HW7

Andrew Shao

2024-10-16

7.4.1 Q1

```
library(r02pro)
```

```
## Warning: package 'r02pro' was built under R version 4.3.3
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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
```

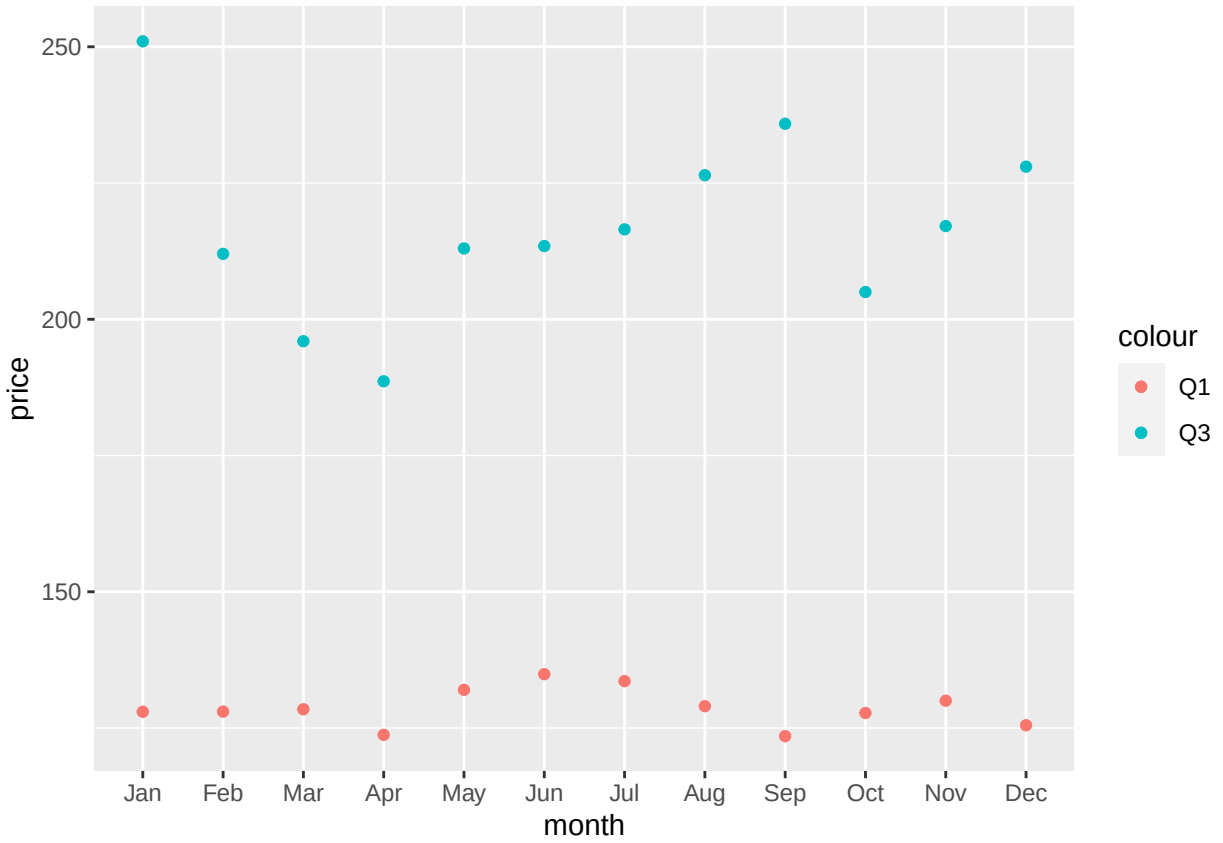
```
ahp %>%
  mutate(age = yr_sold - yr_built) %>%
  select(age, sale_price, kit_qual) %>%
  ggplot() +
  geom_point(aes(x = age,
                 y = sale_price,
                 color = kit_qual))
```

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

As age increases, sale price generally gets lower and the proportion of houses with higher kitchen quality goes down. Also price is generaly higher for houses with better kitchen quality.

7.5.3 Q1

```
ahp %>%
  group_by(mo_sold) %>%
  summarize(price_q1 = quantile(sale_price, 0.25, na.rm = T),
            price_q3 = quantile(sale_price, 0.75, na.rm = T)) %>%
  ggplot(aes(month.abb[mo_sold])) +
  geom_point(aes(y = price_q1, color = 'Q1')) +
  geom_point(aes(y = price_q3, color = 'Q3')) +
  scale_x_discrete(limits = month.abb) +
  xlab('month') +
  ylab('price')
```



The first quartile doesn't vary much , but the third quartile peaks in January and September (School start times?) with the lowest in April.

