

# GlobalClimatePoliciesFinalProject

Michael Grybko

2024-10-05

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.2      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.2      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
```

```
gj_data <- read.csv("~/Desktop/MSDS/ClimateChange/GlobalClimatePolicies/FinalProject/USW00023066.csv")
```

```
fc_data <- read.csv("~/Desktop/MSDS/ClimateChange/GlobalClimatePolicies/FinalProject/USC00053005.csv")
#summary(gj_data)
```

```
# Drop unnecessary columns
```

```
gj_data <- gj_data[-c(1,17:138)]
```

```
# Drop NA entries
```

```
gj_data <- gj_data %>% drop_na()
```

```
# Check for NA entries
```

```
(which(is.na(gj_data)))
```

```
## integer(0)
```

```
# Adjust the temperature values to degrees Celsius from tenths of degrees Celsius
```

```
gj_data$TMAX <- gj_data$TMAX / 10
```

```
gj_data$TMIN <- gj_data$TMIN / 10
```

```
# Adjust the precipitation values to millimeters from tenths of millimeters
```

```
gj_data$PRCP <- gj_data$PRCP / 10
```

```
# Convert the data column to a date format
```

```
gj_data <- gj_data %>%
```

```
  mutate(DATE = ymd(DATE))
```

```
# Filter the data to only include years 1940 through 2023
```

```
gj_data <- gj_data %>%
```

```

filter(year(Date) >= 1940 & year(Date) <= 2023)

summary(gj_data)

##      DATE      LATITUDE      LONGITUDE      ELEVATION
## Min.   :1940-01-01   Min.   :39.13   Min.   :-108.5   Min.   :1470
## 1st Qu.:1961-01-03   1st Qu.:39.13   1st Qu.: -108.5   1st Qu.:1470
## Median :1982-01-02   Median :39.13   Median : -108.5   Median :1470
## Mean   :1982-01-01   Mean   :39.13   Mean   : -108.5   Mean   :1470
## 3rd Qu.:2003-01-01   3rd Qu.:39.13   3rd Qu.: -108.5   3rd Qu.:1470
## Max.   :2023-12-31   Max.   :39.13   Max.   : -108.5   Max.   :1470
##      NAME      PRCP      PRCP_ATTRIBUTES      SNOW
## Length:30677   Min.   : 0.0000   Length:30677   Min.   : 0.000
## Class :character 1st Qu.: 0.0000   Class :character 1st Qu.: 0.000
## Mode  :character Median : 0.0000   Mode  :character Median : 0.000
##                      Mean   : 0.6087                      Mean   : 1.502
##                      3rd Qu.: 0.0000                      3rd Qu.: 0.000
##                      Max.   :47.5000                      Max.   :221.000
## SNOW_ATTRIBUTES  SNWD      SNWD_ATTRIBUTES      TMAX
## Length:30677   Min.   : 0.000   Length:30677   Min.   : -16.10
## Class :character 1st Qu.: 0.000   Class :character 1st Qu.: 8.90
## Mode  :character Median : 0.000   Mode  :character Median : 19.40
##                      Mean   : 6.048                      Mean   : 18.82
##                      3rd Qu.: 0.000                      3rd Qu.: 29.40
##                      Max.   :406.000                      Max.   : 41.70
## TMAX_ATTRIBUTES  TMIN      TMIN_ATTRIBUTES
## Length:30677   Min.   : -30.600   Length:30677
## Class :character 1st Qu.: -2.800   Class :character
## Mode  :character Median : 4.400   Mode  :character
##                      Mean   : 4.626
##                      3rd Qu.: 13.300
##                      Max.   : 25.600

# drop unnecessary columns
fc_data <- fc_data[-c(1,11,12,17:138)]

# Drop NA entries
fc_data <- fc_data %>% drop_na()

# Check for NA entries
(which(is.na(fc_data)))

## integer(0)

# Adjust the temperature values to degrees Celsius from tenths of degrees Celsius
fc_data$TMAX <- fc_data$TMAX / 10
fc_data$TMIN <- fc_data$TMIN / 10

# Adjust the precipitation values to millimeters from tenths of millimeters
fc_data$PRCP <- fc_data$PRCP / 10

# Convert the data column to a date format
fc_data <- fc_data %>%
  mutate(Date = ymd(Date))

```

```
# Filter the data to only include years 1940 through 2023
```

```
fc_data <- fc_data %>%
  filter(year(Date) >= 1940 & year(Date) <= 2023)
```

```
summary(fc_data)
```

```
##      DATE      LATITUDE      LONGITUDE      ELEVATION
## Min.   :1940-01-01   Min.   :40.58   Min.   :-105.1   Min.   :1525
## 1st Qu.:1960-11-20   1st Qu.:40.58   1st Qu.: -105.1   1st Qu.:1525
## Median :1982-01-28   Median :40.58   Median : -105.1   Median :1525
## Mean   :1982-01-19   Mean   :40.58   Mean   : -105.1   Mean   :1525
## 3rd Qu.:2003-01-27   3rd Qu.:40.58   3rd Qu.: -105.1   3rd Qu.:1525
## Max.   :2023-12-31   Max.   :40.58   Max.   : -105.1   Max.   :1525
##      NAME      PRCP      PRCP_ATTRIBUTES      SNOW
## Length:30435   Min.   : 0.000   Length:30435   Min.   : 0.000
## Class :character 1st Qu.: 0.000   Class :character 1st Qu.: 0.000
## Mode  :character Median : 0.000   Mode  :character Median : 0.000
##                      Mean   : 1.071                      Mean   : 3.655
##                      3rd Qu.: 0.000                      3rd Qu.: 0.000
##                      Max.   :117.600                      Max.   :536.000
## SNOW_ATTRIBUTES   TMAX      TMAX_ATTRIBUTES      TMIN
## Length:30435   Min.   :-23.30   Length:30435   Min.   :-40.600
## Class :character 1st Qu.: 10.00   Class :character 1st Qu.: -4.400
## Mode  :character Median : 17.80   Mode  :character Median : 2.200
##                      Mean   : 17.45                      Mean   : 1.958
##                      3rd Qu.: 26.10                      3rd Qu.: 10.000
##                      Max.   : 39.40                      Max.   : 24.400
## TMIN_ATTRIBUTES
## Length:30435
## Class :character
## Mode  :character
##
##
##
```

## Precipitation by Year Grand Junction

There have been four major droughts in Colorado in the 21st century, 2002, 2012, 2018, and 2020 these will be marked with a red asterisks. Years with floods will be marked with a dark blue asterisk.

```
# Sum the PRCP column by year
```

```
gj_yearly_prpc <- gj_data %>%
  mutate(Year = year(Date)) %>%
  group_by(Year) %>%
  summarize(total_prpc = sum(PRPC, na.rm = TRUE), # Total precipitation by year
            ave_prpc = mean(PRPC, na.rm = TRUE)) # Average precipitation by year
```

```
# Plot total precipitation per year
```

```
gj_prpc_year <- ggplot(data = gj_yearly_prpc , aes(x = Year)) +
  geom_line(aes(y = total_prpc, color ="Total Precipitation"), size = 1.5) +
  geom_smooth(aes(y = total_prpc, color ="Smoothed Precipitation"), method ="loess", size =1, linetype=
  # Add a points for drought years with a red asterisk
  geom_point(data = gj_yearly_prpc %>% filter(Year %in% c(2002,2012,2018,2020)),
            aes(x = Year, y = total_prpc),
```

```

        shape = 8, size = 5, color = "red") +
# Add a points for regional floods with a darkblue asterisk
geom_point(data = gj_yearly_prcp %>% filter(Year %in% c(1984,1983,1952,1957)),
          aes(x = Year, y = total_prcp),
            shape = 8, size = 5, color = "darkblue") +
scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
scale_y_continuous(limits=c(100, 400)) +
scale_color_manual(values = c("Total Precipitation" = "green", "Smoothed Precipitation" = "darkgreen"))
labs(title = "Total Precipitation per Year Grand Junction CO",
     x = "Year",
     y = "Total Precipitation in mm") +
theme_minimal() +
theme(plot.title = element_text(size=16, hjust = 0.5, face = "bold"),
      axis.title = element_text(size=12, face = "bold"),
      axis.line = element_line(size = 1, colour = "black"),
      axis.text = element_text(size=10, face="bold"),
      legend.position = c(0.7, 0.98),
      legend.justification = c(0, 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.

