DataReview_Sarah

Sarah Christen

2024-11-18

```
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 become errors
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(patchwork)
library(gapminder)
library(scales)
library(knitr)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
setwd("~/uvaMSDS/stat6021/project2/Project2_Group2_STAT6021")
Data <- read.csv("kc_house_data.csv",header = TRUE)
```

```
which(is.na(Data)) # no N/As in Data
```

```
## integer(0)
```

```
colnames(Data)
```

```
##  [1] "id"            "date"          "price"         "bedrooms"
##  [5] "bathrooms"     "sqft_living"   "sqft_lot"      "floors"
##  [9] "waterfront"    "view"          "condition"     "grade"
## [13] "sqft_above"    "sqft_basement" "yr_built"      "yr_renovated"
## [17] "zipcode"       "lat"           "long"          "sqft_living15"
## [21] "sqft_lot15"
```

```
Data$waterfront <- factor(Data$waterfront)
Data$view <- factor(Data$view)
Data$condition <- factor(Data$condition)
Data$grade <- factor(Data$grade)

houseSales <- nrow(Data)
houses <- length(unique(Data$id))
duplicate_house_id <- Data |>
  group_by(id) |>
  filter(n() > 1) |>
  ungroup()

renovated <- Data[Data$yr_renovated!=0,]
```

```
bedroomData<-Data%>%
  group_by(bedrooms)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
bedroomData
```

```
## # A tibble: 13 x 3
##    bedrooms Counts   Percent
##       <int>  <int>     <dbl>
##  1        0     13 0.000601
##  2        1    199 0.00921
##  3        2   2760 0.128
##  4        3   9824 0.455
##  5        4   6882 0.318
```

```
##  6        5   1601 0.0741
##  7        6    272 0.0126
##  8        7     38 0.00176
##  9        8     13 0.000601
## 10        9      6 0.000278
## 11       10      3 0.000139
## 12       11      1 0.0000463
## 13       33      1 0.0000463
```

```
bathroomData<-Data%>%
  group_by(bathrooms)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
bathroomData
```

```
## # A tibble: 30 x 3
##    bathrooms Counts  Percent
##        <dbl>  <int>    <dbl>
##  1      0        10 0.000463
##  2      0.5       4 0.000185
##  3      0.75     72 0.00333
##  4      1      3852 0.178
##  5      1.25      9 0.000416
##  6      1.5    1446 0.0669
##  7      1.75   3048 0.141
##  8      2      1930 0.0893
##  9      2.25   2047 0.0947
## 10      2.5    5380 0.249
## # i 20 more rows
```

```
floorData<-Data%>%
  group_by(floors)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
floorData
```

```
## # A tibble: 6 x 3
##   floors Counts  Percent
##    <dbl>  <int>    <dbl>
## 1    1    10680 0.494
## 2    1.5   1910 0.0884
## 3    2     8241 0.381
## 4    2.5    161 0.00745
## 5    3      613 0.0284
## 6    3.5      8 0.000370
```

```
waterfrontData<-Data%>%
  group_by(waterfront)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
waterfrontData
```

```
## # A tibble: 2 x 3
```

```
##   waterfront Counts Percent
##   <fct>       <int>   <dbl>
## 1 0           21450 0.992
## 2 1             163 0.00754
```

```
viewData<-Data%>%
  group_by(view)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
viewData
```

```
## # A tibble: 5 x 3
##   view  Counts Percent
##   <fct>  <int>   <dbl>
## 1 0      19489  0.902
## 2 1        332  0.0154
## 3 2        963  0.0446
## 4 3        510  0.0236
## 5 4        319  0.0148
```

```
conditionData<-Data%>%
  group_by(condition)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
conditionData
```

```
## # A tibble: 5 x 3
##   condition Counts Percent
##   <fct>      <int>   <dbl>
## 1 1             30 0.00139
## 2 2            172 0.00796
## 3 3          14031 0.649
## 4 4           5679 0.263
## 5 5           1701 0.0787
```

