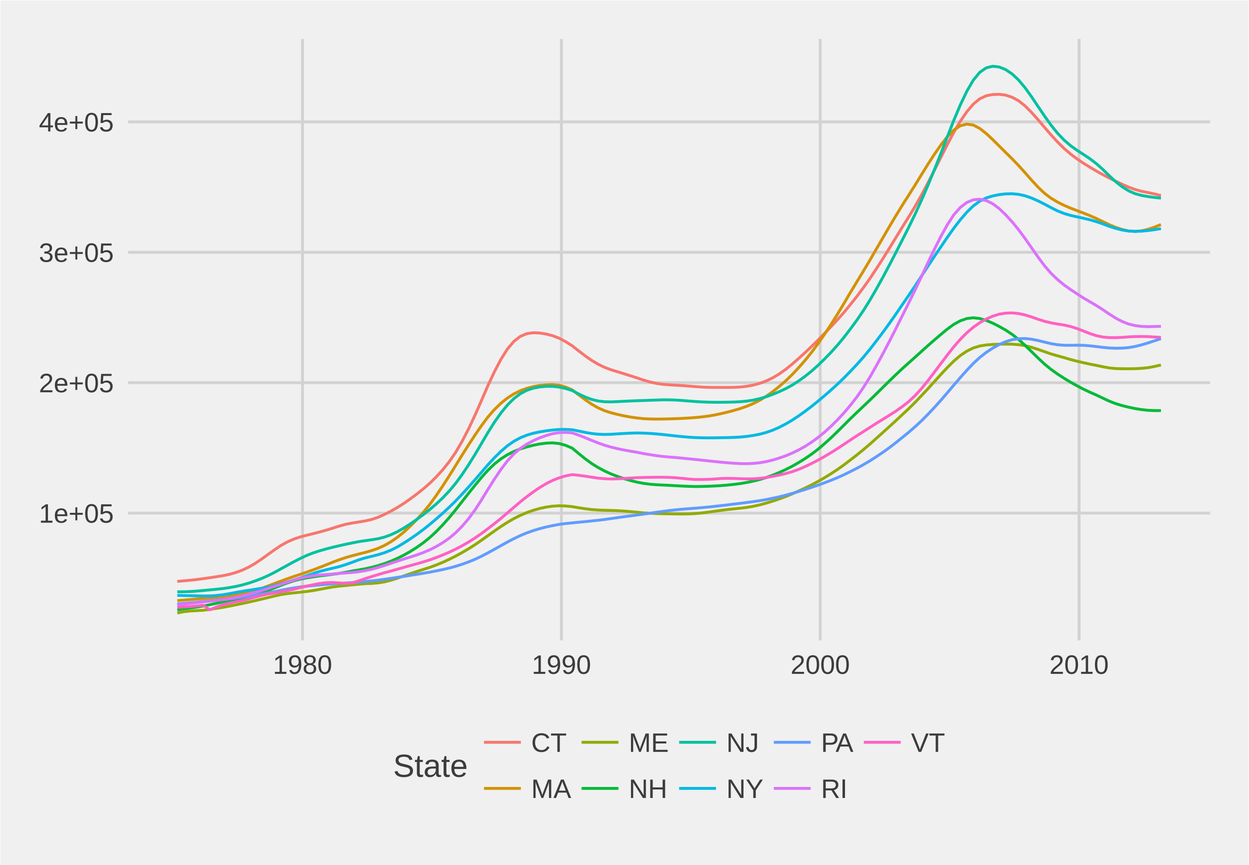
```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_wsj()
```



```r
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_solarized()
```

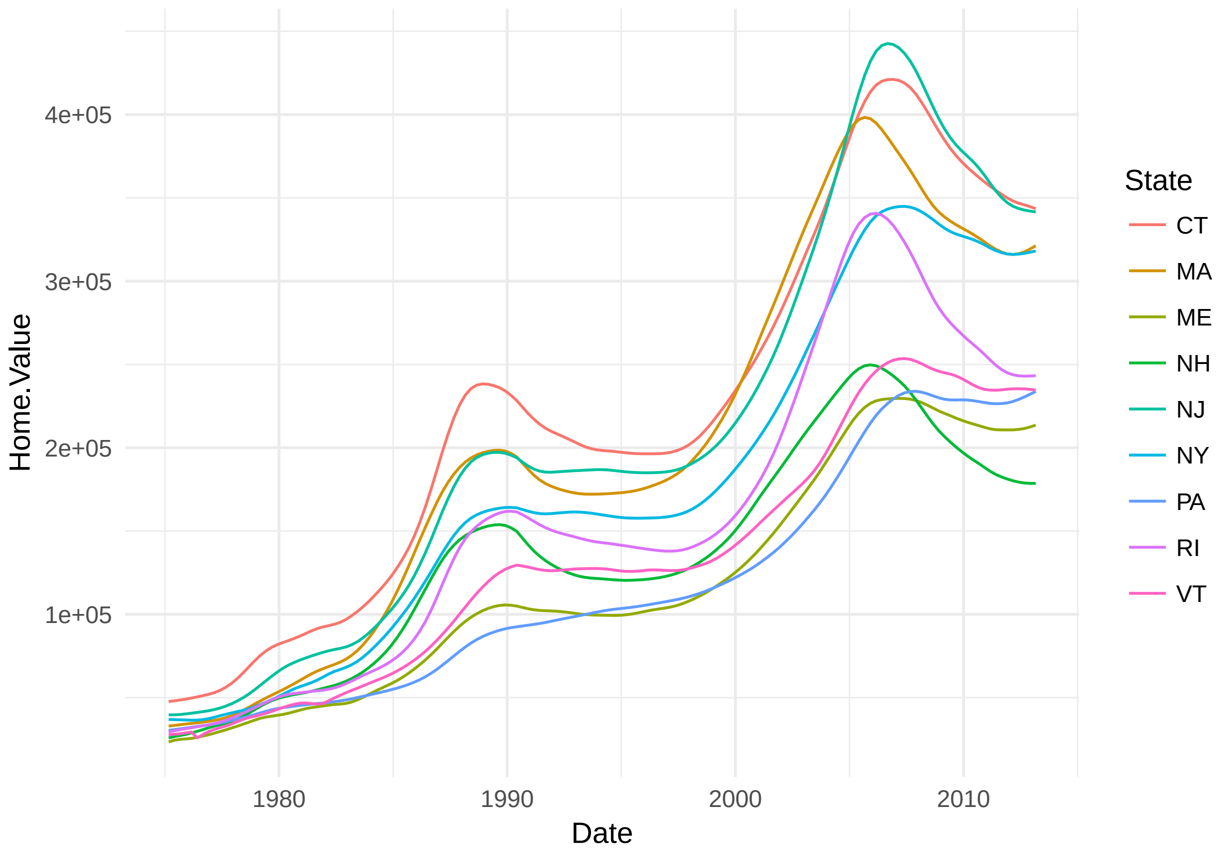


```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_fivethirtyeight()
```



