Koronca7

March 26, 2020

Avtor: Tim Kmecl

[1]:
```
import numpy as np
import seaborn as sns
import math
import matplotlib.pyplot as plt
from matplotlib import cm
import scipy.optimize
import pandas as pd
```

[2]:
```
%config InlineBackend.figure_format = 'retina'
```

[3]:
```
sns.set("paper")
sns.set()
sns.set_style("whitegrid")
sns.set_style("whitegrid", {"axes.facecolor": ".995",
                            'axes.spines.left': False,
                            'axes.spines.top': False,
                            'axes.spines.right': False,
                            'axes.edgecolor': '.5',
                            'axes.grid': True,
                            'grid.color': '.85'})

plt.rcParams["figure.figsize"] = [6.4, 4.8]

plt.rcParams["lines.linewidth"] = 2

plt.rcParams["lines.marker"] = "o"
plt.rcParams["lines.markeredgecolor"] = "w"
plt.rcParams["lines.markeredgewidth"] = "2"
plt.rcParams["lines.markersize"] = 6
palette = "Blues_d"

plt.rcParams['figure.dpi'] = 100

blues_d = cm.Blues(np.linspace(.3, 1, 5))
blues_dd = cm.Blues(np.linspace(.3, 1, 5))
```

```
cmap = np.concatenate((cm.Blues(np.linspace(0.3, 1, 5))[:,:3], 0.3 +  cm.
↪plasma(np.linspace(0.15, 1, 7))[:,:3]*0.7))
cm_stran = cmap[[4, 2, 0, 5,6,7,8,9,10,11, 1, 3]]
cmap = cmap[[4, 3, 2, 1, 0, 5,6,7,8,9,10,11]]

save_dpi=150
cur_img = 0
```

[4]:

```
def utils_plot1(ax, log, title, ylim, xticks):
    if xticks is not None:
        ax.set_xticks(xticks)
    ax.set_title(title + (" [log]" if log else ""))
    ax.set_ylim(0 if (ylim==None and not log) else (0.95 if ylim==None else␣
↪ylim))
    ax.set_yscale("log" if log else "linear")
    ax.xaxis.set_tick_params(labelbottom=True)
    plt.subplots_adjust(wspace=0.13, top=0.91, left=.06, right=.93, bottom=.08)
    ax.set(xlabel=None)
    ax.set_ylabel('')
    ax.set_xlabel('')
    if log:
        ax.grid(axis="y", which="both")

def plot_bar_m1(ax, x, y, log=False, title=None, ylim=None, xticks=None,␣
↪x2=None, y2=None, width=(0.6, 0.5, 0.1), show=True, x3=None, y3=None):
    if not x2 is None and not y2 is None:
        ax.bar(x2, y2, color="lightblue", width=(width[1] if len(width)>1 else␣
↪width[0]))
    if len(x) and len(y):
        ax.bar(x, y, color="b", width=width[0])
    if x3 is not None and y3 is not None:
        ax.bar(x3, y3, width=(width[2] if len(width)>2 else width[0]), color=".
↪25", alpha=.7, linewidth=0)

    utils_plot1(ax, log, title, ylim, xticks)

def plot_line1(ax, x, y, log=False, title=None, ylim=None, xticks=None,␣
↪x2=None, y2=None, m2=True, show=True, color=None):
    if x2 is not None and y2 is not None:
        sns.lineplot(x2, y2, color=("lightblue" if m2 else "orange"),␣
↪marker=(plt.rcParams["lines.marker"] if m2 else ""), ax=ax)
    if len(x) and len(y):
        sns.lineplot(x, y, color=("b" if not color else color), ax=ax)

    utils_plot1(ax, log, title, ylim, xticks)
```

```
def plot_bar1(ax, x, y, log=False, title=None, ylim=None, xticks=None, x2=None,␣
 ↪y2=None):
    if not x2 is None and not y2 is None:
        sns.lineplot(x2, y2, color="orange", marker="", ax=ax)
    if len(x) and len(y):
        ax.bar(x, y, color="b", width=0.7)

    utils_plot1(ax, log, title, ylim, xticks)
```

