Assignment 3

HTHSCI 1M03: Foundations of Data Science

Jasraj_Singh_Johal_johalj11_400434346

Saturday, March 30, 2024

R setup code:

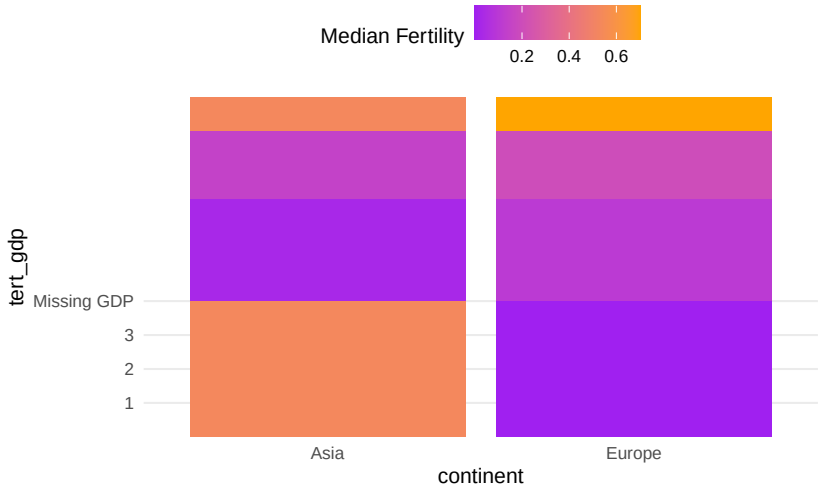
note: i have referred to the book for additional doubts

Question 1

```
data(gapminder, package = "dslabs")
tib_q1 <- gapminder |>
  filter(continent %in% c("Europe", "Asia"), year == 2010) |>
  mutate(tert_gdp = as_factor(ntile(gdp,3)),
         tert_gdp = fct_na_value_to_level(tert_gdp, level = "Missing GDP"),
         continent = fct_drop(continent),
         fr_1m = (fertility / population) * 1e6) |>
  group_by(continent, tert_gdp) |>
  summarize(med_fr_1m = median(fr_1m, na.rm = TRUE)) |>
  ungroup() |>
  complete(continent, tert_gdp) |>
  mutate(med_fr_1m = coalesce(med_fr_1m, 1e-3))
#> `summarise()` has grouped output by 'continent'. You can override using the
#> `.groups` argument.


tib_q1 |>
  ggplot(aes(x = continent, y = tert_gdp, fill = med_fr_1m)) +
  geom_bar(stat = "identity") +  # Using a bar plot for unrelated variables
  scale_fill_gradient(low = "purple", high = "orange") +  # Unintuitive color scheme
  labs(fill = "Median Fertility") +  # Poor labeling
  theme_minimal() +  # Lack of context or explanation
```

```r
theme(legend.position = "top")  # Poor legend placement
```



1. **Misleading Axis Labels**:
   - The x-axis label "continent" and y-axis label "tert_gdp" do not accurately describe the data. "tert_gdp" is converted to a factor without clear representation, and "continent" simply indicates the plotted regions.
2. **Misleading Y-Axis Representation**:
   - Converting "tert_gdp" to a factor suggests it's a categorical variable, leading to confusion. The y-axis should display meaningful numerical values, not arbitrary factor levels.
3. **Misleading Color Coordination**:
   - The fill colors represent the Median Fertility rate, but this isn't clearly indicated in the legend or labels. Using a gradient scale (from red to blue) may imply a continuous scale, which doesn't fit the discrete nature of the data.
4. **Lack of Clarity**:
   - The graph lacks clear context or explanation, making it difficult for viewers to understand what the bars represent and their relationships.
5. **Inconsistent Faceting**:
   - The facet_wrap function separates Europe and Asia into different panels without clear justification. This complicates data interpretation without providing additional insight.
6. **Inadequate Labeling**:

- The fill legend is labeled simply as "Median Fertility," without specifying units. It should indicate that it represents the Median Fertility Rate per 1M people for clarity.

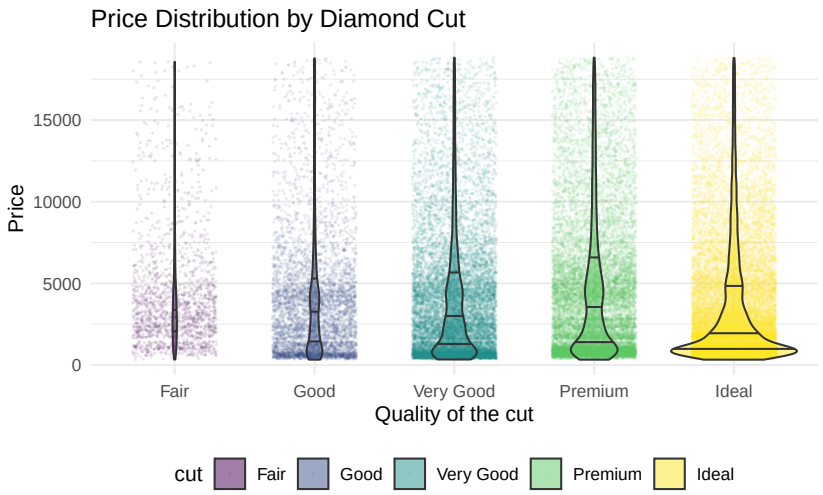
Question 2

```
diamonds |> summary()
#>      carat               cut        color        clarity          depth      
#>  Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00  
#>  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00  
#>  Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194   Median :61.80  
#>  Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75  
#>  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50  
#>  Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00  
#>                                     J: 2808   (Other): 2531                  
#>      table           price             x                y         
#>  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000  
#>  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720  
#>  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710  
#>  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735  
#>  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540  
#>  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900  
#>                                                                   
#>        z         
#>  Min.   : 0.000  
#>  1st Qu.: 2.910  
#>  Median : 3.530  
#>  Mean   : 3.539  
#>  3rd Qu.: 4.040  
#>  Max.   :31.800  
#>
```

```
data(diamonds)

diamonds %>%
  ggplot(aes(x = cut, y = price)) +
  geom_jitter(aes(color = cut), width = 0.3, alpha = 0.1, size = 0.1, height = 0.2) +
  geom_violin(aes(fill = cut), alpha = 0.5, draw_quantiles = c(0.25, 0.5, 0.75), scale="co
  theme_minimal() +
  labs(title = "Price Distribution by Diamond Cut",
```

```
      x = "Quality of the cut",
      y = "Price") +
theme(legend.position = "bottom")
```



1. **Color for Clarity**: Adding color to the jittered points with `aes(color = cut)` helps differentiate between different diamond cut categories. Also made it transparent (`alpha = 0.1`) to avoid overplotting and improve visibility.

2. **Vertical Orientation**: Changing the plot's orientation from horizontal to vertical (setting `aes(x = cut, y = price)`) aligns diamond cut categories along the x-axis and prices along the y-axis. This common orientation improves readability and understanding.

3. **Aesthetic Enhancement**: Adjusting transparency (`alpha = 0.1` for jittered points and `alpha = 0.5` for violins) improves the plot's appearance. Lower transparency helps viewers see individual data points and violin plots more clearly.

4. **Title Addition**: Including a descriptive title ("Price Variation Across Diamond Cuts") provides context and guides interpretation. A clear title helps viewers understand the plot's purpose.

5. **Quantiles for Violin Plots**: Specifying `draw_quantiles = c(0.25, 0.5, 0.75)` indicates which quantiles to display on violin plots. This offers additional insights into price distribution within each diamond cut category.