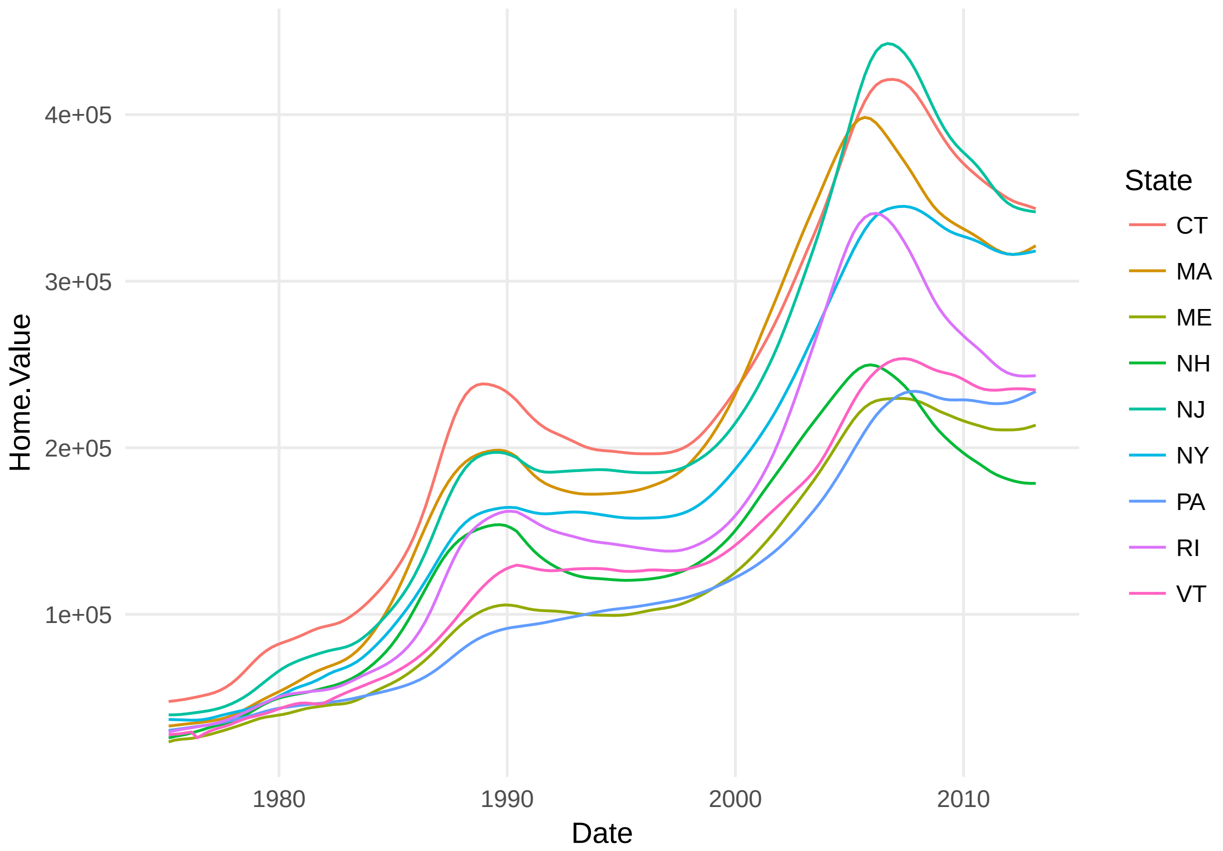
We can also have complete control over the theme by customizing each element ourselves. Let's start with `theme_minimal()`

```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_minimal()
```



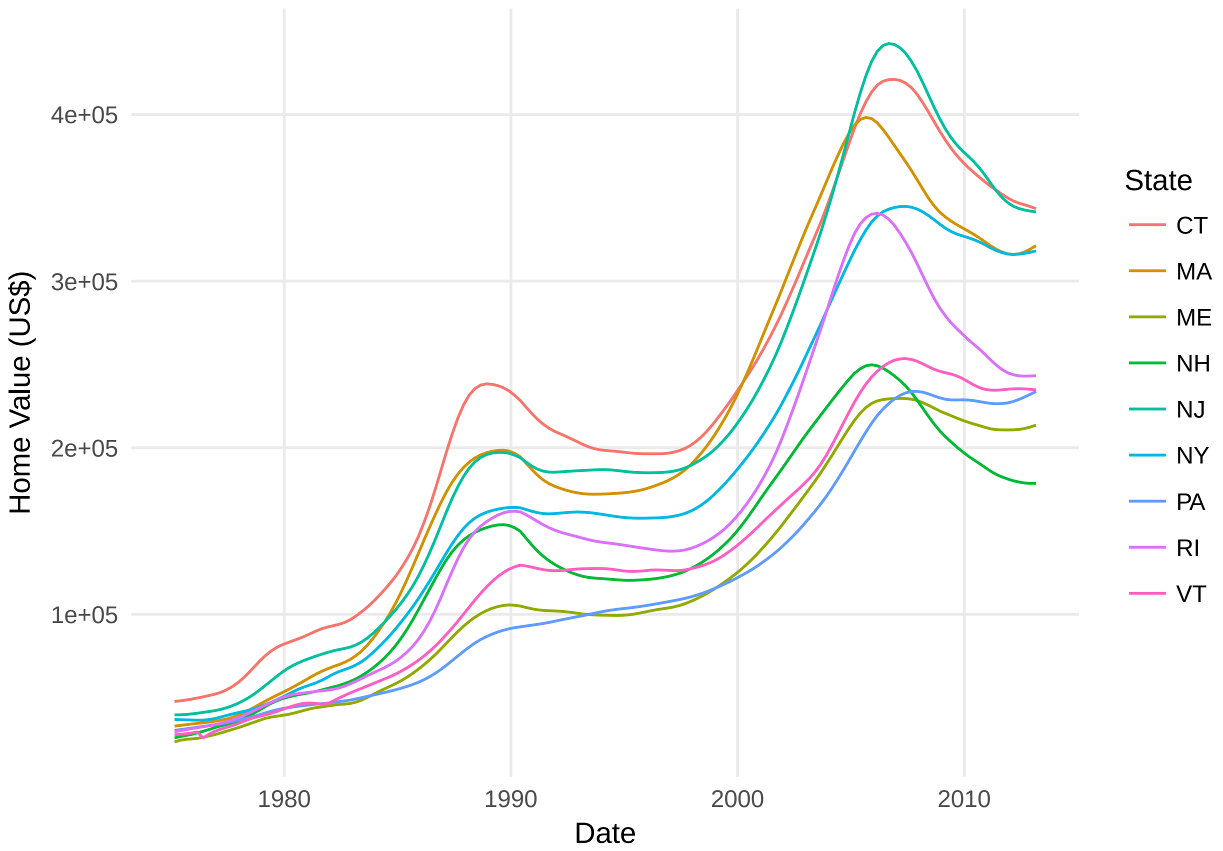
Now remove the minor grid lines

```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_minimal() +
  theme(
    panel.grid.minor = element_blank()
  )
```