[17]:

```
def cat(l):
    return np.concatenate(l)

def reverse_legend(ax, labels1=None):
    handles, labels = ax.get_legend_handles_labels()
    if labels1 is None:
        ax.legend(handles[::-1], labels[::-1])
    else:
        ax.legend(handles[::-1], labels1[::-1])
```

1 Podatki

Vir 1: NIJZ *[Spremljanje koronavirusa SARS-CoV-2 (COVID-19)]*
Vir 2: prirejeno po NIJZ: *[Dnevno spremljanje okužb s SARS-CoV-2 (COVID-19) [graf]]*
Vir 3: Objave vlade na gov.si in twitterju.
Vir 4: Luka Renko et al., COVID-19 Slovenija

Na grafih je *vir 2* temno moder, *vir 1* svetlo moder, *vir 3* pa siv.

- pri *vir 1* so podatki na začetku za ob 14h, od dneva 11 naprej ob ~10-14h na določen dan, zato so na grafu zamaknjeni za 0.5 dneva v levo

- pri *vir 2* so podatki na začetku za konec dneva, od dneva 15 naprej naslednji dan ob 10h

- pri *vir 3* ni točno zabeleženo, na katero obdobje se podatki nanašajo, objavljeno enkrat popoldne

- *vir 4* samo za hospitalizacije ipd. (neuradno!), in po regijah ter starostirh (uradno, posredno iz *vir 2*)

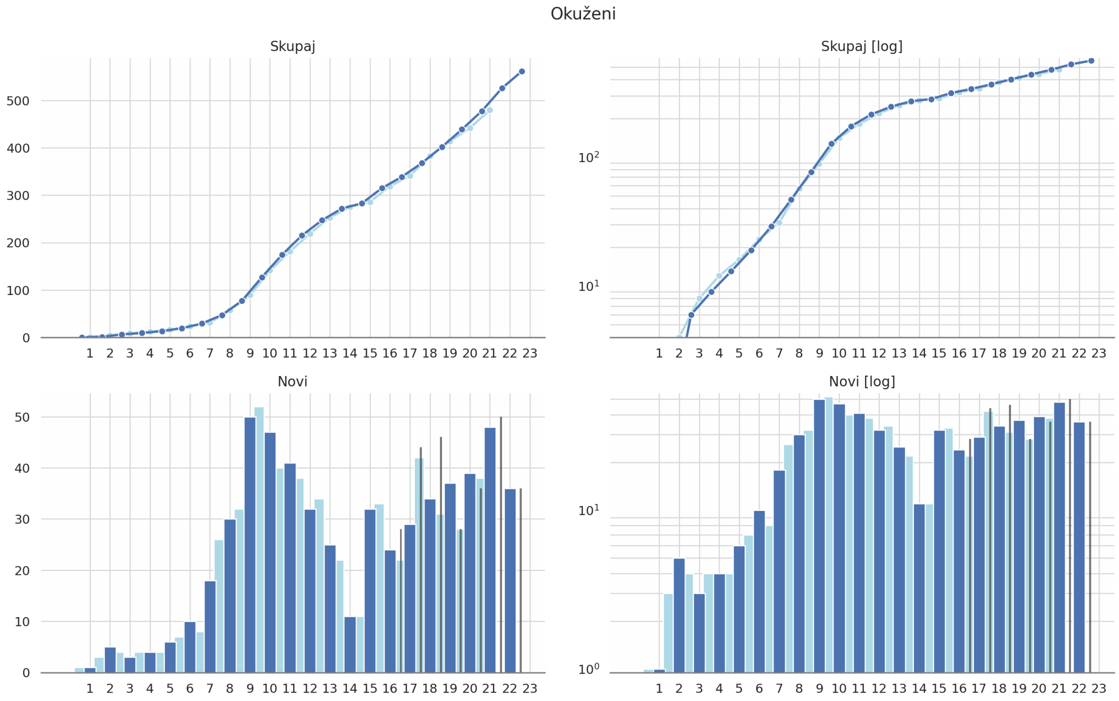
Dan 1 je *4.3.*, zadnjič posodobljeno *26.3.* (zadnji celodnevni podatek velja za en dan nazaj).

[60]:
```
df = pd.read_csv("./korona_slo.csv",  index_col="dan")
#df['datum'] = pd.to_datetime(df['datum'], format='%d.%m.')
```

1.1 Podatki o okuženih

[61]:
```
df["oku_novi_0h"] = df.oku_0h.diff()
df["oku_novi_10h"] = df.oku_10h.diff()
df["oku_novi_14h"] = df.oku_14h.diff()
```

[62]:
```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize=(15, 9),␣
 ↪sharex="all")
fig.suptitle("Okuženi")
dfp = df[4:]
plot_line1(ax1, dfp.index-.4, dfp.oku_0h, False, "Skupaj", xticks=dfp.index,
          x2=dfp.index, y2=dfp.oku_14h)
plot_line1(ax2, dfp.index-.4, dfp.oku_0h, True, "Skupaj", xticks=dfp.index,
          x2=dfp.index, y2=dfp.oku_14h, ylim=4)

plot_bar_m1(ax3, dfp.index-1, dfp.oku_novi_0h, False, "Novi", xticks=dfp.index,
          x2=dfp.index + .45-1, y2=dfp.oku_novi_14h, x3=dfp.index+.55-1, y3=dfp.
 ↪oku_novi_gov)
plot_bar_m1(ax4, dfp.index-1, dfp.oku_novi_0h, True, "Novi", xticks=dfp.index,
          x2=dfp.index + .45-1, y2=dfp.oku_novi_14h, x3=dfp.index+.55-1, y3=dfp.
 ↪oku_novi_gov)
```



1.2 Rast

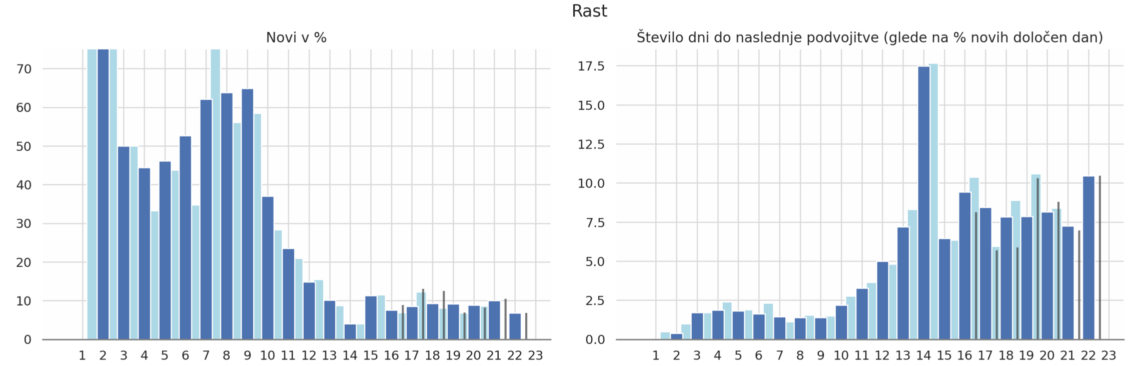
[63]:

```
df["rast_0h"] = [None] + [df.oku_0h[i]/df.oku_0h[i-1] for i in df.index[1:]]
df["rast_14h"] = [None] + [df.oku_14h[i]/df.oku_14h[i-1] for i in df.index[1:]]
df["rast_gov"] = [None] + [1 + df.oku_novi_gov[i]/df.oku_0h[i-1] for i in df.
 ↪index[1:]]
df["podvojitev_0h"] = math.log(2)/np.log(df["rast_0h"])
df["podvojitev_14h"] = math.log(2)/np.log(df["rast_14h"])
df["podvojitev_gov"] = math.log(2)/np.log(df["rast_gov"])
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: RuntimeWarning:
invalid value encountered in long_scalars
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: RuntimeWarning:
divide by zero encountered in long_scalars
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: RuntimeWarning:
invalid value encountered in double_scalars

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: RuntimeWarning:
divide by zero encountered in double_scalars
```

[64]:

```
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(15, 5), sharex=True)
fig.suptitle("Rast", y=0.96)
dfp = df[4:]
plot_bar_m1(ax1, dfp.index-1, (dfp.rast_0h-1)*100, False, "Novi v %",␣
 ↪ylim=(0,75),
          x2=dfp.index - .55, y2=(dfp.rast_14h-1)*100, x3=dfp.index-.45,␣
 ↪y3=(dfp.rast_gov-1)*100, xticks=dfp.index)
plot_bar_m1(ax2, dfp.index-1, dfp.podvojitev_0h, False,
           "Število dni do naslednje podvojitve (glede na % novih določen dan)",
          x2=dfp.index - .55, y2=dfp.podvojitev_14h, x3=dfp.index-.45, y3=dfp.
 ↪podvojitev_gov, xticks=dfp.index)
plt.subplots_adjust(top=0.85, bottom=.15)
```



1.3 Podatki o testih

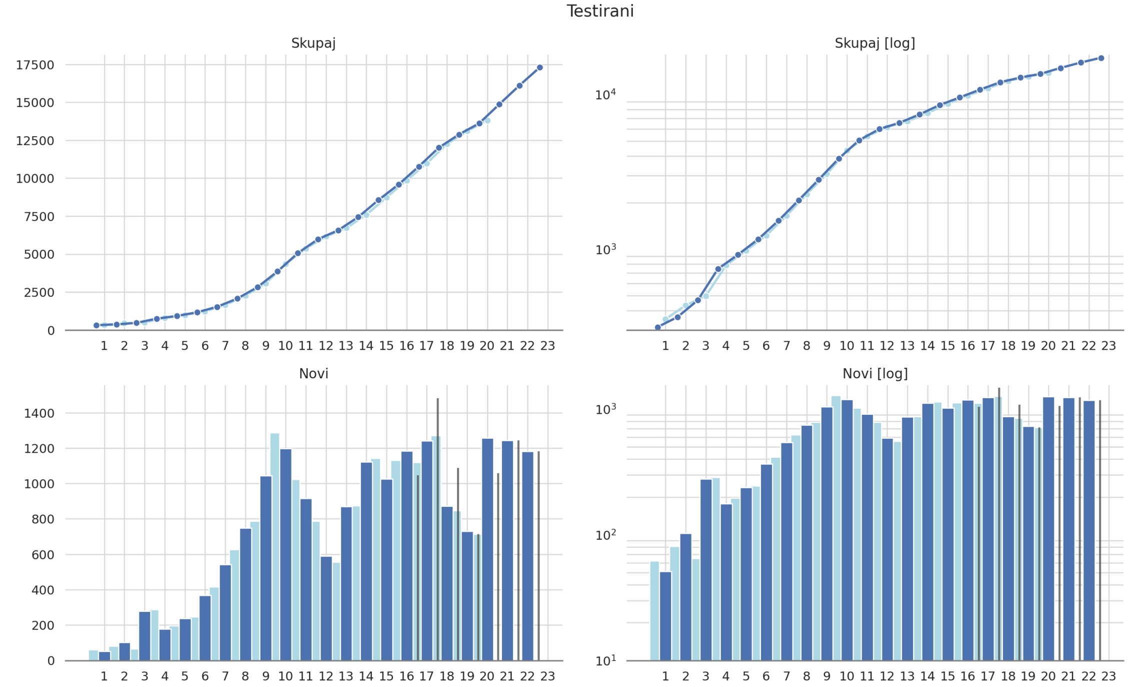
[65]:

```
df["testi_novi_0h"] = df.testi_0h.diff()
df["testi_novi_14h"] = df.testi_14h.diff()
```

