

```
In [1]: import pandas as pd
```

```
In [2]: daily_data = pd.read_csv('C:/Users/Rajendra/Desktop/Data Analysis/Labmentix Inte
```

```
In [3]: hourly_data = pd.read_csv('C:/Users/Rajendra/Desktop/Data Analysis/Labmentix Int
```

```
In [4]: print("Daily data shape:", daily_data.shape)
```

Daily data shape: (947, 18)

```
In [5]: daily_data.sort_values(by='ActivityDate', inplace=True)
```

```
In [6]: print("Daily data columns:", daily_data.columns)
```

Daily data columns: Index(['Id', 'Region', 'ActivityDate', 'TotalSteps', 'TotalDistance\_Kms',  
 'DailyCalories', 'ExtraCalories', 'SedentaryMinutes',  
 'LightlyActiveMinutes', 'FairlyActiveMinutes', 'VeryActiveMinutes',  
 'SedentaryActiveDistance\_Kms', 'LightActiveDistance\_Kms',  
 'ModeratelyActiveDistance\_Kms', 'VeryActiveDistance\_Kms',  
 'TotalMinutesAsleep', 'WeightKg', 'BMI'],  
 dtype='object')

```
In [7]: print("Hourly data shape:", hourly_data.shape)
```

Hourly data shape: (12840, 7)

```
In [8]: print("Hourly data columns:", hourly_data.columns)
```

Hourly data columns: Index(['Id', 'Region', 'ActivityHour', 'StepTotal', 'Calories',  
 'TotalIntensity', 'AverageIntensity'],  
 dtype='object')

```
In [9]: print("\n Daily Data - Head") # Quick Look at daily data  
print(daily_data.head())
```

## Daily Data - Head

	Id	Region	ActivityDate	TotalSteps	TotalDistance_Kms	\
763	7086361926	East	2016-01-05	12390	8.07	
844	8378563200	West	2016-01-05	11419	9.06	
322	3372868164	East	2016-01-05	3077	2.10	
593	5577150313	North	2016-01-05	13368	9.99	
562	5553957443	South	2016-01-05	5164	3.37	

	DailyCalories	ExtraCalories	SedentaryMinutes	LightlyActiveMinutes	\
763	2730	2730	685	258	
844	3369	3369	669	127	
322	1237	1237	842	172	
593	4546	4546	499	178	
562	1747	1747	436	237	

	FairlyActiveMinutes	VeryActiveMinutes	SedentaryActiveDistance_Kms	\
763	15	30	0.0	
844	10	71	0.0	
322	0	0	0.0	
593	72	194	0.0	
562	0	0	0.0	

	LightActiveDistance_Kms	ModeratelyActiveDistance_Kms	\
763	4.85	0.90	
844	2.47	0.56	
322	2.09	0.00	
593	3.24	1.44	
562	3.37	0.00	

	VeryActiveDistance_Kms	TotalMinutesAsleep	WeightKg	BMI
763	2.30	0	0	0
844	6.03	0	0	0
322	0.00	0	0	0
593	5.31	0	0	0
562	0.00	0	0	0

```
In [10]: print("\n Daily Data - Summary Stats")
          print(daily_data.describe())
```

## Daily Data - Summary Stats

	Id	TotalSteps	TotalDistance_Kms	DailyCalories	\
count	9.470000e+02	947.000000	947.000000	947.000000	
mean	4.839669e+09	7643.164731	5.490517	2300.801478	
std	2.422897e+09	5069.010887	3.910297	716.280526	
min	1.503960e+09	0.000000	0.000000	0.000000	
25%	2.320127e+09	3804.500000	2.625000	1830.000000	
50%	4.445115e+09	7443.000000	5.270000	2130.000000	
75%	6.962181e+09	10709.500000	7.705000	2785.500000	
max	8.877689e+09	36019.000000	28.030000	4900.000000	

	ExtraCalories	SedentaryMinutes	LightlyActiveMinutes	\
count	947.000000	947.000000	947.000000	
mean	2300.801478	992.104541	193.658923	
std	716.280526	300.359539	109.301882	
min	0.000000	0.000000	0.000000	
25%	1830.000000	730.500000	127.500000	
50%	2130.000000	1061.000000	199.000000	
75%	2785.500000	1227.500000	265.500000	
max	4900.000000	1440.000000	518.000000	

	FairlyActiveMinutes	VeryActiveMinutes	SedentaryActiveDistance_Kms	\
count	947.000000	947.000000	947.000000	
mean	13.594509	21.030623	0.001626	
std	19.925875	32.760456	0.007335	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	7.000000	4.000000	0.000000	
75%	20.000000	31.500000	0.000000	
max	143.000000	210.000000	0.110000	

	LightActiveDistance_Kms	ModeratelyActiveDistance_Kms	\
count	947.000000	947.000000	
mean	3.349757	0.569715	
std	2.035885	0.881205	
min	0.000000	0.000000	
25%	1.955000	0.000000	
50%	3.390000	0.240000	
75%	4.790000	0.810000	
max	10.710000	6.480000	

	VeryActiveDistance_Kms	TotalMinutesAsleep	WeightKg	BMI
count	947.000000	947.000000	947.000000	947.000000
mean	1.492967	0.990496	0.111932	0.048574
std	2.651501	21.645445	2.434368	1.056424
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.210000	0.000000	0.000000	0.000000
75%	2.030000	0.000000	0.000000	0.000000
max	21.920000	515.000000	53.000000	23.000000

```
In [11]: print("\n Daily Data - Missing Values")
print(daily_data.isnull().sum())
```

## Daily Data - Missing Values

```

Id                                0
Region                            0
ActivityDate                      0
TotalSteps                       0
TotalDistance_Kms                0
DailyCalories                    0
ExtraCalories                    0
SedentaryMinutes                 0
LightlyActiveMinutes            0
FairlyActiveMinutes             0
VeryActiveMinutes               0
SedentaryActiveDistance_Kms     0
LightActiveDistance_Kms        0
ModeratelyActiveDistance_Kms   0
VeryActiveDistance_Kms         0
TotalMinutesAsleep              0
WeightKg                        0
BMI                             0
dtype: int64

```

```

In [12]: print("\n Hourly Data - Head")
         print(hourly_data.head()) # Quick Look at hourly data

```

```

Hourly Data - Head
      Id Region      ActivityHour  StepTotal  Calories  \
0  1503960366  South  2016-05-05 00:00:00         70         56
1  1503960366  South  2016-05-05 01:00:00        113         58
2  1503960366  South  2016-05-05 02:00:00         19         52
3  1503960366  South  2016-05-05 03:00:00          0         47
4  1503960366  South  2016-05-05 04:00:00         44         53

      TotalIntensity  AverageIntensity
0                   6                  0
1                   6                  0
2                   2                  0
3                   0                  0
4                   4                  0

```

```

In [13]: print("\n Hourly Data - Summary Stats")
         print(hourly_data.describe())

```

```

Hourly Data - Summary Stats
      Id      StepTotal      Calories      TotalIntensity \
count  1.284000e+04  12840.000000  12840.000000  12840.000000
mean   4.127640e+09   308.840265    94.099455    1.043925
std    2.254277e+09   664.384803    55.786812    6.775397
min    1.503960e+09    0.000000    0.000000    0.000000
25%    2.022484e+09    0.000000    62.000000    0.000000
50%    4.020333e+09   36.000000    83.000000    0.000000
75%    5.553957e+09   352.000000   105.000000    0.000000
max    8.877689e+09   9769.000000   843.000000   165.000000

      AverageIntensity
count      12840.000000
mean         0.012617
std          0.124171
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          3.000000

```

```

In [14]: print("\n Hourly Data - Missing Values")
         print(hourly_data.isnull().sum())

```

```

Hourly Data - Missing Values
Id      0
Region  2289
ActivityHour  0
StepTotal  0
Calories  0
TotalIntensity  0
AverageIntensity  0
dtype: int64

```

```

In [15]: import matplotlib.pyplot as plt # Convert the ActivityDate to datetime