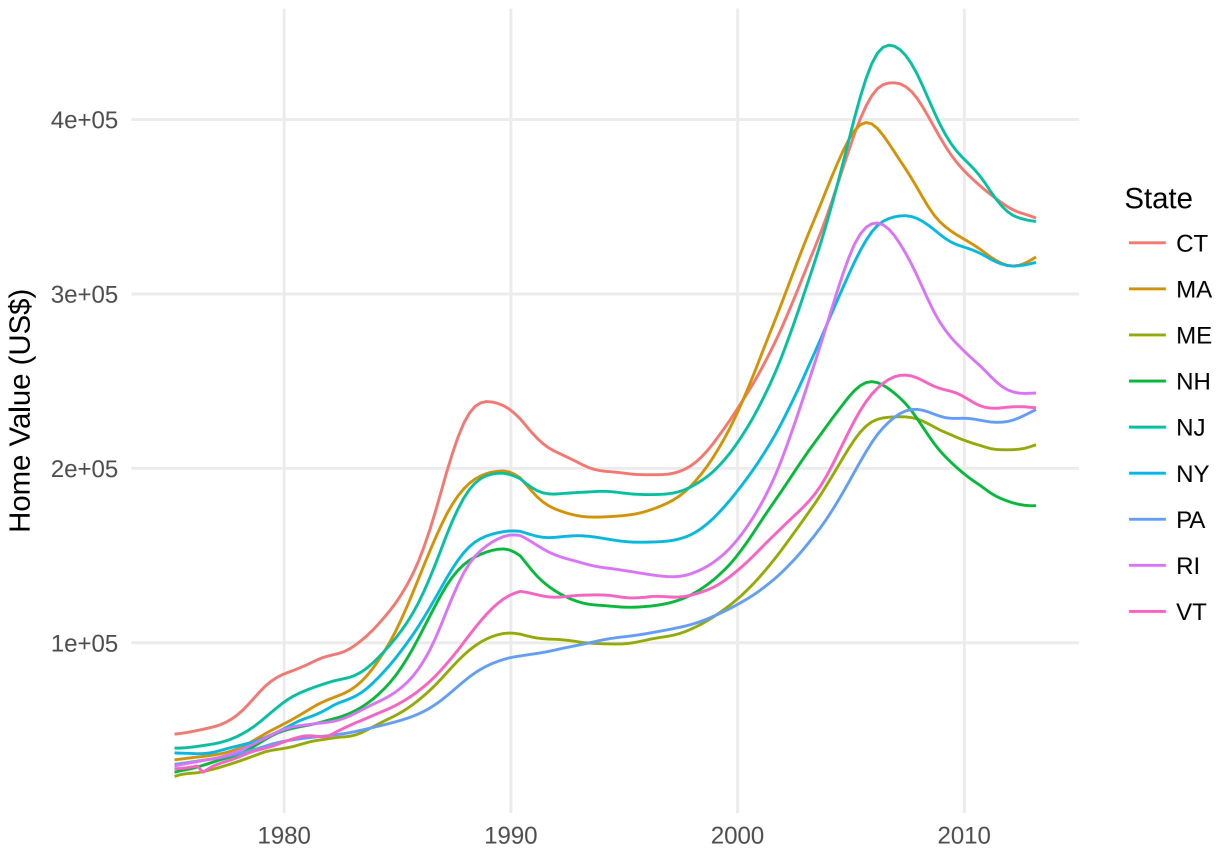
Next, we change the y-axis label

```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_minimal() +
  theme(
    panel.grid.minor = element_blank()
  ) +
  ylab("Home Value (US$)")
```



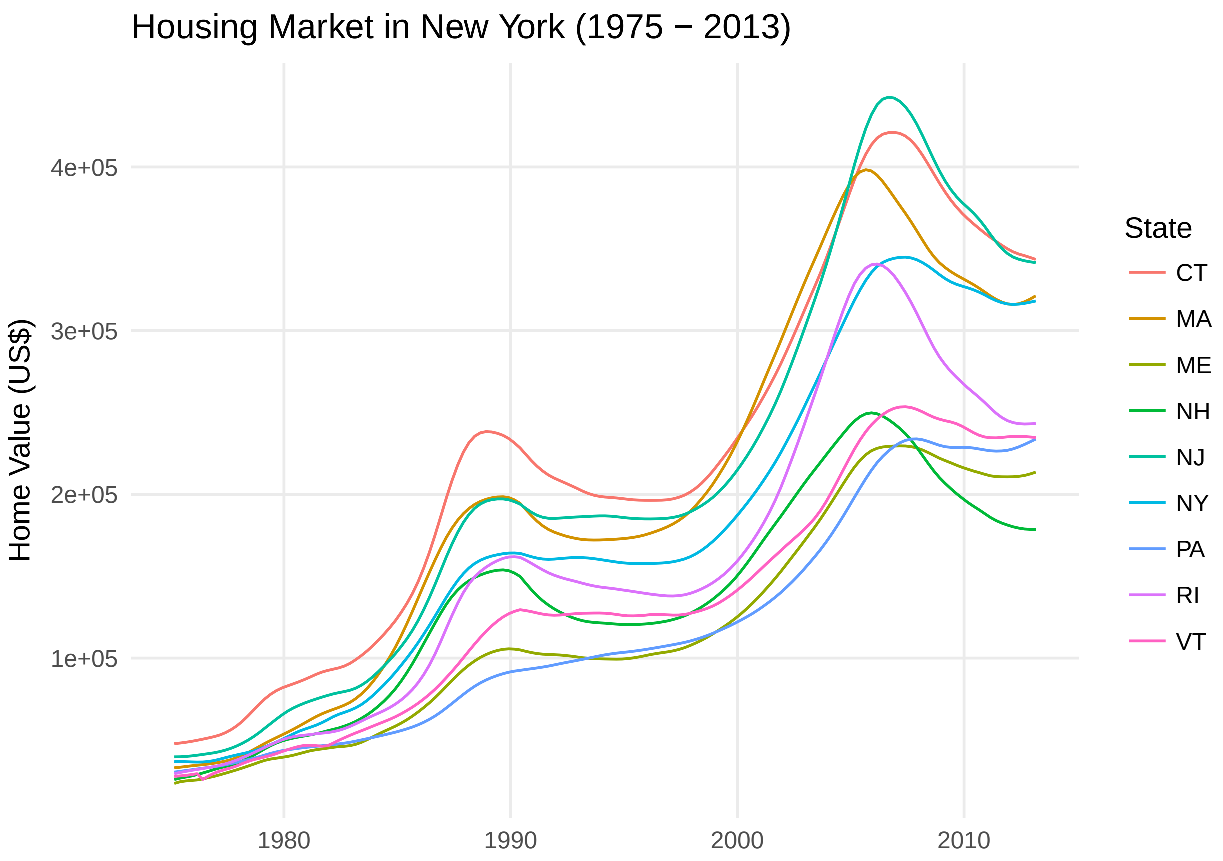
Then remove the x-axis title since the year is self explanatory

```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  ylab("Home Value (US$)")
```



