dataVisulization

2023-02-14

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.1 ✔ purrr 1.0.1  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0  
## ✔ readr 2.1.4 ✔ forcats 1.0.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

mn\_homes <- read\_csv("Data/mn\_homes.csv")

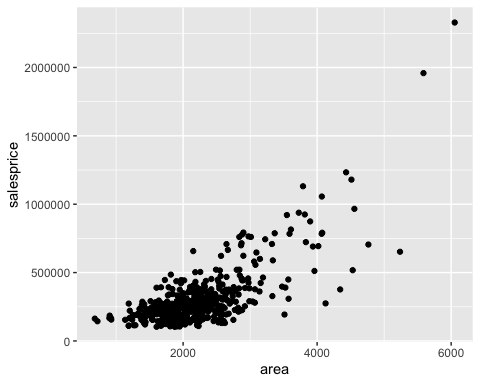
## Rows: 495 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): neighborhood, community  
## dbl (10): saleyear, salemonth, salesprice, area, beds, baths, stories, yearb...  
## lgl (1): fireplace  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(mn\_homes)

## Rows: 495  
## Columns: 13  
## $ saleyear <dbl> 2012, 2014, 2005, 2010, 2010, 2013, 2011, 2007, 2013, 20…  
## $ salemonth <dbl> 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 <dbl> 3937, 1440, 1835, 2016, 2004, 2822, 2882, 1979, 3140, 35…  
## $ beds <dbl> 5, 2, 2, 3, 3, 3, 4, 3, 4, 3, 3, 3, 2, 3, 3, 6, 2, 3, 2,…  
## $ baths <dbl> 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 <dbl> 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 <dbl> 6192, 5160, 5040, 4875, 5060, 6307, 6500, 5600, 6350, 75…  
## $ numfireplaces <dbl> 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…

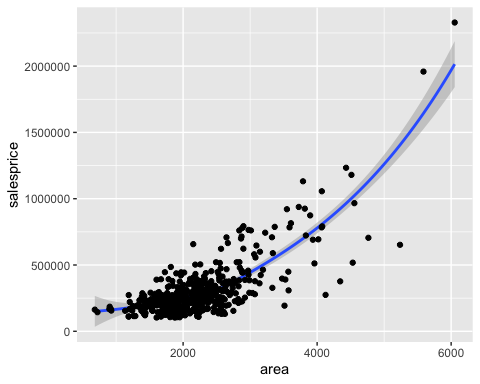
##First Graph We usually use ggplot function to draw plots, it has several components as the following: 1- data: Here we specify the dataFrame that we’ll use 2- mapping: Here you specify the x-axis and y-axis 3- Add drawing component 4- Add lables to the graph 5- Faceting (having different graphs based on different values of a specific variable)

ggplot(data=mn\_homes, mapping=aes(x=area, y=salesprice)) + geom\_point()

 As we see above, geom\_point() represents each record in the dataframe with a point in the plot. It usually helps to detect outliers.There are different types of drawing. For example, we can use geom\_smooth() to represent the relation between two variables with a curved 1ine as you can see below:

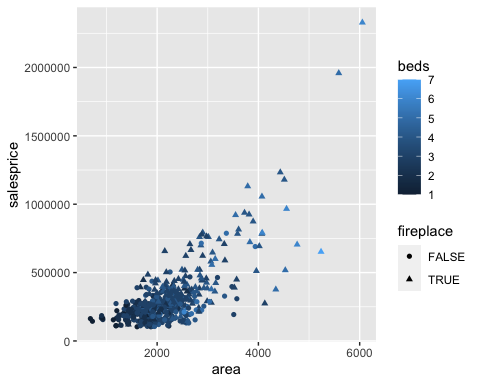
ggplot(data=mn\_homes, mapping=aes(x=area, y=salesprice)) + geom\_smooth() + geom\_point()

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

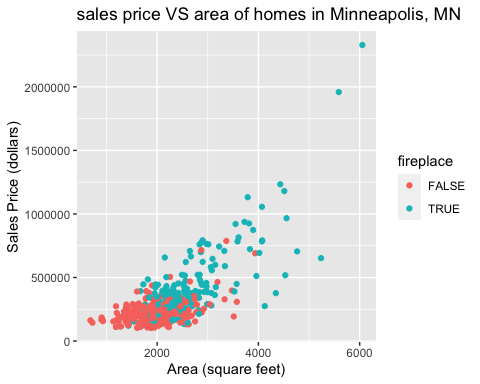
 As we see that we’re able to combine two function of geom() to see the graph

##aes() aes() method also allows us to change some visual properties in the plot including: shape, color, size, and transparency, for example, we can change the color of the points in the plot based on another variable (such as fireplace, beds, baths)

ggplot(data=mn\_homes, mapping=aes(x=area, y=salesprice, color=beds, shape=fireplace)) + geom\_point()

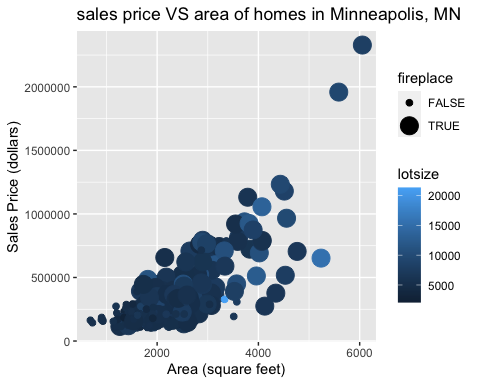
 We can also add lables to the graph using the labs() methods as the following:

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)")



