UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

SEMINARSKA NALOGA

**POVEZOVANJE COVID-19 PODATKOVNIH ZBIRK**

Miha Bogataj in Gašper Štepec

Ljubljana, 2021

**KAZALO**

[1 UVOD 4](#_Toc61086528)

[2 TEHNIČNI PODATKI 5](#_Toc61086529)

[2.1 PostgreSQL 5](#_Toc61086530)

[2.2 Python 5](#_Toc61086531)

[2.3 Podatkovne zbirke 5](#_Toc61086532)

[2.3.1 Amazon AWS 5](#_Toc61086533)

[2.3.2 OECD 5](#_Toc61086534)

[3 PODAtkovna baza 6](#_Toc61086535)

[3.1 Konceptualni model 6](#_Toc61086536)

[3.2 Fizični model 6](#_Toc61086537)

[4 Vstavljanje podatkov v bazo 7](#_Toc61086538)

[4.1 Vzpostavitev povezave z bazo 7](#_Toc61086539)

[4.2 Continent 7](#_Toc61086540)

[4.3 Country 8](#_Toc61086541)

[4.4 Severitiesprediction 9](#_Toc61086542)

[4.5 Unemployment 10](#_Toc61086543)

[4.6 AgeGroup 11](#_Toc61086544)

[4.7 States Measures 12](#_Toc61086545)

[4.8 ByGender 14](#_Toc61086546)

[4.9 Okrožje 15](#_Toc61086547)

[4.10 Hospital 16](#_Toc61086548)

[4.11 Covid19 18](#_Toc61086549)

[4.11.1 Države 18](#_Toc61086550)

[4.11.2 Okrožja 20](#_Toc61086551)

[4.11.3 New York in San Francisco 21](#_Toc61086552)

[4.12 ApprovalRating 23](#_Toc61086553)

[5 PoIZVEDBE 24](#_Toc61086554)

[5.1 Število potrjenih okuženih glede na milijon prebivalcev po kontinentih 24](#_Toc61086555)

[5.2 Države z največjim številom potrjenih okuženih na milijon prebivalcev 26](#_Toc61086556)

[5.3 Države z najmanjšim številom potrjenih okuženih na milijon prebivalcev 27](#_Toc61086557)

[5.4 Primerjava med ZDA, New Yorkom ter San Franciscom 13. 5. 2020 28](#_Toc61086558)

[5.5 Delež starostnih skupin med COVID-19 smrtmi 30](#_Toc61086559)

[5.6 Delež smrti zaradi COVID-19 po starostnih skupinah 31](#_Toc61086560)

[5.7 Okrožja z največjim številom bolnišnic 33](#_Toc61086561)

[5.8 Delež bolnišnic po rangu napovedi za naslednjih 7 dni 34](#_Toc61086562)

[5.9 Prikaz podatkov o okužbah, smrtih ter testiranju v Sloveniji 35](#_Toc61086563)

[5.10 Primerjava novih okužb in smrti v Sloveniji s stopnjo brezposelnosti po mesecih 36](#_Toc61086564)

[5.11 Primerjava novih okužb in smrti v ZDA s stopnjo brezposelnosti ter podporo Donaldu Trumpu po mesecih 38](#_Toc61086565)

[6 Zaključek 40](#_Toc61086566)

[7 VIRI 41](#_Toc61086567)

**KAZALO TABEL**

[Tabela 1: Število okuženih in mrtvih po kontinentih na milijon prebivalcev 25](#_Toc61052060)

[Tabela 2: Države z največjim številom okuženih glede na milijon prebivalcev 26](#_Toc61052061)

[Tabela 3: Države z najmanjšim številom okuženih glede na milijon prebivalcev 27](#_Toc61052062)

[Tabela 4: Primerjava ZDA, New York, San Francisco 13. 5. 2020 29](#_Toc61052063)

[Tabela 5: Delež starostnih skupin med COVID-19 smrtmi 30](#_Toc61052064)

[Tabela 6: Delež smrti zaradi COVID-19 po starostnih skupinah 31](#_Toc61052065)

[Tabela 7: Countyji z največjim številom bolnišnic 33](#_Toc61052066)

[Tabela 8: Prikaz podatkov o okužbah, smrtih ter testiranju v Sloveniji 35](#_Toc61052067)

[Tabela 9: Primerjava novih okužb in smrti s stopnjo brezposelnosti po mesecih v Sloveniji 37](#_Toc61052068)

[Tabela 10: Primerjava novih okužb in smrti v ZDA s stopnjo brezposelnosti ter podporo Donaldu Trumpu po mesecih 38](#_Toc61052069)

**KAZALO GRAFOV**

[Graf 1: Število okuženih po kontinentih na milijon prebivalcev 25](#_Toc61052082)

[Graf 2: Število mrtvih po kontinentih na milijon prebivalcev 25](#_Toc61052083)

[Graf 3: Države z največjim številom okuženih glede na milijon prebivalcev 26](#_Toc61052084)

[Graf 4: Države z najmanjšim številom okuženih glede na milijon prebivalcev 27](#_Toc61052085)

[Graf 5: Delež starostnih skupin med COVID-19 smrtmi 32](#_Toc61052086)

[Graf 6: Delež smrti zaradi COVID-19 po starostnih skupinah 32](#_Toc61052087)

[Graf 7: Delež bolnišnic po rangu napovedi za naslednjih 7 dni 34](#_Toc61052088)

[Graf 8: Prikaz podatkov o novih okužbah in smrtih 35](#_Toc61052089)

[Graf 9: Prikaz podatkov o številu opravljenih ter potrjenih testov ter stopnji potrditve 36](#_Toc61052090)

[Graf 10: Primerjava novih okužb s stopnjo brezposelnosti po mesecih v Sloveniji 37](#_Toc61052091)

[Graf 11: Primerjava novih okužb s stopnjo brezposelnosti v ZDA 39](#_Toc61052092)

[Graf 12: Primerjava novih okužb s podporo Donaldu Trumpu 39](#_Toc61052093)

# UVOD

Konec leta 2019 se je pojavila nova bolezen, ki se je v nekaj mesecih razširala po celem svetu imenovana SARS-COVID-19. Na COVID-19 so se države različno odzvale, npr. ZDA veliko bolj sproščeno kot pa večina evropskih držav. Prav tako so se različno odzvale države znotraj ZDA zato želimo pogledati razlike med njimi. Pregledali bomo tudi podatke dveh ameriških mest za »prvi val«, natančneje za New York ter San Francisco.

Domnevamo, da COVID-19 slabše vpliva na starejše ljudi, zato bomo pogledali porazdelitev smrti zarad COVID-19 po starostnih skupinah ter primerjali število smrti s skupnim številom smrti v vsaki starostni skupini.

Na število smrtnih žrtev med epidemijami močno vplivajo bolnišnice. Zanima nas katera okrožja imajo največje število bolnišnic in kako so bolnišnice rangirane glede na stopnjo napovedi resnosti situacije glede epidemije.

COVID-19 ni vplival le na zdravje ljudi ampak zaradi strožjih ali milejših ukrepov tudi na gospodarstvo. Obdelali bomo podatke za Slovenijo ter ZDA. Pri Sloveniji nas zanima celoten potek epidemije glede na število okuženih, smrti, opravljenih testiranj in podobno. Nato želimo preveriti korelacijo med številom okuženih oseb s procentom brezposelnih ljudi v Sloveniji ter ZDA. Vpliv COVID-19 bomo primerjali tudi s podporo zdaj že skoraj nekdanjemu predsedniku ZDA Donaldu Trumpu.

# TEHNIČNI PODATKI

## PostgreSQL

Za SUPB sva izbrala PostgreSQL, ki je prosto dostopen in nama osebno poznan. PostgreSQL, velikokrat imenovan tudi Postgres, je relacijski SUPB, ki se drži pravil SQL.

## Python

Za izvajanje polnjenja in transformacijskih korakov sva izbrala programski jezik Python. Python je izredno popularen objektno orientiran jezik, ki omogoča enostavno povezavo s podatkovno bazo in polnjenje le-te.

## Podatkovne zbirke

Za vir podatkovnih zbirk sva si izbrala Amazonov AWS ter OECD.

### Amazon AWS

* COVID-19 - World Confirmed Cases, Deaths, and Testing
* USA Hospital Beds - COVID-19 | Definitive Healthcare
* Coronavirus (COVID-19) Data Hub
* COVID-19 United States Reopen and Shut Down Status by State | NY Times
* Donald Trump's Approval Ratings | FiveThirtyEight
* NYC Coronavirus (COVID-19) Data | New York City Department of Health
* COVID-19 Prediction Models Counties & Hospitals | Yu Group (UC Berkeley)
* San Francisco Bay Area COVID-19 Tracker
* Provisional COVID-19 Death Counts by Sex, Age, and State | CDC / NCHS

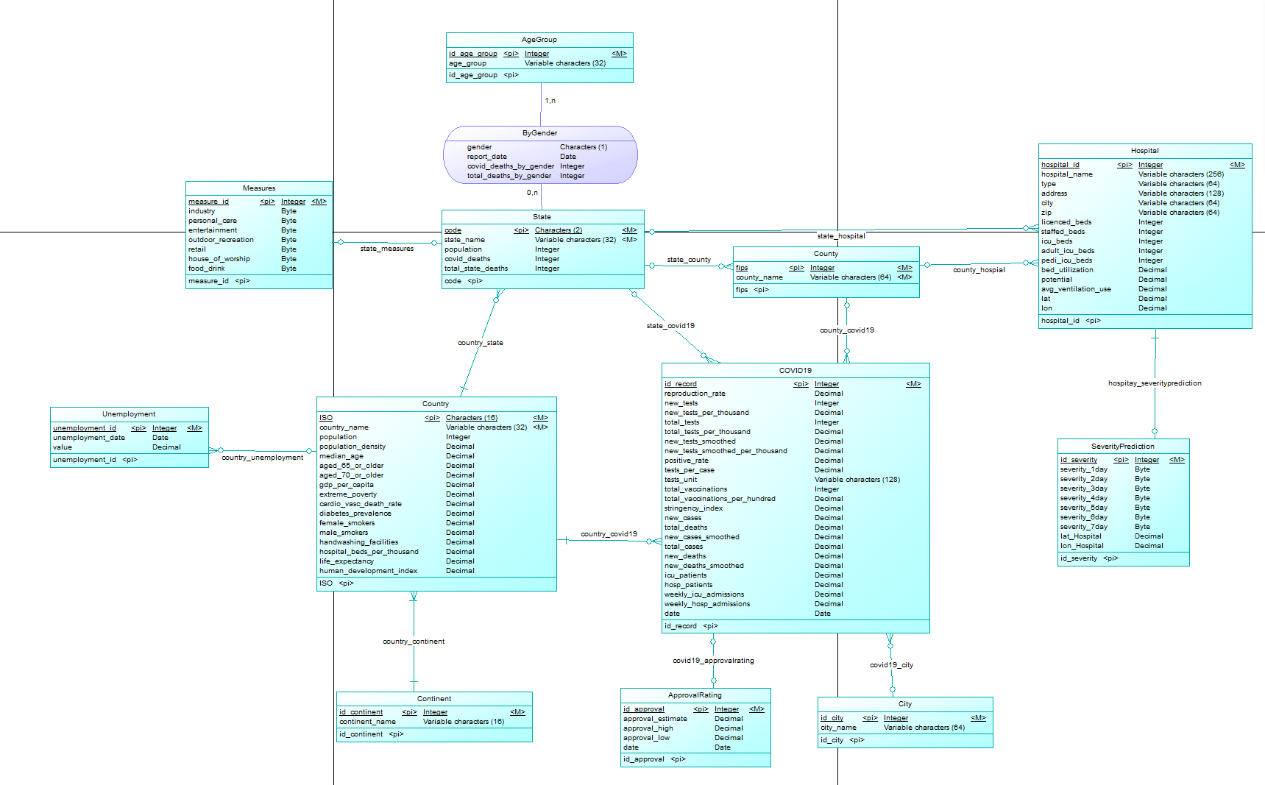
### OECD

* Short-Term Labour Market Statistics

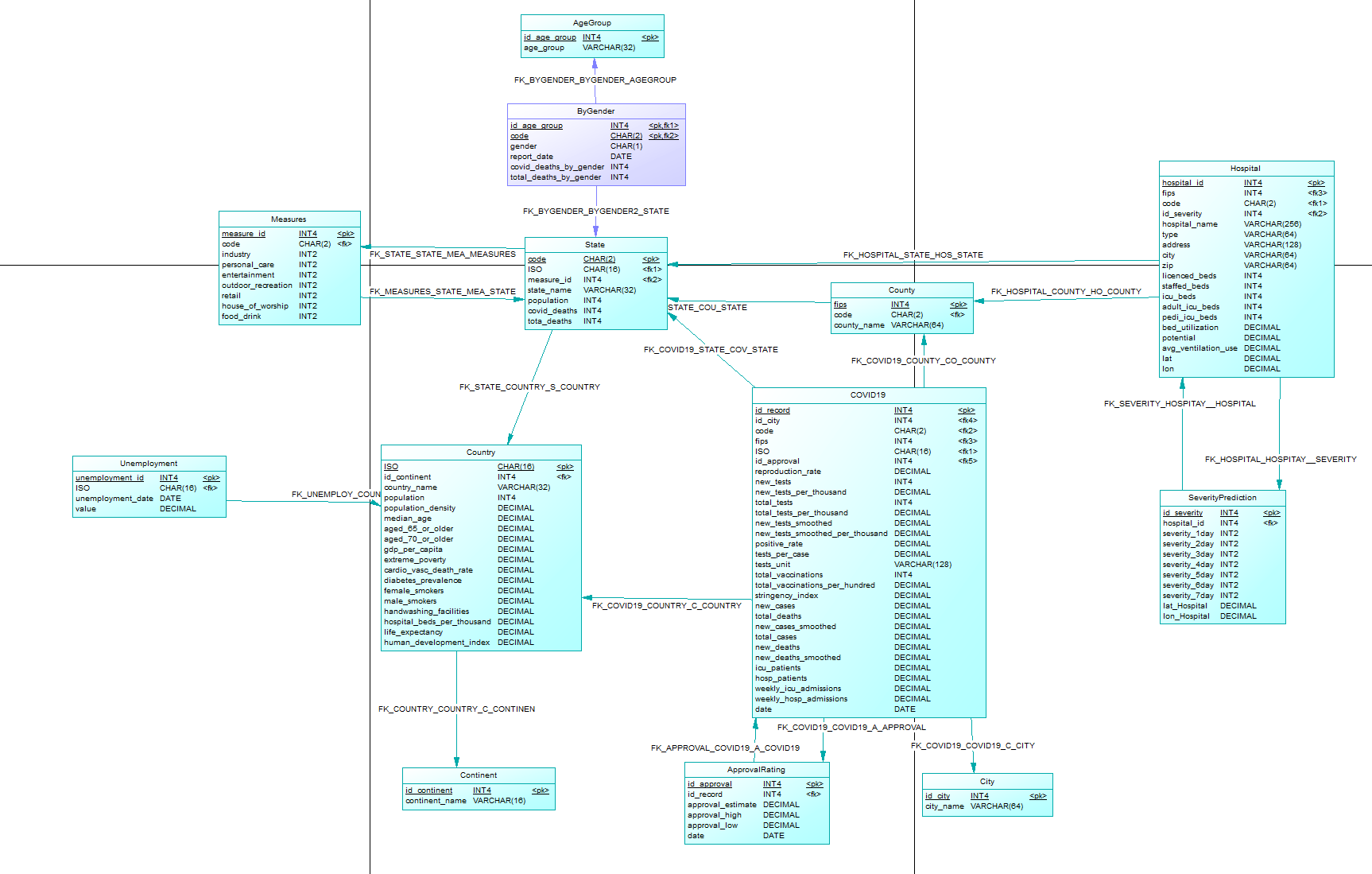
# PODAtkovna baza

## Konceptualni model

Konceptualni model vsebuje 13 entitet in 1 šibki entitetni tip.



## Fizični model



# Povezovanje podatkovnih zbirk

Tabele povezujeva na dva načina: po datumih ter po unikatnih identifikatorjih.

## Identifikatorji

Pri državah je identifikator ISO številka. Pri posameznem podatku o stopnji brezposelnosti, ukrepih, starostni skupini, vpisu o COVID-19, napovedi za 7 dni ter podpori Trumpu je identifikator zaporedna številka vnosa v bazo. Zvezna država ima za identifikator dvočrkovno kratico. Kontinente sva zapisala statično v kodo identifikator pa predstavlja kar zaporedna številka.

## Datumi

Po datumih povezujeva večino podatkov o državah, zveznih državah, bolnišnicah ter COVID-19 podatkov. Bolnišnice in napovedi so povezane glede na id\_severity ter GPS koordinate bolnišnice. Bolnišnica je na county povezana z enolično določeno kodo za county imenovano FIPS. County je povezan z zvezno državo s podatkov o kratici zvezne države, ki ga ima v tabeli tako kot tabela byGender. AgeGroup je na byGender povezana preko identifikatorja starostne skupine katere sva enolično določila sama kar je vidno v programski kodi.

# Vstavljanje podatkov v bazo

## Vzpostavitev povezave z bazo

Za vzpostavitev povezave je potrebna prvo nastavitev ODBC (Open Database Connectivity) gonilnika za PostgreSQL. Ob nastavitivi ODBC DSN (Data Source Name) pa se povezava še poenostavi.

connection\_string = 'DSN=Seminarska'

connection = pyodbc.connect(connection\_string)

## Continent

Ker ni pričakovati spremembe imen ali števila kontinentov sva kontinente zapisala statično v kodo ter jih nato poslala v tabelo *continent*.

continents = {

    0: 'Africa',

    1: 'Asia',

    2: 'Europe',

    3: 'North America',

    4: 'Oceania',

    5: 'South America'

}

def insert\_continents():

    try:

        cursor = connection.cursor()

        for x in continents:

            command = "INSERT INTO continent (id\_continent, continent\_name) \

VALUES ({0}, '{1}')"

            cursor.execute(command.format(x, continents[x]))

        cursor.commit()

    except:

        print("No new continents inserted")

## Country

Tabela *country* vsebuje podatke o državah sva izluščila iz podaktovne zbirke COVID-19 - World Confirmed Cases, Deaths, and Testing.

def insert\_countries():

    try:

f = pd.read\_csv("../datasets/covid-19-world-cases-deaths- \

testing\_dataset\_covid-19-world- \

cases-deaths-testing.csv")

        keep\_col = ['iso\_code', 'continent', 'location', 'population',

'population\_density', 'median\_age', 'aged\_65\_older',

                    'aged\_70\_older', 'gdp\_per\_capita', 'extreme\_poverty',

 'cardiovasc\_death\_rate', 'diabetes\_prevalence',

                    'female\_smokers', 'male\_smokers','handwashing\_facilities',  'hospital\_beds\_per\_thousand',

                    'life\_expectancy', 'human\_development\_index']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        last\_country = ''

        for row in new\_f.itertuples():

            if (row[3] == 'International' or row[3] == 'World'

or last\_country == row[1]):

                continue

            query = "INSERT INTO country (iso, id\_continent, country\_name,  \

population, population\_density, median\_age, \

aged\_65\_or\_older, aged\_70\_or\_older, gdp\_per\_capita, \

extreme\_poverty, cardio\_vasc\_death\_rate, \

diabetes\_prevalence, female\_smokers, male\_smokers, \

handwashing\_facilities, hospital\_beds\_per\_thousand, \

life\_expectancy, human\_development\_index) VALUES" \

                    "('{0}', {1}, '{2}', {3}, {4}, {5}, {6}, {7}, {8}, {9}, \

{10}, {11}, {12}, {13}, {14}, {15}, {16}, {17})"

            cursor.execute(query.format(row[1], get\_continent\_id(row[2]),

row[3].replace('\'', ''), handleNan(row[4]),

                    handleNan(row[5]), handleNan(row[6]), handleNan(row[7]),

handleNan(row[8]), handleNan(row[9]), handleNan(row[10]),

handleNan(row[11]), handleNan(row[12]), handleNan(row[13]),

handleNan(row[14]), handleNan(row[15]), handleNan(row[16]),

                  handleNan(row[17]), handleNan(row[18])))

            last\_country = row[1]

            cursor.commit()

## Severitiesprediction

Tabela *severitiesprediction* vsebuje napoved zasedenost bolnišnice za naslednjih 7 dni. Podatki so vzeti za 6. januar 2021. Ob zapisovanju v bazo vpisujemo podatke o koordinatah bolnišnice v povezavi z identiteto v slovar, saj nam bo ta slovar pomagal pri vpisovanju bolnišnic. Podatke sva pridobila iz podatkovne zbirke COVID-19 Prediction Models Counties & Hospitals | Yu Group (UC Berkeley)

def insert\_severity():

    try:

        f = pd.read\_csv("../datasets/covid-19-yu-group\_dataset\_ \

severity-index.csv")

        keep\_col = ['severity\_1-day', 'severity\_2-day',

'severity\_3-day', 'severity\_4-day',

'severity\_5-day', 'severity\_6-day',

                    'severity\_7-day', 'latitude', 'longitude']

        new\_f = f[keep\_col]

        #new\_f.to\_csv("severity.csv", index=False)

        cursor = connection.cursor()

        delete\_query = "TRUNCATE severityprediction RESTART IDENTITY CASCADE;"

        elementNo = 1

        cursor.execute(delete\_query)

        for row in new\_f.itertuples():

            query = "INSERT INTO severityprediction (severity\_1day, \

severity\_2day, severity\_3day, severity\_4day, \

severity\_5day, severity\_6day, severity\_7day, \

lat\_hospital, lon\_hospital) VALUES" \

                    "({0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8})"

            severity[(row[8],row[7])] = elementNo

            cursor.execute(query.format(row[1], row[2], row[3], row[4],

row[5], row[6], row[7], handleNan(row[8]),

handleNan(row[9])))

            if handleNan(row[9]) != 0 and handleNan(row[8]) != 0:

                severity[(row[8], row[9])] = elementNo

            print(row)

            elementNo += 1

        cursor.commit()

    except:

        print("No new severity records inserted")

## Unemployment

Tabela *unemployment* uporablja podatkovno zbirko Short-Term Labour Market Statistics, ki vsebuje podatke o procentu nezaposlenih prebivalcev članic OECD.

def insert\_unemployment():

    try:

        f = pd.read\_csv("../datasets/STLABOUR\_06012021210015887\_1.csv")

        keep\_col = ['ISO', 'Number date', 'Value']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            try:

                query = "INSERT INTO unemployment (iso, unemployment\_date, \

value) VALUES" \

                        "('{0}', '{1}', {2})"

                cursor.execute(query.format(row[1], row[2] + "-01", row[3]))

            except:

                print(row[1] + " is not a country ISO code, skipped")

        cursor.commit()

    except:

        print("No new unemployment records inserted")

## AgeGroup

Tabela *ageGroup* uporablja podatke o starostnih skupinah, ki sva jih zapisala statično v kodo ter poslala v bazo.

age\_groups = {

    0: 'Under 1 year',

    1: '0-17 years',

    2: '1-4 years',

    3: '5-14 years',

    4: '15-24 years',

    5: '18-29 years',

    6: '25-34 years',

    7: '30-49 years',

    8: '35-44 years',

    9: '45-54 years',

    10: '50-64 years',

    11: '55-64 years',

    12: '65-74 years',

    13: '75-84 years',

    14: '85 years and over'

}

def insert\_age\_group():

    try:

        cursor = connection.cursor()

        for x in age\_groups:

            command = "INSERT INTO agegroup (id\_age\_group, age\_group)" \

"VALUES ({0}, '{1}')"

            cursor.execute(command.format(x, age\_groups[x]))

        cursor.commit()

    except:

        print("No new age groups inserted")

## States Measures

Tabeli *states* in *measures* uporabljata podatkovno zbirko COVID-19 United States Reopen and Shut Down Status by State | NY Times za pridobitev podatkov o trenutnih ukrepih v posameznih zveznih državah ZDA, ter podatkov o samih zveznih državah. Podatke o imenu in okrajšavah zveznih državah zapišemo še v slovar saj nam bodo prišli prav še kasneje.

def insert\_state\_measures():

    try:

        f = pd.read\_csv("../datasets/nyt-states-reopen-status-covid" \

"-19\_dataset\_nyt-states-reopen-status-covid-19.csv")

        keep\_col = ['state\_abbreviation', 'state', 'opened\_food\_and\_drink',

'closed\_houses\_of\_worship', 'closed\_food\_and\_drink',

'opened\_retail', 'opened\_outdoor\_and\_recreation',

'closed\_outdoor\_and\_recreation', 'closed\_entertainment',

'opened\_industries', 'opened\_entertainment',

'opened\_personal\_care', 'opened\_houses\_of\_worship',

'population']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            measure\_id = row[0]

            query = "INSERT INTO state (code, ISO, state\_name, population) \

VALUES" \

 "('{0}', 'USA', '{1}', {2})"

            cursor.execute(query.format(row[1], row[2], row[14]))

            cursor.commit()

            industry\_measure = decide\_measure(None, row[10])

            personal\_care\_measure = decide\_measure(None, row[12])

            entertainment\_measure = decide\_measure(row[9], row[11])

            outdoor\_measure = decide\_measure(row[8], row[7])

            retail\_measure = decide\_measure(None, row[6])

            worship\_measure = decide\_measure(row[4], row[13])

            food\_measure = decide\_measure(row[5], row[3])

            command = "INSERT INTO measures (measure\_id, code, industry, "\

"personal\_care, entertainment, " \

                      "outdoor\_recreation, retail, house\_of\_worship, "

" food\_drink) VALUES" \

   "({0}, '{1}', {2}, {3}, {4}, {5}, {6}, {7}, {8})"

 cursor.execute(command.format(measure\_id, row[1],

industry\_measure, personal\_care\_measure,                          entertainment\_measure, outdoor\_measure,

retail\_measure, worship\_measure, food\_measure))

            cursor.commit()

            command = "UPDATE state SET measure\_id={0} WHERE code='{1}'"

            cursor.execute(command.format(measure\_id, row[1]))

            states[row[1]] = row[2]

            states[row[2]] = row[1]

            cursor.commit()

    except:

        print("No new states inserted")

        cursor = connection.cursor()

        query = "SELECT code, state\_name FROM state"

        cursor.execute(query)

        state\_list = cursor.fetchall()

        for state in state\_list:

            states[state[0]] = state[1]

            states\_inverse[state[1]] = state[0]

## ByGender

Tabela *byGender* uporablja podatkovno zbirko Provisional COVID-19 Death Counts by Sex, Age, and State | CDC / NCHS za pridobitev podatkov o smrtih ljudi glede na starostno skupino, spol ter zvezne države ZDA. Podatek »All sexes« uporabimo še za dodajanje število smrti vsaki zvezni državi.

def insert\_by\_age():

    try:

        f = pd.read\_csv("../datasets/covid-19-death-counts-sex-age"\

"-state\_dataset\_covid-19-death-counts-sex-age-state.csv")

        keep\_col = ['Data as of', 'State', 'Sex', 'Age group',

'COVID-19 Deaths', 'Total Deaths']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            if row[2] not in ['United States', 'Puerto Rico']:

                state = states\_inverse[row[2]]

                sex = row[3]

                age\_group = get\_age\_group\_id(row[4])

                covid\_deaths = handleNan(row[5])

                total\_deaths = handleNan(row[6])

                if age\_group is None and sex == 'All Sexes':

                    query = "UPDATE state SET covid\_deaths={0}, \

total\_deaths={1} WHERE code='{2}'"

                    cursor.execute(query.format(covid\_deaths, total\_deaths,

state))

                elif age\_group is not None and sex != 'All Sexes':

                    query = "INSERT INTO bygender (id\_age\_group, code, "\

"gender, report\_date, covid\_deaths\_by\_gender, "\

"total\_deaths\_by\_gender) VALUES" \

                         "({0}, '{1}', '{2}', '{3}', {4}, {5})"

                    cursor.execute(query.format(age\_group, state,

transformGender(sex), row[1], covid\_deaths,

total\_deaths))

        cursor.commit()

    except:

        print("No new data by gender and age added")

## Okrožje

Tabela *county* uporablja podatkovno zbirko Coronavirus (COVID-19) Data Hub za pridobitev podatkov o posameznih okrožjih, ki se nahajajo v ZDA.

def insert\_county():

    try:

        f = pd.read\_csv("../datasets/published\_PUBLIC\_COVID-19-" \

"Activity\_1609956124\_COVID-19 Activity.csv")

        keep\_col = ['COUNTY\_FIPS\_NUMBER', 'PROVINCE\_STATE\_NAME','COUNTY\_NAME']

        new\_f = f[keep\_col]

        #new\_f.to\_csv("county.csv", index=False)

        cursor = connection.cursor()

        last\_county = ''

        for row in new\_f.itertuples():

            if (row[1] in county or last\_county == row[3] or row[3] == "" or

row[3] == "Unknown" or handleNan(row[1]) == 0 or

row[2] in ["Northern Mariana Islands", "Virgin Islands"]):

                continue

            county[int(row[1])] = row[2]

            query = "INSERT INTO county (fips, code, county\_name) VALUES" \

                    "({0}, '{1}', '{2}')"

            cursor.execute(query.format(int(row[1]), states\_inverse[row[2]],

row[3].replace("'","")))

            last\_county = row[3]

        cursor.commit()

    except:

        print("No new counties inserted")

## Hospital

Tabela *hospital* uporablja podatkovni set COVID-19 Prediction Models Counties & Hospitals | Yu Group (UC Berkeley) za pridobitev podatkov o bolnišnicah v ZDA.

def insert\_hospital():

    try:

        f = pd.read\_csv("../datasets/usa-hospital-beds\_dataset\_usa-" \

"hospital-beds.csv")

        keep\_col = ['OBJECTID', 'FIPS', 'STATE\_NAME', 'HOSPITAL\_NAME',

'HOSPITAL\_TYPE', 'HQ\_ADDRESS', 'HQ\_CITY', 'HQ\_ZIP\_CODE',

                    'NUM\_LICENSED\_BEDS', 'NUM\_STAFFED\_BEDS', 'NUM\_ICU\_BEDS',

'ADULT\_ICU\_BEDS', 'PEDI\_ICU\_BEDS', 'BED\_UTILIZATION',

                    'Potential\_Increase\_In\_Bed\_Capac', 'AVG\_VENTILATOR\_USAGE',

 'Y', 'X']

        #id severity je za STATE\_NAME ampak ga nimamo in bo prazen

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            if handleNan(row[2]) == 0:

                continue

            query\_check =

"SELECT \* FROM county WHERE county.fips = " + str(row[2])

            if (cursor.execute(query\_check).rowcount == 0 or

handleNan(row[17]) == 0 or handleNan(row[18]) == 0):

                continue

            if (row[17], row[18]) in severity:

                sev\_id = severity[(row[17], row[18])]

                query = "INSERT INTO hospital (hospital\_id, fips, code, \

"id\_severity, hospital\_name, type, address, city, \

"zip, licenced\_beds, staffed\_beds," \

                        "icu\_beds, adult\_icu\_beds, pedi\_icu\_beds," \

"bed\_utilization, potential, avg\_ventilation\_use,"\

"lat, lon) VALUES" \

"({0}, {1}, '{2}', {3}, '{4}', '{5}', '{6}', '{7}', "\"'{8}', {9}, {10}, {11}, {12}, {13}, {14}, {15}, "\

"{16}, {17}, {18})"

cursor.execute(query.format(row[1], int(row[2]),

states\_inverse[row[3]], sev\_id,

row[4].replace("'", ""), row[5].replace("'", ""),

row[6].replace("'", ""), row[7].replace("'", ""),

                        row[8], handleNan(row[9]), handleNan(row[10]),

handleNan(row[11]), handleNan(row[12]),

handleNan(row[13]), handleNan(row[14]),

                         handleNan(row[15]), handleNan(row[16]),

handleNan(row[17]), handleNan(row[18])))

            else:

                query = "INSERT INTO hospital (hospital\_id, fips, code, "\

"hospital\_name, type, address, city, zip, " \

"licenced\_beds, staffed\_beds," \

                        " icu\_beds, adult\_icu\_beds, pedi\_icu\_beds, "\

"bed\_utilization, potential, avg\_ventilation\_use," \ "lat, lon) VALUES" \

"({0}, {1}, '{2}', '{3}', '{4}', '{5}', '{6}', '{7}'"\", {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, " \

"{16}, {17})"

                cursor.execute(query.format(row[1], int(row[2]),

states\_inverse[row[3]], row[4].replace("'",""),

                         row[5].replace("'",""), row[6].replace("'",""),

row[7].replace("'",""), row[8], handleNan(row[9]),

handleNan(row[10]), handleNan(row[11]),

handleNan(row[12]), handleNan(row[13]),

handleNan(row[14]), handleNan(row[15]),

handleNan(row[16]), handleNan(row[17]),

handleNan(row[18])))

        cursor.commit()

    except:

        print("No new hospitals inserted")

## Covid19

Tabela *covid19* uporablja podatke več podatkovnih setov. Med drugimi notri vključujemo podatke o covid19 glede držav (COVID-19 - World Confirmed Cases, Deaths, and Testing), countyjev (Coronavirus (COVID-19) Data Hub) ter New Yorka ter San Francisca (NYC Coronavirus (COVID-19) Data | New York City Department of Health in San Francisco Bay Area COVID-19 Tracker).

### Države

def insert\_covid\_data():

    try:

        f = pd.read\_csv("../datasets/covid-19-world-cases-deaths-" \

"testing\_dataset\_covid-19-world-cases-deaths-testing.csv")"

        keep\_col = ['iso\_code', 'date', 'total\_cases', 'new\_cases',

'new\_cases\_smoothed', 'total\_deaths', 'new\_deaths',

                  'new\_deaths\_smoothed', 'total\_cases\_per\_million',

'new\_cases\_per\_million','new\_cases\_smoothed\_per\_million',

'total\_deaths\_per\_million', 'new\_deaths\_per\_million',

                  'new\_deaths\_smoothed\_per\_million', 'reproduction\_rate',

'icu\_patients', 'icu\_patients\_per\_million', 'hosp\_patients',

'hosp\_patients\_per\_million', 'weekly\_icu\_admissions',

               'weekly\_icu\_admissions\_per\_million', 'weekly\_hosp\_admissions', 'weekly\_hosp\_admissions\_per\_million', 'new\_tests',

'total\_tests', 'total\_tests\_per\_thousand',

'new\_tests\_per\_thousand', 'new\_tests\_smoothed',

                  'new\_tests\_smoothed\_per\_thousand', 'positive\_rate',

'tests\_per\_case', 'tests\_units','total\_vaccinations',

'total\_vaccinations\_per\_hundred', 'stringency\_index',

'location']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        cursor.execute('BEGIN')

        cursor.execute('DELETE FROM covid19')

        for row in new\_f.itertuples():

            if (row[36] in ['International', 'World']):

                continue

            query1="INSERT INTO covid19 (ISO, reproduction\_rate, new\_tests, "\

"new\_tests\_per\_thousand, total\_tests, " \

                    "total\_tests\_per\_thousand, new\_tests\_smoothed, " \

  "new\_tests\_smoothed\_per\_thousand, positive\_rate," \

                    "tests\_per\_case, tests\_unit, total\_vaccinations, " \

"total\_vaccinations\_per\_hundred, stringency\_index, " \

                    "date, new\_cases, total\_deaths, new\_cases\_smoothed, " \

"total\_cases, new\_deaths, new\_deaths\_smoothed, " \

"icu\_patients, weekly\_icu\_admissions, " \

"weekly\_hosp\_admissions) " \

                     "VALUES ('{0}', {1}, {2}, {3}, {4}, {5}, {6}, {7}," \

  "{8}, {9}, '{10}', {11}, {12}, {13}, '{14}', " \

                     "{15}, {16}, {17}, {18}, {19}, {20}, {21}, {22}, {23})"

            cursor.execute(query1.format(row[1], handleNan(row[15]),

handleNan(row[24]), handleNan(row[27]), handleNan(row[25]), handleNan(row[26]), handleNan(row[28]), handleNan(row[29]),

                  handleNan(row[30]), handleNan(row[31]), row[32],

handleNan(row[33]), handleNan(row[34]), handleNan(row[35]),

row[2], handleNan(4), handleNan(6), handleNan(5),

handleNan(3), handleNan(7), handleNan(8), handleNan(16),

handleNan(20), handleNan(22)))

        cursor.execute('COMMIT')

        cursor.commit()

    except:

        print("No new COVID-19 reports inserted")

### Okrožja

def insert\_covid\_by\_counties():

    f = pd.read\_csv("../datasets/published\_PUBLIC\_COVID-19" \

"-Activity\_1609956124\_COVID-19 Activity.csv")

    keep\_col = ['PEOPLE\_POSITIVE\_CASES\_COUNT', 'PROVINCE\_STATE\_NAME',

'REPORT\_DATE', 'PEOPLE\_DEATH\_NEW\_COUNT',

               'COUNTY\_FIPS\_NUMBER', 'COUNTRY\_ALPHA\_3\_CODE',

'PEOPLE\_POSITIVE\_NEW\_CASES\_COUNT',

               'PEOPLE\_DEATH\_COUNT', 'PROVINCE\_STATE\_NAME']

    new\_f = f[keep\_col]

    cursor = connection.cursor()

    counter = 0

    for row in new\_f.itertuples():

        if (row[6] != 'USA' or not isinstance(row[9], str) and

math.isnan(row[9]) or row[9] in

["Northern Mariana Islands", "Virgin Islands", "Guam",

"Puerto Rico"]):

            continue

        if not isinstance(row[1], str) and math.isnan(row[5]):

            query = "INSERT INTO covid19 (code, ISO, date, total\_cases,"\

"new\_deaths, new\_cases, total\_deaths)" \

                    "VALUES ('{0}', '{1}', '{2}', {3}, {4}, {5}, {6})"

            cursor.execute(query.format(states\_inverse[row[2]], row[6],

row[3], row[1], row[4], row[7], row[8]))

        else:

            query = "INSERT INTO covid19 (code, fips, ISO, date, " \

"total\_cases, new\_deaths, new\_cases, total\_deaths)" \

                    "VALUES ('{0}', {1}, '{2}', '{3}', {4}, {5}, {6}, {7})"

            cursor.execute(query.format(states\_inverse[row[2]], row[5],

row[6], row[3], row[1], row[4], row[7], row[8]))

        print("County record: {0}".format(counter))

        counter += 1

    cursor.commit()

### New York in San Francisco

def insert\_city():

    try:

        cursor = connection.cursor()

        cursor.execute("INSERT INTO city (id\_city, city\_name) " \

  "VALUES (0, 'New York City')")

        cursor.execute("INSERT INTO city (id\_city, city\_name) " \

"VALUES (1, 'San Francisco')")

        cursor.commit()

    except:

        print("No new cities inserted")

    insert\_new\_york()

    insert\_san\_francisco()

def insert\_new\_york():

    try:

        f = pd.read\_csv("../datasets/nyc-doh-covid-19\_dataset\_nyc-doh-" \

"covid-19-case-hosp-death.csv")

        keep\_col = ['DATE\_OF\_INTEREST', 'CASE\_COUNT', 'HOSPITALIZED\_COUNT',

'DEATH\_COUNT']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        sum\_cases = 0

        sum\_deaths = 0

        for row in new\_f.itertuples():

            sum\_cases += handleNan(row[2])

            sum\_deaths += handleNan(row[4])

            query = "INSERT INTO covid19 (date, new\_cases, total\_cases, " \

"hosp\_patients, new\_deaths, total\_deaths, " \

                     "id\_city, code, iso) " \

                    "VALUES ('{0}', {1}, {2}, {3}, {4}, {5}, 0, 'NY', 'USA')"

            cursor.execute(query.format(row[1], handleNan(row[2]), sum\_cases,

handleNan(row[3]), handleNan(row[4]), sum\_deaths))

        cursor.commit()

    except:

        print("No new New York data inserted")

def insert\_san\_francisco():

    try:

        f = pd.read\_csv("../datasets/covid-19-sf-bay-area-" \

"tracker\_dataset\_covid-19-sf-bay-area-tracker.csv")

        keep\_col = ['date', 'sf\_bay\_area\_total\_cases',

'new\_cases\_in\_sf\_bay\_area', 'growth\_factor',

                  'total\_fatalities\_in\_sf\_bay\_area',

'new\_fatalities\_in\_the\_bay\_area']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            if not isinstance(row[1], str) and math.isnan(row[1]):

                continue

            date = row[1].split(', ')[1].split(' ')

            month = transform\_month\_name(date[0])

            day = date[1]

            new\_date = "2020-" + month + "-" + day

            query = "INSERT INTO covid19 (date, total\_cases, new\_cases, " \

"reproduction\_rate, total\_deaths, new\_deaths, " \

                    "id\_city, code, iso) VALUES "\

" ('{0}', {1}, {2}, {3}, {4}, {5}, 1, 'CA', 'USA')"

            cursor.execute(query.format(

                new\_date,

                handleNan(row[2]),

                handleNan(row[3]),

                handleNan(row[4]),

                handleNan(row[5]),

                handleNan(row[6])

            ))

        cursor.commit()

    except:

        print("No new San Francisco data inserted")

## ApprovalRating

Tabela *approvalRating* uporablja podatke iz podatkovnega seta Donald Trump's Approval Ratings | FiveThirtyEight. Vsebuje podatke podpore zdaj že skoraj nekdanjemu predsedniku ZDA Donaldu Trumpu za obdobje njegovega mandata.

def insert\_approval():

    try:

        f = pd.read\_csv("../datasets/fivethirtyeight-trump-approval" \

"-ratings\_dataset\_approval\_topline.csv")

        keep\_col = ['subgroup', 'modeldate', 'approve\_estimate',

'approve\_hi', 'approve\_lo']

        new\_f = f[keep\_col]

        cursor = connection.cursor()

        for row in new\_f.itertuples():

            if row[1] == 'All polls':

                query = "INSERT INTO approvalrating (approval\_estimate, " \

"approval\_high, approval\_low, date) " \

                        "VALUES ({0}, {1}, {2}, '{3}') RETURNING id\_approval"

                cursor.execute(query.format(row[3], row[4], row[5], row[2]))

                id = cursor.fetchall()[0][0]

                query = "UPDATE covid19 SET id\_approval={0} " \

"WHERE iso='USA' AND date='{1}'"

                cursor.execute(query.format(id, row[2]))

                query = "SELECT id\_record FROM covid19 WHERE iso='USA' " \

"AND date='{0}'"

                cursor.execute(query.format(row[2]))

                id\_record = cursor.fetchall()

                if len(id\_record) > 0:

                    id\_record = id\_record[0][0]

                    query = "UPDATE approvalrating SET id\_record={0}" \

" WHERE id\_approval={1}"

                    cursor.execute(query.format(id\_record, id))

        cursor.commit()

    except:

        print("No new approvals inserted")

# PoIZVEDBE

## Število potrjenih okuženih glede na milijon prebivalcev po kontinentih

**Poizvedba**

SELECT

    continent.continent\_name,

    (SUM(total\_cases) / (

        SELECT SUM(country1.population)

        FROM country AS country1

        INNER JOIN continent AS continent1

        ON country1.id\_continent=continent1.id\_continent

        WHERE continent1.id\_continent=continent.id\_continent

        GROUP BY continent1.id\_continent

        ORDER BY AVG(country1.population))) \* 1000000

        AS total\_cases\_per\_million,

    (SUM(total\_deaths) / (

        SELECT SUM(country1.population)

        FROM country AS country1

        INNER JOIN continent AS continent1

        ON country1.id\_continent=continent1.id\_continent

        WHERE continent1.id\_continent=continent.id\_continent

        GROUP BY continent1.id\_continent

        ORDER BY AVG(country1.population))) \* 1000000

        AS total\_deaths\_per\_million

FROM covid19

    INNER JOIN country ON covid19.iso=country.iso

    INNER JOIN continent ON country.id\_continent=continent.id\_continent

WHERE date=(

    SELECT date AS DateCount

    FROM covid19

    WHERE code IS null

    GROUP BY date

    HAVING COUNT(date) > 100

    ORDER BY date DESC LIMIT 1) AND code IS null

GROUP BY continent.id\_continent

ORDER BY total\_cases\_per\_million DESC

**Rezultati**

|  |  |  |
| --- | --- | --- |
| continent\_name | total\_cases\_per\_million | total\_deaths\_per\_million |
| North America | 40620.74 | 879.74 |
| Europe | 32808.58 | 747.52 |
| South America | 31051.22 | 850.52 |
| Asia | 4558.76 | 74.24 |
| Africa | 2130.06 | 50.74 |
| Oceania | 762.36 | 22.82 |

Tabela 1: Število potrjenih okuženih in mrtvih po kontinentih na milijon prebivalcev

Graf 1: Število potrjenih okuženih po kontinentih na milijon prebivalcev

Graf 2: Število mrtvih po kontinentih na milijon prebivalcev

## Države z največjim številom potrjenih okuženih na milijon prebivalcev

**Poizvedba**

SELECT continent\_name, country\_name, total\_cases, total\_deaths, (total\_cases / population) \* 1000000

AS cases\_per\_million

FROM covid19

INNER JOIN country ON covid19.iso=country.iso

INNER JOIN continent ON country.id\_continent=continent.id\_continent

WHERE date=(

    SELECT date AS DateCount

    FROM covid19

    WHERE code IS null

    GROUP BY date

    HAVING COUNT(date) > 100

    ORDER BY date DESC LIMIT 1)

    AND code IS null

ORDER BY (total\_cases / population) \* 1000000 DESC

LIMIT 5

**Rezultati**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| continent\_name | country\_name | total\_cases | total\_deaths | cases\_per\_million |
| Europe | Andorra | 8249.00 | 84 | 106762.4409 |
| Europe | Montenegro | 49630.00 | 692 | 79020.861 |
| Europe | Luxembourg | 46986.00 | 508 | 75060.3857 |
| Europe | San Marino | 2493.00 | 62 | 73457.48129 |
| Europe | Czechia | 746714.00 | 12070 | 69727.82287 |

Tabela 2: Države z največjim številom potrjenih okuženih glede na milijon prebivalcev

Graf 3: Države z največjim številom potrjenih okuženih glede na milijon prebivalcev

## Države z najmanjšim številom potrjenih okuženih na milijon prebivalcev

**Poizvedba**

SELECT continent\_name, country\_name, total\_cases, total\_deaths, (total\_cases / population) \* 1000000

AS cases\_per\_million

FROM covid19

INNER JOIN country ON covid19.iso=country.iso

INNER JOIN continent ON country.id\_continent=continent.id\_continent

WHERE date=(

    SELECT date AS DateCount

    FROM covid19

    WHERE code IS null

    GROUP BY date

    HAVING COUNT(date) > 100

    ORDER BY date DESC LIMIT 1)

    AND code IS null

ORDER BY (total\_cases / population) \* 1000000 ASC

LIMIT 5

**Rezultati**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| continent\_name | country\_name | total\_cases | total\_deaths | cases\_per\_million |
| Oceania | Vanuatu | 1.00 | 0 | 3.255738239 |
| Asia | Laos | 41.00 | 0 | 5.635308147 |
| Africa | Tanzania | 509.00 | 21 | 8.521079871 |
| Oceania | Samoa | 2.00 | 0 | 10.08013709 |
| Asia | Vietnam | 1497.00 | 35 | 15.3793075 |

Tabela 3: Države z najmanjšim številom okuženih glede na milijon prebivalcev

Graf 4: Države z najmanjšim številom potrjenih okuženih glede na milijon prebivalcev

## Primerjava med ZDA, New Yorkom ter San Franciscom 13. 5. 2020

**Poizvedba**

(SELECT date, country.country\_name, state.state\_name, city.city\_name,

    covid19.new\_cases, covid19.total\_cases, covid19.new\_deaths,

    covid19.total\_deaths FROM covid19

INNER JOIN city ON covid19.id\_city=city.id\_city

INNER JOIN country ON covid19.iso=country.ISO

INNER JOIN state ON covid19.code=state.code

WHERE city.id\_city=0 AND date=(

    (SELECT date FROM covid19

    WHERE id\_city=0 GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19

    WHERE id\_city=1 GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19

    WHERE id\_city IS null AND iso='USA'

    GROUP BY date ORDER BY date DESC LIMIT 1)

    ORDER BY date ASC LIMIT 1)

ORDER BY date DESC)

UNION

(SELECT date, country.country\_name, state.state\_name, city.city\_name,

    covid19.new\_cases, covid19.total\_cases, covid19.new\_deaths,

    covid19.total\_deaths FROM covid19

INNER JOIN city ON covid19.id\_city=city.id\_city

INNER JOIN country ON covid19.iso=country.ISO

INNER JOIN state ON covid19.code=state.code

WHERE city.id\_city=1 AND date=(

    (SELECT date FROM covid19 WHERE id\_city=0

    GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19 WHERE id\_city=1

    GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19 WHERE id\_city IS null AND iso='USA'

    GROUP BY date ORDER BY date DESC LIMIT 1)

    ORDER BY date ASC LIMIT 1)

ORDER BY date DESC)

UNION

(SELECT date, country.country\_name, null AS state\_name, null AS city\_name,

    covid19.new\_cases, covid19.total\_cases, covid19.new\_deaths,

    covid19.total\_deaths FROM covid19

INNER JOIN country ON covid19.iso=country.ISO

WHERE covid19.iso='USA' AND covid19.code IS null AND date=(

    (SELECT date FROM covid19 WHERE id\_city=0

    GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19 WHERE id\_city=1

    GROUP BY date ORDER BY date DESC LIMIT 1)

    UNION

    (SELECT date FROM covid19 WHERE id\_city IS null AND iso='USA'

    GROUP BY date ORDER BY date DESC LIMIT 1)

    ORDER BY date ASC LIMIT 1)

ORDER BY date DESC)

ORDER BY total\_cases DESC

**Rezultati**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| date | country\_  name | state\_  name | city\_  name | new\_  cases | total\_  cases | new\_  deaths | total\_  deaths |
| 2020-05-13 | United States | NULL | NULL | 20305 | 1400298 | 1741 | 87133 |
| 2020-05-13 | United States | New York | New York City | 1331 | 189642 | 72 | 16010 |
| 2020-05-13 | United States | California | San Francisco | 118 | 10308 | 7 | 382 |

Tabela 4: Primerjava ZDA, New York, San Francisco 13. 5. 2020

Iz dobljenih rezultatov lahko lepo vidimo, da je »prvi val« močno prizadel New York City in krepko močneje kot pa San Francisco.

## Delež starostnih skupin med COVID-19 smrtmi

**Poizvedba**

SELECT agegroup.age\_group,(CAST(SUM(covid\_deaths\_by\_gender) AS DECIMAL(7,2)) /

    (SELECT SUM(covid\_deaths) FROM state)) \* 100

    AS covid\_deaths, SUM(total\_deaths\_by\_gender) AS total\_deaths

FROM bygender

INNER JOIN agegroup ON bygender.id\_age\_group=agegroup.id\_age\_group

GROUP BY agegroup.id\_age\_group

ORDER BY covid\_deaths DESC

**Rezultati**

|  |  |  |
| --- | --- | --- |
| age\_group | covid\_deaths [%] | total\_deaths |
| 85 years and over | 31.8643 | 871982.0000 |
| 75-84 years | 27.3947 | 704438.0000 |
| 65-74 years | 21.3276 | 576781.0000 |
| 50-64 years | 14.8694 | 475506.0000 |
| 55-64 years | 11.9190 | 377170.0000 |
| 45-54 years | 4.7882 | 163927.0000 |
| 30-49 years | 4.0679 | 190615.0000 |
| 35-44 years | 1.7552 | 89917.0000 |
| 25-34 years | 0.6232 | 63552.0000 |
| 18-29 years | 0.3066 | 54785.0000 |
| 15-24 years | 0.0951 | 30972.0000 |
| 0-17 years | 0.0093 | 28454.0000 |
| Under 1 year | 0.0000 | 16027.0000 |
| 1-4 years | 0.0000 | 2824.0000 |
| 5-14 years | 0.0000 | 4703.0000 |

Tabela 5: Delež starostnih skupin med COVID-19 smrtmi

Stolpec covid\_deaths nam pove kolikšen delež vseh smrti katerih razlog je COVID-19 predstavlja določena starostna skupina.

V rezultatih sta dva statistična modela razdelitve ljudi v starostne grupe. Za predstavitev podatkov v tortnem diagramu sva si izbrala nam bolj poznanega, ki vsebuje: 0-17 let, 18-29 let, 30-49 let, 50-64 let, 65-74 let, 75-84 let in 85 let in starejši.

Deleži starostnih skupin so razporejene po pričakovanjih, kjer umrli s starostjo nad 50 let predstavljajo 95,46% delež vseh umrlih za COVID-19.

## Delež smrti zaradi COVID-19 po starostnih skupinah

**Poizvedba**

SELECT agegroup.age\_group,(CAST(SUM(covid\_deaths\_by\_gender) AS DECIMAL(7,2)) /

    SUM(total\_deaths\_by\_gender)) \* 100 AS covid\_deaths,

    SUM(total\_deaths\_by\_gender) AS total\_deaths

FROM bygender

INNER JOIN agegroup ON bygender.id\_age\_group=agegroup.id\_age\_group

GROUP BY agegroup.id\_age\_group

ORDER BY covid\_deaths DESC

**Rezultati**

|  |  |  |
| --- | --- | --- |
| age\_group | covid\_deaths [%] | total\_deaths |
| 75-84 years | 11.7319 | 704438.0000 |
| 65-74 years | 11.1552 | 576781.0000 |
| 85 years and over | 11.0241 | 871982.0000 |
| 55-64 years | 9.5334 | 377170.0000 |
| 50-64 years | 9.4337 | 475506.0000 |
| 45-54 years | 8.8118 | 163927.0000 |
| 30-49 years | 6.4381 | 190615.0000 |
| 35-44 years | 5.8888 | 89917.0000 |
| 25-34 years | 2.9582 | 63552.0000 |
| 18-29 years | 1.6884 | 54785.0000 |
| 15-24 years | 0.9266 | 30972.0000 |
| 0-17 years | 0.0984 | 28454.0000 |
| Under 1 year | 0.0000 | 16027.0000 |
| 1-4 years | 0.0000 | 2824.0000 |
| 5-14 years | 0.0000 | 4703.0000 |

Tabela 6: Delež smrti zaradi COVID-19 po starostnih skupinah

Stolpec covid\_deaths nam pove kolikšen delež vseh smrti so smrti katerih razlog je COVID-19 po starostnih skupinah.

V rezultatih sta dva statistična modela razdelitve ljudi v starostne grupe. Za predstavitev podatkov v tortnem diagramu sva si izbrala nam bolj poznanega, ki vsebuje: 0-17 let, 18-29 let, 30-49 let, 50-64 let, 65-74 let, 75-84 let in 85 let in starejši.

Tudi tukaj so rezultati pričakovani. Največ skupnih umrlih najdemo v starostnih skupinah, ki vsebujejo starejše osebe. Največji delež COVID-19 umrlih med vsemi umrlimi je prav tako pričakovno v skupinah starejših oseb.

Graf 5: Delež starostnih skupin med COVID-19 smrtmi

Graf 6: Delež smrti zaradi COVID-19 po starostnih skupinah

## Okrožja z največjim številom bolnišnic

Želimo pridobiti okrožja z največjim številom bolnišnic v ZDA ter dodati stolpec s pripadujočo zvezno državo.

**Poizvedba**

SELECT state.state\_name, county.county\_name, COUNT(county.code) FROM county

FULL OUTER JOIN hospital ON hospital.fips = county.fips

FULL OUTER JOIN state ON state.code = county.code

GROUP BY county.county\_name, county.code, state.state\_name

ORDER BY count DESC

LIMIT 5

**Rezultati**

|  |  |  |
| --- | --- | --- |
| state\_name | county\_name | count |
| California | Los Angeles | 108 |
| Texas | Harris | 74 |
| Illinois | Cook | 68 |
| Arizona | Maricopa | 64 |
| Texas | Tarrant | 47 |

Tabela 7: Okrožja z največjim številom bolnišnic

Rezultati niso presenetljivi saj vsa ta okrožja vsebujejo največja ameriška mesta. Los Angeles County je county z največjim številom prebivalcev v ZDA in vsebuje, kot nam že ime pove, mesto Los Angeles. Harris County vsebuje Houston, Cook County vsebuje Chicago, Maricopa County vsebuje Phoenix ter Tarrant County vsebuje Fort Worth, ki je manjša polovica dvojčka mest, ki vsebuje še Dallas.

## Delež bolnišnic po rangu napovedi za naslednjih 7 dni

Bolnišnice imajo napovedi za naslednjih 7 dni. Vsak dan je označen s številom od 1 do 3, kjer 1 pomeni manj resno napoved in 3 zelo resno napoved. Napovedi za 1 teden sva seštela in sestavila lestvico od 7 do 21.

**Poizvedba**

SELECT

    (severity\_1day + severity\_2day + severity\_3day + severity\_4day +

    severity\_5day + severity\_6day + severity\_7day) AS severity,

    COUNT(severity\_1day + severity\_2day + severity\_3day + severity\_4day +

    severity\_5day + severity\_6day + severity\_7day) AS sum,

    ((CAST(COUNT(severity\_1day+severity\_2day + severity\_3day + severity\_4day +

    severity\_5day + severity\_6day + severity\_7day) AS DECIMAL(7, 2))) /

    (SELECT COUNT(\*) FROM hospital)) \* 100 AS percentage

FROM hospital

INNER JOIN severityprediction ON severityprediction.id\_severity = hospital.id\_severity

GROUP BY severity\_1day + severity\_2day + severity\_3day + severity\_4day + severity\_5day + severity\_6day + severity\_7day

ORDER BY severity DESC

**Rezultati**

Stolpec sum vsebuje število bolnišnic v posameznem rangu napovedi, percentage pa delež bolnišnic v rangu od števila vseh bolnišnic.

|  |  |  |
| --- | --- | --- |
| severity | sum | percentage |
| 21 | 1318 | 26.4181 |
| 20 | 89 | 1.7839 |
| 19 | 43 | 0.8619 |
| 18 | 23 | 0.4610 |
| 17 | 21 | 0.4209 |
| 16 | 23 | 0.4610 |
| 15 | 112 | 2.2449 |
| 14 | 1429 | 28.6430 |
| 13 | 157 | 3.1469 |
| 12 | 44 | 0.8819 |
| 11 | 23 | 0.4610 |
| 10 | 24 | 0.4811 |
| 9 | 25 | 0.5011 |
| 8 | 156 | 3.1269 |
| 7 | 1502 | 30.1062 |

Graf 7: Delež bolnišnic po rangu napovedi za naslednjih 7 dni

## Prikaz podatkov o okužbah, smrtih ter testiranju v Sloveniji

**Poizvedba**

SELECT date, country.country\_name, new\_tests, total\_tests, new\_cases,

    total\_cases, positive\_rate, new\_deaths, total\_deaths FROM covid19

INNER JOIN country ON country.iso = covid19.iso

WHERE country.iso='SVN'

ORDER BY date ASC

**Rezultati**

Ker poizvedba vrne preko preko 300 vrsticjih bomo prikazali le nekaj.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| date | country\_  name | new\_  tests | total\_  tests | new\_  cases | total\_  cases | positive\_  rate | new\_  deaths | total\_  deaths |
| 15.03.2020 | Slovenia | 590 | 3748 | 38 | 219 | 0 | 0 | 1 |
| 16.03.2020 | Slovenia | 871 | 4619 | 34 | 253 | 0 | 0 | 1 |
| 17.03.2020 | Slovenia | 1121 | 5740 | 22 | 275 | 0 | 0 | 1 |
| 18.03.2020 | Slovenia | 1026 | 6766 | 0 | 275 | 0 | 0 | 1 |
| 19.03.2020 | Slovenia | 1184 | 7950 | 11 | 286 | 0.029 | 0 | 1 |
| 20.03.2020 | Slovenia | 1242 | 9192 | 55 | 341 | 0.029 | 0 | 1 |
| 21.03.2020 | Slovenia | 872 | 10064 | 42 | 383 | 0.029 | 0 | 1 |
| 22.03.2020 | Slovenia | 731 | 10795 | 31 | 414 | 0.028 | 1 | 2 |

Tabela 8: Prikaz podatkov o okužbah, smrtih ter testiranju v Sloveniji

Graf 8: Prikaz podatkov o novih okužbah in smrtih

Ker je smrti procentualno nekje okoli 2% jih je na tem grafu težko zaznati. Zaznamo pa redne močne padce v številu novih okužb. Ti padci se dogaja vsako nedeljo, še močnejši pa so vsak ponedeljek, ko očitno testirajo najmanj ljudi, kar prikaže tudi naslednji graf.

Graf 9: Prikaz podatkov o številu opravljenih ter potrjenih testov ter stopnji potrditve

Iz grafa lahko ponovno razberemo redne padce števila opravljenih testov, kar razloži padce v številu novih okužb. Opazimo lahko tudi da je v drugem valu ne močno naraslo le število novih okužb, temveč tudi stopnja pozitivnih testov.

## Primerjava novih okužb in smrti v Sloveniji s stopnjo brezposelnosti po mesecih

**Poizvedba**

SELECT EXTRACT(YEAR FROM unemployment\_date) as year,

    EXTRACT(MONTH FROM unemployment\_date) AS month,

    SUM(new\_cases) AS new\_cases, SUM(new\_deaths) AS new\_deaths,

    unemployment.value FROM unemployment

INNER JOIN country ON country.iso=unemployment.iso

FULL OUTER JOIN covid19 ON covid19.iso = country.iso AND

    EXTRACT(MONTH FROM unemployment\_date)=EXTRACT(MONTH FROM date) AND

    EXTRACT(YEAR FROM unemployment\_date)=EXTRACT(YEAR FROM date)

WHERE country.iso='SVN'

GROUP BY unemployment.unemployment\_id

ORDER BY year ASC, month ASC

**Rezultati**

Stolpec value predstavlja stopnjo brezposelnosti.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| year | month | new\_cases | new\_deaths | value |
| 2019 | **10** | 0 | 0 | 4.4 |
| 2019 | **11** | 0 | 0 | 4.1 |
| 2019 | **12** | 0 | 0 | 3.8 |
| 2020 | **1** | 0 | 0 | 4.1 |
| 2020 | **2** | 0 | 0 | 4.2 |
| 2020 | **3** | 802 | 15 | 4.5 |
| 2020 | **4** | 627 | 76 | 5.4 |
| 2020 | **5** | 44 | 17 | 5.5 |
| 2020 | **6** | 127 | 3 | 5.3 |
| 2020 | **7** | 556 | 8 | 4.8 |
| 2020 | **8** | 727 | 14 | 4.7 |
| 2020 | **9** | 2807 | 17 | 4.7 |
| 2020 | **10** | 28617 | 188 | 4.9 |

Tabela 9: Primerjava novih okužb in smrti s stopnjo brezposelnosti po mesecih v Sloveniji

Graf 10: Primerjava novih okužb s stopnjo brezposelnosti po mesecih v Sloveniji

Iz podatkov lahko vidimo, da je prvi val epidemije COVID-19 vplival na brezposelnost vendar ne močno in nekoliko zakasnjeno kar lahko pripišemo dejstvu, da so vladni ukrepi vključevali nadomestilo delavcem za čakanje na delo kar je povzročilo precej manj odpušanj.

## Primerjava novih okužb in smrti v ZDA s stopnjo brezposelnosti ter podporo Donaldu Trumpu po mesecih

**Poizvedba**

SELECT EXTRACT(YEAR FROM unemployment\_date) as year,

    EXTRACT(MONTH FROM unemployment\_date) AS month,

    SUM(new\_cases) AS new\_cases,

    SUM(new\_deaths) AS new\_deaths, unemployment.value,

    AVG(approvalrating.approval\_estimate) AS approval\_estimate,

    AVG(approvalrating.approval\_high) AS approval\_high,

    AVG(approvalrating.approval\_low) AS approval\_low FROM unemployment

INNER JOIN country ON country.iso=unemployment.iso

FULL OUTER JOIN covid19 ON covid19.iso = country.iso AND

    EXTRACT(MONTH FROM unemployment\_date)=EXTRACT(MONTH FROM date) AND

    EXTRACT(YEAR FROM unemployment\_date)=EXTRACT(YEAR FROM date)

FULL OUTER JOIN approvalrating ON approvalrating.id\_approval = covid19.id\_approval

WHERE country.iso='USA'

GROUP BY unemployment.unemployment\_id

ORDER BY year ASC, month ASC

**Rezultati**

Stolpec value predstavlja stopnjo brezposelnosti. Approval\_estimate predstavlja povprečje vseh anket o podpori predsedniku Donaldu Trumpu v mesescu, approval\_high najvišjo vrednost ter approval\_low najnižjo vrednost v mesecu.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| year | month | new\_  cases | new\_  deaths | value | approval\_  estimate | approval\_  high | approval\_  low |
| 2019 | **10** | NULL | NULL | 3.6 | NULL | NULL | NULL |
| 2019 | **11** | NULL | NULL | 3.5 | NULL | NULL | NULL |
| 2019 | **12** | NULL | NULL | 3.5 | NULL | NULL | NULL |
| 2020 | **1** | 14 | 0 | 3.6 | 42.8943 | 47.3660 | 38.4226 |
| 2020 | **2** | 81 | 2 | 3.5 | 43.5751 | 48.1224 | 39.0278 |
| 2020 | **3** | 448078 | 11761 | 4.4 | 43.5441 | 48.1835 | 38.9047 |
| 2020 | **4** | 1887589 | 131936 | 14.7 | 44.2529 | 48.1722 | 40.3336 |
| 2020 | **5** | 1465219 | 85293 | 13.3 | 43.2798 | 47.2793 | 39.2803 |
| 2020 | **6** | 1684735 | 43178 | 11.1 | 41.2217 | 45.3186 | 37.1247 |
| 2020 | **7** | 3833180 | 52778 | 10.2 | 40.4069 | 44.6650 | 36.1488 |
| 2020 | **8** | 2918610 | 58915 | 8.4 | 41.7198 | 46.3650 | 37.0746 |
| 2020 | **9** | 2403823 | 46527 | 7.9 | 43.1580 | 47.9195 | 38.3964 |
| 2020 | **10** | 3845115 | 47440 | 6.9 | 43.2605 | 47.6385 | 38.8825 |
| 2020 | **11** | 8850234 | 74158 | 6.7 | 44.4510 | 49.5656 | 39.3364 |

Tabela 10: Primerjava novih okužb in smrti v ZDA s stopnjo brezposelnosti ter podporo Donaldu Trumpu po mesecih

Graf 11: Primerjava novih okužb s stopnjo brezposelnosti v ZDA

Graf 12: Primerjava novih okužb s podporo Donaldu Trumpu

Iz grafa 11 lahko vidimo, da je epidemija COVID-19 na brezposelnost v ZDA vplivala precej močneje kot pa na brezposelnost v Sloveniji. To lahko pripišemo nekoliko zakasnelim ukrepim ter dejstvu da v ZDA država ni pomagala z enakimi ali podobnimi ukrepi kot z na primer »čakanjem na delo«.

Iz grafa 12 lahko razberemo da epidemija COVID-19 ni pomagali pri podpori predsedniku ZDA Donaldu Trumpu, dno pa je dosegla julija. Nato se začne dvigovati kar bi lahko večinsko pripisali začetku predsedniške kampanje in stabilizaciji gospodarstva ter brezposelnosti..

# Zaključek

Pri analizi podatkov je razvidno, da je imel SARS-COVID-19 vpliv na svet iz več vidikov. Iz ekonomskega vidika je virus vplival na Slovenijo in ZDA, kjer je zaradi manjšega in kasnejšega odziva, vpliv bil veliko bolj razviden. Ta odziv je povzročil tudi zaznaven padec popularnosti ameriškega predsednika Donalda Trumpa.

Prikazali smo, da virus potrjeno bolj razsaja po manjših razvitih neizoliranih državah (Andora, Črna Gora, Luksemburg, San Marino…), najmanj sledi pa je razvidno pri nerazvitih ali izoliranih državah. Možen razlog za to je manj testiranja (in posledično manj potrjenih primerov).

Tu bi lahko definirali dvodimenzionalen spekter, kjer na posamezno os uvrstimo izoliranost in velikost. V tem primeru bi Slovenijo postavili na spekter med manjše in neizolirane države, ZDA pa med velike in neizolirane države. Pri tem bi pričakovali iz zgornjih podatkov, da bi imel virus večji vpliv na Slovenijo kot ZDA, kar vidimo, da ne drži. Razlog za to je verjetno drugačna notranja politika posameznih držav, ki je pripomogla k drugačnim ekonomskim odzivom države.

Poleg primerjave držav, je razviden podoben trend tudi za cele kontinente. Izolirani in manj razviti kontinenti poročajo o manj primerih SARS-COVID-19. Razlog je verjetno v tem, da manj razviti kontinenti naredijo manj testov. Pri tej primerjavi pa je razvidna še ena informacija: v primerjavi z Evropo in Južna Amerika manj potrjenih okužb a večje število smrtnih žrtev, kar lahko indicira na slabšo kakovost zdravstva. Ta razlika postane še bolj očitna, če upoštevamo, da je povprečna starost v Južni Ameriki nižja kot v Evropi in da je iz podatkov razvidno, da je smrt zaradi virusa bolj verjetna pri starejši populaciji.

Kljub temu da starost vpliva na nevarnost virusa, je pa iz podatkov razvidno, da delež smrti zaradi korone od vseh smrti za določeno starostno skupino postane približno enak pri starosti nad 65 let.

# VIRI

COVID-19 - World Confirmed Cases, Deaths, and Testing  
<https://aws.amazon.com/marketplace/pp/prodview-3b32sjummof5s?qid=1610023807511&sr=0-8&ref_=srh_res_product_title> 09. 01. 2021

USA Hospital Beds - COVID-19 | Definitive Healthcare  
<https://aws.amazon.com/marketplace/pp/prodview-yivxd2owkloha?qid=1610023900123&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

Coronavirus (COVID-19) Data Hub  
<https://aws.amazon.com/marketplace/pp/prodview-a5mqede4xd4c4?qid=1610024218785&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

COVID-19 United States Reopen and Shut Down Status by State | NY Times  
<https://aws.amazon.com/marketplace/pp/prodview-ejbvrkmiwc5so?qid=1610023816933&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

Donald Trump's Approval Ratings | FiveThirtyEight  
<https://aws.amazon.com/marketplace/pp/prodview-qwvtstd6ihv5g?qid=1610023782964&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

NYC Coronavirus (COVID-19) Data | New York City Department of Health  
<https://aws.amazon.com/marketplace/pp/prodview-flc4q2mgzczvm?qid=1610023797926&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

COVID-19 Prediction Models Counties & Hospitals | Yu Group (UC Berkeley)  
<https://aws.amazon.com/marketplace/pp/prodview-px2tvvydirx4o?qid=1610023756902&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

San Francisco Bay Area COVID-19 Tracker  
<https://aws.amazon.com/marketplace/pp/prodview-7bbg5rjbsmimg?qid=1610023699570&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

Provisional COVID-19 Death Counts by Sex, Age, and State | CDC / NCHS  
<https://aws.amazon.com/marketplace/pp/prodview-fbud5zxqg37cm?qid=1610023646923&sr=0-1&ref_=srh_res_product_title> 09. 01. 2021

Short-Term Labour Market Statistics  
<https://stats.oecd.org/index.aspx?queryid=36324> 09. 01. 2012