Homework dataviz dsb10 phat

Phat

2024-07-23

Homework data transformation

dplyr 5 query

```
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(nycflights13)
```

find year , month , day , dep_time, arr_time and sched_dep_time, sched_arr_time of flights between 1 - 15 September and order by day

```
flights %>%
  select(year, month, day,dep_time, arr_time,sched_dep_time, sched_arr_time) %>%
  filter(month == 9 & day >= 1 & day <= 15) %>%
  arrange(day)
```

```
## # A tibble: 13,556 x 7
##     year month   day dep_time arr_time sched_dep_time sched_arr_time
##    <int> <int> <int>    <int>    <int>          <int>          <int>
##  1  2013     9     1        9      343           2359            340
##  2  2013     9     1      117      218           2245           2359
##  3  2013     9     1      508      717            516            800
##  4  2013     9     1      537      849            545            855
##  5  2013     9     1      537      906            545            921
##  6  2013     9     1      549      815            600            850
##  7  2013     9     1      552      843            600            905
##  8  2013     9     1      553      809            600            834
##  9  2013     9     1      554      700            600            716
## 10  2013     9     1      554      803            600            823
## # i 13,546 more rows
```

find top 3 airline name with most flight

```
flights %>%
  count(carrier) %>%
  arrange(-n) %>%
  head(3) %>%
  left_join(airlines)
```

```
## Joining with `by = join_by(carrier)`
```

```
## # A tibble: 3 x 3
##   carrier     n name
##   <chr>   <int> <chr>
## 1 UA      58665 United Air Lines Inc.
## 2 B6      54635 JetBlue Airways
## 3 EV      54173 ExpressJet Airlines Inc.
```

find mean of dep_delay and mean of arr_delay that group by carrier

```
flights %>%
  group_by(carrier) %>%
  drop_na() %>%
  summarise(mean(dep_delay),
            mean(arr_delay))
```

```
## # A tibble: 16 x 3
##    carrier `mean(dep_delay)` `mean(arr_delay)`
##    <chr>               <dbl>             <dbl>
##  1 9E                  16.4               7.38
##  2 AA                   8.57              0.364
##  3 AS                   5.83             -9.93
##  4 B6                  13.0               9.46
##  5 DL                   9.22              1.64
##  6 EV                  19.8              15.8
##  7 F9                  20.2              21.9
##  8 FL                  18.6              20.1
##  9 HA                   4.90             -6.92
## 10 MQ                  10.4              10.8
## 11 OO                  12.6              11.9
## 12 UA                  12.0               3.56
## 13 US                   3.74              2.13
## 14 VX                  12.8               1.76
## 15 WN                  17.7               9.65
## 16 YV                  18.9              15.6
```

find flights that total delay more than 1000 and sort it by total delay in DESC

```
flights %>%
  select(dep_delay,arr_delay,carrier) %>%
  mutate(total_delay = dep_delay + arr_delay) %>%
  filter(total_delay > 1000) %>%
  arrange(-total_delay)
```

```
## # A tibble: 52 x 4
##    dep_delay arr_delay carrier total_delay
```

```
##        <dbl>     <dbl> <chr>         <dbl>
##  1      1301      1272 HA             2573
##  2      1137      1127 MQ             2264
##  3      1126      1109 MQ             2235
##  4      1014      1007 AA             2021
##  5      1005       989 MQ             1994
##  6       960       931 DL             1891
##  7       911       915 DL             1826
##  8       898       895 DL             1793
##  9       896       878 AA             1774
## 10       878       875 MQ             1753
## # i 42 more rows
```

find max and min of total delay of each month

```
flights %>%
  mutate(total_delay = dep_delay + arr_delay) %>%
  group_by(month) %>%
  drop_na() %>%
  summarise(max(total_delay),
            min(total_delay))
```

```
## # A tibble: 12 x 3
##    month `max(total_delay)` `min(total_delay)`
##    <int>              <dbl>              <dbl>
##  1     1               2573                -74
##  2     2               1687                -91
##  3     3               1826                -74
##  4     4               1891                -79
##  5     5               1753               -100
##  6     6               2264                -78
##  7     7               1994                -80
##  8     8               1010                -78
##  9     9               2021                -82
## 10    10               1390                -71
## 11    11               1594                -80
## 12    12               1774                -71
```

