Problem Set 5

Jonah Messinger

2025-03-07

Section 1

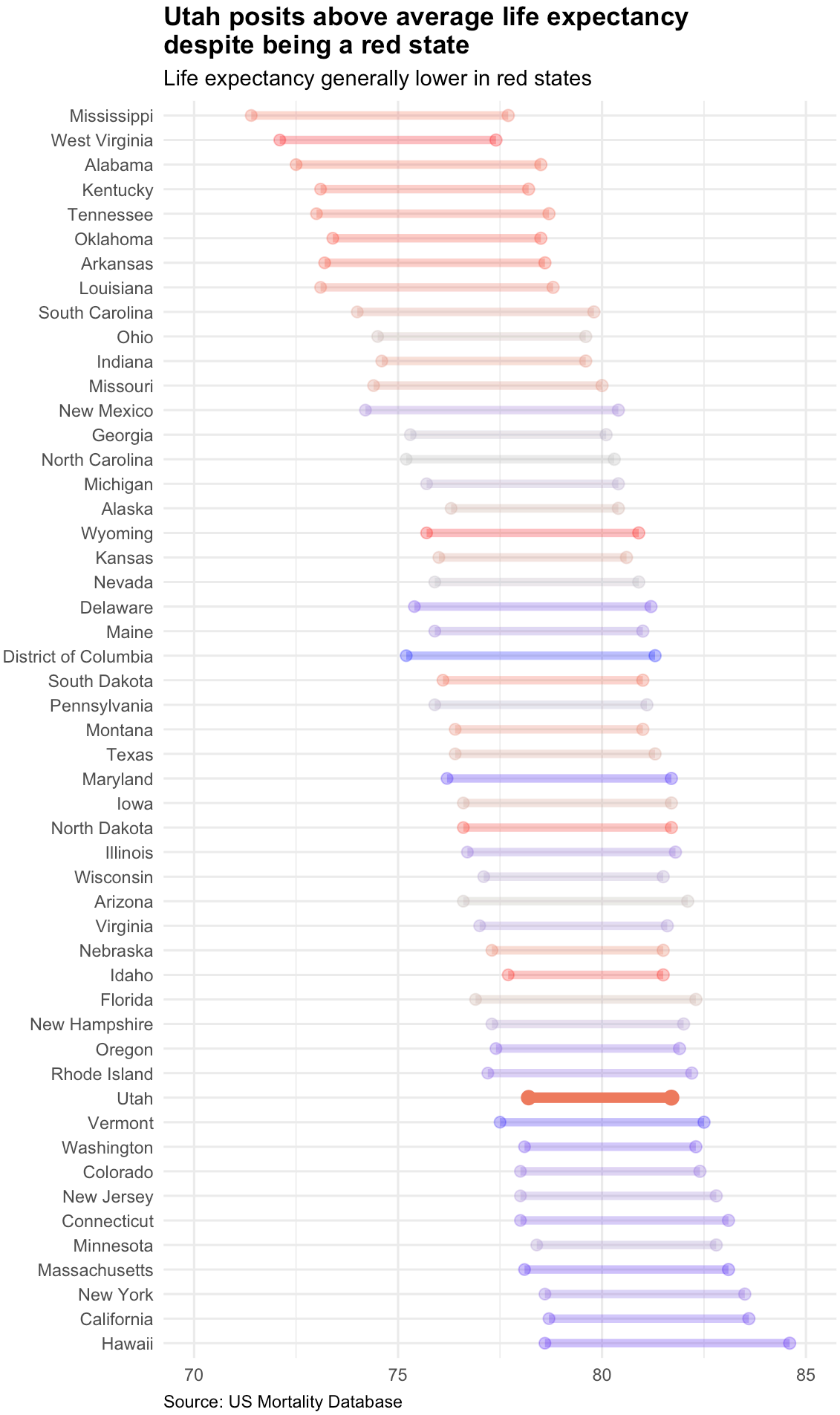
```
# Load the data
url <- "https://en.wikipedia.org/wiki/List_of_U.S._states_and_territories_by_life_expect
ancy"
webpage <- read_html(url)
tables <- html_table(webpage, fill = TRUE)

data <- tables[[2]]

write.csv(data, "life_expectancy.csv")
#Clean the data
data[data == "–"] <- NA
data <- data %>%
  na.omit()

#Remove the periods at the end of the rank column
data$Rank <- gsub("\\.", "", data$Rank)
#Make the rank column numeric
data$Rank <- as.numeric(data$Rank)
data$Male <- as.numeric(data$Male)
data$Female <- as.numeric(data$Female)

colnames(data) <- c("Rank", "State", "Life_Expectancy_19", "Male", "Female")

#Load new data
url2 <- "https://www.presidency.ucsb.edu/statistics/elections/2024"
webpage2 <- read_html(url2)
tables2 <- html_table(webpage2, fill = TRUE)

data2 <- tables2[[1]]

#Remove the first 11 rows
data2 <- data2[-c(1:11),]
#Only keep the first 8 columns
data2 <- data2[,1:8]

colnames(data2) <- c("State", "Total_Votes", "Harris_Votes", "Harris_Pct", "Harris_EV",
"Trump_Votes", "Trump_Pct", "Trump_EV")

data2 <- data2[-c(1:2),]
data2 <- data2[-c(57:60),]
data2 <- data2[-c(31:33),]
data2 <- data2[-c(21:22),]

str(data2)
```

```
## tibble [51 × 8] (S3: tbl_df/tbl/data.frame)
##  $ State       : chr [1:51] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ Total_Votes : chr [1:51] "2,265,090" "338,177" "3,390,161" "1,182,676" ...
##  $ Harris_Votes: chr [1:51] "772,412" "140,026" "1,582,860" "396,905" ...
##  $ Harris_Pct  : chr [1:51] "34.10%" "41.41%" "46.69%" "33.56%" ...
##  $ Harris_EV   : chr [1:51] "" "" "" "" ...
##  $ Trump_Votes : chr [1:51] "1,462,616" "184,458" "1,770,242" "759,241" ...
##  $ Trump_Pct   : chr [1:51] "64.57%" "54.54%" "52.22%" "64.20%" ...
##  $ Trump_EV    : chr [1:51] "9" "3" "11" "6" ...
```

```
#Change all variables but State to numeric
data2$Total_Votes <- as.numeric(gsub("[^0-9]", "", data2$Total_Votes))
data2$Harris_Votes <- as.numeric(gsub("[^0-9]", "", data2$Harris_Votes))
data2$Harris_Pct <- as.numeric(gsub("[^0-9]", "", data2$Harris_Pct))
data2$Harris_EV <- as.numeric(gsub("[^0-9]", "", data2$Harris_EV))
data2$Trump_Votes <- as.numeric(gsub("[^0-9]", "", data2$Trump_Votes))
data2$Trump_Pct <- as.numeric(gsub("[^0-9]", "", data2$Trump_Pct))
data2$Trump_EV <- as.numeric(gsub("[^0-9]", "", data2$Trump_EV))

#Chnage both the Percentage columns to be XX.XX from XXXX
#Place a period before the last two digits
data2$Harris_Pct <- data2$Harris_Pct/100
data2$Trump_Pct <- data2$Trump_Pct/100

#Create a new column for the difference between Harris and Trump vote percentages
data2 <- data2 %>%
  mutate(Difference = Trump_Pct - Harris_Pct)

data_2combine <- data2 %>%
  select(State, Difference)

