STU SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

# Graphics. ggplot2

Dr.Sc. Oleksii Yehorchenkov

Department of Spatial Planning

Funded by the European Union NextGenerationEU

# What is ggplot2

`ggplot2` is one of core packages of `tidyverse`. It is one of the most elegant and most versatile system for making graphs in R. `ggplot2` implements the grammar of graphics, a coherent system for describing and building graphs.

# Loading data

```
1 library(tidyverse)
2 library(ggrepel) # repel overlapping text labels
```

# Example data: housing prices

```r
housing <- read_csv("./data/landdata-states.csv")

# create a subset for 1st quarter 2001
hp2001Q1 <- housing |>
  filter(Date == 2001.25)

head(housing[1:5]) # view first 5 columns
```

```
# A tibble: 6 × 5
  State region  Date Home_Value Structure_Cost
  <chr> <chr>  <dbl>      <dbl>          <dbl>
1 AK    West   2010.     224952         160599
2 AK    West   2010.     225511         160252
3 AK    West   2010.     225820         163791
4 AK    West   2010      224994         161787
5 AK    West   2008      234590         155400
6 AK    West   2008.     233714         157458
```

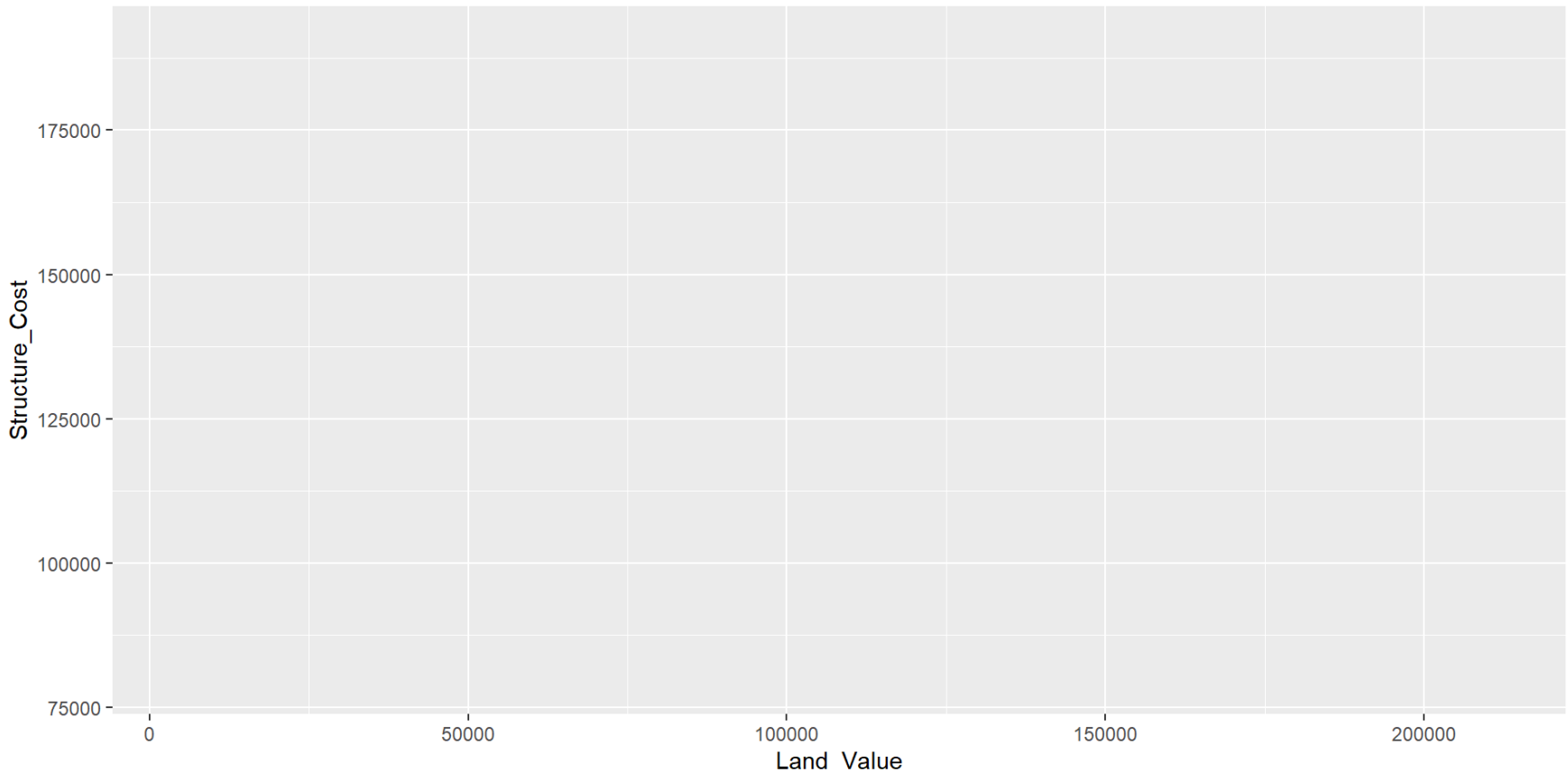
# Step 1: create a blank canvas by specifying data:

```
1  ggplot(data = hp2001Q1)
```

# Step 2: specify aesthetic mappings

now we want to map variables to visual aspects: here we map “Land_Value” and “Structure_Cost” to the x- and y-axes.

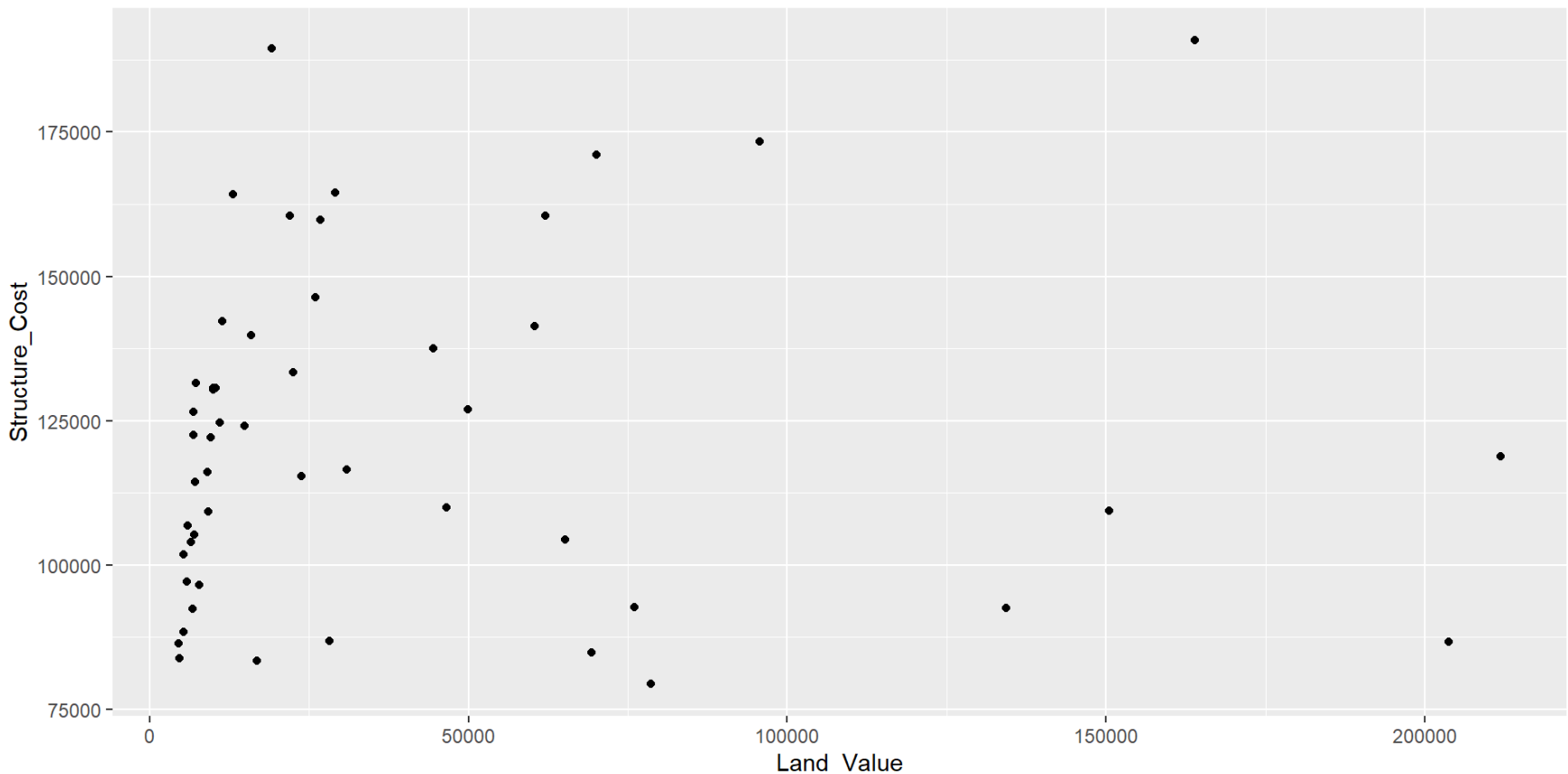
```
ggplot(data = hp2001Q1,
       mapping = aes(x = Land_Value, y = Structure_Cost))
```