Finally, we can add a title to our plot

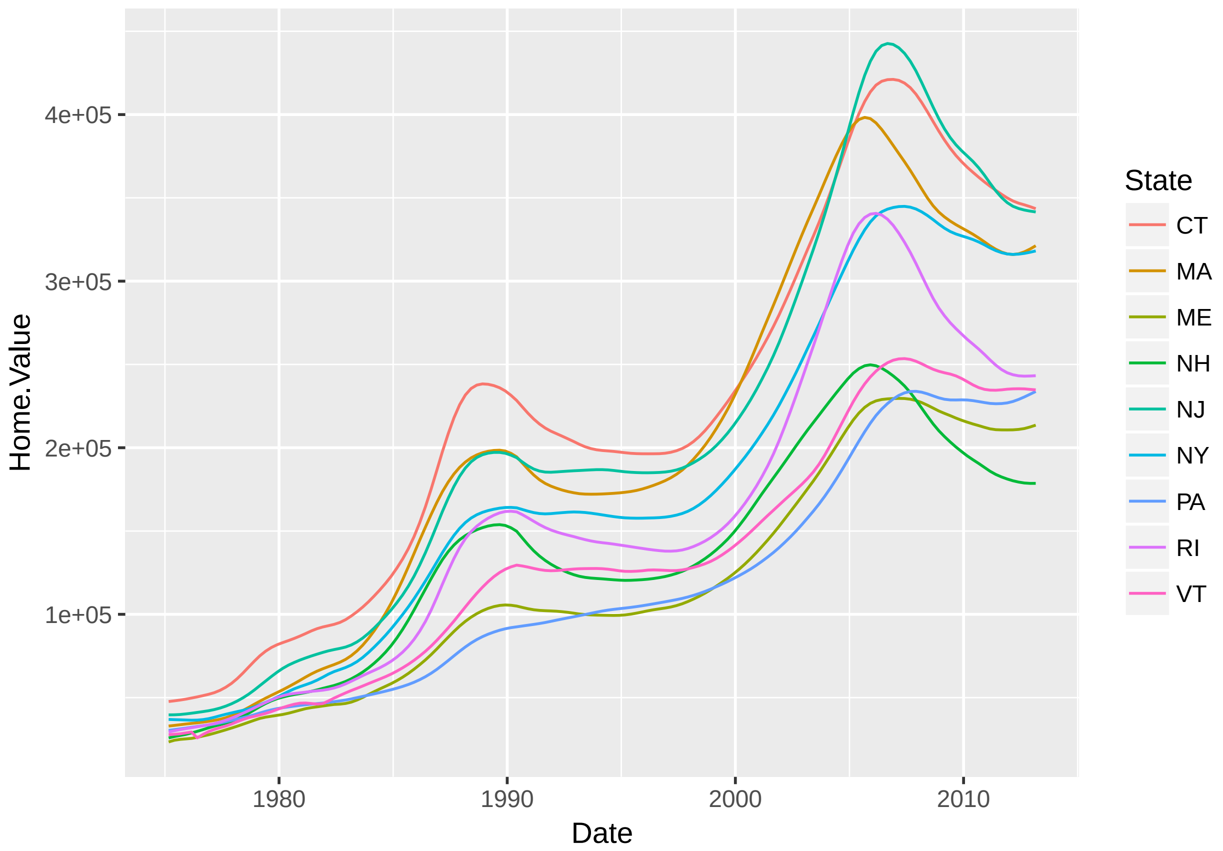
```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line() +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  ylab("Home Value (US$)") +
  ggtitle("Housing Market in New York (1975 - 2013)")
```



12 Facets

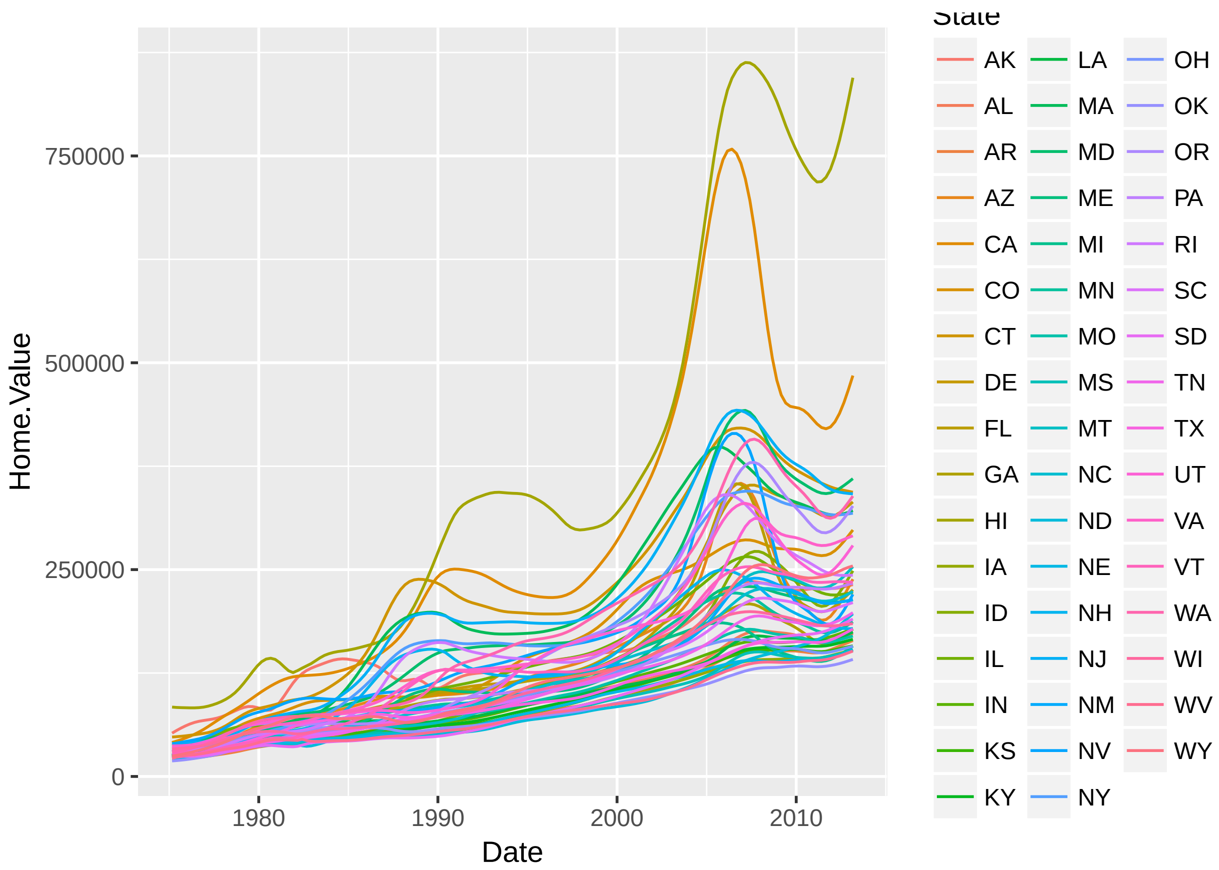
Let's plot the northeast data again

```
ggplot(northeast, aes(x = Date, y = Home.Value, color = State)) +
  geom_line()
```



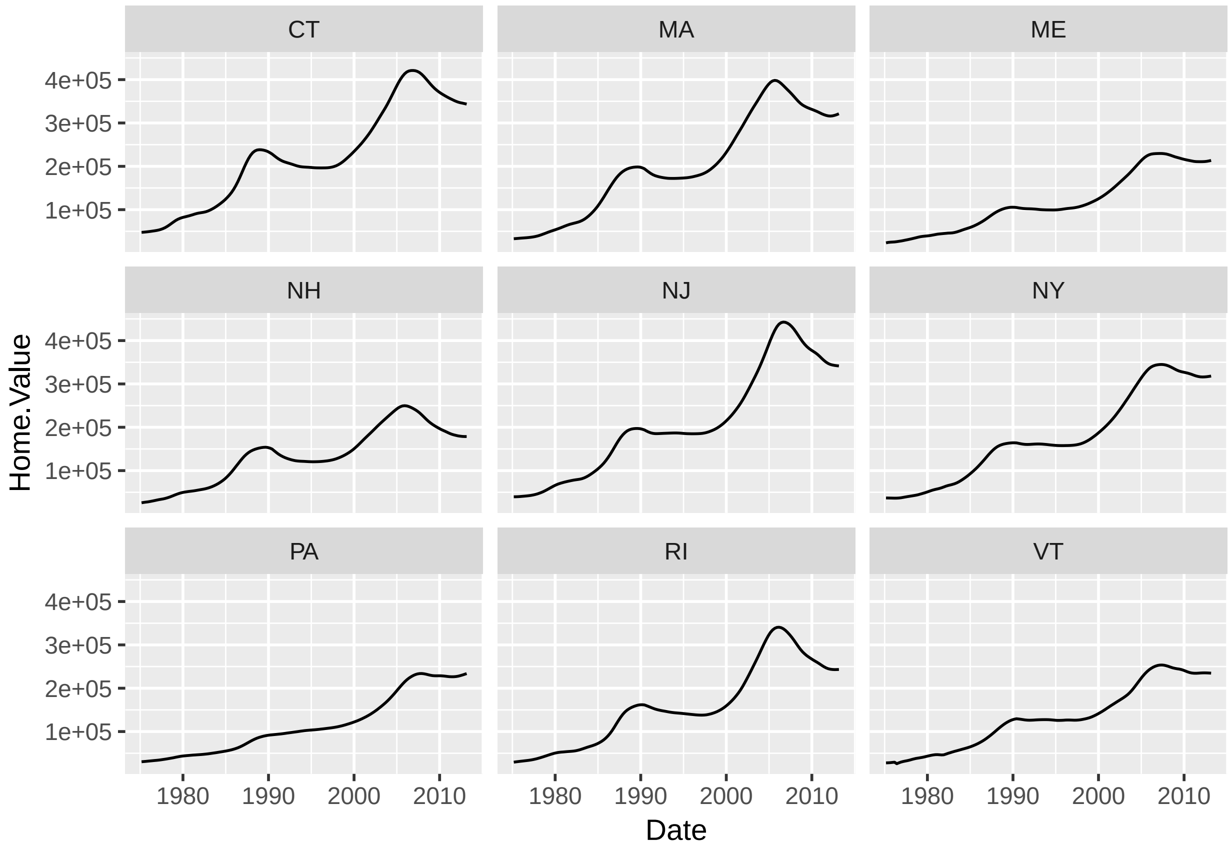
But what if we were to plot the entire dataset?

