Homework_Data_Viz

```
### Library
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(glue)
library(ggeasy)
library(patchwork)
library(ggthemes)
```

```
#data.frame diamonds
data("diamonds")

## check data.frame ofdiamonds
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
sum(is.na(diamonds))
```

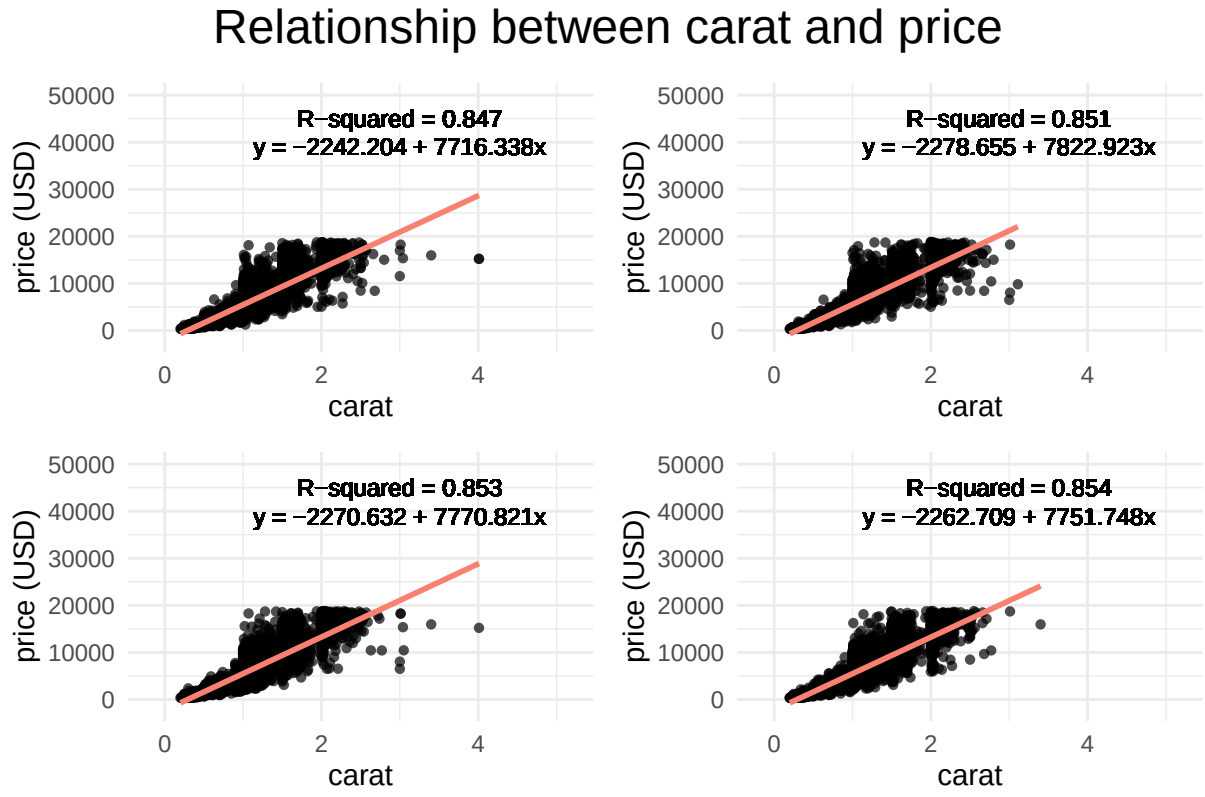
```
## [1] 0
```

First plot

Using For loop to create 4 data.frame and ggplot save as plot_i then combine plot with patchwork library and also adding main title and caption

```
############ For loop

for (i in 1:4) {
  small_diamonds <- sample_n(diamonds, 10000)

  lm_diamonds <- lm(price ~ carat, data =small_diamonds)

  lm_diamonds_R = round(summary(lm_diamonds)$r.squared,3)

  lm_diamonds_intercept <- round(lm_diamonds$coefficients["(Intercept)"],3)

  lm_diamonds_carat <- round(lm_diamonds$coefficients["carat"],3)

  ### First plot diamonds data-set
  small_plot <- ggplot(small_diamonds, aes(carat, price))+
    geom_point(pch = 16, alpha = 0.7) +
    geom_smooth(method = "lm", col = "salmon") +
    geom_text(x = 3, y = 45000,  label = glue("R-squared = {lm_diamonds_R}"), cex =3) +
    geom_text(x =3, y = 39000,
