Data Visualizations on Housing Data

Description

A Given a random sample of 20, 000 housing sales from the San Francisco Bay Area. The data contains a large amount of relevant information for each house sale. However, there may be errors and missing values in the data.

Data Documentary:

The housing dataset contains the following features:

county The county in which the house is sold. city The city in which the house is sold.

zip The zip code in which the house is sold in. street The address of the house.

price The price (in nominal U.S. dollars) of the house.

br The number of bedrooms of the house.

lsqft The lot size of the house, measured in square feet. bsqft The building size of the house, measured in square feet.

year The construction year of the house

date The date when the house was sold.

long The longitude (horizontal) of the house location lat The latitude (vertical) of the house location

1. Load in the housing dataset. Convert the columns to appropriate data types. If you notice any data irregularities, you can potentially write them here. You don’t need to write an answer in your report for this question, but please mark the code for this question in the appendix.

```
## 
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
## 
##     date
```

```
##  [1] "county" "city"   "zip"    "street" "price"  "br"     "lsqft" 
##  [8] "bsqft"  "year"   "date"   "long"   "lat"
```

```
##  [1] 20005 20005 20005 20005  3885  2005  2005  2005  2005  2005  2005
## [12]  2005  2005  2005  2004  2004  2004  2004  2004  2004  2004  2004
## [23]  2004  2004  2004  2004  2004  2004  2004  2004
##  [1] 1900 1900 1900 1900 1900 1898 1898 1896 1896 1890 1890 1890 1890 1889
## [15] 1885 1885   88   88   88   86   82   76   75   74   74   55   37    5
## [29]    3    0
```

2. What timespan does the housing sales cover? What is the timespan of the construction dates of homes?

by looking into the data, only the highest number year 2005 and lowerst number 1885 make logical sense.

```
## Time difference of 1134 days
```

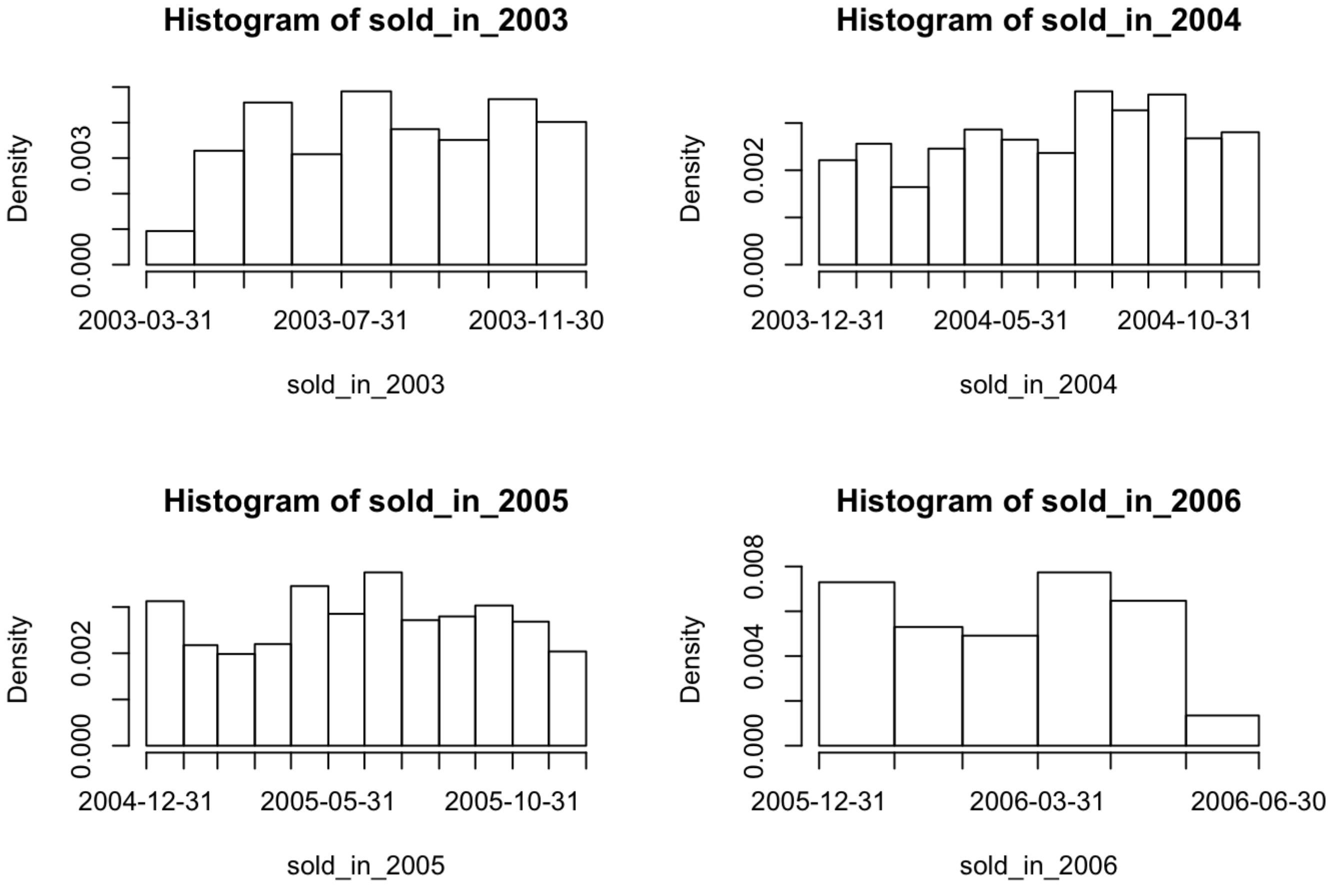
```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
## [1] 120
```

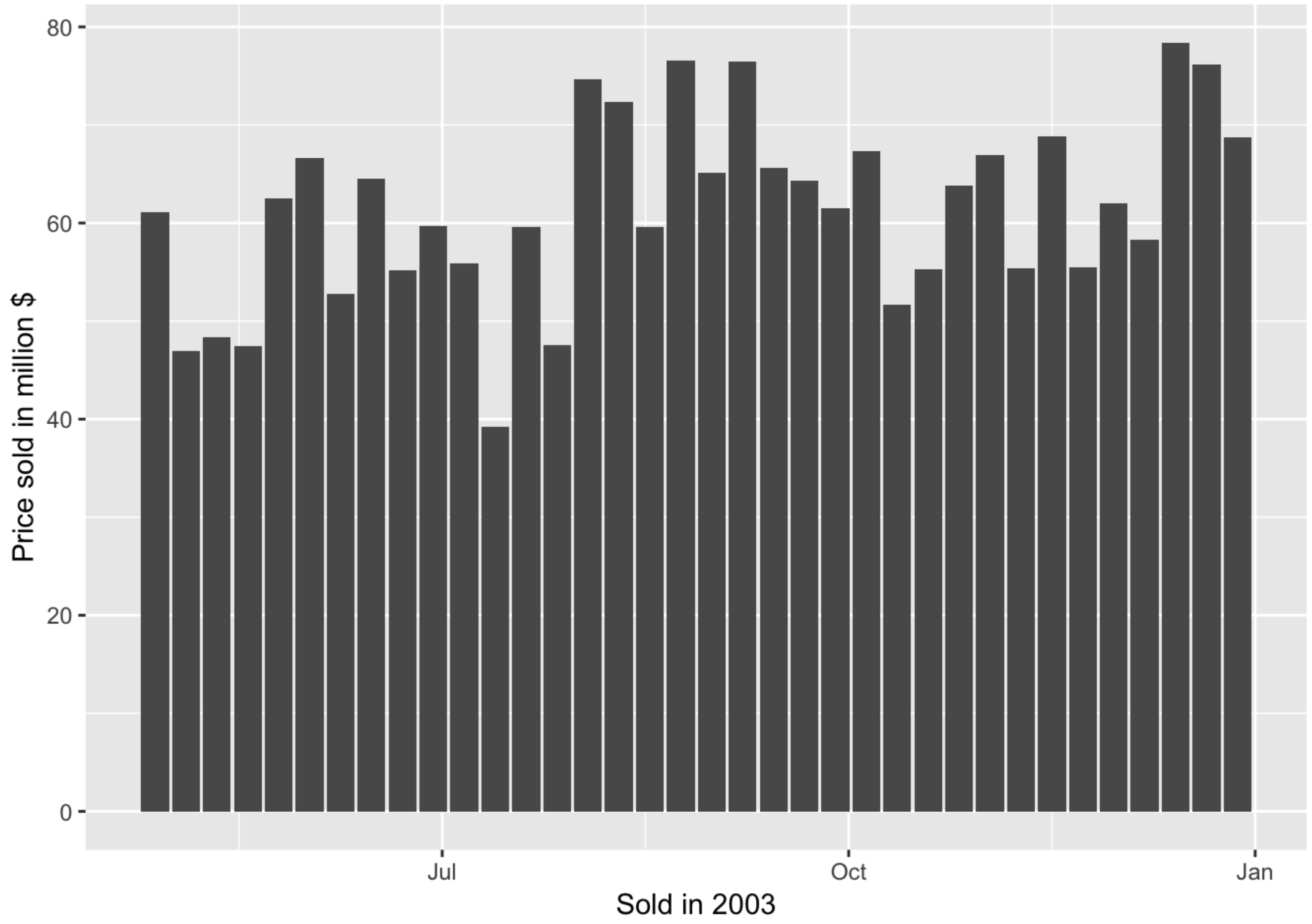
3. Examine the monthly housing sales for this dataset. You will need to look at combinations of both year and month using the date variable. Make two plots:

A plot that shows the number of sales over time.

