Lab3_12Feb

Aicha

2023-02-12

import tidyverse

This library it is package that includes several functions for data importing and processing

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
```

```
mn_homes <- read.csv("data/mn_homes.csv")
glimpse(mn_homes)
```

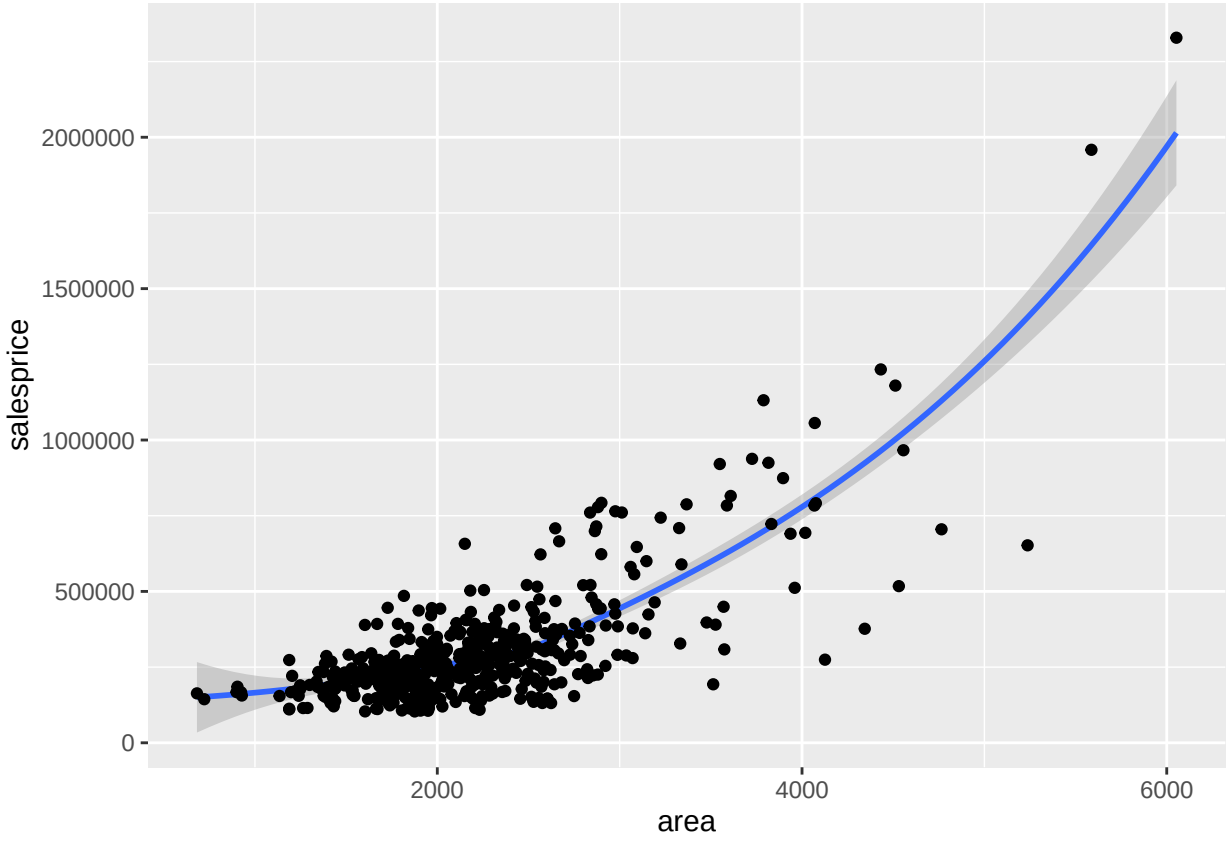
```
## Rows: 495
## Columns: 13
## $ saleyear      <int> 2012, 2014, 2005, 2010, 2010, 2013, 2011, 2007, 2013, 20~
## $ salemonth     <int> 6, 7, 7, 6, 2, 9, 1, 9, 10, 6, 7, 8, 5, 2, 7, 6, 10, 6, ~
## $ salesprice    <dbl> 690467.0, 235571.7, 272507.7, 277767.5, 148324.1, 242871~
## $ area          <int> 3937, 1440, 1835, 2016, 2004, 2822, 2882, 1979, 3140, 35~
## $ beds          <int> 5, 2, 2, 3, 3, 3, 4, 3, 4, 3, 3, 3, 2, 3, 3, 6, 2, 3, 2,~
## $ baths         <int> 4, 1, 1, 2, 1, 3, 3, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1,~
## $ stories       <dbl> 2.5, 1.7, 1.7, 2.5, 1.0, 2.0, 1.7, 1.5, 1.5, 2.5, 1.0, 2~
## $ yearbuilt     <int> 1907, 1919, 1913, 1910, 1956, 1934, 1951, 1929, 1940, 19~
## $ neighborhood  <chr> "Lowry Hill", "Cooper", "Hiawatha", "King Field", "Shing~
## $ community     <chr> "Calhoun-Isles", "Longfellow", "Longfellow", "Southwest"~
## $ lotsize       <int> 6192, 5160, 5040, 4875, 5060, 6307, 6500, 5600, 6350, 75~
## $ numfireplaces <int> 0, 0, 0, 0, 0, 2, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0,~
## $ fireplace     <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, TR~
```

Fist Graph

We usally use ggplot function to draw plots. It has several parameters as the following: 1. data: Here we specify the dataframe that we will use 2. mapping: Here you specify the x-axis and y-axis dimensions (scale) 3. Add drawing component

```
ggplot(data=mn_homes, mapping=aes(x=area, y=salesprice)) + geom_smooth() + geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

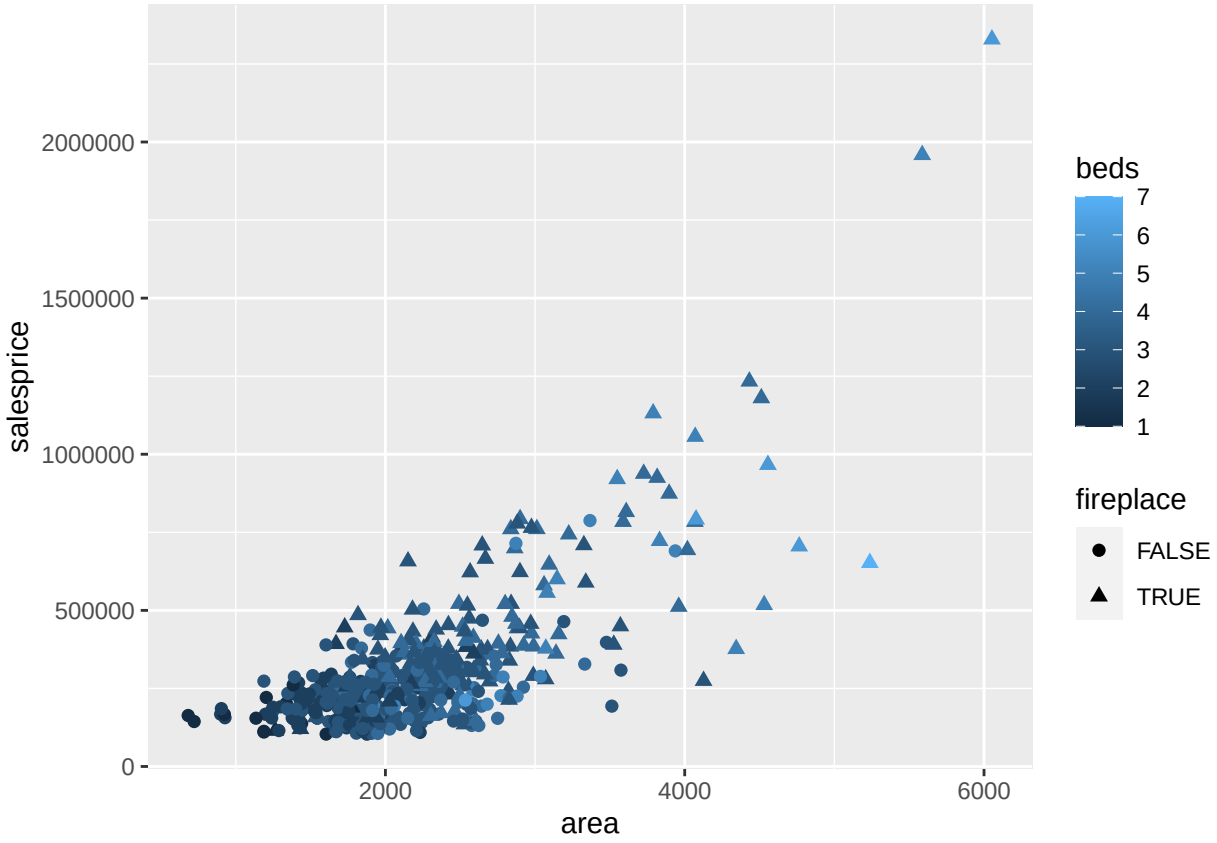


As we can see above geom_point() represents each record in the dataframe with a point in the plot. It usally helps to detect outliers. There are different types of drawings. For example, we can use geom_smooth() to represent the relation between two variables with a curved line as you can see above.