Homework data visualization

prepare data

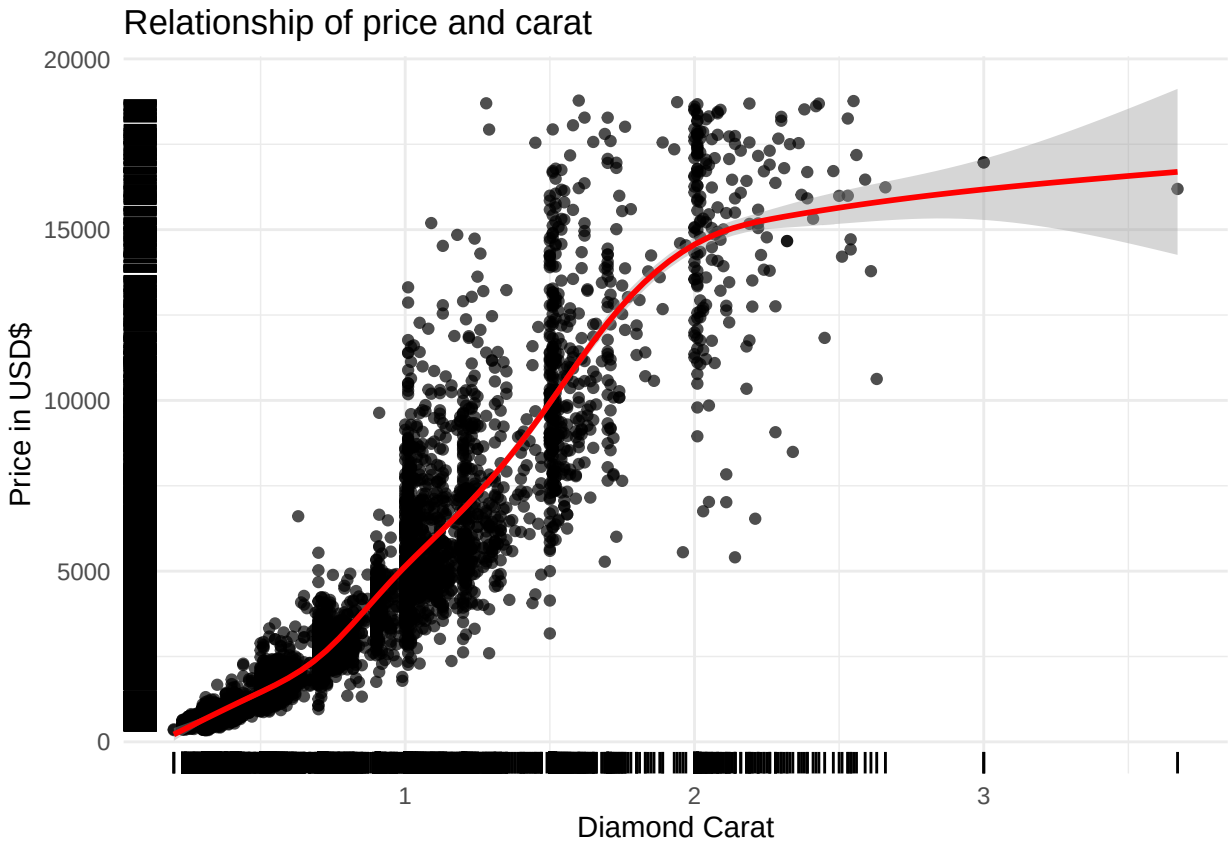
```
set.seed(40)
sample_diamonds = diamonds %>%
  sample_frac(0.1)
```

