Temperature Regime Analysis by City

Pat McCornack

2023-04-20

About:

These are the results of the Climate Analysis performed for my capstone project for DATA 424. In this collaborative project we assessed the vulnerability of Boeing Facilities to climate change impacts based on their geographic location using parameters related to climate, electricity prices, and energy resource mixes. The climate analysis component, found here, was narrowed to analyzing temperature trends for simplicity and as that is the component that will have the most influence on energy costs.

A subset of the charts and summary statistics found here are incorporated into an R Shiny web app that maps Boeing facilities and allows users to dynamically explore the data related to the three facets of our analysis for each facility.

This is currently a work in progress.

Data Source and Information

Dataset: Compiled historical daily temperature and precipitation data for selected 210 U.S. cities.

Citation: Lai, Yuchuan; Dzombak, David (2019): Compiled historical daily temperature and precipitation data for selected 210 U.S. cities. Carnegie Mellon University. Dataset. https://doi.org/10.1184/R1/7890488.v5

This dataset is comprised of individual files of precipitation and temperature records for 210 U.S. cities. Datasets were compiled using the Global Historical Climatological Network - a service provided by NOAA comprised of climatalogical data collected at the weather station level. For each city, data from one or more stations were merged to create a longitudinal dataset with some records beginning before 1900.

The dataset was last updated 2022-01-13 and can be accessed at this link: https://kilthub.cmu.edu/articles/dataset/Compiled_daily_temperature_and_precipitation_data_for_the_U_S_cities/7890488/5

Initial EDA

The climate records for each city are stored in a directory of files identified by the city ID. An accompanying file named 'city_info.csv' contains city names, IDs, and other metadata. The following allows a user to select a city from the dataset for the analysis by specifying the 'city' variable. It then queries the 'city_info' table to find the associated ID and loads in the dataframe for analysis.

The contents of rows with NA are checked before removal.

Initial processing includes changing the 'Date' data type to Date, separating that out to year/month/day features, and creating an average temperature feature 'tavg' using the maximum and minimum temperatures.

An initial overview using the summary statistics doesn't suggest any issues with the data.

```
# Specify the city for analysis
city = 'Seattle'
```

```r
city_info <- read.csv('./raw_data/city_info.csv')

# Load in the data for the city
file_path <- city_info %>%
  filter(Name == city) %>%
  select(ID)
file_path <- paste('./raw_data/', file_path[1,], '.csv', sep = "")
city_df <- read.csv(file_path)

# Check the rows with NAs before removing them
check_na <- city_df[which(is.na(city_df)),]

# Initial data processing
city_df <- city_df %>%
  select(-X, -prcp) %>%
  na.omit() %>%
  mutate(tavg = (tmax + tmin)/2) %>%
  mutate(Date = ymd(Date)) %>%
  mutate(year = year(Date)) %>%
  mutate(month = month(Date)) %>%
  mutate(day = month(Date))

head(city_df)
```

```
##         Date tmax tmin tavg year month day
## 1 1894-01-01   40   36 38.0 1894     1   1
## 2 1894-01-02   40   35 37.5 1894     1   1
## 3 1894-01-03   38   30 34.0 1894     1   1
## 4 1894-01-04   36   28 32.0 1894     1   1
## 5 1894-01-05   37   22 29.5 1894     1   1
## 6 1894-01-06   36   26 31.0 1894     1   1
```

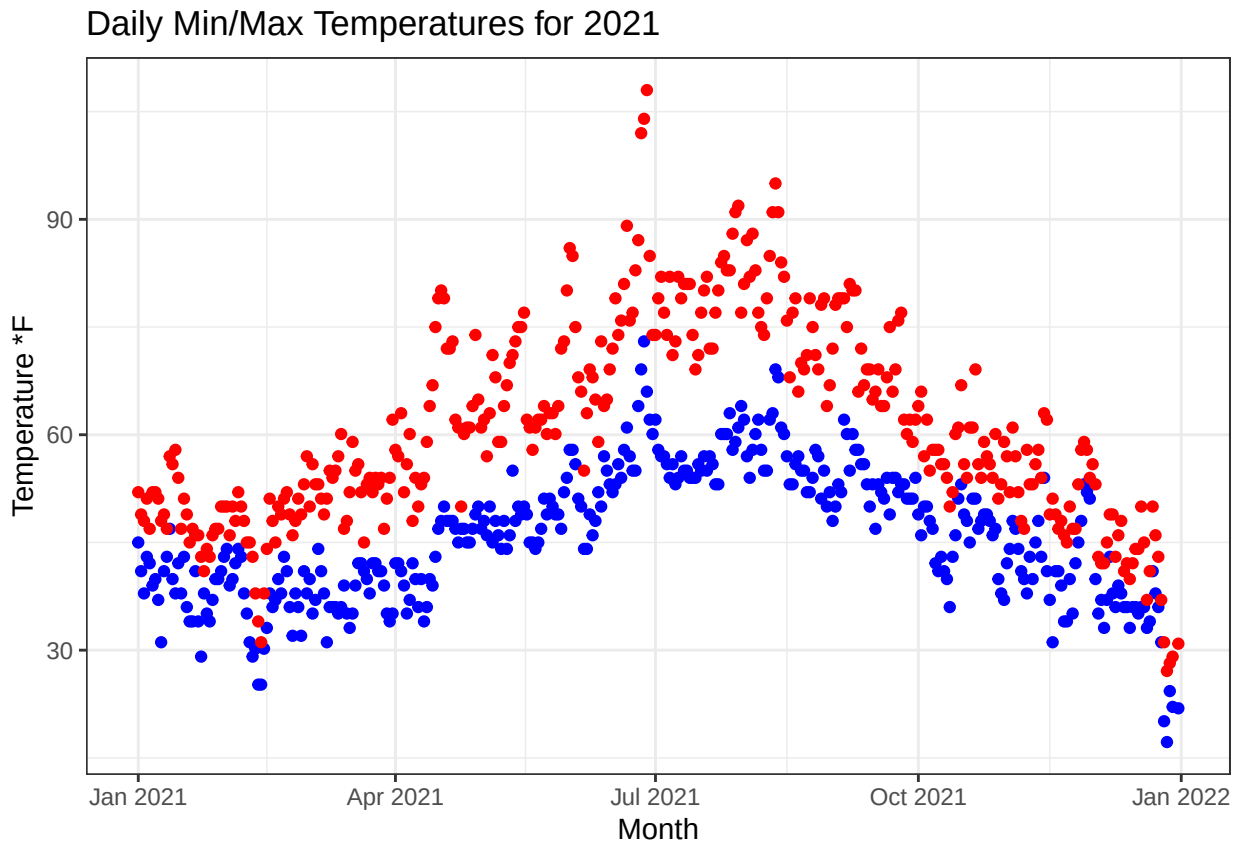
```r
summary(city_df)
```

```
##       Date                tmax             tmin            tavg
##  Min.   :1894-01-01   Min.   : 16.00   Min.   : 0.00   Min.   :10.00
##  1st Qu.:1926-01-13   1st Qu.: 50.00   1st Qu.:39.00   1st Qu.:44.50
##  Median :1958-01-20   Median : 58.00   Median :45.00   Median :51.50
##  Mean   :1958-01-13   Mean   : 59.28   Mean   :44.85   Mean   :52.07
##  3rd Qu.:1990-01-09   3rd Qu.: 68.00   3rd Qu.:52.00   3rd Qu.:60.00
##  Max.   :2021-12-31   Max.   :108.00   Max.   :73.00   Max.   :88.50
##       year          month             day
##  Min.   :1894   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:1926   1st Qu.: 4.000   1st Qu.: 4.000
##  Median :1958   Median : 7.000   Median : 7.000
##  Mean   :1958   Mean   : 6.525   Mean   : 6.525
##  3rd Qu.:1990   3rd Qu.:10.000   3rd Qu.:10.000
##  Max.   :2021   Max.   :12.000   Max.   :12.000
```

