1. Im	dergradute (SIT220) Task porting Libraries: purpose of this code is to import libraries. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer.
impoimpo Num Mata Pano Seab The [2]: weat	ont seaton as spl ont seaborn as spl ont seaborn as spl ont seaborn as spl ont seaborn as sns nPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. plotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy das is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Padding files in notebook. purpose of this code is to read external files such as csv, bt, etc. ther = pd.read_csv('weather.csv', comment = '#') ther.head()
0 1 2 3 4 By	rigi va
0 1 2 3 4	There is the search of the sea
0 1 2 3 4 [9]: weat weat	Figure 1
o 1 2 3 4 11]: weat understand the second temp dewp	EWR 2013
wind wind prec pres visi time dtyp 12]: weat weat 12]: orig year mont day hour temp dewp	<pre>d in float64 d_gust float64 d_gust float64 sure float64 tip float64 tib float64 tib float64 te hour object ther['time_hour'] = pd.to_datetime(weather['time_hour']) ther.dtypes gin object the int64 th int64 th int64 th int64 object float64 co float64 co float64 co float64 co float64</pre>
wind wind prec pres visi time dtyp Chece 13]: pd.u arra Chece	<pre>dir float64 d_speed float64 d_gust float64 cip float64 sure float64 bb float64 bb float64 cb float64 cc ddifferent 'origin' unique(weather['origin']) ay(['EWR', 'JFK', 'LGA'], dtype=object) ck count of 'origin'.</pre>
orig EWR JFK LGA dtyp In 1 Grou	8708 8711
	JFK 2013 1 1 1 3.3 24.08 57.04 250.0 1.692821 1.948065 0.0 1013.5 16.09 2013-01-01 02:00:00 10 JFK 2013 1 1 2 3.9 24.98 56.77 240.0 1.918531 2.207807 0.0 1013.0 16.09 2013-01-01 02:00:00 11 JFK 2013 1 1 3 3.9 26.06 59.37 240.0 1.805676 2.077936 0.0 1013.2 16.09 2013-01-01 04:00:00 12 JFK 2013 1 1 4 3.9 24.98 56.77 260.0 1.467112 1.688323 0.0 1013.2 16.09 2013-01-01 05:00:00 13 LGA 2013 1 1 0 3.9 19.04 44.18 240.0 1.354257 1.558452 0.0 1013.7 16.09 2013-01-01 02:00:00 14 LGA 2013 1 1 1 3.9 19.04 44.18 240.0 1.354257 1.558452 0.0 1013.6 16.09 2013-01-01 02:00:00
16]: JFK JFK. 16]: ir 0 8 1 8 2 8 3 8	
5 8 Hell 17]: dail dail 17]: t 0 20 1 20 2 20 3 20	### ### ##############################
plt. plt. plt. plt.	.figure(figsize = (12, 8)) .plot(daily)FK.time_hour, daily)FK.temp) .title('JFK Daily Average Temperature 2013') .xlabel('Days') .show() JFK Daily Average Temperature 2013 30 - 25 - 20 -
Emperture(Celsi	15
27]: sort sort 27]: t 0 20 1 20 2 20 4 20	tedJFK = dailyJFK.sort_values(by=['temp'], ascending = False).reset_index(drop=True) tedJFK = dailyJFK.sort_values(by=['temp'], ascending = False).reset_index(drop=True) time.hour temp toli3-07-17 30.787500 toli3-07-18 30.195833 toli3-07-19 30.187500 Additional Tasks for Postgraduate (SIT731) Students (*)
In 1	this section, we do the same thing like what we did for JFK. Split data, group by day and calc daily mean temp and plot it all in a graph.
8705 8706 8707 8708 23]: LGA	8 8703 EWR 2013 12 30 19 2.8 21.02 51.95 32.0 1.692821 1.948065 0.0 1017.6 16.09 2013-12-30 20:00:00 8 8705 EWR 2013 12 30 21 0.6 15.98 48.98 32.0 1.467112 1.688323 0.0 1019.8 16.09 2013-12-30 22:00:00 8 8706 EWR 2013 12 30 22 -0.6 12.92 46.74 34.0 1.579967 1.818194 0.0 1020.5 16.09 2013-12-30 23:00:00 8 8707 EWR 2013 12 30 23 -1.7 12.02 48.69 33.0 1.467112 1.688323 0.0 1021.1 16.09 2013-12-30 23:00:00 9 8708 EWR 2013 12 30 23 -1.7 12.02 48.69 33.0 1.467112 1.688323 0.0 1021.1 16.09 2013-12-30 23:00:00
2 3 4 8706 8707 8708 8709	1 1/420
dail 25]: plt. plt. plt. plt. plt. plt. plt. plt.	<pre>lyEWR = EWR.groupby(EWR['time_hour'].dt.date)['temp'].mean().reset_index() lyLGA = LGA.groupby(LGA['time_hour'].dt.date)['temp'].mean().reset_index() .figure(figsize = (12, 8)) .plot(dailyJFK.time_hour, dailyJFK.temp, label='JFK') .plot(dailyLGA.time_hour, dailyEWR.temp, label='EWR') .plot(dailyLGA.time_hour, dailyLGA.temp, label='LGA') .xlabel('Temperture(Celsius)') .legend() .show()</pre>
ture(Celsius)	
As 3. By 26]: #if #if dail	we see, these three airports have similar daily average temp as they are pretty close to each other on the map, so the temp wouldn't be a significant difference. Optional Features using diff function, we can calc the difference between the days next day has greater mean = + next day has greater mean = - next day has greater mean = - next day has greater mean = - next day has smaller mean = - next day has freater mean =
0 20 1 20 2 20 3 20 4 20 27]: plt. plt. for	time_hour temp meanDiff 2013-01-01 3.913636
	5.5
28]: JFK[JFK[#htt	
2 3 4 5 6 7 8 9 10 11	128 2013-02-21 050000 2013-02-21 050000 1516 2013-03-50 060000 2013-03-31 030000 2134 2013-03-31 010000 2013-03-31 030000 2204 2013-04-03 000000 2013-04-03 000000 5375 2013-08-13 040000 2013-08-13 040000 5446 2013-08-19 24000 2013-08-19 24000 5579 2013-08-22 22000 2013-08-23 30000 5670 2013-08-23 300000 2013-08-23 30000 7141 2013-10-26 00000 2013-10-26 00000 710 2013-10-27 01000 2013-10-27 03000
13 14 AS 29]: newJ newJ newJ 2013	736 2013-11-01 07:000 2013-11-01 10:000 2013-11-03 06:000 2013-11-03 06:000 2013-11-03 06:000 2013-11-03 06:000 2013-11-03 06:000 2013-11-04 17:000 2013-11-
2013 2013 2013 2013 2013 2013 2013 2013	3-01-01 03:00:00 8710. JFK 2013.0 1.0 1.0 2.0 3.9 24.98 56.77 240. 1.918531 2.207807 0.0 1013.0 16.09 3-01-01 05:00:00 8711.0 JFK 2013.0 1.0 1.0 3.0 3.9 26.06 59.37 240. 1.805676 2.077936 0.0 1013.2 16.09 3-01-01 05:00:00 8712.0 JFK 2013.0 1.0 1.0 4.0 3.9 24.98 56.77 260.0 1.467112 1.688323 0.0 1013.2 16.09 3-01-01 05:00:00 NaN NaN NaN NaN NaN NaN NaN NaN NaN N
1 2 2 2 2 3 2 4 2 4 2 4 2 4 2 4 4 2 4 4 2 4 4 2 4 4 2 4 4 2 4 4 2 4 4 4 2 4	2013-01-01 01:00:00 8708.0 JFK 2013.0 1.0 1.0 0.0 3.3 23.00 54.51 24.00 1.579967 1.818194 0.0 1014.4 16.09 2013-01-01 02:00:00 8709.0 JFK 2013.0 1.0 1.0 1.0 1.0 3.3 24.08 57.04 25.00 1.692821 1.948065 0.0 1013.5 16.09 2013-01-01 03:00:00 8710.0 JFK 2013.0 1.0 1.0 1.0 2.0 3.9 24.98 56.77 24.00 1.918531 2.207807 0.0 1013.2 16.09 2013-01-01 04:00:00 8711.0 JFK 2013.0 1.0 1.0 1.0 3.0 3.9 26.06 59.37 24.00 1.805676 2.077936 0.0 1013.2 16.09 2013-01-01 05:00:00 8712.0 JFK 2013.0 1.0 1.0 1.0 4.0 3.9 24.98 56.77 26.00 1.467112 1.688323 0.0 1013.2 16.09 2013-01-01 06:00:00 8712.5 NaN 2013.0 1.0 1.0 5.0 3.9 25.52 58.07 26.00 1.354257 1.558452 0.0 1012.6 16.09 2013-01-01 08:00:00 8714.0 JFK 2013.0 1.0 1.0 5.0 3.9 26.06 59.37 27.00 1.281402 1.428581 0.0 1012.6 16.09 2013-01-01 08:00:00 8714.0 JFK 2013.0 1.0 1.0 5.0 3.9 26.06 59.37 27.00 1.281480 1.298710 0.0 1012.6 16.09 2013-01-01 08:00:00 8715.0 JFK 2013.0 1.0 1.0 8.0 4.4 26:06 59.50 26.00 1.467112 1.688323 0.0 1012.6 16.09 2013-01-01 08:00:00 8715.0 JFK 2013.0 1.0 1.0 8.0 4.4 26:06 59.50 26.00 1.467112 1.688323 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 8.0 4.4 26:06 59.50 26.00 1.467112 1.688323 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-01-01 09:00:00 8715.0 JFK 2013.0 1.0 1.0 9.0 4.4 28.04 62.1 25.00 1.692821 1.948065 0.0 1012.6 16.09 2013-
dail 0 20 1 20 2 20 3 20 4 20 5 20 6 20 7 20 8 20 9 20 812]: fig, fig ax1.	
ax2.	.plot(dailyJFK.time_hour, dailyJFK.temp, 'r') atplotlib.lines.Line2D at 0x184feedeaf0>] Interpolate vs Non-Interpolate