# Practice Problems 3

Solutions

# Instructions

To complete your assignment, please follow these steps:

0. **Download R and RStudio (if you haven't already).** See the *Useful Resources* section on Moodle for installation links and videos with step-by-step installation instructions. If you're having issues with this, you'll need to reach out to your instructor and/or come to office hours, so plan accordingly!
1. **Download `practice3.Rmd` from Moodle and save it some place on your computer that you can easily find again.** I strongly encourage you to create a new folder dedicated to homework assignments for this class. See the *File Structure and Organization* video on Moodle for tips on how to do this.
2. **Make sure that the file you downloaded is called `practice3.Rmd` and not `practice3.Rmd.txt`.** The latter often happens when you use Safari on a Mac—try downloading the file using a different browser (e.g., Chrome) instead, or edit the file name (as explained in the *File Structure and Organization* video).
3. **Open `practice3.Rmd` in RStudio.** See the *Intro to R and RStudio*, *R Data Types*, and *R Error Messages and Troubleshooting* videos.
4. **Update the section number, author, and due date** on the third, fourth, and fifth lines of the file.
5. **Make sure you've already installed the `dplyr`, `readr`, and `ggplot2` packages.** Open the `Packages` panel (usually in the bottom right corner) to see the list of all packages that are already installed. Look to see if `dplyr`, `readr`, and `ggplot2` are listed here. If they're on this list, that means they're already installed and you're good to go. If any packages are missing from this list, type `install.packages('packagename')` in the *Console* (usually in the bottom left corner) and hit enter. See the *R Packages* video.
6. **Try *Knitting* your document:** click the `Knit` button at the top of this screen (look for yarn and needle). See the *Intro to RMarkdown* video. A dialogue box may pop up asking you if you want to install some packages: click "Yes." If you encounter any error messages, get in touch with your instructor or the preceptors.
7. **Answer the questions in Parts 1–3.** Click `Knit` occasionally along the way to make sure everything looks okay.
8. Once you're done with all parts, **click `Knit` one final time.** This will turn your R Markdown document into a nicely formatted HTML file.
9. **Look at the HTML file to make sure it looks like you want it to**: graphs appearing, no error messages, no data print out that lasts 50 pages, etc.
10. Once your HTML file pops up and you've checked that it looks like what you want, **click `Open in Browser` and then Print and select "Save as pdf".**
11. **Turn in two files to Moodle:** this `.Rmd` file, and the knitted `.pdf` version you generated in Step 10.

# Part 1. Saratoga Homes Data

Answer the following questions about the `homes.csv` dataset.

# Load Data in R

Enter code in the code chunk below to load the `homes.csv` dataset into R. Make sure that you've downloaded the `.csv` file from Moodle (see Activity 5) and it's saved in the same folder as this RMD. Don't forget to load any packages that you might need first!

```
# load package first (if needed)
library(readr)

# then read in dataset
homes <- read_csv('/Users/kgrinde/Documents/GitHub/stat155/activities/homes.csv')
```

```
## Rows: 1728 Columns: 16
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (16): Price, Lot.Size, Waterfront, Age, Land.Value, New.Construct, Centr...
```

```
##
## 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.
```

> Remember: when you read in your dataset, you can call it whatever you want! I called it "homes" here but you might have used a different name. That's okay! The key is to be consistent in using the same name throughout the same R Markdown document.

> Also note: if your CSV is not stored in the same folder as your RMD, then you can give R the full path to the file so it knows which folder to look in. In this case, my "homes.csv" file is stored in a folder I set up for in-class activities for this class. The long /file/path/ above tells R where to find that folder.

# Preparing and Cleaning Data

Enter code in the code chunk below to create new versions of the `Central.Air` and `New.Construct` variables that R will know are categorical. Call these new variables `Central.Air.Cat` and `New.Construct.Cat`. Don't forget to load any packages that you might need first! (*Hint: see Activity 5!*)

```
# load packages first (if needed)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# then create new categorical variables
homes <- homes %>%
  mutate(Central.Air.Cat = factor(Central.Air),
         New.Construct.Cat = factor(New.Construct))
```

We need to load the `dplyr` package in order to use the `%>%` and `mutate` functions.

