Project Report
Data Analytics Traineeship Program 2024

On

“Analysis of Fitness Data”

28th August 2024

MedTourEasy

Acknowledgements

I am immensely grateful for the traineeship opportunity I had with MedTourEasy, which allowed me to delve deep into the **"Analysis of Fitness Data".** This experience was invaluable in enhancing my understanding in data analytics, as well as contributing significantly to my personal and professional growth.

I would like to express my deepest gratitude to **Mr. Ankit Hasija**, my mentor during this project. His guidance and expertise were instrumental in helping me navigate the complexities of the project. I am sincerely thankful for his support, which made this individual project a fulfilling and enriching learning experience.

Finally, I would like to extend my heartfelt thanks to **MedTourEasy** for providing me with this opportunity. The insights and skills I gained during this traineeship will undoubtedly contribute to my future endeavours in the field of data analytics.

Table of Contents

S.No.	Content	Page No.
	Abstract	1
1.	Introduction	2
	1.1 About the Company	2
	1.2 Project Outline: Objective and Deliverables	2
2.	Methodology	4
	2.1 Flow of the Project	4
	2.2 Programming Tools	4
3.	Implementation	6
	3.1 Defining Problem Statement	6
	3.2 Data Collection	6
	3.3 Cleaning and Preprocessing	6
	3.4 Visualizations and Observations	8
	3.5 Summarization	13
	3.6 Gather Insights	14
4.	Conclusion and Future Scope	16
5.	References	18

Abstract

With the rapid rise in the popularity of fitness trackers, fitness freaks worldwide collect and analyse their training activity data to keep themselves motivated and to consistently monitor their progress. This project aimed to answer some key questions like: How fast, long, and intense was my run today? Have I succeeded with my training goals? What is my progress? And more, using the exported data from the Runkeeper app.

The project involved reviewing the raw data, followed by the initial steps of data cleaning, and preprocessing which includes handling duplicates and missing values through methods like dropping and mean imputation respectively. Various visualizations were created to analyse the running statistics, including the running averages and heart rate comparisons. To meet the set goals, a benchmark of 1000 km per year was set and compared against the average running distances per year from the data. The trend of running distances were also visualized to assess progress over time.

This study highlights the significance of tracking fitness metrics for maintaining a healthy lifestyle and contributes invaluably in h field of healthcare by showcasing the benefits of data-driven fitness monitoring. The methodologies applied serves as a template for others looking to analyse their training data to optimize performance and reach their goals.

1. Introduction

1.1 About the Company

MedTourEasy is a global healthcare company which provides the informational resources needed to evaluate your global options. MedTourEasy provides analytical solutions to our partner healthcare providers globally.

1.2 Project Outline: Objective and Deliverables

With the surge in popularity of fitness trackers, an increasing number of runners globally are leveraging technology, such as smartphones and smartwatches, to collect detailed data on their physical activities. This data serves as a powerful tool for motivation, allowing runners to track and analyse various aspects of their performance. The objective of this project is to systematically import, clean, and analyse seven years' worth of training data, spanning from 2012 through 2018, which was exported from the Runkeeper app. The data, organized in a CSV file where each row represents a single training activity, provides a comprehensive record of the runner's performance over time. By analysing this extensive dataset, the project aims to offer a deep understanding of the runner's performance metrics, identify trends, and assess progress towards achieving set goals. The methods and strategies developed can be applied to other datasets, enabling broader applications for those looking to optimize their fitness routines and track their progress effectively.

The following key questions were addressed and analysed through the course of the project, resulting in the following deliverables:

a. How fast, long, and intense was my run today?
   - Preparation of running data for the last 4 years (from 2015 to 2018).
   - Calculated metrics like average running distance, speed, calories burned, climb, average heart rate (bpm) for those last 4 years.
   - Calculated weekly averages of the last 4 years by studying the same metrics.
   - Determined the estimated trainings per week the runner had on average.

b. Have I succeeded with my training goals?
   - Set the target goal of running 1000 km per year.
   - Visualized annual running distance from 2013 to 2018 and compared against target.

c. Am I progressing?
   - Decompose weekly running distance using statsmodels.api and compare it to the raw data.