```

```

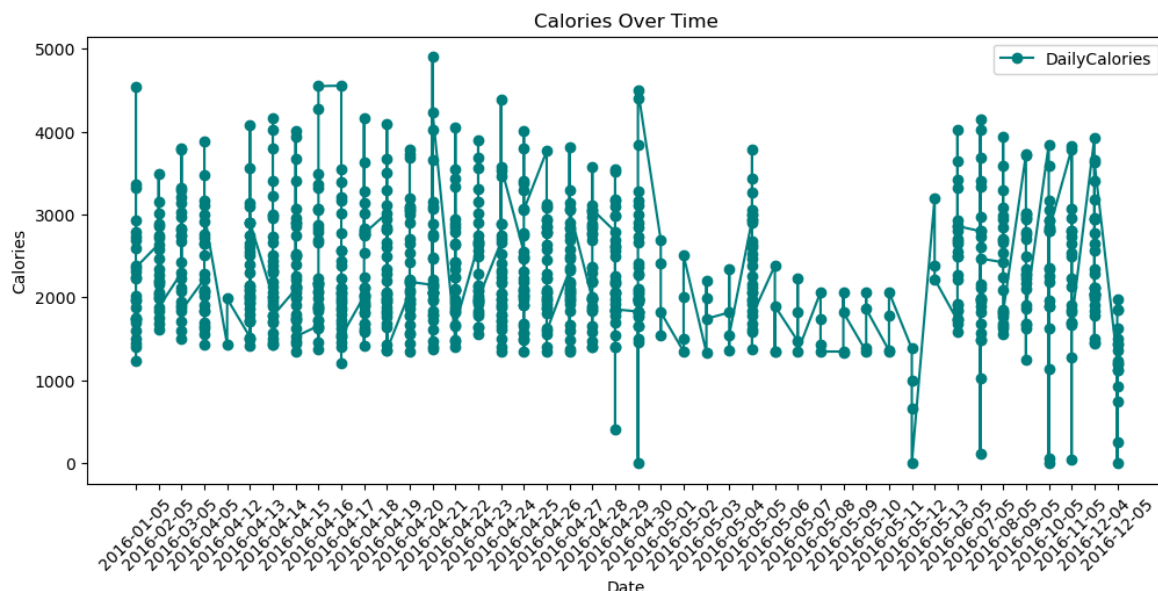
In [16]: # Plot Total Steps over time

```

```

In [17]: plt.figure(figsize=(12, 5))
         plt.plot(daily_data['ActivityDate'], daily_data['DailyCalories'], color='teal',
         plt.xlabel('Date')
         plt.ylabel('Calories')
         plt.title('Calories Over Time')
         plt.legend()
         plt.xticks(rotation=45)
         plt.show()

```



```
In [18]: hourly_data['ActivityHour'] = pd.to_datetime(hourly_data['ActivityHour'])
```

```
In [19]: # Count unique users per day
users_per_day = hourly_data.groupby(hourly_data['ActivityHour'].dt.date)['Id'].n
users_per_day.columns = ['Date', 'UserCount']
```

```
In [20]: # Convert ActivityHour to datetime if not already
hourly_data['ActivityHour'] = pd.to_datetime(hourly_data['ActivityHour'])
```

```
In [21]: # Create ActiveHours flag
hourly_data['ActiveFlag'] = hourly_data['StepTotal'].apply(lambda x: 1 if x > 0
```

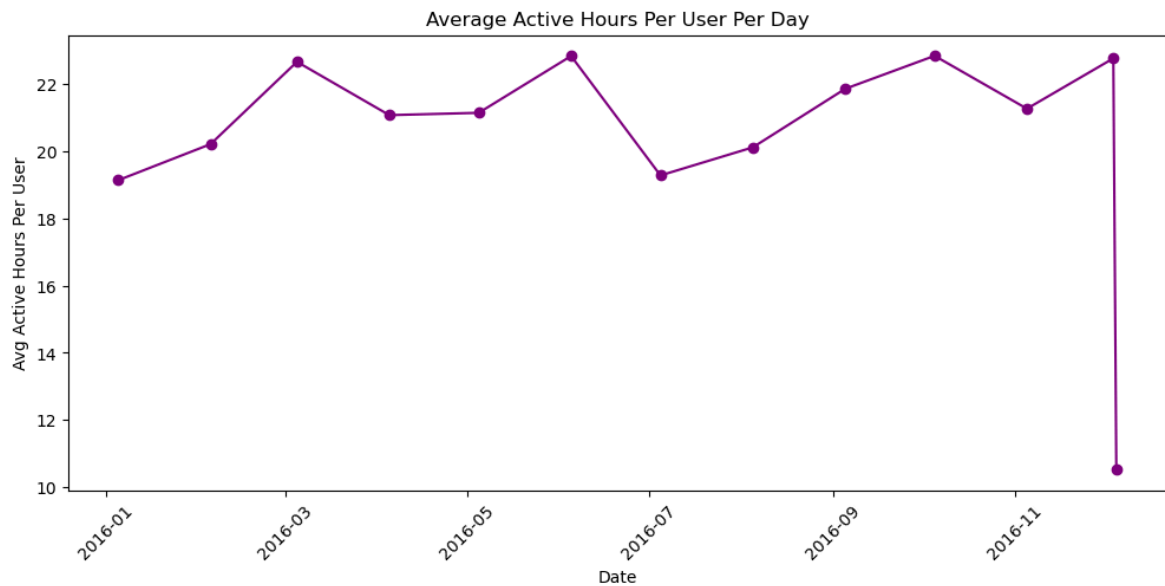
```
In [22]: # Calculate Active Hours per Day
active_hours_per_day = hourly_data.groupby(hourly_data['ActivityHour'].dt.date)[
active_hours_per_day.columns = ['Date', 'ActiveHours']
```

```
In [23]: # Merge with active hours
active_hours_per_day = active_hours_per_day.merge(users_per_day, on='Date')
active_hours_per_day['AvgActiveHoursPerUser'] = active_hours_per_day['ActiveHour
```

```
In [24]: print(active_hours_per_day.head())
```

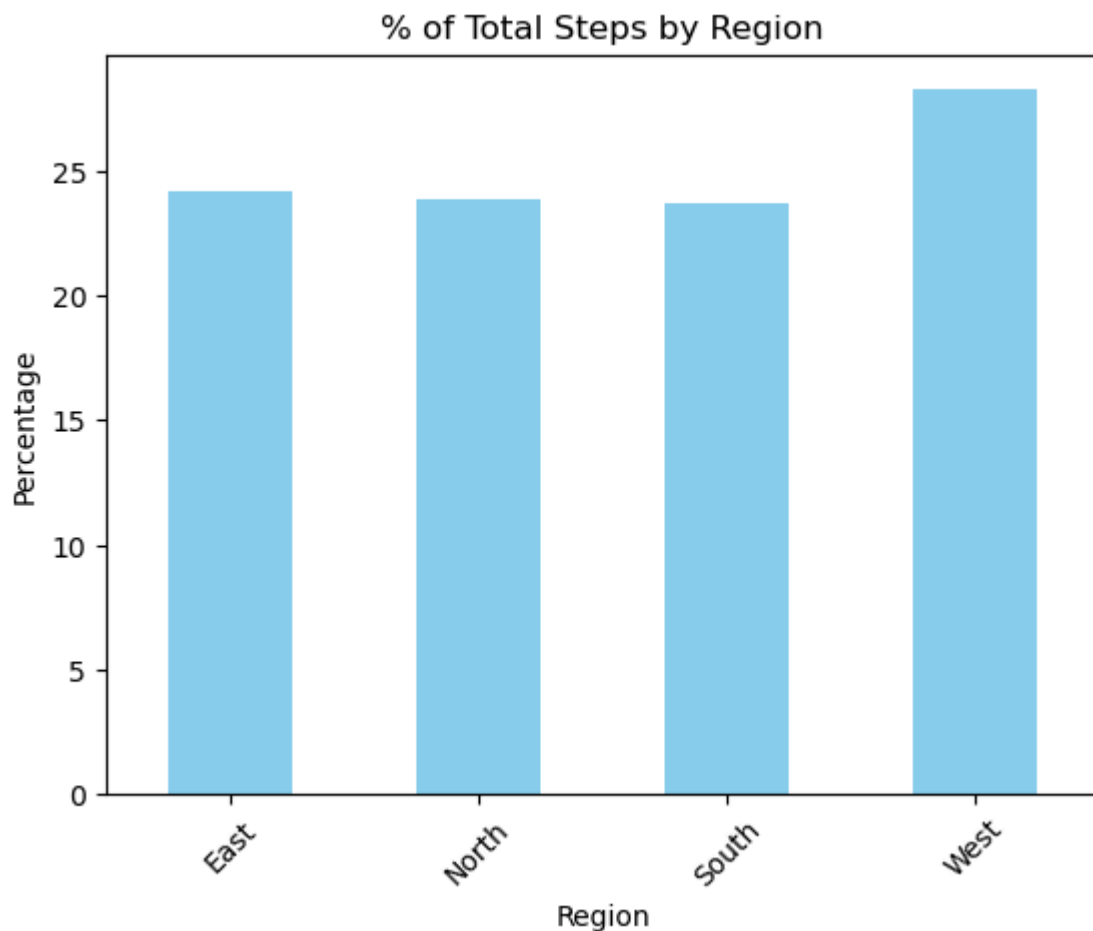
	Date	ActiveHours	UserCount	AvgActiveHoursPerUser
0	2016-01-05	574	30	19.133333
1	2016-02-05	586	29	20.206897
2	2016-03-05	657	29	22.655172
3	2016-04-05	611	29	21.068966
4	2016-05-05	613	29	21.137931

```
In [25]: # Plot
plt.figure(figsize=(12, 5))
plt.plot(active_hours_per_day['Date'], active_hours_per_day['AvgActiveHoursPerUs
plt.xlabel('Date')
plt.ylabel('Avg Active Hours Per User')
plt.title('Average Active Hours Per User Per Day')
plt.xticks(rotation=45)
plt.show()
```



```
In [26]: # % of Total Steps by Region
region_steps = daily_data.groupby("Region")["TotalSteps"].sum()
region_percent = round(region_steps / region_steps.sum() * 100, 2)
```

```
In [27]: region_percent.plot(kind='bar', color='skyblue', title='% of Total Steps by Region')
plt.ylabel('Percentage')
plt.xticks(rotation=45)
plt.show()
```



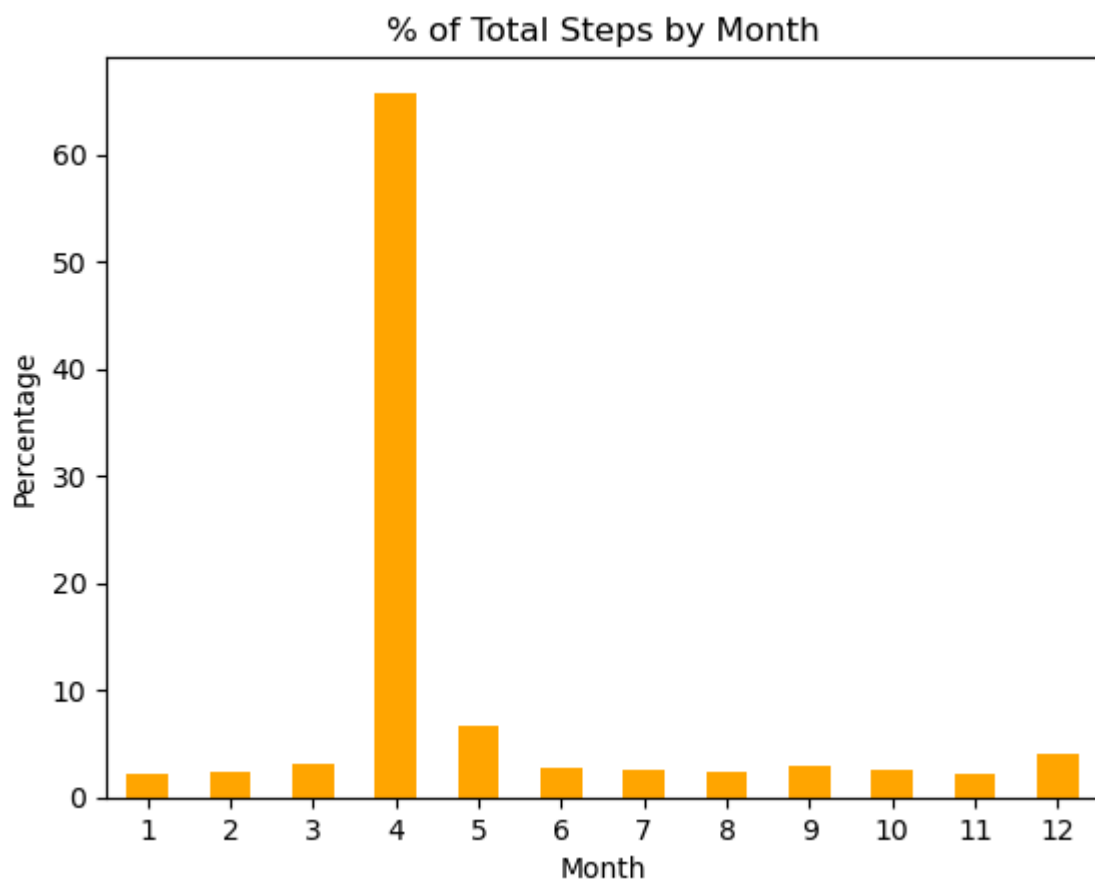
```
In [30]: # % of Users Averaging 10,000+ Steps
user_avg_steps = daily_data.groupby("Id")["TotalSteps"].mean()
```

```
highly_active = (user_avg_steps >= 10000).sum()
total_users = user_avg_steps.count()
```

```
In [31]: active_percent = round(highly_active / total_users * 100,2)
print(f"{active_percent}% of users are highly active(10000+ steps/day)")
```

21.21% of users are highly active(10000+ steps/day)

```
In [32]: # % of Total Steps by Month
daily_data['Month'] = pd.to_datetime(daily_data['ActivityDate']).dt.month
monthly_steps = daily_data.groupby("Month")["TotalSteps"].sum()
monthly_percent = round(monthly_steps / monthly_steps.sum() * 100, 2)
monthly_percent.plot(kind='bar', color='orange', title='% of Total Steps by Month')
plt.ylabel('Percentage')
plt.xticks(rotation=0)
plt.show()
```

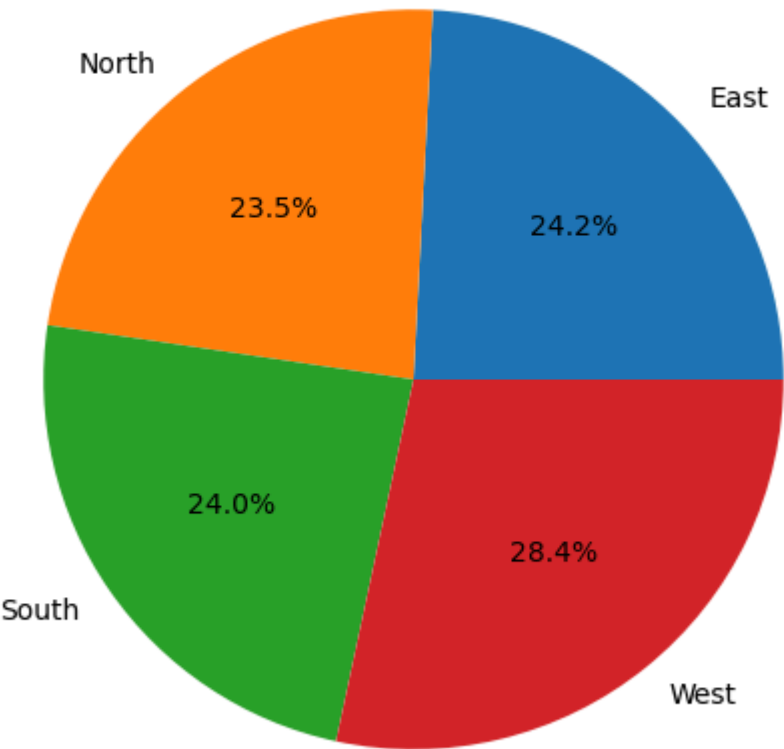


```
In [35]: # % of Calories by Region
region_cals = daily_data.groupby("Region")["DailyCalories"].sum()
region_cal_percent = round(region_cals / region_cals.sum() * 100,2)

region_cal_percent.plot(kind='pie', autopct='%1.1f%%', title='Calories Burned by Region')
plt.ylabel('')
plt.show()
```



Calories Burned by Region



```
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```