aes()

aes() method also allows us to change some visual propreties in the plot including: shape, color, size, and transperancy. For example, we can change the color of the points in the plot based on another variable (such as fireplace) => which allow visualization of more than 2 variables (3 or more) => Example, shape=23, fill="blue", color="darkred" size = 20 Note: not all visualization make sense

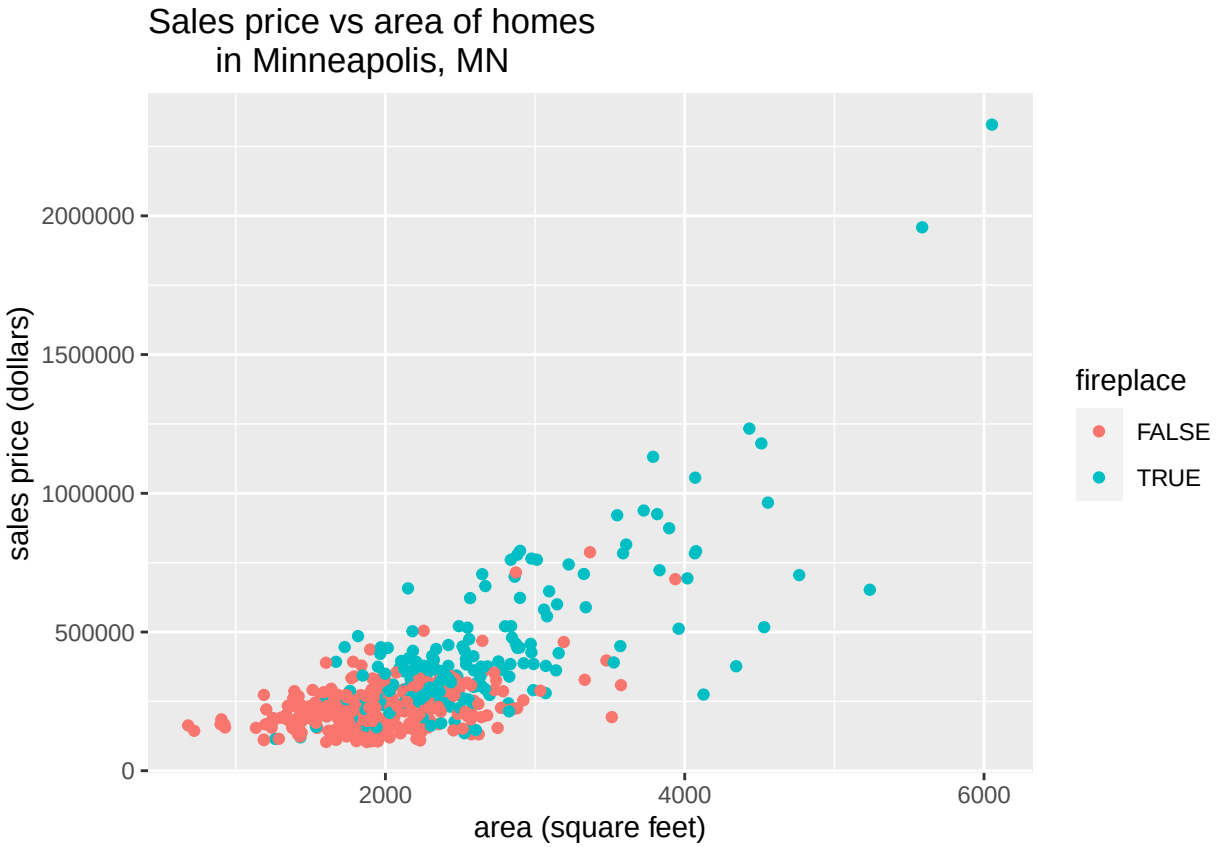
```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice, color=beds,
                   shape=fireplace)) + geom_point(size=2)
```



Labels

We can also add labels to graph using labs() methods as the following:

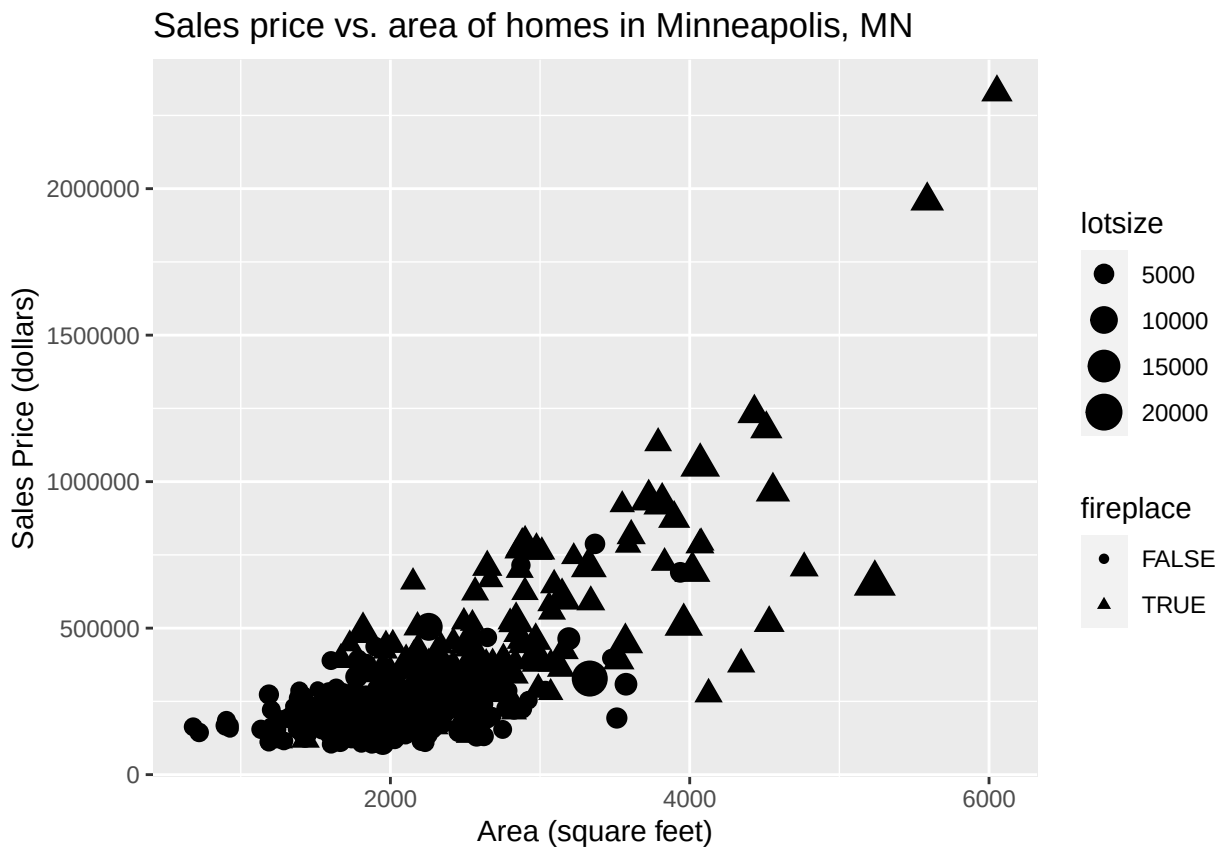
```
ggplot(data=mn_homes,
       mapping=aes(x=area,
                   y=salesprice, color=fireplace)) +
  geom_point() +
  labs(title = "Sales price vs area of homes
       in Minneapolis, MN", x="area (square feet)",
       y="sales price (dollars)")
```



Practice

Question: Are the above visualizations effective? Why or why not? How might you improve them? No this visualization is not effetive we can't notice a clear difference between sizes we can try changing the true false to shape intead of colors.