              label = glue("y = {lm_diamonds_intercept} + {lm_diamonds_carat}x"),
              cex = 3) +
    xlim(-0.2,5.2) + ylim(-2000,50000) +
    labs(#title = "Relationship between carat and price",
         x = "carat",
         y = "price (USD)",
         #caption = "Source: Diamonds from ggplot2 package",
         family ="Serif", size = 18) +
    theme_minimal() +
    #ggeasy::easy_center_title() +
    theme(plot.title = element_text(size = 20))
  assign(paste("plot",i, sep = "_"), small_plot)
  i = i + 1
}

### using patchwork library to combine all ggplot
plot_all <- (plot_1 + plot_2) / (plot_3 + plot_4)

## Adding main title, caption and arrange main title position
plot_all + plot_annotation(title = "Relationship between carat and price",
                           caption = "Source: Diamonds from ggplot2 package",
                           theme = theme(plot.title = element_text(size = 18, hjust = 0.5)))
```

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



Source: Diamonds from ggplot2 package

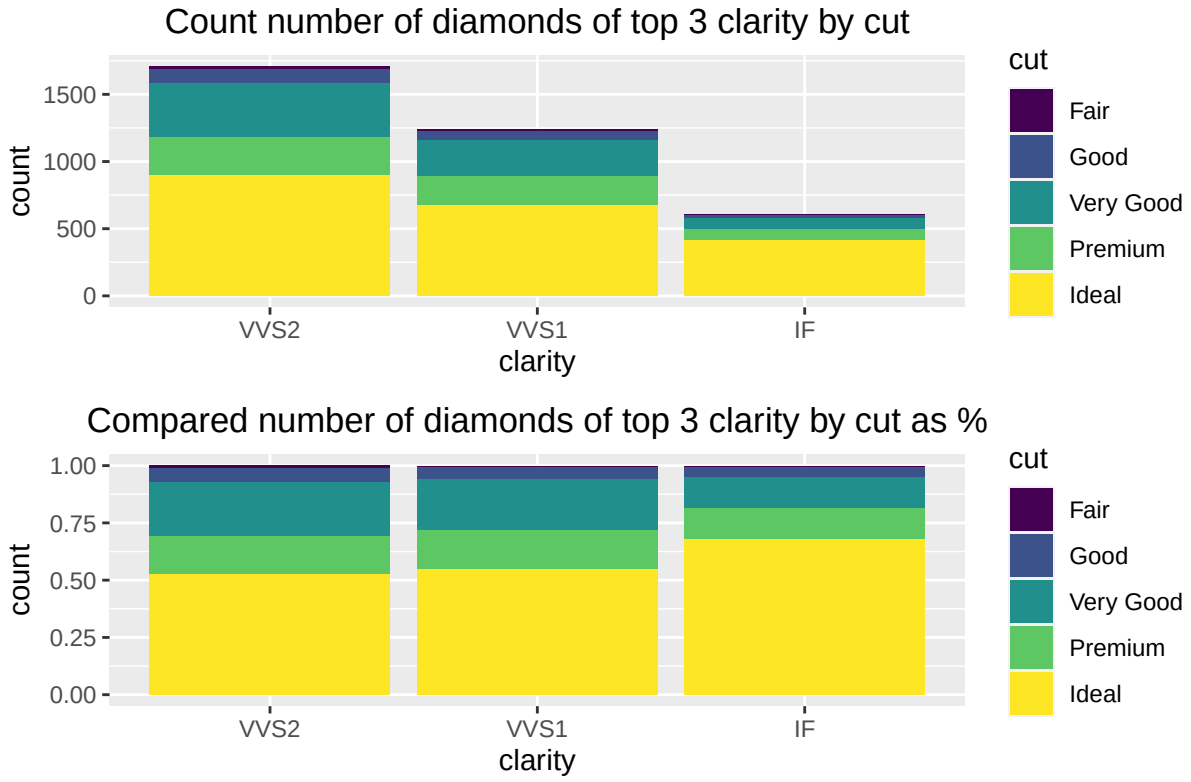
Second Plot

clarity = measurement of how clear the diamond is

I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best)