# Step 3: add geometric objects:

here we use `geom_point()` to add a layer with point (dot) elements as the geometric shapes to represent the data.

```
1  ggplot(data = hp2001Q1, aes(x = Land_Value, y = Structure_Cost)) +
2    geom_point()
```



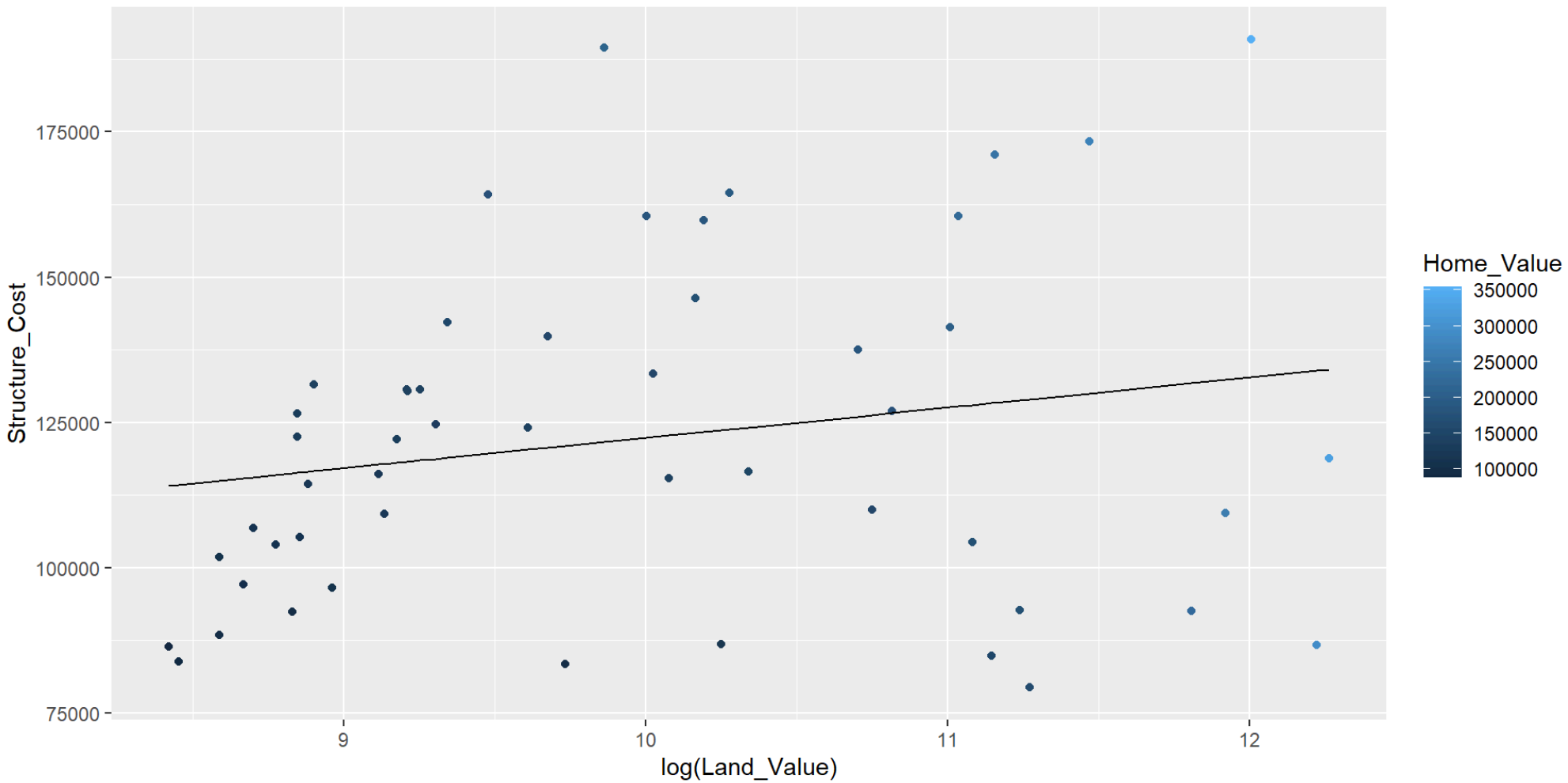
# Lines (prediction line)

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 — this is referred to as **aesthetic inheritance**. Our plot could use a regression line:

```
hp2001Q1$pred_SC <- lm(Structure_Cost ~ log(Land_Value),
                       data = hp2001Q1) |>
    predict()
```

# Lines (prediction line)

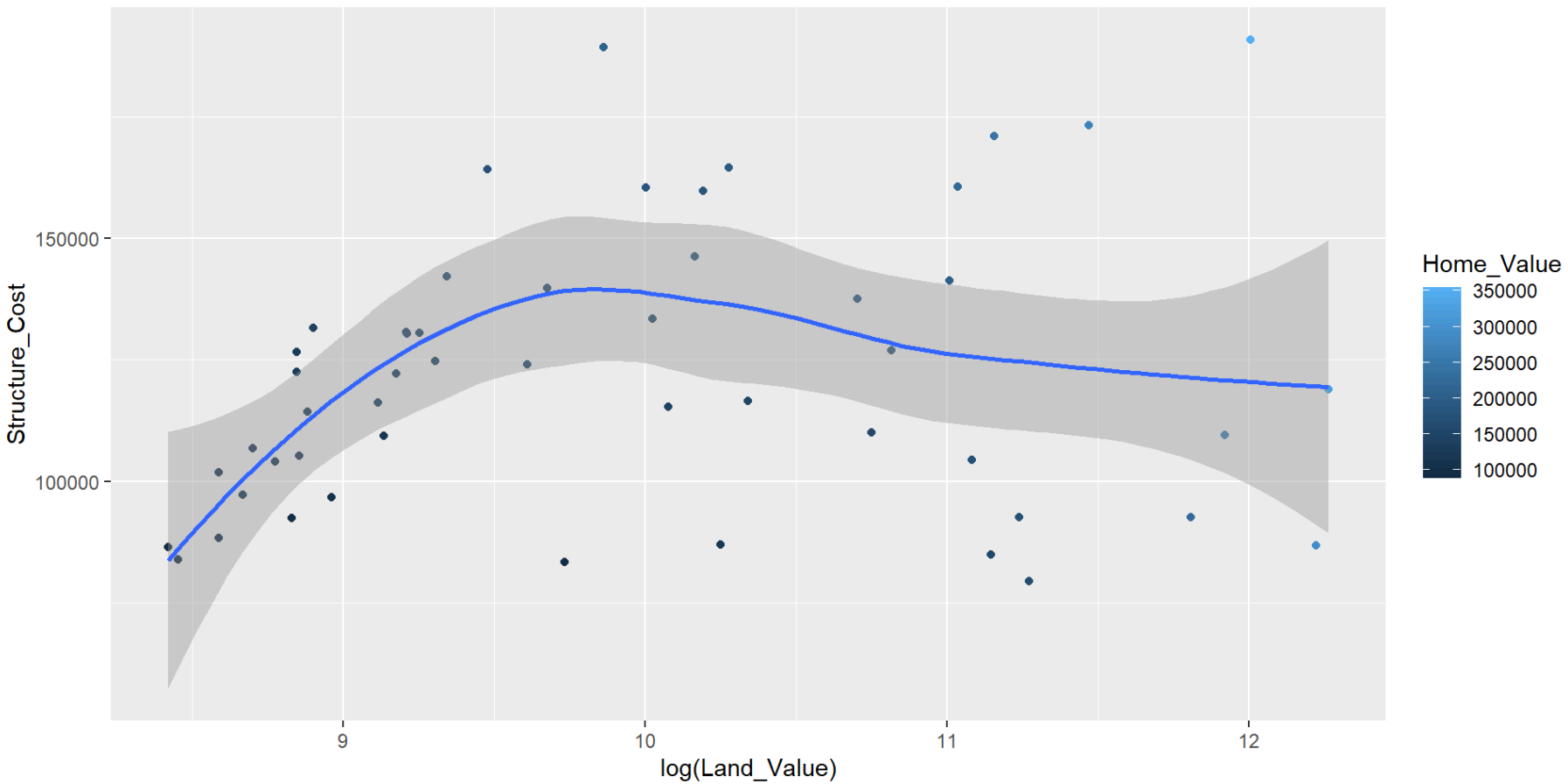
```
p1 <- ggplot(hp2001Q1, aes(x = log(Land_Value), y = Structure_Cost))
p1 + geom_point(aes(color = Home_Value)) +
     geom_line(aes(y = pred_SC))
```