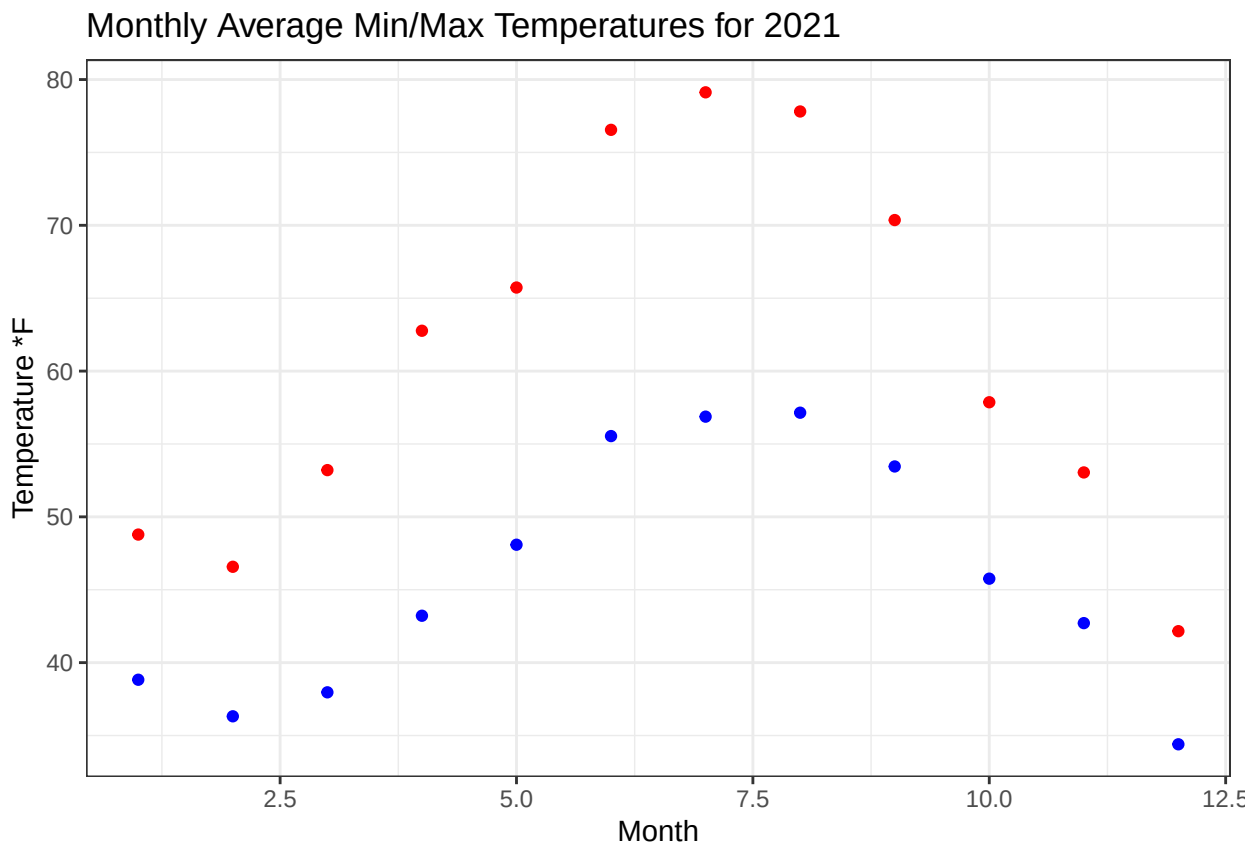
Preliminary Visualizations

The following plots display temperature trends for the year of 2019 as a spot check for abnormalities. Red dots represent maximum temperatures. Blue dots represent minimum temperatures

```r
city_2021 <- city_df %>% filter(year == '2021')

# Daily Min/Max
ggplot(city_2021) +
  geom_point(aes(x = Date, y = tmin), color = 'blue') +
  geom_point(aes(x = Date, y = tmax), color = 'red') +
  ggtitle("Daily Min/Max Temperatures for 2021") +
  ylab('Temperature *F') +
  xlab('Month') +
  theme_bw()
```




```r
# Aggregate by month to get average monthly maxes and mins
monthly_avg <- city_2021 %>%
  group_by(month) %>%
  summarise(tmin_avg = mean(tmin),
            tmax_avg = mean(tmax),
            .groups = 'drop')

ggplot(monthly_avg) +
  geom_point(aes(x = month, y = tmin_avg), color = 'blue') +
  geom_point(aes(x = month, y = tmax_avg), color = 'red') +
  ggtitle("Monthly Average Min/Max Temperatures for 2021") +
  ylab('Temperature *F') +
  xlab('Month') +
  theme_bw()
```



Heating and Cooling Degree Days

Degree days are an indicator of energy usage in an area. A degree day compares the mean temperature to 65 degrees F, and the difference is the number of degree days. The value of 65 is chosen as the typical temperature that is maintained in a room by heating/cooling.