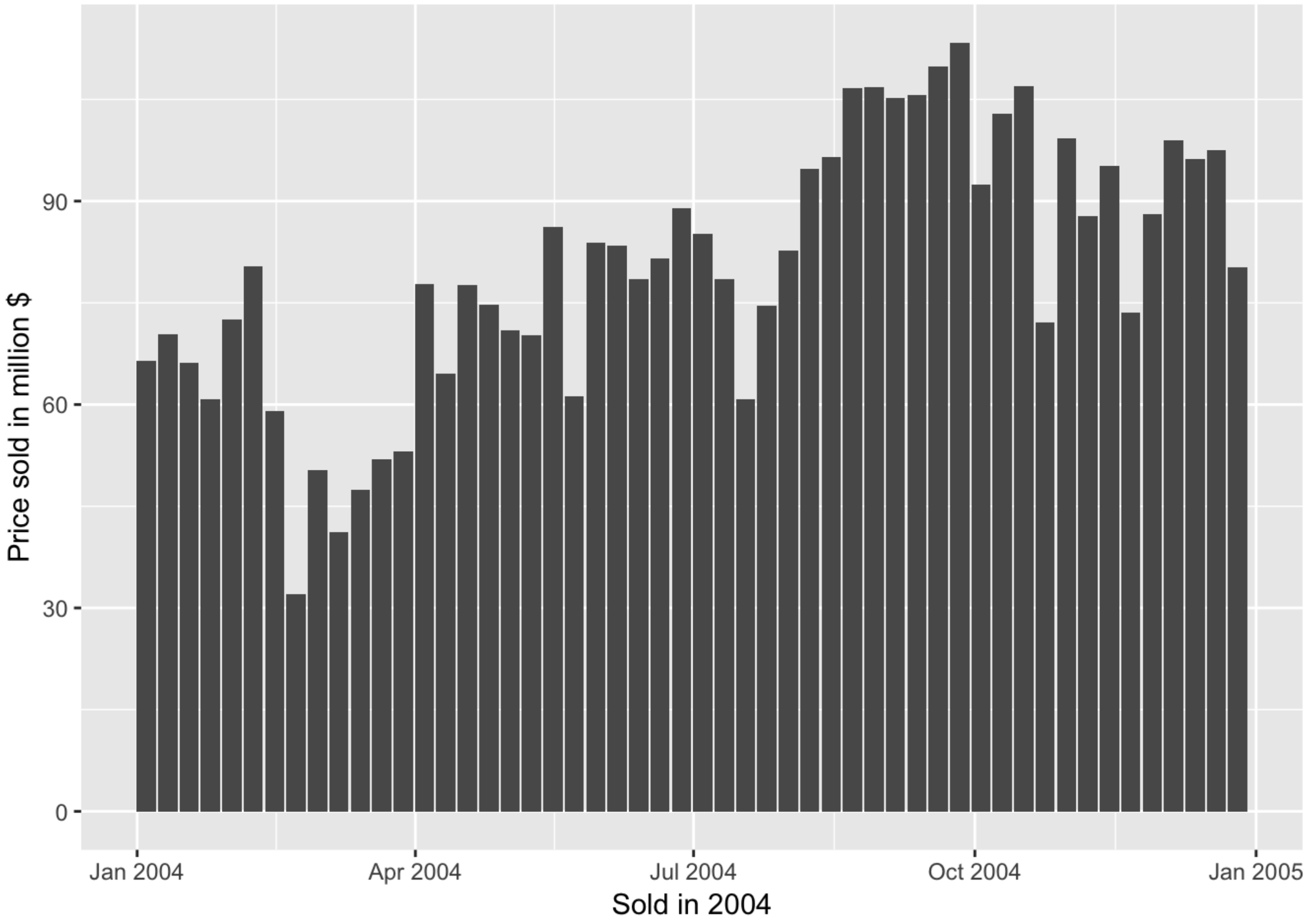
A plot that shows the average house price over time.



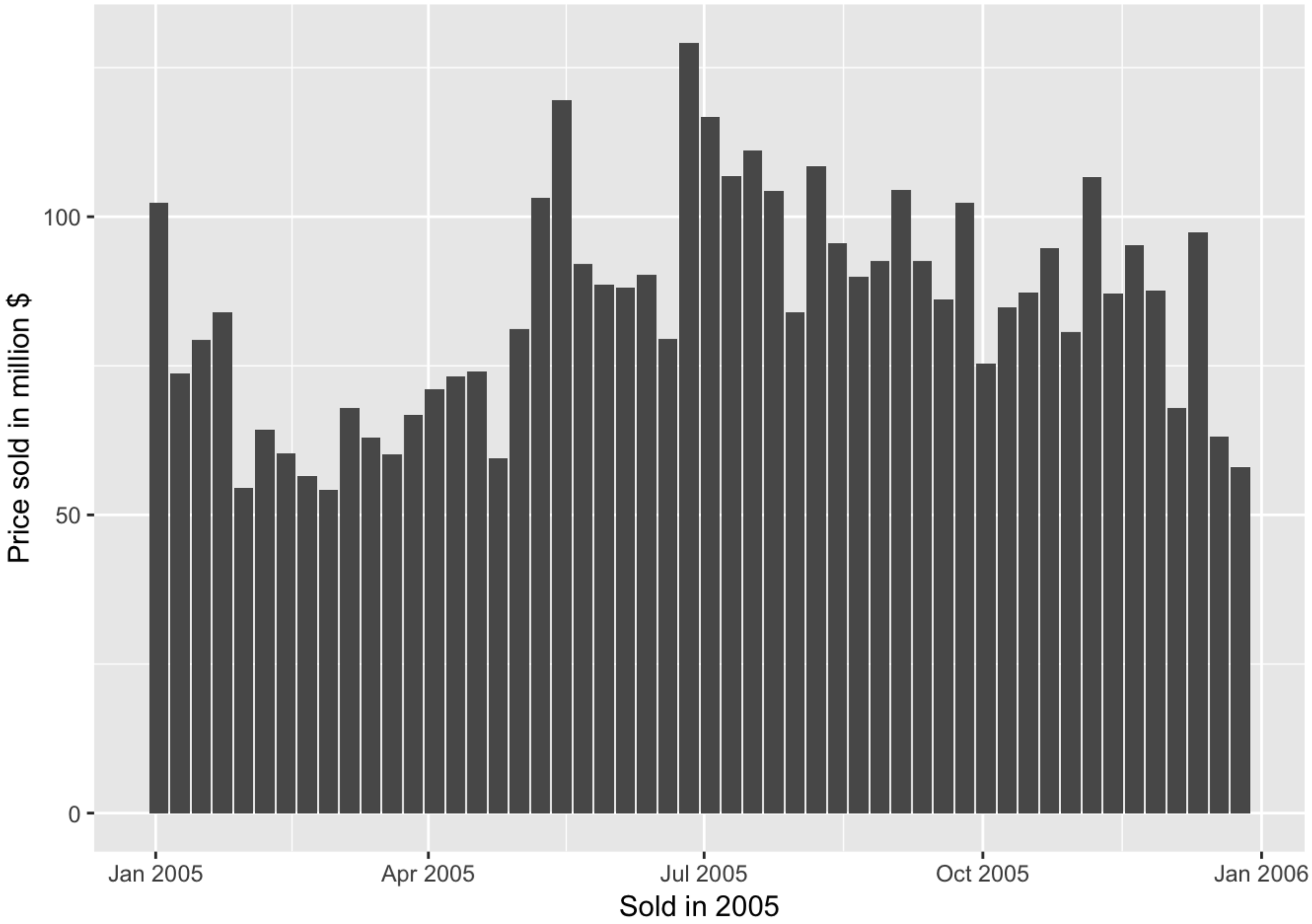
```
## Warning: Removed 2 rows containing missing values (position_stack).
```