This plot were select only top 3 (VVS2, VVS1, IF) clarity to plot

```
clarity_filter <- diamonds %>%
                    filter(clarity == c("VVS2", "VVS1", "IF"))

clarity_plot <- ggplot(clarity_filter, aes(clarity, fill = cut))
clarity_plot1 <- clarity_plot + geom_bar() +
                 labs(title = "Count number of diamonds of top 3 clarity by cut") +
                 ggeasy::easy_center_title()

clarity_plot2 <- clarity_plot + geom_bar(position = "fill") +
                 labs(title = "Compared number of diamonds of top 3 clarity by cut as %") +
                 ggeasy::easy_center_title()


(clarity_plot1 / clarity_plot2) +
  plot_annotation(caption = "source: diamonds data.frame from ggplot2")
```

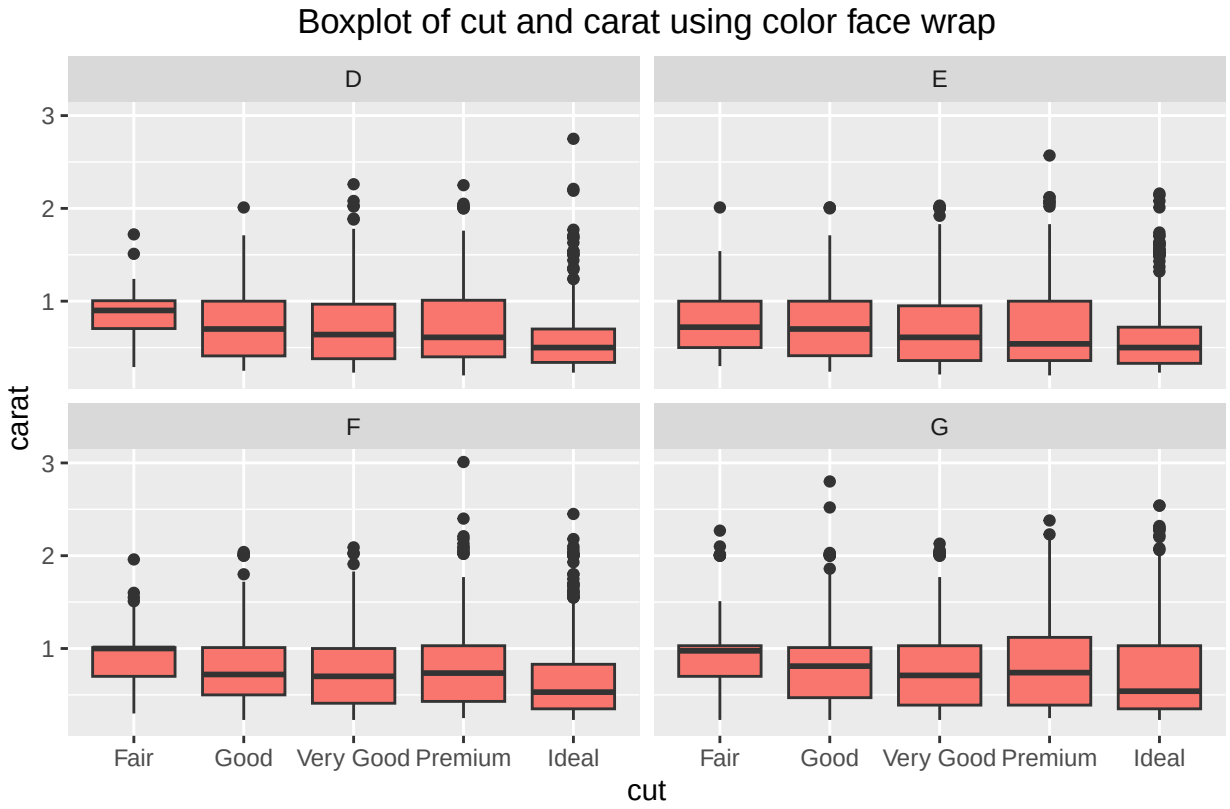


source: diamonds data.frame from ggplot2

Third Plot

boxplot of cut in each color using facet wrap

```
## filter only top 4 diamond color (D, E, F, G)
color_filter <- diamonds %>%
                filter(color == c("D", "E", "F", "G"))

## plot
ggplot(color_filter, aes(cut, carat, fill = "salmon")) + geom_boxplot() +
       facet_wrap(~ color, ncol = 2) +
       labs(title = "Boxplot of cut and carat using color face wrap",
            caption = "source: diamonds data.frame from ggplot2") +
       ggeasy::easy_center_title() +
       theme(legend.position="none")
```



