STATS472_final_draft

2024-04-17

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Loading Packages

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(stringr)
library(knitr)
#install.packages("remotes")
#remotes::install_github("ateucher/lutz")
library(lutz)
#if (!requireNamespace("nnet", quietly = TRUE)) install.packages("nnet")
library(nnet)
#install.packages("kableExtra")
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

library(tidyr)
#install.packages("viridis")
library(viridis)

## Loading required package: viridisLite
```

Cleaning the Data

```
# Load the CSV file into a data frame

daily<- read.csv("combineddailyAQI.csv")
tzn_counties<- read.csv("uscounties.csv")
daily$Date <- as.Date(daily$Date)
daily <- daily %>%
  mutate(
    Year = year(Date),
    Month = month(Date)
  )

cleaned_daily <- daily %>%
  filter(!is.na(Year), !is.na(Month), !is.na(AQI))

cleaned_daily$AQI <- as.numeric(as.character(cleaned_daily$AQI))

monthly_summary <- daily %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(
    AvgAQI = mean(AQI),
    .groups = 'drop'
  )

## Warning: There were 131 warnings in `summarise()`.
## The first warning was:
## ℹ In argument: `AvgAQI = mean(AQI)`.
## ℹ In group 1: `Year = 2013`, `Month = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 130 remaining warnings.

daily$Date <- as.Date(daily$Date, format = "%Y-%m-%d")


monthly_summary <- daily %>%
```

```
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(
    AvgAQI = mean(AQI, na.rm = TRUE),
    .groups = 'drop'
  )
```

```
## Warning: There were 131 warnings in `summarise()`.
## The first warning was:
## ℹ In argument: `AvgAQI = mean(AQI, na.rm = TRUE)`.
## ℹ In group 1: `Year = 2013`, `Month = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 130 remaining warnings.
```

```
cleaned_daily$AQI <- as.numeric(as.character(cleaned_daily$AQI))
monthly_summary <- cleaned_daily %>%
  group_by(State.Name, county.Name, Year, Month) %>%
  summarize(AvgAQI = mean(AQI, na.rm = TRUE), .groups = 'drop')

cleaned_seasoned <- cleaned_daily %>%
  mutate(Month = month(Date),
         Season = case_when(
           Month %in% c(3, 4, 5) ~ "Spring",
           Month %in% c(6, 7, 8) ~ "Summer",
           Month %in% c(9, 10, 11) ~ "Autumn",
           Month %in% c(12, 1, 2) ~ "Winter",
           TRUE ~ NA_character_
         ))
```

Clean dataset for the timezone Dataset

```
#tzn_counties <- tzn_counties %>%
  #select(-county_ascii)

combined_df <- left_join(cleaned_daily, tzn_counties, by = c("State.Name" = "
state_name", "county.Name" = "county"))
```

```
## Warning in left_join(cleaned_daily, tzn_counties, by = c(State.Name = "sta
te_name", : Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 120079 of `x` matches multiple rows in `y`.
## ℹ Row 297 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