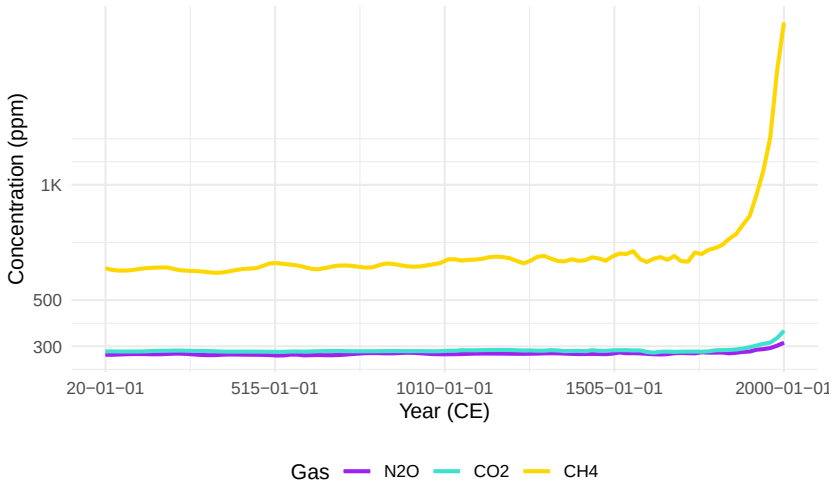
6. **Removal of Boxplots**: As violins already represent data distribution, removing boxplots reduces redundancy and clutter. This streamlines the plot while still conveying valuable information through violins and jittered points. I consider this a better approach for visualizing the data as violin is a modern boxplot.

7. **Legend Position**: Shifting the legend to the bottom (`theme(legend.position = "bottom")`) prevents overlap with the plot's main content. Placing the legend at the bottom is a common practice for horizontal plots and improves overall aesthetics.

Overall, better clarity, aesthetics, and information, making it easier to analyze!

Question 3

```
data(greenhouse_gases)
greenhouse_gases |>
  mutate(gas = factor(gas, levels = c("N2O", "CO2", "CH4"))) |>
  ggplot(aes(x = year, y = concentration, color = gas)) +
  geom_line() +
  geom_line(size = 1) +
  scale_y_continuous(breaks = c(300, 500, 1000), labels = c(300, 500, "1K")) +
  scale_x_continuous(breaks = c(20,515,1010,1505,2000), labels = c("20-01-01", "515-01-01"
  scale_color_manual(values = c("purple", "turquoise", "gold"),
                     labels = c("N2O", "CO2", "CH4")) +
  labs(x = "Year (CE)", y = "Concentration (ppm)", color = "Gas") +
  theme_minimal() +
  theme(legend.position = "bottom")
#> Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
#> i Please use `linewidth` instead.
```



```r
data(greenhouse_gases)

greenhouse_gases |> summary()
#>       year          gas             concentration
#>  Min.   :  20   Length:300         Min.   : 260.0
#>  1st Qu.: 515   Class :character   1st Qu.: 269.7
#>  Median :1010   Mode  :character   Median : 279.7
#>  Mean   :1010                      Mean   : 416.2
#>  3rd Qu.:1505                      3rd Qu.: 641.0
#>  Max.   :2000                      Max.   :1703.4
```

```r
greenhouse_gases |>
  mutate(gas = factor(gas, levels = c("N2O", "CO2", "CH4"))) |>
  ggplot(aes(x = year, y = gas, fill = concentration)) +
  geom_bar(stat = "identity") +
  geom_bar(stat = "identity", color = "white",
           fill = "transparent", width = 0.9) +
  scale_fill_gradientn(
    colors = c("#dbf4e3","#78D5B5","#3A538F","#35284F","black"),
                       limits = c(100, 1703.4), oob = scales::squish) +
  scale_y_discrete(limits = c("N2O", "CO2", "CH4")) +
  scale_x_continuous(breaks = c(20,90000), labels = c(1800,2000)) +
  labs(x = "Year", y = "Gas",
       fill = "Concentration (ppm)") +
  theme_minimal() +
  theme(legend.position = "right")
```