TECHNICAL UNIVERSITY OF MUNICH

CHAIR OF TRANSPORTATION SYSTEMS ENGINEERING

STATISTICAL LEARNING AND DATA ANALYTICS FOR TRANSPORTATION SYSTEMS

PROBLEM SET 3

Author	*Matriculation Number*
NATALIA ALLMI	03755024

PROBLEM 1

Identify if the following problems are within the scope of statistical learning. If it is a statistical learning problem, please indicate the characteristics of the problem (e.g., supervised or unsupervised learning, which are the potential dependent and independent variables, if it is a regression, classification, or clustering problem, and justify your answers. If it is not a statistical learning problem, provide reasons).

- **Network-level taxi demand prediction for the city of Munich**: this is a time series statistical learning problem that uses different data to predict taxi demand. This is a supervised learning model that can use diverse algorithms such as random forests and neural networks. It is a regression problem that uses as input variables (independent variables) areas/subdivisions of space, location of vehicle, weather conditions, population demand based on cell phone data, time of day, day of the week, traffic conditions, amongst others [1]. The output variables (dependent variable) is the taxi demand.

- **Identification of lane-changing maneuvers from trajectory data:** This could be a statistical learning problem that uses trajectory data as the independent variable to determine whether there is a lane-changing maneuver. This would be a supervised classification method. It is supervised because a ground truth is needed to train a model, and it is classification because the output indicates if there is a lane-changing maneuver or not.

- **Optimal taxi dispatching strategy based on passenger waiting time and operator cost:** This can be a statistical learning problem depending on how it's approached. If we think of it as a problem where we must optimize a certain function, it is a statistical learning problem. The inputs will be waiting time and operator costs and this is what we want to minimize. The output will be taxi dispatch interval (or strategy). This is an unsupervised learning problem.

- **Advance detection of e-scooter battery malfunction for preventive maintenance:** This is a statistical learning problem that uses data about the battery, that come from different sensors, as the independent variables to determine whether the battery is malfunctioning (dependent variable). This problem can be categorized as a supervised learning classification problem. It's supervised because a ground truth of when a battery malfunc-

[1] https://link.springer.com/article/10.1007/s13177-020-00248-9

tions would be used to train a model. It's a classification problem because the outcome is battery malfunctions/battery is well.

- **Map matching using probe vehicle data:** This is a statistical learning problem that maps recorded geographical data to a model of the real world. Map matching algorithms often use Hidden Markov Models which are a supervised machine learning models (we have a ground truth). The independent variables (input) are time stamps with location coordinates (probe vehicle data) and the dependent variable (outcome variable) is the location in the real world model. The time and location are associated to a vehicle.

- **Quantifying increase in traffic speed violations during COVID-19 pandemic:** This is not a statistical learning problem. Quantifying an increase in speed violations during the pandemic can be done by comparing violations pre- and post-pandemic. One way to do this would be to forecast pre-pandemic speed violations to the pandemic period and compare with real number of speed violations. Although forecasting can be considered a statistical learning problem quantifying the increase in itself is not a statistical learning problem.

- **Investigating crowding at public transport facilities during 9-Euro ticket in Germany:** This could be a statistical learning problem if crowding is monitored using computer vision. Deep learning models are used to track people and crowd density estimation. In this case, the input variable can be images or videos of a public transport facility. The output variable can be number of people or crowd density. These are supervised learning methods. If the outcome is number of people, the model uses object detection (which classifies objects) to detect people, after which we can have a count of the total number.

- **Identification of sentiment towards public transport using Twitter feeds**: Sentiment analysis (a field of NLP) is a statistical learning problem. It can be either supervised or unsupervised. It is a classification problem because text is labeled as positive, neutral, or negative (or a scale that reflects different levels of this basic one). For the supervised version, after building a training set any prediction method can be used to predict sentiment class. Unsupervised methods uses rules or a dictionary to quantify emotions. The output (dependent variable) is the sentiment, and the input (independent variables) is the pre-processed text, or variables generated from this text.

- **In-vehicle real-time parking spot recommendation system**: In vehicle real-time parking spot recommendation systems can be thought of as a two-process problem. The first is the detection of available spaces and the second is the recommendation of parking spaces. The first process can be performed, for example, using neural networks or Hidden Markov Models [2]. The second part, the recommendation, is a rule-based system that ultimately provides a suggestion to a user (that can be personalized). The input of this problem (independent variables) are variables such as: current location of user, destination, day of week, time of day, traffic conditions, weather conditions, known parking spaces. It can also use historical data. The output of the first process will be the probability of the parking space being available, which will be used for the final output, which is the parking spot recommendation. Recommendation systems can be supervised, in which case they are classification problems (you have feedback of whether a user liked the recommendation or not an use that as labels), or they can be unsupervised, in which case clustering is used [3].

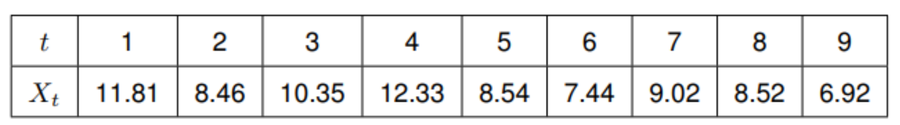
- **Identifying the travel mode based on joint probability of trip statistics (trip length, travel speed, etc.) from the floating car data (FCD):** This is a statistical learning problem that uses joint probability of trip statistics (as input/independent variable) to identify travel mode (output/dependent variable). This problem can be supervised or unsupervised, depending on whether there is a ground truth available. In the case of supervised learning this would be a classification problem and in the case of unsupervised learning this would be a clustering problem.

[2] https://www.researchgate.net/publication/295092249_Recognition_and_Recommendation_of_Parking_Places

[3] https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e

PROBLEM 2

1. Suppose we obtain a traffic occupancy time series of length nine from a loop detector as follows (in percentage).



t	1	2	3	4	5	6	7	8	9
X_t	11.81	8.46	10.35	12.33	8.54	7.44	9.02	8.52	6.92

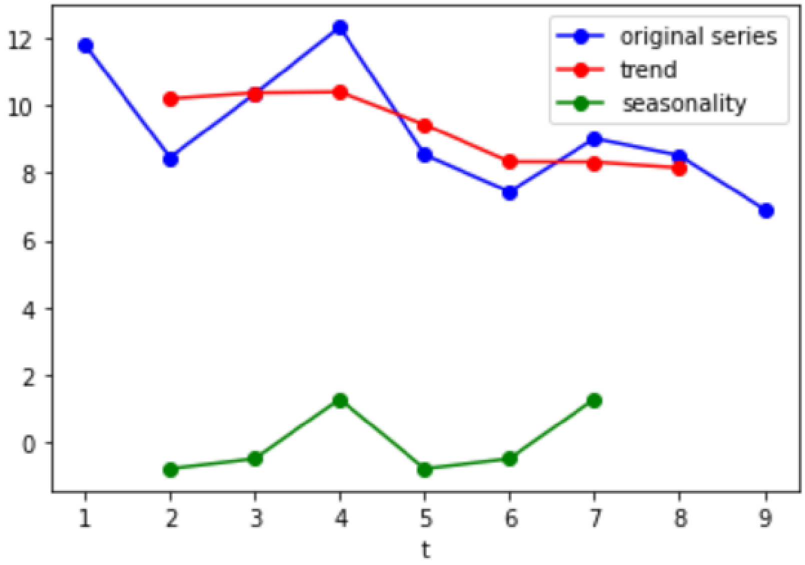
Suppose we know this is a periodic time series with periodicity d=3. Estimate the seasonal component from this time series and subtract the estimated seasonal component from the original series. You are required to do this without using the automatic decompose function in Python or R.

(a) Show your estimated seasonal component.

(b) Write down your time series after removing the seasonality.

a)

The seasonal component is estimated using local filtering. (All the calculations can be seen in the notebook.)



b)

Time series can be decomposed as:

$X_t = m_t + s_t + Y_t$

Where m_t is the trend component, s_t is the seasonal component, and Y_t is a random noise component.

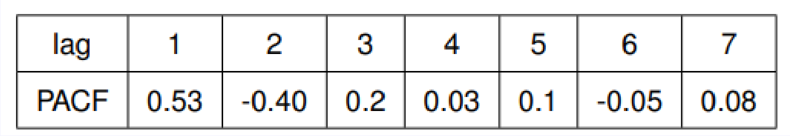
Therefore, if we remove the seasonal component we will have $X_t - s_t$

$X_t = [[11.81, 8.46, 10.35, 12.33, 8.54, 7.44, 9.02, 8.52, 6.92]$

$s_t = [1.279, -0.788, -0.491, 1.279, -0.788, -0.491, 1.279, -0.788, -0.491]$

$X_t - s_t = [10.531, 9.248, 10.841, 11.051, 9.328, 7.931, 7.741, 9.308, 7.411]$

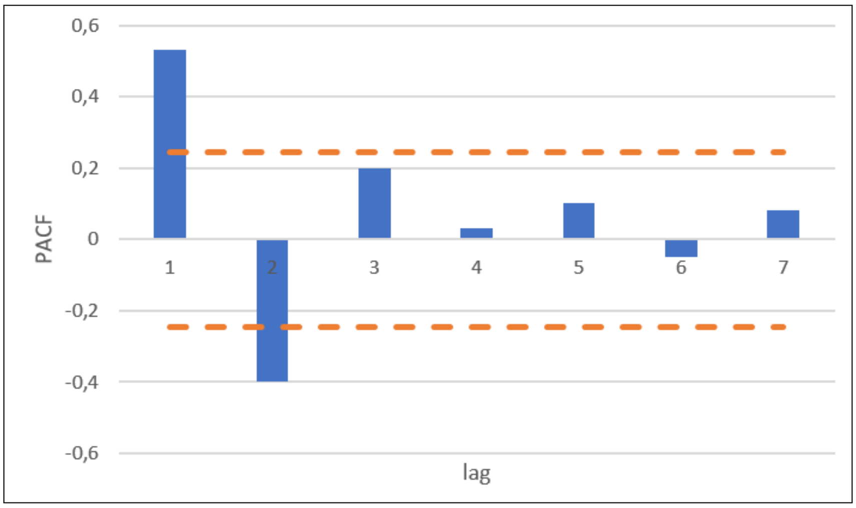
2. For a traffic time series of length 64, the sample partial autocorrelations are given as:



lag	1	2	3	4	5	6	7
PACF	0.53	-0.40	0.2	0.03	0.1	-0.05	0.08

Which models should we consider in this case, and why?

The following shows a plot of the partial autocorrelations at different lags. The sample partial autocorrelation function is defined as the correlation between Y_t and Y_{t-k} after removing the effect of the intervening variables Y_{t1}, Y_{t2}, $\ldots$, Y_{t-k+1}. The PACF detects the p length of AR(p) model.



The partial autocorrelation function for an AR(p) process cuts off after the lag exceeds the order p of the process. If the sample PACF falls between the plotted bounds $1.96/sqrt(n)$ (horizontal dashed lines) for lags $h > p$, then an AR(p) model is suggested. In this case $n = 64$

and therefore, the value for the bounds is 0.245.

In the plot, the lags cut-off at lag 2 (after lag 2 the values are no longer significant). Therefore the process would be AR(2). Depending on what the ACF plot looks like, if the autocorrelation function cuts off after the lag exceeds the order q of the process, then we would have a MA(q). In that case we would have an ARMA(2,0,q). Otherwise it would just be an AR(2).

3. An MA(1) process is given by the following equation:

$Y_t = 7\varepsilon_{t-1} + \varepsilon_t$

Write down the ACF for this process.

ACF(k) is a measure of the correlation between Y_t and Y_{t+k} for a given lag k.

$\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$ meaning that the ε_t are identically, independently distributed, each with a normal distribution having mean 0 and the same variance.

For an MA(1) process [4],

- Mean is $E(Y_t) = \mu$
- Variance is $Var(Y_t) = \sigma_\varepsilon^2(1+\theta_1^2)$

Autocorrelation function (ACF) is:

$$ACF(0) = 1$$
$$ACF(1) = \frac{\theta_1}{1+\theta_1^2} = \frac{7}{1+7^2} = \frac{7}{50} = 0.14$$
$$ACF(k) = 0 \text{ for } k \geq 2$$

4. For clustering using a mixture of two Gaussians, you are given four data points in 1-D.

x=[1,2,20,40]

The result of the E-step is the following matrix:

$$\begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \\ 0 & 1 \\ 0.7 & 0.3 \end{bmatrix}$$

Determine the mixing weights and means after M-step.

[4] https://online.stat.psu.edu/stat510/lesson/2/2.1

The result of the E-step are the responsibilities $\gamma(Z_{nk})$. Where n refers to a data point and k to a component. In this case there are 4 data points and 2 components. Therefore, the result of the E-step is:

$$\begin{bmatrix} \gamma(Z_{11}) & \gamma(Z_{12}) \\ \gamma(Z_{21}) & \gamma(Z_{22}) \\ \gamma(Z_{31}) & \gamma(Z_{32}) \\ \gamma(Z_{41}) & \gamma(Z_{42}) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \\ 0 & 1 \\ 0.7 & 0.3 \end{bmatrix}$$

For the M-step we re-estimate the parameters using the responsibilities from the E-step.

Means $\longrightarrow$ $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk}) X_n$
Mixing Coefficients $\longrightarrow$ $\pi_k^{new} = \frac{N_k}{N}$
where $N_k = \sum_{n=1}^{N} \gamma(Z_{nk})$

$N_1 = \sum_{n=1}^{4} \gamma(Z_{n1}) = 0.5 + 1 + 0 + 0.7 = 2.2$
$N_2 = \sum_{n=1}^{4} \gamma(Z_{n2}) = 0.5 + 0 + 1 + 0.3 = 1.8$

The new mixing weights are:
$\pi_1 = \frac{N_1}{N} = \frac{2.2}{4} = 0.55$
$\pi_2 = \frac{N_2}{N} = \frac{1.8}{4} = 0.45$

The new means are:
$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{4} \gamma(Z_{n1}) X_n = \frac{1}{2.2} * (0.5 * 1 + 1 * 2 + 0 * 20 + 0.7 * 40) = 13.86$
$\mu_2 = \frac{1}{N_2} \sum_{n=1}^{4} \gamma(Z_{n2}) X_n = \frac{1}{1.8} * (0.5 * 1 + 0 * 2 + 1 * 20 + 0.3 * 40) = 18.06$

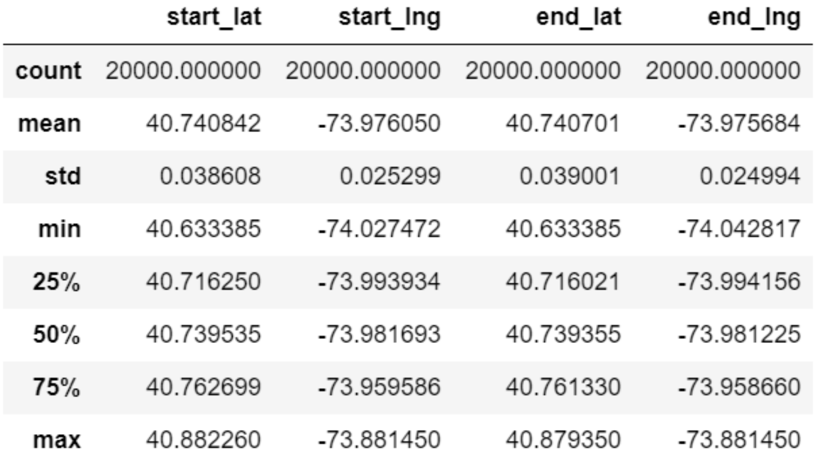
PROBLEM 3

Download the bike share.csv data 1 from Moodle. Data consists of start location, end location, start time, and end time of the trips, among other attributes. The service operator wants to identify origin-destination pairs with most trips for focused resource allocation. You are required to analyze the data and use a clustering method to divide the trips into sets of similar trips: trips with “similar” start AND end points. i.e., for the trips in the same cluster, the maximum distance between the start locations should be less than 0.05 deg, AND the maximum distance between the end locations should also be less than 0.05 deg.

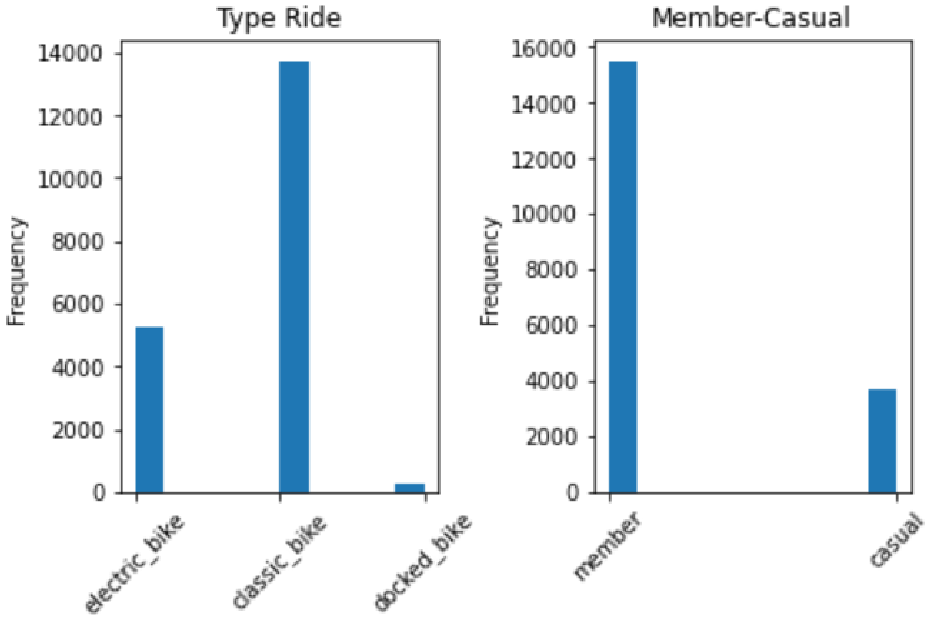
All the calculations are in the attached notebook.

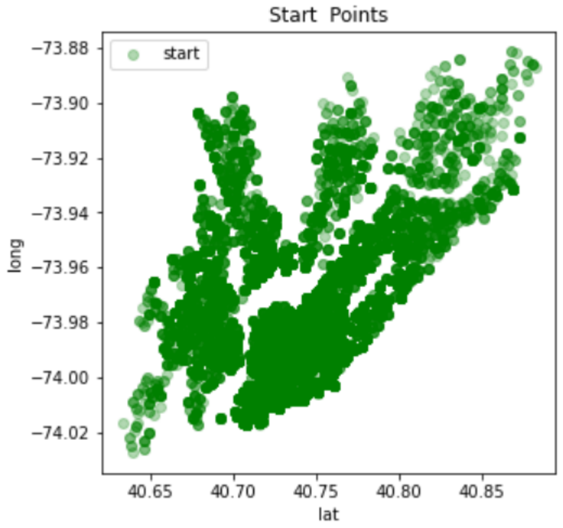
1) Visualize the data and describe the summary statistics.

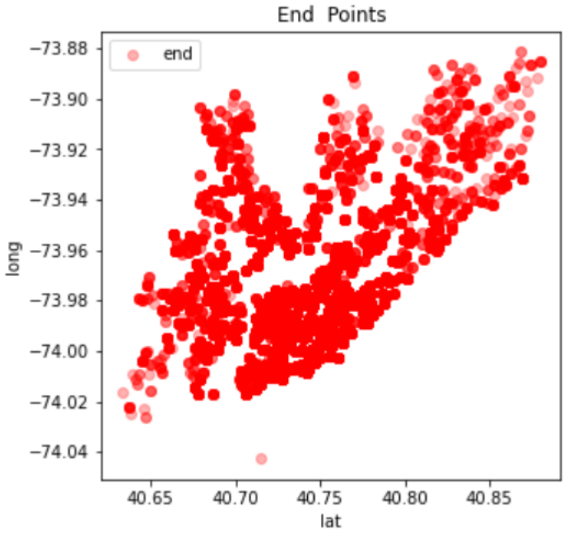
First I read the data and verify that there are no missing values. Next I obtain the summary statistics and plots for the categorical data, as well as the start and end points for the trips.



	start_lat	start_lng	end_lat	end_lng
count	20000.000000	20000.000000	20000.000000	20000.000000
mean	40.740842	-73.976050	40.740701	-73.975684
std	0.038608	0.025299	0.039001	0.024994
min	40.633385	-74.027472	40.633385	-74.042817
25%	40.716250	-73.993934	40.716021	-73.994156
50%	40.739535	-73.981693	40.739355	-73.981225
75%	40.762699	-73.959586	40.761330	-73.958660
max	40.882260	-73.881450	40.879350	-73.881450

