Data Wrangling in R

Fall 2022, MATH8050: Homework 2
Your Name, Section XXX

Due September 14, 12:00 PM

General instructions for homeworks: Please follow the uploading file instructions according to the syllabus. You will give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used. Your code must be completely reproducible and must compile.

Note: Exact colors in your homework solutions may be different from what's in the homework since colors may be displayed differently across different operating platforms and the version of R softwares, however, the pattern of colors should be the same. For example, different colors are used for different groups. The font size or style on labels are allowed to be different as long as it is clearly shown in your figures.

Advice: Start early on the homeworks and it is advised that you not wait until the last day. While the professor and the TA's check emails, they will be answered in the order they are received and last minute help will not be given.

Commenting code Code should be commented. See the tidyverse style guide for questions regarding commenting or how to write code https://style.tidyverse.org/index.html. No late homework's will be accepted.

R Working Environment

Please load all the packages used in the following R chunk before the function `sessionInfo()`

```
# load packages
library(tidyverse)
library(lubridate)
library(patchwork)
library(sf)
library(scico)

sessionInfo()
## R version 4.1.3 (2022-03-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.3.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
##  [1] scico_1.3.1     sf_1.0-8        patchwork_1.1.2 lubridate_1.8.0
##  [5] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.8     purrr_0.3.4
##  [9] readr_2.1.2     tidyr_1.2.0     tibble_3.1.6    ggplot2_3.3.5
## [13] tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.9          class_7.3-20        assertthat_0.2.1    digest_0.6.29
##  [5] utf8_1.2.2          R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [9] reprex_2.0.1        evaluate_0.15       e1071_1.7-9         httr_1.4.2
## [13] pillar_1.7.0        rlang_1.0.2         readxl_1.4.0        rstudioapi_0.13
## [17] rmarkdown_2.13      munsell_0.5.0       proxy_0.4-26        broom_0.8.0
## [21] compiler_4.1.3      modelr_0.1.8        xfun_0.30           pkgconfig_2.0.3
## [25] htmltools_0.5.2     tidyselect_1.1.2    fansi_1.0.3         crayon_1.5.1
## [29] tzdb_0.3.0          dbplyr_2.1.1        withr_2.5.0         grid_4.1.3
## [33] jsonlite_1.8.0      gtable_0.3.0        lifecycle_1.0.1     DBI_1.1.2
## [37] magrittr_2.0.3      units_0.8-0         scales_1.2.0        KernSmooth_2.23-20
## [41] cli_3.2.0           stringi_1.7.6       fs_1.5.2            xml2_1.3.3
## [45] ellipsis_0.3.2      generics_0.1.2      vctrs_0.4.1         tools_4.1.3
## [49] glue_1.6.2          hms_1.1.1           fastmap_1.1.0       yaml_2.3.5
## [53] colorspace_2.0-3    classInt_0.4-3      rvest_1.0.2         knitr_1.38
## [57] haven_2.5.0
```

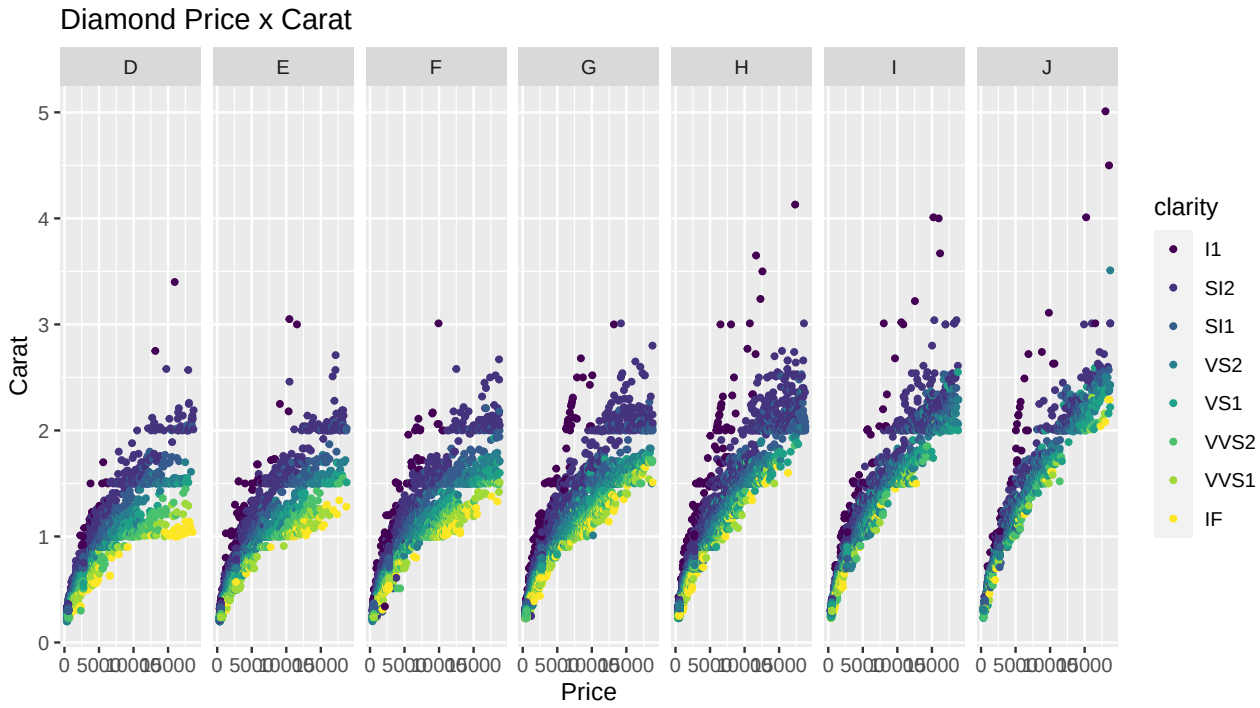
Total points on assignment: 10 (reproducibility) + 20 (Q1) + 35 (Q2) + 10 (Q3) + 25 (Q4)

Reproducibility component: 10 points.

1. (20 pts total, equally weighted) The `diamonds` dataset

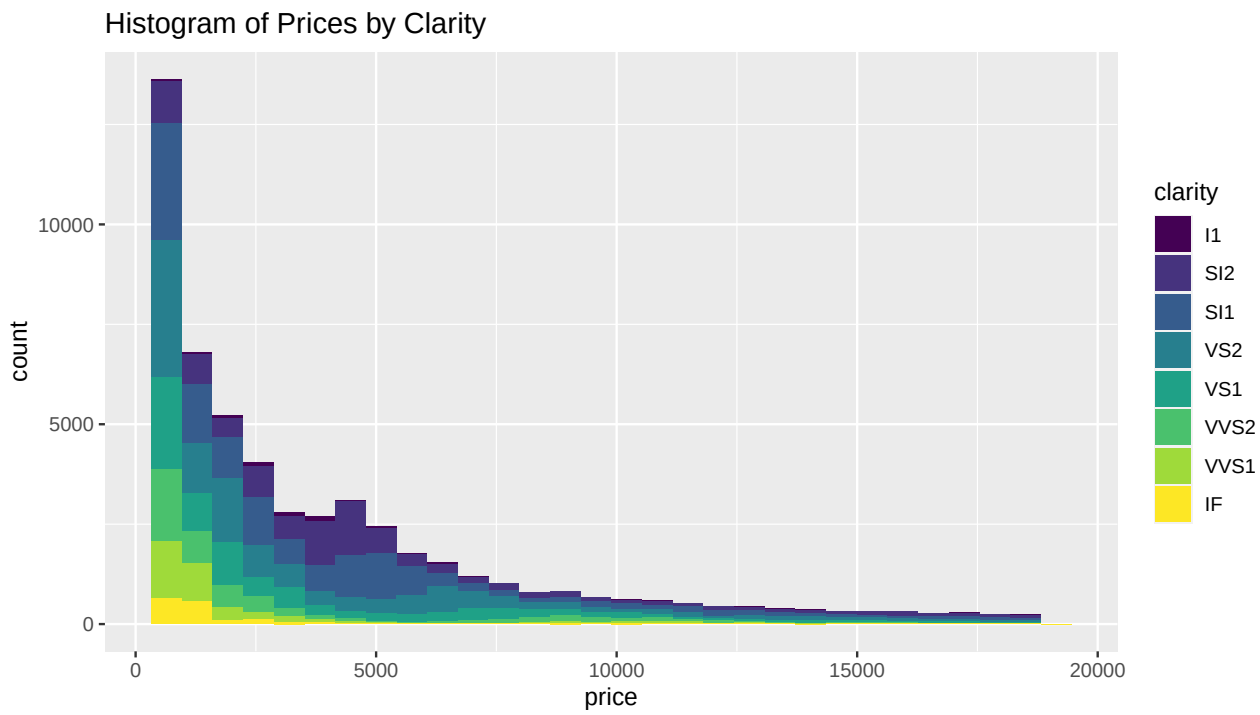
a. Replicate the following scatter plot

```
ggplot(data = diamonds, aes(x=price, y=carat)) +
  geom_point(aes(col = clarity), size = 1) +
  facet_grid(~color) +
  ggtitle("Diamond Price x Carat") +
  xlab("Price") +
  ylab("Carat")
```