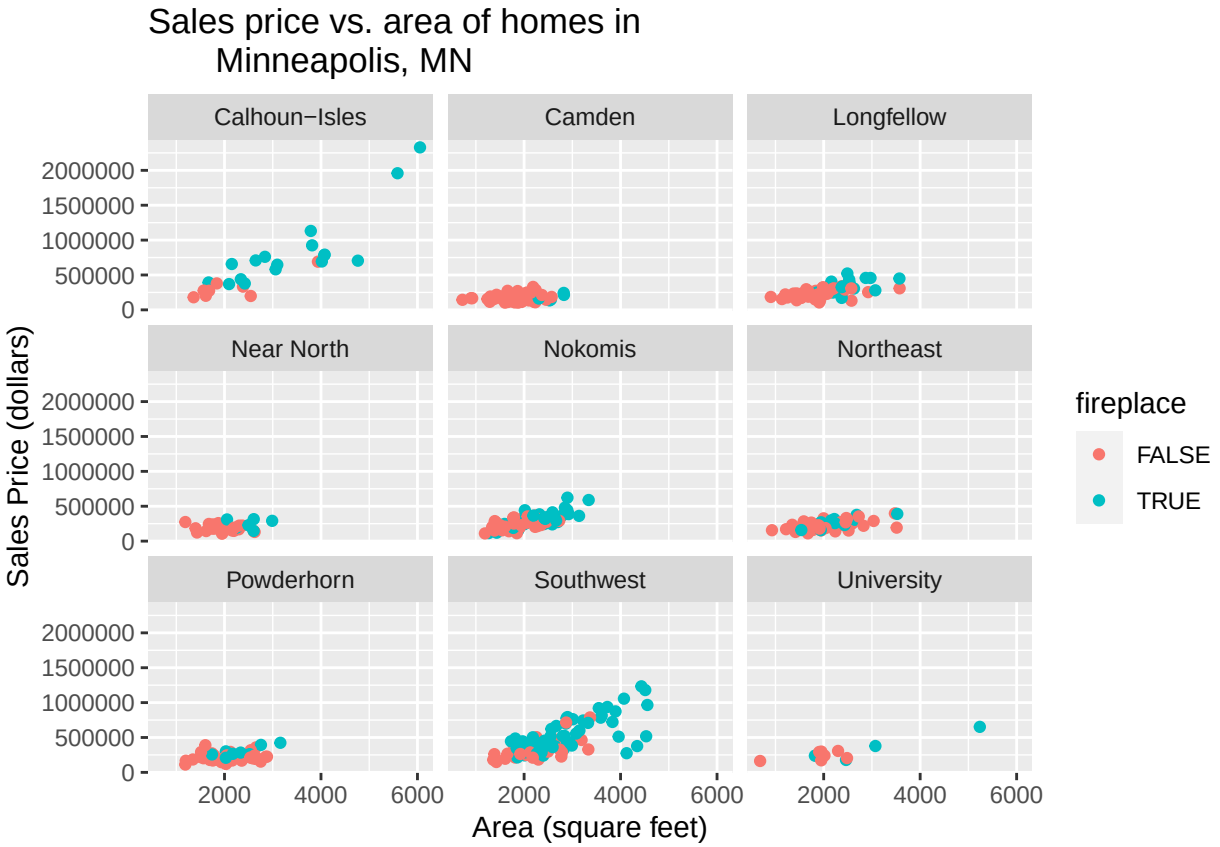
```
ggplot(data = mn_homes,
mapping = aes(x = area, y = salesprice,
shape = fireplace,
size = lotsize)) +
geom_point() +
labs(title = "Sales price vs. area of homes in Minneapolis, MN",
x = "Area (square feet)", y = "Sales Price (dollars)")
```



Faceting

Faceting: having different graphs based on different values of specific variable. For example: if we want to make a similar graph to what we have above but for wich community (which a variable in our dataframe), we can do the following. => in faceting you can chnage the number of rows. A. Facet Wrap

```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice,
                   color=fireplace)) +
  geom_point() +
  labs(title = "Sales price vs. area of homes in
       Minneapolis, MN",
       x = "Area (square feet)",
       y = "Sales Price (dollars)") +
  facet_wrap(~community)
```

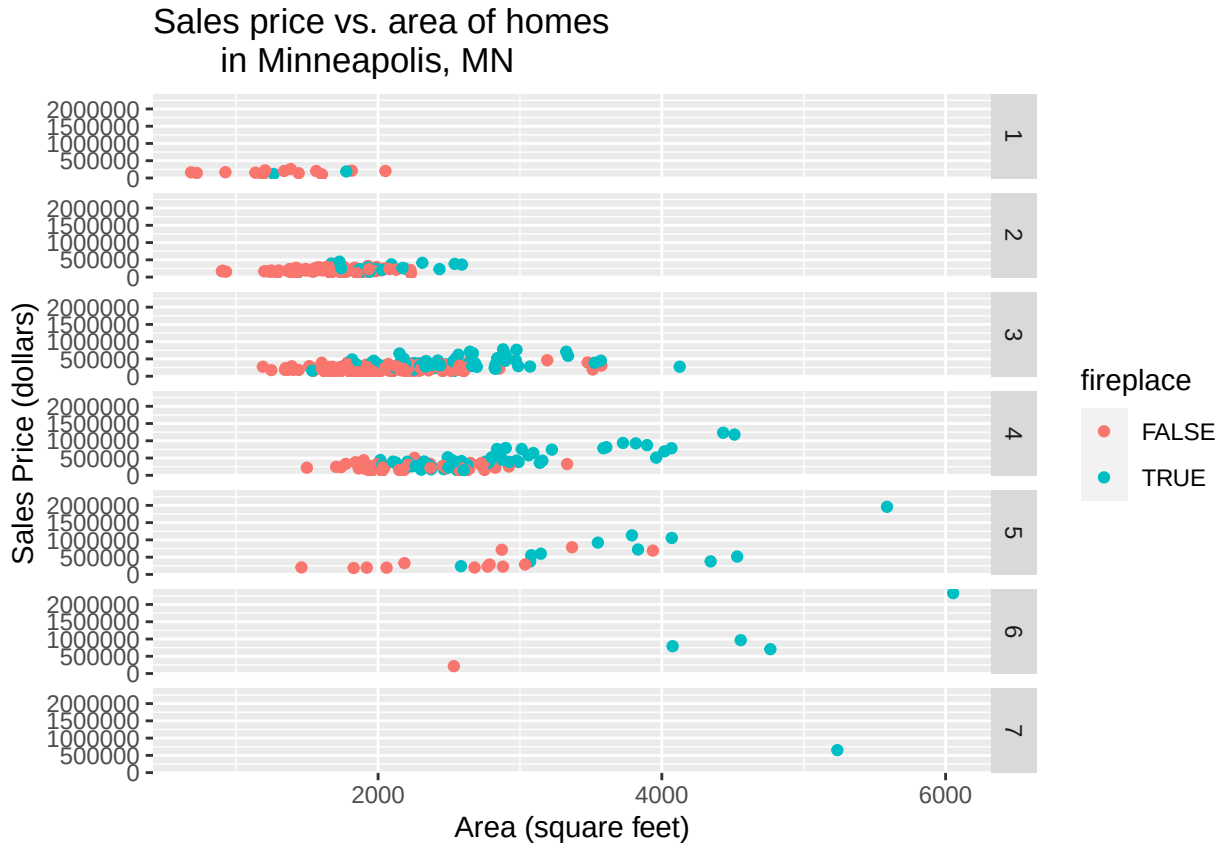


B. Facet Grid In facet grid you can plot: 1. columns: facet_grid(. ~ beds)

```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice,
                   color=fireplace)) +
  geom_point() +
  labs(title = "Sales price vs. area of homes
       in Minneapolis, MN",
       x = "Area (square feet)",
       y = "Sales Price (dollars)") +
  facet_grid(.~beds)
```