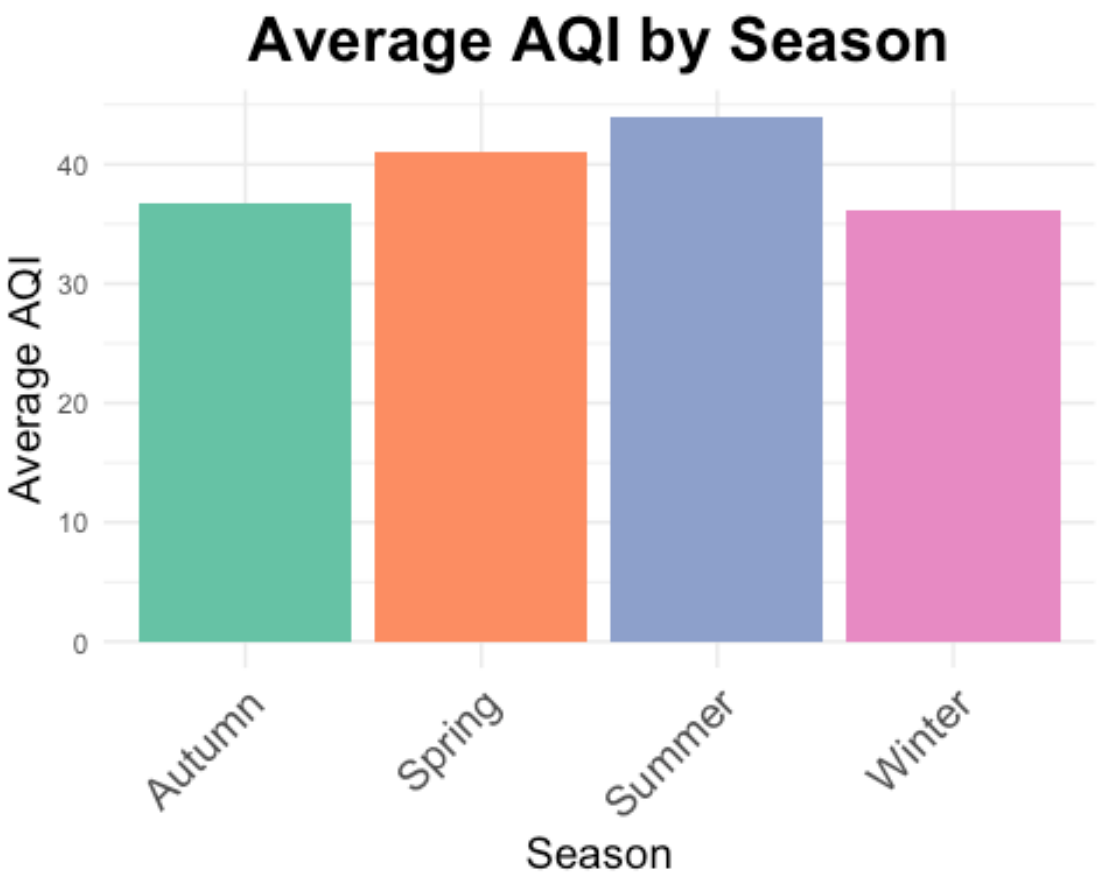
```
# Adding time zone information to our combined daily.
combined_df$tz <- tz_lookup_coords(lat = combined_df$lat, lon = combined_df$l
ng, method = "fast")
```

```
## Warning: Using 'fast' method. This can cause inaccuracies in time zones
##   near boundaries away from populated ares. Use the 'accurate'
##   method if accuracy is more important than speed.
```

```
cleaned_combined_df <- combined_df %>%
  filter(!is.na(tz))

cleaned_combined_df <- cleaned_combined_df %>%
  mutate(Month = month(Date),
         Season = case_when(
           Month %in% c(3, 4, 5) ~ "Spring",
           Month %in% c(6, 7, 8) ~ "Summer",
           Month %in% c(9, 10, 11) ~ "Autumn",
           Month %in% c(12, 1, 2) ~ "Winter",
           TRUE ~ NA_character_ # Default case
         ))

cleaned_combined_df <- cleaned_combined_df[, !names(cleaned_combined_df) %in%
c('State.Code', 'County.Code', 'population', 'Defining.Site','county_fips')]

cleaned_combined_df1 <- cleaned_combined_df[, c("AQI", "Season", "tz")]
```