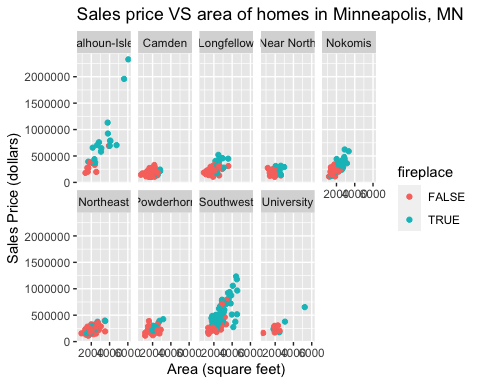
ggplot(data=mn\_homes, mapping=aes(x=area, y=salesprice, color=lotsize, size=fireplace)) + geom\_point() + labs(title="sales price VS area of homes in Minneapolis, MN", x="Area (square feet)", y="Sales Price (dollars)")

## Warning: Using size for a discrete variable is not advised.



##Faceting We can also have different graphs based on different values of a specific variable. For example, if we want make a similar graph to what we have above but for each community (community is a variable in our dataFrame), we can do the following:

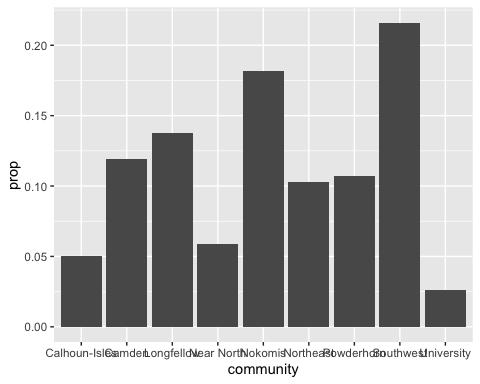
ggplot(data=mn\_homes, mapping=aes(x=area, y=salesprice, color=fireplace)) + geom\_point() + labs(title="Sales price VS area of homes in Minneapolis, MN", x="Area (square feet)", y="Sales Price (dollars)") + facet\_wrap(~community, nrow=2)



Another type of plotting is the bar plot.it’s use to visualize categorical variables

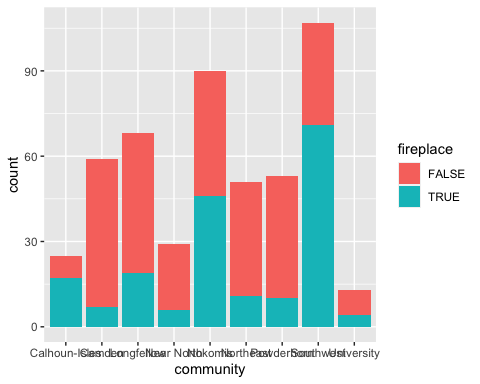
ggplot(data=mn\_homes, mapping=aes(x=community, y=..prop..,group=1)) + geom\_bar()

## Warning: The dot-dot notation (`..prop..`) was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(prop)` instead.



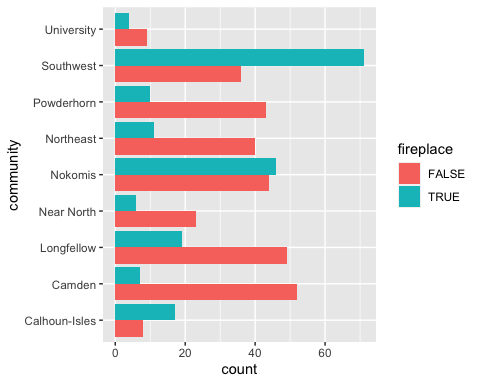
we can also color the bars using some variables as the following

ggplot(data=mn\_homes, mapping=aes(x=community, fill=fireplace)) + geom\_bar()



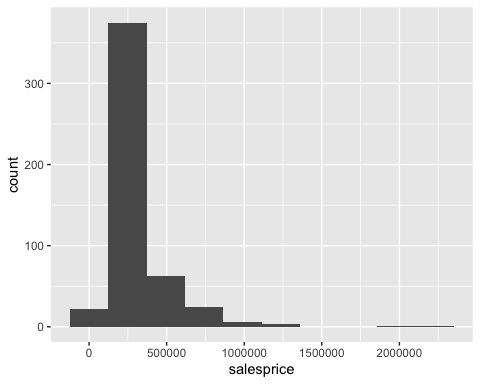
sometimes, we prefer to add the mapping parameter to the geom\_chart function and not in the ggplot function, also we can use the coord\_flip method to swap between the horizontal and the vertical coordinates if needed

ggplot(data=mn\_homes) + geom\_bar(mapping=aes(x=community, fill=fireplace), position="dodge") + coord\_flip()



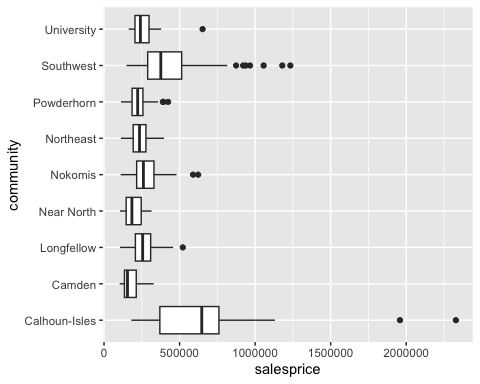
As we used bar chart for categorical variables, we can use histogram charts for continues variables as the following:

ggplot(data=mn\_homes, mapping=aes(x=salesprice)) + geom\_histogram(bins=10)



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

ggplot(data=mn\_homes, mapping=aes(x=community, y=salesprice)) + geom\_boxplot() + coord\_flip()



## 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.