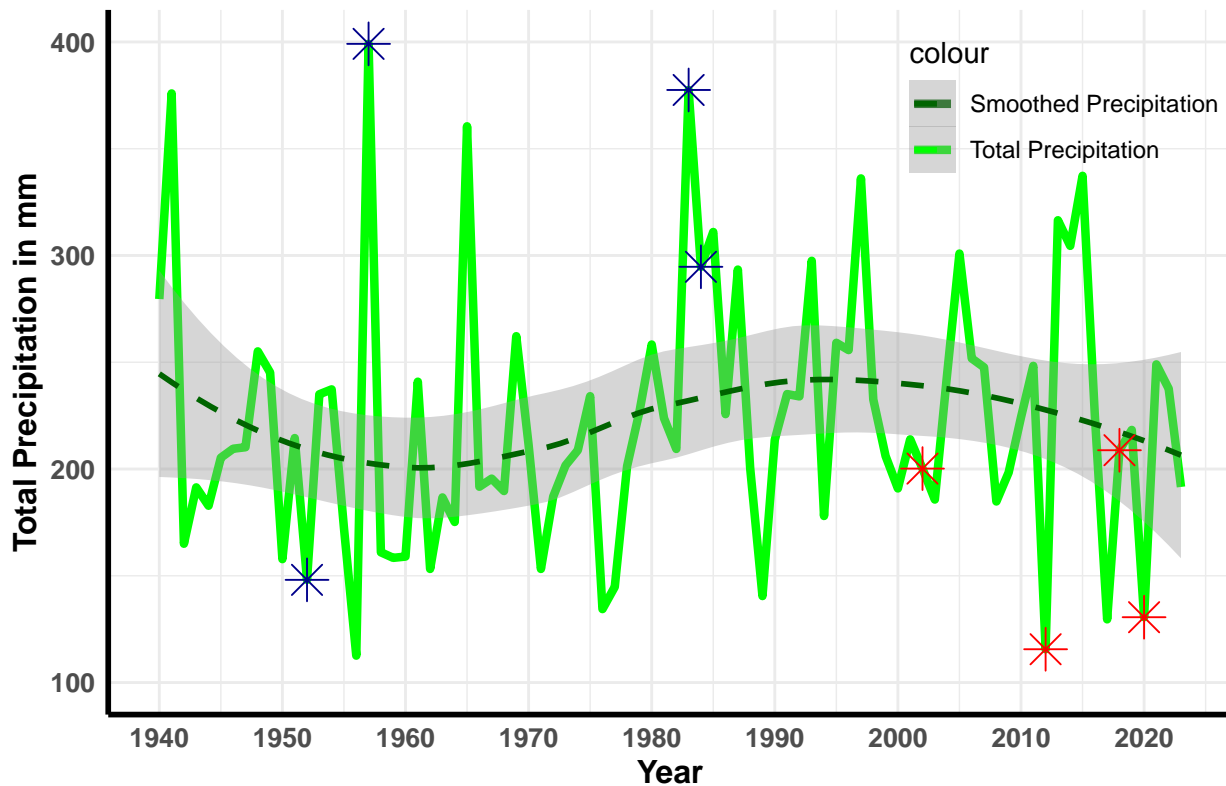
# save plots as jpeg
ggsave("gj_prcp_year.jpeg", width = 10, height = 6, dpi = 300)

## `geom_smooth()` using formula = 'y ~ x'
gj_prcp_year

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

```

## Total Precipitation per Year Grand Junction CO



```
# Sum the PRCP column by year
gj_yearly_snow <- gj_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(total_snow = sum(SNOW, na.rm = TRUE), # Total precipitation by year
            ave_snow = mean(SNOW, na.rm = TRUE)) # Average precipitation by year

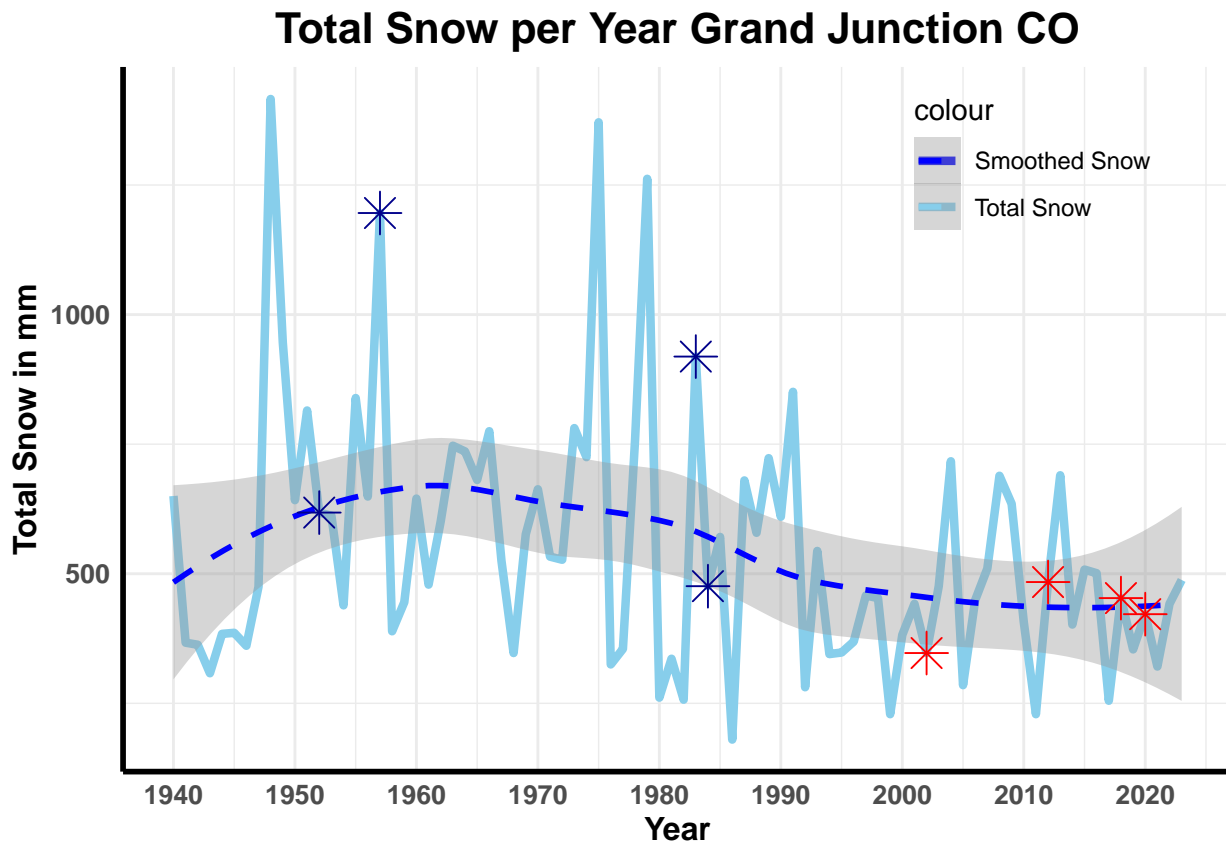
# Plot total snow per year
gj_snow_year <- ggplot(data = gj_yearly_snow, aes(x = Year)) +
  geom_line(aes(y = total_snow, color = "Total Snow"), size = 1.5) +
  geom_smooth(aes(y = total_snow, color = "Smoothed Snow"), method = "loess", size = 1, linetype = 2) +
  # Add a points for drought years with a red asterisk
  geom_point(data = gj_yearly_snow %>% filter(Year %in% c(2002, 2012, 2018, 2020)),
            aes(x = Year, y = total_snow),
            shape = 8, size = 5, color = "red") +
  # Add a points for regional floods with a darkblue asterisk
  geom_point(data = gj_yearly_snow %>% filter(Year %in% c(1984, 1983, 1952, 1957)),
            aes(x = Year, y = total_snow),
            shape = 8, size = 5, color = "darkblue") +
  scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
  scale_color_manual(values = c("Total Snow" = "skyblue", "Smoothed Snow" = "blue")) +
  labs(title = "Total Snow per Year Grand Junction CO",
       x = "Year",
       y = "Total Snow in mm") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, hjust = 0.5, face = "bold"),
        axis.title = element_text(size = 12, face = "bold"),
```

```
axis.line = element_line(size = 1, colour = "black"),
axis.text = element_text(size=10, face="bold"),
legend.position = c(0.7, 0.98),
legend.justification = c(0, 1))

# save plots as jpegs
ggsave("gj_snow_year.jpeg", width = 10, height = 6, dpi = 300)

## `geom_smooth()` using formula = 'y ~ x'
gj_snow_year

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



## Temperature Trends for Grand Junction

The warmest year to date in Colorado is 2012, which will be marked with an asterisk. To examine trends by decade the deviation from the mean maximum or minimum temperature were plotted.

```
gj_yearly_max_temp <- gj_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(ave_max_temp = mean(TMAX, na.rm = T))

# plot average max temp by year with smoothed line
gj_max_temp_year <- ggplot(data = gj_yearly_max_temp , aes(x = Year)) +
  geom_line(aes(y = ave_max_temp, color ="Average Max Temp"), size = 1.5) +
```

```

geom_smooth(aes(y = ave_max_temp, color = "Smoothed Temp"), method = "loess", size = 1, linetype=2) +
# Add a point for 2012 with an asterisk
geom_point(data = gj_yearly_max_temp %>% filter(Year == 2012),
  aes(x = Year, y = ave_max_temp),
  shape = 8, size = 5, color = "black") + # Asterisk for 2012
scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
scale_y_continuous(limits=c(16, 21)) +
scale_color_manual(values = c("red", "darkred"),
  labels = c("Average Max Temp", "Smoothed Max Temp")) +
labs(title = "Average Max Temperature by Year Grand Junction CO",
  x = "Year",
  y = "Max Temp °C") +
theme_minimal() +
theme(plot.title = element_text(size=16, hjust = 0.5, face = "bold"),
  axis.title = element_text(size=12, face = "bold"),
  axis.line = element_line(size = 1, colour = "black"),
  axis.text = element_text(size=10, face="bold"),
  legend.position = c(0.025, 0.95),
  legend.justification = c(0, 1))

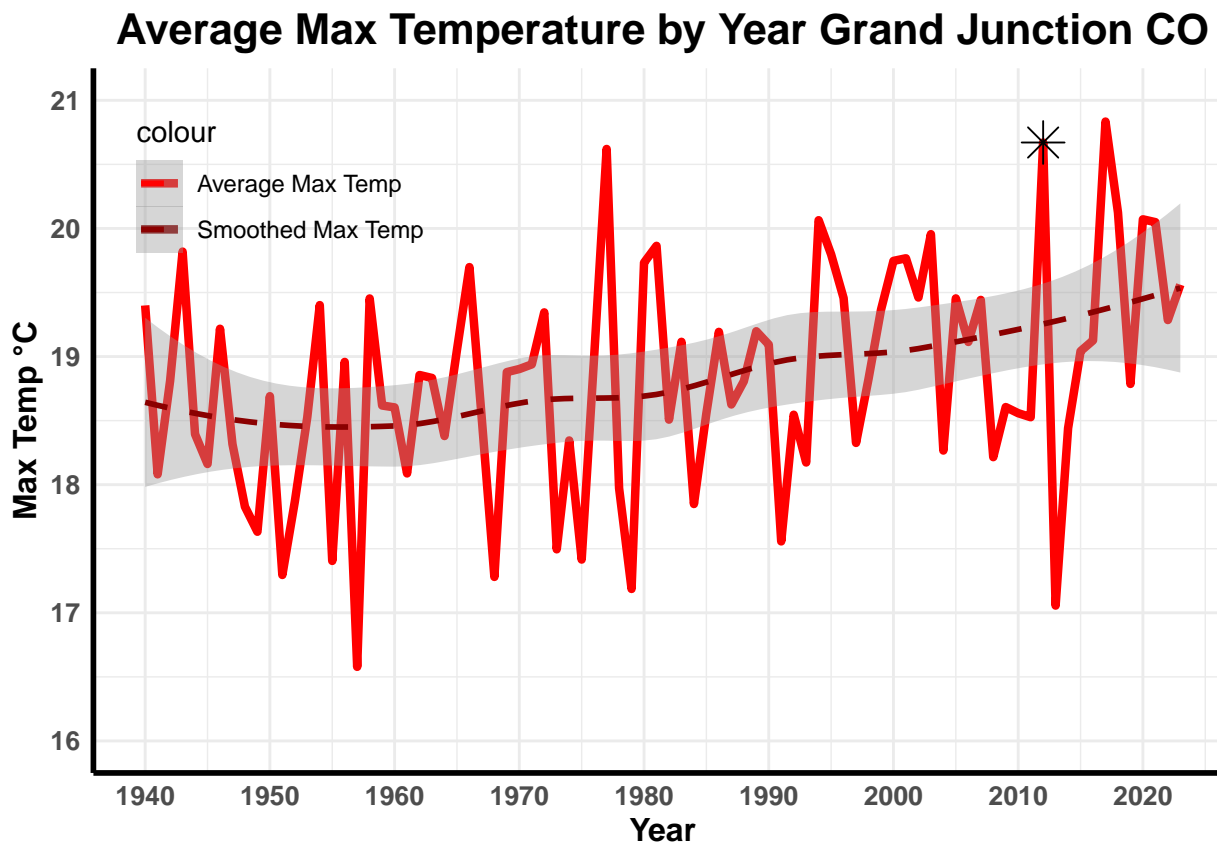
ggsave("gj_max_temp_year.jpeg", width = 10, height = 6, dpi = 300)

```

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

```
gj_max_temp_year
```

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



```

# Calculate the mean of TMAX
gj_ave_max_temp <- mean(gj_data$TMAX)
# Group by decade and calculate the difference from mean
gj_decade_max_temp <- gj_data %>%
  mutate(Year = year(DATE),
         Decade = (Year %/% 10) * 10) %>%
  filter(year(DATE) < 2020) %>%
  group_by(Decade) %>%
  summarize(
    diff_ave_max_temp = (mean(TMAX, na.rm = TRUE) - gj_ave_max_temp),
    sd_max_temp = sd(TMAX, na.rm = TRUE), # Standard deviation of max temperature
    count = n() # Number of observations per decade
  )
# Compute standard error
gj_decade_max_temp <- gj_decade_max_temp %>%
  mutate(se_max_temp = sd_max_temp / sqrt(count))

# Categorize each decade as positive or negative
gj_decade_max_temp <- gj_decade_max_temp %>%
  mutate(temp_category = ifelse(diff_ave_max_temp >= 0, "Positive", "Negative"))

# Create the bar plot with color gradient based on the value of diff_ave_max_temp
gj_max_temp_decade <- ggplot(data = gj_decade_max_temp,
                             aes(x = as.factor(Decade),
                                y = diff_ave_max_temp,
                                fill = diff_ave_max_temp)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = diff_ave_max_temp - se_max_temp,
                  ymax = diff_ave_max_temp + se_max_temp),
               width = 0.2, color = "black") +
  scale_fill_gradient2(low = "skyblue1", mid = "white", high = "firebrick2",
                      midpoint = 0,
                      name = "Temp Deviation") +
  labs(title = "Changes in Average Max Temperature by Decade in Grand Junction",
       x = "Decade",
       y = "Deviation from Average Max Temperature (°C)") +
  theme_minimal() +
  theme(plot.title = element_text(size=14, hjust = 0.5, face = "bold"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size = 10, face = "bold"),
        axis.line = element_line(size = 1, colour = "black"))

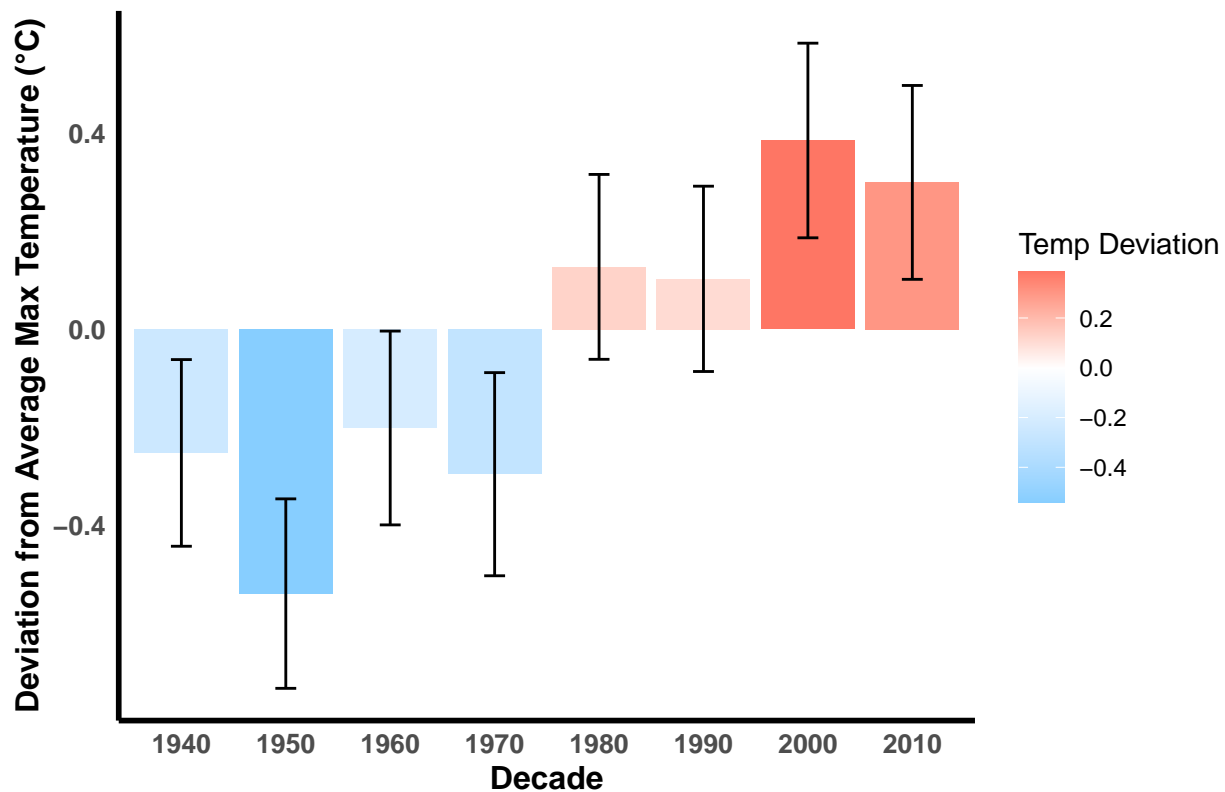
# Save the updated plot as a JPEG file
ggsave("gj_max_temp_decade_anomalies_scaled.jpeg", plot = gj_max_temp_decade, width = 10, height = 6, dpi = 300)

# Display the plot
gj_max_temp_decade

```



## anges in Average Max Temperature by Decade in Grand Junction



```

gj_yearly_min_temp <- gj_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(ave_min_temp = mean(TMIN, na.rm = TRUE))

gj_min_temp_year <- ggplot(data = gj_yearly_min_temp , aes(x = Year)) +
  geom_line(aes(y = ave_min_temp, color = "Average Min Temp"), size = 1.5) +
  geom_smooth(aes(y = ave_min_temp, color = "Smoothed Temp"), method = "loess", size = 1, linetype=2) +
  # Add a point for 2012 with an asterisk
  geom_point(data = gj_yearly_min_temp %>% filter(Year == 2012),
    aes(x = Year, y = ave_min_temp),
    shape = 8, size = 5, color = "black") + # Asterisk for 2012
  scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
  scale_color_manual(values = c("blue", "navyblue"),
    labels = c("Average Min Temp", "Smoothed Min Temp")) +
  labs(title = "Average Minimum Temperature by Year Grand Junction CO",
    x = "Year",
    y = "Average Min Temp °C") +
  theme_minimal() +
  theme(plot.title = element_text(size=16, hjust = 0.5, face = "bold"),
    axis.title = element_text(size=12, face = "bold"),
    axis.line = element_line(size = 1, colour = "black"),
    axis.text = element_text(size=10, face="bold"),
    legend.position = c(0.75, 0.95),
    legend.justification = c(0, 1))

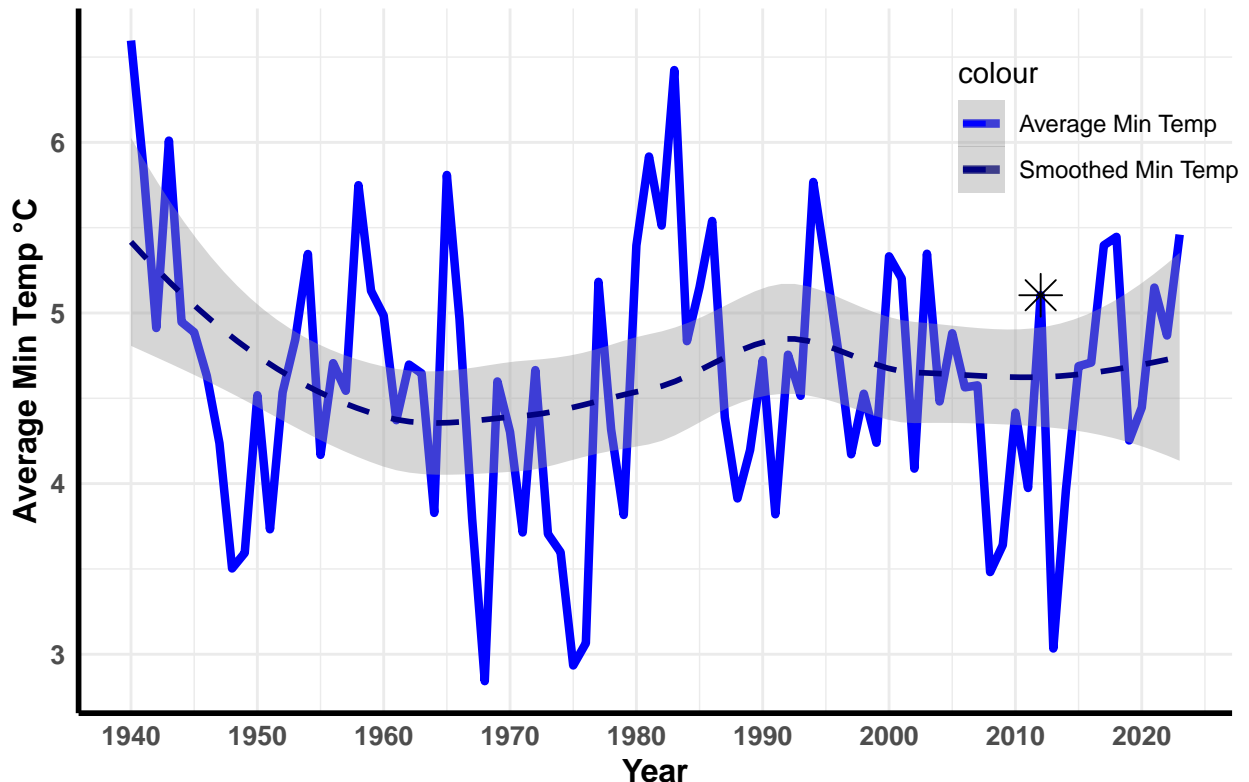
ggsave("gj_min_temp_year.jpeg", width = 10, height = 6, dpi = 300)

```

```
## `geom_smooth()` using formula = 'y ~ x'
gj_min_temp_year
```

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

## Average Minimum Temperature by Year Grand Junction CO



```
# Calculate the mean of TMIN
gj_ave_min_temp <- mean(gj_data$TMIN)
# Group by decade and calculate the difference from mean
gj_decade_min_temp <- gj_data %>%
  mutate(Year = year(DATE),
         Decade = (Year %/% 10) * 10) %>%
  filter(year(DATE) < 2020) %>%
  group_by(Decade) %>%
  summarize(
    diff_ave_min_temp = (mean(TMIN, na.rm = TRUE) - gj_ave_min_temp),
    sd_min_temp = sd(TMIN, na.rm = TRUE), # Standard deviation of max temperature
    count = n() # Number of observations per decade
  )
# Compute standard error
gj_decade_min_temp <- gj_decade_min_temp %>%
  mutate(se_min_temp = sd_min_temp / sqrt(count))

# Categorize each decade as positive or negative
gj_decade_min_temp <- gj_decade_min_temp %>%
  mutate(temp_category = ifelse(diff_ave_min_temp >= 0, "Positive", "Negative"))

# Create the bar plot with color gradient based on the value of diff_ave_max_temp
```

```

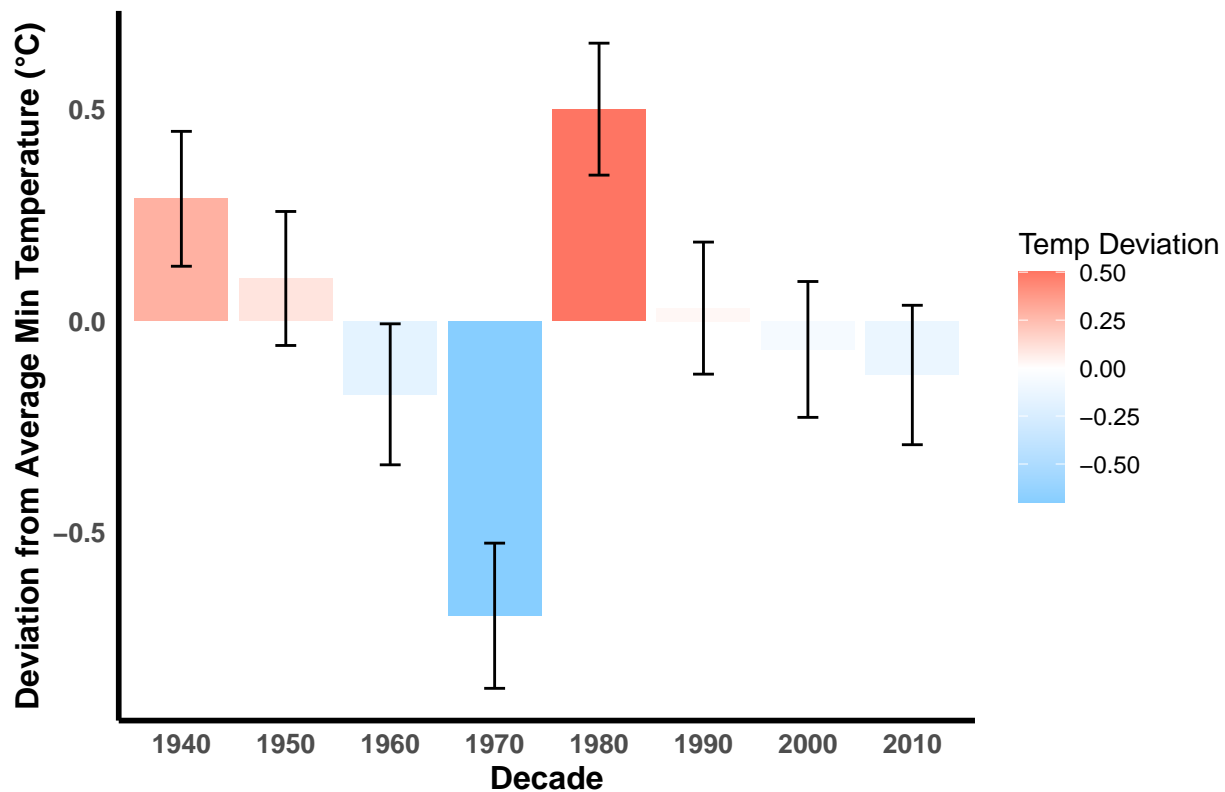
gj_min_temp_decade <- ggplot(data = gj_decade_min_temp,
                             aes(x = as.factor(Decade),
                                y = diff_ave_min_temp,
                                fill = diff_ave_min_temp)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = diff_ave_min_temp - se_min_temp,
                   ymax = diff_ave_min_temp + se_min_temp),
               width = 0.2, color = "black") +
  scale_fill_gradient2(low = "skyblue1", mid = "white", high = "firebrick2",
                      midpoint = 0,
                      name = "Temp Deviation") +
  labs(title = "Changes in Average Min Temperature by Decade in Grand Junction",
       x = "Decade",
       y = "Deviation from Average Min Temperature (°C)") +
  theme_minimal() +
  theme(plot.title = element_text(size=14, hjust = 0.5, face = "bold"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size = 10, face = "bold"),
        axis.line = element_line(size = 1, colour = "black")
  )

# Save the bar plot as a JPEG file
ggsave("gj_min_temp_decade.jpeg", width = 10, height = 6, dpi = 300)

(gj_min_temp_decade)

```

## anges in Average Min Temperature by Decade in Grand Junction



Precipitation by Year For Fort Collins There have been four major droughts in Colorado in the 21st century, 2002, 2012, 2018, and 2020 these will be marked with a red asterisks. Years with floods will be marked with a dark blue asterisk.

```
# Sum the PRCP column by year
fc_yearly_prpc <- fc_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(total_prpc = sum(PRCP, na.rm = TRUE), # Total precipitation by year
            ave_prpc = mean(PRCP, na.rm = TRUE)) # Average precipitation by year

# Plot total precipitation per year
fc_prpc_year <- ggplot(data = fc_yearly_prpc, aes(x = Year)) +
  geom_line(aes(y = total_prpc, color = "Total Precipitation"), size = 1.5) +
  geom_smooth(aes(y = total_prpc, color = "Smoothed Precipitation"), method = "loess", size = 1, linetype = "dashed")
# Add a points with an asterisk
geom_point(data = fc_yearly_prpc %>% filter(Year %in% c(2002, 2012, 2018, 2020)),
           aes(x = Year, y = total_prpc),
           shape = 8, size = 5, color = "red") +
scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
# Add a points for regional floods with a darkblue asterisk
geom_point(data = fc_yearly_prpc %>% filter(Year %in% c(2013, 1997)),
           aes(x = Year, y = total_prpc),
           shape = 8, size = 5, color = "darkblue") +
scale_y_continuous(limits = c(100, 400)) +
scale_color_manual(values = c("Total Precipitation" = "green", "Smoothed Precipitation" = "darkgreen"))
labs(title = "Total Precipitation per Year Fort Collins CO",
     x = "Year",
```

```

y = "Total Precipitation in mm") +
theme_minimal() +
theme(plot.title = element_text(size=16, hjust = 0.5, face = "bold"),
      axis.title = element_text(size=12, face = "bold"),
      axis.line = element_line(size = 1, colour = "black"),
      axis.text = element_text(size=10, face="bold"),
      legend.position = c(0.7, 0.98),
      legend.justification = c(0, 1))

```

```

# save plot as jpeg
ggsave("fc_prctp_year.jpeg", width = 10, height = 6, dpi = 300)

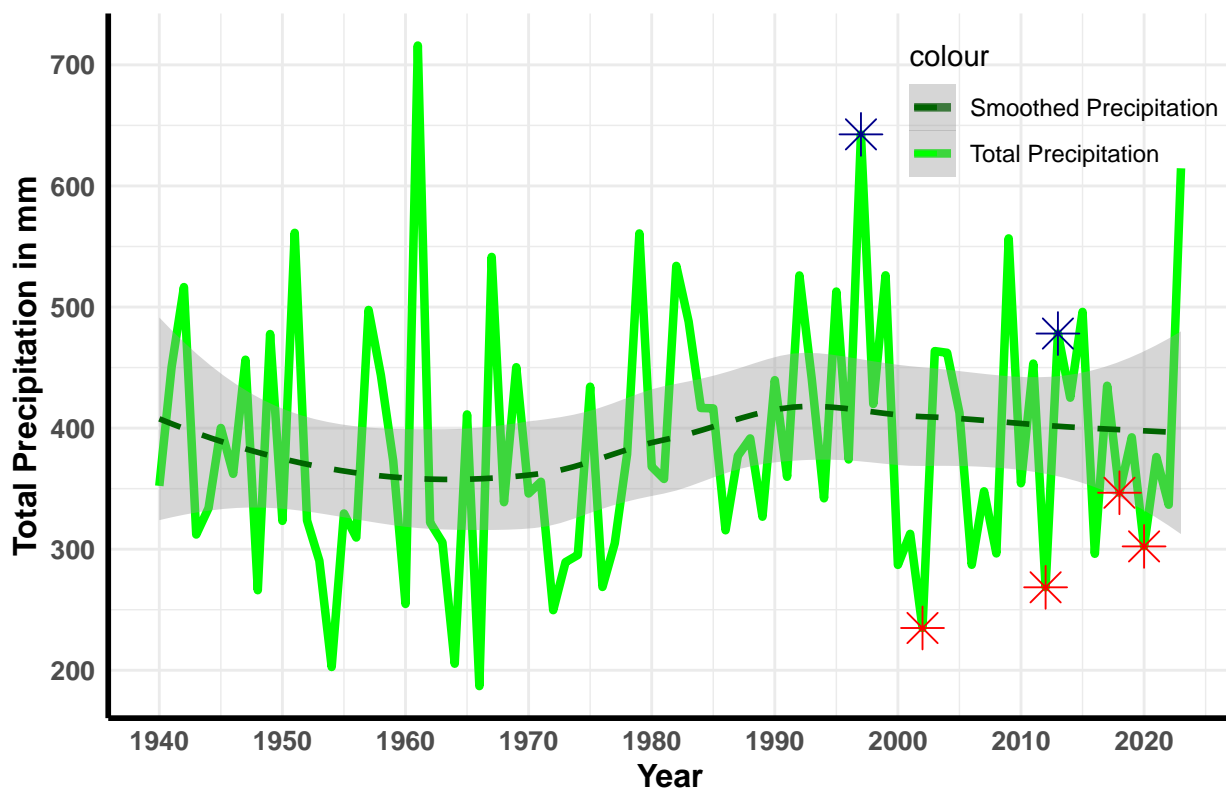
```

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

```
fc_prctp_year
```

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

## Total Precipitation per Year Fort Collins CO



```

# Sum the PRCP column by year
fc_yearly_snow <- fc_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(total_snow = sum(SNOW, na.rm = TRUE), # Total precipitation by year
            ave_snow = mean(SNOW, na.rm = TRUE)) # Average precipitation by year

# Plot total precipitation per year
fc_snow_year <- ggplot(data = fc_yearly_snow, aes(x = Year)) +
  geom_line(aes(y = total_snow, color = "Total Snow"), size = 1.5) +

```

```

geom_smooth(aes(y = total_snow, color = "Smoothed Snow"), method = "loess", size = 1, linetype = 2) +
# Add a points with an asterisk
geom_point(data = fc_yearly_snow %>% filter(Year %in% c(2002, 2012, 2018, 2020)),
  aes(x = Year, y = total_snow),
  shape = 8, size = 5, color = "red") +
# Add a points for regional floods with a darkblue asterisk
geom_point(data = fc_yearly_snow %>% filter(Year %in% c(2013, 1997)),
  aes(x = Year, y = total_snow),
  shape = 8, size = 5, color = "darkblue") +
scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
scale_color_manual(values = c("Total Snow" = "skyblue", "Smoothed Snow" = "blue")) +
labs(title = "Total Snow per Year Fort Collins CO",
  x = "Year",
  y = "Total Snow in mm") +
theme_minimal() +
theme(plot.title = element_text(size = 16, hjust = 0.5, face = "bold"),
  axis.title = element_text(size = 12, face = "bold"),
  axis.line = element_line(size = 1, colour = "black"),
  axis.text = element_text(size = 10, face = "bold"),
  legend.position = c(0.7, 0.98),
  legend.justification = c(0, 1))

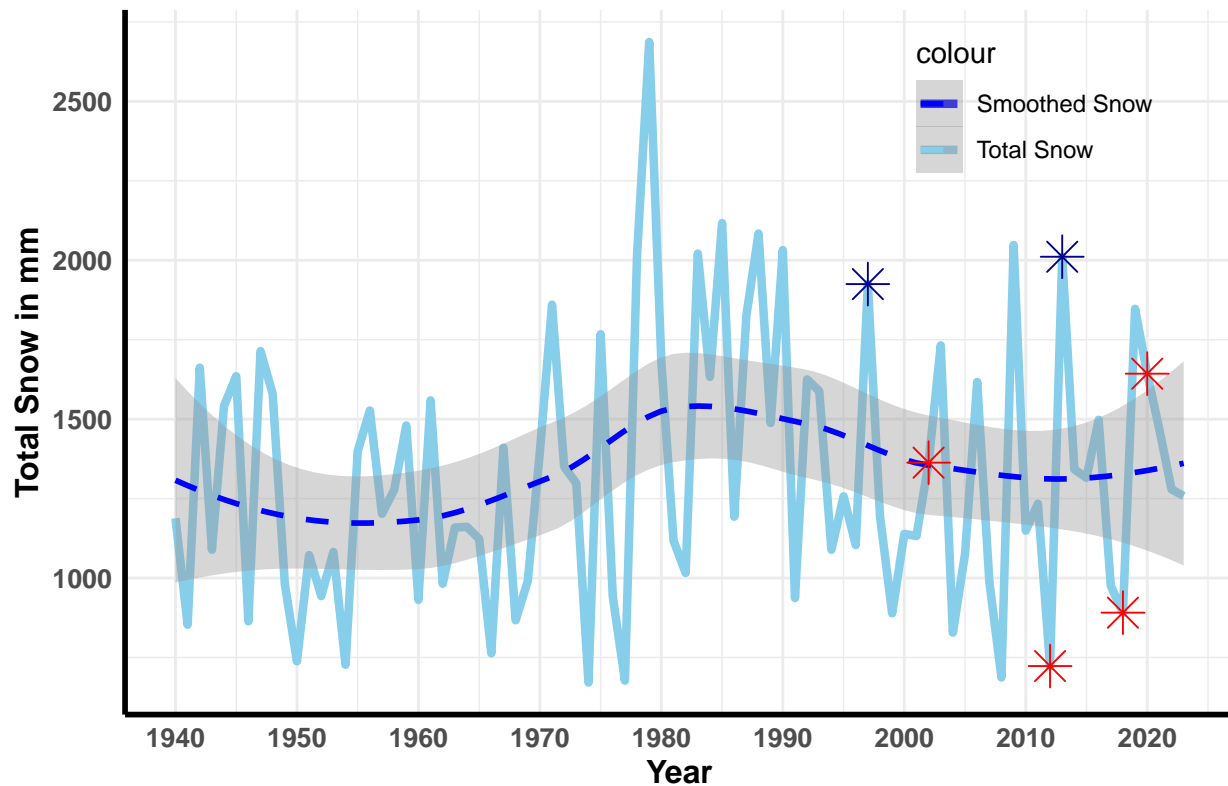
# save plots as jpegs
ggsave("fc_snow_year.jpeg", width = 10, height = 6, dpi = 300)

## `geom_smooth()` using formula = 'y ~ x'
fc_snow_year

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

