athota_hw3

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

Problem 1: Loading and cleaning (25 points)

a. Load the data into a dataframe called `ca_pa`. (Hint: one way is to use `read.csv()`.)

```
#collabarated with Omkar kulkarni
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(dplyr)
ca_pa = read.csv("calif_penn_2011.csv")#reads the data
```

b. How many rows and columns does the dataframe have?

```r
ncol(ca_pa)#outputs number of columns for the dataframe
```

```
[1] 34
```

```r
nrow(ca_pa)#ouytputs number of rows for the dataframe
```

```
[1] 11275
```

c. Run this command, and explain, in words, what it does:

```r
colSums(is.na(ca_pa))#gives the sum of each columns without null values
```

d. Remove any row containing an NA value. There are many ways to do this; one possibility is using the function `na.omit()`, which takes a dataframe and returns a new dataframe, omitting any row containing an NA value. You may also use `dplyr` operations.

```r
new_ca_pa <- na.omit(ca_pa)#creates new dataframe without null values
```

e. How many rows did (d) eliminate?

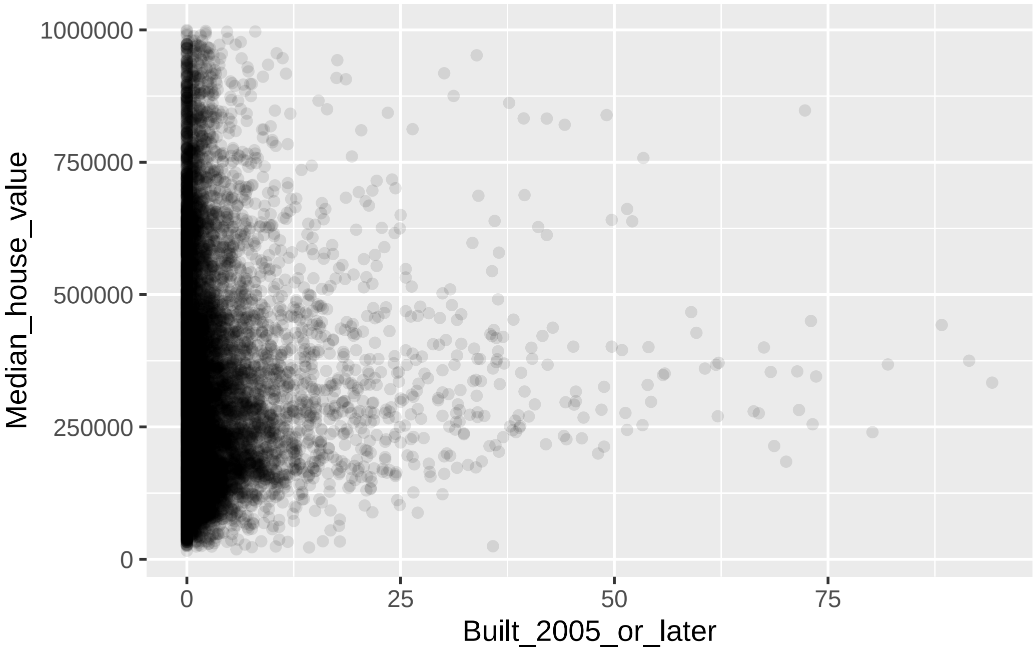
```r
nrow(ca_pa)-nrow(new_ca_pa)#outputs the nuumber of rows elimanated by substracting
```

```
[1] 670
```