EDA#1 Average AQI by Season

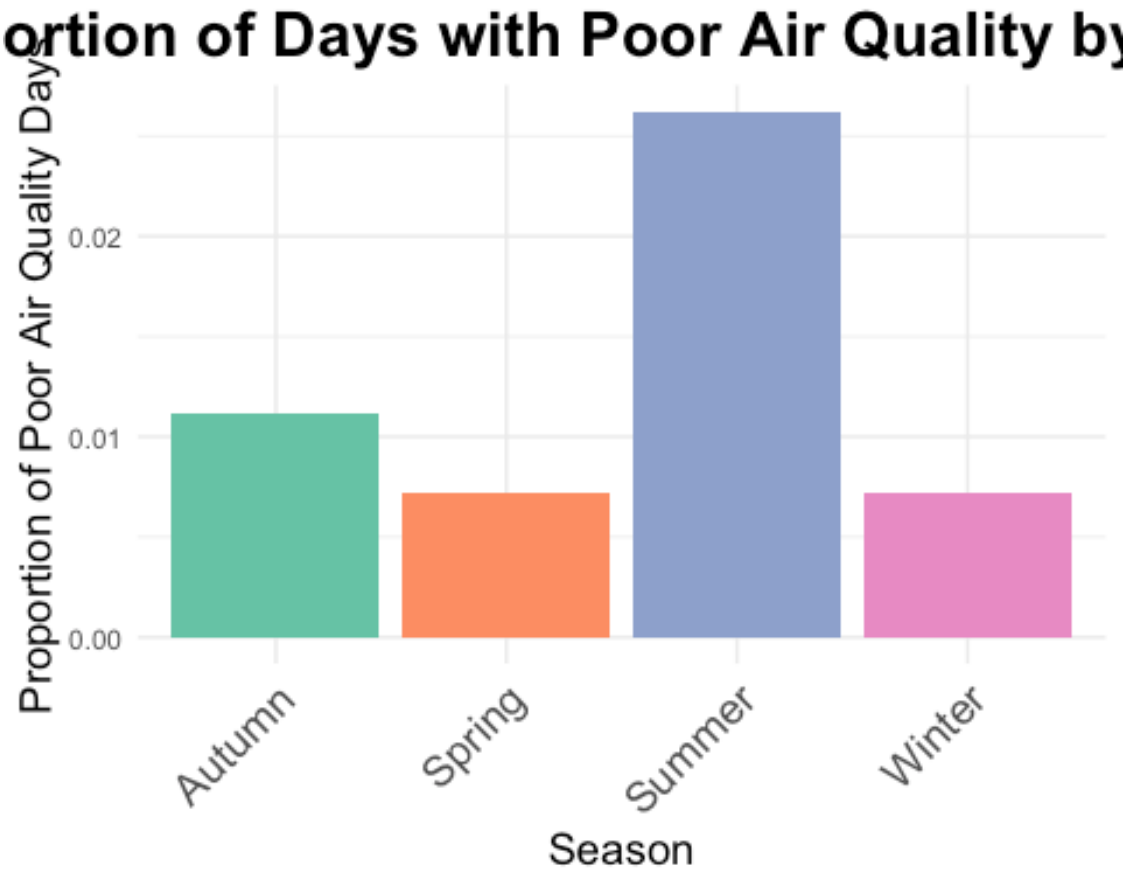
```
avg_aqi_by_season <- cleaned_seasoned %>%
  group_by(Season) %>%
  summarise(AvgAQI = mean(AQI, na.rm = TRUE))



ggplot(avg_aqi_by_season, aes(x = Season, y = AvgAQI, fill = Season)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average AQI by Season",
       x = "Season",
       y = "Average AQI") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") +
  guides(fill = FALSE) +
  theme(
    axis.text.x = element_text(size = 14, angle = 45, hjust = 1, vjust = 1),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5)
  )

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none
" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



EDA#2 Proportion of Days with Poor Air Quality by Season

```
cleaned_seasoned <- cleaned_seasoned %>%
  mutate(PoorAirQuality = AQI > 100)

proportion_poor_aqi_by_season <- cleaned_seasoned %>%
  group_by(Season) %>%
  summarise(
    TotalDays = n(),
    PoorQualityDays = sum(PoorAirQuality, na.rm = TRUE),
    ProportionPoor = PoorQualityDays / TotalDays
  )

ggplot(proportion_poor_aqi_by_season, aes(x = Season, y = ProportionPoor, fil
l = Season)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Proportion of Days with Poor Air Quality by Season",
    x = "Season",
    y = "Proportion of Poor Air Quality Days"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") +
```

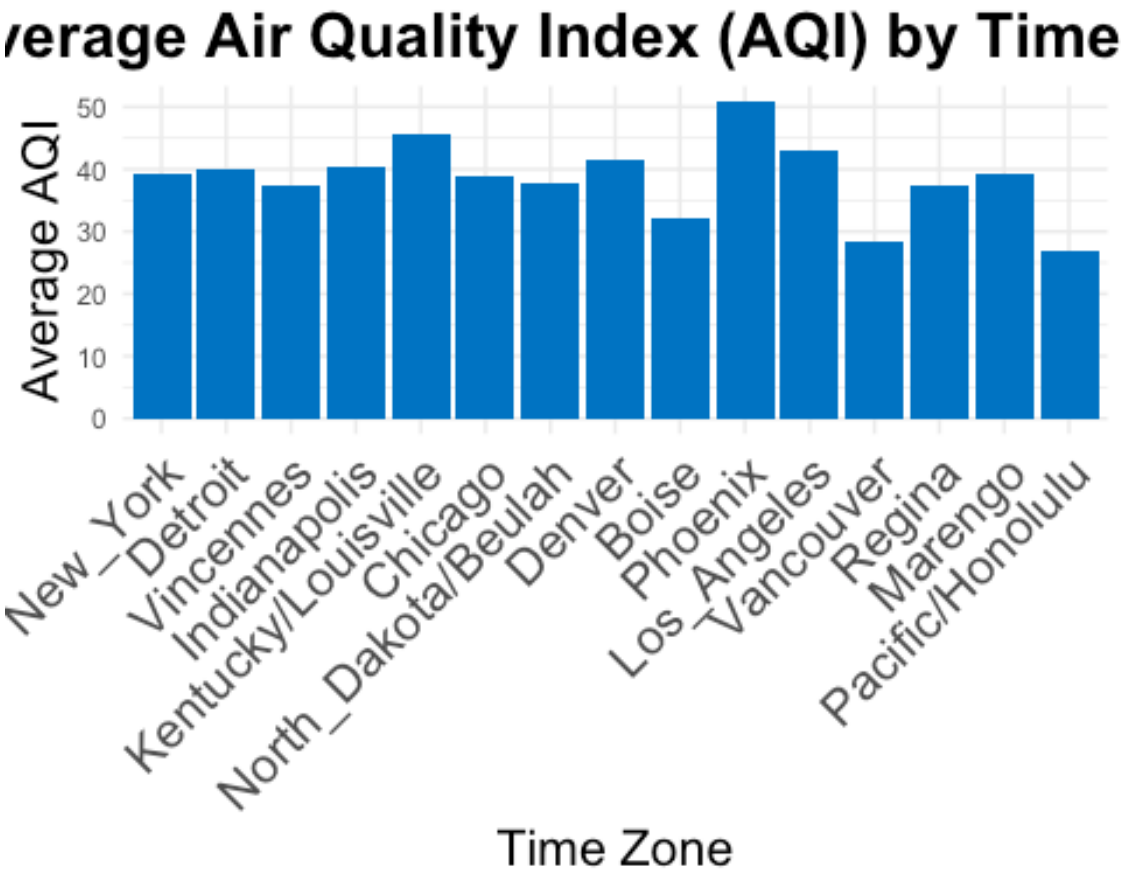
```
  guides(fill = FALSE)+
  theme(
    axis.text.x = element_text(size = 14, angle = 45, hjust = 1, vjust = 1),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5)
  )
```



EDA#3 Average Air Quality Index (AQI) by Time Zone

```
##EDA
cleaned_combined_df$tz <- sub("America/", "", cleaned_combined_df$tz)
cleaned_combined_df$tz <- sub("Indiana/", "", cleaned_combined_df$tz)
average_aqi_by_tzn <- cleaned_combined_df %>%
  group_by(tz) %>%
  summarise(AverageAQI = mean(AQI, na.rm = TRUE))


tz_order <- c("New_York", "Detroit", "Vincennes", "Indianapolis",
              "Kentucky/Louisville", "Chicago", "North_Dakota/Beulah",
              "Denver","Boise", "Phoenix", "Los_Angeles", "Vancouver", "Regin
a",
              "Marengo", "Pacific/Honolulu")
```

```r
average_aqi_by_tzn$tz <- factor(average_aqi_by_tzn$tz, levels = tz_order)


ggplot(average_aqi_by_tzn, aes(x = tz, y = AverageAQI)) +
  geom_bar(stat = "identity", fill = "#0073C2FF") +
  labs(
    title = "Average Air Quality Index (AQI) by Time Zone",
    x = "Time Zone",
    y = "Average AQI"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1, size = 16),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16)
  )
```




```r
cleaned_combined_df <- cleaned_combined_df %>%
  mutate(Month = month(Date),
         Season = case_when(
```

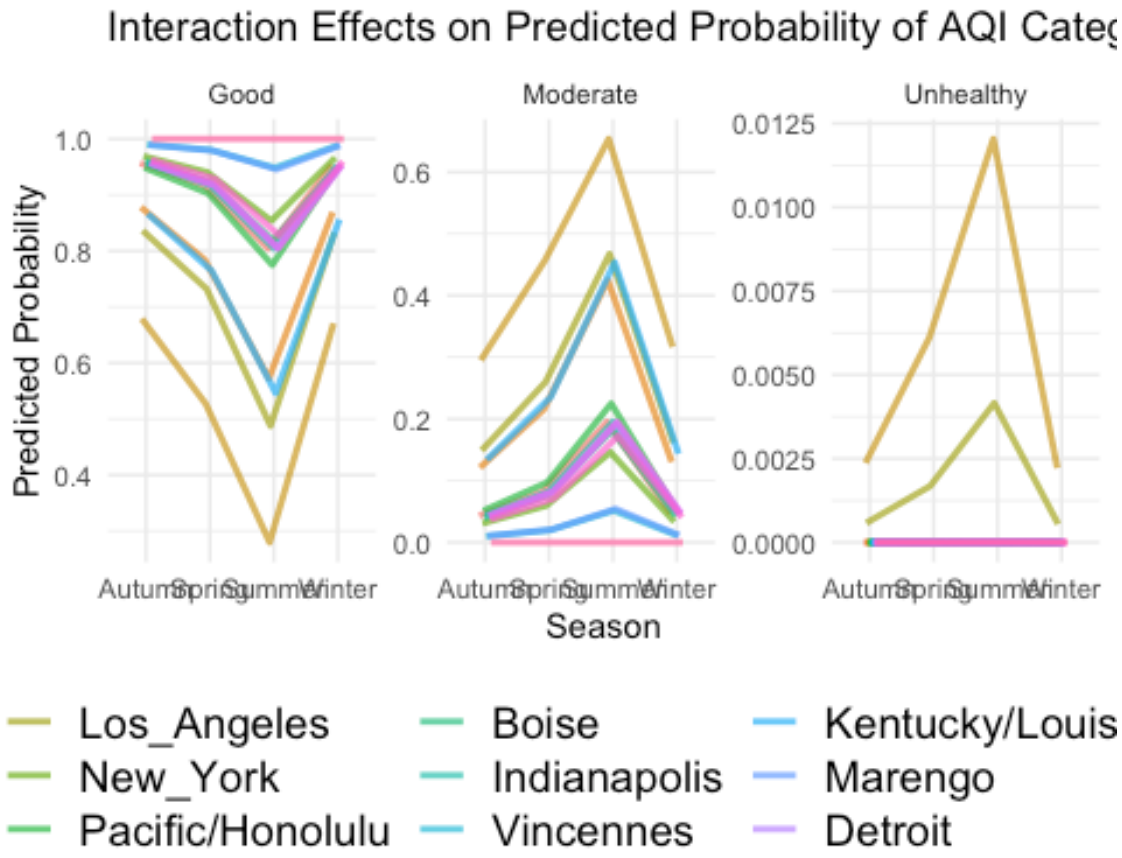
```
            Month %in% c(3, 4, 5) ~ "Spring",
            Month %in% c(6, 7, 8) ~ "Summer",
            Month %in% c(9, 10, 11) ~ "Autumn",
            Month %in% c(12, 1, 2) ~ "Winter",
            TRUE ~ NA_character_
          ))

cleaned_combined_df <- cleaned_combined_df[, !names(cleaned_combined_df) %in%
c('State.Code', 'County.Code', 'population', 'Defining.Site','county_fips')]

cleaned_combined_df1 <- cleaned_combined_df[, c("AQI", "Season", "tz")]

df_aggregated <- cleaned_combined_df %>%
  group_by(State.Name, county.Name, tz, Season, Year) %>%
  summarise(AverageAQI = mean(AQI, na.rm = TRUE), .groups = 'drop')