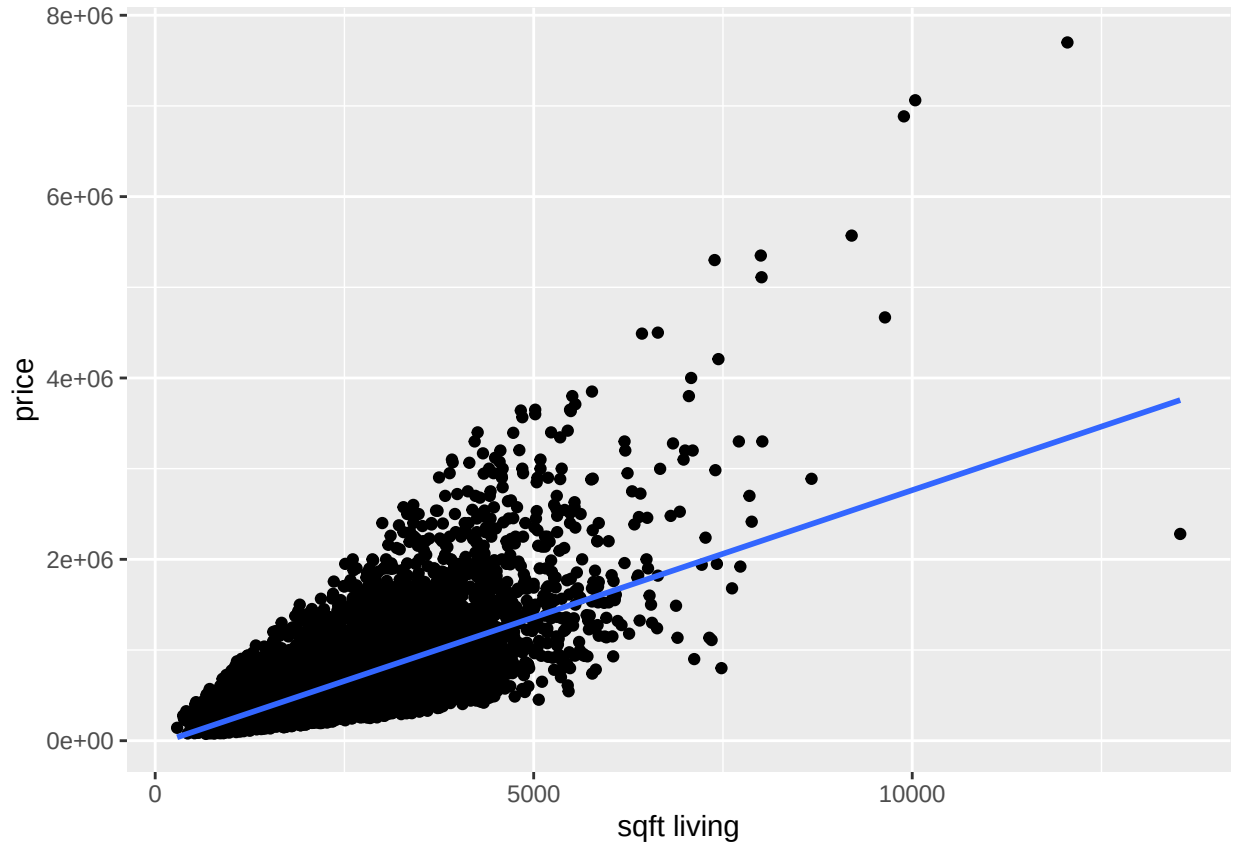
```
gradeData<-Data%>%
  group_by(grade)%>%
  summarize(Counts=n())%>%
  mutate(Percent=Counts/nrow(Data))
gradeData
```

```
## # A tibble: 12 x 3
##    grade Counts   Percent
##    <fct>  <int>     <dbl>
##  1 1          1 0.0000463
##  2 3          3 0.000139
##  3 4         29 0.00134
##  4 5        242 0.0112
##  5 6       2038 0.0943
##  6 7       8981 0.416
##  7 8       6068 0.281
##  8 9       2615 0.121
```

```
##  9 10      1134 0.0525
## 10 11       399 0.0185
## 11 12        90 0.00416
## 12 13        13 0.000601
```