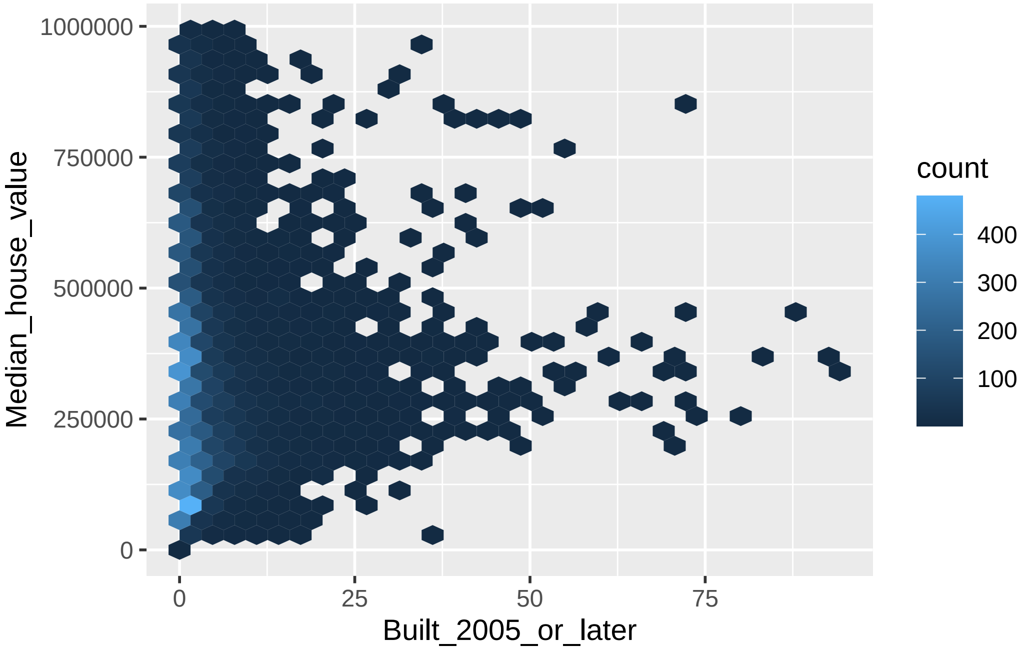
Problem 2: This Very New House (25 points)

a. The variable `Built_2005_or_later` indicates the percentage of houses in each Census tract built since 2005. Plot `Median_house_value` against this variable (`Median_house_value` should be on the y-axis). Is there overplotting? How can you improve on this scatterplot? Produce this plot.

```
new_ca_pa %>%
  ggplot(mapping = aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_point(alpha = 0.1)
```

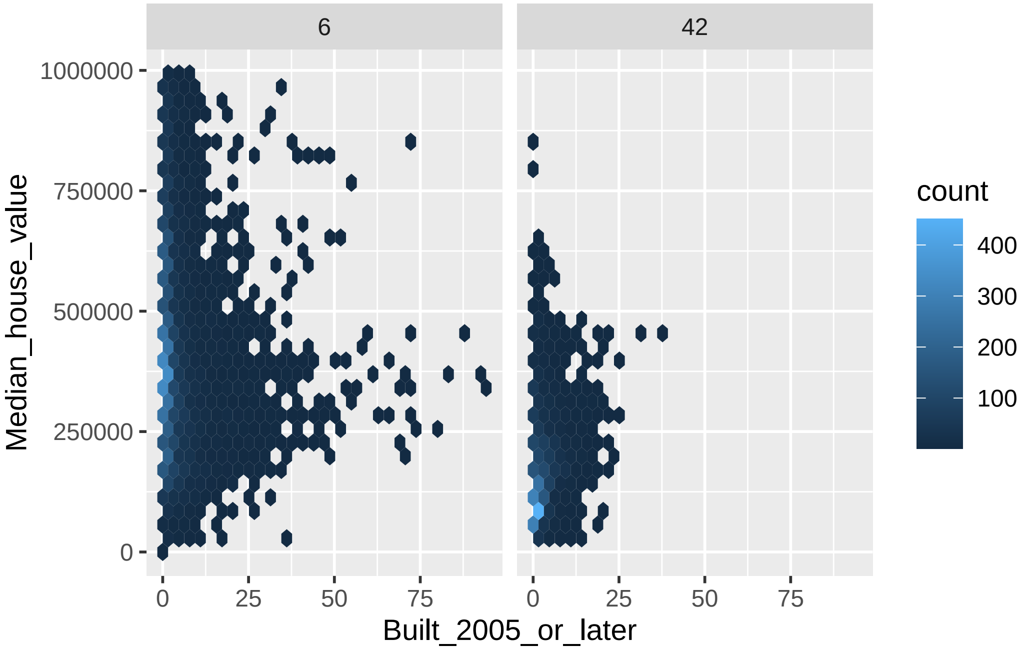


```
new_ca_pa %>%
  ggplot(mapping = aes(x = Built_2005_or_later, y = Median_house_value))+
  geom_hex()
```