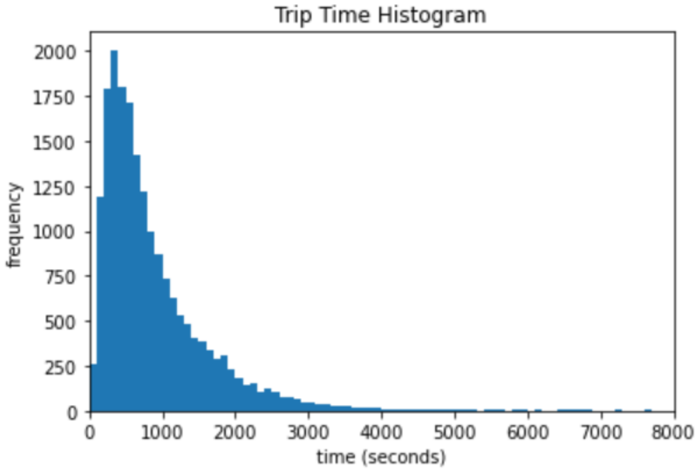


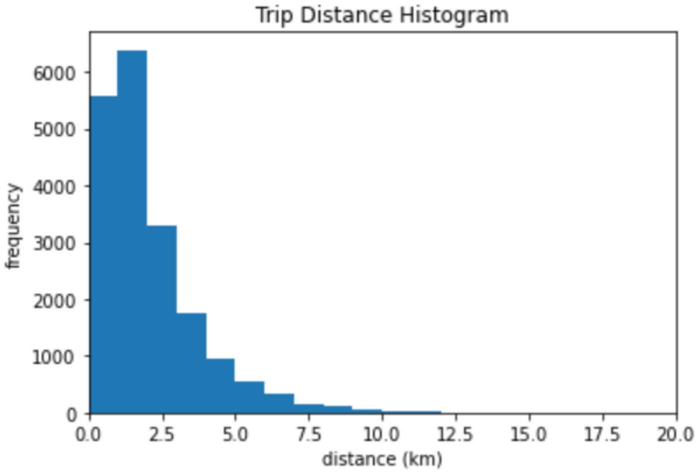


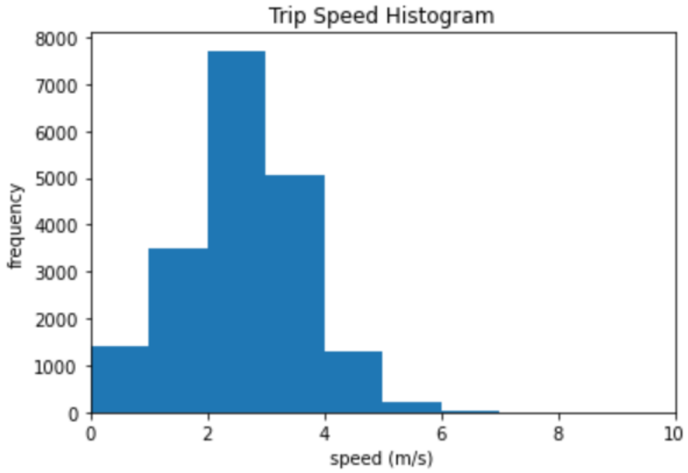


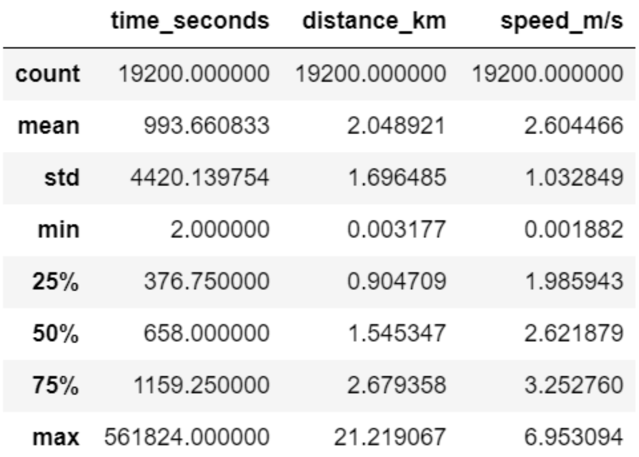
Additionally, I filter out the trips that have the same start and end position because I consider these trips as wrongly reported, so i eliminate them from the data set. Also, I remove an outlier point that for the end points is under -74.04 longitude.

2) Compute trip-time, trip distance, and trip-speed statistics from the given data. Plot their distributions. Discuss these quantities with respect to the other features in the data









	time_seconds	distance_km	speed_m/s
count	19200.000000	19200.000000	19200.000000
mean	993.660833	2.048921	2.604466
std	4420.139754	1.696485	1.032849
min	2.000000	0.003177	0.001882
25%	376.750000	0.904709	1.985943
50%	658.000000	1.545347	2.621879
75%	1159.250000	2.679358	3.252760
max	561824.000000	21.219067	6.953094

The distributions and statistics show that the mean speed is 2.6 m/s. This is odd, and a point to consider given that the average bike speed is 7 m/s. Although, this may be due to the fact that the trips are in a city with stoplights. Next, I will remove outliers.

3) Step-wise describes your approach to identify similar trips based on the criteria mentioned above and which clustering algorithm is suitable here.

In order to cluster the trips and fulfill the proposed criteria, I will use the agglomerative hierarchical clustering with complete linkage. I use this algorithm because it allows me to specify the maximum distance between points within a cluster. However, because of the way the distance is calculated, this can't be done in 4-d. This means I will have to perform separate clustering for the start locations and end locations.

1. The first step is to cluster the start locations and end locations separately. Using agglomerative hierarchical clustering we can specify the distance threshold and not specify the number of clusters, which means the algorithm will find this on its own. For this I use a pre-computed distance matrix calculated using the harvesine formula.

2. Once I obtain the clusters for start and end, I group the trips by start cluster label and end cluster label, and count the number of trips each set has.

3. Once I have the list of trips per set of clusters I choose the set of labels that have the highest amount of trips, being careful not to have any label repeated in the groups for start and end clusters, and assign them new labels that will be the definitive clusters. This means many trips will be left out of the data.

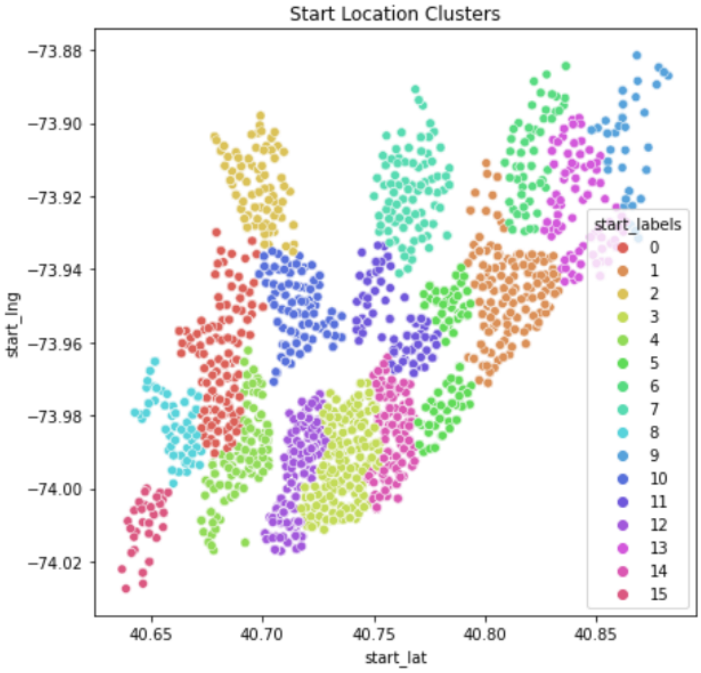
4. To increase the amount of data used in the clustering, I repeat steps 1-3 with the trips that haven't been assigned a cluster.

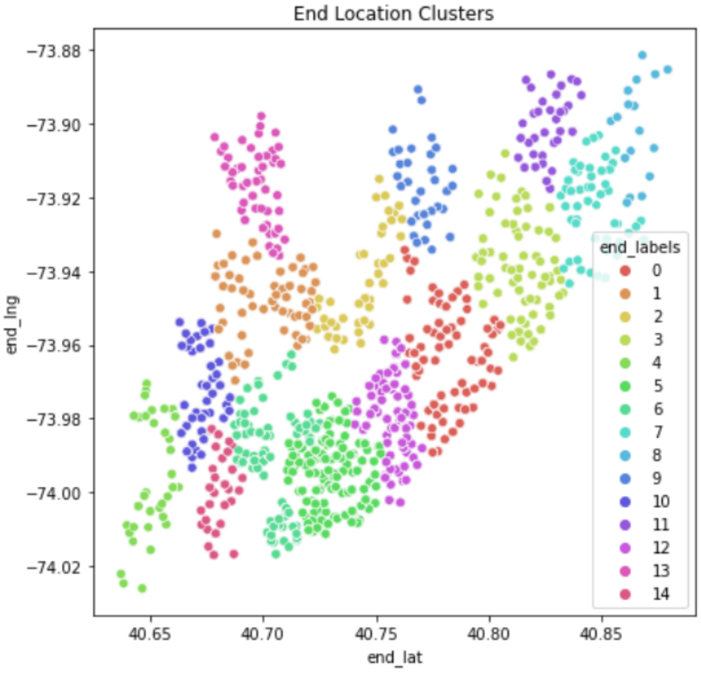
5. With the resulting clusters I check that the distance requirements are fulfilled for the start and

end locations.

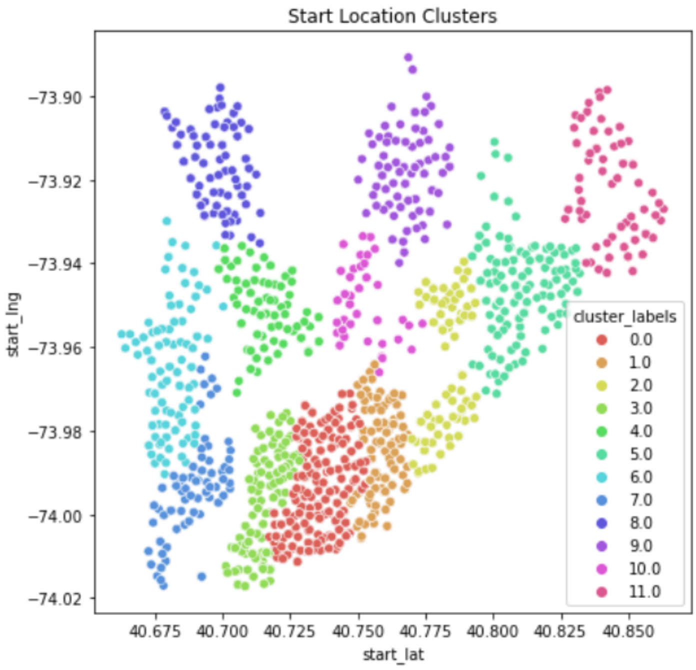
4) Apply your approach, group the trips, and visualize the clusters using different plots.

