Analyzing Factors Affecting Running Performance

In this project, we aim to explore how different variables may affect running performance in various track events, especially for marathons. The main variables we are analyzing include nationality, age, and environmental conditions, like temperature. Here are our main research questions:

1. **Does temperature affect marathon performance? Increase or decrease in finish time?**
    - To answer this, we decided to use a dataset from the Berlin Marathon containing all of the finishing times and the average temperature on each raceday from 1974 to 2019. After creating an area chart and scatterplot & regression model displaying the average temperature and average runtimes for each year of the Berlin marathon, we agree with the assumption that temperature does indeed make a difference in performance.
2. **Are certain nationalities consistently dominant in specific track events?**
    - To answer this, we created a several heatmaps from the olympic track results containing each athlete's nationality, event, and gender. From the heatmaps, we can see that the United States is the most dominant in track events, but usually this is because they have participated for a long time in the olympics and have sent a larger amount of athletes due to other nations. However, proportionately, we cannot definitively determine if certain nations are significantly better than others. However, athletes of African nationality are known to produce very good track athletes.
3. **Have average finish times for marathons and ultra-marathons improved significantly over the decades? How much?**
    - To answer this, we created a line plot in Altair from the Ultramarathon events data containing each athlete's event, event year, and performance. After creating the line plot, we have seen that there is a slight increase for each long distance running event over time. However, because there is so much fluctuation still, we cannot definitievly conclude improvement in racetimes over the years.
4. **Does peak performance age vary based on event type (short vs. long distance events)?**

- To answer this, we created a line plot in altair from the olympic sport events data containing each athlete's event and age. After creating the bar plots for each gender, we have seen that shorter events have an average younger athlete age than longer distance events like the marathon.

Challenge Goals

We had two main challenge goals.

1. **Multiple Datasets -** Involve the use of at least multiple datasets (4) in our project
2. **New Library -** Utilize the library Altair to create data visualizations to better answer our research questions and understand our findings

Collaboration and Conduct

Students are expected to follow Washington state law on the Student Conduct Code for the University of Washington. In this course, students must:

- Indicate on your submission any assistance received, including materials distributed in this course.
- Not receive, generate, or otherwise acquire any substantial portion or walkthrough to an assessment.
- Not aid, assist, attempt, or tolerate prohibited academic conduct in others.

Update the following code cell to include your name and list your sources. If you used any kind of computer technology to help prepare your assessment submission, include the queries and/or prompts. Submitted work that is not consistent with sources may be subject to the student conduct process.

In [1]:
```
your_name = "Lance Garcia, Sean Liu"
sources = [
    "Google AI 'How to convert HH:MM:SS format into to seconds'",
    "Googled 'How to convert seconds into HH:MM:SS (below)'",
    "https://www.geeksforgeeks.org/python-program-to-convert-seconds-i


    "Googled 'how to fix value error in python', used for try except(b
    "https://www.digitalocean.com/community/tutorials/python-valueerro

    "Googled 'layered area chart Altair",
```

```
    "https://altair-viz.github.io/gallery/layered_area_chart.html?utm_
    "https://altair-viz.github.io/user_guide/compound_charts.html?utm_

    "Googled 'how to make scatterplot and regression line in Altair",
    "https://www.geeksforgeeks.org/introduction-to-altair-in-python/#"
    "https://www.geeksforgeeks.org/scatter-plot-with-regression-line-u

    "Googled 'how to  make a heatmap in Altair",
    "https://altair-viz.github.io/gallery/histogram_heatmap.html"
]

assert your_name != "", "your_name cannot be empty"
assert ... not in sources, "sources should not include the placeholder
assert len(sources) >= 6, "must include at least 6 sources, inclusive
```

Data Settings and Methods

In [2]:
```
# libraries imported
import pandas as pd
import altair as alt
import numpy as np
```

Berlin Marathon Data

Our first dataset holds data from the Berlin Marathon from 1974-2019, containing athlete information, finishing times, raceday temperature, and more. We are using this data primarily for answering the first research question of how temperature affects marathon performance.

In [3]:
```
# Race Results
berlin_marathon = pd.read_csv("Berlin_Marathon_data_1974_2019.csv", lo
# Raceday Temperatures
berlin_weather = pd.read_csv("Berlin_Marathon_weather_data_since_1974.

def time_to_seconds(time_str):
    """
    takes in a time string in HH:MM:SS format
    and returns the time back in seconds
    """
    try:
        h, m, s = map(int, time_str.split(":"))
        return h * 3600 + m * 60 + s
    except ValueError:
        return None

# Adds a time column (in seconds) to the data frame
berlin_marathon["TIME_SECONDS"] = berlin_marathon["TIME"].apply(time_t
```

```
# def to_HHMMSS(seconds):
#     """
#     takes in a time string in seconds
#     and returns the time back in HH:MM:SS format
#     """
#     seconds = seconds % (24 * 3600)
#     hour = seconds // 3600
#     seconds %= 3600
#     minutes = seconds // 60
#     seconds %= 60
#     return "%d:%02d:%02d" % (hour, minutes, seconds)

# Calculates average finish time per year and puts it into a new data
avg_per_year = berlin_marathon.groupby("YEAR")["TIME_SECONDS"].mean().
# Merge the weather data based on the "YEAR"
avg_per_year = avg_per_year.merge(berlin_weather, on="YEAR", how="left

display(avg_per_year)
```