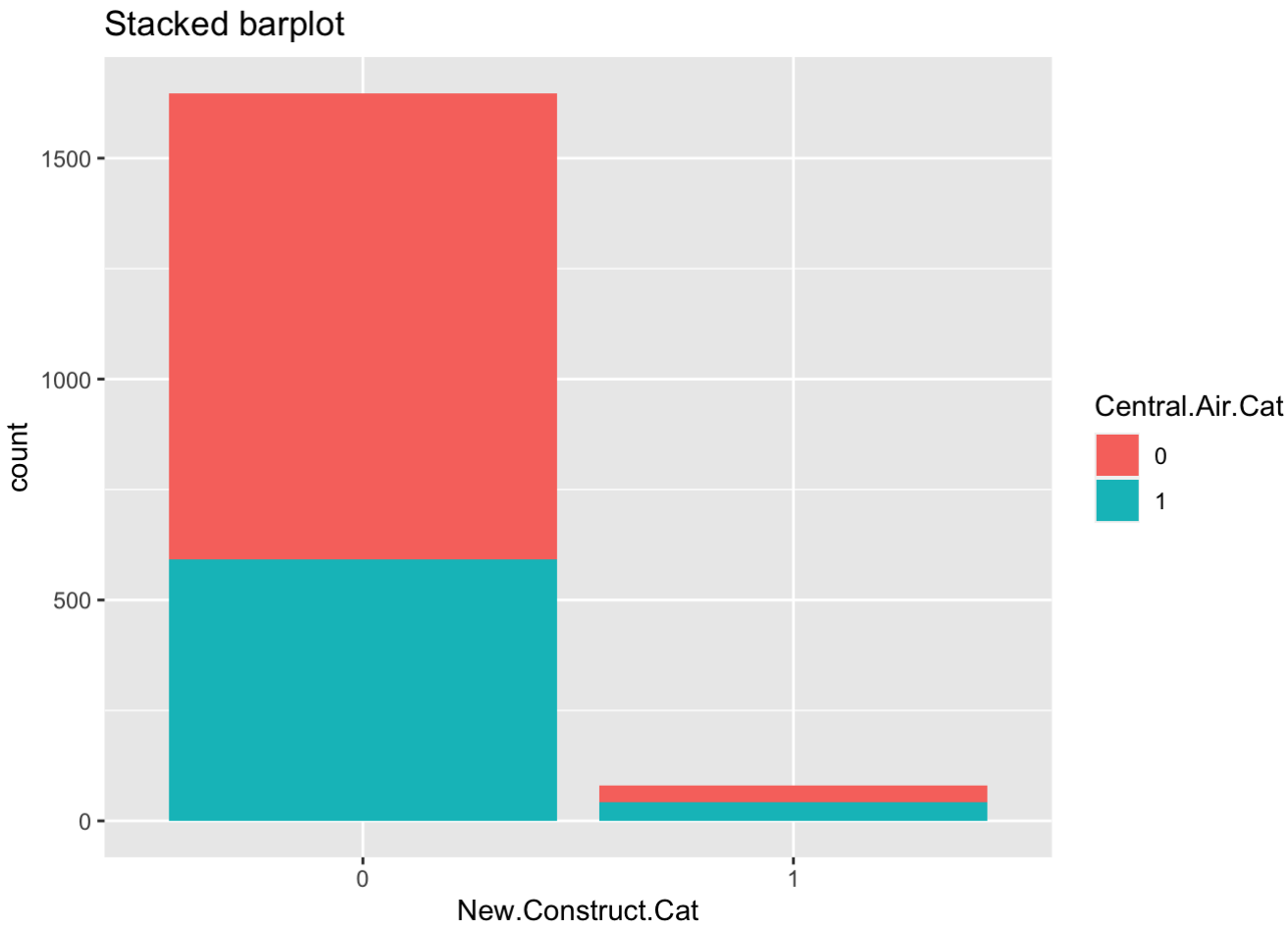
The `factor` function is what tells R to treat a variable as categorical. We use the `mutate` function to add two new variables to our dataset, called `Central.Air.Cat` and `New.Construct.Cat`, which are created by turning the original `Central.Air` and `New.Construct` variables into factors.

# Data Viz - Two Categorical Variables

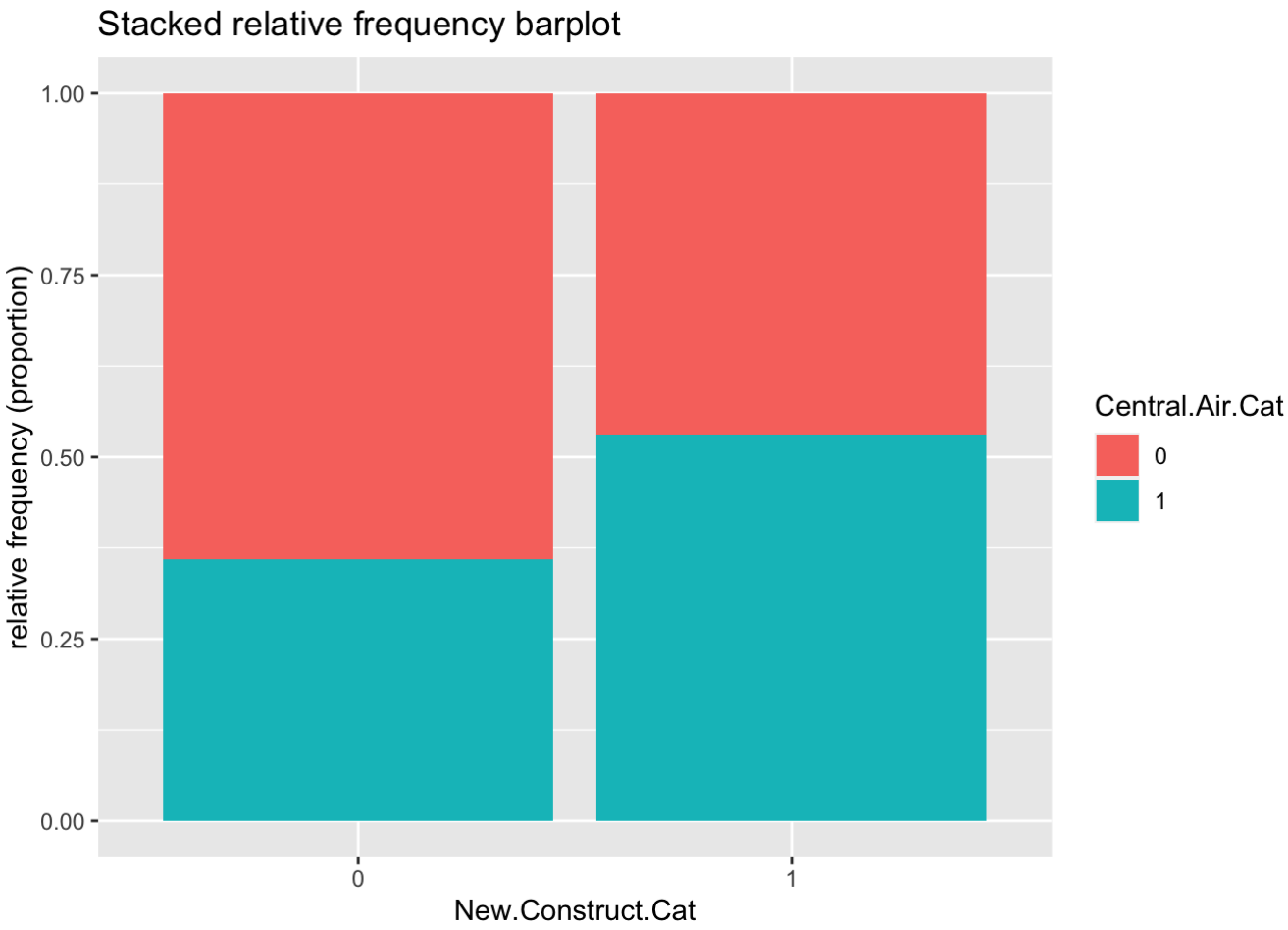
Create one graph that summarizes the relationship between `Central.Air.Cat` and `New.Construct.Cat`. Then, write up a short paragraph describing what you learn from that data visualization. Write this summary using good sentences that tell a story and do not resemble a checklist. Don't forget to consider the context of how the data were collected.