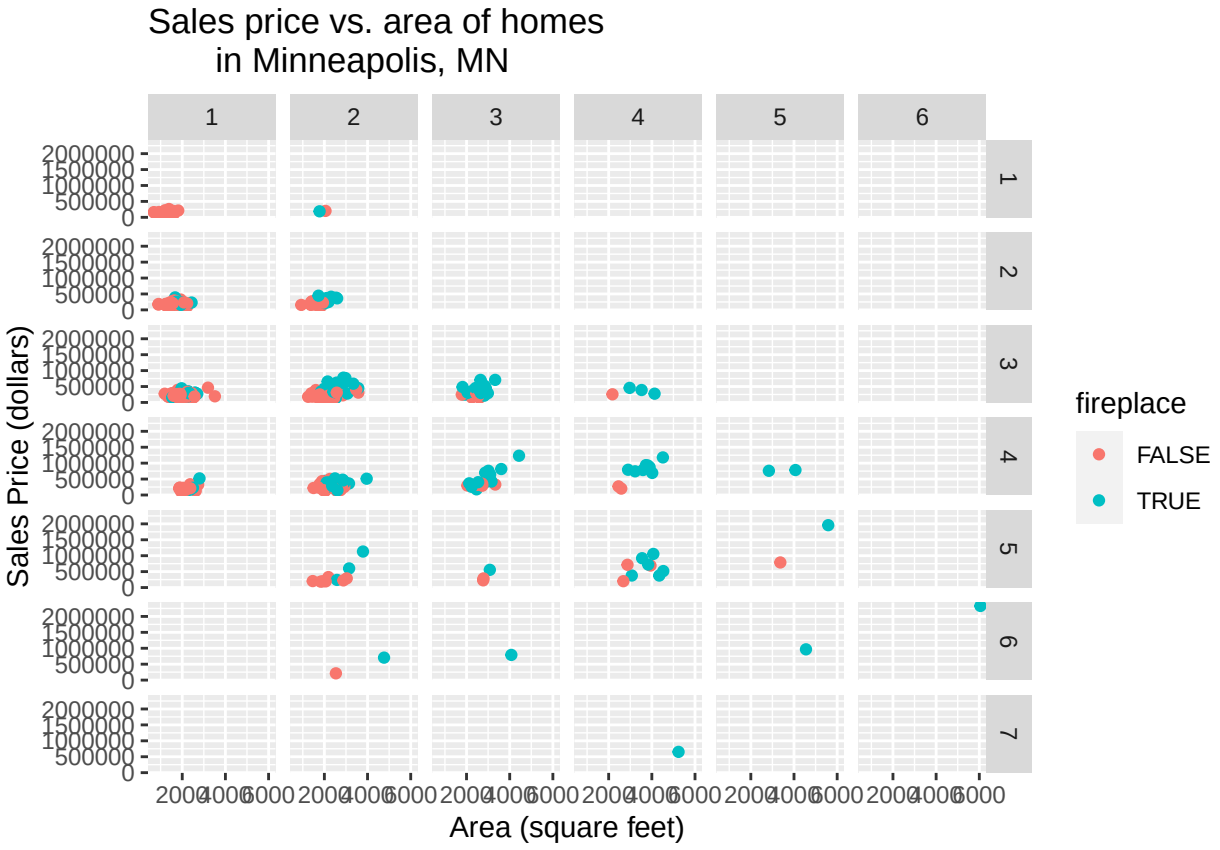
2. rows: facet_grid(beds ~ .)

```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice,
                   color=fireplace)) +
  geom_point() +
  labs(title = "Sales price vs. area of homes
       in Minneapolis, MN",
       x = "Area (square feet)",
       y = "Sales Price (dollars)") +
  facet_grid(beds~.)
```



3. 2 variables: facet_grid(beds ~ baths)

```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice,
                   color=fireplace)) +
  geom_point() +
  labs(title = "Sales price vs. area of homes
       in Minneapolis, MN",
       x = "Area (square feet)",
       y = "Sales Price (dollars)") +
  facet_grid(beds~baths)
```



Facet_Wrap VS Facet_Grid

The facet_grid() function will produce a grid of plots for each combination of variables that you specify, even if some plots are empty. The facet_wrap() function will only produce plots for the combinations of variables that have values, which means it won't produce any empty plots.