7.5.3 Q2

```
ahp %>%
  mutate(less_30yo = (yr_sold - yr_built) <= 30) %>%
  remove_missing(vars = c('less_30yo', 'sale_price')) %>%
  group_by(less_30yo) %>%
  summarize(min_price = min(sale_price),
            med_price = median(sale_price),
            max_price = max(sale_price))
```

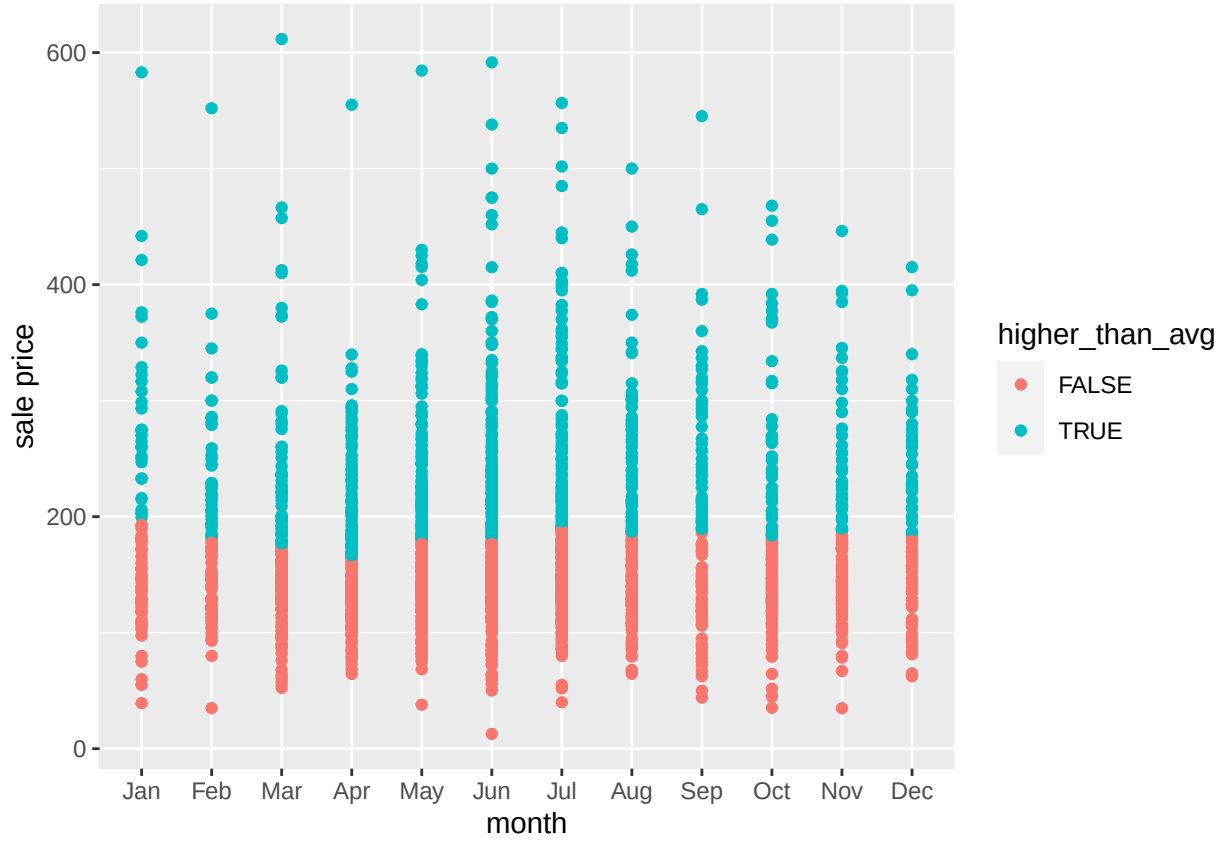
```
## Warning: Removed 5 rows containing missing values.
```

```
## # A tibble: 2 x 4
##   less_30yo min_price med_price max_price
##   <lgl>          <dbl>     <dbl>     <dbl>
## 1 FALSE           12.8       135      475
## 2 TRUE           114.        211      612.
```

7.6.3 Q1

```
ahp %>%
  group_by(mo_sold) %>%
  remove_missing(vars = c('mo_sold', 'sale_price')) %>%
  mutate(higher_than_avg = sale_price > mean(sale_price)) %>%
  ggplot() +
  geom_point(aes(x = month.abb[mo_sold],
                 y = sale_price,
                 color = higher_than_avg)) +
  scale_x_discrete(limits = month.abb) +
  xlab('month') +
  ylab('sale price')
```

```
## Warning: Removed 4 rows containing missing values.
```



There is more higher spread of prices in houses with higher than average prices. The average doesn't fluctuate very much.

