Introduction to ggplot2

# Getting set up

Before continuing with this document

* Open RStudio on your computer OR use the Virtual Desktop **BCOB\_BCC app**
* Watch the adding packages video in D2L, and add the package called ggplot2
* Download the data set called HOUSEDATA, which is available in the Data Sets folder in Top Hat (you likely already have downloaded that file, as it was used in the first reading assignment).

# Explanation of variables in the HOUSEDATA data set

The HOUSEDATA data set consists of information regarding home sales in a specific city for one year. The commands we will use today are:

* PRICE: the selling price of a house, in thousand dollars
* SIZE: the size of the house, measured in square feet
* AGE: the age of the house, in years
* LOCATION: North or South
* LOT: corner lot or non-corner lot
* AGENT: selling agent of the house (values are 1, 2, 3, 4, and this is a categorical variable)

# Graphical Methods

Linear regression is one of the most commonly used methods in statistical analysis – let’s make some graphs associated with Linear Regression!

In order to be able to complete the following questions, it is necessary that all of the following are true:

* You have installed ggplot2 to RStudio
* You have successfully compiled the library(ggplot2) command
* You have imported the HOUSEDATA data set into RStudio

# **Data visualization using ggplot2**

Building a plot in ggplot2 is like painting a picture. We begin with a canvas, and then add layers of additional information.

These layers of additional information are created geometric functions (such as bar charts, box plots, scatter plots, line graphs, etc), and aesthetic mappings indicate exactly what should be graphed – what variables are used and how.

ggplot(data = <*name of data set*>) +

<geom\_*type of geometric function*>(mapping = aes(<*how to use variables*>))

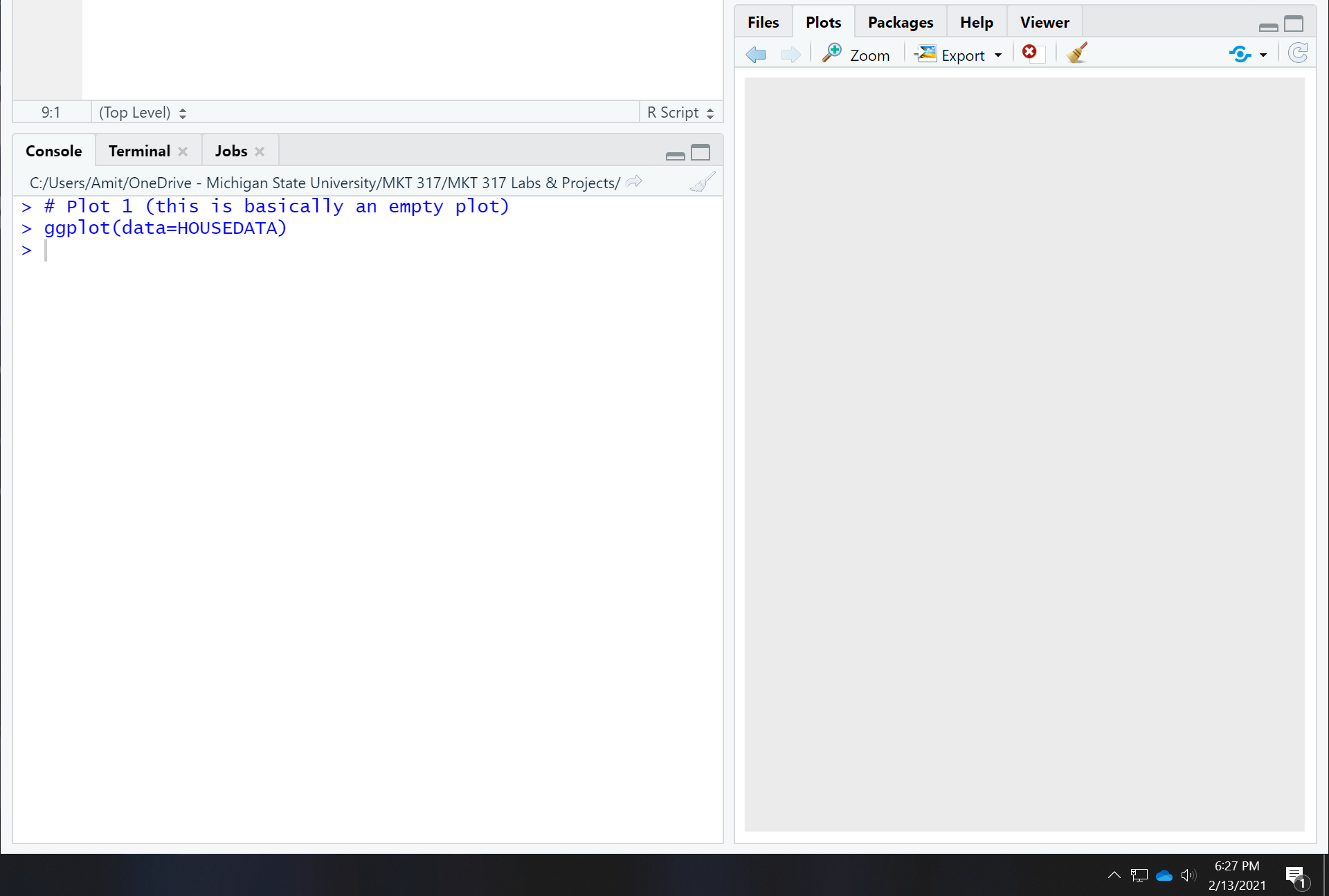
## **Getting started and scatter plots**

**Plot 1**

Let’s begin with our canvas.

ggplot(data=HOUSEDATA)

Screenshot of what you see in the “plots” window (hint: you will probably think you made an error – this is not an interesting plot).



**Plot 2**

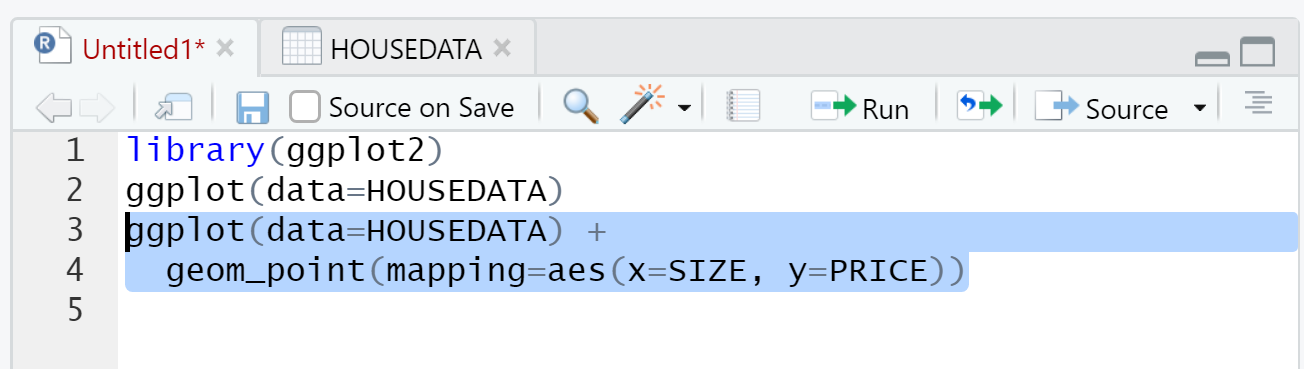
After creating our canvas, let’s add a scatter plot (this is a “point” geometric object).

The aesthetics (aes) mapping will indicate which variables are used and how the variables are used – we will use the variable SIZE on the x-axis, and the variable PRICE on the y-axis.