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



The plot is not very informative anymore. We can use facets to split the plot based on the `State`

```
ggplot(northeast, aes(x = Date, y = Home.Value)) +
  geom_line() +
  facet_wrap(~State, ncol = 3)
```



13 Challenge

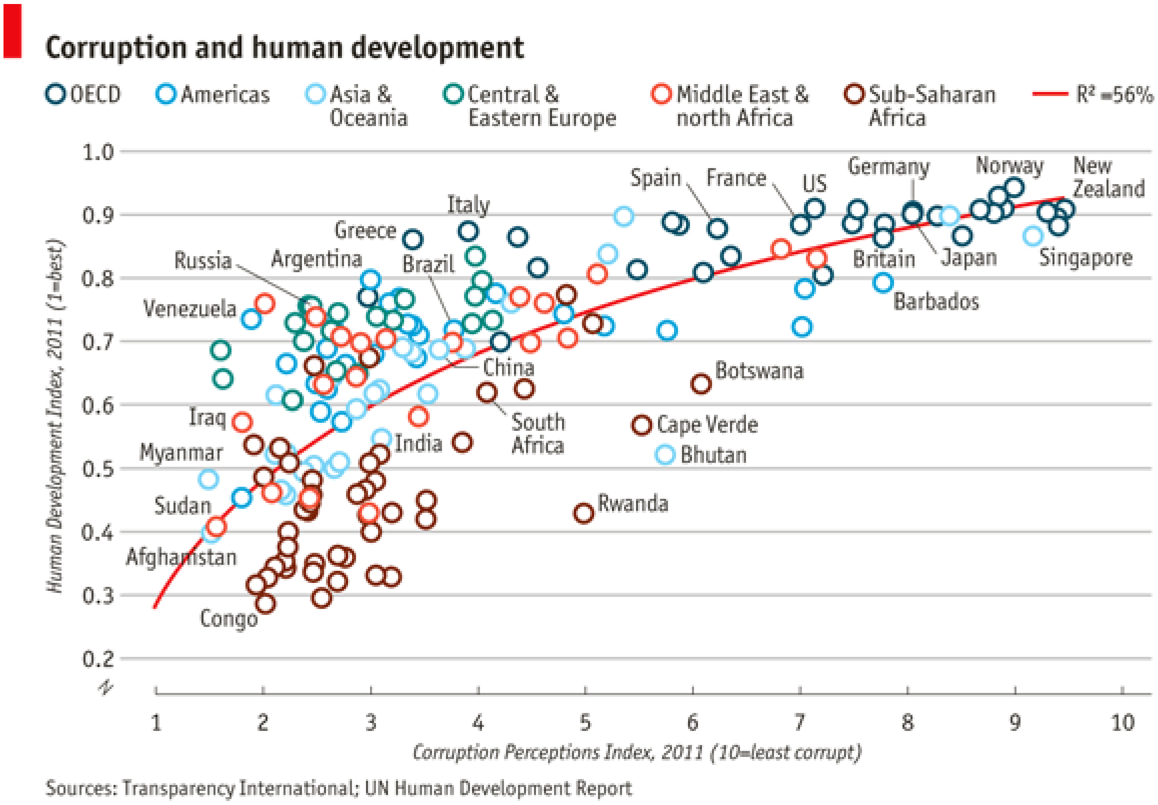
13.1 Recreating the Economist Graph

```
econ <- read.csv("https://raw.githubusercontent.com/altaf-ali/ggplot_tutorial/master/data/economist.csv

head(econ)
```

```
##   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
```

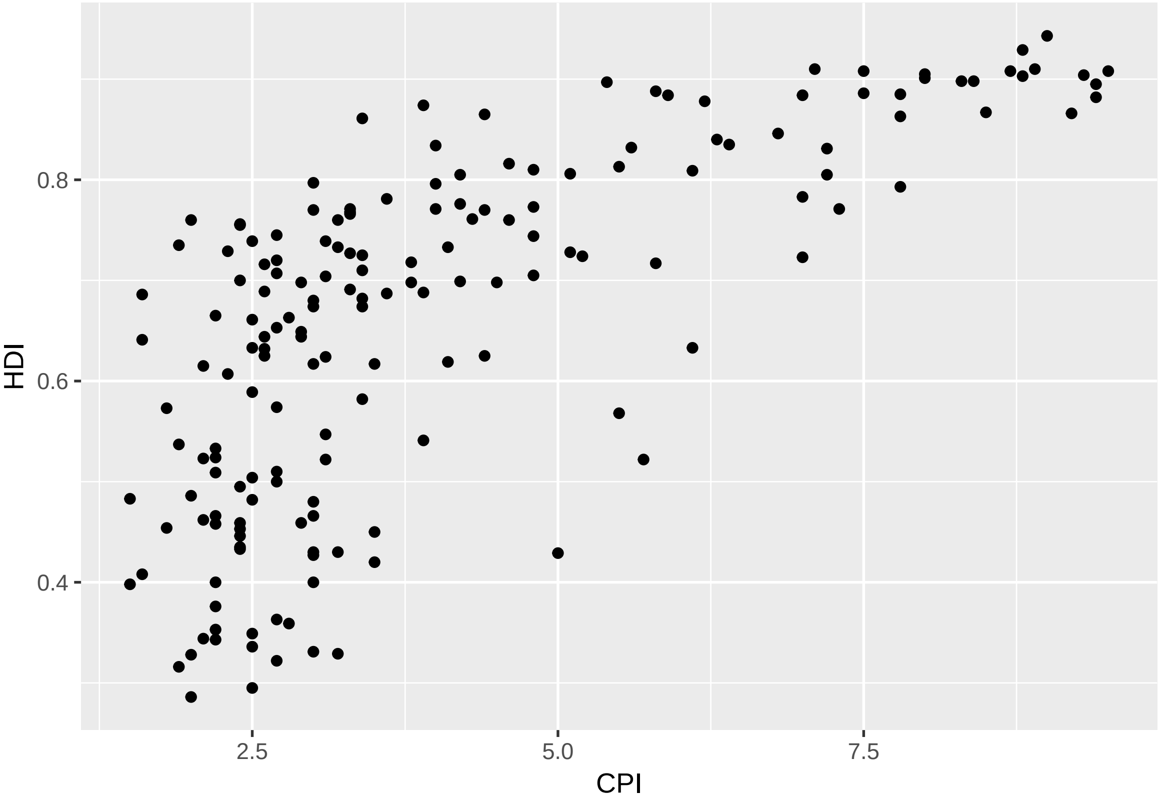
1. Create a scatter plot of the economist data with `CPI` on the x-axis and `HDI` on the y-axis