# Smoothers

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

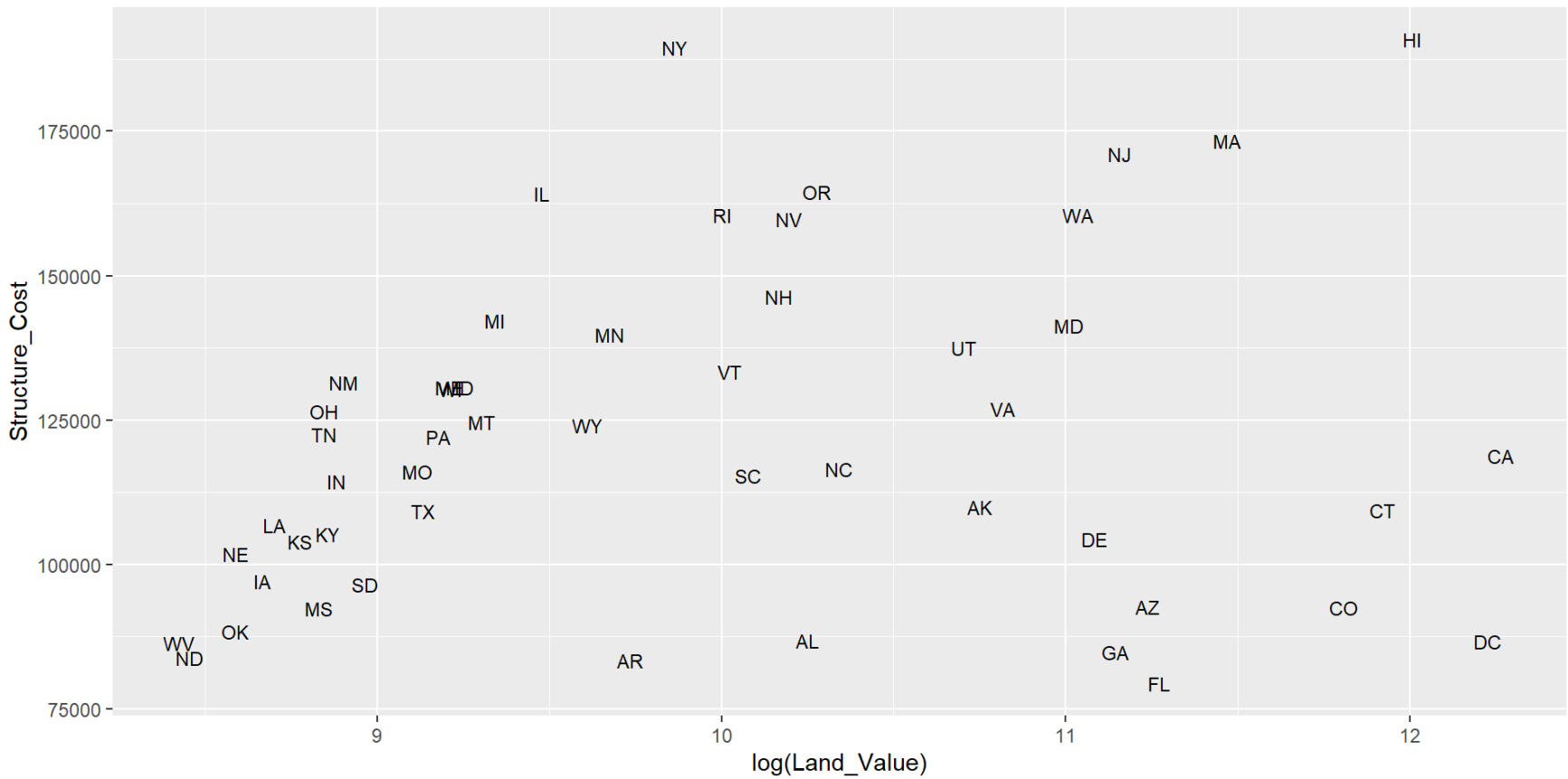
```
p1 +
    geom_point(aes(color = Home_Value)) +
    geom_smooth()
```



# Text (label points)

Each geom accepts a particular set of mappings; for example `geom_text()` accepts a `label` mapping.

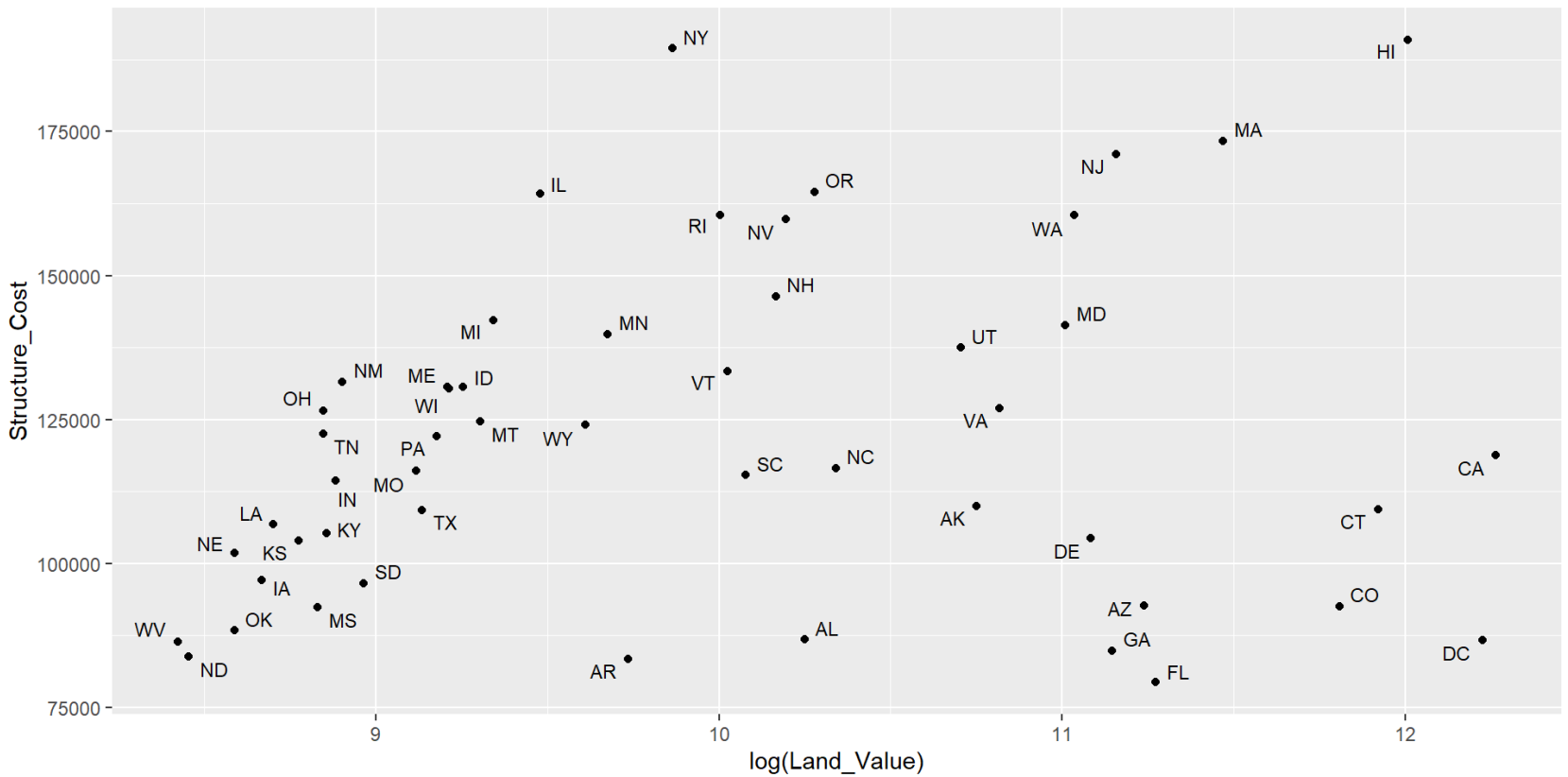
```
1  p1 +
2      geom_text(aes(label = State), size = 3)
```



# Text (label points)

But what if we want to include points and labels? We can use `geom_text_repel()` to keep labels from overlapping the points and each other.