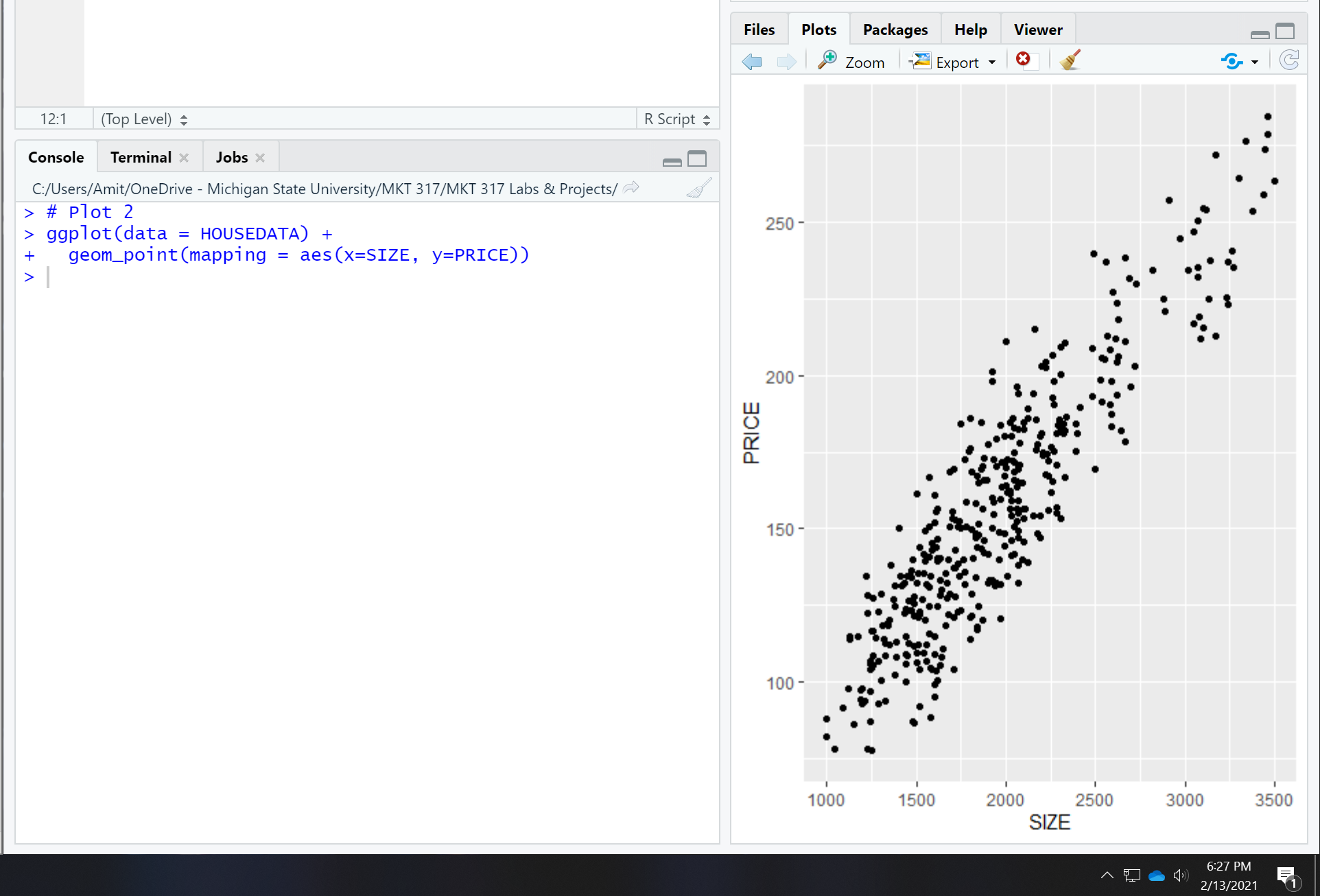
ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE))

It is important that either both pieces of this command are on the same line or that the plus sign is at the end of the first line of code. Be sure to compile the entire command!



Screenshot of plot (it should be a scatter plot):



**Plot 3**

Let’s look at ways that we can add more information!

**Color in a scatter plot**

We can add/modify color in several ways:

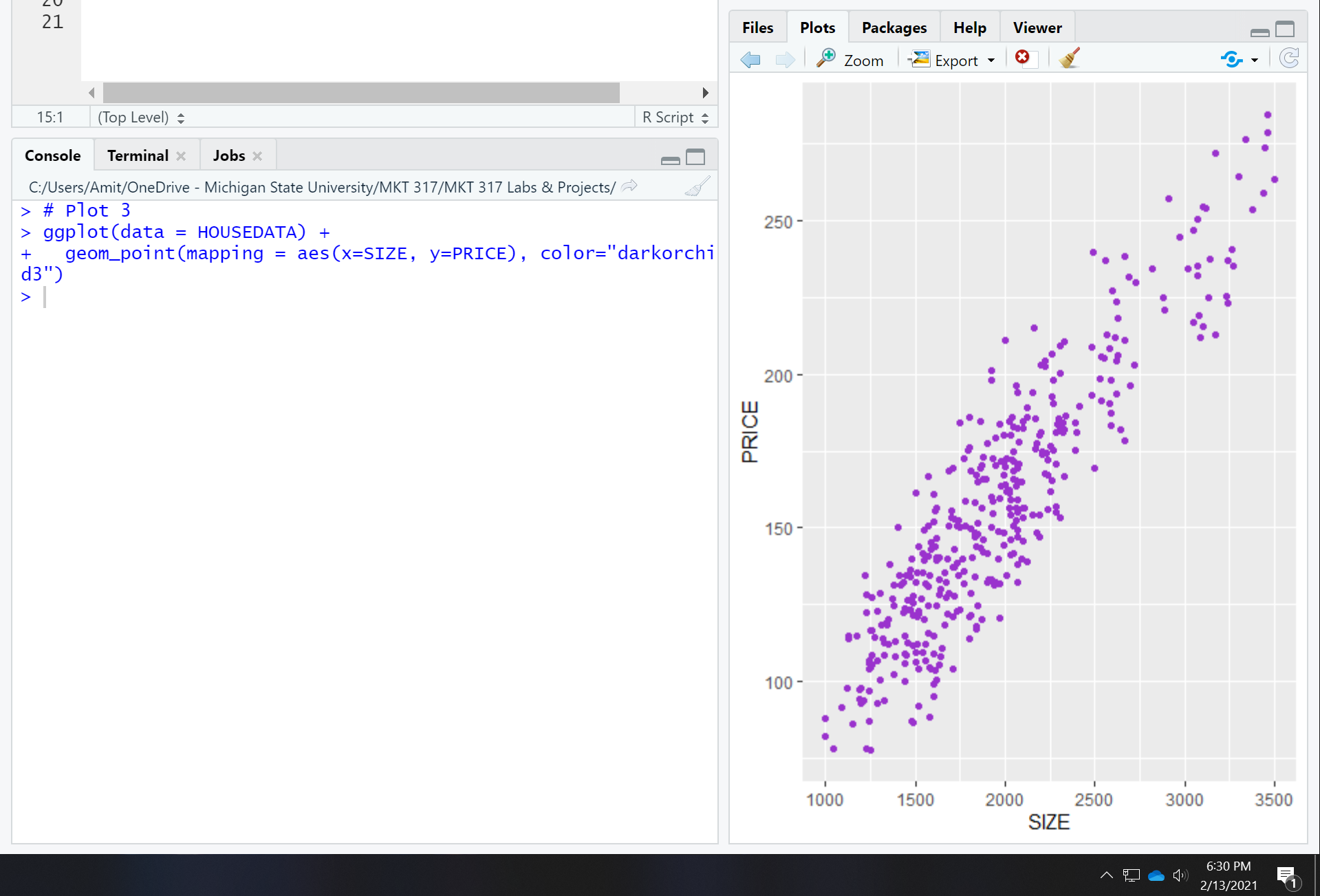
* We can make all dots the same, custom, color.
* We can color-code the dots by a variable and use default colors.
* We can color-code the dots by a variable and use custom colors.

To change the color of all the dots, the color command should not be included as an aesthetic mapping – this is because we are not color-coding based on the values of a variable

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE), color="darkorchid3")

Paste a screenshot of your plot below:



**Plot 4**

Use the site <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

Pick a color that you like. Create a scatter plot where the x-axis is SIZE, the y-axis is PRICE, and the dots are your favorite color (other than darkorchid3)

R code that you used: (in screenshot)

Screenshot of your plot:



**Plot 5**

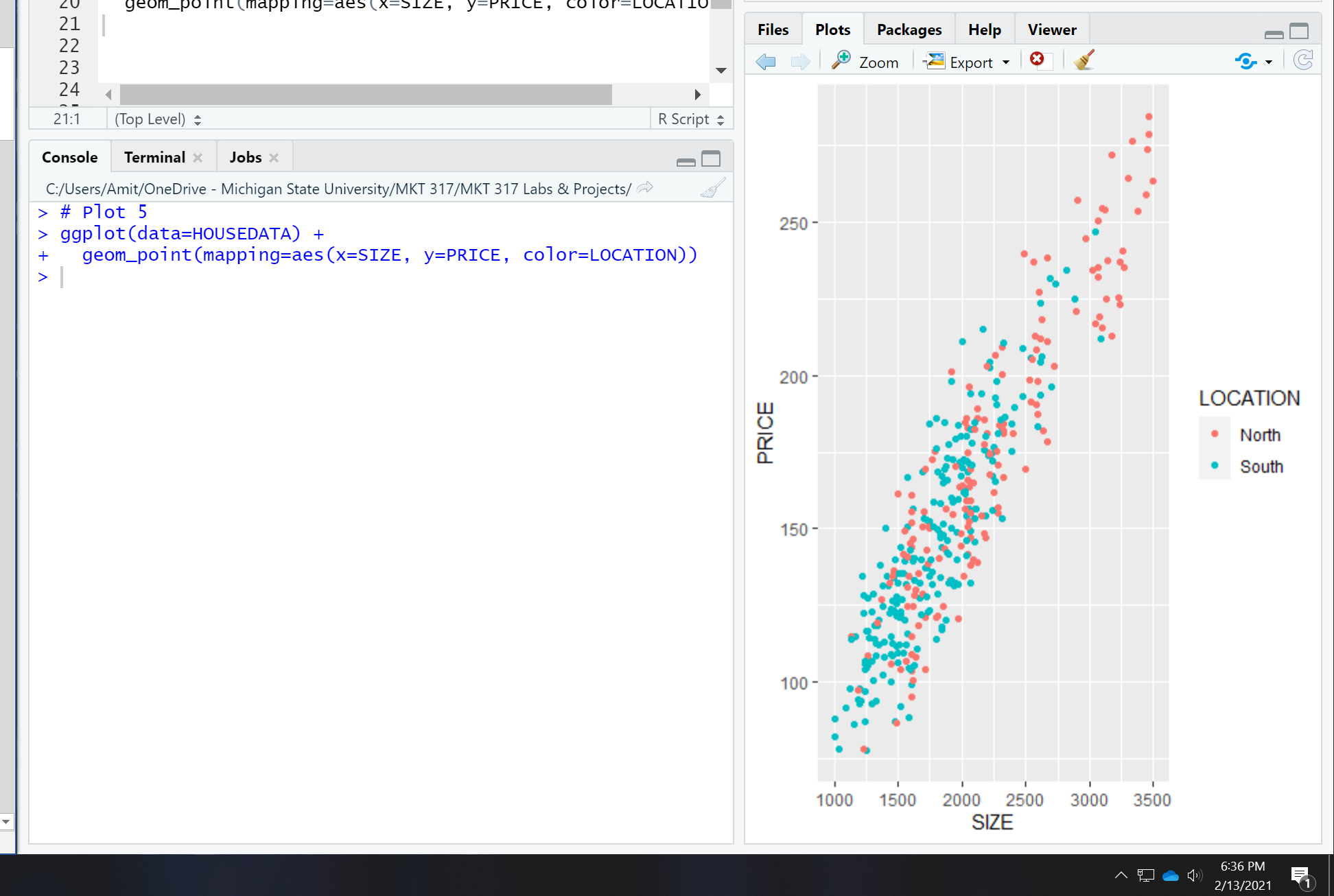
To color-code by a categorical variable, we indicate the color **within** the aes part of the command. When the color is a variable, we do not use quotation marks.

**This should be just one line or code (not two!!) The plus needs to be on the same line as the ggplot() command!**

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=LOCATION))

Screenshot of your plot:



**Plot 6**

To use non-default colors, we can use a color theme, or we can manually assign colors. To manually assign colors, we will add another component to the graph where we manually define the scale for the colors.

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=LOCATION)) +

scale\_color\_manual(values=c("purple", "green"))

Screenshot of your plot:

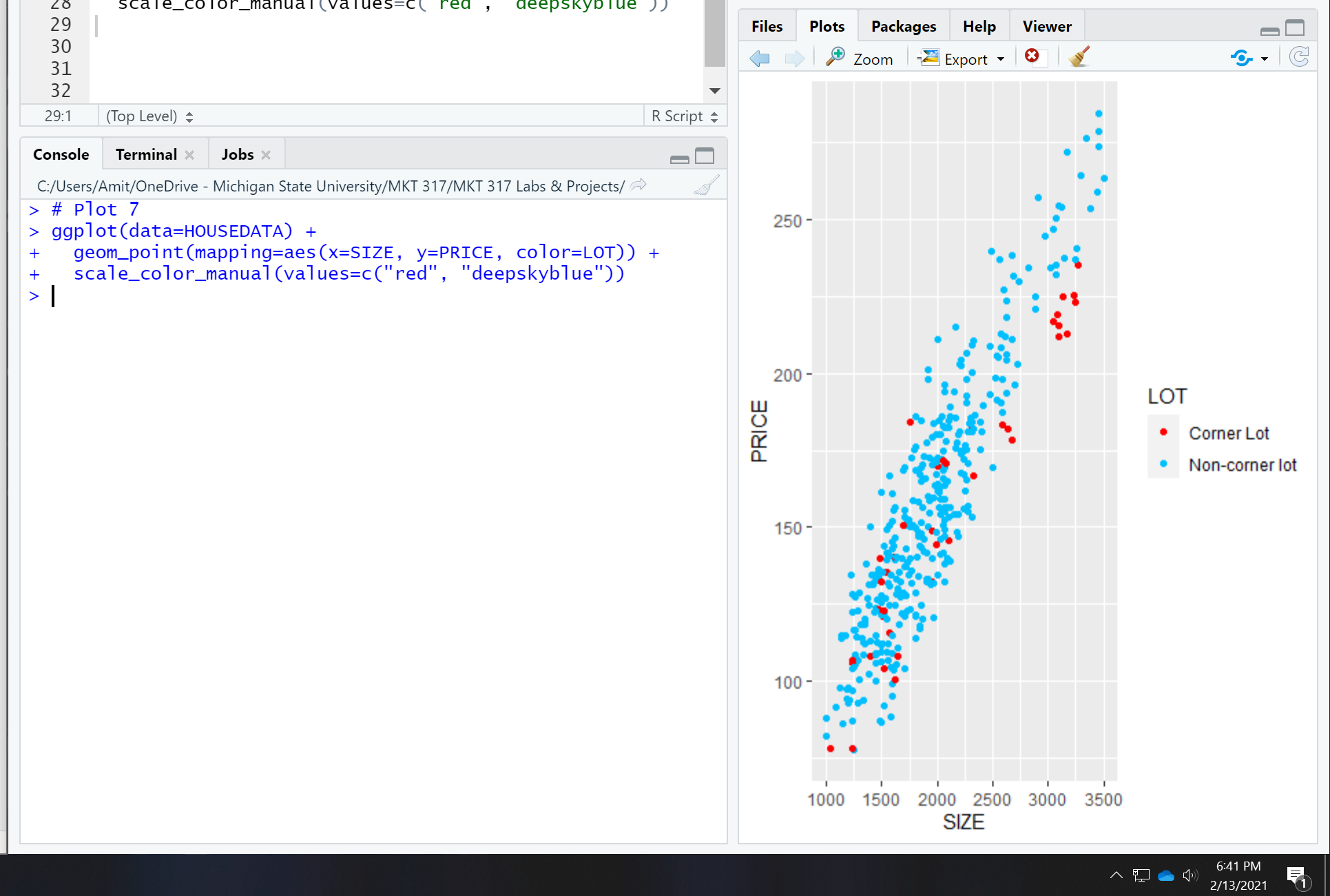


**Plot 7**

Create a scatter plot where the x-axis is SIZE, the y-axis is PRICE, and the dots are color-coded by the variable LOT

R command for your plot: (in screenshot)

Screenshot of your plot:



## **Adding a linear trend line layer to a scatter plot**

**Plot 8**

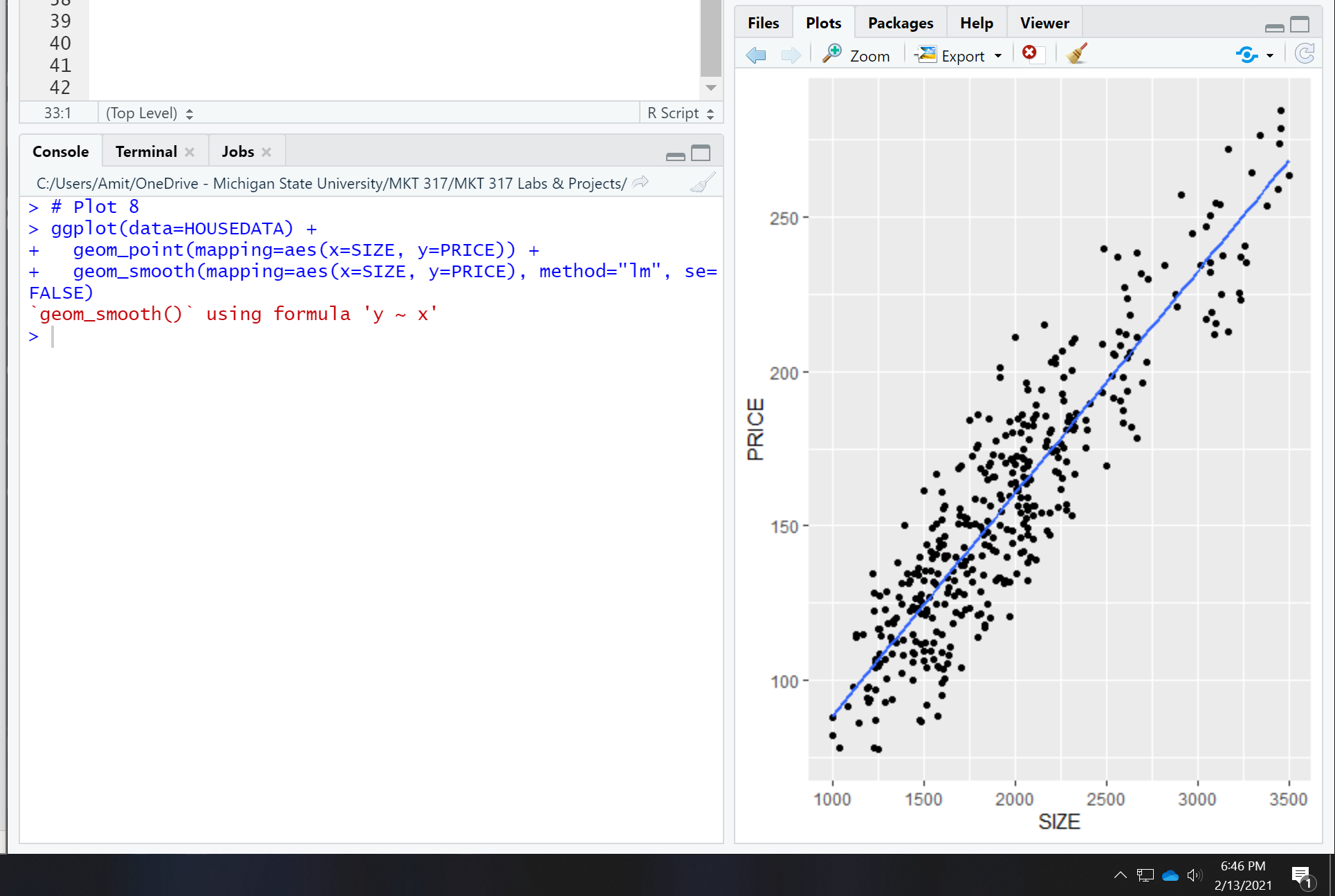
To graph a linear (straight-line) model, we add an additional layer to the plot – we will add a smooth curve, and the shape of the curve will be linear (lm in the command below is an abbreviation for linear model). The se=FALSE removes an “error band” from around the curve.