Library nycflight13 and flights df

```
library(nycflights13)

#data.frame flights
data("flights")
data("airlines")

## check head and column name of flights
str(flights)
```

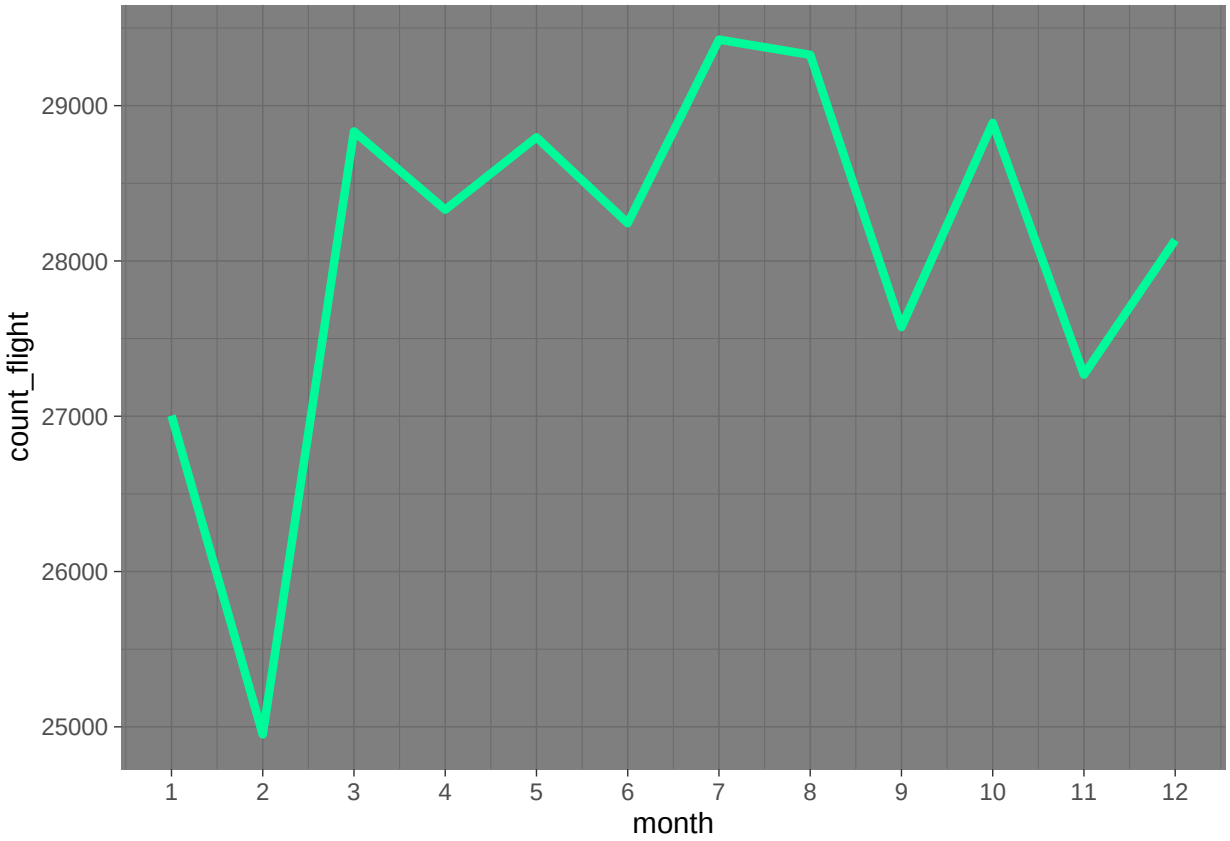
```
## tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
```

```
##  $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

Fourth Plot

Trend lines of average flights in each month in dplyr pipeline

```
flights %>%
  group_by(month) %>%
  summarize(n = n()) %>%
  rename(count_flight = n) %>%
  head(12) %>%
  ggplot(aes(month, count_flight)) +
  geom_line(linewidth = 1.5, col = "mediumspringgreen") +
  scale_x_continuous(limits = c(1, 12), breaks = seq(1, 12, by = 1)) +
  theme_dark()
```



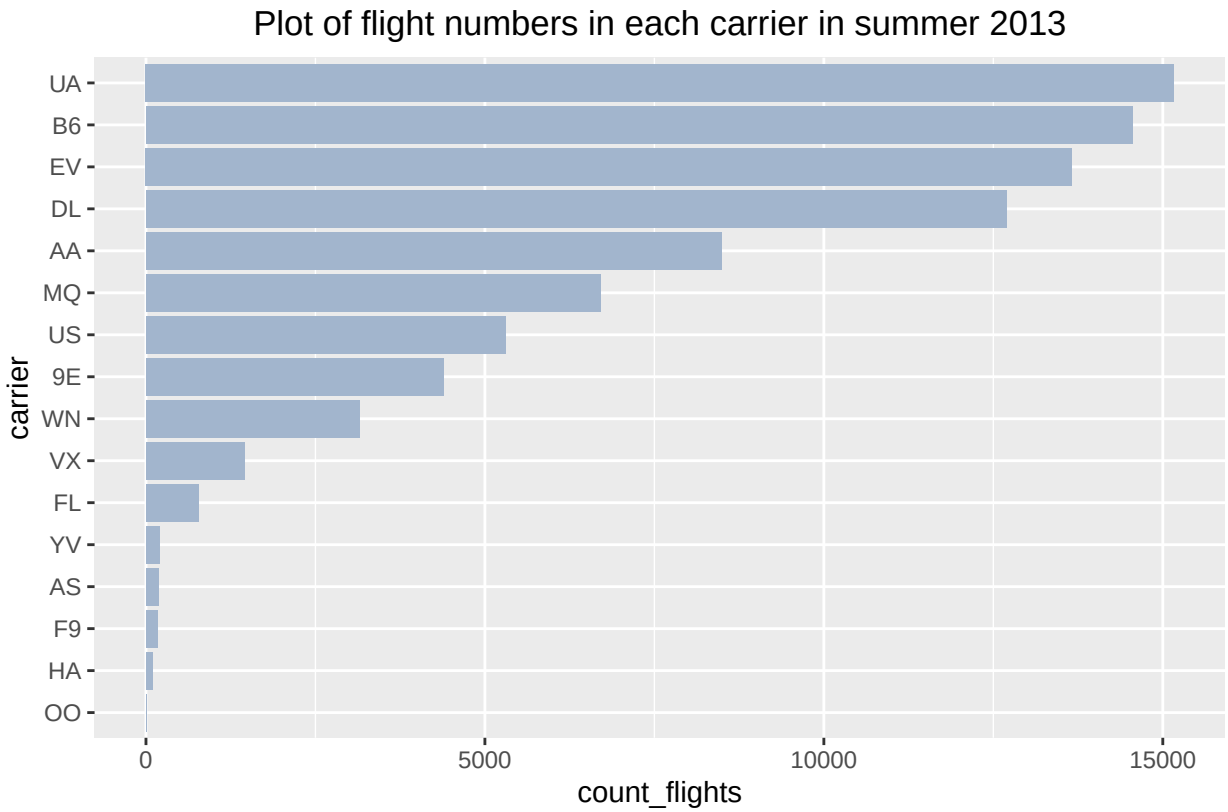
Fifth Plot

Flights in summer 2013 (June to August, 6-8)

```r
air <- airlines
air <- air %>%
  rename(short_name = carrier)

summer_flights <- flights %>%
  filter(year == 2013, between(month, 6, 8)) %>%
  count(carrier) %>%
  arrange(n) %>%
  rename(count_flights = n)