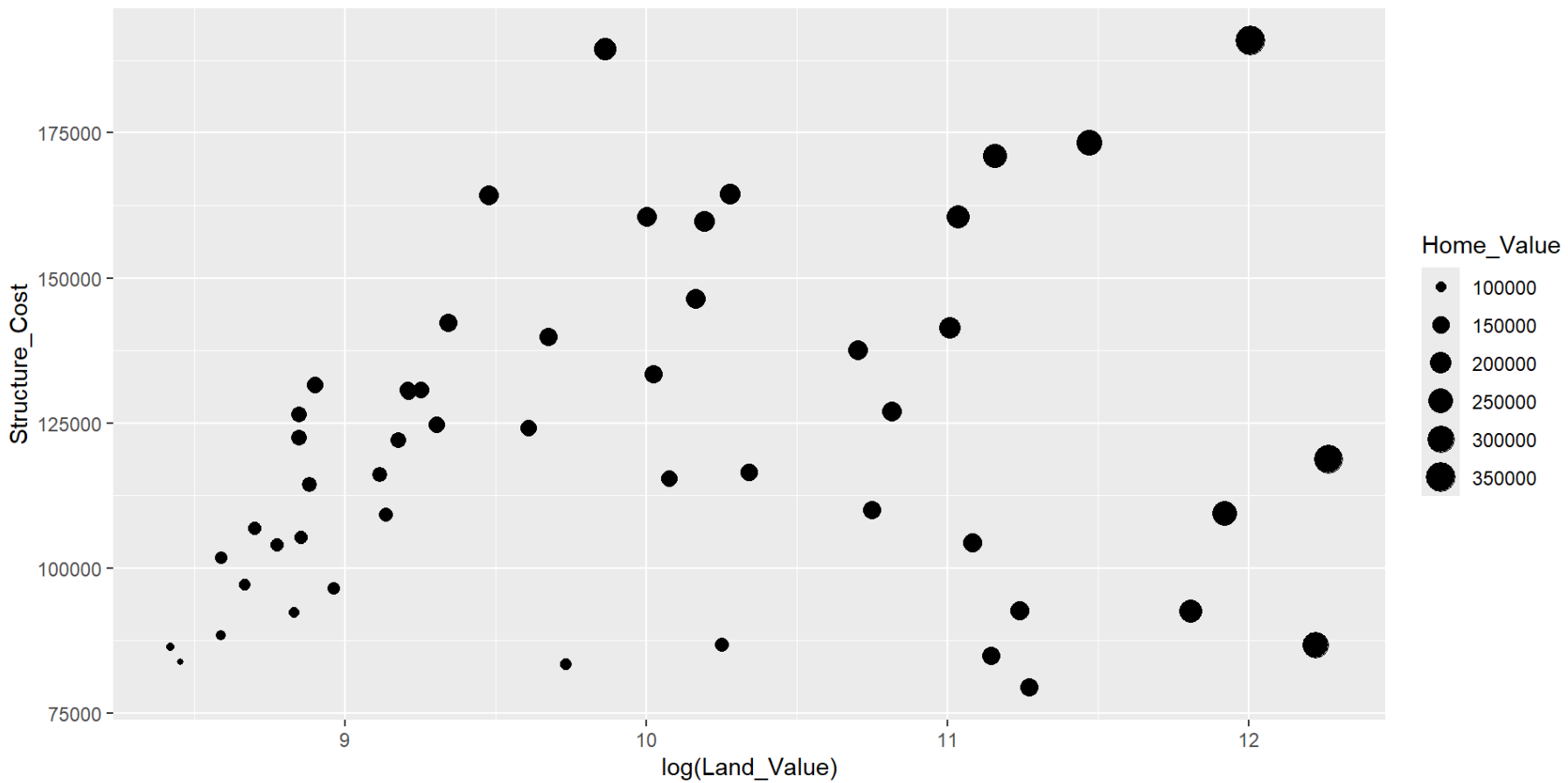
```
p1 +
    geom_point() +
    geom_text_repel(aes(label = State), size = 3)
```



# Aesthetic mapping VS assignment

1. Variables are **mapped** to aesthetics inside the `aes()` function.

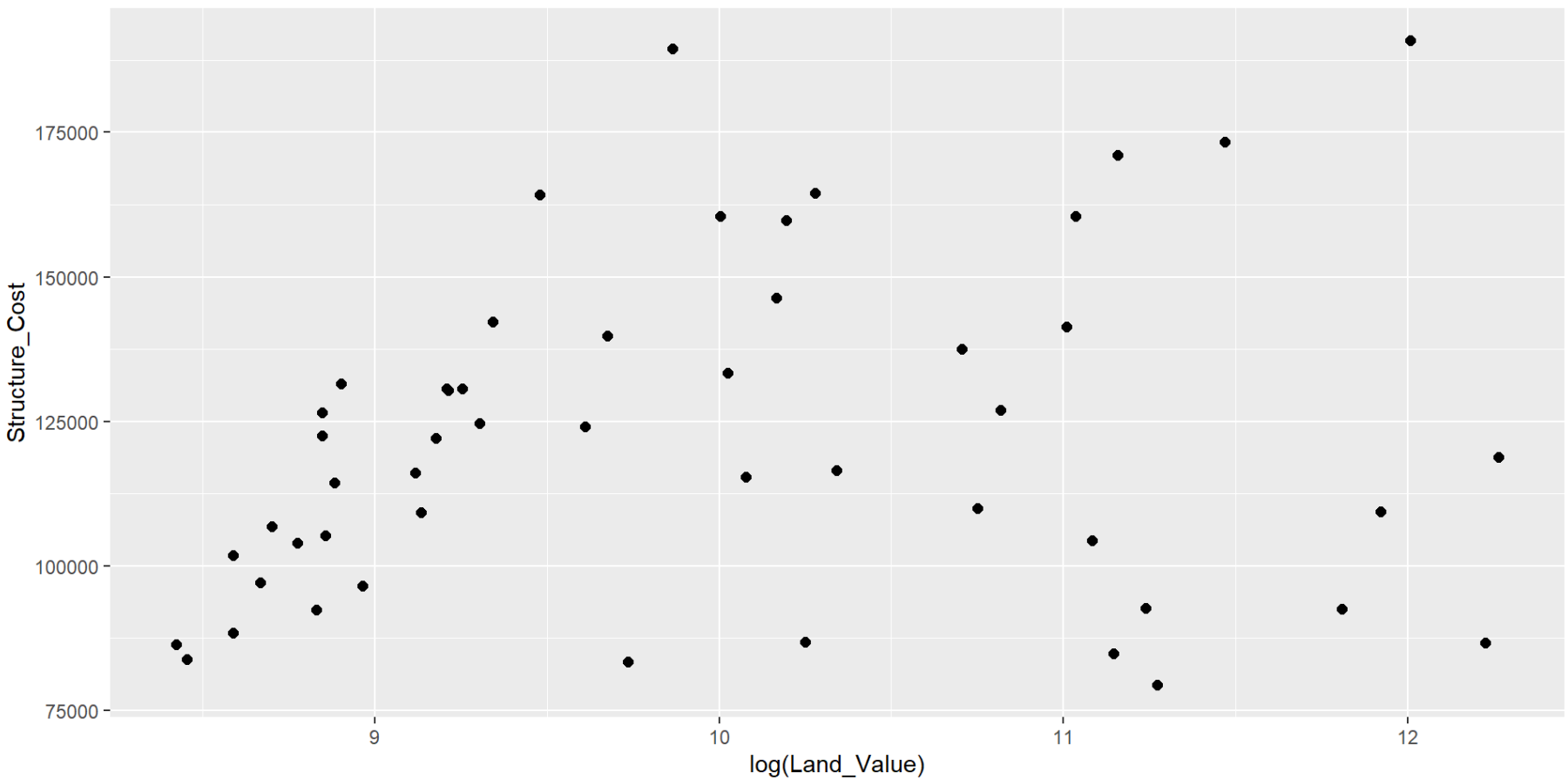
```
p1 +
    geom_point(aes(size = Home_Value))
```