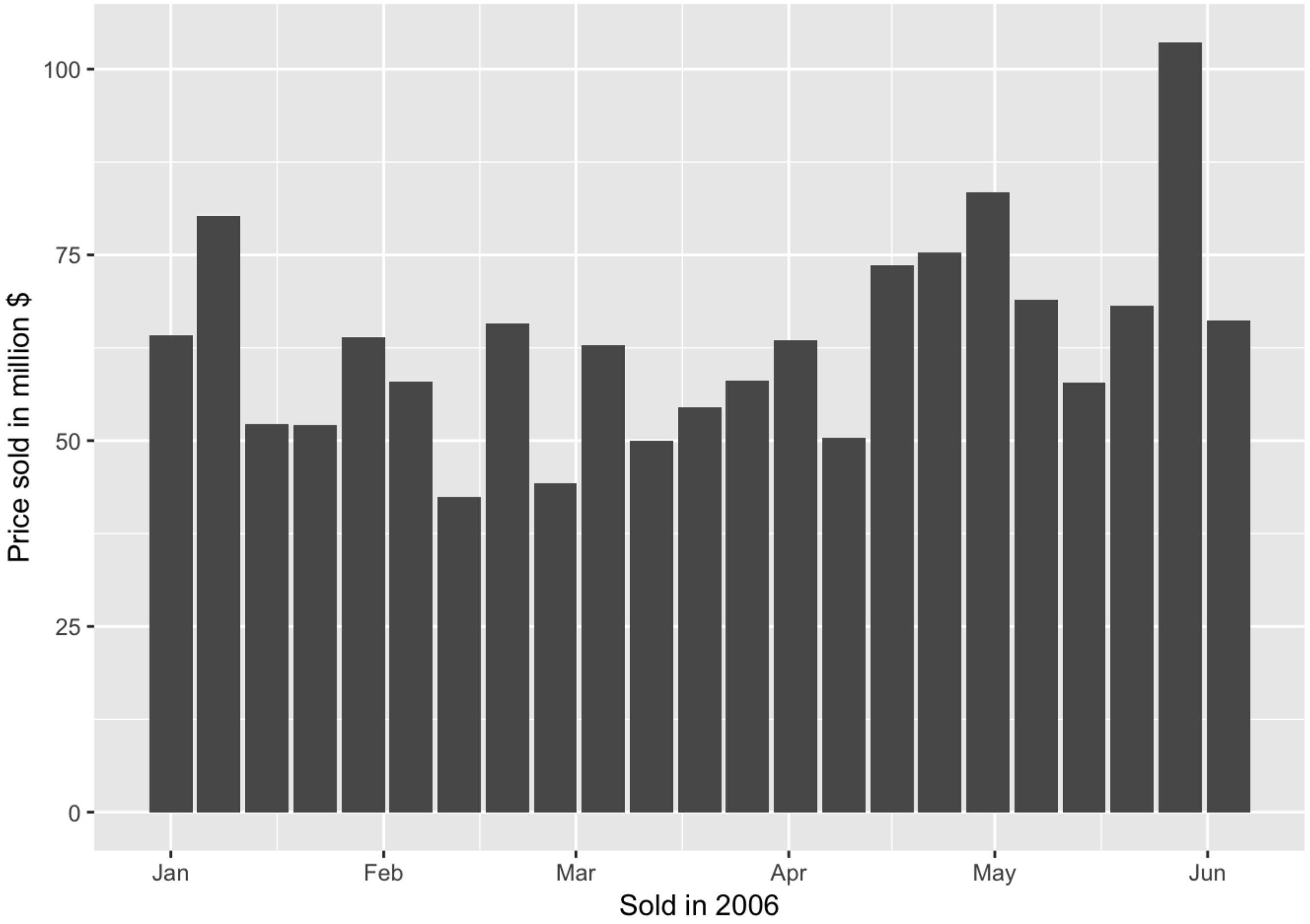


```
## Warning: Removed 5 rows containing missing values (position_stack).
```



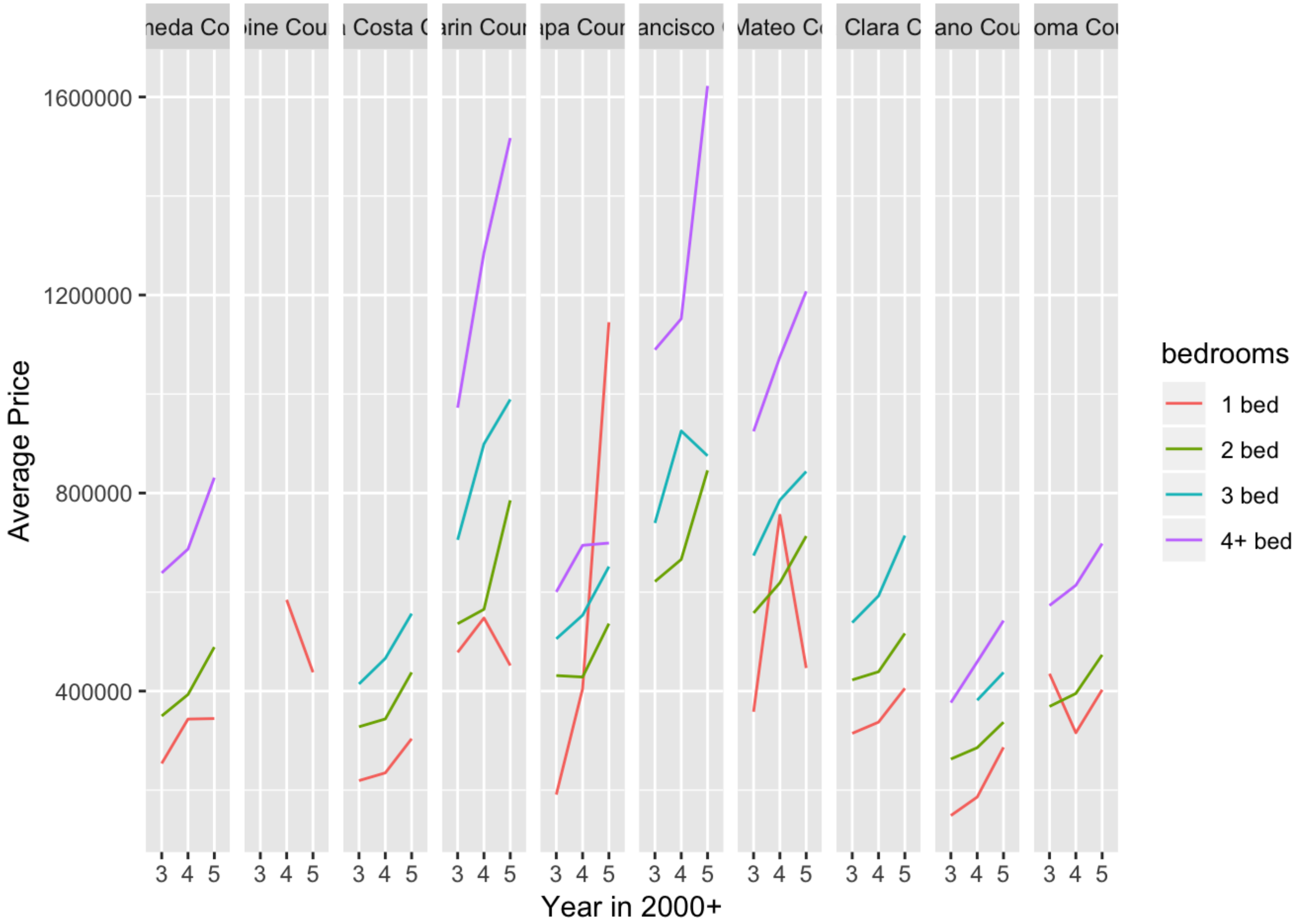
```
## Warning: Removed 2 rows containing missing values (position_stack).
```


Price sold in million $
0
50
100
Jan 2005
Apr 2005
Jul 2005
Oct 2005
Jan 2006
Sold in 2005


Price sold in million $
0
25
50
75
100
Jan
Feb
Mar
Apr
May
Jun
Sold in 2006

4. Make a line plot that shows how the average price depends on county, bedrooms, and sale year. For the bedrooms variable, define the levels to be 1,2,3,4+ bedrooms. For the sale year variable, use the levels 2003,2004,2005. Use all levels of the county variable. Each line should have three points corresponding to year. You will need separate lines for each combination of county and br.

```
## Warning: Removed 1 rows containing missing values (geom_path).
```

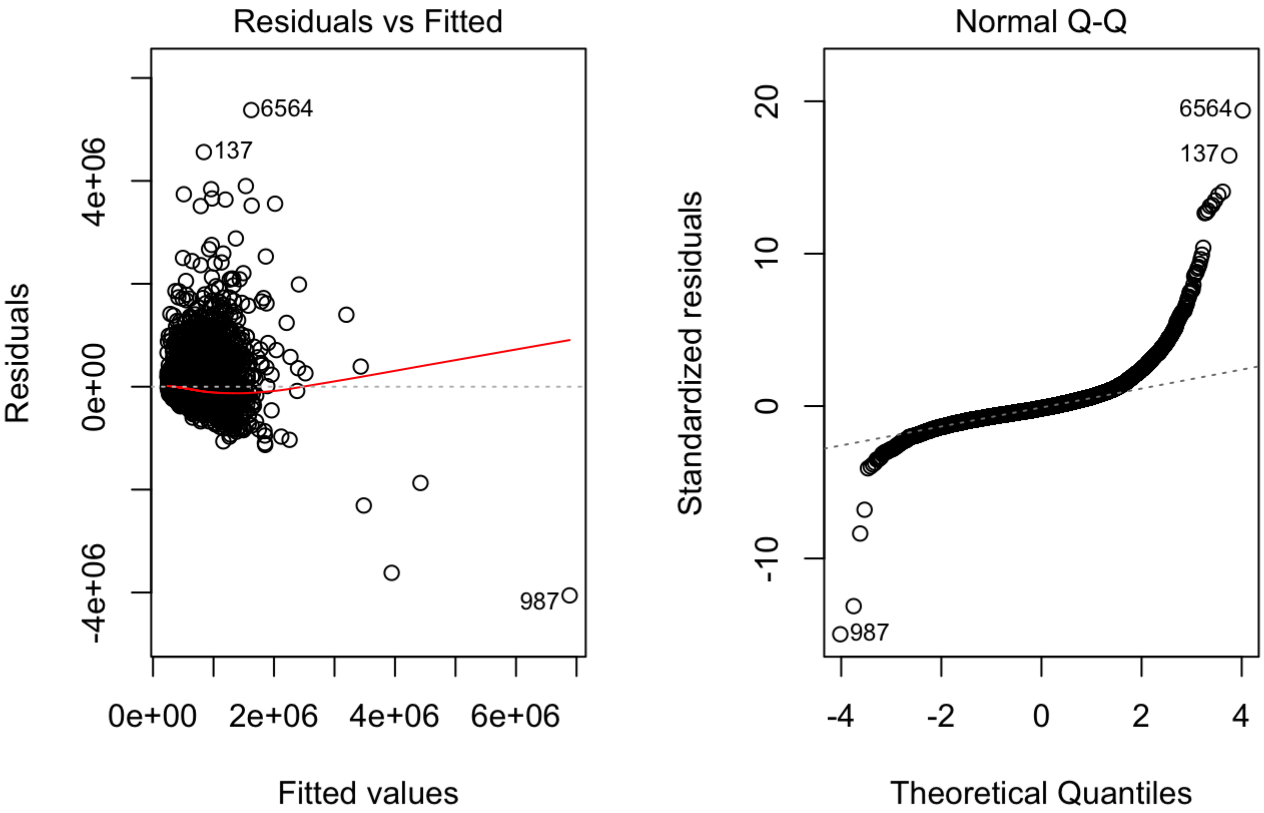


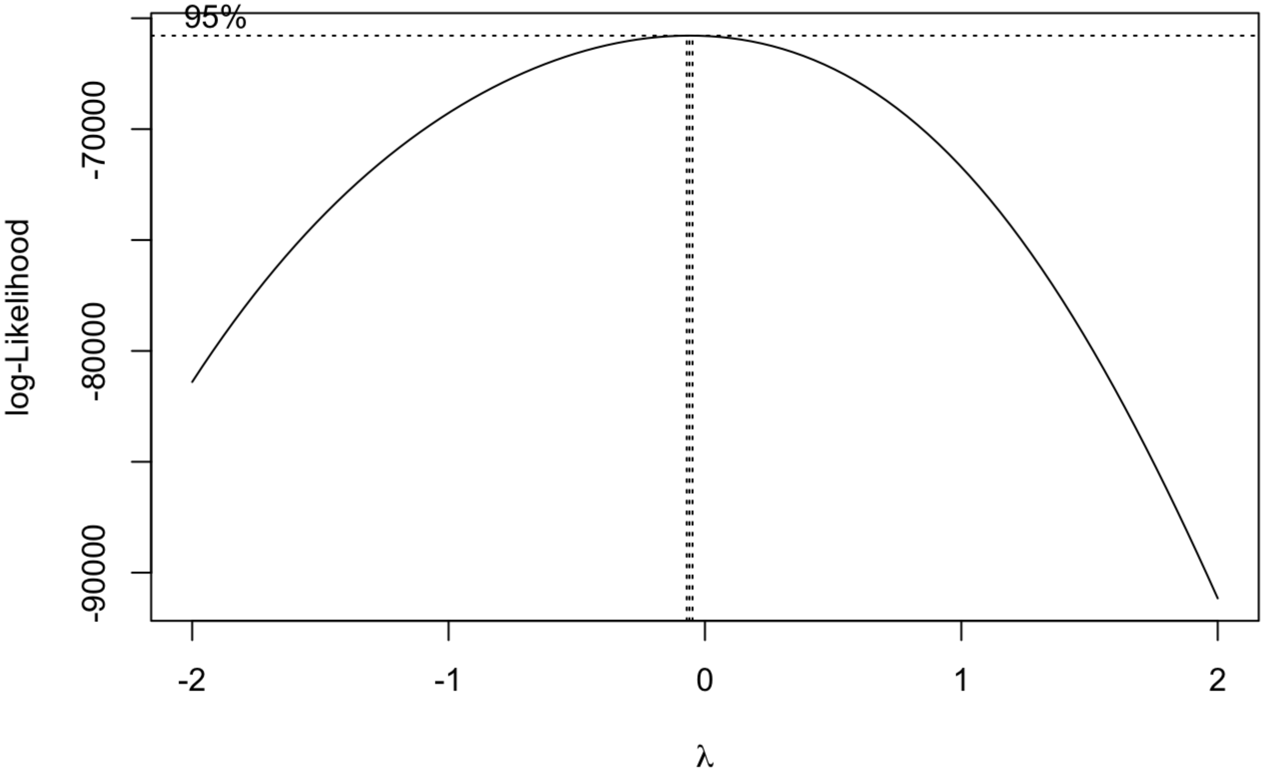
5. Do all housing sales within a given city only occur in one county? Why or why not? Justify your answer. If you have any cities that have sales in more than one county, list them and report how many cities you find.

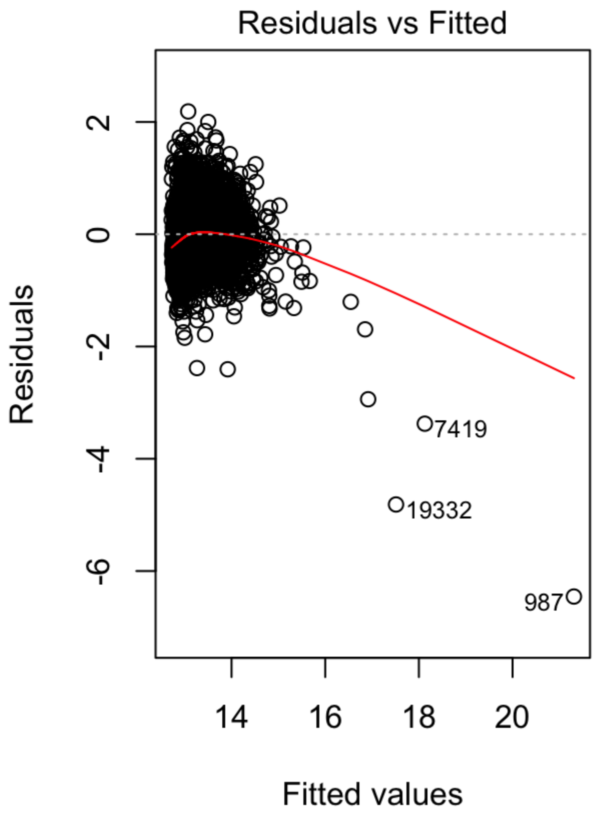
```
## San Francisco       Vallejo
##           128           157
```

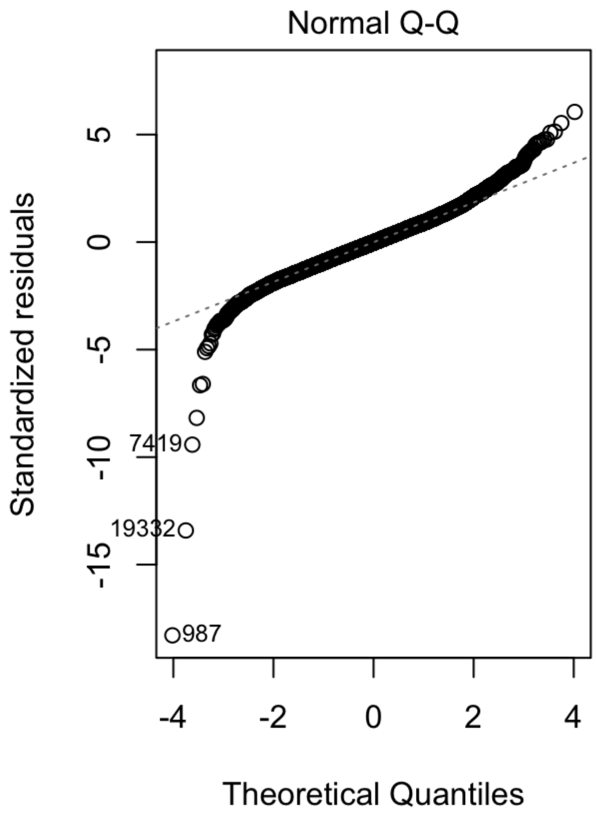
```
## [1] "San Francisco County" "Alpine County"
## [1] "Solano County" "Napa County"
```

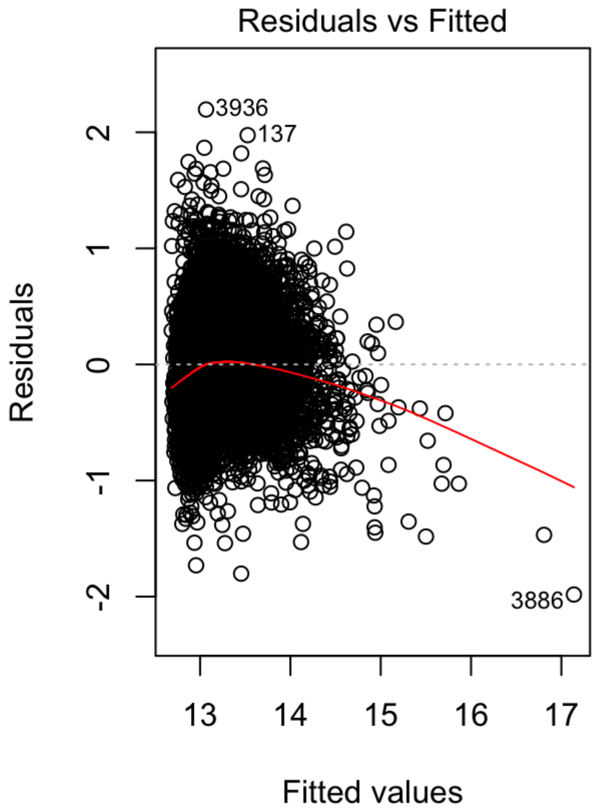
6. Fit a linear regression model that uses bsqft to predict price. Make sure to use appropriate diagnostics (e.g. Q-Q plots, Box-Cox transformations). Removing extreme outliers is okay for this problem. Do not worry if you’re not too familiar with Linear Regression, any required methods will be covered during class.

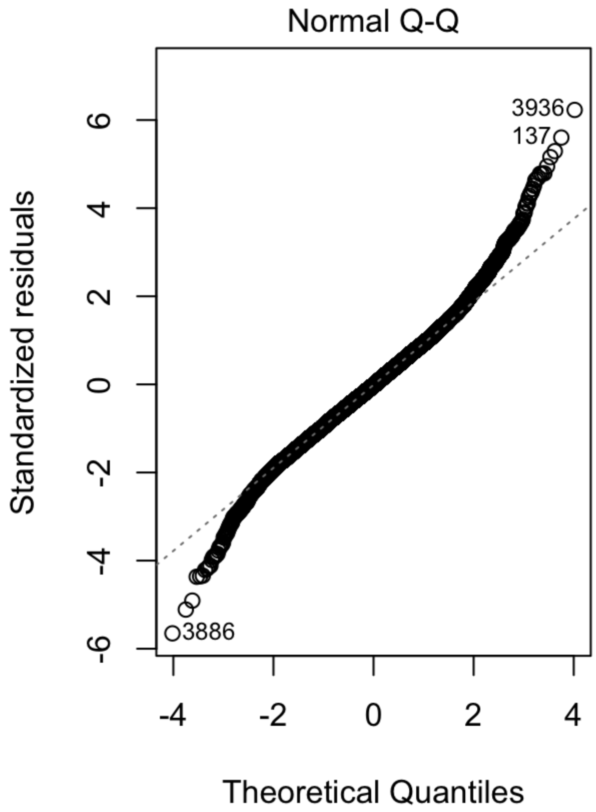

Residuals vs Fitted
Residuals
Fitted values
4e+06
0e+00
-4e+06
0e+00
2e+06
4e+06
6e+06
6564
137
987
Normal Q-Q
Standardized residuals
Theoretical Quantiles
20
10
0
-10
-4
-2
0
2
4
6564
137
987


95%
log-Likelihood
-70000
-80000
-90000
-2
-1
0
1
2
λ


Residuals vs Fitted
Residuals
2
0
-2
-4
-6
7419
19332
987
14
16
18
20
Fitted values


Normal Q-Q
Standardized residuals
5
0
-5
-10
-15
7419
19332
987
-4
-2
0
2
4
Theoretical Quantiles


Residuals vs Fitted
Residuals
2
1
0
-1
-2
3936
137
3886
13
14
15
16
17
Fitted values


Normal Q-Q
Standardized residuals
6
4
2
0
-2
-4
-6
3936
137
3886
-4
-2
0
2
4
Theoretical Quantiles

7. Now fit a linear regression model that uses both bsqft and lsqft to predict price. Do not include any transformations or diagnostics. Using R, conduct a hypothesis test for H_0 : $\beta_{bsqft} \geq \beta_{lsqft}$ vs. H_1 : $\beta_{bsqft} < \beta_{lsqft}$.

For your convenience, you may assume the following as true:

$$Var(\hat{\beta}_{bsqft} - \hat{\beta}_{lsqft}) = Var(\hat{\beta}_{bsqft}) + Var(\hat{\beta}_{lsqft})$$

p-value is close to 1

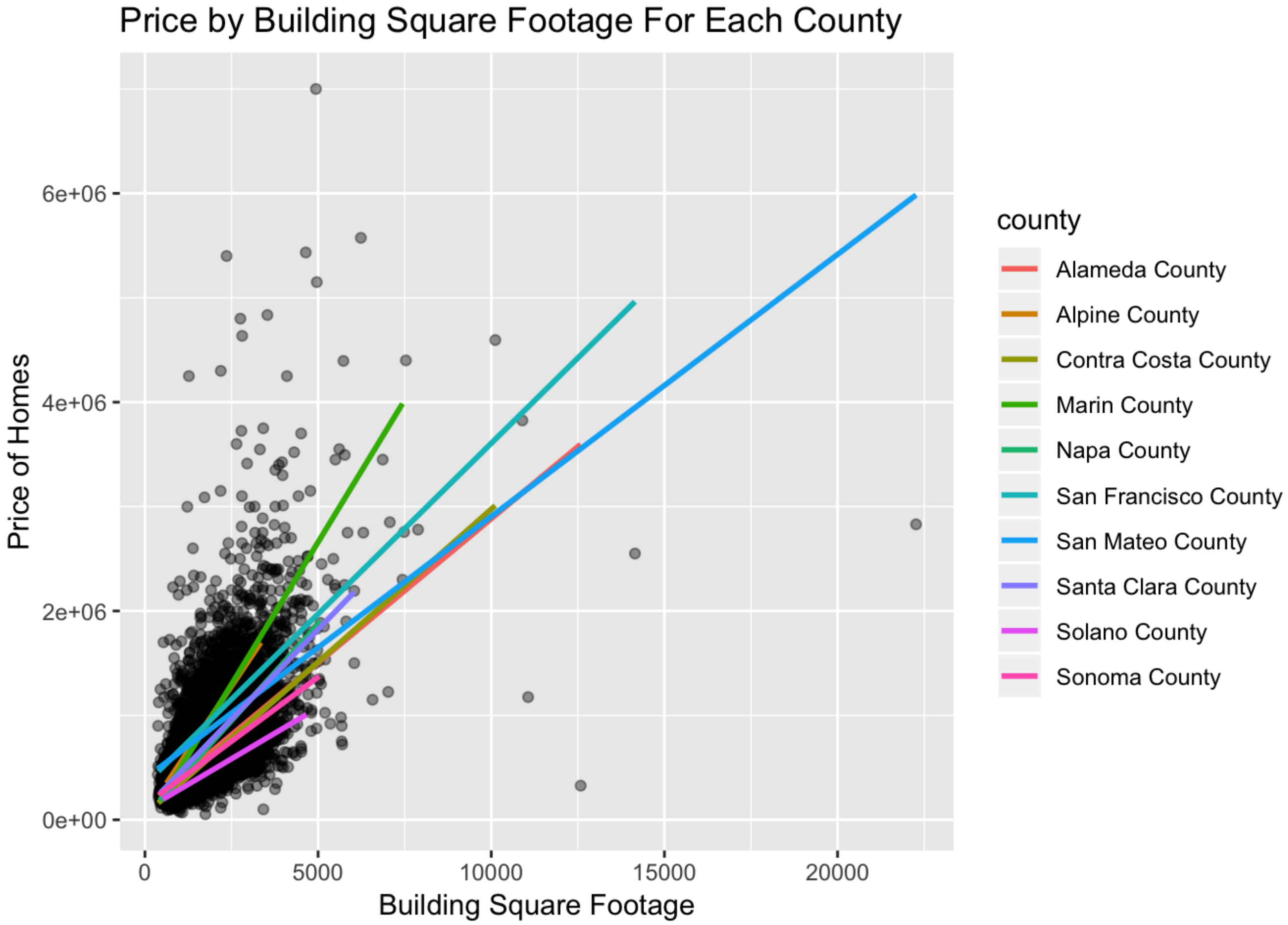
Decision: fail to reject H0 at 5% significance level because p-value > 0.05

```
##
## Call:
## lm(formula = price ~ lsqft + bsqft, data = Hdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4032929  -144063   -40375    91569  5380517
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.253e+05  5.483e+03  22.850   <2e-16 ***
## lsqft       -6.121e-04  8.333e-04  -0.735    0.463
## bsqft        3.026e+02  3.038e+00  99.609   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 282200 on 15599 degrees of freedom
##   (4398 observations deleted due to missingness)
## Multiple R-squared:  0.3889, Adjusted R-squared:  0.3888
## F-statistic:  4963 on 2 and 15599 DF,  p-value: < 2.2e-16
```

```
## [1] 1
```

8. Fit an individual regression line using bsqft and price for each separate county

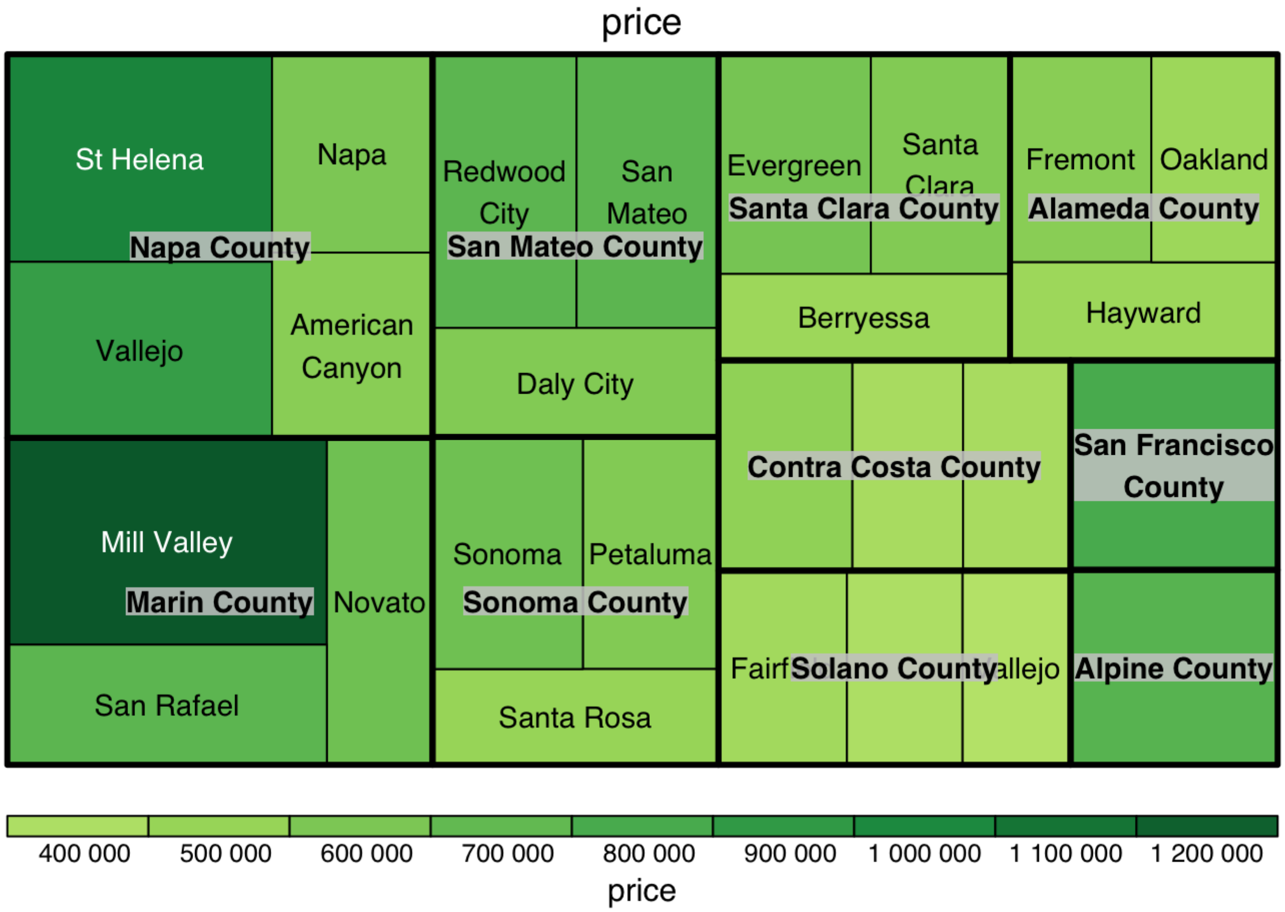
Do not write repeated calls of the lm() function. Instead, use an appropriate apply function. Draw each individual regression line in a single plot. Make sure to distinguish the lines. Can we conclude that the regression lines depend on county? Hint: Are the lines parallel or not?



9. For this question, you will be developing a treemap plot. Read the following steps carefully:

It is of interest to visualize the average prices for the three cities with the most sales in each county. Use appropriate subsetting methods to find the corresponding ”top three” cities for each county. Note that some counties may have only 1 or 2 cities in which housing sales occur. This will not affect the end result.

For the treemap, make sure that the order of the indexing has the city variable ”nested” in the county variable. This would require specifying index=c(county,city) when making the treemap in R.



10 Your last question is to learn how to develop a heatmap. Read the following steps carefully:

The objective of this question is to examine how price and frequency of San Francisco housing sales depend on location. You are tasked with making two heatmaps: 1. A heatmap that is colorized based on the number of houses in a cell 2. A heatmap that is colorized by the average price of the houses within a cell

Each cell is a .01 (longitude) × .01 (latitude) square. To implement this in R, you will want to perform the following transformation to the location variables:

SFdata$long2 < -round(SFdata$long,2)

SFdata$lat2 < -round(SFdata$lat,2)

Create appropriate factor variables for latitude and longitude that include levels that are not present in the dataset. You can use the levels argument to implement this.

For each plot, consider using the image() function to make a heatmap. You can use heatmap() or heatmap.2() or anything you want, but image() is better when overlaying lines. You will need to include the following arguments:

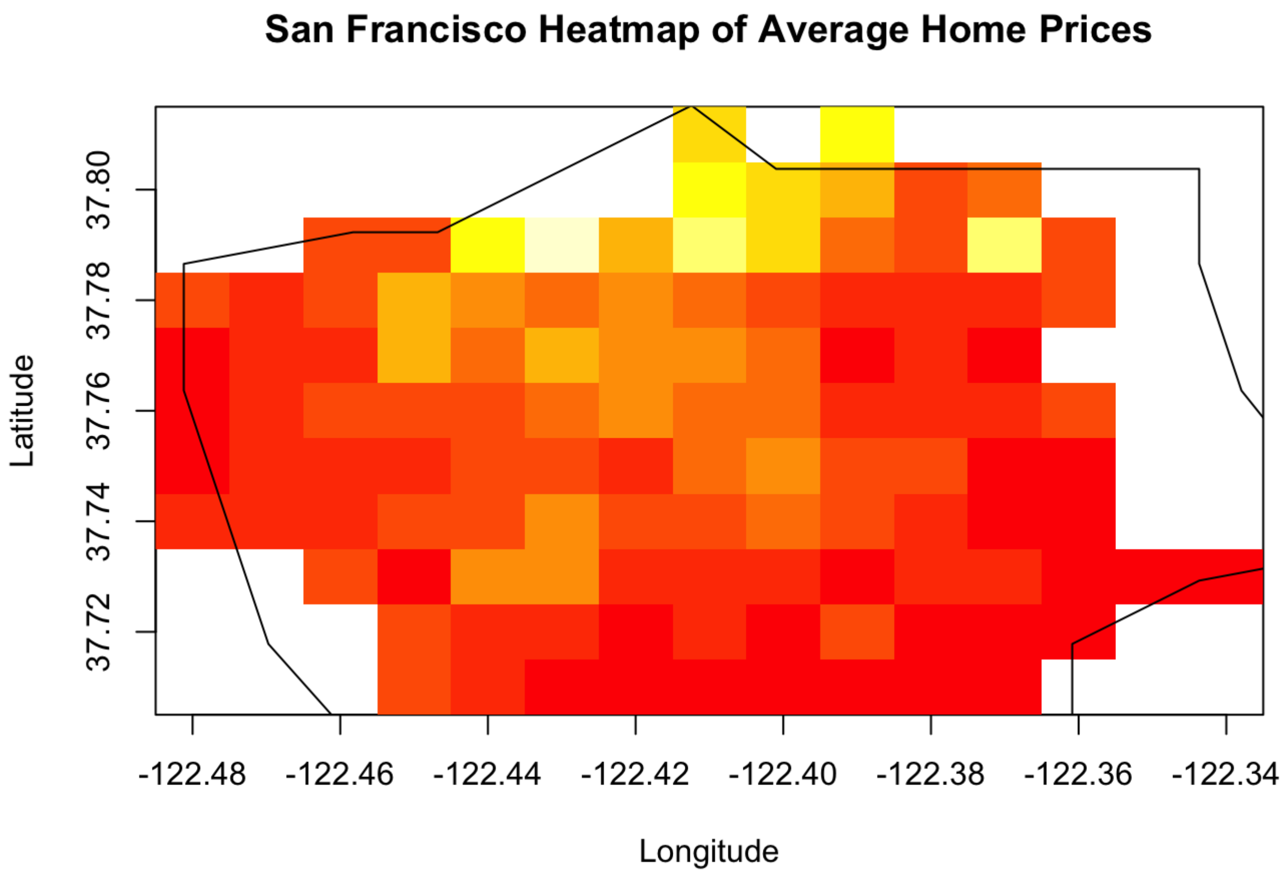
(a) x A length m vector with the levels of your longitude factor variable

(b) y A length n vector with the levels of your latitude factor variable

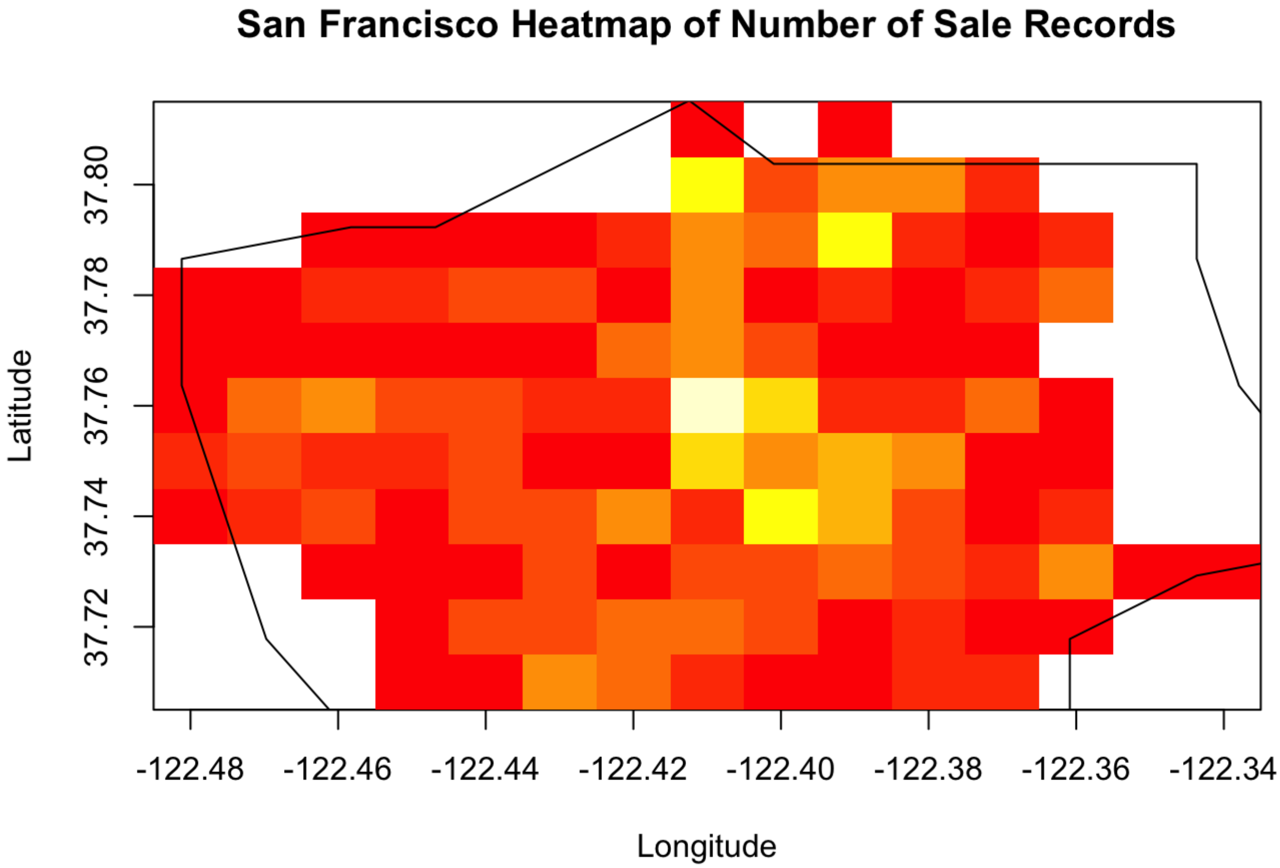
(c) z An m × n matrix object that contains the corresponding values for each cell. If there are no houses in a specific cell, then it should be reported as NA.

Finally, use the maps package to sketch in the borders of San Francisco around the tiles.

heatmap for average housing price



heatmap for # of sale records


San Francisco Heatmap of Number of Sale Records
Latitude
37.72
37.74
37.76
37.78
37.80
Longitude
-122.48
-122.46
-122.44
-122.42
-122.40
-122.38
-122.36
-122.34

R Appendix

```
knitr::opts_chunk$set(echo = FALSE)

Hdata <- read.csv(file = "~/Desktop/STA/STA141A/hw2/housing.csv")
library(stringr)
library(lubridate)

names(Hdata)
head(sort(Hdata$year, decreasing = TRUE) , n = 30)
tail(sort(Hdata$year, decreasing = TRUE) , n = 30)
Hdata$year <- str_replace(Hdata$year, "20005", "2005") # typo fixed.
Hdata$year <- str_replace(Hdata$year, "3885", "1885") # typo fixed.
typo_ind <- which(as.numeric(Hdata$year) < 100)
Hdata$year[typo_ind] <- "NA" #removed unknowns
year <- Hdata$year
Hdata$county <- str_replace( Hdata$county, pattern = "county", replacement = "County") %
>% factor()
Hdata$county <- str_replace( Hdata$county, pattern = "Franciscoe", replacement = "Franci
sco") %>% factor()
Hdata$br <- cut(Hdata$br,
                c(0,1,2,3,Inf),
                c("1 bed", "2 bed", "3 bed", "4+ bed")
                  )   # got helped by Jody Zhou
Hdata$price[Hdata$price == 0] = NA
date <- as.Date(Hdata$date)

timeSpanSalesCover <- max(date) - min(date)
timeSpanSalesCover

#by looking into the data, only the highest number year 2005 and lowerst number 1885 mak
e logical sense.

timeSpanConstruction <- max(as.numeric(year), na.rm = TRUE) - min(as.numeric(year), na.r
m = TRUE)
timeSpanConstruction
#timespan for construction is 120 years

library(ggplot2)

date_sold <- ymd(Hdata$date)


price_sold <- Hdata$price

date_df <- data.frame(date = date_sold,
                      The_year_sold <- as.numeric(format(date_sold, format = "%Y")),
                      the_month_sold <- as.numeric(format(date_sold, format = "%m")),
                      The_day_sold <- as.numeric(format(date_sold, format = "%d" )))

#source: https://stackoverflow.com/questions/4078502/split-date-data-m-d-y-into-3-separa
te-columns

sold_in_2003_ind <- which(date_df$The_year_sold....as.numeric.format.date_sold..forma
t.....Y... == 2003)
```

```
sold_in_2003 <- date[sold_in_2003_ind]
price_sold_2003 <- price_sold[sold_in_2003_ind]#FIXME EACH MON

sold_in_2004_ind <- which(date_df$The_year_sold....as.numeric.format.date_sold..forma
t.....Y... == 2004)
sold_in_2004 <- date[sold_in_2004_ind]
price_sold_2004 <- price_sold[sold_in_2004_ind]#FIXME EACH MON

sold_in_2005_ind <- which(date_df$The_year_sold....as.numeric.format.date_sold..forma
t.....Y... == 2005)
sold_in_2005 <- date[sold_in_2005_ind]
price_sold_2005 <- price_sold[sold_in_2005_ind]#FIXME EACH MON

sold_in_2006_ind <- which(date_df$The_year_sold....as.numeric.format.date_sold..forma
t.....Y... == 2006)
sold_in_2006 <- date[sold_in_2006_ind]
price_sold_2006 <- price_sold[sold_in_2006_ind]#FIXME EACH MON
#
#########Number of sales over time
par(mfrow = c(2,2))
hist(sold_in_2003 , breaks = "month")
hist(sold_in_2004 , breaks = "month")
hist(sold_in_2005 , breaks = "month")
hist(sold_in_2006 , breaks = "month")

df1 = data.frame(sold_in_2003, price_sold_2003)
df2 = data.frame(sold_in_2004, price_sold_2004)
df3 = data.frame(sold_in_2005, price_sold_2005)
df4 = data.frame(sold_in_2006, price_sold_2006)
########### Average over time

ggplot(df1 , aes(df1$sold_in_2003, df1$price_sold_2003/1000000)) + geom_bar(stat = "iden
tity") + xlab("Sold in 2003") + ylab("Price sold in million $")
ggplot(df2 , aes(df2$sold_in_2004, df2$price_sold_2004/1000000)) + geom_bar(stat = "iden
tity") + xlab("Sold in 2004") + ylab("Price sold in million $")
ggplot(df3 , aes(df3$sold_in_2005, df3$price_sold_2005/1000000)) + geom_bar(stat = "iden
tity") + xlab("Sold in 2005") + ylab("Price sold in million $")
ggplot(df4 , aes(df4$sold_in_2006, df4$price_sold_2006/1000000)) + geom_bar(stat = "iden
tity") + xlab("Sold in 2006") + ylab("Price sold in million $")

library(zoom)
year_level <- date[c(sold_in_2003_ind , sold_in_2004_ind , sold_in_2005_ind)]

qn4_df <- data.frame( theYear = Hdata$date,
                      bedroom = Hdata$br,
                      county = Hdata$county,
                      price = Hdata$price)

below_06 = subset(data.frame(Hdata[, c("county", "price", "br")], year = year(Hdata$dat
e)%%100), year(Hdata$date) < 2006)
below_06_county = split(below_06, below_06$county)

below_06_county_br = lapply(below_06_county, function(x) split(x, x$br))
```

```
below_06_county_br_year = lapply(below_06_county_br, function(x) lapply(x, function(y) t
apply(y$price, y$year, mean)))
list_name = names(unlist(below_06_county_br_year))
below_06_df = data.frame(avg_price = as.numeric(unlist(below_06_county_br_year)), do.cal
l("rbind", strsplit(list_name, "\\.")))
names(below_06_df) = c("avg_price", "county", "bedrooms", "year")

below_06_df$bedrooms = factor(below_06_df$bedrooms)

ggplot(below_06_df, aes(x = year, y = avg_price, col = bedrooms)) + geom_line(aes(group
 = bedrooms)) + facet_grid(. ~ county) + ylab("Average Price") + xlab("Year in 2000+")

city_county = sapply(split(Hdata$county, Hdata$city), function(x) length(unique(as.chara
cter(x))))

which(city_county > 1)

# For city: "San Francisco"
as.character(unique(subset(Hdata$county, Hdata$city == "San Francisco")))

# For city: "Vallejo"
as.character(unique(subset(Hdata$county, Hdata$city == "Vallejo")))
m1 = lm(price~bsqft, data = Hdata)
par(mfrow=c(1,2))
plot(m1, which = c(1,2)) # suggest transformation

library(MASS)
par(mfrow=c(1,1))
boxcox(m1) # log transformation is suggested

lm_df = data.frame(x = Hdata$bsqft, y = log(Hdata$price))
lm_df = lm_df[complete.cases(lm_df),]

m2 = lm(y~x, data = lm_df)
par(mfrow=c(1,2))
plot(m2, which = c(1,2))

# remove outliers
lm_df_clean = lm_df[-1*as.numeric(which(rstandard(m2) < -5)),]

m3 = lm(y~x, data = lm_df_clean)
par(mfrow=c(1,2))
plot(m3, which = c(1,2)) # final model

m4 = lm(price ~ lsqft + bsqft, data = Hdata)
summary(m4)

beta_b = summary(m4)$coef[3,1]
beta_l = summary(m4)$coef[2,1]
se_b = summary(m4)$coef[3,2]
se_l = summary(m4)$coef[2,2]

ts = (beta_b-beta_l)/sqrt(se_b^2+se_l^2)
n = length(m4$residuals) # 15602
```

```
pt(ts, n-3) # p-value close to 1
# fail to reject H0 at 5% significance level because p-value > 0.05

#has extra unwanted chart
#q8_df = split(Hdata[,c("price", "bsqft")], Hdata$county)
#lm_coef = do.call("rbind", lapply(q8_df, function(x) summary(lm(x$price~x$bsqft))$coef[,1]))

#plot(Hdata$bsqft, Hdata$price, xlab = "building size of the house", ylab = "sale price", main = "Sale price vs building size \n with fitted regression lines by county")
#sapply(1:nrow(lm_coef), function(x) abline(a = lm_coef[x,1], b = lm_coef[x, 2], col=x))
#legend("bottomright", rownames(lm_coef), col=1:nrow(lm_coef), lty=1, cex = 0.8)

ggplot(Hdata, aes(x = bsqft, y = price, col = county)) +
geom_point(alpha = 0.4, col = "black", na.rm = T) + geom_smooth(method = 'lm', se = F, na.rm = T) +
labs(title = "Price by Building Square Footage For Each County",
x = "Building Square Footage", y = "Price of Homes")


library(treemap)

Hdata$city = as.character(Hdata$city)
county_split = split(Hdata[,c("county", "city", "price")], Hdata$county)
high_freq_city_list = lapply(county_split, function(x) names(sort(table(x$city), decreasing=TRUE)[1:3]))
high_freq_city = as.character(unlist(high_freq_city_list))

Hdata_f = Hdata[Hdata$city %in% high_freq_city,]
county_split_f = split(Hdata_f[,c("county", "city", "price")], Hdata_f$county)
avg = lapply(county_split_f, function(x) tapply(x$price, x$city, function(y) mean(y, na.rm=TRUE)))

q9_df = data.frame(price = as.numeric(unlist(avg)), do.call("rbind", strsplit(names(unlist(avg)), "\\.")))
names(q9_df) = c("price", "county", "city")

treemap(q9_df,
        index=c("county", "city"),
        vSize="price",
        vColor="price",
        type="value",
        format.legend = list(scientific = FALSE, big.mark = " "))
#  source code: piazza. @252

library(maps)

SFdata<-subset(Hdata,Hdata$city=="San Francisco")
SFdata$long2<-round(as.numeric(as.character(SFdata$long)),2)
SFdata$lat2<-round(as.numeric(as.character(SFdata$lat)),2)

long_range<-range(SFdata$long2,na.rm=TRUE)
long_seq<-seq(long_range[1],long_range[2],.01)
SFdata$longF<-factor(SFdata$long2,levels=long_seq)
```

```
lat_range<-range(SFdata$lat2,na.rm = T)
lat_seq<-seq(lat_range[1],lat_range[2],.01)
SFdata$latF<-factor(SFdata$lat2,levels=lat_seq)

SF_agg<-aggregate(price~latF+longF,SFdata,function(x) c(mean(x),length(x)),drop=FALSE)
house_prices<-matrix(SF_agg$price[,1],nlevels(SFdata$longF),nlevels(SFdata$latF),byrow=TRUE)
house_counts<-matrix(SF_agg$price[,2],nlevels(SFdata$longF),nlevels(SFdata$latF),byrow=TRUE)
# heatmap for average housing price
sf_border=map('county','california,san francisco',plot = F)
image(x = as.numeric(levels(SFdata$longF)) + .03, y = as.numeric(levels(SFdata$latF)), z = house_prices, #create heatmap
      xlab = "Longitude", ylab = "Latitude", main = "San Francisco Heatmap of Average Home Prices")
lines(sf_border$x,sf_border$y)
# heatmap for # of sale records
sf_border=map('county','california,san francisco',plot = F)
image(x = as.numeric(levels(SFdata$longF)) + .03, y = as.numeric(levels(SFdata$latF)), z = house_counts, #create heatmap
      xlab = "Longitude", ylab = "Latitude", main = "San Francisco Heatmap of Number of Sale Records")
lines(sf_border$x,sf_border$y)
```