b. Make a new plot, or pair of plots, which breaks the plot in (a) out by state (use your improved version of the scatterplot), for just California and Pennsylvania. Note that the state is recorded in the `STATEFP` variable, with California being state 6 and Pennsylvania state 42. What do you learn from this figure? Is there a difference between the two states?

```
new_ca_pa |>
  ggplot(mapping = aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_hex() +
  facet_wrap(~STATEFP)
```



c. What is the median percentage of houses built in 2005 or later (in the entire data set, i.e., California and Pennsylvania)? Create a new binary variable for whether the Census tract has percentage greater or less than this median. Make a visualization for the median house prices, broken down by this new variable. What do you learn from this figure?

```
median_price <- median(new_ca_pa$Median_house_value)
print(median_price)
```

```
[1] 311100
```

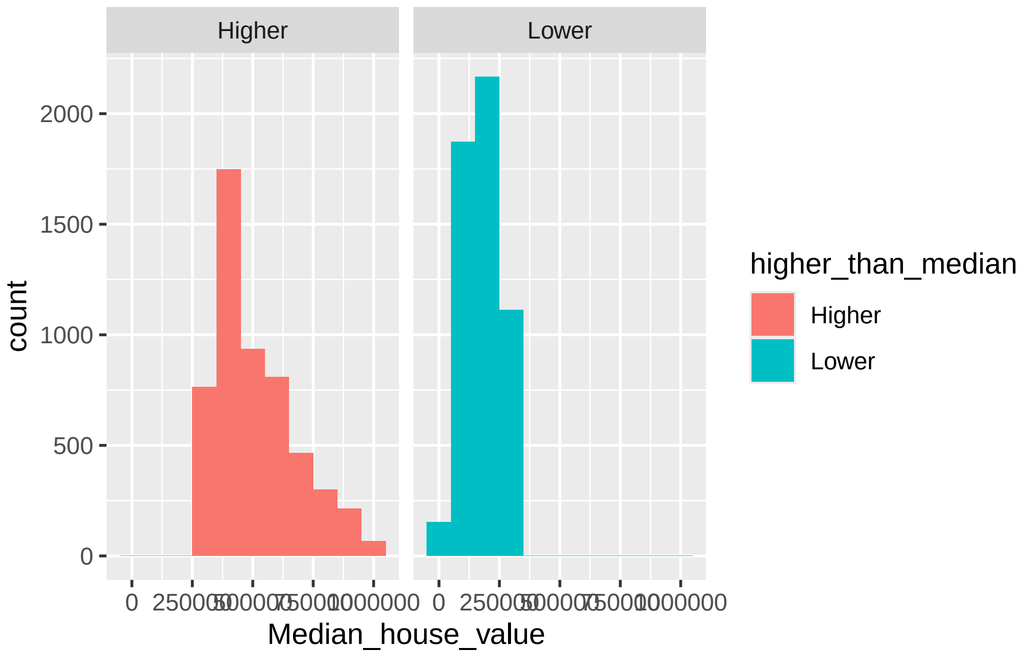
```
new_ca_pa$higher_than_median <- if_else(new_ca_pa$Median_house_value > median_price, "Higher'
head(new_ca_pa)
```

```
  X    GEO.id2 STATEFP COUNTYFP TRACTCE POPULATION LATITUDE LONGITUDE
2 2 6001400200       6        1  400200       1974 37.84829 -122.2495
3 3 6001400300       6        1  400300       4865 37.84027 -122.2544
4 4 6001400400       6        1  400400       3703 37.84845 -122.2573
5 5 6001400500       6        1  400500       3517 37.84894 -122.2647
6 6 6001400600       6        1  400600       1571 37.84223 -122.2649
7 7 6001400700       6        1  400700       4206 37.84172 -122.2721
                                GEO.display.label Median_house_value Total_units
2 Census Tract 4002, Alameda County, California             909600         929
```

```
3 Census Tract 4003, Alameda County, California             748700        2655
4 Census Tract 4004, Alameda County, California             773600        1911
5 Census Tract 4005, Alameda County, California             579200        1703
6 Census Tract 4006, Alameda County, California             480800         781
7 Census Tract 4007, Alameda County, California             460800        1977
  Vacant_units Median_rooms Mean_household_size_owners
2           37          6.0                       2.53
3          134          4.6                       2.45
4           68          5.0                       2.04
5           71          4.5                       2.66
6           65          4.8                       2.58
7          236          4.3                       2.72
  Mean_household_size_renters Built_2005_or_later Built_2000_to_2004
2                        1.81                   0                1.2
3                        1.66                   0                0.0
4                        2.19                   0                0.2
5                        1.72                   0                0.2
6                        2.18                   0                0.0
7                        2.15                   0                0.6
  Built_1990s Built_1980s Built_1970s Built_1960s Built_1950s Built_1940s
2         0.0         1.3         6.1         6.5         1.0        10.8
3         2.3         3.2         5.2         8.3         5.3         7.8
4         1.3         0.0         4.9         4.3         8.0        10.4
5         1.1         1.9         3.7         5.8         6.0         7.5
6         1.2         1.4         1.0         6.5        19.7        17.0
7         1.8         2.2         3.3         0.8         9.4         9.7
  Built_1939_or_earlier Bedrooms_0 Bedrooms_1 Bedrooms_2 Bedrooms_3 Bedrooms_4
2                  73.2        3.0       16.4       27.4       34.4       17.5
3                  68.0       11.5       28.4       29.2       20.4        7.9
4                  71.1        5.2       27.7       33.7       21.9        7.3
5                  73.8        4.9       30.2       38.1       19.3        5.4
6                  53.1        3.5       20.4       40.1       30.7        4.6
7                  72.4        8.2       22.3       43.2       16.7        6.5
  Bedrooms_5_or_more Owners Renters Median_household_income
2                1.2   66.0    34.0                  111667
3                2.7   45.1    54.9                   66094
4                4.2   45.0    55.0                   87306
5                2.1   43.6    56.4                   62386
6                0.8   51.0    49.0                   55658
7                3.1   32.2    67.8                   38646
  Mean_household_income higher_than_median
2                195229             Higher
3                105877             Higher
```

```
4                  106248             Higher
5                   74604             Higher
6                   73933             Higher
7                   56705             Higher
```

```
new_ca_pa |>
  ggplot(mapping = aes(x = Median_house_value, fill = higher_than_median)) +
  geom_histogram(binwidth = 100000) +
  facet_wrap(~higher_than_median)
```