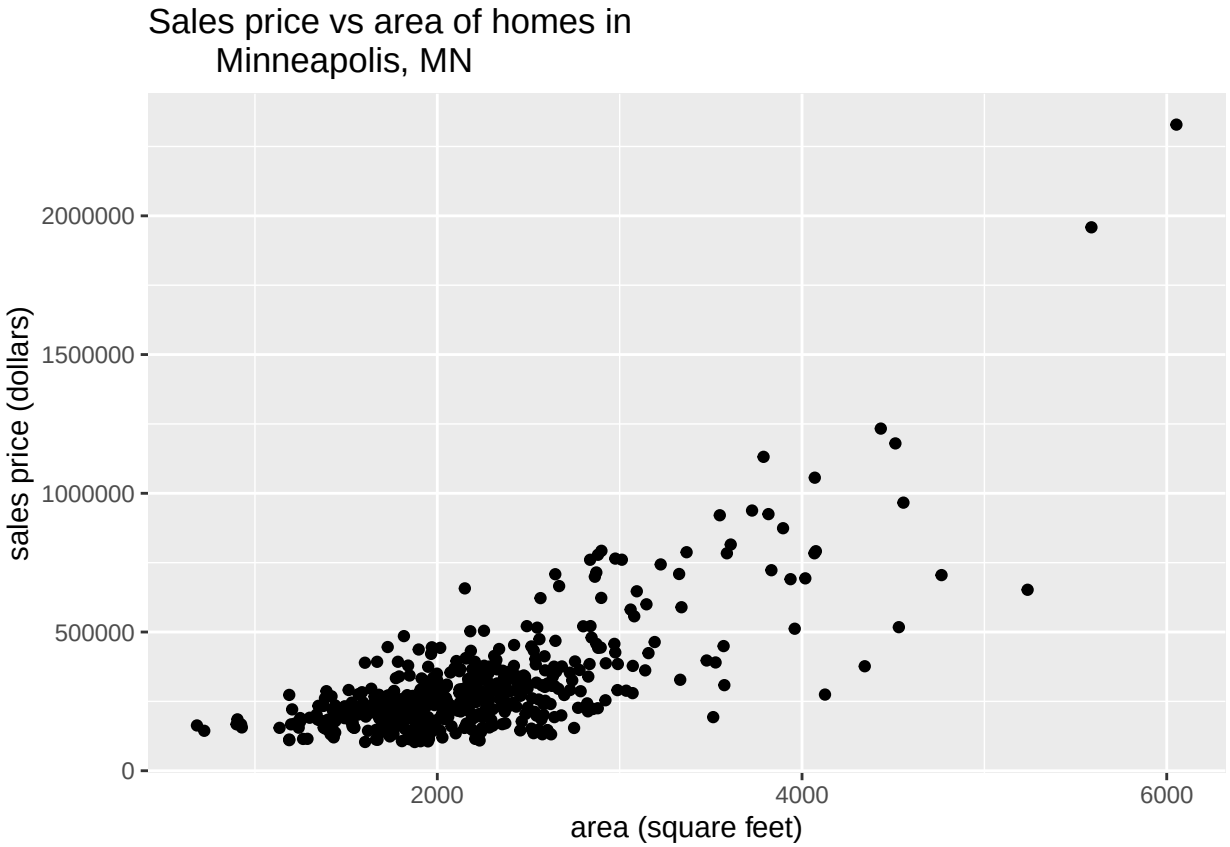
Hint

If you need help ?ggplot

Practice

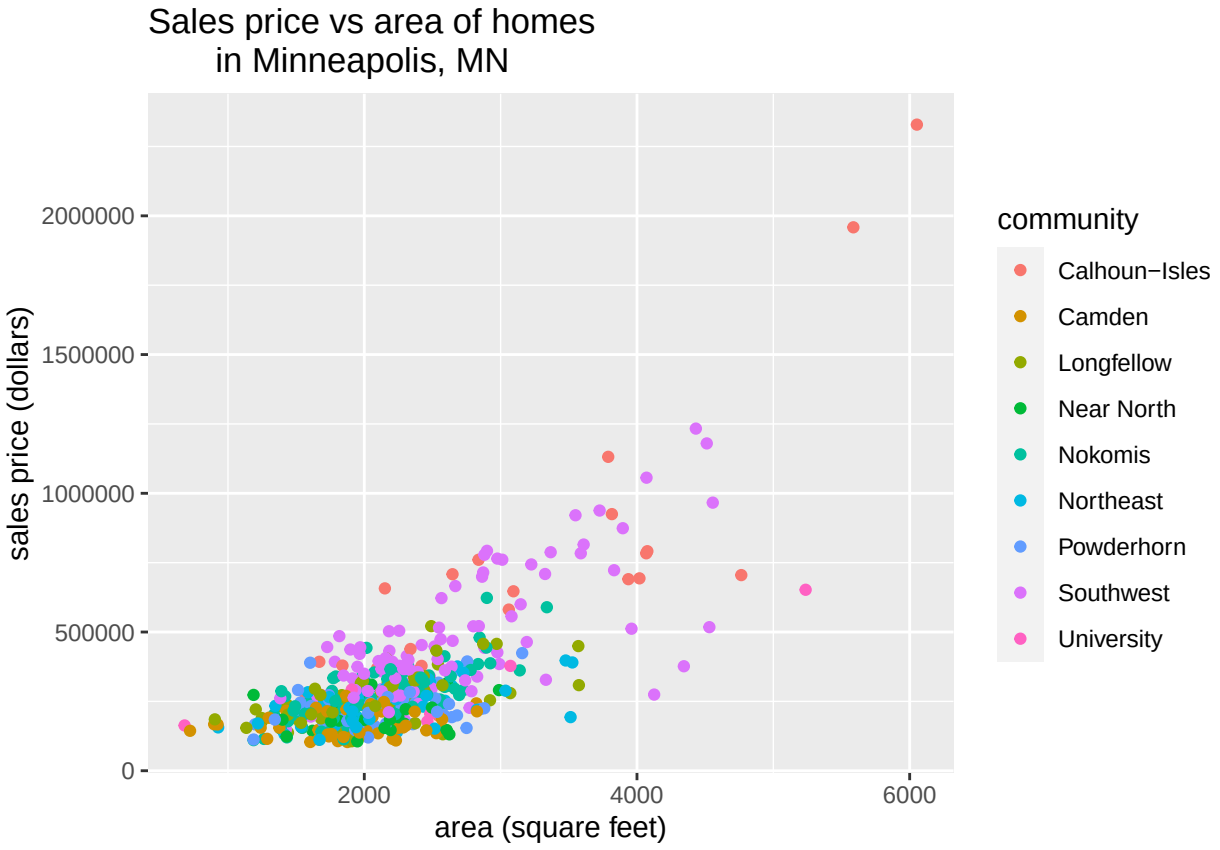
Create a scatterplot using variables of your choosing using the `mn_homes` data.

```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice)) +
  geom_point() +
  labs(title = "Sales price vs area of homes in
       Minneapolis, MN",
       x="area (square feet)",
       y="sales price (dollars)")
```



Modify your scatterplot above by coloring the points for each community.

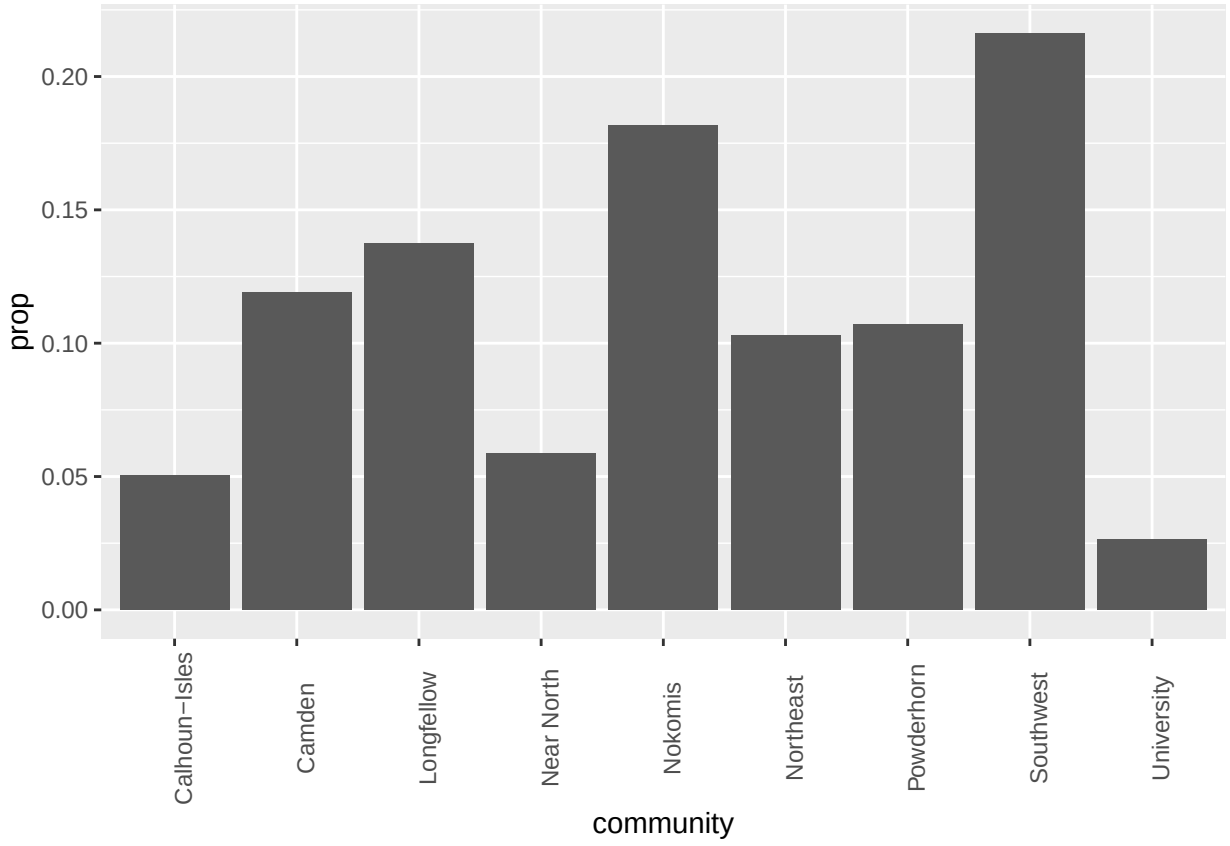
```
ggplot(data=mn_homes,
       mapping=aes(x=area, y=salesprice,
                   color=community)) +
  geom_point() +
  labs(title = "Sales price vs area of homes
       in Minneapolis, MN", x="area (square feet)",
       y="sales price (dollars)")
```