#Merge the two datasets
data <- merge(data, data_2combine, by = "State")

#Make a new variable that is only a 1 for Utah
data <- data %>%
  mutate(Utah = ifelse(State == "Utah", 1, 0))

#Create a new variable Difference_Rank that is the rank of the difference variable
data <- data %>%
  mutate(Difference_Rank = rank(Difference))
```

```
ggplot(data) +
  geom_dumbbell(aes(y = fct_reorder(State, Rank), x = Male, xend = Female, color = Diffe
rence_Rank, size = Utah, alpha = Utah)) +
  scale_x_continuous(limits = c(70, 85)) +
  scale_color_gradientn(colors = c("blue", "grey", "red")) +
  scale_size_continuous(range = c(2, 2.5)) +
  scale_alpha_continuous(range = c(0.3, 1)) +
  labs(title = "Utah posits above average life expectancy\ndespite being a red state",
       subtitle = "Life expectancy generally lower in red states",
       caption = "Source: US Mortality Database") +
  theme_minimal() +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        legend.position = "none",
        plot.title = element_text(face = "bold"),
        plot.caption = element_text(hjust = 0))
```

```
## Warning: Using the `size` aesthetic with geom_segment was deprecated in ggplot2 3.4.
0.
## ℹ Please use the `linewidth` aesthetic instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Utah posits above average life expectancy despite being a red state
Life expectancy generally lower in red states
Mississippi
West Virginia
Alabama
Kentucky
Tennessee
Oklahoma
Arkansas
Louisiana
South Carolina
Ohio
Indiana
Missouri
New Mexico
Georgia
North Carolina
Michigan
Alaska
Wyoming
Kansas
Nevada
Delaware
Maine
District of Columbia
South Dakota
Pennsylvania
Montana
Texas
Maryland
Iowa
North Dakota
Illinois
Wisconsin
Arizona
Virginia
Nebraska
Idaho
Florida
New Hampshire
Oregon
Rhode Island
Utah
Vermont
Washington
Colorado
New Jersey
Connecticut
Minnesota
Massachusetts
New York
California
Hawaii
70
75
80
85
Source: US Mortality Database

I decided to use a dumbbell plot to show the life expectancy and sort by lowest to highest because I wanted to highlight shared characteristics in the lower and higher states. I used the color of the line to show the difference in vote percentage between Harris and Trump, while using a gradient to show by how much each candidate won. I also used the size of the line to highligh Utah, a red state with an above average life expectancy. The alpha also is used to help highlight Utah. The title of the plot is “Utah posits above average life expectancy despite being a red state” and the subtitle is “Life expectancy generally lower in red states”. The plot shows that Utah has a higher life expectancy than most states, despite being a red state. The plot also shows that life expectancy is generally lower in red states.

Section 2: Life Expectancy Over Time

```
#Load the data
data_slope <- tables[[4]]

colnames(data_slope) <- c("State", "2019", "2010", "2000", "1990", "1980", "1970", "196
0", "1950", "1940", "Difference 1940-2019")

#Remove NAs
data_slope[data_slope == "N/A"] <- NA
data_slope <- data_slope %>%
  na.omit()

#Make the 1940 variable numeric
data_slope$`1940` <- as.numeric(data_slope$`1940`)

#Change the observation of state "District ofColumbia" to "District of Columbia"
data_slope$State[data_slope$State == "District ofColumbia"] <- "District of Columbia"

#Create a new variable for only the states with the smallest and largest differences
##Binary variable that is 1 for District of Columbia and 2 for Oklahoma
data_slope <- data_slope %>%
  mutate(change = ifelse(State %in% c("District of Columbia", "Oklahoma"), 1, 0)) %>%
  mutate(change = as.factor(change)) %>%
  arrange(change)

#Make a variable where DC is 1 and Oklahoma is 2, everything else is 0
data_slope <- data_slope %>%
  mutate(change2 = ifelse(State == "District of Columbia", 1, ifelse(State == "Oklahom
a", 2, 0))) %>%
  mutate(change2 = as.factor(change2))



#Make a slope chart
##Use values from 1940 and 2019
ggplot(data_slope) +
  geom_segment(aes(x = 1940, xend = 2019, y = `1940`, yend = `2019`, group = reorder(Sta
te, change), color = change2, alpha = change, size = change)) +
  geom_point(aes(x = 1940, y = `1940`, color = change2, alpha = change), size = 3) +
  geom_point(aes(x = 2019, y = `2019`, color = change2, alpha = change), size = 3) +
  geom_rect(aes(xmin = 2021, xmax = 2035, ymin = 55, ymax = 82), fill = "white") +
  scale_x_continuous(limits = c(1940, 2035), breaks = seq(1940, 2019, 79)) +
  scale_color_manual(values = c("grey", "blue", "red")) +
  scale_alpha_manual(values = c(0.4, 1)) +
  scale_size_manual(values = c(1, 2)) +
  geom_label(data = data_slope %>% filter(change == 1),
    aes(x = 2020, y = `2019`, label = State), nudge_y = 1, nudge_x = -1) +
  coord_cartesian(xlim = c(1940, 2030)) +
  labs(title = "District of Columbia has seen the largest increase \nin life expectancy
since 1940",
       subtitle = "Oklahoma has seen the smallest increase",
       caption = "Source: Global Data Lab") +
  theme_minimal() +
```

```
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        legend.position = "none",
        panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_line(color = "grey", size = 0.5),
        plot.title = element_text(face = "bold", margin = margin(l = -14, b = 5)),
        plot.subtitle = element_text(margin = margin(l = -14)),
        plot.caption = element_text(hjust = -0.07, vjust = -1))
```

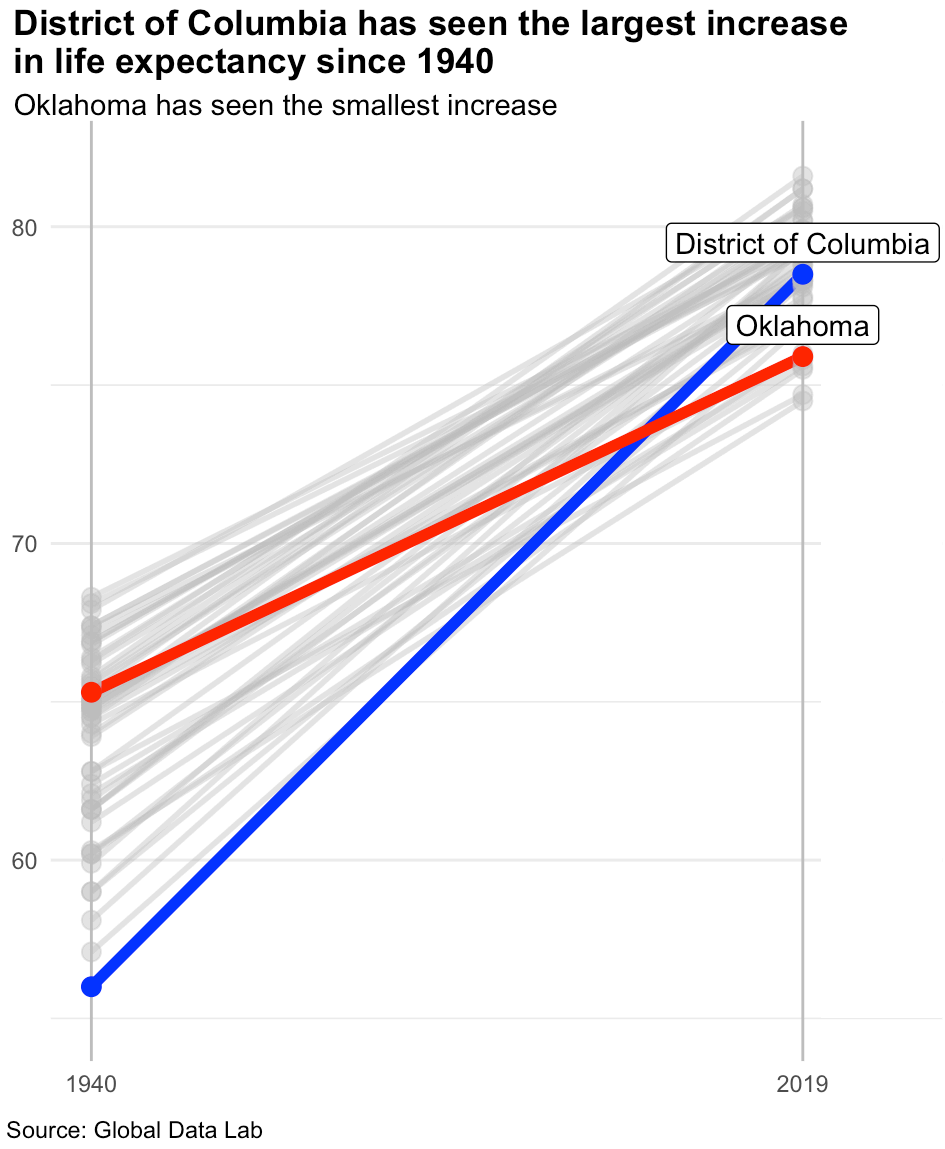
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## ℹ Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in geom_rect(aes(xmin = 2021, xmax = 2035, ymin = 55, ymax = 82), : All aesth
etics have length 1, but the data has 52 rows.
## ℹ Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



I decided to use a slope chart to show the change in life expectancy over time because I wanted to show the general trends of the states. I used color to highlight the states with the smallest and largest differences in life expectancy, with red being used for Oklahoma and blue for the District of Columbia. I thought that made sense of OK is usually considered a red state and DC a blue, tyring to play on some pre-attentive features. I used the alpha layer to fade all of the other states into the background, that way they could be used as reference but did not hinder the main point of the graph. I removed the legend and instead just highlighted our two states of interest. I also made sure that these lines of interest were brought to the front of the plot. The title is active and aptly describes the trend seen between the greatest increase in life expectancy and the smallest increase. I also removed the x-axis minor grids lines since it was a slope chart, making the only relevvant grid lines the major ones of start and finish.

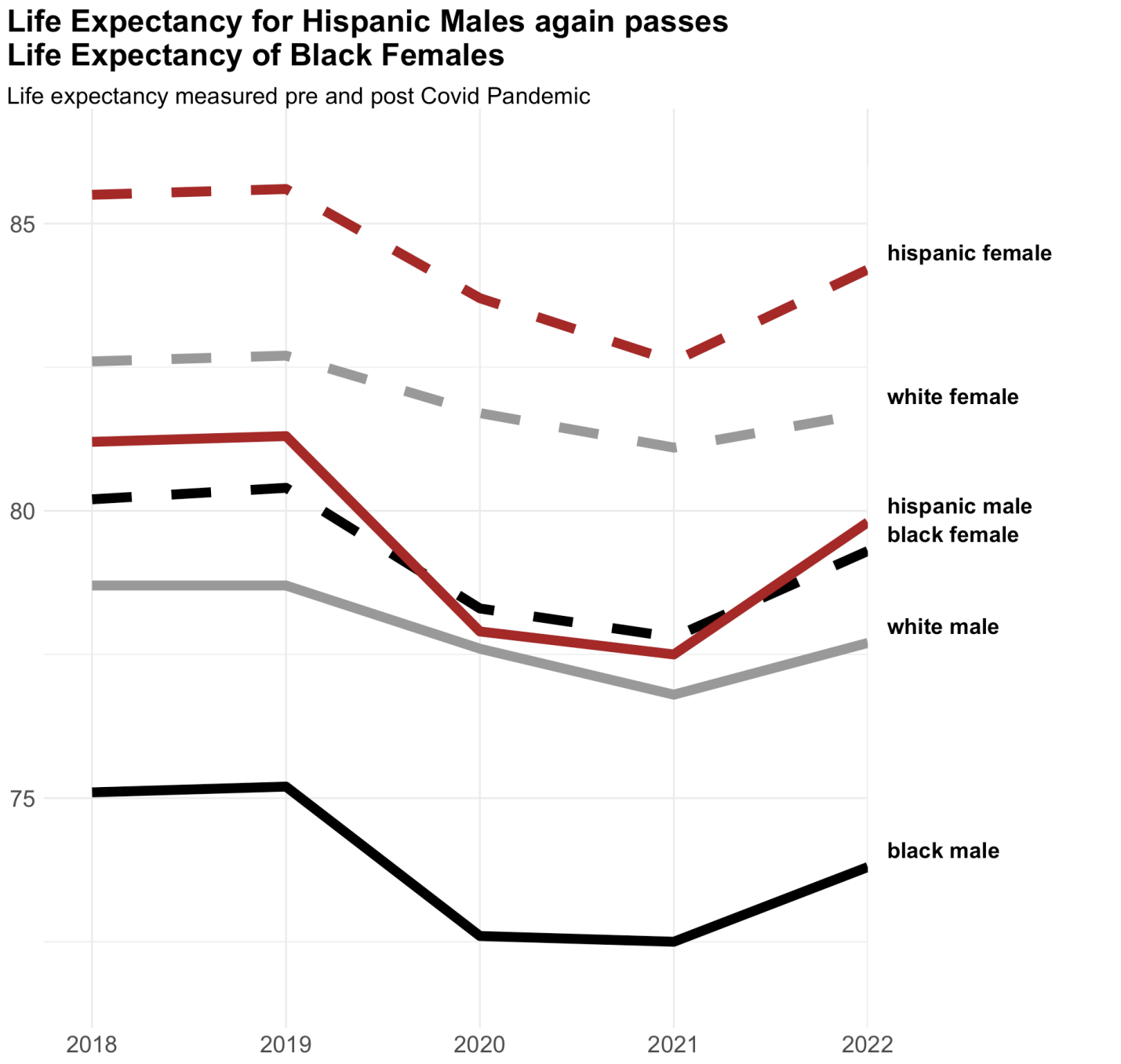
Section 3: Life Expectancy by Race and Gender

```
#Load the data
data3 <- read_csv("~/Desktop/POLI 301/Problem Set 5/data_cdc_LifeExp.csv")
```

```
## Rows: 42 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): race, gender
## dbl (2): year, life_exp
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Filter the data to include only hispanic, white and black individuals
data3 <- data3 %>%
  filter(race == "hispanic" | race == "white" | race == "black")

#Make a line plot
ggplot(data3) +
  geom_line(aes(x = year, y = life_exp, group = interaction(race, gender), color = race,
linetype = gender), size = 3) +
  geom_rect(aes(xmin = 2022, xmax = 2023.4, ymin = 71, ymax = 86), fill = "white") +
  geom_text(data = data3 %>% filter(year == 2022),
            aes(x = year, y = life_exp, label = paste(race, gender)),
            nudge_x = 0.1, nudge_y = 0.3, hjust = 0, size = 5, fontface = "bold") +
  scale_color_manual(values = c("black", "brown", "grey60")) +
  scale_linetype_manual(values = c("dashed", "solid")) +
  scale_x_continuous(limits = c(2018, 2023.5), breaks = seq(2018, 2022, 1)) +
  scale_y_continuous(limits = c(71, 87), expand = c(0, 0), breaks = seq(75, 85, 5)) +
  coord_cartesian(xlim = c(2018, 2023)) +
  labs(title = "Life Expectancy for Hispanic Males again passes\nLife Expectancy of Blac
k Females",
       subtitle = "Life expectancy measured pre and post Covid Pandemic",
       caption = "Source: CDC, National Vital Statistics Reports, 2018-2022") +
  theme_minimal() +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        plot.title = element_text(face = "bold", size = 20, margin = margin(l = -2, b =
5)),
        plot.subtitle = element_text(size = 15, margin = margin(l = -2, t = 5)),
        plot.title.position = "plot",
        legend.position = "none",
        plot.caption = element_text(hjust = 0, size = 15, margin = margin(t = 20, l = -2
2)))
```

```
## Warning in geom_rect(aes(xmin = 2022, xmax = 2023.4, ymin = 71, ymax = 86), : All aes
thetics have length 1, but the data has 30 rows.
## ℹ Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



This graph can be used to highlight the differences in life expectancy between racial and gender groups. The “big takeaway” that I chose to highlight was the proximity of life expectancy between Hispanic males and Black Female, specifically highlighting how Hispanic males one again surpass and now have a higher life expectancy than Black females. This can be used to write an article regarding how Covid-19 may have more negatively affected Hispanic men, casuing the steeper decline in life expectancy we see in the graph. The fact that the life expectancy then returns to normal, or rather above that of a Black female, can be used as further evidence that the period of low life expectancy was due to Covid-19.

Section 4: Replication-ish

```
#Load the data
data4 <- read_csv("~/Desktop/POLI 301/Problem Set 5/UN_LifeExp.csv")
```

```
## Rows: 1584 Columns: 6
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): Region, subregion, country or area *, ISO3 Alpha-code, ISO2 Alpha-code
## dbl (3): Index, Year, Life_Exp
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Rename the columns
colnames(data4) <- c("Index", "Country", "ISO3", "ISO2", "Year", "Life_Exp")

#Create a binary variable for the United States
data4 <- data4 %>%
  mutate(US = ifelse(Country == "United States of America", 1, 0)) %>%
  mutate(US = as.factor(US))

#Create a new variable that is binary 1 for the US and Japan and 0 for all other countri
es
data4 <- data4 %>%
  mutate(US_Japan = ifelse(Country %in% c("United States of America", "Japan"), 1, 0)) %
>%
  mutate(US_Japan = as.factor(US_Japan))

#Create a new data set for the Average Life Expectancy in each year
data4_avg <- data4 %>%
  group_by(Year) %>%
  summarise(Avg_Life_Exp = mean(Life_Exp))

#Create a line plot
ggplot(data4) +
  geom_line(aes(x = Year, y = Life_Exp, group = Country, color = US, size = US, alpha =
US_Japan)) +
  geom_text(data = data4 %>% filter(Country %in% c("United States of America", "Japan"))
%>% filter(Year == 2023),
            aes(x = Year, y = Life_Exp, label = paste0(Life_Exp, " years"), color = US),
            nudge_x = 0.5, nudge_y = 0.5, hjust = 0, size = 4) +
  geom_text(data = data4 %>% filter(Country == "Japan") %>% filter(Year == 2023),
            aes(x = Year, y = Life_Exp, label = Country, color = US),
            nudge_x = 0.5, nudge_y = 2, hjust = 0, size = 4) +
  geom_line(data = data4_avg, aes(x = Year, y = Avg_Life_Exp), color = "grey40", size =
1.5) +
  geom_text(data = data4_avg %>% filter(Year == 2023),
            aes(x = Year, y = Avg_Life_Exp, label = paste0(round(Avg_Life_Exp, 1), " yea
rs")),
            color = "grey30", nudge_x = 0.5, nudge_y = 0.5, hjust = 0, size = 4) +
  scale_color_manual(values = c("0" = "grey40", "1" = "#C18B00")) +
  scale_size_manual(values = c(0.5, 1.5)) +
  scale_alpha_manual(values = c(.2, 1)) +
  scale_y_continuous(expand = c(0, 0), breaks = seq(60, 85, 5), limits = c(64, 87), labe
ls = function(x) paste0(x, " years")) +
  scale_x_continuous(expand = c(0, 1), breaks = seq(1980, 2025, 10), limits = c(1980, 20
28)) +
  labs(title = "American life expectancy is dropping – and it's not all covid's fault",
       subtitle = "Life Expectancy in wealthy countries",
       caption = "Source: United Nations, Department of Economic and Social Affairs, Pop
ulation Division.") +
  annotate("text", x = 2016, y = 76, label = "United States", color = "#C18B00", size =
4.5, fontface = "bold") +
  annotate("text", x = 1997, y = 84, label = "Average of high-income countries", color =
"grey30", size = 4, hjust = 1) +
```

```
  geom_curve(aes(x = 1997.25, y = 83.5, xend = 2007, yend = 78.75),
             curvature = -0.15, arrow = arrow(type = "open", angle = 40, length = unit
(0.06, "inches")), color = "grey30", size = 0.15) +
  theme_minimal() +
  theme(legend.position = "none",
        panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.y = element_line(color = "grey", size = 0.5),
        axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        plot.caption = element_text(hjust = -0.2, vjust = -1,
                                    size = 8),
        plot.title = element_text(face = "bold", size = 15, hjust = -0.42),
        plot.subtitle = element_text(size = 12, hjust = -0.122))
```

```
## Warning in geom_curve(aes(x = 1997.25, y = 83.5, xend = 2007, yend = 78.75), : All ae
sthetics have length 1, but the data has 1584 rows.
## ℹ Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```