	YEAR	TIME_SECONDS	PRECIP_mm	SUNSHINE_hrs	CLOUD_hrs	ATMOS_PR
0	1974	15052.725410	0.0	0.20	7.0	
1	1975	14735.639485	4.2	9.40	3.0	
2	1976	13177.877814	3.0	10.20	3.3	
3	1977	12382.859649	11.2	0.00	8.0	
4	1979	12584.328829	0.0	5.40	2.0	
5	1981	13590.503535	0.1	3.60	7.3	
6	1982	13874.243329	0.7	10.10	1.0	
7	1983	13181.544102	0.0	9.80	2.3	
8	1984	12754.513847	3.9	0.10	7.3	
9	1985	12744.029076	0.0	8.70	5.0	
10	1986	12873.595388	0.0	6.50	5.0	
11	1987	12915.768290	0.0	9.90	1.0	
12	1988	12786.573742	4.2	0.00	7.0	
13	1989	12914.692959	0.1	1.80	5.7	
14	1990	13025.278080	0.5	0.10	7.3	
15	1991	13336.133360	0.0	0.90	7.3	
16	1992	13764.708457	0.0	10.20	0.3	
17	1993	13383.207219	0.0	0.70	7.3	
18	1994	13647.329281	0.0	1.90	4.3	

19	1995	13710.812047	0.0	7.40	4.7
20	1996	13266.034020	7.0	0.00	8.0
21	1997	13784.094313	0.0	11.40	1.0
22	1998	14260.089368	0.0	9.90	1.0
23	1999	14178.979351	1.2	0.00	6.3
24	2000	14850.627825	0.0	5.70	6.3
25	2001	14441.143223	4.5	0.00	7.5
26	2002	14458.792613	0.0	0.00	7.3
27	2003	14729.884757	9.8	7.40	4.3
28	2004	14600.538791	0.6	0.60	6.5
29	2005	15050.660521	0.0	10.60	2.0
30	2006	15372.031906	0.0	11.30	0.1
31	2007	14816.475066	0.0	0.80	7.0
32	2008	14820.991688	0.0	10.60	2.2
33	2009	15170.031758	0.0	10.90	0.6
34	2010	14746.745390	29.8	0.00	7.8
35	2011	14997.533360	0.0	11.30	0.8
36	2012	14790.325066	0.0	8.90	1.6
37	2013	14718.761687	0.0	10.70	1.4
38	2014	14869.665953	0.0	9.78	4.5
39	2015	14830.948902	0.0	8.93	2.0
40	2016	15158.214581	0.0	10.43	5.6
41	2017	15136.338296	0.3	1.50	7.5
42	2018	15539.874585	0.0	7.75	5.6
43	2019	14559.722042	8.0	0.70	6.8

Olympic Track Event Results from 1896 to 2016

This is our second dataset, which holds data on the winners of each Olympic track-specific event result since the late 1800s! This dataset contains a variety of different columns, including athlete name, their nationality, their event, and medal earned. We are using this dataset for the second rewsearch question on how

much of a factor nationality plays in medal attainment.

In [4]:
```
oly_track_results = pd.read_csv("results.csv")

# includes the full names of the country abbreviations
regions = pd.read_csv("noc_regions.csv")

# added a the full name of the country of the nationality abbreviation
oly_track_results = oly_track_results.merge(regions, left_on="National
del (oly_track_results["notes"], oly_track_results["NOC"])

# filter to only include solely running events
men_filtered = oly_track_results.loc[oly_track_results["Event"].isin(
    ['100M Men', '200M Men', '400M Men', '800M Men', '1500M Men', '500
    )]
women_filtered = oly_track_results.loc[oly_track_results["Event"].isin
    ['100M Women', '200M Women', '400M Women', '800M Women', '1500M Wo
    )]
```

Olympic Games Results from 1896-2016 (all sports)

Our third dataset holds data on all athletes who particpated /across all sports of the Olympic Summer and Winter games from 1896-2016.