Bar

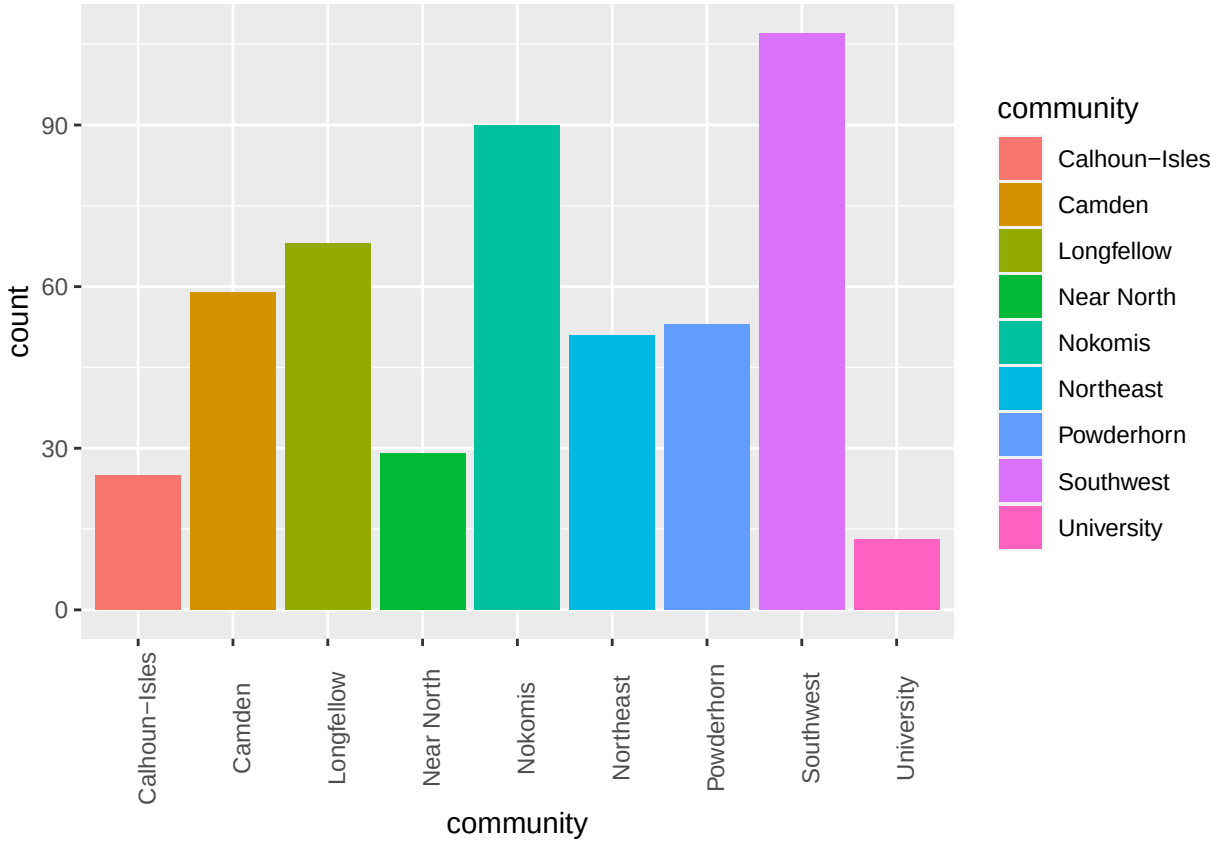
Another type of plotting is the bar plot, it used to visulaize categorical variables.

```
ggplot(data=mn_homes,
       mapping=aes(x=community,
                   y =after_stat(prop),
                   group=1)) +
  geom_bar()+
  theme(axis.text.x = element_text(angle = 90))
```



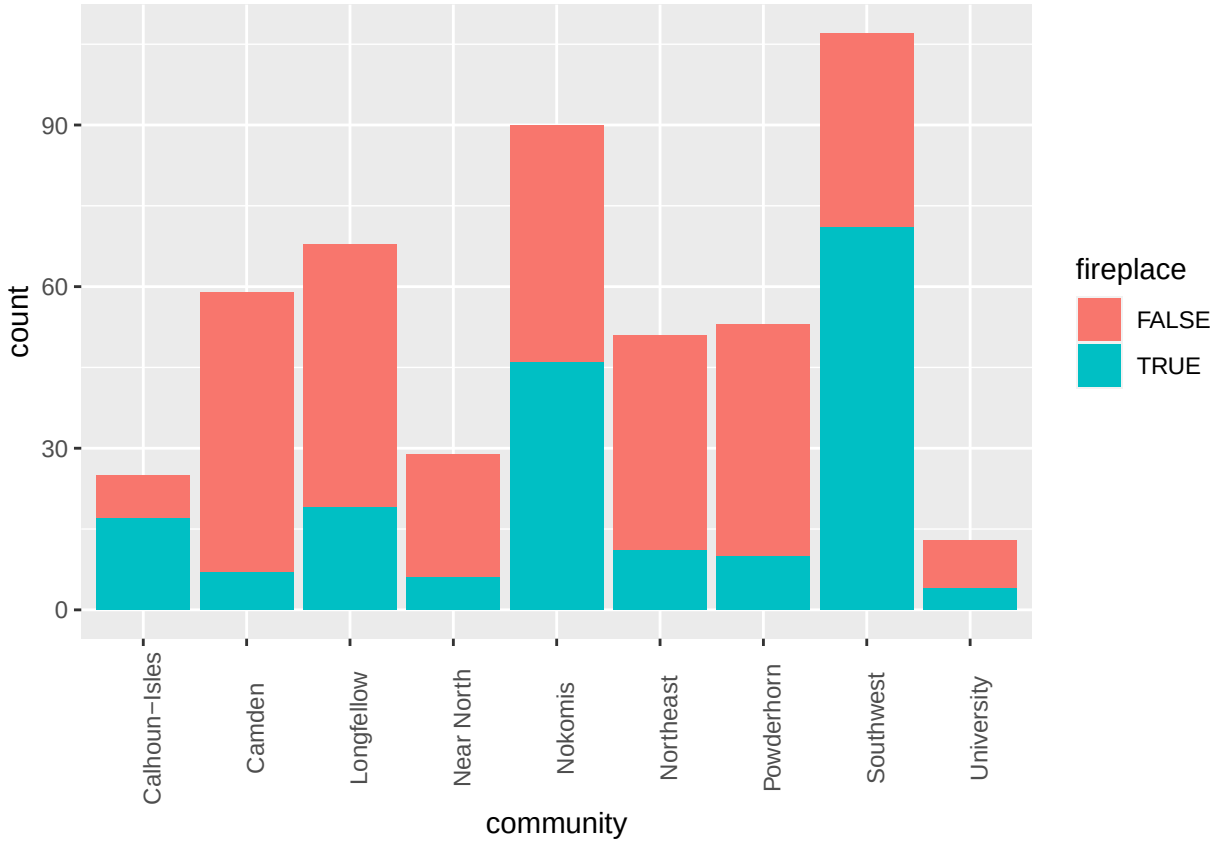
Coloring Bars

```
ggplot(data=mn_homes,
       mapping=aes(x=community, fill=community)) +
  geom_bar()+
  theme(axis.text.x = element_text(angle = 90))
```



