Colab Link: https://colab.research.google.com/drive/1jY9qyikv6Sqju0qYzgmJZyDYyWufxqr1?usp=sharing

```
import pandas as pd
```

```
!gdown 1XbB_dq6tH1D16Lg1OiKLXJuitQ8FdcvP
```

```
Downloading...
From: https://drive.google.com/uc?id=1XbB_dq6tH1D16Lg1OiKLXJuitQ8FdcvP
To: /content/weather.csv
100% 12.0k/12.0k [00:00<00:00, 16.7MB/s]
```

```
!wget "https://drive.google.com/uc?export=download&id=1XbB_dq6tH1D16Lg1OiKLXJuitQ8F
```

```
--2022-05-18 15:39:31--  https://drive.google.com/uc?export=download&id=1XbB_c
Resolving drive.google.com (drive.google.com)... 142.250.141.102, 142.250.141.
Connecting to drive.google.com (drive.google.com)|142.250.141.102|:443... conr
HTTP request sent, awaiting response... 303 See Other
Location: https://doc-0o-14-docs.googleusercontent.com/docs/securesc/ha0ro937c
Warning: wildcards not supported in HTTP.
--2022-05-18 15:39:31--  https://doc-0o-14-docs.googleusercontent.com/docs/sec
Resolving doc-0o-14-docs.googleusercontent.com (doc-0o-14-docs.googleuserconte
Connecting to doc-0o-14-docs.googleusercontent.com (doc-0o-14-docs.googleuserc
HTTP request sent, awaiting response... 200 OK
Length: 11982 (12K) [text/csv]
Saving to: 'weather.csv'

weather.csv          100%[===================>]  11.70K  --.-KB/s    in 0s

                                         /s) - 'weather.csv' saved [11982/11982]
```

Saving...

```
weather = pd.read_csv("weather.csv")
```

```
weather.head()
```

	year	month	element	day1	day2	day3	day4	day5	d
0	2018	1	max	17.573016	19.796815	22.412495	17.813163	20.165825	17.060
1	2018	1	min	22.725760	21.007865	17.730792	18.045290	20.766734	18.656
2	2018	2	max	19.015120	19.261805	17.510713	21.080425	17.915749	19.082
3	2018	2	min	18.653843	22.818600	21.842673	21.958159	22.523078	18.535
4	2018	3	max	20.741115	19.704016	17.039811	20.703908	22.714125	17.205

5 rows × 34 columns

```
weather.shape # wide data, or rectangular data
```

```
(22, 34)
```

```
weather_melt = pd.melt(weather, id_vars=["year", "month", "element"],
        var_name="day",value_name="temp")
```

```
weather_tidy = weather_melt.pivot_table(index=["year", "month", "day"], columns="el
```

```
# melt --> melt the columns into a single column, columns ---> column values in row
# pivot_table --> convets values of a column into seperate columns
```

```
weather_tidy.to_csv("weather_tidy.csv", sep=",")
```

```
!ls
```

```
sample_data  weather.csv  weather_tidy.csv
```

```
# Uber use-case - how to handle timestamp data
```

```
!gdown 1TL2hWkMWtD1ExVgaQhWP6A2swR8F8cVB
```

```
Downloading...
From: https://drive.google.com/uc?id=1TL2hWkMWtD1ExVgaQhWP6A2swR8F8cVB
To: /content/UberDrives.csv
100% 86.4k/86.4k [00:00<00:00, 70.5MB/s]
```

Saving... ×

```
v")
```

```
data.head()
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
	1/5/2016	1/5/2016		Fort			

```
data.shape
```

```
(1156, 7)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1156 entries, 0 to 1155
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   START_DATE*  1156 non-null   object 
 1   END_DATE*    1155 non-null   object 
 2   CATEGORY*    1155 non-null   object 
 3   START*       1155 non-null   object 
 4   STOP*        1155 non-null   object 
 5   MILES*       1156 non-null   float64
 6   PURPOSE*     653 non-null    object 
dtypes: float64(1), object(6)
memory usage: 63.3+ KB
```

```
data.tail()
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/2016 21:32	12/31/2016 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site
	12/31/2016	12/31/2016					Temporary

```
data.drop(1155, axis=0, inplace=True)
```

Saving…

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Kar?chi	Kar?chi	0.7	Meeting
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
	12/31/2016	12/31/2016					Temporary

```
data.isnull().sum(axis=0)
```

```
START_DATE*      0
END_DATE*        0
CATEGORY*        0
START*           0
STOP*            0
MILES*           0
PURPOSE*       502
dtype: int64
```

```
data.describe()
```

	MILES*
count	1155.000000
mean	10.566840
std	21.579106
min	0.500000
25%	2.900000
50%	6.000000
75%	10.400000
max	310.300000

```
data.describe(include="object")
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	PURPOSE*
count	1155	1155	1155	1155	1155	653
unique	1154	1154	2	177	188	10
top	6/28/2016 23:34	6/28/2016 23:59	Business	Cary	Cary	Meeting
freq	2	2	1078	201	203	187

Saving...

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
492	6/28/2016 23:34	6/28/2016 23:59	Business	Durham	Cary	9.9	Meeting

```
data.drop_duplicates(inplace=True)
```

```
data.head()
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
	1/5/2016	1/5/2016		Fort			

```
data.columns = [col_name[:-1] for col_name in data.columns]
```

```
data
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit
...	...	...	...	...	...	...	...
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Kar?chi	Kar?chi	0.7	Meeting
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site

```
# how to handle timestamp data
```

```
data["START_DATE"] = pd.to_datetime(data["START_DATE"])
```

Saving… ×

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	2016-01-01 21:11:00	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	2016-01-02 01:25:00	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	2016-01-02 20:25:00	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
	2016-01-05	1/5/2016		Fort			

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   START_DATE  1154 non-null   datetime64[ns]
 1   END_DATE    1154 non-null   object
 2   CATEGORY    1154 non-null   object
 3   START       1154 non-null   object
 4   STOP        1154 non-null   object
 5   MILES       1154 non-null   float64
 6   PURPOSE     652 non-null    object
dtypes: datetime64[ns](1), float64(1), object(5)
memory usage: 72.1+ KB
```

```
data["END_DATE"] = pd.to_datetime(data["END_DATE"])
```

```
data
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit
...	...	...	...	...	...	...	...
1150	2016-12-31 01:07:00	2016-12-31 01:14:00	Business	Kar?chi	Kar?chi	0.7	Meeting
	13:24:00	13:42:00	Business	Kar?chi	Unknown Location	3.9	Temporary Site

Saving…

```
data.loc[data["START_DATE"] == data["END_DATE"]]
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
751	2016-09-06 17:49:00	2016-09-06 17:49:00	Business	Unknown Location	Unknown Location	69.1	NaN
761	2016-09-16 07:08:00	2016-09-16 07:08:00	Business	Unknown Location	Unknown Location	1.6	NaN
798	2016-10-08 15:03:00	2016-10-08 15:03:00	Business	Karachi	Karachi	3.6	NaN

```
data.loc[data["START_DATE"] == data["END_DATE"]].index
```

```
Int64Index([751, 761, 798, 807], dtype='int64')
```

```
data.drop([751, 761, 798, 807], inplace=True, axis=0)
data.shape
```

```
(1150, 7)
```

```
ts = data['START_DATE'][0]
ts
```

```
Timestamp('2016-01-01 21:11:00')
```

```
ts.year
```

```
2016
```

```
ts.month
```

```
1
```

```
ts.day
```

```
1
```

```
ts.month_name()
```

```
'January'
```

```
ts.day_name()
```

```
'Friday'
```

Saving…

```
21
```

```
data['END_DATE'].dt.year
```

```
0       2016
1       2016
2       2016
3       2016
4       2016
        ...
1150    2016
1151    2016
1152    2016
1153    2016
1154    2016
Name: END_DATE, Length: 1150, dtype: int64
```

```
data['END_DATE'].dt.month_name()
```

```
0        January
```

```
1         January
2         January
3         January
4         January
            ...
1150     December
1151     December
1152     December
1153     December
1154     December
Name: END_DATE, Length: 1150, dtype: object
```

```
# What is the shortest journey made? - miles
# What is the longest journey made? - miles
# What is the average journey made? - miles
# How many years of data do we have? - count
```

```
data.describe()
```

	MILES
count	1150.000000
mean	10.538957
std	21.552360
min	0.500000
25%	2.900000
50%	6.000000
75%	10.400000

Saving…

```
data["MILES"].mean()
```

```
10.538956521739115
```

```
data["START_DATE"].dt.year.nunique()
```

```
1
```

```
data["END_DATE"].dt.year.nunique()
```

```
1
```

```
# Basic Deacriptive Statistics
```

```
# Measures of central tendency
```

```
data["MILES"].mean()

     10.538956521739115
```

```
data["MILES"].min(), data["MILES"].max()

     (0.5, 310.3)
```

```
# robust estimator to extreme values
```

```
data["MILES"].median()

     6.0
```

```
# mean, median, mode
```

```
data["PURPOSE"].value_counts(dropna=False)

     NaN                 498
     Meeting             186
     Meal/Entertain      160
     Errand/Supplies     128
     Customer Visit      101
     Temporary Site       50
     Between Offices      18
     Moving                4
     Airport/Travel        3
     Charity ($)           1
     Commute               1
     Name: PURPOSE, dtype: int64
```

Saving…

```
data["PURPOSE"].mode()

     0    Meeting
     dtype: object
```

```
# Measures of dispersion/variability
```

```
data["MILES"].std()

     21.552359680264498
```

```
from scipy import stats
```

```
stats.median_absolute_deviation(data["MILES"])

     5.337359999999999
```

```
# estimate of percentiles
```

```
import numpy as np
np.percentile(data['MILES'], 50)

    6.0
```

```
np.percentile(data['MILES'], 30)

    3.2
```

```
np.percentile(data['MILES'], 98)

    63.79399999999992
```

```
# mean abs dev, mean sq deviation (variance), standard deviation,
```

```
# IQR - Interquartile Range - 75th percentile - 25th percentile - Q3-Q1
np.percentile(data['MILES'], 75) - np.percentile(data['MILES'], 25)

    7.5
```

```
# Outlier Decection - 1.5*IQR
```

```
# [Q1 - 1.5*IQR, Q3 + 1.5*IQR]
```

Saving... ×

✓ 0s completed at 23:16 ● ×