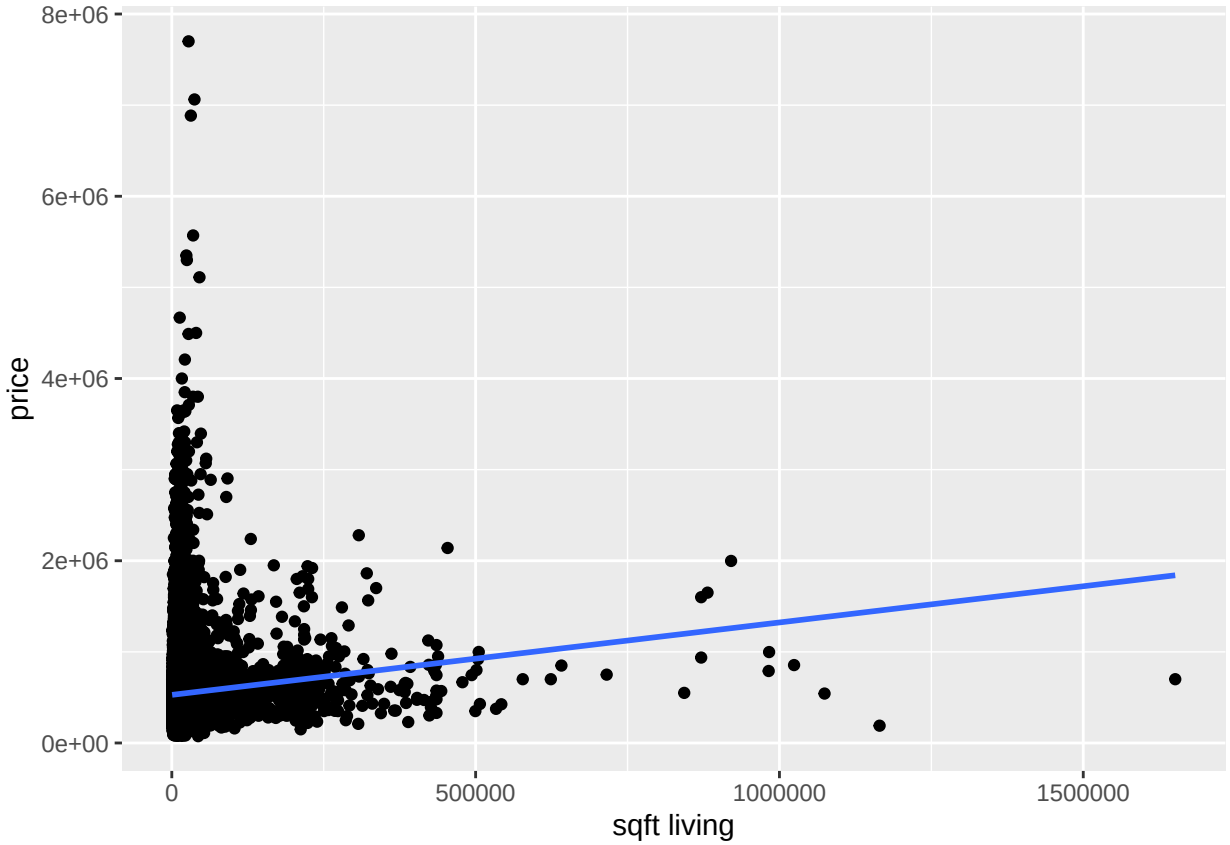
```
ggplot2::ggplot(Data, aes(x=sqft_living, y=price))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)+
  labs(x="sqft living", y="price")
```

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




```
ggplot2::ggplot(Data, aes(x=sqft_lot, y=price))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)+
  labs(x="sqft living", y="price")
```