Heating degree days (HDD) occur when the mean temperature is below 65 deg. F, and are an indicator of how cold a region is.

Cooling degree days (CDD) occur when the mean temperature is above 65 deg. F, and are an indicator of how hot a region is.

The formula for the number of degree days on a given day is:
$|mean - 65|$ where it is a HDD if the mean is under 65 deg. F and it is a CDD if the temperature is above 65 deg. F.

Trends in degree days in a region over time can be an indicator of how the climate is changing at that location - and how energy needs are changing at that location.
Below, the DD column is the number of degree days for a given day. If DD is positive they are CDD, if DD is negative then they are HDD.

Degree days for 2021

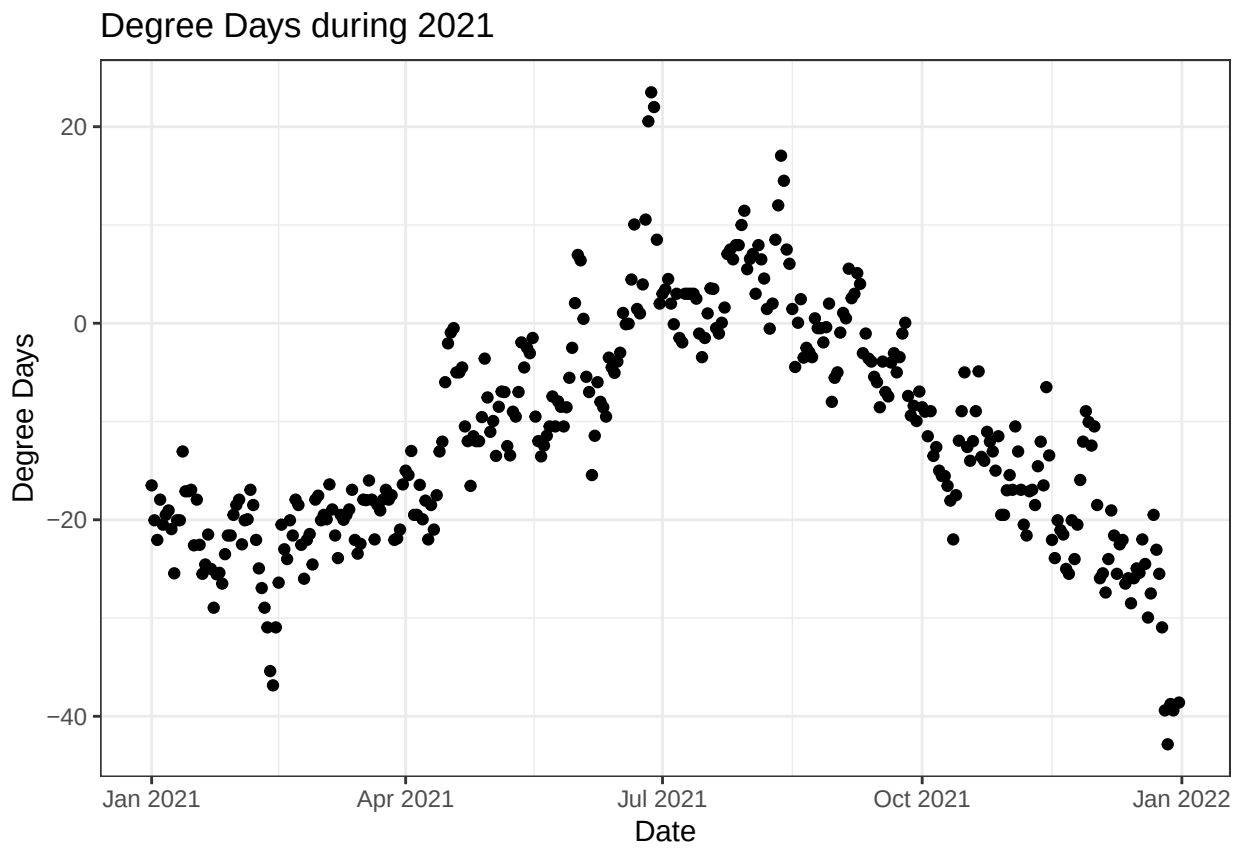
The following computes HDD/CDD for 2021 and creates visualizations.

The trend for Seattle suggests that energy is more typically used for heating rather than cooling, except for a three month time span during the summer. This makes sense given Seattle's latitude and maritime climate. Also noticeable is the spike in CDDs caused by the summer 2021 heat wave that hit the Pacific Northwest.

```r
# Functions to create HDD/CDD columns
calc_CDD <- function(deg.day){
  if (deg.day > 0)
  {
    return(deg.day)
  }
  else {
    return(0)
  }
}

calc_HDD <- function(deg.day){
  if (deg.day < 0)
  {
    return(abs(deg.day))
  }
  else {
    return(0)
  }
}

# Add degree day column then use that to create HDD and CDD columns
city_2021 <- city_2021 %>%
  mutate(DD = tavg - 65)

city_2021$CDD <- sapply(city_2021$DD, calc_CDD)
city_2021$HDD <- sapply(city_2021$DD, calc_HDD)

head(city_2021)
```

```
##         Date tmax tmin  tavg year month day     DD CDD   HDD
## 1 2021-01-01 52.0 45.0 48.50 2021     1   1 -16.50   0 16.50
## 2 2021-01-02 48.9 41.0 44.95 2021     1   1 -20.05   0 20.05
## 3 2021-01-03 48.0 37.9 42.95 2021     1   1 -22.05   0 22.05
## 4 2021-01-04 51.1 43.0 47.05 2021     1   1 -17.95   0 17.95
## 5 2021-01-05 46.9 42.1 44.50 2021     1   1 -20.50   0 20.50
## 6 2021-01-06 52.0 39.0 45.50 2021     1   1 -19.50   0 19.50
```

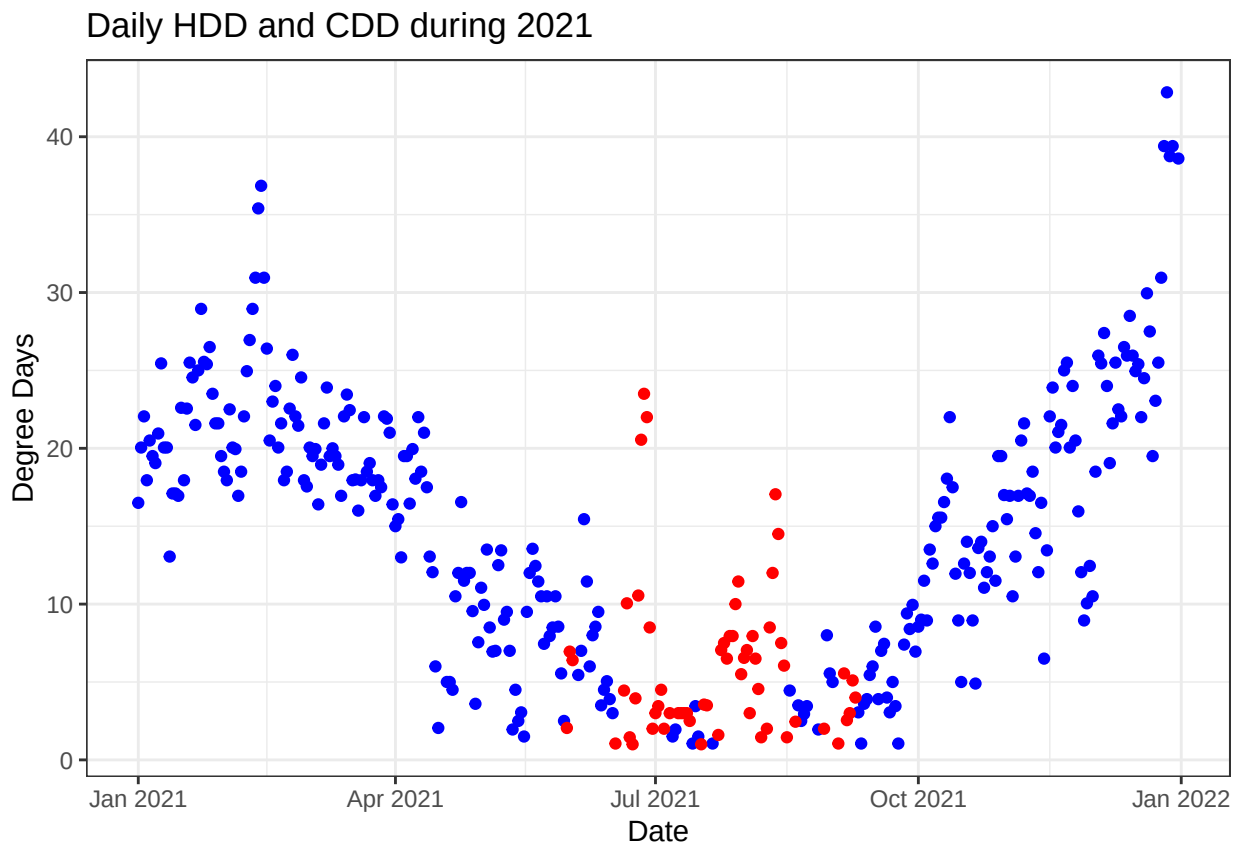
```r
# Visualization of DD over the year
ggplot(city_2021, aes(Date, DD)) +
  geom_point() +
  ggtitle('Degree Days during 2021') +
  xlab('Date') +
  ylab('Degree Days') +
  theme_bw()
```



