````
---
title: "cs09: Weather Impacts on Buffalo Traffic"
subtitle: "Using Open Data APIs"
format:
  html: default
editor: visual
---

## Data Sources

Two complementary datasets:

1.  Weather Data – NOAA's Climate Data Online (CDO) API Provides daily temperature, rainfall, and snowfall from local weather stations.
2.  Traffic Incident Data – City of Buffalo's Open Data Portal Contains daily reports of 911 and 311 calls, including traffic incidents.

## Retrieve the Data

Data was retrieved using httr:httr, content:httr, and jsonlite:endJSON

```{r, message=FALSE, warning=FALSE}
library(httr)
library(jsonlite)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)

library(knitr)

# ==== USER SETTINGS ====
cdo_token <- "fiIVgrYYVdiXYBLIairHbbComGlDHqrM"

lat <- 42.8864
lon <- -78.8784
start_date <- "2024-10-01"
end_date   <- "2025-03-31"

# ==== API CALL ====
data_url <- "https://www.ncei.noaa.gov/cdo-web/api/v2/data"
params <- list(
  datasetid = "GHCND",
  stationid = "GHCND:USW00014733",   # Buffalo Airport
  startdate = start_date,
  enddate = end_date,
  datatypeid = "TMIN,PRCP,SNOW",
  units = "metric",
  limit = 1000
)
```

## Write the API Call Using httr

```{r, message=FALSE, warning=FALSE}

#Had to use add_headers inside of the GET call to apply my access token
#Had to use as = "text" to convert data to a format endJSON could read

response <- GET(data_url, query = params, add_headers(Token = cdo_token)) %>%
  content(as = "text")

noaa <- fromJSON(response)
````
````

```
#Pulling the proper data from the json text

noaa <- noaa$results

#Converted date column using as_date
noaa <- noaa %>%
  mutate(date = as_date(noaa$date))

#Used pivot_wider to give the variables their own columns
noaa_table <- noaa %>%
  pivot_wider(names_from = datatype, values_from = value)

#Used group_by to sort the data by date, and then summarize(across(everything())) to fill in
missing data for each date in the each row
noaa_table <- noaa_table %>%
  mutate(attributes = NULL) %>%
  group_by(date) %>%
  summarise(across(everything(), na.omit))

#Used !duplicate to take out duplicated rows in my noaa_table data (only keeping one row for each
date)
noaa_table <- noaa_table[!duplicated(noaa_table), ]

#Used kable to see some of the data
kable(noaa_table[1:10, ], format = "pipe")

```

```{r}

#Plotted temperature and snowfall over time
ggplot(data = noaa_table) + geom_line(aes(x = date, y = SNOW), color = "darkgreen", linetype =
"dashed") + geom_line(aes(x = date, y = TMIN), color = "darkblue") + labs(title = "Daily Min
Temperature and Snowfall (Buffalo Airport)", x = " Date", y = "Temperature(C)/Snow")


```

```{r}

# ==== API CALL ====

#Set parameters for traffic data; Had to set limit to download all of the availale data
socrata_url <- "https://data.buffalony.gov/resource/6at3-hpb5.json"
soc_params <- list(
  "$limit" = 5000,
  "$where" =
  "report_date >= '2024-10-01' AND
  report_date <= '2025-03-31'"
)

#Same GET call as for weather data, minus the access token
traffic_res <- GET(socrata_url, query = soc_params) %>%
  content(as = "text")

traffic <- fromJSON(traffic_res)

#Converted JSON output to tibble
traffic <- as_tibble(traffic)

#Used kable to see some of the data in a table
kable(traffic[1:10, ], format = "pipe")

#Converted to date column, grouped by date, and used n() from dplyr to count the complaintids on
each date and show that value in the complaintid column
```

```
traffic <- traffic %>%
  mutate(date = as_date(report_date)) %>%
  group_by(date) %>%
  summarise(complaintid = n())

#Graphed daily trends
ggplot(traffic) + geom_line(aes(x = date, y = complaintid)) + labs(title = "Daily Traffic
Accidents in Buffalo", x = "Date", y = "Count")
```

Merge Weather and Traffic Data

```{r, message=FALSE, warning=FALSE}

#Combined weather and traffic data
weather_traffic <- noaa_table %>% left_join(traffic)

#Used replace_na() to make all NA values = 0; Used wday to create a weekday column
weather_traffic <- weather_traffic %>%
  replace_na(list(x = 0)) %>%
  mutate(date = as_date(date)) %>%
  mutate(weekday = wday(date, label = TRUE))

#Created this vector to divide the data by snowfall
custom_breaks <- c(-1, 10, 50, 100, 150, 500)

#Used cut and my vector to create a column grouping snow into bins to be used in the boxplots
later
weather_traffic <- weather_traffic %>%
  mutate(snow_cut = cut(SNOW, breaks = custom_breaks))

ggplot(weather_traffic) + geom_line(aes(x = date, y = complaintid), color = "red") +
geom_line(aes(x = date, y = SNOW), color = "darkgreen", linetype = "dashed") + labs(title =
"Accidents vs. Snowfall Overtime", x = "Date", y = "Accidents/Snow")
```

Visual Exploration - Accidents vs. Weather

```{r}
#Accidents by day of the week boxplot
weekday_box <- boxplot(complaintid~weekday, data = weather_traffic, xlab = "Day of Week", ylab =
"Accident Count", main = "Accidents by Day of the Week")

```

```{r}
#Accidents by snowfall amount boxplot
snow_box <- boxplot(complaintid~snow_cut, data = weather_traffic,  xlab = "Snowfall (mm)", ylab =
"Accident Count", main = "Accidents by Snowfall")

```

```{r}
#Linear graph of Accidents vs. Snowfall
ggplot(data = weather_traffic, aes(x = SNOW, y = complaintid)) + geom_point(aes(color = weekday))
+ geom_smooth(method = "lm") + labs(title = "Accidents vs. Snowfall", x = "Snow (mm)", y =
"Accident Count", color = "Day of the Week")
```

```{r}
#Linear graph of Accidents vs. Rainfall
weather_traffic <- weather_traffic %>%
  mutate(log_precip = log1p(PRCP))

ggplot(data = weather_traffic, aes(x = PRCP, y = complaintid)) + geom_point(aes(color = weekday))
```

```
+ geom_smooth(method = "lm") + labs(title = "Accidents vs. Rainfall", x = "Rain (mm)", y =
"Accident Count")
```

Fit a Linear Model

```{r}
#Regression model (complaintid relationship to snow and precip) showing expected increases in prcp
and snow with increased accidents (prcp seems to have a stronger influence)
lm_model <- lm(complaintid ~ SNOW + PRCP, data = weather_traffic)

tidy_model <- broom::tidy(lm_model)

kable(tidy_model, format = "pipe")
```