There are multiple options of graphs we could make here! In all cases, we need to make sure we load the `ggplot2` package before we can use the `ggplot()` and related `geom` functions. In order to make a mosaic plot, we also need to load the `ggmosaic` package.

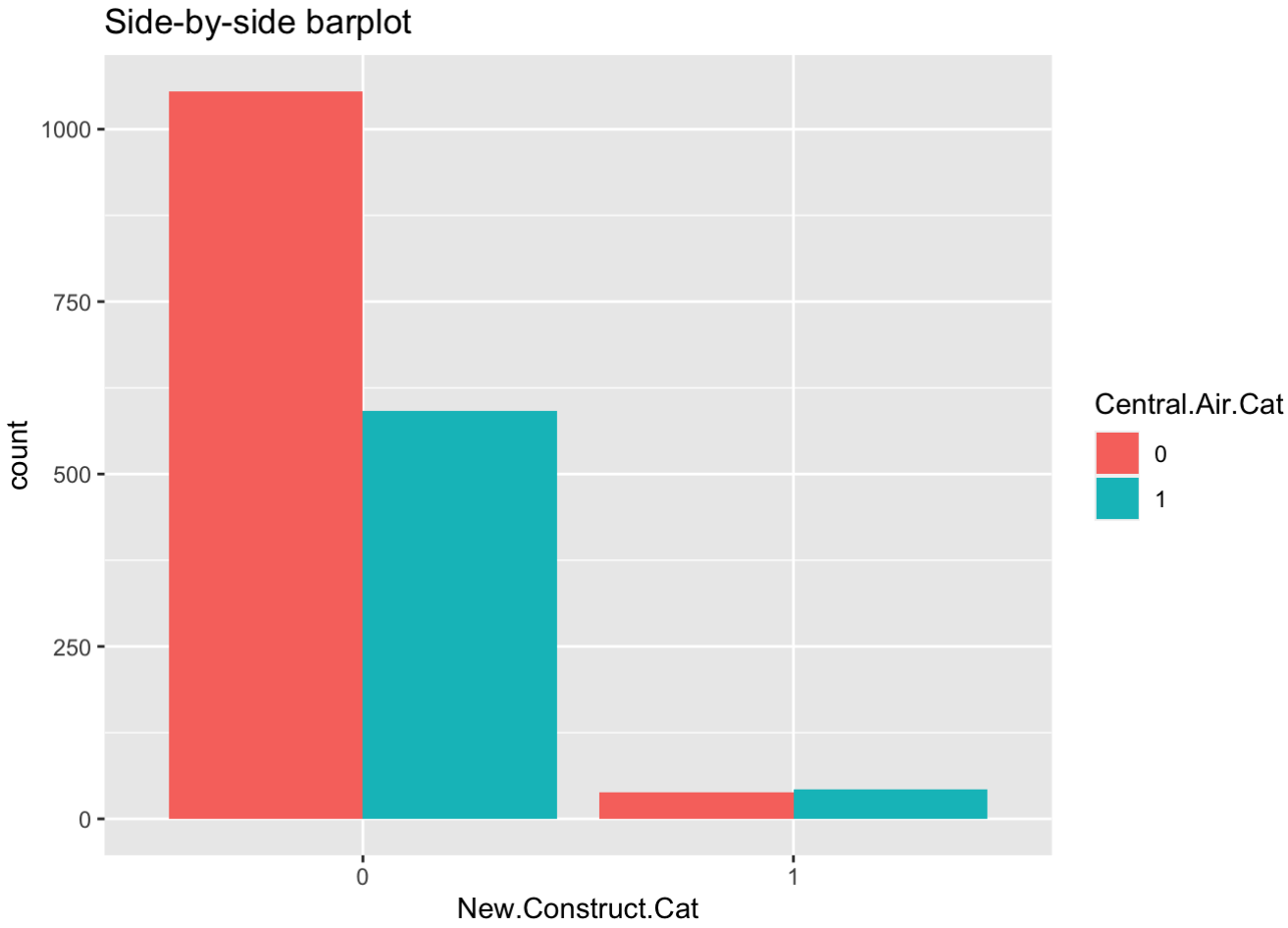
```
# load packages first
library(ggplot2)

# option 1: stacked barplot
homes %>%
  ggplot(aes(x = New.Construct.Cat, fill = Central.Air.Cat)) +
  geom_bar() + # this line says to make a stacked barplot
  ggtitle('Stacked barplot')
```