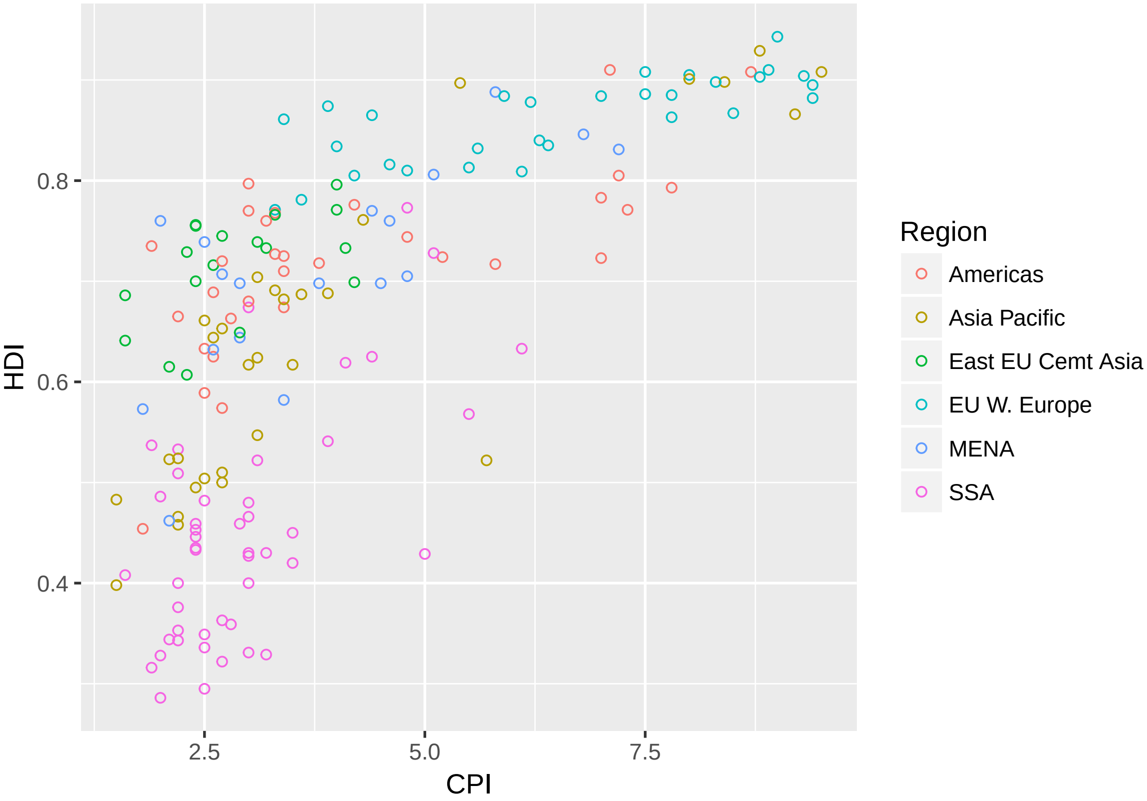
Corruption and human development

Sources: Transparency International; UN Human Development Report

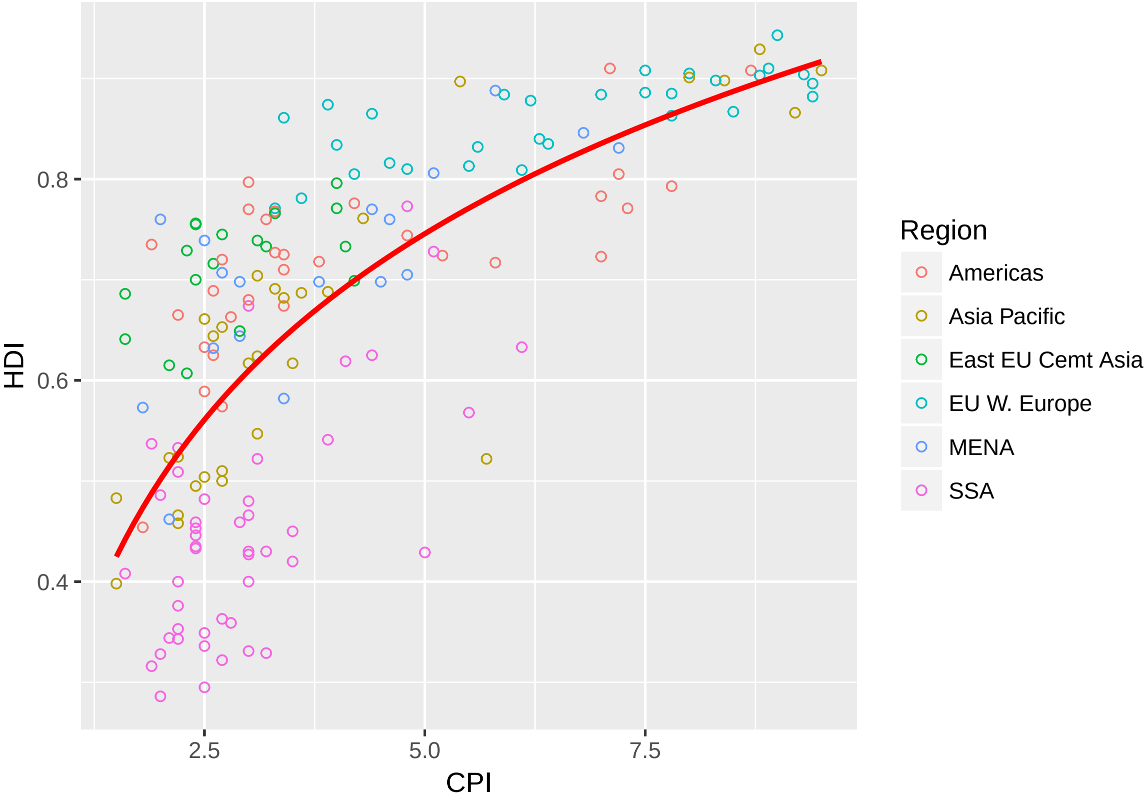
Figure 3:



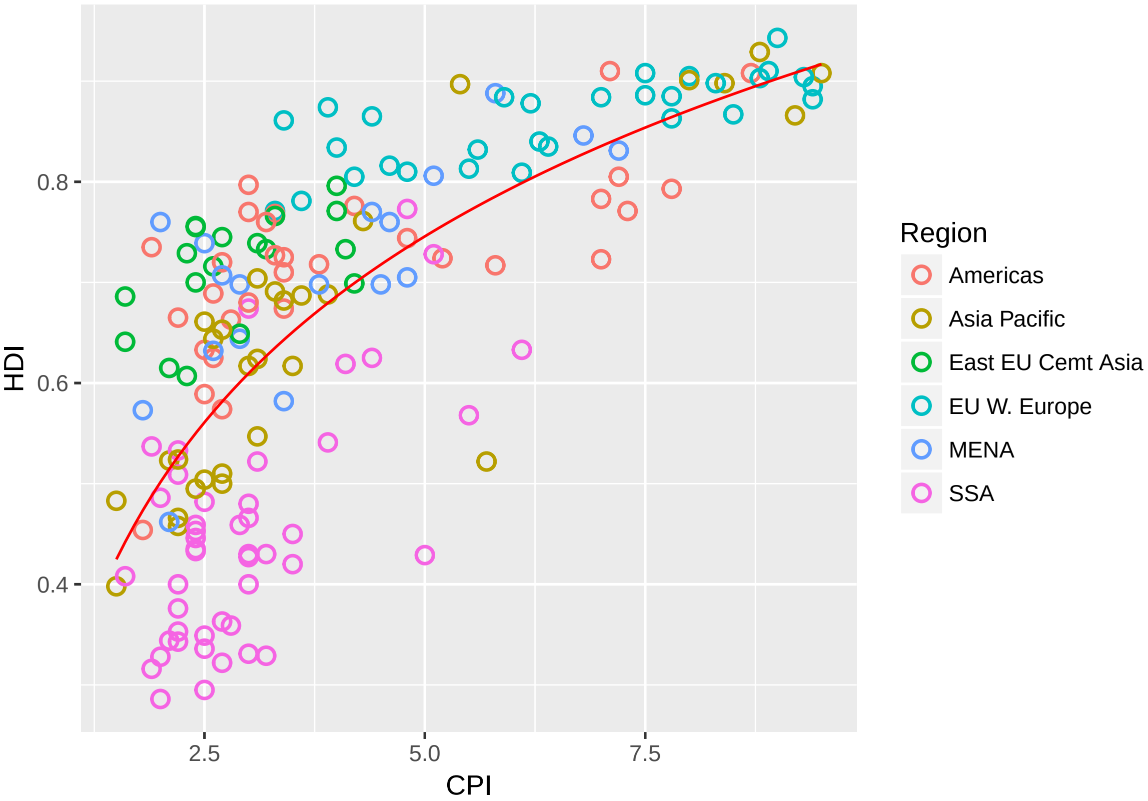
2. Color the points based on `Region` using hollow points



3. Add a trend line



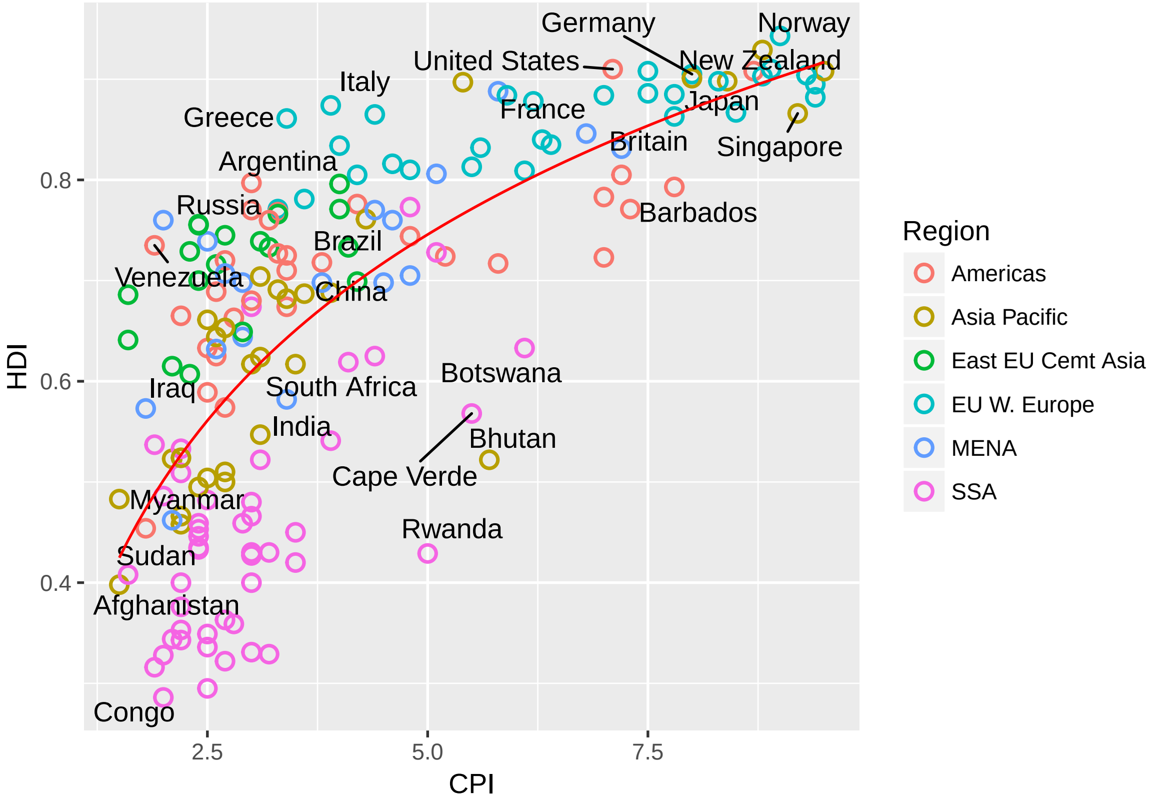
4. The trend line is too thick compared to the circles so we need to adjust it appropriately



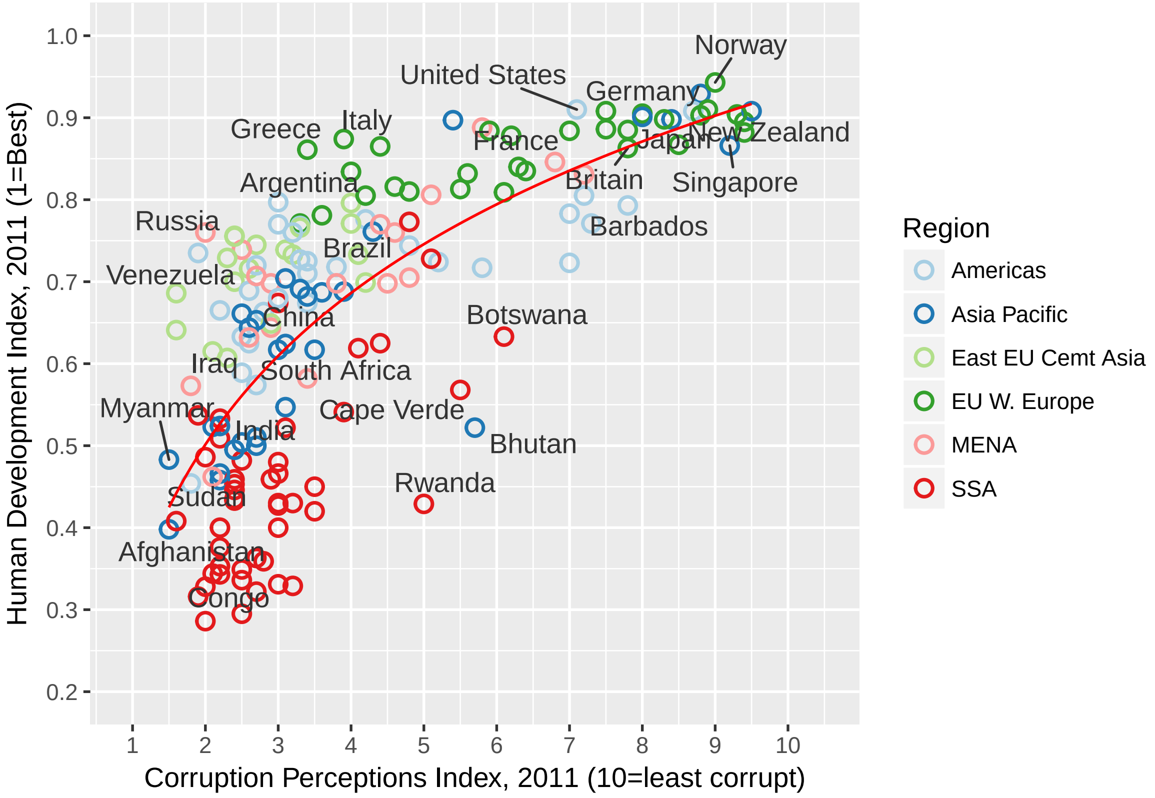
5. Add text labels to the points

   HINT: Create a subset of countries to label since we don't want to label every point