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE)) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE), method="lm", se=FALSE)

screenshot of your plot:



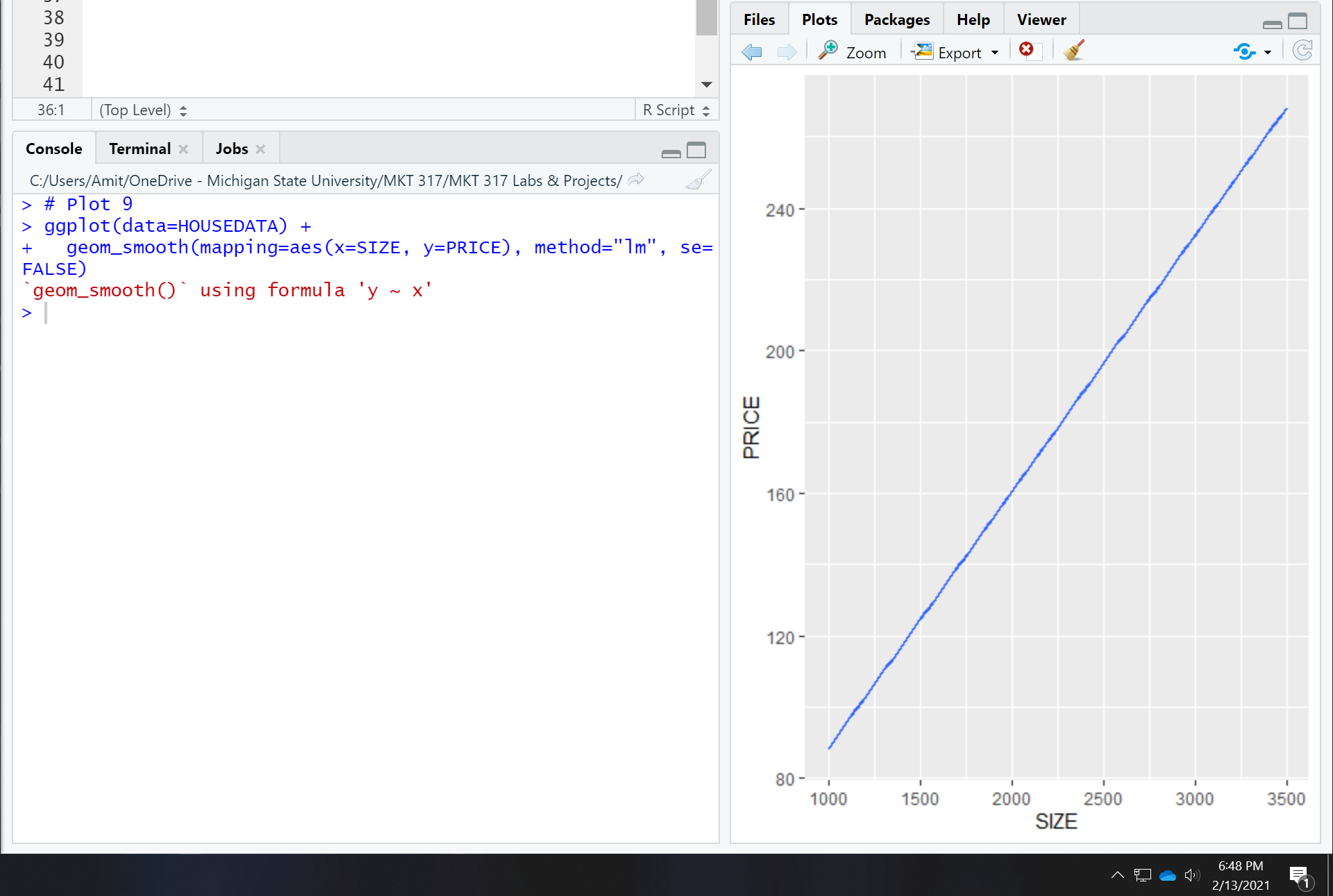
**Plot 9 – what if we left out the geom\_point() layer?**

Two of the three lines are the same as the plot above – how is it different? (no data points)

ggplot(data=HOUSEDATA) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE), method="lm", se=FALSE)

screenshot of your plot:



**Plot 10**

What R commands would create a scatter plot where the x-axis is SIZE, the y-axis is PRICE, the dots are color-coded by LOT, and there is a linear trend line given for both lot types?

Goal:



**Which of these four sets of commands work? Try them all to find out!**

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE), color=LOT) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE), method="lm", se=FALSE)

screenshot of your plot:

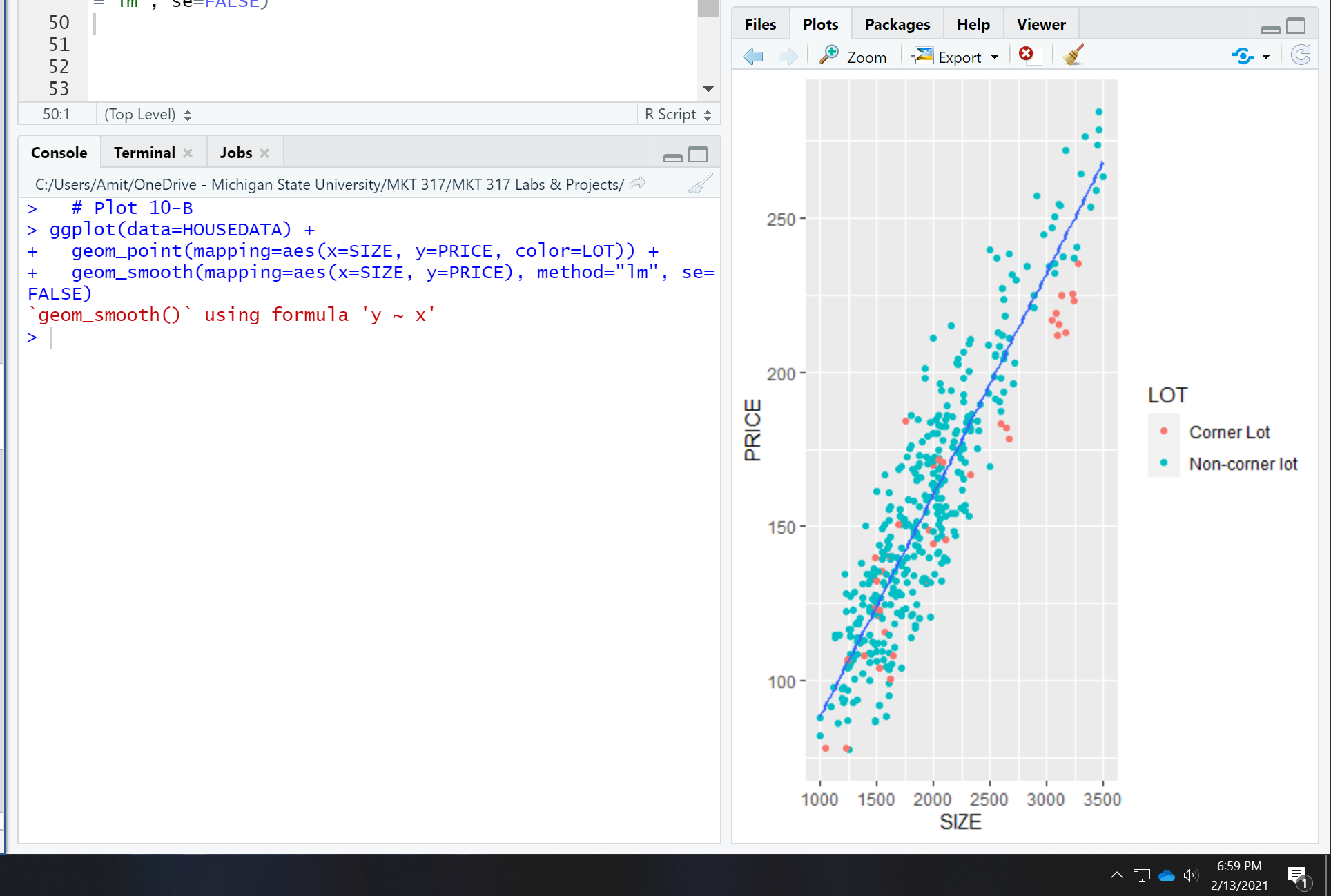
No plot to show, as command is incorrect. It is missing a second parentheses after “color=LOT)” the next command below corrects this.