1.Visualize relationship of price and carat

```
sample_diamonds %>%
  ggplot(aes(x=carat,y = price)) +
  geom_point(alpha=0.7) +
  geom_smooth(col = 'red') +
  geom_rug() +
```

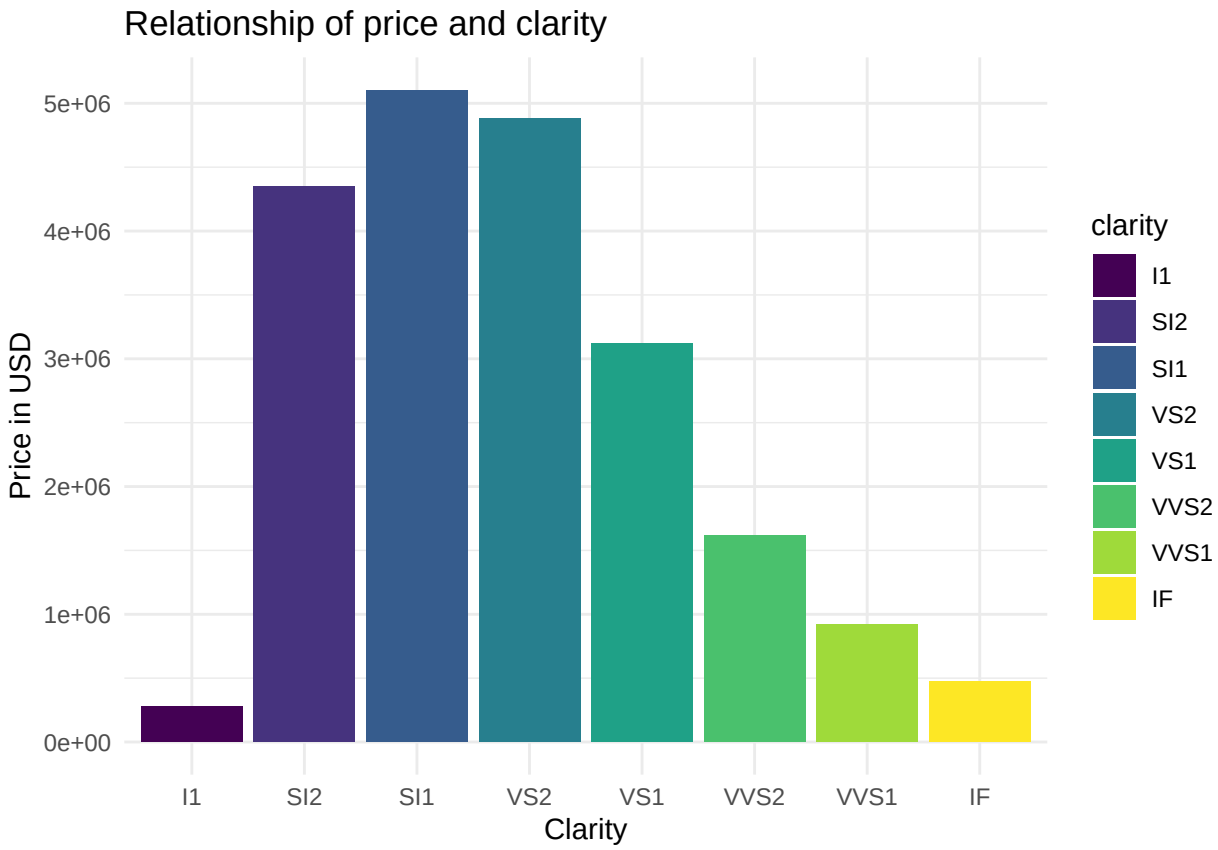
```
  theme_minimal() +
  labs(title = 'Relationship of price and carat',
       x = 'Diamond Carat',
       y = 'Price in USD$')
```