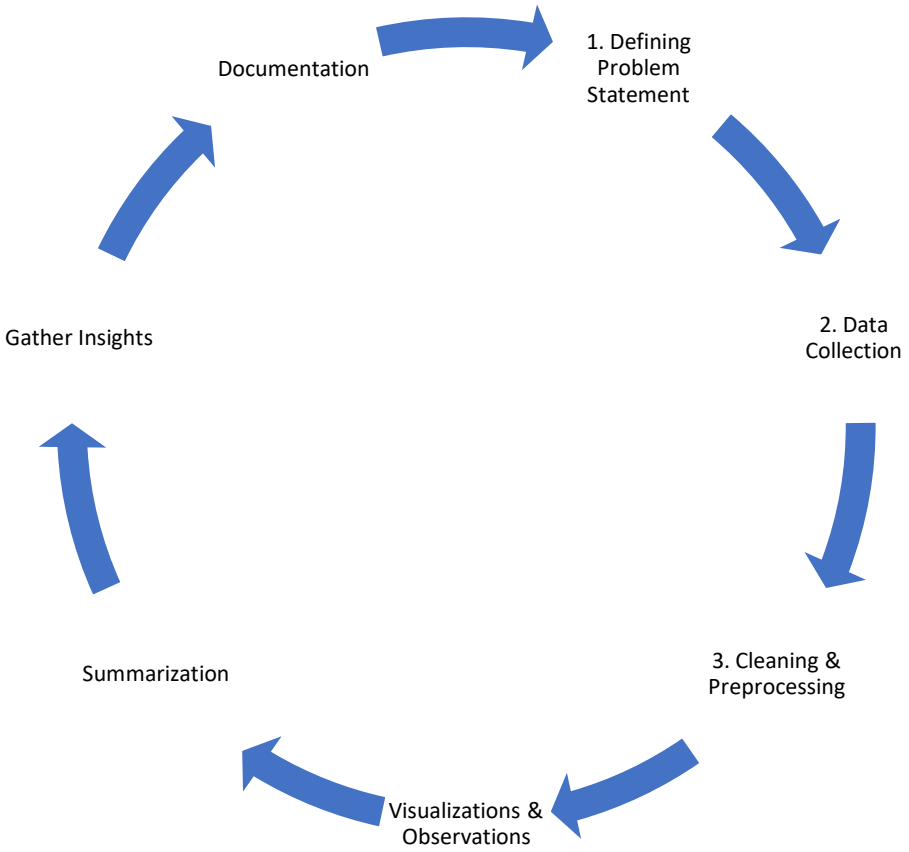
- Distribution plot of heart rate by training intensity to account of average heart rate for a particular training intensity level.

d. What were my best achievements?
    - Created a detailed summary report. Highlight individual best running scores under ‘Fun Facts’ summary. Create the statistical summary using groupby() and stack() functions.
    - Answered some fun questions using this summary. Determined the total number of shoes gone through for Forest Gump’s route.

2. Methodology

2.1 Flow of the Project

The following steps were followed to successfully meet the objectives of the project. Each step is discussed in detail in Section 3.



2.2 Programming Tools

Python

Python is a versatile and widely-used programming language known for its simplicity and readability. It is highly favoured in various fields, including data analytics, due to its extensive libraries and frameworks that make data manipulation, analysis, and visualization more efficient. Python's syntax is intuitive, making it accessible to beginners while being powerful enough for experts to perform complex data-related tasks.

In data analytics, Python is used in the following ways:

- **Data Collection**: Automating data extraction from various sources, including APIs, databases, and web scraping.
- **Data Cleaning and Preprocessing**: Handling missing data, filtering outliers, and transforming raw data into a usable format using libraries like Pandas and NumPy.
- **Exploratory Data Analysis (EDA)**: Summarizing and visualizing data to uncover patterns, trends, and relationships using tools like Matplotlib, Seaborn, and Plotly.
- **Statistical Analysis**: Performing statistical tests, regression analysis, and other mathematical computations to derive insights from data.
- **Machine Learning**: Building predictive models with libraries like Scikit-learn, TensorFlow, and Keras to forecast trends and make data-driven decisions.
- **Data Visualization**: Creating interactive and static visualizations to present data findings clearly and effectively to stakeholders.
- **Automation and Reporting**: Automating repetitive tasks and generating reports that compile insights and analytics into a consumable format.

Jupyter Notebook

Jupyter Notebook is an open-source, web-based interactive computing environment that allows data analysts and scientists to create and share documents that contain live code, equations, visualizations, and narrative text. It is particularly popular in the field of data analytics due to its versatility and ease of use. Jupyter Notebooks provide a seamless interface where analysts can perform data cleaning, exploratory data analysis, statistical modelling, and data visualization all within a single document. The ability to mix code execution with rich text and visual outputs makes it an ideal tool for documenting the entire data analysis process, facilitating collaboration, and producing reproducible research. Its support for various programming languages, especially Python, further enhances its utility in data analytics, enabling users to interactively explore datasets and develop insights in a dynamic and iterative manner.

3. Implementation

3.1 Defining Problem Statement

It is a clear and concise description of the business problem that needs to be addressed. It identifies the gap or problem that exists, the significance of the problem, and the objectives of the analysis.

3.2 Data Collection

Data collection is the method for gathering and measuring information from a variety of sources to obtain a detailed overview of the area of interest. The dataset helps to address specific questions, determine outcomes, and forecast future patterns.

For this project, the data was exported from Runkeeper. The data is a CSV file where each row is a single training activity and each column is a specific training metric. This data was then imported into the coding environment from internal or external sources using **pandas**.

3.3 Cleaning and Preprocessing

Data is a crucial element in both Analytics and Machine Learning. In various computing and business contexts, data is essential. However, in real-world scenarios, data often comes with issues such as incompleteness, inconsistency, or missing values. When data is compromised, it can hinder the process and lead to inaccurate results. Therefore, data cleaning is a fundamental aspect of data science.

Data cleaning involves identifying and addressing incorrect, incomplete, inaccurate, irrelevant, or missing data. This process includes modifying, replacing, or removing data as necessary to ensure its quality and reliability.

With reference to the Runkeeper dataset, it may contain many duplicates, null values or incorrect values which could be a result of inconsistency in data reporting. Hence, we used several important functions to perform data cleaning process.

drop(): This function was used to delete the unnecessary columns from the dataset by simply dropping them.

value_counts(): This function allows us to calculate each activity type counts which was implied on ‘Type’ column.

str.replace(): To replace a certain value to another, we use this function. For our dataset, we rename the ‘Other’ values to ‘Unicycling’ in the ‘Type’ column.

isnull.sum(): The missing values in each column were counted using this function.

```
In [5]: # Define list of columns to be deleted
cols_to_drop = ['Friend\'s Tagged','Route Name','GPX File','Activity Id','Calories Burned', 'Notes']

# Delete unnecessary columns
# ... YOUR CODE FOR TASK 2 ...
df_ativities = df_activities.drop(columns = cols_to_drop)

# Count types of training activities
activity_counts = df_activities['Type'].value_counts()
display(activity_counts)

# Rename 'Other' type to 'Unicycling'
df_activities['Type'] = df_activities['Type'].str.replace('Other', 'Unicycling')

# Count missing values for each column
# ... YOUR CODE FOR TASK 2 ...
missing_values = df_activities.isnull().sum()
display(missing_values)
```

```
Running    459
Cycling     29
Walking     18
Other        2
Name: Type, dtype: int64
```

```
Activity Id                  0
Type                         0
Route Name                 507
Distance (km)                0
Duration                     0
Average Pace                 0
Average Speed (km/h)         0
Calories Burned              0
Climb (m)                    0
Average Heart Rate (bpm)   214
Friend's Tagged            508
Notes                      277
GPX File                     4
dtype: int64
```

As we can see from the last output, there are 214 missing entries for my average heart rate. To address the missing values in the average heart rate (bpm) column, we implemented the mean imputation method. **Mean imputation** is a method in each the null values are replaced by the average value. The process was done by following these steps.

- Calculate the sample mean for Average Heart Rate (bpm) for the 'Cycling' activity type. Assign the result to avg_hr_cycle.
- Filter the df_activities for the 'Cycling' activity type. Create a copy of the result using copy() and assign the copy to df_cycle.
- Fill in the missing values for Average Heart Rate (bpm) in df_cycle with int(avg_hr_cycle) using the fillna() method.
- Count the missing values for all columns in df_run.

```
In [6]: # Calculate sample means for heart rate for each training activity type
        avg_hr_run = df_activities[df_activities['Type'] == 'Running']['Average Heart Rate (bpm)'].mean()
        avg_hr_cycle = df_activities[df_activities['Type'] == 'Cycling']['Average Heart Rate (bpm)'].mean()

        # Split whole DataFrame into several, specific for different activities
        df_run = df_activities[df_activities['Type'] == 'Running'].copy()
        df_walk = df_activities[df_activities['Type'] == 'Walking'].copy()
        df_cycle = df_activities[df_activities['Type'] == 'Cycling'].copy()

        # Filling missing values with counted means
        df_walk['Average Heart Rate (bpm)'].fillna(110, inplace=True)
        df_run['Average Heart Rate (bpm)'].fillna(int(avg_hr_run), inplace=True)
        # ... YOUR CODE FOR TASK 3 ...
        df_cycle['Average Heart Rate (bpm)'].fillna(int(avg_hr_cycle), inplace=True)

        # Count missing values for each column in running data
        # ... YOUR CODE FOR TASK 3 ...
        missing_values_run = df_run.isnull().sum()
        display(missing_values_run)

        Activity Id                   0
        Type                          0
        Route Name                  458
        Distance (km)                 0
        Duration                      0
        Average Pace                  0
        Average Speed (km/h)          0
        Calories Burned               0
        Climb (m)                     0
        Average Heart Rate (bpm)      0
        Friend's Tagged             459
        Notes                       237
        GPX File                      4
        dtype: int64
```

3.4 Visualizations and Observations

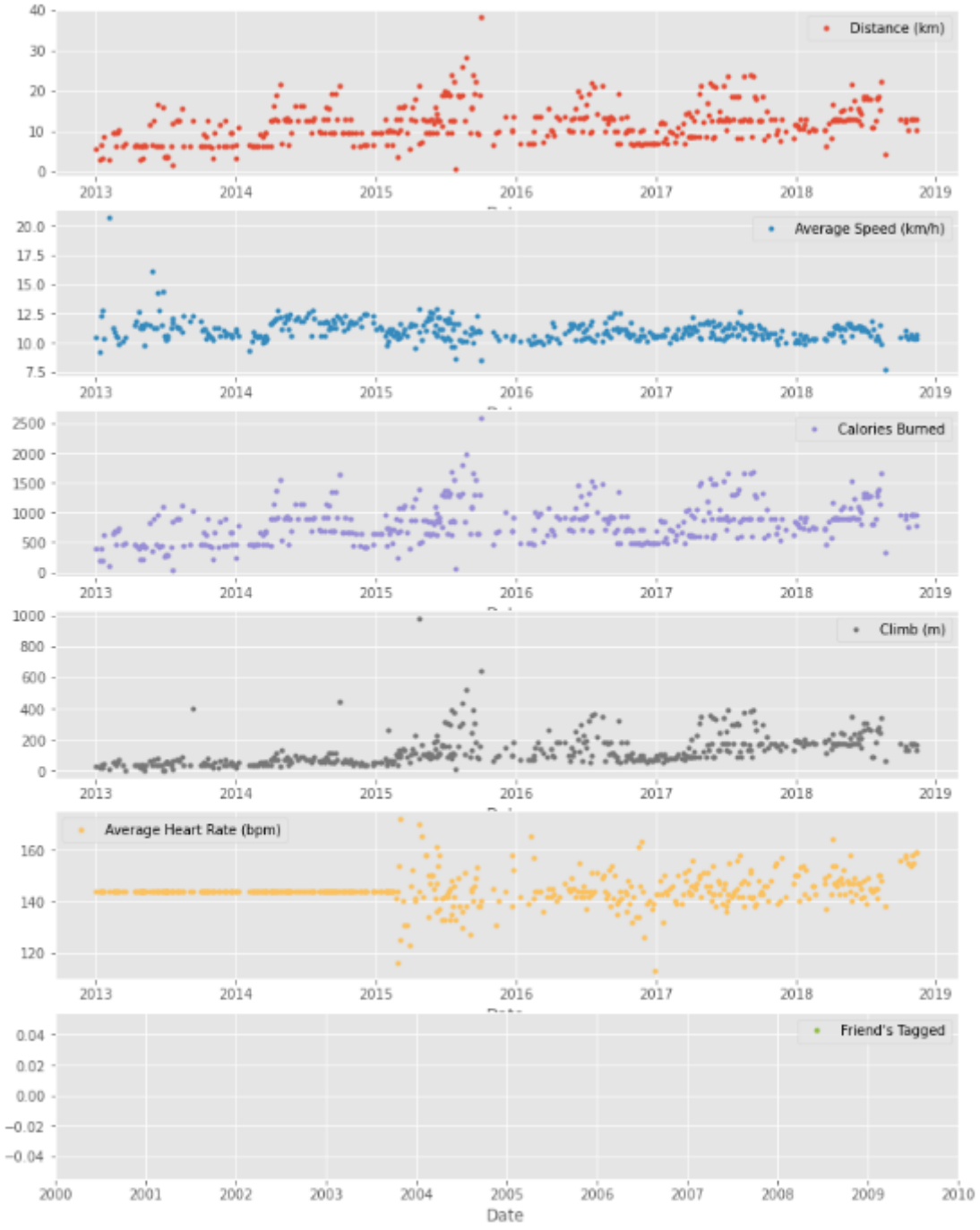
Visualizing the running data:

Since most of the activities in the data were running (459 in number) and only 29, 18, and 2 instances for cycling, walking and unicycling respectively, so the focus would primarily be on plotting the different running metrics.

The visuals were created using four subplots, one for each running metric. Each plot has a common x-axis (Date column) and different y-axis mentioned in the plot legends shown below. The visualizations were performed using **matplotlib.pyplot.**

The plots give us a detailed idea of how the performance of the runner looked like between the year 2013 to 2018.

- The average speed was maintained between 10-12 km/hr.
- The average calories burned were about 1000 cal.
- The average climb was made around 200m.
- The average heart rate remained between 140-160 bpm.



Running Statistics:

Running is known to help people stay mentally and physically healthy and productive at any age. When you are in a group of runners, some of the common questions you hear among them are like: What is your average distance? How fast do you run? Do you measure your heart rate? How often do you train?

In the dataset, since the heart rate was not measured before the year 2015, hence we had a look at the averages from the year 2015 to 2018. The time series span of the data is grouped using resample() function. The resampling is done annually and weekly.

The results of the calculations were as follows.

Average run in the last 4 years:

Date	Distance (km)	Average Speed (km/h)	Calories Burned	Climb (m)	Average Heart Rate (bpm)	Friend's Tagged
2015-12-31	13.602805	10.998902	932.906138	160.170732	143.353659	NaN
2016-12-31	11.411667	10.837778	796.152777	133.194444	143.388889	NaN
2017-12-31	12.935176	10.959059	914.164706	169.376471	145.247059	NaN
2018-12-31	13.339063	10.777969	952.359375	191.218750	148.125000	NaN

Weekly averages of the last 4 years:

Date	Distance (km)	Average Speed (km/h)	Calories Burned	Climb (m)	Average Heart Rate (bpm)	Friend's Tagged
2015-01-04	9.780000	11.120000	654.000000	51.0	144.0	NaN
2015-01-11	NaN	NaN	NaN	NaN	NaN	NaN
2015-01-18	9.780000	11.230000	654.500000	51.0	144.0	NaN
2015-01-25	NaN	NaN	NaN	NaN	NaN	NaN
2015-02-01	9.893333	10.423333	707.624448	58.0	144.0	NaN
...	...	...	...	...	...	...
2018-10-14	12.620000	10.840000	928.000000	146.5	157.5	NaN
2018-10-21	10.290000	10.410000	764.000000	133.0	155.0	NaN
2018-10-28	13.020000	10.730000	967.000000	170.0	154.0	NaN
2018-11-04	12.995000	10.420000	963.500000	170.0	156.5	NaN
2018-11-11	11.640000	10.535000	864.000000	149.0	159.0	NaN

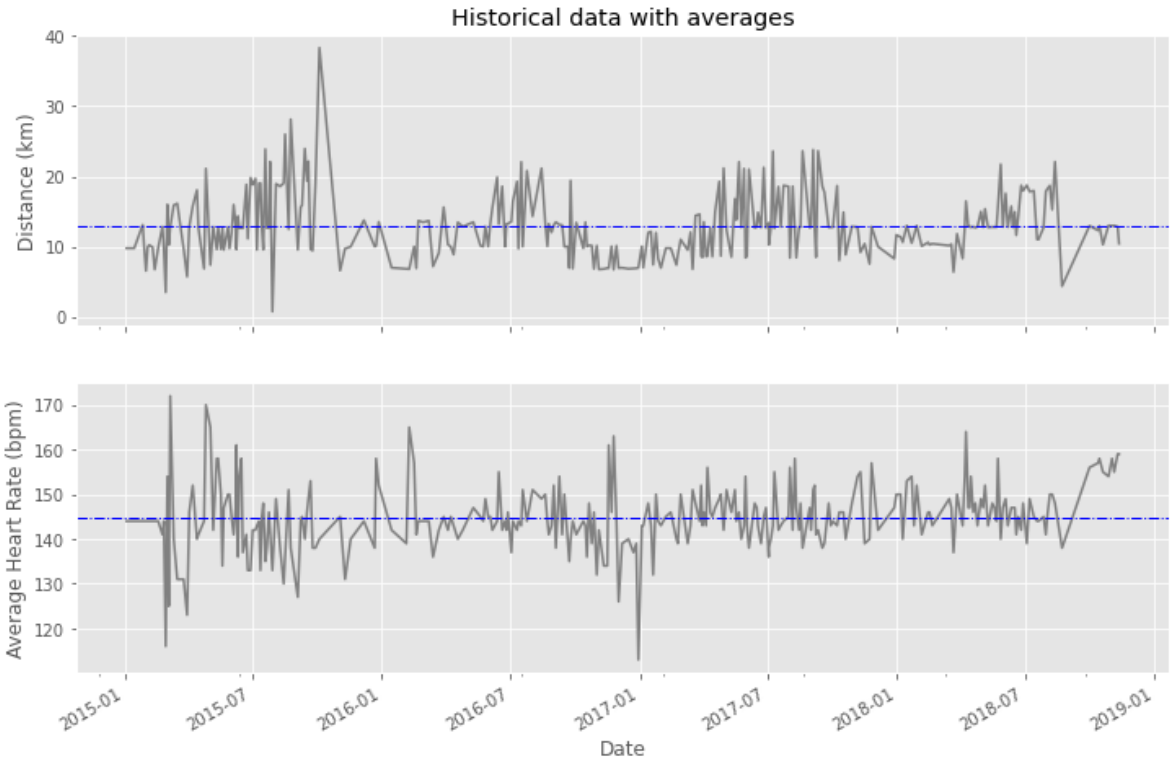
202 rows × 6 columns

Thus, average training per week: **1.5**

Visualization with Averages:

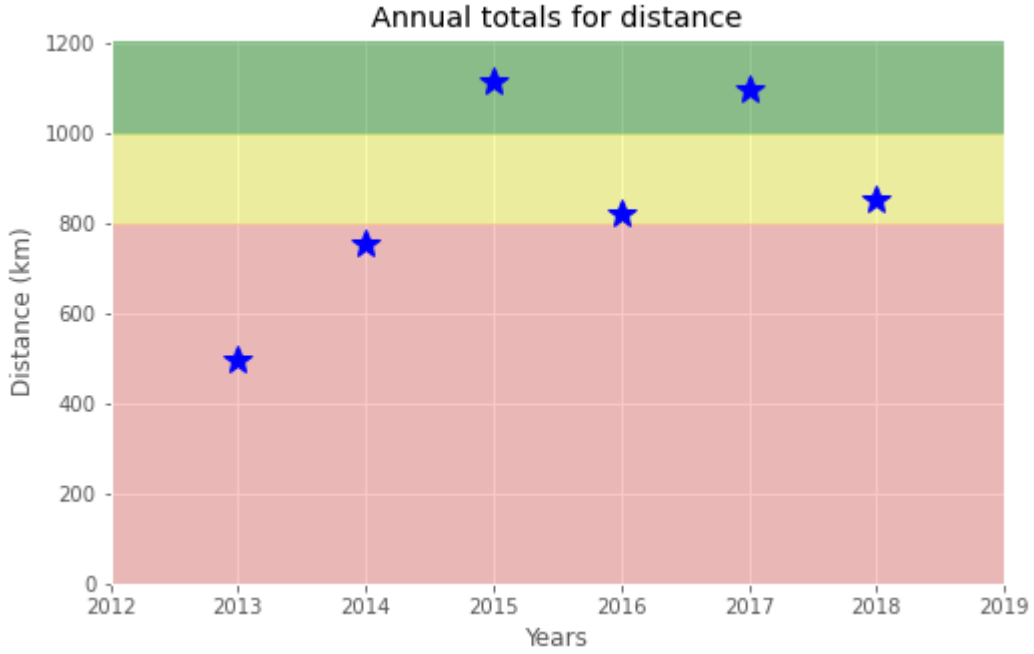
We plotted a long-term average of the runner's distance run and their heart rate with their raw data to visually compare the averages of each training session.

The results show two instances where the heart rate critically dropped below 120 bpm.



Did I reach my goals?

With a running target set of 1000 km per year, we visualized the runner's annual running distance from 2013 to 2018. The stars in the green region indicate success.

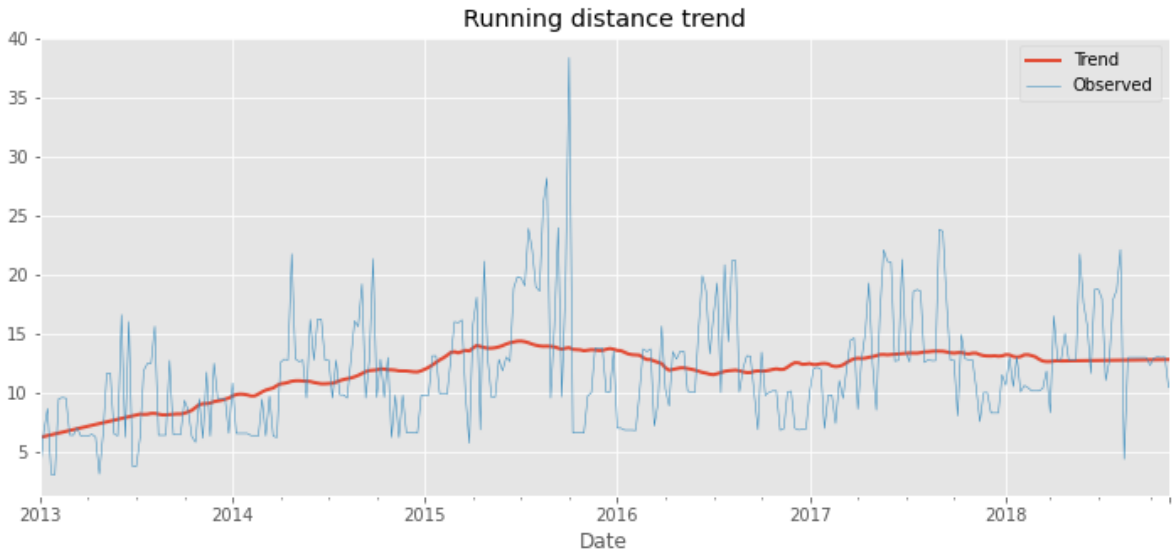


The plot describes that the years 2015 and 2017 were the two successful years where the runner was able to achieve their set goals. The weakest years were the starting ones i.e., 2013 and 2014. The most significant growth was seen between the years 2013 to 2015.

Am I progressing?

In terms of the running skills, is the runner progressing? This is where we attempt to visualize the trend of the running distance to what is observed.
The weekly distance run is decomposed using statsmodel.api library and was compared to the raw data.

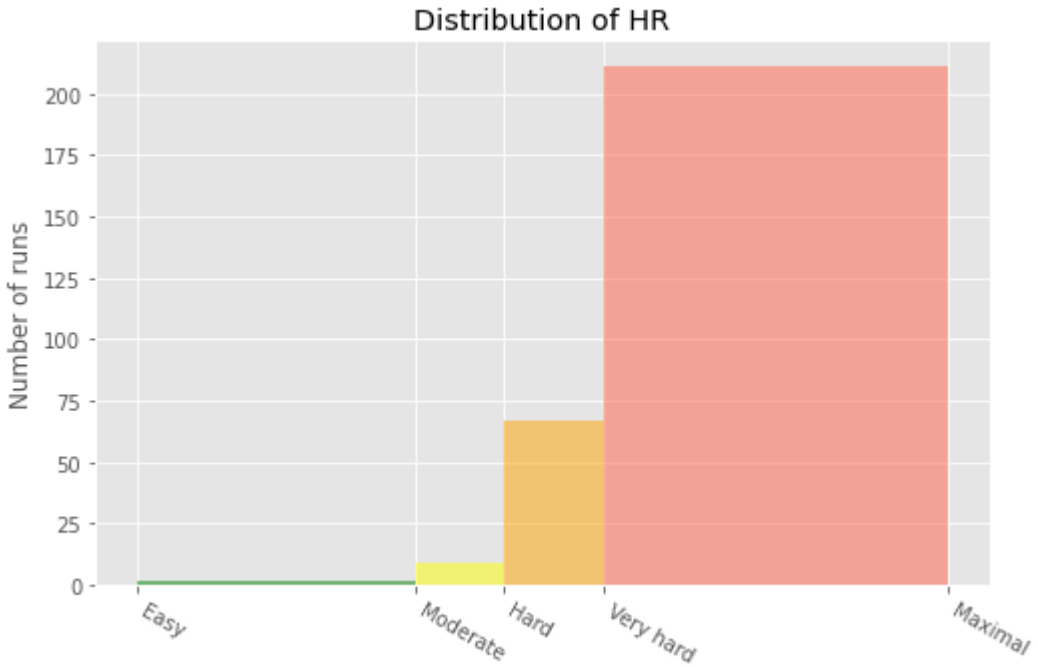
statsmodel.api: This library provides classes and functions for statistical modelling and hypothesis testing. Particularly, the **statsmodel.tsa.seasonal** module is used for time series decomposition.



The trend displays a decent progression with a more stagnant performance in the later years.

Training intensity:

Heart rate is a popular metric used to measure training intensity. Depending on age and fitness level, heart rates are grouped into different zones that people can target depending on training goals. A target heart rate during moderate-intensity activities is about 50-70% of maximum heart rate, while during vigorous physical activity it is about 70-85% of maximum.
We created a distribution plot of the heart rate data by training intensity. It will be a visual presentation for the number of activities from predefined training zones.



3.5 Summarization

A detailed summary table was created of the runner's training data by applying the following steps:

- We created two tables.
- The first table, is a summary of distance and climb variables for each training activity.
- The second table, is a statistical summary for the average speed, climb, and distance variables for each training activity.
- Use the stack() method to display a compact form of full summary report.