df_aggregated$AQI_Category <- cut(df_aggregated$AverageAQI,
                        breaks = c(-Inf, 50, 100, 150, 200, 300, Inf),
                        labels = c("Good", "Moderate", "Unhealthy for Sensitiv
e Groups",
                                   "Unhealthy", "Very Unhealthy", "Hazardous")
,
                        include.lowest = TRUE)
```

Multinomial Logistic Regression

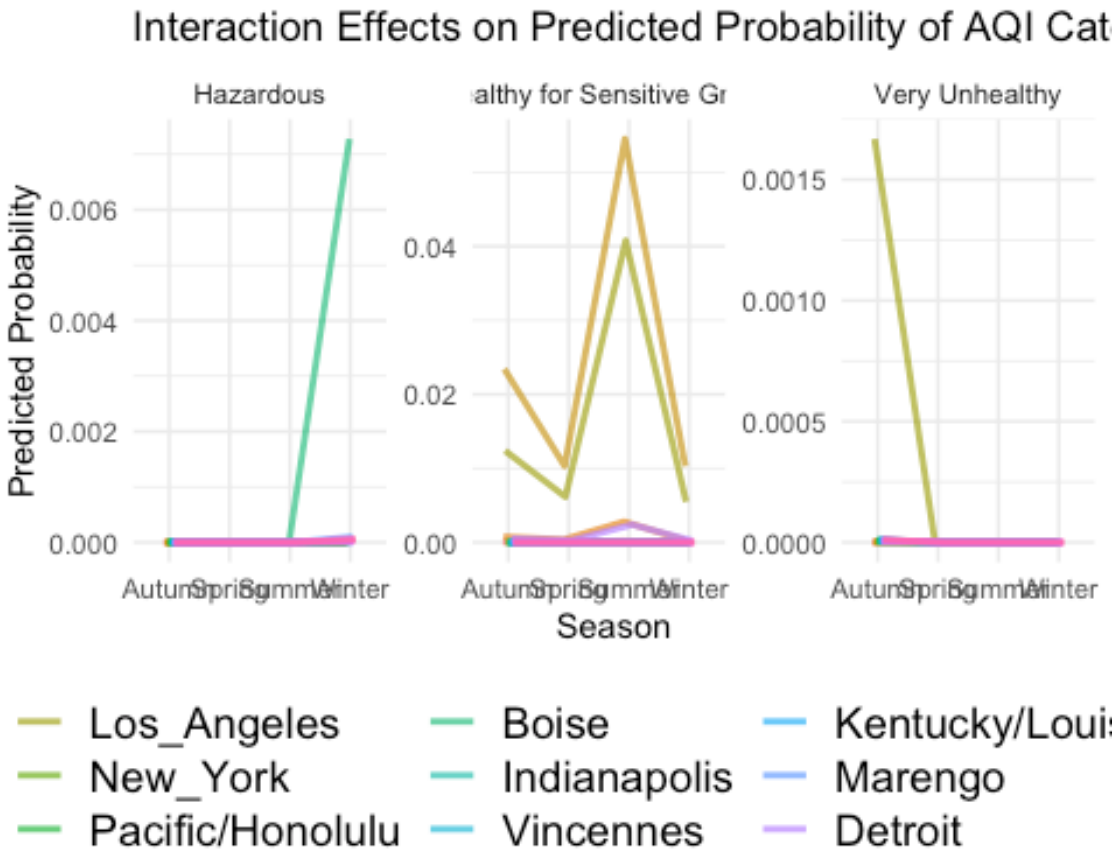
```
mlr_model <- multinom(AQI_Category ~ Season + tz, data = df_aggregated)

## # weights:  114 (90 variable)
## initial  value 72544.757390
## iter  10 value 14536.193877
## iter  20 value 13891.301601
## iter  30 value 13856.674445
## iter  40 value 13854.032499
## iter  50 value 13853.815783
## iter  60 value 13853.752758
## final  value 13853.734004
## converged

seasons <- unique(df_aggregated$Season)
time_zones <- unique(df_aggregated$tz)
prediction_data <- expand.grid(Season = seasons, tz = time_zones)

predicted_probs <- predict(mlr_model, newdata = prediction_data, type = "prob
s")

prediction_data <- cbind(prediction_data, predicted_probs)

predicted_probs <- predict(mlr_model, newdata = prediction_data, type = "prob
s")
```

```r
predicted_probs_df <- as.data.frame(predicted_probs)

prediction_data <- cbind(prediction_data, predicted_probs_df)

prediction_data_long <- pivot_longer(prediction_data,
                                     cols = -c(Season, tz),
                                     names_to = 'AQI_Category',
                                     values_to = 'PredictedProbability')

number_of_tz <- prediction_data_long %>%
  pull(tz) %>%
  unique() %>%
  length()

first_three_categories <- c("Good", "Moderate", "Unhealthy")
prediction_data_first <- filter(prediction_data_long, AQI_Category %in% first
_three_categories)

remaining_categories <- setdiff(unique(prediction_data_long$AQI_Category), fi
rst_three_categories)
prediction_data_second <- filter(prediction_data_long, AQI_Category %in% rema
ining_categories)