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



2.Visualize relationship of price and clarity

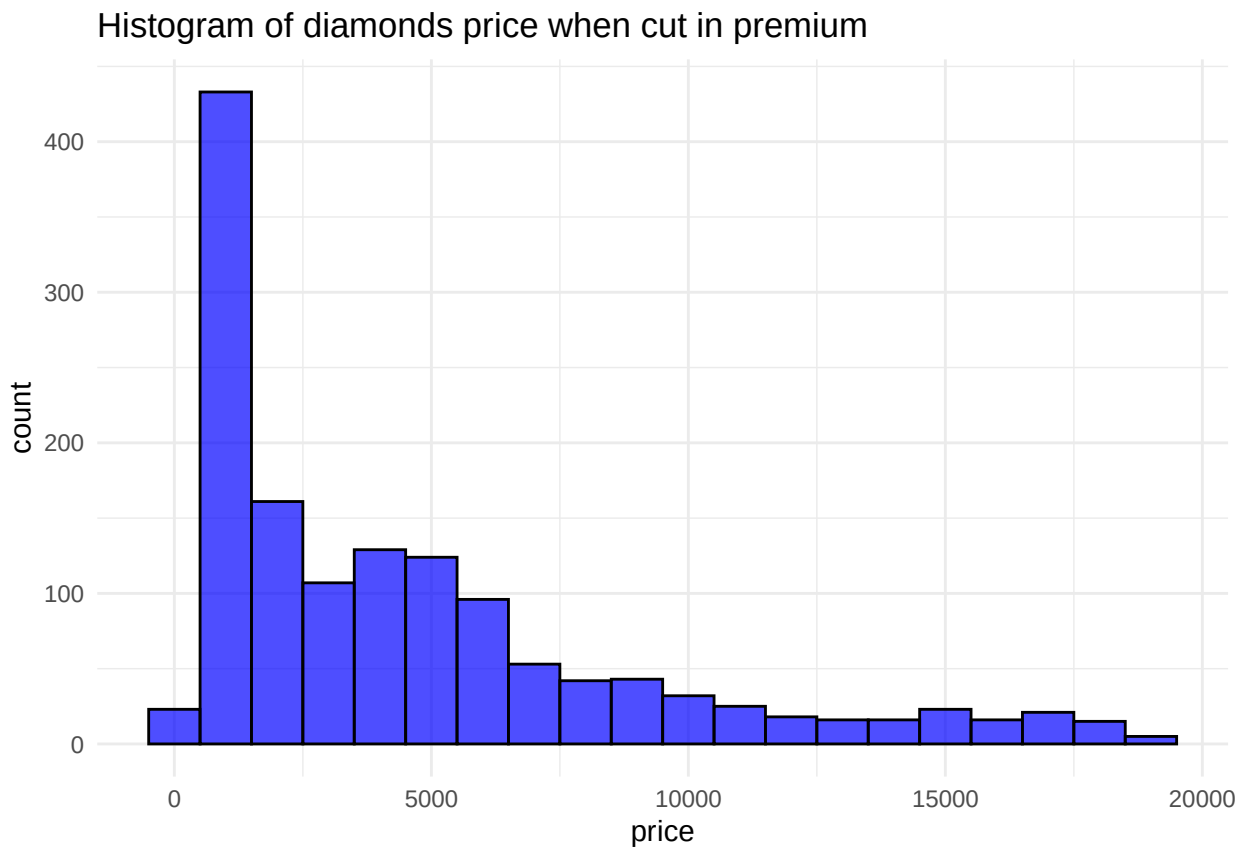
```
sample_diamonds %>%
  ggplot(aes(x=clarity,y=price,fill = clarity)) +
  geom_col() +
  theme_minimal() +
  labs(title = 'Relationship of price and clarity',
       x = 'Clarity',
       y = 'Price in USD')
```