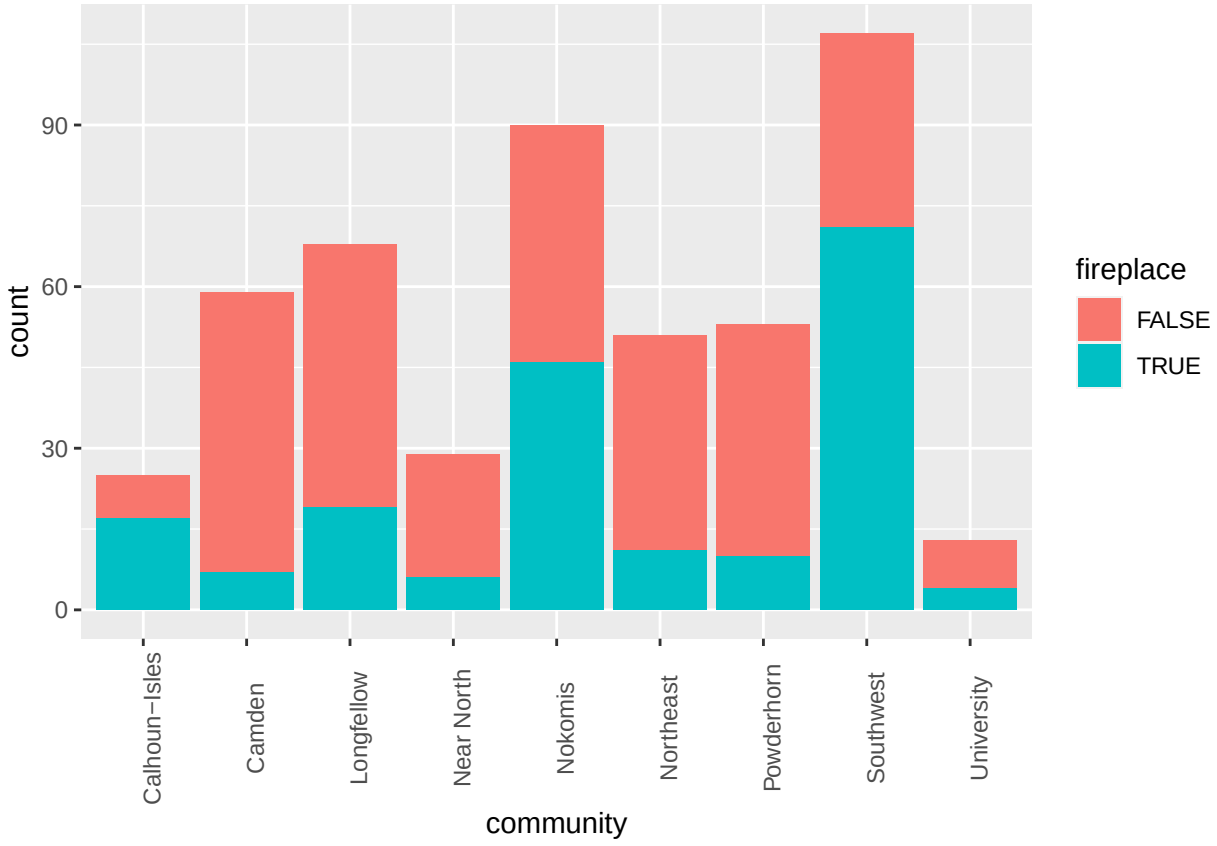
We can also color bars using some variables as the following

```
ggplot(data=mn_homes,
       mapping=aes(x=community, fill=fireplace)) +
  geom_bar()+
  theme(axis.text.x = element_text(angle = 90))
```



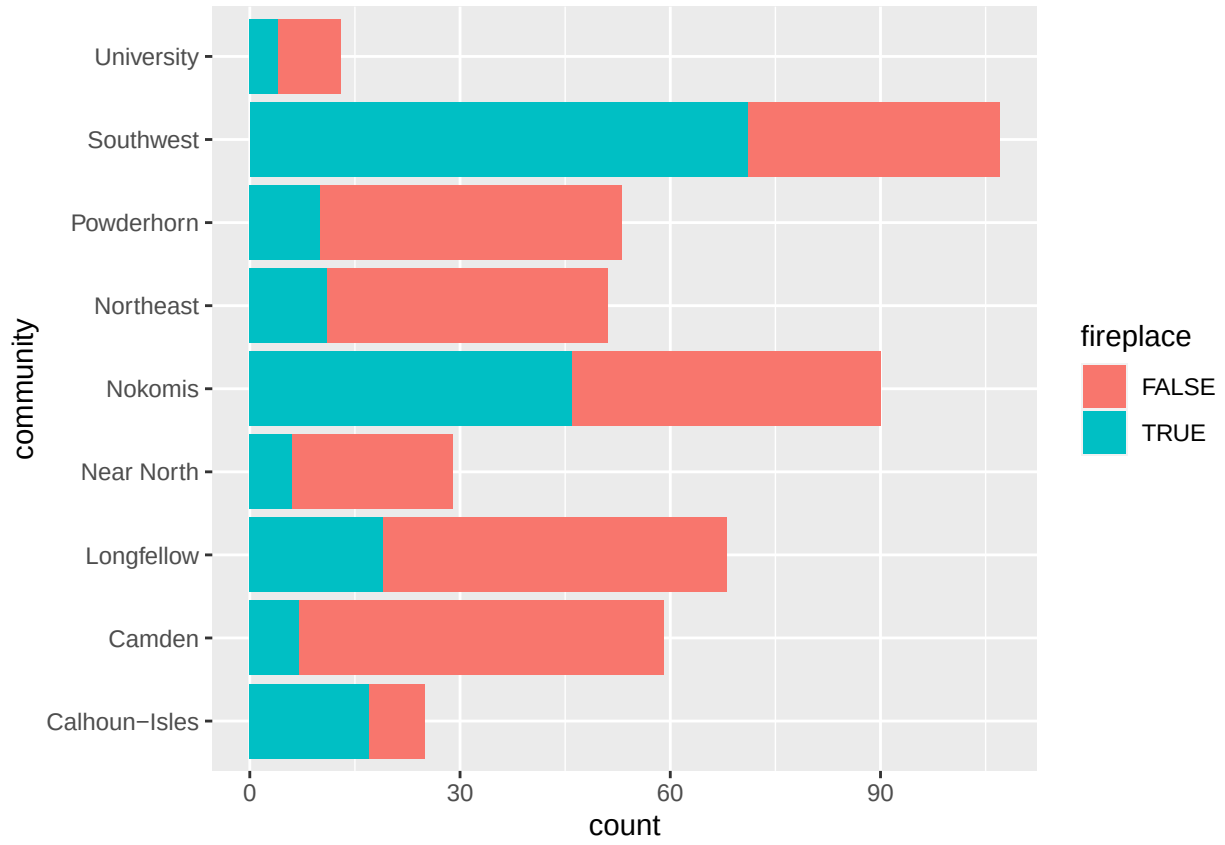
Sometimes we prefer to add the mapping parameter to the geom_chart function and not in the ggplot function in case we would like to plot more than one

```
ggplot(data=mn_homes) +
  geom_bar(mapping=aes(x=community, fill=fireplace))+
  theme(axis.text.x = element_text(angle = 90))
```



Flipping cordinates:

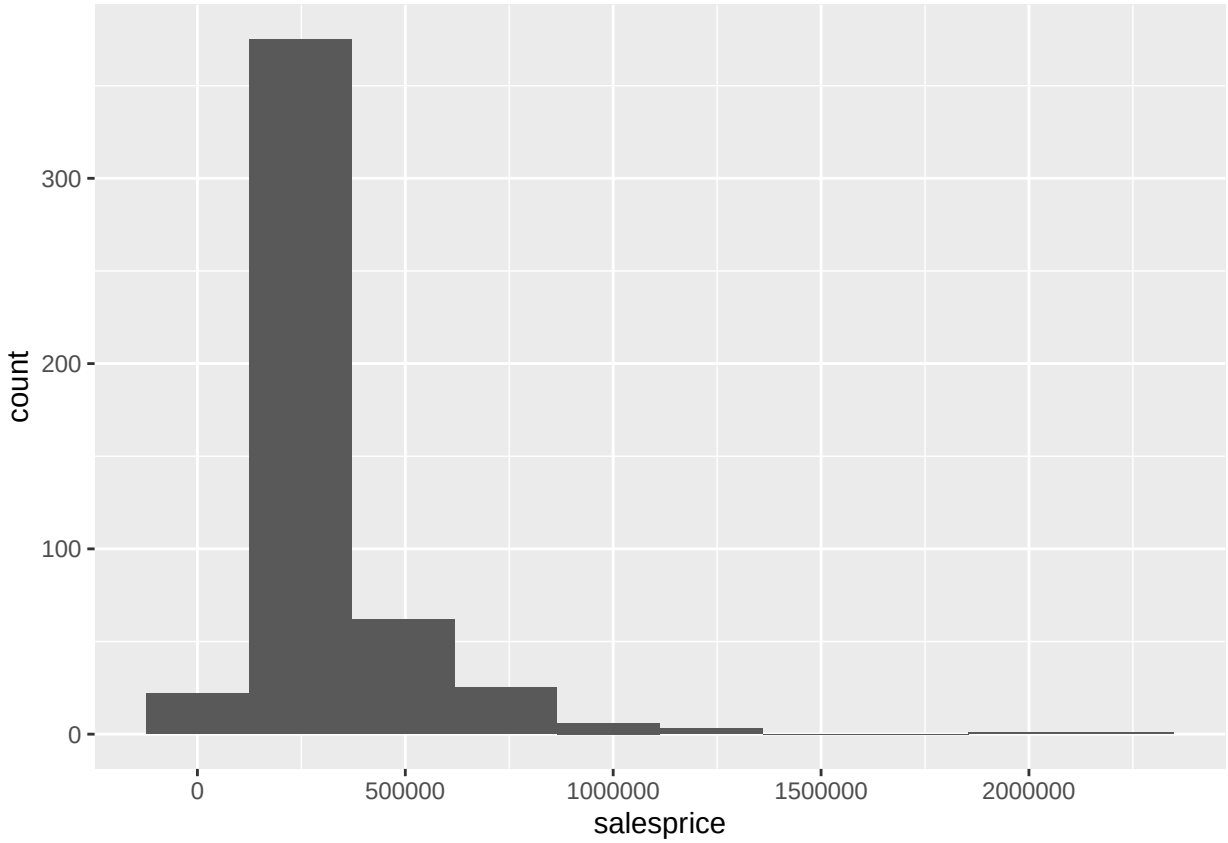
```
ggplot(data=mn_homes) +
  geom_bar(mapping=aes(x=community, fill=fireplace))+
  coord_flip()
```



