

Covid-19 Pandemisi

Covid-19 Nedir?

- İlk olarak Çin'in Wuhan bölgesinde, 2019 yılı Aralık ayının başında görülüp, Dünya Sağlık Örgütü (DSÖ/WHO) tarafından virüsün resmi adı SARS-CoV-2 (Şiddetli Akut Solunum Sendromu-Koronavirus-2) olarak belirlenmiştir.
- Dünya Sağlık Örgütü virüsün neden olduğu hastalığı tanımlamak için Covid-19 terimini kullanmaktadır.
- 30 Ocak 2020'de Covid-19, Dünya Sağlık Örgütü tarafından küresel bir sağlık acil durumu ilan edilmiştir.
- 11 Mart 2020 tarihinde ise virüs pandemi, yani küresel salgın hastalık olarak ilan edilmiştir.

Kaynak: <https://www.acibadem.edu.tr/covid-19/hakkinda>

▸ Veri Dosyasını Alma

↳ 5 cells hidden

▼ Veriyi Görselleştirme

```
#Gerrekli kütüphanelerin eklenmesi
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import math
import folium
from folium.plugins import TimestampedGeoJson
import matplotlib.ticker as ticker
import matplotlib.animation as animation
import matplotlib.colors as mc
import colorsys
from IPython.display import HTML
import random
import datetime
```

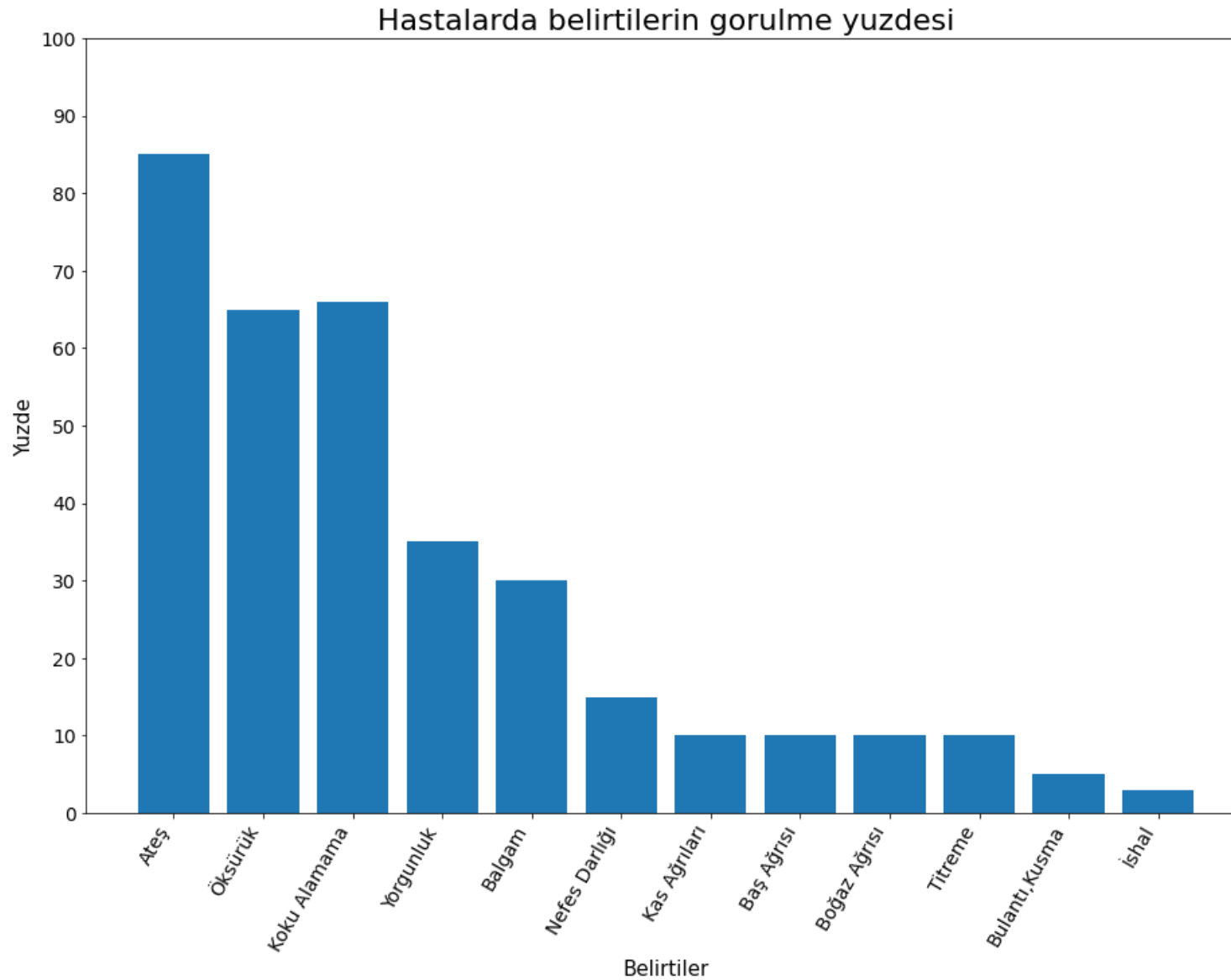
Covid-19 Belirtileri Nelerdir?

- Yeni Koronavirüs Hastalığı genel olarak ateş, kuru öksürük ve ilerleyen durumlarda solunum güçlüğü ile seyreden bir hastalıktır.
- Bunlara ilave olarak boğaz ağrısı, baş ağrısı bulantı-kusma, ishal, burun akıntısı, kas-eklem ağrısı, iştahsızlık gibi farklı yakınmalar da görülebilmektedir
- Hastalık bazı genç ve sağlıklı kişilerde hiçbir yakınmaya yol açmazken, bazı kişilerde daha hafif bir klinik tablo ile seyredebilmekte, ancak özellikle, diyabet (şeker hastalığı), tansiyon yüksekliği, akciğer hastalığı, bağışıklık sistemi hastalığı bulunan kişilerde daha ağır ve hastane koşullarında tedavi gerektiren pnömoni (zatürre), akut solunum yetmezliği, böbrek yetmezliği gibi ciddi klinik tabloların ortaya çıkmasına neden olabilmekte, bazı vakalarda ise ölümle sonuçlanabilmektedir.

Kaynak: <https://www.acibadem.edu.tr/covid-19/hakkinda>, #<https://www.sozcu.com.tr/2020/saglik/corona-virusunun-yetiskinlerde-ve-cocuklardaki-belirtileri-neler-5765732/>

```
fig = plt.figure()
ax = fig.add_axes([1,1,2,2])
belirtiler = ['Ateş', 'Öksürük', 'Koku Alamama', 'Yorgunluk', 'Balgam','Nefes Darlığı','Kas Ağrıları','Baş Ağrısı','Boğaz Ağrısı', 'Titreme','Bulantı,Kusma','İshal']
yuzdeler = [85,65,66,35,30,15,10,10,10,10,5,3]
ax.bar(belirtiler,yuzdeler)
plt.gca().set_xticklabels(belirtiler, rotation=60, horizontalalignment= 'right')
plt.title("Hastalarda belirtilerin gorulme yuzdesi", fontsize=22)
plt.yticks(np.arange(0, 110, step=10))
plt.ylabel('Yuzde', fontsize=15)
plt.xlabel('Belirtiler', fontsize=15)
plt.show()
```





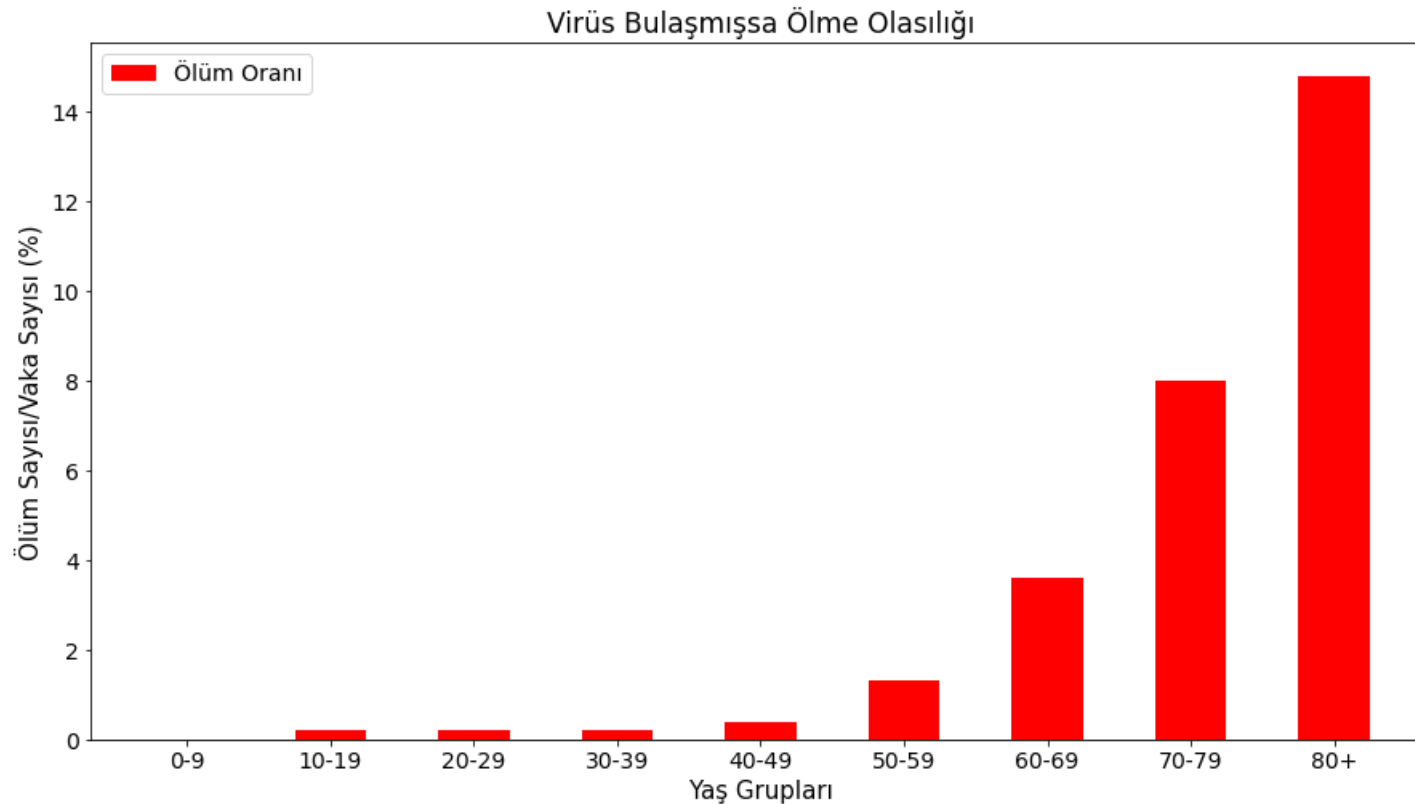
Covid-19 Hangi Gruplarda Daha Ağır Seyrediyor?

- Covid-19 özellikle 65 yaş üstü ve kronik hastalığı olan kişileri etkilediği biliniyor.

- Kanser hastalarını, hamileleri, bağışıklık sistemi zayıf olanları, yüksek tansiyon hastalarını, şeker hastalarını, akciğer hastalıkları ve KOAH gibi ek hastalıkları olan insanları daha fazla etkilediği gözlemlenmiş bulunmaktadır.
- Covid-19 belirtileri çocuklarda da ender olarak görülse de, hastalığı hafif semptomlarla atlattıkları gözlemlenmektedir.
- Türkiye'deki nüfusun genç olması ve obezite oranının Amerika gibi ülkelere göre düşük olması bizi bu konuda şanslı kılıyor.

```
data_by_age =[0,0.2,0.2,0.2,0.4,1.3,3.6,8.0,14.8]
plt.figure(figsize=(15,8))
plt.bar(["0-9" ,"10-19","20-29","30-39","40-49","50-59","60-69","70-79","80+"],
        data_by_age,width=0.5,color="red",label="Ölüm Oranı")
plt.ylabel("Ölüm Sayısı/Vaka Sayısı (%)", fontsize = 15)
plt.xlabel("Yaş Grupları", fontsize = 15)
plt.legend(fontsize=14)
plt.title("Virüs Bulaşmışsa Ölme Olasılığı",fontsize = 17)
#https://www.worldometers.info/coronavirus/coronavirus-age-sex-demographics/
```

Text(0.5, 1.0, 'Virüs Bulaşmışsa Ölme Olasılığı')



Türkiye Nüfusunun Yaş Dağılımı

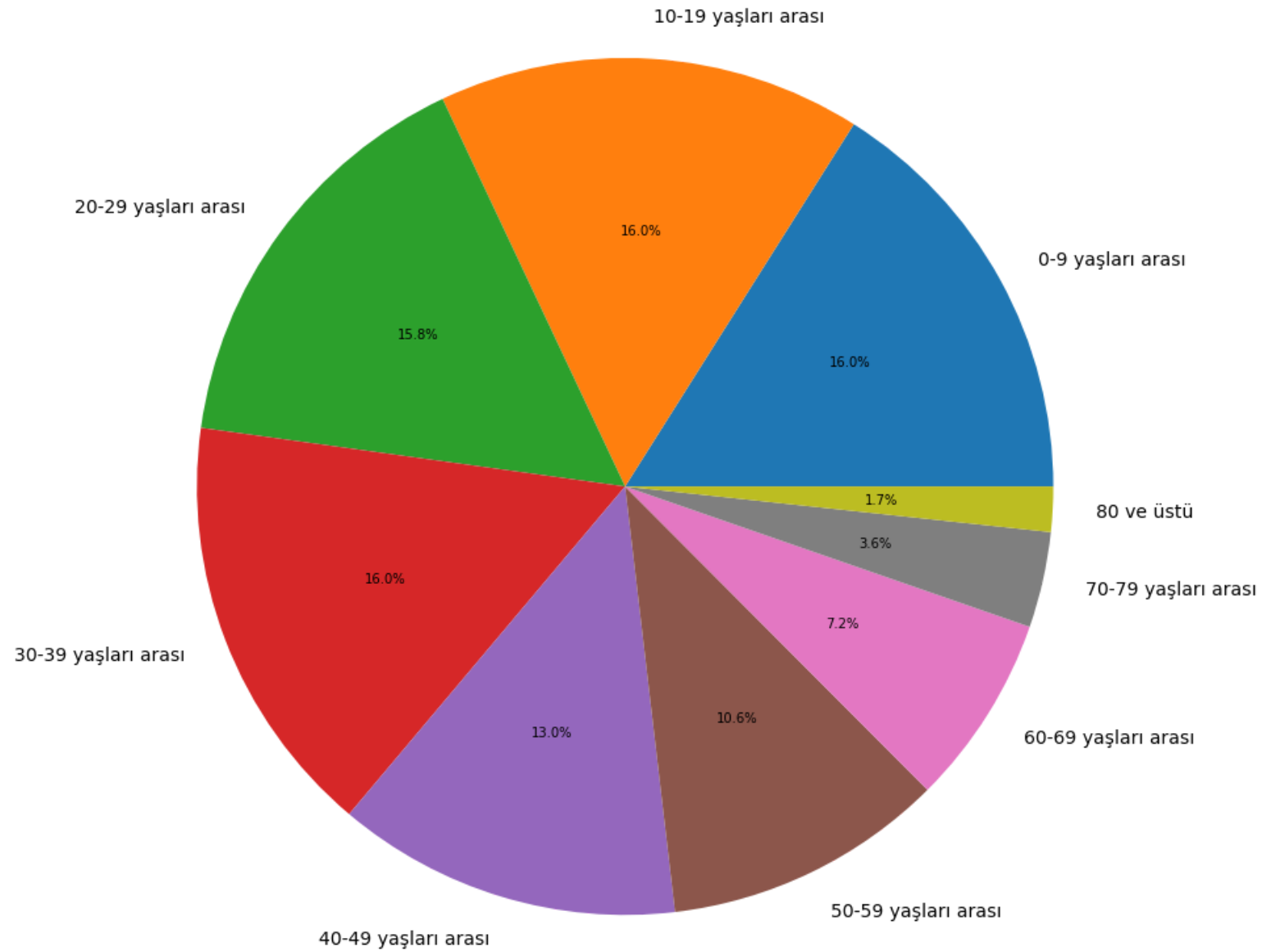
- Türkiye gelişmekte olan bir ülke olduğu için nüfusu gençtir.
- Bu da Türkiye'deki ölüm oranının düşük olmasında bir etkidir.

```
plt.figure(figsize=(15,15))
plt.pie([12796739,12752362,12611764,12804744,10382831,8471980,5755485,2882542,1356424],
        labels=["0-9 yaşları arası","10-19 yaşları arası","20-29 yaşları arası","30-39 yaşları arası",
                "40-49 yaşları arası","50-59 yaşları arası","60-69 yaşları arası","70-79 yaşları arası","80 ve üstü"],autopct='%1.1f%%')
plt.rc(['xtick','ytick'], labelsize=15)
plt.title("Türkiye'deki yaş gruplarının nüfusu", fontsize=15)
#https://tr.wikipedia.org/wiki/T%C3%BCrkiye\_demografisi
```



```
Text(0.5, 1.0, "Türkiye'deki yaş gruplarının nüfusu")
```

Türkiye'deki yaş gruplarının nüfusu



COVID-19 nasıl bulaşır?

- Virüs, kişiden kişiye burun ve / veya ağızdan küçük enfekte damlacıklar yoluyla bulaşabilir, COVID-19 olan bir kişi öksürdüğünde veya nefes verdiğinde yayılabilir.
- Bu damlacıklar kişinin etrafındaki nesnelere ve yüzeylere inebilir.
- Diğer kişiler daha sonra bu nesnelere ve / veya yüzeylere dokunarak enfeksiyonu kapar, sonra gözlerine ve / veya burnuna ve / veya ağızına dokunur.
- İnsanlar ayrıca, COVID-19'lu damlacıkları öksüren veya nefes veren bir kişiden damlacıklarla nefes aldıklarında da bulaşabilirler.
- Bu, hasta olan bir kişiden en az 1 metre uzaklıktaki sosyal mesafenin önemini açıklar.

Kaynak: <https://www.gau.edu.tr/duyuru/18754/covid-19>

```
plt.figure(figsize=(20,10))

plt.plot(datatoplam.groupby('Date').sum().index,
         datatoplam.groupby('Date').sum()['Confirmed'], label = "Dünyadaki Toplam Vaka Sayısı")
plt.plot(datatoplam.groupby('Date').sum().index,
         datatoplam.groupby('Date').sum()['Deaths'], label = "Dünyadaki Toplam Ölüm Sayısı")

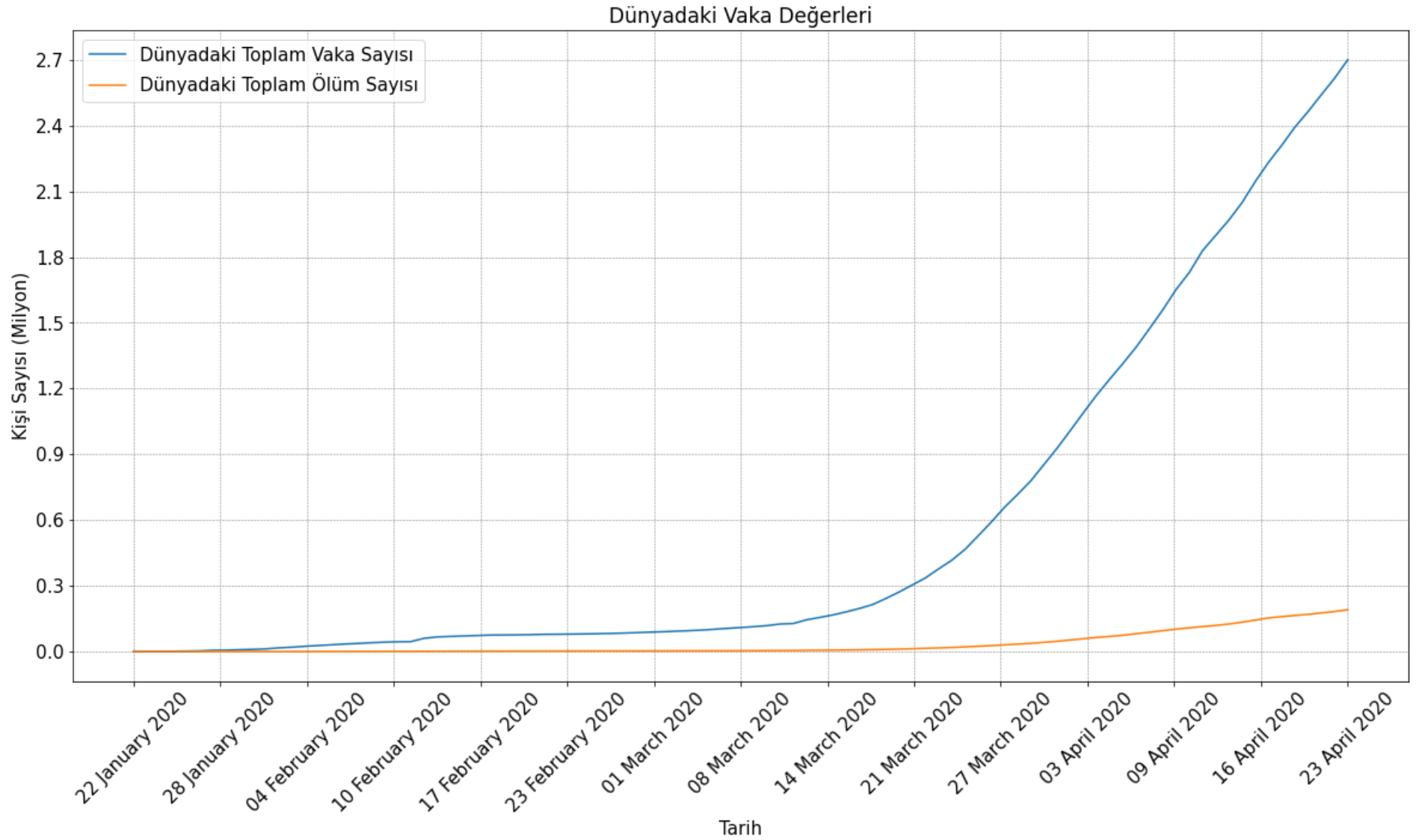
plt.rc(['xtick','ytick'], labels=14)
plt.yticks(np.linspace(0, max(datatoplam.groupby('Date').sum()['Confirmed']), 10),
          np.round(np.linspace(0, max(datatoplam.groupby('Date').sum()['Confirmed']), 10)/1000000,2))
plt.xticks(np.linspace(data['Date'].min().value,
                      data['Date'].max().value, 15).astype('datetime64[ns]'),
          pd.to_datetime(np.linspace(data['Date'].min().value,
                      data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Kişi Sayısı (Milyon)', fontsize=15)
plt.title('Dünyadaki Vaka Değerleri', fontsize=17)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f47a8dc18>



COVID-19 Neden Hızlı Yayılıyor?

- Yeni tip koronavirüsün en belirgin özelliklerinden biri hızlı yayılması. Genetik bilimci Prof. Dr. Selim Badur'a göre yeni tip koranavirüs yeni bir antijenik (vücuda girdiğinde antikor oluşmasına yol açan virüs, bakteri, parazit gibi protein yapısında bir madde) tip olması nedeni ile insanlar

arasında kendisine karşı gelişmiş bir immünite (bağışıklık) henüz söz konusu değil.

- Diğer bir tanımlama ile, ilk kez karşılaşılan bu yeni etkene karşı hiçbir insanda daha önceden var olan bir bağışıklık yok; bu nedenle virüs çok süratle yayılıyor.
- Koronavirüsün 2 haftalık kuluçka süresi, hastalık belirtilerini gizliyor ve hastalığın kuluçka süresinde daha zor tespit edilebilmesine yol açıyor.
- Koronavirüs'ün karşılaştığımız virüs çeşidi ile daha önce tanışmamız oluşumuz virüsün karakteristik özelliklerini bilmemeyişimize neden oldu. Hangi yollarla bulaştığı, virüsün etkileri ve etkilediği organlar gün geçtikçe keşfedilmeye başlandı. İlk başlarda DSÖ dahil birçok kurum yanılgılara dahi düştü.
- Amerika, İtalya gibi ülkeler, virüsü ilk başlarda ciddiye almadılar ve günlük hayatlarına devam ettiler. Festivaller ve kalabalık etkinlikleri düzenleyen İtalya, Sınırlarını kapatmakta geç kalan Amerika hala ağır bedeller ödüyor.
- Çin Halk Cumhuriyeti'nin verileri geç açıklaması da Dünya'nın olayı geç fark etmesine neden oldu

Yukarıda saydığımız sebepler COVID-19'un hızlı yayılmasında öngördüğümüz büyük etkenler

```
total_incidents = data_map[data_map['Confirmed'] > 0]['Confirmed'].sum()
```

```
def geojsons(df):
    features = []

    for _, row in df.iterrows():
        feature = {
            'type' : 'Feature',
            'geometry' : {
                'type' : 'Point',
                'coordinates' : [row['Long'],row['Lat']]
            },
            'properties' : {
                'time' : pd.to_datetime(row['Date'], format = '%Y-%m-%d').__str__(),
                'style' : {'color' : ''},
                'icon' : 'circle',
                'iconstyle': {
                    'fillColor' : 'red',
                    'fillOpacity' : 0.8,
                    'stroke' : 'true',
                    'radius' : math.pow(abs(row['Confirmed']), 0.22)
                }
            }
        }

        features.append(feature)
    return features

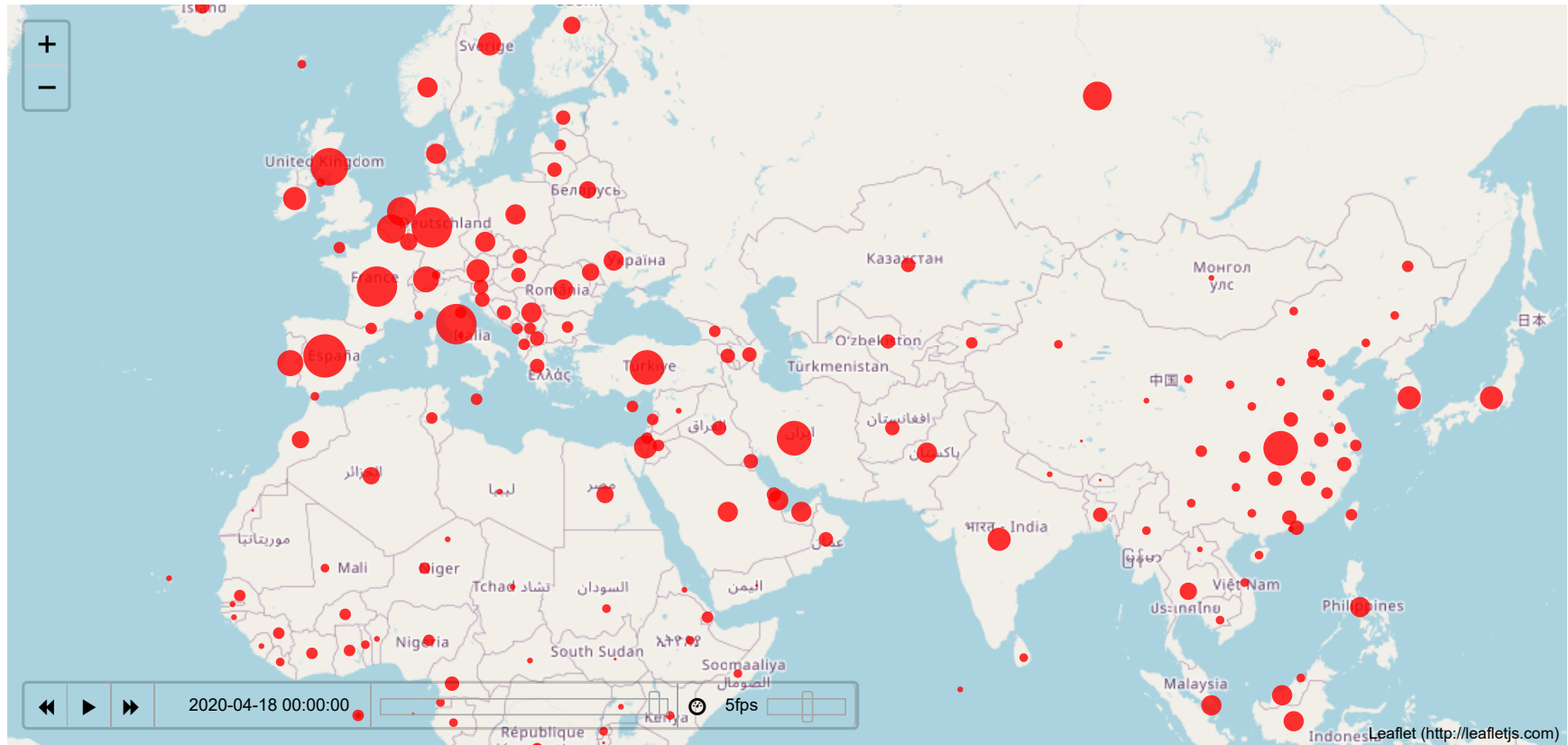
start_geojson = geojsons(data_map[data_map['Confirmed'] > 0])

m = folium.Map(width=1000, height=512, location = [50, 30], zoom_start = 3, tiles = 'OpenStreetMap')
```

```
m = Folium.Map(width=1000,height=512,location = [50,50],zoom_start = 5,tiles = 'OpenStreetMap',
```

```
TimestampedGeoJson(
    start_geojson,
    period = 'P1D',
    duration = 'PT1M',
    transition_time = 200,
    auto_play = True,
).add_to(m)
m
```

↪



Salgın Asya'da Başlamasına Rağmen Avrupa ve Amerika Asya'yı Nasıl Geçti?

- Yeni tip koronavirüsün açığa çıkardığı bir gerçek de sınırların küresel krizler ya da salgınlar karşısında anlamını yitirmesi. Nitekim dünya genelinde 100'den fazla ülkede ve bölgede bu salgın -farklı düzeylerde olsa da- baş göstermiş durumda.
- Covid-19'un bulaşma zincirindeki yüksek nüfuslu ülkelerin, özellikle İtalya, İspanya ve Amerika'nın önlem almakta gecikmeleri, alınan önlemlerin sınırlı kalması, sağlık alt yapısındaki eksiklikler (tanı merkezlerinin azlığı, genel taramanın yapılmaması vs), toplumsal alışkanlık farklılıklarının getirdiği salgını hafifseme gibi nedenler salgının yayılmasına neden oldu.
- Amerika gibi ülkelerde Covid-19 tedavisinin(ilk başlarda) ücretsiz olmaması insanları zora soktu.
- Virüs'ün ortalama 2 haftalık kuluçka süresinin olduğu biliniyor. Belki de bu 2 haftalık süre içerisinde hasta olan diğer kişilerin çoktan ülkelere dönmüş olabileceği de bu varsayımlarımız içinde.
- Aşağıdaki animasyonda birçok ülkedeki vaka sayısının Çin'deki vaka sayısının katlarına ulaştığı görünmektedir.

```
#Bar yarış grafiğinde kullanmak için veri dosyasının bazı kolonlarından yeni bir df
df = data[['Country/Region', 'Date', 'Confirmed', 'Deaths']]
```

```
#Tarih formatını, salgının .inci günü şekline çevirmek için kod bloğu
def kacinci_gun(a):
    return (a - df['Date'].min()).days+(data['Date'].min()-pd.Timestamp('2019-12-01')).days
```

```
print("Veri dosyası salgının başlangıcından {} gün sonra yazılmaya başlandı.".format((data['Date'].min()-pd.Timestamp('2019-12-01')).days))
df['Date'] = df['Date'].apply(kacinci_gun)
```

```
❏ Veri dosyası salgının başlangıcından 52 gün sonra yazılmaya başlandı.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy.
"""
```

```
#Veri dosyasındaki en fazla vakaya sahip olan 20 ülkeden bir df oluşturduk
current_time = df['Date'].max()
dff = (df[df['Date'].eq(current_time)]
       .sort_values(by='Confirmed', ascending=False)
       .head(20))
```

```
#Random renk almak ve rengi hex formatından rgba formatına çeviren fonksiyonlar
def random_renk():
    r = lambda: random.randint(0,255)
```

```

r = '%02X%02X%02X' % (r(),r(),r())
return r

def transform_color(color, amount = 0.5):
    try:
        c = mc.cnames[color]
    except:
        c = color
    c = colorsys.rgb_to_hls(*mc.to_rgb(c))
    return colorsys.hls_to_rgb(c[0], 1 - amount * (1 - c[1]), c[2])

#Bir sözlük oluşturup ülkelere random renk atadık
colors = dict()

for i in dff['Country/Region'].tolist():
    colors.update({i: random_renk()})

#Bar yarışı grafiğinin yenileme fonksiyonu
dfb = dff
def draw_barchart(time):
    dff = df[(df['Date'].eq(time)) & (df['Country/Region'].isin(dfb['Country/Region']))].sort_values(by='Confirmed', ascending=True).tail(10)
    ax.clear()
    ax.barh(dff['Country/Region'], dff['Confirmed'], color=[transform_color(colors[x], 1) + (0.825,) for x in dff['Country/Region']],
            , height = 0.8, edgecolor=[transform_color(colors[x], 0.9) for x in dff['Country/Region']], linewidth = '2')
    dx = dff['Confirmed'].max() / 300
    for i, (value, name) in enumerate(zip(dff['Confirmed'], dff['Country/Region'])):
        ax.text(value+dx, i+10/50, name, size=14, weight=600, ha = 'left', va = 'center')
        ax.text(value+dx, i-10/50, f'{value:,.0f}', size=14, ha = 'left', va = 'center')
    # ... polished styles
    ax.text(1, 0.4, str(time)+'. Gün', transform=ax.transAxes, color='#777777', size=46, ha='right', weight=800)
    ax.text(0, 1.015, 'Salgının Başlangıcından Günümüze En Çok Vakaya Sahip Ülkeler ', transform=ax.transAxes, size=35, weight=600, color='#444')
    ax.xaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
    ax.xaxis.set_ticks_position('top')
    ax.tick_params(axis = 'x', colors = '#fff', labelsiz = 28)
    ax.set_yticks([])
    ax.set_axisbelow(True)
    ax.margins(0, 0.01)
    #ax.grid(which = 'major', axis = 'x', linestyle = '-')

    plt.locator_params(axis = 'x', nbins = 4)
    plt.box(False)
    plt.subplots_adjust(left = 0.075, right = 0.75, top = 0.825, bottom = 0.05, wspace = 0.2, hspace = 0.2)

#Bar race grafiğini oluşturduk ve video olarak çıktısını aldık
fig, ax = plt.subplots(figsize=(20, 8))
animator = animation.FuncAnimation(fig, draw_barchart, frames=range(df['Date'].min(), df['Date'].max()+1))
HTML(animator.to_html5_video())

```

```
print(animatör.save_video())  
#animatör.save("Ülkelerde Vaka Sayıları.mp4", fps = 5, bitrate = 1800)
```



0:18 / 0:18

Salgının Başlangıcından Günümüze En Çok Vakaya Sahip Ülkeler



52. Gün

```
Germany
0
India
0
Iran
0
Ireland
0
```

ABD’de Koronavirüs Vaka Sayısı Neden Yüksek?

• Harvard’dan Prof. Dr. Gökhan Hotamışlıgil bu konuyu şu yorumları yapıyor:

Amerika’nın bu duruma gelmesine 2 kısımda bakmak lazım. Salgının sıra dışı bir virüsle yayıldığı fark edilince ki buna ocak ayı ortaları diyelim, o süreç içerisinde ABD yönetiminin bunu birinci sebep olarak ciddiye almaması, ABD’ye hafif şekilde uğrayacağı yönünde bir söylemle işin üzerine ciddiyetle eğilmemesi...

• İkinci süreç, geçirdiğimiz 2-3 sene içerisinde ABD’nin çok önemli, bilime dayalı, hastalıklar bakımından bütün tedbirleri alan kuruluşunun yetkinliklerinin azaltılmış olması... Yönetimin bilim insanlarının tavsiyelerinden uzak bir davranış içine girmiş olması. Yetkinlik ve yönetim problemleri nedeniyle önlemlerin gecikmesi önemli bir zaman kaybına yol açtı. 60-70 gün gibi bir süreç önlem alınmadan geçirildi.

• Amerika başkanı Donald Trump, ekonomik ve siyasi çıkarlarını bu salgının önüne koyarak büyük bir hata yaptı böylelikle tedbirleri çok geç aldı. Salgının yayılımı kontrolden çıktı ve New York gibi eyaletlerde sağlık sistemi gelen büyük dalga karşısında felce uğradı.

Peki buradan çıkarmamız gereken sonuç nedir?

1. Bu durum bilim kuruluşlarının önemini ortaya koydu.
2. Siyasi ve Ekonomik çıkarların halk sağlığının önünde tutulmaması gerektiğini gösterdi.

```
dts = data[(data['Date']==data['Date'].max()) & (data['Confirmed'] > 1000)][['Country/Region', 'Deaths', 'Confirmed']].set_index('Country/Region').sort_index()
ks = 0.4
```

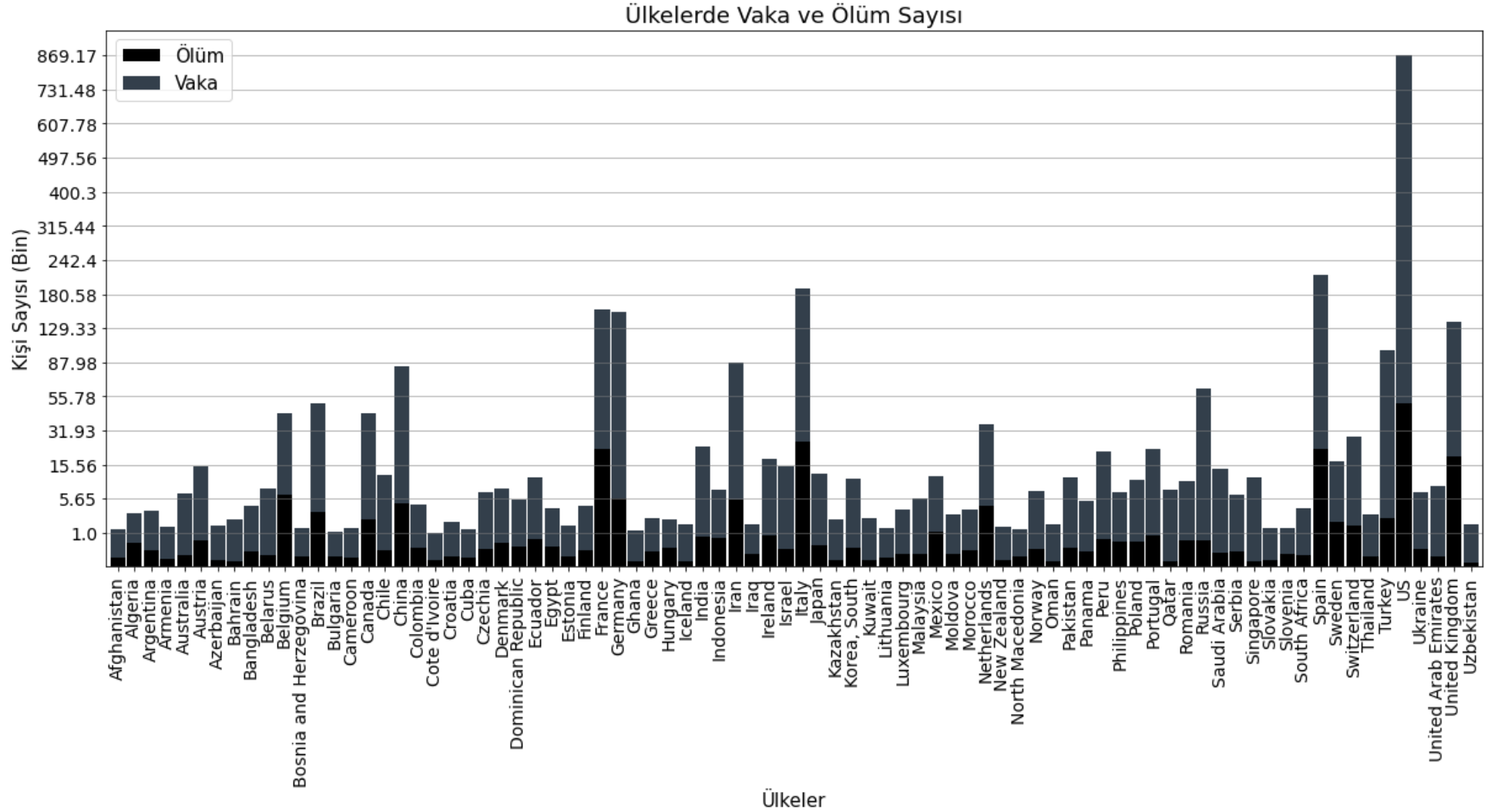
```
dts['Deaths'] = dts['Deaths']**ks
dts['Confirmed'] = dts['Confirmed']**ks - dts['Deaths']
```

```
dts.plot(kind='bar', width=0.9, figsize=(20,8), stacked=True, color=['#000', '#333F4B'])
plt.xticks(np.linspace(1000**ks, dts[['Confirmed','Deaths']].max().sum(), 15),
            np.round((np.linspace(1000**ks, ((dts['Confirmed']).max() + dts['Deaths'].max())**(1/ks))**ks, 15)**(1/ks)).astype('int')/1000,2))
plt.xlabel('Ülkeler',fontsize=15)
plt.ylabel('Kişi Sayısı (Bin)', fontsize=15)
plt.title("Ülkelerde Vaka ve Ölüm Sayısı", fontsize=18)
```



```
plt.grid(axis='y', color='gray', linestyle='-', linewidth=0.5)
plt.legend(['Ölüm', 'Vaka'], loc='upper left', fontsize=15)
```

<matplotlib.legend.Legend at 0x7f0f47b02898>



Koronavirüs Kaynaklı Ölüm

• Corona virüsler (CoV), hafif düzeyli soğuk algınlığından şiddetli solunum sıkıntısına kadar farklı tablolara neden olan büyük bir virüs ailesidir.

Coronavirus'ların insanlarda dolaşımda olan alt tipleri (HCoV-229E, HCoV-OC43, HCoV-NL63 ve HKU1-CoV) çoğunlukla soğuk algınlığına sebep

olan virüslerdir.

- SARS-CoV, 21. yüzyılın ilk uluslararası sağlık acil durumu olarak 2003 yılında, daha önceden bilinmeyen bir virüs halinde ortaya çıkmış olup yüzlerce insanın hayatını kaybetmesine neden olmuştur.
- Yaklaşık 10 yıl sonra Coronavirus ailesinden, daha önce insan ya da hayvanlarda varlığı gösterilmemiş olan MERSCoV (Middle East Respiratory Syndrome Coronavirus) Eylül 2012'de ilk defa insanlarda Suudi Arabistan'da tanımlanmış; ancak daha sonra aslında ilk vakaların Nisan 2012'de Ürdün Zarqa'daki bir hastanede görüldüğü ortaya çıkmıştır. SARS Coronavirus'u ile uzaktan bağlantılı olmasına rağmen, yaşanmış olan SARS tecrübesinden ötürü endişe oluşturmuştur.
- Yaklaşık 10 yıl sonra Coronavirus ailesinden, daha önce insan ya da hayvanlarda varlığı gösterilmemiş olan MERSCoV (Middle East Respiratory Syndrome Coronavirus) Eylül 2012'de ilk defa insanlarda Suudi Arabistan'da tanımlanmış; ancak daha sonra aslında ilk vakaların Nisan 2012'de Ürdün Zarqa'daki bir hastanede görüldüğü ortaya çıkmıştır. SARS Coronavirus'u ile uzaktan bağlantılı olmasına rağmen, yaşanmış olan SARS tecrübesinden ötürü endişe oluşturmuştur.
- Aşağıdaki Dünyadaki Toplam Ölüm Değerleri grafiğinde Covid-19 kaynaklı ölüm eğrisi ve günlük ölüm sayısına ait eğri verilmiştir.

Kaynak: <https://hisarhospital.com/corona-virusu-kendisi-kucuk-ama-tahribati-buyuk/>

```
plt.figure(figsize=(20,8))

plt.plot(datatoplam.groupby('Date').sum().index, datatoplam.groupby('Date').sum()['Deaths'], label = "Dünyadaki Toplam Ölüm Miktarı")
plt.plot(datatoplam.groupby('Date').sum().index, artis(datatoplam.groupby('Date').sum()['Deaths'].tolist()), label = "Dünyaki Günlük Ölüm Miktarı")

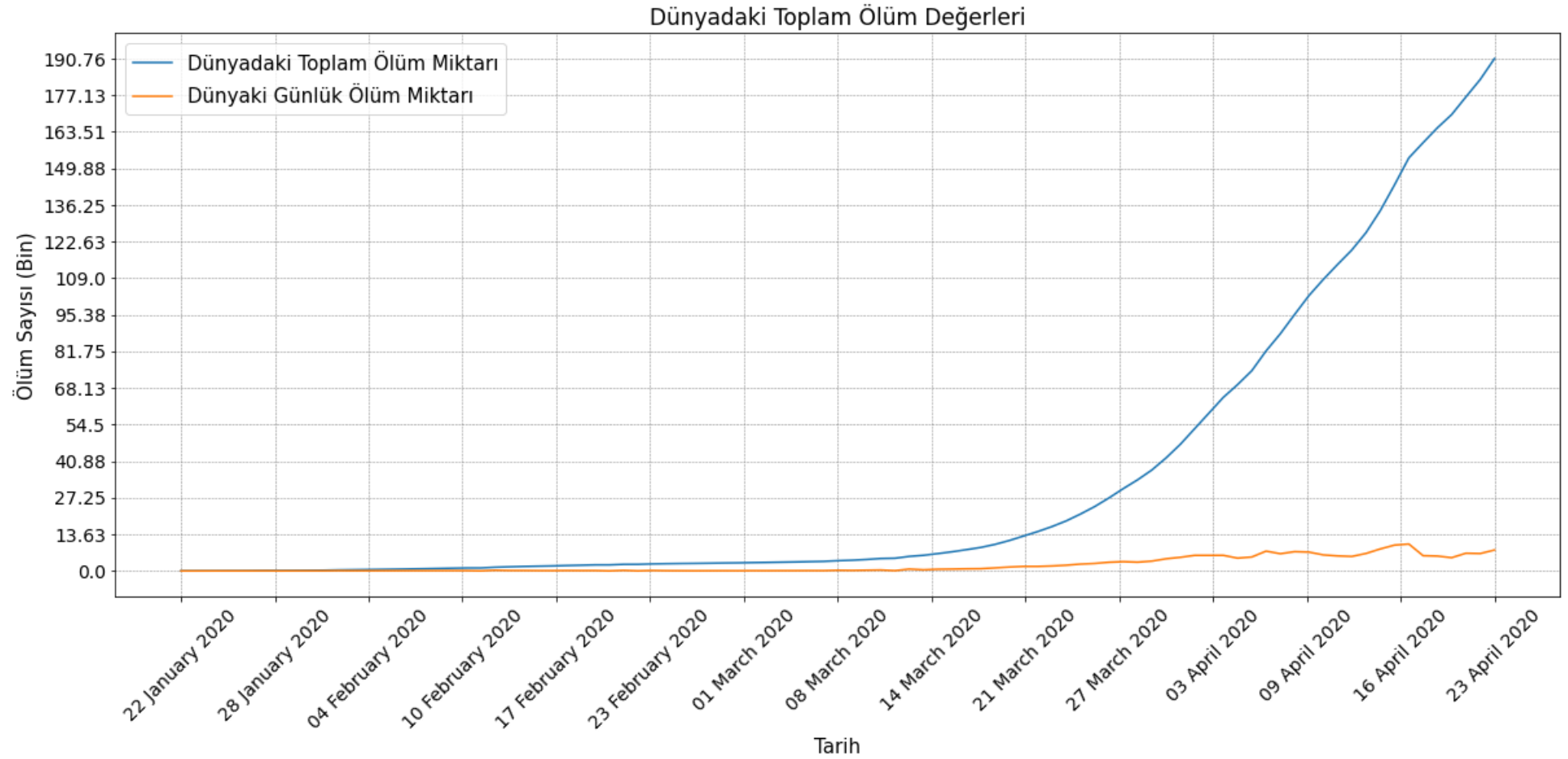
plt.rc(['xtick','ytick'], labels=14)
plt.yticks(np.linspace(0, max(datatoplam.groupby('Date').sum()['Deaths']), 15),
           np.round(np.linspace(0, max(datatoplam.groupby('Date').sum()['Deaths']), 15)/1000,2))
plt.xticks(np.linspace(data['Date'].min().value,
                        data['Date'].max().value, 15).astype('datetime64[ns]'),
           pd.to_datetime(np.linspace(data['Date'].min().value,
                                     data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Ölüm Sayısı (Bin)', fontsize=15)
plt.title('Dünyadaki Toplam Ölüm Değerleri', fontsize=17)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```

↩

<matplotlib.legend.Legend at 0x7f0f45ca3ac8>



Ülkelerin Ölüm Eğrileri

- Avrupa devletlerinin ve Amerika Birleşik Devletlerinin geç aldığı önlemler, vatandaşların önlemlere uymaması ve salgını hafife alması vaka sayısının hızla artmasına neden olmuştur.
- Vakanın hızlı artışı, ölümlerin ani artışını tetiklemiştir.
- Devletlerin sağlık sistemlerinin ani artan vaka ve ölüm sayısını taşıyamadığı ve bu durum zaten hızlı şekilde artan ölüm sayısını daha da hızlandırdı.
- Çin, salgının başından itibaren aldığı önlemlerle ve halkın bu önlemlere büyük ölçüde uymasıyla, salgını bastırmayı başardı.

- Aşağıdaki Bazı Ülkelerdeki Ölüm Değerleri grafiğinde ülkelerin ölüm eğrisi verilmiştir. Grafikte Çin ile diğer ülkeler arasındaki ölüm artış hızı açıkça görünüyor.

```
plt.figure(figsize=(20,8))

plt.plot(data[(data['Country/Region'] == 'Turkey')]['Date'],
         data[(data['Country/Region'] == 'Turkey')]['Deaths'], c='Red', label = "Türkiye")
plt.plot(data[(data['Country/Region'] == 'Italy')]['Date'],
         data[(data['Country/Region'] == 'Italy')]['Deaths'], c='Green', label = "İtalya")
plt.plot(data[(data['Country/Region'] == 'US')]['Date'],
         data[(data['Country/Region'] == 'US')]['Deaths'], c='Blue', label = "Amerika")
plt.plot(data[(data['Country/Region'] == 'Spain')]['Date'],
         data[(data['Country/Region'] == 'Spain')]['Deaths'], c='Yellow', label = "İspanya")
plt.plot(data[(data['Country/Region'] == 'Iran')]['Date'],
         data[(data['Country/Region'] == 'Iran')]['Deaths'], c='Black', label = "İran")
plt.plot(data[(data['Country/Region'] == 'China')]['Date'],
         data[(data['Country/Region'] == 'China')]['Deaths'], c='Violet', label = "Çin")

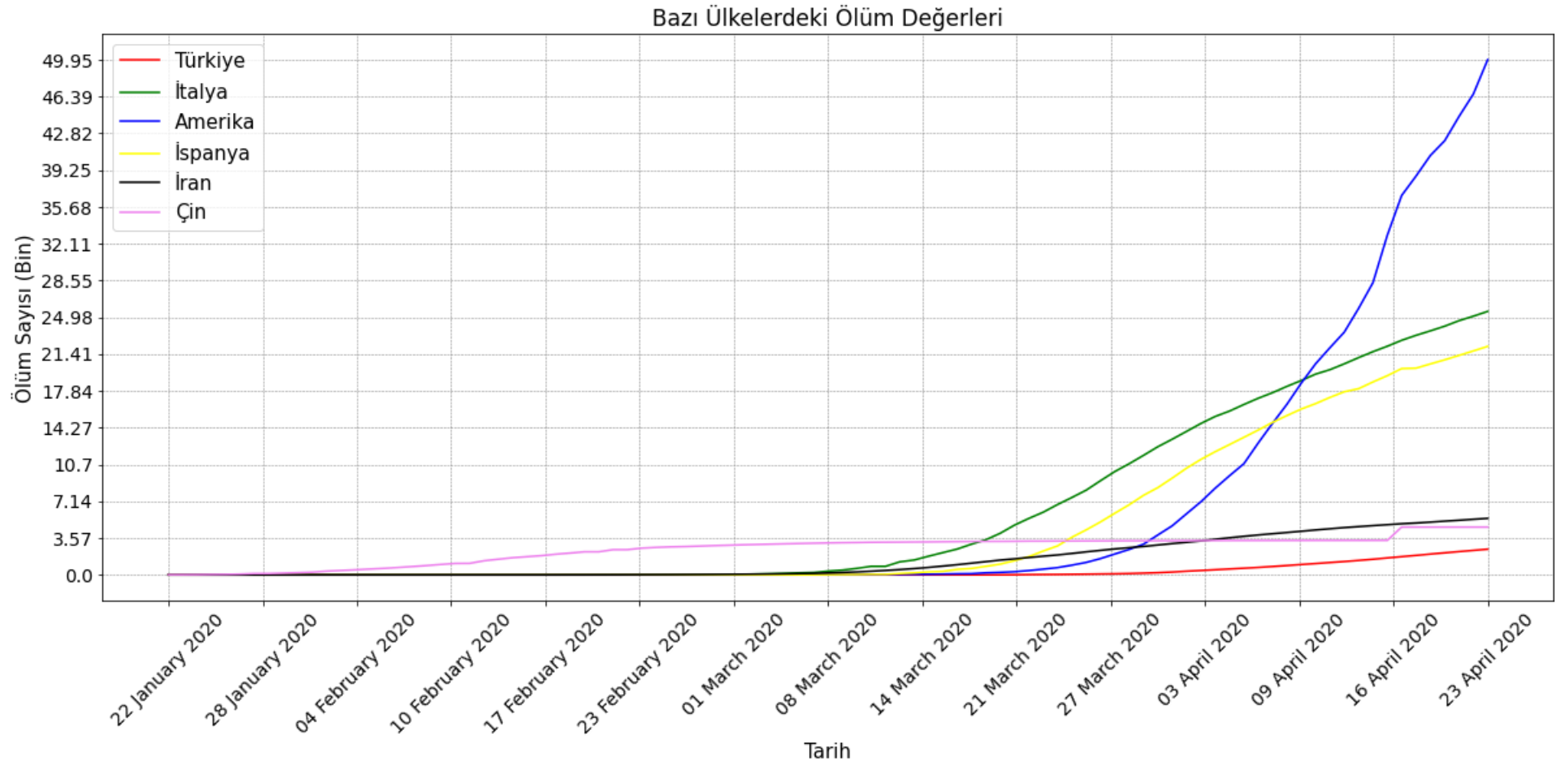
plt.rc(['xtick','ytick'], labels=14)
plt.yticks(np.linspace(0, max(data[(data['Country/Region'] == 'US')]['Deaths']), 15),
          np.round(np.linspace(0, max(data[(data['Country/Region'] == 'US')]['Deaths']), 15)/1000,2))
plt.xticks(np.linspace(data['Date'].min().value,
                      data['Date'].max().value, 15).astype('datetime64[ns]'),
          pd.to_datetime(np.linspace(data['Date'].min().value,
                                     data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Ölüm Sayısı (Bin)', fontsize=15)
plt.title("Bazı Ülkelerdeki Ölüm Değerleri", fontsize=17)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f461c3320>



Türkiye'de Koronavirüs Pandemisi

- Dünya geneline yayılan 2019-20 koronavirüs salgını'nın Türkiye'deki ilk tespit edilen COVID-19 vakası Türkiye Cumhuriyeti Sağlık Bakanlığı tarafından 10 Mart 2020'de açıklandı.
- Salgın Türkiye'de sosyal, ekonomik, siyasî, iktisadî, idarî, hukukî, askerî ve dinî alanlarda birçok önemli etkilere ve sonuçlara neden olan radikal kararlar alınmasına neden oldu.
- Salgın Türkiye'de sosyal, ekonomik, siyasî, iktisadî, idarî, hukukî, askerî ve dinî alanlarda birçok önemli etkilere ve sonuçlara neden olan radikal kararlar alınmasına neden oldu.

- Ülkede ilk, orta ve liselerde eğitim-öğretim askıya alınırken; bütün üniversitelerde bahar dönemi dersleri iptal edilerek sınavlar ertelendi.
- Diyanet İşleri Başkanlığı cuma namazı başta olmak üzere cami ve mescitlerde cemaatle namaza ara verildiğini duyurdu.
- Tüm restaurantlar, cafe, müze, dersane, kurs, alışveriş merkezi, kahvehane, spor salonları, konser mekanları, dernek, düğün salonu, piknik ve ören yerleri, park ve bahçeler ziyaretçilere kapatıldı.
- Ülkedeki bütün futbol ligleri ertelendi ve tüm spor müsabakaları bir sonraki açıklamaya kadar iptal edildi
- Koronavirüs salgınının cezaevlerine sıçraması ve oluşabilecek kamu düzeninin bozulması riski nedeniyle bazı suçlarda infaz indirimi öngören ve denetimli serbestlik süresinin geçici olarak 3 yıla çıkarılmasını öngören yasa mecliste kabul edilerek yasalaştı ve 90 bin hükümlüye tahliye yolu açıldı.
- Hükûmet, salgının yayılım hızını düşürmek ve insanlar arasında sosyal mesafeyi korumak için ilk olarak 65 yaş ve üstü kişilere sokağa çıkma yasağı uygulaması başlattı. Daha sonra bu kısıtlamayı 20 yaş ve altı çocuk ve gençleri de kapsayacak şekilde genişletti.
- 11-12 Nisan 2020'de ise, 2000 yılındaki nüfus sayımından 20 yıl sonra ilk kez geniş çaplı bir sokağa çıkma yasağı ilan edildi. İçişleri Bakanlığı Türkiye'nin 30 büyükşehir belediyesi ile akciğer hastalıklarının sık görüldüğü Zonguldak ilinde haftasonu iki gün boyunca sokağa çıkma yasağı ilan edildi. Sokağa çıkmağa yasağı ilerleyen zamanlarda tekrarlandı.
- Aşağıdaki Türkiye'de Covid-19 Eğrileri grafiğinde Covid-19 vaka, ölüm ve iyileşme eğrileri bulunmaktadır.

Kaynak: https://tr.wikipedia.org/wiki/T%C3%BCrkiye%27de_2020_koronavir%C3%BCs_pandemisi

```
plt.figure(figsize=(20,8))

d = data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] != 0)]

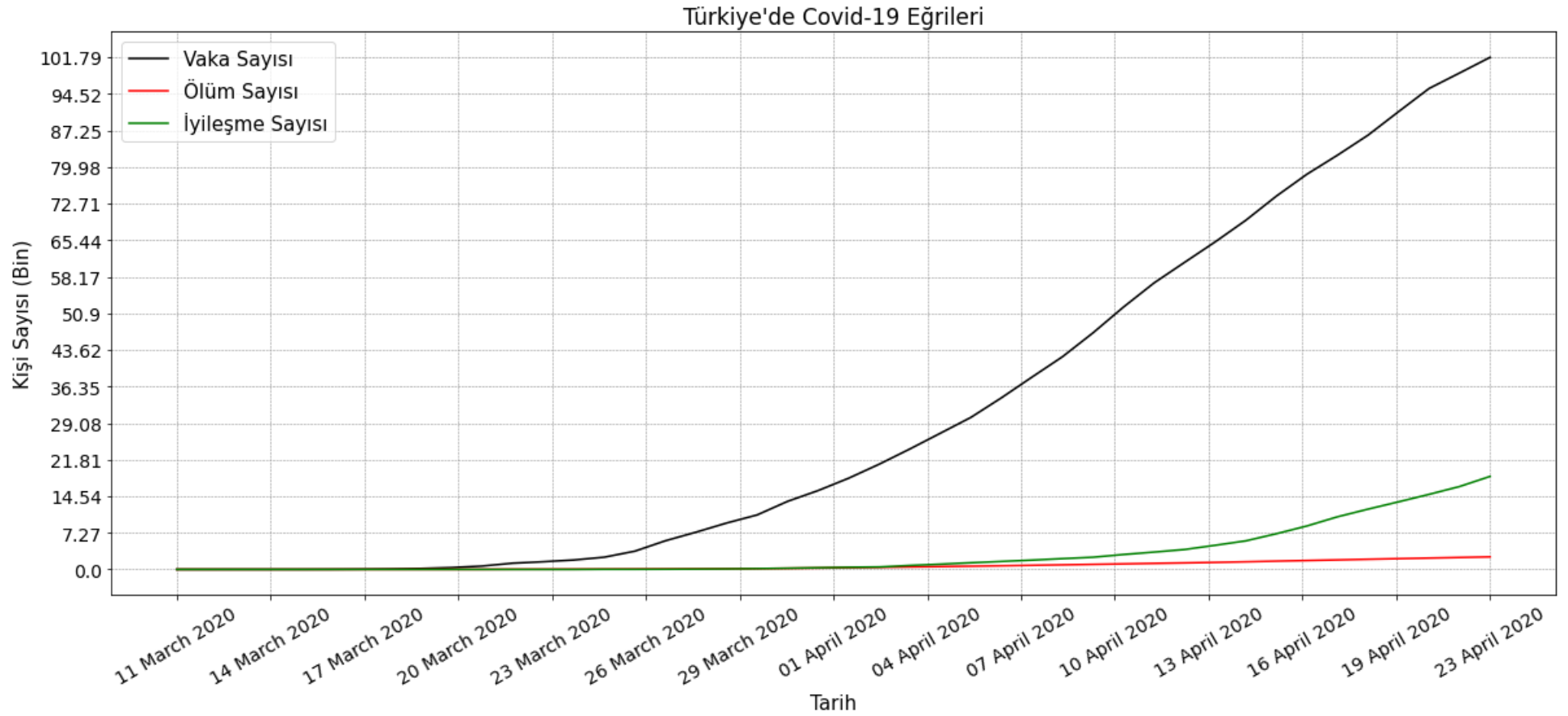
plt.plot(d['Date'], d['Confirmed'], c='Black', label = "Vaka Sayısı")
plt.plot(d['Date'], d['Deaths'], c='Red', label = "Ölüm Sayısı")
plt.plot(d['Date'], d['Recovered'], c='Green', label = "İyileşme Sayısı")

plt.rc(['xtick','ytick'], labelsiz=14)
plt.yticks((np.linspace(0, max(data[(data['Country/Region'] == 'Turkey') & (data['Deaths'] != 0)][['Confirmed']], 15)),
            np.round((np.linspace(0, max(data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] != 0)][['Confirmed']], 15)/1000),2))
plt.xticks(np.linspace(d['Date'].min().value, d['Date'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(d['Date'].min().value, d['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=30)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Kişi Sayısı (Bin)', fontsize=15)
plt.title("Türkiye'de Covid-19 Eğrileri", fontsize=17)
plt.grid(linestyle='dashed', color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f4781c8d0>



Türkiye'de Koronavirüs Pandemisi

- Sağlık Bakanı Fahrettin Koca, 1 Nisan 2020'de yaptığı açıklamada koronavirüs vakalarının tüm Türkiye'ye yayıldığını açıkladı.
- 19 Nisan itibarıyla Türkiye vaka sayısında, salgının ilk başladığı ülke olan Çin'i geride bıraktı.
- 10 Nisan'da paylaşılan verilere göre 28 bin vaka ile İstanbul en çok vakanın olduğu şehir olurken, onu sırasıyla İzmir, Ankara, Kocaeli ve Konya takip etti.
- Dışişleri Bakanı Mevlüt Çavuşoğlu 18 Nisan'da yaptığı açıklamada yurt dışında koronavirüsten 342 Türk vatandaşının hayatını kaybettiğini açıkladı.

Kaynak: https://tr.wikipedia.org/wiki/T%C3%BCrkiye%27de_2020_koronavir%C3%BCs_pandemisi

```

dts = data[(data['Confirmed'] > 0) & (data['Country/Region'] == 'Turkey')][['Date', 'Deaths', 'Recovered', 'Confirmed']].set_index('Date').sort_index()
ks = 0.4

dts['Deaths'] = dts['Deaths']**ks
dts['Recovered'] = dts['Recovered']**ks - dts['Deaths']
dts.loc[dts['Recovered'] < 0, 'Recovered'] = 0
dts['Confirmed'] = dts['Confirmed']**ks - (dts['Recovered'] + dts['Deaths'])

dts.index = pd.to_datetime(dts.index).strftime('%d %B %Y')

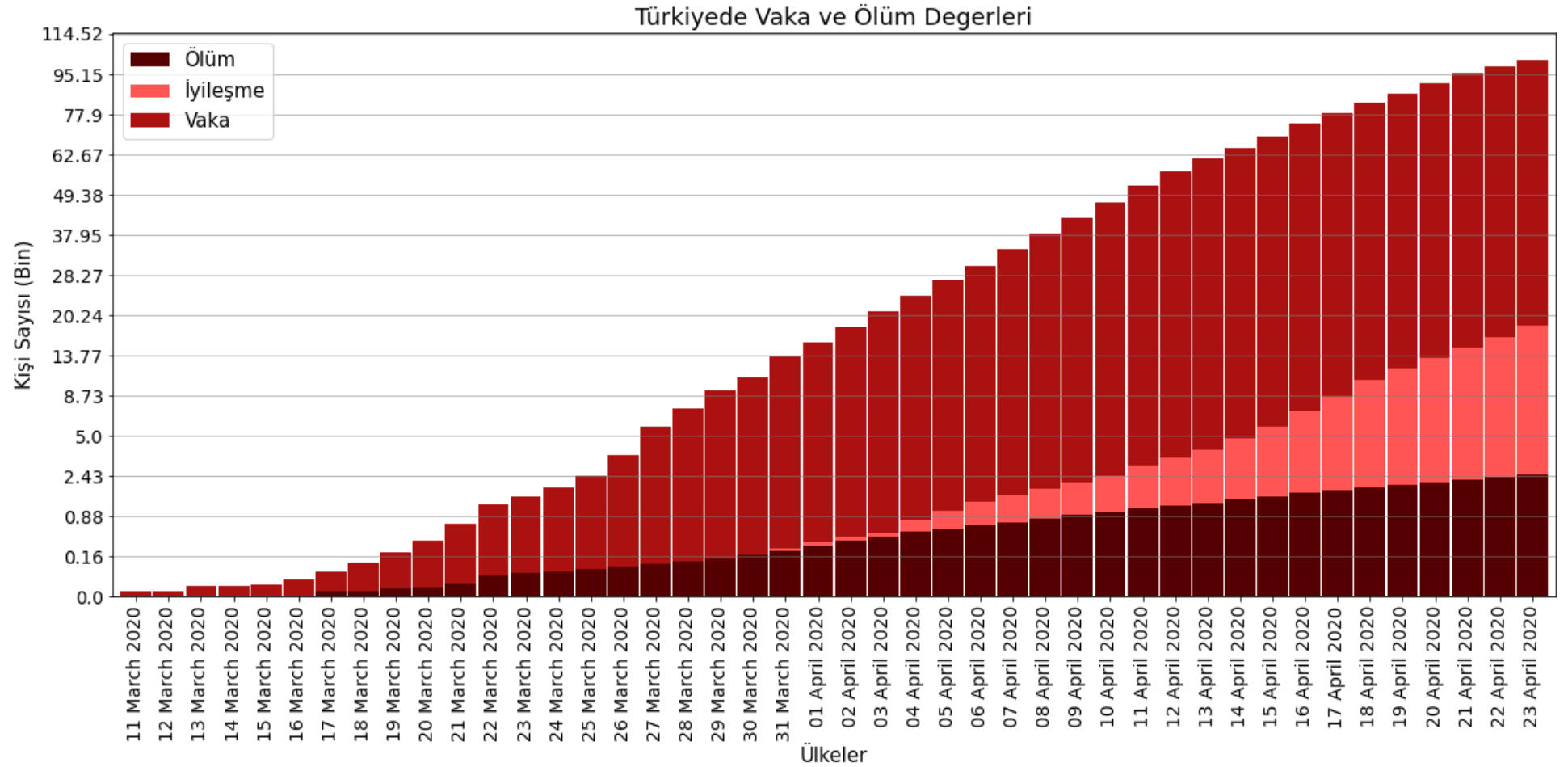
dts.plot(kind='bar', width=0.95, figsize=(20,8), stacked=True, color=['#500', '#F55', '#A11'])

plt.yticks(np.linspace(0, dts[['Confirmed', 'Recovered', 'Deaths']].max().sum(), 15),
           np.round((np.linspace(0, ((dts['Confirmed'].max()+ dts['Recovered'].max() + dts['Deaths'].max())**(1/ks))**ks, 15)**(1/ks)).astype('int')/1000,2))
plt.xlabel('Ülkeler',fontsize=15)
plt.ylabel('Kişi Sayısı (Bin)', fontsize=15)
plt.title("Türkiyede Vaka ve Ölüm Degerleri", fontsize=18)
plt.grid(axis='y', color='gray', linestyle='-', linewidth=0.5)
plt.legend(['Ölüm', 'İyileşme', 'Vaka'], loc='upper left',fontsize=15)

```




```
<matplotlib.legend.Legend at 0x7f0f47830cc0>
```



Koronavirüs nasıl tedavi edilir?

- DSÖ verilerine göre yeni tip koronavirüse yakalananların yüzde 80'i, hatta yüzde 85'i özel bir tedaviye ihtiyaç duymaksızın hastalığı atlatabiliyor.
- Koronavirüs bir bakteri olmadığı için antibiyotikler işe yaramaz. Şu anda koronavirüse özgü bir tedavi yoktur. Koronavirüs tedavisi ve koronavirüs aşısı geliştirme çalışmaları sürmektedir.
- Sıtma tedavisinde kullanılan klorokin Şubat 2020'de Çin'de tedavi için araştırılmış, öncül sonuçlar maddeyi etkin göstermiştir.
- Klorokin ve hidroklorokin, ikinci madde daha güçlü ve güvenli olacak şekilde SARS-CoV-2'yi in vitro inhibe etmektedir.

- COVID-19'un çoğu vakası solunum cihazıyla gerçekleştirilebilecek mekanik ventilasyon (makine yoluyla solunum desteği) gerektirmese de vakaların belli bir yüzdesinde bu gerekmektedir.
- Aşağıdaki Türkiyede Ölüm ve İyileşme Eğrileri grafiğinde Türkiye'deki iyileşme ve ölüm eğrileri bulunmaktadır.

Kaynak: <https://www.medicalpark.com.tr/coronavirus/hg-2287>

Kaynak2: https://tr.wikipedia.org/wiki/Koronavir%C3%BCs_hastal%C4%B1%C4%9F%C4%B1_2019#Tedavi

Kaynak3: <https://t24.com.tr/haber/20-soruda-yeni-tip-koronavirus-covid-19-hakkinda-her-sey,868277>

```
plt.figure(figsize=(20,8))

d = data[(data['Country/Region'] == 'Turkey') & (data['Deaths'] > 0)]
plt.plot(d['Date'], d['Deaths'], c='Red', label = "Ölüm Sayısı")
plt.plot(d['Date'], d['Recovered'], c='Green', label = "İyileşme Sayısı")

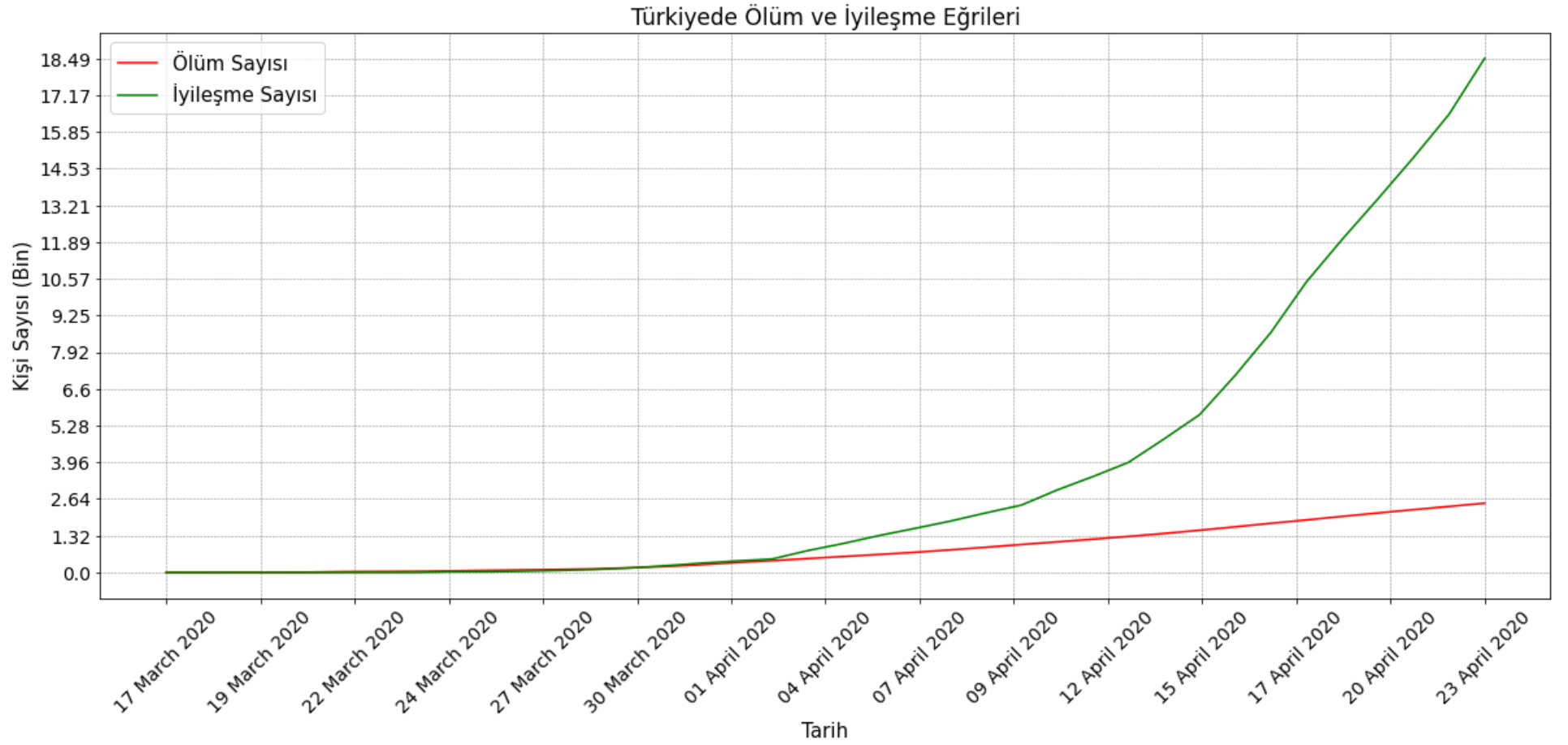
plt.rc(['xtick','ytick'], labels=14)
plt.yticks((np.linspace(0, max(d['Recovered']), 15)), np.round((np.linspace(0, max(d['Recovered']), 15)/1000),2))
plt.xticks(np.linspace(d['Date'].min().value,d['Date'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(d['Date'].min().value, d['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Kişi Sayısı (Bin)', fontsize=15)
plt.title("Türkiyede Ölüm ve İyileşme Eğrileri", fontsize=17)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f4780bf60>



Günlük Vaka Artışı

- Günlük vaka artışı salgının ilerleyişi açısından önemlidir.
- Plato dönemi vaka artış ve ölüm artış hızının duraklamaya girdiği, stabilleşmenin başladığı dönemdir.
- Hastalığın en üst sınırı yani zirve noktasını görmesi pik noktası olarak adlandırılıyor.
- Aşağıdaki Bazı Ülkelerde Günlük Vaka Artışı grafiğinde ülkelerin günlük vaka artışını görebiliriz. Günlük vaka artışının genel olarak düşmeye başladığı görünüyor.

```
d = data[(data['Country/Region'] == 'US')]
```

```

plt.figure(figsize=(20,8))
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Turkey')]['Confirmed'].tolist()), c='Red', label = "Türkiye")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Italy')]['Confirmed'].tolist()), c='Green', label = "İtalya")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'US')]['Confirmed'].tolist()), c='Blue', label = "Amerika")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Spain')]['Confirmed'].tolist()), c='Yellow', label = "İspanya")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Iran')]['Confirmed'].tolist()), c='Black', label = "İran")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'China')]['Confirmed'].tolist()), c='Violet', label = "Çin")

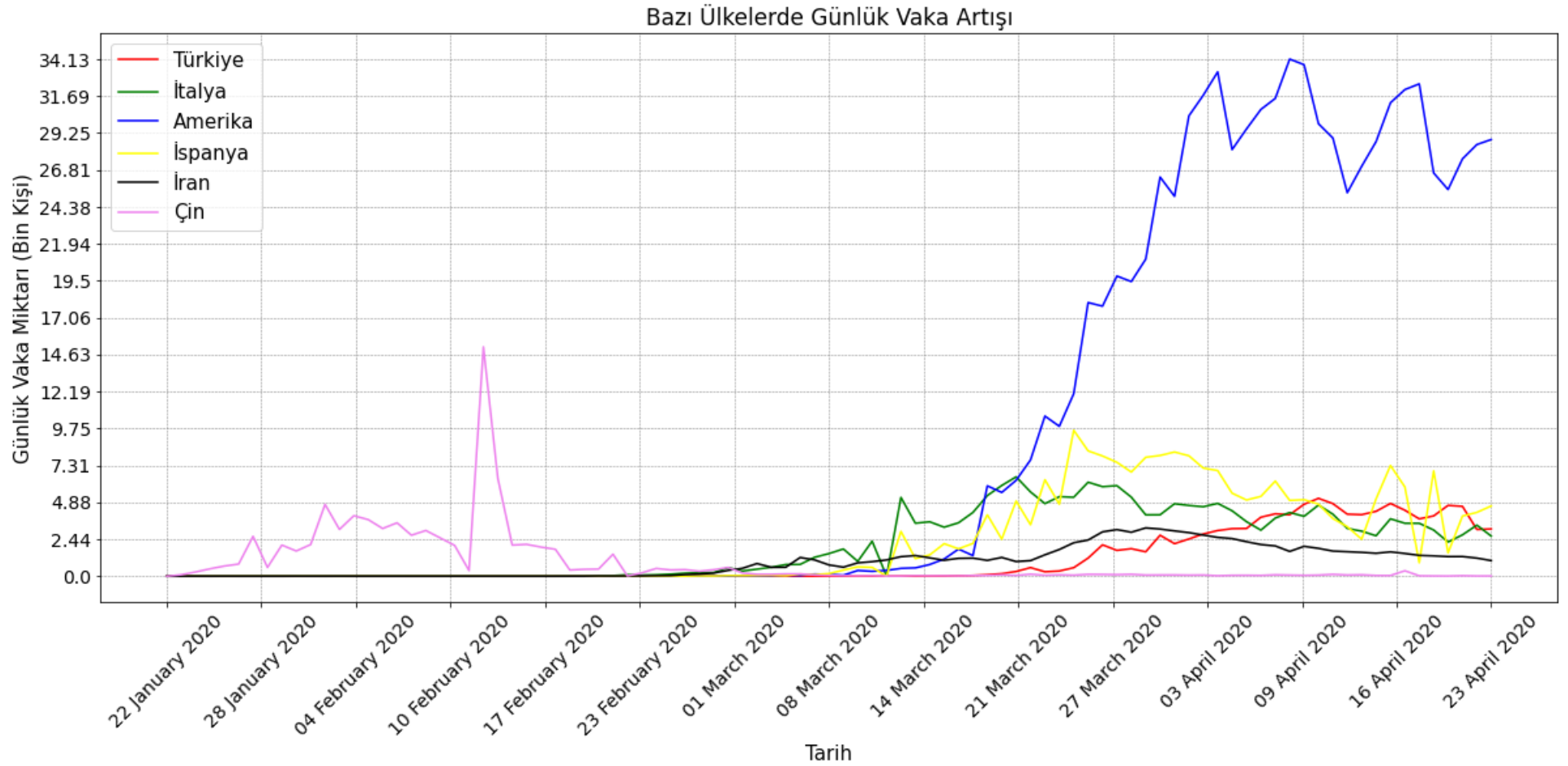
plt.yticks((np.linspace(0, max(artis(d['Confirmed'].tolist()), 15)), np.round((np.linspace(0, max(artis(d['Confirmed'].tolist()), 15)/1000),2))
plt.xticks(np.linspace(data['Date'].min().value,
                        data['Date'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(data['Date'].min().value,
                                      data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.rc(['xtick','ytick'], labels=14)
plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Günlük Vaka Miktarı (Bin Kişi)', fontsize=15)
plt.title('Bazı Ülkelerde Günlük Vaka Artışı', fontsize=17)
plt.grid(color='gray', linestyle='dashed', linewidth=0.5)
plt.legend(fontsize=15)

```



<matplotlib.legend.Legend at 0x7f0f47c9c470>



```
gv = list()
for i, row in data[(data['Confirmed'] > 20000) & (data['Date'] == data['Date'].max())].iterrows():
    gv.append([row['Country/Region'],artis(data[data['Country/Region'] == row['Country/Region']]['Confirmed'].tolist())])
```

Türkiye'de Günlük Vaka Artışı

- Vaka sayısı ilk vakadan bu yana hızlı bir artış göstermişti.
- Vaka sayısı artmaya devam ediyor fakat son günlerde vaka artış hızında bir düşüş var.

• Aşağıdaki Türkiye'deki Günlük Vaka Artışı grafiğinde günlük vaka artışını inceleyebiliriz. Eğer günlük vaka artışı 5 bin miktarına tekrar

```
d = data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] != 0)]
dg = df[(df['Country/Region'] == 'Turkey') & (df['Confirmed'] != 0)]
plt.figure(figsize=(20,8))
plt.rc(['xtick','ytick'], labels=14)
plt.xticks(np.arange(0, max(artis(d['Confirmed']).tolist()+500, 500))
plt.xticks(np.linspace(d['Date'].min().value, d['Date'].max().value, 15).astype('datetime64[ns]'), np.linspace(dg['Date'].min(), dg['Date'].max(), 15).astype('int'))
plt.plot(d['Date'], artis(d['Confirmed'].tolist()), marker='o', c='black', label = "Türkiye")
plt.xlabel('Dünyadaki İlk Vakadan Sonra Geçen Gün', fontsize=15)
plt.ylabel('Günlük Vaka Miktarı (Kişi)', fontsize=15)
plt.title("Türkiye'deki Günlük Vaka Artışı", fontsize=17)
plt.grid(color='gray', linestyle='dashed', linewidth=0.5)
plt.legend(fontsize=15, loc=2)
```

☐ <matplotlib.legend.Legend at 0x7f0f47ddc080>



Günlük Ölüm Artışı

- Günlük vaka sayısındaki gibi günlük ölüm sayısında da genel olarak bir düşüş görünüyor.
- Covid-19 için spesifik bir tedavi olmasa da hangi tedavilerin daha etkili olduğunun öğrenilmesi tedavilerin sonuç vermesinde etkili oldu.
- Aşağıdaki Bazı Ülkelerde Günlük Vaka Artışı grafiğinde ülkelerin günlük ölüm sayısını görebiliriz. Önlemlerin devam etmesiyle, günlük vaka ve ölüm sayısının düşmeye devam edecek gibi görünüyor.

```
d = data[(data['Country/Region'] == 'China')]

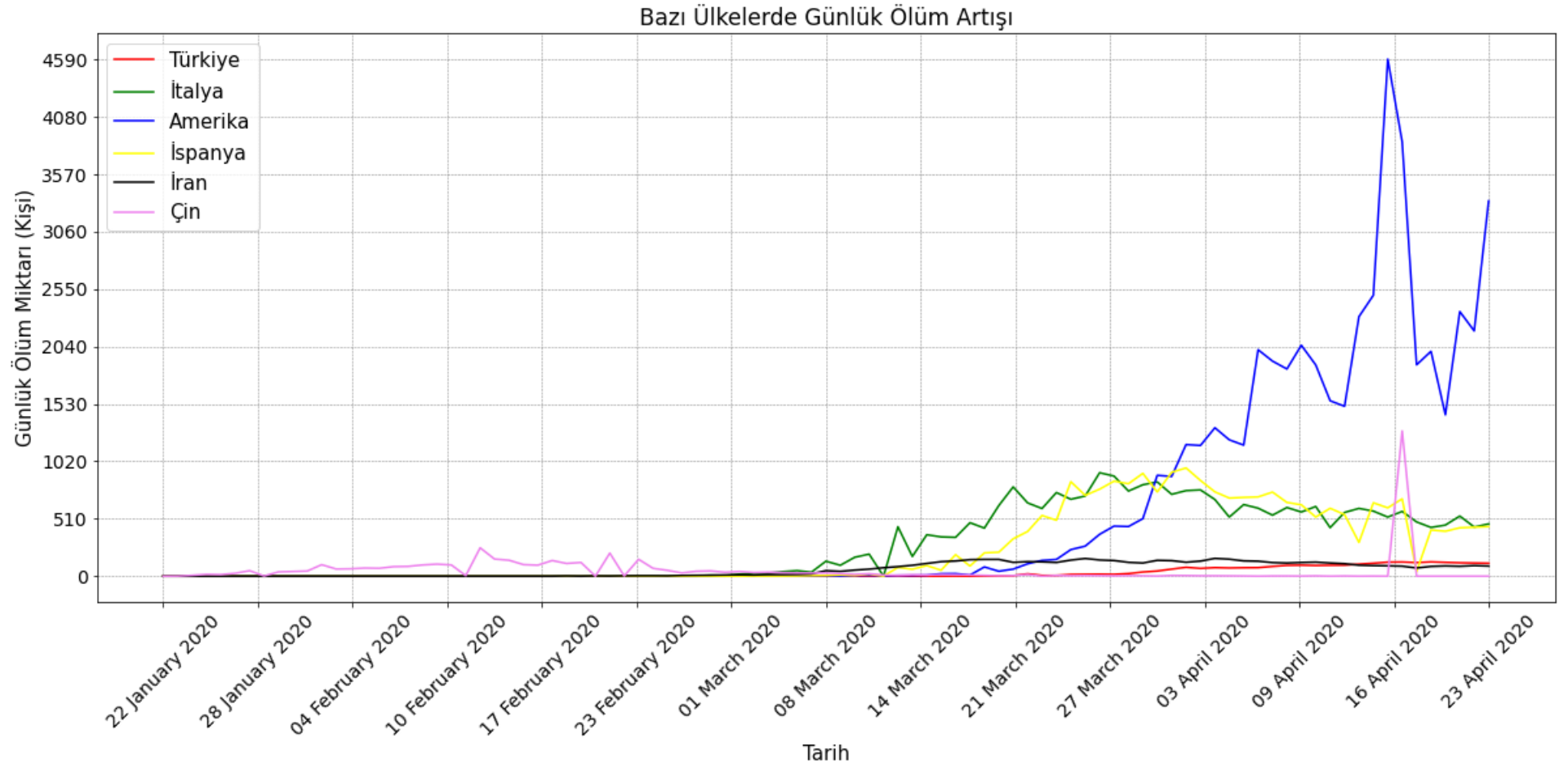
plt.figure(figsize=(20,8))

plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Turkey')]['Deaths'].tolist()), c='Red', label = "Türkiye")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Italy')]['Deaths'].tolist()), c='Green', label = "İtalya")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'US')]['Deaths'].tolist()), c='Blue', label = "Amerika")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Spain')]['Deaths'].tolist()), c='Yellow', label = "İspanya")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'Iran')]['Deaths'].tolist()), c='Black', label = "İran")
plt.plot(d['Date'], artis(data[(data['Country/Region'] == 'China')]['Deaths'].tolist()), c='Violet', label = "Çin")
plt.yticks((np.linspace(0, max(artis(data[(data['Country/Region'] == 'US')]['Deaths'].tolist()), 10).astype('int')//10)*10)
plt.xticks(np.linspace(data['Date'].min().value,
                        data['Date'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(data['Date'].min().value,
                                      data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.rc(['xtick','ytick'], labels=14)
plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Günlük Ölüm Miktarı (Kişi)', fontsize=15)
plt.title('Bazı Ülkelerde Günlük Ölüm Artışı', fontsize=17)
plt.grid(color='gray', linestyle='dashed', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f46197908>



Türkiye'de Günlük Ölüm Artışı

- İlk vakadan kısa bir süre sonra ölümler başlamıştı.
- Günlük ölüm sayısı, günlük vaka sayısı gibi son günlerde düşüşte görünüyor.
- Bu düşüşleri uygulanan karantinaların ve tecrübe edilen tedavilerin bir sonucu olarak görebiliriz.
- Aşağıdaki Türkiye'deki Günlük Ölüm Artışı grafiğinde günlük ölüm artışını görebiliyoruz. Günlük ölümün tekrar yükselişe geçmeyeceğini varsayarsak peak noktasına ulaşmış diyebiliriz.


```

d = data[(data['Country/Region'] == 'Turkey') & (data['Deaths'] > 0)]
dg = df[(df['Country/Region'] == 'Turkey') & (df['Deaths'] > 0)]
ilk_vaka = df[(df['Country/Region'] == 'Turkey') & (df['Confirmed'] != 0)][['Date']].min()

plt.figure(figsize=(20,8))
plt.plot(d['Date'], artis(d['Deaths'].tolist()), marker='x', c='Red', label = "Türkiye")

plt.rc(['xtick','ytick'], labels=14)
plt.yticks(np.round(np.linspace(0, max(artis(d['Deaths'].tolist())), 12)))
plt.xticks(np.linspace(d['Date'].min().value, d['Date'].max().value, 15).astype('datetime64[ns]'),
            np.linspace(dg['Date'].min()-ilk_vaka, dg['Date'].max()-ilk_vaka, 15).astype('int'))

plt.xlabel("Türkiye'deki İlk Vakadan Sonra Geçen Gün", fontsize=15)
plt.ylabel("Günlük Ölüm Miktarı (Kişi)", fontsize=15)
plt.title("Türkiye'deki Günlük Ölüm Artışı", fontsize=17)
plt.grid(color='gray', linestyle='dashed', linewidth=0.5)
plt.legend(fontsize=15, loc=2)

```

☐ <matplotlib.legend.Legend at 0x7f0f47ba95f8>



```
data_test = pd.read_csv('https://raw.githubusercontent.com/yasnrk/Covid-19Analiz/master/TestSayisi.csv')
data_test['Date'] = pd.to_datetime(data_test['Date'])
```

Covid-19 Testi

- Dünya Sağlık Örgütü, hastalığın tanısı için çeşitli test protokolleri açıklamıştır.
- Standart test yöntemi, gerçek zamanlı revers transkriptaz polimeraz zincir reaksiyonu (rRT-PCR) olmaktadır. Bu test, kişiden alınacak nazofarengeal sürüntü veya balgam gibi solunum yollarından alınmış örnekler ile gerçekleştirilebilmektedir.
- PCR testleri, doğruluk oranının yüksekliği nedeniyle altın standarttır fakat test sonucu almanın yavaş olması (en az 4 ila 6 saat), deneyimli personel ihtiyacı ve lojistik olarak merkezî bir konumda yapılması gereksinimi yönünden testin yapılması zordur.
- 23 Mart 2020 itibarıyla, SARS-CoV-2'ye karşı üretilen IgM ve IgG antikorlarını tespit edebilen, 15 dakikada sonuç veren testler üretilmiş, Çin'de onaylanmasının yanı sıra Avrupa'da CE işareti almıştır.
- Bu testlerin doğruluk oranı PCR'a göre daha düşüktür, örneğin testlerden biri için %13'lük bir yanlış negatif (hastalığa sahip olmakla beraber negatif sonuç) oranı verilmektedir.
- Aşağıdaki Bazı Ülkelerdeki Test Sayıları grafiğinde test sayılarının eğrisi bulunmaktadır.

Kaynak: https://tr.wikipedia.org/wiki/Koronavir%C3%BCs_hastal%C4%B1%C4%9F%C4%B1_2019#Tan%C4%B1

```
plt.figure(figsize=(20,8))
test_limit = 100000

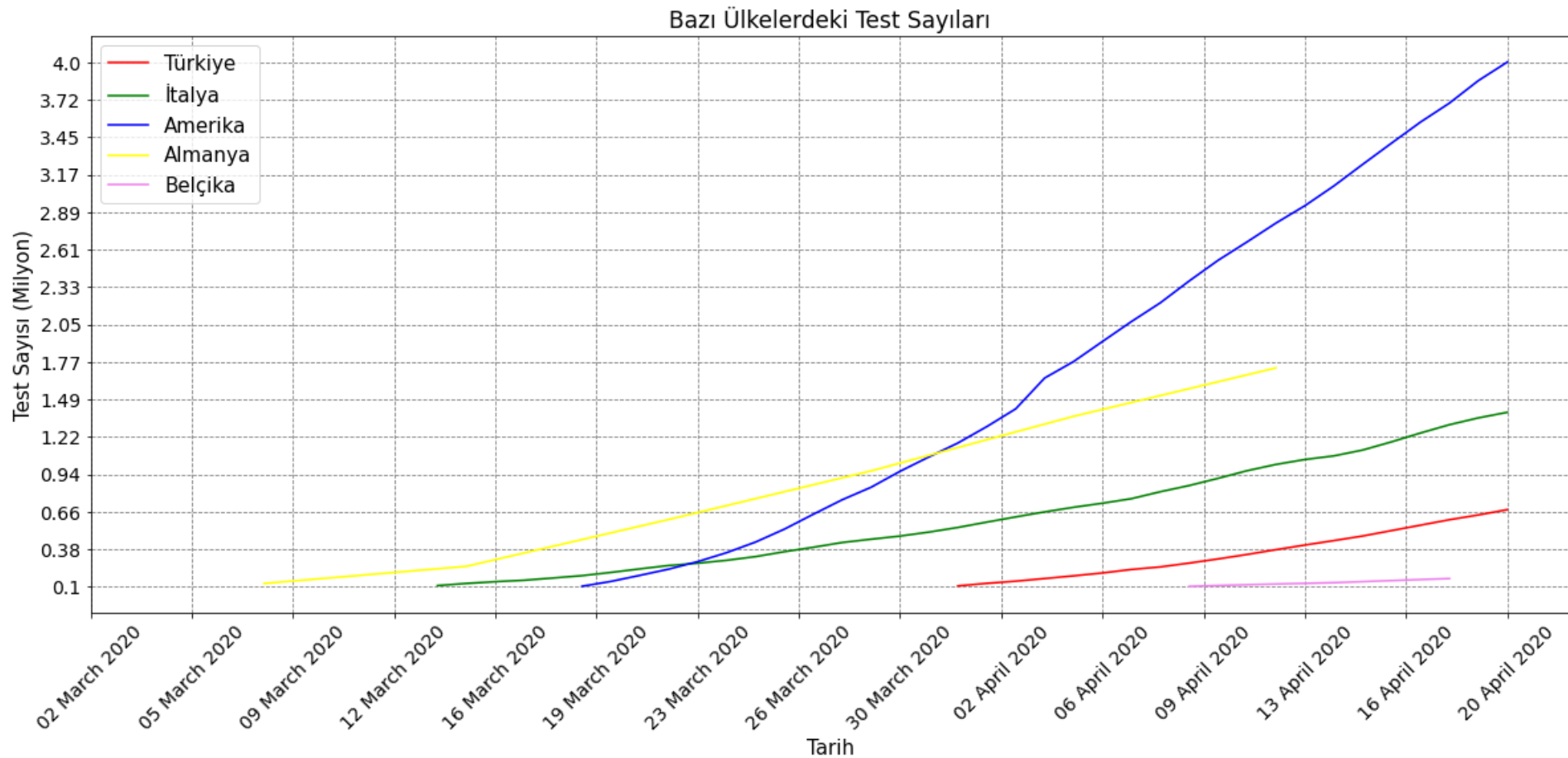
plt.plot(data_test[(data_test['Entity'] == 'Turkey') & (data_test['Total tests'] > test_limit)][['Date'],
            data_test[(data_test['Entity'] == 'Turkey') & (data_test['Total tests'] > test_limit)][['Total tests'], c='Red', label = "Türkiye")
plt.plot(data_test[(data_test['Entity'] == 'Italy') & (data_test['Total tests'] > test_limit)][['Date'],
            data_test[(data_test['Entity'] == 'Italy') & (data_test['Total tests'] > test_limit)][['Total tests'], c='Green', label = "İtalya")
plt.plot(data_test[(data_test['Entity'] == 'United States') & (data_test['Total tests'] > test_limit)][['Date'],
            data_test[(data_test['Entity'] == 'United States') & (data_test['Total tests'] > test_limit)][['Total tests'], c='Blue', label = "Amerika")
plt.plot(data_test[(data_test['Entity'] == 'Germany')][['Date'],
            data_test[(data_test['Entity'] == 'Germany') & (data_test['Total tests'] > test_limit)][['Total tests'], c='Yellow', label = "Almanya")
plt.plot(data_test[(data_test['Entity'] == 'Belgium') & (data_test['Total tests'] > test_limit)][['Date'],
            data_test[(data_test['Entity'] == 'Belgium') & (data_test['Total tests'] > test_limit)][['Total tests'], c='Violet', label = "Belçika")

plt.yticks(np.linspace(test_limit, data_test['Total tests'].max(), 15),
            np.round(np.linspace(test_limit, data_test['Total tests'].max(),15)/100000, 2))
plt.xticks(np.linspace(data_test[(data_test['Total tests'] > test_limit)][['Date']].min().value,
            data_test[(data_test['Total tests'] > test_limit)][['Date']].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(data_test[(data_test['Total tests'] > test_limit)][['Date']].min().value,
            data_test[(data_test['Total tests'] > test_limit)][['Date']].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Test Sayısı (Milyon)', fontsize=15)
```

```
plt.title("Bazı Ülkelerdeki Test Sayıları", fontsize=17)
plt.grid( color='gray', linestyle='dashed', linewidth=0.8)
plt.legend(fontsize=15)
```

↳ <matplotlib.legend.Legend at 0x7f0f4525ea58>



```
plt.figure(figsize=(20,8))
vaka_limiti = 10
```

```
plt.rc(['xtick','ytick'], labels=14)
plt.plot(data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] >= vaka_limiti)][Date],
         data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] >= vaka_limiti)][Confirmed], c='Red', label = "Türkiye")
plt.plot(data[(data['Country/Region'] == 'Italy') & (data['Confirmed'] >= vaka_limiti)][Date],
         data[(data['Country/Region'] == 'Italy') & (data['Confirmed'] >= vaka_limiti)][Confirmed], c='Green', label = "İtalya")
plt.plot(data[(data['Country/Region'] == 'US') & (data['Confirmed'] >= vaka_limiti)][Date],
         data[(data['Country/Region'] == 'US') & (data['Confirmed'] >= vaka_limiti)][Confirmed], c='Blue', label = "Amerika")
```

```
plt.plot(data[(data['Country/Region'] == 'Germany') & (data['Confirmed'] >= vaka_limiti)][ 'Date'],
         data[(data['Country/Region'] == 'Germany') & (data['Confirmed'] >= vaka_limiti)][ 'Confirmed'], c='Yellow', label = "Almanya")
plt.plot(data[(data['Country/Region'] == 'Belgium') & (data['Confirmed'] >= vaka_limiti)][ 'Date'],
         data[(data['Country/Region'] == 'Belgium') & (data['Confirmed'] >= vaka_limiti)][ 'Confirmed'], c='Violet', label = "Belçika")

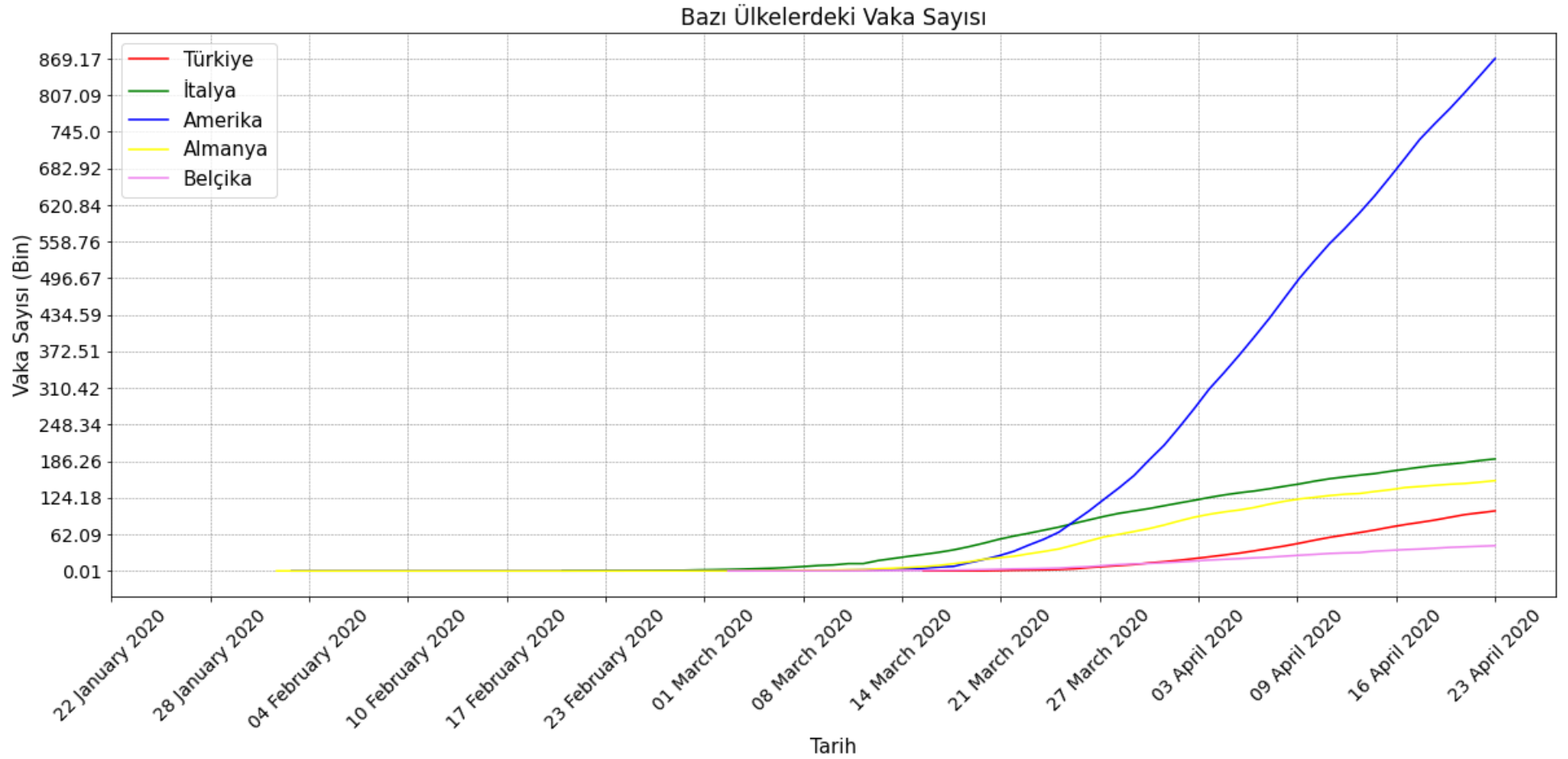
plt.yticks(np.linspace(vaka_limiti, max(data[(data['Country/Region'] == 'US')][ 'Confirmed']), 15),
           np.round(np.linspace(vaka_limiti, max(data[(data['Country/Region'] == 'US')][ 'Confirmed']), 15)/1000,2))
plt.xticks(np.linspace(data[(data['Confirmed'] >= vaka_limiti)][ 'Date'].min().value,
                       data[(data['Confirmed'] >= vaka_limiti)][ 'Date'].max().value, 15).astype('datetime64[ns]'),
           pd.to_datetime(np.linspace(data[(data['Confirmed'] >= vaka_limiti)][ 'Date'].min().value,
                                     data[(data['Confirmed'] >= vaka_limiti)][ 'Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Vaka Sayısı (Bin)', fontsize=15)
plt.title("Bazı Ülkelerdeki Vaka Sayısı", fontsize=17)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f45c20e10>



Covid-19 Ölüm Oranı

- Dünya Sağlık Örgütü, hastalığın tanısı için çeşitli test protokolleri açıklamıştır.
 - Farklı ülkelerdeki ölüm oranları karşılaştırıldığında ortaya çok değişken bir tablo çıkıyor.
 - Dünya Sağlık Örgütü'ne (DSÖ) göre bu farkın altında yatan başlıca nedenler ülkelerin demografik yapısı, tıbbi olanakları ve yapılan test sayısı.
 - Örneğin, İtalya'da hafif semptomlara sahip vakaların Almanya'ya oranla çok daha az test edilmesi ve bu nedenle istatistiklerde yer almaması.
- Bu durum sadece ağır hastaların tespitine ve dolayısıyla ölüm oranının yükselmesine neden oluyor.

- Aşağıdaki Bazı Ülkelerdeki Ölüm Oranı grafiğinde ölüm oranlarının eğrisi bulunmaktadır. Türkiye'de test sayısının yüksek olması, ülke nüfusunun genç olması ve sağlık sisteminin mevcut durumu kaldırabilmesi Türkiye'deki ölüm oranının düşük olmasını sağlamıştır.

Referans: <https://www.dw.com/tr/koronavir%C3%BCs-%C3%BCkelerde-%C3%B6l%C3%BCm-oranlar%C4%B1-neden-bu-kadar-farkl%C4%B1/a-52939908>

```
plt.figure(figsize=(20,8))

vaka_limiti = 10

d = data[(data['Country/Region'] == 'Turkey') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='red', label='Türkiye')

d = data[(data['Country/Region'] == 'Italy') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='Green', label='İtalya')

d = data[(data['Country/Region'] == 'US') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='Blue', label='Amerika')

d = data[(data['Country/Region'] == 'Spain') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='Yellow', label='İspanya')

d = data[(data['Country/Region'] == 'Iran') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='Black', label='İran')

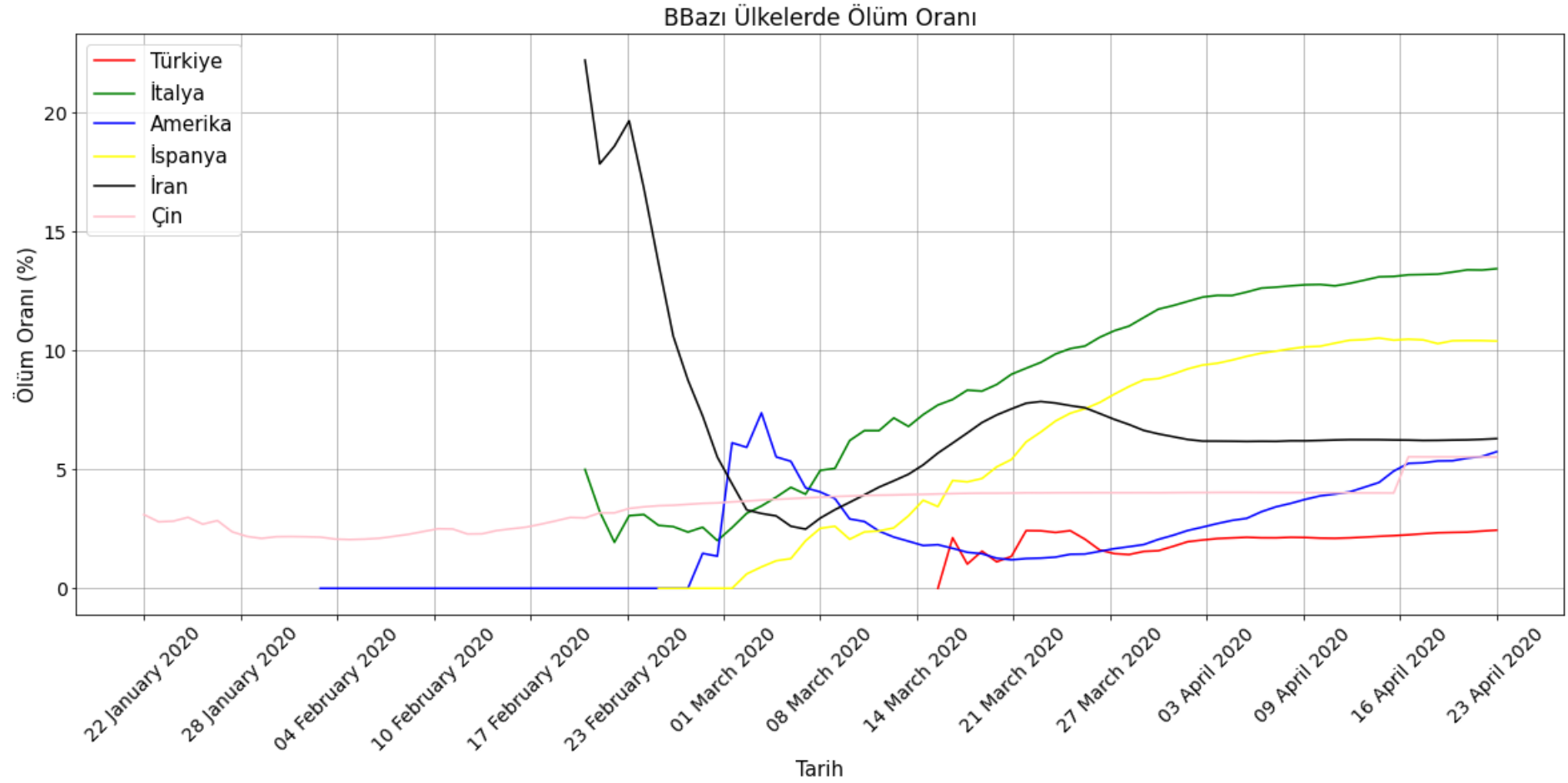
d = data[(data['Country/Region'] == 'China') & (data['Confirmed'] >= vaka_limiti)]
plt.plot(d['Date'], 100*d['Deaths']/d['Confirmed'], c='Pink', label='Çin')

plt.rc(['xtick','ytick'], labels=14)
plt.xticks(np.linspace(data['Date'].min().value,
                        data['Date'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(data['Date'].min().value,
                                      data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Ölüm Oranı (%)', fontsize=15)
plt.title('Bazı Ülkelerde Ölüm Oranı', fontsize=17)
plt.grid(color='gray', linestyle='-', linewidth=0.5)
plt.legend(fontsize=15, loc=2)
```



<matplotlib.legend.Legend at 0x7f0f45b64748>



Covid-19 Ölüm Oranı

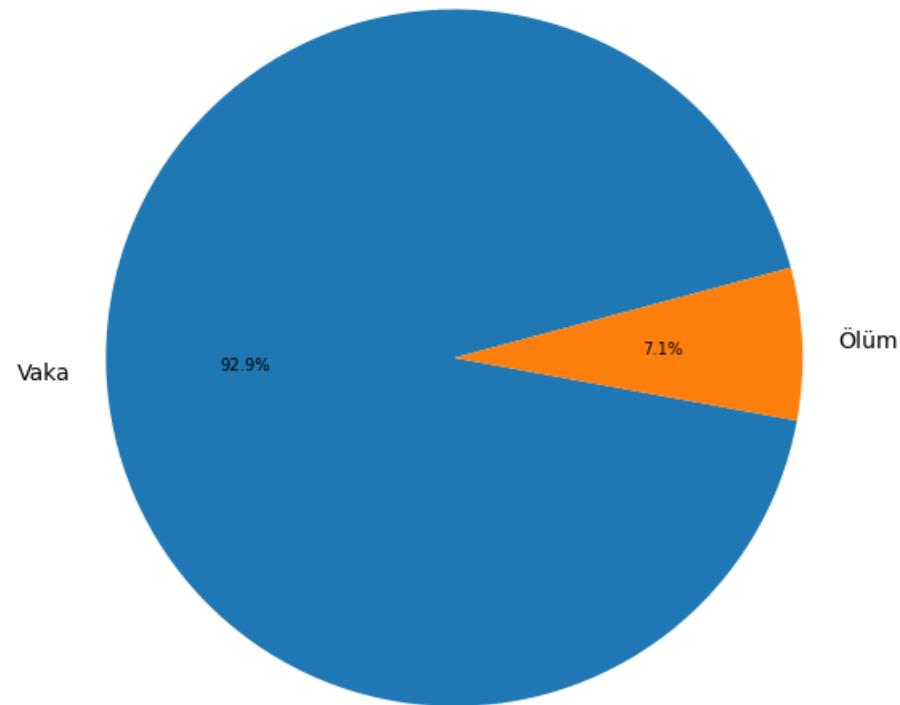
- Dünyadaki ölüm oranı pasta grafiği olarak aşağıda görünüyor.
- Yapılan bir araştırmaya göre corona virüs ölüm oranı, hafif veya asemptomatik vakalar da dahil edilince daha önceki tahminlerden daha düşük seviyeye geriliyor.

Kaynak: <https://www.ntv.com.tr/galeri/teknoloji/bilim-insanlarindan-yeni-iddia-corona-virusun-olum-orani-daha->

```
plt.figure(figsize=(10,10))
plt.rc(['xtick','ytick'], labels=14)
d = datatoplaml.groupby('Date').sum()[-1:]
plt.pie([d['Confirmed'][0]-d['Deaths'][0],d['Deaths'][0]], labels=['Vaka','Ölüm'], autopct='%1.1f%%', startangle=15)
plt.title('Dünyadaki Ölüm ve Vaka Oranları', fontsize=15)
```

Text(0.5, 1.0, 'Dünyadaki Ölüm ve Vaka Oranları')

Dünyadaki Ölüm ve Vaka Oranları



```
salginlar = {'MERS-CoV (11.12.2017-Birleşik Arap Emirlikleri)':35,'Sarı Humma (Aralık 2017-Nijerya)':21.1,
            'Difteri (Ağustos/Aralık 2017-Yemen ve Bangladeş)': 10.5, 'Kolera (Ocak/Kasım 2017-Kenya)':1.9,
            'Margburg Humması (Uganda ve Kenya)':50, 'Deng Ateşi (28.09.2017- Burkina Faso) ': 0.2}
```

""Kaynak: <http://dergipark.org.tr/tr/download/article-file/546064>"""

Text(0.5, 1.0, 'Kaynak: <http://dergipark.org.tr/tr/download/article-file/546064>')

Salgınların Ölüm Oranı

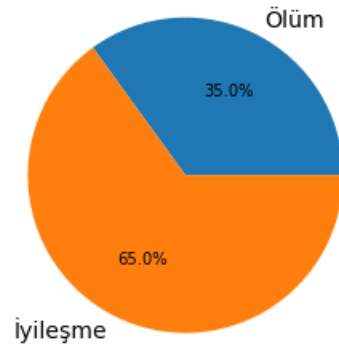
- Geçmişteki salgınlara baktığımızda, bugünkü yaşadığımız salgının öldürücülüğünün düşük olduğunu görebiliriz fakat koronavirüs salgını çok büyük kitlelere ulaşmayı başardı.
- Aşağıda bazı salgınların ölüm oranları görünüyor. Başlıklarındaki tarih ve konumlar salgının son görüldüğü tarih ve yerdir.

Kaynak: https://www.ntv.com.tr/galeri/teknoloji/bilim-insanlarindan-yeni-iddia-corona-virusun-olum-orani-daha-dusuk,8C_wBSzsH0uXDNgJ4jmYbA/Wb9LlpJW5kqF_VXIfc82dA

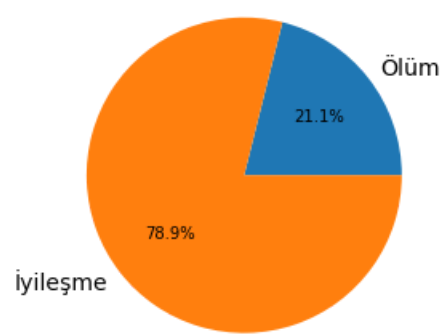
```
fig, axes= plt.subplots(2,3, figsize=(15,10))
for i, ax in enumerate(axes.flatten()):
    veri_isim = list(salgınlar.keys())
    veri_oran = list(salgınlar.values())
    ax.pie([veri_oran[i],100-veri_oran[i]], autopct="%.1f%%", labels=['Ölüm', 'İyileşme'])
    ax.set_title(veri_isim[i])
plt.show()
```



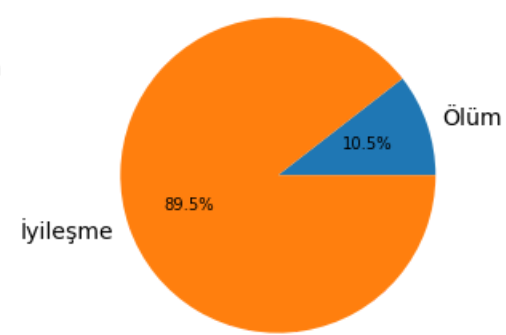
MERS-CoV (11.12.2017-Birleşik Arap Emirlikleri)



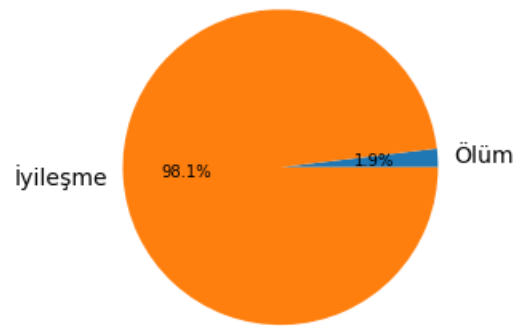
Sarı Humma (Aralık 2017-Nijerya)



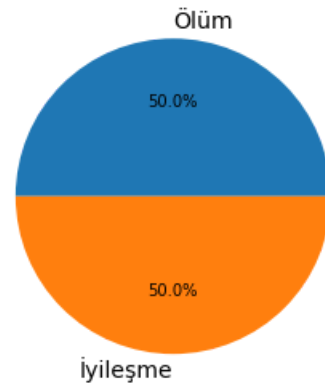
Difteri (Ağustos/Aralık 2017-Yemen ve Bangladeş)



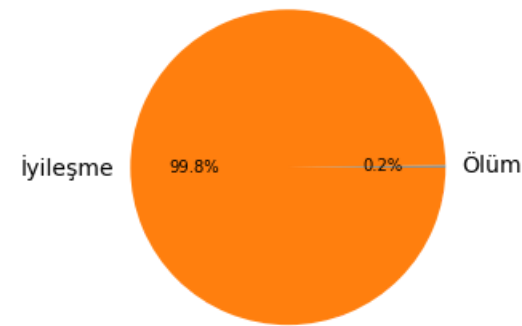
Kolera (Ocak/Kasım 2017-Kenya)



Margburg Humması (Uganda ve Kenya)



Deng Ateşi (28.09.2017- Burkina Faso)

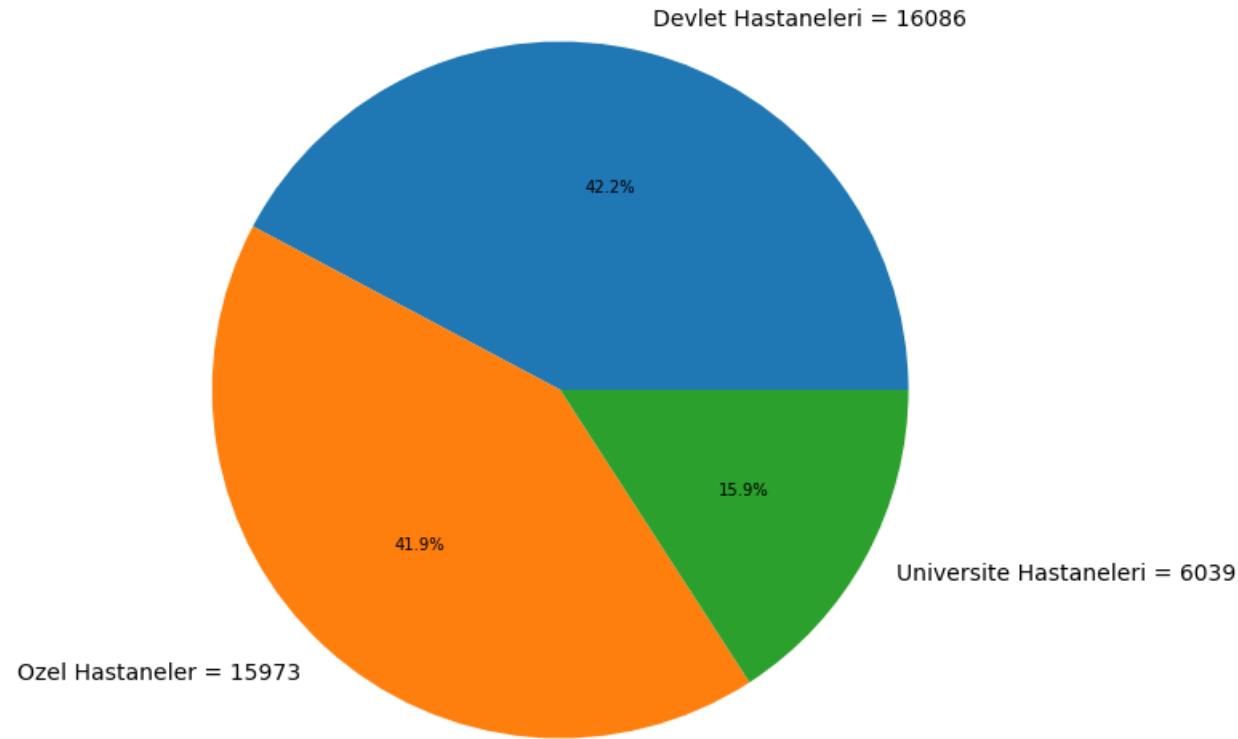


```
plt.figure(figsize=(10,10))
plt.rc(['xtick','ytick'], labels=14)
plt.pie([16086,15973,6039],labels=['Devlet Hastaneleri = 16086','Ozel Hastaneler = 15973','Universite Hastaneleri = 6039'], autopct='%1.1f%%')
plt.title("Türkiye'deki yoğun bakım yatak sayıları ", fontsize=15) #https://mail.google.com/mail/u/0/#inbox/FMfcgxwHMsSvDmDkknMWXTRDBNJCHNNv
```



```
Text(0.5, 1.0, "Türkiye'deki yoğun bakım yatak sayıları ")
```

Türkiye'deki yoğun bakım yatak sayıları



Salgın ve Ekonomi

- Tüm dünyada milyonlarca kişinin hastalanmasına neden olan koronavirüs, ekonomide de büyük hasarlara yol açtı.
- Birçok ekonomik faaliyete ara verilmek zorunda kalındı.
- Petrol'den Dolar'a birçok ekonomik yapı hasara uğradı. Amerika Merkez Bankası bu dönemde karşılıksız para dahi bastı piyasayı canlı tutmak için. Kısa vadede yangını durdurmak için etkili olsa da uzun vadede Dolar'ın büyük darbeler alacağını öngörebiliriz.
- Türkiye'de Dolar'ın yükselmesinin sebeplerinden bir tanesi de ülkeye para girdi çıktısı olmaması ve Türk Lirası'nın Dolar karşısında da değer kaybetmesinde yol açtı

- Büyük Türk şirketlerinin başta THY olmak üzere ekonomik çıkmaza girebileceği hatta bazı şirketleri iflasa kadar sürükleyebileceği ihtimali doğuyor
- Küçük esnafların, günlük işlerde çalışan insanların aile içi ekonomilerine büyük etkilerde bulunabileceği hatta koronavirüs sonrası işsizlik oranında çok büyük artışlar gözlemlenebileceği öngörülebilir.

```
data_doviz = pd.read_csv('https://raw.githubusercontent.com/kynemre/Covid_19/master/USD_TRY%20Ge%C3%A7mi%C5%9F%20Verileri.csv')
```

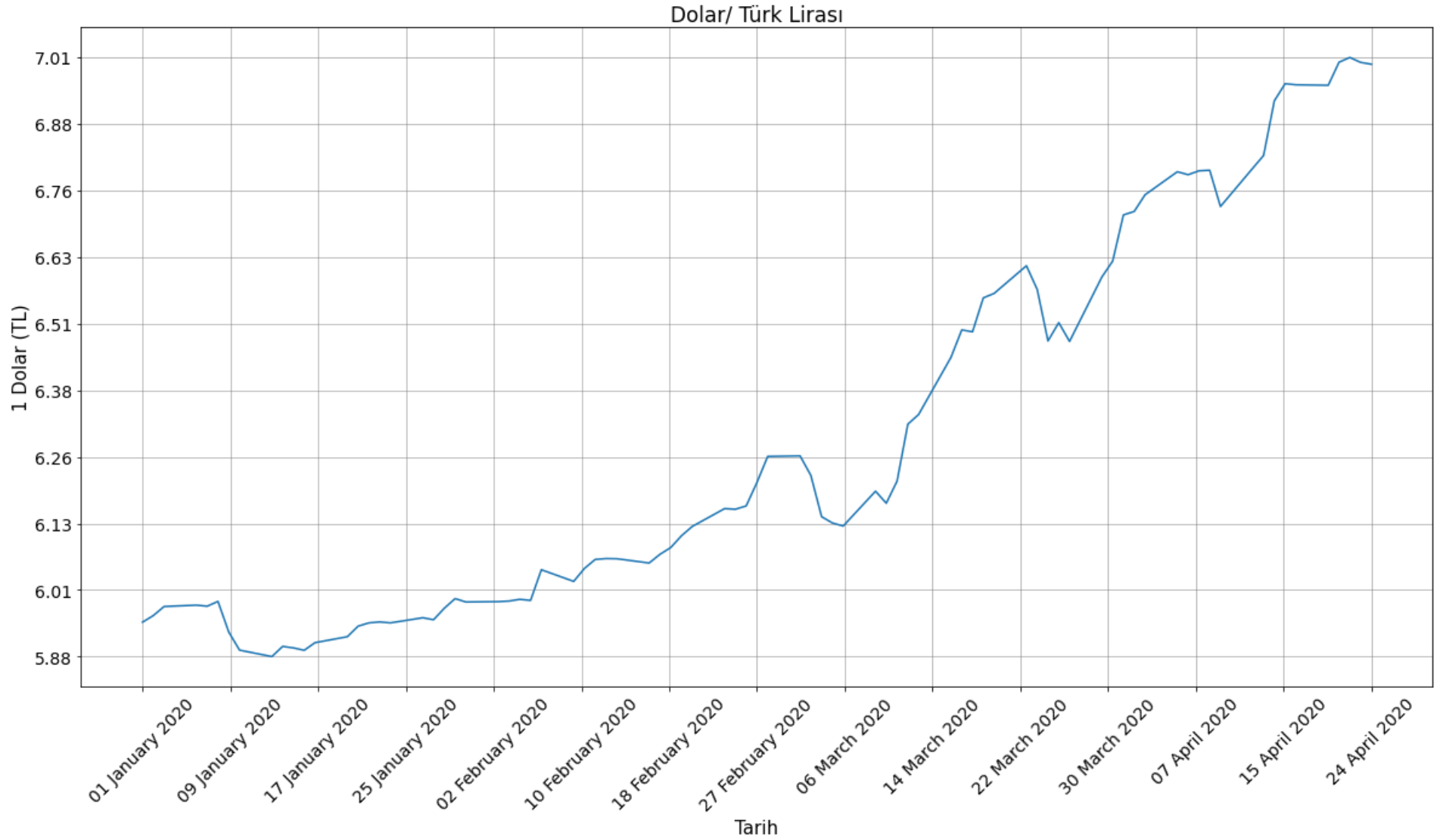
```
def to_float(data, kolon):
    data[kolon] = data[kolon].str.strip('%')
    return data[kolon].str.replace(',', '.').astype('float32')
```

```
data_doviz['Şimdi'] = to_float(data_doviz, 'Şimdi')
data_doviz['Açılış'] = to_float(data_doviz, 'Açılış')
data_doviz['Yüksek'] = to_float(data_doviz, 'Yüksek')
data_doviz['Düşük'] = to_float(data_doviz, 'Düşük')
data_doviz['Fark %'] = to_float(data_doviz, 'Fark %')
data_doviz['Tarih'] = data_doviz['Tarih'].str.replace(r'(\d+).(\d+).(\d+)', r'\3-\2-\1')
data_doviz['Tarih'] = pd.to_datetime(data_doviz['Tarih'])
```

```
plt.figure(figsize=(20,10))
plt.plot(data_doviz['Tarih'], data_doviz['Yüksek'])
```

```
plt.yticks(np.linspace(min(data_doviz['Yüksek']), max(data_doviz['Yüksek']), 10), np.round(np.linspace(min(data_doviz['Yüksek']), max(data_doviz['Yüksek']), 10), 2))
plt.xticks(np.linspace(data_doviz['Tarih'].min().value,
                        data_doviz['Tarih'].max().value, 15).astype('datetime64[ns]'),
            pd.to_datetime(np.linspace(data_doviz['Tarih'].min().value,
                                       data_doviz['Tarih'].max().value, 15)).strftime('%d %B %Y'), rotation=45)
plt.rc(['xtick', 'ytick'], labels=14)
plt.xlabel('Tarih', fontsize=15)
plt.ylabel('1 Dolar (TL)', fontsize=15)
plt.title('Dolar/ Türk Lirası', fontsize=17)
plt.grid(color='gray', linestyle='-', linewidth=0.5)
plt.tick_params(axis='x', rotation = 45)
```





Aylara Kadar Gerçekleşen Uçak Trafikçi ve Ziyaretçi Sayıları

•Türkiye'deki uçak trafiğinin Mart aylarında önceki yıla göre düşüş gösterdiğini görüyoruz. Bazı ülkelere yapılan seyahat sınırlamaları ve yapılan seyahat uyarılarının bu konuda etkili olduğunu görüyoruz.

•Ülkemize gelen turist sayılarında -İran hariç- büyük düşüşler bulunmakta. Bunun ülkemize turizm ekonomisi olarak büyük etkilerinin olabileceğini söyleyenebilir. Özellikle turizm ekonomisi ile geçinen yerlerde büyük sorunlara yol açacaktır.

•İran'dan ülkemizde gelen ziyaretçi sayısının önceki yıllara göre daha az düşüş yaşamasını iki nedeni olabilir.

1. İran 2019 yılında yaşadığı iç ve dış sorunlarından dolayı ziyaretçi Türkiye'ye gelen ziyaretçi sayısının çok düşük olması.
2. Salgının başlangıcından kısa bir süre sonra İran'da vaka ve ölüm sayısında artış olması ve virüsten kaçmak isteyen vatandaşların

Türkiye'ye gelme verileri

```
data_ucus_ocak = pd.read_excel('https://raw.githubusercontent.com/kynemre/Covid_19/master/Ucus_Verileri/Ocak.xlsx', header= 2)
data_ucus_subat = pd.read_excel('https://raw.githubusercontent.com/kynemre/Covid_19/master/Ucus_Verileri/%C5%9Eubat.xlsx', header= 2)
data_ucus_mart = pd.read_excel('https://raw.githubusercontent.com/kynemre/Covid_19/master/Ucus_Verileri/Mart.xlsx', header= 2)
```

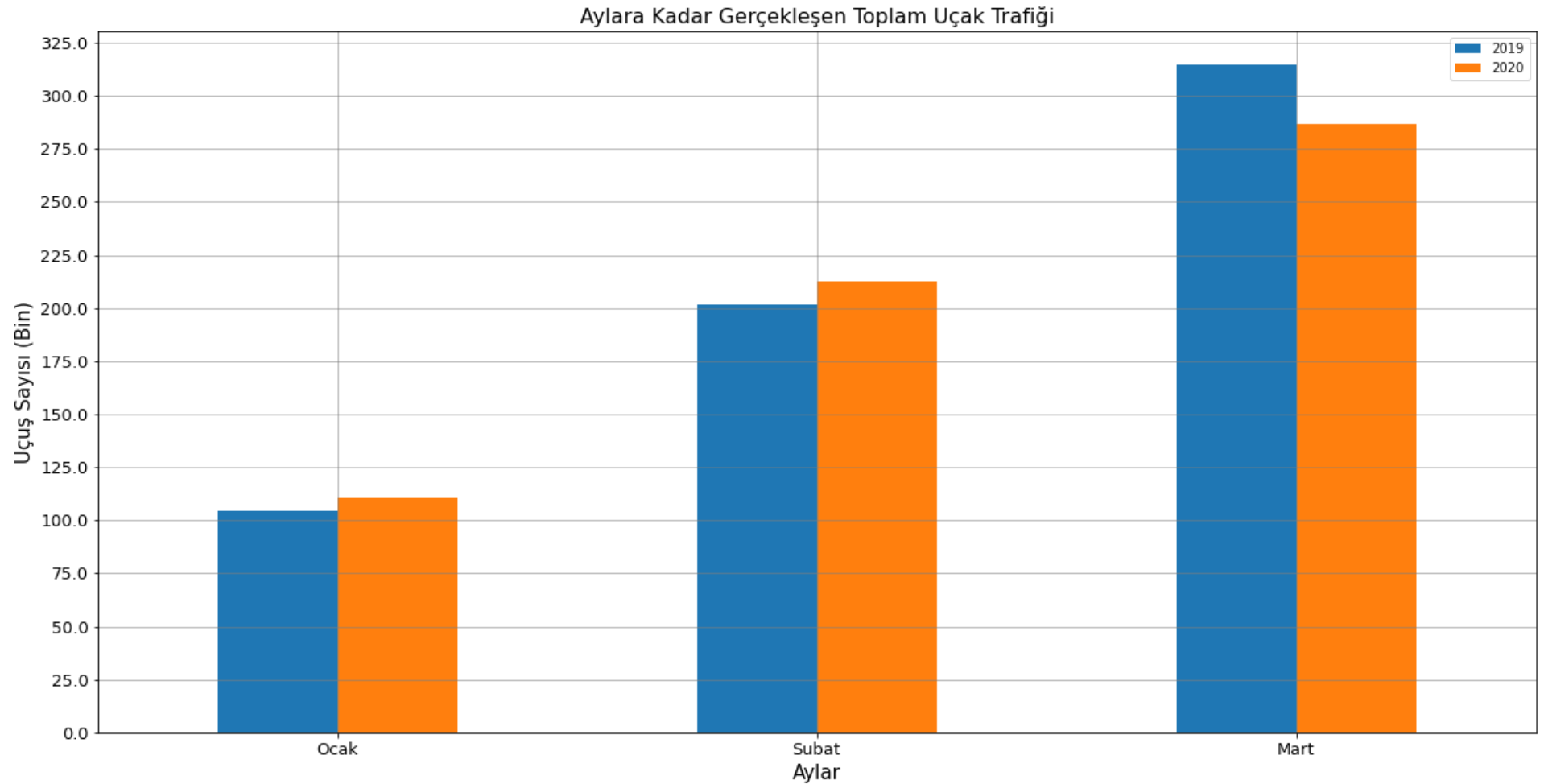
```
data_ucus_ocak.rename(columns={'Unnamed: 0': 'Havalimanı'}, inplace=True)
data_ucus_subat.rename(columns={'Unnamed: 0': 'Havalimanı'}, inplace=True)
data_ucus_mart.rename(columns={'Unnamed: 0': 'Havalimanı'}, inplace=True)
```

```
aylar = ['Ocak', 'Subat', 'Mart']
veriler = [[data_ucus_ocak[data_ucus_ocak['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam'],
            data_ucus_ocak[data_ucus_ocak['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam.1']],
            [data_ucus_subat[data_ucus_subat['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam'],
            data_ucus_subat[data_ucus_subat['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam.1']],
            [data_ucus_mart[data_ucus_mart['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam'],
            data_ucus_mart[data_ucus_mart['Havalimanı'] == 'TÜRKİYE GENELİ'].iloc[0]['Toplam.1']]]
veriler = pd.DataFrame(veriler)
veriler.columns = [2019,2020]
veriler.index = aylar
```

```
veriler.plot(kind='bar', fontsize=13, figsize=(20,10))
plt.yticks(np.arange(0, veriler[[2019,2020]].max().max()+25000, 25000),np.arange(0, veriler[[2019,2020]].max().max()+25000, 25000)//1000)
plt.xlabel('Aylar',fontsize=15)
plt.ylabel('Uçuş Sayısı (Bin)', fontsize=15)
plt.title("Aylara Kadar Gerçekleşen Toplam Uçak Trafiği", fontsize=16)
plt.tick_params(axis='x', rotation = 0)
plt.grid(color='gray', linestyle='-', linewidth=0.5)
plt.legend()
```



<matplotlib.legend.Legend at 0x7f0f4bea41d0>



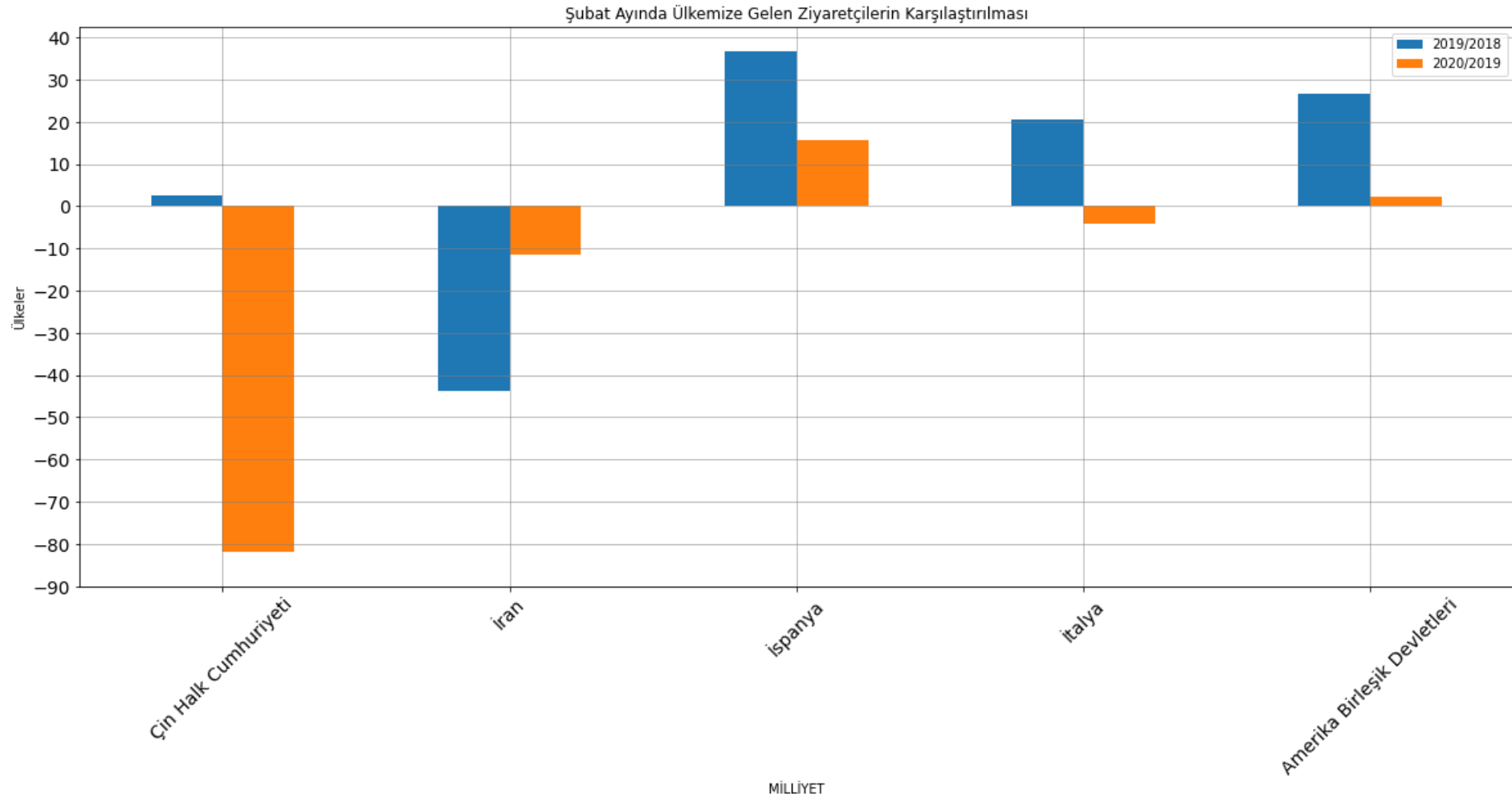
```
data_turizm = pd.read_excel('https://raw.githubusercontent.com/kynemre/Covid_19/master/Turizm_Data_Subat.xls', sheet_name= 'Milliyet', header=2)
```

```
data_turizm.set_index('MİLLİYET', inplace = True)
```

```
data_tur = data_turizm[(data_turizm.index == 'Çin Halk Cumhuriyeti') | (data_turizm.index == 'İtalya')
| (data_turizm.index == 'Amerika Birleşik Devletleri') | (data_turizm.index == 'İran')
| (data_turizm.index == 'İspanya')][['2019/2018', '2020/2019']].copy()
```

```
data_tur.plot(kind='bar', width = 0.5, figsize=(20,8))
```

```
plt.yticks(np.arange((data_tur.min().min()/10)*10, data_tur.max().max()+10, 10))
plt.title('Şubat Ayında Ülkemize Gelen Ziyaretçilerin Karşılaştırılması')
plt.ylabel('Yılların Yüzdelik Farkı')
plt.ylabel('Ülkeler')
plt.tick_params(axis='x', rotation=45)
plt.grid(color='gray', linestyle='-', linewidth=0.5)
```



Geçmiş Verilerden Tahminlerde Bulunulması

• Salgınların seyri genel olarak birbirine benziyor. İlk vakadan sonra, kısa bir süreliğine yavaş bir yükseliş gerçekleşiyor. Salgının orta zamanlarında hızlı bir artışla günlük vaka ve ölüm sayıları pik değerine ulaşır plato durumuna geçiyorlar. Ardından günlük vaka ve ölüm sayıları yok denecek kadar azalıyor.

• Aşağıdaki grafiklerde salgını hastırmayı hasaran Çin'e bakarak bir tahmin grafiği çizdirdik

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
```

#Doğrusal modelleme

```
lr = LinearRegression()#obje oluşturma
```

#özellikleri ve özelliklerle kestireceğimiz hedeflenen değeri belirleyelim

```
y = data[data['Country/Region'] == 'China']['Deaths']#hedef, series
```

```
x = data[data['Country/Region'] == 'China']['Confirmed']#kullanılacak özellik, dataframe
```

```
lr.fit(x,y)
```

```
>>> LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Linear regresyonun hata payı fazla olduğu için işlemleri polinom regresyon ile yapıyoruz.

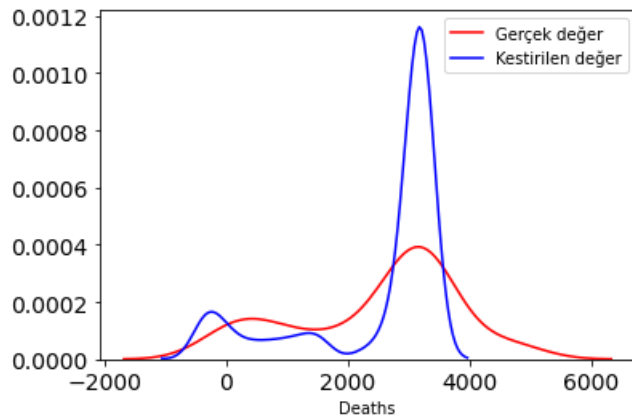
```
y_kestirilen = lr.predict(x)#modelle özellikten kestirim yapılması
```

#Gerçek ve kestirilen değerlerin dağılımlarının çizimi

```
ax1 = sns.distplot(y,hist=False,color='r',label='Gerçek değer')
```

```
sns.distplot(y_kestirilen,hist=False,color='b',label='Kestirilen değer',ax=ax1)
```

```
>>> <matplotlib.axes._subplots.AxesSubplot at 0x7f0f456a6240>
```



```
plt.figure(figsize=(20,8))
```

```
plt.plot(data[data['Country/Region'] == 'China']['Date'], y, c='red', label='Gerçekleşen Ölüm Sayısı')
```

```
plt.plot(data[data['Country/Region'] == 'China']['Date'], y_kestirilen, c='blue', label='Tahmin Edilen Ölüm Sayısı')
```

```
plt.plot(data[data['Country/Region'] == 'China']['Date'], y_restrictions, c='blue', label='Tartışılan Ülken Ölüm Sayısı',  
         linestyle='dashed', linewidth=0.5)  
  
plt.xticks(np.linspace(0, max(data[data['Country/Region'] == 'China']['Deaths']), 15),  
           np.round(np.linspace(0, max(data[data['Country/Region'] == 'China']['Deaths']), 15)/1000,2))  
plt.xticks(np.linspace(data['Date'].min().value,  
                        data['Date'].max().value, 15).astype('datetime64[ns]'),  
           pd.to_datetime(np.linspace(data['Date'].min().value,  
                                       data['Date'].max().value, 15)).strftime('%d %B %Y'), rotation=45)  
  
plt.xlabel('Tarih', fontsize=15)  
plt.ylabel('Ölüm Sayısı (Bin)', fontsize=15)  
plt.title('Dünyadaki Toplam Ölüm Değerleri', fontsize=17)  
  
plt.grid(linestyle='dashed', color='gray', linewidth=0.5)  
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f456e0588>

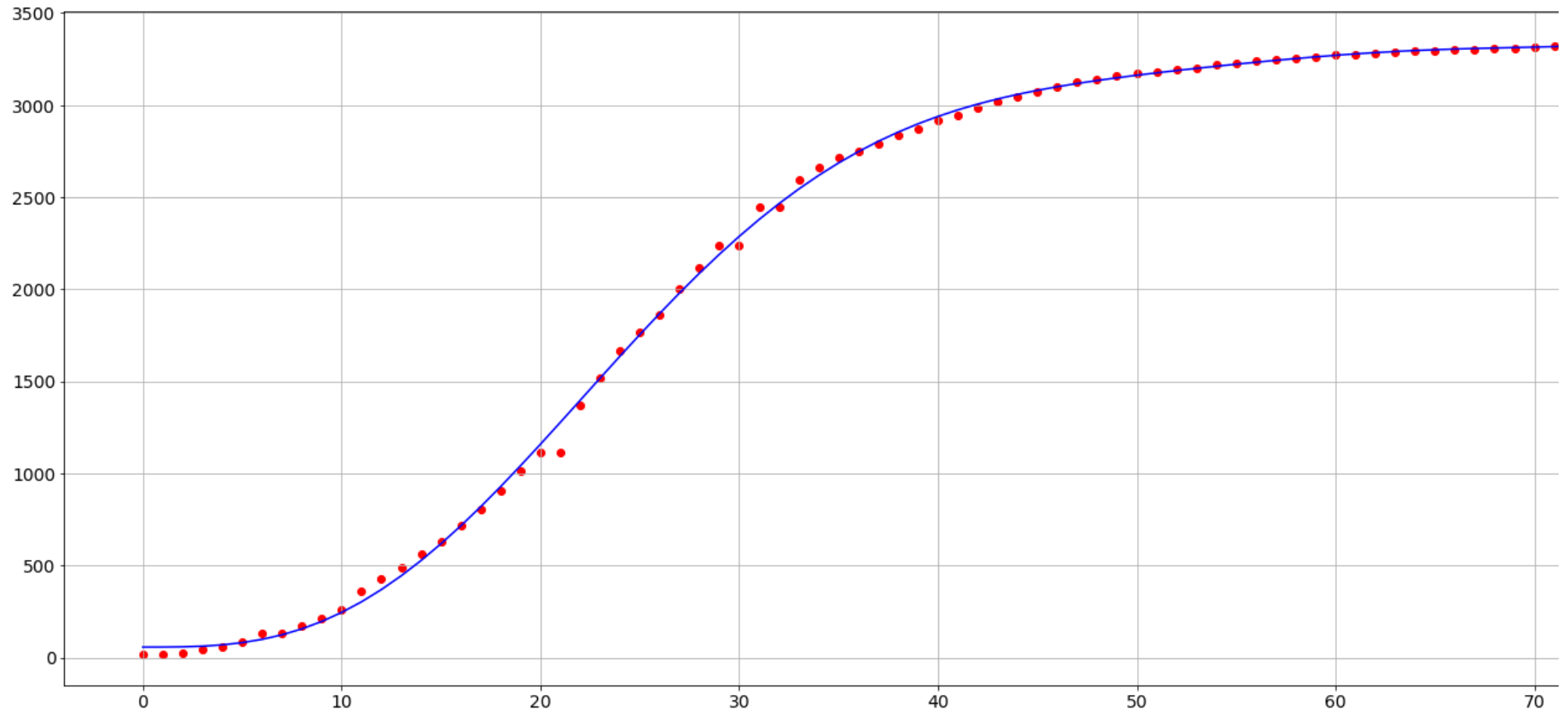
```
X = df[df['Country/Region'] == 'China']['Date'].iloc[:80].values-52
y = data[data['Country/Region'] == 'China']['Deaths'].iloc[:80].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

```
lr = LinearRegression()
poly = PolynomialFeatures(degree=9)
```

```
egitim_poly = poly.fit_transform(X)
lr.fit(egitim_poly, y)
predict = lr.predict(egitim_poly)
```

```
plt.figure(figsize=(25,10))
plt.scatter(X, y, color='red')
plt.plot(X, predict, color='blue')
plt.grid()
plt.show()
```

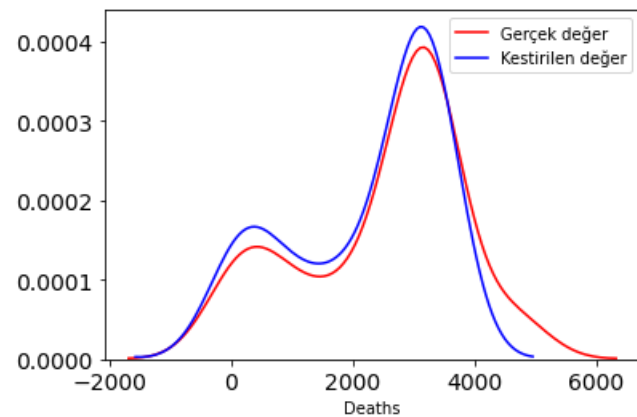




```
ax1 = sns.distplot(data[data['Country/Region'] == 'China']['Deaths'],hist=False,color='r',label='Gerçek değer')  
sns.distplot(predict,hist=False,color='b',label='Kestirilen değer',ax=ax1)
```



<matplotlib.axes._subplots.AxesSubplot at 0x7f0f455bed30>



```
plt.figure(figsize=(25,10))

plt.plot(data[data['Country/Region'] == 'China']['Date'][:80], y, c='red', label='Gerçekleşen Ölüm Sayısı ')
plt.plot(data[data['Country/Region'] == 'China']['Date'][:80], predict, c='blue', label='Tahmin Edilen Ölüm Sayısı')

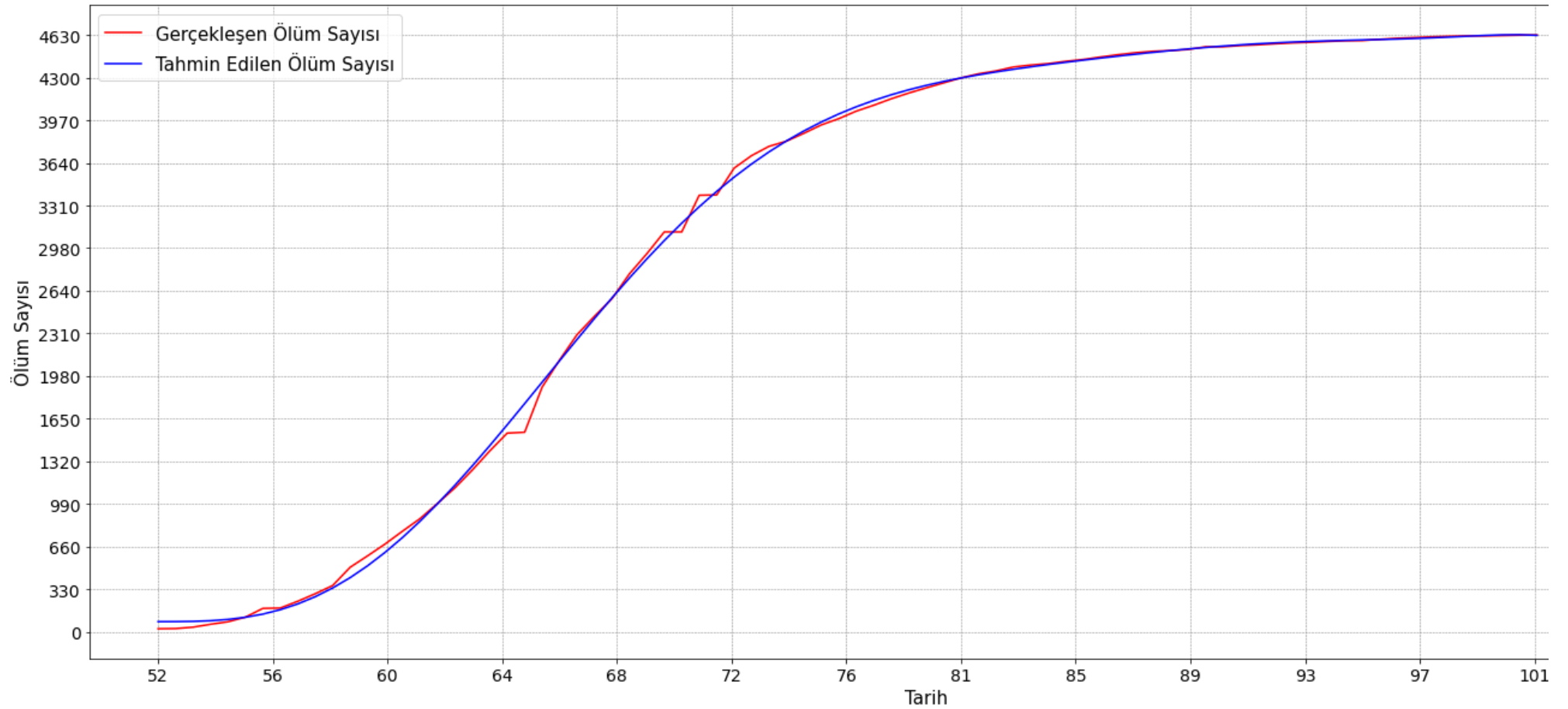
plt.yticks(np.linspace(0, max(y), 15),
            ((np.linspace(0, max(data[data['Country/Region'] == 'China']['Deaths']), 15)//10)*10).astype('int'))
plt.xticks(np.linspace(data['Date'].min().value,
                        data['Date'].max().value, 15).astype('datetime64[ns]'),
            np.linspace(52, (data['Date'][:80].max() - data['Date'].min()).days+52, 20).astype('int'))

plt.xlabel('Tarih', fontsize=15)
plt.ylabel('Ölüm Sayısı', fontsize=15)
plt.title('gh', fontsize=1)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```

↩

<matplotlib.legend.Legend at 0x7f0f454fea58>



```
tahmin = lr.predict(poly.fit_transform(np.arange(80).reshape(-1,1)))
```

Geçmiş Verilerden Tahminlerde Bulunulması

- Türkiye'deki tahmin edilen ve şu anki ölüm eğrisi aşağıdaki gibidir.
- Grafiğin şu anki halinde tahminler ve gerçek değerler oldukça uyumlu. Günlük vaka ve ölüm sayılarına bakılırsa da ölüm artışının bu seyrinde gideceği öngörülebilir.

```
plt.figure(figsize=(25,10))
```

```
d = data[(data['Country/Region'] == 'Turkey') & (data['Deaths'] > 17)][['Deaths']]
plt.plot(pd.date_range(data[data['Country/Region'] == 'China']['Date'].min(),
                      periods=len(d)), d, marker='o', c='red', label="Türkiye Ölüm Verisinin Eğrisi")
plt.plot(data[data['Country/Region'] == 'China']['Date'][:80], tahmin, c='Black', label='Tahmin')

plt.yticks(np.linspace(0, max(y), 15),
           ((np.linspace(0, max(data[data['Country/Region'] == 'China']['Deaths']), 15)//10)*10).astype('int'))
plt.xticks(np.linspace(data['Date'].min().value,
                      data['Date'].max().value, 15).astype('datetime64[ns]'),
           np.linspace(11, (data['Date'].max() - data['Date'].min()).days+52, 20).astype('int'))

plt.xlabel('Salgının Başlangıcından Geçen Gün', fontsize=15)
plt.ylabel('Ölüm Sayısı', fontsize=15)
plt.title('Türkiyede Ölüm Eğrisi ve Ölüm Eğrisi Tahmini', fontsize=1)

plt.grid(linestyle='dashed' ,color='gray', linewidth=0.5)
plt.legend(fontsize=15)
```



<matplotlib.legend.Legend at 0x7f0f4541bc18>

