Homework9_DS-GA1013

April 28, 2021

Name: Zhuoyuan Xu (Kallen)

NetID: zx1137

Problem 1

(a) The sound propagation can be characterized by the convolution of the emitted sound and an impulse response if we can assume the system is a linear, translation invariant function.

(b) The impulse response can be explained as the output signal that represents the reaction of a system (as a function of time in this problem) in response to some external changes (an impulse which is a basis, or a really short sound covering a large frequency band). Thus, the value of zero may indicate the system does not create some response before 25 ms. This may relates to the time of the sound reaching the receiver/collector, i.e. the time needed for the signal travelling from the source to the receiver.

(c) The first spike corresponds to the smallest time, and has the highest "y" value on this plot, and this correspond to the sound of source. Similarly, the second spike corresponds to the second smallest time, and the first echo/reveberation. Also, according to my research online about some terminologies, the first spike represents the impulse response to the direct sound which has high sound pressure but short duration. The second spike represents the first order or early reflection of the sound.

(d) One way to measure the impulse responses by bursting a balloon is to first set up receivers in omnidirectional mode, set the receiver input volume low to prevent potential loss of wave-form sound information and put them to the chosen positions. Then, we allow the room to become silent. After that, we can pop a balloon quickly and wait until the reverberation of impulses completes and the room returns to a normal noise floor. In this experiment, bursting a balloon gives a loud and short sound that has relatively uniform radiation pattern, and covers a large frequency band. This assumes the system is transition invariant for calculation.

An limitation of the approach is that bursting a balloon is not translation invariant and it decays as time goes on. The types and the sizes of the chosen balloon can affect the energy and performances in different frequency bands and other conditions that affect the final measurements of the impulse response. It is hard to control the experiment. Bursting the balloon is only an approximation to the theories.

(e) This experiment is similar to the commonly known sine sweep. The chosen sinusoidal sound with a specific frequency has its DFT to be a spike. We can measure the sinusoidal sound and the corresponding output of the system, and then perform DFT on both of them to get corresponding DFT coefficients. Since sinusoidal sound can be viewed as linear translation-invariant, we can see that the DFT coefficients of the output can be calculated by the multiplication of the DFT coefficients of the sinusoidal sound and those of the impulse response. Then we can get the DFT coefficients of the impulse response, and perform IDFT to get the actual impulse response itself. We can perform such experiment with known sinusoidal sounds in a wide frequency range.

Problem 2

Given $x * y = x$, from Theorem 3.5 of the lecture notes,

$$\hat{x[k]} = \hat{x[k]}\hat{y[k]}$$

Since x is bandlimited with $\hat{x[k]} = 0$ for all $|k| > k_c$, we can get

$$\hat{y[k]} = 1 \text{ for all } |k| \leq k_c$$

and some arbitrary number for $|k| > k_c$.

From IDFT, the n-th entry of y can be written as

$$\begin{aligned}
y[n] &= \frac{1}{N}\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \hat{y[k]} exp(\frac{i2\pi kn}{N}) \\
&= \frac{1}{N}(\hat{y}[-\frac{N-1}{2}]exp(\frac{i2\pi n}{N}\frac{-(N-1)}{2}) + ... + \hat{y}[-k_c]exp(\frac{i2\pi n}{N}(-k_c)) + ... \\
&+ \hat{y}[k_c]exp(\frac{i2\pi n}{N}(k_c)) + ... + \hat{y}[\frac{N-1}{2}]exp(\frac{i2\pi n}{N}\frac{N-1}{2})) \\
&= \frac{1}{N}(\hat{y}[-\frac{N-1}{2}]exp(\frac{i2\pi n}{N}\frac{-(N-1)}{2}) + ... \\
&+ 1exp(\frac{i2\pi n}{N}(-k_c)) + ... + 1exp(\frac{i2\pi n}{N}(k_c)) + ... + \hat{y}[\frac{N-1}{2}]exp(\frac{i2\pi n}{N}\frac{N-1}{2}))
\end{aligned}$$

We can write the l_2 norm of y as

$$||y||_2 \propto y^T y \propto \frac{1}{N^2}\hat{y}^T F F^* \hat{y} \propto \hat{y}^T \hat{y}$$

and this shows the l_2 norm of y is proportional to the l_2 norm of the DFT coefficients.

Since y is a vector with the smallest l_2 norm which should be larger than or equal to 0, from the above expression of y, y can have $\hat{y}[k] = 0$ for all $|k| > k_c$ or the sum of terms having $|k| > k_c$ equal to 0 since we have already determined the DFT coefficients of y from $-k_c$ to k_c.

The previous equation can be written as

$$\begin{aligned}
y[n] &= \frac{1}{N}\sum_{k=-k_c}^{k_c} exp(\frac{i2\pi kn}{N}) \\
&= \frac{1}{N}(exp(\frac{i2\pi n}{N}(-k_c)) + ... + exp(\frac{i2\pi n}{N}k_c)) \\
&= \frac{1}{N}\frac{exp(\frac{i2\pi n(-k_c)}{N}) - exp(\frac{i2\pi(k_c+1)}{N})}{1 - exp(\frac{i2\pi n}{N})} \\
&= \frac{sin(\frac{n\pi}{N}(2k_c+1))}{Nsin(\frac{n\pi}{N})}
\end{aligned}$$

From the above equation, we can see that entries of y should be real-valued.

Problem 3

(a) If we define $\phi_k[j] = exp(-\frac{2\pi kj}{N})$, then the DFT coefficients of the k-th term of $a_{\tilde{x}}$ can be represented as

$$\begin{aligned}
a\hat{x}[k] &= \sum_{j=0}^{N} a_{\tilde{x}}[j]\phi_k[j] \\
&= a_{\tilde{x}}[0] + \sum_{i=1}^{\frac{N-1}{2}} (a_{\tilde{x}}[i]\phi_k[i] + a_{\tilde{x}}[N-i]\phi_k[N-i])
\end{aligned}$$

From definition 4.1, $a_{\tilde{x}}[n] = a_{\tilde{x}}[-n] = a_{\tilde{x}}[N-n]$, so the above expression can be written as

$$\begin{aligned}
a\hat{x}[k] &= a_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (a_{\tilde{x}}[p]\phi_k[p] + a_{\tilde{x}}[N-p]\phi_k[N-p]) \\
&= a_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (a_{\tilde{x}}[p](\phi_k[p] + \phi_k[N-p])) \\
&= a_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (a_{\tilde{x}}[p](cos(\frac{2\pi kp}{N}) + isin(\frac{2\pi kp}{N}) + cos(\frac{2\pi k(N-p)}{N}) + isin(\frac{2\pi k(N-p)}{N}))) \\
&= a_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (2a_{\tilde{x}}[p]cos(\frac{2\pi kp}{N}))
\end{aligned}$$

Since $a_{\tilde{x}}$ has real values, $a_{\tilde{x}}[0]$ is real. The DFT coefficients of $a_{\tilde{x}}$ are real.

We can also write the expression for $\hat{a}_{\tilde{x}}[N-k]$ as

$$\begin{aligned}
a\hat{x}[N-k] &= \sum_{j=0}^{N} a_{\tilde{x}}[j]\phi_{N-k}[j] \\
&= a_{\tilde{x}}[0] + \sum_{i=1}^{\frac{N-1}{2}} (a_{\tilde{x}}[N-i]\phi_k[N-i] + a_{\tilde{x}}[i]\phi_k[i]) \\
&= a_{\tilde{x}}[0] + \sum_{i=1}^{\frac{N-1}{2}} a_{\tilde{x}}[i](\phi_k[N-i] + \phi_k[i])
\end{aligned}$$

This expression will have the similar result with that of $\hat{a}_{\tilde{x}}$, and thus $\hat{a}_{\tilde{x}}[k] = \hat{a}_{\tilde{x}}[N-k]$ for $k = 1, ..., \frac{N-1}{2}$.

(b)

$$
\begin{aligned}
\Sigma_{\tilde{x}}[j_2, j_1] &= e_{j_2} a_{\tilde{x}}^{\downarrow j_1} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} exp(-\frac{i2\pi k j_1}{N}) \hat{a}_{\tilde{x}}[k] \phi_k[j_2] \\
&= \frac{1}{N} \sum_{k=0}^{N-1} exp(-\frac{i2\pi k j_1}{N}) exp(\frac{i2\pi k j_2}{N}) \hat{a}_{\tilde{x}}[k] \\
&= \frac{1}{N} \sum_{k=0}^{N-1} exp(\frac{i2\pi k (j_2 - j_1)}{N}) \hat{a}_{\tilde{x}}[k]
\end{aligned}
$$

with e_{j_2} a vector that only has 1 at its j_2-th position and others all 0.

(c) Given the expression $\Sigma_{\tilde{x}}[j_2, j_1] = \frac{1}{N} \sum_{k=0}^{N-1} exp(\frac{i2\pi k(j_2 - j_1)}{N}) \hat{a}_{\tilde{x}}[k]$ and $\hat{a}_{\tilde{x}}[k] = \hat{a}_{\tilde{x}}[N-k]$ for $k = 1, ..., \frac{N-1}{2}$,

$$
\begin{aligned}
\Sigma_{\tilde{x}}[j_2, j_1] &= \frac{1}{N} \sum_{k=0}^{N-1} exp(\frac{i2\pi k(j_2 - j_1)}{N}) \hat{a}_{\tilde{x}}[k] \\
&= \frac{1}{N} (\hat{a}_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (\hat{a}_{\tilde{x}}[p] exp(\frac{i2\pi p(j_2 - j_1)}{N}) + \hat{a}_{\tilde{x}}[N-p] exp(\frac{i2\pi (N-p)(j_2 - j_1)}{N}))) \\
&= \frac{1}{N} (\hat{a}_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (2\hat{a}_{\tilde{x}}[p] cos(\frac{2\pi p(j_2 - j_1)}{N})) \\
&= \frac{1}{N} (\hat{a}_{\tilde{x}}[0] + \sum_{p=1}^{\frac{N-1}{2}} (2\hat{a}_{\tilde{x}}[p] (cos(\frac{2\pi p j_2}{N}) cos(\frac{2\pi p j_1}{N}) + sin(\frac{2\pi p j_2}{N}) sin(\frac{2\pi p j_1}{N}))) \\
&= \frac{1}{N} \hat{a}_{\tilde{x}}[0] + \frac{2}{N} \sum_{p=1}^{\frac{N-1}{2}} (\hat{a}_{\tilde{x}}[p] sin(\frac{2\pi p j_2}{N}) sin(\frac{2\pi p j_1}{N}) + \hat{a}_{\tilde{x}}[N-p] cos(\frac{2\pi (N-p) j_1}{N}) cos(\frac{2\pi (N-p) j_2}{N}))
\end{aligned}
$$

The above expression is exactly the decomposition $\Sigma_{\tilde{x}}[j_2, j_1] = \sum_{k=0}^{N-1} \lambda_k u_k[j_1] u_k[j_2]$ given the expressions of the orthonormal vectors. Also, the values of $\lambda_0, ..., \lambda_k$ is the DFT coefficients $\hat{a}_{\tilde{x}}$, so $\lambda_0 = \hat{a}_{\tilde{x}}[0], ..., \lambda_k = \hat{a}_{\tilde{x}}[k]$ correspondingly.

Problem 4

(a)

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import acf, acovf
from statsmodels.graphics.tsaplots import plot_acf
```

[2]:
```
power_data = pd.read_csv('household_power_consumption_days.csv')
```

[3]:
```
power_data['Global_active_power(kW)'].isnull().sum()
```

[3]: 0

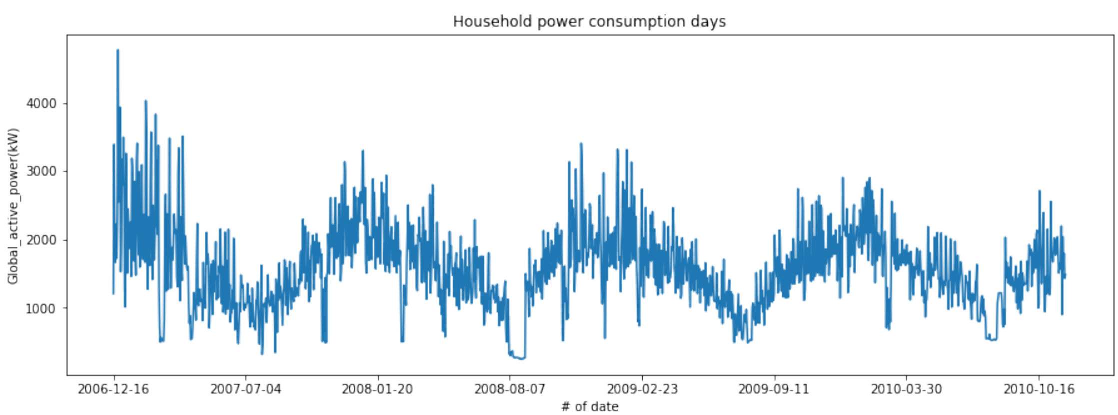
[4]:
```
power_data
```

[4]:

	datetime	Global_active_power(kW)
0	2006-12-16	1209.176
1	2006-12-17	3390.460
2	2006-12-18	2203.826
3	2006-12-19	1666.194
4	2006-12-20	2225.748
…	…	…
1437	2010-11-22	2041.536
1438	2010-11-23	1577.536
1439	2010-11-24	1796.248
1440	2010-11-25	1431.164
1441	2010-11-26	1488.104

[1442 rows x 2 columns]

[5]:
```
# plot original data to observe changes
fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(power_data['datetime'], power_data['Global_active_power(kW)'])
ax.set_title('Household power consumption days')
ax.set_xlabel('# of date')
ax.set_ylabel('Global_active_power(kW)')
ax.xaxis.set_major_locator(plt.MaxNLocator(10))
```

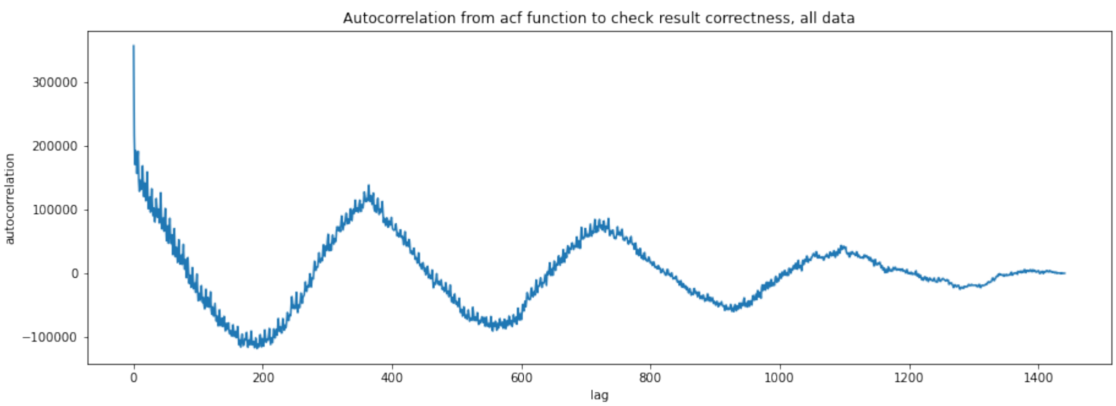


[6]:
```
# use acf to check the result
```

```
year_covaf = acf(power_data['Global_active_power(kW)']-np.
↪mean(power_data['Global_active_power(kW)']), fft=False,␣
↪nlags=len(power_data['Global_active_power(kW)'])-1)
year_covaf = np.var(power_data['Global_active_power(kW)']) * year_covaf
```

[7]:
```
fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(year_covaf)
ax.xaxis.set_major_locator(plt.MaxNLocator(10))
ax.set_xlabel('lag')
ax.set_ylabel('autocorrelation')
ax.set_title('Autocorrelation from acf function to check result correctness,␣
↪all data')
```

[7]:
```
Text(0.5, 1.0, 'Autocorrelation from acf function to check result correctness,
all data')
```



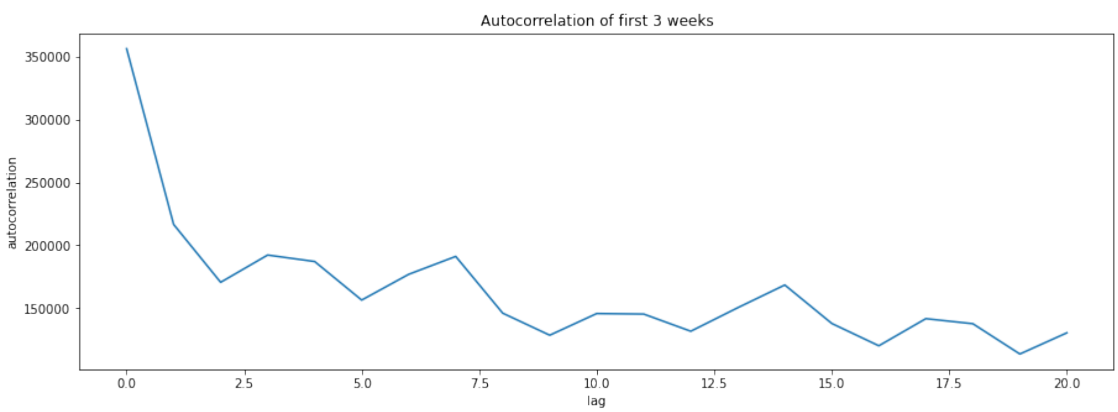
[8]:
```
# manually calculate autocorrelation
def autocorrelation(x):
    N = len(x)
    mean = np.mean(x)
    x_center = np.array([i-mean for i in x])
    autocorr = np.correlate(x_center, x_center, mode='full')
    res = autocorr[autocorr.size//2:]/autocorr[autocorr.size//2]
    res = res * np.var(x)

    #atcorr = [1]
    #for i in range(1, len(x)):
    #    temp = np.hstack((xp[N-i:N],xp[0:N-i]))
    #    current_corr = np.corrcoef(xp, temp)[0, 1]
    #    atcorr.append(current_corr)
    #x_mat = np.array(x)
    #for i in range(1, len(x)):
```

```
#     temp = np.hstack((x[N-i:N],x[0:N-i]))
#     x_mat = np.vstack((x_mat, np.array(temp)))
#covariance = np.cov(x_mat)[0]
return res
```

[9]:
```
covaf = autocorrelation(power_data['Global_active_power(kW)'])
```

[10]:
```
fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(covaf[0:21])
ax.xaxis.set_major_locator(plt.MaxNLocator(10))
ax.set_title('Autocorrelation of first 3 weeks')
ax.set_xlabel('lag')
ax.set_ylabel('autocorrelation')
```

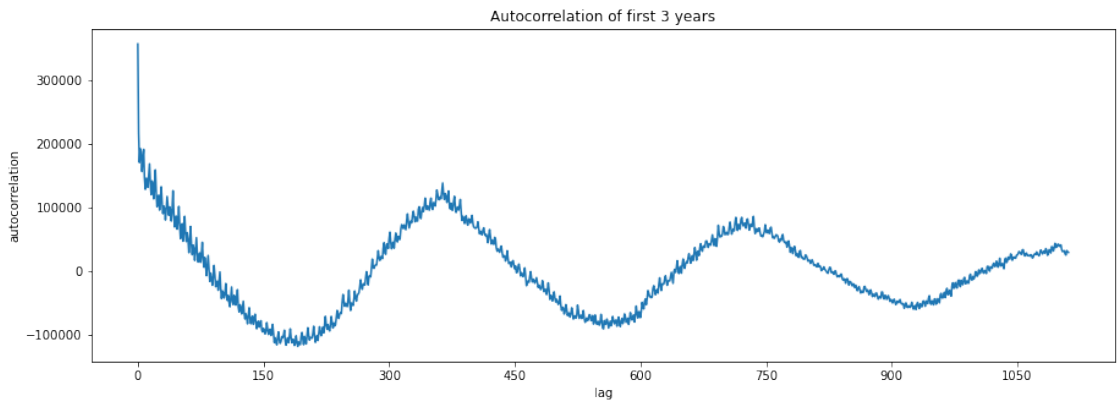
[10]:
```
Text(0, 0.5, 'autocorrelation')
```



The above plot represents the autocorrelation from the first 3 weeks window. It shows a generally decreasing trend with minor fluctuations. Autocorrelation is a mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. The decreasing trend in our plot shows a decrease in similarity as lags increase.

[11]:
```
fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(covaf[power_data['datetime'] < '2010-01-01'])
ax.xaxis.set_major_locator(plt.MaxNLocator(10))
ax.set_title('Autocorrelation of first 3 years')
ax.set_xlabel('lag')
ax.set_ylabel('autocorrelation')
```

[11]:
```
Text(0, 0.5, 'autocorrelation')
```

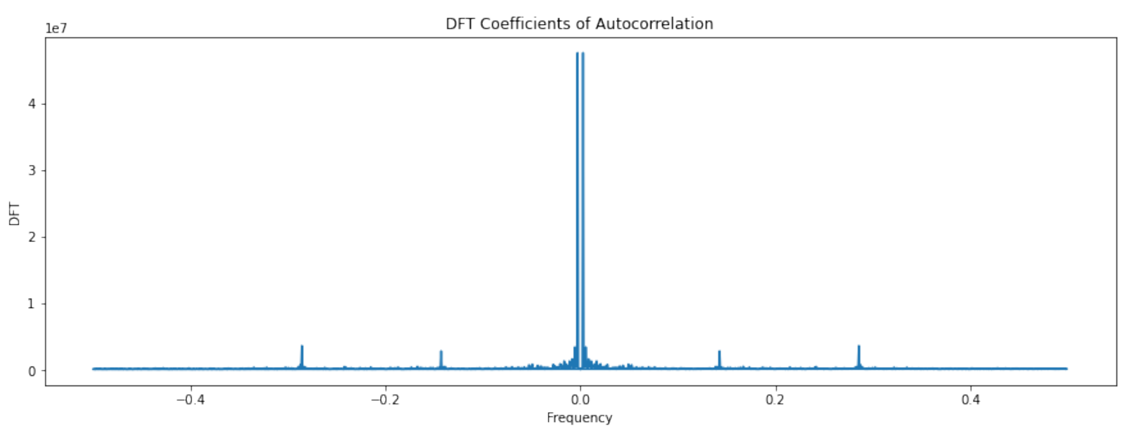


The above plot has a larger window size of the first 3 years. Different from the previous plot, the plot shows an oscillating trend around 0, with the oscillation decreasing as lag increases. This shows the similarity of the time series with its lagged versions first decreases and then increases, and so on. We can see there is a periodical variation in our autocorrelation which is consistent with our original data.

Also, from the above autocorrelation plots, it seems the given dataset is not stationary. Generally, the autocorrelation will drop to 0 relatively quickly. However, our plots do not fit into this observation because the autocorrelation values oscillates around 0 as the number of lags increases.

(b)

[12]:
```
dft_year = np.fft.fft(year_covaf)
year_freq = np.fft.fftfreq(len(dft_year))

fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(year_freq, np.real(dft_year))
ax.set_title('DFT Coefficients of Autocorrelation')
ax.set_xlabel('Frequency')
ax.set_ylabel('DFT')
plt.savefig('dftyear.pdf')
```



The magnitude plot of the DFT coefficients is symmetric, and has its maximum spikes very close to frequency = 0. Spikes further away from 0 than the maximum have much smaller values than the maximum, but they have values relatiely close to each other. The spikes that seems to be the closest to the maximum have magnitudes almost similar to those of spikes at frequency close to -0.25, -0.15, 0.15 and 0.25. This can be a helpful tool for identifying the dominant cyclical behavior or periodicity of the given data. Our plot seems to be consistent with the original data because the original data shows a periodic behavior that has local minimums in “summer” time (about July) and maximum in “winter” time (about December or January).

(c) The stationary signal assumes that a system do not significantly change overtime. The assumptions help simplify our calculation. For example, the mean and variance would not change significantly as time goes under the assumption, and it is consistent with the need to assume that each data point is independent of one another. However, this assumption can hardly be satisfied in real-world application, and also it does not show trends or seasonality of the dataset.