# Aesthetic mapping VS assignment

2. Constants are **assigned** to aesthetics outside the `aes()` call

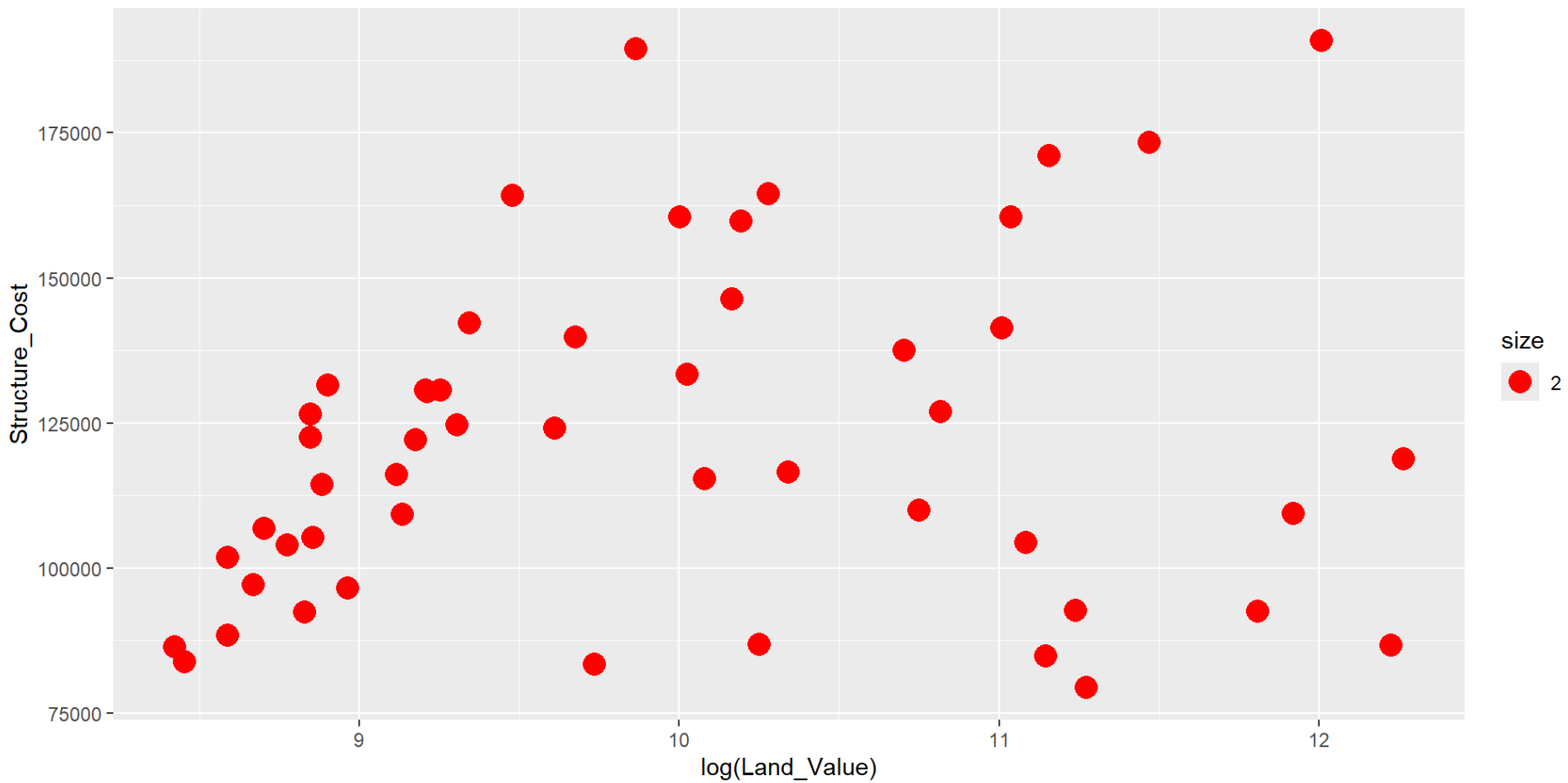
```
1  p1 +
2      geom_point(size = 2)
```



# Aesthetic mapping VS assignment

This sometimes leads to confusion, as in this example:

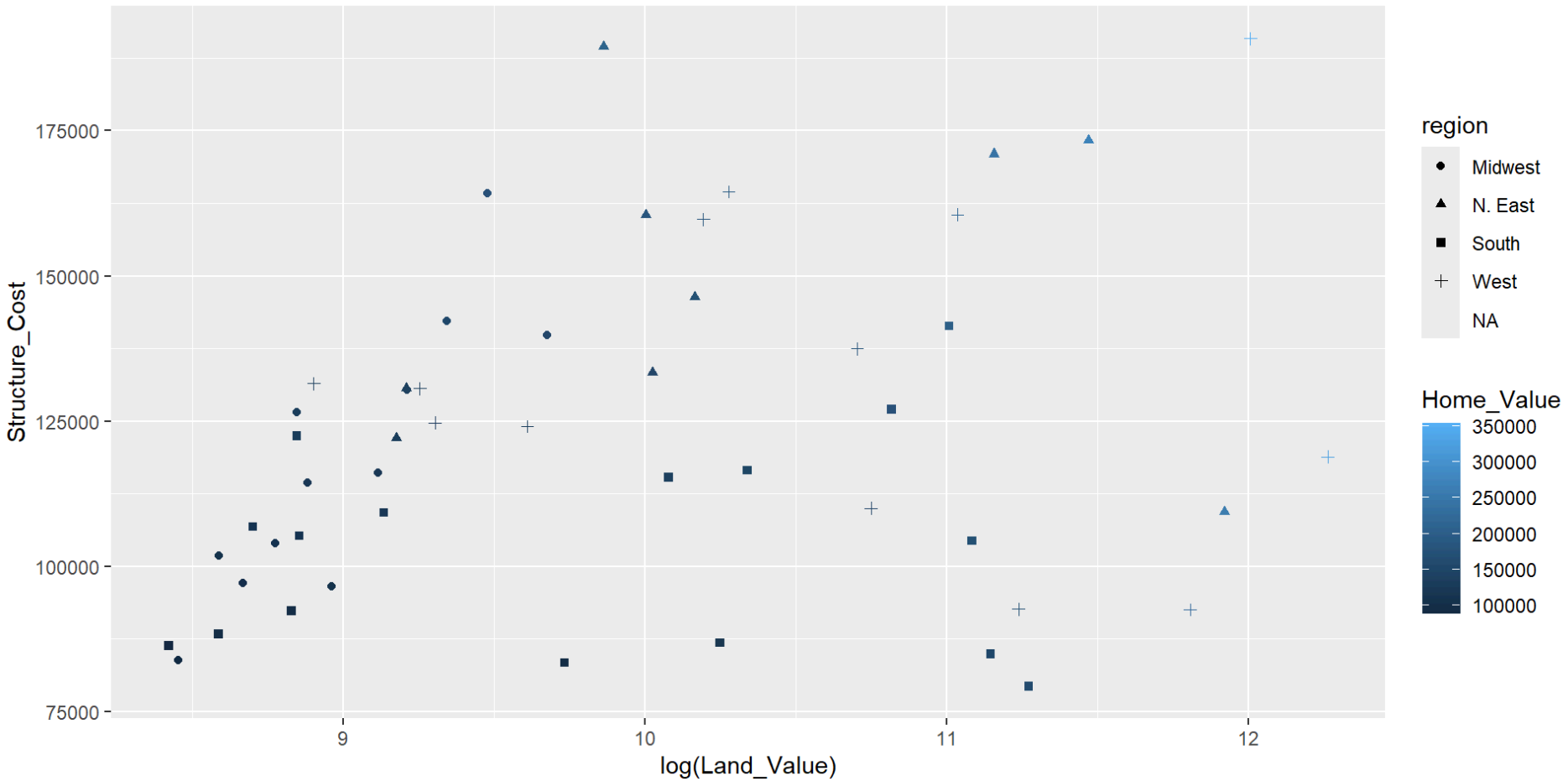
```
p1 +
    geom_point(aes(size = 2),# incorrect! 2 is not a variable
               color="red") # this is fine -- all points red
```



# Mapping variables to other aesthetics

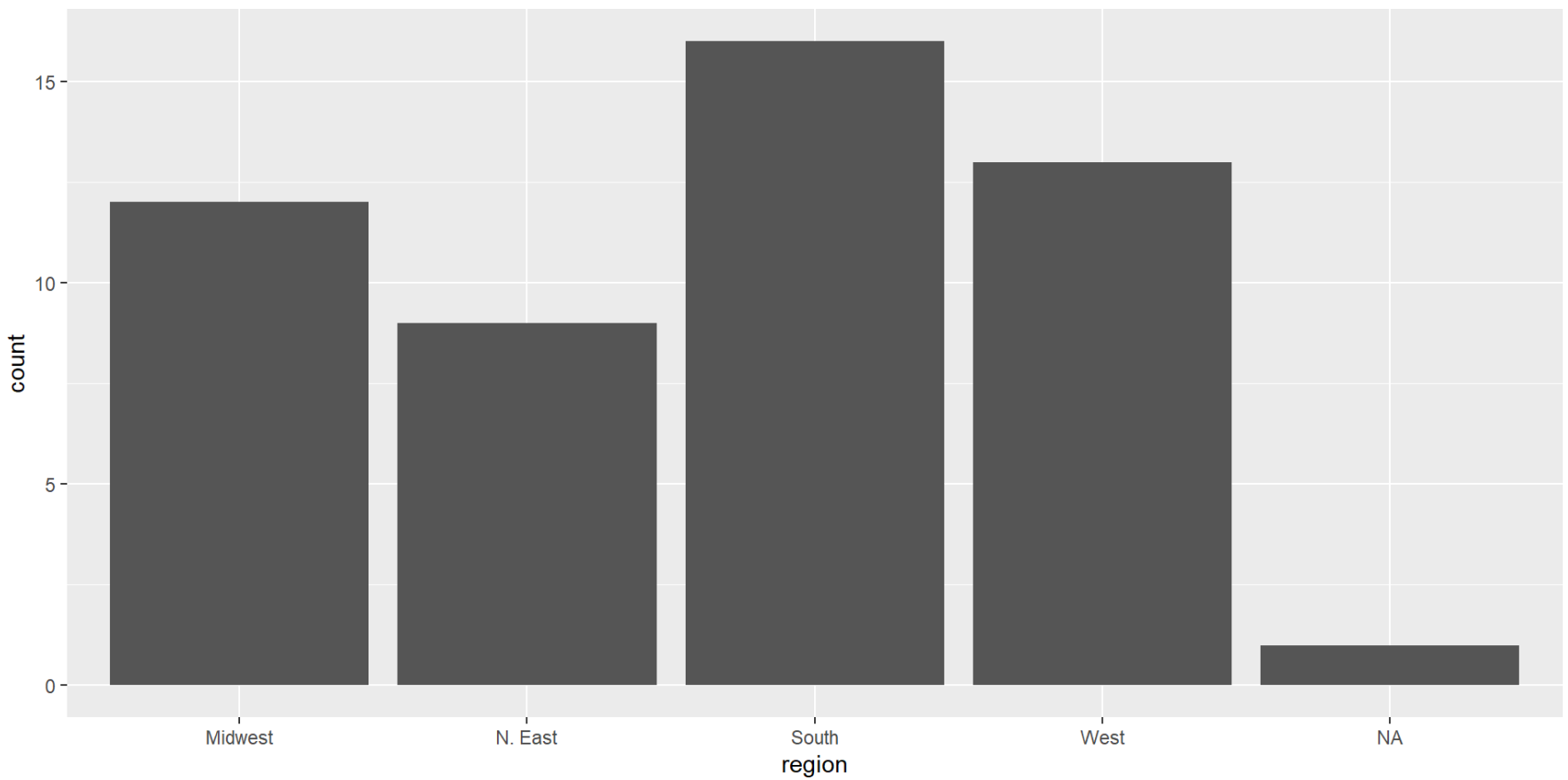
Other aesthetics are mapped in the same way as x and y in the previous example.

```
p1 +
    geom_point(aes(color = Home_Value, shape = region))
```



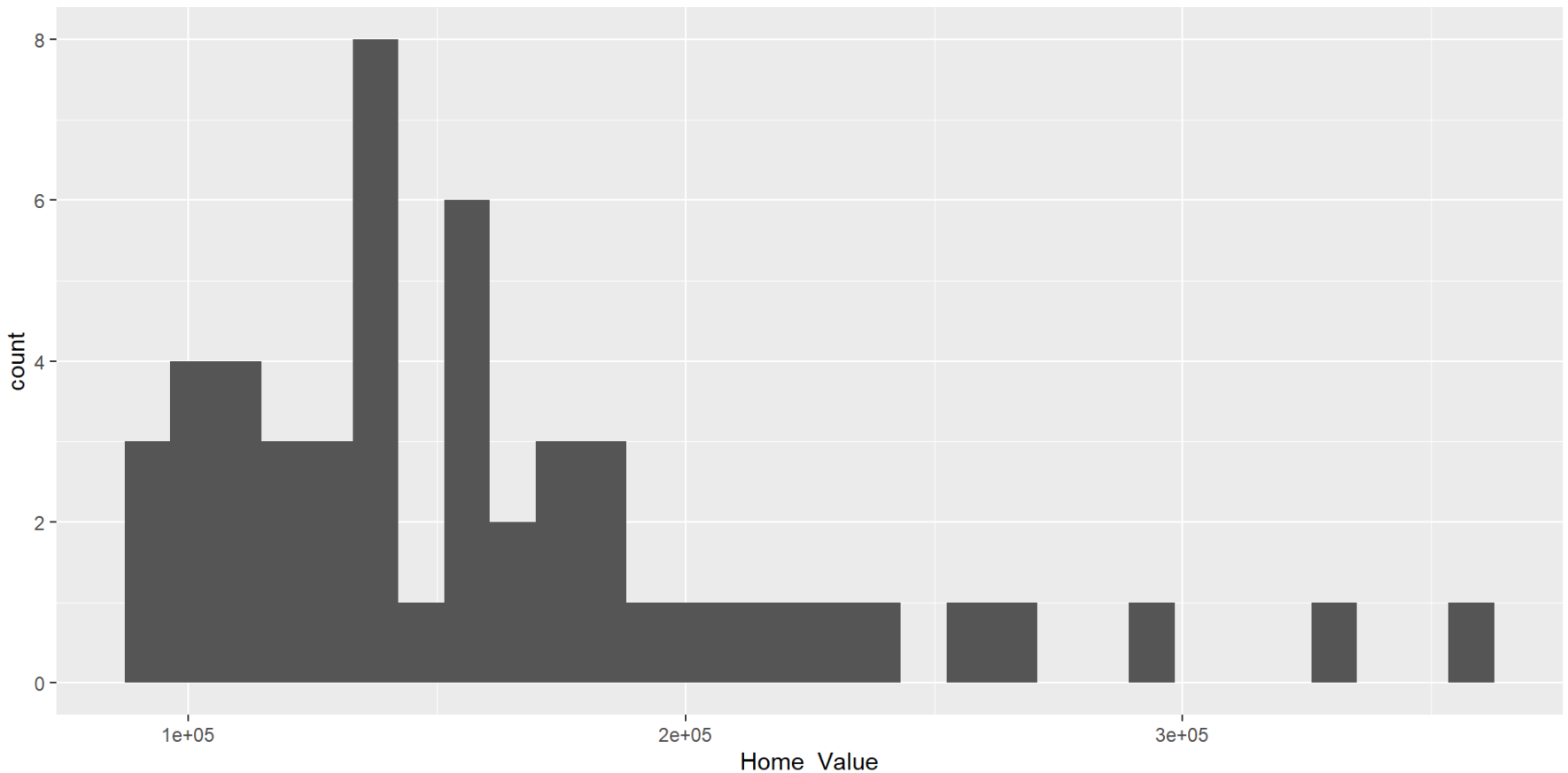
# Visualizing distributions. Categorical variable

```
1  ggplot(hp2001Q1, aes(x = region)) +
2    geom_bar()
```



# Visualizing distributions. Numerical variable

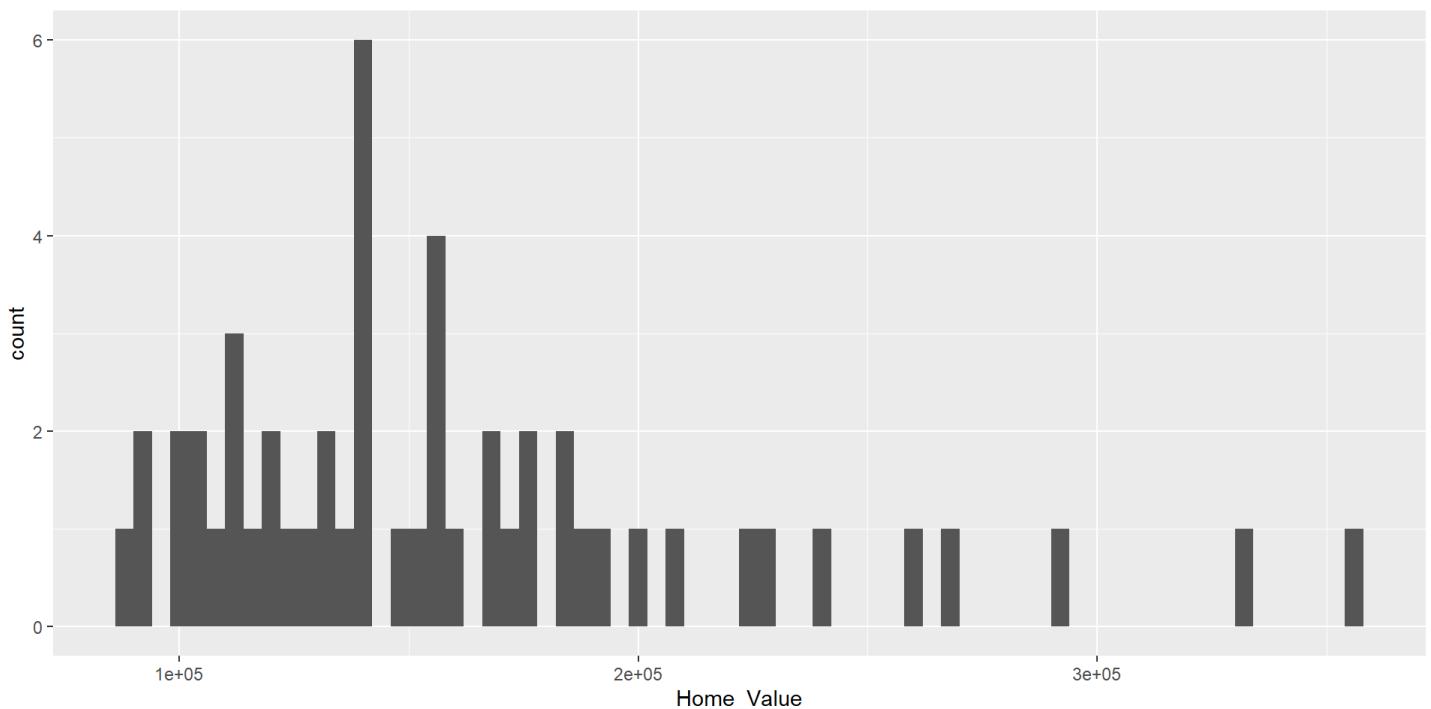
```
ggplot(hp2001Q1, aes(x = Home_Value)) +
  geom_histogram()
```



# Visualizing distributions. Numerical variable

We can change the binning scheme by passing the `binwidth` argument to the `geom_histogram` function

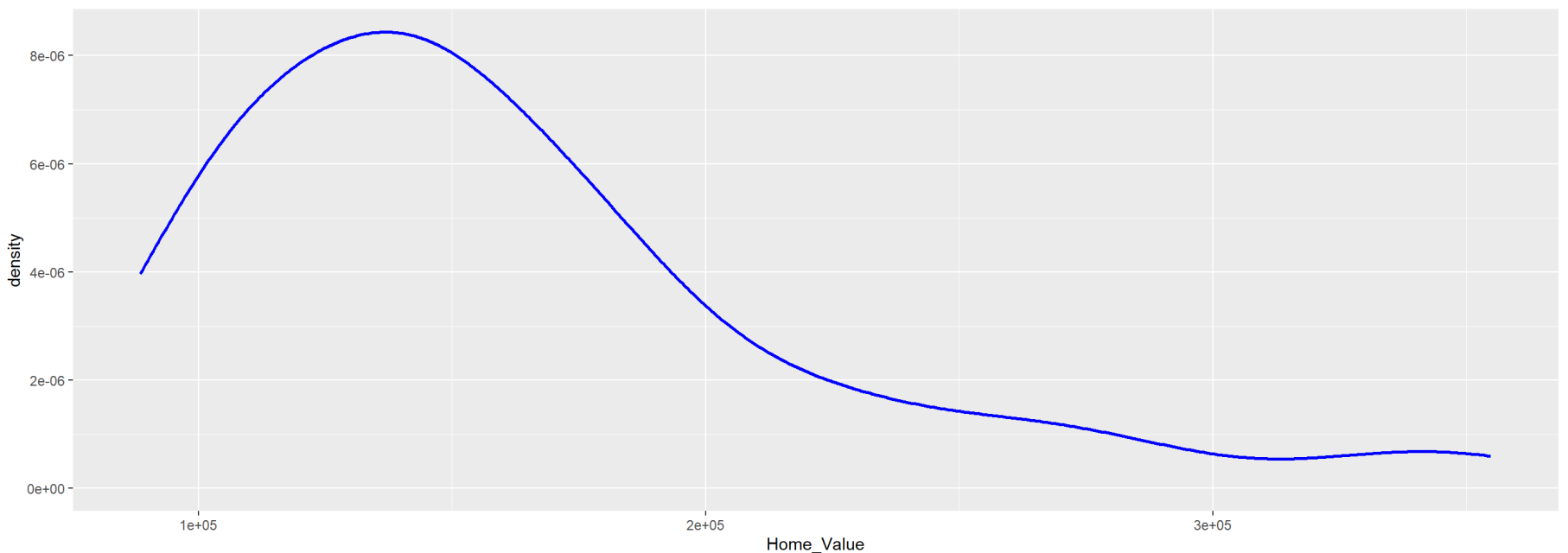
```
1 ggplot(hp2001Q1, aes(x = Home_Value)) +
2   geom_histogram(binwidth = 4000)
```



# Visualizing distributions. Numerical variable

An alternative visualization for distributions of numerical variables is a density plot. A density plot is a smoothed-out version of a histogram and a practical alternative, particularly for continuous data that comes from an underlying smooth distribution.

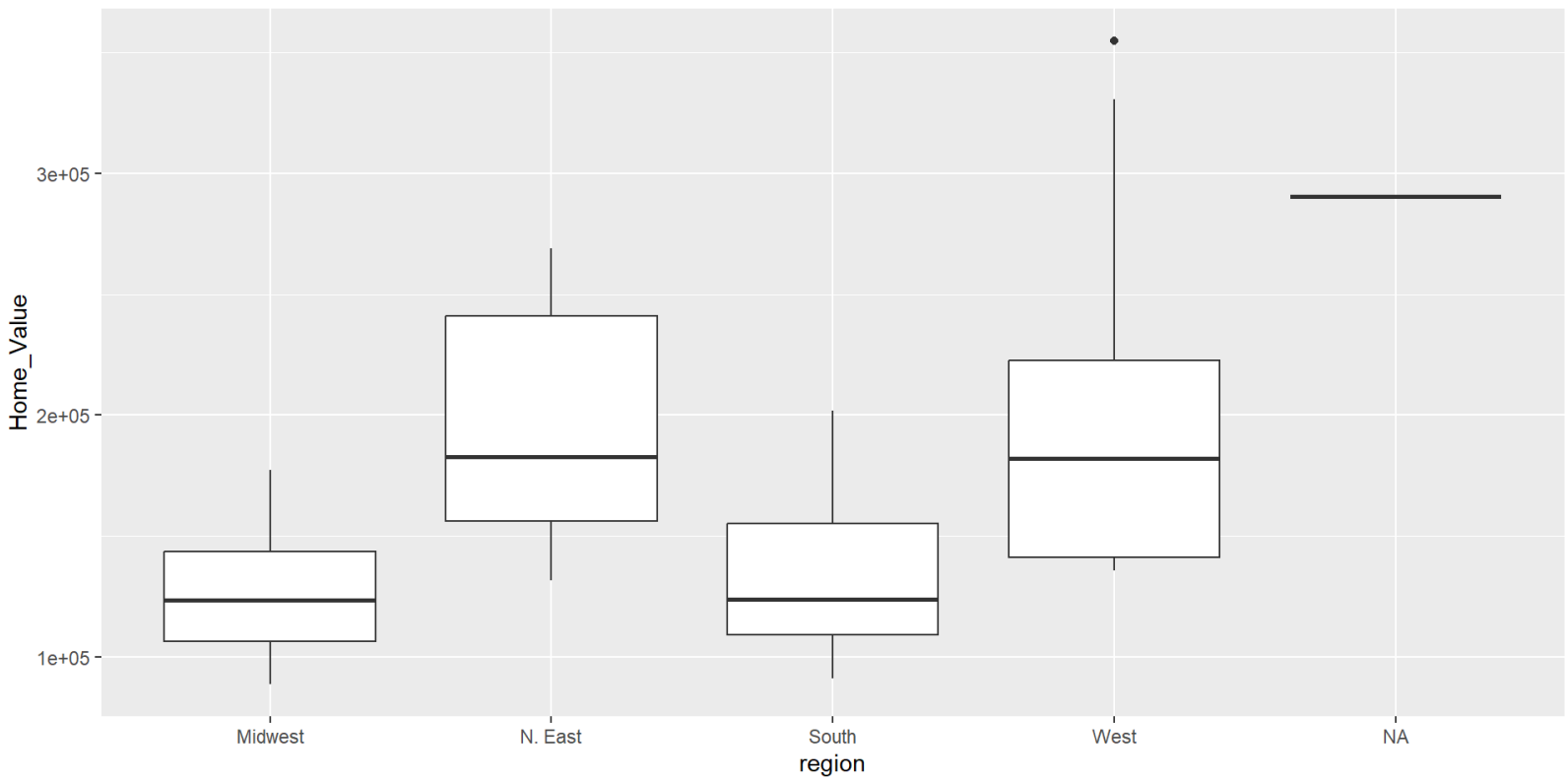
```
1  ggplot(hp2001Q1, aes(x = Home_Value)) +
2    geom_density(linewidth = 1, color="blue")
```



# Visualizing relationships

To visualize the relationship between a numerical and a categorical variable we can use side-by-side box plots. A boxplot is a type of visual shorthand for measures of position (percentiles) that describe a distribution.