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




```
price_mean <- dollar_format()(mean(Data$price,na.rm = TRUE))
price_median <- dollar_format()(median(Data$price,na.rm = TRUE))
price_min <- dollar_format()(min(Data$price))
price_max <- dollar_format()(max(Data$price))

bedrooms_mean <- round(mean(Data$bedrooms,na.rm = TRUE),2)
bedrooms_median <- round(median(Data$bedrooms,na.rm = TRUE),2)
bedrooms_min <- round(min(Data$bedrooms),2)
bedrooms_max <- round(max(Data$bedrooms),2)

bathrooms_mean <- round(mean(Data$bathrooms,na.rm = TRUE),2)
bathrooms_median <- round(median(Data$bathrooms,na.rm = TRUE),2)
bathrooms_min <- round(min(Data$bathrooms),2)
bathrooms_max <- round(max(Data$bathrooms),2)

sqftliving_mean <- round(mean(Data$sqft_living,na.rm = TRUE),2)
sqftliving_median <- round(median(Data$sqft_living,na.rm = TRUE),2)
sqftliving_min <- round(min(Data$sqft_living),2)
sqftliving_max <- round(max(Data$sqft_living),2)

sqftlot_mean <- round(mean(Data$sqft_lot,na.rm = TRUE),2)
sqftlot_median <- round(median(Data$sqft_lot,na.rm = TRUE),2)
sqftlot_min <- round(min(Data$sqft_lot),2)
sqftlot_max <- round(max(Data$sqft_lot),2)

floors_mean <- round(mean(Data$floors,na.rm = TRUE),2)
```

```
floors_median <- round(median(Data$floors,na.rm = TRUE),2)
floors_min <- round(min(Data$floors),2)
floors_max <- round(max(Data$floors),2)

view_mean <- round(mean(as.numeric(Data$view),na.rm = TRUE),2)
view_median <- round(median(as.numeric(Data$view),na.rm = TRUE),2)
view_min <- round(min(as.numeric(Data$view)),2)
view_max <- round(max(as.numeric(Data$view)),2)

condition_mean <- round(mean(as.numeric(Data$condition),na.rm = TRUE),2)
condition_median <- round(median(as.numeric(Data$condition),na.rm = TRUE),2)
condition_min <- round(min(as.numeric(Data$condition)),2)
condition_max <- round(max(as.numeric(Data$condition)),2)

grade_mean <- round(mean(as.numeric(Data$grade),na.rm = TRUE),2)
grade_median <- round(median(as.numeric(Data$grade),na.rm = TRUE),2)
grade_min <- round(min(as.numeric(Data$grade)),2)
grade_max <- round(max(as.numeric(Data$grade)),2)

sqftabove_mean <- round(mean(Data$sqft_above,na.rm = TRUE),2)
sqftabove_median <- round(median(Data$sqft_above,na.rm = TRUE),2)
sqftabove_min <- round(min(Data$sqft_above),2)
sqftabove_max <- round(max(Data$sqft_above),2)

sqftbasement_mean <- round(mean(Data$sqft_basement,na.rm = TRUE),2)
sqftbasement_median <- round(median(Data$sqft_basement,na.rm = TRUE),2)
sqftbasement_min <- round(min(Data$sqft_basement),2)
sqftbasement_max <- round(max(Data$sqft_basement),2)

yrbuilt_mean <- round(mean(Data$yr_built,na.rm = TRUE),2)
yrbuilt_median <- round(median(Data$yr_built,na.rm = TRUE),2)
yrbuilt_min <- round(min(Data$yr_built),2)
yrbuilt_max <- round(max(Data$yr_built),2)

sqftliving15_mean <- round(mean(Data$sqft_living15,na.rm = TRUE),2)
sqftliving15_median <- round(median(Data$sqft_living15,na.rm = TRUE),2)
sqftliving15_min <- round(min(Data$sqft_living15),2)
sqftliving15_max <- round(max(Data$sqft_living15),2)

sqftlot15_mean <- round(mean(Data$sqft_lot15,na.rm = TRUE),2)
sqftlot15_median <- round(median(Data$sqft_lot15,na.rm = TRUE),2)
sqftlot15_min <- round(min(Data$sqft_lot15),2)
sqftlot15_max <- round(max(Data$sqft_lot15),2)

variableNames <- c('Price','Bedrooms','Bathrooms','Sqft Living','Sqft Lot','Floors','View','Condition',
varMean <- c(price_mean,bedrooms_mean,bathrooms_mean,sqftliving_mean,sqftlot_mean,floors_mean,view_mean
varMedian <- c(price_median,bedrooms_median,bathrooms_median,sqftliving_median,sqftlot_median,floors_me
varMin <- c(price_min,bedrooms_min,bathrooms_min,sqftliving_min,sqftlot_min,floors_min,view_min,conditi
varMax <- c(price_max,bedrooms_max,bathrooms_max,sqftliving_max,sqftlot_max,floors_max,view_max,conditi
summary_variables <- data.frame(variableNames,varMean,varMedian,varMin,varMax)
colnames(summary_variables) <- c('Variable','Mean','Median','Minimum','Maximum')

kable(summary_variables)
```

Variable	Mean	Median	Minimum	Maximum
Price	$540,088	$450,000	$75,000	$7,700,000
Bedrooms	3.37	3	0	33
Bathrooms	2.11	2.25	0	8
Sqft Living	2079.9	1910	290	13540
Sqft Lot	15106.97	7618	520	1651359
Floors	1.49	1.5	1	3.5
View	1.23	1	1	5
Condition	3.41	3	1	5
Grade	6.66	6	1	12
Sqft Above	1788.39	1560	290	9410
Sqft Basement	291.51	0	0	4820
Yr Built	1971.01	1975	1900	2015
Sqft Living 15	1986.55	1840	399	6210
Sqft Lot 15	12768.46	7620	651	871200