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=LOT)) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE), method="lm", se=FALSE)

screenshot of your plot:

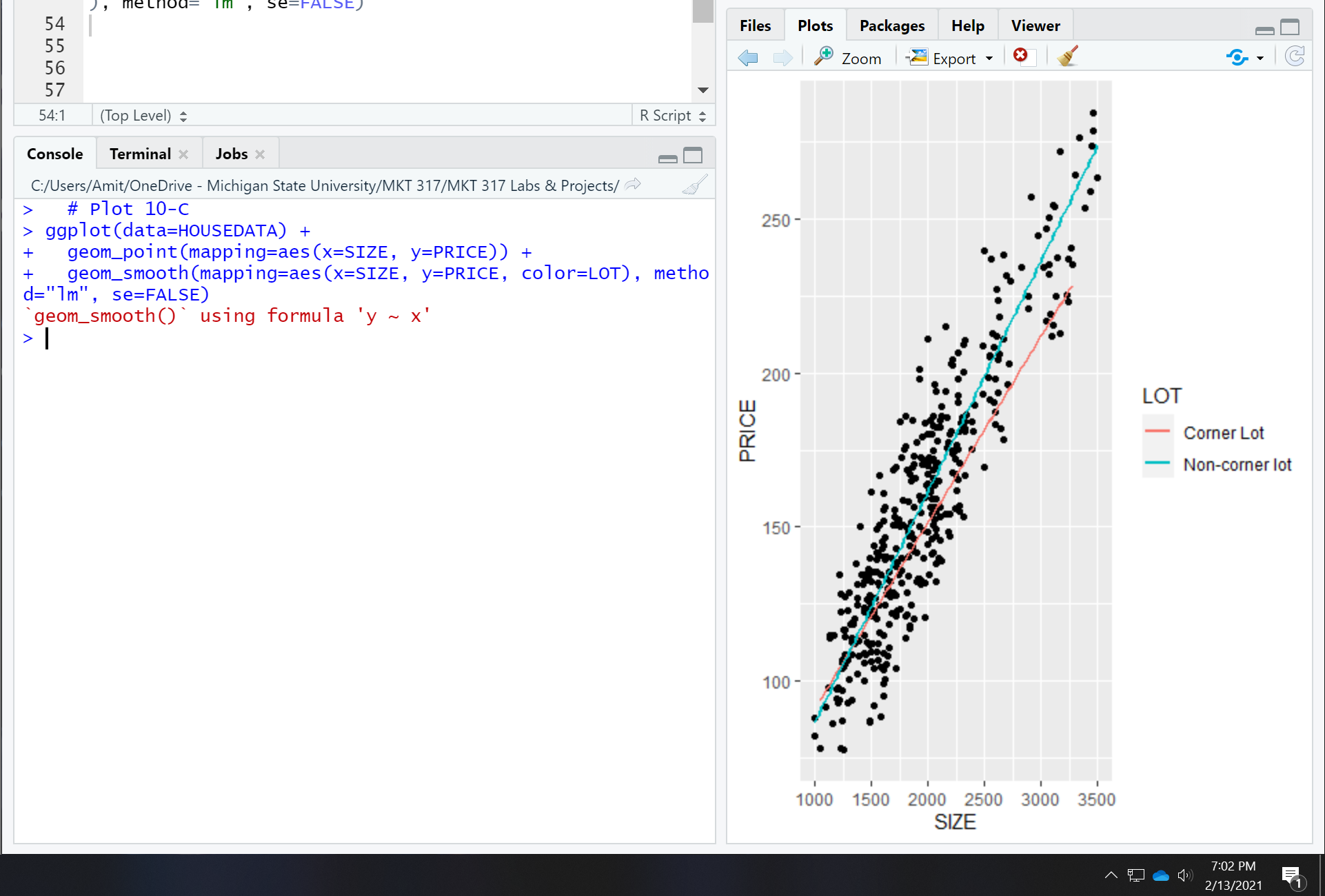


ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE)) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=LOT), method="lm", se=FALSE)

screenshot of your plot:

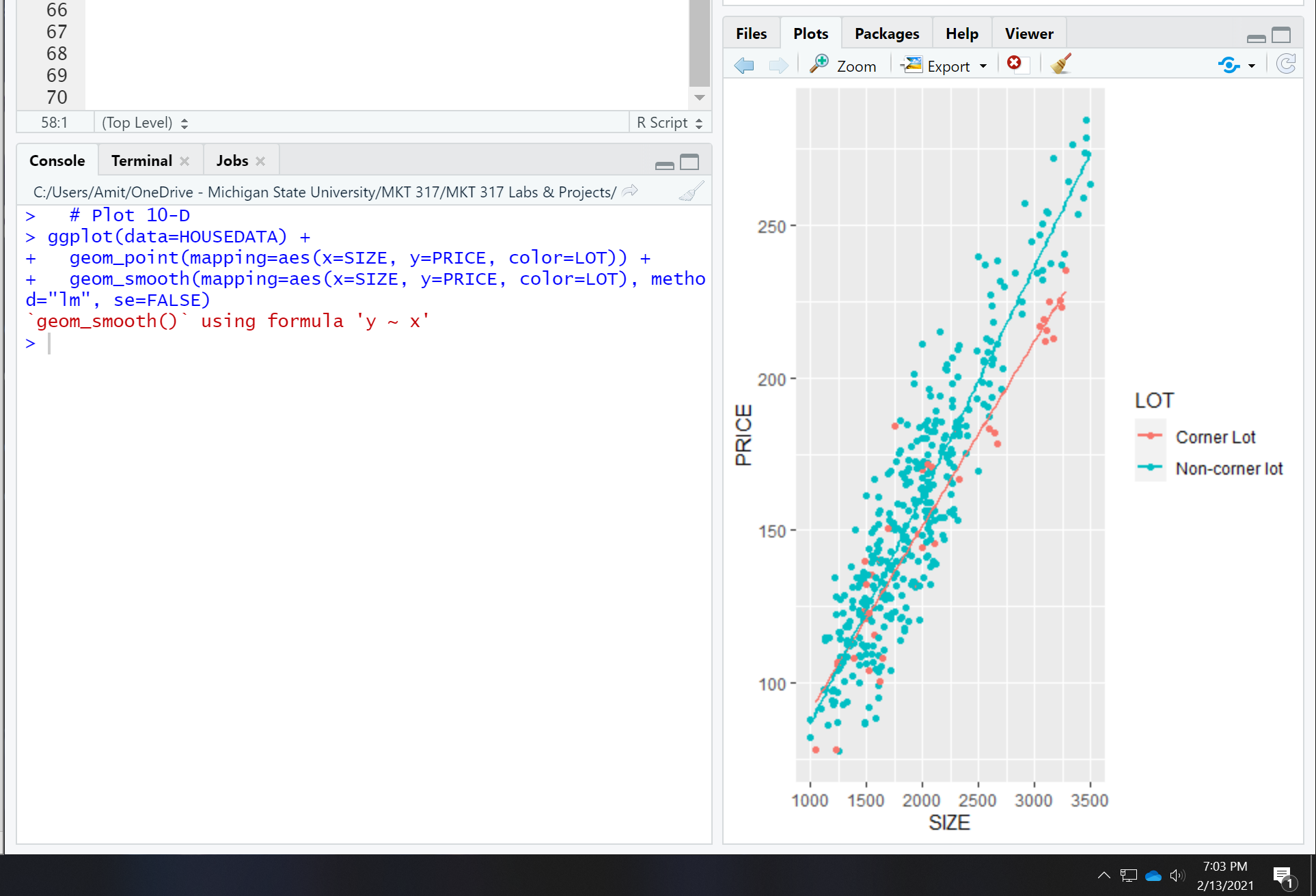


ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=LOT)) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=LOT), method="lm", se=FALSE)

screenshot of your plot:



For the three plots above that weren’t what we wanted – what was incorrect with the code?

Plot A was missing a parentheses which made the command fail, Plot 2 fixed the mistake but only had 1 trend line, and Plot 3 didn’t have colored dots.