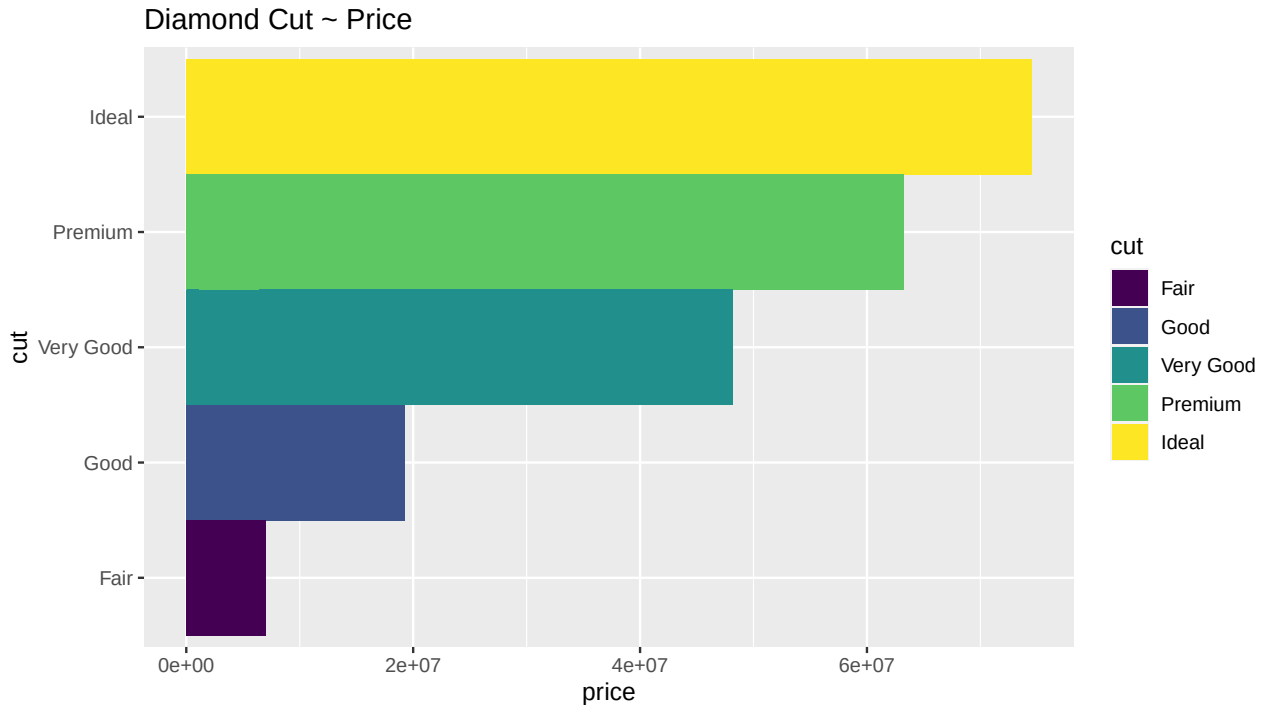
b. Replicate the following plot

```
ggplot(diamonds, aes(x=price, fill=clarity)) +
  geom_histogram() +
  ggtitle("Histogram of Prices by Clarity")
```



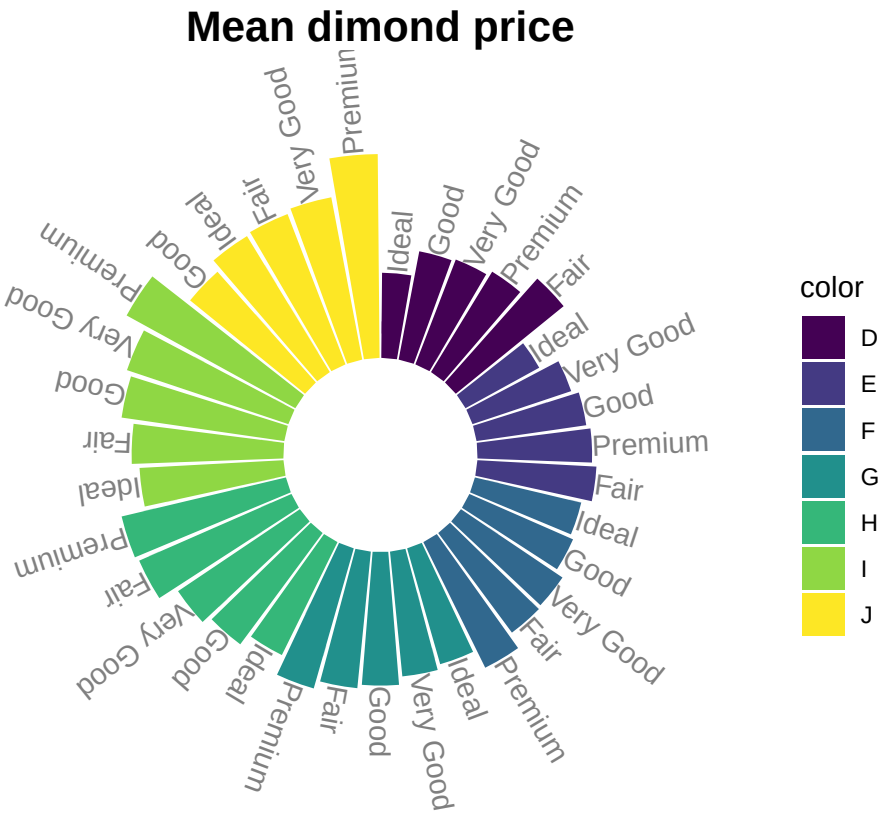
c. Replicate the following plot

```
ggplot(data = diamonds, aes(x=cut, y = price, fill = cut)) +
  geom_bar(width = 1, stat = "identity") +
  ggtitle("Diamond Cut ~ Price") +
  coord_flip()
```



d. For the `diamonds` dataset, replicate the following plot.

```
df <- diamonds %>%
  group_by(cut, color) %>%
  summarise(price = mean(price)) %>%
  arrange(color, price) %>%
  ungroup() %>%
  mutate(id = row_number(),
         angle = 90 - 360 * (id - 0.5) / n())

df %>%
  ggplot(aes(factor(id), price, fill = color, group = cut, label = cut)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  geom_text(hjust = 0, angle = df$angle, alpha = .5) +
  coord_polar() +
  ggtitle('Mean dimond price') +
  ylim(-3000, 7000) +
  theme_void() +
  theme(plot.title = element_text(hjust = 0.5, size = 16, face = 'bold'))
```