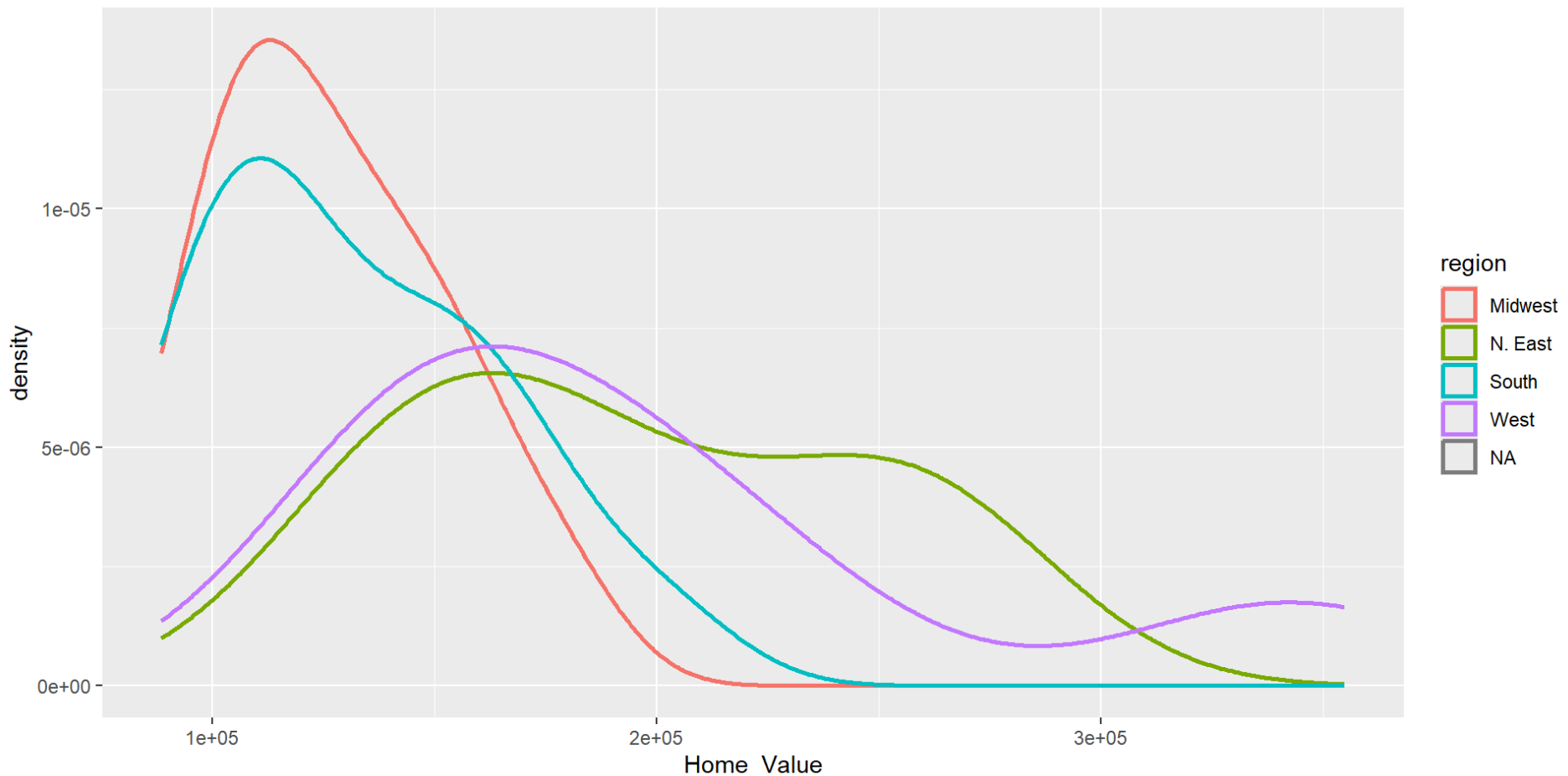
```
1  ggplot(hp2001Q1, aes(x = region, y = Home_Value)) +
2    geom_boxplot()
```



# Visualizing relationships

Alternatively, we can make density plots with `geom_density()`.

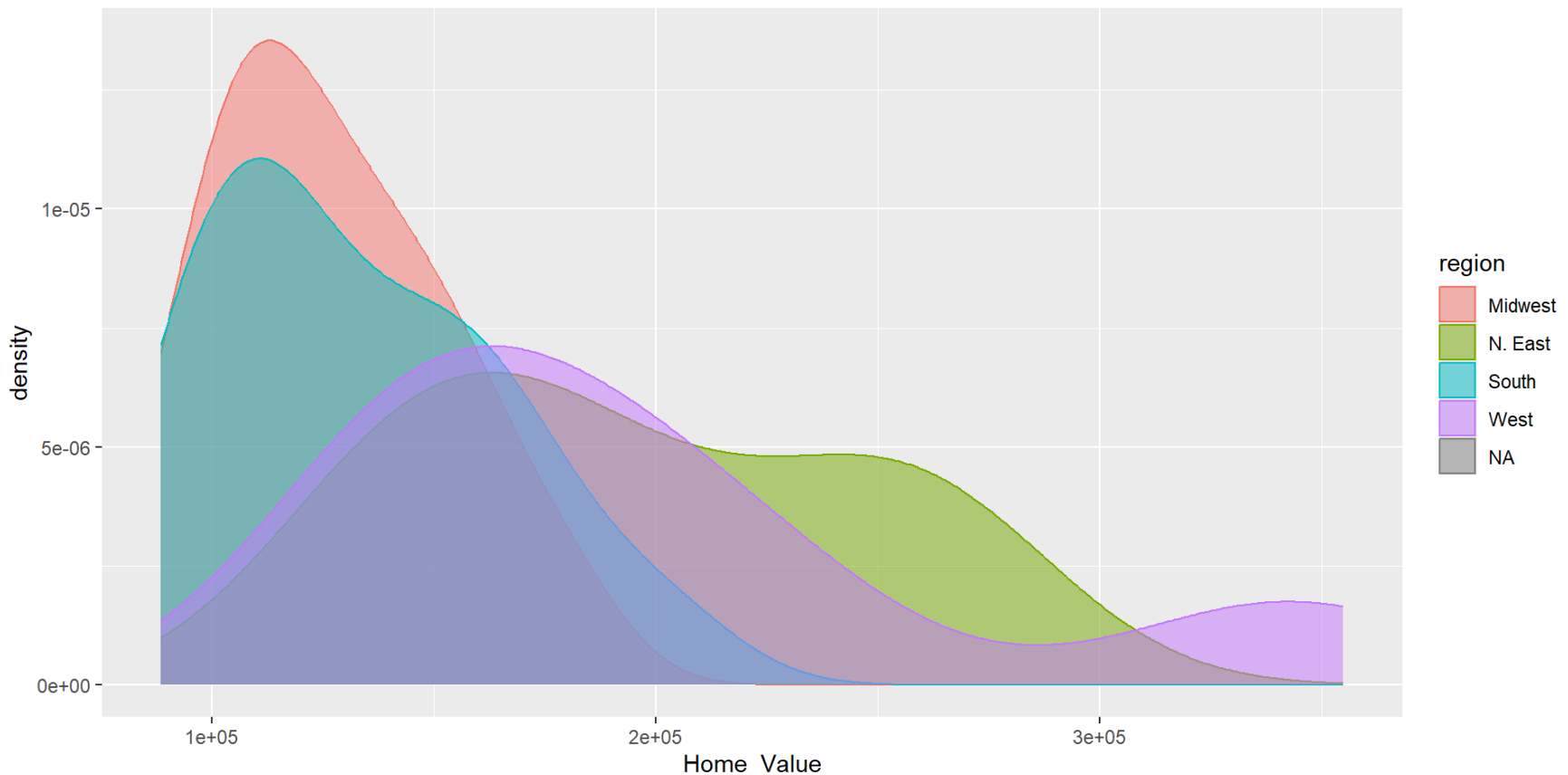
```
1  ggplot(hp2001Q1, aes(x = Home_Value, color = region)) +
2    geom_density(linewidth = 1)
```



# Visualizing relationships

Additionally, we can map species to both `color` and `fill` aesthetics and use the `alpha` aesthetic to add transparency to the filled density curves.

```
1  ggplot(hp2001Q1, aes(x = Home_Value, color = region, fill = region)) +
2    geom_density(alpha = 0.5)
```



# Saving your plots

Once you've made a plot, you might want to get it out of R by saving it as an image that you can use elsewhere. That's the job of `ggsave()`, which will save the plot most recently created to disk:

```
ggplot(hp2001Q1, aes(x = Home_Value, color = region, fill = region)) +
  geom_density(alpha = 0.5)

ggsave(filename = "houses_plot.png")
```

# References

R for Data Science 2e, Hadley Wickham, Mine Cetinkaya-Rundel, Garrett Grolemund