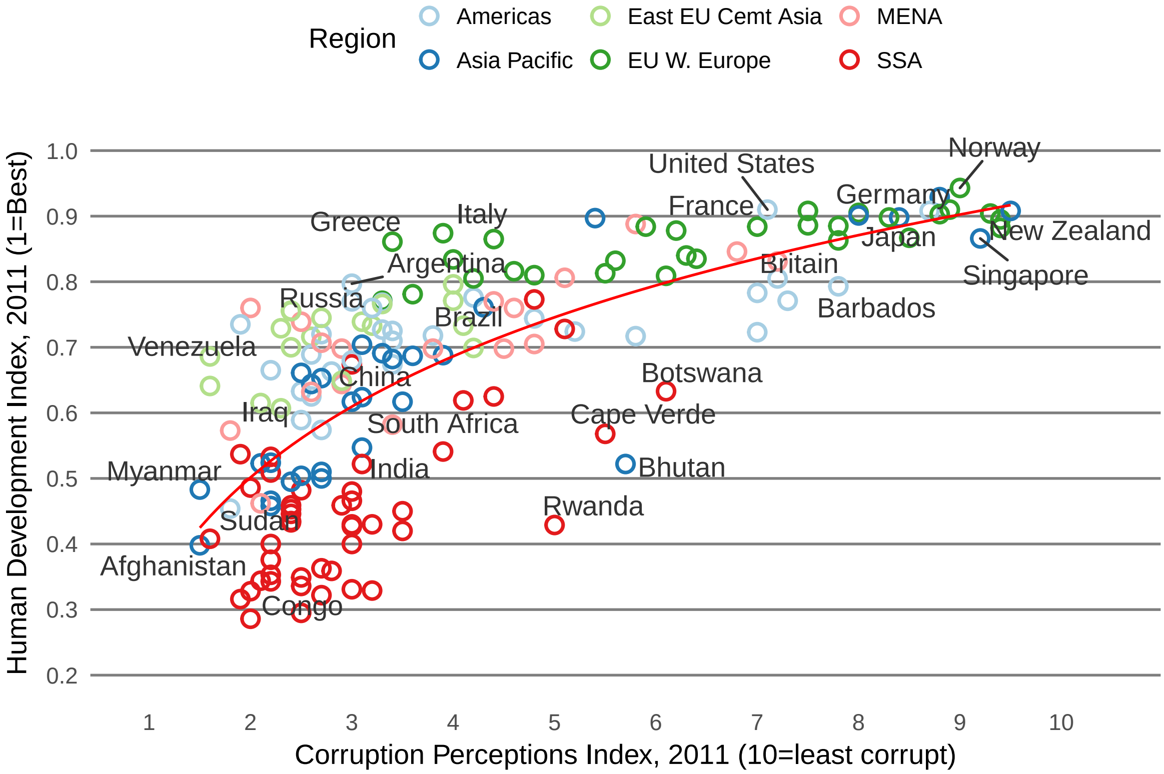
```
target_countries <- c(
  "Russia", "Venezuela", "Iraq", "Myanmar", "Sudan",
  "Afghanistan", "Congo", "Greece", "Argentina", "Brazil",
  "India", "Italy", "China", "South Africa", "Spane",
  "Botswana", "Cape Verde", "Bhutan", "Rwanda", "France",
  "United States", "Germany", "Britain", "Barbados", "Norway", "Japan",
  "New Zealand", "Singapore"
)
labeled_countries <- subset(econ, Country %in% target_countries)
```



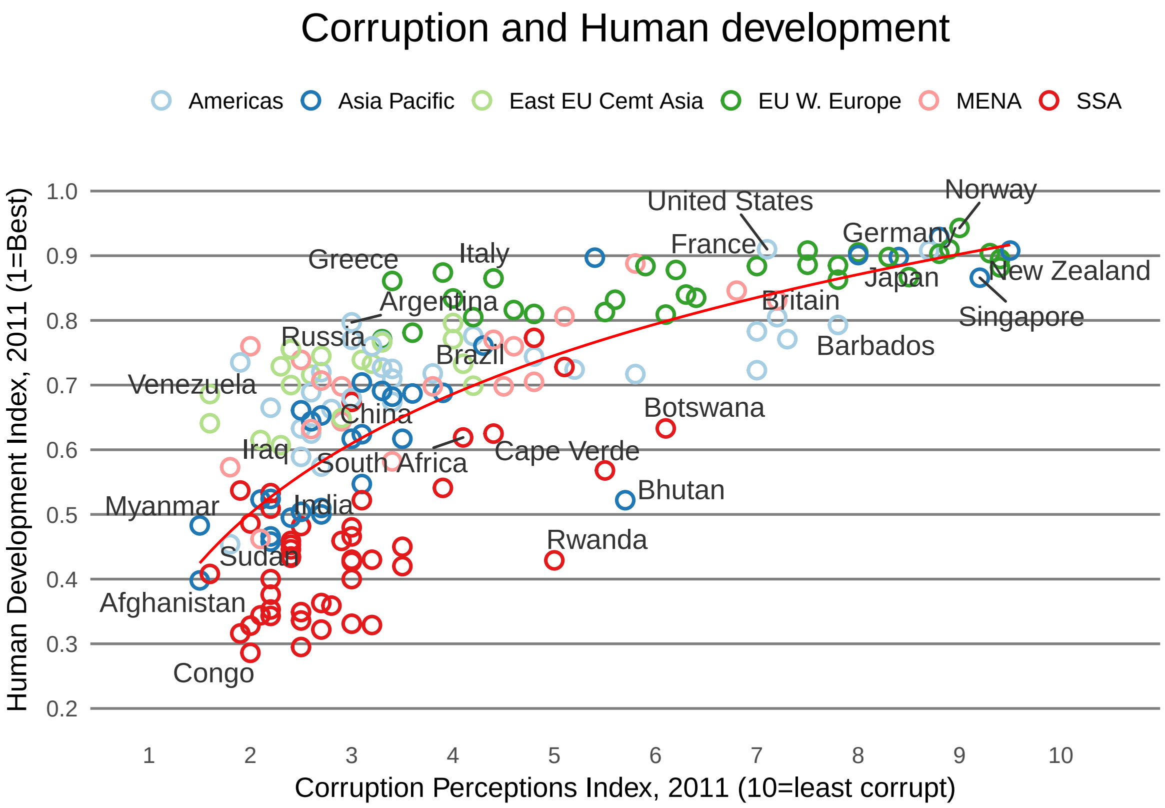
6. Adjust the x and y scales and use Color Brewer pallete `Paired`.



7. Remove vertical grid lines and move the legend



8. Add title “Corruption and Human development”


Corruption and Human development
○ Americas
○ Asia Pacific
○ East EU Cemt Asia
○ EU W. Europe
○ MENA
○ SSA
Human Development Index, 2011 (1=Best)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Corruption Perceptions Index, 2011 (10=least corrupt)
1
2
3
4
5
6
7
8
9
10
Norway
United States
Germany
France
Greece
Italy
Japan
New Zealand
Argentina
Britain
Singapore
Russia
Brazil
Barbados
Venezuela
Botswana
China
Iraq
Cape Verde
South Africa
Bhutan
Myanmar
India
Rwanda
Sudan
Afghanistan
Congo