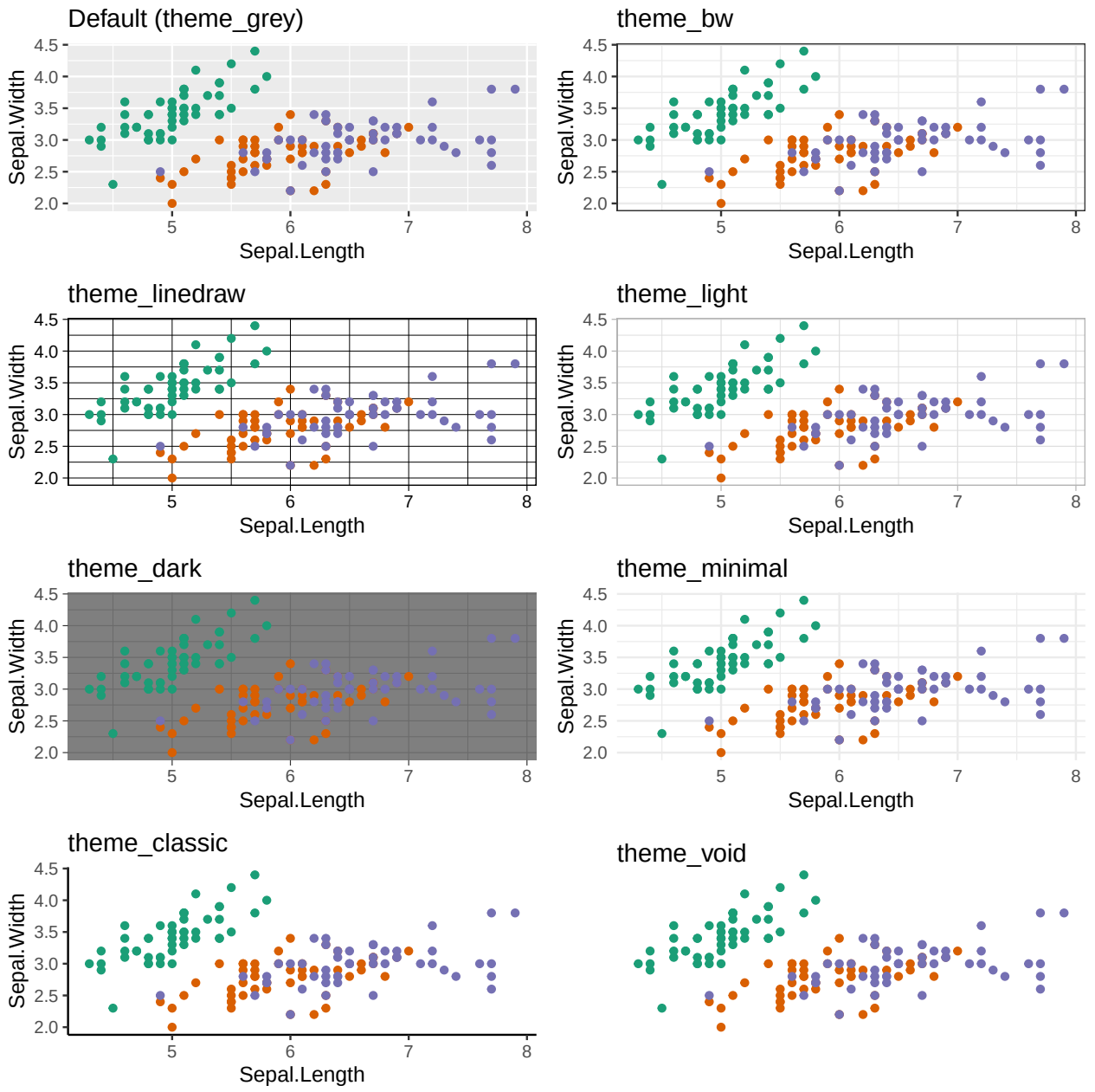


2. (35 pts total, equally weighted) We use `tydiverse` package to generate various plots with the `iris` dataset.

a. For the `iris` dataset, replicate the following plot

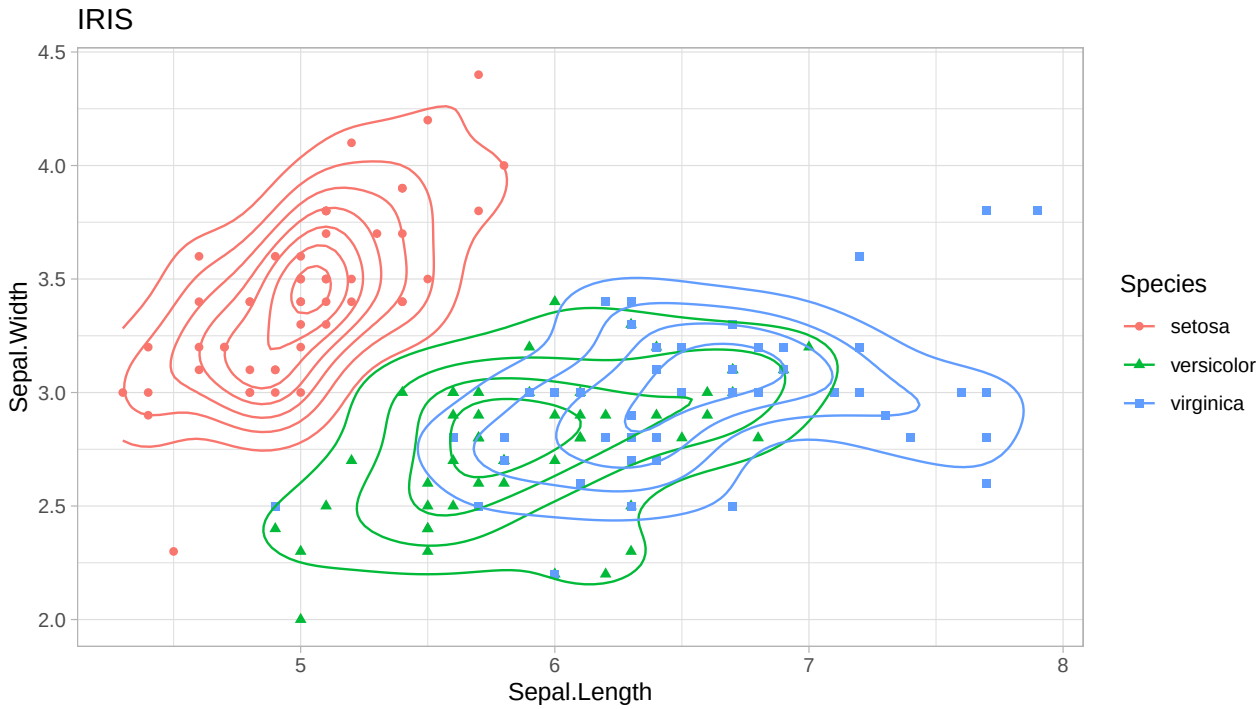
```
base_plot <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(show.legend = FALSE) + scale_color_brewer(palette = "Dark2")

default_plot_title <- base_plot + labs(title = "Default (theme_grey)")

bw_plot_title <- base_plot + labs(title = "theme_bw") + theme_bw()

linedraw_plot_title <- base_plot + labs(title = "theme_linedraw") + theme_linedraw()

light_plot_title <- base_plot + labs(title = "theme_light") + theme_light()

dark_plot_title <- base_plot + labs(title = "theme_dark") + theme_dark()

minimal_plot_title <- base_plot + labs(title = "theme_minimal") + theme_minimal()

classic_plot_title <- base_plot + labs(title = "theme_classic") + theme_classic()

void_plot_title <- base_plot + labs(title = "theme_void") + theme_void()

wrap_plots(default_plot_title, bw_plot_title, linedraw_plot_title,
          light_plot_title, dark_plot_title, minimal_plot_title,
          classic_plot_title, void_plot_title,
```

```
        ncol = 2, byrow = TRUE)
```



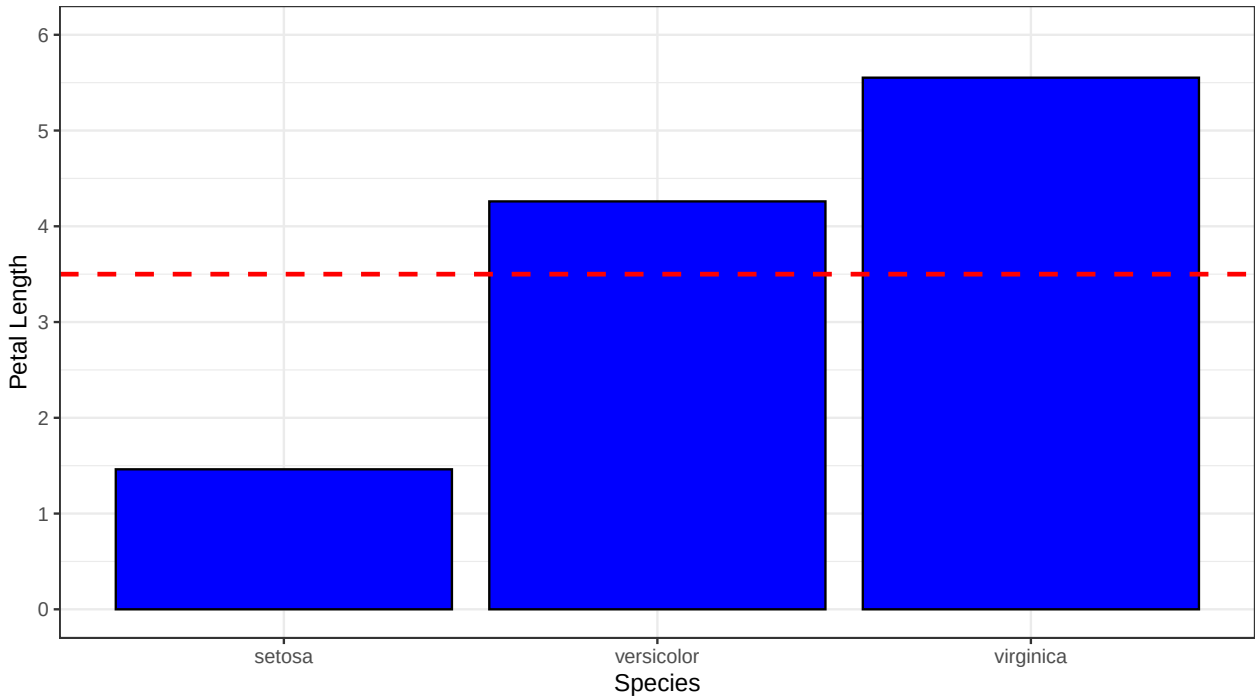
b. For the `irish` dataset, replicate the following plot.

```
iris %>%
  ggplot(aes(Sepal.Length, Sepal.Width, color = Species, shape = Species)) +
  geom_point() +
  geom_density2d() +
  ggtitle('IRIS') +
  theme_light()
```



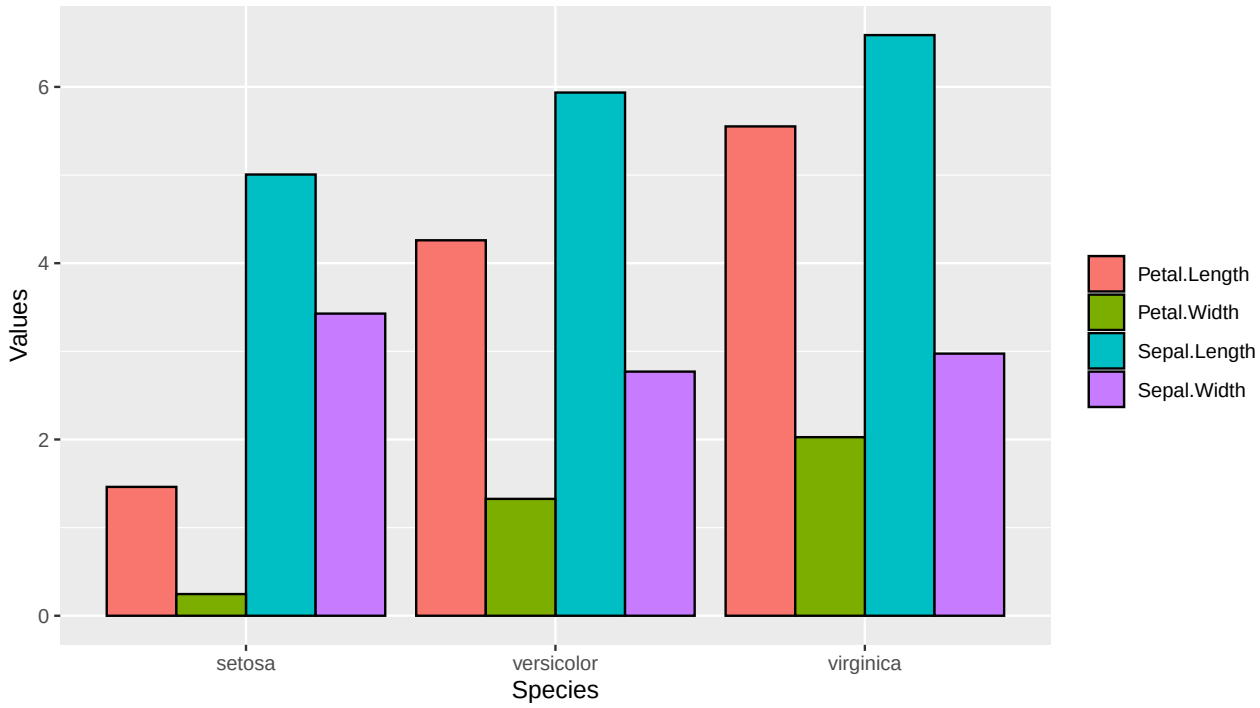
c. Compute the mean Petal Length under each species and then replicate the following plot. Make sure that you only use the `tidyverse` package for this problem.

```
df = as_tibble(iris)

df1 = df %>%
  group_by(Species) %>%
  summarize(
    across(
      .cols = c(Sepal.Length:Petal.Width),
      .fns = mean,
    )
  )

df1 %>%
  ggplot(aes(x=Species, y=Petal.Length)) +
  geom_bar(stat="identity", fill="blue", color="black") +
  xlab("Species") + ylab("Petal Length") + theme_bw() +
  scale_y_continuous(breaks=seq(0, 6, by=1), limits=c(0,6)) +
  geom_hline(yintercept = 3.5, linetype="dashed",
             color="red", size=1)
```



d. Combine variables by species and then replicate the following plot. Make sure that you only use the `tidyverse` package for this problem.

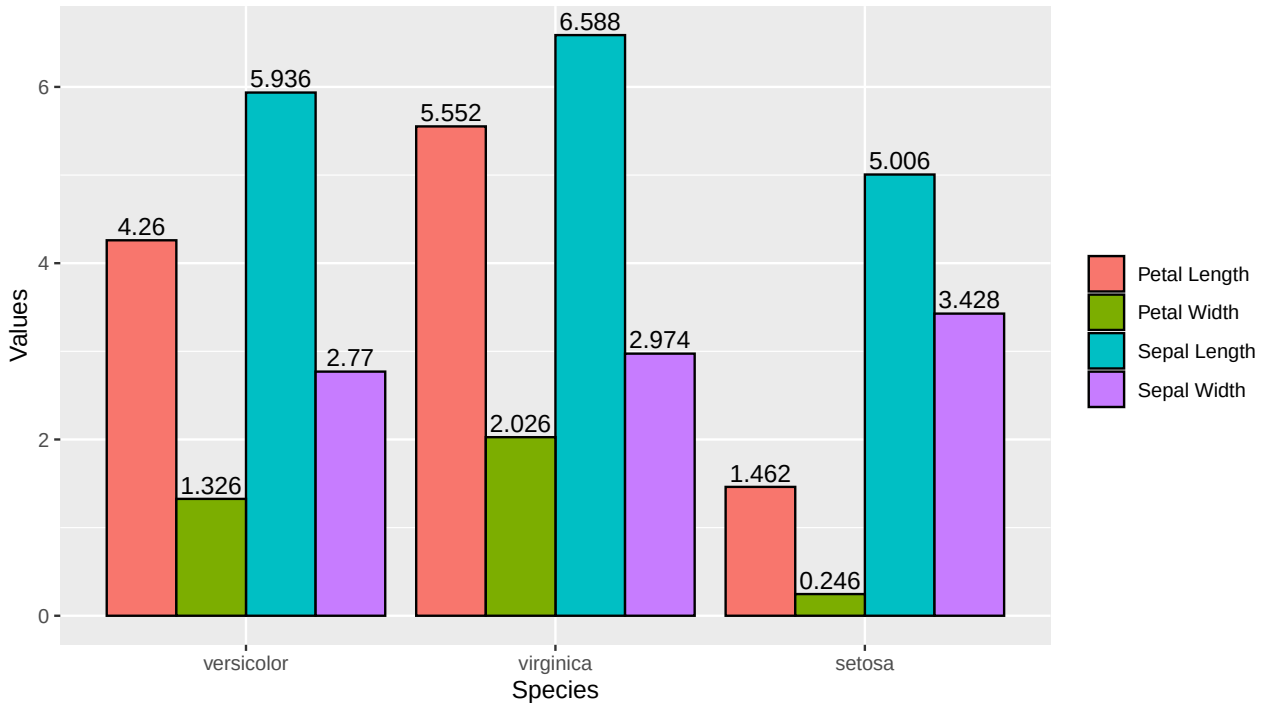
```
df1 %>%
  pivot_longer(
    cols=Sepal.Length:Petal.Width,
    names_to="variable",
    values_to = "value"
  ) %>%
  ggplot(aes(x=Species, y=value, group=variable, fill=variable)) +
  labs(x="Species", y="Values") +
  geom_bar(stat="identity", color="black", position="dodge") +
  scale_fill_discrete(name=NULL)
```



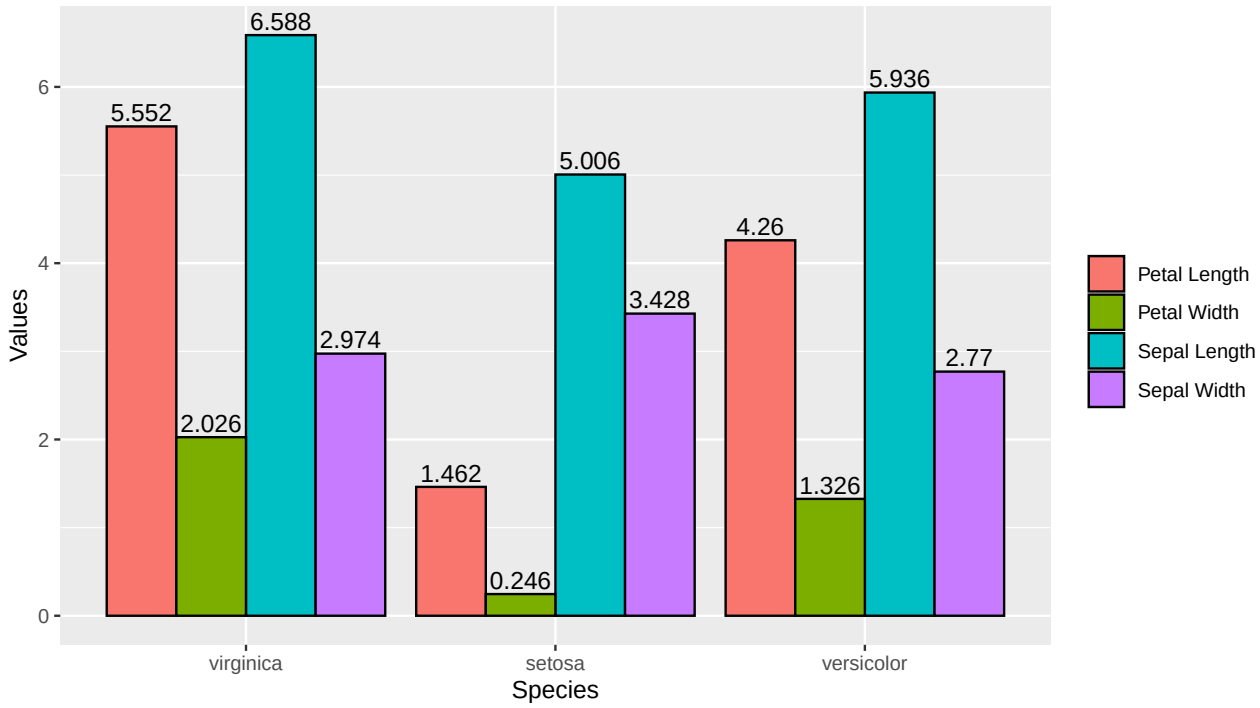
e. Order the species according to the order `virginica`, `setosa`, and `versicolor`, and replicate the following plot. Make sure that you only use the `tidyverse` package for this problem.

Both solutions are accepted below.

```
df1$Species = df1$Species %>%
  fct_reorder(.x=c("virginica", "setosa", "versicolor"))

df1 %>%
  pivot_longer(
    cols=Sepal.Length:Petal.Width,
    names_to="variable",
    values_to = "value"
  ) %>%
  ggplot(aes(x=Species, y=value, group=variable, fill=variable)) +
  labs(x="Species", y="Values") +
  scale_fill_discrete(name=NULL,
                      labels=c("Petal Length", "Petal Width",
                               "Sepal Length", "Sepal Width")) +
  geom_bar(stat="identity", color="black", position="dodge") +
  geom_text(aes(label=value, vjust=-0.3),
                position = position_dodge(0.9))
```

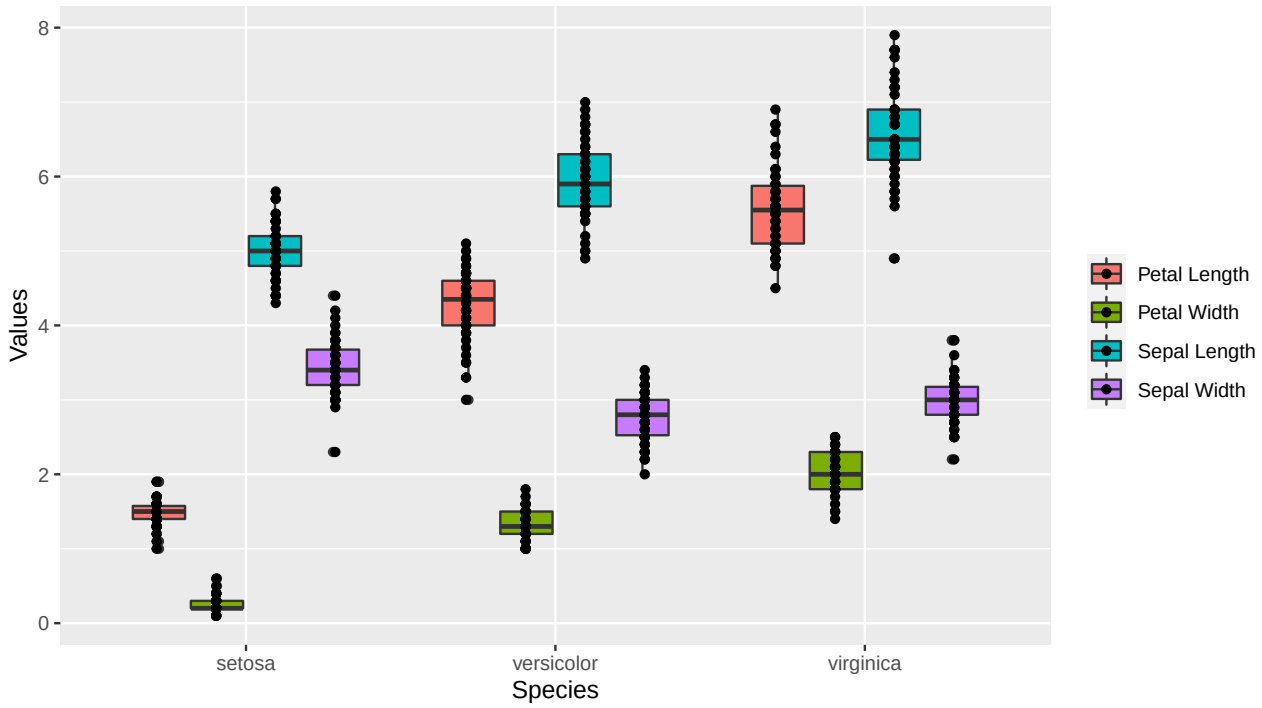



```
df1$Species = df1$Species %>%
  fct_reorder(.x=c("virginica", "setosa", "versicolor"))
df1$Species = ordered(df1$Species,
                      levels=c("virginica",
                               "setosa",
                               "versicolor")
                      )

df1 %>%
  pivot_longer(
    cols=Sepal.Length:Petal.Width,
    names_to="variable",
    values_to = "value"
  ) %>%
  ggplot(aes(x=Species, y=value, group=variable, fill=variable)) +
  labs(x="Species", y="Values") +
  scale_fill_discrete(name=NULL,
                      labels=c("Petal Length", "Petal Width",
                               "Sepal Length", "Sepal Width")) +
  geom_bar(stat="identity", color="black", position="dodge") +
  geom_text(aes(label=value, vjust=-0.3),
                position = position_dodge(0.9))
```



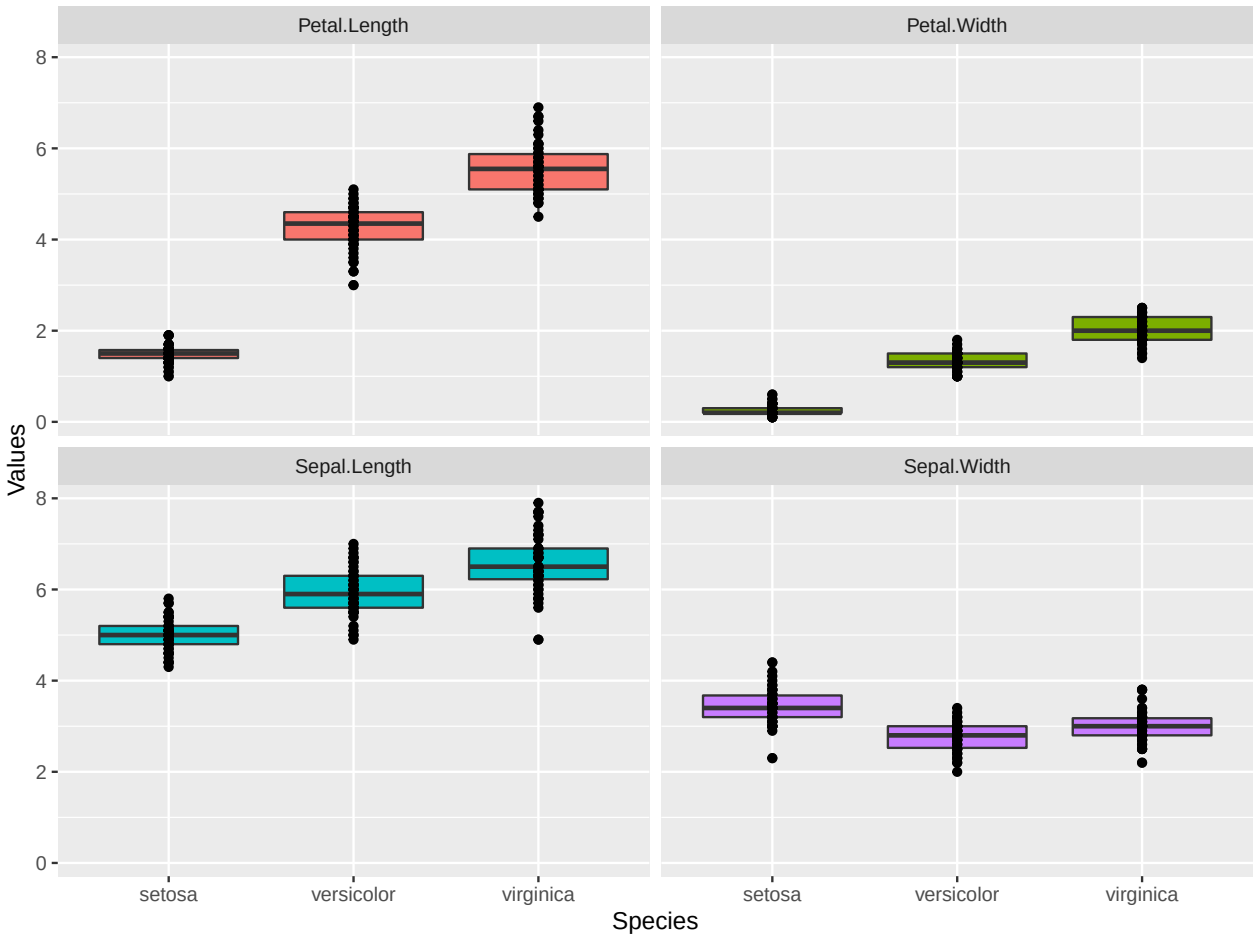
f. Add small amount of random variation to the location of each point using `geom_jitter` and replicate the following boxplot, where each characteristics of species corresponds to a boxplot and these boxplots are grouped by species.

```
df %>%
  pivot_longer(
    cols=Sepal.Length:Petal.Width,
    names_to="variable",
    values_to = "value"
  ) %>%
  ggplot(aes(x=Species, y=value, fill=variable)) +
  labs(x="Species", y="Values") +
  geom_boxplot() +
  geom_jitter(position=position_dodge(0.77)) +
  scale_fill_discrete(name=NULL,
                      labels=c("Petal Length", "Petal Width",
                               "Sepal Length", "Sepal Width"))
```



g. Generate the boxplots faceted for each species and replicate the following plot.

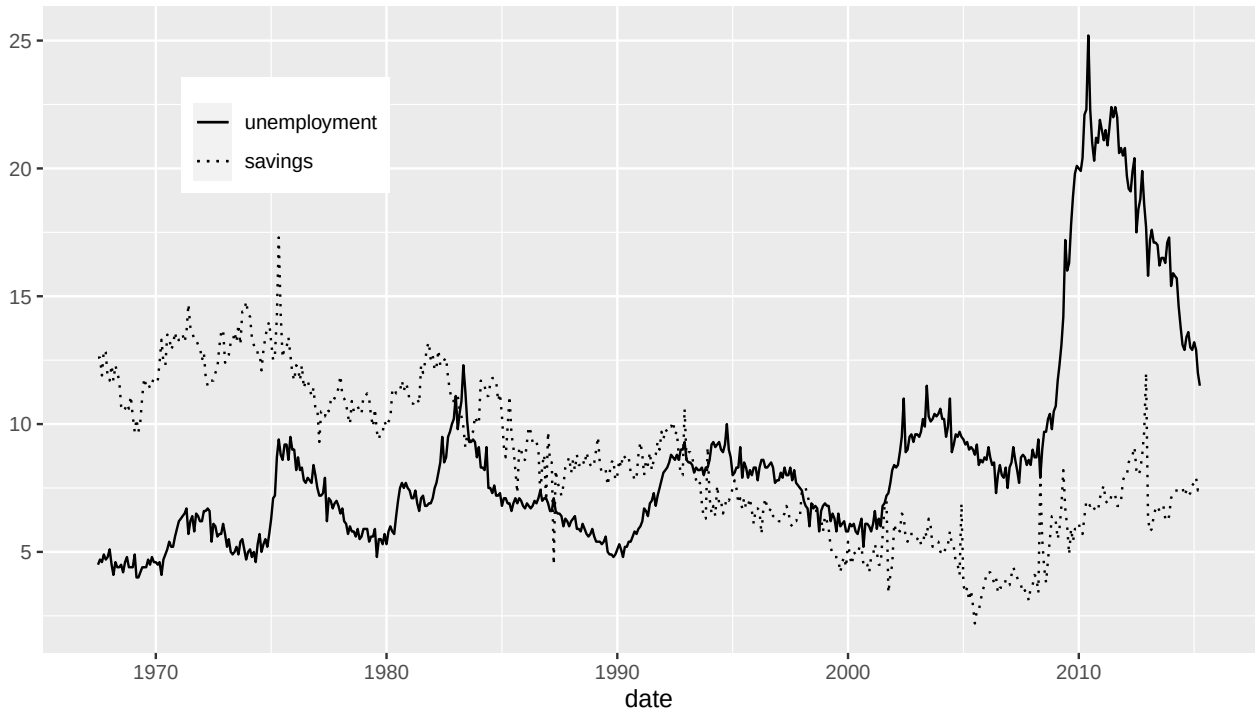
```
df %>%
  pivot_longer(
    cols=Sepal.Length:Petal.Width,
    names_to="variable",
    values_to = "value"
  ) %>%
  ggplot(aes(x=Species, y=value, fill=variable)) +
  labs(x="Species", y="Values") +
  geom_boxplot() +
  facet_wrap(~variable, ncol=2) +
  geom_jitter(position=position_dodge(0.77)) +
  theme(legend.position = "none")
```



3. (10 pts total, equally weighted) Use the `economics` dataset from the ggplot2 package answer the following questions

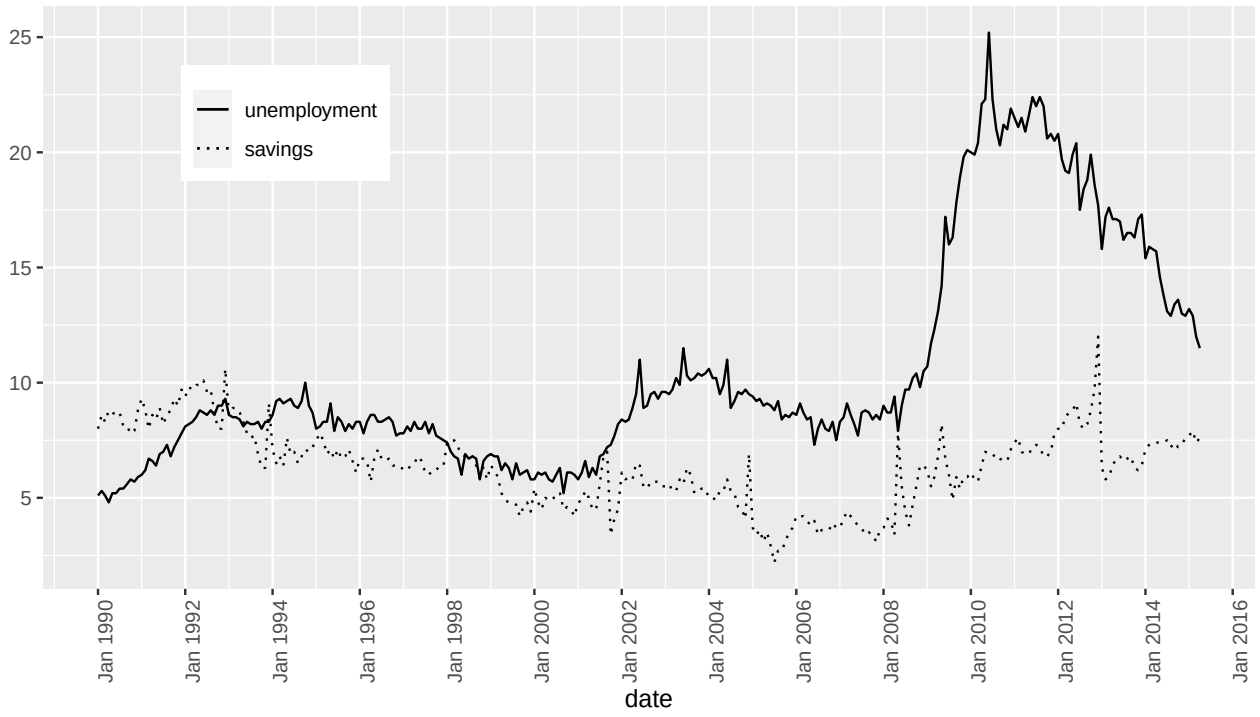
a. Replicate the following figure mentioned in Lecture 2 for the ggplot2 package

```
ggplot(economics) +
    geom_line(aes(x=date, y=uempmed,
                  linetype="unemployment")) +
    geom_line(aes(x=date, y=psavert,
                  linetype="savings")) +
    scale_linetype_manual(name=NULL,
      values=c(
      "unemployment"="solid",
      "savings"="dotted"
    )) +
    theme(legend.position = c(0.2, 0.8)) +
    ylab("")
```



b. Replicate the following figure, where the date starts from the year 1990.

```
economics %>%
  filter(year(date)>=1990) %>%
  ggplot() +
    geom_line(aes(x=date, y=uempmed,
                  linetype="unemployment")) +
    geom_line(aes(x=date, y=psavert,
                  linetype="savings")) +
    scale_linetype_manual(name=NULL,
      values=c(
      "unemployment"="solid",
      "savings"="dotted"
    )) +
    theme(legend.position = c(0.2, 0.8),
          axis.text.x = element_text(angle = 90)
          ) +
    ylab("") +
    scale_x_date(date_breaks="2 year",
                 date_labels="%b %Y"
                 )
```



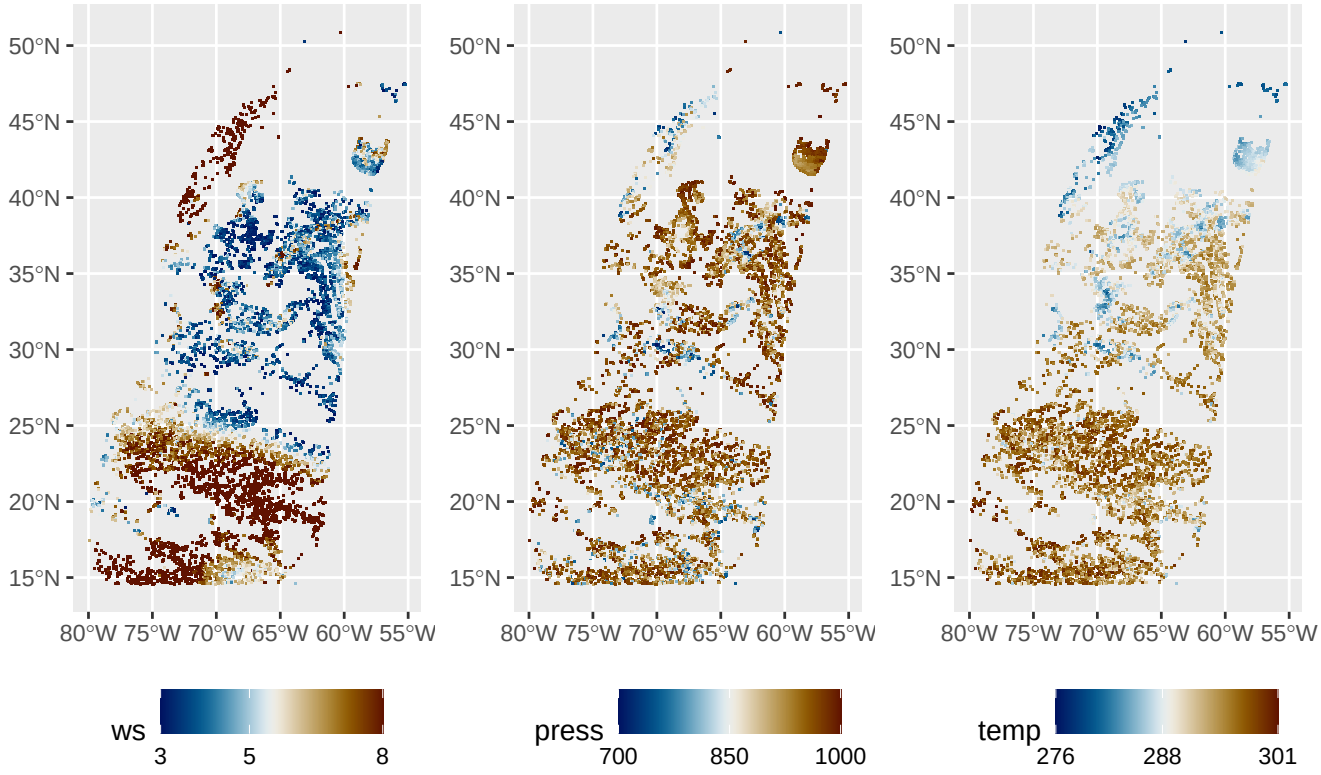
4. (25 pts total) Work with the GOES-R dataset mentioned in class

a. (4pts) load the `DMWC_G16.nc` dataset in R, extract variables: `wind_speed`, `wind_direction`, `lat`, `lon`, `time`, `pressure`, `temperature`, `local_zenith_angle`, `solar_zenith_angle`, `DQF`, save it into a data frame as shown below

```
dat = data.frame(ws=wind_speed, wd=wind_direction,
                 lat=lat, lon=lon, time=time,
                 press=pressure, temp=temperature,
                 lza=local_zenith_angle,
                 sza=solar_zenith_angle,
                 DQF=DQF)
```

b. (8pts) Convert the data frame `dat` into an `sf` object named `df`, where only observations with `DQF` equal to 0 are kept as in Lecture 3, and then replicate the following figure with the following requirements:

- using the **filled square** shape with size .1
- using the `scico::vik` color palette
- using the `wrap_plot()` function or the pip operator "+" to arrange the columns



c. (5pts) In the `df` data frame, pivot the variables `ws`, `press`, `temp` into longer format and give it a new name `variable` with their values stored in the new variable `value`. Then save this new dataset into a tibble `p` and print out the first 6 observations in this new data frame. You should obtain the following output

```
## Simple feature collection with 6 features and 7 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -63.0746 ymin: 50.24621 xmax: -60.31003 ymax: 50.88714
## Geodetic CRS:  WGS 84
## # A tibble: 6 x 8
##      wd       time   lza   sza   DQF            geometry variable  value
##   <dbl>      <dbl> <dbl> <dbl> <int>          <POINT [°]> <chr>     <dbl>
## 1  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) ws        29.6
## 2  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) press    746.
## 3  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) temp     280.
## 4  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) ws         3.42
## 5  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) press    989.
## 6  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) temp     278.
```

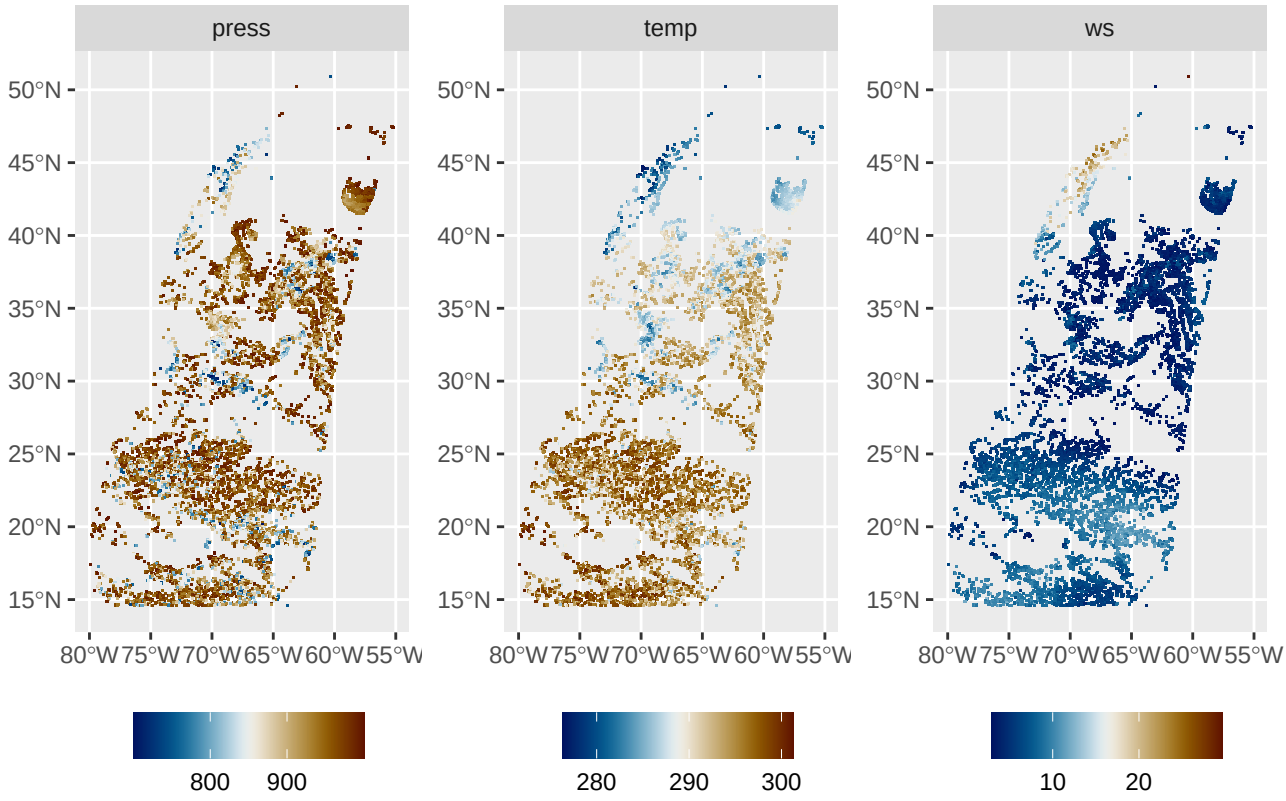
```
p = df %>%
  pivot_longer(cols=c(ws, press, temp),
               names_to="variable",
               values_to="value")
p %>% head(n=6)
## Simple feature collection with 6 features and 7 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -63.0746 ymin: 50.24621 xmax: -60.31003 ymax: 50.88714
```

```
## Geodetic CRS:  WGS 84
## # A tibble: 6 x 8
##      wd       time   lza   sza   DQF             geometry variable  value
##   <dbl>      <dbl> <dbl> <dbl> <int>          <POINT [°]> <chr>     <dbl>
## 1  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) ws        29.6
## 2  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) press    746.
## 3  209. 656121674.  60.0  77.1     0 (-60.31003 50.88714) temp     280.
## 4  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) ws         3.42
## 5  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) press    989.
## 6  263. 656121674.  58.7  78.3     0  (-63.0746 50.24621) temp     278.
```

d. (8pts) Replicate the exact figure with the following requirements:

- using the **filled square** shape with size .1
- using the `scico::vik` color palette

```
ps = split(p, f=p$variable)
gg = lapply(ps, function(x){
  ggplot(x, aes(color=value)) +
    geom_sf(size=.1, shape=15) +
    facet_wrap(~variable) +
    paletteer::scale_color_paletteer_c(name=NULL,
                                       palette="scico::vik",
                                       n.breaks=3,
                                       oob=scales::squish) +
    theme(legend.position="bottom", legend.box = "horizontal")
})

wrap_plots(gg, nrow=1)
```


press
temp
ws
50°N
45°N
40°N
35°N
30°N
25°N
20°N
15°N
80°W
75°W
70°W
65°W
60°W
55°W
800
900
280
290
300
10
20