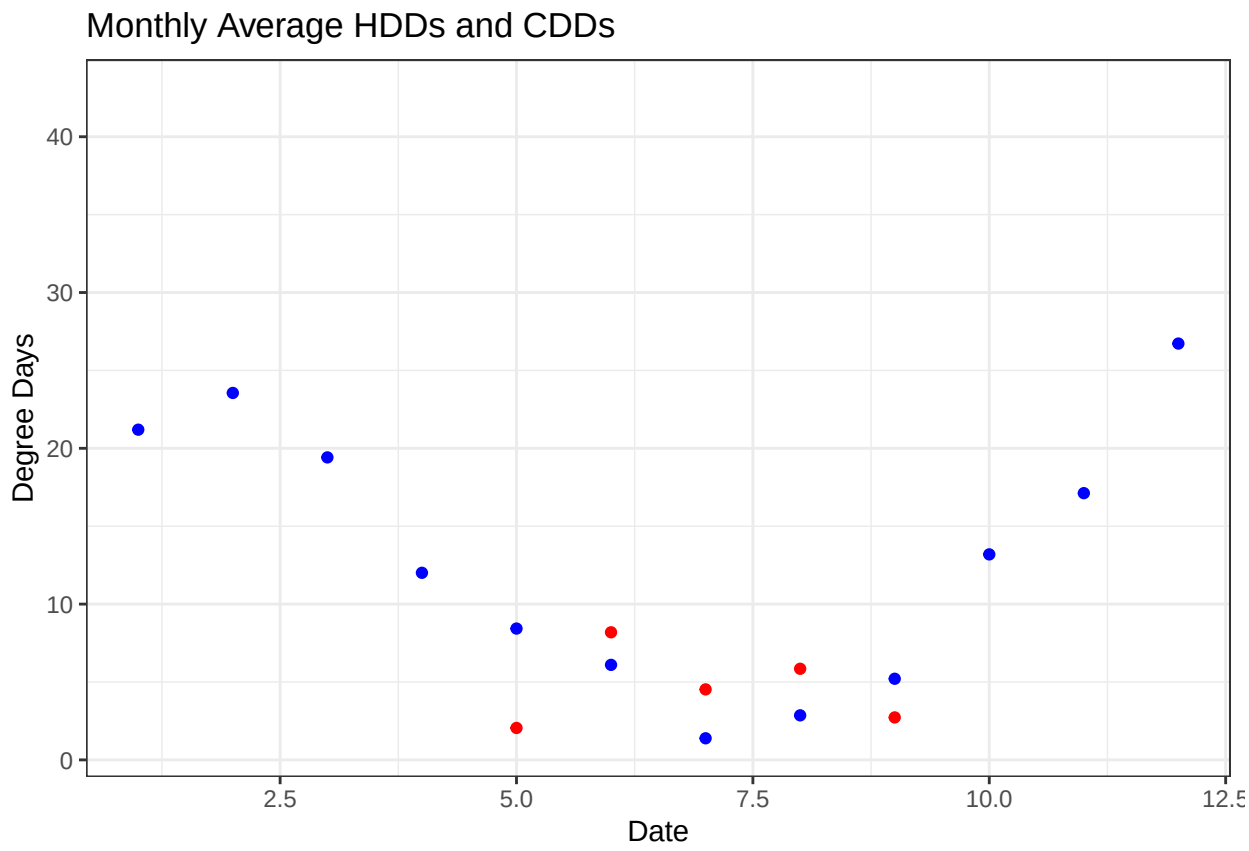

```
# Break DDs out to HDD and CDD and plot both
ggplot(city_2021) +
  geom_point(aes(Date, HDD), color = "blue") +
  geom_point(aes(Date, CDD), color = "red") +
  ylim(1, max(city_2021$HDD)) +
  ggtitle('Daily HDD and CDD during 2021') +
  xlab('Date') +
  ylab('Degree Days') +
  theme_bw()
```

```
## Warning: Removed 77 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 304 rows containing missing values (`geom_point()`).
```




```
# Filter out rows where HDD/CDD are 0, then aggregate by month
city_2021_HDD <- city_2021 %>%
  filter(HDD != 0) %>%
  group_by(month) %>%
  summarise(avg_HDD = mean(HDD))
city_2021_CDD <- city_2021 %>%
  filter(CDD != 0) %>%
  group_by(month) %>%
  summarise(avg_CDD = mean(CDD))

# Plot monthly average HDD and CDD over 2021
ggplot() +
  geom_point(data = city_2021_HDD, mapping = aes(month, avg_HDD), color = "blue") +
  geom_point(data = city_2021_CDD, mapping = aes(month, avg_CDD), color = "red") +
  ylim(1, max(city_2021$HDD)) +
  ggtitle('Monthly Average HDDs and CDDs') +
  xlab('Date') +
  ylab('Degree Days') +
  theme_bw()
```