3.Visualize histogram of carat when cut in premium

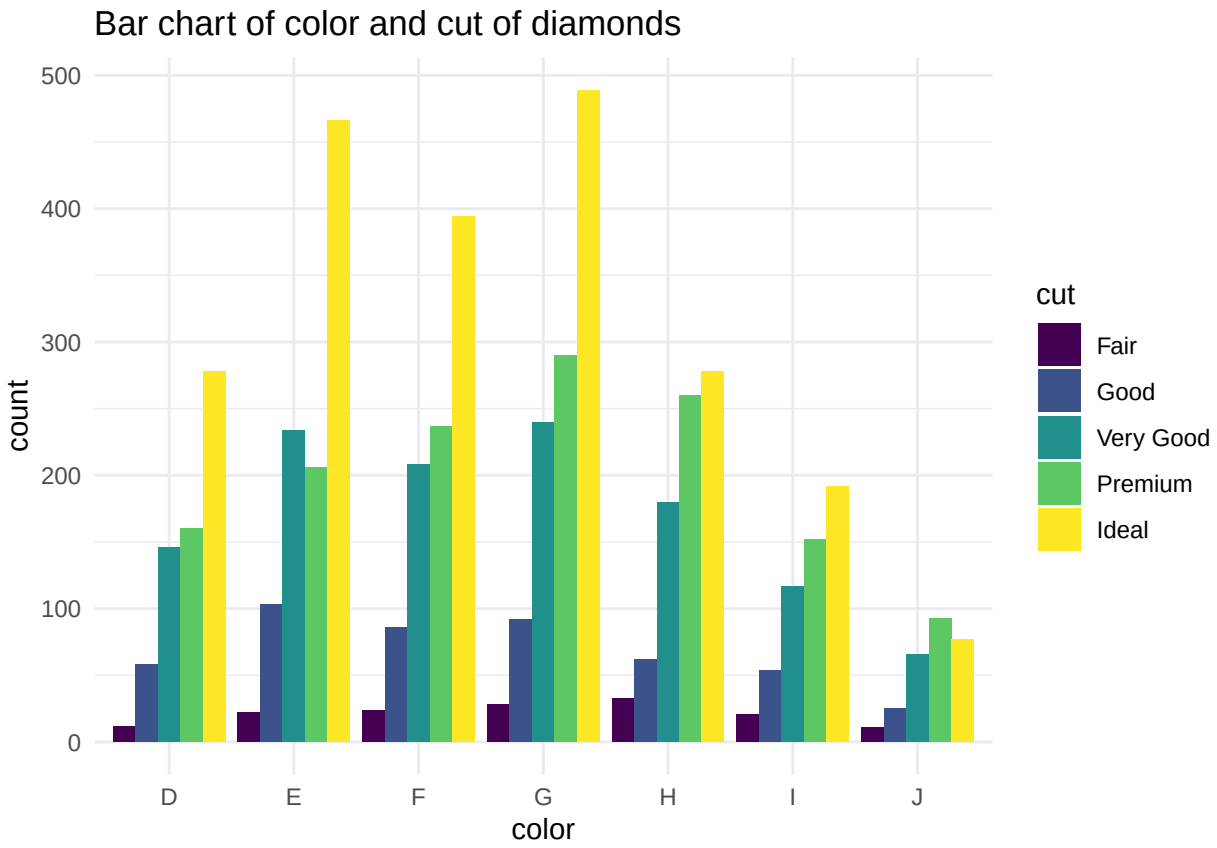
```
filter_diamonds = sample_diamonds %>%
  filter(cut == 'Premium')

filter_diamonds %>%
  ggplot(aes(price)) +
  geom_histogram(binwidth = 1000, color = 'black',fill = 'blue',alpha = 0.7) +
  theme_minimal() +
  labs(title = 'Histogram of diamonds price when cut in premium')
```