7.6.3 Q2

```
ahp %>%
  group_by(oa_cond) %>%
```

```
  summarise(n = n(),
            avg_sale_price = mean(sale_price, na.rm = T)) %>%
  filter(n >= 30)
```

```
## # A tibble: 6 x 3
##   oa_cond     n avg_sale_price
##     <dbl> <int>          <dbl>
## 1       3    35           99.8
## 2       4    70          114.
## 3       5  1165          207.
## 4       6   367          149.
## 5       7   270          155.
## 6       8   101          156.
```

7.6.3 Q3

```
ahp %>%
  group_by(yr_remodel) %>%
  mutate(r = rank(desc(sale_price), ties.method = 'first')) %>%
  filter(r <= 2) %>%
  select(yr_remodel, sale_price) %>%
  arrange(yr_remodel, desc(sale_price))
```

```
## # A tibble: 122 x 2
## # Groups:   yr_remodel [61]
##    yr_remodel sale_price
##         <dbl>      <dbl>
##  1       1950       257.
##  2       1950       256
##  3       1951       155
##  4       1951       141
##  5       1952       166
##  6       1952       146.
##  7       1953       225
##  8       1953       217
##  9       1954       156.
## 10       1954       150.
## # i 112 more rows
```

7.6.3 Q4

```
ahp %>%
  group_by(kit_qual, central_air) %>%
  mutate(r_asc = rank(sale_price, ties.method = 'first'),
         r_desc = rank(desc(sale_price), ties.method = 'first'),
         max_price = max(sale_price, na.rm = T)) %>%
  filter(r_asc == 1 | r_desc == 1) %>%
  arrange(desc(max_price), sale_price) %>%
  select(!c(r_asc, r_desc, max_price))
```

```
## # A tibble: 16 x 56
## # Groups:   kit_qual, central_air [8]
##    dt_sold    yr_sold mo_sold yr_built yr_remodel bldg_class bldg_type
##    <date>       <dbl>   <dbl>    <dbl>      <dbl>      <dbl> <chr>
##  1 2009-05-03    2009       5     1917       2007         70 1Fam
##  2 2010-03-03    2010       3     2009       2010         20 1Fam
##  3 2010-04-03    2010       4     1946       2006         20 1Fam
##  4 2007-06-14    2007       6     2006       2007         20 1Fam
##  5 2007-09-05    2007       9     1910       1950         50 1Fam
##  6 2010-03-07    2010       3     1959       1997         20 1Fam
##  7 2006-06-10    2006       6     1920       1950         30 1Fam
##  8 2006-09-06    2006       9     1979       1979         20 1Fam
##  9 2010-06-10    2010       6     1923       1970         30 1Fam
## 10 2007-11-04    2007      11     1918       1950         70 1Fam
## 11 2007-04-19    2007       4     1946       1950         20 1Fam
## 12 2009-07-13    2009       7     1916       1994         75 1Fam
## 13 2008-06-18    2008       6     1920       1950         50 1Fam
## 14 2009-10-15    2009      10     1925       2007         50 1Fam
## 15 2007-01-18    2007       1     1946       1950         20 1Fam
## 16 2009-10-15    2009      10     1949       2005         50 1Fam
## # i 49 more variables: house_style <chr>, zoning <chr>, neighborhd <chr>,
## #   oa_cond <dbl>, oa_qual <dbl>, func <chr>, liv_area <dbl>, `1fl_area` <dbl>,
## #   `2fl_area` <dbl>, tot_rms <dbl>, bedroom <dbl>, bathroom <dbl>, kit <dbl>,
## #   kit_qual <chr>, central_air <chr>, elect <chr>, bsmt_area <dbl>,
## #   bsmt_cond <chr>, bsmt_exp <chr>, bsmt_fin_qual <chr>, bsmt_ht <chr>,
## #   ext_cond <chr>, ext_cover <chr>, ext_qual <chr>, fdn <chr>, fence <chr>,
## #   fp <dbl>, fp_qual <chr>, gar_area <dbl>, gar_car <dbl>, gar_cond <chr>, ...
```