Problem 3: Nobody Home (25 points)

The vacancy rate is the fraction of housing units which are not occupied. The dataframe contains columns giving the total number of housing units for each Census tract, and the number of vacant housing units.

a. Add a new column to the dataframe which contains the vacancy rate. What is mean vacancy rate?

```
new_ca_pa<- new_ca_pa |>
  mutate(vacancy_rate = Vacant_units / Total_units)

mean(new_ca_pa$vacancy_rate)
```

```
[1] 0.08888789
```

b. What is the standard deviation? Calculate this using just basic arithmetic operations (`+` or `sum()`, `-`, …) and `length()`, then use the `sd()` function to make sure that you get the same result.

```
sqrt(sum((new_ca_pa$vacancy_rate - mean(new_ca_pa$vacancy_rate))^2) / (nrow(new_ca_pa) - 1))
```

```
[1] TRUE
```

Problem 4: County Investigation (25 points)

The column `COUNTYFP` contains a numerical code for counties within each state.

a. We are interested in San Francisco County (county 75 in California), Yolo (county 113 in California), and Allegheny County (county 3 in Pennsylvania). Create a new data frame with just the relevant rows. What is the median home value in Yolo county?

```
county_new_ca <- new_ca_pa |>
  filter((STATEFP == 6 & COUNTYFP == 75) |
         (STATEFP == 6 & COUNTYFP == 113) |
          STATEFP == 42 & COUNTYFP == 3)

county_new_ca %>%
  filter(county_new_ca$COUNTYFP == 113) %>%
  summarize(median_yolo_county = median(Median_house_value))
```

```
  median_yolo_county
1             361700
```

b. For San Francisco, Yolo and Allegheny Counties, what were the average percentages of housing built since 2005?

```
county_new_ca |>
  group_by(COUNTYFP) |>
    summarize(avg_percentages_housing2005 = mean(Built_2005_or_later))
```