In [5]:
```
# Data Containing All of the Olympic Games Results over the past 120 y
oly_all_results = pd.read_csv("athlete_events.csv")

# These are the track events for men we are examining
mens_events = [
    "Athletics Men's 100 metres",
    "Athletics Men's 200 metres",
    "Athletics Men's 400 metres",
    "Athletics Men's 800 metres",
    "Athletics Men's 1,500 metres",
    "Athletics Men's 5,000 metres",
    "Athletics Men's Marathon"
]
# filter to only include solely male distance events
oly_filtered_m = oly_all_results.loc[oly_all_results["Event"].isin(men
# filter to only include solely male distance events with finalists
oly_filtered_m_finalists = oly_filtered_m.loc[oly_filtered_m["Medal"].

# These are the track events for women we are examining
womens_events = [
    "Athletics Women's 100 metres",
    "Athletics Women's 200 metres",
    "Athletics Women's 400 metres",
    "Athletics Women's 800 metres",
    "Athletics Women's 1,500 metres",
    "Athletics Women's 5,000 metres",
```

```
    "Athletics Women's Marathon"
]
# filter to only include solely female distance events
oly_filtered_w = oly_all_results.loc[oly_all_results["Event"].isin(wom
# filter to only include solely female distance events with finalists
oly_filtered_w_finalists = oly_filtered_w.loc[oly_filtered_w["Medal"].

display(oly_filtered_m_finalists)

display(oly_filtered_w_finalists)
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	G
191	86	Jos Manuel Abascal Gmez	M	26.0	182.0	67.0	Spain	ESP	Su
720	411	Gezahgne Abera	M	22.0	166.0	58.0	Ethiopia	ETH	Su
915	519	Harold Maurice Abrahams	M	24.0	183.0	75.0	Great Britain	GBR	Su
6260	3512	Nijel Carlos Amilfitano Amos	M	18.0	179.0	60.0	Botswana	BOT	Su
8475	4667	Sad Aouita	M	24.0	175.0	58.0	Morocco	MAR	Su
...	...	...	...	...	...	...	...	...	
266986	133579	Juan Carlos Zabala Boyer	M	20.0	165.0	55.0	Argentina	ARG	Su
268010	134080	Emil Ztopek	M	25.0	182.0	72.0	Czechoslovakia	TCH	Su
268012	134080	Emil Ztopek	M	29.0	182.0	72.0	Czechoslovakia	TCH	Su
268014	134080	Emil Ztopek	M	29.0	182.0	72.0	Czechoslovakia	TCH	Su
269909	135042	Kazimierz Franciszek Zimny	M	25.0	172.0	60.0	Poland	POL	Su

588 rows × 15 columns

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Game
747	428	Elvan Abeylegesse	F	25.0	159.0	40.0	Turkey	TUR	200 Summe
6220	3494	Judith Florence "Judy" Amoore-Pollock	F	24.0	163.0	55.0	Australia	AUS	196 Summe
6757	3788	Grete Andersen-Waitz	F	30.0	172.0	53.0	Norway	NOR	198 Summe
8380	4613	Nataliya Nikolayevna Antyukh	F	23.0	182.0	69.0	Russia	RUS	200 Summe
9211	5069	Yuko Arimori (-Wilson)	F	25.0	166.0	47.0	Japan	JPN	199 Summe
...	...	...	...	...	...	...	...	...	.
265362	132794	Valentina Mikhaylovna Yegorova (Vasilyeva-)	F	28.0	155.0	50.0	Unified Team	EUN	199 Summe
265363	132794	Valentina Mikhaylovna Yegorova (Vasilyeva-)	F	32.0	155.0	50.0	Russia	RUS	199 Summe
268293	134243	Monika Zehrt (-Landgraf)	F	19.0	168.0	56.0	East Germany	GDR	197 Summe
269377	134789	Zhou Chunxiu	F	29.0	162.0	45.0	China	CHN	200 Summe
270000	135074	Elfi Zinn (Rost-)	F	22.0	165.0	55.0	East Germany	GDR	197 Summe

288 rows × 15 columns

UltraMarathon Running Data

This dataset holds recorded data on athletes who particpated in various ultramarathon (>26 mile races) events from 1798-2018. We are using this dataset primarily for exploring whether marathon / ultramarathon performances have increased over the years.

In [6]:
```
# Data containing a bunch of different long distance events
um_results = pd.read_csv("TWO_CENTURIES_OF_UM_RACES.csv")

# finds the top 10 most common elements in a given dataframe and colum
def top_10_common(df, column_name):
    """
    Returns the 10 most common values in a given column with their cou
    df -  DataFrame to analyze and column_name - Column to count value
    Returns a DataFrame with the top 10 most common values and their c
    """
    return df[column_name].value_counts().head(11)
top_10 = top_10_common(um_results, "Event distance/length")

# including only results from the 10 most common events in the data
most_common = ["50km","100km","50mi","56km","87km","100mi","60km","45k
um_results_mc = um_results[um_results["Event distance/length"].isin(mo

# remove the trailing 'h' in the performance times
um_results_mc["Athlete performance"] = um_results_mc["Athlete performa

# transform the HH:MM:SS format into seconds only
um_results_mc["Athlete performance"] = pd.to_timedelta(um_results_mc["

um_results_mc
```

```
/var/folders/lc/qq0mf0k931b_w3zbk4yycds00000gn/T/ipykernel_42873/3621788333.py:2: DtypeWarning: Columns (11) have mixed types. Specify dtype option on import or set low_memory=False.
  um_results = pd.read_csv("TWO_CENTURIES_OF_UM_RACES.csv")
/var/folders/lc/qq0mf0k931b_w3zbk4yycds00000gn/T/ipykernel_42873/3621788333.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  um_results_mc["Athlete performance"] = um_results_mc["Athlete performance"].str.rstrip(" h")
/var/folders/lc/qq0mf0k931b_w3zbk4yycds00000gn/T/ipykernel_42873/3621788333.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  um_results_mc["Athlete performance"] = pd.to_timedelta(um_results_mc["Athlete performance"]).dt.total_seconds()
```

Out[6]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	
0	2018	06.01.2018	Selva Costera (CHI)	50km	22	17499.0	
1	2018	06.01.2018	Selva Costera (CHI)	50km	22	18945.0	E
2	2018	06.01.2018	Selva Costera (CHI)	50km	22	19004.0	
3	2018	06.01.2018	Selva Costera (CHI)	50km	22	20053.0	
4	2018	06.01.2018	Selva Costera (CHI)	50km	22	21254.0	
...	...	...	...	...	...	...	
7461181	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	43177.0	

7461182	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	43301.0
7461183	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	43406.0
7461184	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	43406.0
7461185	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	43559.0

3994071 rows × 13 columns

Results

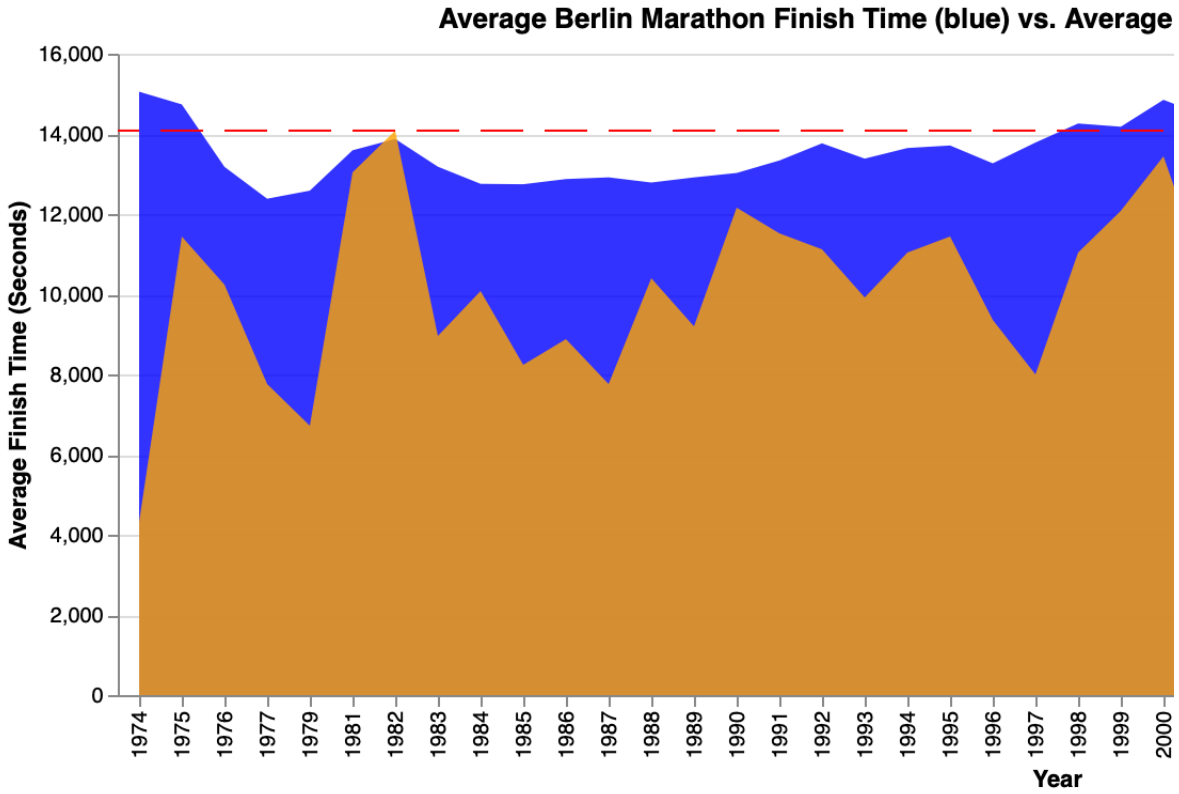
Q1 - Does Temperature Affect Running Performance?

- So for this question, we created an area map and a scatter plot to help better visualize the affect of temperature on the Berlin Marathon running performance times. Again, we used the Berlin marathon data from 1974-2019. We decided to use the Berlin Marathon Dataset because it included data on raceday temperature.
- From the results, we assume that temperature does indeed make a difference in affecting running performances. It does make logical sense that the hotter it gets, the harder it is to run.
    1. Let's first examine the area chart below: In this chart you can see blue -- representing the average time in seconds to complete the marathon for a certain year and also orange -- representing the average temperature on the day of the race. Lastly, the red dotted line represents the average time to complete the marathon across all the years of the data. Looking at the data here, we can see that when temperature goes up, the time it takes to finish also generally tends to rise as well.

2. Next, let's look at the scatter plot. The regression line we calculated was **TIME_SECONDS = 91.69 * AVG_TEMP_C + 12907.67**. This shows that there is a slight positive slope and further shows that there is likely a relationship between the two.

In [7]:
```
# Plots the average Berlin marathon finishing time for each year in bl
finishing_times = alt.Chart(avg_per_year).mark_area(opacity=0.8, color
    x=alt.X("YEAR:N", title="Year"),
    y=alt.Y("TIME_SECONDS:Q", title="Average Finish Time (Seconds)"),
    tooltip=["YEAR", "TIME_SECONDS"]
)
# finishing_times

# Plots the Berlin marathon terperature for each year in orange
temp = alt.Chart(avg_per_year).mark_area(opacity=0.8, color="orange").
    x=alt.X("YEAR:N"),
    y=alt.Y("AVG_TEMP_C:Q", title="Average Temperature (Celsius)"),
    tooltip=["YEAR", "AVG_TEMP_C"]
)
# temp

# Calculate the overall average finishing time across all years
overall_avg_time = avg_per_year["TIME_SECONDS"].mean()

# Create a horizontal rule for the average finishing time
avg_time_line = alt.Chart(pd.DataFrame({"y": [overall_avg_time]})).mar
    color="red", strokeDash=[20,10]
).encode(
    y=alt.Y("y:Q", title=None, axis=None)
)

# # layers the two charts on top of each other
finishing_times_temp = alt.layer(finishing_times, temp, avg_time_line)
    title="Average Berlin Marathon Finish Time (blue) vs. Average Temp
)

finishing_times_temp
```

Out[7]:

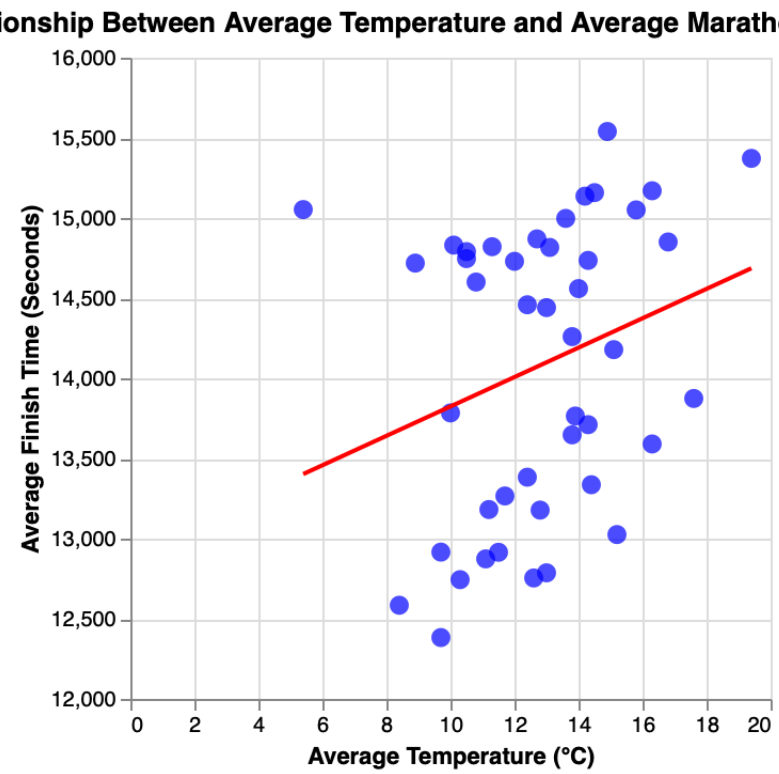


In [8]:

```
# Scatter plot of average temperature vs. average finishing time
scatter = alt.Chart(avg_per_year).mark_circle(size=80, color="blue", o
    x=alt.X("AVG_TEMP_C:Q", title="Average Temperature (°C)"),
    y=alt.Y("TIME_SECONDS:Q", title="Average Finish Time (Seconds)", s
    tooltip=["YEAR", "AVG_TEMP_C", "TIME_SECONDS"]
)

# Added a regression line
regression = scatter.transform_regression(
    "AVG_TEMP_C", "TIME_SECONDS", method="linear"
).mark_line(color="red")

# Combine scatter plot and trend line
temp_vs_time_chart = (scatter + regression).properties(
    title="Relationship Between Average Temperature and Average Marath
)

# Perform regression and extract parameters separately with numpy
slope, intercept = np.polyfit(avg_per_year["AVG_TEMP_C"], avg_per_year
# Print regression equation
print(f"Regression equation: TIME_SECONDS = {slope:.2f} * AVG_TEMP_C +

temp_vs_time_chart
```

```
Regression equation: TIME_SECONDS = 91.69 * AVG_TEMP_C + 12907.67
```

Out[8]:



Q2 - Are certain nationalities dominant in certain track events or in track in general?

- To answer this question, we used our olympic track event results dataset and created three heatmaps to better visualize of the medals earned across different nationalities.

- From our results, we cannot definitively conclude that nationalities constitute to track event dominance. However, we can see that the United States, with such a large population,does have the most medals, which is likely the case because they consistently send the most athletes to the olympics and have particpated in the Olympics every year for many, many years.

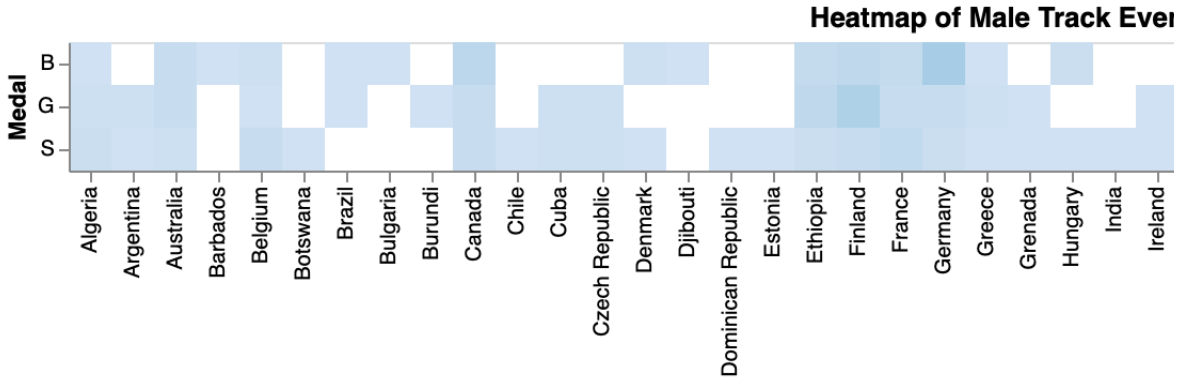
    1. The first heatmap shows the medals earned by male runners from each nationality. As you can see, USA has the most. However, there is a good distribution between other countries. Great Britain is a close second.

    2. The second heatmap is the same as the first, but it shows the medals earned by female runners from each nationality. As you can see, USA has the most.

    3. The final heatmap is shows the distribution of medals earned by nationality across various different specific track events. USA wins here also.

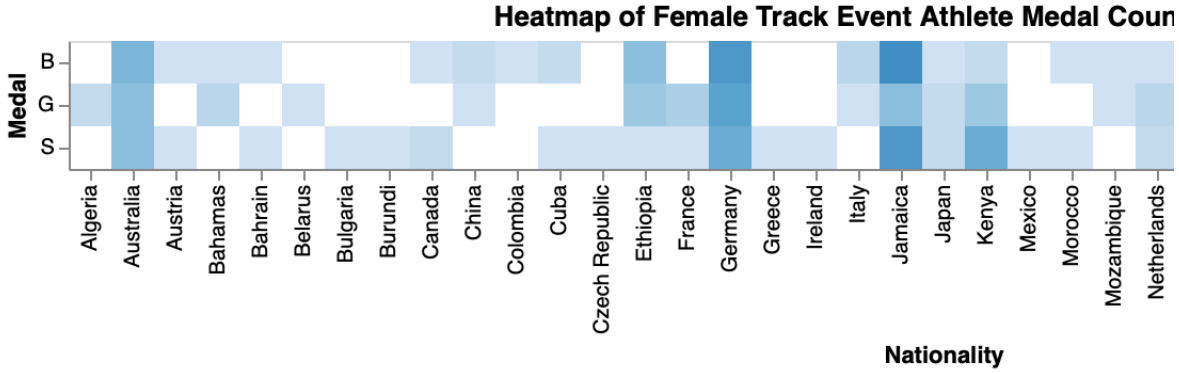
In [9]:
```
# Group by nationality and medal type, and count the occurrences
male_medals_count = men_filtered.groupby(['region', 'Medal']).size().r
# Create a heatmap to show the number of medals
male_finalist_heatmap = alt.Chart(male_medals_count).mark_rect().encod
    x=alt.X('region:N', title='Nationality'),
    y=alt.Y('Medal:N', title='Medal'),
    color=alt.Color('Count:Q', scale=alt.Scale(scheme='blues'), title=
    tooltip=['region', 'Medal', 'Count']
).properties(
    title="Heatmap of Male Track Event Athlete Medal Counts by Nationa
)
male_finalist_heatmap
```

Out[9]:



In [10]:
```
# Group by nationality and medal type, and count the occurrences
female_medals_count = women_filtered.groupby(['region', 'Medal']).size
# Create a heatmap to show the number of medals
female_finalist_heatmap = alt.Chart(female_medals_count).mark_rect().e
    x=alt.X('region:N', title='Nationality'),
    y=alt.Y('Medal:N', title='Medal'),
    color=alt.Color('Count:Q', scale=alt.Scale(scheme='blues'), title=
    tooltip=['region', 'Medal', 'Count']
).properties(
    title="Heatmap of Female Track Event Athlete Medal Counts by Natio
)
female_finalist_heatmap
```

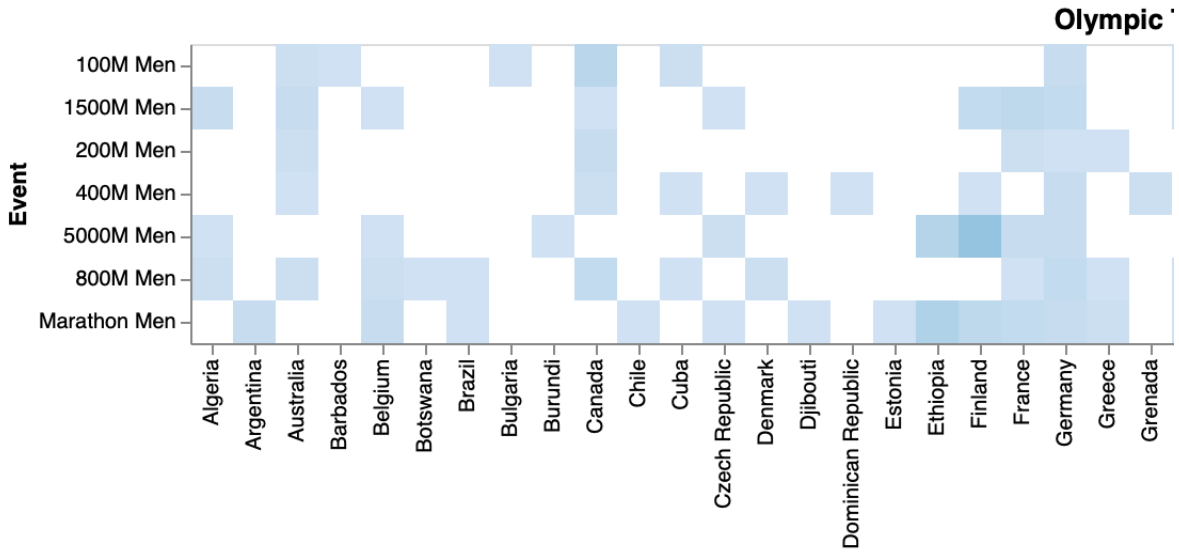
Out[10]:



In [11]:
```
# Group by event and nationality
event_medal_counts = men_filtered.groupby(["region", "Event"]).size().
```

```
# Create heatmap
event_heatmap = alt.Chart(event_medal_counts).mark_rect().encode(
    y=alt.Y("Event:N", title="Event"),
    x=alt.X("region:N", title="Nationality", sort="x"),  # Sort by mos
    color=alt.Color("Count:Q", scale=alt.Scale(scheme="blues")),  # He
    tooltip=["Event", "region", "Count"]
).properties(
    title="Olympic Track Event Dominance by Nationality"
)
event_heatmap
```

Out[11]:



Q3 - Have average finish times for ultra-marathons improved significantly over the decades?

- To answer this question, we used our ultramarathon track events dataset, containing tens of thousands of different athletes competing in a wice variety of different ultramarathon events.

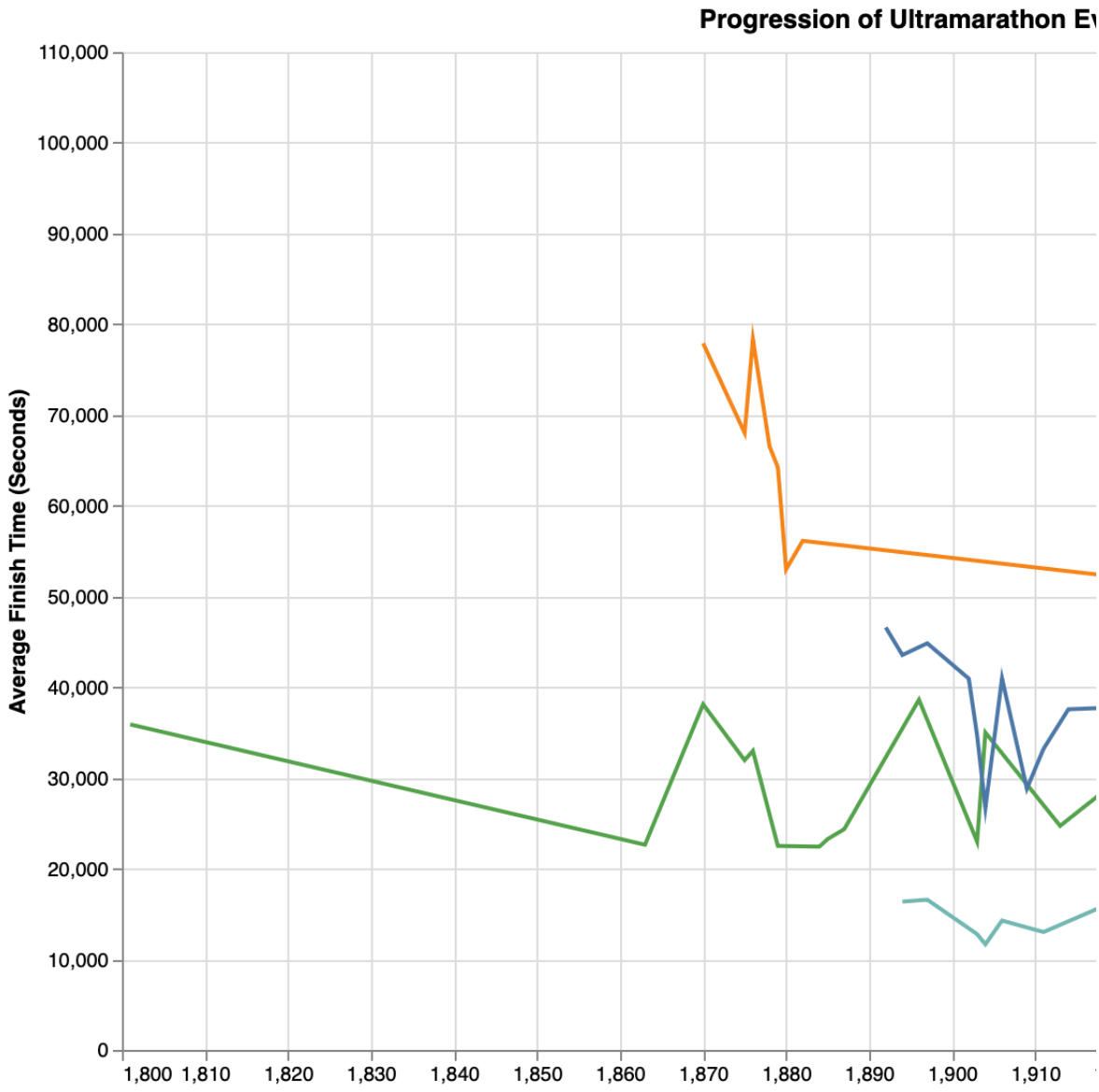
    1. From our results in our line plot, we do see that there is a slight decrease in performance (the times got higher). However, this definitiely does not allow us to say that performance has data in the earlier years of the data. In contrast to the data, we believe that with a increease in better training standards and sports technologies, people are aactually in fact running these events faster than previously before.

In [23]:

```
# Group by Year and Event to get average time per event per year
um_avg_finishes = um_results_mc.groupby(["Year of event", "Event dista

# Create the line chart
line_chart = alt.Chart(um_avg_finishes).mark_line().encode(
    x=alt.X("Year of event", title="Year", axis=alt.Axis(labelAngle=0)
```

```
    y=alt.Y("Athlete performance:Q", title="Average Finish Time (Secon
    color="Event distance/length:N",  # Different colors for each even
    tooltip=["Year of event", "Event distance/length", "Athlete perfor
).properties(
    title="Progression of Ultramarathon Event Finishing Times Over the
    width=1000,
    height=500
)
line_chart
```

Out[23]:



Q4 - Does peak performance age vary based on event type (short vs. long distance)?

- To answer this question, we used our Olympic Sports Events (not track specific) data set because it included each athlete's ages. We filtered out the data for only track events and the athletes that won a medal. Then we

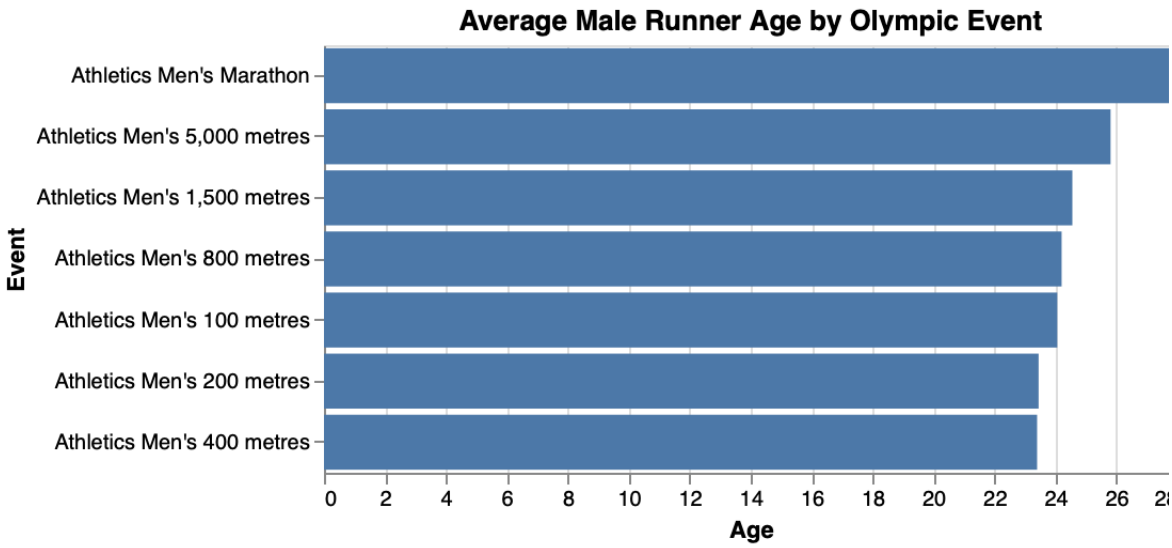
created two bar plots showing the average age of athletes per Olympic track event, one for each gender of course.

1. Looking at the two different bar plots, we can see that the shorter distance events for both genders have younger athletes. From this, we can generally assume that younger athletes are more explosive and better suited for short-distance events that require fast-twitch muscle power and brute force. However, for longer distance events like the marathon, the average age is higher. This shows that older athletes are still perfectly capable of performing at an elite level at an older age. It shows that age is less importantof a success factor for longer events.

2. A graph that we were not able to create was the distribution of race times per track event per age. I wasn't able to do this because I coldn't quite merge my other Olympic dataset togehter with this one. The other one had the race times, but this one had the athlete ages. After some difficulties, I was unfortunately unable to create such a graph comibining both age and time variables.

In [20]:
```
m_finalists_avg_age = oly_filtered_m_finalists.groupby('Event')['Age']
# Create the Altair chart
chart1 = alt.Chart(m_finalists_avg_age).mark_bar().encode(
    x='Age:Q',
    y=alt.Y('Event:N', sort='-x'),
    tooltip=['Event:N', 'Age:Q']
).properties(
    title='Average Male Runner Age by Olympic Event',
    height=200,
    width=400
)

w_finalists_avg_age = oly_filtered_w_finalists.groupby('Event')['Age']
# Create the Altair chart
chart2 = alt.Chart(w_finalists_avg_age).mark_bar().encode(
    x='Age:Q',
    y=alt.Y('Event:N', sort='-x'),
    tooltip=['Event:N', 'Age:Q']
).properties(
    title='Average Female Runner Age by Olympic Event',
    height=200,
    width=400
)

chart1 | chart2
```

Out[20]:



Implications and Limitations

- **Who might benefit from your analysis and who might be harmed by it?**
  - Other athletes looking to study about sports performance can probably take something away positive from our analysis! I don't think our analysis excludes anyone in particular. We are just analyzing historic data, not predicting anything.
- **What about the data setting might have impacted your results?**
  - Our data is a little outdated and probably has lots of skew. There is probably more data in the more recent years than in prior years due to poor documentation historically, esepcially in the ultramarathon (long distance) events dating back centuries.
- **Explain at least 3 limitations of your analysis and how others should or shouldn't be advised to use your conclusions.**?

1. Variation Across Events: Peak performance age varies between track events. Sprinters (100m, 200m) often peak in their mid-20s, while marathoners and ultra-marathoners typically peak later, in their 30s. The average across all events could obscure these differences.

2. Exclusion of Other Factors: The analysis focuses only on age, ignoring factors like injury history, training, and nutrition. These impact when athletes peak, especially in endurance sports like marathons and ultra-marathons.

3. Some athletes, especially in endurance events, may peak later and have longer careers. This analysis doesn't consider how long athletes can

perform at high levels, especially in marathons and ultra-marathons.