4.visualize color and cut by using bar charts

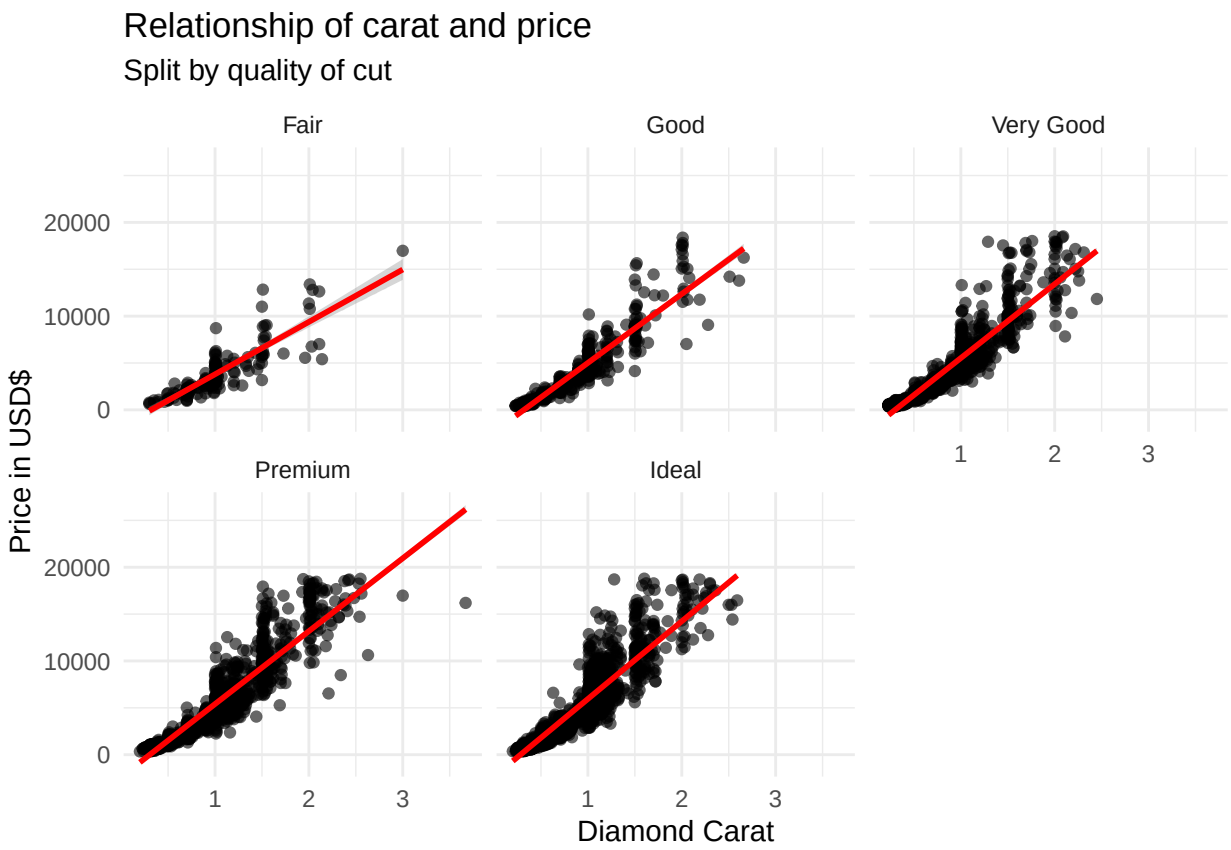
```
sample_diamonds %>%
  ggplot(aes(color,fill = cut)) +
  geom_bar(position = 'dodge') +
  theme_minimal() +
  labs(title = 'Bar chart of color and cut of diamonds')
```



5.visualize relation of carat and price split it by cut

```
sample_diamonds %>%
  ggplot(aes(x=carat,y=price)) +
  geom_point(alpha = 0.6,size = 1.5)+
  geom_smooth(method = 'lm',col = 'red') +
  facet_wrap(~cut,ncol = 3) +
  theme_minimal() +
  labs(title = 'Relationship of carat and price',
       subtitle = 'Split by quality of cut',
       x = 'Diamond Carat',
       y = 'Price in USD$')
```

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


Relationship of carat and price
Split by quality of cut
Fair
Good
Very Good
Premium
Ideal
Price in USD$
20000
10000
0
1
2
3
Diamond Carat