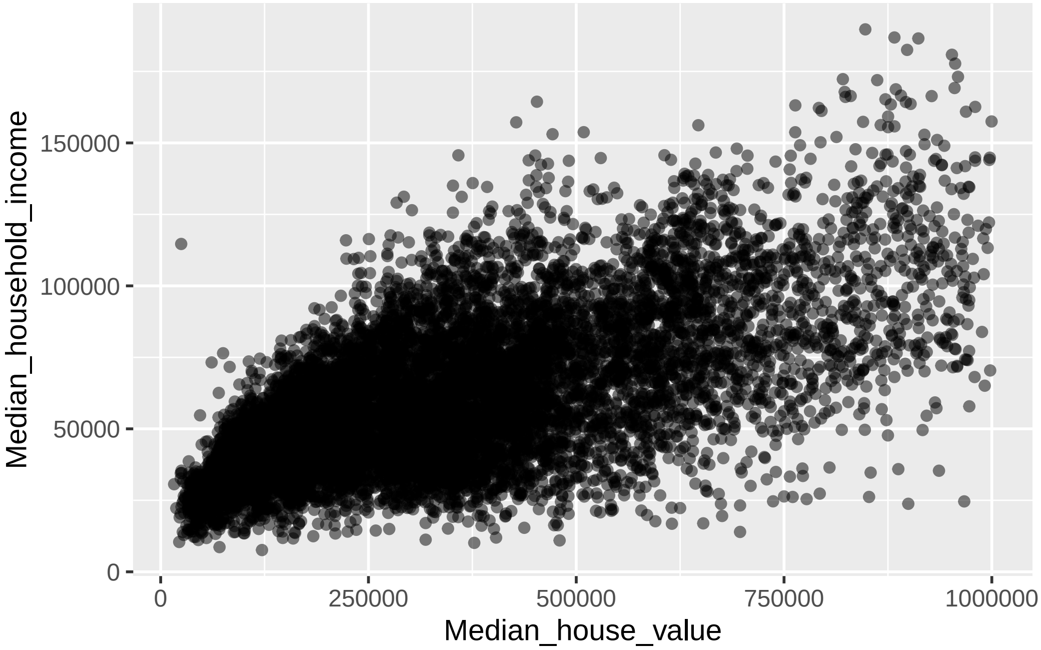
```
# A tibble: 3 x 2
  COUNTYFP avg_percentages_housing2005
     <int>                       <dbl>
1        3                        1.47
2       75                        2.23
3      113                        6.04
```

```
#outputs the average percentages of housing built since 2005 in each county.
```

c. What is the (Pearson) correlation coefficient between median house value and the median household income in (i) the whole data, (ii) all of California, (iii) all of Pennsylvania, (iv) San Francisco County? First make scatterplots and guess, then compute these in R. What do you learn about the relationship between median house values and median household income?

d.

```
new_ca_pa |>
  ggplot(mapping = aes(x = Median_house_value, y = Median_household_income)) +
  geom_point(alpha = 0.5)
```