#Plot for first slide
ggplot(prediction_data_first, aes(x = Season, y = PredictedProbability, color
= tz, group = tz)) +
  geom_line(size = 1, position = position_dodge(width = 0.2), alpha = 0.7) +
  facet_wrap(~AQI_Category, scales = 'free_y', ncol = 3) +
  labs(x = 'Season', y = 'Predicted Probability', color = 'Time Zone') +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 15)) +
  ggtitle('Interaction Effects on Predicted Probability of AQI Categories')
```

```
## 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.
```




```r
# Plot for the second slide
ggplot(prediction_data_second, aes(x = Season, y = PredictedProbability, colo
r = tz, group = tz)) +
 geom_line(size = 1, position = position_dodge(width = 0.2), alpha = 0.7) +
  facet_wrap(~AQI_Category, scales = 'free_y', ncol = 3) +
  labs(x = 'Season', y = 'Predicted Probability', color = 'Time Zone') +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 15)) +
  ggtitle('Interaction Effects on Predicted Probability of AQI Categories')
```

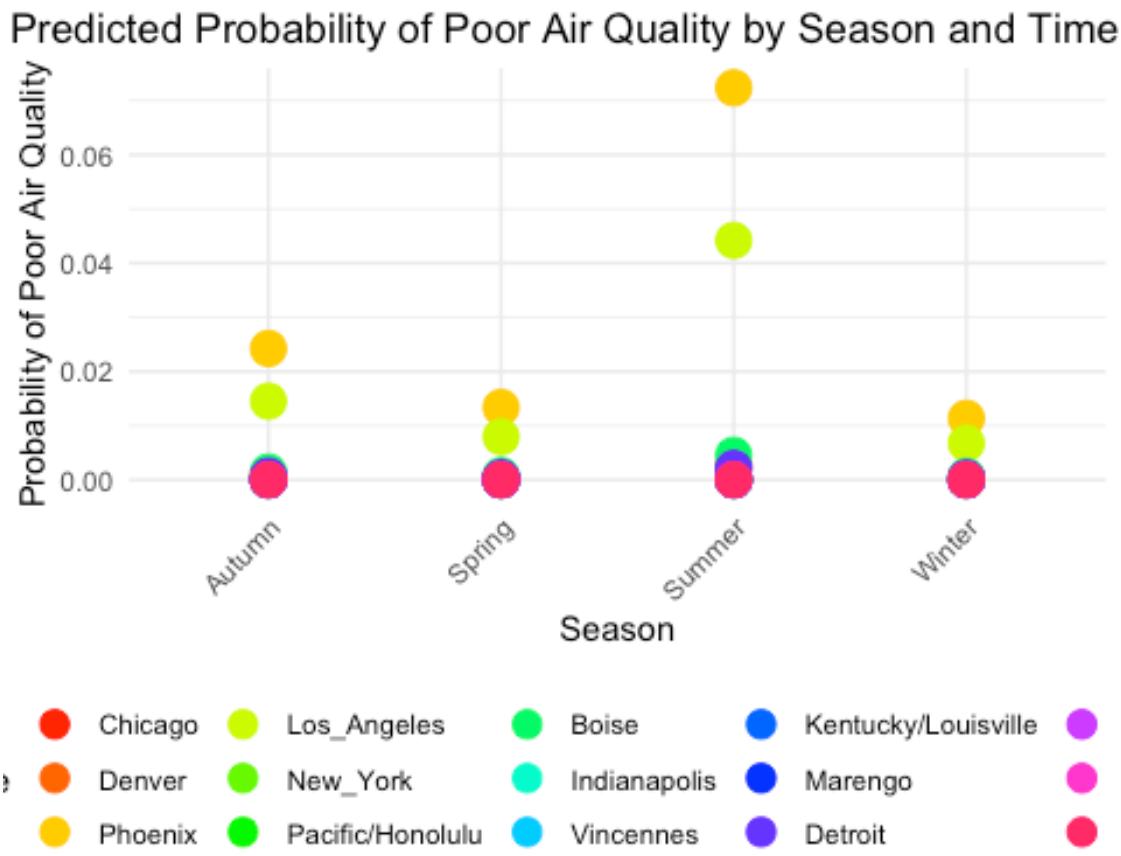