**Plot 11:**

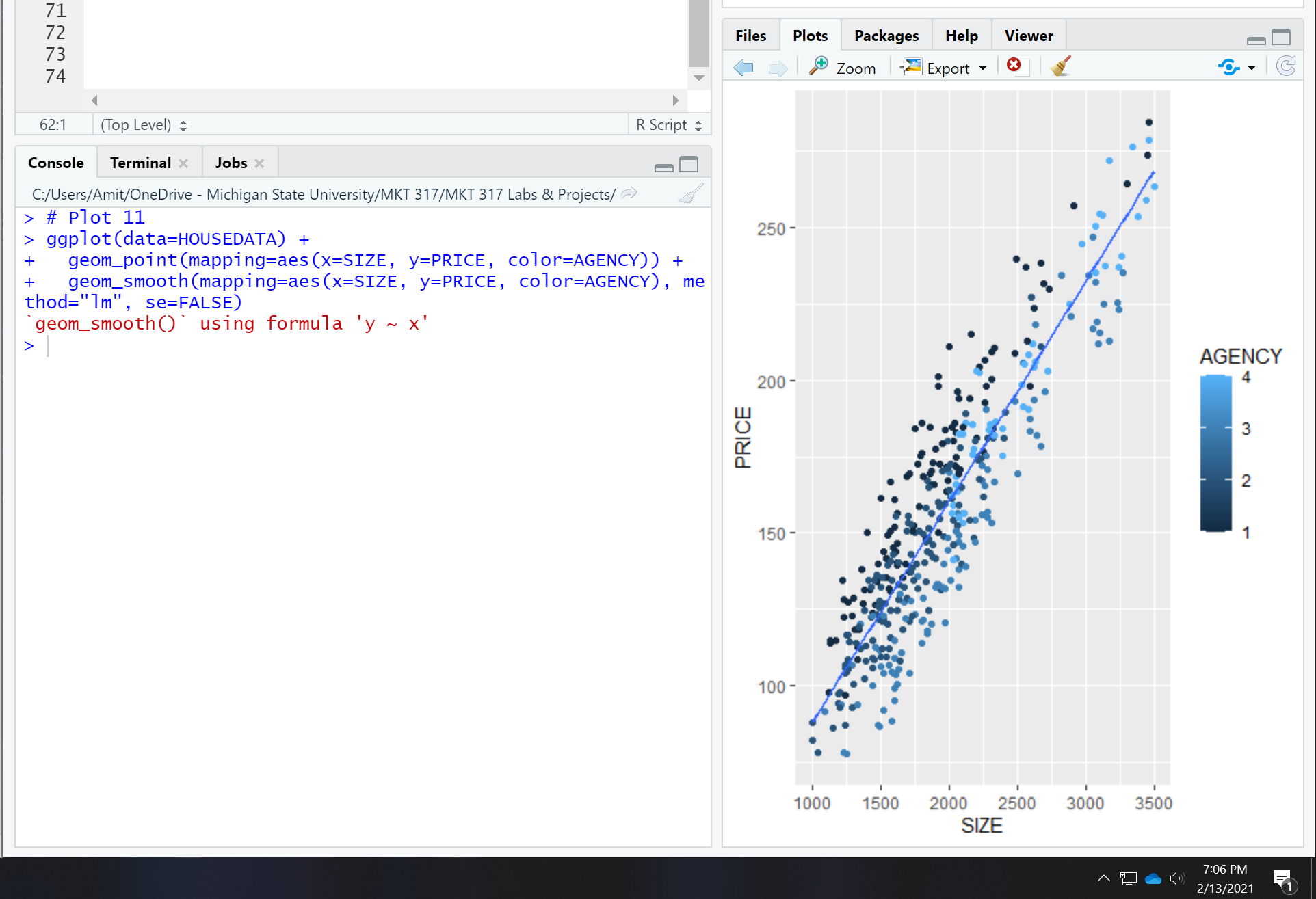
What happens if we want to color-code by the numeric categorical variable agency?

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=AGENCY)) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=AGENCY), method="lm", se=FALSE)

screenshot of your plot:



## **Re-formatting categorical variables.**

When you have a categorical variable, use as.factor(variable name) instead of the original variable!

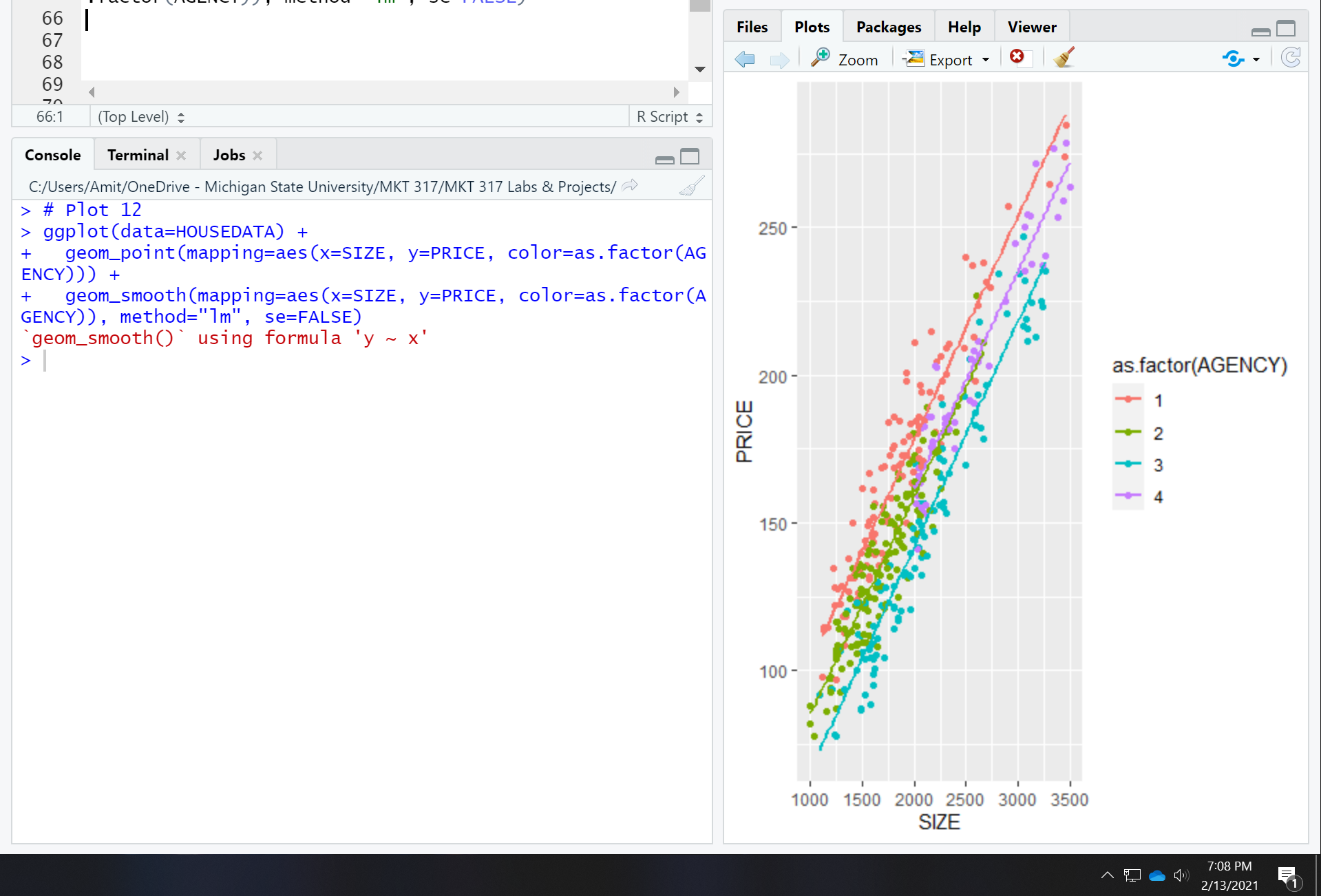
**Plot 12: How is this different form plot 11?** Has 4 separate trend lines and separate colors instead of a gradient.

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE)

screenshot of your plot:



## **Changing the theme.**

What if we want to print our graph? The background would look nicer in white. Let’s use the black and white theme!

**Plot 13:**

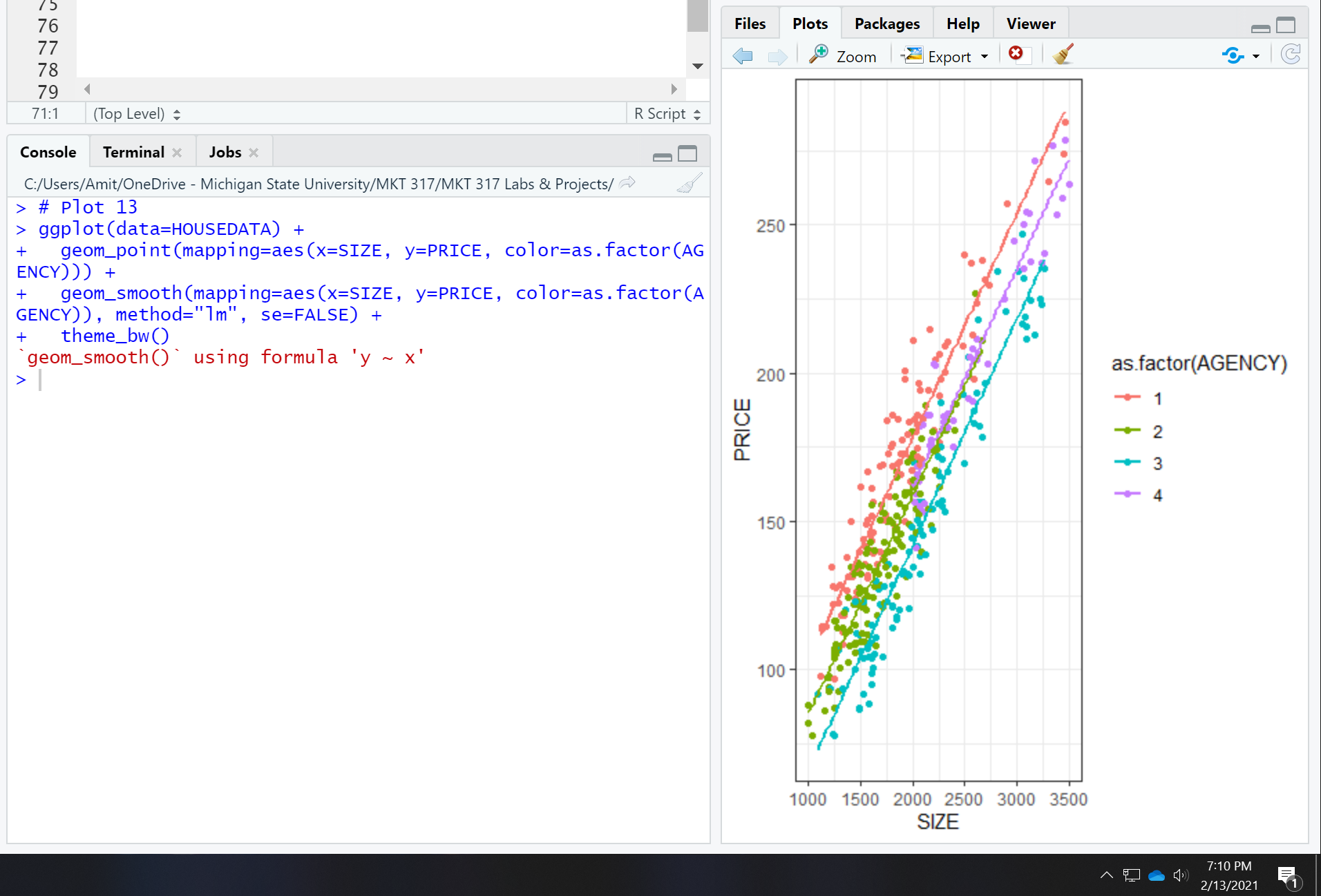
ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE) +

theme\_bw()

screenshot of your plot:



## **Changing the legend and axis titles.**

The axis titles can be changed with the labs commands (labels)

**Plot 14:**

ggplot(data=HOUSEDATA) +

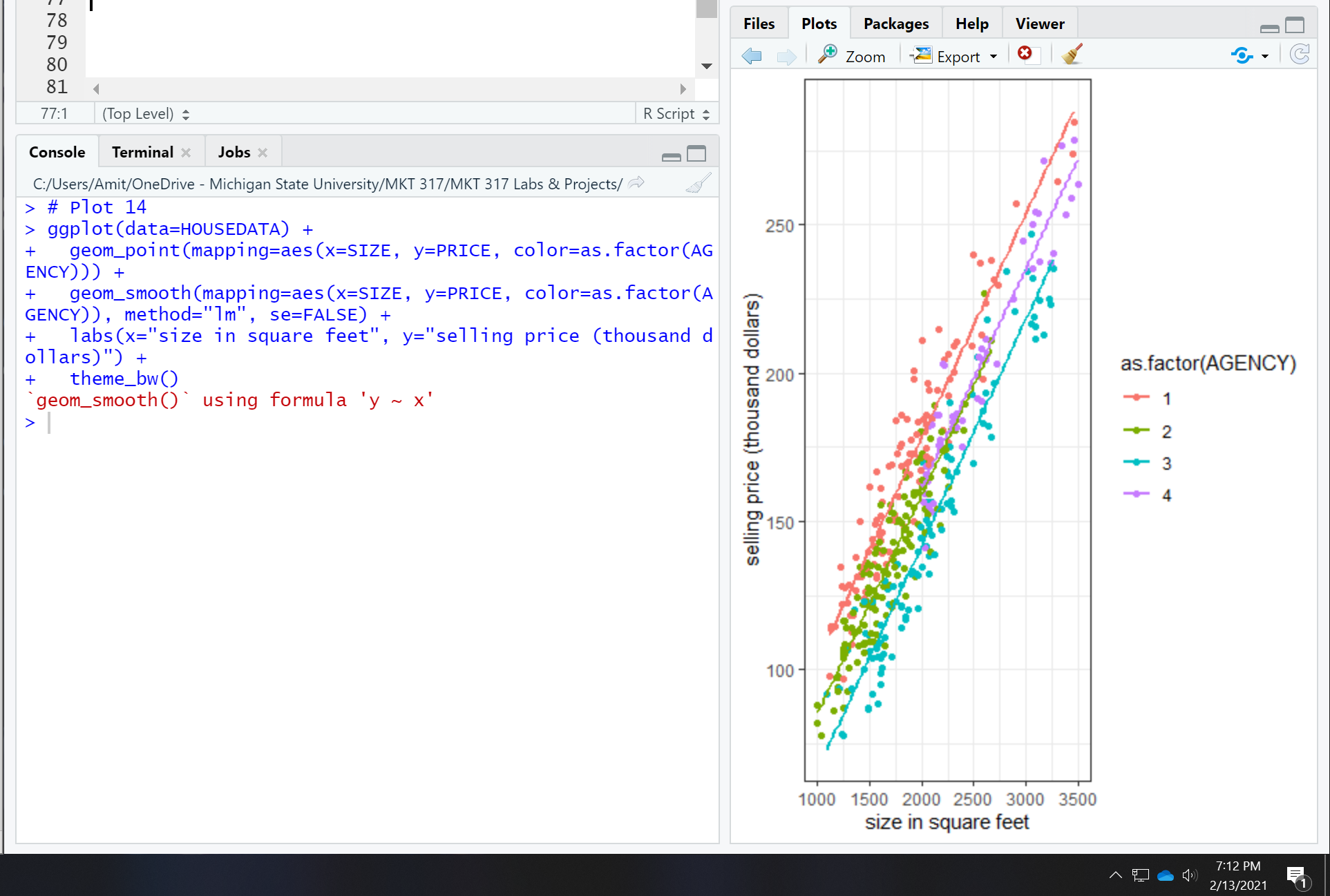
geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE) +

labs(x="size in square feet", y="selling price (thousand dollars)") +

theme\_bw()

screenshot of your plot:



**Plot 15:**

Since we are color-coding by AGENCY, we can also use the labs command to give a name of the legend!

ggplot(data=HOUSEDATA) +

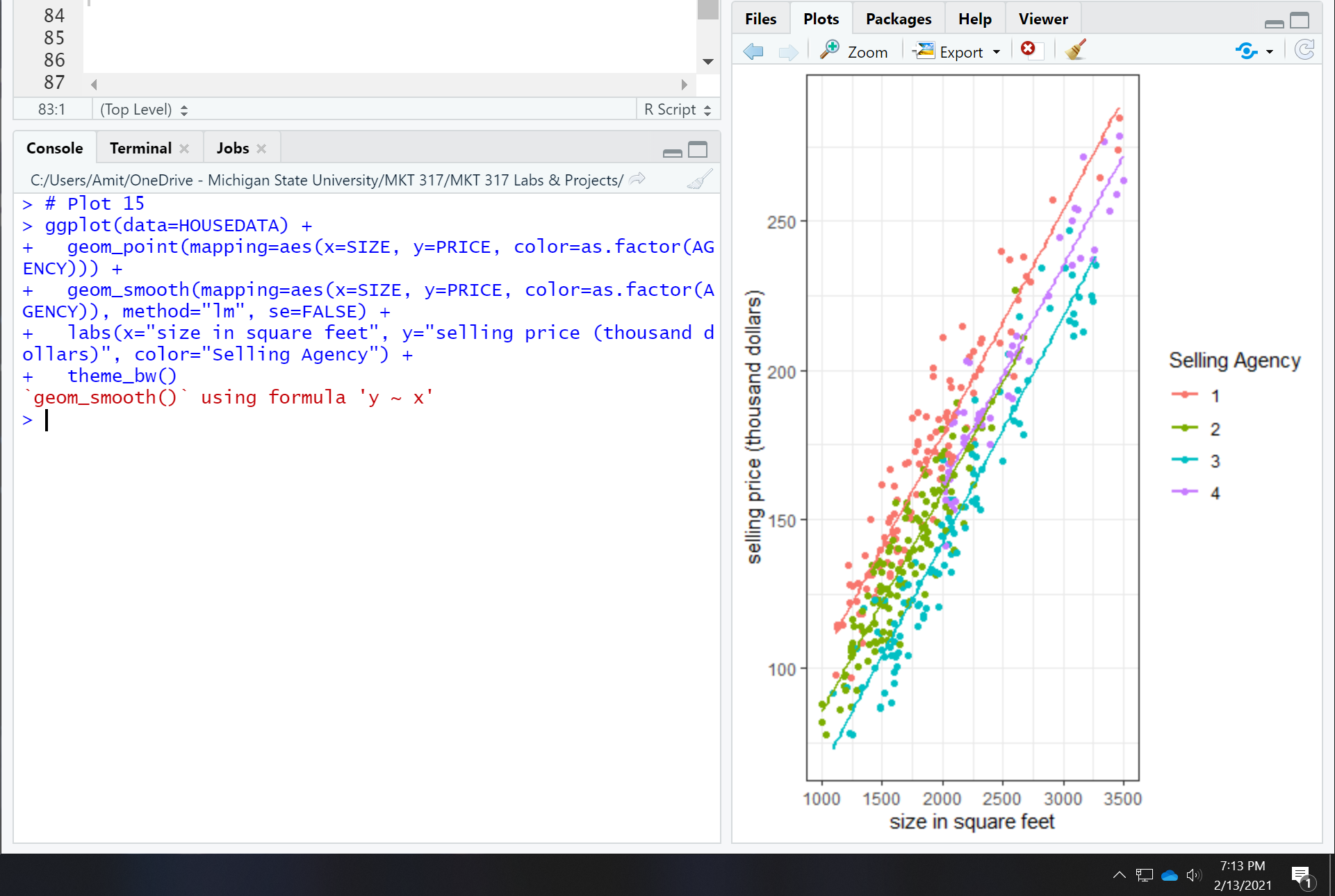
geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE) +

labs(x="size in square feet", y="selling price (thousand dollars)", color="Selling Agency") +

theme\_bw()

screenshot of your plot:



**Plot 16:**

Wait! PRICE is money – can we put a $ next to it? Answer: yes! But we need the scales package. The part of the command below that adds dollar signs is in green.

library(scales)

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

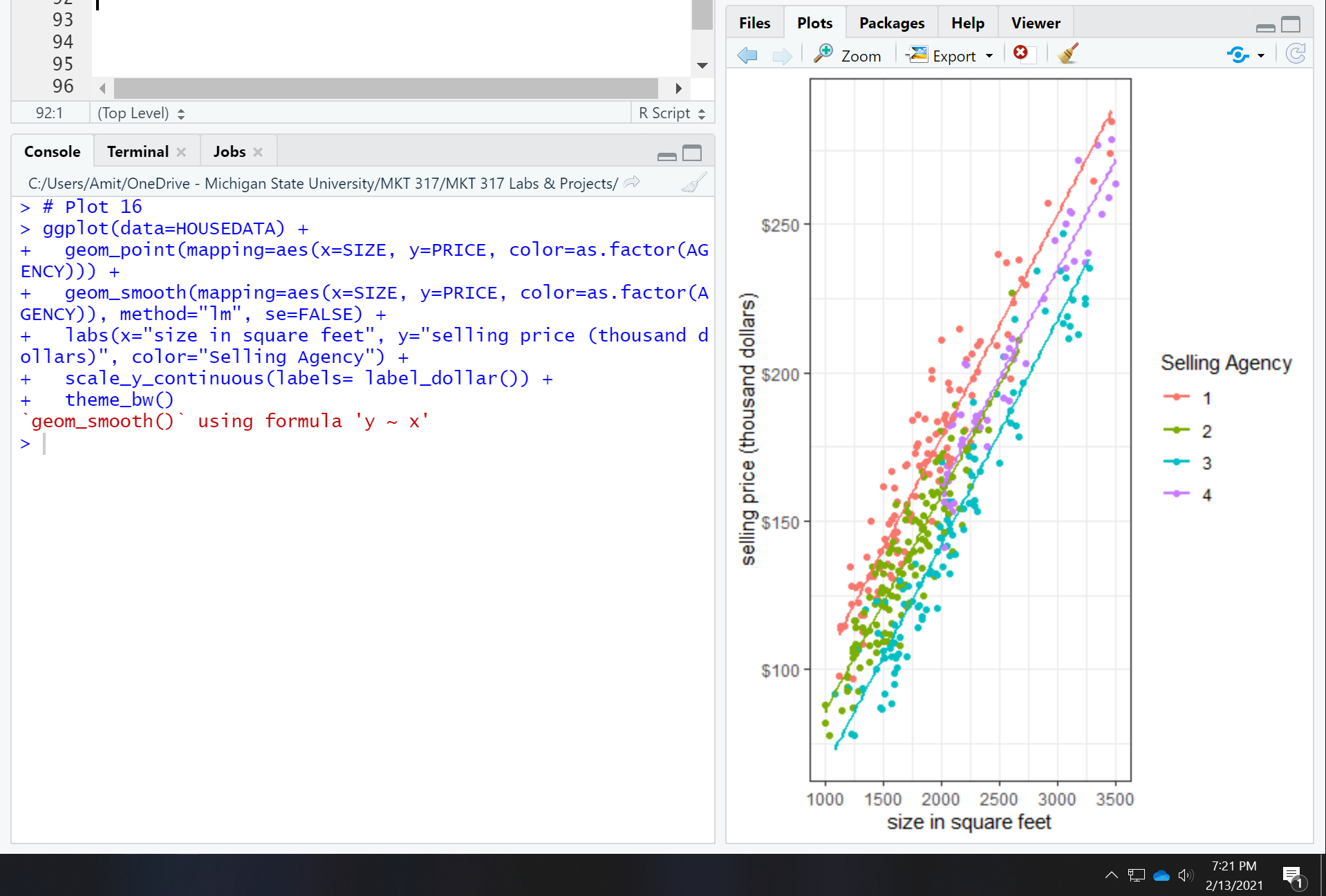
geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE) +

labs(x="size in square feet", y="selling price (thousand dollars)", color="Selling Agency") +

scale\_y\_continuous(labels= label\_dollar()) +

theme\_bw()

screenshot of your plot:



**Plot 17: We can add more labels to the numbers on the y-axis! This is also using the Scales package.**

library(scales)

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY))) +

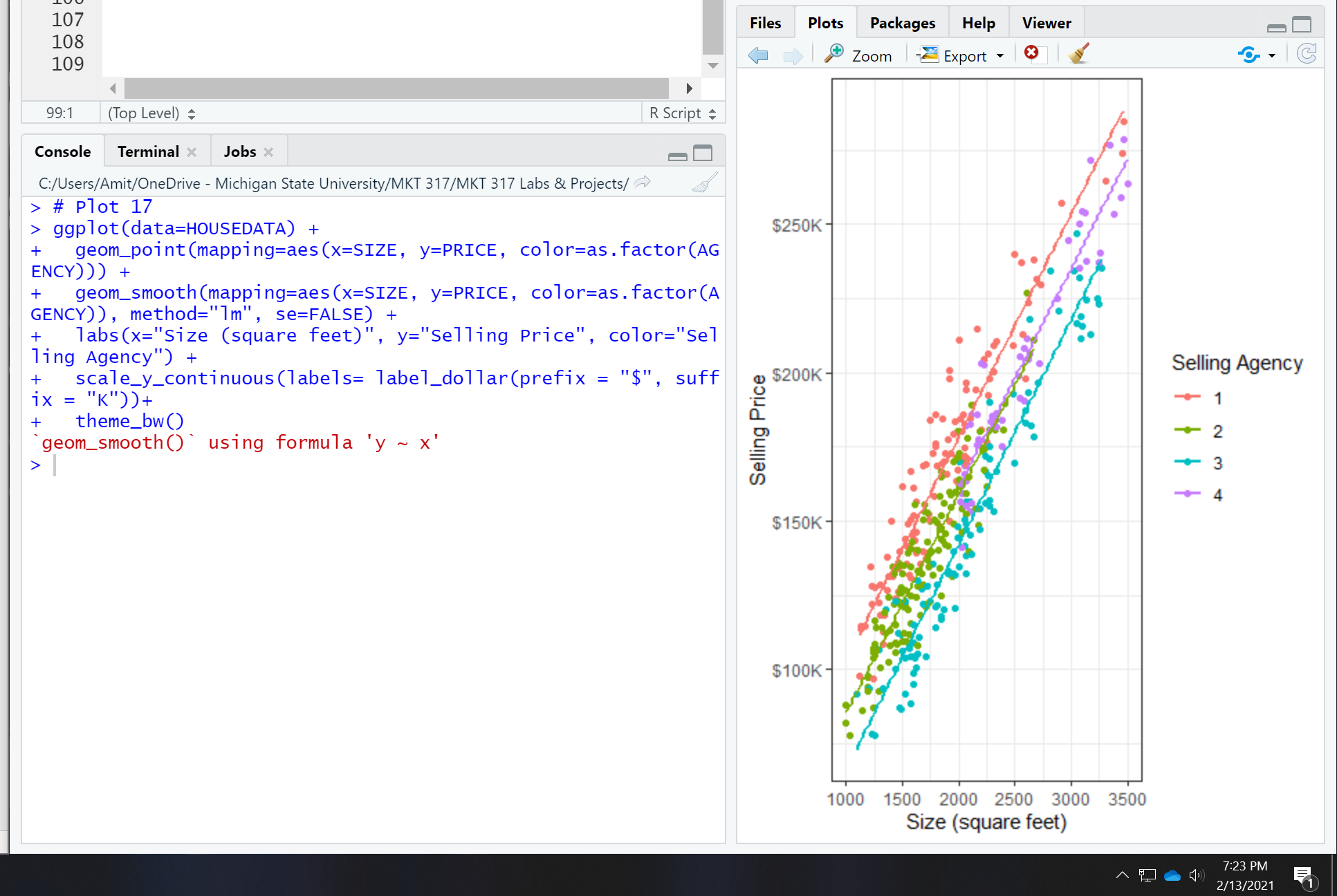
geom\_smooth(mapping=aes(x=SIZE, y=PRICE, color=as.factor(AGENCY)), method="lm", se=FALSE) +

labs(x="Size (square feet)", y="Selling Price", color="Selling Agency") +

scale\_y\_continuous(labels= label\_dollar(prefix = "$", suffix = "K"))+

theme\_bw()

screenshot of your plot:



**Plot 18:**

We can also re-format the data within the plot command! By multiplying PRICE by 1000, we are changing the units from Selling Price (thousand dollars) to Selling Price (dollars)

ggplot(data=HOUSEDATA) +

geom\_point(mapping=aes(x=SIZE, y=PRICE\*1000, color=as.factor(AGENCY))) +

geom\_smooth(mapping=aes(x=SIZE, y=PRICE\*1000, color=as.factor(AGENCY)), method="lm", se=FALSE) +

labs(x="Size (square feet)", y="Selling Price", color="Selling Agency") +

scale\_y\_continuous(labels=scales::dollar\_format()) +

theme\_bw()

screenshot of your plot:

