In [78]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

path = "./data/Mammoth Springs Yellowstone.csv"
df = pd.read_csv(path)
df.head()
```

Out[78]:

	STATION	NAME	DATE	DAPR	MDPR	PRCP	SNOW	SNWD	TMAX
0	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-01	NaN	NaN	0.01	0.5	10.0	32.0
1	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-02	NaN	NaN	0.00	0.0	10.0	28.0
2	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-03	NaN	NaN	0.00	0.0	10.0	28.0
3	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-04	NaN	NaN	0.00	0.0	10.0	27.0
4	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-05	NaN	NaN	0.00	0.0	10.0	25.0

In [79]:
```
# Find the number of staions (Theres only one)
stations = df["STATION"].unique()
print(stations)
```

```
['USC00489905']
```

Introduction to the Region:

In this i am using Yellowstone Park Mammoth, Wy US as my location of choice. I thought Yellowstone would be a good place to get a wide range of temperature and rain/snow which can show a good amount of data. For the data in Yellowstone it has a great area that gets all kinds of weather being in a sweet spot in the country.

In [80]:
```
# Key missing values TMAX, TMIN, PRCP
cleaned = df.copy()

cleaned.dropna(subset=["TMAX", "TMIN", "PRCP"], inplace=True)
```

```
cleaned.drop(columns=["DAPR", "MDPR"], inplace=True)

cleaned.info()

cleaned.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 706 entries, 0 to 719
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   STATION  706 non-null    object 
 1   NAME     706 non-null    object 
 2   DATE     706 non-null    object 
 3   PRCP     706 non-null    float64
 4   SNOW     706 non-null    float64
 5   SNWD     706 non-null    float64
 6   TMAX     706 non-null    float64
 7   TMIN     706 non-null    float64
dtypes: float64(5), object(3)
memory usage: 49.6+ KB
```

Out[80]:

	PRCP	SNOW	SNWD	TMAX	TMIN
count	706.000000	706.000000	706.000000	706.000000	706.000000
mean	0.040921	0.194334	3.252125	54.893768	28.259207
std	0.101796	0.756060	6.160400	20.669264	15.687713
min	0.000000	0.000000	0.000000	-8.000000	-31.000000
25%	0.000000	0.000000	0.000000	37.250000	17.000000
50%	0.000000	0.000000	0.000000	53.000000	30.000000
75%	0.020000	0.000000	3.000000	73.000000	41.000000
max	0.960000	9.500000	23.000000	96.000000	63.000000

In [81]:
```
cleaned["MONTH"] = pd.to_datetime(cleaned["DATE"]).dt.month
cleaned["YEAR"] = pd.to_datetime(cleaned["DATE"]).dt.year
cleaned["MEAN_TEMP"] = (cleaned["TMAX"] + cleaned["TMIN"]) / 2
```

In [82]:
```
cleaned23 = cleaned[cleaned["YEAR"] == 2023]
cleaned23.head()
```

Out[82]:

	STATION	NAME	DATE	PRCP	SNOW	SNWD	TMAX	TMIN	MONTH
0	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-01	0.01	0.5	10.0	32.0	11.0	1
1	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-02	0.00	0.0	10.0	28.0	8.0	1
2	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-03	0.00	0.0	10.0	28.0	6.0	1
3	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-04	0.00	0.0	10.0	27.0	13.0	1
4	USC00489905	YELLOWSTONE PARK MAMMOTH, WY US	2023-01-05	0.00	0.0	10.0	25.0	13.0	1

TMAX and TMIN Temperature

Looking at Temperature the total mean for TMAX is 54 degrees and for TMIN 28 degrees. If we look at the temperature grouped by months for the years of 2023-2024 the hottest month average temperature is July at 83.5 and the coldest average min is January at 10.

In [83]:

```
grouped = (
    cleaned.groupby("MONTH")[["MEAN_TEMP", "TMAX", "TMIN", "PRCP", "SNOW"]]
    .agg(
        {
            "MEAN_TEMP": "mean",
            "TMAX": "mean",
            "TMIN": "mean",
            "PRCP": "sum",
            "SNOW": "sum",
        }
    )
    .reset_index()
)
grouped
```

Out[83]:

	MONTH	MEAN_TEMP	TMAX	TMIN	PRCP	SNOW
0	1	19.653226	29.306452	10.000000	1.40	19.2
1	2	23.008772	33.561404	12.456140	2.45	28.5
2	3	26.274194	38.516129	14.032258	3.37	42.0
3	4	37.549020	50.921569	24.176471	1.23	5.8
4	5	48.745763	62.593220	34.898305	1.65	4.8
5	6	57.025000	72.166667	41.883333	3.56	0.0
6	7	65.408333	83.516667	47.300000	3.06	0.0
7	8	62.637931	78.620690	46.655172	4.13	0.0
8	9	56.112069	72.206897	40.017241	3.15	0.0
9	10	44.409836	58.901639	29.918033	2.94	14.5
10	11	30.870690	41.517241	20.224138	0.89	8.0
11	12	27.833333	37.616667	18.050000	1.06	14.4

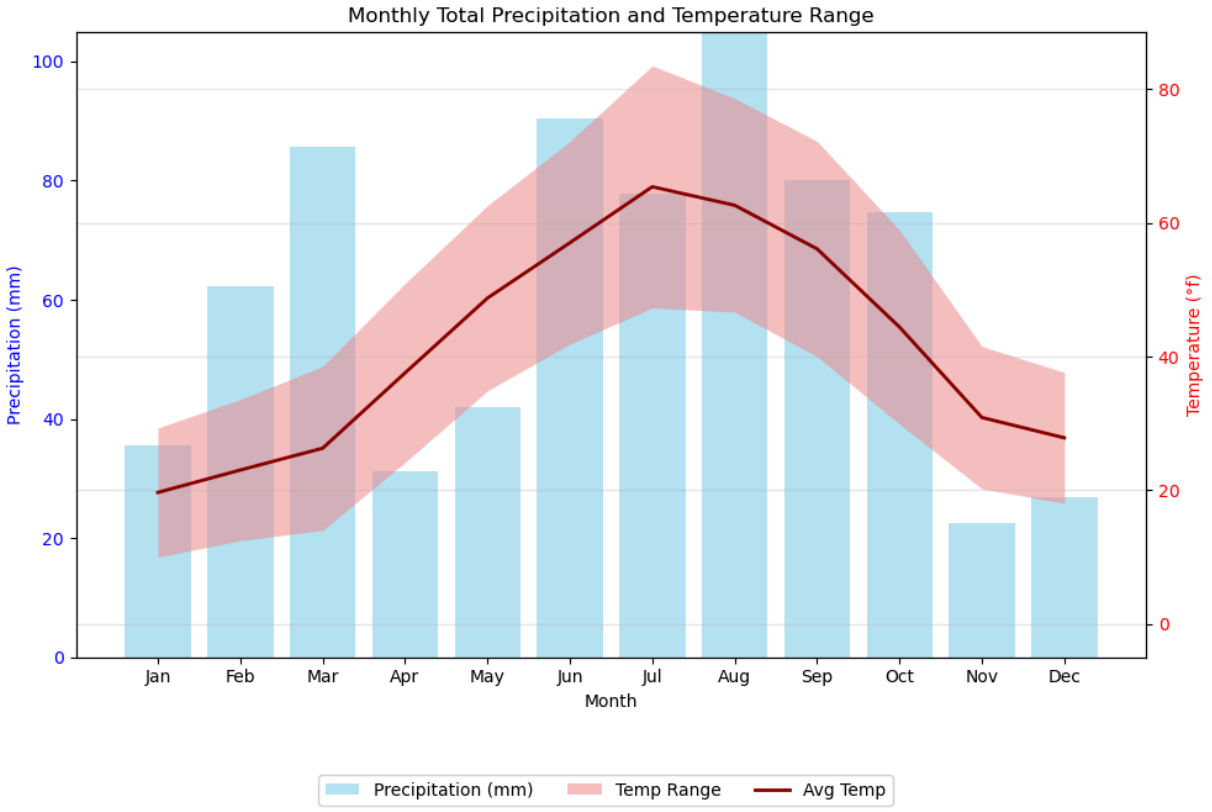
PRCP, TMIN, TMAX, TAVG, Combo Plot

Getting to the charts the first one here is a combo plot with Monthly total precipitation and the temperature as a fill_between which shows the max, min, and average for each month. By looking at this you can see the total precipitation actually happens more during the summer months than it does in the spring months with November and December bringing up the rear with the least amount of precipitation and June and Aug being the most amount of precipitation.

In [84]:

```
# KEEP
# Create plot
fig, ax1 = plt.subplots(figsize=(10, 6))

# Data
x = grouped["MONTH"]
y1 = grouped["TMAX"]
y2 = grouped["TMIN"]

# Secondary y-axis for temperatures
ax2 = ax1.twinx()
ax2.fill_between(
    x, y1, y2, alpha=0.5, color="lightcoral", linewidth=0, label="Temp Range
)
ax2.plot(x, (y1 + y2) / 2, linewidth=2, color="darkred", label="Avg Temp")
ax2.set_ylabel("Temperature (°f)", color="red")
ax2.tick_params(axis="y", labelcolor="red")

# Dynamic temperature y-limits
```

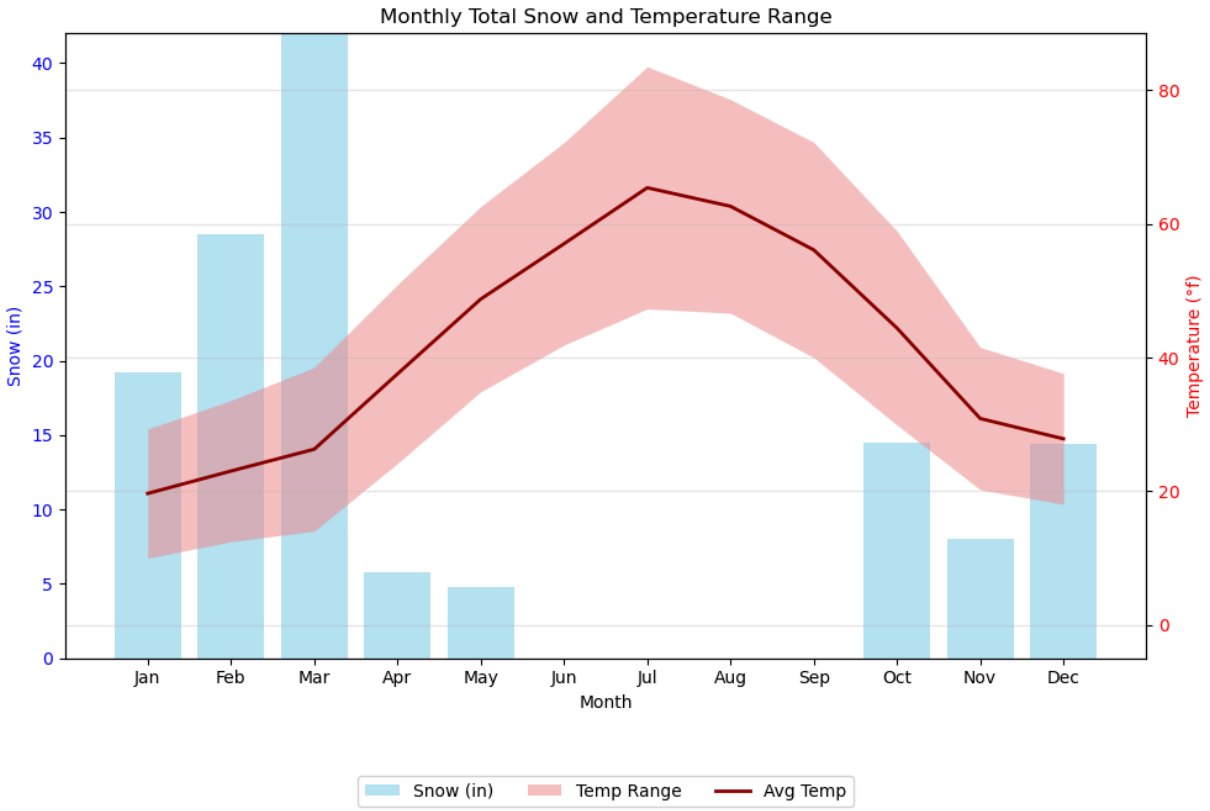
```
y_min = min(y2.min(), 0) - 5
y_max = y1.max() + 5
ax2.set_ylim(y_min, y_max)

# Bar plot for PRCP
ax1.bar(
    x, grouped["PRCP"] * 25.4, color="skyblue", label="Precipitation (mm)",
)
ax1.set_xlabel("Month")
ax1.set_ylabel("Precipitation (mm)", color="blue")
ax1.tick_params(axis="y", labelcolor="blue")
ax1.set_ylim(0, (grouped["PRCP"] * 25.4).max())

# Custom x-ticks
ax1.set_xticks(np.arange(1, 13))
ax1.set_xticklabels(
    ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "
)

# Title and legend
plt.title("Monthly Total Precipitation and Temperature Range")
fig.legend(loc="upper center", bbox_to_anchor=(0.5, -0.05), ncol=3)

# Grid
ax2.grid(True, alpha=0.3)

# Show plot
plt.tight_layout()
plt.show()
```



Snow Temperature Range

This Chart shows Total Snow and Temperature Range showing February and March being the most and a strong drop off into April.

In [85]:
```
# KEEP
# Create plot
fig, ax1 = plt.subplots(figsize=(10, 6))

# Data
x = grouped["MONTH"]
y1 = grouped["TMAX"]
y2 = grouped["TMIN"]

# Secondary y-axis for temperatures
ax2 = ax1.twinx()
ax2.fill_between(
    x, y1, y2, alpha=0.5, color="lightcoral", linewidth=0, label="Temp Range
)
ax2.plot(x, (y1 + y2) / 2, linewidth=2, color="darkred", label="Avg Temp")
ax2.set_ylabel("Temperature (°f)", color="red")
ax2.tick_params(axis="y", labelcolor="red")

# Dynamic temperature y-limits
y_min = min(y2.min(), 0) - 5
y_max = y1.max() + 5
ax2.set_ylim(y_min, y_max)

# Bar plot for PRCP
ax1.bar(x, grouped["SNOW"], color="skyblue", label="Snow (in)", alpha=0.6)
ax1.set_xlabel("Month")
ax1.set_ylabel("Snow (in)", color="blue")
ax1.tick_params(axis="y", labelcolor="blue")
ax1.set_ylim(0, grouped["SNOW"].max())

# Custom x-ticks
ax1.set_xticks(np.arange(1, 13))
ax1.set_xticklabels(
    ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "
)

# Title and legend
plt.title("Monthly Total Snow and Temperature Range")
fig.legend(loc="upper center", bbox_to_anchor=(0.5, -0.05), ncol=3)

# Grid
ax2.grid(True, alpha=0.3)

# Show plot
plt.tight_layout()
plt.show()
```

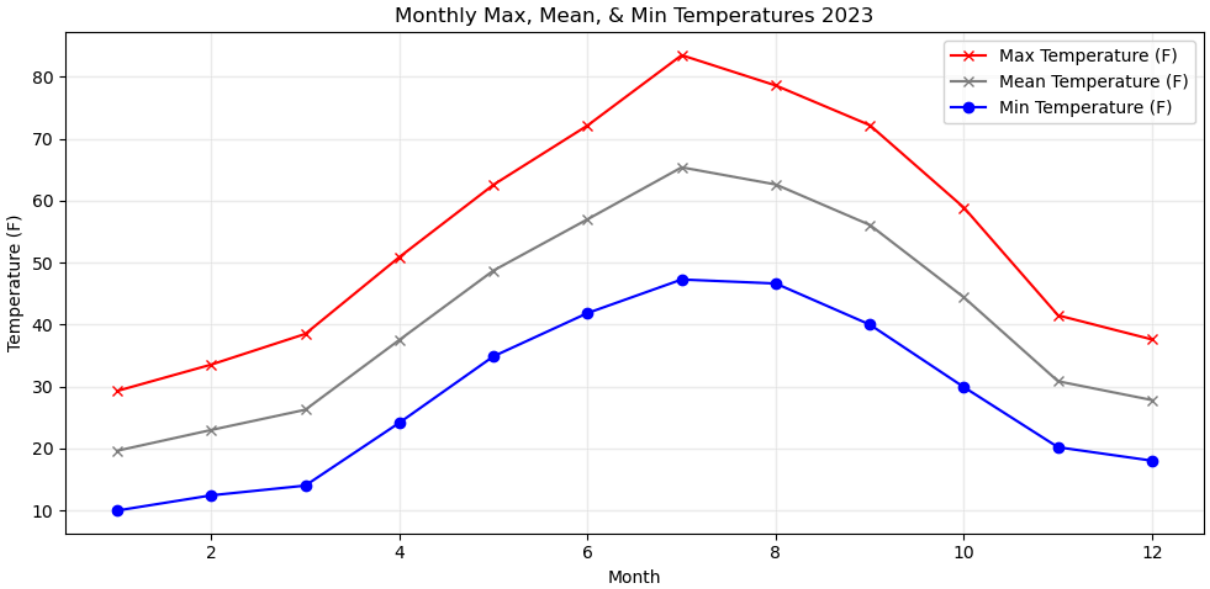


Temperature Line Plot

This Chart shows what the fill_between is but focused more on the temperatures only.

In [86]:
```
# KEEP
fig, ax1 = plt.subplots(figsize=(10, 5))

plt.plot(
    grouped["MONTH"],
    grouped["TMAX"],
    color="red",
    marker="x",
    label="Max Temperature (F)",
)
plt.plot(
    grouped["MONTH"],
    (grouped["TMAX"] + grouped["TMIN"]) / 2,
    color="grey",
    marker="x",
    label="Mean Temperature (F)",
)
plt.plot(
    grouped["MONTH"],
    grouped["TMIN"],
    color="blue",
    marker="o",
    label="Min Temperature (F)",
)
```

