Introduction to R Graphics with ggplot2 (IQSS)

Code

Samuel Chan

This is a weekend exercise, using materials from a workshop offered by Harvard University's Institute for Quantitative Social Science (IQSS) program. The original workshop notes can be found on the IQSS website (http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html).

Some default set-up

Hide

```
options(width=70)
theme_set(theme_bw())
```

Grammar of Graphics

The basic idea: independently specify plot building blocks and combine them to create just about any kind of graphical display you want. Building blocks of a graph include:

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

Take a look at the housing prices dataset.

Hide

```
housing <- read.csv("datasets/landdata-states.csv")
head(housing[,1:5])
```

	State <fctr>	region <fctr>	Date <dbl>	Home.Value <int>	Structure.Cost <int>
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

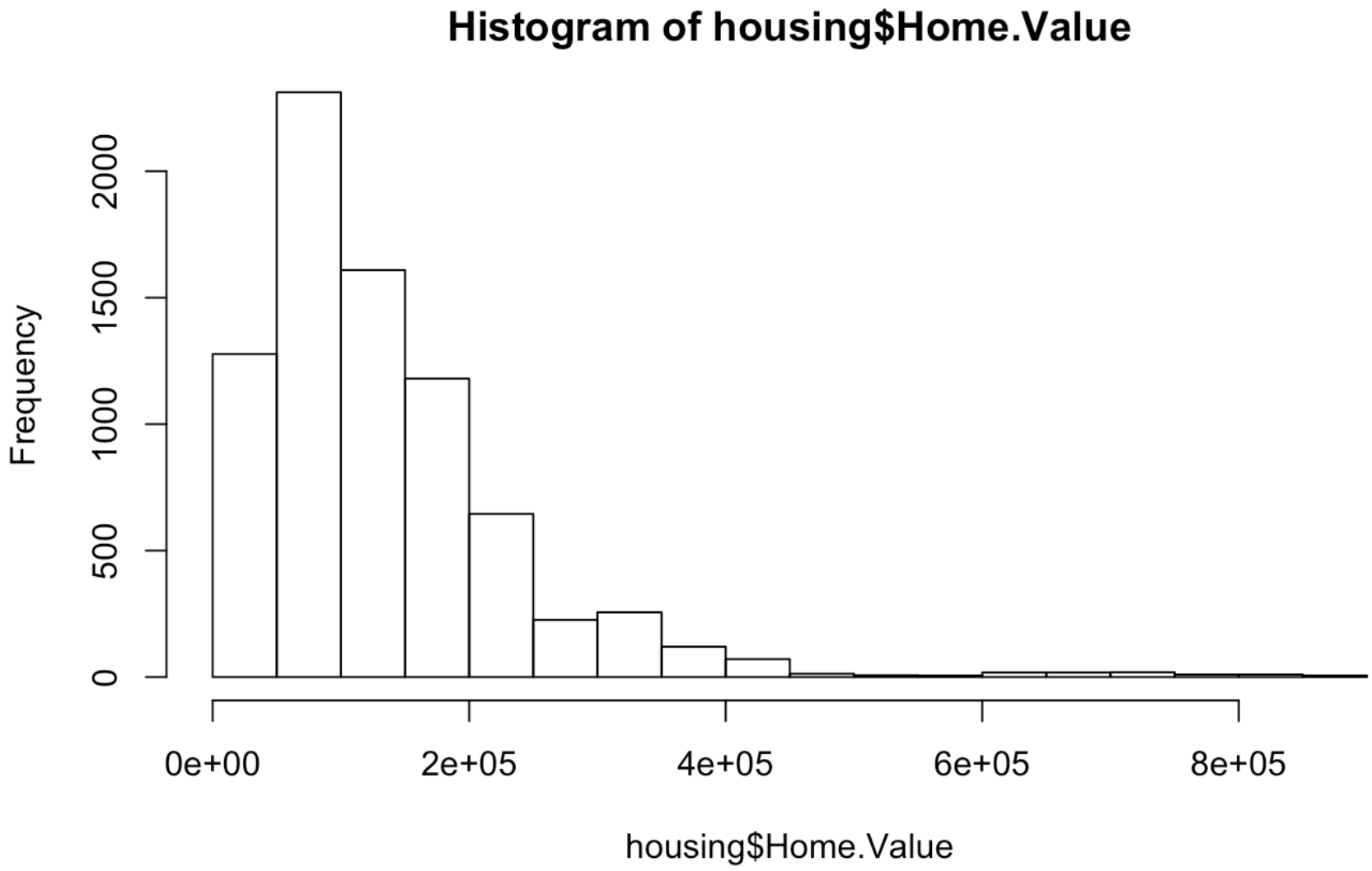
6 rows

`Base` graphics VS `ggplot` for histograms

Base graphics histogram example:

Hide

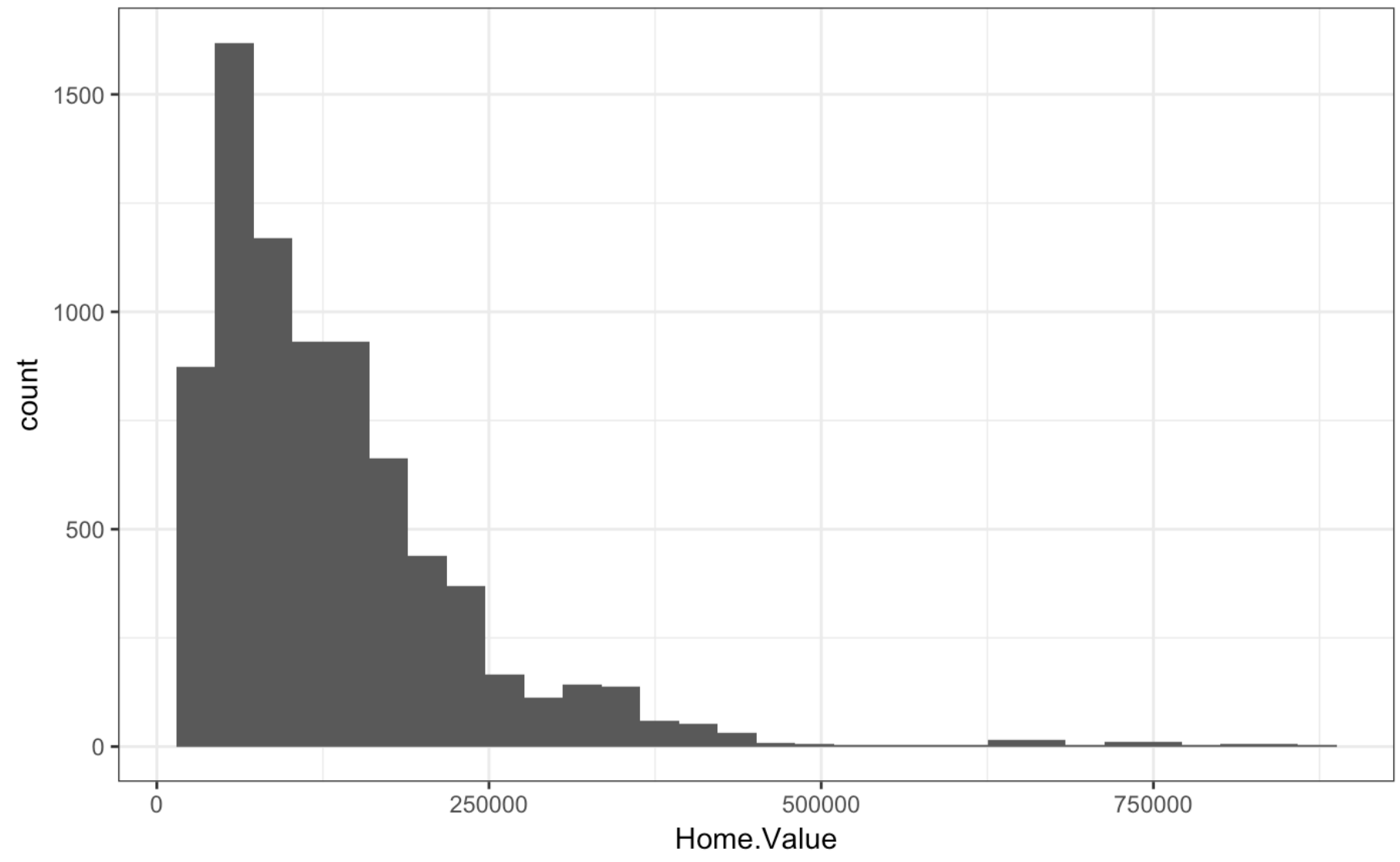
```
hist(housing$Home.Value)
```



`ggplot2` histogram example:

Hide

```
library(ggplot2)
ggplot(housing, aes(x=Home.Value)) + geom_histogram()
```



`Base` graphics VS `ggplot` for scatterplots

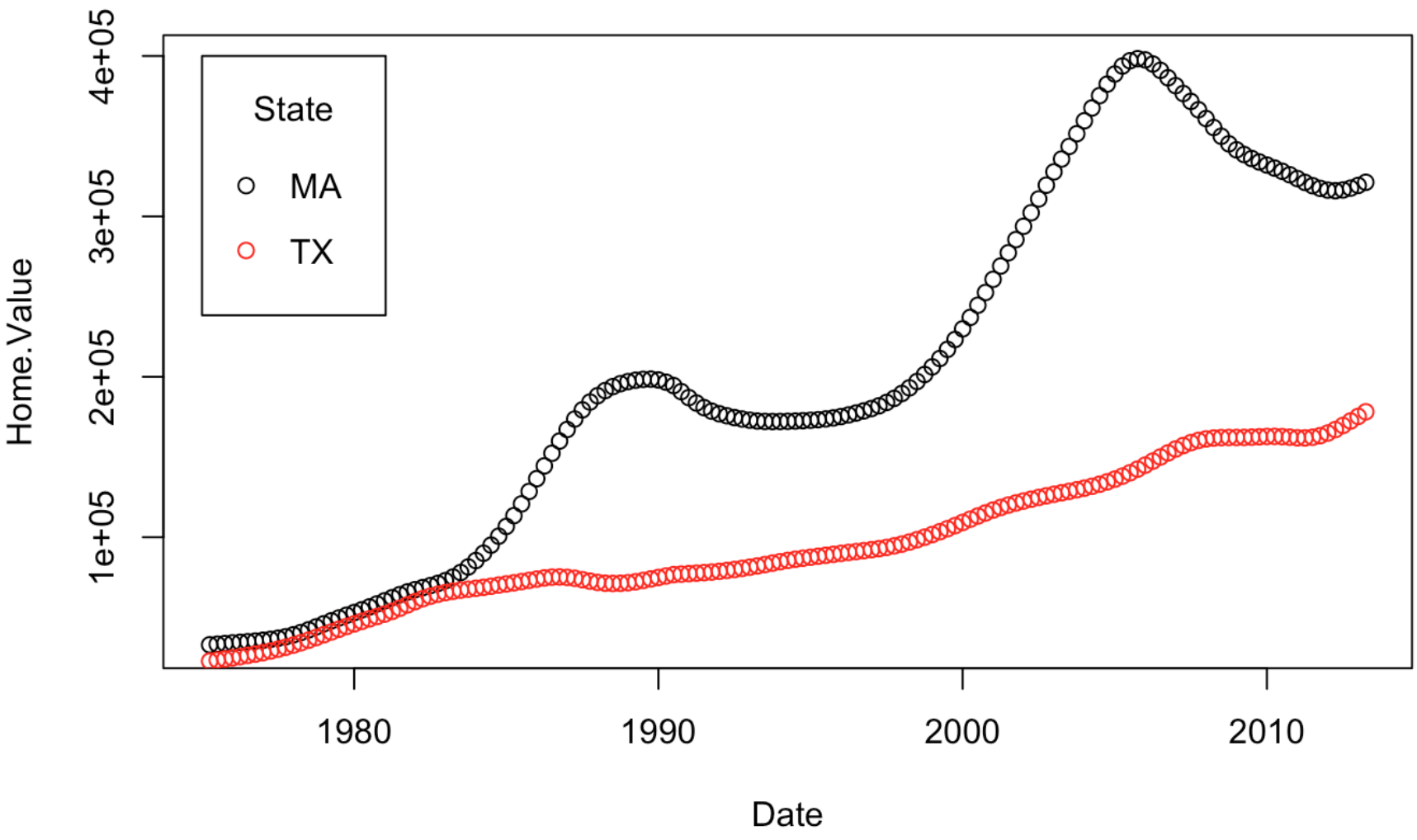
Base colored scatterplot example:

Hide

```
plot(Home.Value ~ Date, data=subset(housing, State == "MA"))
points(Home.Value ~ Date, col="red", data=subset(housing, State == "TX"))
```

Hide

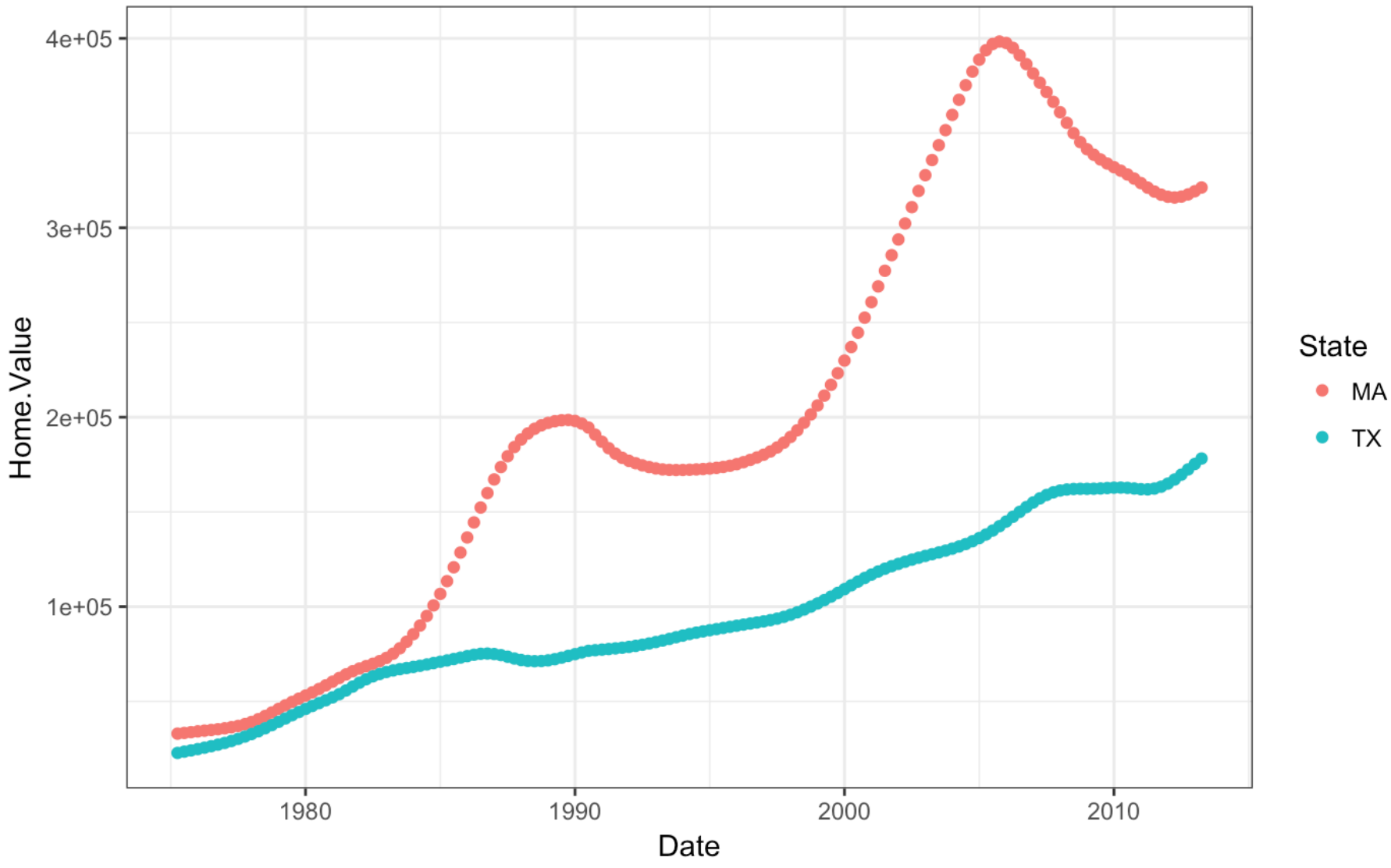
```
# legend(x, y, legend) where x and y are coordinates to position the legend
legend(1975, 400000, c("MA","TX"), title="State", col=c("black", "red"), pch=c(1,1))
```



`ggplot` colored scatterplot example:

Hide

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



Geometric Objects and Aesthetics

Aesthetic Mapping

In ggplot, aesthetic means something we can see. Examples are:

- Position (i.e. on the x and y axis)
- Color (“outside” color)
- Fill (“inside” color)

- Shape (of points)
- Linetype
- Size

Geometric objects are the actual marks we put on a plot. Examples are:

- Points (geom_point, for scatter plots, dot plots, etc)
- Lines (geom_line, for time series, trend lines, etc)
- Boxplot (geom_boxplot, for, well, boxplots!)

A plot must have at least one geom; there is no upper limit. You can add a geom to a plot using the + operator. We can get a list of available geometric objects using the code below:

Hide

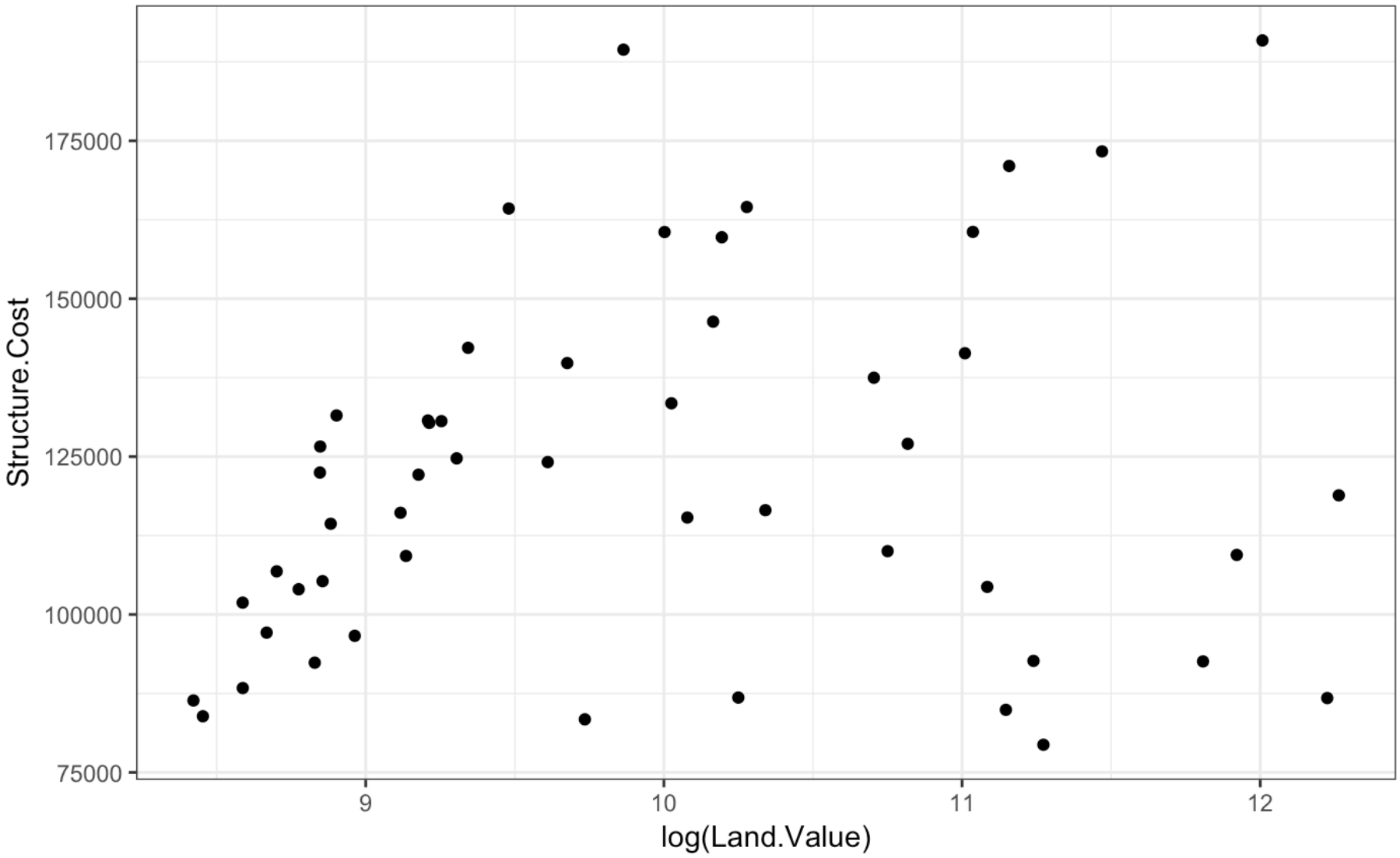
```
help.search("geom_", package = "ggplot2")
```

Points (scatterplot)

`geom_point` requires mappings for x and y, all others optional

Hide

```
hp.q12001 <- subset(housing, Date == 2001.25)
ggplot(hp.q12001, aes(x = log(Land.Value), y = Structure.Cost)) + geom_point()
```

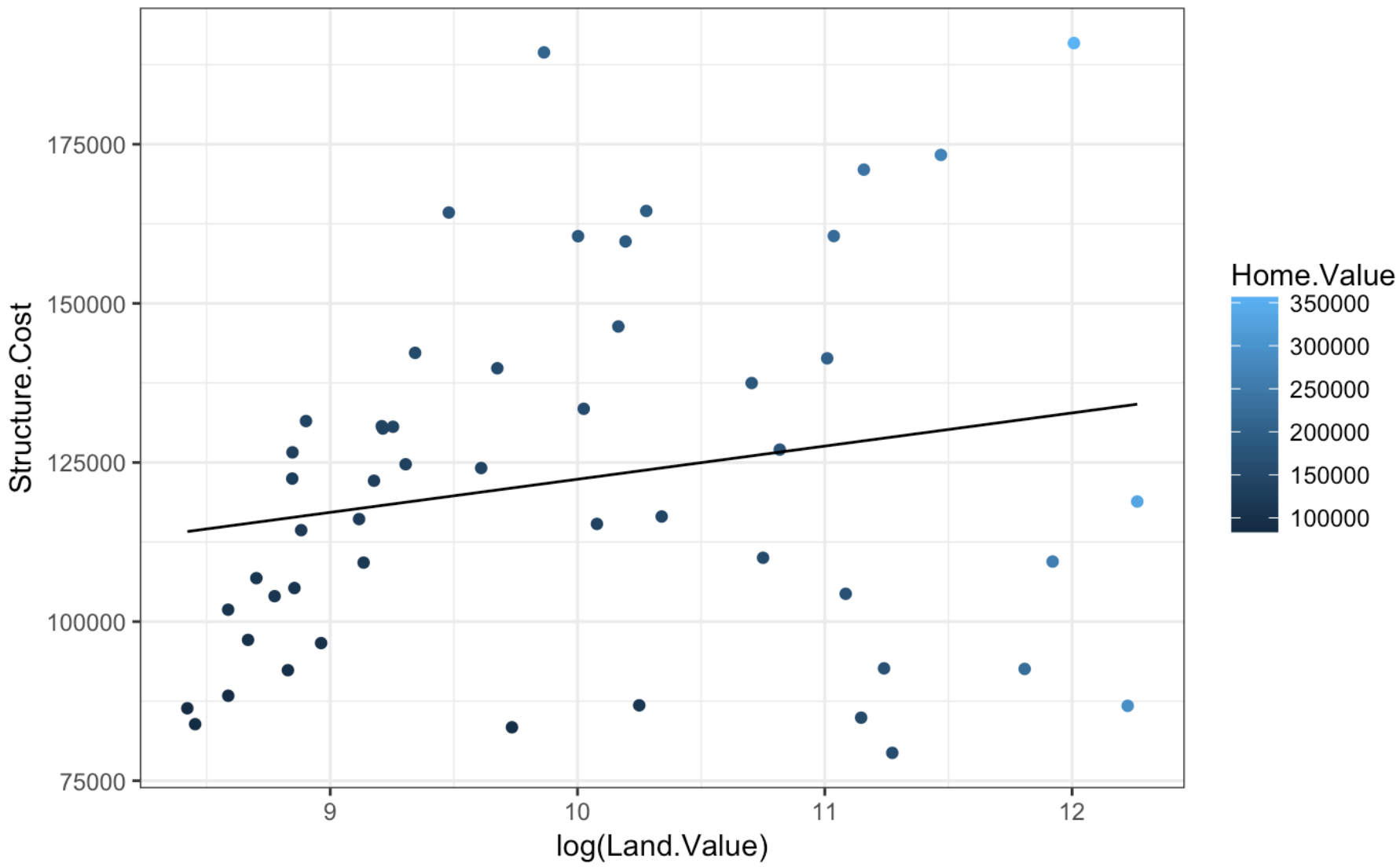


Lines (Prediction lines)

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

Hide

```
# Create a .SC variable (structure cost), that is just a value of the predict function() using lm
hp.q12001$pred.SC <- predict(lm(Structure.Cost ~ log(Land.Value), data = hp.q12001))
p1 <- ggplot(hp.q12001, aes(x = log(Land.Value), y = Structure.Cost)) + geom_point(aes(color = Home.Value)) + geo
m_line(aes(y = pred.SC))
p1
```

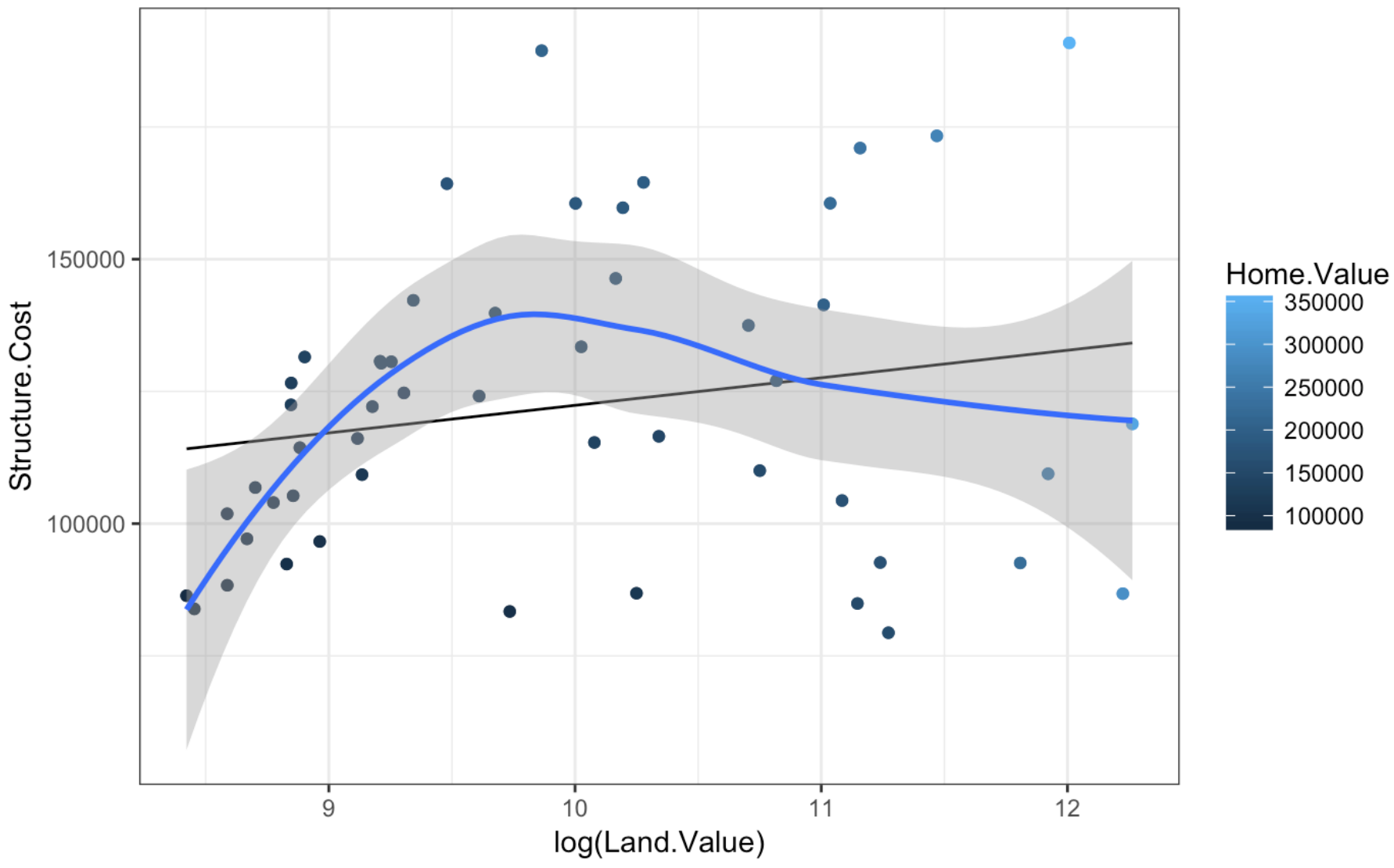


Smoothers

Not all geometric objects are simple shapes–the smooth geom includes a line and a ribbon.

Hide

```
p1 + geom_point(aes(color=Home.Value)) + geom_smooth()
```

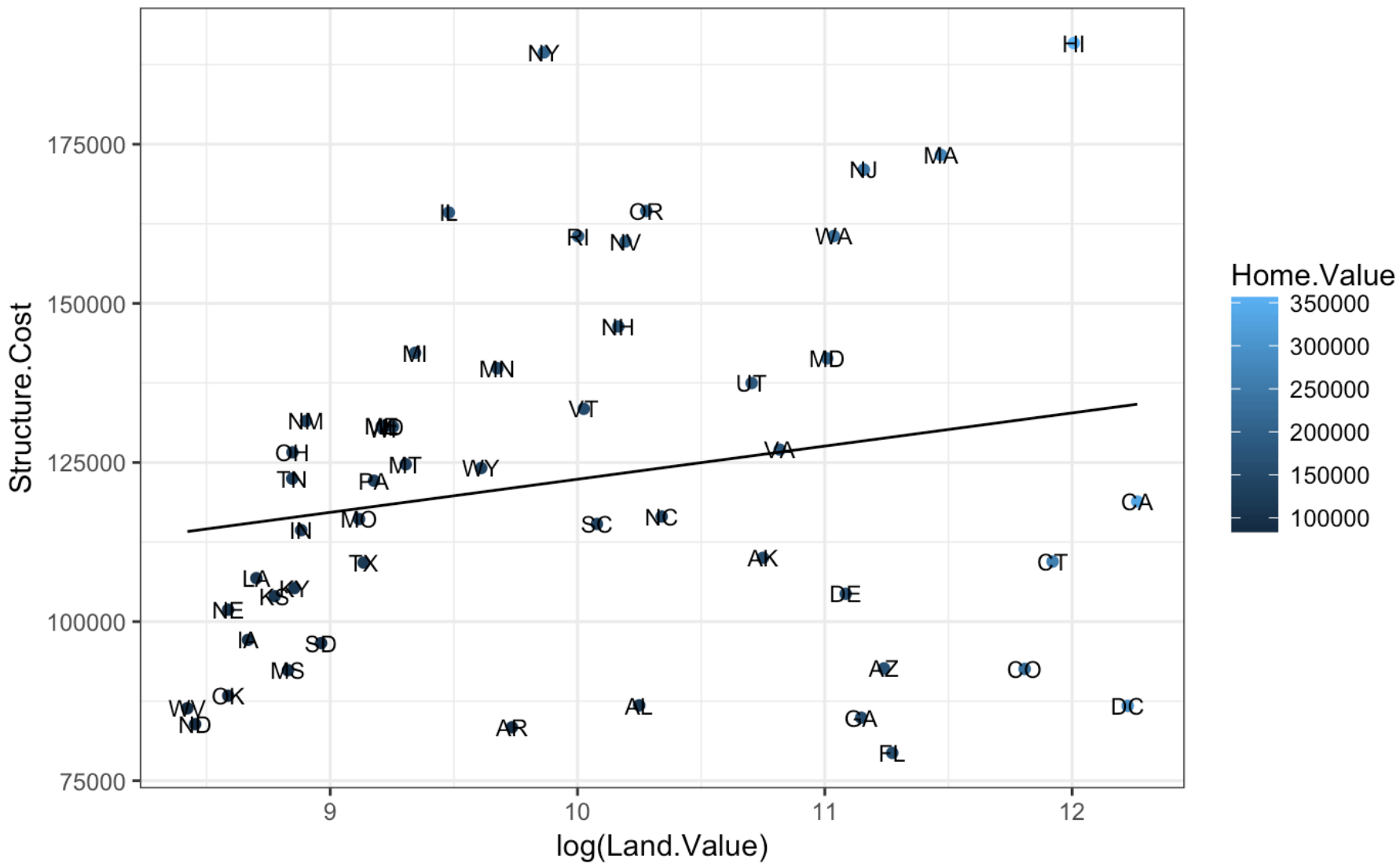


Text (label points)

Each `geom` accepts a particular set of mappings. For example, geom_text() accepts a labels mapping.

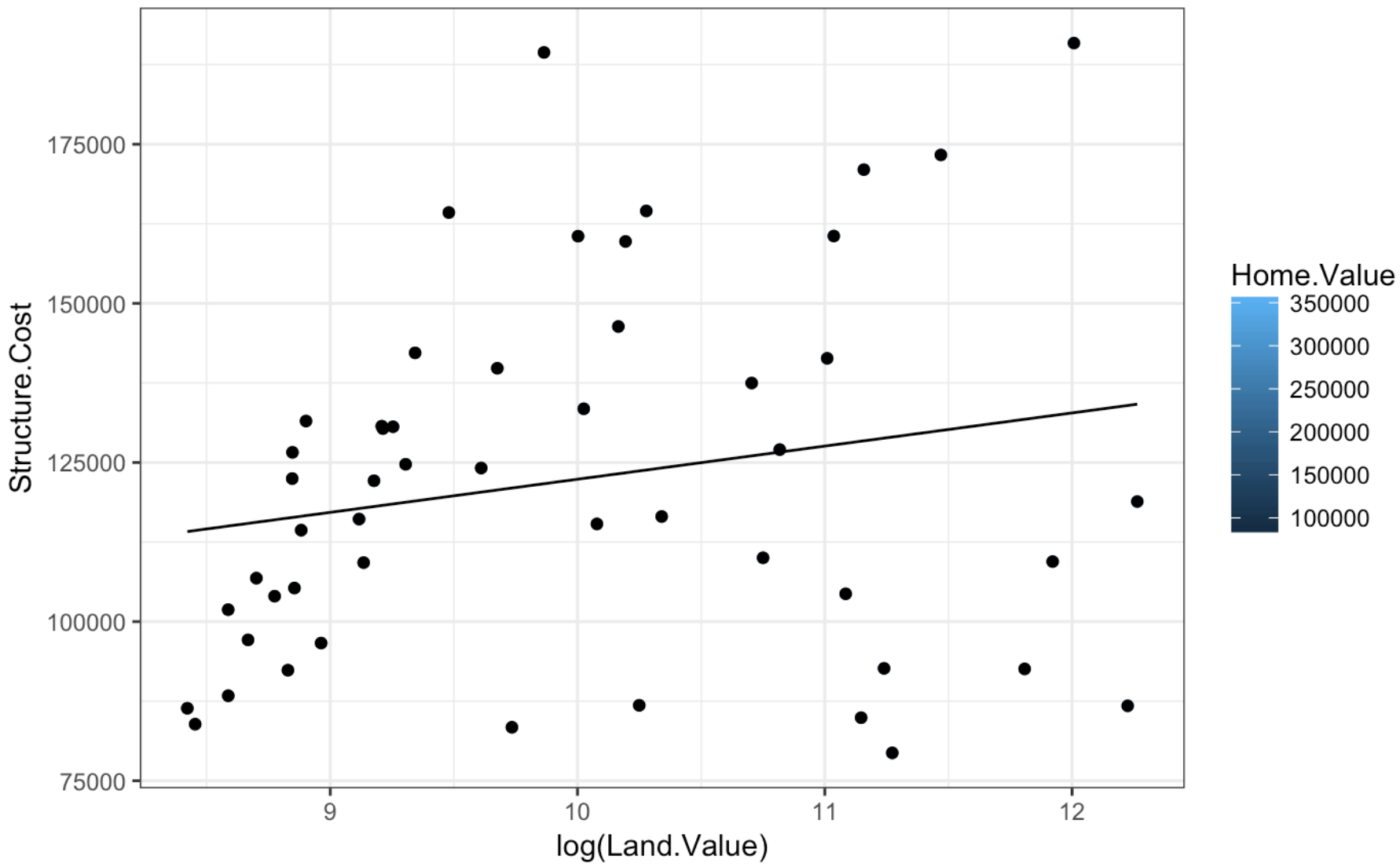
Hide

```
p1 + geom_text(aes(label = State), size = 3)
```



Hide

```
### install.packages("ggrepel")
library(ggrepel)
p1 + geom_point() + geom_text_repel(aes(label = State), size =3)
```

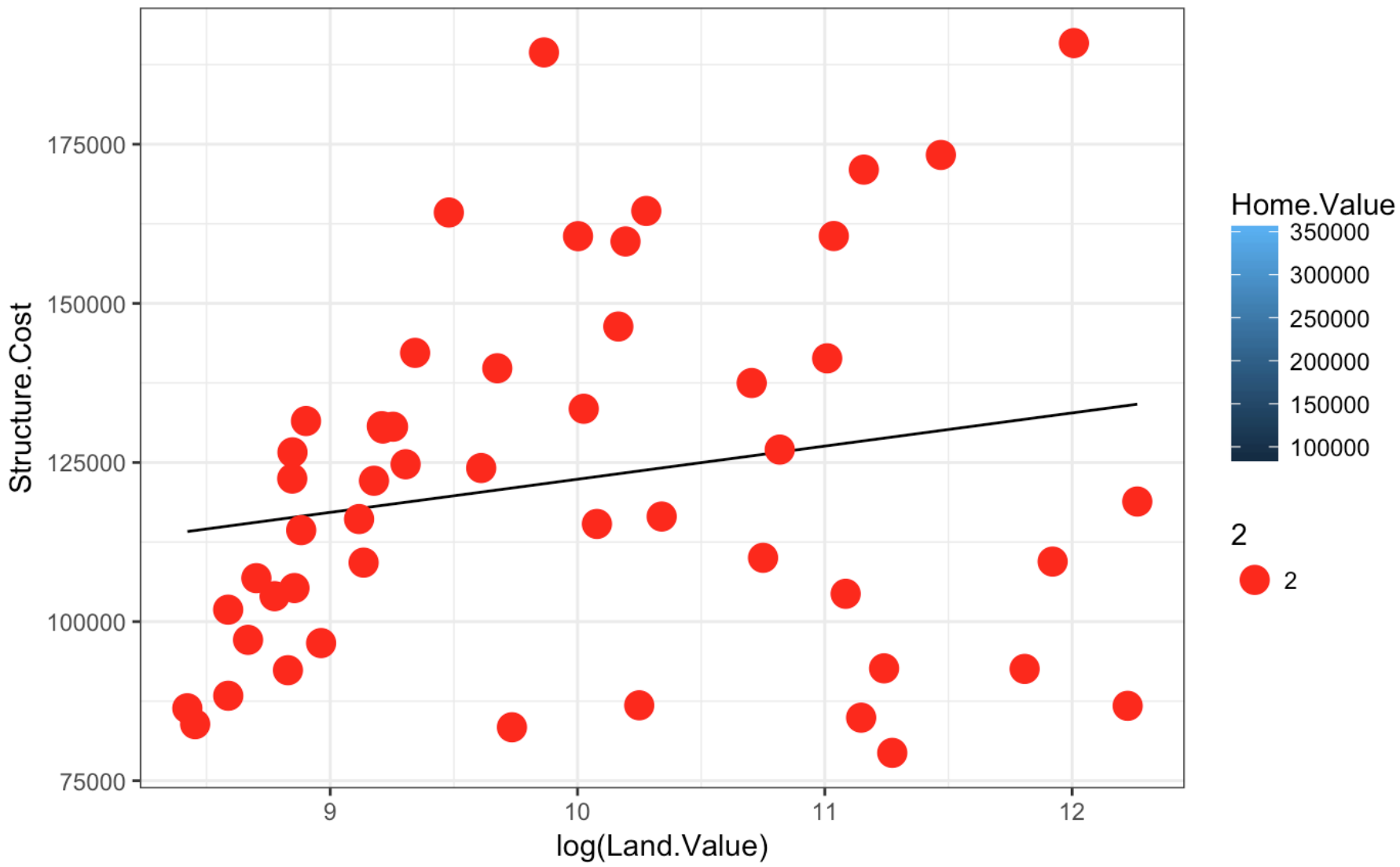


Aesthetic Mapping vs Assignment

Note that variables are mapped to aesthetics with the aes() function, while fixed aesthetics are set outside the aes() call. This sometimes leads to confusion, as in this example:

Hide

```
p1 + geom_point(aes(size=2), # incorrect! 2 is not a variable
                color = "red") # this is fine -- all points red
```

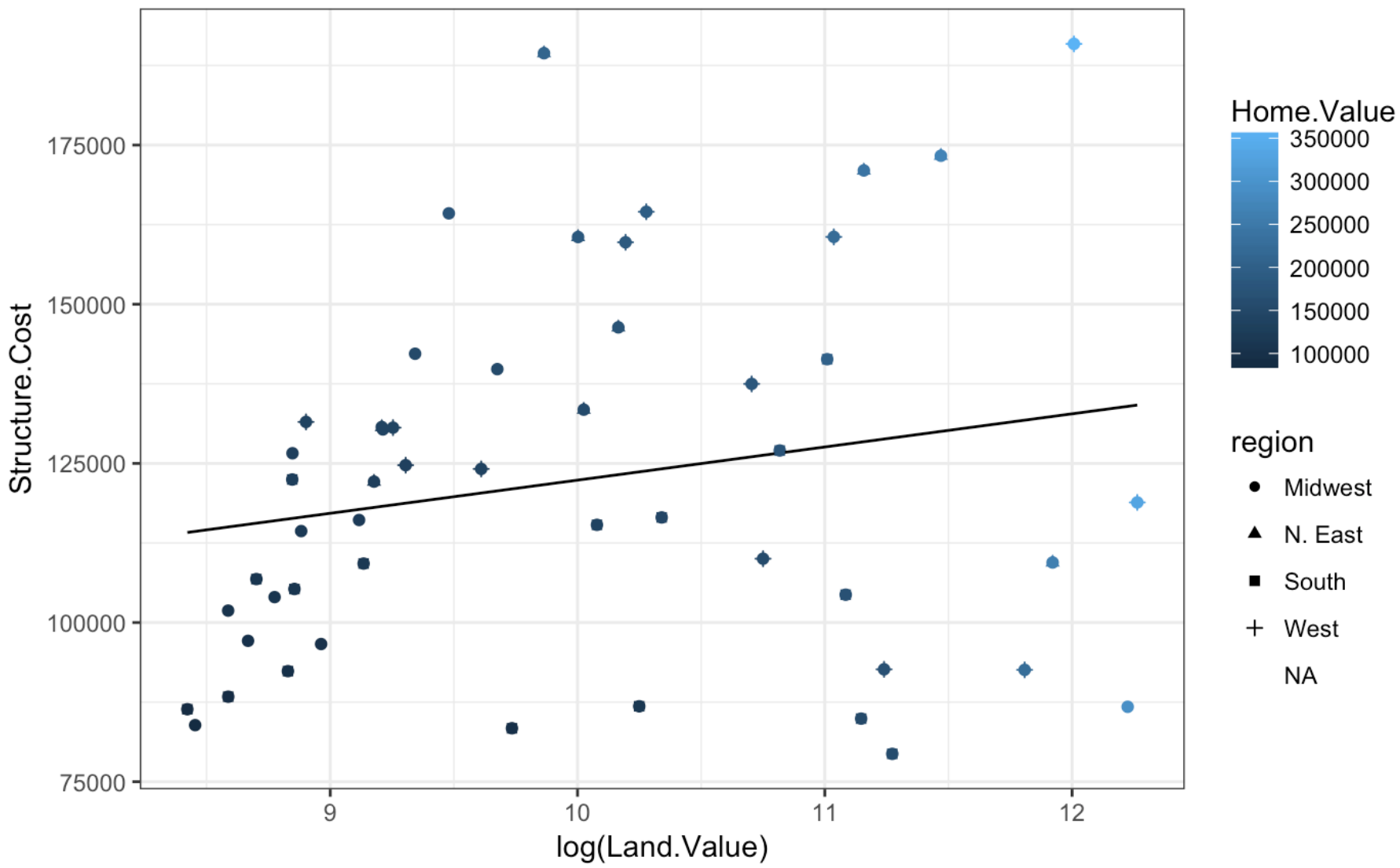


Mapping variables to other aesthetics

Other aesthetics are mapped in the same way as x and y in the previous example.

Hide

```
p1 + geom_point(aes(color=Home.Value, shape = region))
```



Exercise 1

For this exercise, we read in the `datasets/EconomistData.csv` file.

Hide

```
dat <- read.csv("datasets/EconomistData.csv")
head(dat)
```

	X	Country	HDI.Rank	HDI	CPI	Region
	<int>	<fctr>	<int>	<dbl>	<dbl>	<fctr>
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

6 rows

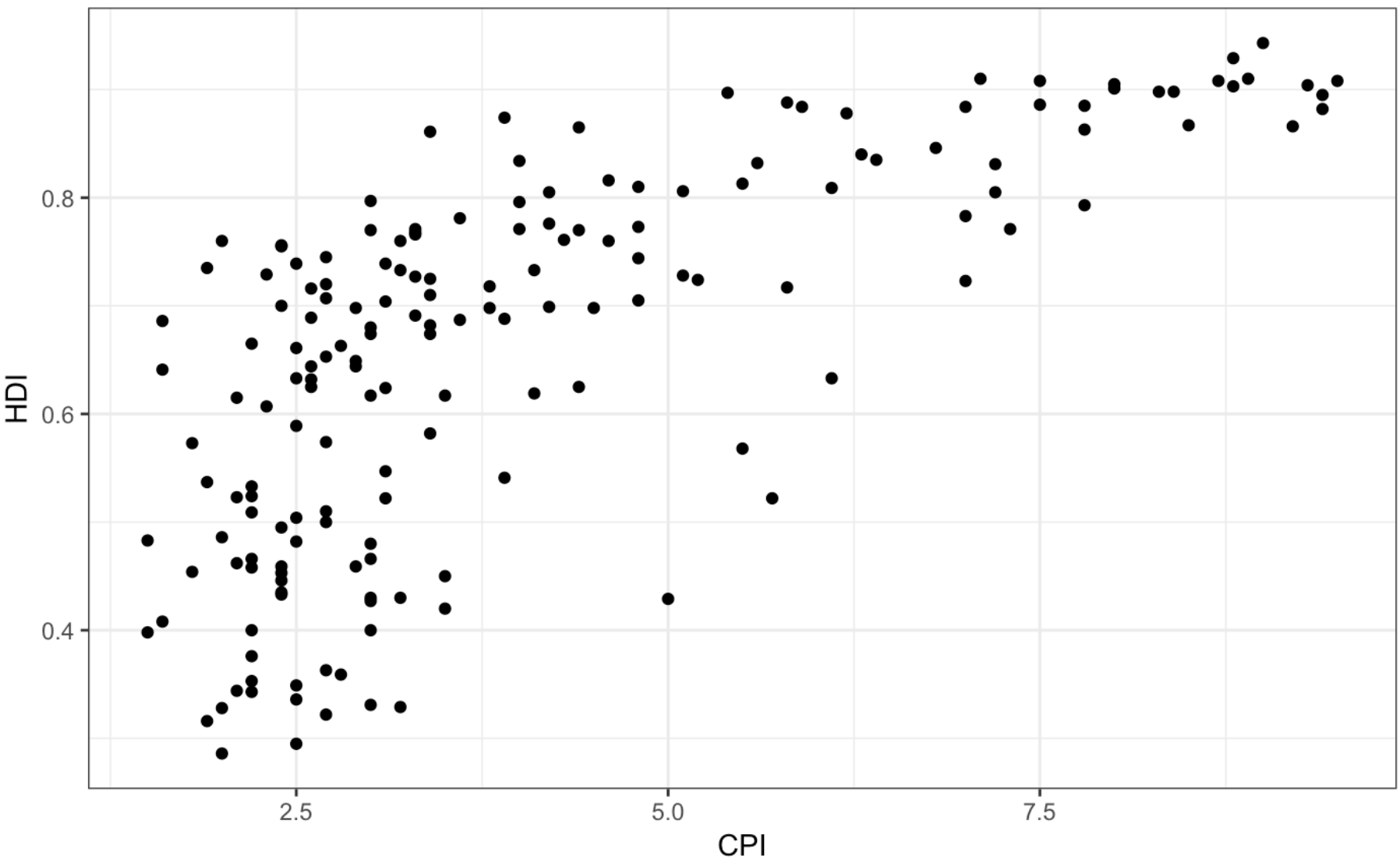
The original sources for these data are Transparency.org (http://www.transparency.org/content/download/64476/1031428) and UNDP (http://hdrstats.undp.org/en/indicators/display_cf_xls_indicator.cfm?indicator_id=103106&lang=en).

These data consist of Human Development Index and Corruption Perception Index scores for several countries.

1. Create a scatter plot with CPI on the x axis and HDI on the y axis.
2. Color the points blue.
3. Map the color of the the points to Region.
4. Make the points bigger by setting size to 2
5. Map the size of the points to HDI.Rank

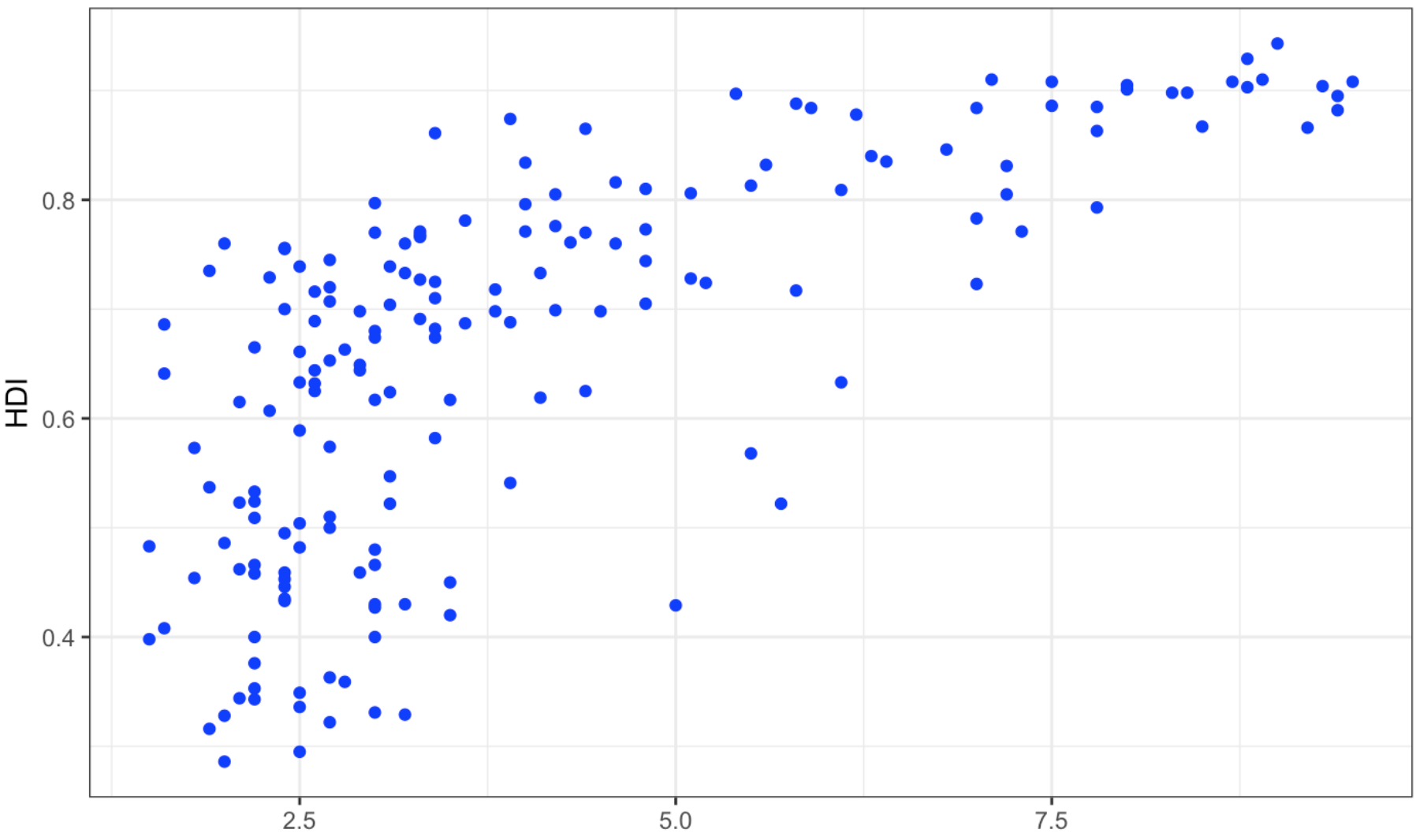
Hide

```
# Create a scatterplot with CPI on the x-axis and HDI on the y-axis
plot <- ggplot(data = dat, aes(x=CPI, y=HDI))
plot + geom_point()
```



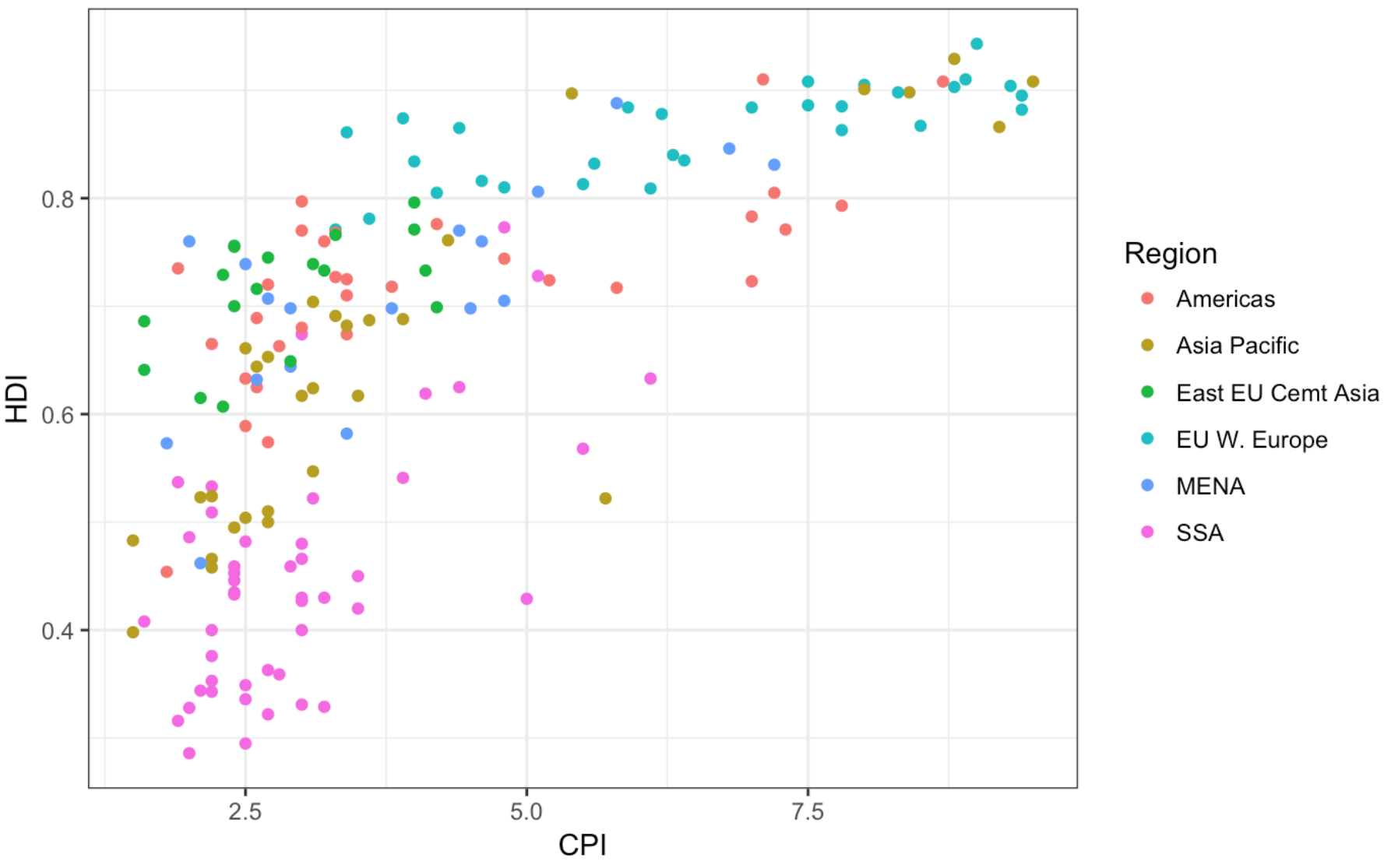
Hide

```
# Color the points blue
plot + geom_point(col="blue")
```



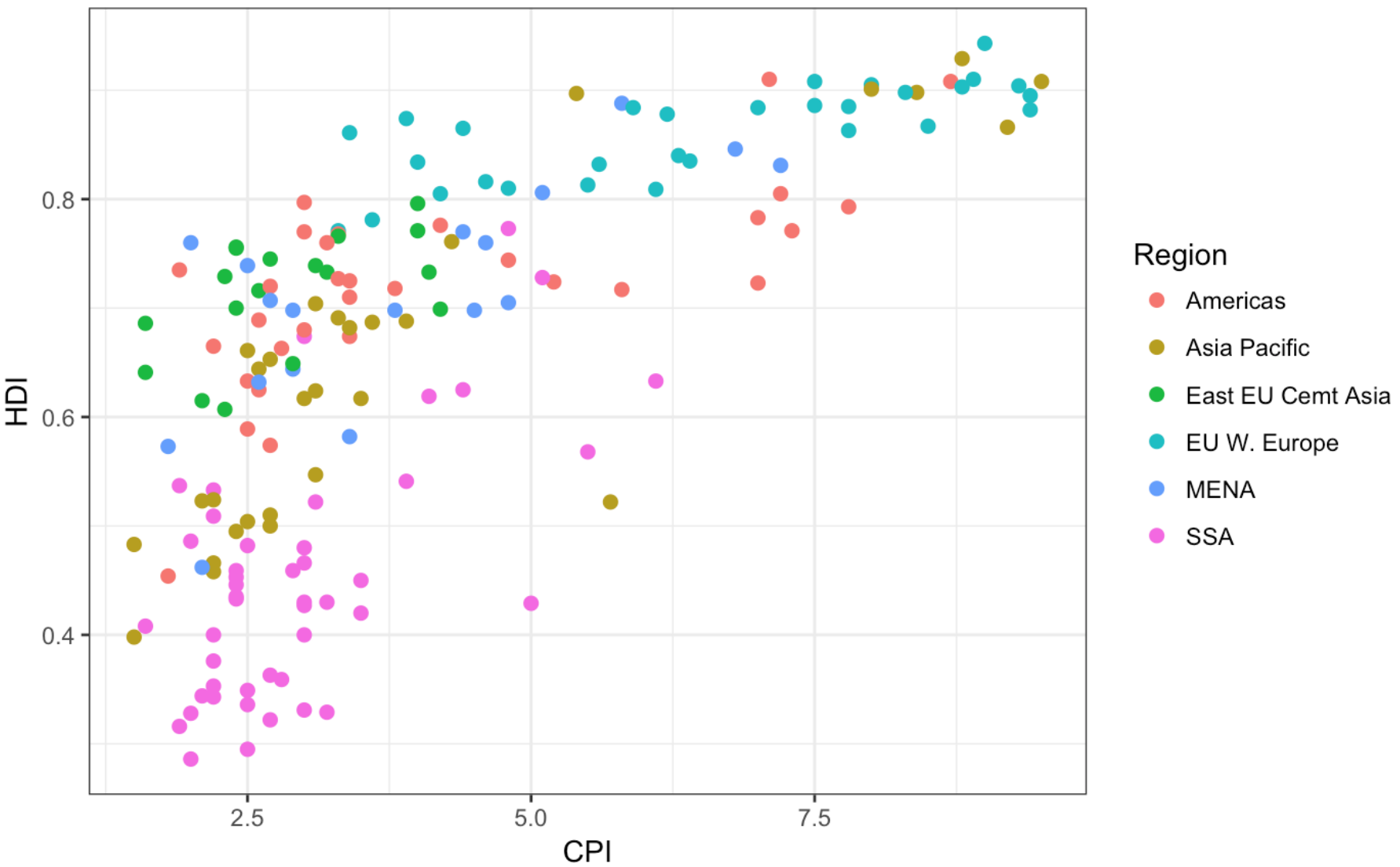
Hide

```
# Map the color of the points to Region
plot + geom_point(aes(col = Region))
```



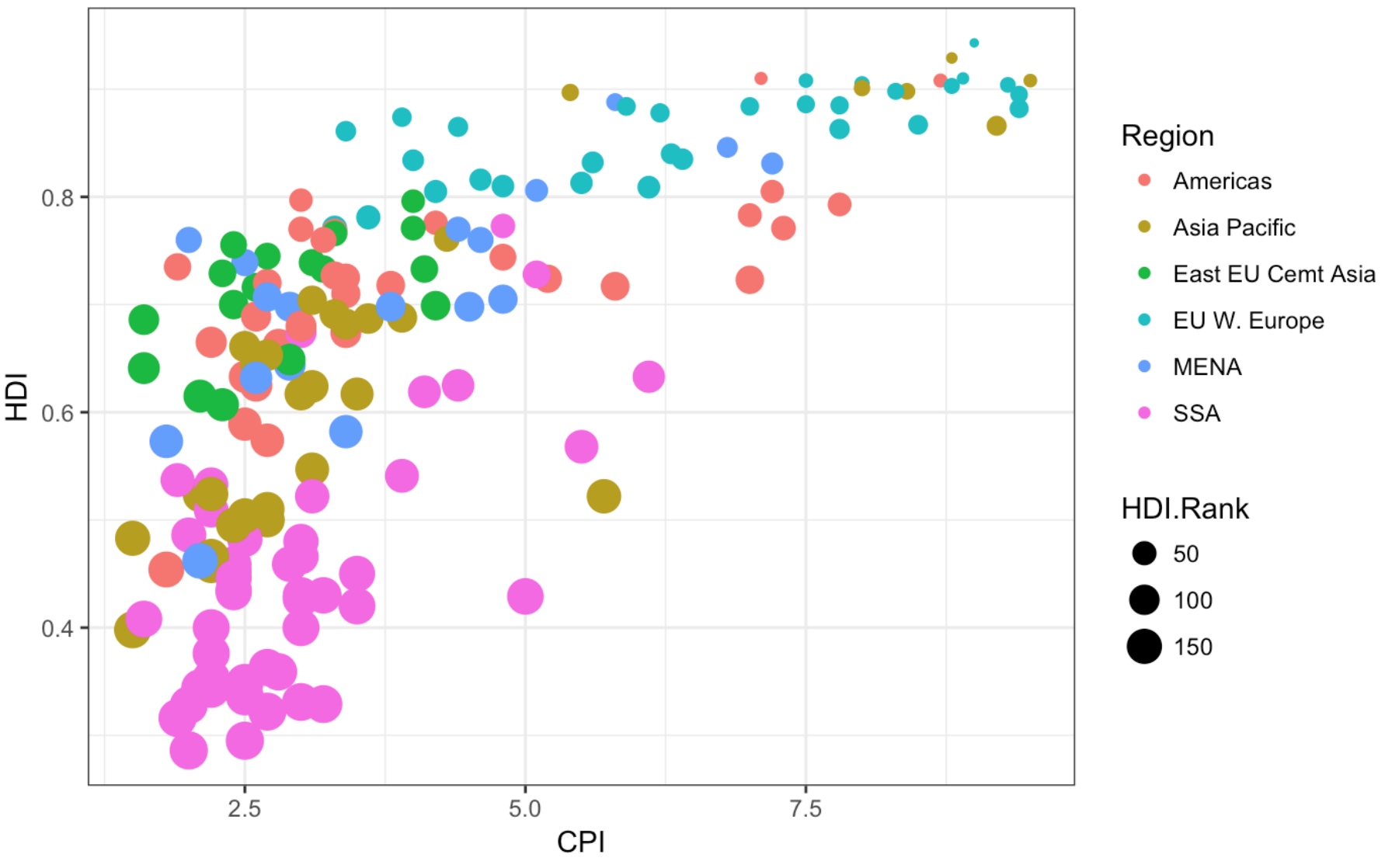
Hide

```
# Make the points bigger by setting size to 2
plot + geom_point(aes(color = Region), size = 2)
```



Hide

```
# Map the size of the points to HDI.Rank
plot + geom_point(aes(color = Region, size = HDI.Rank))
```



Statistical Transformation

Some plot types (such as scatterplots) do not require transformations–each point is plotted at x and y coordinates equal to the original value. Other plots, such as boxplots, histograms, prediction lines etc. require statistical transformations:

- For a boxplot the y values must be transformed to the median and 1.5(IQR)
- For a smoother plot the y values must be transformed into predicted values

Each `geom` has a default statistic, but these can be changed. For example, the default statistic for geom_bar is `stat_bin` :

Hide

```
args(geom_histogram)
```

```
function (mapping = NULL, data = NULL, stat = "bin", position = "stack",
    ..., binwidth = NULL, bins = NULL, na.rm = FALSE, show.legend = NA,
    inherit.aes = TRUE)
NULL
```

Hide

```
args(stat_bin)
```

```
function (mapping = NULL, data = NULL, geom = "bar", position = "stack",
    ..., binwidth = NULL, bins = NULL, center = NULL, boundary = NULL,
    breaks = NULL, closed = c("right", "left"), pad = FALSE,
    na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)
NULL
```

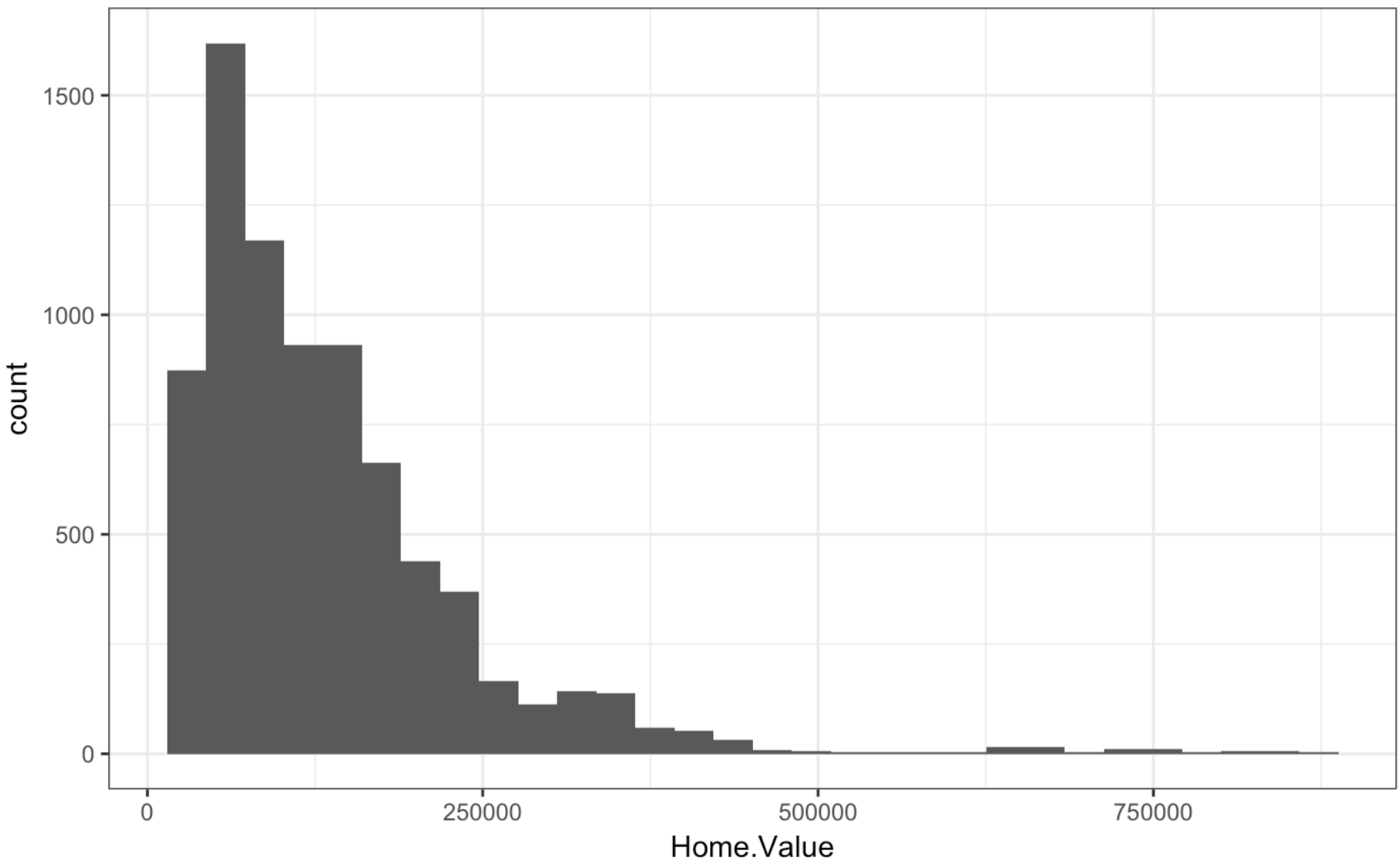
Setting statistical transformation arguments

Arguments to stat_ functions can be passed through geom_ functions. This can be slightly annoying because in order to change it you have to first determine which stat the geom uses, then determine the arguments to that stat.

For example, here is the default histogram of Home.Value:

Hide

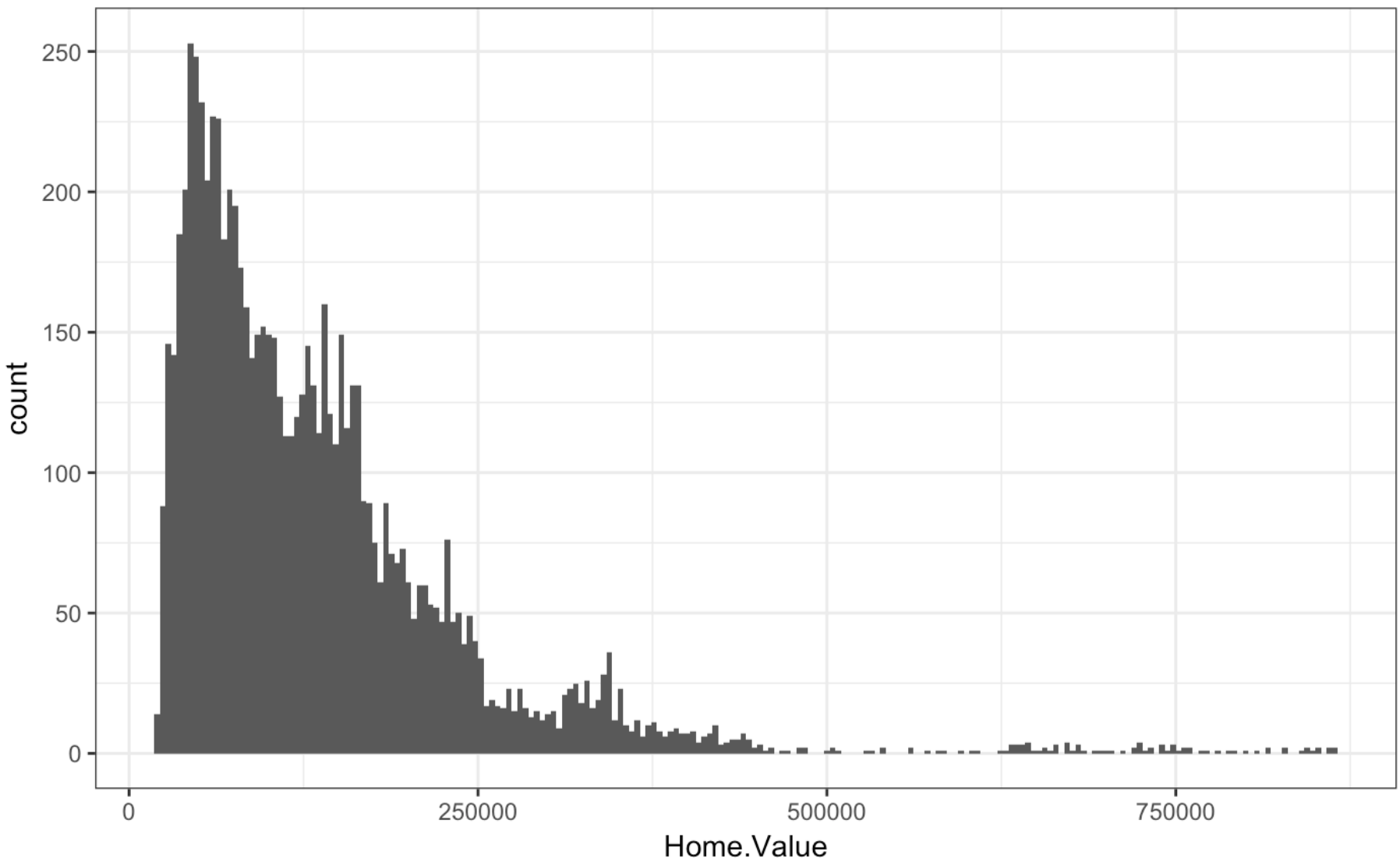
```
p2 <- ggplot(housing, aes(x=Home.Value))
p2 + geom_histogram()
```



Reasonable by default, but we can change it by passing the `binwidth` argument to the `stat_bin` function:

Hide

```
p2 + geom_histogram(stat = "bin", binwidth=4000)
```



Changing the statistical transformation

Sometimes the default statistical transformation is not what you need. This is often the case with pre-summarized data:

Hide

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

	State <fctr>	Home.Value <dbl>
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

1-10 of 12 rows Previous **1** 2 Next

The following code will throw `Error: stat_count() must not be used with a y aesthetic.` :

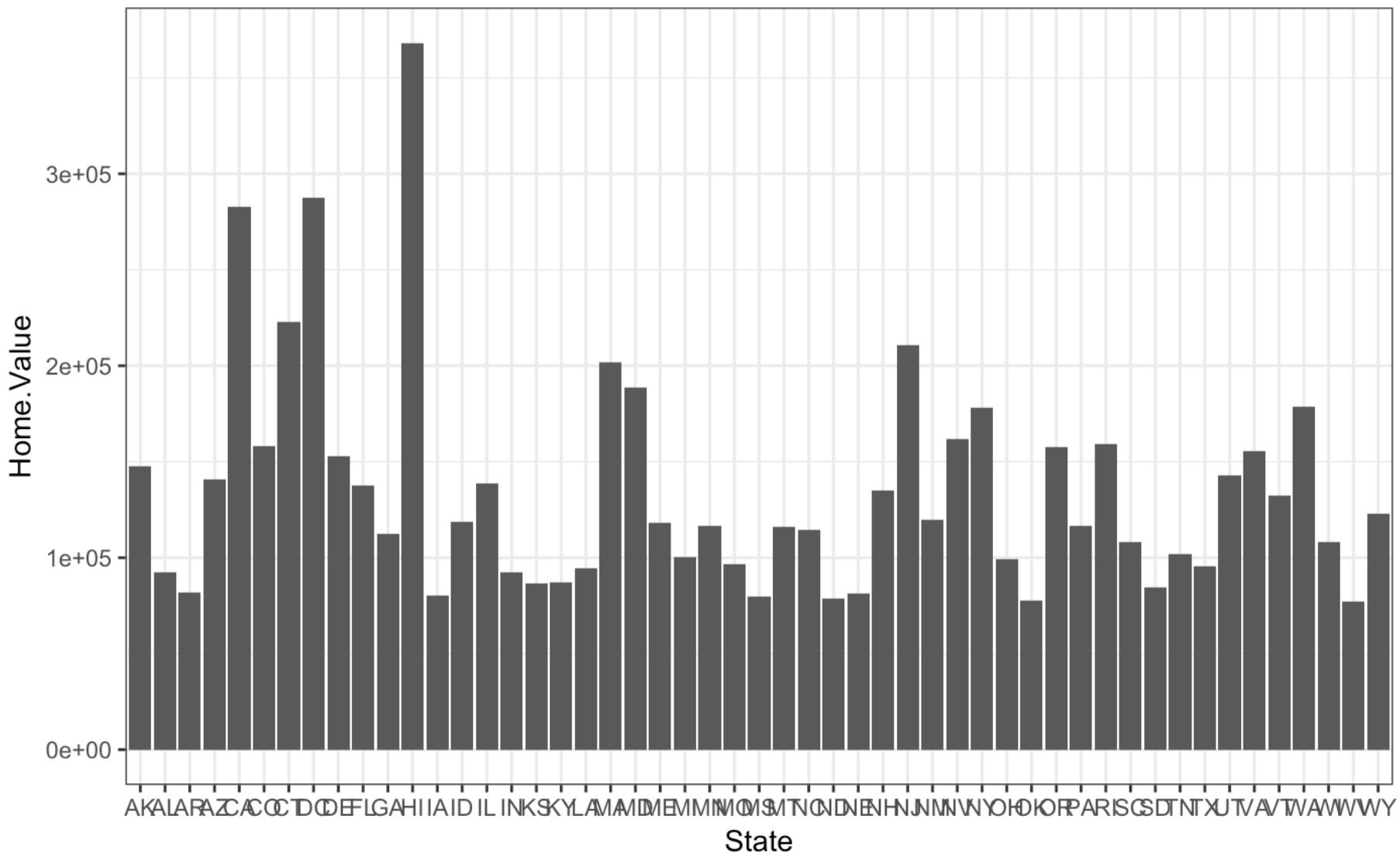
Hide

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

This is because we are taking binned (summarized by default) and our summarized data and requesting ggplot to bin and summarize it again. Recall that geom_bar defaults to stat = stat_count. We can override that by explicitly telling geom_bar to use a different statistical transformation function:

Hide

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

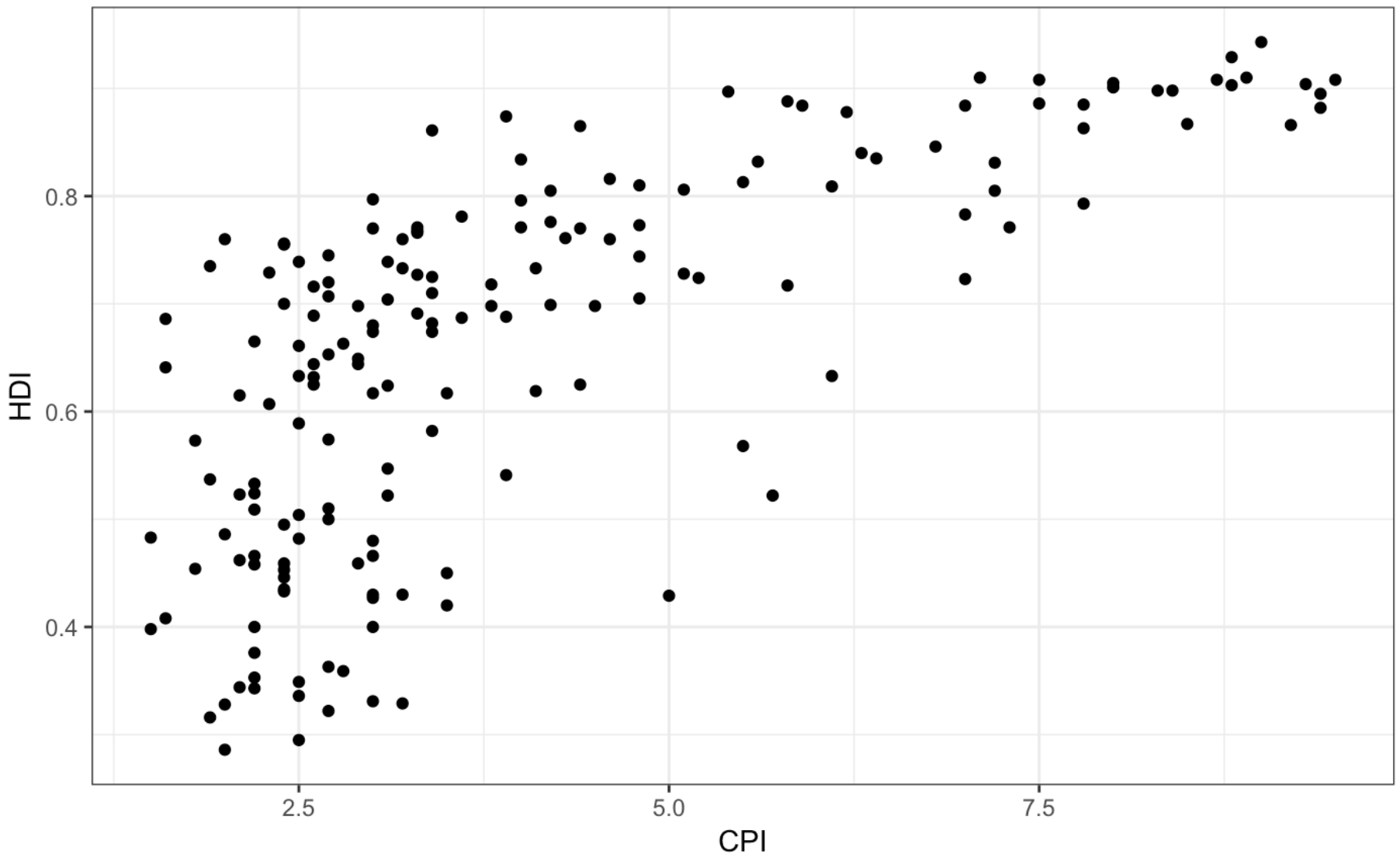


Exercise 2

1. Re-create a scatter plot with CPI on the x axis and HDI on the y axis (as you did in the previous exercise).
2. Overlay a smoothing line on top of the scatter plot using `geom_smooth`
3. Overlay a smoothing line on top of the scatter plot using `geom_smooth`, but use a linear model for the predictions. Hint: see `?stat_smooth`.
4. Overlay a smoothing line on top of the scatter plot using `geom_line`. Hint: change the statistical transformation.
5. BONUS: Overlay a smoothing line on top of the scatter plot using the default loess method, but make it less smooth. Hint: see `?loess`.

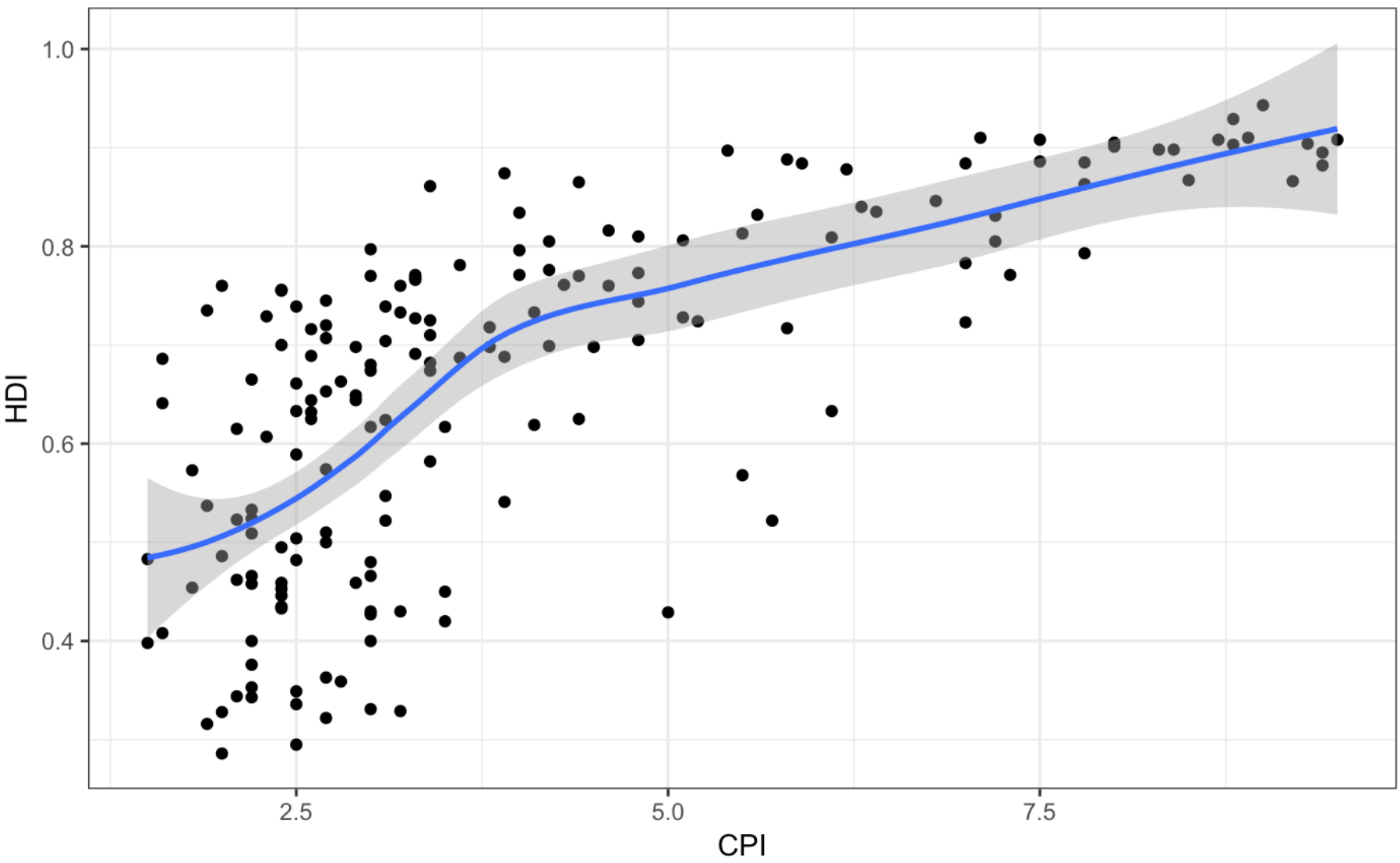
Hide

```
plot2 <- ggplot(data = dat, aes(x=CPI, y=HDI)) + geom_point()
plot2
```



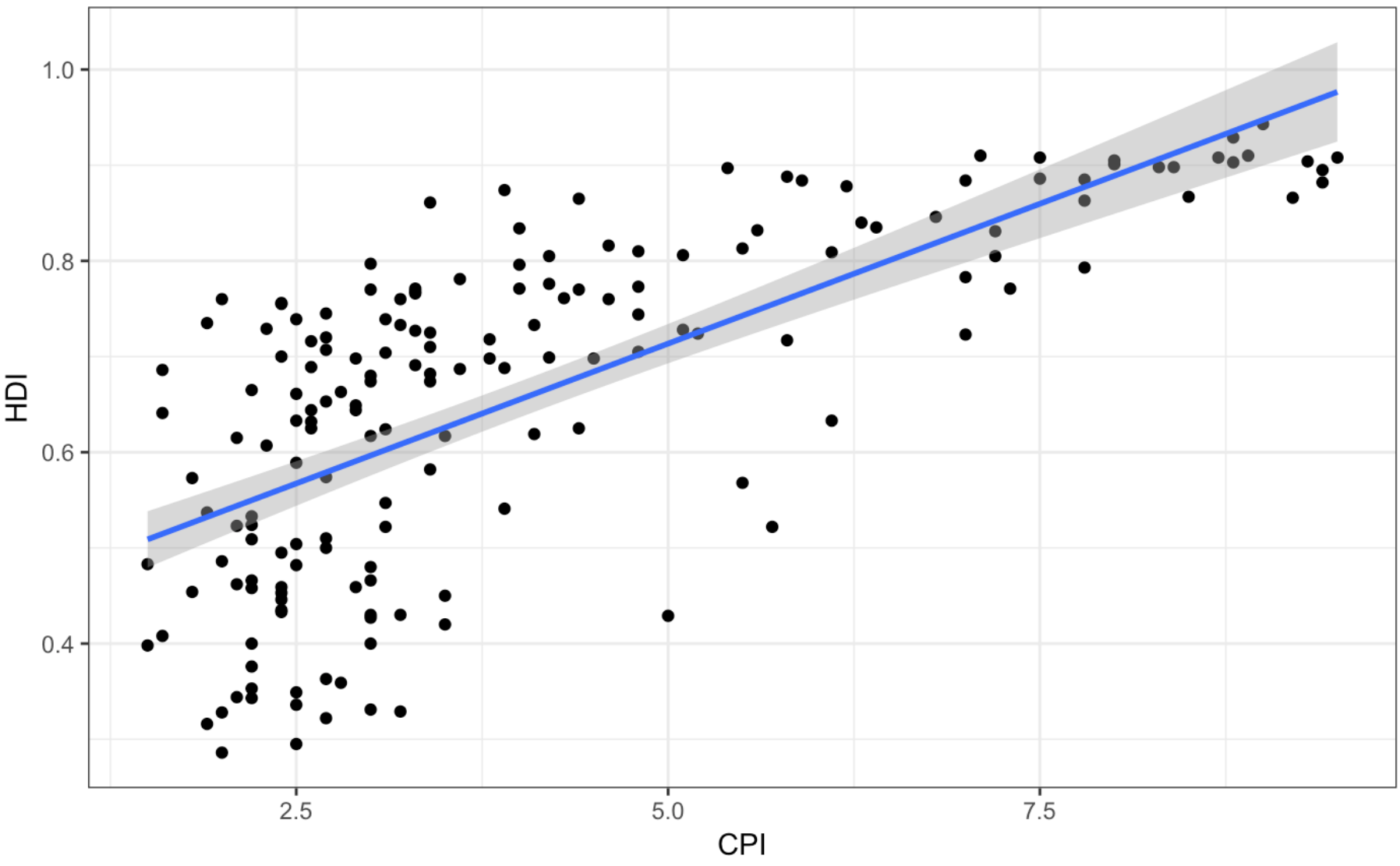
Hide

```
plot2 + geom_smooth()
```



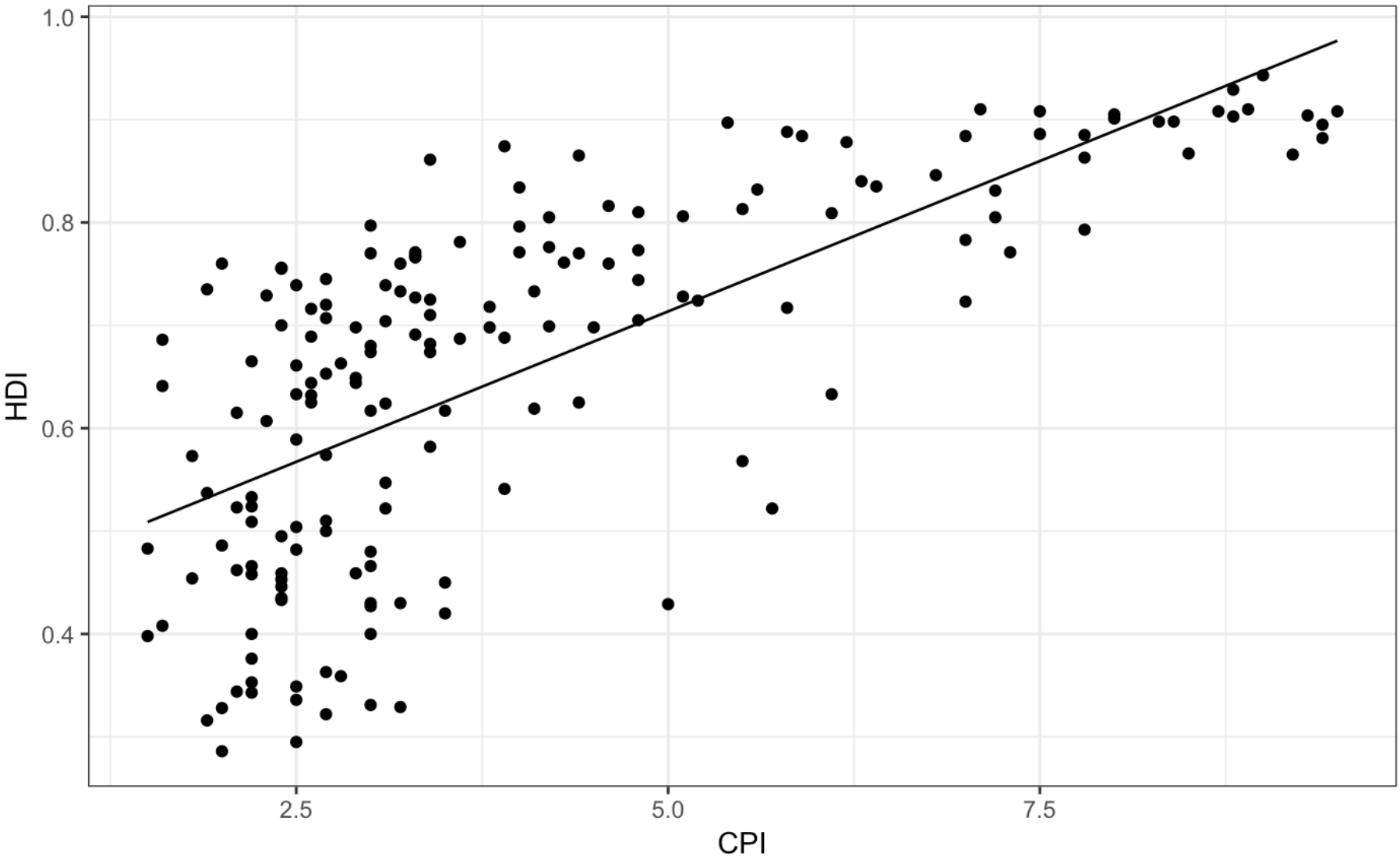
Hide

```
# Overlay a smoothing line on top of the scatter plot using `geom_smooth`, but use a linear model for the predict
ions. Hint: see `?stat_smooth`.
plot2 + geom_smooth(method = "lm")
```



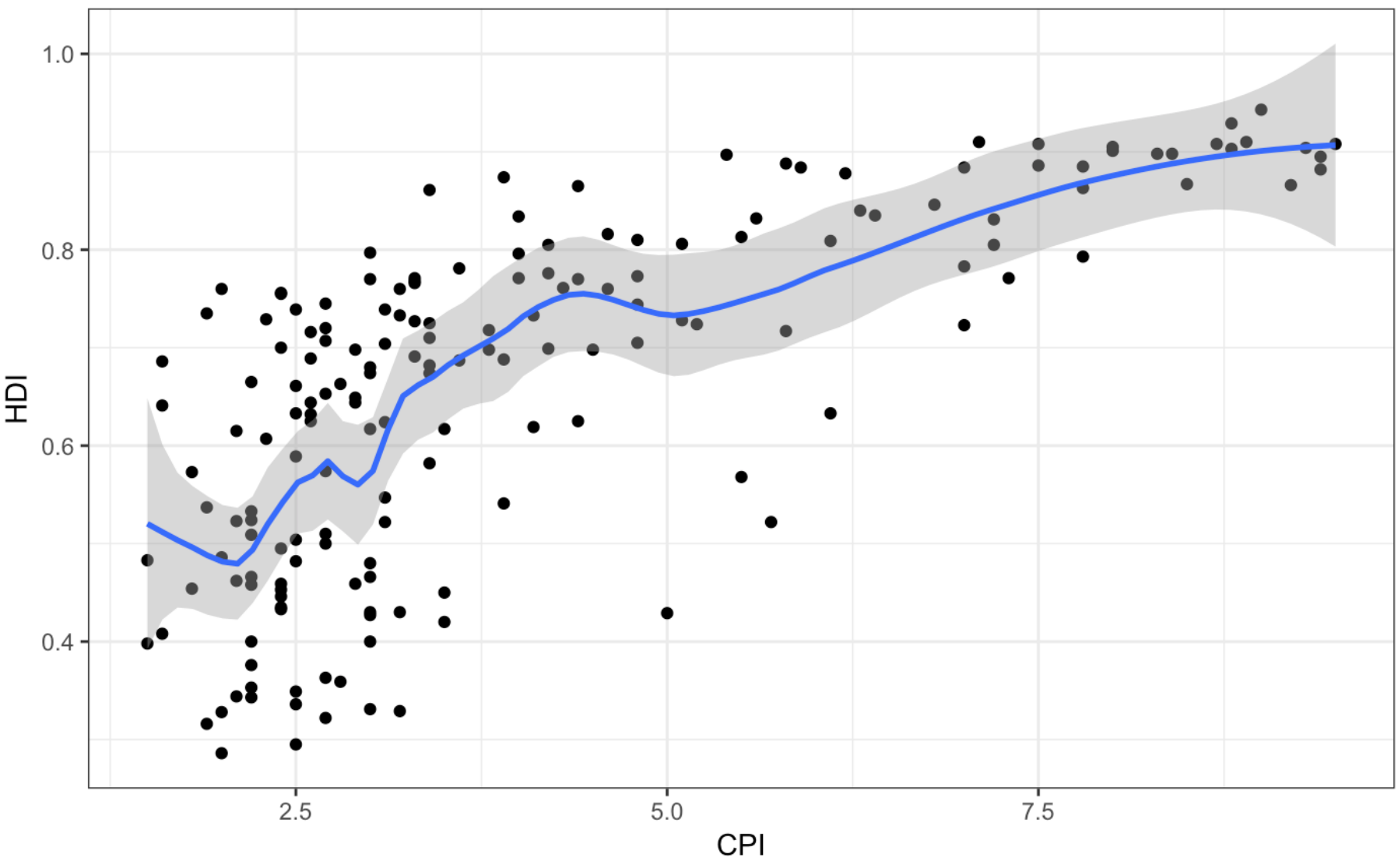
Hide

```
# Overlay a smoothing line on top of the scatter plot using `geom_line`. Hint: change the statistical transformat
ion.
plot2 + geom_line(aes(y= predict(lm(HDI ~ CPI, data = dat))))
```



Hide

```
# Bonus: Overlay a smoothing line on top of the scatter plot using the default loess method, but make it less smo
oth. Hint: see `?loess`.
plot2 + geom_smooth(method = "loess", span = 0.3)
```



As observed, we can use `span` to control the "wiggliness" of the default loess smoother. The span is the fraction of points used to fit each local regression. We use 0.3 or a smaller number than that to achieve a wigglier curver.

Scales

Scales: Controlling aesthetic mapping

Aesthetic mapping (i.e., with `aes()`) only says that a variable should be mapped to an aesthetic. It doesn't say how that should happen. For example, when mapping a variable to shape with `aes(shape = x)` you don't say what shapes should be used. Similarly, `aes(color = z)` doesn't say what colors should be used. Describing what colors/shapes/sizes etc. to use is done by modifying the corresponding *scale*. In `ggplot2` scales include:

- Position
- Color and fill
- Size
- Shape
- Line type

Scales are modified with a series of functions using a `scale_<aesthetic>_<type>` naming scheme. Try typing `scale_<tab>` to see a list of scale modification functions.

Common scale arguments

- `name`
  - the first argument gives the axis or legend title
- `limits`
  - the minimum and maximum of the scale
- `breaks`
  - the points along the scale where labels should appear
- `labels`
  - the labels that appear at each break

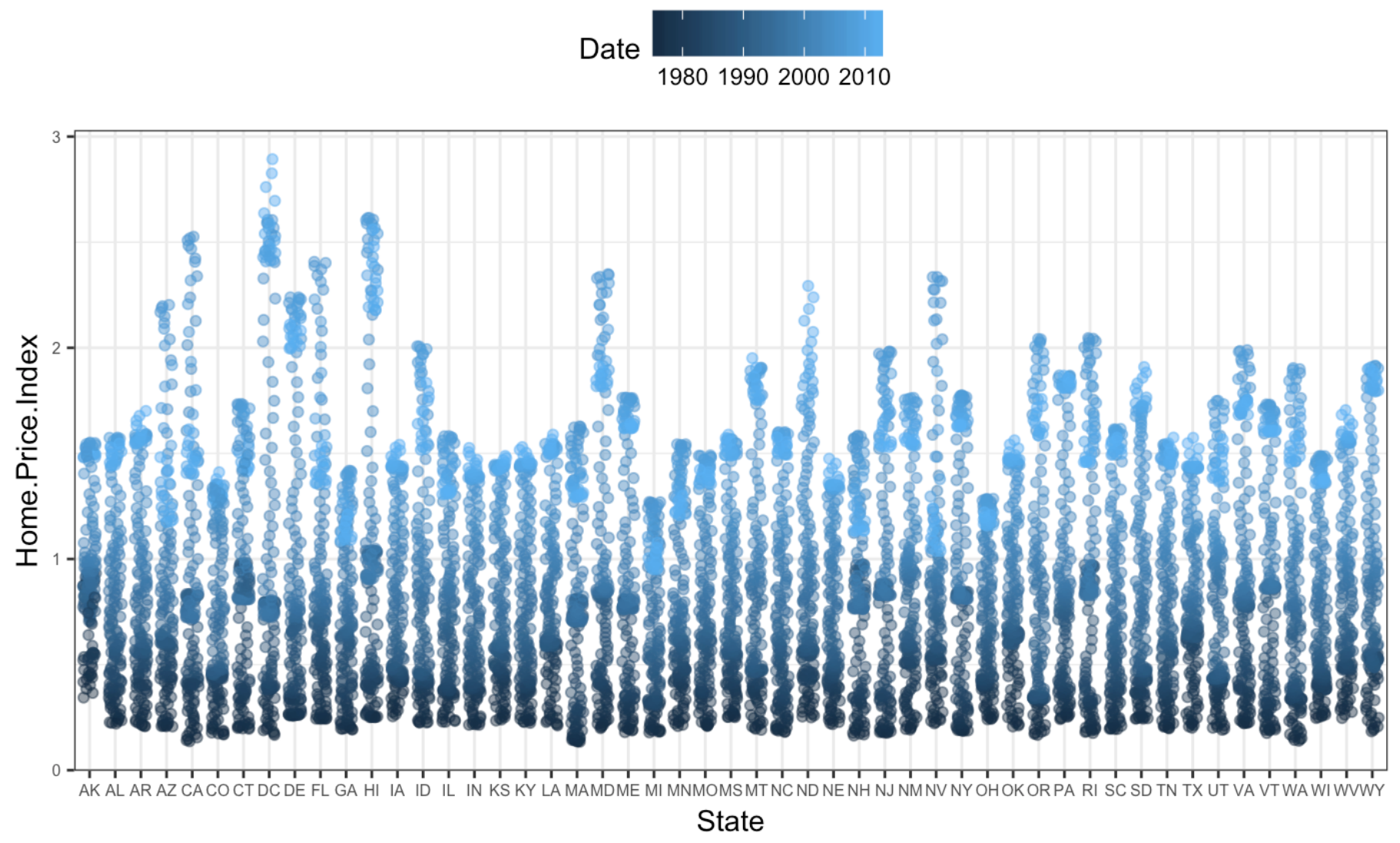
Specific scale functions may have additional arguments; for example, the `scale_color_continuous` function has arguments `low` and `high` for setting the colors at the low and high end of the scale.

Scale modification examples

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

Hide

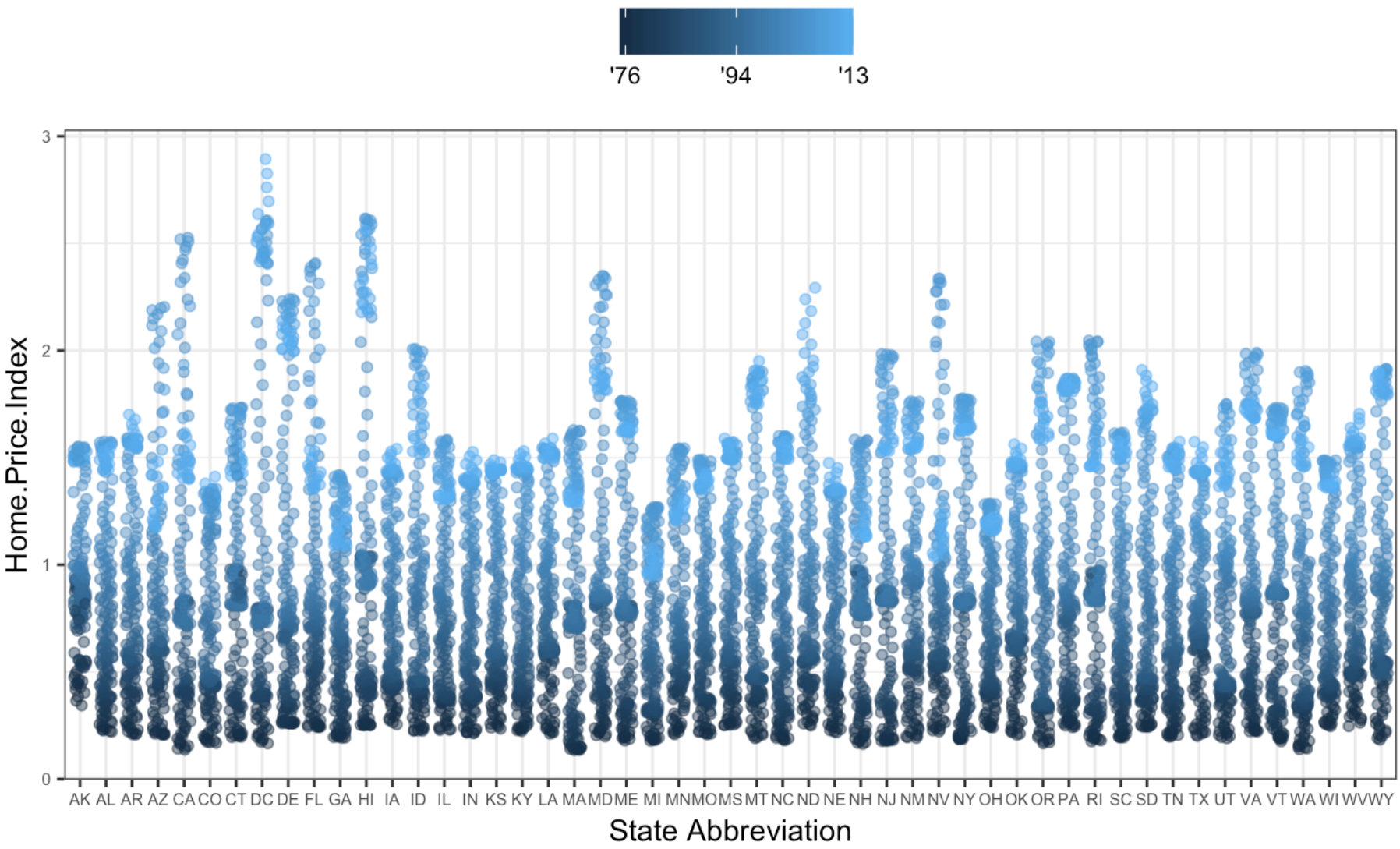
```
p3 <- ggplot(housing, aes(x=State, y=Home.Price.Index)) + theme(legend.position = "top", axis.text =
element_text(size=6))
p4 <- p3 + geom_point(aes(color=Date), alpha=0.5, size=1.5, position = position_jitter(width=0.25, height=0))
p4
```



Now modify the breaks for the x axis and color scales

Hide

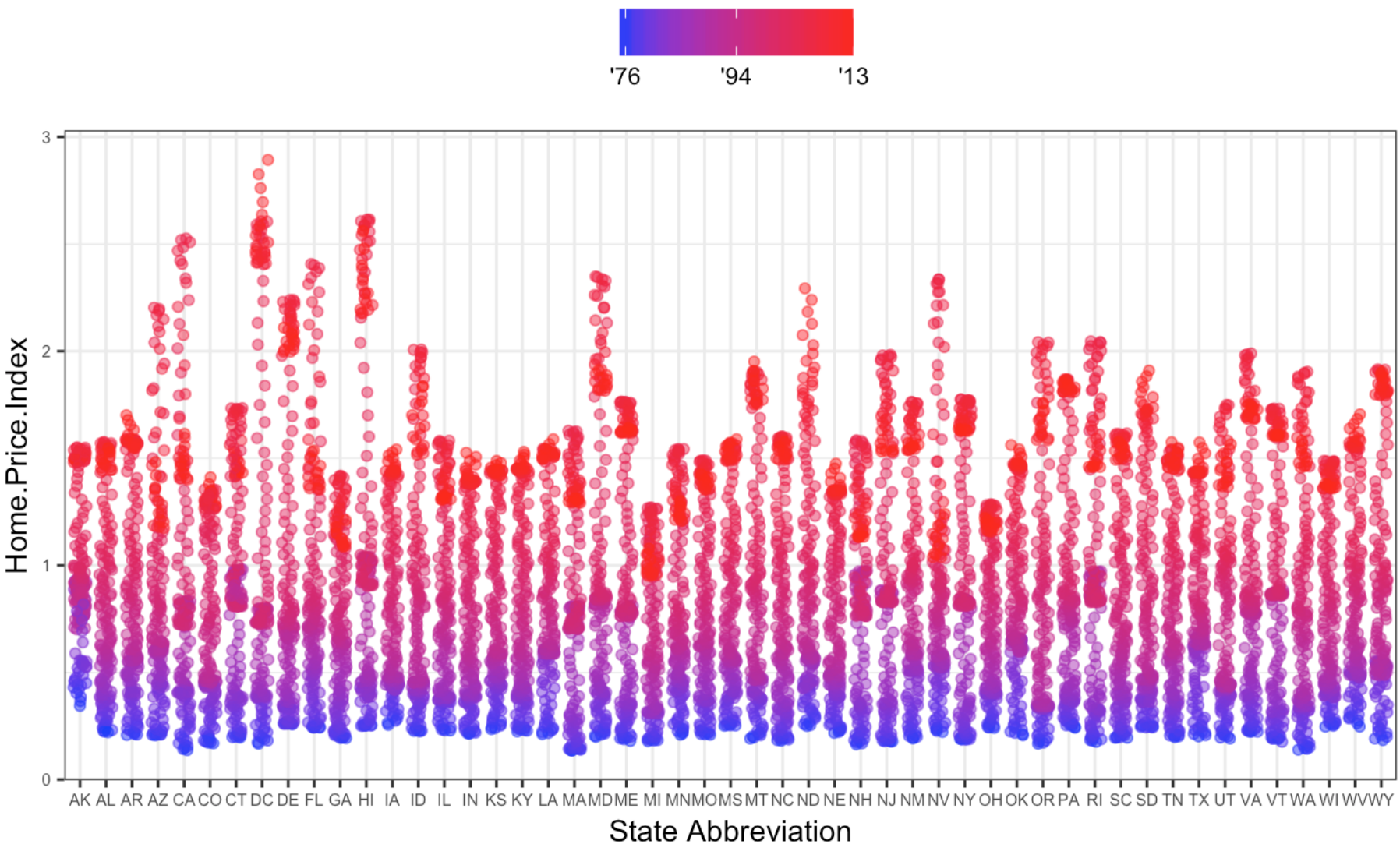
```
p4 + scale_x_discrete(name="State Abbreviation") + scale_color_continuous(name="", breaks = c(1976, 1994, 2013),
labels = c("'76", "'94", "'13"))
```



Next change the low and high values to blue and red:

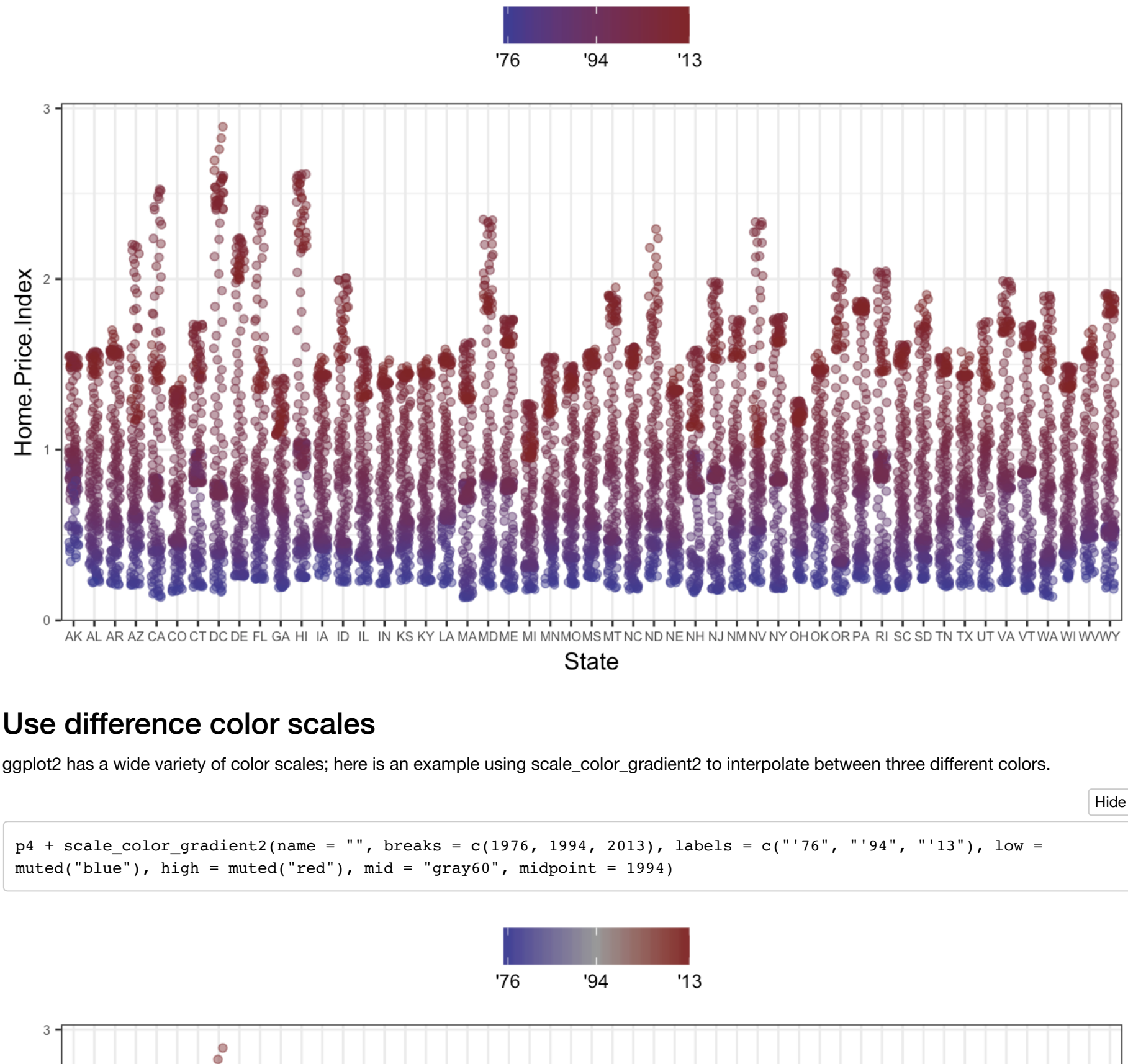
Hide

```
p4 + scale_x_discrete("State Abbreviation") + scale_color_continuous(name = "", breaks = c(1976, 1994, 2013), lab
els = c("'76", "'94", "'13"), low="blue", high="red")
```



Hide

```
library(scales)
p4 +
  scale_color_continuous(name="",
                         breaks = c(1976, 1994, 2013),
                         labels = c("'76", "'94", "'13"),
                         low = muted("blue"), high = muted("red"))
```

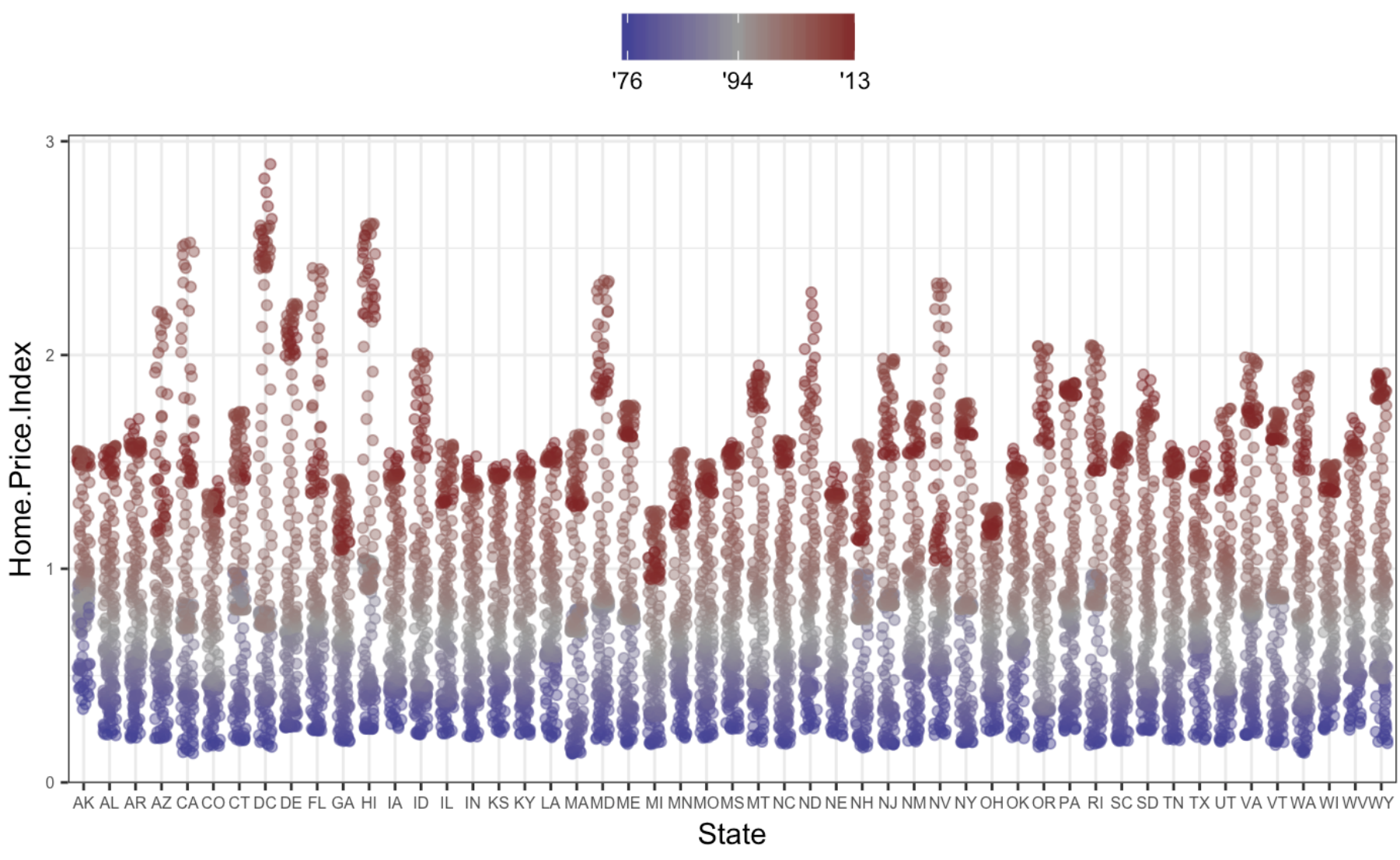


Use difference color scales

ggplot2 has a wide variety of color scales; here is an example using scale_color_gradient2 to interpolate between three different colors.

Hide

```
p4 + scale_color_gradient2(name = "", breaks = c(1976, 1994, 2013), labels = c("'76", "'94", "'13"), low =
muted("blue"), high = muted("red"), mid = "gray60", midpoint = 1994)
```

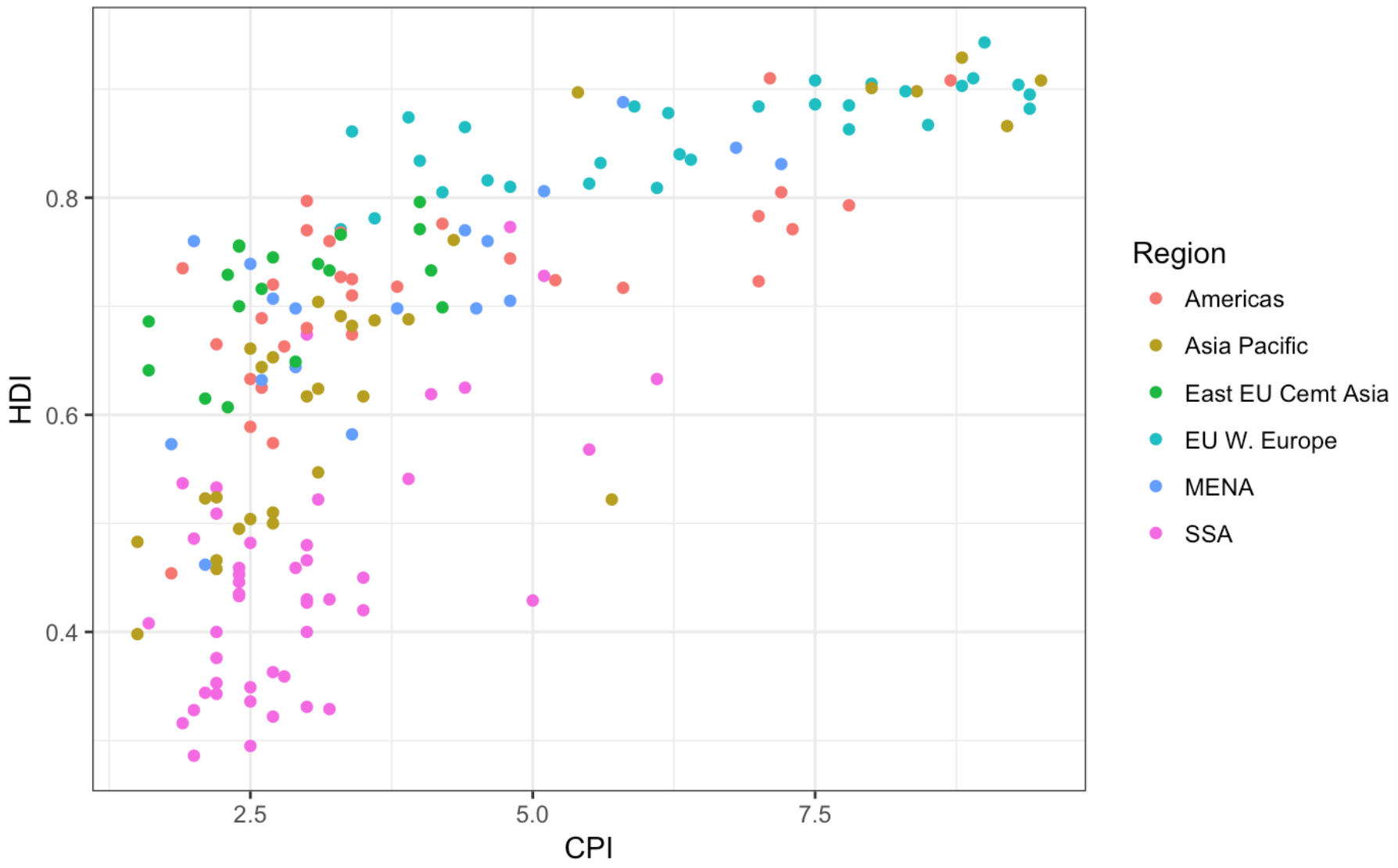


Exercise 3

1. Create a scatter plot with CPI on the x axis and HDI on the y axis. Color the points to indicate region.
2. Modify the x, y, and color scales so that they have more easily-understood names (e.g., spell out “Human development Index” instead of “HDI”).
3. Modify the color scale to use specific values of your choosing. Hint: see `?scale_color_manual` .

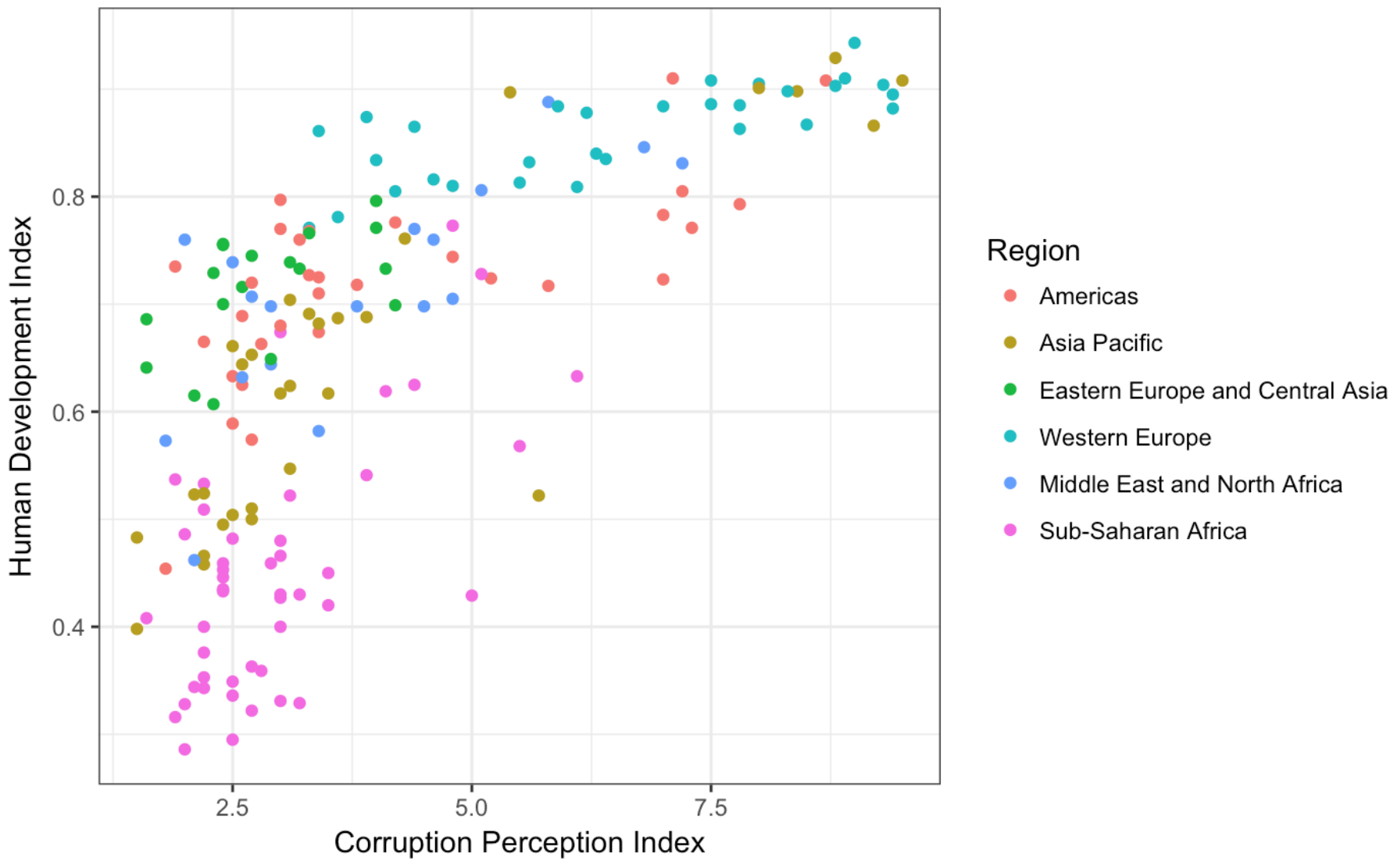
Hide

```
# Create a scatterplot with CPI on the x axis and HDI on the y axis. Color the points to indicate region
plot3 <- ggplot(data = dat, aes(x=CPI, y=HDI)) + geom_point(aes(color=Region))
plot3
```



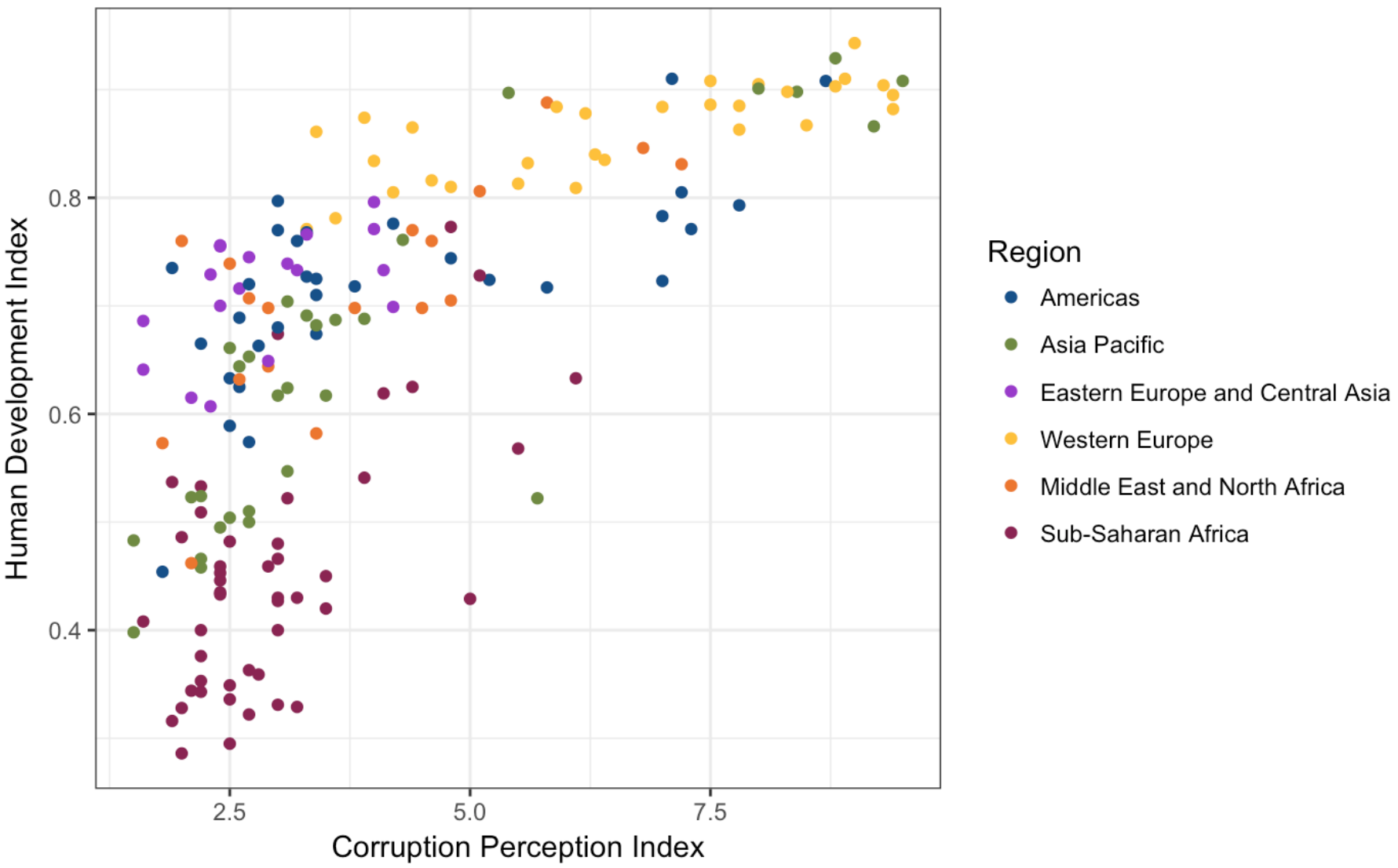
Hide

```
# Modify the x, y, and color scales so that they have more easily-understood names (e.g., spell out "Human develo
pment Index" instead of "HDI").
plot3 + scale_x_continuous(name="Corruption Perception Index") + scale_y_continuous(name="Human Development Inde
x") + scale_color_discrete(labels=c("Americas", "Asia Pacific", "Eastern Europe and Central Asia", "Western Europ
e", "Middle East and North Africa", "Sub-Saharan Africa"))
```



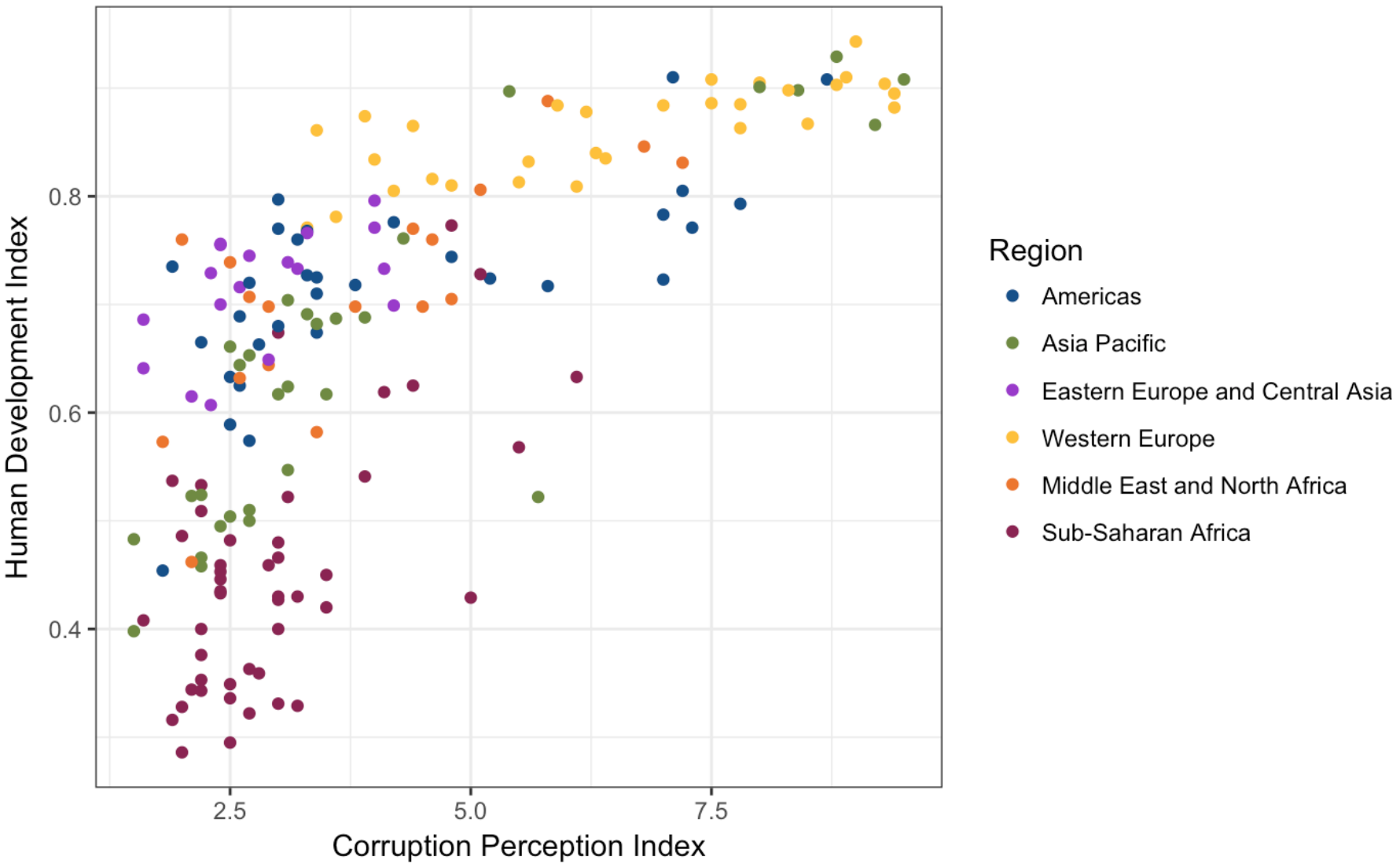
Hide

```
# Modify the color scale to use specific values of your choosing. Hint: see `?scale_color_manual`
plot3 + scale_x_continuous(name="Corruption Perception Index") + scale_y_continuous(name="Human Development Inde
x") + scale_color_manual(name = "Region", values = c("dodgerblue4", "darkolivegreen4", "darkorchid3", "goldenrod
1", "chocolate2", "violetred4"), labels=c("Americas", "Asia Pacific", "Eastern Europe and Central Asia", "Western
 Europe", "Middle East and North Africa", "Sub-Saharan Africa"))
```



Hide

```
# Bonus: Add geom_text
plot3 + scale_x_continuous(name="Corruption Perception Index") + scale_y_continuous(name="Human Development Inde
x") + scale_color_manual(name = "Region", values = c("dodgerblue4", "darkolivegreen4", "darkorchid3", "goldenrod
1", "chocolate2", "violetred4"), labels=c("Americas", "Asia Pacific", "Eastern Europe and Central Asia", "Western
 Europe", "Middle East and North Africa", "Sub-Saharan Africa")) + geom_text_repel(aes(label=Country), data =
dat[dat$Country %in% c("Singapore", "Malaysia", "Indonesia", "Vietnam","Philippines", "Thailand"),], size=2)
```



Faceting

Faceting concepts

Faceting is `ggplot2` parlance for small multiples. The idea is to create separate graphs for subsets of data. `ggplot2` offers two functions for creating small multiples:

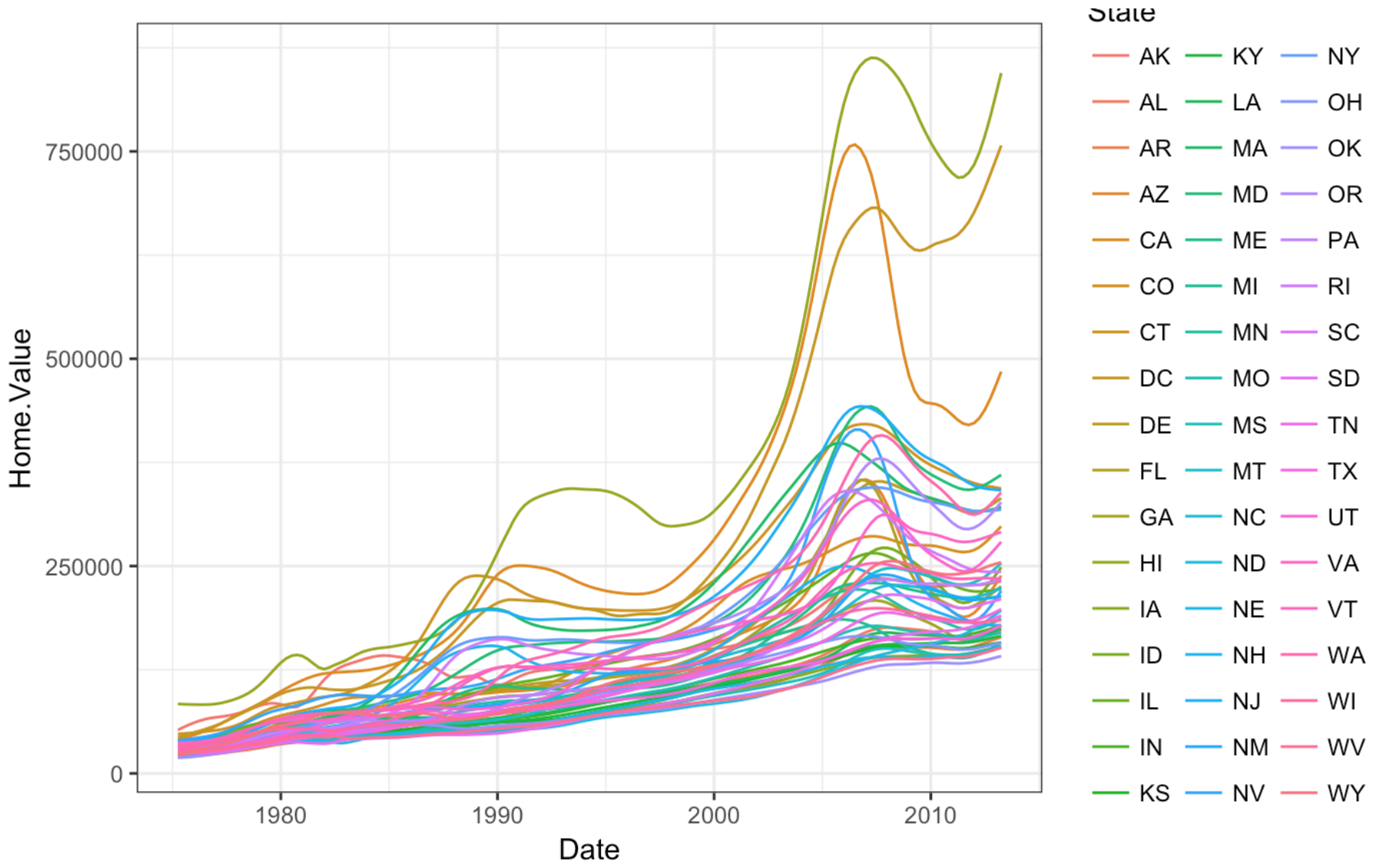
- facet_wrap(): define subsets as the levels of a single grouping variable
- facet_grid(): define subsets as the crossing of two grouping variabls

This facilitates comparison among plots, not just of geoms within a plot.

What is the trend in housing prices in each state?

Hide

```
# Start by using a technique we already know: map State to color
p5 <- ggplot(housing, aes(x = Date, y = Home.Value))
p5 + geom_line(aes(color = State))
```



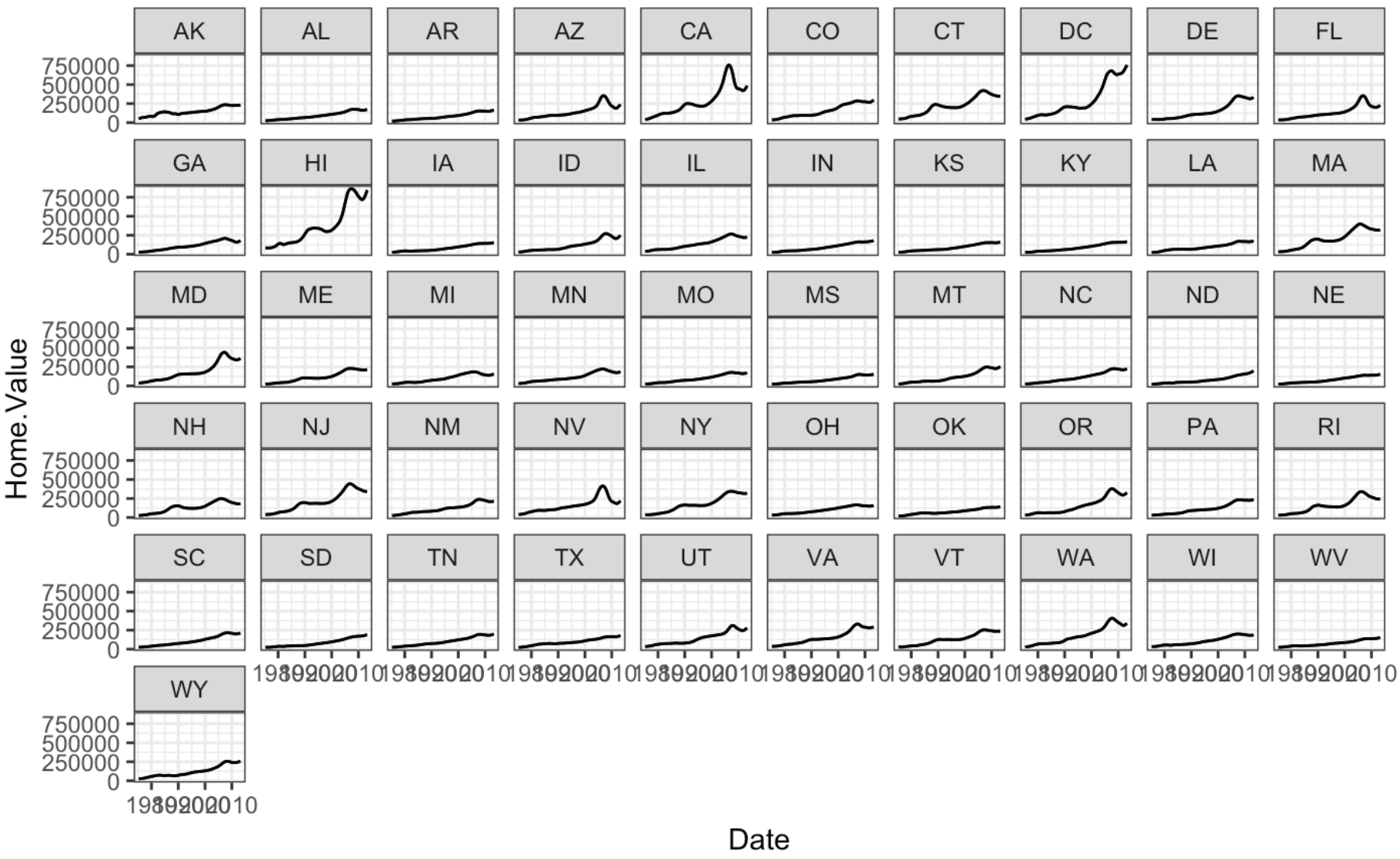
There are two problems here–there are too many states to distinguish each one by color, and the lines obscure one another.

Faceting to the rescue

We can remedy the deficiencies of the previous plot by faceting by state rather than mapping state to color.

Hide

```
p5 <- p5 + geom_line() + facet_wrap(~State, ncol=10)
p5
```



There is also a facet_grid() function for faceting in two dimensions.

Themes

The ggplot2 theme system handles non-data plot elements such as:

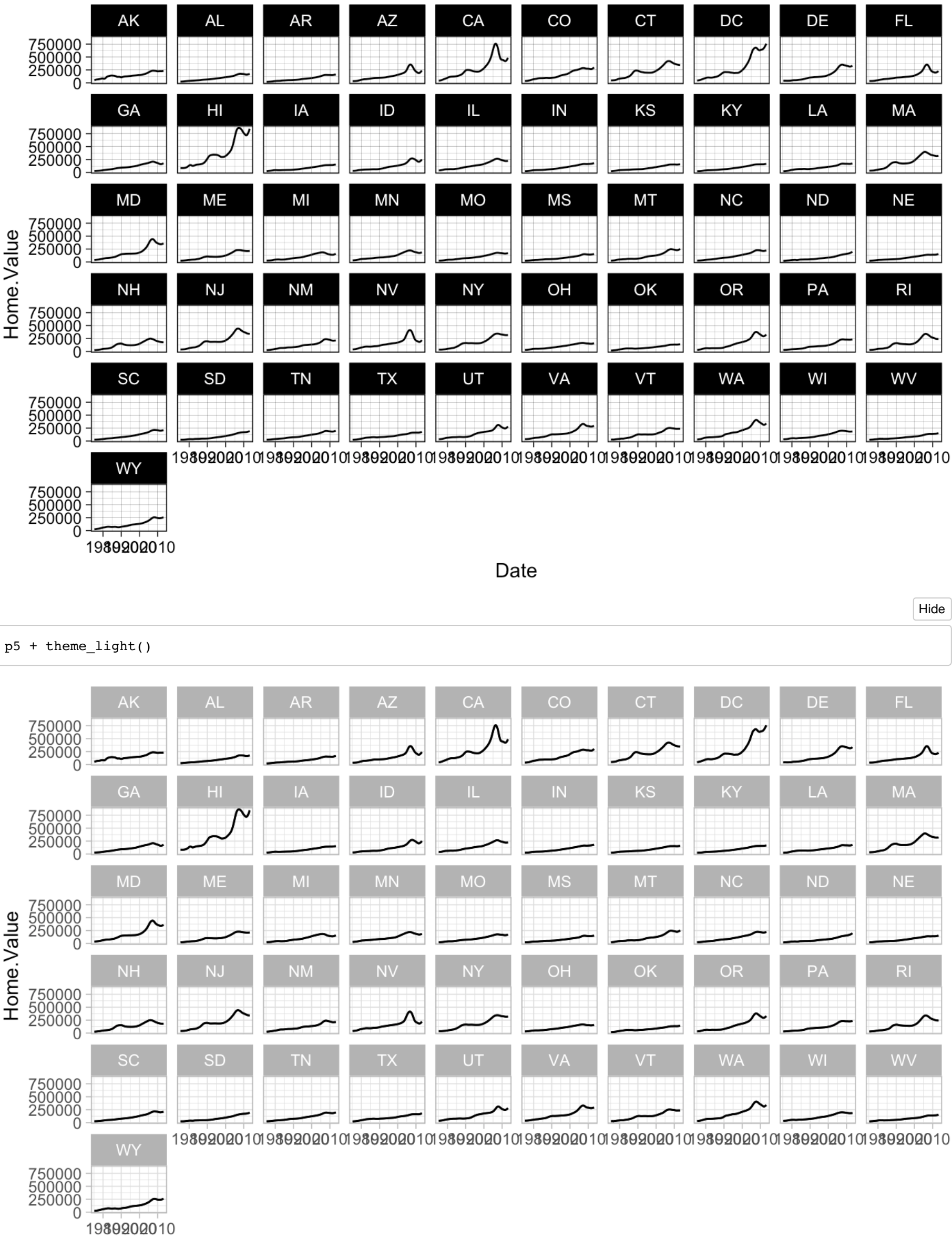
- Axis labels
- Plot background
- Facet label backround
- Legend appearance

Built-in themes include:

- theme_gray() (default)
- theme_bw()
- theme_classc()

Hide

```
p5 + theme_linedraw()
```



Hide

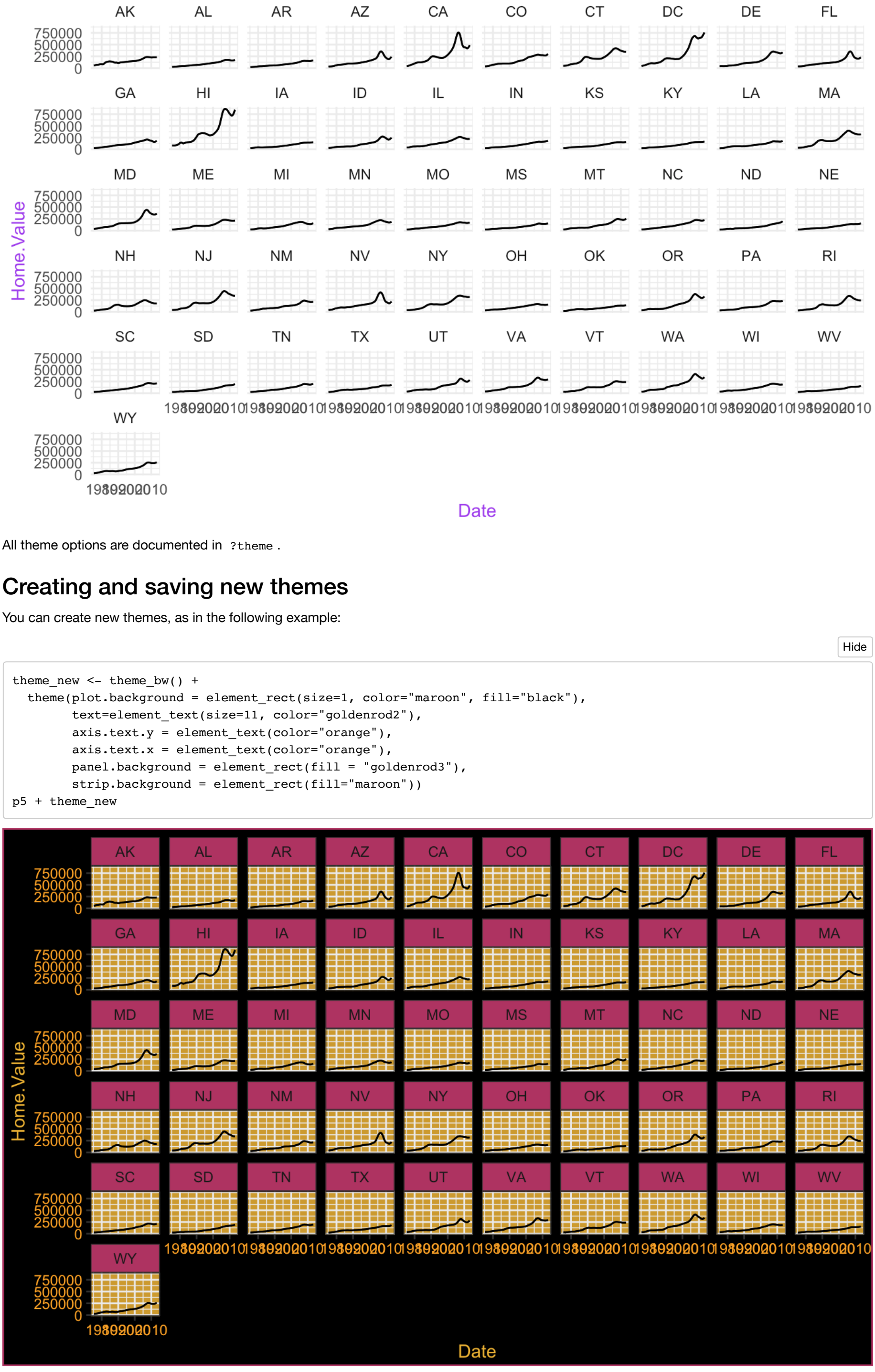
```
p5 + theme_light()
```


Overriding theme defaults

Specific theme elements can be overridden using `theme()`:

Hide

```
p5 + theme_minimal() + theme(text = element_text(color="purple"))
```



All theme options are documented in `?theme` .

Creating and saving new themes

You can create new themes, as in the following example:

Hide

```
theme_new <- theme_bw() +
  theme(plot.background = element_rect(size=1, color="maroon", fill="black"),
        text=element_text(size=11, color="goldenrod2"),
        axis.text.y = element_text(color="orange"),
        axis.text.x = element_text(color="orange"),
        panel.background = element_rect(fill = "goldenrod3"),
        strip.background = element_rect(fill="maroon"))
p5 + theme_new
```


Challenge: Recreate an Economist graph

Building off of the graphics you created in the previous exercises, put the finishing touches to make it as close as possible to the original economist graph (http://www.economist.com/node/21541178)

Hide

```
# Import the fonts installed on our system. Only do this once.
# library(extrafont)
# font_import()
# Register the font with the PDF output device
# loadfonts()
# Rename / Re-order the factor levels before the plot
dat$Region <- factor(dat$Region, levels = c("EU W. Europe", "Americas", "Asia Pacific", "East EU Cemt Asia", "MEN
A", "SSA"))
# List of countries with labels
countryList <- c("New Zealand", "Singapore", "Norway", "Japan","Germany", "Britain", "Barbados", "US", "France",
"Spain", "Botswana", "Cape Verde", "Bhutan", "Italy", "Greece", "Brazil", "Argentina", "China", "South Africa",
"Rwanda", "India", "Congo", "Russia", "Venezuela", "Iraq", "Myanmar", "Sudan", "Afghanistan")
# Develop our new theme
econ_theme <- theme_bw() +
  theme(aspect.ratio = 3/7,
        legend.text = element_text(size = 7, family="Arial Narrow"),
        legend.position = c(0,1),
        legend.justification = "left",
        legend.direction = "horizontal",
        legend.spacing = unit(2, "lines"),
        plot.margin = unit(c(0.5, 1, 0.5, 0.5), units="line"),
        axis.title.y=element_text(face="italic"),
        axis.title.x=element_text(face="italic"),
        plot.title = element_text(face = "bold", size=13),
        panel.border = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(size=.1, color="grey66"),
        panel.grid.minor.y = element_blank())
econ <- ggplot(data = dat, aes(x=CPI, y=HDI)) +
  geom_point(aes(color=Region), size=1.5, fill=4, shape=1, stroke = 1.2) +
  geom_smooth(method = "lm", formula = y ~ log(x), se=FALSE, color="red", linetype=1, weight=2) +
  geom_text_repel(aes(label=Country), data = dat[dat$Country %in% countryList,], size=2.2) +
  scale_x_continuous(name="Corruption Perceptions Index, 2011(10=least corrupt)", breaks=seq(1,10,by=1),
limits=c(1,10)) +
  scale_y_continuous(name="Human Development Index, 2011(1=best)", breaks=seq(0, 1.0, by=0.1), limits=
c(0.2,1.0)) +
  scale_color_manual(name = "", values = c("#2d6074", "#29adde", "#bbe9fb", "#188c81", "#f2523a", "#7c2510"), lab
els=c("EU W. Europe"="OECD", "Americas" = "Americas", "Asia Pacific"="Asia & Oceania", "East EU Cemt Asia"="Centr
al & Eastern Europe", "MENA"="Middle East & north Africa", "SSA"="Sub-Saharan Africa")) +
  labs(title="Corruption and human development") +
  guides(color=guide_legend(nrow=1)) +
  econ_theme
econ
```