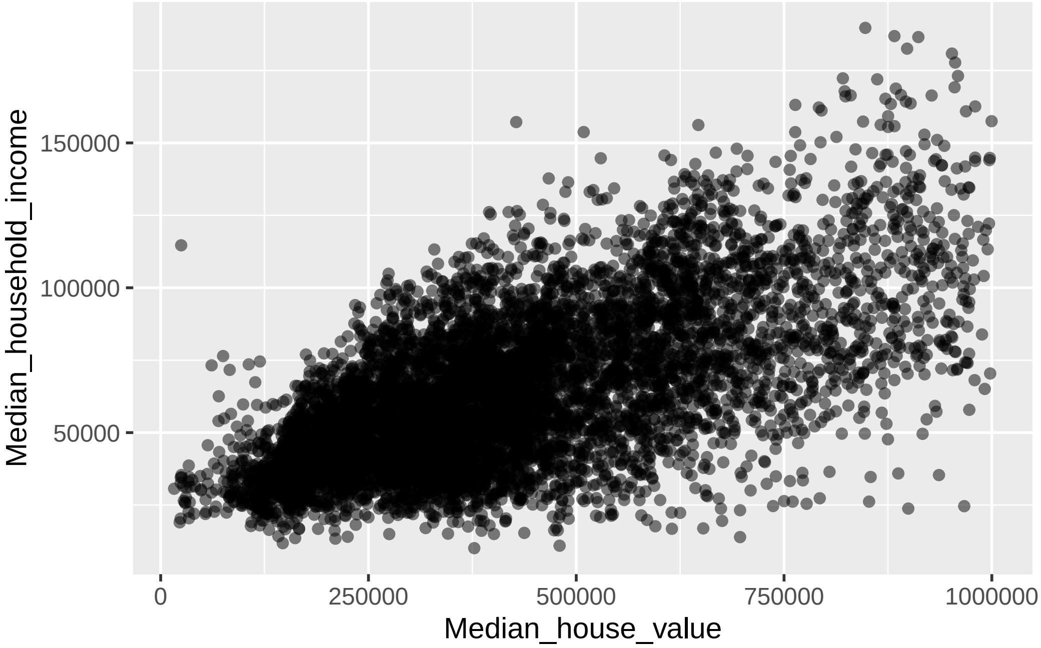


```r
#I am guessing 0.7

cor(new_ca_pa$Median_house_value, new_ca_pa$Median_household_income)
```

```
[1] 0.6402729
```

ii.

```r
new_ca <- new_ca_pa |>
  filter(STATEFP == 6)

ggplot(data = new_ca, mapping = aes(x = Median_house_value, y = Median_household_income)) +
geom_point(alpha = 0.5)
```

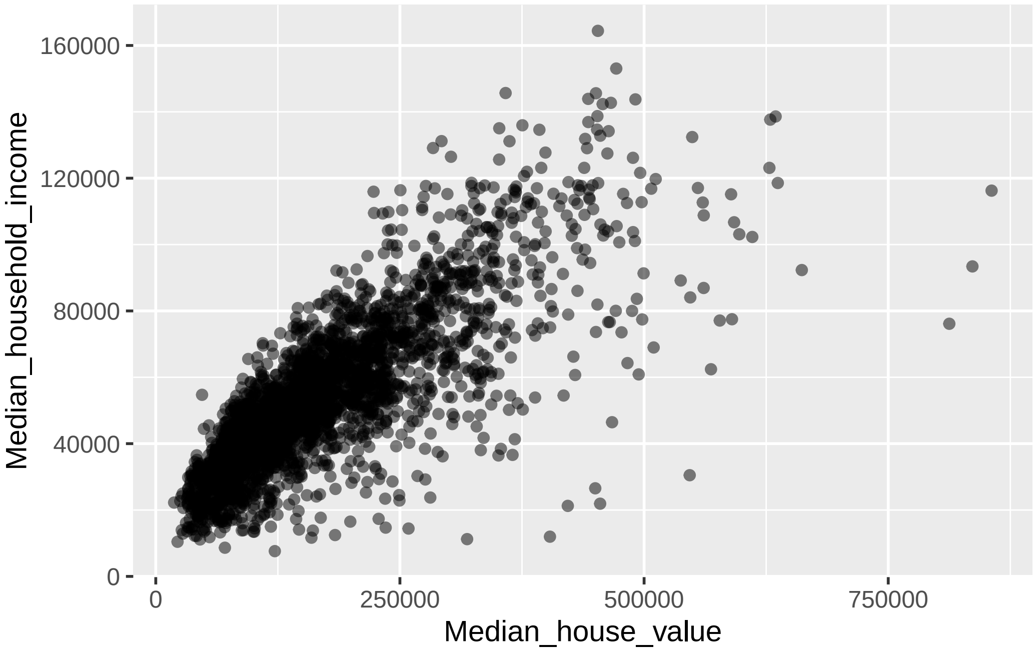


```r
#I am gonna guess 0.67
cor(new_ca$Median_house_value, new_ca$Median_household_income)
```

```
[1] 0.6438247
```

iii

```
new_pa <- new_ca_pa |>
  filter(STATEFP == 42)

ggplot(data = new_pa, mapping = aes(x = Median_house_value, y = Median_household_income)) +
geom_point(alpha = 0.5)
```

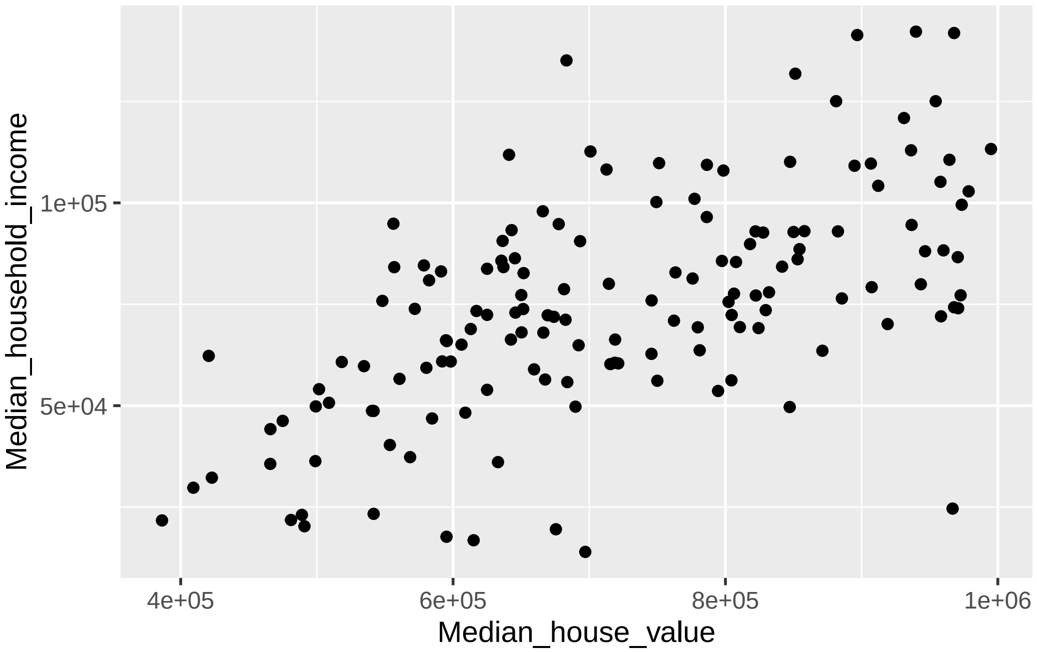


```
#I am guessing 0.75
cor(new_pa$Median_house_value, new_pa$Median_household_income)
```

```
[1] 0.7948716
```

iv.

```
new_sfc <-new_ca_pa |>
  filter(STATEFP == 6 & COUNTYFP == 75)

ggplot(data = new_sfc, mapping = aes(x = Median_house_value, y = Median_household_income)) +
geom_point(alpha = 1)
```



```r
# I am guessing 0.4

cor(new_sfc$Median_house_value, new_sfc$Median_household_income)
```

```
[1] 0.6006226
```

Appendix

```r
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
 [5] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
 [9] ggplot2_3.5.1   tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] gtable_0.3.5      jsonlite_1.8.9    compiler_4.4.1    tidyselect_1.2.1
 [5] scales_1.3.0      yaml_2.3.10       fastmap_1.2.0     lattice_0.22-6
 [9] hexbin_1.28.4     R6_2.5.1          labeling_0.4.3    generics_0.1.3
[13] knitr_1.48        munsell_0.5.1     pillar_1.9.0      tzdb_0.4.0
[17] rlang_1.1.4       utf8_1.2.4        stringi_1.8.4     xfun_0.48
[21] timechange_0.3.0  cli_3.6.3         withr_3.0.1       magrittr_2.0.3
[25] digest_0.6.37     grid_4.4.1        rstudioapi_0.16.0 hms_1.1.3
[29] lifecycle_1.0.4   vctrs_0.6.5       evaluate_1.0.1    glue_1.8.0
[33] farver_2.1.2      fansi_1.0.6       colorspace_2.1-1  rmarkdown_2.28
[37] tools_4.4.1       pkgconfig_2.0.3   htmltools_0.5.8.1
```