```
plt.xlabel("Month")
plt.ylabel("Temperature (F)")
plt.title("Monthly Max, Mean, & Min Temperatures 2023")
plt.legend()
plt.grid(True, alpha=0.2)
plt.tight_layout()
plt.show()
```

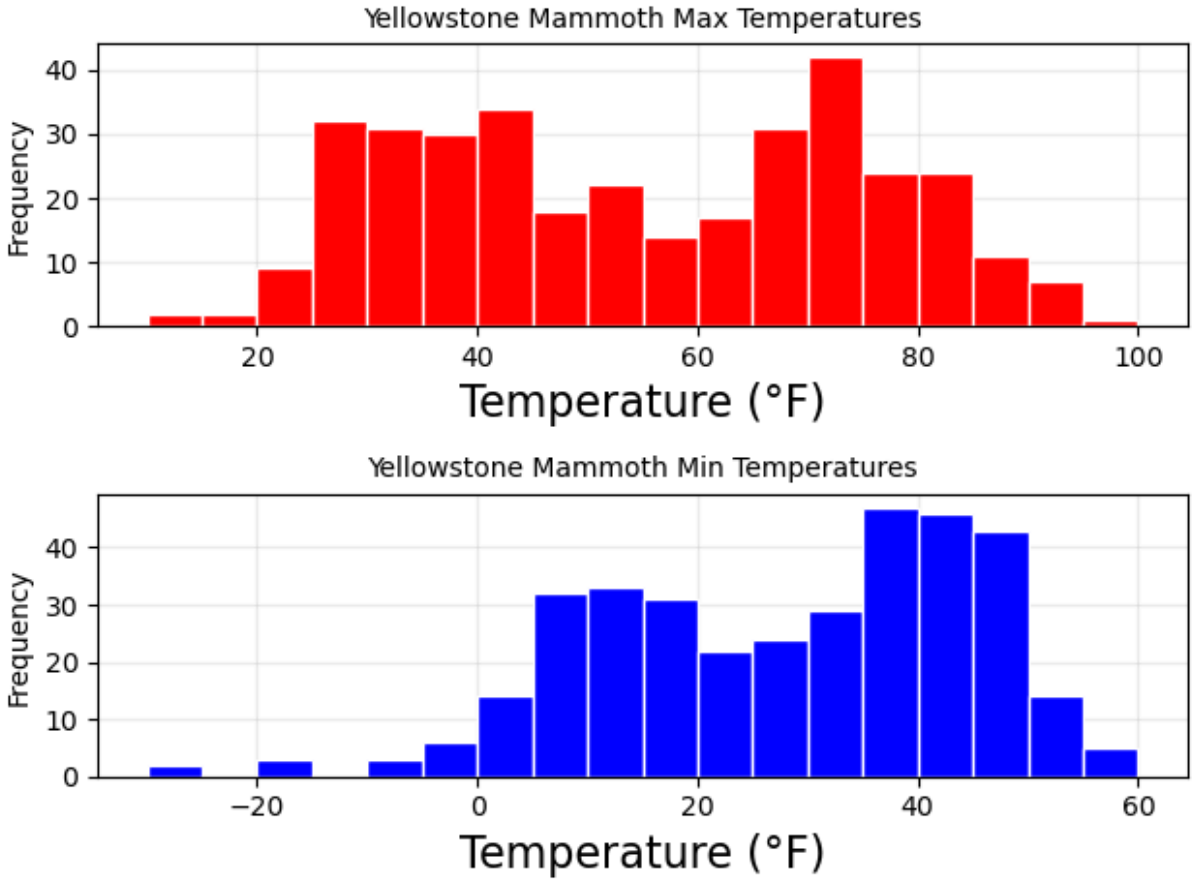


TMAX & TMIN Stacked Bar Graph

This is a stacked chart showing the frequency of the TMAX and TMIN. This shows theres a lot more even distribution in the TMAX than the TMIN with some temps getting really cold with a better cleaned distribution 0-50 degrees with some outliners below 0 and even below -20.

In [87]:
```
max_bins = range(10, 105, 5)
min_bins = range(-30, 65, 5)
fig, (ax1, ax2) = plt.subplots(2, 1)

cleaned23.hist(column="TMAX", ax=ax1, bins=max_bins, facecolor="red", edgecc
cleaned23.hist(
    column="TMIN", ax=ax2, bins=min_bins, facecolor="blue", edgecolor="white
)

ax1.set_title("Yellowstone Mammoth Max Temperatures", fontsize=10)
ax2.set_title("Yellowstone Mammoth Min Temperatures", fontsize=10)

ax1.set_xlabel("Temperature (°F)", fontsize=16)
ax2.set_xlabel("Temperature (°F)", fontsize=16)

ax1.set_ylabel("Frequency", fontsize=10)
ax2.set_ylabel("Frequency", fontsize=10)

ax1.grid(alpha=0.25)
```

```
ax2.grid(alpha=0.25)
ax1.set_axisbelow(True)
ax2.set_axisbelow(True)

plt.tight_layout()
plt.show()
```



In [ ]: