

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RACUNALNIŠTVO IN INFORMATIKO

# Povezovanje COVID-19 podatkovnih zbirk

Seminarska naloga pri predmetu

Tehnologija in upravljanje podatkov

**Študenta**

Eva Bizilj

Nikolay Vasilev

**Mentor**

izr. prof. dr. Matjaž Kukar

Januar 2022, Ljubljana

# Vsebina

<b>Uvod .....</b>	<b>4</b>
<b>Predpostavke.....</b>	<b>5</b>
<b>Težave s podatki in podatkovnimi zbirkami.....</b>	<b>6</b>
<b>Pristop k problemski domeni in tehnologije .....</b>	<b>8</b>
<b>PowerDesigner.....</b>	<b>9</b>
<b>ASP.NET Web API .....</b>	<b>17</b>
<b>Swagger .....</b>	<b>22</b>
<b>Postman.....</b>	<b>23</b>
<b>Oddaljen strežnik .....</b>	<b>24</b>
Zaledje .....	25
Podatkovna baza .....	27
Prožilec .....	32
<b>Relacijska podatkovna baza (MySQL) .....</b>	<b>34</b>
<b>RStudio.....</b>	<b>35</b>
<b>Poizvedbe in rezultati.....</b>	<b>36</b>
<b>Predpostavka 1 .....</b>	<b>36</b>
Poizvedbe.....	36
Rezultati.....	44
<b>Predpostavka 2 .....</b>	<b>44</b>
Poizvedbe.....	44
Rezultati.....	50
<b>Predpostavka 3 .....</b>	<b>51</b>
Poizvedbe.....	51
Rezultati.....	55
<b>Predpostavka 4 .....</b>	<b>57</b>
Poizvedbe.....	58
Rezultati.....	61
<b>Analiza podatkov.....</b>	<b>61</b>
<b>Predpostavka 1 .....</b>	<b>62</b>
Koda.....	63

Rezultati.....	64
<b>Predpostavka 2 .....</b>	<b>65</b>
Koda.....	65
Rezultati.....	66
<b>Predpostavka 3 .....</b>	<b>69</b>
Koda.....	69
Rezultati.....	72
<b>Predpostavka 4 .....</b>	<b>73</b>
Koda.....	73
Rezultati.....	74
<b>Sklepi in ugotovitve .....</b>	<b>75</b>
<b>Zaključek.....</b>	<b>79</b>
<b>Slovarček neznanih besed .....</b>	<b>80</b>
<b>Viri .....</b>	<b>82</b>

## Uvod

V seminarSKI nalogI sva se lotila obdelave in analize podatkov. Pri tem sva uporabila tri podatkovne zbirke in sicer Covid-19 sledilnik, REST Countries (podatki o državah) in JHU CSSE (Center for Systems Science and Engineering at Johns Hopkins University). Zanimali so naju predvsem podatki povezani s koronavirusno boleznijo COVID-19, ki jo povzroča virus SARS-CoV-2 v Sloveniji.

Temo »Povezovanje COVID-19 podatkovnih zbirk« sva si izbrala, ker želiva bolje razumeti trende epidemije virusa COVID-19, njegove posledice ter poudariti smiselnost cepljenja proti virusu, ki med nami vztraja že skoraj 2 leti.

Pregledala sva zgoraj omenjene podatkovne zbirke, identificirala attribute ter naredila podatkovni model s pomočjo orodja PowerDesigner. Nato sva vzpostavila podatkovno bazo, ki v realnem času pridobiva podatke iz podatkovnih zbirk. Da bi pridobljene podatke bolje razumela in prišla do zanimivih zaključkov sva oblikovala nekaj smiselnih poizvedb nad tabelami ter jih analizirala in vizualizirala v RStudio.

## Predpostavke

Pred samou obdelavo in analizo podatkov sva postavila štiri predpostavke.

Glede na najino »laično« razumevanje virusa Covid-19 sva postavila naslednje predpostavke:

- 1. predpostavka:** Cepljenje proti Covid-19 zmanjša število smrti.  
Primerjala sva število smrti, število okuženih ter število cepljenih med obdobjem 2. in 4. vala epidemije Covid-19 v Sloveniji.
- 2. predpostavka:** Povprečna starost umrlih se je 4. valu v primerjavi z 2. valom zmanjšala.  
Primerjala sva število umrlih ter delež cepljenih po starostnih obdobjih v obdobju 2. in 4. vala v Sloveniji.
- 3. predpostavka:** V državah, kjer je večji delež cepljenih ljudi, je manjši delež hospitalizacij na intenzivni negi.  
Primerjala sva evropske države glede na delež cepljenih ter delež hospitalizacij na intenzivni negi.
- 4. predpostavka:** Različica delta je bolj kužna a manj smrtonosna od originalne različice virusa Covid-19.  
Primerjavala sva število okuženih, hospitaliziranih in število smrti v obdobju, ko je prevladovala originalna in delta različica virusa Covid-19 v Sloveniji.

Opomba: Vpliv cepljenja pri tem nisva upoštevala.

### Opombe

1.11.2020 – 1.2.2021 (analizirano obdobje 2. vala)

12.9.2021 - 12.12.2021 (analizirano obdobje 4. vala)

4.3.2020 - 1.4.2021 (prevladujoča originalna različica virusa Covid -19): 393 dni

11.7.2021 – 12.12.2021 (prevladujoča delta različica virusa Covid -19): 154 dni

4.3.2020 - 12.12.2021 (analizirano obdobje epidemije)

## Težave s podatki in podatkovnimi zbirkami

Pri seminarski nalogi sva imela težave predvsem pri iskanju uporabnih podatkov.

Soočila sva se z naslednjimi težavami:

- neurejenost JSON podatkov (slaba JSON struktura)
- šumi (manjkajoči in napačno vnešeni podatki, npr. za vse države niso znani podatki o številu zasedenih intenzivnih posteljah)
- premalo podatkov (samo za neko kratko časovno obdobje)
- neuporabni podatki (npr. samo število smrti na 100.000 prebivalcev in ne dejansko število smrti)

Primer neurejenosti JSON podatkov (vir: podatkovna zbirka data.europe.eu)

```
{
  "records" : [
    {
      "dateRep" : "14/12/2020",
      "day" : "14",
      "month" : "12",
      "year" : "2020",
      "cases" : 746,
      "deaths" : 6,
      "countriesAndTerritories" : "Afghanistan",
      "geoId" : "AF",
      "countryterritoryCode" : "AFG",
      "popData2019" : 38041757,
      "continentExp" : "Asia",
      "Cumulative_number_for_14_days_of_COVID-19_cases_per_100000" : "9.01377925"
    },
    {
      "dateRep" : "13/12/2020",
      "day" : "13",
      "month" : "12",
      "year" : "2020",
      "cases" : 298,
      "deaths" : 9,
      "countriesAndTerritories" : "Afghanistan",
      "geoId" : "AF",
      "countryterritoryCode" : "AFG",
      "popData2019" : 38041757,
      "continentExp" : "Asia",
      "Cumulative_number_for_14_days_of_COVID-19_cases_per_100000" : "7.05277624"
    }
  ]
}
```

Problem je, da ni gručeno po datumu ali državi. Zaradi tega se je potrebno sprehoditi po vseh zapisih (*records*) – kartezični produkt vseh opazovanih datumov in držav.

Časovna zahtevnost sprehoda po takem JSON zapisu je  $O(m * k)$  namesto  $O(m)$  ali  $O(k)$ , pri čemer velja  $m * k$  je št. vseh zapisov,  $m$  je število vseh datumov in  $k$  je število vseh držav.

Pri tem pa tudi velja, da ima vsak datum za vsako državo le en podatkovni paket, ki vsebuje število smrti, število aktivnih primerov ... tisti dan.

## Primer urejenosti JSON podatkov

Gručeno po datumu in nato po kraju.

```
1 [ {  
2   "date": "2021-01-10T00:00:00",  
3   "region": [  
4     {  
5       "regionName": "lj",  
6       "dailyActiveCases": 5282,  
7       "deceasedToDate": 722,  
8       "firstVaccineToDate": 8985,  
9       "secondVaccineToDate": 1  
10      }  
11    ]  
12  },  
13  {  
14    "date": "2021-01-11T00:00:00",  
15    "region": [  
16      {  
17        "regionName": "lj",  
18        "dailyActiveCases": 5146,  
19        "deceasedToDate": 727,  
20        "firstVaccineToDate": 9327,  
21      }  
22    ]  
23  }
```

Zaradi naštetih težav nekaterih zbirk podatkov, kot je npr. Kaggle in data.europe.eu nisva uporabila, saj bi nama to povzročalo velike težave z razčlenjevanjem (paršanjem) podatkov in obravnavo napak na zaledju.

Bila sva presenečena, da imajo tako velike in znane spletne strani tako »neurejene« podatke ter veliko težav na njihovih spletnih strežnikih.

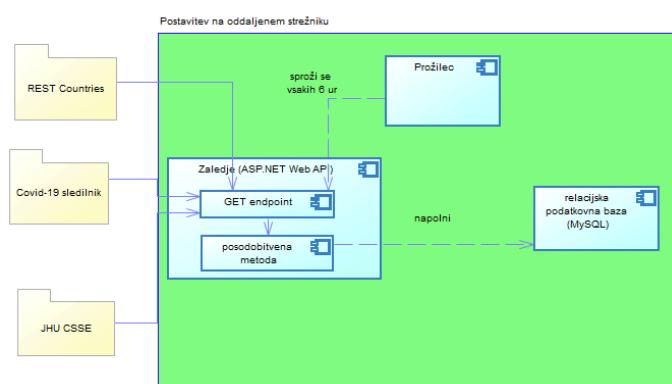
## Pristop k problemski domeni in tehnologije

Za izdelavo seminarske naloge sva uporabila naslednje tehnologije:

- PowerDesigner
- relacijsko podatkovno bazo (MySQL)
- ASP.NET Web API
- Node.js
- oddaljen strežnik Microsoft Azure
- Swagger
- Postman
- RStudio

Odločila sva se, da v seminarski nalogi ne bova uporabila programskega jezika Python, saj smo z njim že veliko delali na vajah oz. predavanjih in sva se že lela naučiti nekaj novega.

Za zaledje sva zato uporabila ASP.NET Web API in programski jezik C#.

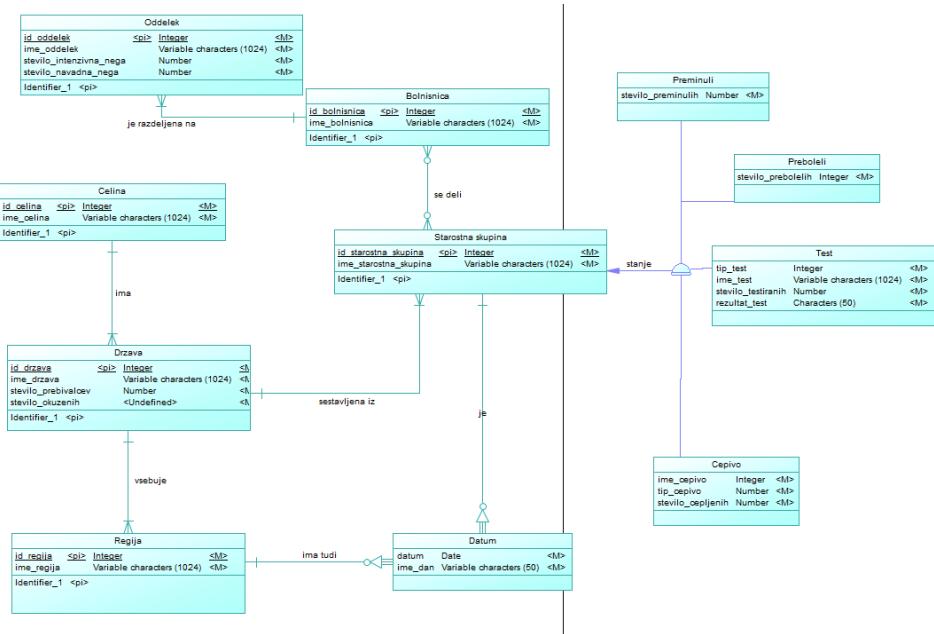


Zeleni pravokotnik na zgornji sliki predstavlja oddaljen strežnik Microsoft Azure, kamor sva postavila vso zaledje (ASP.NET Web API), Javascript prožilec (uporaba CRON jobs) in relacijsko podatkovno bazo (MySQL).

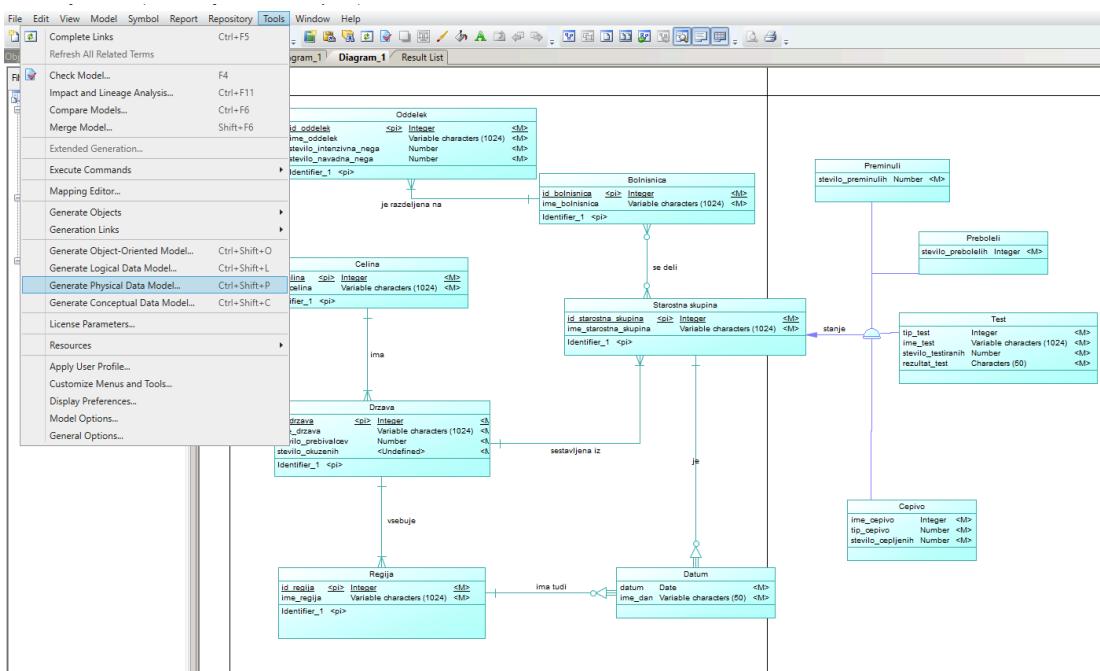
Prožilec na vsakih N-ur pokliče GET endpoint (npr. `\slovenia\cases`) na zaledju, ki pridobi podatke iz podatkovnih zbirk. Nato se pokliče posodobitvena metoda v funkciji, katera se sproži ob koncu življenskega cikla GET endpointa in doda vrstice v podatkovno bazo. Na podlagi podatkov v podatkovni bazi sva naredila SQL poizvedbe in nato pridobljene podatke s poizvedbami analizirala in vizualizirala v RStudiu.

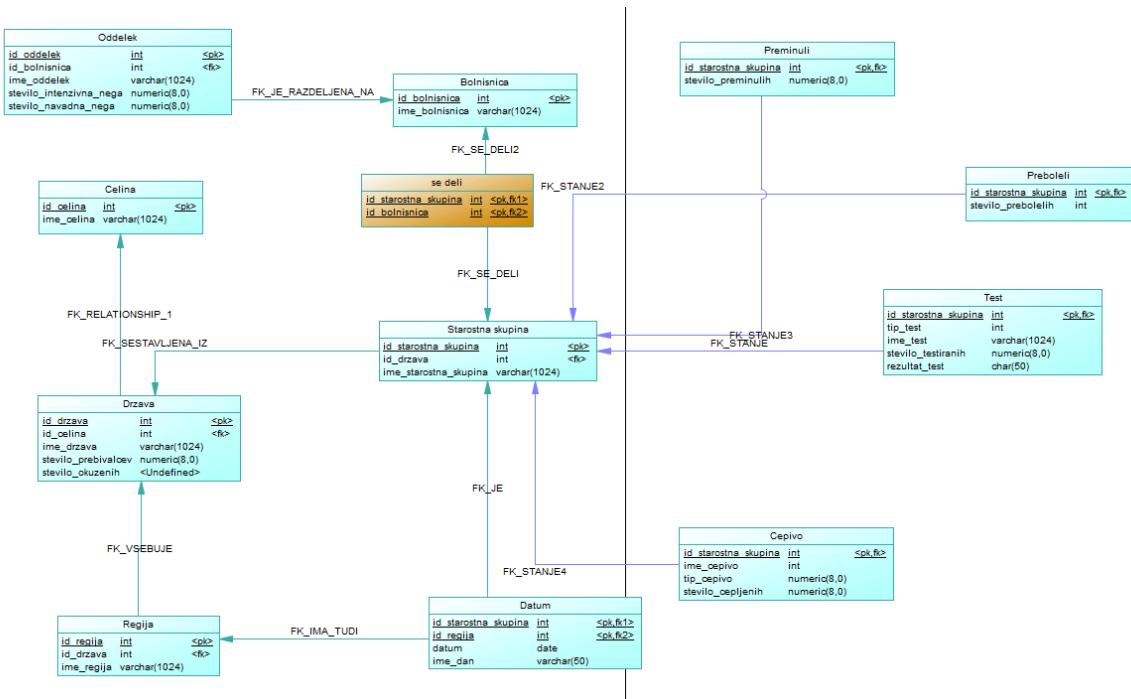
## PowerDesigner

Z orodjem PowerDesigner sva izdelala konceptualni podatkovni model, ki vsebuje 11 povezanih tabel.

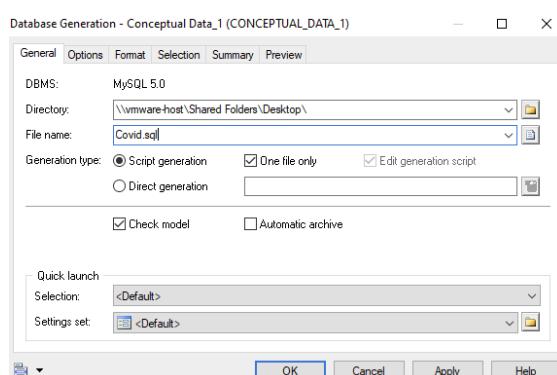
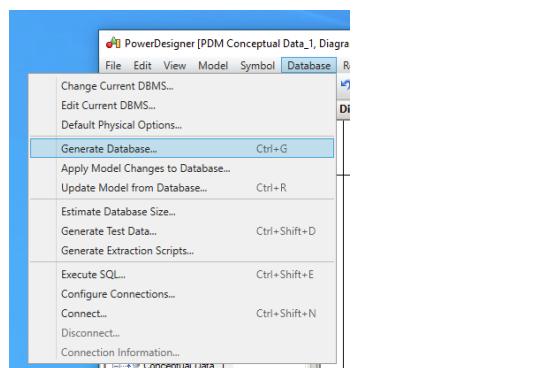


Dobljeni model sva nato pretvorila v logični model.





Logični model sva s pomočjo PowerDesigner-ja pretvorila v fizični model (SQL skripto).



## SQL skripta

```
/*=====
/* DBMS name: MySQL 5.0
/*=====*/
drop table if exists BOLNISNICA;

drop table if exists CELINA;

drop table if exists CEPIVO;

drop table if exists DATUM;

drop table if exists DRZAVA;

drop table if exists ODDELEK;

drop table if exists PREBOLELI;

drop table if exists PREMINULI;

drop table if exists REGIJA;

drop table if exists SE_DELI;

drop table if exists STAROSTNA_SKUPINA;

drop table if exists TEST;

/*=====
/* Table: BOLNISNICA
/*=====*/
create table BOLNISNICA
(
    ID_BOLNISNICA      int not null,
    IME_BOLNISNICA     varchar(1024) not null,
    primary key (ID_BOLNISNICA)
);
```

```

/*=====
/* Table: CELINA
=====
create table CELINA
(
    ID_CELINA      int not null,
    IME_CELINA     varchar(1024) not null,
    primary key (ID_CELINA)
);

/*=====
/* Table: CEPIVO
=====
create table CEPIVO
(
    ID_STAROSTNA_SKUPINA int not null,
    IME_CEPIVO        int not null,
    TIP_CEPIVO        numeric(8,0) not null,
    STEVILO_CEPLJENIH numeric(8,0) not null,
    primary key (ID_STAROSTNA_SKUPINA)
);

/*=====
/* Table: DATUM
=====
create table DATUM
(
    ID_STAROSTNA_SKUPINA int not null,
    ID_REGIJA        int not null,
    DATUM            date not null,
    IME_DAN          varchar(50) not null,
    primary key (ID_STAROSTNA_SKUPINA, ID_REGIJA)
);

```

```

/*=====
/* Table: DRZAVA
=====
create table DRZAVA
(
    ID_DRZAVA      int not null,
    ID_CELINA      int not null,
    IME_DRZAVA     varchar(1024) not null,
    STEVILO_PREBIVALCEV numeric(8,0) not null,
    STEVILO_OKUZENIH  char(10) not null,
    primary key (ID_DRZAVA)
);

/*=====
/* Table: ODDELEK
=====
create table ODDELEK
(
    ID_ODDELEK      int not null,
    ID_BOLNISNICA   int not null,
    IME_ODDELEK     varchar(1024) not null,
    STEVILO_INTENZIVNA_NEGA numeric(8,0) not null,
    STEVILO_NAVADNA_NEGA numeric(8,0) not null,
    primary key (ID_ODDELEK)
);

/*=====
/* Table: PREBOLELI
=====
create table PREBOLELI
(
    ID_STAROSTNA_SKUPINA int not null,
    STEVILO_PREBOLELIH  int not null,
    primary key (ID_STAROSTNA_SKUPINA)
);

```

```

/*=====
/* Table: PREMINULI
=====
create table PREMINULI
(
    ID_STAROSTNA_SKUPINA int not null,
    STEVILO_PREMINULIH numeric(8,0) not null,
    primary key (ID_STAROSTNA_SKUPINA)
);

/*=====
/* Table: REGIJA
=====
create table REGIJA
(
    ID_REGIJA      int not null,
    ID_DRZAVA      int not null,
    IME_REGIJA     varchar(1024) not null,
    primary key (ID_REGIJA)
);

/*=====
/* Table: SE_DELI
=====
create table SE_DELI
(
    ID_STAROSTNA_SKUPINA int not null,
    ID_BOLNISNICA      int not null,
    primary key (ID_STAROSTNA_SKUPINA, ID_BOLNISNICA)
);

```

```

/*=====
/* Table: STAROSTNA_SKUPINA          */
/*=====*/
create table STAROSTNA_SKUPINA
(
    ID_STAROSTNA_SKUPINA int not null,
    ID_DRZAVA      int not null,
    IME_STAROSTNA_SKUPINA varchar(1024) not null,
    primary key (ID_STAROSTNA_SKUPINA)
);

/*=====
/* Table: TEST                      */
/*=====*/
create table TEST
(
    ID_STAROSTNA_SKUPINA int not null,
    TIP_TEST      int not null,
    IME_TEST      varchar(1024) not null,
    STEVILO_TESTIRANIH numeric(8,0) not null,
    REZULTAT_TEST   char(50) not null,
    primary key (ID_STAROSTNA_SKUPINA)
);

alter table CEPIVO add constraint FK_STANJE4 foreign key (ID_STAROSTNA_SKUPINA)
    references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;

alter table DATUM add constraint FK_IMA_TUDI foreign key (ID_REGIJA)
    references REGIJA (ID_REGIJA) on delete restrict on update restrict;

alter table DATUM add constraint FK_JE foreign key (ID_STAROSTNA_SKUPINA)
    references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;

alter table DRZAVA add constraint FK_RELATIONSHIP_1 foreign key (ID_CELINA)
    references CELINA (ID_CELINA) on delete restrict on update restrict;

alter table ODDELEK add constraint FK_JE_RAZDELJENA_NA foreign key (ID_BOLNISNICA)
    references BOLNISNICA (ID_BOLNISNICA) on delete restrict on update restrict;

```

```
alter table PREBOLELI add constraint FK_STANJE2 foreign key (ID_STAROSTNA_SKUPINA)
references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;

alter table PREMINULI add constraint FK_STANJE3 foreign key (ID_STAROSTNA_SKUPINA)
references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;

alter table REGIJA add constraint FK_VSEBUJE foreign key (ID_DRZAVA)
references DRZAVA (ID_DRZAVA) on delete restrict on update restrict;

alter table SE_DELI add constraint FK_SE_DELI foreign key (ID_STAROSTNA_SKUPINA)
references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;

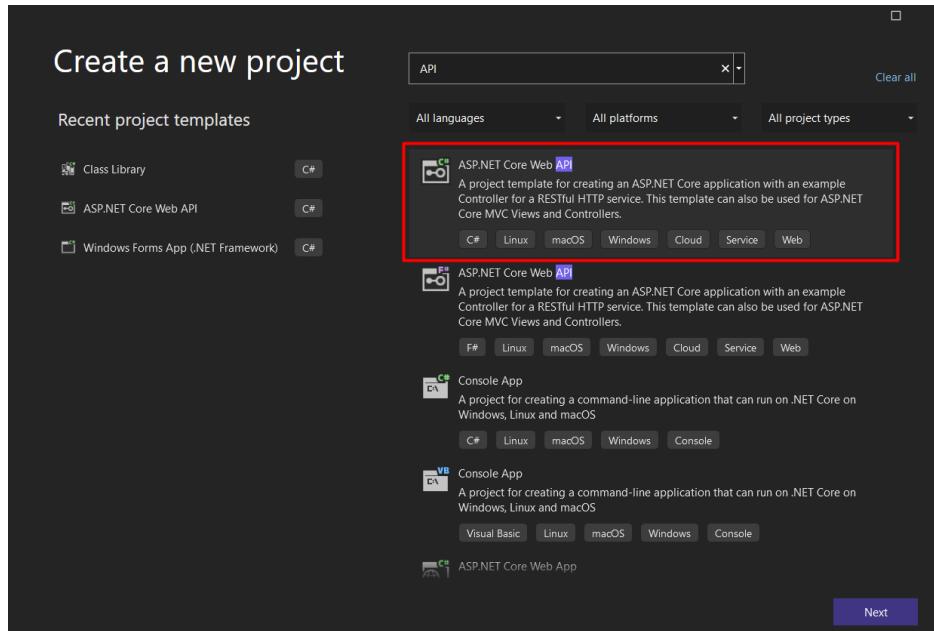
alter table SE_DELI add constraint FK_SE_DELI2 foreign key (ID_BOLNISNICA)
references BOLNISNICA (ID_BOLNISNICA) on delete restrict on update restrict;

alter table STAROSTNA_SKUPINA add constraint FK_SESTAVLJENA_IZ foreign key (ID_DRZAVA)
references DRZAVA (ID_DRZAVA) on delete restrict on update restrict;

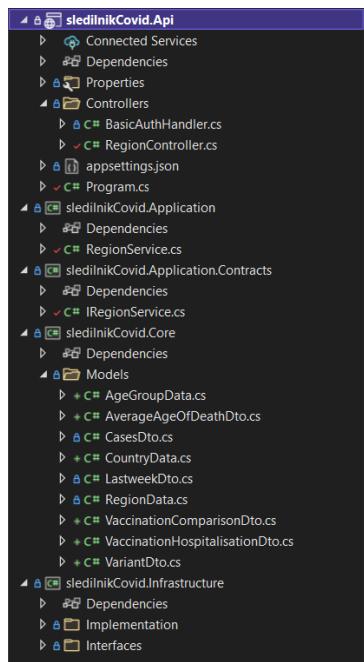
alter table TEST add constraint FK_STANJE foreign key (ID_STAROSTNA_SKUPINA)
references STAROSTNA_SKUPINA (ID_STAROSTNA_SKUPINA) on delete restrict on update restrict;
```

## ASP.NET Web API

Odpremo projekt v Visual Studio in izberemo ASP.NET Core Web API.



Spodnja slika prikazuje zgradbo najinega projekta sledilnikCovid.



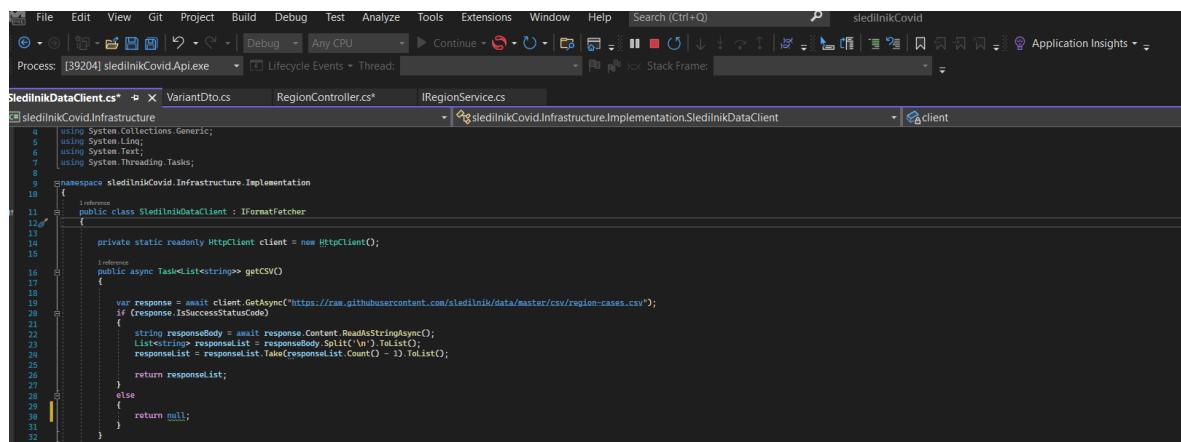
V zaledju sva uporabila pristope Dependency Injection (DI), Separation of Concerns (SOC) in Layer-Structured-Architecture (LSA).

Nivoji na zaledju:

- API (vsebuje controller)
- Application.Contracts (vmesniki poslovne logike)
- Application (implementacija poslovne logike)
- Core (shramba vseh podatkovnih modelov, v našem primeru zgolj DTO-jev)
- Infrastructure (vmesniki in implementacije povezav na zunanje API-je, formatiranje datotečnih struktur)

Uporabila sva tudi pristop Basic authentication (BasicAuth), ki zahteva uporabniško ime in geslo za dostop do endpointov na zaledju. S tem sva onemogočila dostop nepooblaščenim do podatkov.

Spodnja slika prikazuje primer pridobivanja datotečne strukture .csv z uporabo standardne HTTP knjižnice.



```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace sledilnikCovid.Infrastructure.Implementation
{
    interface ISledilnikDataClient : IFormatFetcher
    {
        public async Task<List<string>> getCSV()
        {
            var response = await client.GetAsync("https://raw.githubusercontent.com/sledilnik/data/master/csv/region-cases.csv");
            if (response.IsSuccessStatusCode)
            {
                string responseBody = await response.Content.ReadAsStringAsync();
                List<string> responseList = responseBody.Split('\n').ToList();
                responseList = responseList.Take(responseList.Count - 1).ToList();
            }
            else
            {
                return null;
            }
        }
    }
}
```

Primer pridobivanja imen regij iz nerazčlenjene .csv datoteke.

```
public List<Tuple<string, int>> GetListOfRegions(List<string> header)
{
    List<Tuple<string, int>> regions = new List<Tuple<string, int>>();
    string regionPrev = "", regionNext = "";

    for (int i = 1; i < header.Count; i++)
    {
        string[] splitAttribute = header[i].Split('.');
        regionPrev = splitAttribute[1];

        if (regionNext != regionPrev)
        {
            regions.Add(new Tuple<string, int>(splitAttribute[1], i));
            regionNext = regionPrev;
        }
    }

    return regions;
}
```

Primer pridobivanja podatkov za vsako regijo.

```
public async Task<List<CasesDto>> FetchCases()
{
    List<CasesDto> listCasesDto = new List<CasesDto>();

    List<string> listOfRows = await getCSV();

    List<string> header = listOfRows.ElementAt(0).Split(',').ToList();
    listOfRows = listOfRows.Skip(1).ToList();

    List<Tuple<string, int>> allRegions = GetListOfRegions(header);

    allRegions.RemoveAll(t => t.Item1 == "foreign" || t.Item1 == "unknown");

    //index = 0 je header
    for (int i = 0; i < listOfRows.Count; i++)
    {
        List<string> splitRow = listOfRows[i].Split(',').ToList();

        DateTime currentDate = DateTime.Parse(splitRow[0]);

        List<RegionData> listRegion = new List<RegionData>();

        int index = 0;

        allRegions.ForEach(region => {
            var dac = (splitRow[allRegions[index].Item2] == "") ? 0
                : int.Parse(splitRow[allRegions[index].Item2]);
            var dtd = (splitRow[allRegions[index].Item2 + 2] == "") ? 0
                : int.Parse(splitRow[allRegions[index].Item2 + 2]);
            var fvtd = (splitRow[allRegions[index].Item2 + 3] == "") ? 0
                : int.Parse(splitRow[allRegions[index].Item2 + 3]);
            var svtd = (splitRow[allRegions[index].Item2 + 4] == "") ? 0
                : int.Parse(splitRow[allRegions[index].Item2 + 4]);

            RegionData tempRegion = new RegionData
            {
                RegionName = allRegions[index].Item1,
                DailyActiveCases = dac, //indeks
                DeceasedToDate = dtd, //indeks + 2
                FirstVaccineToDate = fvtd, //indeks + 3
                SecondVaccineToDate = svtd //indeks + 4
            };
            index++;
            listRegion.Add(tempRegion);
        });

        CasesDto temp = new CasesDto
        {
            Date = currentDate,
            Region = listRegion
        };

        listCasesDto.Add(temp);
    }
}

return listCasesDto;
}
```

## GET endpointi za pridobivanje potrebnih podatkov

Prikazuje GET endpoint za slovenia/cases.

S tem pridobimo podatkovni model, ki vsebuje ime regije, dnevno število primerov, število umrlih ter cepljenih s prvim in drugim odmerkom.

```
/// <summary>
/// Vrne seznam z informacijami o podatkih osnovanih na datumu in regiji
/// </summary>
/// <param name="region"></param>
/// <param name="from"></param>
/// <param name="to"></param>
/// <returns>Vrne seznam primerov</returns>
/// <response code="200">Vrne seznam primerov glede na željeno regijo v danem državi</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("slovenia/cases")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<CasesDto>>> GetCases(string? country, string? region, DateTime? from, DateTime? to)
{
    //collect syntax and semantic errors
    var validRegions = new List<string> {"lj", "ce", "kr", "nm", "lk", "kp", "mb", "ns", "ng", "po", "sg", "za"};
    var exceptionStack = new List<string>();
    Datetime ddate;

    if (region != null && !validRegions.Contains(region))
        exceptionStack.Add("Invalid region.");
    if (from > to)
        exceptionStack.Add("From date cannot be later than To date");
    //evaluate syntax and semantic errors
    if (exceptionStack.Count == 0)
    {
        var data = await _regionService.FetchDataCases(region, from, to);
        return Ok(data);
    }
    else
        return BadRequest(exceptionStack);
}
```

Prikazuje GET endpoint za slovenia/lastweek.

Vrne seznam vseh regij z vsoto aktivnih primerov v zadnjem tednu.

```
/// <summary>
/// Vrne seznam vseh regij z vsoto aktivnih primerov v zadnjem tednu
/// </summary>
/// <returns>Seznam danih vseh regij glede na regijo</returns>
/// <response code="200">Vrne seznam regij z njihovo korespondečno vsoto</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("slovenia/lastweek")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<LastweekDto>>> GetLastweek()
{
    var data = await _regionService.FetchDataLastWeek();
    return Ok(data);
}
```

Prikazuje GET endpoint za slovenia/vaccinationComparison.

Vrne podatke o številu smrti, najvišje dnevno število smrti, število okuženih (pozitivnih) in število cepljenih v Sloveniji v obdobju analiziranega 2. in 4. vala.

```
/// <summary>
/// Vrne podatke o številu smrti, najvišje dnevno število smrti, število okuženih (pozitivnih) in število cepljenih v Sloveniji v obdobju analiziranega 2. in 4. vala
/// </summary>
/// <returns>Objekt VaccinationComparisonOto</returns>
/// <response code="200">Vrne ustrezne podatke</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("slovenia/vaccinationComparison")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<VaccinationComparisonOto>>> GetVaccinationComparison(DateTime fromDateWave1, DateTime fromDateWave2, DateTime toDateWave1, DateTime toDateWave2)
{
    var data = await _regionService.FetchVaccinationComparison(fromDateWave1, fromDateWave2, toDateWave1, toDateWave2);
    return Ok(data);
}
```

Prikazuje GET endpoint za slovenia/averageAgeOfDeath.

Vrne število umrlih, število cepljenih, število okuženih (pozitivnih) po starostnih skupinah v obdobju analiziranega 2. in 4. vala.

```
/// <summary>
/// Vrne število umrlih, število cepljenih, število okuženih (pozitivnih) po starostnih skupinah v obdobju analiziranega 2. in 4. vala
/// </summary>
/// <returns>Seznam objektov AverageAgeOfDeathDto</returns>
/// <response code="200">Vrne ustrezne podatke</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("slovenia/averageAgeOfDeath")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<AverageAgeOfDeathDto>>> Get(DateTime fromDateWave1, DateTime fromDateWave2, DateTime toDateWave1, DateTime toDateWave2)
{
    var data = await _regionService.FetchAverageAgeOfDeath(fromDateWave1, fromDateWave2, toDateWave1, toDateWave2);
    return Ok(data);
}
```

Prikazuje GET endpoint za europe/vaccinationHospitalisation.

Vrne 3 evropske države, kjer je precepljenost prebivalstva najvišja in 3 evropske države, kjer je precepljenost najnižja ter pri le-teh vrne delež hospitalizacij na intenzivni negi.

```
/// <summary>
/// Vrne 3 evropske države, kjer je precepljenost prebivalstva najvišja in 3 evropske države, kjer je precepljenost najnižja ter pri le-teh vrne delež hospitalizacij na intenzivni negi
/// </summary>
/// <returns>Seznam objektov VaccinationHospitalisation</returns>
/// <response code="200">Vrne ustrezne podatke</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("europe/vaccinationHospitalisation")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<VaccinationHospitalisation>>> GetVaccinationHospitalisation()
{
    var data = await _regionService.FetchVaccinationHospitalisation();
    return Ok(data);
}
```

Prikazuje GET endpoint za slovenia/variantInfectionLevel.

Vrne število okuženih, hospitalizacij na navadni in intenzivni negi ter število smrti v obdobju, ko je prevladovala originalna in delta različica.

```
/// <summary>
/// Vrne število okuženih, hospitalizacij na navadni in intenzivni negi ter število smrti v obdobju, ko je prevladovala originalna in delta različica
/// </summary>
/// <returns>Seznam objektov VaccinationHospitalisation</returns>
/// <response code="200">Vrne ustrezne podatke</response>
/// <response code="400">Vrne izjemo glede na neustrezne vhodne podatke uporabnika</response>
[HttpGet]
[Route("slovenia/variantInfectionLevel")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
public async Task<ActionResult<List<VariantDto>>> GetVariantInfectionLevel()
{
    var data = await _regionService.FetchVariantInfectionLevel();
    return Ok(data);
}
```

## Swagger

Testiranje GET endpoint-a /slovenia/cases v Swagger-ju.

The screenshot shows the Swagger UI interface for testing a GET endpoint. At the top, there are three input fields for query parameters:

- region: "lj"
- from: "1/10/2021"
- to: "3/10/2021"

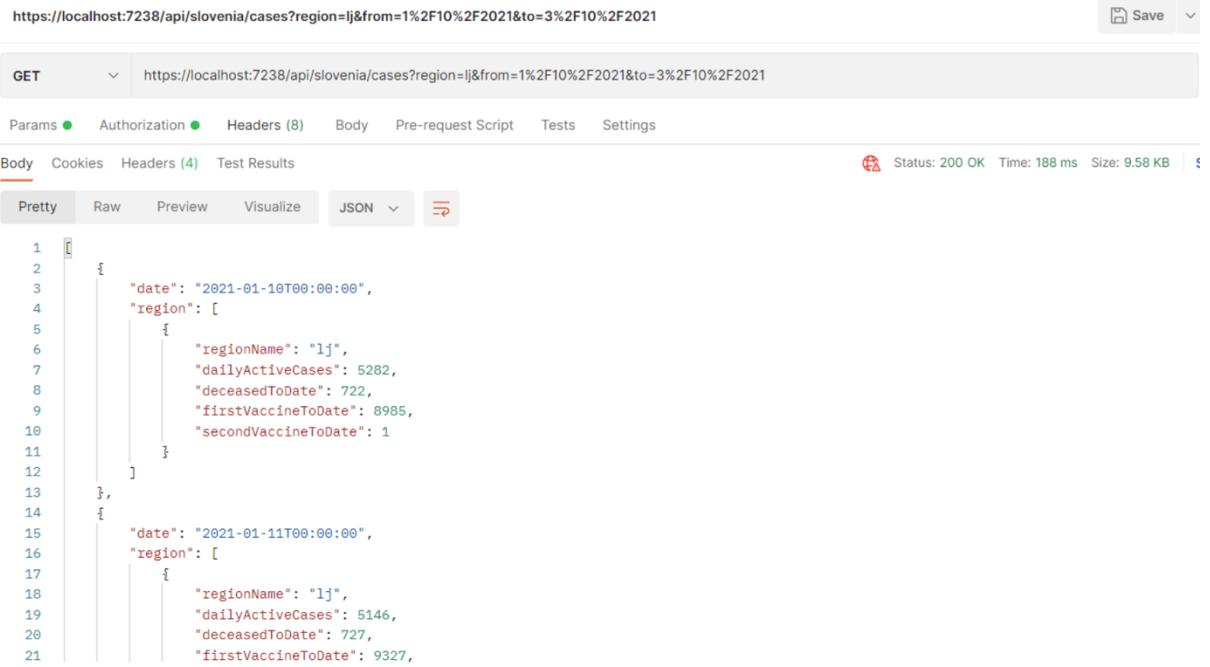
Below these fields are two buttons: "Execute" (in blue) and "Clear".

Under the "Responses" section, there are three tabs: "Curl", "Request URL", and "Server response".

- "Curl": Shows the command: curl -X 'GET' \ 'https://localhost:7238/api/slovenia/cases?region=lj&from=1%2F10%2F2021&to=3%2F10%2F2021' \H 'accept: application/json'
- "Request URL": Shows the URL: https://localhost:7238/api/slovenia/cases?region=lj&from=1%2F10%2F2021&to=3%2F10%2F2021
- "Server response": Shows the HTTP status code 200 and the response body as JSON. The JSON array contains three objects, each representing a date range and a region (lj). The first object is for the date range "2021-01-10T00:00:00" to "2021-01-11T00:00:00". The second object is for the date range "2021-01-11T00:00:00" to "2021-01-12T00:00:00". The third object is for the date range "2021-01-12T00:00:00" to "2021-01-13T00:00:00". Each object has a "region" field containing "lj" and a "dailyActiveCases" field with values 5282, 5146, and 5146 respectively. Other fields like "deceasedToDate", "firstVaccineToDate", and "secondVaccineToDate" are also present.

## Postman

Testiranje GET endpoint /slovenia/cases v Postman-u.



The screenshot shows the Postman interface with the following details:

- URL: <https://localhost:7238/api/slovenia/cases?region=lj&from=1%2F10%2F2021&to=3%2F10%2F2021>
- Status: 200 OK
- Time: 188 ms
- Size: 9.58 KB
- Method: GET
- Headers (8) selected
- Body tab selected
- JSON response (Pretty) displayed:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
{
  "date": "2021-01-10T00:00:00",
  "region": [
    {
      "regionName": "lj",
      "dailyActiveCases": 5282,
      "deceasedToDate": 722,
      "firstVaccineToDate": 8985,
      "secondVaccineToDate": 1
    }
  ],
  {
    "date": "2021-01-11T00:00:00",
    "region": [
      {
        "regionName": "lj",
        "dailyActiveCases": 5146,
        "deceasedToDate": 727,
        "firstVaccineToDate": 9327,
      }
    ]
  }
}
```

## Oddaljen strežnik

Za oddaljen strežnik sva uporabila Microsoft Azure, kamor sva postavila zaledje, relacijsko podatkovno bazo in prožilec.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a blue header bar with the 'Microsoft Azure' logo, a search bar, and various navigation icons. On the right side of the header, it shows the user's email 'eb2044@student.uni-lj.si' and the text 'UNIVERZA V LJUBLJANI' next to a small profile picture.

The main content area starts with a 'Welcome to Azure!' message and a note about not having a subscription, followed by three promotional cards:

- Start with an Azure free trial**: Get \$200 free credit toward Azure products and services, plus 12 months of popular free services. Buttons: [Start](#) and [Learn more](#).
- Manage Azure Active Directory**: Manage access, set smart policies, and enhance security with Azure Active Directory. Buttons: [View](#) and [Learn more](#).
- Access student benefits**: Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status. Buttons: [Explore](#) and [Learn more](#).

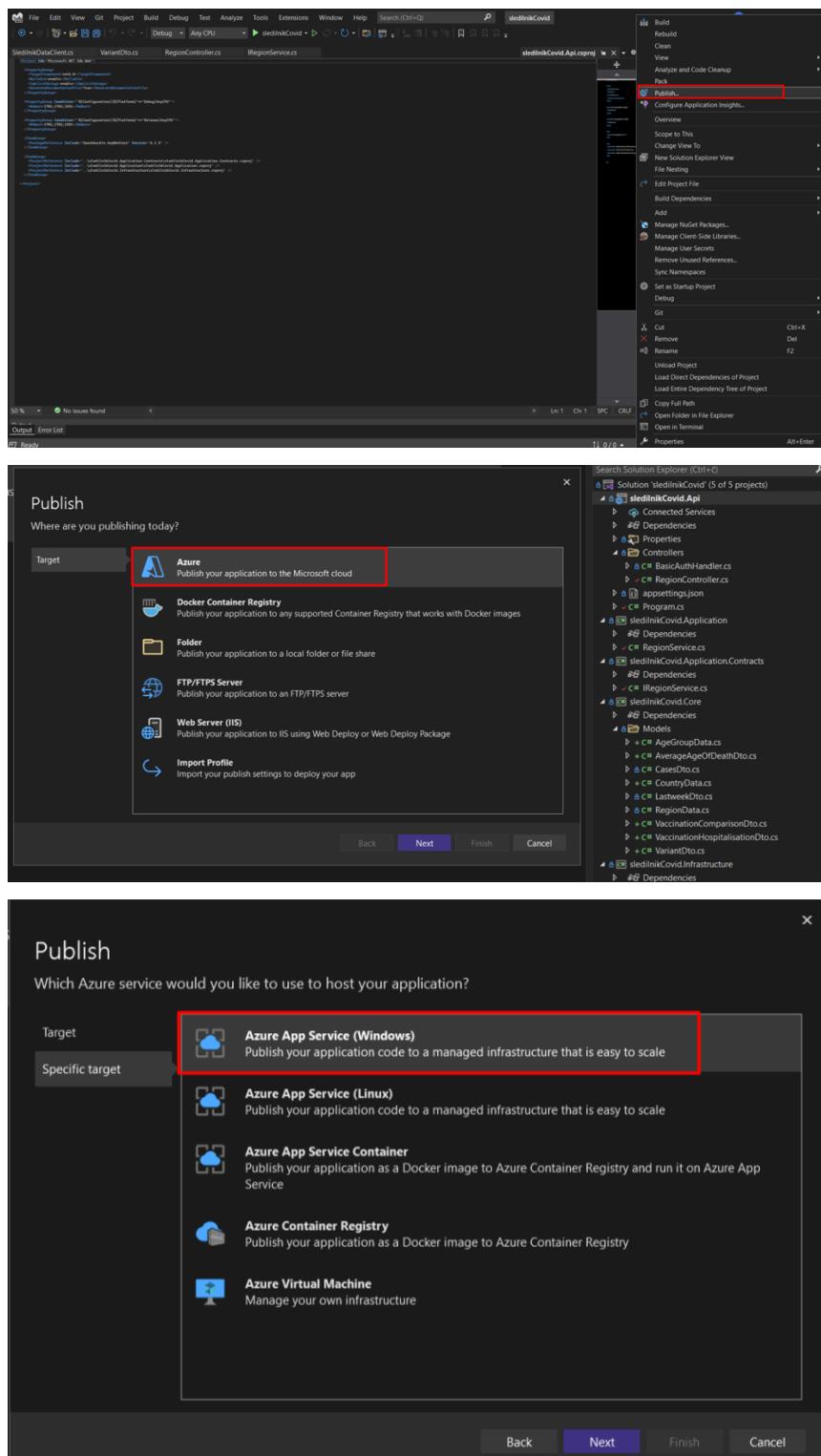
Below these cards is a section titled 'Azure services' containing icons for creating resources, quickstart center, virtual machines, app services, storage accounts, SQL databases, Azure Cosmos DB, Kubernetes services, function apps, and a link to 'All services'.

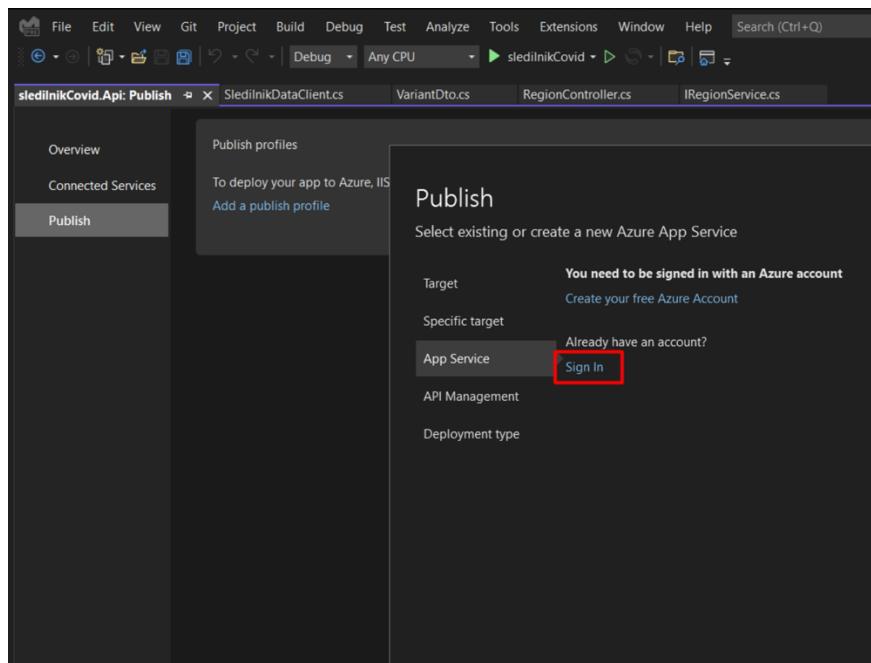
Under 'Navigate', there are links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'.

At the bottom left, there's a 'Tools' section.

## Zaledje

### Koraki za postavitev zaledja na oddaljen strežnik Microsoft Azure





Naslednjih korakov ne prikaževo zaradi osebnih podatkov.

## Podatkovna baza

### Koraki za postavitev podatkovne baze na oddaljen strežnik Microsoft Azure

Home > Create a resource >

#### Azure Database for MySQL Microsoft



#### Azure Database for MySQL Add to Favorites

Microsoft

★ 3.7 (200 Azure ratings)

[Create](#)

[Overview](#) [Plans](#) [Usage Information + Support](#) [Reviews](#)

Azure Database for MySQL is a MySQL database service built on Microsoft's scalable cloud infrastructure for application developers. Leverage your existing open-source MySQL skills and tools and scale on-the-fly without downtime to efficiently deliver existing and new applications with reduced operational overhead. Built-in features maximize performance, availability, and security. Azure Database for MySQL empowers developers to focus on application innovation instead of database management tasks.

Home > Azure Database for MySQL servers >

#### Select Azure Database for MySQL deployment option Microsoft

Microsoft

##### How do you plan to use the service?



###### Flexible server

Best for production workloads that require zone resiliency, predictable performance, maximum control with IOPs scaling, custom maintenance window, cost optimization controls and simplified developer experience.

[Create](#)

[Learn More ↗](#)



###### Single server

Best for existing applications already leveraging single server. Designed for basic database management functions, such as patching, backups and zonal high availability with minimal user configuration.

[Create](#)

[Learn More ↗](#)

## Flexible server

Microsoft

### Basics (Change)

Subscription	Azure for Students
Resource group	FRI
Server name	coviddata
Server admin login name	evabizilj
Location	West Europe
Availability zone	No preference
High availability	Not Enabled
MySQL version	5.7
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 GiB storage
Backup retention period (in days)	7 day(s)
Storage Auto-growth	Enabled
Geo-redundancy	Not Enabled

### Networking (Change)

Connectivity method	Public access (allowed IP addresses)
Allow public access from any Azure service within Azure to this server	No
Firewall rules	0
SSL/TLS	SSL is enforced and TLS version is 1.2. This can be changed after server is created. <a href="#">Learn more</a>

### Tags (Change)

Create

< Previous

Download a template for automation

Home >



MySqlFlexibleServer\_932ceb0d122f4dbeb0f22cb778950eea | Overview

X

Search (Cmd+/) < Delete Cancel Redeploy Refresh

Overview Inputs Outputs Template

Deployment is in progress

Deployment name: MySqlFlexibleServer\_932ceb0d122f4dbeb0f22cb778950eea Start time: 12/19/2021, 9:56:51 PM  
Subscription: Azure for Students Correlation ID: 8b964405-55fa-4d59-8c29-e90645853851  
Resource group: FRI

Deployment details (Download)

Resource	Type	Status	Operation details
coviddata	Microsoft.DBforMySQL/flexibleServers	Accepted	<a href="#">Operation details</a>

## coviddata

Azure Database for MySQL flexible server

Search (Cmd+/)	<	Delete	Reset password	Restore	Restart	Stop	Refresh	Feedback
Overview								
Activity log								
Access control (IAM)								
Tags								
Diagnose and solve problems								
Settings								

Subscription : Azure for Students

Server name : coviddata.mysql.database.azure.com

Subscription ID : 73715ce1-98fe-4fd7-bbc1-dae9b400d6a4

Server admin login name : evabizilj

Resource group : FRI

Configuration : Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 GiB storage

Status : Available

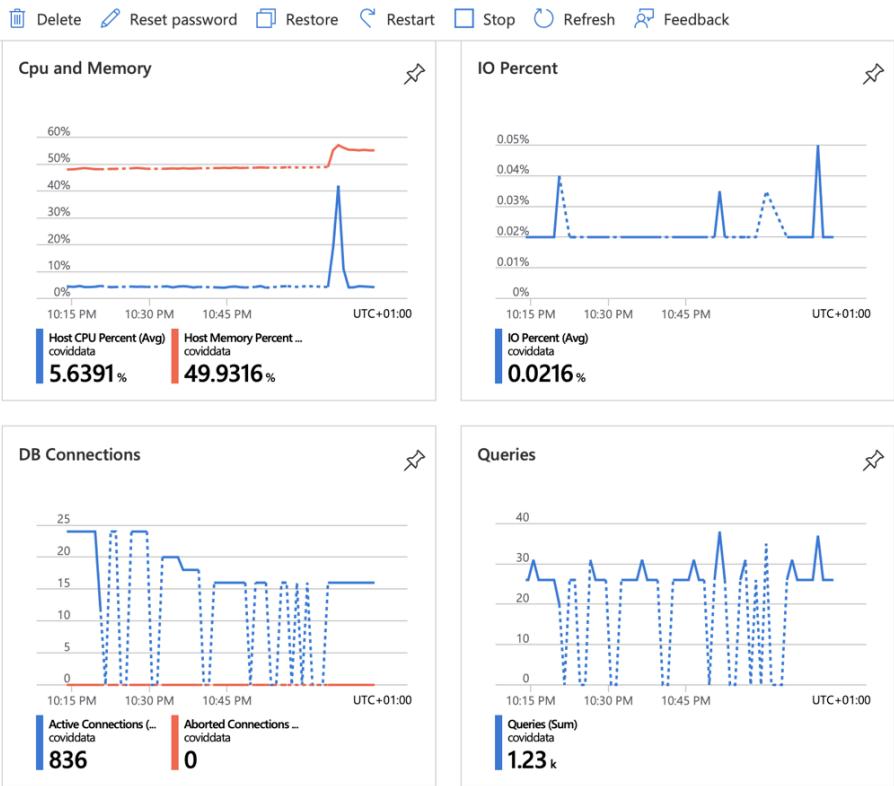
MySQL version : 5.7

Location : West Europe

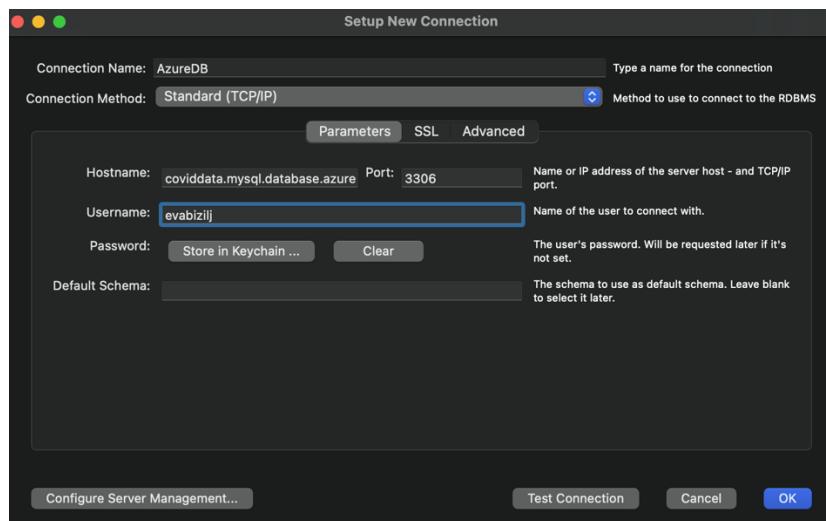
Availability zone : 1

Tags (Edit) : [Click here to add tags](#)

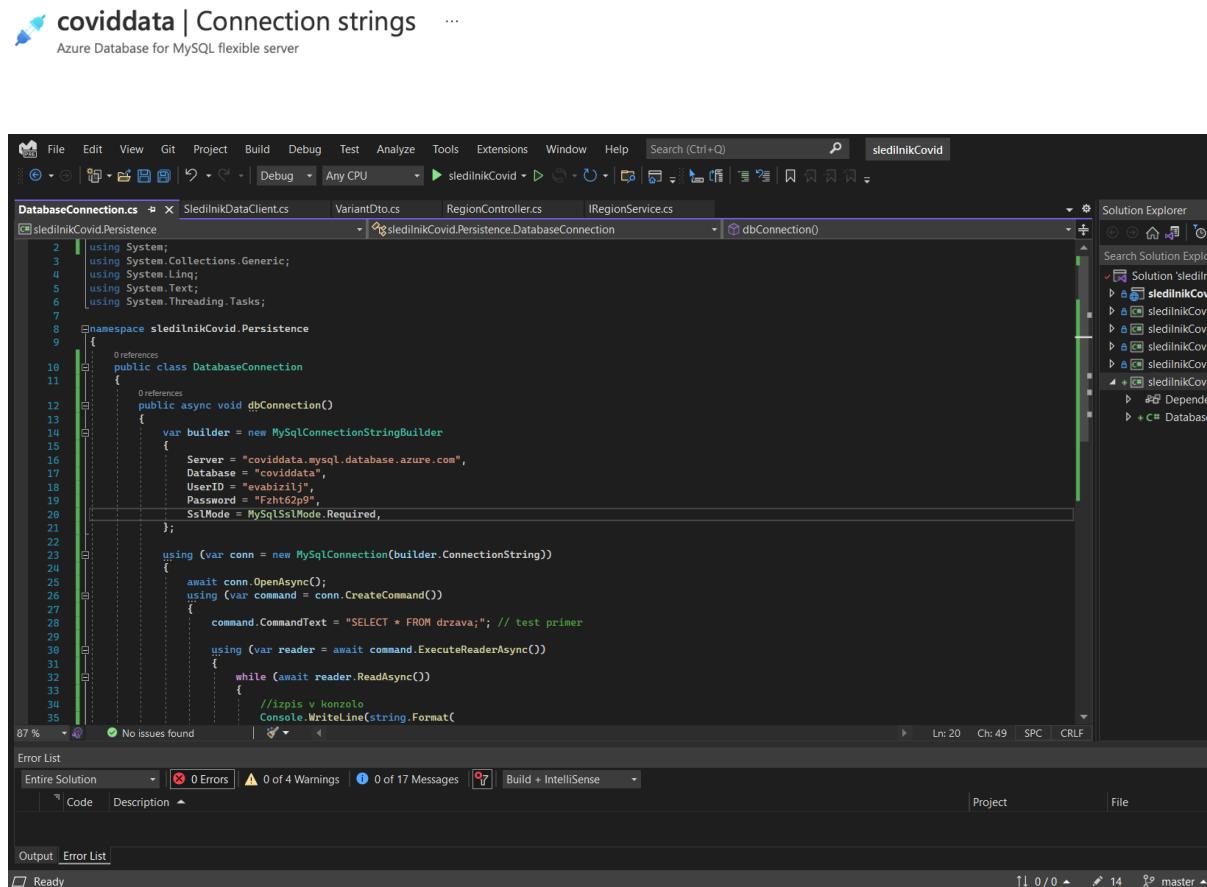
JSON View



Z MySQL Workbench se povežemo na AzureDB (Azure Database for MySQL - Flexible Server), da bomo lahko dostopali do podatkov, ki jih bomo pridobili iz podatkovnih zbirk.



Na AzureDB se povežemo še z .NET (C#) in sicer tako, da uporabimo Connection String.



The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** Shows "sledilnikCovid" as the active project.
- Solution Explorer:** Displays the project structure under "sledilnikCovid".
- Toolbox:** Standard Visual Studio toolbox icons.
- Code Editor:** The file "DatabaseConnection.cs" is open, showing C# code for connecting to an Azure MySQL database. The code uses a MySQLConnectionStringBuilder to define connection parameters like Server, Database, UserID, Password, and SslMode.
- Error List:** Shows "No issues found".
- Output:** Shows "Ready".
- Status Bar:** Shows "Ln: 20 Ch: 49 SPC CRLF".

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace sledilnikCovid.Persistence
8 {
9     public class DatabaseConnection
10    {
11        public async void dbConnection()
12        {
13            var builder = new MySqlConnectionStringBuilder
14            {
15                Server = "coviddata.mysql.database.azure.com",
16                Database = "coviddata",
17                UserID = "evabizily",
18                Password = "Fzht629",
19                SslMode = MySqlSslMode.Required,
20            };
21
22            using (var conn = new MySqlConnection(builder.ConnectionString))
23            {
24                await conn.OpenAsync();
25                using (var command = conn.CreateCommand())
26                {
27                    command.CommandText = "SELECT * FROM drzava"; // test primer
28
29                    using (var reader = await command.ExecuteReaderAsync())
30                    {
31                        while (await reader.ReadAsync())
32                        {
33                            //izpis v konzolo
34                            Console.WriteLine(string.Format(
35                                ));
```

## Prožilec

Prožilec na vsakih 6 ur proži GET endpoint metodo v kontrolerju znotraj ASP.NET Web API-ja. Nato se izvede endpoint metoda, ki pridobi podatke iz podatkovnih zbirk preko zunanjega API klica. Podatki se nato ustrezno razčlenijo in obdelajo ter v primerni obliki pošljejo na podatkovno bazo.

```
import {CREDENTIALS} from '~/constants.utils.js'

const cron = require("node-cron");
const axios = require("axios");
const port = 3000;

const BASE_URL = "https://coviddata.mysql.database.azure.com/api/Region";

app.listen(port, () => {
  console.log(`Success! Aplikacija se izvaja na portu: ${port}.`);
});

const authTrigger = async () => {
  await axios.basicAuth(`.../auth`, {CREDENTIALS});
};

// Nastavimo CRON jobs, da se izvedejo vsakih 6 ur periodično.
// Vsi parametri so NULLABLE, torej niso potrebni za izvedbo zaledne procedure API-ja

// Vrne vse primere z informacijami o podatkih osnovanih na datumu in regiji
export cron.schedule("0 */6 * * *", const fetchCases = async (country, region, from, to) => {
  const response = await axios.get(`${BASE_URL}/cases?
  country=${country}&
  region=${region}&
  from=${from}&
  to=${to}
  `);
  const data = response.data;
  return data;
});

// Vrne seznam vseh regij z vsoto aktivnih primerov v zadnjem tednu
export cron.schedule("0 */6 * * *", const fetchLastweek = async () => {
  const response = await axios.get(`${BASE_URL}/lastweek`);
  const data = response.data;
  return data;
});

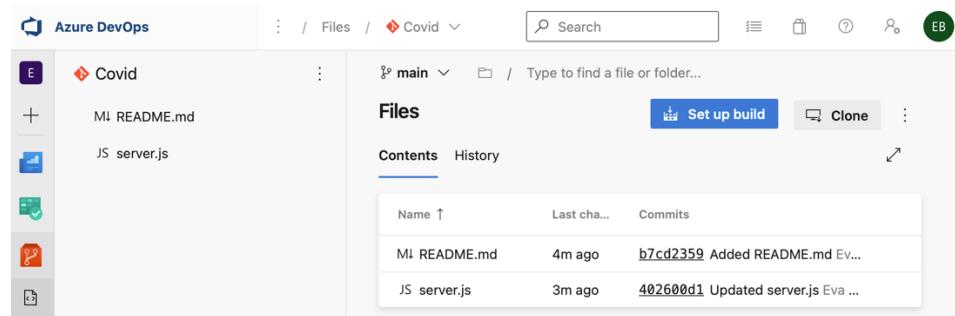
// Vrne podatke o številu smrtev, najvišje dnevno število smrtev, število okuženih in število cep. v SLO
export cron.schedule("0 */6 * * *", const fetchVaccinationComparison = async (from1, from2, to1, to2) => {
  const response = await axios.get(`${BASE_URL}/vaccinationComparison?
  fromDateWave1=${from1}&
  fromDateWave2=${from2}&
  toDateWave1=${to1}&
  toDateWave2=${to2}
  `);
  const data = response.data;
  return data;
});

// Vrne število umrlih, število cepljenih, število okuženih po starostnih skupinah v obdobju 2. in 4. vala
export cron.schedule("0 */6 * * *", const fetchAverageAgeOfDeath = async (from1, from2, to1, to2) => {
  const response = await axios.get(`${BASE_URL}/averageAgeOfDeath?
  fromDateWave1=${from1}&
  fromDateWave2=${from2}&
  toDateWave1=${to1}&
  toDateWave2=${to2}
  `);
  const data = response.data;
  return data;
});

// Vrne 3 evropske države, kjer je precepljenost najvišja in 3, kjer je najnižja
export cron.schedule("0 */6 * * *", const fetchVaccinationHospitalisation = async () => {
  const response = await axios.get(`${BASE_URL}/vaccinationHospitalisation`);
  const data = response.data;
  return data;
});

// Vrne št. okuženih, hospitalizacij na navadni in intenzivni negi ter št. smrtev v obdobju DELTA variante
export cron.schedule("0 */6 * * *", const fetchVariantInfectionLevel = async () => {
  const response = await axios.get(`${BASE_URL}/variantInfectionLevel`);
  const data = response.data;
  return data;
});
```

## Koraki za postavitev prožilca na oddaljen strežnik Microsoft Azure



## Create Function App ...

Basics    Hosting    Networking (preview)    Monitoring    Tags    Review + create

### Summary



### Details

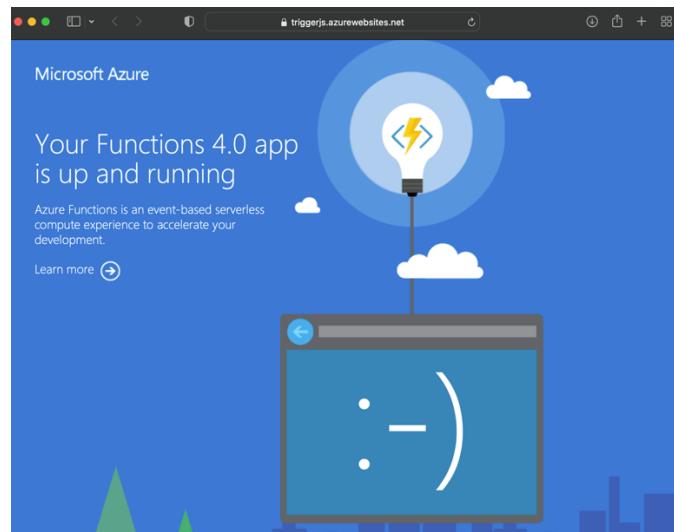
Subscription	73715ce1-98fe-4fd7-bbc1-dae9b400d6a4
Resource Group	Covid
Name	triggerjs
Runtime stack	Node.js 14 LTS

### Hosting

#### Storage (New)

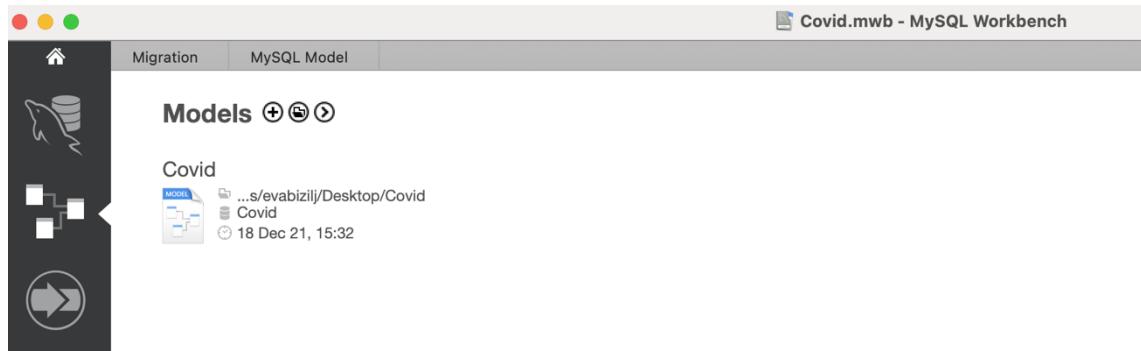
Storage account storageaccountcovidac80

Na koncu pa še preverimo delovanje prožilca.



## Relacijska podatkovna baza (MySQL)

Preden sva vzpostavila podatkovno bazo na oddaljenem strežniku Microsoft Azure, sva jo testirala na lokalnem strežniku (*localhost*). Pri tem sva uporabila MySQL Workbench.

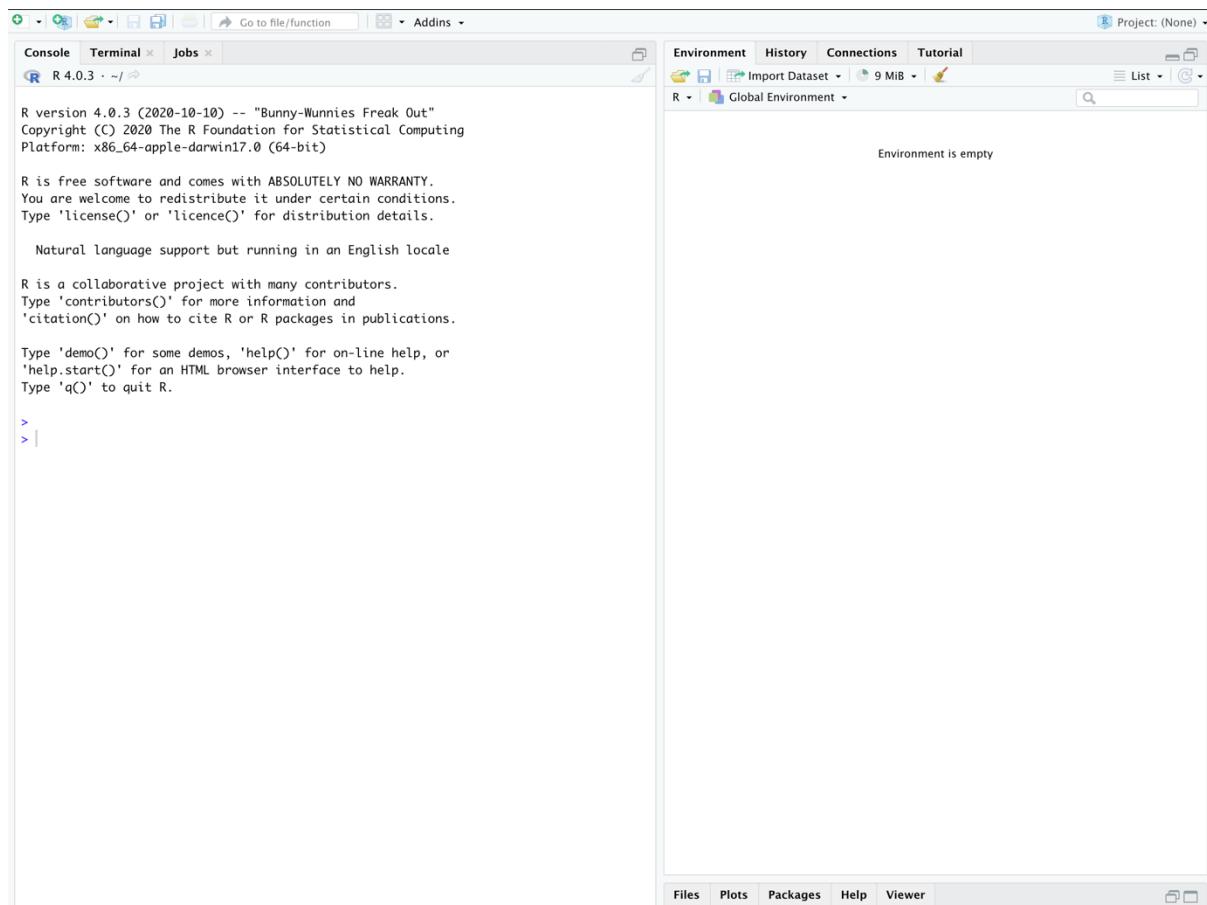


A detailed screenshot of the MySQL Workbench interface. On the left, the 'SCHEMAS' panel shows the 'bookstore' and 'Covid' databases, with 'Covid' expanded to show its tables: BOLNISNIKA, CELINA, CEPIVO, DATUM, and DRZAVA. The 'DRZAVA' table is selected. The 'Object Info' panel shows the table structure for DRZAVA, including columns ID\_DRZAVA, ID\_CELINA, IME\_DRZAVA, STEVILLO\_PREBIVALCEV, and STEVILLO\_OKUZENIH. On the right, a query editor window displays a SELECT statement: 'SELECT ID\_CELINA, IME\_DRZAVA, STEVILLO\_PREBIVALCEV FROM Covid.DRZAVA'. Below the query editor is a 'Result Grid' showing the results of the query. The results are as follows:

## RStudio

Za analizo podatkov sva uporabila programski jezik R in njegovo knjižnico ggplot2 za prikaz grafov in nekaterih bolj naprednih izračunov.

RStudio sva uporabila za delovno okolje, ker je uporabniku najbolj prijazen.



Preden sva podatke vnesla v RStudio so le-ti prešli fazo predprocesiranja (SQL poizvedbe). Podatke pridobljene s SQL poizvedbami sva shranila v .csv datoteko.

Ker ima R precej težav pri dekodiranju .csv datotek (npr. težavno razčlenjevanje atributnih vrednosti v vrsticah zaradi neustreznih delimiterjev), sva datoteko naprej pretvorila v .xlsx format in jo nato v tem formatu tudi odprla.

# Poizvedbe in rezultati

S poizvedbami nad podatkovno bazo sva testirala najine predpostavke.

## Predpostavka 1

Cepljenje proti Covid-19 zmanjša število smrti.

### Poizvedbe

1. Vrne število smrti v obdobju analiziranega 2. vala v Sloveniji.

```
SELECT SUM(p.STEVILO_PREMINULIH) as STEVILO_SMRTI_SLOVENIJA_2
FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-11-01'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC1.csv ×
1 STEVILO_SMRTI_SLOVENIJA_2
2 3184
```

2. Vrne število smrti v obdobju analiziranega 4. vala v Sloveniji.

```
SELECT SUM(p.STEVILO_PREMINULIH) as STEVILO_SMRTI_SLOVENIJA_4 FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2021-09-12'
        AND d.DATUM <= '2021-12-12'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC2.csv ×
1 STEVILO_SMRTI_SLOVENIJA_4
2 940
```

3. Vrne najvišje dnevno število smrti v obdobju analiziranega 2. vala v Sloveniji.

```
SELECT MAX(p.STEVILO_PREMINULIH) as MAX_SMRTI_SLOVENIJA_2
FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-11-01'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

	poizvedbaC3.csv	X
1	MAX_SMRTI_SLOVENIJA_2	
2	66	

4. Vrne najvišje dnevno število smrti v obdobju analiziranega 4. vala v Sloveniji.

```
SELECT MAX(p.STEVILO_PREMINULIH) as MAX_SMRTI_SLOVENIJA_4
FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2021-09-12'
        AND d.DATUM <= '2021-12-12'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC4.csv ×
1 MAX_SMRTI_SLOVENIJA_4
2 24
```

5. Vrne število okuženih (pozitivnih) v obdobju analiziranega 2. vala v Sloveniji.

```
SELECT SUM(ts.STEVILO_TESTIRANIH) as STEVILO_POZITIVNIH_SLOVENIJA_2
FROM TEST ts
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-11-01'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA
WHERE ts.REZULTAT_TEST = "positive";
```

Rezultat

```
poizvedbaC5.csv ×
1 STEVILO_POZITIVNIH_SLOVENIJA_2
2 132767
```

6. Vrne število okuženih (pozitivnih) v obdobju analiziranega 4. vala v Sloveniji.

```
SELECT SUM(ts.STEVILO_TESTIRANIH) as STEVILO_POZITIVNIH_SLOVENIJA_4
FROM TEST ts
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2021-09-12'
        AND d.DATUM <= '2021-12-12'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA
WHERE ts.REZULTAT_TEST = "positive";
```

### Rezultat

```
poizvedbaC6.csv ×
1 STEVILO_POZITIVNIH_SLOVENIJA_4
2 164434
```

7. Vrne število cepljenih od začetka epidemije do konca analiziranega 2. vala v Sloveniji.

```
SELECT SUM(c.STEVILO_CEPLJENIH) as STEVILO_CEPLJENIH_SLOVENIJA_2
FROM CEPIVO c
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-04-03'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON c.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

poizvedbaC7.csv X

1	STEVIL0_CEPLJENIH_SLOVENIJA_2
2	27521

8. Vrne število cepljenih od začetka epidemije do konca analiziranega 4. vala v Sloveniji.

```
SELECT SUM(c.STEVILO_CEPLJENIH) as STEVILO_CEPLJENIH_SLOVENIJA_4
FROM CEPIVO c
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-04-03'
        AND d.DATUM <= '2021-12-12'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON c.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC8.csv ×
1 STEVILO_CEPLJENIH_SLOVENIJA_4
2 1176493
```

## **Rezultati**

<b>Poizvedba</b>	<b>Rezultat</b>
število smrti v obdobju analiziranega 2. vala v Sloveniji	3184
število smrti v obdobju analiziranega 4. vala v Sloveniji	940
najvišje dnevno število smrti v obdobju analiziranega 2. vala v Sloveniji	66
najvišje dnevno število smrti v obdobju analiziranega 4. vala v Sloveniji	24
število okuženih (pozitivnih) v obdobju analiziranega 2. vala v Sloveniji	132767
število okuženih (pozitivnih) v obdobju analiziranega 4. vala v Sloveniji	164434
število cepljenih od začetka epidemije do konca analiziranega 2. vala v Sloveniji	27521
število cepljenih od začetka epidemije do konca analiziranega 4. vala v Sloveniji	1176493

## Predpostavka 2

Povprečna starost umrlih se je 4. valu v primerjavi z 2. valom zmanjšala.

### Poizvedbe

9. Vrne število umrlih po starostnih skupinah v obdobju analiziranega 2. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(p.STEVILO_PREMINULIH) as STEVILO_SMRTI_SLOVENIJA_2
FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-11-01'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC9.csv ×
1 IME_STAROSTNA_SKUPINA,STEVILLO_SMRTI_SLOVENIJA_2
2 sk_0-4,0
3 sk_5-14,0
4 sk_15-24,0
5 sk_25-34,2
6 sk_35-44,17
7 sk_45-54,34
8 sk_55-64,127
9 sk_65-74,469
10 sk_75-84,1139
11 sk_85+,1665
```

10. Vrne število umrlih po starostnih skupinah v obdobju analiziranega 4. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(p.STEVILO_PREMINULIH) as STEVILO_SMRTI_SLOVENIJA_4
FROM PREMINULI p
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2021-09-12'
        AND d.DATUM <= '2021-12-12'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON p.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

Rezultat

```
poizvedbaC10.csv ×
1 IME_STAROSTNA_SKUPINA,STEVILO_SMRTI_SLOVENIJA_4
2 sk_0-4,0
3 sk_5-14,0
4 sk_15-24,1
5 sk_25-34,0
6 sk_35-44,8
7 sk_45-54,21
8 sk_55-64,87
9 sk_65-74,194
10 sk_75-84,332
11 sk_85+,357
```

11. Vrne število cepljenih po starostnih skupinah od začetka epidemije do konca analiziranega 2. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(c.STEVILO_CEPLJENIH) as  
STEVILO_CEPLJENIH_SLOVENIJA_2  
FROM CEPIVO c  
JOIN (  
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA  
    FROM DRZAVA dr  
    JOIN (  
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA  
        FROM DATUM d  
        JOIN STAROSTNA_SKUPINA ss  
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA  
        WHERE d.DATUM >= '2020-04-03'  
        AND d.DATUM <= '2021-02-01'  
    ) t1  
    ON dr.ID_DRZAVA = t1.ID_DRZAVA  
    WHERE dr.IME_DRZAVA = "Slovenia"  
) t  
ON c.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA  
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC11.csv ×  
1 IME_STAROSTNA_SKUPINA,STEVILO_CEPLJENIH_SLOVENIJA_2  
2 sk_0-4,0  
3 sk_5-14,0  
4 sk_15-24,421  
5 sk_25-34,3136  
6 sk_35-44,4599  
7 sk_45-54,5621  
8 sk_55-64,5778  
9 sk_65-74,2247  
10 sk_75-84,2254  
11 sk_85+,3465
```

12. Vrne število cepljenih po starostnih skupinah od začetka epidemije do konca analiziranega 4. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(c.STEVILO_CEPLJENIH) as  
STEVILO_CEPLJENIH_SLOVENIJA_4  
FROM CEPIVO c  
JOIN (  
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA  
    FROM DRZAVA dr  
    JOIN (  
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA  
        FROM DATUM d  
        JOIN STAROSTNA_SKUPINA ss  
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA  
        WHERE d.DATUM >= '2020-04-03'  
        AND d.DATUM <= '2021-12-12'  
    ) t1  
    ON dr.ID_DRZAVA = t1.ID_DRZAVA  
    WHERE dr.IME_DRZAVA = "Slovenia"  
) t  
ON c.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA  
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

### Rezultat

```
poizvedbaC12.csv ×  
1 IME_STAROSTNA_SKUPINA,STEVILO_CEPLJENIH_SLOVENIJA_4  
2 sk_0-4,0  
3 sk_5-14,32857  
4 sk_15-24,78604  
5 sk_25-34,122651  
6 sk_35-44,169034  
7 sk_45-54,194272  
8 sk_55-64,213260  
9 sk_65-74,199340  
10 sk_75-84,114772  
11 sk_85+,45555
```

13. Vrne število okuženih (pozitivnih) po starostnih skupinah analiziranega 2. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(ts.STEVILO_TESTIRANIH) as STEVILO_POZITIVNIH_SLOVENIJA_2
FROM TEST ts
JOIN (
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA
    FROM DRZAVA dr
    JOIN (
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA
        FROM DATUM d
        JOIN STAROSTNA_SKUPINA ss
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA
        WHERE d.DATUM >= '2020-11-01'
        AND d.DATUM <= '2021-02-01'
    ) t1
    ON dr.ID_DRZAVA = t1.ID_DRZAVA
    WHERE dr.IME_DRZAVA = "Slovenia"
) t
ON ts.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA
WHERE ts.REZULTAT_TEST = "positive"
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

### Rezultat

	IME_STAROSTNA_SKUPINA, STEVILO_POZITIVNIH_SLOVENIJA_2
1	IME_STAROSTNA_SKUPINA, STEVILO_POZITIVNIH_SLOVENIJA_2
2	sk_0-4, 785
3	sk_5-14, 1781
4	sk_15-24, 10307
5	sk_25-34, 21632
6	sk_35-44, 25397
7	sk_45-54, 25961
8	sk_55-64, 19635
9	sk_65-74, 11118
10	sk_75-84, 8985
11	sk_85+, 7166

14. Vrne število okuženih (pozitivnih) po starostnih skupinah analiziranega 4. vala v Sloveniji.

```
SELECT t.IME_STAROSTNA_SKUPINA, SUM(ts.STEVILO_TESTIRANIH) as  
STEVILO_POZITIVNIH_SLOVENIJA_4  
FROM TEST ts  
JOIN (  
    SELECT t1.ID_STAROSTNA_SKUPINA, t1.IME_STAROSTNA_SKUPINA  
    FROM DRZAVA dr  
    JOIN (  
        SELECT ss.ID_DRZAVA, ss.ID_STAROSTNA_SKUPINA, ss.IME_STAROSTNA_SKUPINA  
        FROM DATUM d  
        JOIN STAROSTNA_SKUPINA ss  
        ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA  
        WHERE d.DATUM >= '2021-09-12'  
        AND d.DATUM <= '2021-12-12'  
    ) t1  
    ON dr.ID_DRZAVA = t1.ID_DRZAVA  
    WHERE dr.IME_DRZAVA = "Slovenia"  
) t  
ON ts.ID_STAROSTNA_SKUPINA = t.ID_STAROSTNA_SKUPINA  
WHERE ts.REZULTAT_TEST = "positive"  
GROUP BY t.IME_STAROSTNA_SKUPINA;
```

## Rezultat

```
poizvedbaC14.csv ×  
1 IME_STAROSTNA_SKUPINA,STEVILO_POZITIVNIH_SLOVENIJA_4  
2 sk_0-4,3413  
3 sk_5-14,27132  
4 sk_15-24,20661  
5 sk_25-34,22785  
6 sk_35-44,31863  
7 sk_45-54,24493  
8 sk_55-64,16781  
9 sk_65-74,10446  
10 sk_75-84,4821  
11 sk_85+,2039
```

## Rezultati

1. Vrne število umrlih po starostnih skupinah v obdobju analiziranega 2. vala v Sloveniji.
2. Vrne število umrlih po starostnih skupinah v obdobju analiziranega 4. vala v Sloveniji.
3. Vrne število cepljenih po starostnih skupinah od začetka epidemije do konca analiziranega 2. vala v Sloveniji.
4. Vrne število cepljenih po starostnih skupinah od začetka epidemije do konca analiziranega 4. vala v Sloveniji.
5. Vrne število okuženih (pozitivnih) po starostnih skupinah analiziranega 2. vala v Sloveniji.
6. Vrne število okuženih (pozitivnih) po starostnih skupinah analiziranega 4. vala v Sloveniji.

ID	sk_0-4	sk_5-14	sk_15-24	sk_25-34	sk_35-44	sk_45-54	sk_55-64	sk_65-74	sk_75-84	sk_85+
1	0	0	0	2	17	34	127	469	1139	1665
2	0	0	1	0	8	21	87	194	332	357
3	0	0	421	3136	4599	5621	5778	2247	2254	3465
4	0	32857	78604	122651	169034	194272	213260	199340	114772	45555
5	785	1781	10307	21632	25397	25961	19635	11118	8985	7166
6	3413	27132	20661	22785	31863	24493	16781	10446	4821	2039

### Predpostavka 3

V državah, kjer je večji delež cepljenih ljudi, je manjši delež hospitalizacij na intenzivni negi.

#### Poizvedbe

15. Vrne 3 evropske države, kjer je precepljenost prebivalstva najvišja ter delež hospitalizacij na intenzivni negi.

```
SELECT t6.DATUM, t6.IME_DRZAVA, (SUM(c.STEVILO_CEPLJENIH) / t6.STEVILO_PREBIVALCEV) AS  
DELEZ_CEPLJENIH, (SUM(t6.STEVILO_INTENZIVNA_NEGA) * 1000000 / t6.STEVILO_PREBIVALCEV) AS  
DELEZ_MIL_INTENZIVNA_NEGA  
FROM CEPIVO c  
JOIN (  
    SELECT t5.IME_DRZAVA, t5.ID_STAROSTNA_SKUPINA, t5.STEVILO_INTENZIVNA_NEGA,  
t5.STEVILO_PREBIVALCEV, d.DATUM  
    FROM DATUM d  
    JOIN (  
        SELECT t4.IME_DRZAVA, t4.ID_STAROSTNA_SKUPINA, o.STEVILO_INTENZIVNA_NEGA,  
t4.STEVILO_PREBIVALCEV  
        FROM ODDELEK o  
        JOIN (  
            SELECT t3.IME_DRZAVA, t3.ID_STAROSTNA_SKUPINA, b.ID_BOLNISNICA,  
t3.STEVILO_PREBIVALCEV  
            FROM BOLNISNICA b  
            JOIN (  
                SELECT t2.IME_DRZAVA, t2.ID_STAROSTNA_SKUPINA, sd.ID_BOLNISNICA,  
t2.STEVILO_PREBIVALCEV  
                FROM SE_DELI sd  
                JOIN (  
                    SELECT t1.IME_DRZAVA,  
ss.ID_STAROSTNA_SKUPINA, t1.STEVILO_PREBIVALCEV  
                    FROM STAROSTNA_SKUPINA ss  
                    JOIN (  
                        SELECT d.ID_DRZAVA, d.IME_DRZAVA, d.STEVILO_PREBIVALCEV  
                        FROM DRZAVA d  
                        JOIN REGIJA r  
                            ON d.ID_DRZAVA = r.ID_REGIJA  
                            WHERE r.IME_REGIJA = "Europe"  
                    ) t1
```

```
    ON ss.ID_DRZAVA = t1.ID_DRZAVA
) t2
    ON sd.ID_STAROSTNA_SKUPINA = t2.ID_STAROSTNA_SKUPINA
) t3
    ON t3.ID_BOLNISNICA = b.ID_BOLNISNICA
) t4
    ON o.ID_BOLNISNICA = t4.ID_BOLNISNICA
) t5
    ON d.ID_STAROSTNA_SKUPINA = t5.ID_STAROSTNA_SKUPINA
) t6
    ON c.ID_STAROSTNA_SKUPINA = t6.ID_STAROSTNA_SKUPINA
GROUP BY t.IME_DRZAVA
ORDER BY DELEZ_CEPLJENIH DESC
LIMIT 3;
```

16. Vrne 3 evropske države, kjer je precepljenost prebivalstva najnižja ter delež hospitalizacij na intenzivni negi.

```
SELECT t6.DATUM, t6.IME_DRZAVA, (SUM(c.STEVILO_CEPLJENIH) / t6.STEVILO_PREBIVALCEV) AS  
DELEZ_CEPLJENIH, (SUM(t6.STEVILO_INTENZIVNA_NEGA) * 1000000 / t6.STEVILO_PREBIVALCEV) AS  
DELEZ_MIL_INTENZIVNA_NEGA  
FROM CEPIVO c  
JOIN (  
    SELECT t5.IME_DRZAVA, t5.ID_STAROSTNA_SKUPINA, t5.STEVILO_INTENZIVNA_NEGA,  
t5.STEVILO_PREBIVALCEV, d.DATUM  
    FROM DATUM d  
    JOIN (  
        SELECT t4.IME_DRZAVA, t4.ID_STAROSTNA_SKUPINA, o.STEVILO_INTENZIVNA_NEGA,  
t4.STEVILO_PREBIVALCEV  
        FROM ODDELEK o  
        JOIN (  
            SELECT t3.IME_DRZAVA, t3.ID_STAROSTNA_SKUPINA, b.ID_BOLNISNICA,  
t3.STEVILO_PREBIVALCEV  
            FROM BOLNISNICA b  
            JOIN (  
                SELECT t2.IME_DRZAVA, t2.ID_STAROSTNA_SKUPINA, sd.ID_BOLNISNICA,  
t2.STEVILO_PREBIVALCEV  
                FROM SE_DELI sd  
                JOIN (  
                    SELECT t1.IME_DRZAVA,  
ss.ID_STAROSTNA_SKUPINA, t1.STEVILO_PREBIVALCEV  
                    FROM STAROSTNA_SKUPINA ss  
                    JOIN (  
                        SELECT d.ID_DRZAVA, d.IME_DRZAVA, d.STEVILO_PREBIVALCEV  
                        FROM DRZAVA d  
                        JOIN REGIJA r  
                            ON d.ID_DRZAVA = r.ID_REGIJA  
                            WHERE r.IME_REGIJA = "Europe"  
                    ) t1  
                    ON ss.ID_DRZAVA = t1.ID_DRZAVA  
                ) t2  
                ON sd.ID_STAROSTNA_SKUPINA = t2.ID_STAROSTNA_SKUPINA  
            ) t3  
            ON t3.ID_BOLNISNICA = b.ID_BOLNISNICA
```

```
) t4
    ON o.ID_BOLNISNICA = t4.ID_BOLNISNICA
) t5
    ON d.ID_STAROSTNA_SKUPINA = t5.ID_STAROSTNA_SKUPINA
) t6
    ON c.ID_STAROSTNA_SKUPINA = t6.ID_STAROSTNA_SKUPINA
GROUP BY t.IME_DRZAVA
ORDER BY DELEZ_CEPLJENIH ASC
LIMIT 3;
```

## Rezultati

Zgornji poizvedbi sva še dodatno sflitrirala - iskane podatke vzameta 1 x na mesec do vključno 2021-04-12.

poizvedbaC15.csv ×

	DATUM,IME_DRZAVA,DELEZ_CEPLJENIH,DELEZ_MIL_INTENZIVNA_NEGA
1	2021-12-12,Portugal,88.81,14.06
2	2021-12-12,Malta,84.08,3.88
3	2021-12-12,Iceland,82.34
4	2021-11-12,Portugal,87.78,6.39
5	2021-11-12,Malta,83.41,5.81
6	2021-11-12,Iceland,81.41
7	2021-10-12,Portugal,86.46,5.51
8	2021-10-12,Malta,82.16,0.00
9	2021-10-12,Iceland,80.92
10	2021-09-12,Malta,80.66,5.81
11	2021-09-12,Portugal,79.96,11.9
12	2021-09-12,Iceland,77.48
13	2021-08-12,Malta,77.77,5.81
14	2021-08-12,Iceland,74.36
15	2021-08-12,Portugal,67.34,17.8
16	2021-07-12,Iceland,71.84
17	2021-07-12,Malta,68.09,1.94
18	2021-07-12,Hungary,54.00
19	2021-06-12,San Marino,63.45
20	2021-06-12,Malta,50.17,0.00
21	2021-06-12,United Kingdom,43.68,2.48
22	2021-05-12,San Marino,37.11
23	2021-05-12,Monaco,32.83
24	2021-05-12,United Kingdom,27.7,1.99
25	2021-04-12,Monaco,27.48
26	2021-04-12,San Marino,24.69
27	2021-04-12,Serbia,17.49,36.77

poizvedbaC16.csv ×

	DATUM,IME_DRZAVA,DELEZ_CEPLJENIH,DELEZ_MIL_INTENZIVNA_NEGA
1	2021-12-12,Ukraine,28.71
2	2021-12-12,Bulgaria,26.67,90.91
3	2021-12-12,Moldova,23.77
4	2021-11-12,Moldova,22.41
5	2021-11-12,Bosnia and Herzegovina,22.08
6	2021-11-12,Ukraine,19.8
7	2021-10-12,Belarus,18.53
8	2021-10-12,Bosnia and Herzegovina,15.57
9	2021-10-12,Ukraine,14.48
10	2021-09-12,Belarus,15.41
11	2021-09-12,Bosnia and Herzegovina,13.06
12	2021-09-12,Ukraine,11.03
13	2021-08-12,Bosnia and Herzegovina,10.04
14	2021-08-12,Kosovo,9.82
15	2021-08-12,Ukraine,5.95
16	2021-07-12,Bosnia and Herzegovina,5.02
17	2021-07-12,Kosovo,3.7
18	2021-07-12,Ukraine,2.8
19	2021-06-12,Bosnia and Herzegovina,2.0
20	2021-06-12,Kosovo,1.0
21	2021-06-12,Ukraine,0.55
22	2021-05-12,Bosnia and Herzegovina,1.09
23	2021-05-12,Moldova,0.67
24	2021-05-12,Ukraine,0.01
25	2021-04-12,Bosnia and Herzegovina,0.31
26	2021-04-12,Belarus,0.28
27	2021-04-12,Moldova,0.2

Zaradi manjkajočih podatkov o številu zasedenih intenzivnih posteljah pri nekaterih državah, sva poizvedbi znotraj MySQL še dodatno sfiltrirala (prikažeta samo države, ki imajo ta podatek).

poizvedbaC15-sfiltrirano.csv ×

```
1 DATUM,IME_DRZAVA,DELEZ_CEPLJENIH,DELEZ_MIL_INTENZIVNA_NEGL  
2 2021-12-12,Portugal,88.81,14.06  
3 2021-12-12,Malta,84.08,3.88  
4 2021-12-12,Spain,80.7,10.8  
5 2021-11-12,Portugal,87.78,6.39  
6 2021-11-12,Malta,83.41,5.81  
7 2021-11-12,Spain,81.41,8.43  
8 2021-10-12,Portugal,86.46,5.51  
9 2021-10-12,Malta,82.16,0.00  
10 2021-10-12,Spain,79.07,10.4  
11 2021-09-12,Portugal,80.66,5.81  
12 2021-09-12,Malta,79.96,11.9  
13 2021-09-12,Spain,75.52,25.07  
14 2021-08-12,Malta,78.09,5.81  
15 2021-08-12,Portugal,67.65,16.62  
16 2021-08-12,Netherlands,66.2,12.23  
17 2021-07-12,Malta,68.09,1.94  
18 2021-07-12,United Kingdom,51.31,7.07  
19 2021-07-12,Spain,47.36,15.38  
20 2021-06-12,Malta,50.17,0.00  
21 2021-06-12,United Kingdom,43.68,2.48  
22 2021-06-12,Cyprus,30.77,8.93  
23 2021-05-12,United Kingdom,27.77,1.99  
24 2021-05-12,Malta,25.29,9.69  
25 2021-05-12,Serbia,24.69,15.05  
26 2021-04-12,Serbia,17.49,36.77  
27 2021-04-12,Malta,14.35,25.19  
28 2021-04-12,United Kingdom,11.52,5.53
```

poizvedbaC16-sfiltrirano.csv ×

```
1 DATUM,IME_DRZAVA,DELEZ_CEPLJENIH,DELEZ_MIL_INTENZIVNA_NEGL  
2 2021-12-12,Slovakia,43.47,53.11  
3 2021-12-12,Romania,40.01,37.75  
4 2021-12-12,Bulgaria,26.67,90.91  
5 2021-11-12,Slovakia,42.64,43.77  
6 2021-11-12,Romania,35.22,90.6  
7 2021-11-12,Bulgaria,23.42,109.47  
8 2021-10-12,Slovakia,41.77,17.76  
9 2021-10-12,Romania,29.1,87.62  
10 2021-10-12,Bulgaria,19.86,68.44  
11 2021-09-12,Slovakia,40.6,4.21  
12 2021-09-12,Romania,27.23,31.37  
13 2021-09-12,Bulgaria,17.93,55.1  
14 2021-08-12,Slovakia,38.38,0.55  
15 2021-08-12,Romania,25.98,5.12  
16 2021-08-12,Bulgaria,15.42,15.66  
17 2021-07-12,Finland,24.65,1.44  
18 2021-07-12,Romania,24.44,2.67  
19 2021-07-12,Bulgaria,12.36,15.08  
20 2021-06-12,Czechia,19.56,2.33  
21 2021-06-12,Finland,13.37,1.26  
22 2021-06-12,Bulgaria,10.11,35.67  
23 2021-05-12,Luxembourg,5.96,47.26  
24 2021-05-12,Bulgaria,5.25,77.57  
25 2021-05-12,Finland,3.91,4.33  
26 2021-04-12,Slovakia,5.51,41.39  
27 2021-04-12,Finland,1.79,7.57  
28 2021-04-12,Bulgaria,1.66,115.56
```

## Predpostavka 4

Različica delta je bolj kužna a manj smrtonosna od originalne različice virusa Covid-19.

Opomba: Vpliv cepljenja pri tem nisva upoštevala.

## Poizvedbe

17. Vrne število okuženih, število hospitalizacij na navadni in intenzivni negi ter število smrti v obdobju, ko je prevladovala originalna različica virusa Covid-19 v Sloveniji.

```
SELECT t5.STEVILO_POZITIVNIH, t5.STEVILO_INTENZIVNA_NEZA, t5.STEVILO_NAVADNA_NEZA,  
SUM(p.STEVILO_PREMINULIH) AS STEVILO_SMRTI  
FROM PREMINULI p  
JOIN (  
    SELECT t4.ID_STAROSTNA_SKUPINA, t4.STEVILO_INTENZIVNA_NEZA, t4.STEVILO_NAVADNA_NEZA,  
    SUM(ts.STEVILO_TESTIRANIH) AS STEVILO_POZITIVNIH  
    FROM TEST ts  
    JOIN (  
        SELECT t3.ID_STAROSTNA_SKUPINA, o.STEVILO_INTENZIVNA_NEZA, o.STEVILO_NAVADNA_NEZA  
        FROM ODDELEK o  
        JOIN (  
            SELECT t2.ID_STAROSTNA_SKUPINA, b.ID_BOLNISNICA  
            FROM BOLNISNICA b  
            JOIN (  
                SELECT t1.ID_STAROSTNA_SKUPINA, sd.ID_BOLNISNICA  
                FROM SE_DELI sd  
                JOIN (  
                    SELECT ss.ID_STAROSTNA_SKUPINA  
                    FROM DATUM d  
                    JOIN STAROSTNA_SKUPINA ss  
                    ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA  
                    WHERE d.DATUM >= '2020-03-04'  
                    AND d.DATUM <= '2021-04-01'  
                ) t1  
                ON sd.ID_STAROSTNA_SKUPINA = t1.ID_STAROSTNA_SKUPINA  
            ) t2  
            ON t2.ID_BOLNISNICA = b.ID_BOLNISNICA  
        ) t3  
        ON o.ID_BOLNISNICA = t3.ID_BOLNISNICA  
    ) t4
```

```
ON ts.ID_STAROSTNA_SKUPINA = t4.ID_STAROSTNA_SKUPINA  
WHERE ts.REZULTAT_TEST="positive"  
) t5  
ON p.ID_STAROSTNA_SKUPINA = t5.ID_STAROSTNA_SKUPINA;
```

## Rezultat

poizvedbaC17.csv ×

1	STEVILO_POZITIVNIH,STEVILO_INTENZIVNA_NEZA,STEVILO_NAVADNA_NEZA,STEVILO_SMRTI
2	218266,27498,136709,4054

18. Vrne število pozitivnih primerov, število hospitalizacij na navadni in intenzivni negi in število smrti v obdobju, ko je prevladovala delta različica virusa Covid-19 v Sloveniji.

```
SELECT t5.STEVILO_POZITIVNIH, t5.STEVILO_INTENZIVNA_NEGA, t5.STEVILO_NAVADNA_NEGA,  
SUM(p.STEVILO_PREMINULIH) AS STEVILO_SMRTI  
FROM PREMINULI p  
JOIN (  
    SELECT t4.ID_STAROSTNA_SKUPINA, SUM(t4.STEVILO_INTENZIVNA_NEGA +  
t4.STEVILO_NAVADNA_NEGA) AS STEVILO_HOSPITALIZIRANIH, SUM(ts.STEVILO_TESTIRANIH) AS  
STEVILO_POZITIVNIH  
    FROM TEST ts  
    JOIN (  
        SELECT t3.ID_STAROSTNA_SKUPINA, o.STEVILO_INTENZIVNA_NEGA, o.STEVILO_NAVADNA_NEGA  
        FROM ODDELEK o  
        JOIN (  
            SELECT t2.ID_STAROSTNA_SKUPINA, b.ID_BOLNISNICA  
            FROM BOLNISNICA b  
            JOIN (  
                SELECT t1.ID_STAROSTNA_SKUPINA, sd.ID_BOLNISNICA  
                FROM SE_DELI sd  
                JOIN (  
                    SELECT ss.ID_STAROSTNA_SKUPINA  
                    FROM DATUM d  
                    JOIN STAROSTNA_SKUPINA ss  
                    ON d.ID_STAROSTNA_SKUPINA = ss.ID_STAROSTNA_SKUPINA  
                    WHERE d.DATUM >= '2021-07-11'  
                    AND d.DATUM <= '2021-12-12'  
                ) t1  
                ON sd.ID_STAROSTNA_SKUPINA = t1.ID_STAROSTNA_SKUPINA  
            ) t2  
            ON t2.ID_BOLNISNICA = b.ID_BOLNISNICA  
        ) t3  
        ON o.ID_BOLNISNICA = t3.ID_BOLNISNICA  
    ) t4  
    ON ts.ID_STAROSTNA_SKUPINA = t4.ID_STAROSTNA_SKUPINA  
    WHERE ts.REZULTAT_TEST="positive"  
) t5  
ON p.ID_STAROSTNA_SKUPINA = t5.ID_STAROSTNA_SKUPINA
```

## Rezultat

poizvedbaC18.csv	
1	STEVILO_POZITIVNIH,STEVILO_INTENZIVNA_NEGA,STEVIVO_NAVADNA_NEGA,STEVIVO_SMRTI
2	182790,16542,50737,986

## Rezultati

različica	število okuženih	število hospitalizacij (navadna nega)	število hospitalizacij (intenzivna nega)	število smrti
originalna	218266	27498	136709	4054
delta	182790	16542	50737	986

# Analiza podatkov

Z analizo podatkov v programskem jeziku R in okolju RStudio sva poskušala potrditi najine predpostavke.

Import Excel Data

File/URL:  
~/Desktop/predpostavka3\_1.xlsx

Data Preview:

DATUM (double)	IME_DRZAVA (character)	DELEZ_CEPLJENIH (character)	DELEZ_MIL_INTENZIVNA_NEGA (character)
2021-12-12	Portugal	88.81	14.06
2021-12-12	Malta	84.08	3.88
2021-12-12	Spain	80.7	10.8
2021-11-12	Portugal	87.78	6.39
2021-11-12	Malta	83.41	5.81
2021-11-12	Spain	81.41	8.43
2021-10-12	Portugal	86.46	5.51
2021-10-12	Malta	82.16	0.00
2021-10-12	Spain	79.07	10.4
2021-09-12	Portugal	80.66	5.81
2021-09-12	Malta	79.96	11.9
2021-09-12	Spain	75.52	25.07
2021-08-12	Malta	78.09	5.81
2021-08-12	Portugal	67.65	16.62
2021-08-12	Netherlands	66.2	12.23
2021-07-12	Malta	68.09	1.94

Previewing first 50 entries.

Import Options:

Name: predpostavka3_1	Max Rows:	<input checked="" type="checkbox"/> First Row as Names
Sheet: Default	Skip:	<input type="text" value="0"/> <input checked="" type="checkbox"/> Open Data Viewer
Range: A1:D10	NA:	

Code Preview:

```
library(readxl)
predpostavka3_1 <- read_excel("Desktop/predpostavka3_1.xlsx")
View(predpostavka3_1)
```

RStudio

predpostavka3\_1

DATUM	IME_DRZAVA	DELEZ_CEPLJENIH	DELEZ_MIL_INTENZIVNA_NEGA
1 2021-12-12	Portugal	88.81	14.06
2 2021-12-12	Malta	84.08	3.88
3 2021-12-12	Spain	80.7	10.8
4 2021-11-12	Portugal	87.78	6.39
5 2021-11-12	Malta	83.41	5.81
6 2021-11-12	Spain	81.41	8.43
7 2021-10-12	Portugal	86.46	5.51
8 2021-10-12	Malta	82.16	0.00
9 2021-09-12	Spain	79.07	10.4
10 2021-09-12	Portugal	80.66	5.81
11 2021-09-12	Malta	79.96	11.9
12 2021-09-12	Spain	75.52	25.07
13 2021-08-12	Malta	78.09	5.81
14 2021-08-12	Portugal	67.65	16.62
15 2021-08-12	Netherlands	66.2	12.23
16 2021-07-12	Malta	68.09	1.94
17 2021-07-12	United Kingdom	51.31	7.07
18 2021-07-12	Spain	47.36	15.38
19 2021-06-12	Malta	50.17	0.00
20 2021-06-12	United Kingdom	43.68	2.48

Showing 1 to 20 of 27 entries. 4 total columns

Environment History Connections Tutorial

Global Environment

```
R - Global Environment
```

Name	Type	Length	Size	Value
Bulgaria_cepl...	character	4	304 B	chr [1:4] "26.67" "23.42..."
Bulgaria_datum	POSIXct	4	368 B	POSIXct[1:4], format: "..."
Bulgaria_int...	character	4	304 B	chr [1:4] "90.91" "109.4..."
cepljnih	character	27	1.7 KB	chr [1:27] "88.81" "84.0..."
Chechia_datum	POSIXct	1	344 B	2021-06-12 02:00:00 CEST
Chechia_int...	character	1	112 B	"2.33"
Cyprus_ceplje...	character	1	112 B	"30.77"
Cyprus_datum	POSIXct	1	344 B	2021-06-12 02:00:00 CEST
data	data.frame	2	1.7 KB	18 obs. of 2 variables
datum	POSIXct	27	776 B	POSIXct[1:27], format: "..."
delez_stevilo...	numeric	10	176 B	num [1:10] 0 0 0.06 0..."
delez_stevilo...	numeric	10	176 B	num [1:10] 0 0 0.1 0 0.8..."
Finland_datum	POSIXct	4	368 B	POSIXct[1:4], format: "..."
Finland_int...	character	4	304 B	chr [1:4] "1.44" "1.26"...
int_nega	character	27	1.7 KB	chr [1:27] "53.11" "37.7..."
Luxembourg_dat...	POSIXct	1	344 B	2021-05-12 02:00:00 CEST
Luxembourg_int...	character	1	112 B	"47.26"
Malta_cepljen...	character	4	304 B	chr [1:4] "84.08" "83.41..."
Malta_datum	POSIXct	4	368 B	POSIXct[1:4], format: "..."
Malta_int_nega	character	4	304 B	chr [1:4] "3.88" "5.81"...
oznaka_datum	character	8	584 B	chr [1:8] "Moj" "Junij" ...
plot	gg	9	8.1 KB	List of 9
point_nega	character	27	1.7 KB	chr [1:27] "53.11" "37.7..."
poizvedbo15	tbl_df	4	5.8 KB	27 obs. of 4 variables
poizvedbo16	tbl_df	4	5.9 KB	27 obs. of 4 variables
poizvedbo15	tbl_df	5	6.5 KB	48 obs. of 5 variables
Portugal_cepl...	character	4	304 B	chr [1:4] "88.81" "87.78..."
Portugal_datum	POSIXct	4	368 B	POSIXct[1:4], format: "..."
Portugal_int...	character	4	304 B	chr [1:4] "14.06" "6.39"...
predpostavka1	tbl_df	8	1.8 KB	1 obs. of 8 variables
predpostavka2	tbl_df	11	3 KB	6 obs. of 11 variables
predpostavka3	tbl_df	4	5.8 KB	27 obs. of 4 variables

Console Terminal Jobs

```
R 4.0.3 - / ~
> library(readxl)
> predpostavka3_1 <- read_excel("Desktop/predpostavka3_1.xlsx")
> View(predpostavka3_1)
> |
```

## Predpostavka 1

Cepljenje proti Covid-19 zmanjša število smrti.

### Koda

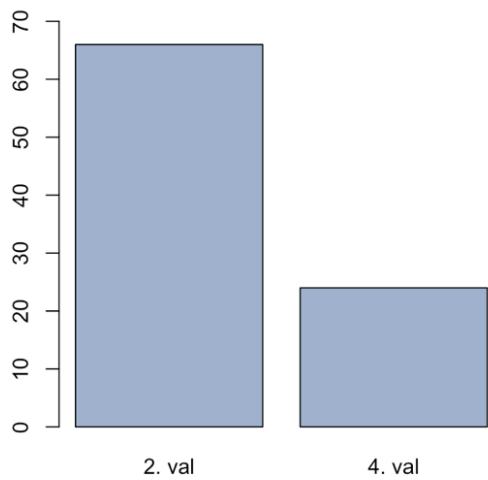
```
install.packages("ggplot2")
library(readxl)

predpostavka1 <- read_excel("Desktop/Covid/predpostavka1.xlsx")
View(predpostavka1)

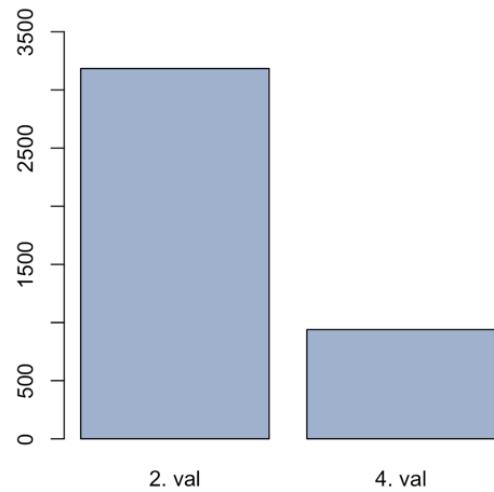
barplot(height=c(predpostavka1$stevilo_smerti_2, predpostavka1$stevilo_smerti_4), names=c("2. val", "4. val"),
       col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,3500), main = "Število smrti")
barplot(height=c(predpostavka1$max_st_smerti_2, predpostavka1$max_st_smerti_4), names=c("2. val", "4.
       val"), col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,70), main = "Najvišje število smrti v enem dnevu")
barplot(height=c(predpostavka1$stevilo_poz_2, predpostavka1$stevilo_poz_4), names=c("2. val", "4. val"),
       col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,200000), main = "Število okuženih (pozitivnih)")
barplot(height=c(predpostavka1$stevilo_cep_2, predpostavka1$stevilo_cep_4), names=c("2. val", "4. val"),
       col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,1300000), main = "Število polno cepljenih")
```

## Rezultati

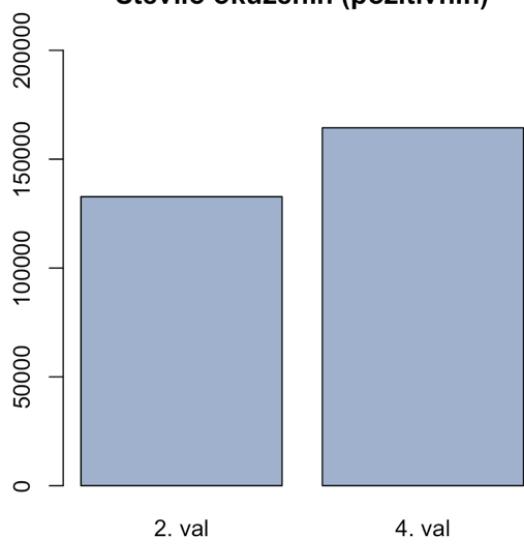
Najvišje število smrti v enem dnevu



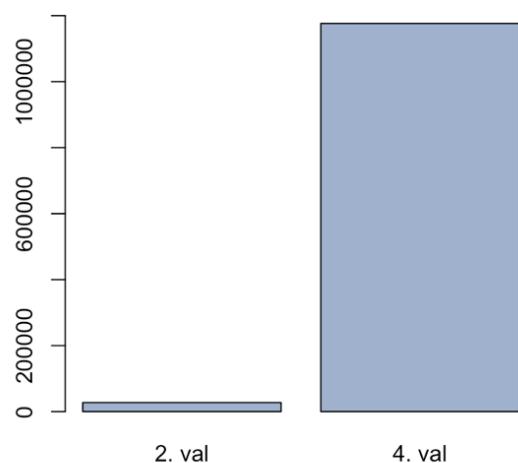
Število smrti



Število okuženih (pozitivnih)



Število polno cepljenih



## Predpostavka 2

Povprečna starost umrlih se je 4. valu v primerjavi z 2. valom zmanjšala.

### Koda

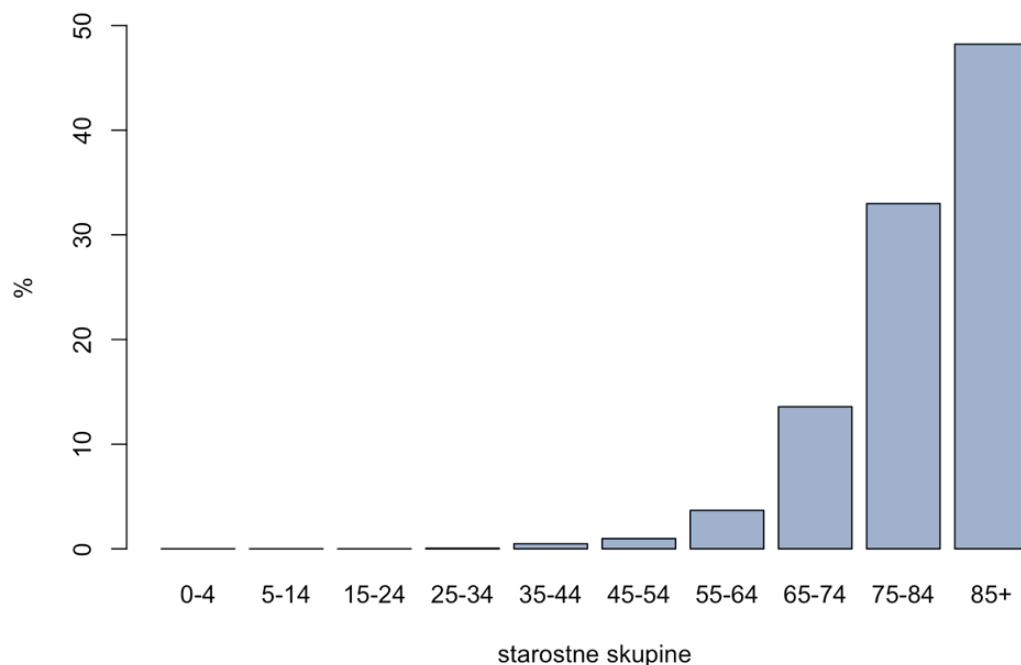
```
predpostavka2 <- read_excel("Desktop/Covid/predpostavka2.xlsx")
View(predpostavka2)

delez_stevilo_smrti_2 <- stevilo_smrti_2/sum(stevilo_smrti_2)*100
delez_stevilo_smrti_4 <- stevilo_smrti_4/sum(stevilo_smrti_4)*100
delez_stevilo_poz_2 <- stevilo_poz_2/sum(stevilo_poz_2)*100
delez_stevilo_poz_4 <- stevilo_poz_4/sum(stevilo_poz_4)*100
starostne_skupine <- c("0-4", "5-14", "15-24", "25-34", "35-44", "45-54", "55-64", "65-74", "75-84", "85+")

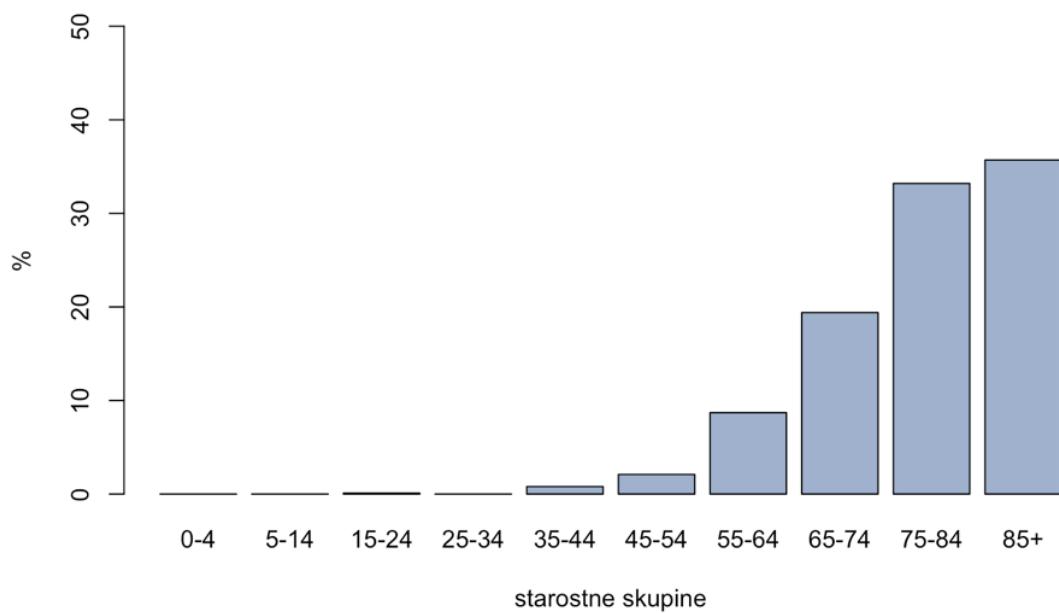
barplot(height=delez_stevilo_smrti_2, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,50),
xlab="starostne skupine", ylab="%",main = "Delež smrti po starostnih skupinah v obdobju analiziranega 2.
vala")
barp <- barplot(height=delez_stevilo_smrti_2, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6),
ylim=c(0,60), xlab="starostne skupine", ylab="%",main = "Delež smrti po starostnih skupinah v obdobju
analiziranega 2. vala")
text(barp, x + 2, labels = round(delez_stevilo_smrti_2, 2))
barplot(height=delez_stevilo_smrti_4, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,50),
xlab="starostne skupine", ylab="%",main = "Delež smrti po starostnih skupinah v obdobju analiziranega 4.
vala")
barplot(height=stevilo_cep_2 / st_preb * 100, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6),
ylim=c(0,10), xlab="starostne skupine", ylab="%",main = "Delež polno cepljenih po starostnih skupinah v
obdobju analiziranega 2. vala")
barplot(height=stevilo_cep_4 / st_preb * 100, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6),
ylim=c(0,100), xlab="starostne skupine", ylab="%",main = "Delež polno cepljenih po starostnih skupinah v
obdobju analiziranega 4. vala")
barplot(height=delez_stevilo_poz_2, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,30),
xlab="starostne skupine", ylab="%",main = "Delež okuženih (pozitivnih) po starostnih skupinah v obdobju
analiziranega 2. vala")
barplot(height=delez_stevilo_poz_4, names=starostne_skupine, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,30),
xlab="starostne skupine", ylab="%",main = "Delež okuženih (pozitivnih) po starostnih skupinah v obdobju
analiziranega 4. vala")
```

## Rezultati

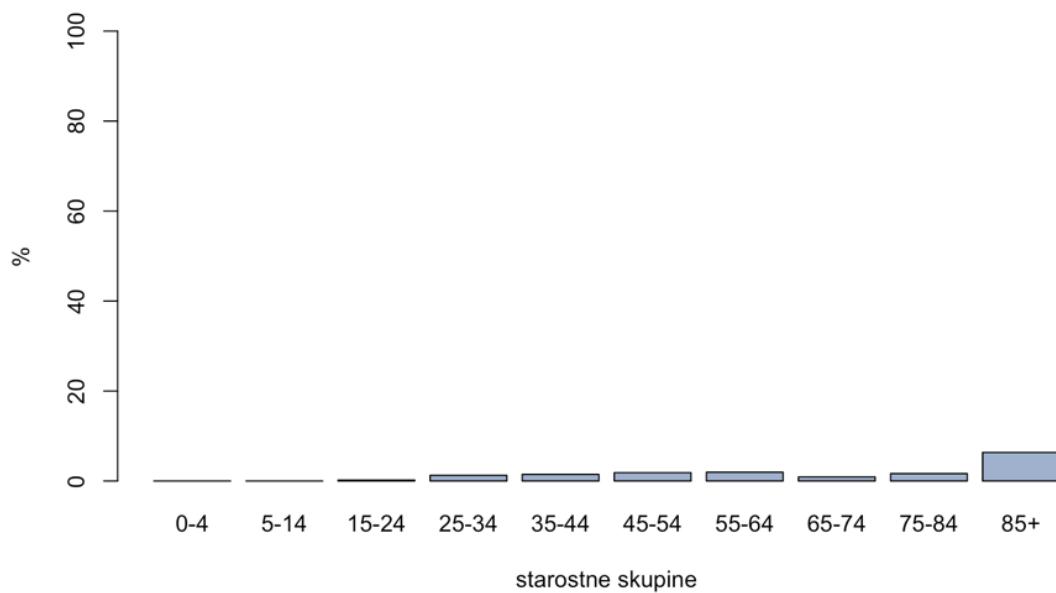
**Delež smrti po starostnih skupinah v obdobju analiziranega 2. vala**



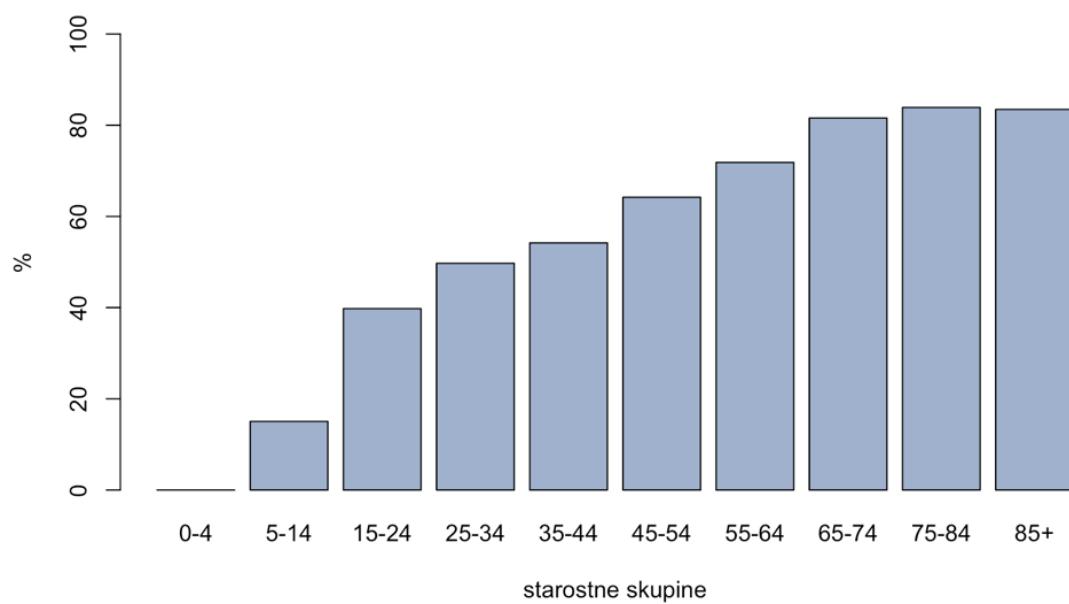
**Delež smrti po starostnih skupinah v obdobju analiziranega 4. vala**



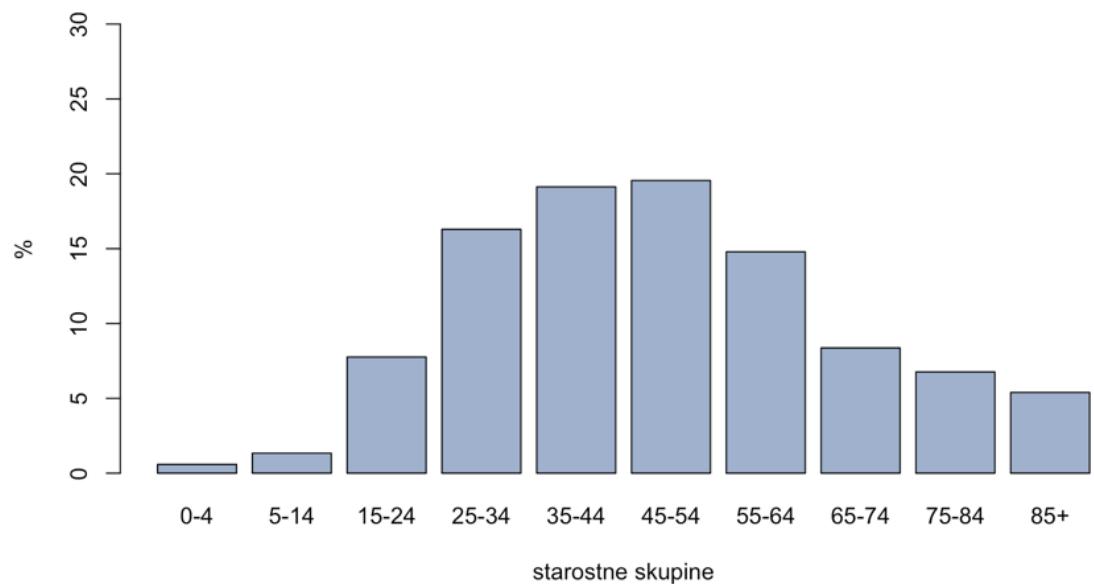
**Delež polno cepljenih po starostnih skupinah v obdobju analiziranega 2. vala**



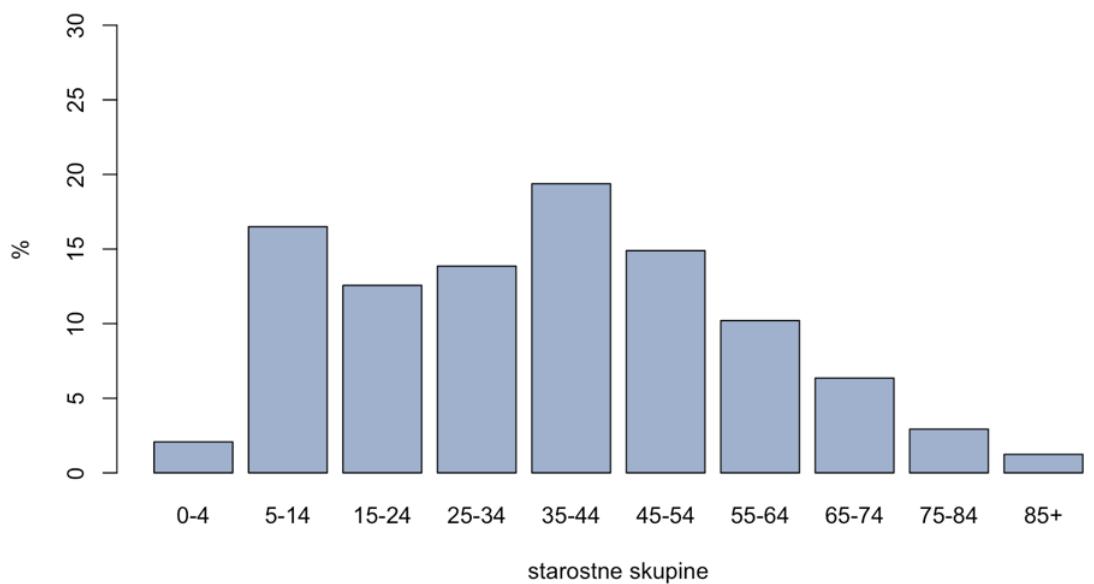
**Dlež polno cepljenih po starostnih skupinah v obdobju analiziranega 4. vala**



**Delež okuženih (pozitivnih) po starostnih skupinah v obdobju analiziranega 2. vala**



**Delež okuženih (pozitivnih) po starostnih skupinah v obdobju analiziranega 4. vala**



### Predpostavka 3

V državah, kjer je večji delež cepljenih ljudi, je manjši delež hospitalizacij na intenzivni negi.

#### Koda

```
predpostavka3_1 <- read_excel("Desktop/Covid/predpostavka3_1.xlsx")
predpostavka3_2 <- read_excel("Desktop/Covid/predpostavka3_2.xlsx")
View(predpostavka3_1)
View(predpostavka3_2)

cepljenih <- predpostavka3_1$DELEZ_CEPLJENIH
datum <- predpostavka3_1$DATUM
plot(datum, cepljenih, height=10, type="n", xlab="Mesec", ylab="%", main="Delenj cepljenih (zadnje 3 mesece leta 2021)")
Portugal_datum <- c(predpostavka3_1$DATUM[1], predpostavka3_1$DATUM[4],
predpostavka3_1$DATUM[7], predpostavka3_1$DATUM[10])
Malta_datum <- c(predpostavka3_1$DATUM[2], predpostavka3_1$DATUM[5], predpostavka3_1$DATUM[8],
predpostavka3_1$DATUM[11])
Spain_datum <- c(predpostavka3_1$DATUM[3], predpostavka3_1$DATUM[6], predpostavka3_1$DATUM[9],
predpostavka3_1$DATUM[12])
Slovakia_datum <- c(predpostavka3_2$DATUM[1], predpostavka3_2$DATUM[4],
predpostavka3_2$DATUM[7], predpostavka3_2$DATUM[10])
Romania_datum <- c(predpostavka3_2$DATUM[2], predpostavka3_2$DATUM[5],
predpostavka3_2$DATUM[8], predpostavka3_2$DATUM[11])
Bulgaria_datum <- c(predpostavka3_2$DATUM[3], predpostavka3_2$DATUM[6],
predpostavka3_2$DATUM[9], predpostavka3_2$DATUM[12])
Portugal_cepljenih <- c(predpostavka3_1$DELEZ_CEPLJENIH[1], predpostavka3_1$DELEZ_CEPLJENIH[4],
predpostavka3_1$DELEZ_CEPLJENIH[7], predpostavka3_1$DELEZ_CEPLJENIH[10])
Malta_cepljenih <- c(predpostavka3_1$DELEZ_CEPLJENIH[2], predpostavka3_1$DELEZ_CEPLJENIH[5],
predpostavka3_1$DELEZ_CEPLJENIH[8], predpostavka3_1$DELEZ_CEPLJENIH[11])
Spain_cepljenih <- c(predpostavka3_1$DELEZ_CEPLJENIH[3], predpostavka3_1$DELEZ_CEPLJENIH[6],
predpostavka3_1$DELEZ_CEPLJENIH[9], predpostavka3_1$DELEZ_CEPLJENIH[12])
Slovakia_cepljenih <- c(predpostavka3_2$DELEZ_CEPLJENIH[1], predpostavka3_2$DELEZ_CEPLJENIH[4],
predpostavka3_2$DELEZ_CEPLJENIH[7], predpostavka3_2$DELEZ_CEPLJENIH[10])
Romania_cepljenih <- c(predpostavka3_2$DELEZ_CEPLJENIH[2], predpostavka3_2$DELEZ_CEPLJENIH[5],
predpostavka3_2$DELEZ_CEPLJENIH[8], predpostavka3_2$DELEZ_CEPLJENIH[11])
Bulgaria_cepljenih <- c(predpostavka3_2$DELEZ_CEPLJENIH[3], predpostavka3_2$DELEZ_CEPLJENIH[6],
predpostavka3_2$DELEZ_CEPLJENIH[9], predpostavka3_2$DELEZ_CEPLJENIH[12])
```

```

lines(Portugal_datum, Portugal_cepljenih, type="o", col="red")
lines(Malta_datum, Malta_cepljenih, type="o", col="blue")
lines(Spain_datum, Spain_cepljenih, type="o", col="green")
lines(Slovakia_datum, Slovakia_cepljenih, type="o", col="gray")
lines(Romania_datum, Romania_cepljenih, type="o", col="brown")
lines(Bulgaria_datum, Bulgaria_cepljenih, type="o", col="black")
legend(x = "bottomleft", legend= c("Portugal", "Malta", "Spain", "Slovakia", "Romania", "Bulgaria"),
col=c("red", "blue", "green", "gray", "brown", "black"), lwd=3)

int_nega <- predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEGA
datum <- predpostavka3_2$DATUM
plot(datum, int_nega, type="n", xlab="Mesec", ylab="Število na mil. prebivalcev", main="Število bolnikov na
intenzivni negi (zadnje 3 mesece leta 2021)")
Portugal_datum <- c(predpostavka3_1$DATUM[1], predpostavka3_1$DATUM[4],
predpostavka3_1$DATUM[7], predpostavka3_1$DATUM[10])
Malta_datum <- c(predpostavka3_1$DATUM[2], predpostavka3_1$DATUM[5], predpostavka3_1$DATUM[8],
predpostavka3_1$DATUM[11])
Spain_datum <- c(predpostavka3_1$DATUM[3], predpostavka3_1$DATUM[6], predpostavka3_1$DATUM[9],
predpostavka3_1$DATUM[12])
Slovakia_datum <- c(predpostavka3_2$DATUM[1], predpostavka3_2$DATUM[4],
predpostavka3_2$DATUM[7], predpostavka3_2$DATUM[10])
Romania_datum <- c(predpostavka3_2$DATUM[2], predpostavka3_2$DATUM[5],
predpostavka3_2$DATUM[8], predpostavka3_2$DATUM[11])
Bulgaria_datum <- c(predpostavka3_2$DATUM[3], predpostavka3_2$DATUM[6],
predpostavka3_2$DATUM[9], predpostavka3_2$DATUM[12])
Portugal_int_nega <- c(predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[1],
predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[4], predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[7],
predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[10])
Malta_int_nega <- c(predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[2],
predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[5], predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[8],
predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[11])
Spain_int_nega <- c(predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[3],
predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[6], predpostavka3_1$DELEZ_MIL_INTENZIVNA_NEGA[9],
poizvedba15$DELEZ_MIL_INTENZIVNA_NEGA[12])
Slovakia_int_nega <- c(predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEGA[1],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEGA[4], predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEGA[7],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEGA[10])

```

```

Romania_int_nega <- c(predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[2],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[5], predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[8],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[11])

Bulgaria_int_nega <- c(predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[3],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[6], predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[9],
predpostavka3_2$DELEZ_MIL_INTENZIVNA_NEZA[12])

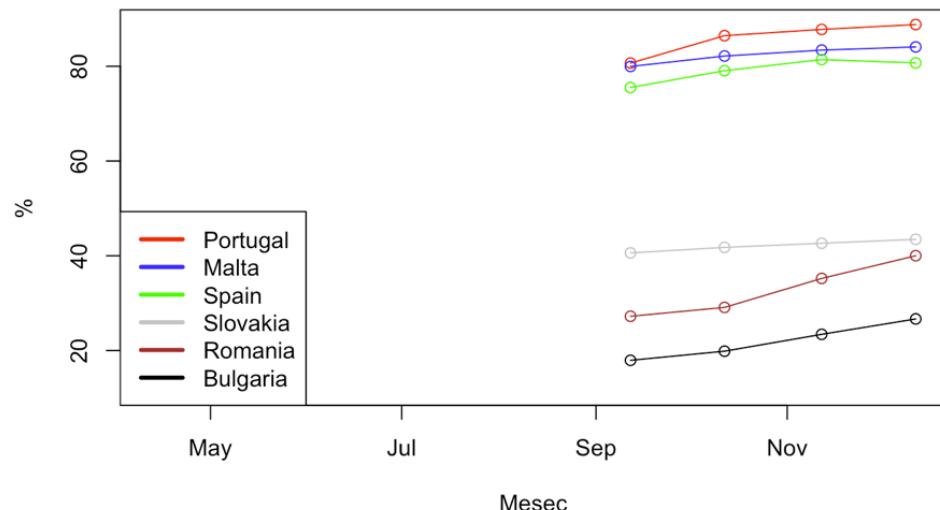
lines(Portugal_datum, Portugal_int_nega, type="o", col="red")
lines(Malta_datum, Malta_int_nega, type="o", col="blue")
lines(Spain_datum, Spain_int_nega, type="o", col="green")
lines(Slovakia_datum, Slovakia_int_nega, type="o", col="gray")
lines(Romania_datum, Romania_int_nega, type="o", col="brown")
lines(Bulgaria_datum, Bulgaria_int_nega, type="o", col="black")

legend(x = "bottomleft", legend= c("Portugal", "Malta", "Spain", "Slovakia", "Romania", "Bulgaria"),
col=c("red", "blue", "green", "gray", "brown", "black"), lwd=3)

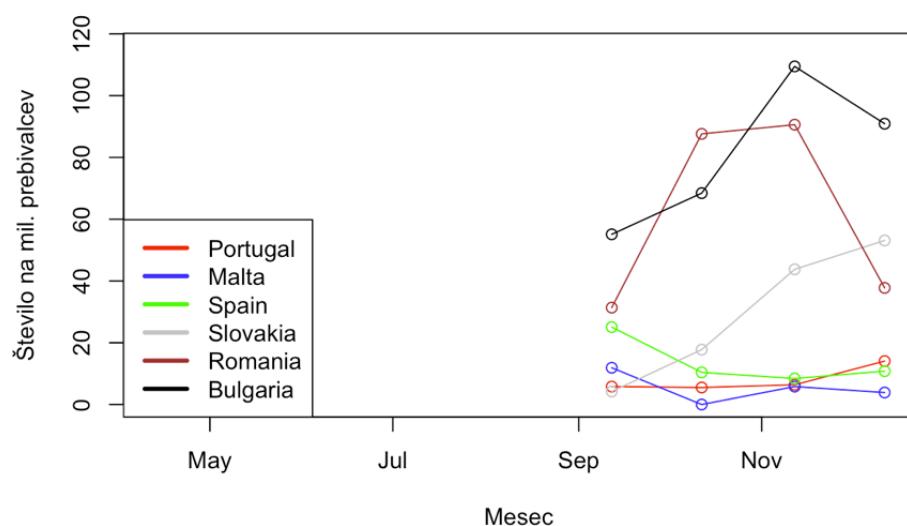
```

## Rezultati

**Delež cepljenih (zadnje 3 mesece leta 2021)**



**Število bolnikov na intenzivni negi (zadnje 3 mesece leta 2021)**



## Predpostavka 4

Različica delta je bolj kužna a manj smrtonosna od originalne različice virusa Covid-19.

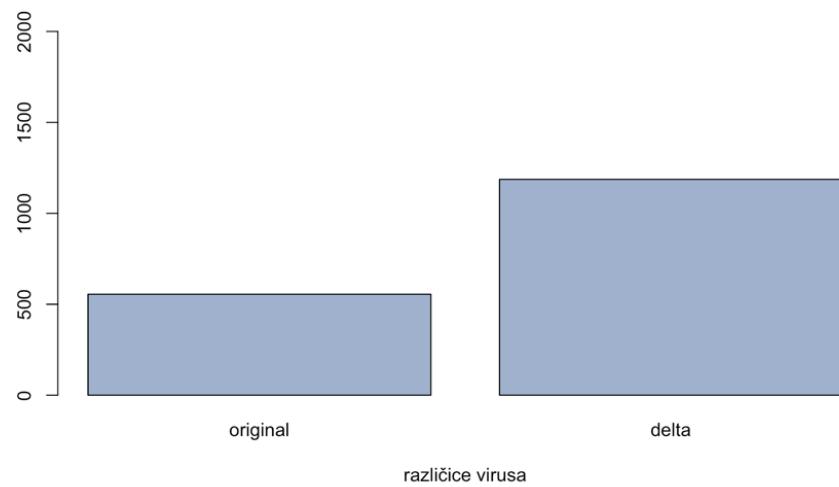
### Koda

```
predpostavka4_1 <- read_excel("Desktop/predpostavka4_1.xlsx")
View(predpostavka4_1)
predpostavka4_2 <- read_excel("Desktop/predpostavka4_2.xlsx")
View(predpostavka4_2)

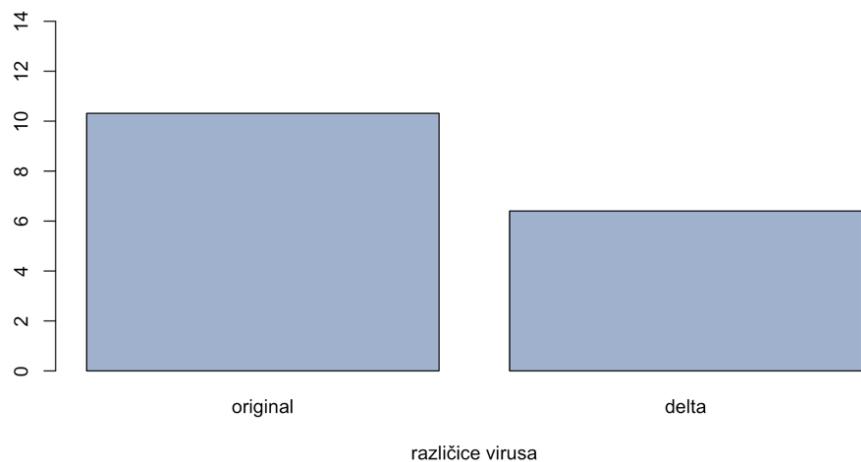
delez_okuzenih_original <- predpostavka4_1$STEVILO_POZITIVNIH / dur_original
delez_okuzenih_delta <- predpostavka4_2$STEVILO_POZITIVNIH / dur_delta
razlicice <- c("original", "delta")
stevilo_poz <- c(delez_okuzenih_original,delez_okuzenih_delta)
barplot(height=stevilo_poz, names=razlicice, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,2000), xlab="različice
virusa",main = "Povprečno število okuženih (pozitivnih) na dan")
stevilo_smrti <- c(predpostavka4_1$STEVILO_SMRTI, predpostavka4_2$STEVILO_SMRTI)
barplot(height=stevilo_smrti, names=razlicice, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,5000), xlab="različice
virusa",main = "Število smrti")
stevilo_intenzivna <- c(predpostavka4_1$STEVILO_INTENZIVNA_NEGA,
predpostavka4_2$STEVILO_INTENZIVNA_NEGA)
barplot(height=stevilo_intenzivna, names=razlicice, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,40000), xlab="različice
virusa",main = "Število bolnikov na intenzivni negi")
stevilo_smrti_dan <- c(predpostavka4_1$STEVILO_SMRTI / dur_original, predpostavka4_2$STEVILO_SMRTI /
dur_delta)
barplot(height=stevilo_smrti_dan, names=razlicice, col=rgb(0.2,0.4,0.6,0.6), ylim=c(0,15), xlab="različice
virusa",main = "Povprečno število smrti na dan")
sample_data1 <- data.frame(starostne_skupine,delez_stevilo_smrti_2)
plot<-ggplot(sample_data1, aes(starostne_skupine,delez_stevilo_smrti_2)), geom_bar(stat = "identity"),
geom_text(aes(label = signif(delez_stevilo_smrti_2)), nudge_y = 3)
plot
```

## Rezultati

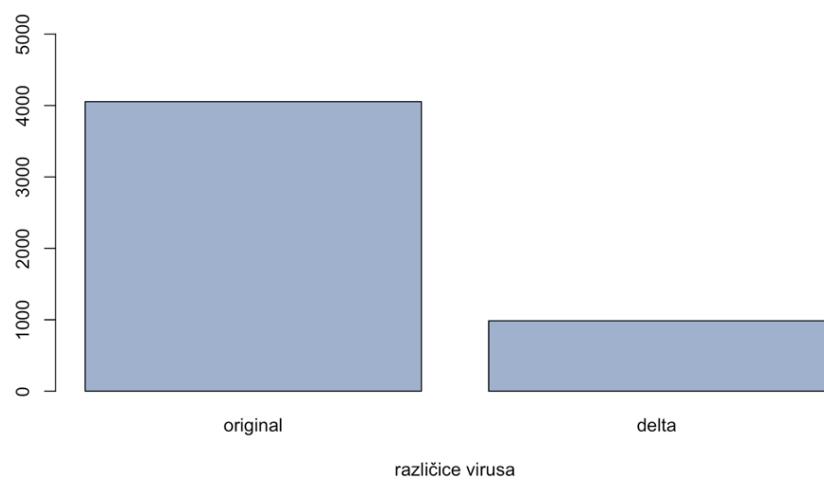
Povprečno število okuženih (pozitivnih) na dan



Povprečno število smrti na dan



Število smrti



## Sklepi in ugotovitve

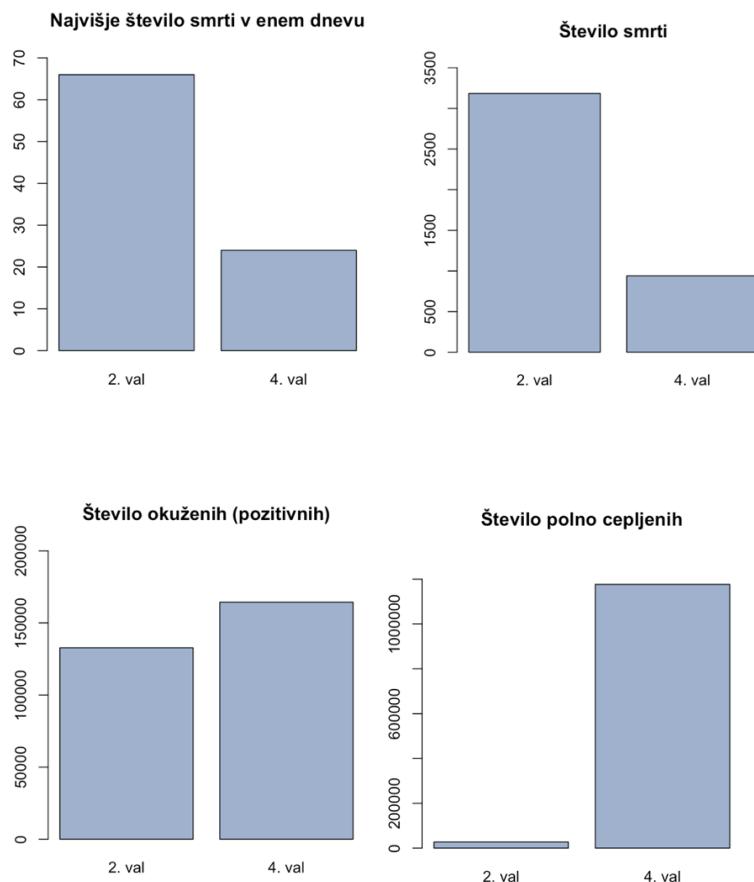
Glede na rezultate analize podatkov lahko potrdita vse najine zastavljene predpostavke.

### Prva predpostavka

Rezultati so pokazali, da cepljenje proti Covid-19 zmanjša število smrti, kar je bilo signifikantno in v korelacijski s postavljenim predpostavkom.

Iz analize podatkov je razvidno, da je bilo v 2. valu, ko je bilo cepljenih približno 40 x manj ljudi, kar 3 x več smrti kot v 4. valu.

Kljub višjemu številu okuženih oz. obolelih za Covid-19 v obdobju 4. vala lahko iz rezultatov sklepamo, da cepivo zelo dobro poskrbi za zaščito pred hudim potekom bolezni Covid-19 (manj smrti).



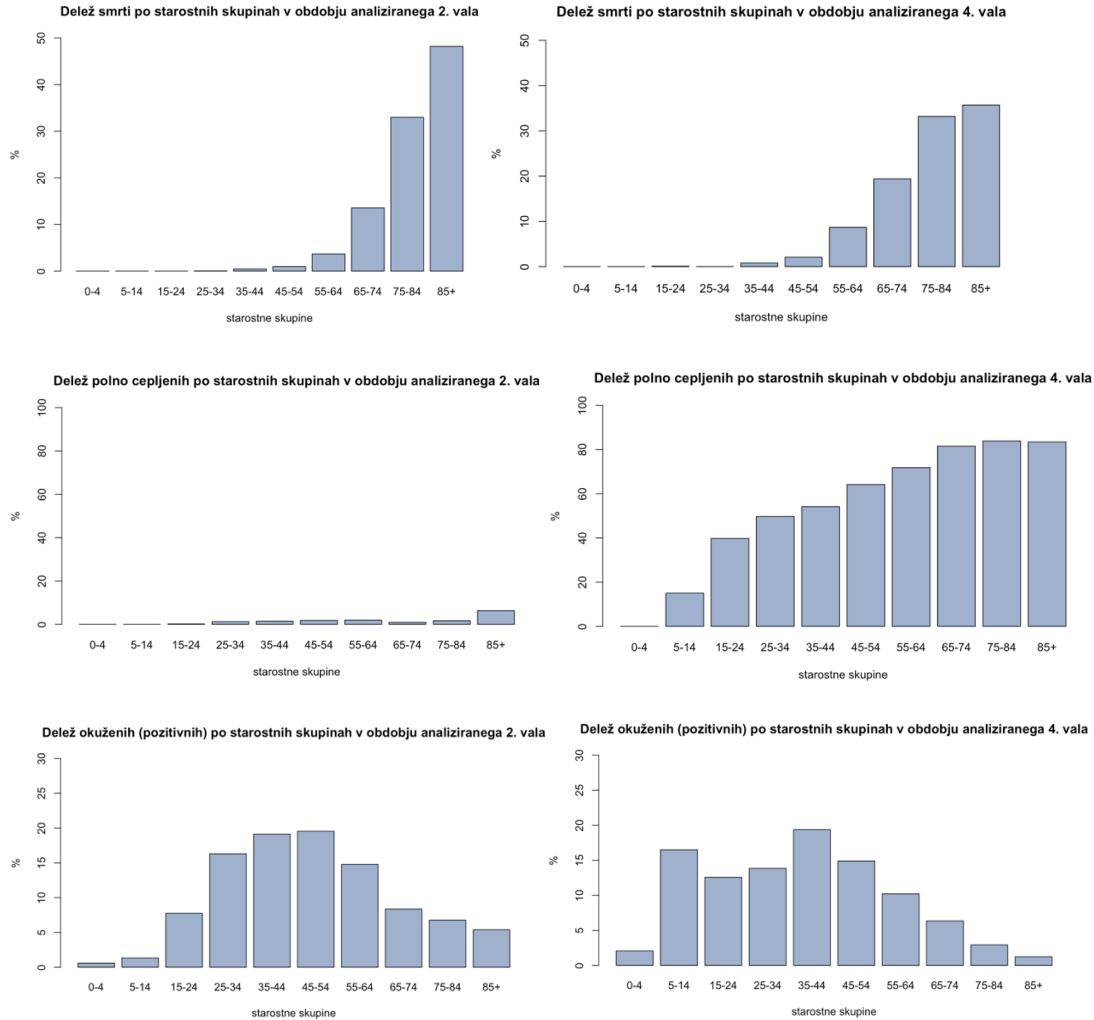
## Druga predpostavka

Rezultati so pokazali, da je se je v 4. valu v primerjavi z 2. valom zmanjšala povprečna starost umrlih, kar je bilo signifikantno in v korelaciji s postavljenou predpostavko.

Iz grafov je razvidno, da se je delež umrlih s starejšimi starostnimi skupinami zmanjšal.

Ker nisva imela podatkov o starosti umrlih, nisva mogla izračunati povprečne starosti umrlih.

Namreč tovrstni podatki za leto 2021 še niso dostopni v podatkovni bazi Statističnega urada Republike Slovenije (SURS, SiStat).



Po najinem mnenju je razlog za tak rezultat naslednji:

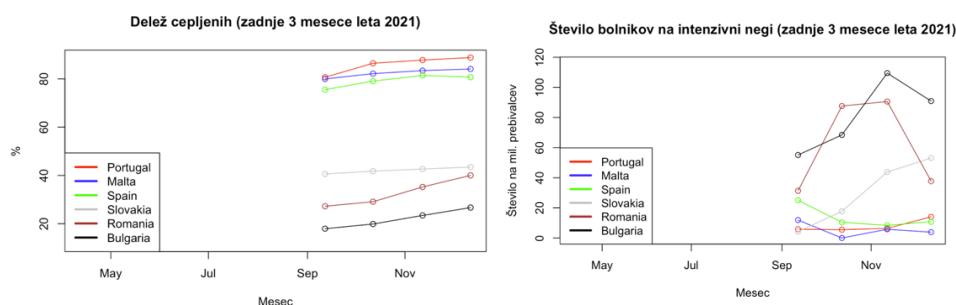
V starostni skupini nad 75 let je precepljenost prebivalcev Slovenije najvišje, kar pripomore k temu, da so manj dovzetni za okužbo in posledično je manj možnosti, da bodo umrli zaradi Covid-19.

### Tretja predpostavka

Rezultati so pokazali, da je v državah, kjer je večji delež cepljenih, manjši delež hospitalizacij na intenzivni negi, kar je bilo signifikantno in v korelacijski s postavljenou predpostavko.

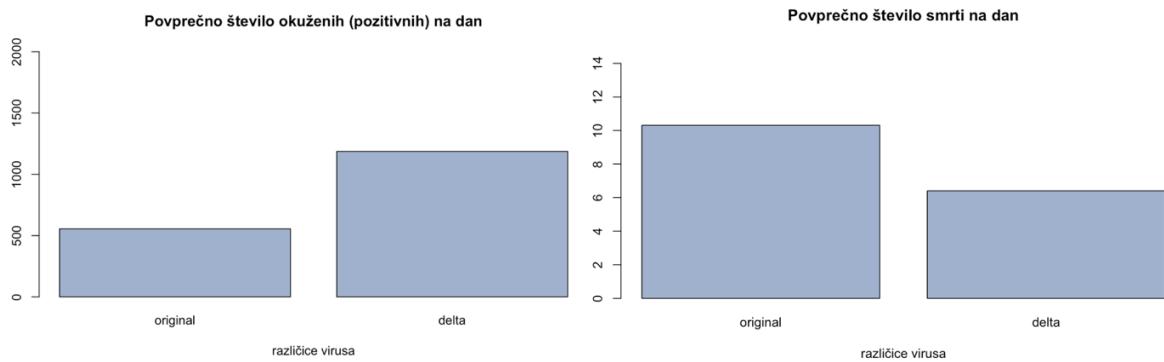
Tu velja omeniti, da sva izločila nekatere države, ker ni bilo na voljo podatkov o številu bolnikov na intenzivni negi.

Za analizo podatkov sva vzela le zadnje 3 mesece leta 2021, da sva med seboj lahko primerjala iste države (poleti in spomladji so so si države »mesta« po deležu cepljenih malce izmenjavala).



## Četrta predpostavka

Rezultati so pokazali, da je različica delta bolj kužna a manj smrtonosna od originalne različice virusa Covid-19, kar je bilo signifikantno in v korelaciji s postavljenou predpostavko.



## Zaključek

V seminarški nalogi sva se naučila precej o obdelavi in analizi podatkov. Spoznala sva, da je le malo podatkovnih zbirk kakovostnih in posledično primernih za obdelavo in analizo.

Presenetilo naju je tudi, da na oddaljenem strežniku Microsoft Azure hitro zmanjka creditov in postane plačljivo (kot študenti imamo samo 100 kreditov za uporabo v enem letu po aktivaciji računa).

 Free trial subscriptions can move to pay as you go to get Azure Database for MySQL free for 12 months. Each month you get 750 hours free with Burstable B1MS instance and 32 GiB storage. At the end of 12 months you'll be billed at pay-as-you-go rates for any usage. [Learn more](#)

Kar bi bilo še zanimivo, da bi se analiziralo:

- primerjava poteka Covid-19 epidemije med Slovenijo in svetom
- analiza učinkov epidemioloških ukrepov in omejitve na število okuženih in umrlih
- analiza, koliko več ljudi bi umrlo, če v obdobju 4. vala, ne bi imeli cepiva

## Slovarček neznanih besed

Angleška kratica	Angleška beseda	Razlaga pojma
/	PowerDesigner	orodje za izdelavo modelov (npr. konceptualni model)
	Rstudio	orodje za analizo podatkov
JSON	JavaScript Object Notation	odprt standardni format, ki omogoča prenos podatkov po spletu
	ASP.NET Web API	za gradnjo REST API-jev na katerekoli platformi s C#
	Swagger	dokumentacija API-jev in program za izvajanje klicev API-jev
	Postman	program za izvajanje klicev API-jev
DBMS	Database Management System	sistem za upravljanje s podatkovnimi bazami, primer je MySQL
	MySQL	odprtakodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL
	CRON jobs	omogoča avtomatizacijo zaganjanja enot programske opreme
	endpoint	dostopna točka na zaledju
API	Application Programming Interface	programski vmesnik
REST	Representational State Transfer	arhitekturni pristop za izdelavo omrežnih sistemov

DI	Dependency Injection	tehnika ustvarjanja odvisnosti med datotekami in datotečnimi sistemi
SOC	Separation of Concerns	pristop, pri katerem se razdeli shema kodne zbirke na poglavja, direktorije
LSA	Layer-Structured-Architecture	razdelitev celotnega datotečnega sistema glede na tip logike (poslovna, ...)
DTO	Data Transfer Object	vmesnik za podatkovne modele
BasicAuth	Basic authentication	sistem verifikacije z uporabniškim imenom in gesлом

## Viri

- <https://covid-19.sledilnik.org/sl/stats>
- <https://coronavirus.jhu.edu>
- <https://restcountries.com>