```

## Total Snow per Year Fort Collins CO



## Temperature Trends for Fort Collins

The warmest year to date in Colorado is 2012, which will be marked with an asterisk. To examine trends by decade the deviation from the mean maximum or minimum temperature were plotted.

```
fc_yearly_max_temp <- fc_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(ave_max_temp = mean(TMAX, na.rm = T))

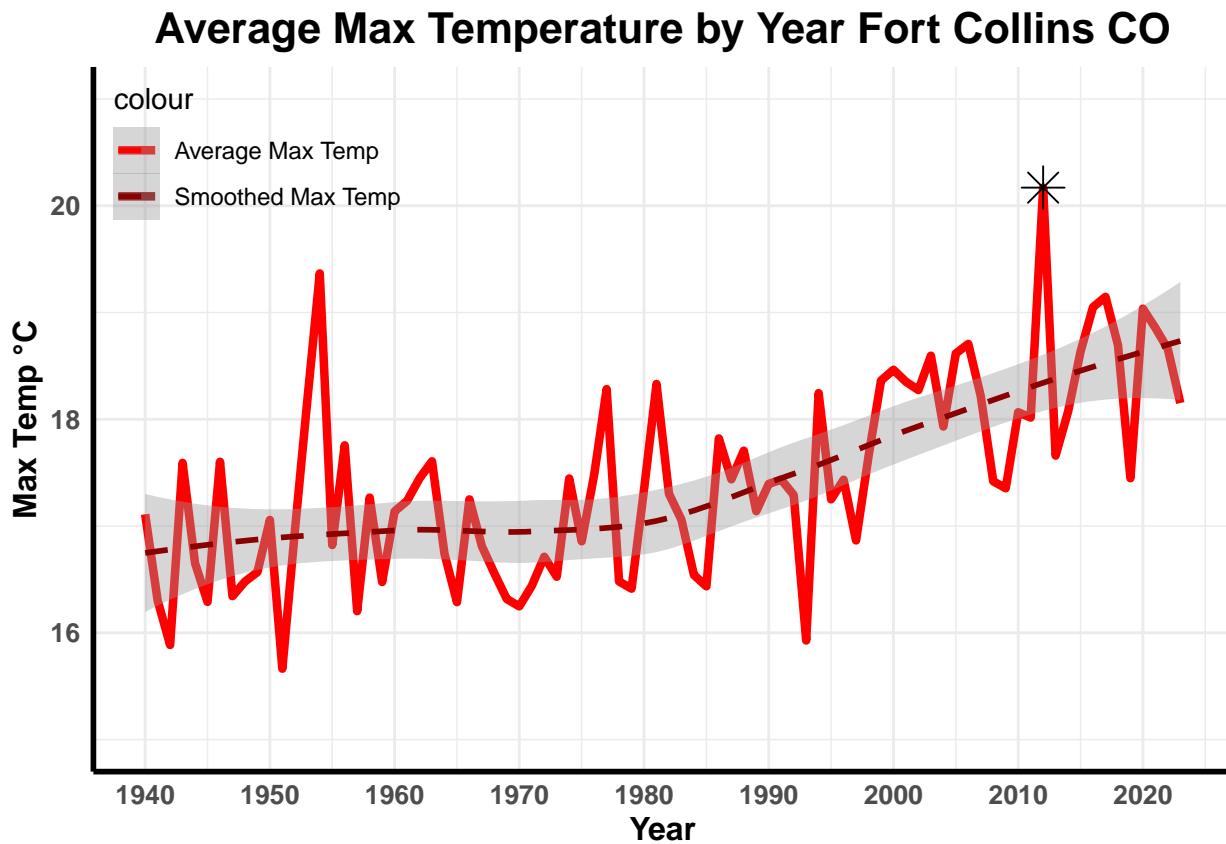
# plot average max temp by year with smoothed line
fc_max_temp_year <- ggplot(data = fc_yearly_max_temp , aes(x = Year)) +
  geom_line(aes(y = ave_max_temp, color = "Average Max Temp"), size = 1.5) +
  geom_smooth(aes(y = ave_max_temp, color = "Smoothed Temp"), method = "loess", size = 1, linetype=2) +
  # Add a point for 2012 with an asterisk
  geom_point(data = fc_yearly_max_temp %>% filter(Year == 2012),
             aes(x = Year, y = ave_max_temp),
             shape = 8, size = 5, color = "black") + # Asterisk for 2012
  scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
  scale_y_continuous(limits=c(15, 21)) +
  scale_color_manual(values = c("red", "darkred"),
                     labels = c("Average Max Temp", "Smoothed Max Temp")) +
  labs(title = "Average Max Temperature by Year Fort Collins CO",
       x = "Year",
       y = "Max Temp °C") +
  theme_minimal() +
  theme(plot.title = element_text(size=16, hjust = 0.5, face = "bold"),
```

```
axis.title = element_text(size=12, face = "bold"),
axis.line = element_line(size = 1, colour = "black"),
axis.text = element_text(size=10, face="bold"),
legend.position = c(0.005, 0.995),
legend.justification = c(0, 1))

ggsave("fc_max_temp_year.jpeg", width = 10, height = 6, dpi = 300)
```

```
## `geom_smooth()` using formula = 'y ~ x'
fc_max_temp_year
```

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



```
# Calculate the mean of TMAX
fc_ave_max_temp <- mean(fc_data$TMAX)
# Group by decade and calculate the difference from mean
fc_decade_max_temp <- fc_data %>%
  mutate(Year = year(DATE),
         Decade = (Year %/% 10) * 10) %>%
  filter(year(DATE) < 2020) %>%
  group_by(Decade) %>%
  summarize(
    diff_ave_max_temp = (mean(TMAX, na.rm = TRUE) - fc_ave_max_temp),
    sd_max_temp = sd(TMAX, na.rm = TRUE), # Standard deviation of max temperature
    count = n() # Number of observations per decade
  )
# Compute standard error
```



```

fc_decade_max_temp<- fc_decade_max_temp %>%
  mutate(se_max_temp = sd_max_temp / sqrt(count))

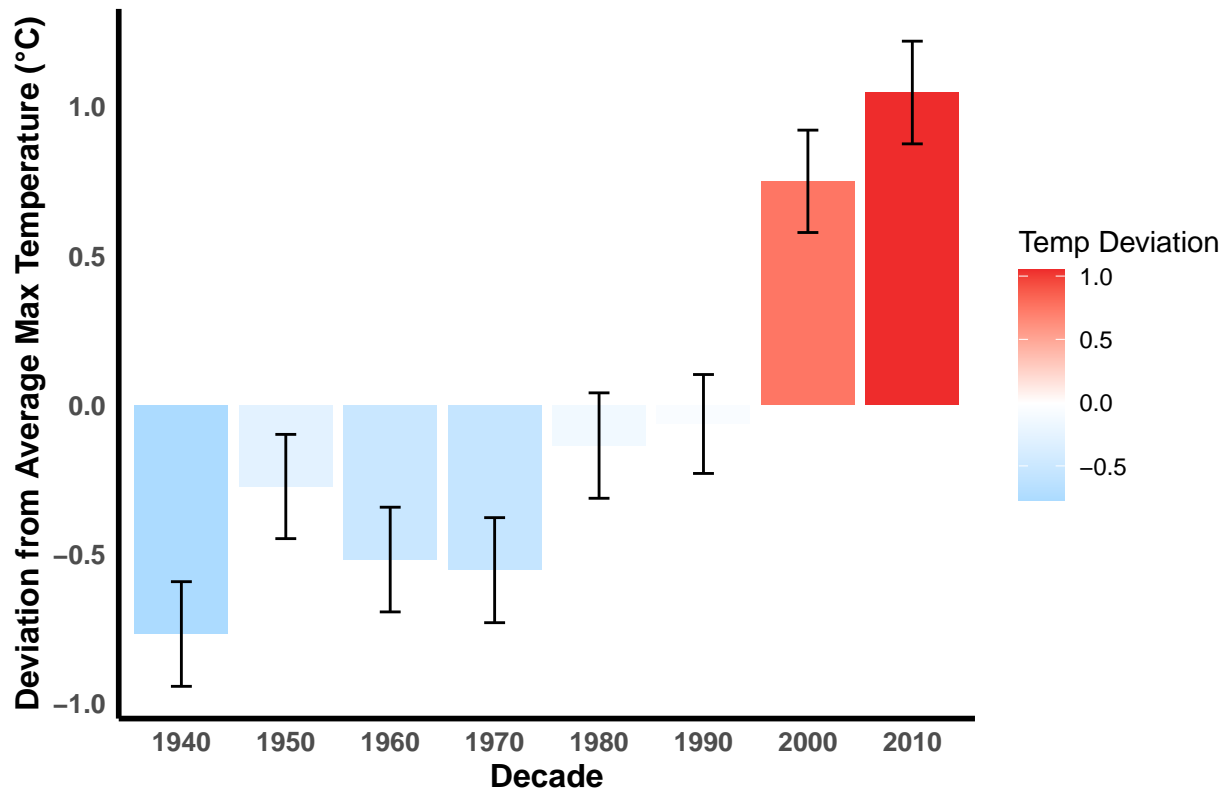
# Categorize each decade as positive or negative
fc_decade_max_temp <- fc_decade_max_temp %>%
  mutate(temp_category = ifelse(diff_ave_max_temp >= 0, "Positive", "Negative"))

# Create the bar plot with color gradient based on the value of diff_ave_max_temp
fc_max_temp_decade <- ggplot(data = fc_decade_max_temp,
                             aes(x = as.factor(Decade),
                                 y = diff_ave_max_temp,
                                 fill = diff_ave_max_temp)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = diff_ave_max_temp - se_max_temp,
                   ymax = diff_ave_max_temp + se_max_temp),
               width = 0.2, color = "black") +
  scale_fill_gradient2(low = "skyblue1", mid = "white", high = "firebrick2",
                      midpoint = 0,
                      name = "Temp Deviation") +
  labs(title = "Changes in Average Max Temperature by Decade in Fort Collins",
       x = "Decade",
       y = "Deviation from Average Max Temperature (°C)") +
  theme_minimal() +
  theme(plot.title = element_text(size=14, hjust = 0.5, face = "bold"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size = 10, face = "bold"),
        axis.line = element_line(size = 1, colour = "black")
  )
# Save the bar plot as a JPEG file
ggsave("fc_max_temp_decade.jpeg",width = 10, height = 6, dpi = 300)

(fc_max_temp_decade)

```

## Changes in Average Max Temperature by Decade in Fort Collins



```
fc_yearly_min_temp <- fc_data %>%
  mutate(Year = year(DATE)) %>%
  group_by(Year) %>%
  summarize(ave_min_temp = mean(TMIN, na.rm = TRUE))

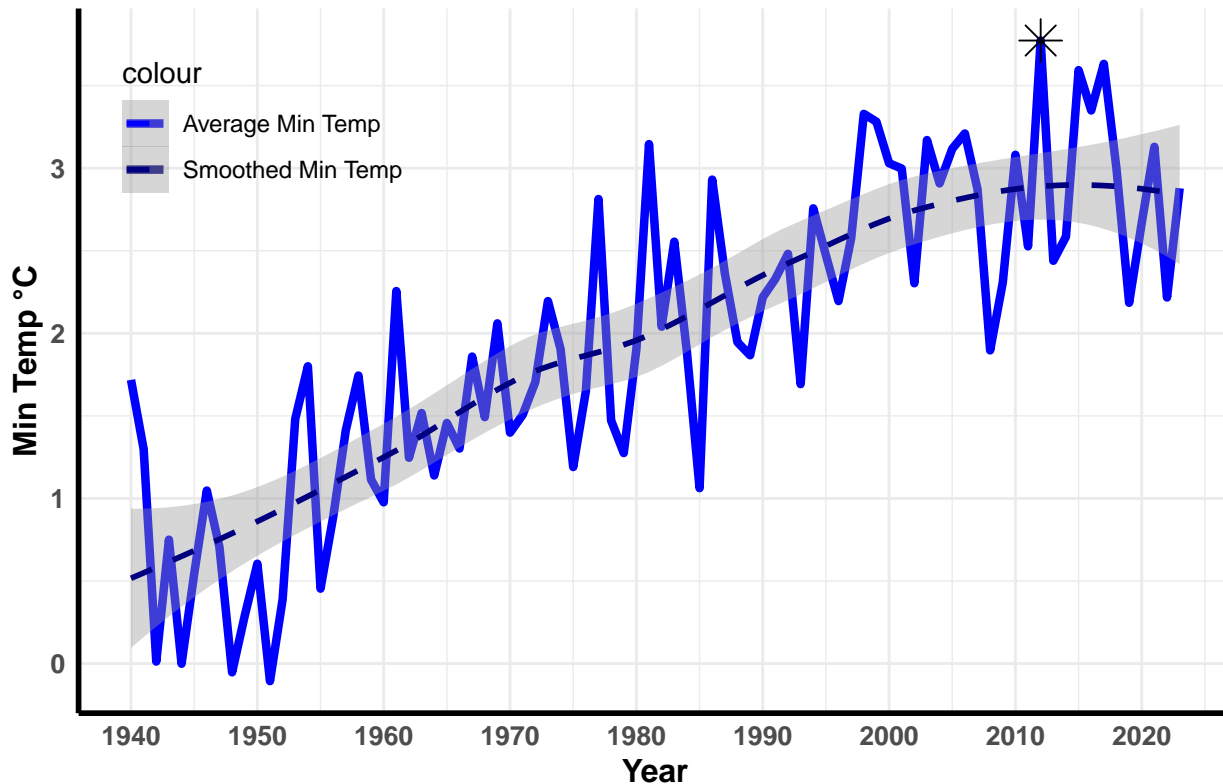
fc_min_temp_year <- ggplot(data = fc_yearly_min_temp , aes(x = Year)) +
  geom_line(aes(y = ave_min_temp, color = "Average Min Temp"), size = 1.5) +
  geom_smooth(aes(y = ave_min_temp, color = "Smoothed Temp"), method = "loess", size = 1, linetype=2) +
  # Add a point for 2012 with an asterisk
  geom_point(data = fc_yearly_min_temp %>% filter(Year == 2012),
             aes(x = Year, y = ave_min_temp),
             shape = 8, size = 5, color = "black") + # Asterisk for 2012
  scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
  scale_color_manual(values = c("blue", "navyblue"),
                    labels = c("Average Min Temp", "Smoothed Min Temp")) +
  labs(title = "Average Minimum Temperature by Year Fort Collins CO",
       x = "Year",
       y = "Min Temp °C") +
  theme_minimal() +
  theme(plot.title = element_text(size=16,hjust = 0.5, face = "bold"),
        axis.title = element_text(size=12, face = "bold"),
        axis.line = element_line(size = 1, colour = "black"),
        axis.text = element_text(size=10, face="bold"),
        legend.position = c(0.025, 0.95),
        legend.justification = c(0, 1))

ggsave("fc_min_temp_year.jpeg", width = 10, height = 6, dpi = 300)
```

```
## `geom_smooth()` using formula = 'y ~ x'
fc_min_temp_year
```

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

## Average Minimum Temperature by Year Fort Collins CO



```
# Calculate the mean of TMIN
fc_ave_min_temp <- mean(fc_data$TMIN)
# Group by decade and calculate the difference from mean
fc_decade_min_temp <- fc_data %>%
  mutate(Year = year(DATE),
         Decade = (Year %/% 10) * 10) %>%
  filter(year(DATE) < 2020) %>%
  group_by(Decade) %>%
  summarize(
    diff_ave_min_temp = (mean(TMIN, na.rm = TRUE) - fc_ave_min_temp),
    sd_min_temp = sd(TMIN, na.rm = TRUE), # Standard deviation of max temperature
    count = n() # Number of observations per decade
  )
# Compute standard error
fc_decade_min_temp <- fc_decade_min_temp %>%
  mutate(se_min_temp = sd_min_temp / sqrt(count))

# Categorize each decade as positive or negative
fc_decade_min_temp <- fc_decade_min_temp %>%
  mutate(temp_category = ifelse(diff_ave_min_temp >= 0, "Positive", "Negative"))

# Create the bar plot with color gradient based on the value of diff_ave_max_temp
```

```

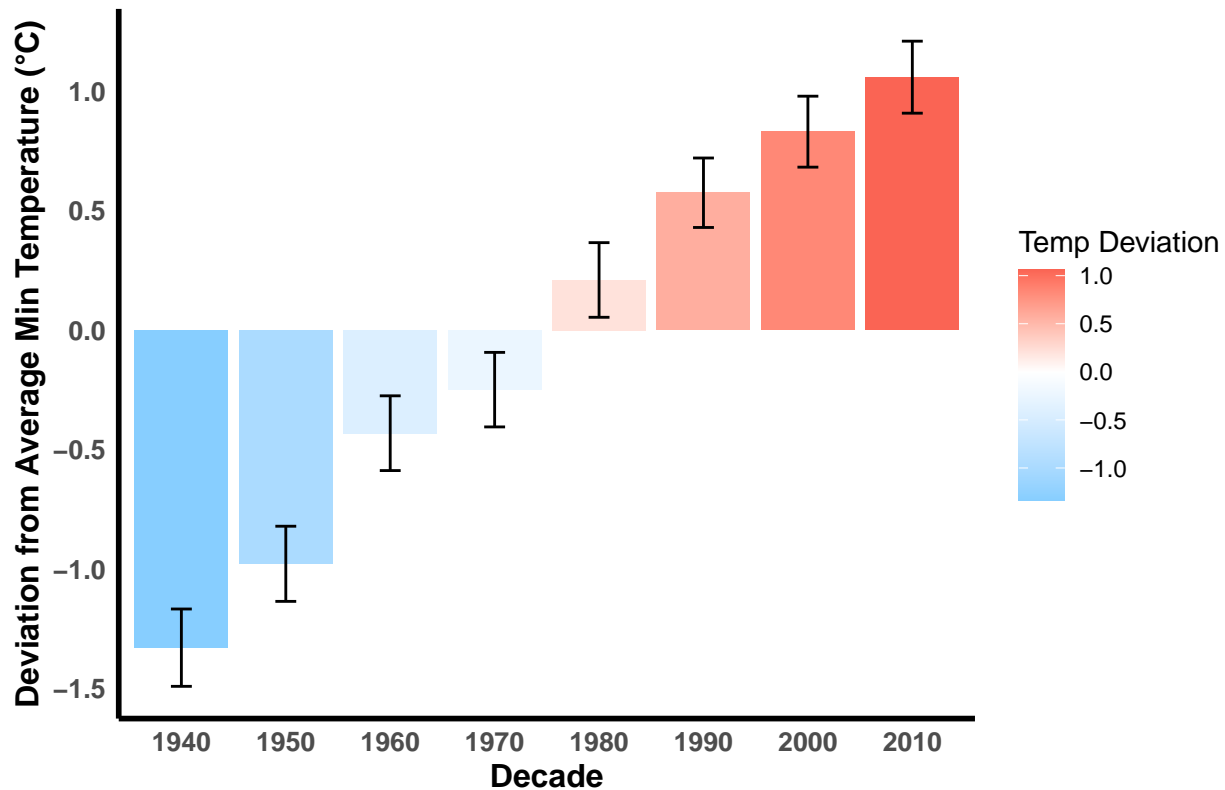
fc_min_temp_decade <- ggplot(data = fc_decade_min_temp,
                             aes(x = as.factor(Decade),
                                y = diff_ave_min_temp,
                                fill = diff_ave_min_temp)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = diff_ave_min_temp - se_min_temp,
                   ymax = diff_ave_min_temp + se_min_temp),
               width = 0.2, color = "black") +
  scale_fill_gradient2(low = "skyblue1", mid = "white", high = "firebrick2",
                      midpoint = 0,
                      name = "Temp Deviation") +
  labs(title = "Changes in Average Min Temperature by Decade in Fort Collins",
       x = "Decade",
       y = "Deviation from Average Min Temperature (°C)") +
  theme_minimal() +
  theme(plot.title = element_text(size=14, hjust = 0.5, face = "bold"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size = 10, face = "bold"),
        axis.line = element_line(size = 1, colour = "black")
  )

# Save the bar plot as a JPEG file
ggsave("fc_min_temp_decade.jpeg", width = 10, height = 6, dpi = 300)

(fc_min_temp_decade)