[66]:

```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize=(15, 9),␣
 ↪sharex=True)
fig.suptitle("Testirani")
dfp = df[4:]

plot_line1(ax1, dfp.index-.4, dfp.testi_0h, False, "Skupaj", xticks=dfp.index,
          x2=dfp.index, y2=dfp.testi_14h)
plot_line1(ax2, dfp.index-.4, dfp.testi_0h, True, "Skupaj", xticks=dfp.index,
          x2=dfp.index, y2=dfp.testi_14h, ylim=300)

plot_bar_m1(ax3, dfp.index-1, dfp.testi_novi_0h, False, "Novi", xticks=dfp.
 ↪index,
          x2=dfp.index + .45-1, y2=dfp.testi_novi_14h, x3=dfp.index+.55-1,␣
 ↪y3=dfp.testi_novi_gov)
```

```
plot_bar_m1(ax4, dfp.index-1, dfp.testi_novi_0h, True, "Novi", xticks=dfp.index,
          x2=dfp.index + .45-1, y2=dfp.testi_novi_14h, x3=dfp.index+.55-1,␣
 ↪y3=dfp.testi_novi_gov, ylim=10)
```

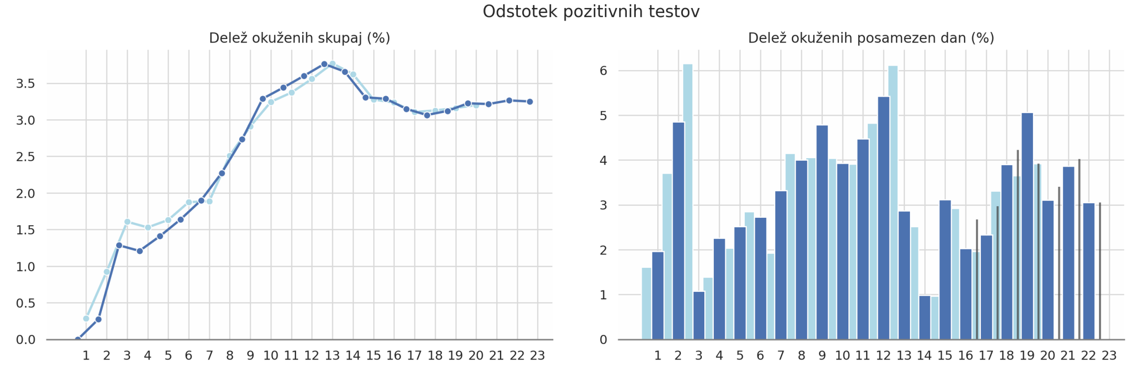


1.4 Razmerja med testi in okuženimi

[67]:
```
df["testi_procent_0h"] = df.oku_0h/df.testi_0h
df["testi_procent_14h"] = df.oku_14h/df.testi_14h

df["testi_novi_procent_0h"] = df.oku_novi_0h/df.testi_novi_0h
df["testi_novi_procent_14h"] = df.oku_novi_14h/df.testi_novi_14h
df["testi_novi_procent_gov"] = df.oku_novi_gov/df.testi_novi_gov
```

[68]:
```
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(15, 5), sharex=True)
fig.suptitle("Odstotek pozitivnih testov", y=0.96)
dfp = df[4:]

plt.subplots_adjust(top=0.85, bottom=.15)

plot_line1(ax1, dfp.index-.4, dfp.testi_procent_0h*100, False, "Delež okuženih␣
 ↪skupaj (%)", xticks=dfp.index,
          x2=dfp.index, y2=dfp.testi_procent_14h*100)
```

```
plot_bar_m1(ax2, dfp.index-1, dfp.testi_novi_procent_0h*100, False, "Delež␣
↪okuženih posamezen dan (%)", xticks=dfp.index,
          x2=dfp.index + .45-1, y2=dfp.testi_novi_procent_14h*100, x3=dfp.
↪index+.55-1, y3=dfp.testi_novi_procent_gov*100)
plt.subplots_adjust(top=0.85, bottom=.15)
```