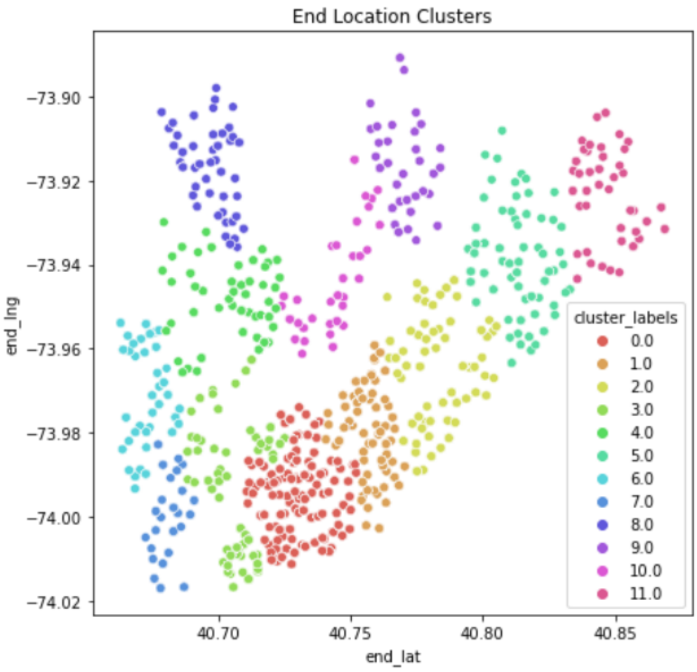
For the first step I obtain the following clustering:



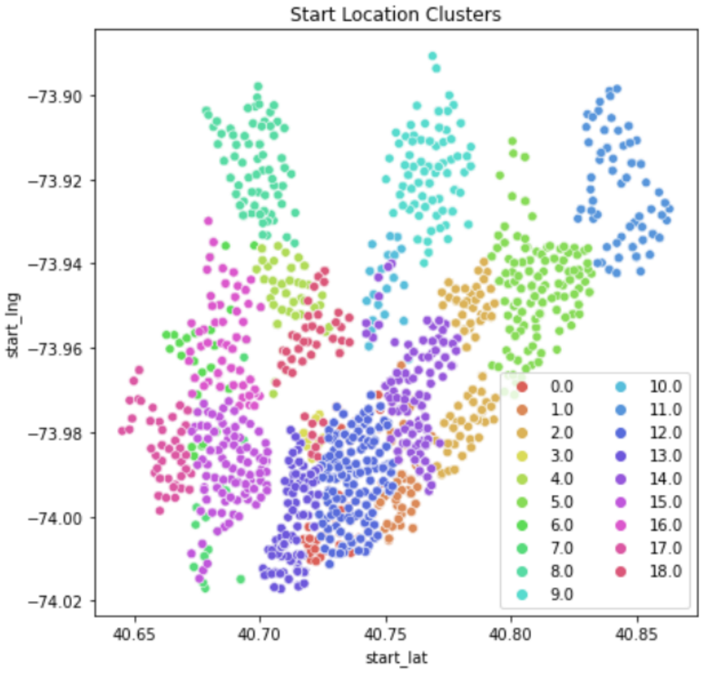


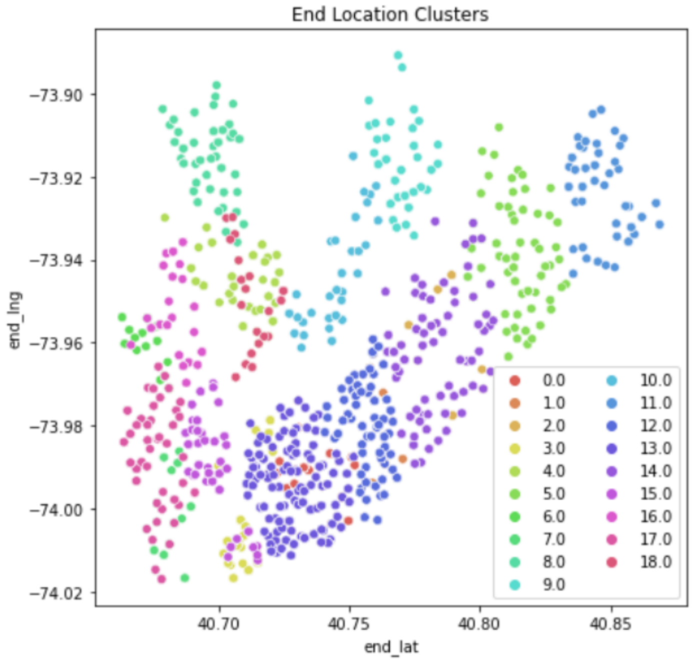
After step 3 the following clusters are obtained:





However, this only constitutes 45% of the data, which is why I repeat steps 1-3 with the trips that do not have a cluster. This leads to the following final clustering configuration that uses 66% of the original data. All clusters have more than 100 trips.





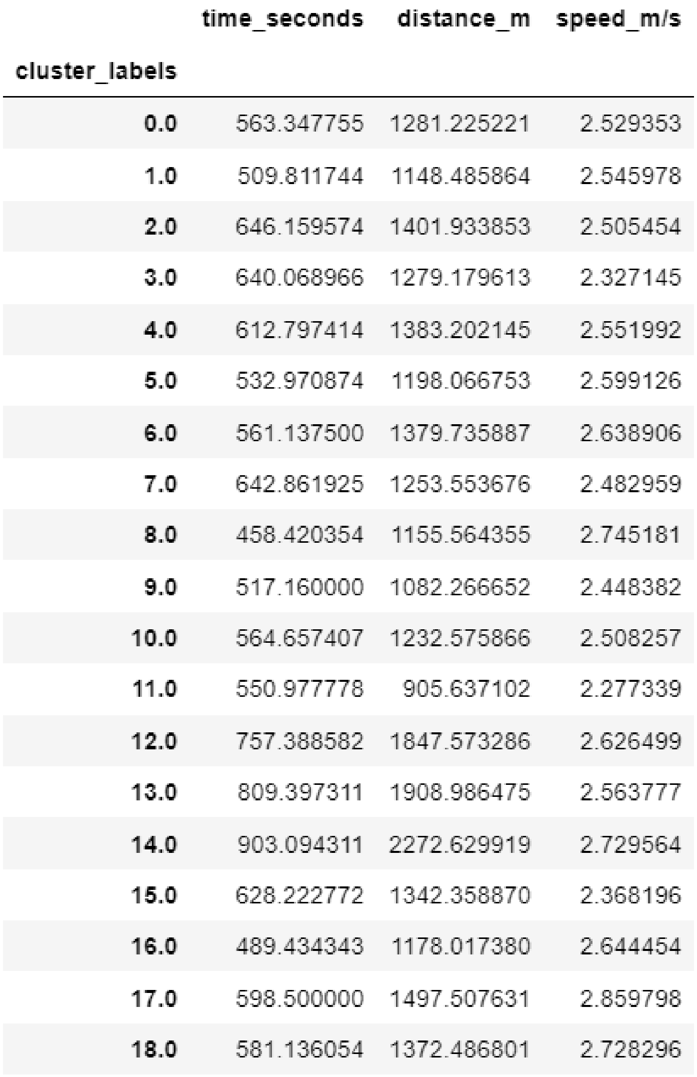
The distance check shows that for all start and end locations the clusters comply with the criteria (see notebook).

5) Evaluate your clustering with the help of clustering performance measures.

The clustering is evaluated using the silhoutte coefficient which combines ideas of both cohesion and separation. It varies between -1 and 1. Closer to 1 is better, and values near 0 indicate overlapping clusters. Negative values indicate that points are assigned to the wrong cluster. This score for all the clusters is 0.186. This indicates that there are some overlapping clusters.

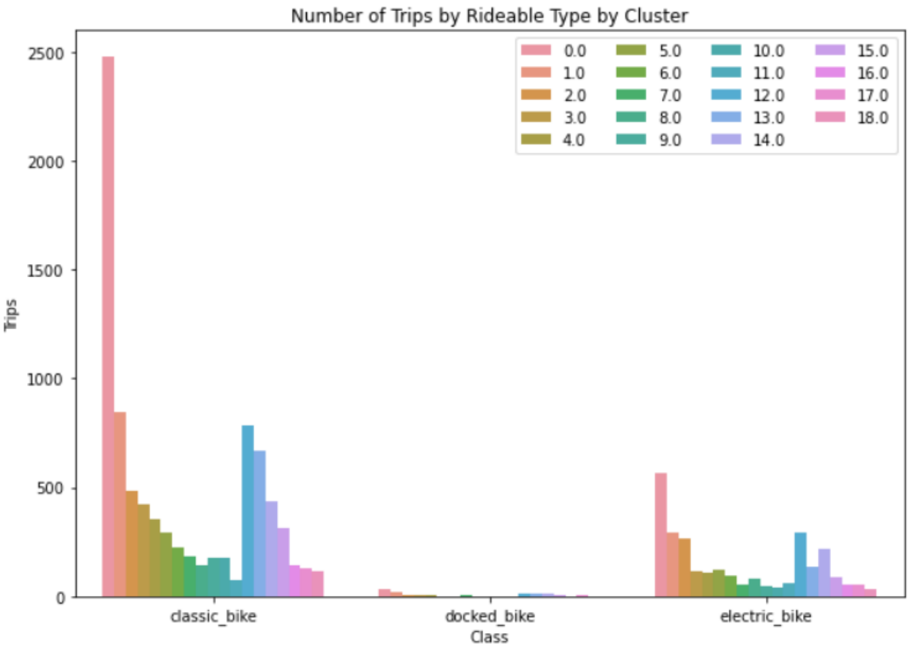
Repeating steps 1-3 two times results in some overlapping clusters. However, these clusters can overlap for start locations and not so much for the end locations or vice versa. In some cases, further analysis could be done to determine if its worth keeping so many clusters. This is also a matter of trade-off between score and the percentage of data we want to have included in the clusters.

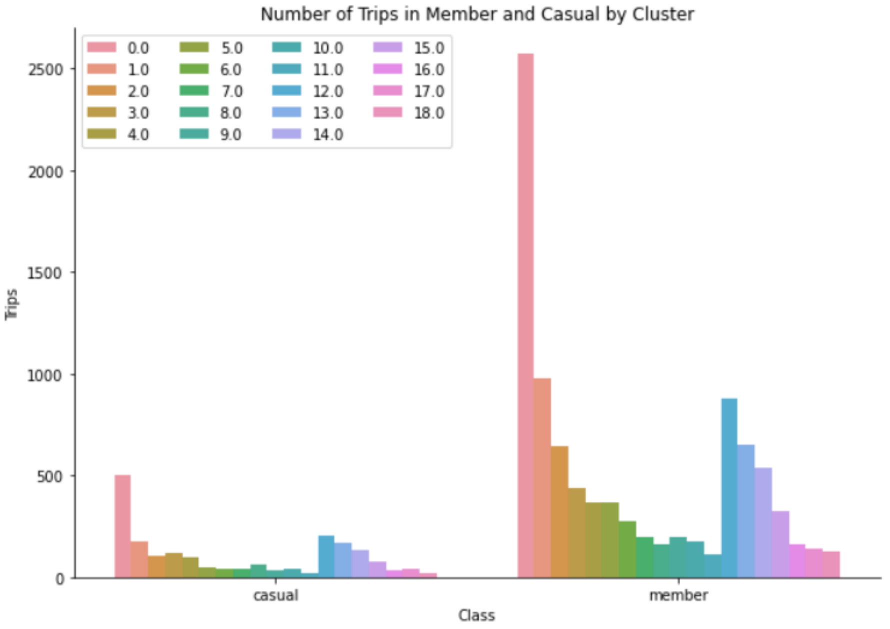
6) Briefly discuss the trip statistics for major trip groups/ clusters.



	time_seconds	distance_m	speed_m/s
cluster_labels			
0.0	563.347755	1281.225221	2.529353
1.0	509.811744	1148.485864	2.545978
2.0	646.159574	1401.933853	2.505454
3.0	640.068966	1279.179613	2.327145
4.0	612.797414	1383.202145	2.551992
5.0	532.970874	1198.066753	2.599126
6.0	561.137500	1379.735887	2.638906
7.0	642.861925	1253.553676	2.482959
8.0	458.420354	1155.564355	2.745181
9.0	517.160000	1082.266652	2.448382
10.0	564.657407	1232.575866	2.508257
11.0	550.977778	905.637102	2.277339
12.0	757.388582	1847.573286	2.626499
13.0	809.397311	1908.986475	2.563777
14.0	903.094311	2272.629919	2.729564
15.0	628.222772	1342.358870	2.368196
16.0	489.434343	1178.017380	2.644454
17.0	598.500000	1497.507631	2.859798
18.0	581.136054	1372.486801	2.728296

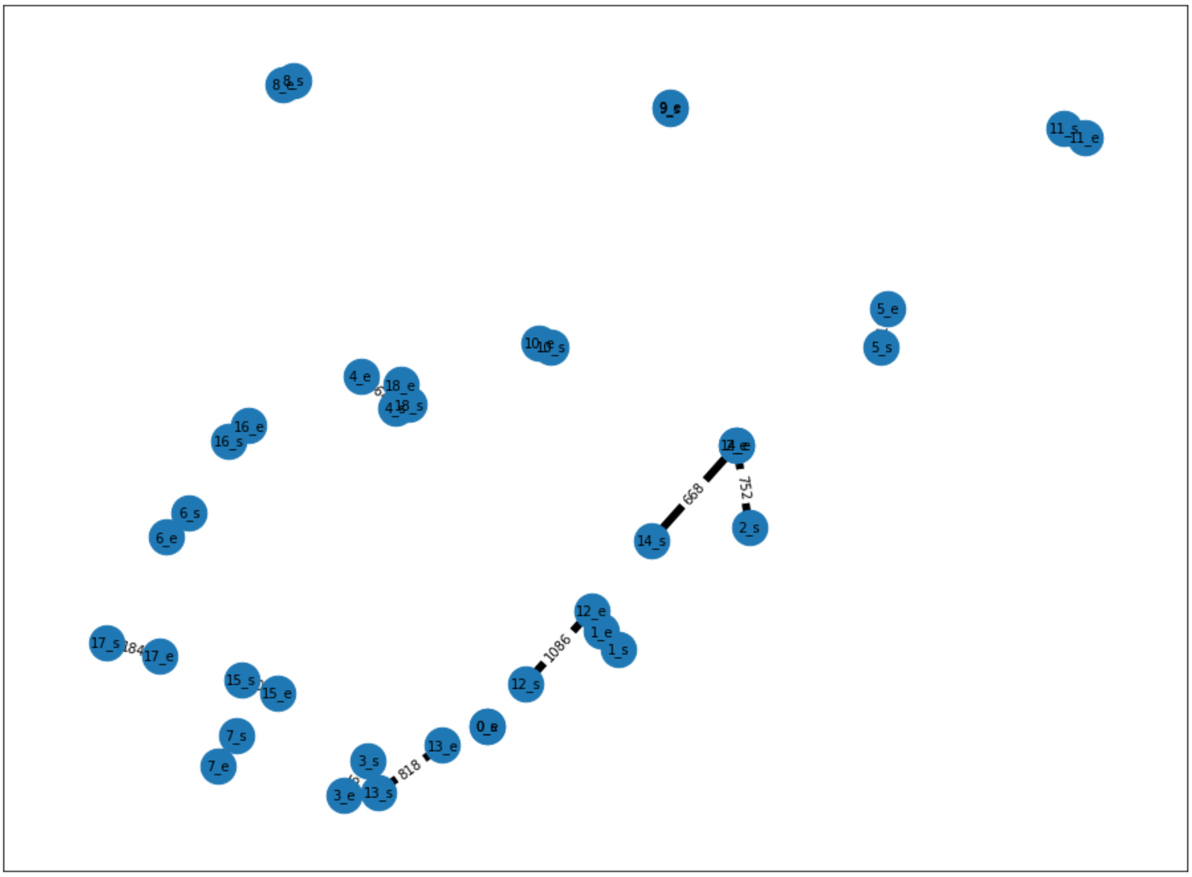
Regarding time, clusters 8 and 16 have the shortest trip times, while 12,13, and 14 have much longer times. For the distance metric, cluster 11 has the shortest distances and cluster 14 has the longest distances. In regards to speed, cluster 11, 3, and 15 have the lowers speeds, while clusters 8,14,17, and 18 have the highest speeds.





For the categorical variables type of ride and member/casual the trend of proportion seems to be similar.

7) Plot a network or graph visualization where nodes represent the origin and destination centers and the thickness of edges between nodes represent the number of trips assigned to that trip group.

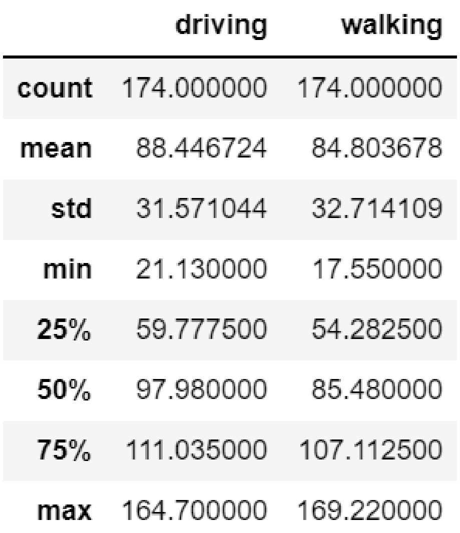

8_e
8_s
9_e
11_s
11_e
5_e
5_s
10_e
10_s
4_e
18_e
4_s
18_s
16_e
16_s
6_s
6_e
14_e
668
752
14_s
2_s
12_e
1_e
1_s
1086
12_s
17_s
184
17_e
15_s
15_e
7_s
7_e
3_s
13_e
0_e
818
3_e
13_s

PROBLEM 4

Download the apple mobility trends.csv from Moodle. This data shows walking and driving trends 2 for the city of Munich. The data were collected during the COVID-19 pandemic and are quantified relative to the base value (=100) on 13.01.2020. You are tasked to analyze this time-series data and develop forecasting models to predict the driving trends. Use the model you estimated to perform forecasts for the driving trends for the last 14 days in the data, i,e, use data for the last 14 days as test data. Show details of your modeling and forecasting process step by step with reasoning/discussion. Organize your response into your approach and the obtained results and must include the following components:

All the calculations are in the attached notebook.

1) Statistical summary of the original data.



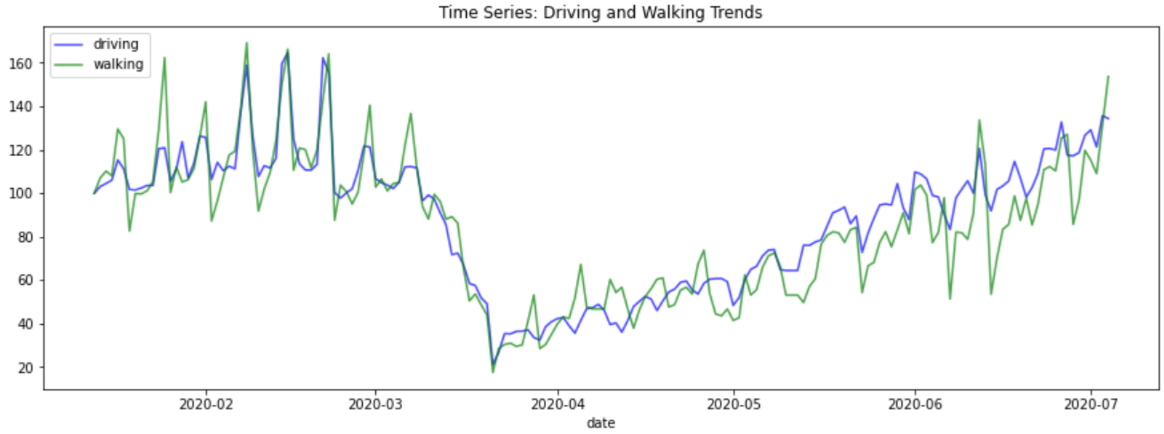
	driving	walking
count	174.000000	174.000000
mean	88.446724	84.803678
std	31.571044	32.714109
min	21.130000	17.550000
25%	59.777500	54.282500
50%	97.980000	85.480000
75%	111.035000	107.112500
max	164.700000	169.220000

2) Data visualization and discussing the patterns.

Trends: at the beginning the trend seems to remain constant. However, from approximately March 2020 until the end of the month, there seems to be a decrease in the measurements for walking and driving. The trend appears to be reversed towards the end of March and the measurements for both driving and walking increase until July 2020.

Seasonality: there seems to be some seasonality at certain points for the time-series. For example, between February and March there appear to be seasonality components in the time series for both walking and driving. It seems logical to assume a seasonality period of a week (7 days) because driving and walking trends would change, for example, whether it is a weekend or a weekday.

Variance: there are certain periods where the variance appears to be greater than in other. Therefore, the variance appears to be time-dependent.

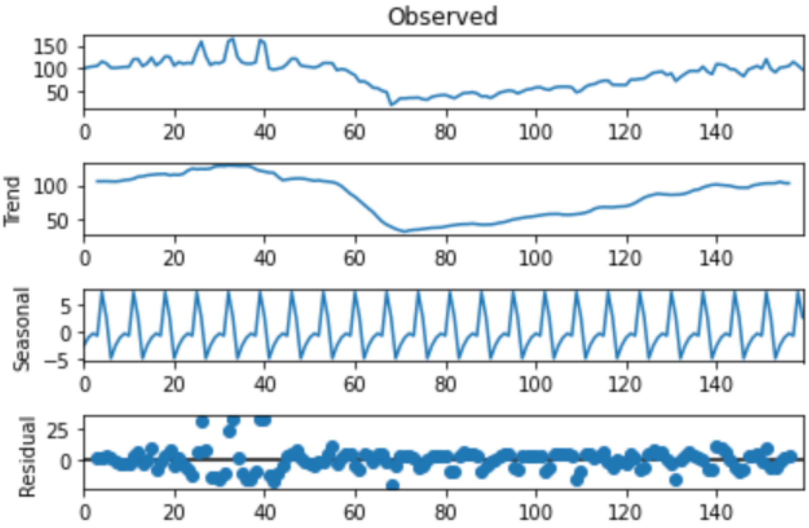


3) Data split into training and test data.

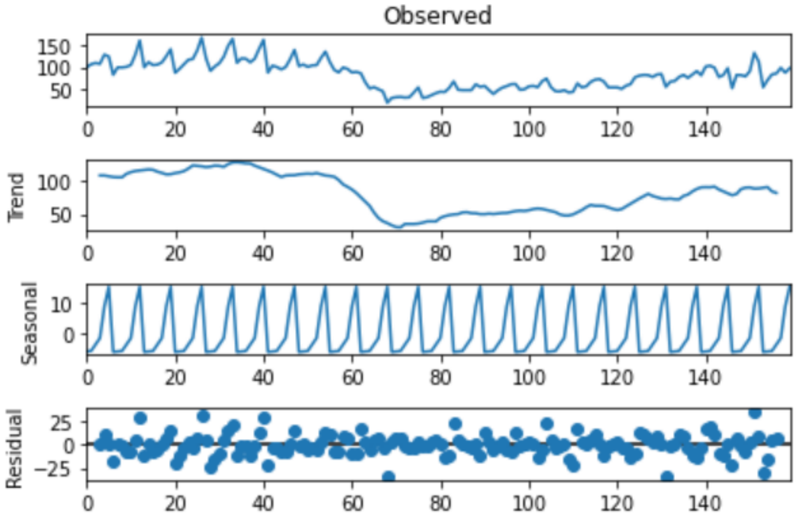
The last 14 days of the time-series are left out as the test set.

4) Decomposition of both the time series.

For driving trends:



For walking trends:



5) Testing the time-series for stationarity using statistical tools.

Augmented Dickey Fuller (ADF) test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test are both tests to check for stationarity.

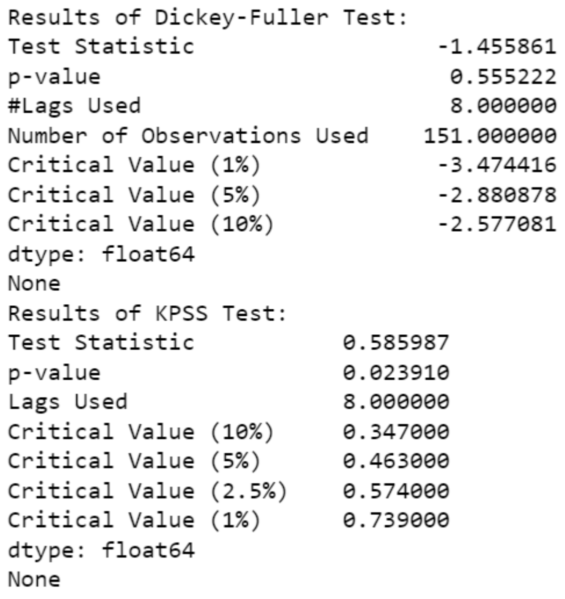
ADF: To determine if the series is stationary or not. Hypothesis of this test are:

- Null Hypothesis: The series has a unit root (is not stationary).
- Alternate Hypothesis: The series has no unit root.

If the null hypothesis in failed to be rejected, this test may provide evidence that the series is non-stationary.

KPSS :KPSS is another test for checking the stationarity of a time series. Null and alternate hypothesis for the KPSS test are opposite that of the ADF test.

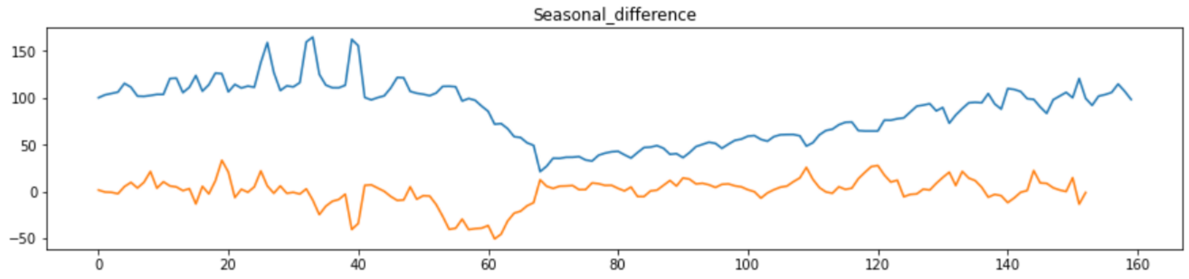
- Null Hypothesis: The process is trend stationary.
- Alternate Hypothesis: The series has a unit root (series is not stationary).



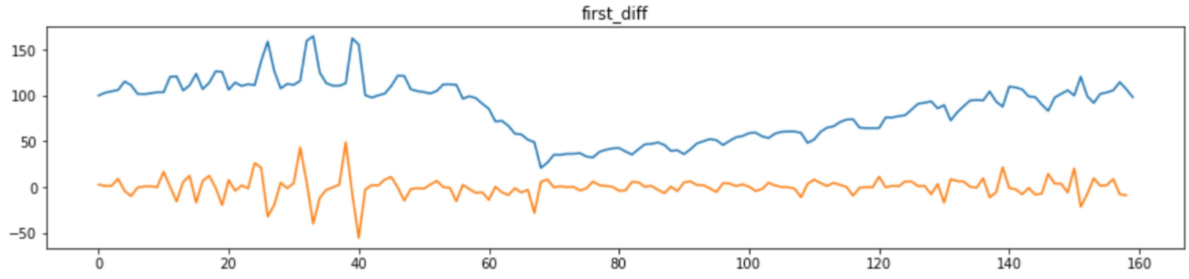
```
Results of Dickey-Fuller Test:
Test Statistic                  -1.455861
p-value                          0.555222
#Lags Used                       8.000000
Number of Observations Used    151.000000
Critical Value (1%)             -3.474416
Critical Value (5%)             -2.880878
Critical Value (10%)            -2.577081
dtype: float64
None
Results of KPSS Test:
Test Statistic           0.585987
p-value                  0.023910
Lags Used                8.000000
Critical Value (10%)     0.347000
Critical Value (5%)      0.463000
Critical Value (2.5%)    0.574000
Critical Value (1%)      0.739000
dtype: float64
None
```

At a significance level of 5%, we can't reject the null hypothesis for the Dickey-Fuller test, meaning the series would not be stationary. For the KPSS test, the null hypothesis is rejected at the 5% significance level, which indicates that the series is non-stationary. Therefore, because both tests conclude that the series is not stationary, the series is not stationary. Because the series is not stationary, we need to make the data stationary by applying differences.

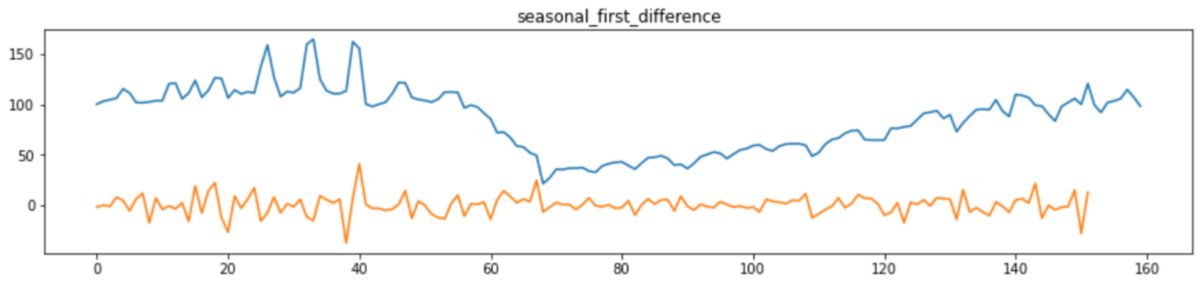
Seasonal differencing using 7 days as the period, results in:



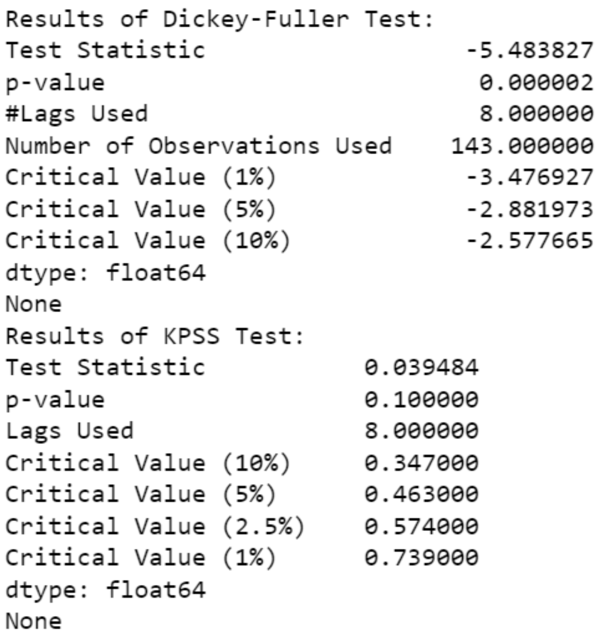
First differencing to eliminate trend:



First differencing and seasonal differencing:



I use the first difference with seasonal differencing because if I just use seasonal differencing there is still a hint of trend, while if I just use first differencing there remains a hint of seasonality. I perform the ADF and KPSS test on the differenced series:

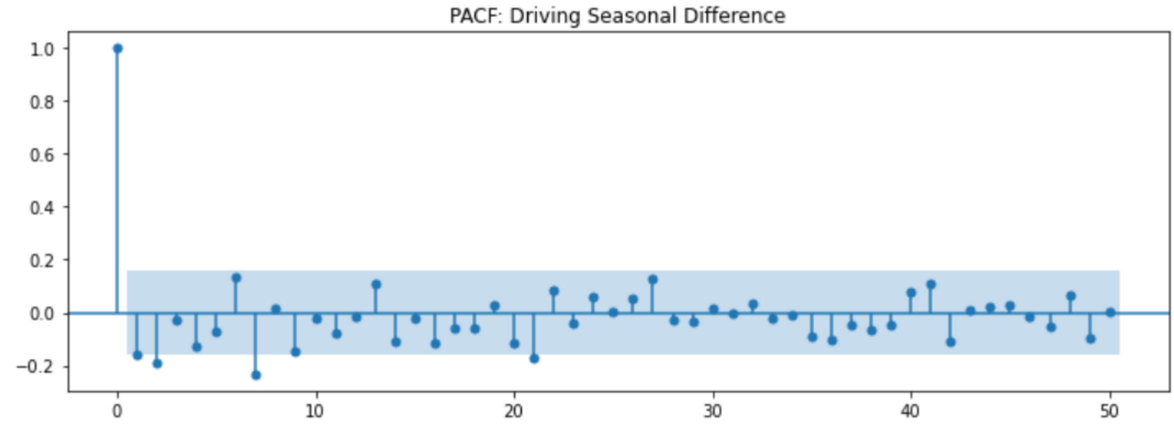


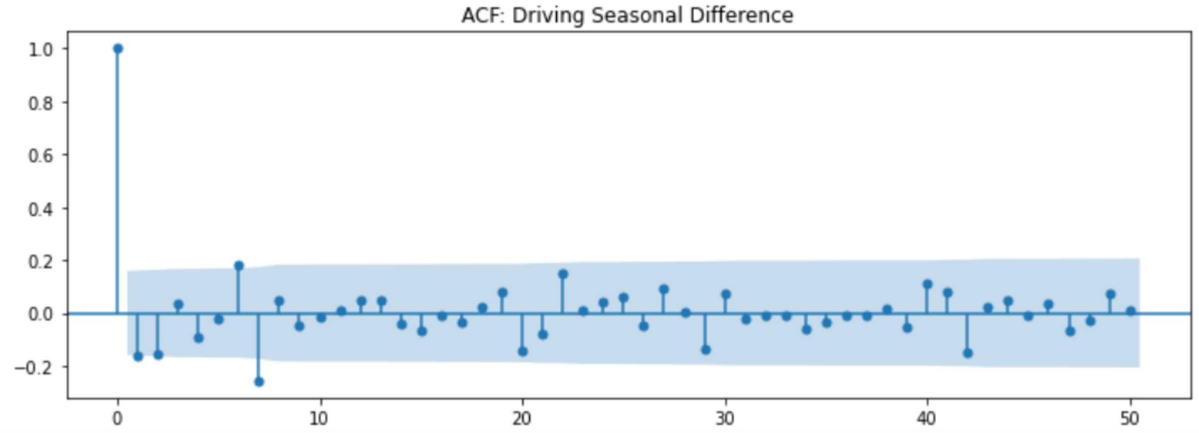
```
Results of Dickey-Fuller Test:
Test Statistic                  -5.483827
p-value                          0.000002
#Lags Used                       8.000000
Number of Observations Used    143.000000
Critical Value (1%)             -3.476927
Critical Value (5%)             -2.881973
Critical Value (10%)            -2.577665
dtype: float64
None
Results of KPSS Test:
Test Statistic           0.039484
p-value                  0.100000
Lags Used                8.000000
Critical Value (10%)     0.347000
Critical Value (5%)      0.463000
Critical Value (2.5%)    0.574000
Critical Value (1%)      0.739000
dtype: float64
None
```

At a 5% significance level, we can reject the ADF null hypotheses (time series not stationary), so we can say the time series is stationary. For the KPSS test, we can't reject the null hypotheses (time series stationary), therefore we can say the time series is stationary.

6) ACF and PACF analysis.

For the stationary time series, the ACF and PACF plots are:





The model will be of time SARIMA because there is a seasonal component. Therefore the model has the components ARIMA$(p,d,q)(P,D,Q)_s$. We know d and D will be 1 because the series is a first difference. Additionally, $s = 7$ because we applied seasonal differencing for a period of 7 days. To determine the seasonal component:

- For the ACF plot, we see a spike at lag 7 but no other significant spikes.
- For the PACF plot, there is an exponential decay in the seasonal (i.e., at lags 7, 14, 21)

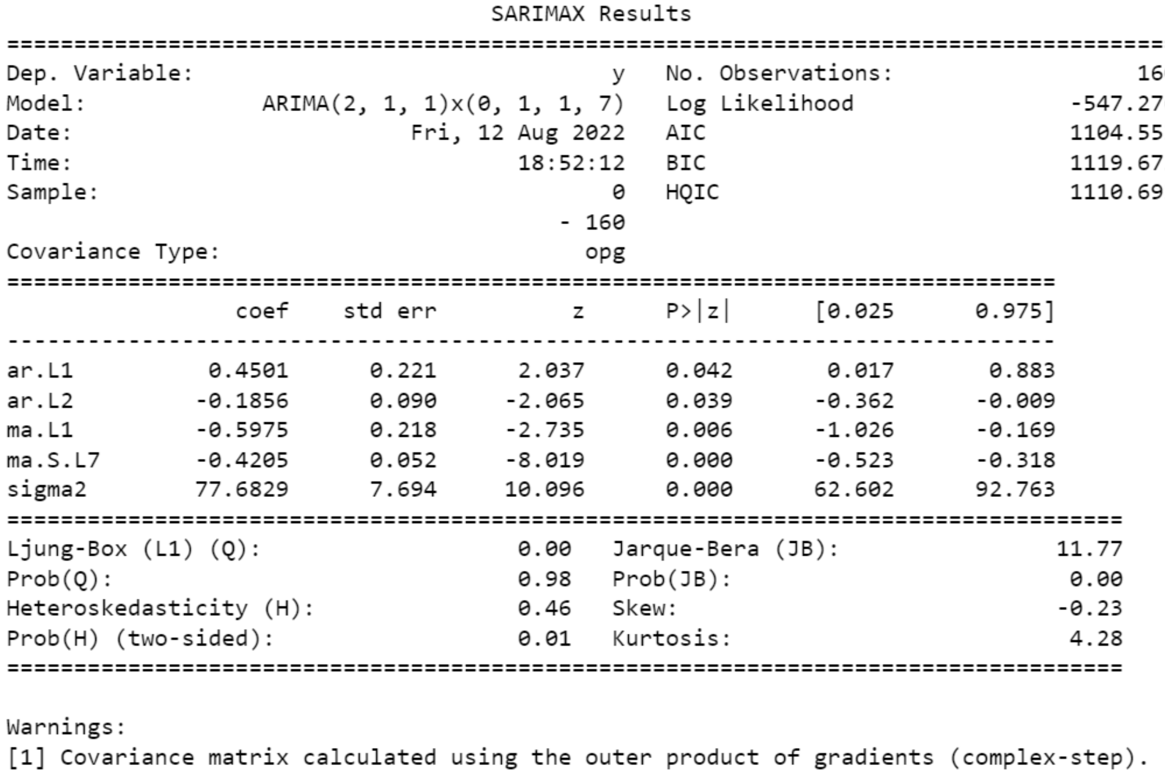
Therefore we would have an ARIMA$(0,0,0)(0,0,1)_7$. To determine the non-seasonal component we look at the first lags:

- For the ACF plot, only the first lag appears to be significant. Therefore, $q = 1$.
- For the PACF plot, the lags are insignificant (cut-off) after lag 2. Therefore, $p = 2$.

The complete model I use as a starting point is: ARIMA$(2,1,1)(0,1,1)_7$.

7) Model specification, parameter estimation, and significance test.

The result of the model fit shows the parameter estimation, and that at a 5% significance level the coefficients are significant.



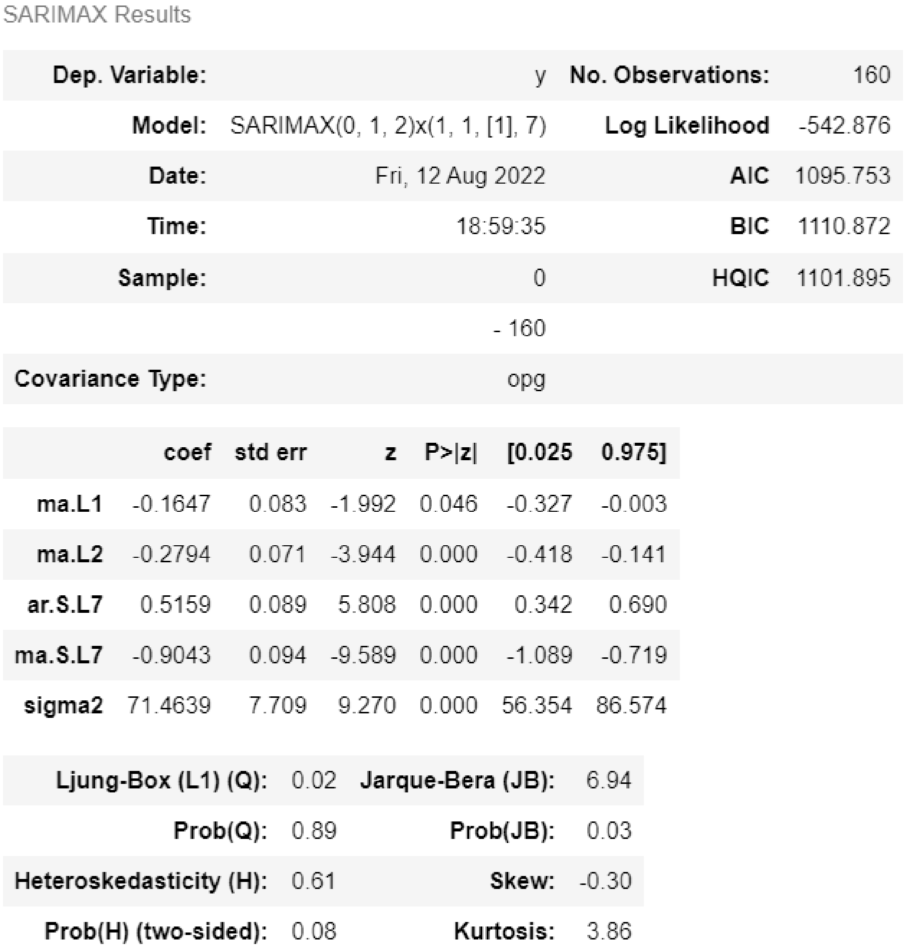
```
                                     SARIMAX Results
=========================================================================================
Dep. Variable:                                  y   No. Observations:                  160
Model:             ARIMA(2, 1, 1)x(0, 1, 1, 7)   Log Likelihood                -547.276
Date:                            Fri, 12 Aug 2022   AIC                           1104.553
Time:                                    18:52:12   BIC                           1119.672
Sample:                                         0   HQIC                          1110.695
                                            - 160
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4501      0.221      2.037      0.042       0.017       0.883
ar.L2         -0.1856      0.090     -2.065      0.039      -0.362      -0.009
ma.L1         -0.5975      0.218     -2.735      0.006      -1.026      -0.169
ma.S.L7       -0.4205      0.052     -8.019      0.000      -0.523      -0.318
sigma2        77.6829      7.694     10.096      0.000      62.602      92.763
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                11.77
Prob(Q):                              0.98   Prob(JB):                         0.00
Heteroskedasticity (H):               0.46   Skew:                            -0.23
Prob(H) (two-sided):                  0.01   Kurtosis:                         4.28
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

From the statistical summary and the visualization (points 1) and 2) we can see that the driving and walking trends are similar, and there doesn't seem to be a dynamic where when the driving trends go down the walking trends go up or vice-versa. Therefore, I don't think that including walking trends as a predictor will help improve the model, which is why I won't include it as an exogenous variable.

8) Criteria for model selection.

For model selection, I use the auto-arima function. The auto-ARIMA process seeks to identify the most optimal parameters for an ARIMA model, based on the AIC. The function yields as the best model ARIMA$(0,1,2)(1,1,1)_7$.



SARIMAX Results

Dep. Variable:	y	**No. Observations:**	160
Model:	SARIMAX(0, 1, 2)x(1, 1, [1], 7)	**Log Likelihood**	-542.876
Date:	Fri, 12 Aug 2022	**AIC**	1095.753
Time:	18:59:35	**BIC**	1110.872
Sample:	0	**HQIC**	1101.895
	- 160		
Covariance Type:	opg		

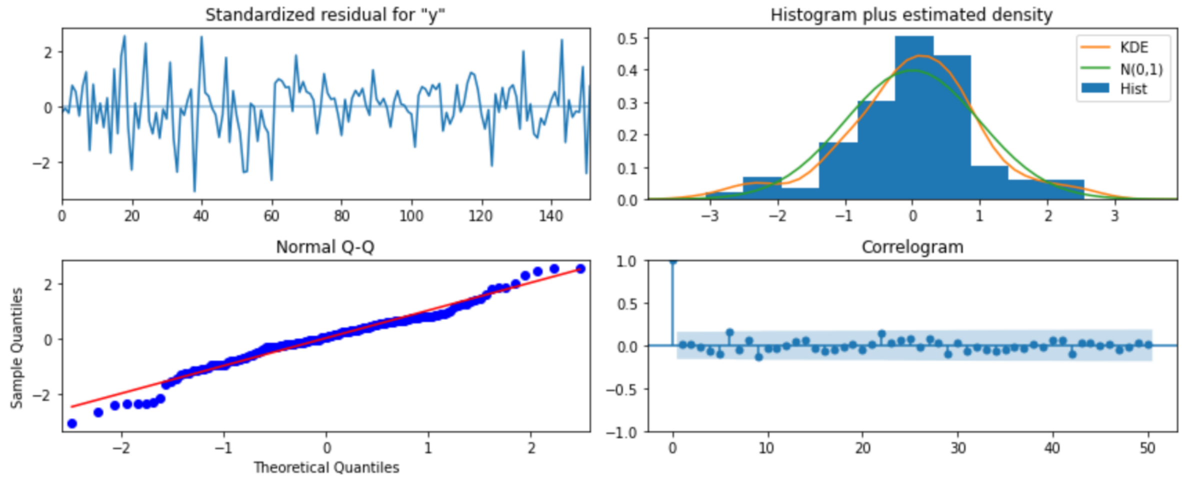
	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ma.L1	-0.1647	0.083	-1.992	0.046	-0.327	-0.003
ma.L2	-0.2794	0.071	-3.944	0.000	-0.418	-0.141
ar.S.L7	0.5159	0.089	5.808	0.000	0.342	0.690
ma.S.L7	-0.9043	0.094	-9.589	0.000	-1.089	-0.719
sigma2	71.4639	7.709	9.270	0.000	56.354	86.574

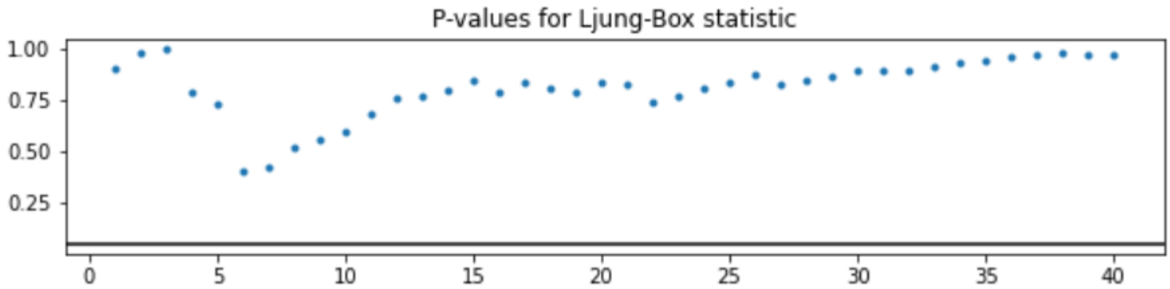
Ljung-Box (L1) (Q):	0.02	**Jarque-Bera (JB):**	6.94
Prob(Q):	0.89	**Prob(JB):**	0.03
Heteroskedasticity (H):	0.61	**Skew:**	-0.30
Prob(H) (two-sided):	0.08	**Kurtosis:**	3.86

9) Model diagnosis.

If the model is correctly specified and the parameter estimates are reasonably close to the true values, then the residuals should have nearly the properties of white noise. They should behave roughly like independent, identically distributed normal variables with zero means and common standard deviations.

- The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
- The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.
- The residuals have constant variance.
- The residuals are normally distributed.





The results seem to show that the stated conditions are met. Although, at the extremes the Normal Q-Q plot shows some deviations. The Ljung Box statistic uses the following hypothesis:

•H0: The residuals are independently distributed.

•HA: The residuals are not independently distributed; they exhibit serial correlation.

The p-values obtained from the Ljung-Box statistic are all larger than 0.05, which indicates that at a 5% level of significance we fail to reject the null hypothesis, which would indicate that the residuals are independently distributed.

10) Error metrics and forecasting performance on test data.

Error Metrics: I use one scale-dependent, one percentage-error metric, and one scale-free metric.

Scale-dependent:

- Mean Absolute Error (MAE)

Percentage-error metrics:

- Mean Absolute Percentage Error (MAPE)

Scale-free:

- R-Squared

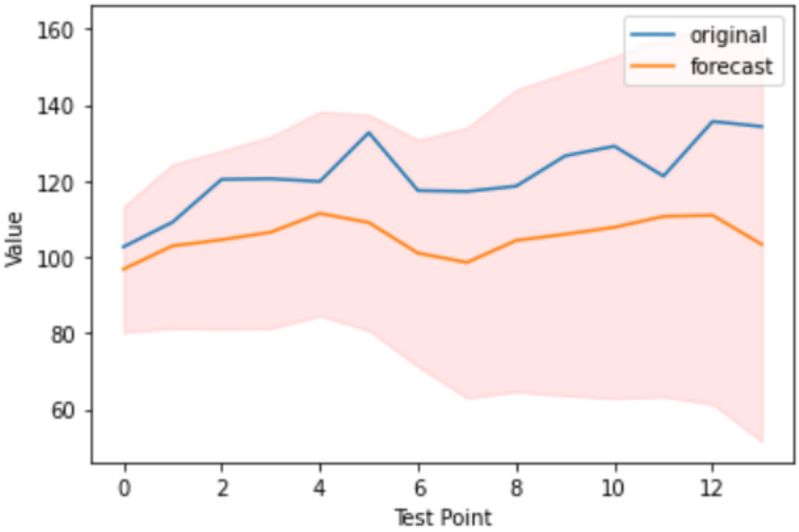
Results:

Metric	Train	Test
MAE	7.24	16.55
MAPE	0.1	0.13
R2	0.84	-3.02

These results show that the test set doesn't perform as well as the train set. In particular, the R2 for the test set indicates the predictions are not accurate.

11) Plot original time series, forecasted values, and confidence intervals.

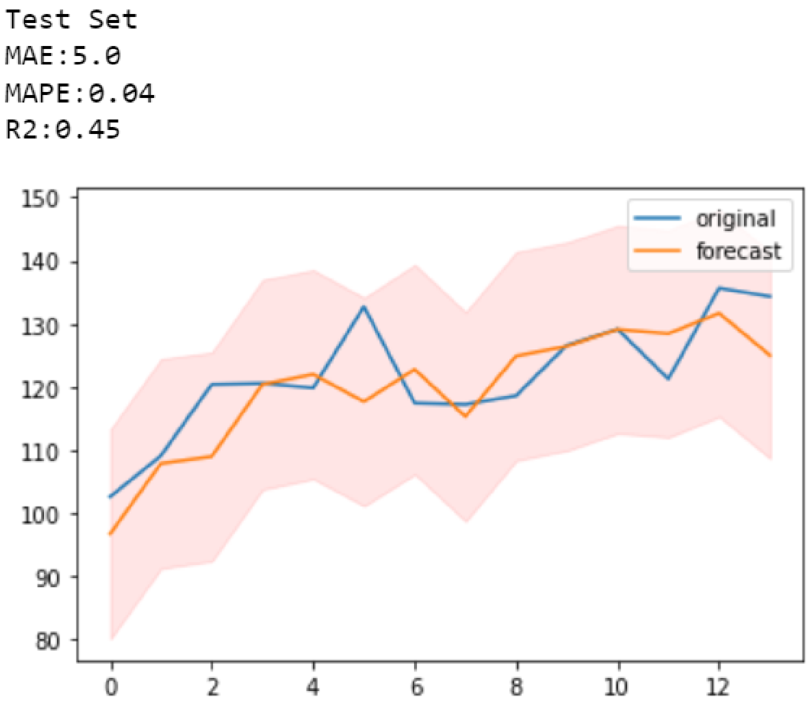
In the following plot we can see for each test data point the value corresponding to the original time series and the forecast. The confidence interval is indicated by the shading.



The plot shows that there is a gap in the prediction and the original time-series values. Additionally, as the test point is further away from the time of the model building, the confidence interval becomes larger.

12) Now repeat steps (7) and (10) by using a model which predicts one step at a time, and each step uses the new data to update the model. Compare the results with the previous model.

The following results are obtained for a forecast made one step at a time, where the model used was updated at each step.



```
Test Set
MAE:5.0
MAPE:0.04
R2:0.45
```

Compared to the previous model this one performs much better, obtaining an R2 for the test set of 0.45. Additionally, the plot shows closer values for the forecast to the original time-series, and a similar size for the confidence interval at every point.

13) Imagine if you could collect more data; which other predictors would you add to the model to improve its performance?

The covid pandemic implied many changes in mobility trends. In particular, many people started working from home, which inevitably affects driving trends. Therefore, it would be interesting to include data on the number of companies that implemented home-office work. Additionally, it would be good to incorporate data on the industries or companies that shut down during the beginning, to account for the decreasing trend, as well as the period where schools where closed.

PROBLEM 5

Answer the following in True/False and provide reason(s).

The reference used for the items related to clustering is the book *Introduction to Data Mining* by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar.

- **For large datasets, agglomerative hierarchical clustering is faster than K-means.**

  FALSE. The time requirements for K-means are modest. They are linear in their in the number of data points: $O(I*K*m*n)$. For agglomerative hierarchical clustering the time requirement is square to the number of data points: $O(n^2)$.

- **In the DBSCAN algorithm, an increase in the value of parameters $MinPts$ and Eps will lead to the identification of more clusters.**

  FALSE. Eps are data objects within a distance/ radius ε from a point. $MinPts$ are the minimum number of points in neighborhood for density. If the number of $MinPts$ increases then the number of clusters becomes smaller. Additionally, as Eps increases clusters start to merge together, therefore obtaining a smaller number of clusters. AS an extreme case, we can imagine the $MinPTs$ equal to the total number of data points and Eps large enough to encompass them all in one cluster. On the other extreme, if $MinPTs$ is equal to 1, and Eps is very small, each data point is its own cluster.

- **If DBSCAN is applied to randomly generated data from a uniform distribution, it will fail to identify any clusters.**

  FALSE. Almost every clustering algorithm finds clusters in datas set that have no natural cluster structure. (Refer to example in *Introduction to Data Mining - Section 8.5 Cluster Evaluation* that shows clusterse found by DBSCAN for 100 points randomly uniformly distributed.)

- **Space complexity of DBSCAN does not increase rapidly for high dimensional datasets.**

  TRUE. For DBSCAN, space complexity for high-dimensional data is O(m), where m is number of data points, because it is only necessary to keep a small amount of data for each point.

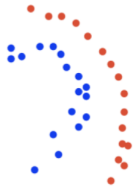
- **K-means algorithm always manages to minimize its objective function.**

  TRUE. K-means always minimizes its objective function SSE. However, this only guarantees finding a local minimum with respect to the SSE because they are based on optimizing the objective function for specific selections of centroids and clusters.

- **Critical intervals of PACF plot will shrink with an increase in time-series length.**
  TRUE. The confidence interval is calculated as $1.96/sqrt(n)$. As n (time-series length) increases, 1.96 will be divided by a larger number, therefore obtaining a smaller number for the upper and lower bounds of the critical interval.

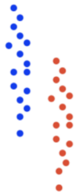
- **Generally, for a time series model, the forecasting accuracy decreases with an increase in the forecasting horizon.**
  TRUE. As the horizon increases there is more uncertainty involved in the prediction. A forecast for next month will be more accurate than a forecast for 6 months. This is why it's necessary to update forecasts as new data becomes available.

- **The following cluster assignment could have been the result of a hierarchical clustering model with a single linkage.**



  TRUE. The single linkage distance method is good at handling non-elliptical shape, in contrast to the complete link method which favors globular shapes.

- **The following cluster assignment could have been the result of the K-means model.**



  FALSE. K-means has difficulty detecting natural clusters when they have non-spherical shapes.

- **The following AR(1) process is not stationary:** $Y_t = 7Y_{t-1} + \varepsilon_t$**.**
  TRUE. A requirement for a stationary AR(1) is that $|\phi_1| < 1$[5], and in this case $|\phi_1| = 7$.

[5] https://online.stat.psu.edu/stat510/lesson/1/1.2