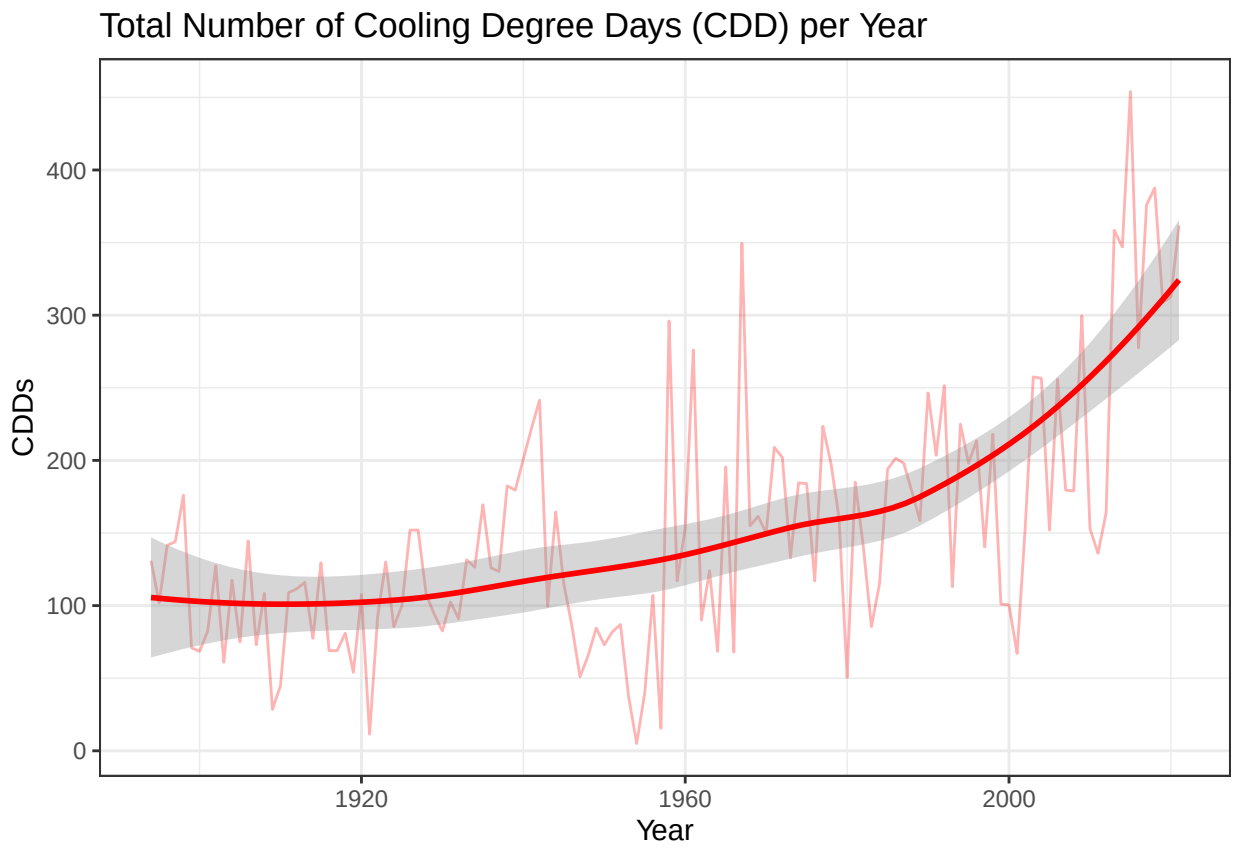
Degree days for the entire dataset

In order to see the trend of how climate change might be effecting energy usage the HDDs and CDDs are aggregated to get yearly totals which are plotted. The plots show the actual data, which have a high variance between years, with a smoothed trend superimposed over top to assess change. For Seattle the number of CDDs has been growing since around 1920, with the rate of increase steepening significantly starting around 1980. The number of HDDs has been decreasing since around 1960 with the rate of decline steepening over time as well. Both of these suggest a warming climate.

```
# Add the DD column
city_df <- city_df %>%
  mutate(DD = tavg - 65)

# Add the CDD and HDD columns
city_df$CDD <- sapply(city_df$DD, calc_CDD)
city_df$HDD <- sapply(city_df$DD, calc_HDD)

# Aggregate yearly averages and sums of HDD and CDD
city_HDD <- city_df %>%
  filter(HDD != 0) %>%
  group_by(year) %>%
  summarise(avg_HDD = mean(HDD),
            sum_HDD = sum(HDD))

city_CDD <- city_df %>%
  filter(CDD != 0) %>%
```

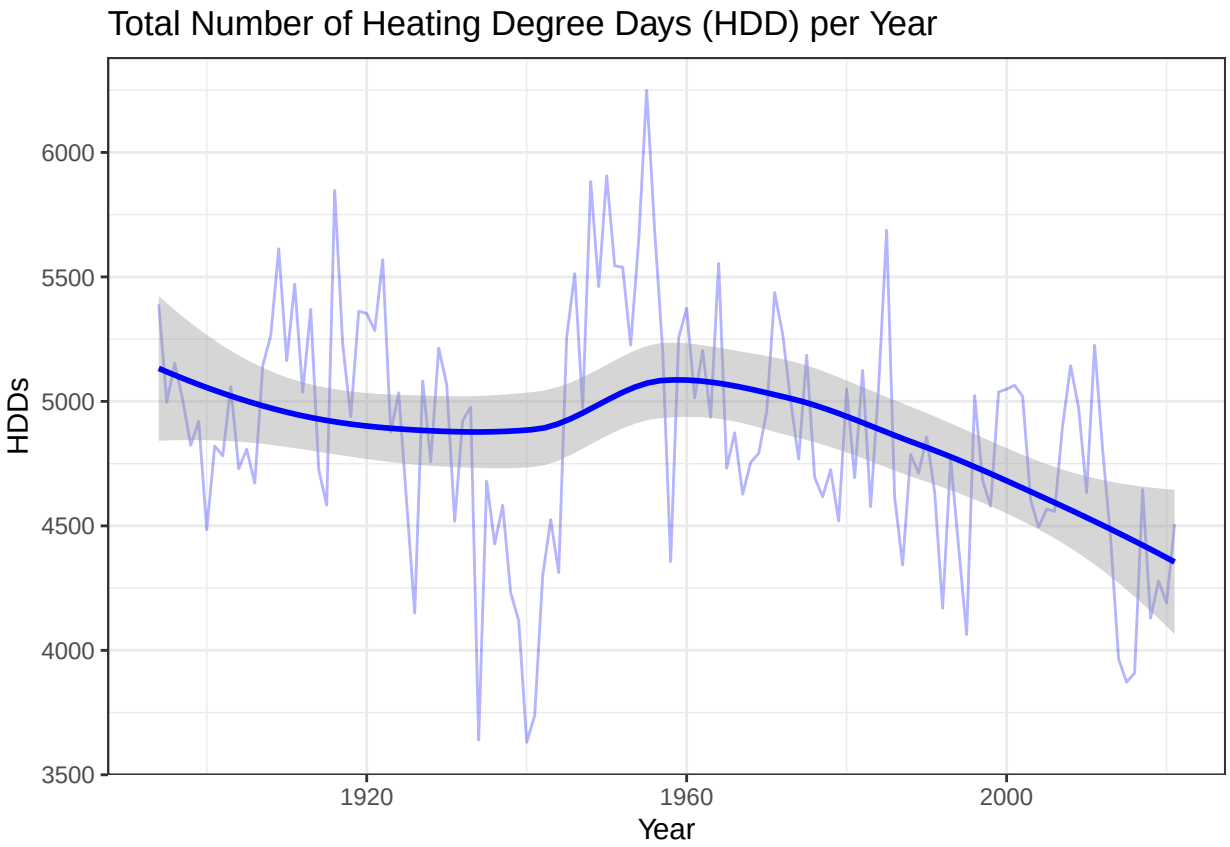
```
  group_by(year) %>%
  summarise(avg_CDD = mean(CDD),
            sum_CDD = sum(CDD))

# Plot the yearly sums of HDD
ggplot() +
  geom_line(data = city_CDD, mapping = aes(year, sum_CDD), color = 'red', alpha = 0.3) +
  geom_smooth(data = city_CDD, mapping = aes(year, sum_CDD), color = 'red') +
  ggtitle("Total Number of Cooling Degree Days (CDD) per Year") +
  ylab("CDDs") +
  xlab("Year") +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
# Plot the yearly sums of CDD
ggplot() +
  geom_line(data = city_HDD, mapping = aes(year, sum_HDD), color = 'blue', alpha = 0.3) +
  geom_smooth(data = city_HDD, mapping = aes(year, sum_HDD), color = 'blue') +
  ggtitle("Total Number of Heating Degree Days (HDD) per Year") +
  ylab("HDDs") +
  xlab("Year") +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Extreme heat and cold days

An alternate measure of the changing climate that is more interpretable are extremely hot and extremely cold days. I have defined these as days where the maximum temperature exceeded the 95th percentile of 'tmax' for the entire dataset and where the minimum temperature was below the 5th percentile of 'tmin'.

As the climate changes it is expected that there will be more climate variability, especially at the extreme ends of the temperature range, and this metric aims to capture that.

Alternatively, this metric could be defined as some threshold over the average high or under the average low. This ideas was initially explored but discarded in preference of the method using percentile thresholds. The code related to this has been left in for legacy purposes.

Build the variables

First build the variables used to find the number of extremely hot/cold days per year. These columns assign a 1 to the respective column if that day exceeded the threshold, and a 0 if not. The data is then aggregated to find the yearly totals of days that were over or under the thresholds.

95% of days in the entire dataset have maximum temperatures under 81 deg. F. Any day with a maximum temperature over this value is an xheat day.

5% of days in the entire dataset have a minimum temperature under 31 deg. F. Any day with a minimum temperature under this value is an xcold day.