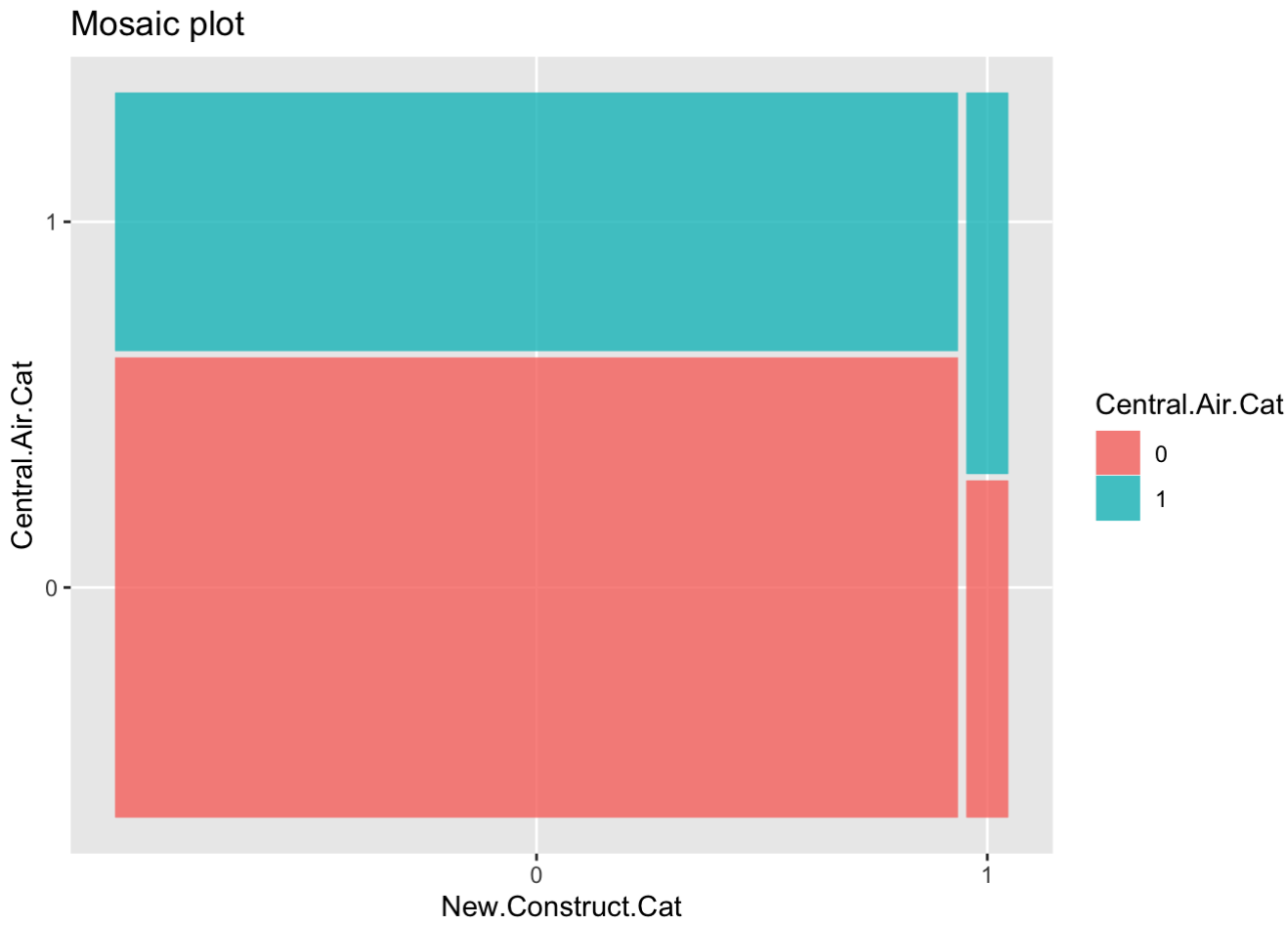

```
# option 2: stacked relative frequency barplot
homes %>%
  ggplot(aes(x = New.Construct.Cat, fill = Central.Air.Cat)) +
  geom_bar(position = 'fill') + # this line says to make a stacked relative frequency ba
rplot
  ggtitle('Stacked relative frequency barplot') +
  ylab('relative frequency (proportion)')
```




```
# option 3: side-by-side barplot
homes %>%
  ggplot(aes(x = New.Construct.Cat, fill = Central.Air.Cat)) +
  geom_bar(position = 'dodge') + # this line says to make a side-by-side barplot
  ggtitle('Side-by-side barplot')
```




```
# option 4: mosaic plot
library(ggmosaic)
homes %>%
  ggplot() +
  geom_mosaic(aes(x = product(Central.Air.Cat, New.Construct.Cat), fill = Central.Air.Ca
t)) +
  ggtitle('Mosaic plot')
```


Mosaic plot
Central.Air.Cat
1
0
0
1
New.Construct.Cat
Central.Air.Cat
0
1

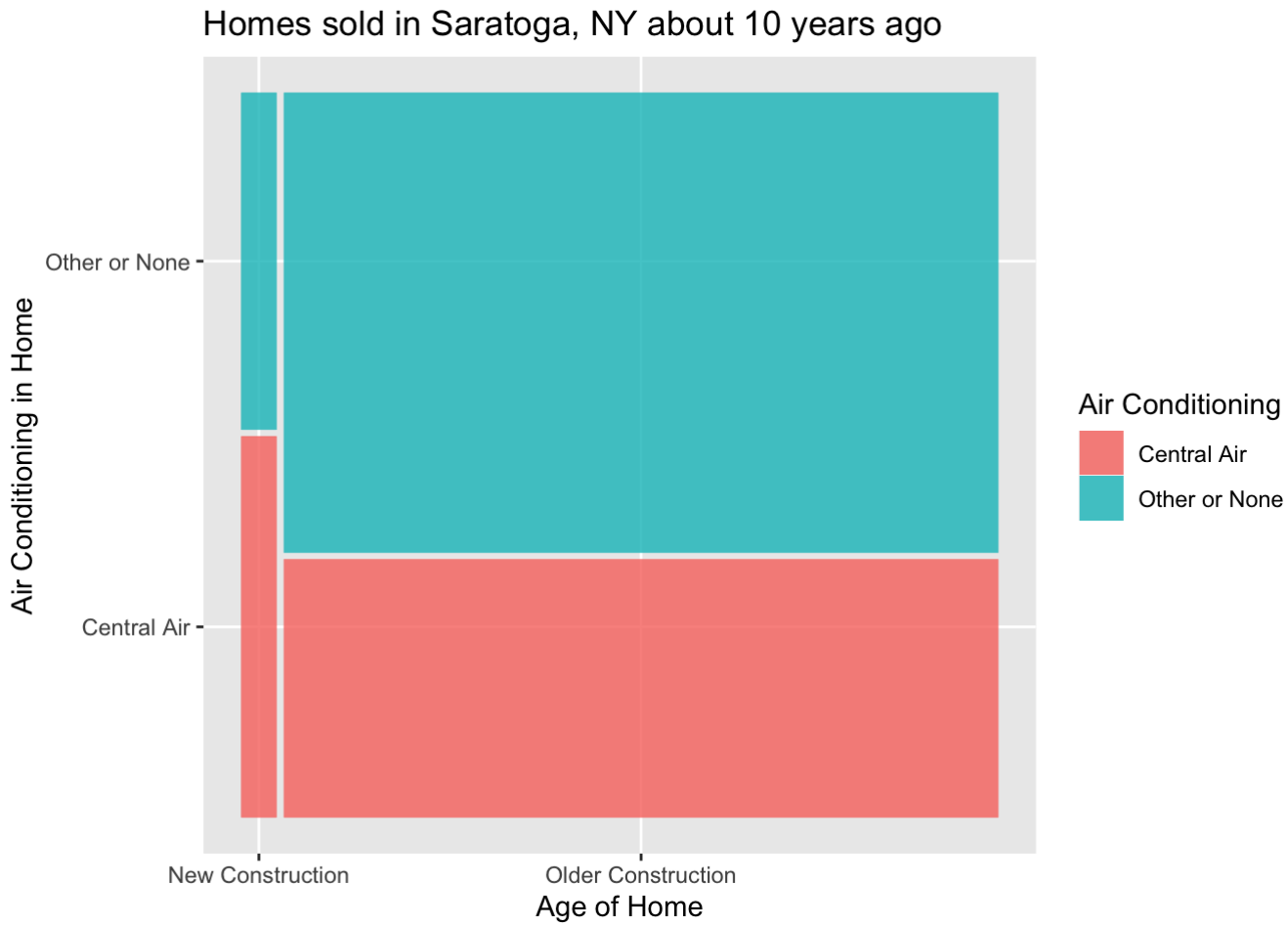
Your summary will likely vary based on which type of plot you decided to make. In any case, you'll want to make sure you commented on the *relationship* between these two variables, since that was the goal of making this plot. Is central air more common in new construction homes, or older homes? (In my opinion, it is easiest to answer this question by looking at either the stacked relative frequency barplot or the mosaic plot: just over half of new construction homes have central air, whereas only about a third of older homes in this area have central air.)

It would also be good to comment on which type of home is more common: new or old construction. We can learn this from the stacked frequency barplot, side-by-side barplot, or mosaic plot, but unfortunately this is *not* something we can tell from the stacked *relative frequency* barplot (option 2 above). In the other three plots it is apparent that there are far more older homes in this area than there are new construction homes.

Otherwise, make sure that your description provides relevant contextual details (e.g., homes in Saratoga, NY about 10 years ago) and talks about the variables in common terms rather than using dataset-specific jargon (e.g., "new construction" rather than "New.Construct.Cat"). Try to write in a way that has a flowing narrative and doesn't just sound like you're listing off details about the plot.

Finally, note that all of the visualizations above have room for improvement! In particular, I think it would be helpful to make the axis labels and color legend more informative, and to add a title providing some context on what the cases in this dataset represent. Here's some code that makes that updates for the mosaic plot:

```
homes %>%
  # give the categories more informative labels
  mutate(Central.Air.Cat = ifelse(Central.Air.Cat == "1", "Central Air", "Other or None"
),
         New.Construct.Cat = ifelse(New.Construct.Cat == "1", "New Construction", "Older
Construction")) %>%
  # make mosaic plot
  ggplot() +
  geom_mosaic(aes(x = product(Central.Air.Cat, New.Construct.Cat), fill = Central.Air.Ca
t)) +
  # add axis and color legend label
  labs(x = 'Age of Home', y = 'Air Conditioning in Home', fill = 'Air Conditioning') +
  # title with data context details
  ggtitle('Homes sold in Saratoga, NY about 10 years ago')
```

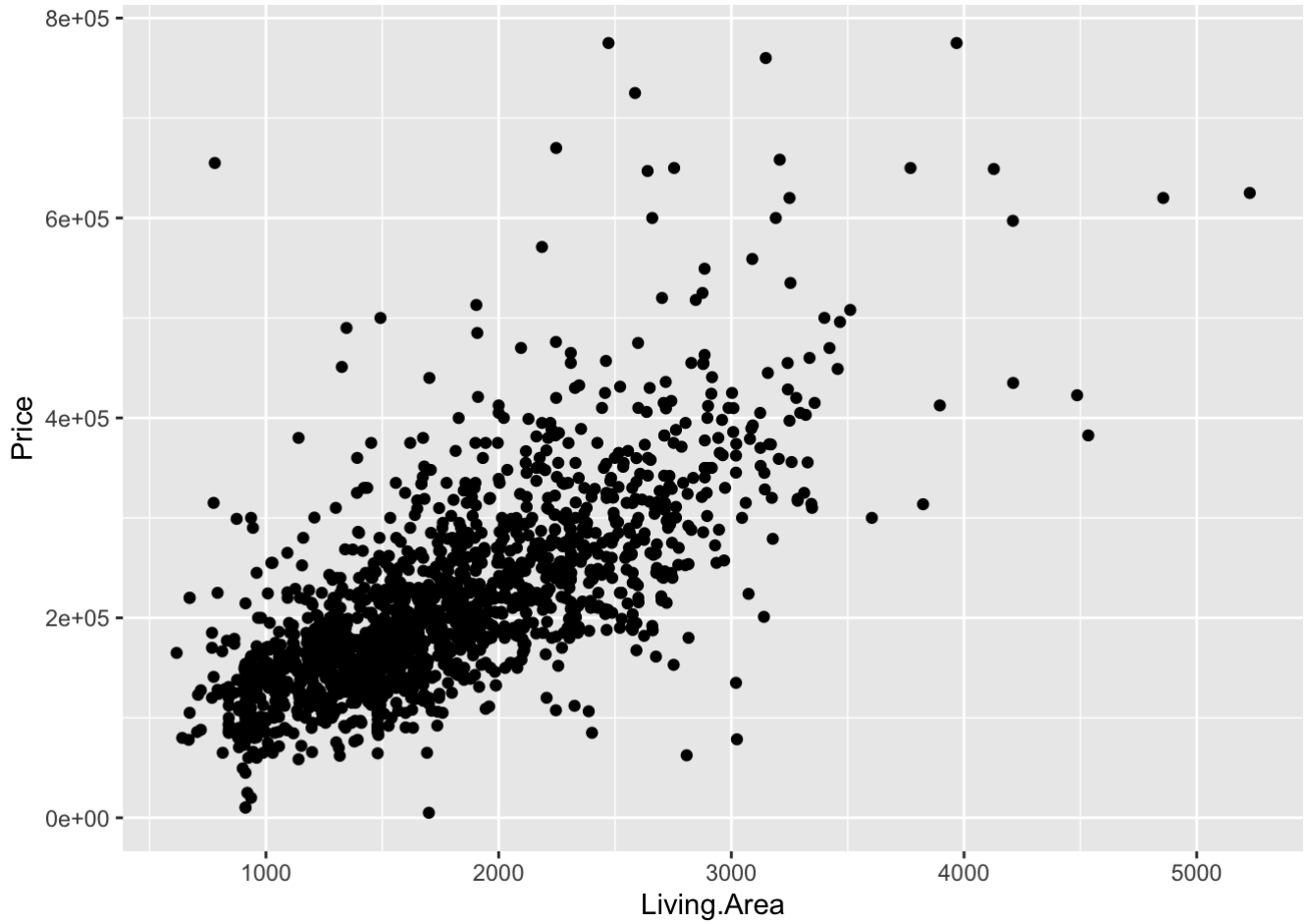



# Data Viz - Two Quantitative Variables

Create one graph and calculate one relevant numerical summary to summarize the relationship between `Living.Area` and `Price`. Then, write up a short paragraph describing what you learn from those summaries. Write this summary using good sentences that tell a story and do not resemble a checklist. Don't forget to consider the context of how the data were collected.

> When comparing two quantitative variables, we have one main option for graphical summary (a scatterplot) and one main option for numerical summary (correlation). Your code should look something like I have below:

```
# graphical summary (scatterplot)
homes %>%
  ggplot(aes(x = Living.Area, y = Price)) +
  geom_point()
```




```
# numerical summary (correlation)
homes %>%
  summarize(cor(Living.Area, Price))
```

```
## # A tibble: 1 × 1
##   `cor(Living.Area, Price)`
##                       <dbl>
## 1                     0.712
```

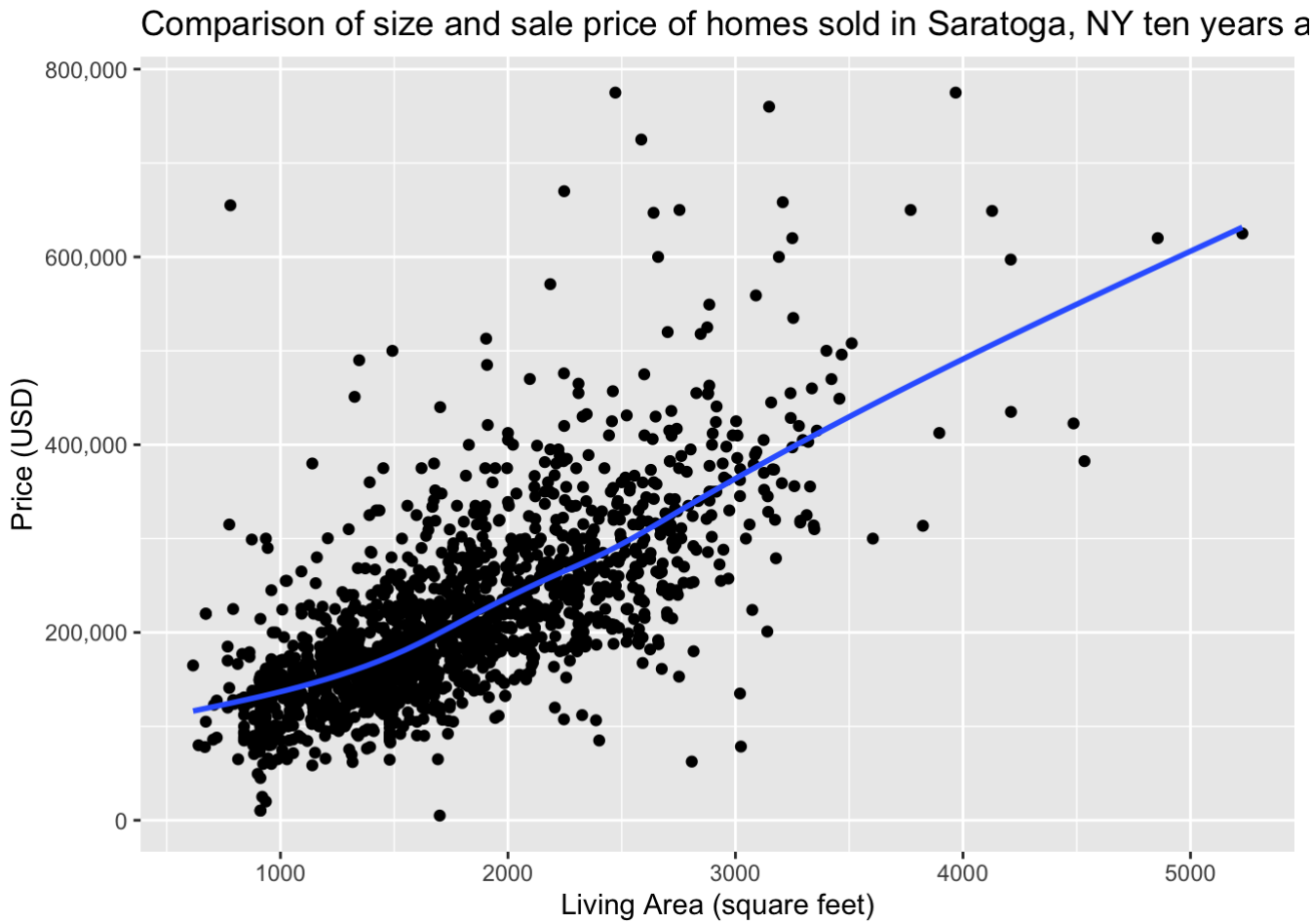
Living area and home price appear to be strongly related (as we might have suspected even before looking at these data!). We see a moderately strong positive linear relationship (correlation = 0.71) between home size and price. There is one outlier that might be interesting to explore further: the roughly 750 square foot house that sold for about $650,000. It deviates from the overall trend noticeably and is worth looking at in more detail: *why* is this house so expensive for its size? Otherwise the trends we observe in these data largely match our prior expectation—larger houses tend to cost more.

Notice that the description above comments on direction (positive), form (linear), strength (moderately strong, including the correlation as numerical evidence), and unusual features (outlier). I provided relevant contextual details, including the units of each variable, and talked about each variable "in words" rather than using their names from the dataset (i.e., I talked about "living area" and "home price" rather than "Living.Area" and "Price"). I also reflected on whether the patterns we observe in these data match our expectation, and concluded with a clear takeaway message about what we learn from this plot: larger houses tend to cost more.

As before, there are multiple improvements that we can make to this plot. Here are a few:

```
# graphical summary (scatterplot)
homes %>%
  ggplot(aes(x = Living.Area, y = Price)) +
  geom_point() +
  scale_y_continuous(labels = scales::comma) + # get rid of R's non-standard scientific
 notation
  labs(x = 'Living Area (square feet)', y = 'Price (USD)') + # update axis labels
  geom_smooth(se = F) + # add a trend line
  ggtitle('Comparison of size and sale price of homes sold in Saratoga, NY ten years ag
o') # add title with context
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




# Part 2. Project Data

Answer the following questions about your chosen project dataset.

# Load Data in R

Look through the information for the possible project data sets on Moodle. Choose one data set to work with (or get approval from Kelsey to use a different dataset). Then, use the code chunk provided below to read your chosen data set into R. Don't forget to load any packages you might need (unless you've already done so earlier in this RMD).

Note: you can use a different dataset for this assignment than you did last week, but moving forward you'll stick with the same data for the rest of the semester. Make sure to choose something that seems interesting to you!

> If you're using the NHANES data for your project, your code should look something like this:

```
# load package first (if needed)
library(NHANES)

# then read in dataset
data(NHANES)
```

Otherwise, you're likely using the `read_csv` function to read in your data from a CSV file. Since we already loaded the `readr` package in an earlier code chunk, we don't need to do it again here:

```
# read in dataset
project <- read_csv('projectdata.csv')
```

Note: in the example above, my code reads in a dataset called "projectdata.csv" and saves it as an object called `project`. You can give your data whatever* name you like! Just make sure to use the same name throughout. For example, if you named your data `kiva` (`kiva <- read_csv('...')`) then run the `ggplot` function on "kiva" as well (`kiva %>% ggplot(...)` in the next question).

*there are a few rules to naming objects in R. Names can't start with a number, can't include some special characters like @, /, &, or spaces, and shouldn't be the name of something that already exists (e.g., don't name your data "read_csv"). I recommend using a name that's short and informative, so it's easy to type and easy to remember what you called it.

Another note: the `read_csv()` function prints out quite a bit of information when it loads in the dataset. You can hide these messages by adding `message = FALSE` to the first line of your code chunk. It should like something like this:
`{r name-of-chunk, message = FALSE}`.

# Data Viz - One Quantitative and One Categorical

Pick one quantitative variable and one categorical variable from your dataset that you would like to explore. Produce a data visualization, and calculate relevant numerical summaries, to summarize the relationship between those two variables. Then, write up a short paragraph describing what you learn from those summaries. Write this summary using good sentences that tell a story and do not resemble a checklist. Don't forget to consider the context of how the data were collected.

You have multiple options for graphical and numerical summaries when comparing a quantitative and categorical variable.

In terms of graphical summaries, you could make side-by-side histograms, side-by-side boxplots, or side-by-side density plots. See the code below.

For numerical summaries, I would recommend calculating at least one measure of the *center* (e.g., mean, median) and at least one measure of the *spread* (e.g., standard deviation, IQR, range) for your quantitative variable, stratified by each category of the categorical variable. We use the `group_by` function to stratify by the categorical variable, and then the `summarize` function to calculate those measures of center and spread on the quantitative variable.

```
# graphical summary: side-by-side histograms
project %>%
  ggplot(aes(x = QuantVar)) +
  geom_histogram() +
  facet_grid(CatVar ~ .)

# graphical summary: side-by-side boxplots
project %>%
  ggplot(aes(x = CatVar, y = QuantVar)) +
  geom_boxplot()

# graphical summary: side-by-side densities
project %>%
  ggplot(aes(x = QuantVar, fill = CatVar)) +
  geom_density(alpha = 0.5)

# numerical summary: stratified mean, median, sd, and IQR (don't need all)
project %>%
  group_by(CatVar) %>%
  summarize(mean = mean(QuantVar),
            med = median(QuantVar),
            sd = sd(QuantVar),
            iqr = IQR(QuantVar))
```

Note: if you have any missing values in your quantitative variable, you'll need to filter those first!

```
# numerical summary: stratified mean, median, sd, and IQR (don't need all)
# (after first filtering out missing values of QuantVar)
project %>%
  filter(!is.na(QuantVar)) %>% # filter out cases missing QuantVar
  group_by(CatVar) %>%
  summarize(mean = mean(QuantVar),
            med = median(QuantVar),
            sd = sd(QuantVar),
            iqr = IQR(QuantVar))
```

Your written description should comment on any similarities/differences in the shape, center, spread, and unusual features of the distribution of your quantitative variable between the categories of your categorical variable. For example, does one group have a higher mean/median? Is one group more spread out? Does one group have more outliers or cases? After reflecting on these details, make sure to provide a takeaway message about what this suggests about if/how these two variables seem to be related. Do all of this while writing about the variables *in context* (reminding us what the cases are, providing units for the quantitative variable, etc.) and avoiding any dataset-specific jargon (writing out the names of variables and categories in common terms rather than just using the way they are named in the dataset, etc.). The key is to make sure this summary is accessible to a wide audience and clearly communicates a story about how these two variables are related.

# Ethical Considerations

For your chosen data set, consider who benefits and who may be harmed from this data being collected, this data being publicly available, and the potential ways you might use (and/or already have used) the data.

Benefits: comment on who benefits from this data being collected, who benefits from this data being publicly available, who benefits from the potential ways you might use (and/or have already used) the data

Harms: comment on who may be harmed from this data being collected, who may be harmed from this data being publicly available, who may be harmed from the potential ways you might use (and/or have already used) the data

# Part 3. Knit

Once you've answered all the questions above, you'll need to `Knit` this document and then save that knitted document as a PDF. See the assignment *Instructions* at the top of this document and on Moodle for more details.