summer_flights
```

```
## # A tibble: 16 x 2
##    carrier count_flights
##    <chr>           <int>
##  1 OO                  6
##  2 HA                 92
##  3 F9                168
##  4 AS                184
##  5 YV                195
##  6 FL                778
##  7 VX               1458
##  8 WN               3151
##  9 9E               4387
## 10 US               5301
## 11 MQ               6702
## 12 AA               8495
## 13 DL              12695
## 14 EV              13660
## 15 B6              14558
## 16 UA              15165
```

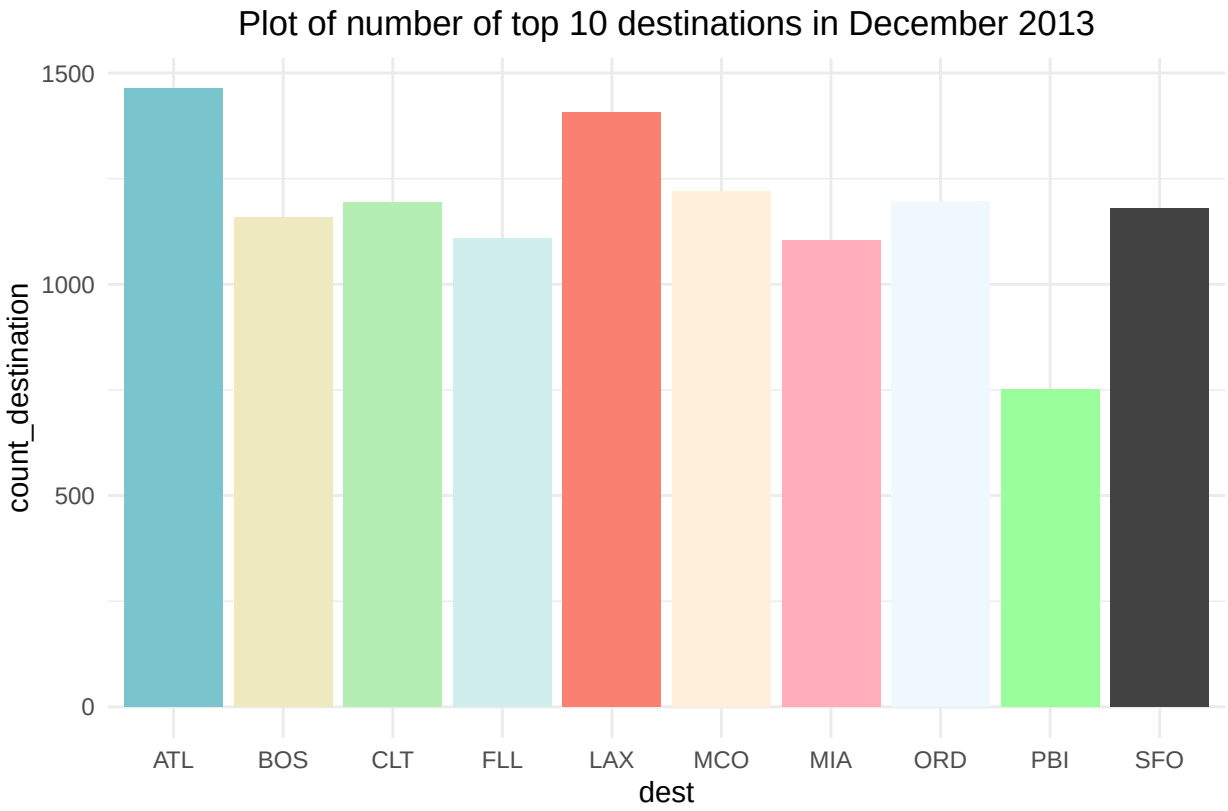
```r
summer_flights$carrier<-factor(summer_flights$carrier, levels = summer_flights$carrier)

ggplot(summer_flights, aes(count_flights, carrier)) +
  geom_col(fill = "lightsteelblue3") + theme_gray() +
  labs(title = "Plot of flight numbers in each carrier in summer 2013",
       caption = "Source: flights data.frame from nycflights13") +
  ggeasy::easy_center_title()
```



Sixth Plot

Top 10 destination in December 2013

```
DEC_des <- flights %>%
  filter(year == 2013, month == 12) %>%
  count(dest) %>%
  arrange(desc(n)) %>%
  rename(count_destination = n)

top10_DEC_des <-DEC_des[1:10,]

## color df for easy input in ggplot
ten_color <- c("cadetblue3", "salmon", "antiquewhite1", "aliceblue",
               "darkseagreen2", "gray26", "lemonchiffon2", "lightcyan2",
               "lightpink1", "palegreen1" )

ggplot(top10_DEC_des, aes(count_destination, dest)) +
  geom_col(fill = ten_color) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Plot of number of top 10 destinations in December 2013",
       caption = "Source: flights data.frame from nycflights13") +
  ggeasy::easy_center_title()
```



Thanks for your attention