Histogram

We can use histogram charts for continous variables as the following:

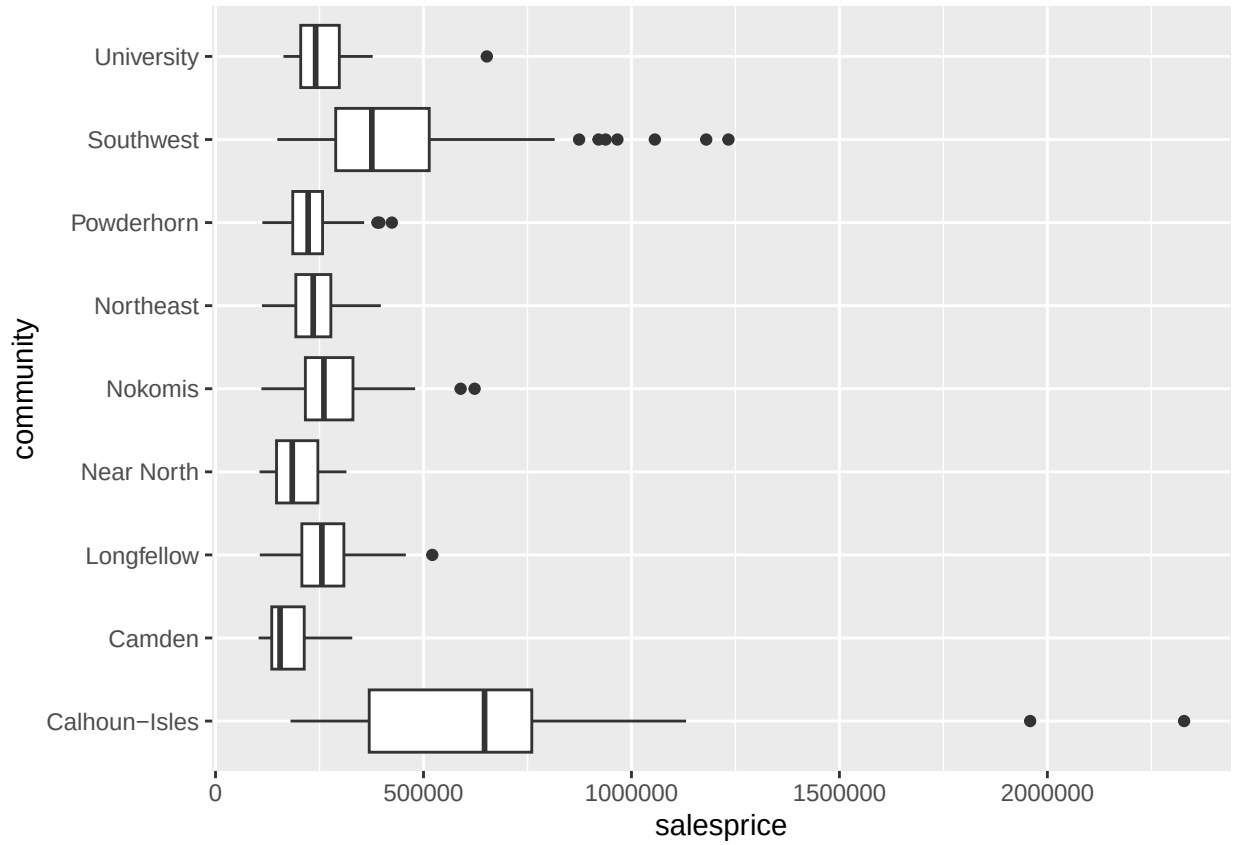
```
ggplot(data = mn_homes,
       mapping = aes(x=salesprice)) +
  geom_histogram(bins=10)
```



Boxplot

Boxplot charts are used also to represent numerical variables distribution across levels of categorical variables. For example, in the chart below we will show the distribuition of house prices across different communities.

```
ggplot(data=mn_homes, mapping=aes(x=community, y=salesprice)) + geom_boxplot() + coord_flip()
```



How to read it

box is the majority of values, thick line for average value, dots for outliers, line min and max value

Eval and Echo

=> Commands added to r: **{r: echo=TRUE/FALSE, eval=TRUE/FALSE}** => echo if true prints code in pdf if not does not print it => eval execute code if true and if false does not execute it (doesn't even check if its true) ## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

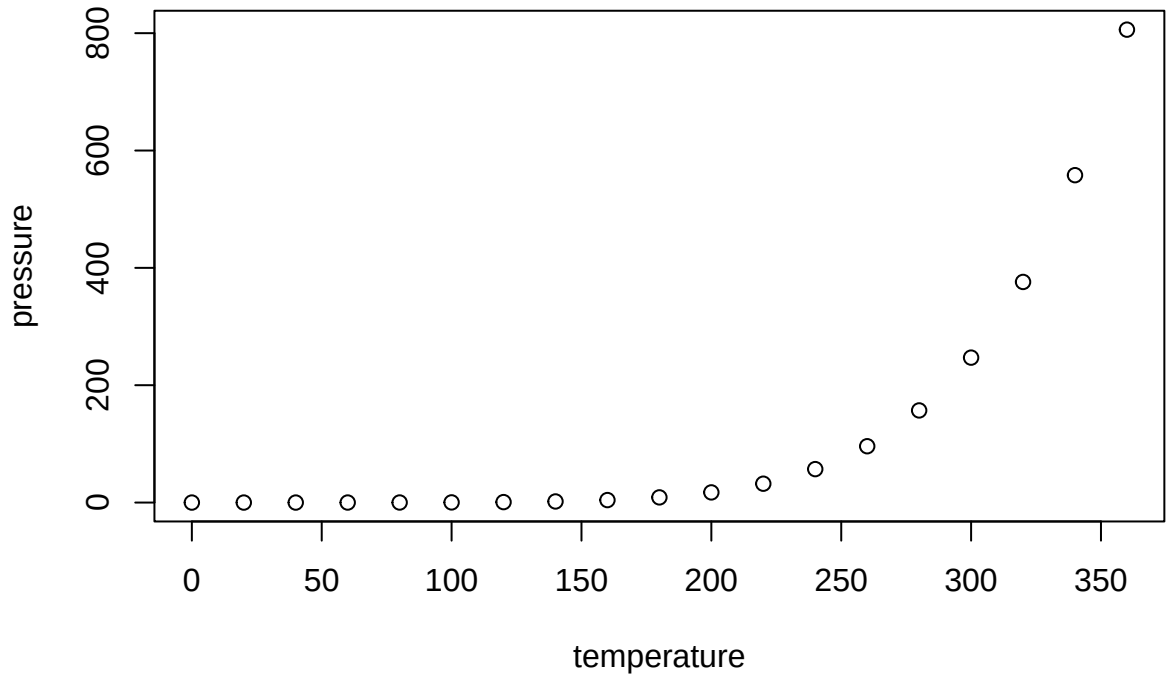
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.