```
# Find the average monthly highs/lows for each month
avg_monthly_temps <- city_df %>%
  group_by(month) %>%
```

```
  summarise(avg_high = mean(tmax),
            avg_low = mean(tmin))

# Join the average monthly temperatures to the city climate data
city <- city_df %>% inner_join(avg_monthly_temps, by = 'month')

# Functions to find extreme temperature days using averages
is_xheat_avgs <- function(tmax, avg_high){
  if(tmax > (avg_high + 10))
  {
    return(1)
  }
  else
  {
    return(0)
  }
}
is_xcold_avgs <- function(tmin, avg_low){
  if(tmin < (avg_low - 10))
  {
    return(1)
  }
  else
  {
    return(0)
  }
}

# Create extreme heat/cold day columns based on average
city$xheat_avgs <- mapply(is_xheat_avgs, city$tmax, city$avg_high)
city$xcold_avgs <- mapply(is_xcold_avgs, city$tmin, city$avg_low)


# Define 'extreme temperature' thresholds using percentiles
high_thresh <- quantile(city$tmax, .95)
low_thresh <- quantile(city$tmin, .05)

# Functions to find extreme temperature days using percentiles
is_xheat_perc <- function(tmax, perc_high){
  if(tmax > perc_high)
  {
    return(1)
  }
  else
  {
    return(0)
  }
}
is_xcold_perc <- function(tmin, perc_low){
  if(tmin < perc_low)
  {
    return(1)
  }
```

```
  else
  {
    return(0)
  }
}

# Create extreme heat/cold day coluns based on percentile
city$xheat_perc <- mapply(is_xheat_perc, city$tmax, high_thresh)
city$xcold_perc <- mapply(is_xcold_perc, city$tmin, low_thresh)


# Aggregate the yearly extreme temp days to find totals per year.
xdays_yearly <- city %>%
  group_by(year) %>%
  summarise(sum_xheat_avgs = sum(xheat_avgs),
            sum_xcold_avgs = sum(xcold_avgs),
            sum_xheat_perc = sum(xheat_perc),
            sum_xcold_perc = sum(xcold_perc),
            max_temp = max(tmax))
head(xdays_yearly)
```

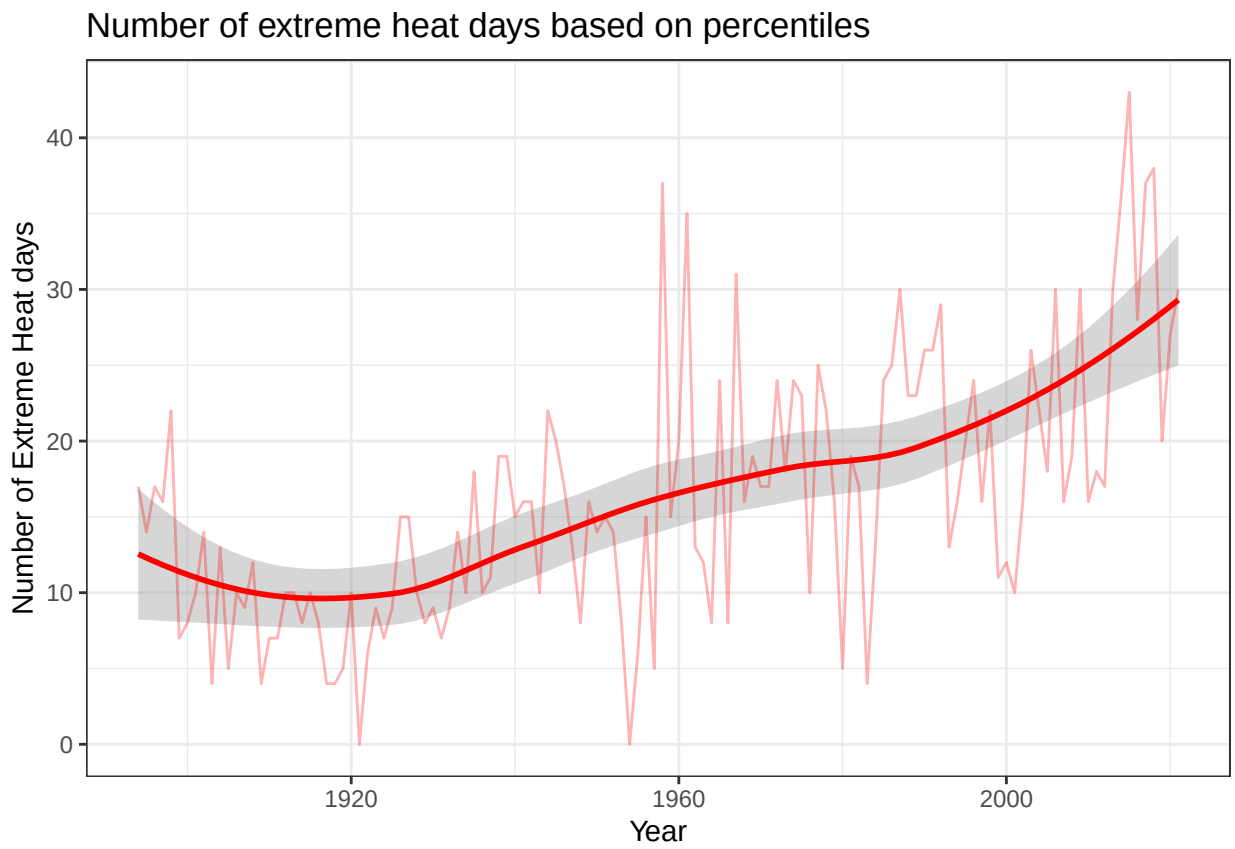
```
## # A tibble: 6 x 6
##    year sum_xheat_avgs sum_xcold_avgs sum_xheat_perc sum_xcold_perc max_temp
##   <dbl>          <dbl>          <dbl>          <dbl>          <dbl>    <dbl>
## 1  1894             15              8             17             23       88
## 2  1895             22              4             14              8       90
## 3  1896             23             13             17             23       93
## 4  1897             23              4             16             13       90
## 5  1898             32              2             22              9       92
## 6  1899             11              9              7             11       90
```

Visualize the yearly extreme heat/cold days

The plots below show similar trends to the Degree Day charts, but they're more interpretable due to the scale of the y-axis. In 2021 there were nearly 30 days over 81 deg. F while in 1920 there weren't even 10.

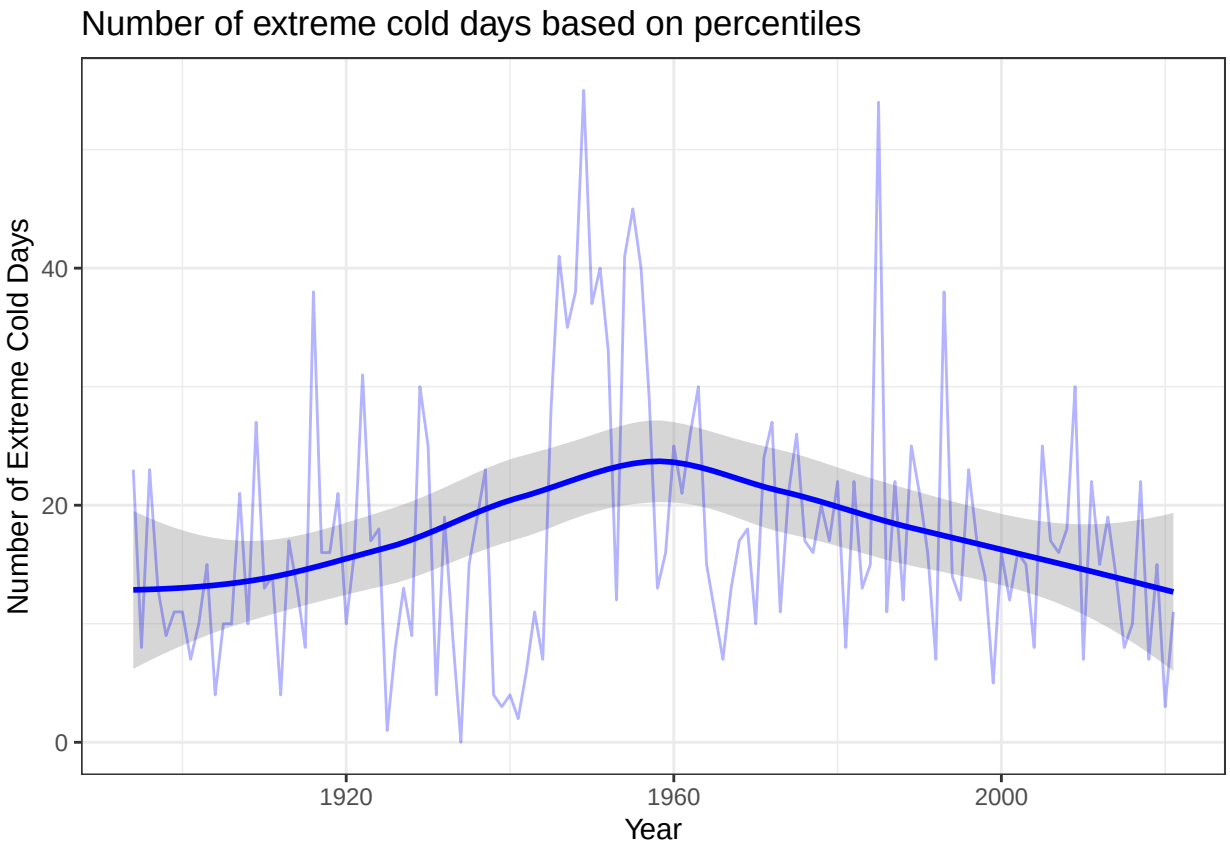
```
# Plot the yearly number of xheat days
ggplot(xdays_yearly) +
  geom_smooth(aes(year, sum_xheat_perc), color = "red") +
  geom_line(aes(year, sum_xheat_perc), color = "red", alpha = 0.3) +
  ggtitle("Number of extreme heat days based on percentiles") +
  ylab("Number of Extreme Heat days") +
  xlab('Year') +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
# Plot the yearly number of xcold days
ggplot(xdays_yearly) +
  geom_smooth(aes(year, sum_xcold_perc), color = "blue") +
  geom_line(aes(year, sum_xcold_perc), color = "blue", alpha = 0.3) +
  ggtitle("Number of extreme cold days based on percentiles") +
  ylab("Number of Extreme Cold Days") +
  xlab('Year') +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



A note on the analysis

While it would be interesting to break the data apart into seasons and analyze it, this analysis was done in support of developing the web app which was designed with simplicity in mind. Including more figures would clutter the interface and seasonality was not relevant to the problem statement.

Forecasting

Goal: Check if SARIMA modeling is appropriate for this task. Does it assume stationarity? If it's appropriate then apply SARIMA forecast to extreme temperature days data. The idea is to predict both physical risk and financial risk (due to energy usage) that is the result of climate change for each city and neighboring Boeing facilities.

Autoregressive integrated moving average (ARIMA) is a forecasting algorithm that utilizes historical time-series data to predict future values. ARIMA takes into account both past values and past forecast errors...

Summary Statistics

Goal: Calculate simple metrics that can be used to quantify how the local climate is changing. This will likely be the percent change between the previous five-year average of extreme temperature days and the previous 50 (?) year average. A similar metric using the forecast model will also be calculated. These statistics will be incorporated into the Shiny app.