STAT 3280 Homework 4

Your Name

October 17, 2022

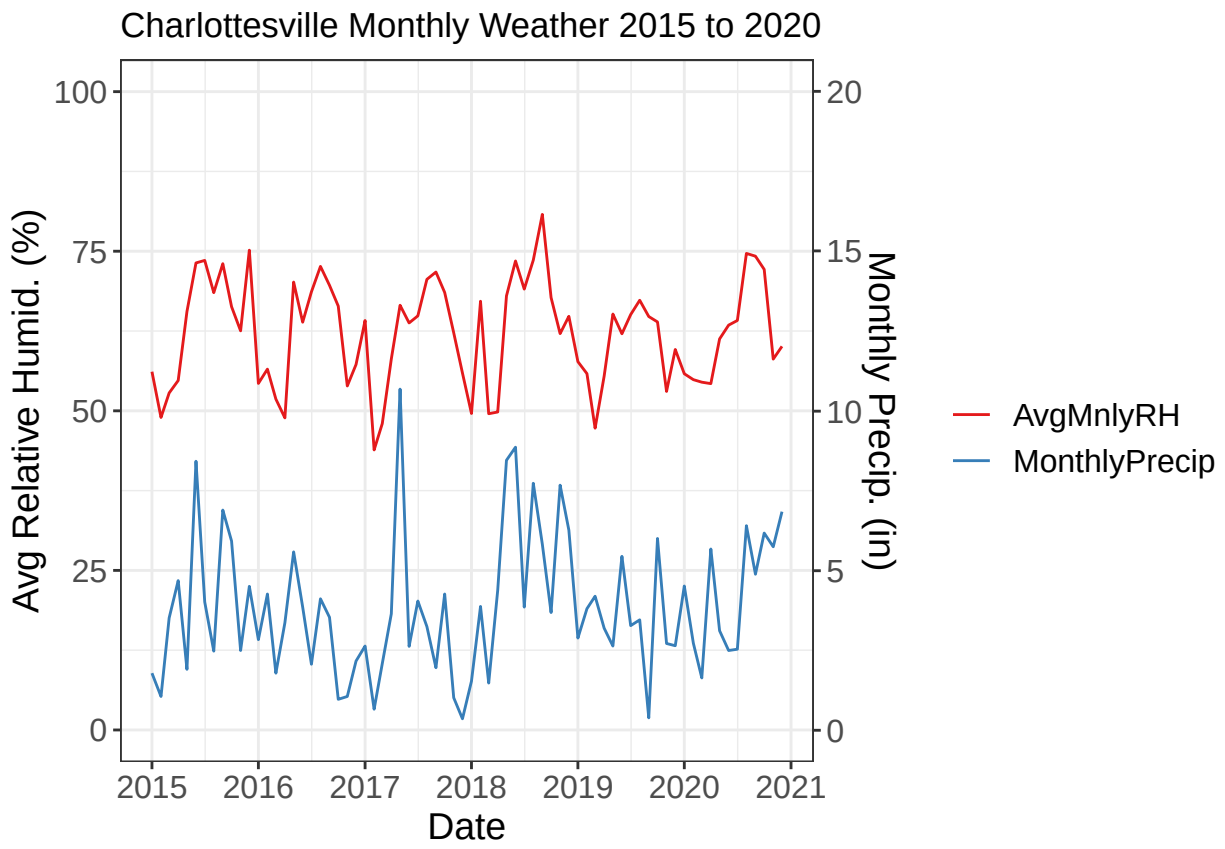
.Rmd file can be found on Collab under Resources/Assigments

Q1: Using the `CVALB_NOAAWeather_Archive` data, create a temporal plot that shows the average relative monthly humidity (%) and the total monthly precipitation for all months in the years 2015 to 2020. Note the formula for relative humidity below in the code section, where `tempC` is the temperature in degrees Celsius, and `dewpC` is the dew point in degrees Celsius. Only use a single plotting area (do not facet). Ensure colors, labels, and themes make the data and message easy to understand.

```
my_theme <- theme_bw() +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14))

mnth <- CvilleWeather %>%
  filter(DATE >= "2015-01-01" & DATE <= "2020-12-31") %>%
  mutate(month = format(DATE, "%Y-%m-01")) %>%
  mutate(dewpC = (DEWP - 32) * 5 / 9,
         tempC = (TEMP - 32) * 5 / 9) %>%
  group_by(month) %>%
  summarise(mprcp = sum(PRCP) * 5,
            mrh = mean(100 * exp(17.625 * dewpC / (243.04 + dewpC)) /
              exp(17.625 * tempC / (243.04 + tempC)))) %>%
  mutate(month = as.Date(month)) %>%
  rename(AvgMnlyRH = mrh, MonthlyPrecip = mprcp) %>%
  pivot_longer(2:3, values_to = "value", names_to = "measurement")

p1 <- ggplot(mnth) +
  geom_line(aes(x = month, y = value, color = measurement)) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(limit = c(0,100), sec.axis = sec_axis(trans = ~. /5,
                                          name = "Monthly Precip. (in)")) +
  scale_color_brewer("", palette = "Set1") +
  labs(x = "Date", y = "Avg Relative Humid. (%)",
       title = "Charlottesville Monthly Weather 2015 to 2020") +
  my_theme
p1
```




```
# RH = 100 * exp(17.625 * dewpC / (243.04 + dewpC)) /
#    exp(17.625 * tempC / (243.04 + tempC)))
```

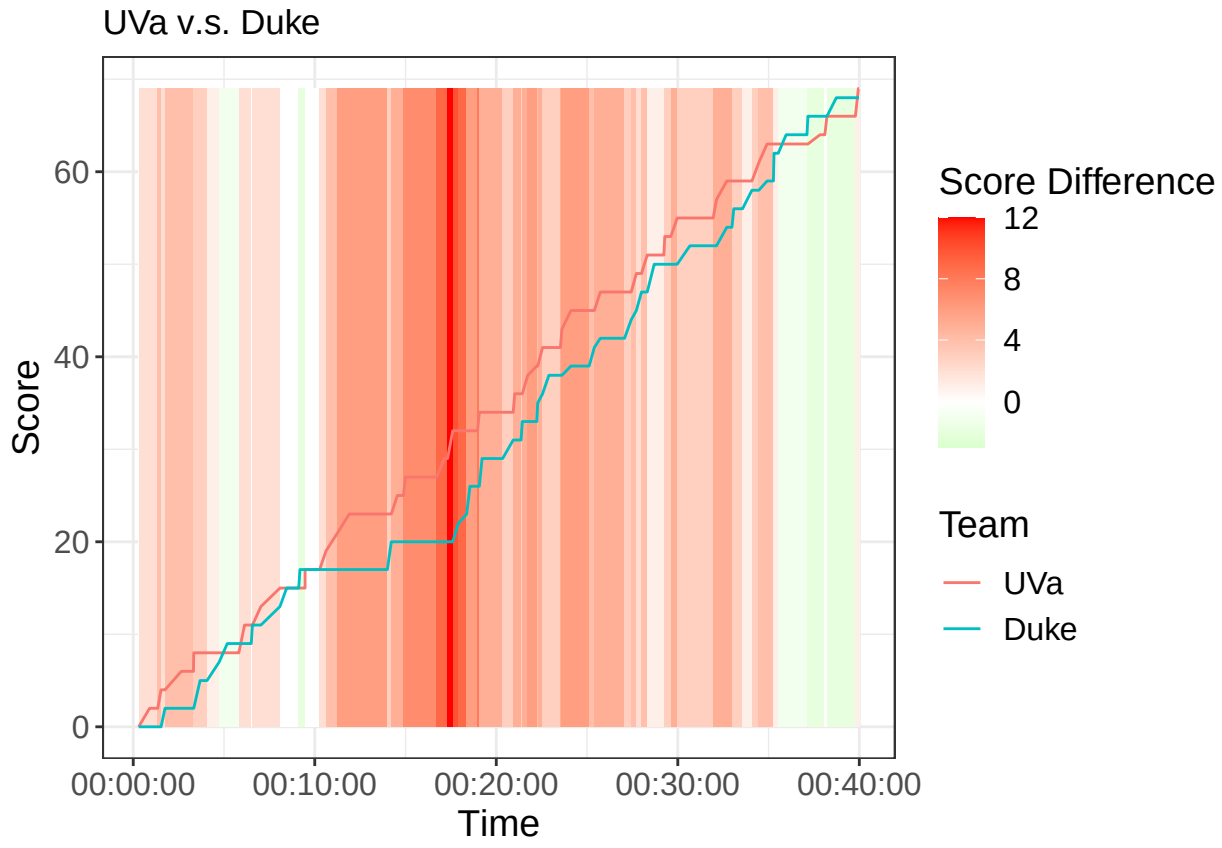
Q2: Using the `UVA_Duke_020722` data, create a temporal plot that shows the score for each team over time in the basketball game. UVA is `away_score` and Duke is `home_score`. Format time as `hh:mm:ss` with `00:00:00` corresponding to the start of the game. Shade the background of the plot with a color that tells the score difference for a corresponding horizontal axis time (hint: use a `geom_rect()` object). Ensure colors, labels, and themes make the data and message easy to understand.

```
UVA_Duke <- UVA_Duke_020722 %>%
  select(away_score, home_score, secs_remaining) %>%
  mutate(score_diff = away_score - home_score,
         secs = 2400 - secs_remaining) %>%
  mutate(secs = as.POSIXct("2022-02-01") +
           as.difftime(secs, unit = "secs")) %>%
  pivot_longer(1:2, names_to = "Team", values_to = "score")

p2 <- ggplot(UVA_Duke) +
  geom_rect(aes(xmin = lag(secs), xmax = secs, ymin = 0, ymax = max(score),
                fill = score_diff)) + scale_fill_gradient2("Score Difference",
                                                           low = "green",
                                                           mid = "white",
                                                           high = "red") +
  geom_line(aes(x = secs, y = score, color = Team)) +
  scale_x_datetime(date_labels = "%T") +
```

```
  scale_color_hue(labels = c("away_score" = "UVa", "home_score" = "Duke")) +
  labs(x = "Time", y = "Score", title = "UVa v.s. Duke") + my_theme
p2
```

```
## Warning: Removed 1 rows containing missing values (geom_rect).
```



Q3: Create a candlestick plot using the `YahooFin` data set for the share price of Google (`GOOG`) in the last three months of 2019 (Q4). A candlestick plot should have a `geom_rect()` element that corresponds to the difference in `open` and `close` prices on a given day. Shade negative returns as red, and positive returns as green. Draw a line from the box to each `high` and `low` observation for a day. You may use the following link as a reference for clarification on plot type https://datavizcatalogue.com/methods/candlestick_chart.html.

```
Goog <- YahooFin %>%
  filter(symbol == "GOOG" &
           date >= "2019-10-01" &
           date <= "2019-12-31")

p2 <- Goog %>%
  ggplot(aes(x = date, y = close)) +
  geom_candlestick(aes(open = open, high = high, low = low,
                       close = close), fill_up = "green", fill_down = "red",
                   colour_up = "black", colour_down = "black") +
  labs(title = "2019 Q4 Google Stock Price (Green = Gain, Red = Loss)", x = "Date",
       y = "Share Price ($)") +
```

```
  scale_x_date("Date", date_breaks = c("2 weeks"), date_labels = "%b %d") +
  theme_bw()
p2
```

Q4: Create a network graph using the `State_to_State_Migration` and the `StateAbbrev` data. For the year 2015, create a directed network that shows the connections where estimated migration exceeded 25,000 people. Label each node with the two letter state abbreviation. Use a directed `join` statement and only retain states that are in the `StateAbbrev` labeling set. Ensure colors, labels, and themes make the data and message easy to understand.

```
data2015 <- Migration %>%
  filter(year == 2015 & estimate > 25000) %>%
  select(state_from, state_to, year) %>%
  rename(State = state_from) %>%
  right_join(state_abbrev) %>%
  rename(code_from = Code, state_from = State, State = state_to) %>%
  select(-Abbrev) %>%
  right_join(state_abbrev) %>%
  rename(code_to = Code) %>%
  select(code_from, code_to, year) %>%
  drop_na()
```

```
## Joining, by = "State"
## Joining, by = "State"
```

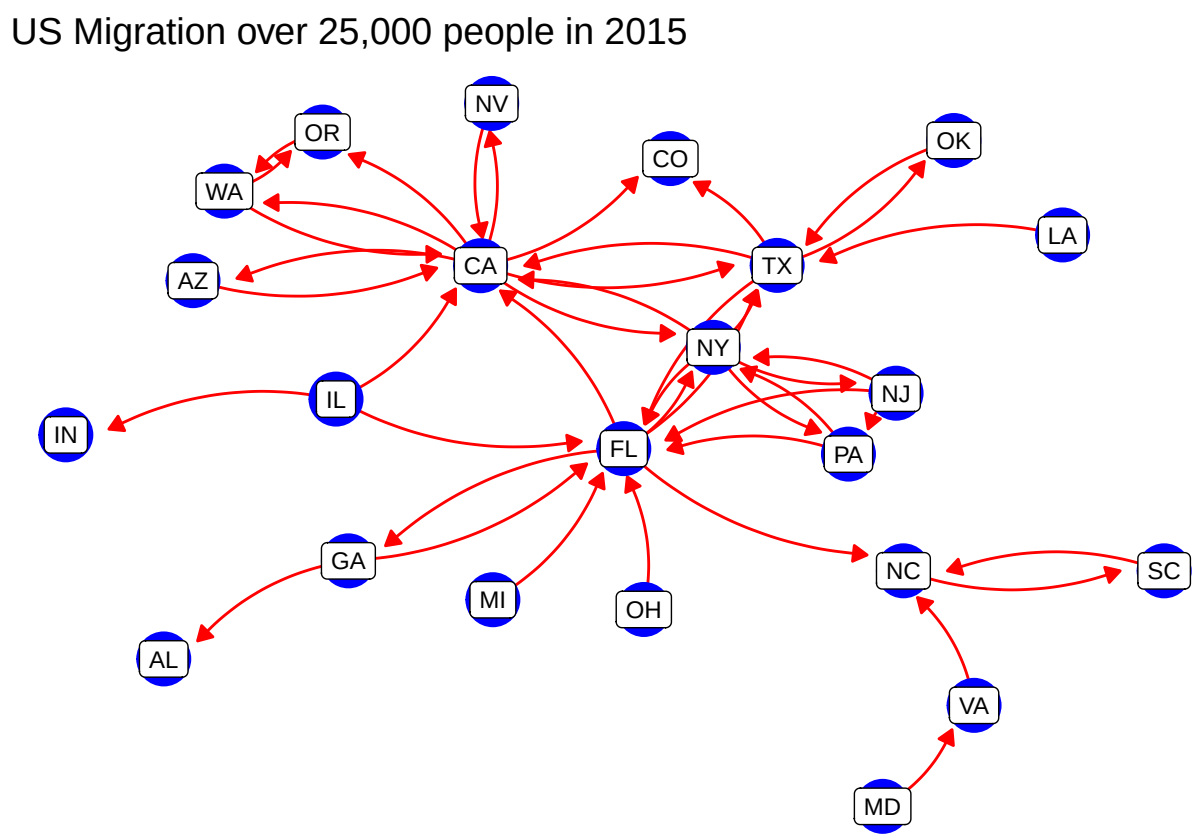
```
q4 <- get.adjacency(graph.data.frame(data2015[,1:2]))
q4_2015 <- network(q4, directed = T)
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```
network::set.edge.attribute(q4_2015, "year", c(data2015$year))

p4 <- ggplot(ggnetwork(q4_2015, layout = "fruchtermanreingold",
                       arrow.gap = 0.04),
             aes(x = x, xend = xend, y = y, yend = yend)) +
  geom_edges(color = "red", curvature = 0.18,
             arrow = arrow(length = unit(6, "pt"), type = "closed")) +
  geom_nodes(color = "blue", size = 9) +
  geom_nodelabel(aes(label = vertex.names), size = 3) +
  theme_blank() +
  labs(title = "US Migration over 25,000 people in 2015")

p4
```



Q5: Create the same graph as in Q4 for the year 2019. Ensure colors, labels, and themes make the data and message easy to understand.

```
data2019 <- Migration %>%
  filter(year == 2019 & estimate > 25000) %>%
  select(state_from, state_to, year) %>%
  rename(State = state_from) %>%
  right_join(state_abbrev) %>%
  rename(code_from = Code, state_from = State, State = state_to) %>%
  select(-Abbrev) %>%
  right_join(state_abbrev) %>%
  rename(code_to = Code) %>%
  select(code_from, code_to, year) %>%
  drop_na()
```

```
## Joining, by = "State"
## Joining, by = "State"
```

```
q5 <- get.adjacency(graph.data.frame(data2019[,1:2]))
q5_2019 <- network(q5, directed = T)
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```
network::set.edge.attribute(q5_2019, "year", c(data2019$year))

p5 <- ggplot(ggnetwork(q5_2019, layout = "fruchtermanreingold",
                       arrow.gap = 0.04),
             aes(x = x, xend = xend, y = y, yend = yend)) +
  geom_edges(color = "red", curvature = 0.18,
             arrow = arrow(length = unit(6, "pt"), type = "closed")) +
  geom_nodes(color = "blue", size = 9) +
  geom_nodelabel(aes(label = vertex.names), size = 3) +
  labs(title = "US Migration over 25,000 people in 2019") +
  theme_blank()

p5
```

US Migration over 25,000 people in 2019