2022 2 학기 SDA 프로젝트 보고서

2022.12.15. 데이터테크놀로지학과, 김광영(60211642), 송지은(60211670), 신유빈(60211673)

1. 초기 문제정의

1.1. 기획 의도

최근 다시 뉴스나 신문같은 여러 미디어에서 COVID-19 우세종에 대한 이야기가 나오고 있다. 팀원 중 김광영의 가족 구성원 중 대다수가 간호사인 관계로 감염병에 대한 관심이 다른 학우들보다 많아 그러한 이슈를 더 관심있게 지켜보게 되었다. 델타 변이가 우세종일 때 보았던 뉴스에서 “바이러스가 진화할 수록 치명률은 낮아지고 감염률은 높아지게 진화한다.”라는 문장이 기억에 남아 데이터 분석을 통해 이 문장이 사실인지 알아보기 위해 실험을 계획하게 되었다.

1.2. 문제 정의

A. 우리 조는 “코로나 바이러스가 진화를 거듭함에 따라 감염률이 증가하고 치명률이 감소한다”는 가정을 증명할 것이다.

B. 현재 한국에서 보인 코로나 바이러스의 우세종은 델타, 오미크론, 그 외 이렇게 세 가지로 분류할 수 있다.

C. 이에 따라 귀무가설(H0)은 “코로나 바이러스가 진화를 거듭해도 감염률과 치명률은 변화가 없다.”로 세우고 이를 기각할 수 있음을 증명할 것이다.

1.3. 단어 정의

A. 우세종은 국내 검출율(국내 + 해외유입사례)이 50%를 넘을 때를 기준으로 파악한다.

B. 감염률(p_i)는 1 일 감염자수/(전체 국민 수 - 전날까지의 누적 감염자 수)로 계산한다. (여기서 i 는 infection 의 약자)

C. 치명률(p_l)은 1 일 사망자수/누적확진자수로 계산한다. (여기서 l 은 lethality 의 약자)

2. 실행 계획

2.1. 확진자와 사망자에 대한 데이터는 한국의 질병관리청 홈페이지를 이용해서 얻을 것이다.

2.2. 우세종은 WHO 에서 얻은 변이에 대한 정보와 밑의 차트를 참고한다.

A. 22 년 1 월 24 일 - 오미크론 우세종

B. 21 년 7 월 25~31 일 - 델타 변이 우세종

2.3. 3 개의 집단으로 분류됨으로 ANOVA 를 사용해 집단 간 평균의 다름을 증명하고, 이후 2 개 집단에 대한 비교를 진행 해 감염률의 상승과 치명률의 감소를 볼 것이다.

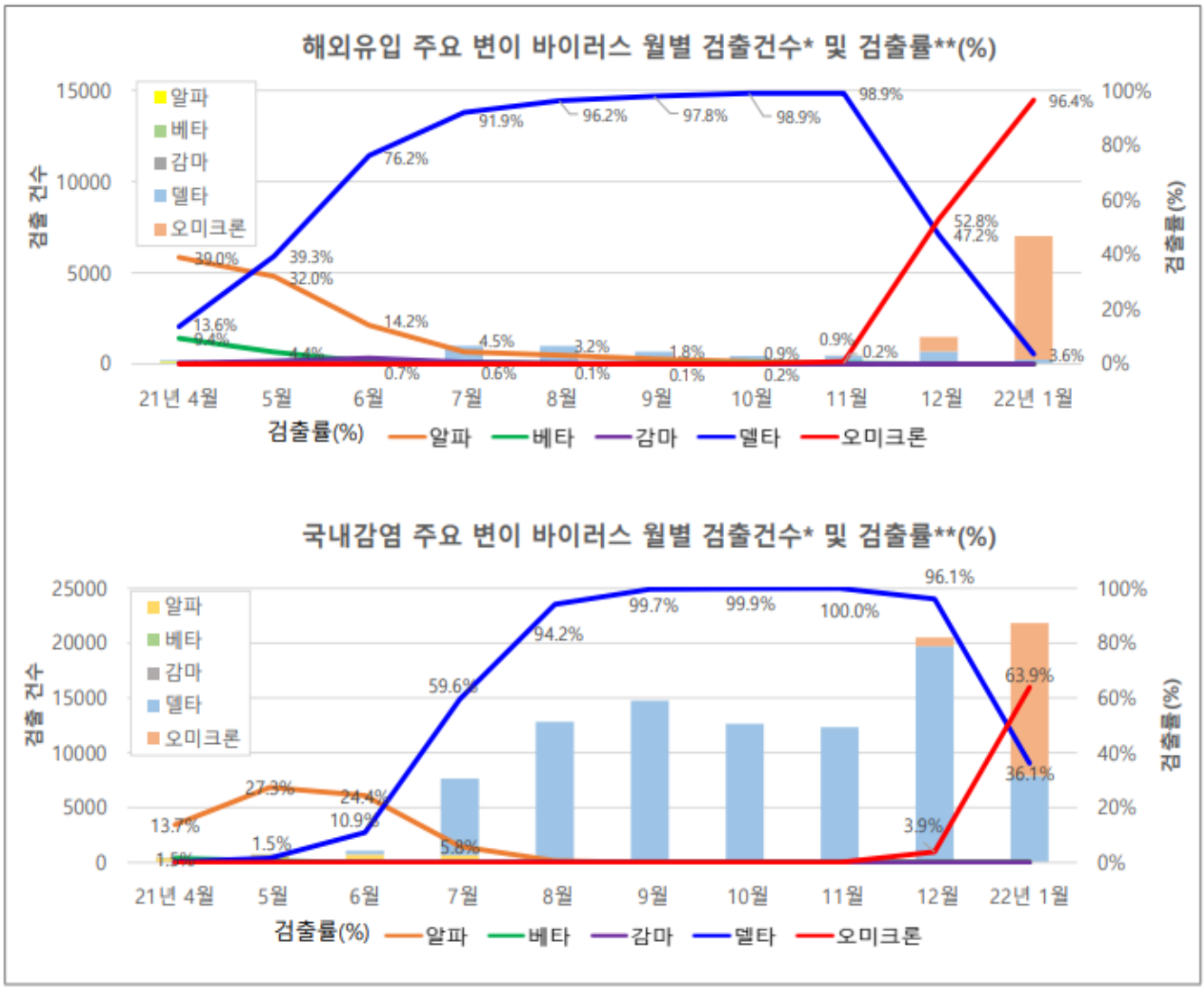


그림 3. 주요 변이 바이러스(VOC) 확인 건수 및 월별 검출률

*확진일 기준, **변이 바이러스 검출률(%) = (변이 바이러스 수 / 분석 건수) ×100

3. 팀프로젝트 결과 보고서

3.1. 단어 정의 수정

A. 설계를 하고 분석을 실행하면서 치명률의 정의에서 2 가지 문제가 발생했다. 첫 번째 문제는 처음 정의한 치명률의 정의에서 야기됐는데, 코로나 발병 초기에서 치명률이 너무나 작아 검정의 진행되지 않았다. 따라서 이는 한겨레의 미래&과학 기사(https://www.hani.co.kr/arti/science/science_general/964555.html)에 나온 기사의 치명률 정의를 따르기로 했다. 두 번째 문제는 사망자의 사망원인이 사망자로 집계된 당일의 우세종에 영향을 받았나 하는 사실이다. 그래서 우리는 이를 17 일 이전에 확진자에서 사망자가 나올 확률이 높다고 이야기했다.

B. 치명률 : (누적 사망자 수 / 17 일 전 누적 확진자 수)로 계산한다.

3.2. 진행 과정

A. 데이터를 먼저 날짜를 기준으로 3 가지 군으로 분류했다. 여기서는 오미크론, 델타, 그 외 변이로 분류했다.

B. 이후 감염률, 치명률을 위에서 내린 정의에 따라 구하고 신천지 집단 감염 사태의 데이터를 outlier 처리해 데이터에서 제외했다.

C. 이후 정규성 검증인 Shapiro test 와 등분산검증인 fligner test 를 진행했다.

D. 세 집단의 평균의 다름을 증명해야 하기 때문에 Kruskal-Wallis Test 를 진행해 다름을 밝혔다. 위에서부터 그 외, 델타 변이, 오미크론 순이다.

```
[1. 감염률 데이터 분석]
[감염률 정규성 검증]
Shapiro Test-statistics : 0.8539105653762817, p-value : 7.790196369742615e-21
Shapiro Test-statistics : 0.835544228553772, p-value : 7.48299046424844e-13
Shapiro Test-statistics : 0.805178165435791, p-value : 7.125169911372918e-19

[감염률 등분산 검증]
Fligner Test-statistics : 683.7901745808464, p-value : 3.2873822050209747e-149

[감염률 Kruskal-Wallis Test]
Kruskal-Wallis Test-statistics : 804.3463264524516, p-value : 2.1797889228493447e-175
```

E.

```
[2. 치명률 데이터 분석]
[치명률 정규성 검증]
Shapiro Test-statistics : 0.9608435034751892, p-value : 4.450002033529188e-10
Shapiro Test-statistics : 0.9434995055198669, p-value : 1.8121364746548352e-06
Shapiro Test-statistics : 0.5536781549453735, p-value : 3.1924320684309763e-27

[치명률 등분산 검증]
Fligner Test-statistics : 148.9384212608489, p-value : 4.554416576790575e-33
[치명률 Kruskal-Wallis Test]
Kruskal-Wallis Test-statistics : 798.2136625406547, p-value : 4.678487671713772e-174
```

F. 위의 결과를 살펴보면, delta, omi, etc 모두 정규성을 띄지 않고, 등분산성을 가지지 않는다. 또한 Kruskal Wallis test 의 p-value 가 적정 유의수준 0.05 보다 작아서 귀무가설인 각 군의 평균이 같다를 기각한다. 따라서 감염률과 치명률의 각 군의 평균이 적어도 하나 다름을 관찰 할 수 있다.

```
# Wilcoxon rank-sum test 실행

# df_covid_etc
# df_covid_delta
# df_covid_omi

print("감염률")
test_stat, p_val = stats.ranksums(df_covid_omi['감염률'],df_covid_delta['감염률'])
print("rank-sum Test-statistics omi-delta : {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.ranksums(df_covid_etc['감염률'],df_covid_delta['감염률'])
print("rank-sum Test-statistics etc-delta: {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.ranksums(df_covid_omi['감염률'],df_covid_etc['감염률'])
print("rank-sum Test-statistics omi-ect: {}, p-value : {}".format(test_stat, p_val))

print("치명률")
test_stat, p_val = stats.ranksums(df_covid_omi['치명률'],df_covid_delta['치명률'])
print("rank-sum Test-statistics omi-delta : {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.ranksums(df_covid_etc['치명률'],df_covid_delta['치명률'])
print("rank-sum Test-statistics etc-delta: {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.ranksums(df_covid_omi['치명률'],df_covid_etc['치명률'])
print("rank-sum Test-statistics omi-ect: {}, p-value : {}".format(test_stat, p_val))
```

```
감염률
rank-sum Test-statistics omi-delta : 17.552398698639596, p-value : 5.702014276527441e-69
rank-sum Test-statistics etc-delta: -19.15796396197209, p-value : 8.305931597068906e-82
rank-sum Test-statistics omi-ect: 23.91238424208976, p-value : 2.276712518052433e-126
치명률
rank-sum Test-statistics omi-delta : -17.272640142245073, p-value : 7.560317451411355e-67
rank-sum Test-statistics etc-delta: 17.197760405441255, p-value : 2.759946139893418e-66
rank-sum Test-statistics omi-ect: -21.255452659635345, p-value : 2.935181598757898e-100
```

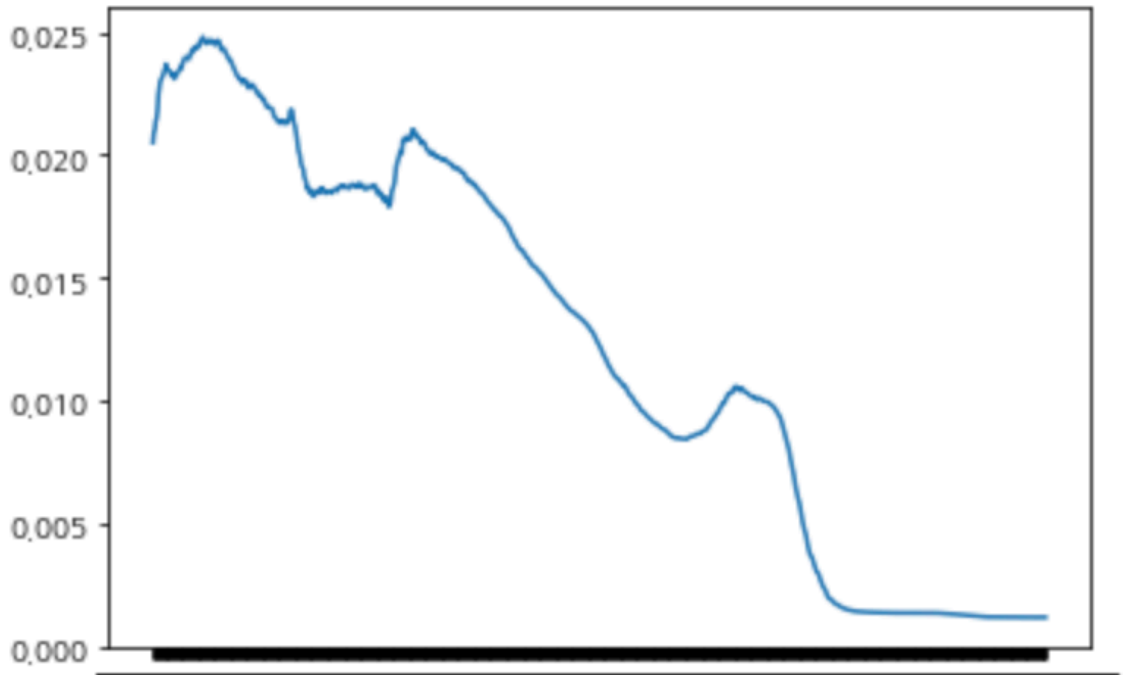
G. 마지막으로 각각의 군끼리 rank-sum test 를 실행한다. 각각 군들의 p-value 가 모두 유의수준 0.05 보다 작으므로 귀무가설을 기각한다. 따라서 각 군들의 평균은 다르다는 결론을 얻을 수 있다. bartlett test 를 통해 유의수준을 0.05/3 까지 낮췄어도 귀무가설을 기각 할 수 있다.

3.3. 설계 평가 및 문제점

실험의 데이터, 실험 설계 문제가 있었다고 생각한다.

A. 이 실험에서 유의미한 데이터를 얻어 내고 싶었으면, 코로나 상황 초기에 있었던, 구로 콜센터, 신천지, 이태원 클럽 이 세가지 집단 감염에서의 n 차 감염자들의 데이터를 모두 제거해야 했다. 즉, outlier 의 정의와 처리가 매우 미숙했다. 처음에 예상 했던 "치명률은 변이가 진행할 수록(코로나 발병 후 시간이 진행 될 수록)

감소한다"와는 다른 구간이 4 곳이나 보인다.



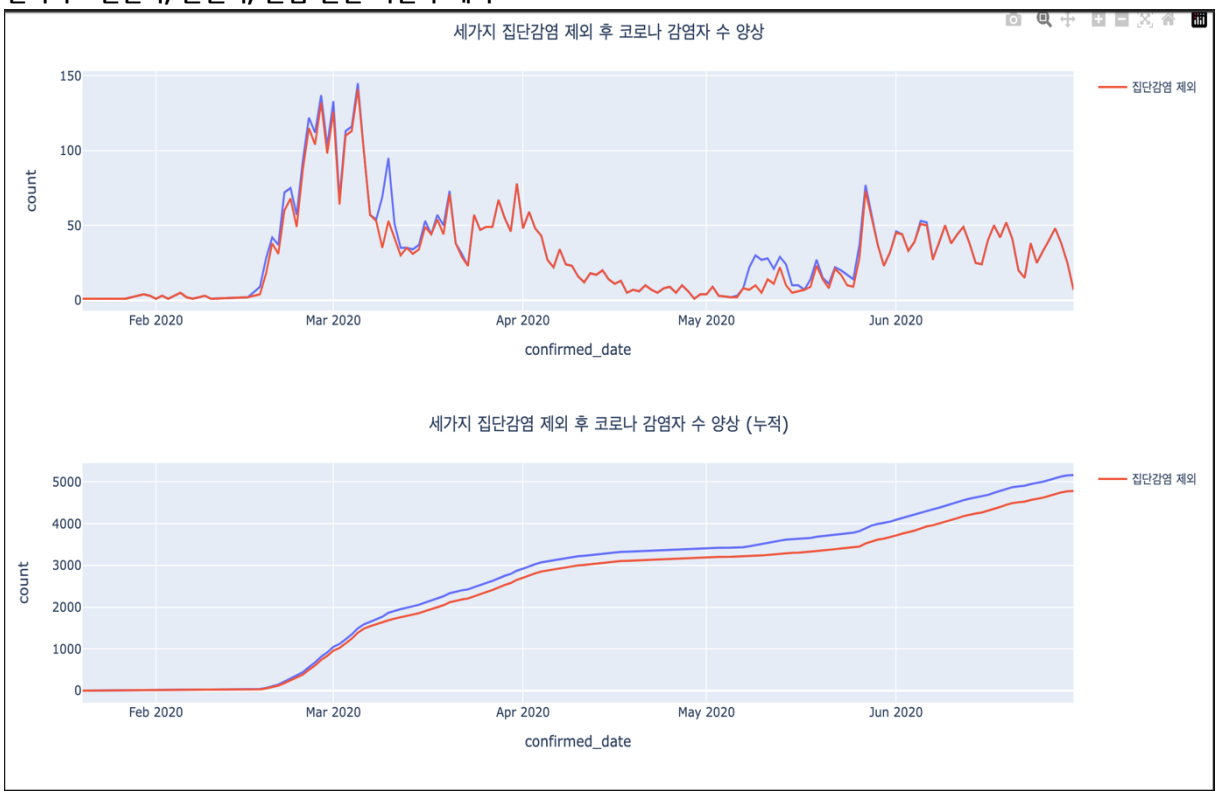
B. 또한 우리가 SDA 수업시간에 배운 다중 선형 회귀 분석까지의 내용은 처음에 설계한 실험 목적인 "k 배의 감염률을 가지고 p 배의 치명률을 가진다"를 유도하기에는 충분하지 않았다.

C. 또한 분석 과정에서 우리가 배운 정규성검정이나 등분산성검정은 유의미하지 않았던 실험이라고 생각한다. 실험에서 증명을 원하는 데이터는 각 군별로 감염자 비율은 증가해야 했고, 치명률은 계속 낮아져야 했다. 따라서 각 군들은 정규분포를 따를 수 없다.

D. 특정 변이가 우세종이라고 해서, 모두 그 변이의 환자라는 가정을 실험을 진행할 때는 인지하지 못하고 기저사실로 실험을 진행했는데, 정확한 데이터를 살펴보기를 원했으면 각 변이의 비율을 실험을 진행할 때 반영해야 했다.

3.4. 추가 분석 : 집단 감염자 제외 시도

A. 전처리 : 신천치, 콜센터, 클럽 관련 확진자 제외



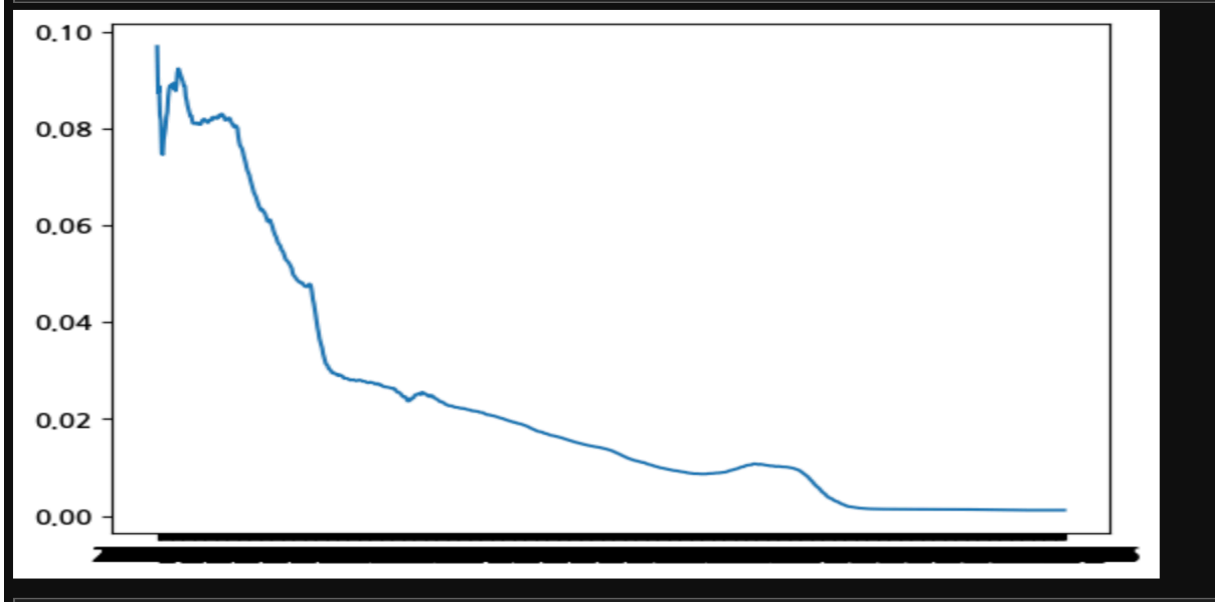
B. 데이터가 정제된 pc_res 의 일일 감염자 수를 구해 원래 우리가 사용했던 데이터의 알맞은 날짜까지 교체했다.

C. 데이터가 사망자 수는 구할 수 없다, 하지만 감염자수가 변경되면 치명률도 이 데이터에 맞게 수정될 것이 다라고 예측하고 실험을 진행한다.

D. 치명률의 plot 이다. 제일 처음 그래프가 신천치 데이터를 제거 하고 실행했던 처음 실험이다. 두 번째 그래프는 이상치 제거 후 그래프이며, 마지막 그래프는 이상치 제거가 없는 그래프이다.

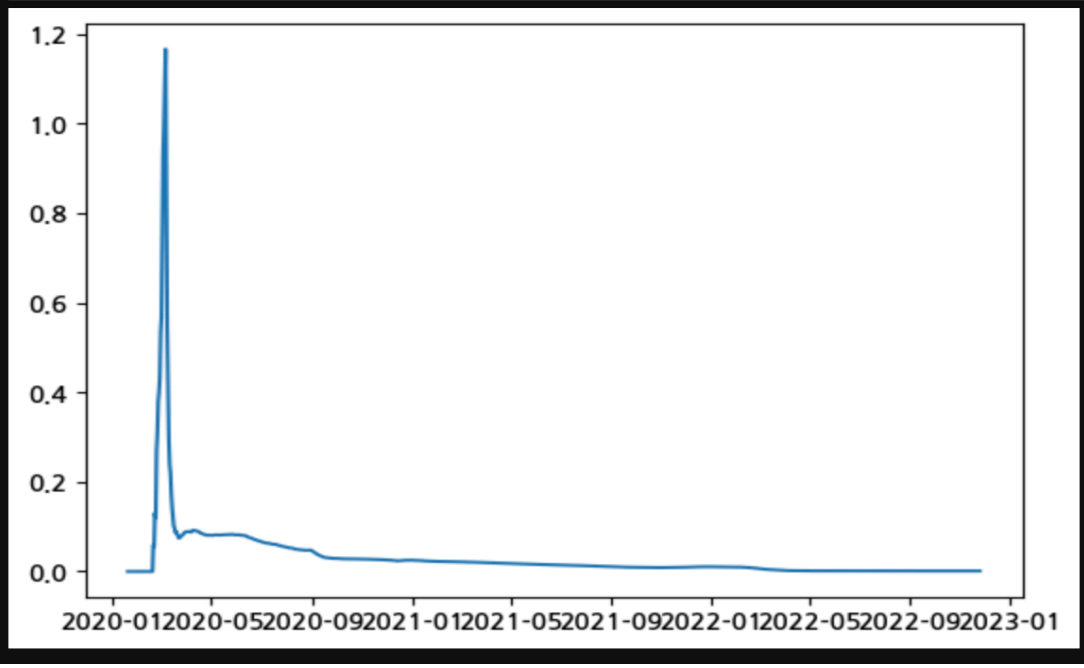
E.

```
plt.plot(df_covid3['일자'], df_covid_rmOutlier['치명률'])
plt.show()
```



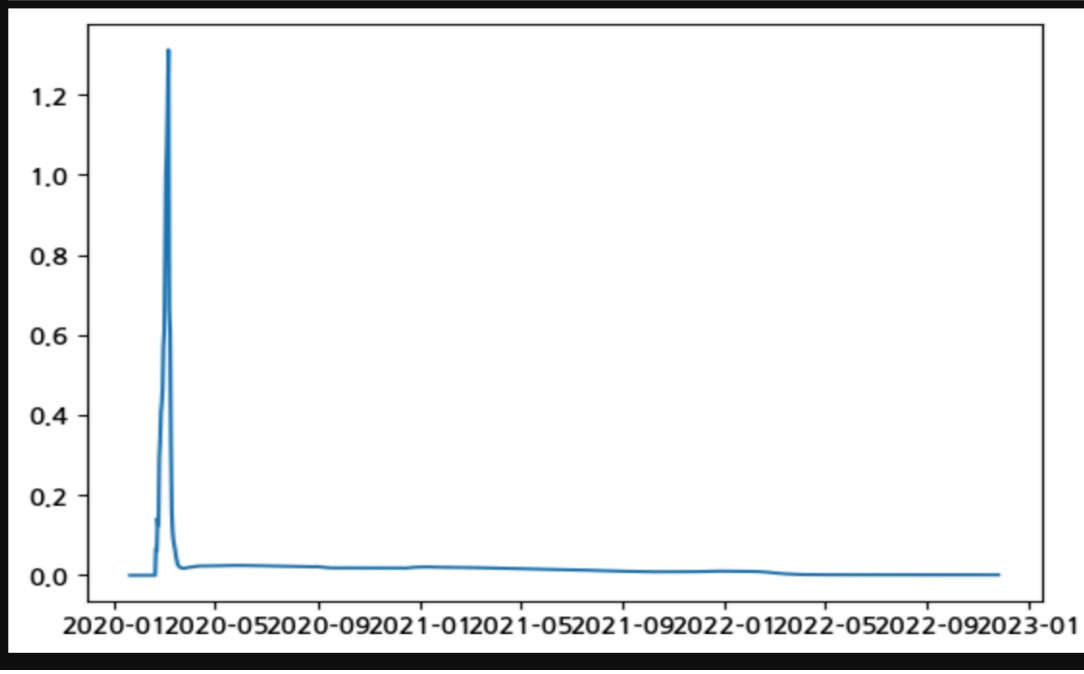
F.

```
[69]: plt.plot(df_covid2['일자'], df_covid['치명률'])
      plt.show()
```



G.

```
[70]: plt.plot(df_covid2['일자'], df_temp['치명률'])
      plt.show()
```



H. 서로 같은 분포인지 알아보자

I.

```
print("[1. 2차 실험 치명률 데이터 분석]")
t_stat, pval = stats.kstest(df_covid["치명률"], df_temp['치명률'])

print("이상치 제거 없음과 KS test : {}, p-value : {}".format(t_stat, pval))

t_stat, pval = stats.kstest(df_covid_rmOutlier["치명률"], df_covid['치명률'])

print("신천지 데이터 제거와 KS test : {}, p-value : {}".format(t_stat, pval))
```

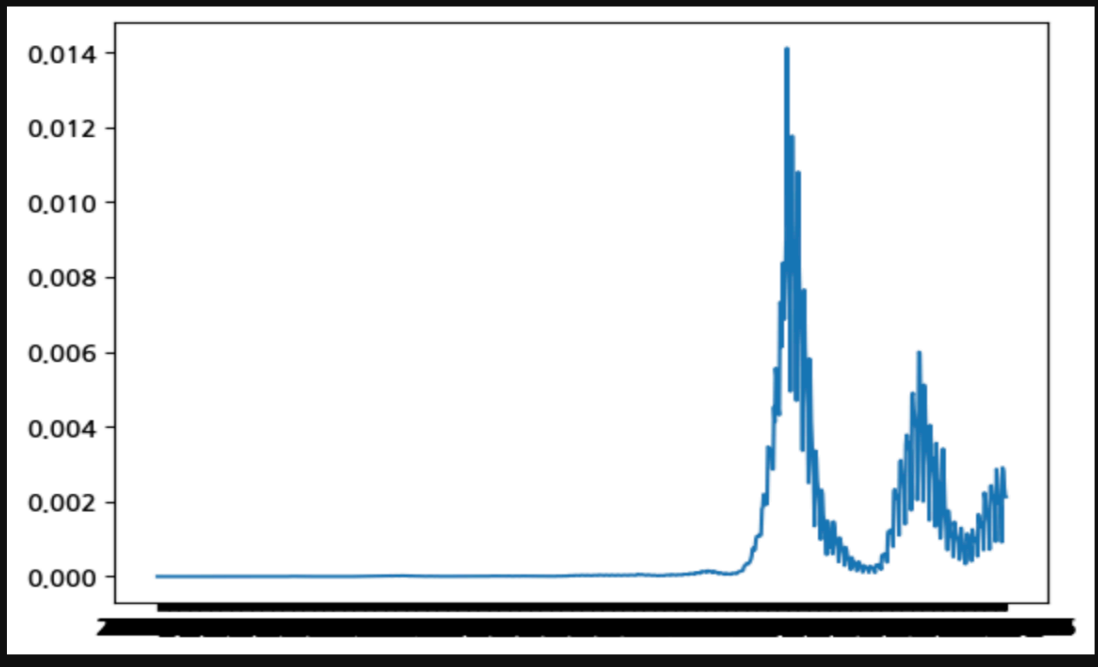
```
[1. 2차 실험 치명률 데이터 분석]
이상치 제거 없음과 KS test : 0.2696737044145873, p-value : 9.916277545251918e-34
신천지 데이터 제거와 KS test : 0.029750479846449136, p-value : 0.7451170727582392
```

J. 치명률은 이상치를 제거하지 않은 데이터와 비교했을 때, p-value 가 유의수준 0.05 보다 작기 때문에 다른 분포를 가지고 있음을 알 수 있다.

K. 감염률도 살펴보았다. (위에서부터 신천지 제거, 이상치 제거, 이상치 제거 없는 그래프)

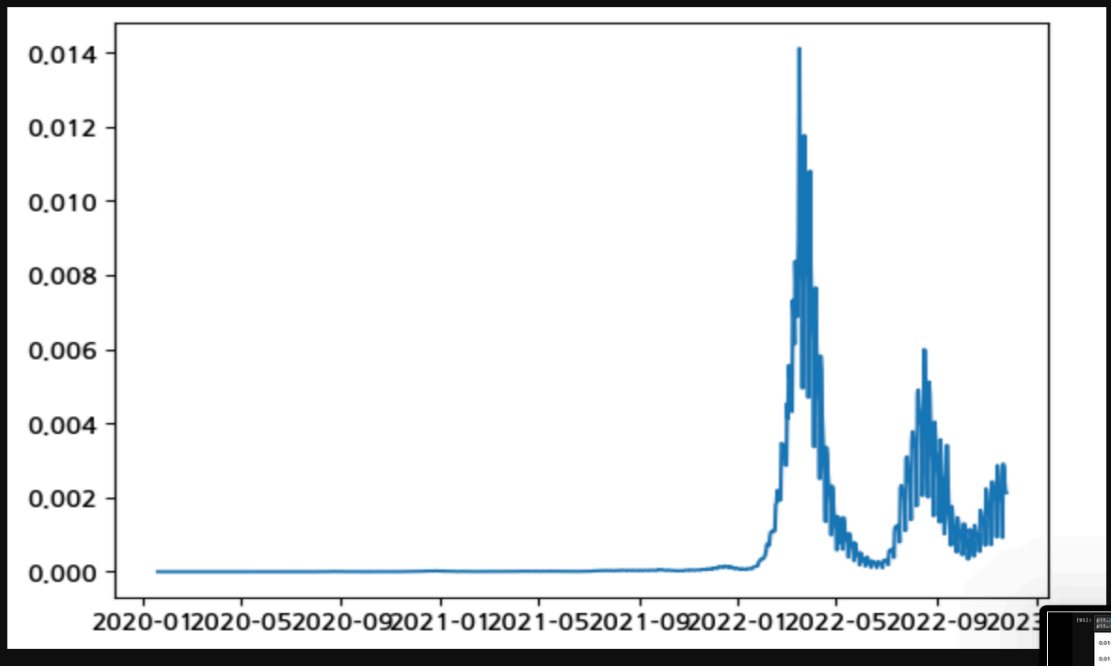
L.

```
[91]: plt.plot(df_covid3['일자'], df_covid_rmOutlier['감염률'])
      plt.show()
```



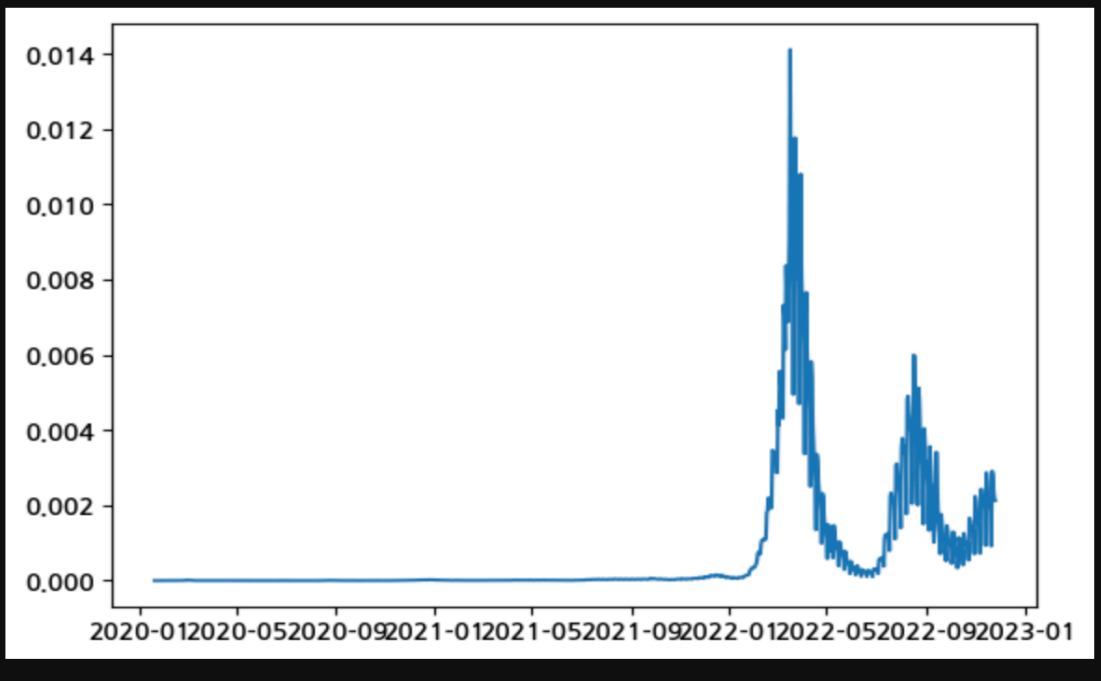
M.

```
[71]: plt.plot(df_covid2['일자'], df_covid['감염률'])
      plt.show()
```



N.

```
[72]: plt.plot(df_covid2['일자'], df_temp['감염률'])
      plt.show()
```



O. 위의 plot 을 살펴보면 별로 큰 차이를 느끼지 못했지만, 명확히 수치화 된 데이터가 없기 때문에 실험을 계속 진행하였다.

P.

- 분포가 서로 같은지 비교

```
print("[1. 2차 실험 감염률 데이터 분석]")
t_stat, pval = stats.kstest(df_covid["감염률"], df_temp['감염률'])

print("이상치 제거 없음과 KS test : {}, p-value : {}".format(t_stat, pval))

t_stat, pval = stats.kstest(df_covid_rmOutlier["감염률"], df_covid['감염률'])

print("신천지 데이터 제거와 KS test : {}, p-value : {}".format(t_stat, pval))
```

```
[1. 2차 실험 감염률 데이터 분석]
이상치 제거 없음과 KS test : 0.0345489443378119, p-value : 0.5630291856169715
신천지 데이터 제거와 KS test : 0.04309556982374777, p-value : 0.29228665913646174
```

Q. 감염률은 두 비교 모두 p-value 가 유의수준 0.05 보다 충분히 크다. 따라서 모두 같은 분포를 따른다.

R.

```
print("[1. 2차 실험 치명률 데이터 분석]")
t_stat, pval = stats.kstest(df_covid["치명률"], df_temp['치명률'])

print("이상치 제거 없음과 KS test : {}, p-value : {}".format(t_stat, pval))

t_stat, pval = stats.kstest(df_covid_rmOutlier["치명률"], df_covid['치명률'])

print("신천지 데이터 제거와 KS test : {}, p-value : {}".format(t_stat, pval))
```

```
[1. 2차 실험 치명률 데이터 분석]
이상치 제거 없음과 KS test : 0.2696737044145873, p-value : 9.916277545251918e-34
신천지 데이터 제거와 KS test : 0.029750479846449136, p-value : 0.7451170727582392
```

S. 이상치를 제거하지 않은 데이터와 이상치를 모두 제거한 데이터를 비교했을 때, p-value 가 유의수준 0.05 보다 작으므로 서로 같은 분포가 아니다.

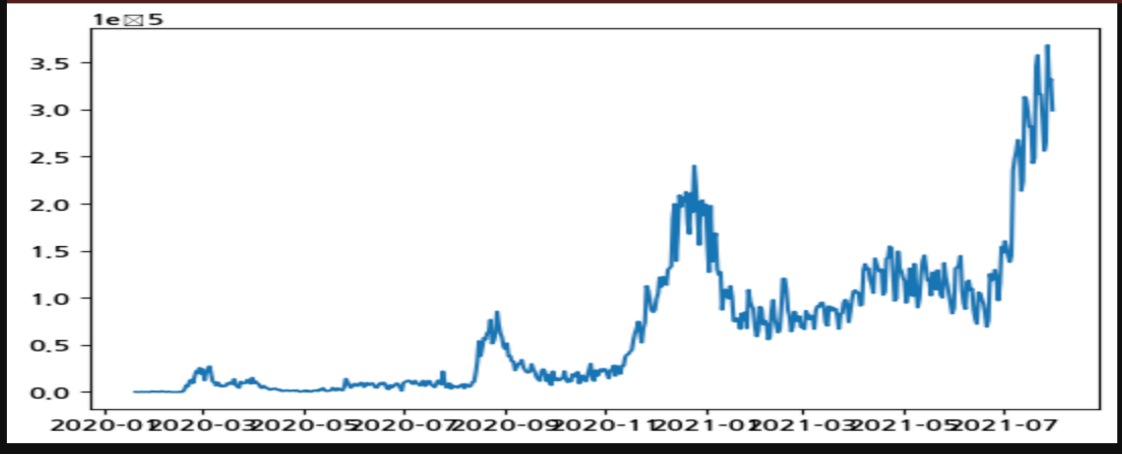
T. 따라서 이상치를 제거한 실험을 다시 진행한다. 감염률, 치명률 plot 이다. 순서대로 etc, delta, omi 변이 순이다.

U.

```
plt.plot(df_covid_etc['감염률'])
plt.show()
```

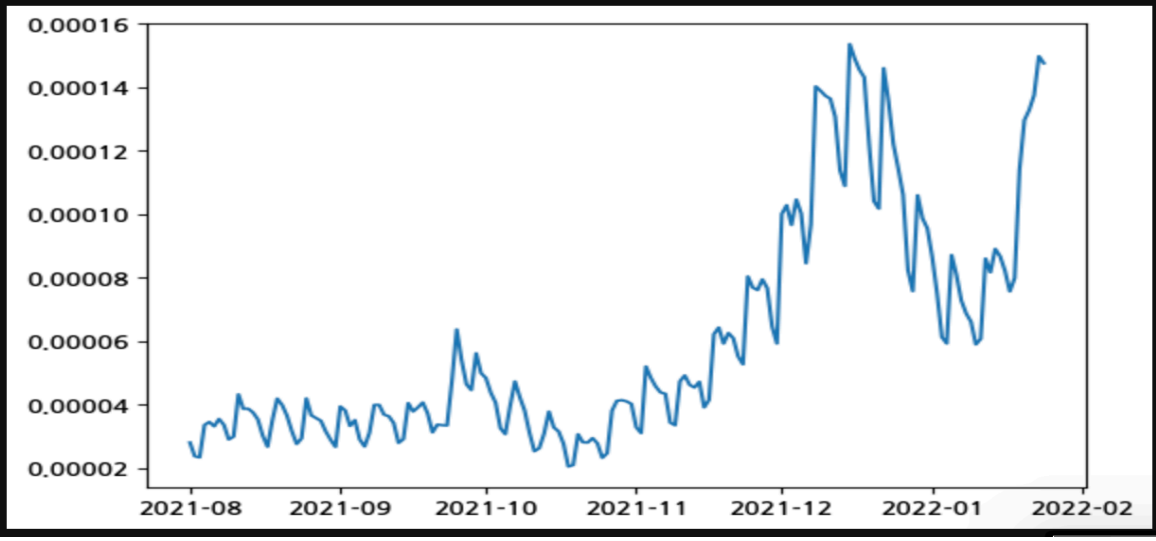
```
/opt/anaconda3/envs/SDA/lib/python3.9/site-packages/IPython/core/pylabtoolserWarning:

Glyph 8722 (\N{MINUS SIGN}) missing from current font.
```



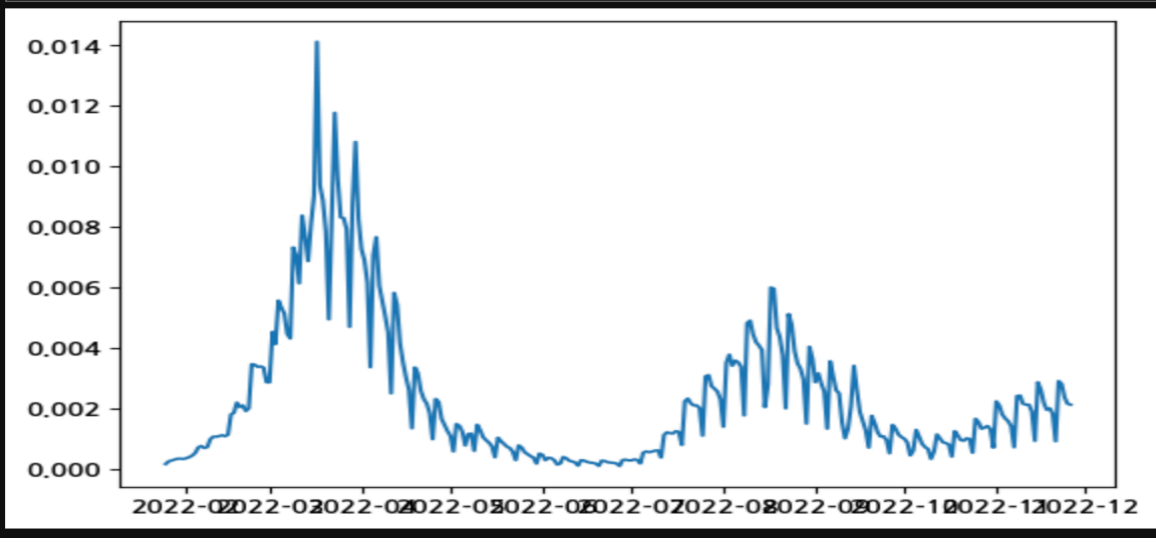
V.

```
]: plt.plot(df_covid_delta['감염률'])
   plt.show()
```



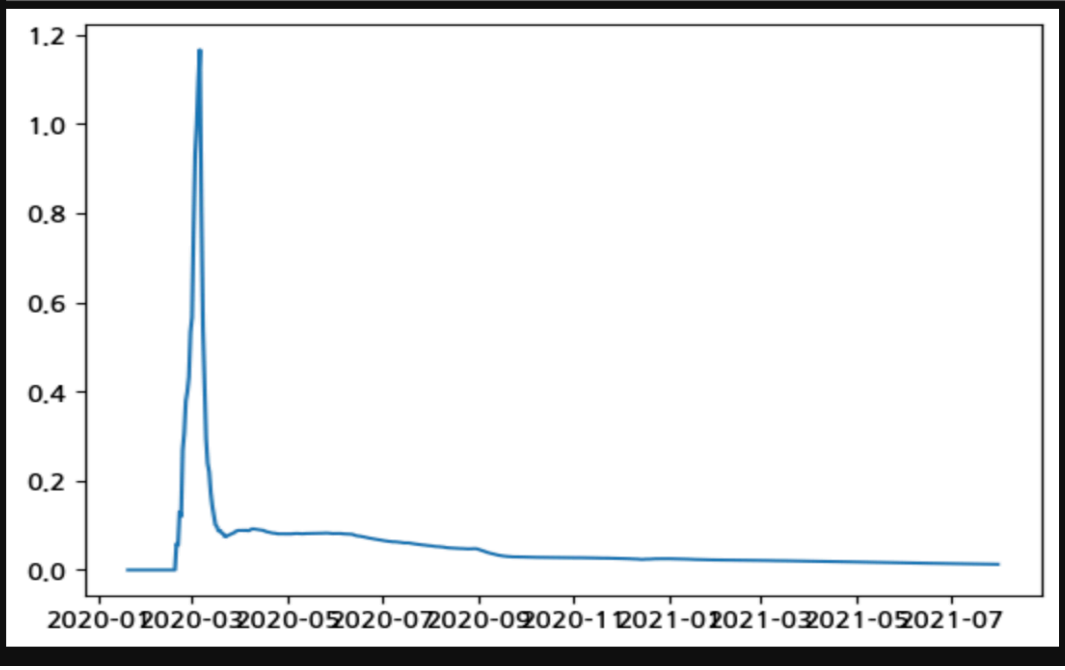
W.

```
]: plt.plot(df_covid_omi['감염률'])
   plt.show()
```



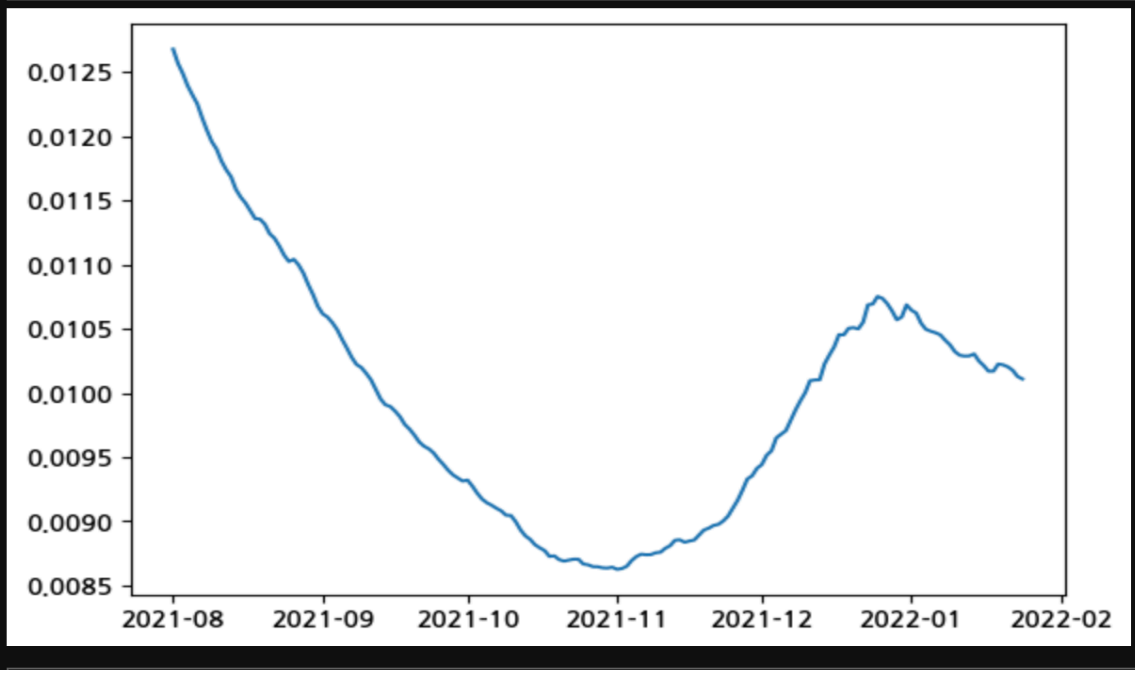
X.

```
[97]: plt.plot(df_covid_etc['치명률'])
      plt.show()
```



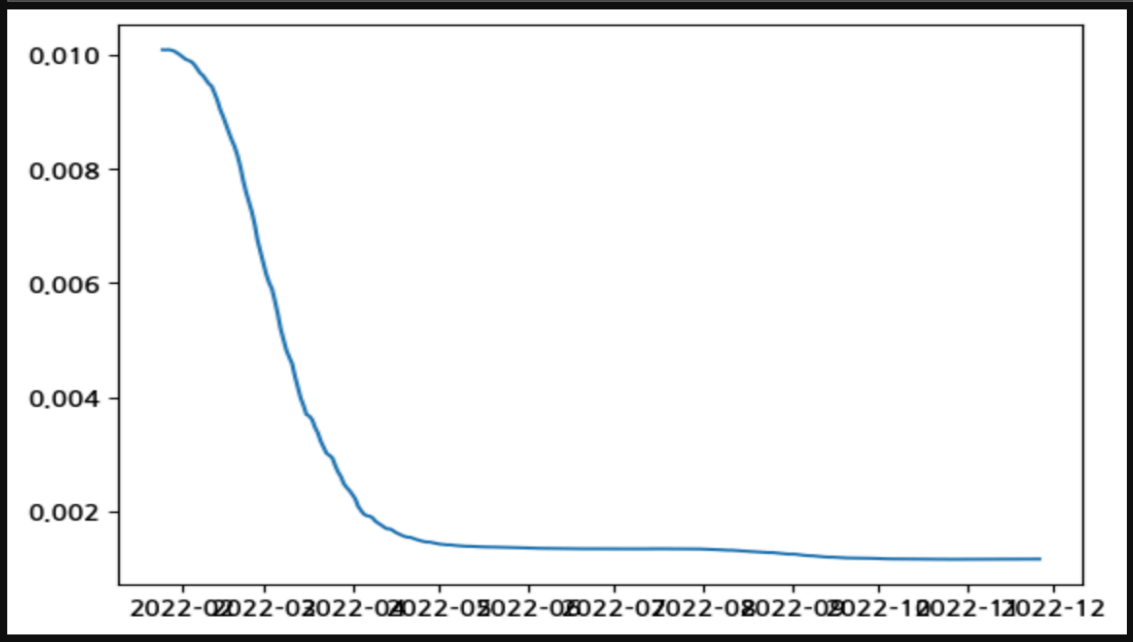
Y.

```
8]: plt.plot(df_covid_delta['치명률'])
    plt.show()
```



Z.

```
9]: plt.plot(df_covid_omi['치명률'])
    plt.show()
```



AA. .

```
### - 2차 실험 진행 : 이상치 제거 데이터로 진행
```

```
]: #정규성 검증
print("[1. 2차실험 감염률 데이터 분석]")
print("[감염률 정규성 검증]")
test_stat, p_val = stats.shapiro(df_covid_etc["감염률"])
print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.shapiro(df_covid_delta["감염률"])
print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
test_stat, p_val = stats.shapiro(df_covid_omi["감염률"])
print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
print()
print("[감염률 등분산 검증]")

test_stat, p_val = stats.fligner(df_covid_etc["감염률"],df_covid_delta["감염률"],df_covid_
print("Fligner Test-statistics : {}, p-value : {}".format(test_stat, p_val))
print()
#정규성을 만족하지 않으므로 Kruskal-Wallis Test 실행
print("[감염률 Kruskal-Wallis Test]")
test_stat, p_val = stats.kruskal(df_covid_etc["감염률"], df_covid_delta["감염률"], df_covi
print("Kruskal-Wallis Test-statistics : {}, p-value : {}".format(test_stat, p_val))
print()
```

```
[1. 2차실험 감염률 데이터 분석]
[감염률 정규성 검증]
Shapiro Test-statistics : 0.8239768147468567, p-value : 2.6924469561135414e-24
Shapiro Test-statistics : 0.8355443477630615, p-value : 7.483093463454826e-13
Shapiro Test-statistics : 0.8051600456237793, p-value : 7.111323941894546e-19

[감염률 등분산 검증]
Fligner Test-statistics : 738.2296005975451, p-value : 4.9599699054499084e-161

[감염률 Kruskal-Wallis Test]
Kruskal-Wallis Test-statistics : 843.910687056595, p-value : 5.586279949873831e-184
```

```
]: #2 치명률 데이터 분석
   print("[2. 2차 실험 치명률 데이터 분석]")
   #타입변환
   df_covid_etc = df_covid_etc.astype({'치명률' : 'float'})
   df_covid_delta = df_covid_delta.astype({'치명률' : 'float'})
   df_covid_omi = df_covid_omi.astype({'치명률' : 'float'})

   #정규성 검증
   print("[치명률 정규성 검증]")
   test_stat, p_val = stats.shapiro(df_covid_etc["치명률"])
   print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
   test_stat, p_val = stats.shapiro(df_covid_delta["치명률"])
   print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
   test_stat, p_val = stats.shapiro(df_covid_omi["치명률"])
   print("Shapiro Test-statistics : {}, p-value : {}".format(test_stat, p_val))
   print()

   #등분산 검증
   print("[치명률 등분산 검증]")
   # 정규분포 아닐 시 bartlett사용 불가 -> non-parametric method인 fligner test 사용(by 60211642)
   test_stat, p_val = stats.fligner(df_covid_etc["치명률"],df_covid_delta["치명률"],df_covid_
   print("Fligner Test-statistics : {}, p-value : {}".format(test_stat, p_val))

   #정규성을 만족하지 않으므로 Kruskal-Wallis Test 실행
   print("[치명률 Kruskal-Wallis Test]")
   test_stat, p_val = stats.kruskal(df_covid_etc["치명률"], df_covid_delta["치명률"], df_covi
   print("Kruskal-Wallis Test-statistics : {}, p-value : {}".format(test_stat, p_val))
   print()
```

```
[2. 2차 실험 치명률 데이터 분석]
[치명률 정규성 검증]
Shapiro Test-statistics : 0.3306838274002075, p-value : 7.826952572486266e-41
Shapiro Test-statistics : 0.9401803612709045, p-value : 9.566617791278986e-07
Shapiro Test-statistics : 0.5527125000953674, p-value : 3.0214849858232097e-27

[치명률 등분산 검증]
Fligner Test-statistics : 499.17347377856834, p-value : 4.0351289015256053e-109
[치명률 Kruskal-Wallis Test]
Kruskal-Wallis Test-statistics : 677.2507188012573, p-value : 8.647189566037621e-148
```

BB.

CC. 실험의 결론은 위의 첫 번째 실험의 결론과 같다.

3.5. 데이터 참조

A. 데이콘의 포스트 코로나 데이터 시각화 경진대회에서 집단 감염자 제외한 코드를 copy 해 진행했다.

B. https://dacon.io/competitions/official/235618/codeshare/1501?page=1&dtype=random

C. 추가 데이터는 https://www.kaggle.com/datasets/kimjihoo/coronavirusdataset 에서 가져왔다.

3.6. 회고

A. 교수님이 쓰레기 데이터로 분석을 시작하면 쓰레기가 나올 수 없다 라는 말을 몸소 체험한 분석이었다.

B. 치명률 정의에 너무 매몰되어서 실험은 진행한 것이 실패를 하는 중요한 요인이었다.

C. 치명률 정의에 너무 시간을 낭비해 결국은 시간에 쫓겨 마감하고 이상한 결론이 나올 수 밖에 없었다.

3.7. 깃허브 링크

https://github.com/Jieun-Song/2022-2-SDATeamProject

감사합니다.