```

## Changes in Average Min Temperature by Decade in Fort Collins



```
fc_seasonal_max_temp <- fc_data %>%
  mutate(Year = year(DATE),
         Month = month(DATE),
         Season = case_when(
           Month %in% c(12, 1, 2) ~ "Winter",
           Month %in% c(3, 4, 5) ~ "Spring",
           Month %in% c(6, 7, 8) ~ "Summer",
           Month %in% c(9, 10, 11) ~ "Fall"
         )) %>%
  group_by(Year, Season) %>%
  summarize(ave_max_temp = mean(TMAX, na.rm = TRUE))
```

## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.

```
fc_fall_max_temp <- fc_seasonal_max_temp %>%
  filter(Season == "Fall")

# plot average max temp by month with smoothed line
fc_temp_plot_fall <- ggplot(data = fc_fall_max_temp, aes(x = Year)) +
  geom_line(aes(y = ave_max_temp, color = "Average Max Temp"), size = 1.5) +
  geom_smooth(aes(y = ave_max_temp, color = "Smoothed Temp"), method = "loess", size = 1, linetype=2) +
  scale_x_continuous(breaks = seq(1940, 2023, by = 10)) +
  scale_color_manual(values = c("red", "darkred"),
                    labels = c("Average Max Temp", "Smoothed Max Temp")) +
  labs(title = "Average Fall Max Temperature Fort Collins CO",
       x = "Year",
       y = "Average Max Temperature (°C)") +
```

```

theme_minimal() +
theme(plot.title = element_text(size=16,hjust = 0.5, face = "bold"),
      axis.title = element_text(size=12, face = "bold"),
      axis.line = element_line(size = 1, colour = "black"),
      axis.text = element_text(size=10, face="bold"),
      legend.position = c(0.005, 0.995),
      legend.justification = c(0, 1),
      panel.grid.minor = element_blank())

ggsave("fc_temp_plot_fall.jpeg", plot = fc_temp_plot_fall, width = 10, height = 6, dpi = 300)

```

```

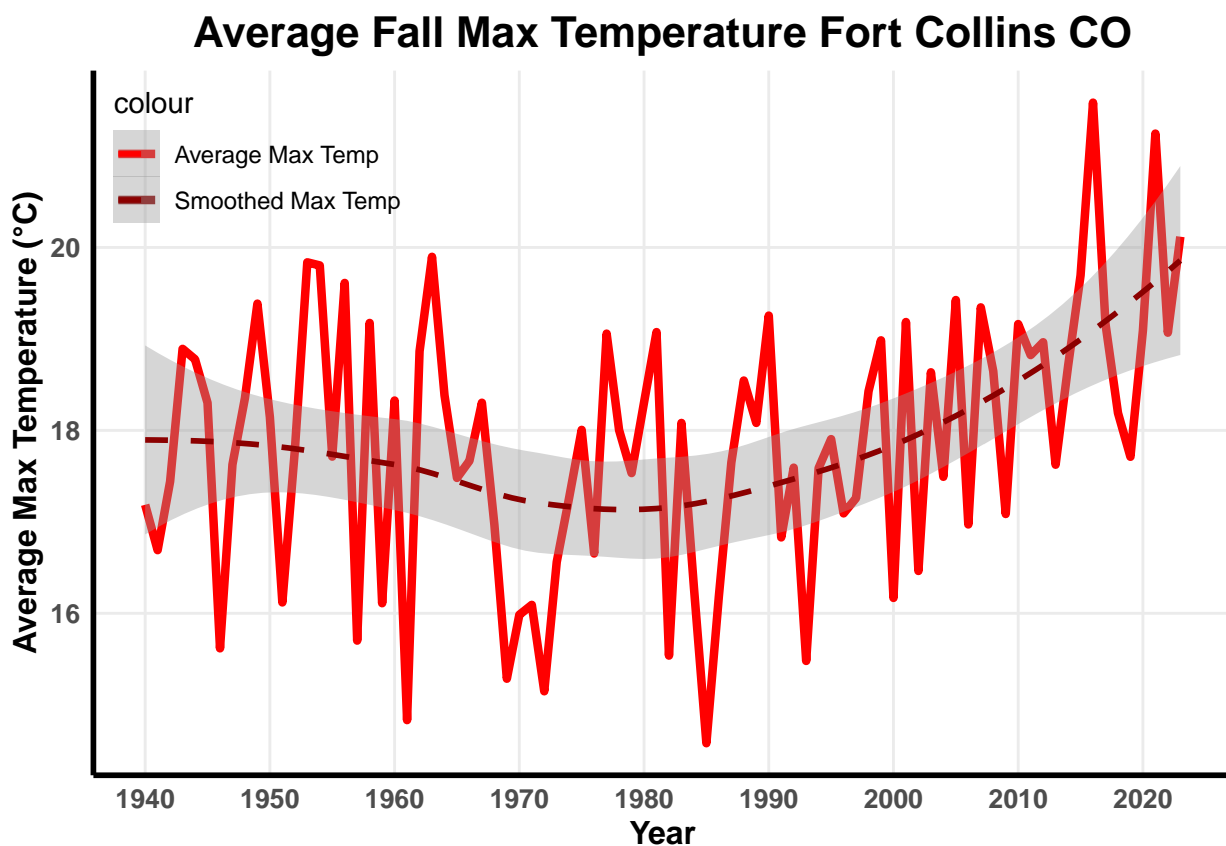
## `geom_smooth()` using formula = 'y ~ x'
fc_temp_plot_fall

```

```

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

```



## Linear Models

Simple linear models with either maximum or minimum temperature as the response and date as the predictor for Grand Junction and Fort Collins. All models show a significant relationship between temperature and date except for the minimum temperature data for Grand Junction.

```

gj.max.lm <- lm(TMAX ~ DATE, data = gj_data)
summary(gj.max.lm)

```

```

##
## Call:

```

```
## lm(formula = TMAX ~ DATE, data = gj_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.658 -10.122   0.455  10.626  22.342
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.865e+01  7.522e-02 247.988  < 2e-16 ***
## DATE        3.744e-05  7.612e-06   4.918  8.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.81 on 30675 degrees of freedom
## Multiple R-squared:  0.0007879, Adjusted R-squared:  0.0007553
## F-statistic: 24.19 on 1 and 30675 DF, p-value: 8.787e-07
```

```
gj.min.lm <- lm(TMIN ~ DATE, data = gj_data)
summary(gj.min.lm)
```

```
##
## Call:
## lm(formula = TMIN ~ DATE, data = gj_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -35.243  -7.436  -0.224   8.672  21.009
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.637e+00  6.230e-02  74.424  <2e-16 ***
## DATE        -2.385e-06  6.305e-06  -0.378    0.705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.779 on 30675 degrees of freedom
## Multiple R-squared:  4.666e-06, Adjusted R-squared: -2.793e-05
## F-statistic: 0.1431 on 1 and 30675 DF, p-value: 0.7052
```

```
fc.max.lm <- lm(TMAX ~ DATE, data = fc_data)
summary(fc.max.lm)
```

```
##
## Call:
## lm(formula = TMAX ~ DATE, data = fc_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -40.922  -7.698   0.500   8.890  22.132
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.715e+01  6.690e-02 256.4  <2e-16 ***
## DATE        6.755e-05  6.755e-06  10.0  <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.45 on 30433 degrees of freedom
## Multiple R-squared:  0.003276,    Adjusted R-squared:  0.003243
## F-statistic:   100 on 1 and 30433 DF,  p-value: < 2.2e-16

fc.min.lm <- lm(TMIN ~ DATE, data = fc_data)
summary(fc.min.lm)

##
## Call:
## lm(formula = TMIN ~ DATE, data = fc_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -41.533  -6.689   0.092   8.053  21.630
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.559e+00  5.934e-02  26.28  <2e-16 ***
## DATE        9.061e-05  5.991e-06   15.12  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.273 on 30433 degrees of freedom
## Multiple R-squared:  0.00746,    Adjusted R-squared:  0.007427
## F-statistic: 228.7 on 1 and 30433 DF,  p-value: < 2.2e-16
```