Totals for different training types:

Type	Distance (km)	Climb (m)
Cycling	680.58	6976
Running	5224.50	57278
Walking	33.45	349

Summary statistics of different training types:

Type		Average Speed (km/h)	Climb (m)	Distance (km)
Cycling	25%	16.980000	139.000000	15.530000
	50%	19.500000	199.000000	20.300000
	75%	21.490000	318.000000	29.400000
	count	29.000000	29.000000	29.000000
	max	24.330000	553.000000	49.180000
	mean	19.125172	240.551724	23.468276
	min	11.380000	58.000000	11.410000
	std	3.257100	128.960289	9.451040
	total	NaN	6976.000000	680.580000
Running	25%	10.495000	54.000000	7.415000
	50%	10.980000	91.000000	10.810000
	75%	11.520000	171.000000	13.190000
	count	459.000000	459.000000	459.000000
	max	20.720000	982.000000	38.320000
	mean	11.056296	124.788671	11.382353
	min	5.770000	0.000000	0.760000
	std	0.953273	103.382177	4.937853
	total	NaN	57278.000000	5224.500000
Walking	25%	5.555000	7.000000	1.385000
	50%	5.970000	10.000000	1.485000
	75%	6.512500	15.500000	1.787500
	count	18.000000	18.000000	18.000000
	max	6.910000	112.000000	4.290000
	mean	5.549444	19.388889	1.858333
	min	1.040000	5.000000	1.220000
	std	1.459309	27.110100	0.880055
	total	NaN	349.000000	33.450000

3.6 Gather Insights

The following insights were gathered after observing the full summary of the cleaned data.

- The data represents 6 years, 2 months and 21 days of training and the runner specifies that they used 7 pairs of running shoes throughout this running history.
- Average distance covered by the runner was 11.38 km.
- The longest distance covered was 38.32 km.
- The highest climb made was 982 m.
- The total climb made was 57,278 m.
- The total number of km run was 5224 km.
- Total runs done were 459

We also developed an interesting case study of Forest Gump where we figure how many pairs of shoes would Forest Gump require to complete the run that he made.

The average pair of shoes required would be: 5224 // (5224/7) = **6 pairs of shoes!**

4. Conclusion and Future Scope

The "Analysis of Fitness Data" project provided a comprehensive look into the various aspects of a runner's performance over a period of seven years, using data exported from the Runkeeper app. The project effectively demonstrated the importance of tracking fitness metrics to maintain a healthy lifestyle and achieve fitness goals. Through systematic data cleaning, preprocessing, and visualization, the project was able to answer critical questions about the runner's performance, including speed, distance, heart rate, and goal achievement.

The insights gained from the data, such as identifying trends in running distances and heart rate, assessing progress towards goals, and highlighting the runner's best achievements, underscore the value of data-driven fitness monitoring. The methodologies used in this project, including mean imputation, resampling, and statistical modelling, provide a robust framework for analysing similar datasets in the future.

Overall, this project not only highlights the significance of fitness tracking but also showcases how data analytics can be applied to optimize performance and provide actionable insights for individuals aiming to improve their health and fitness.

The "Analysis of Fitness Data" project has laid a strong foundation for further exploration and analysis in the field of fitness tracking and data analytics. The future scope of this project includes:

1. **Development of Interactive Dashboards**:
    - Creating interactive dashboards to provide real-time insights into fitness metrics. These dashboards can allow users to track their progress, set new goals, and visualize trends over time, making the data more accessible and actionable.
2. **Integration with Other Fitness Data Sources**:
    - Expanding the dataset by integrating data from other fitness apps and wearables to provide a more comprehensive view of the user's overall fitness and health. This could include sleep data, diet tracking, and other health metrics.
3. **Advanced Predictive Modelling**:
    - Implementing machine learning models to predict future performance based on past data. For example, predicting the likelihood of achieving a specific running goal or identifying potential injuries based on training intensity and frequency.
4. **Personalized Fitness Recommendations**:
    - Developing algorithms that provide personalized fitness recommendations based on the user’s historical data, such as suggested workout plans, recovery periods, and nutrition advice.
5. **Exploration of New Metrics**:
    - Exploring new metrics, such as stress levels or recovery times, to gain deeper insights into the runner's overall well-being and performance. This could

involve incorporating data from heart rate variability (HRV) or other advanced fitness metrics.

6. **Expanding the Scope to Other Sports**:
    - Adapting the analysis framework to other sports and activities, such as cycling, swimming, or strength training, to provide a broader application of the methodologies developed in this project.

5. References

Data Collection:

https://drive.google.com/uc?export=download&id=1O--TsE3O2orEDieV7tU2pp0ndMTYekQB

Jupyter Notebook:

https://drive.google.com/file/d/1DjI2jogKGMrPsgGQ05H25i0M37smFt1J/view?usp=sharing