```r
# Redefining the AQI categories to binary outcome
df_aggregated2 <- df_aggregated %>%
  mutate(PoorAirQuality = as.factor(AverageAQI > 100))

# Binary logistic regression model
binary_lr_model <- glm(PoorAirQuality ~ Season + tz, data = df_aggregated2, f
amily = binomial)

summary(binary_lr_model)

##
## Call:
## glm(formula = PoorAirQuality ~ Season + tz, family = binomial,
##     data = df_aggregated2)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -6.5381     1.0205  -6.407 1.49e-10 ***
## SeasonSpring             -0.6073     0.3517  -1.727  0.08426 .
## SeasonSummer              1.1443     0.2464   4.644 3.41e-06 ***
## SeasonWinter             -0.7717     0.3711  -2.080  0.03756 *
## tzChicago                -3.1398     1.4154  -2.218  0.02654 *
## tzDenver                 -0.5519     1.0971  -0.503  0.61492
```

```
## tzDetroit                -0.7763     1.4159  -0.548  0.58350
## tzIndianapolis          -16.2499  1454.3224  -0.011  0.99109
## tzKentucky/Louisville   -16.2207  5031.8773  -0.003  0.99743
## tzLos_Angeles             2.3196     1.0071   2.303  0.02126 *
## tzMarengo               -16.3053  7582.5703  -0.002  0.99828
## tzNew_York              -16.2432   383.1600  -0.042  0.96619
## tzNorth_Dakota/Beulah   -16.2207  5031.8773  -0.003  0.99743
## tzPacific/Honolulu      -16.2124  3611.1469  -0.004  0.99642
## tzPhoenix                 2.8427     1.0340   2.749  0.00597 **
## tzRegina                -16.2207  7116.1491  -0.002  0.99818
## tzVancouver             -16.2207  7116.1491  -0.002  0.99818
## tzVincennes             -16.2207  5031.8773  -0.003  0.99743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1613.4  on 40487  degrees of freedom
## Residual deviance: 1137.1  on 40470  degrees of freedom
## AIC: 1173.1
##
## Number of Fisher Scoring iterations: 21
```

```
seasons <- unique(df_aggregated$Season)
time_zones <- unique(df_aggregated$tz)
season_timezone_predictions <- expand.grid(Season = seasons, tz = time_zones)

season_timezone_predictions$PoorAirQualityProb <- predict(binary_lr_model, ne
wdata = season_timezone_predictions, type = "response")

season_timezone_predictions$group_id <- interaction(season_timezone_predictio
ns$Season, season_timezone_predictions$tz)
more_colors <- grDevices::rainbow(length(unique(season_timezone_predictions$t
z)))
ggplot(season_timezone_predictions, aes(x = Season, y = PoorAirQualityProb, c
olor = tz)) +
  geom_point(size = 5) +
  scale_color_manual(values = more_colors) +
  labs(
    title = "Predicted Probability of Poor Air Quality by Season and Time Zon
e",
    x = 'Season',
    y = 'Probability of Poor Air Quality',
    color = 'Time Zone'
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1),  # Adjust text angle f
or legibility
```

```
    plot.title = element_text(hjust = 0.5)  # Center the plot title
  ) +
  guides(color = guide_legend(override.aes = list(size=4)))
```




```
#Bigger plot for slides
ggplot(season_timezone_predictions, aes(x = Season, y = PoorAirQualityProb, c
olor = tz)) +
  geom_point(size = 5) +
  geom_line()+
  scale_color_manual(values = more_colors) +
  labs(
    title = "Predicted Probability of Poor Air Quality by Season and Time Zon
e",
    x = 'Season',
    y = 'Probability of Poor Air Quality',
    color = 'Time Zone'
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    legend.text = element_text(size = 16), # Increase legend text size
    legend.title = element_text(size = 14), # Increase legend title size
    axis.text.x = element_text(angle = 45, hjust = 1),  # Adjust text angle f
```

```
or legibility
    plot.title = element_text(hjust = 0.5)  # Center the plot title
  ) +
  guides(color = guide_legend(override.aes = list(size = 6))) # Increase the
size of the points in the legend

## `geom_line()`: Each group consists of only one observation.
## ℹ Do you need to adjust the group aesthetic?
```