1.5 Hospitalizacije, regije in starost *(vir 4)*

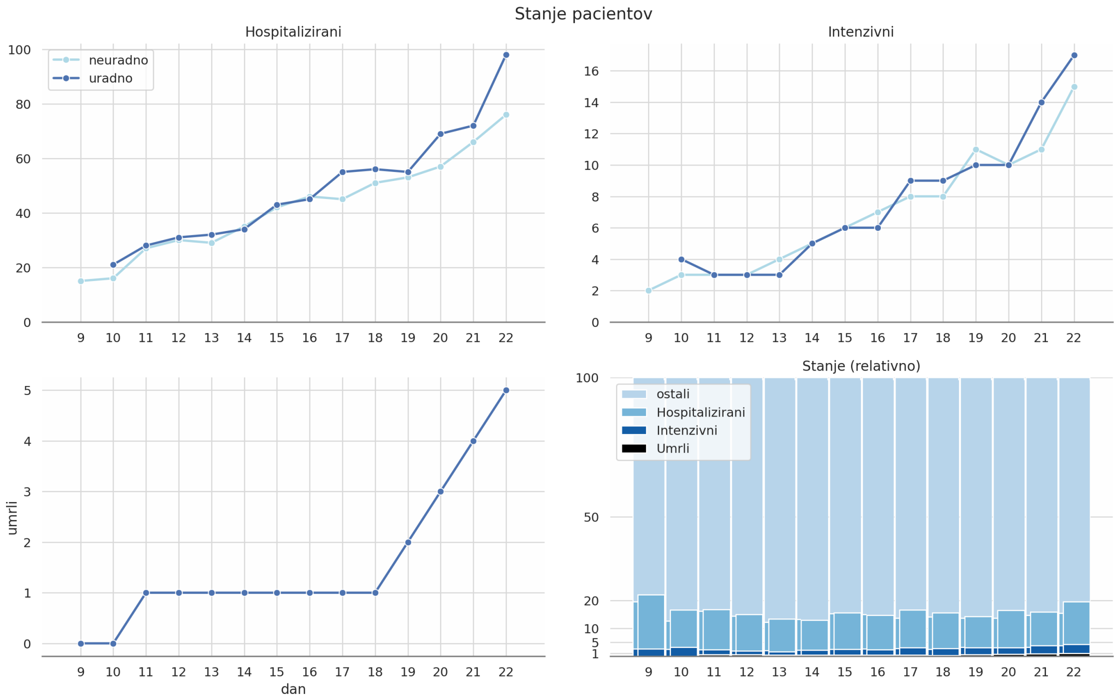
1.5.1 Hospitalizacije

[69]:
```
df["hospitalizirani_rel"] = df["hospitalizirani"]/df["oku_0h"]
df["intenzivni_rel"] = df["intenzivni"]/df["oku_0h"]
df["umrli_rel"] = df["umrli"]/df["oku_0h"]

df["hospitalizirani_neur_rel"] = df["hospitalizirani_neur"]/df["oku_0h"]
df["intenzivni_neur_rel"] = df["intenzivni_neur"]/df["oku_0h"]
```

[119]:
```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize=(15, 9),␣
↪sharex=True)
fig.suptitle("Stanje pacientov", y=0.96)
dfp = df[12:][pd.notna(df.hospitalizirani)]

plot_line1(ax1, dfp.index[1:], dfp["hospitalizirani"][1:], False,␣
↪"Hospitalizirani", xticks=dfp.index,
          x2=dfp.index, y2=dfp["hospitalizirani_neur"])
plot_line1(ax2, dfp.index[1:], dfp["intenzivni"][1:], False, "Intenzivni",␣
↪xticks=dfp.index,
          x2=dfp.index, y2=dfp["intenzivni_neur"])

ax1.legend(["neuradno", "uradno"])

sns.lineplot(x=dfp.index, y=dfp.umrli, color="b", ax=ax3)
```

```
utils_plot1(ax4, False, "Umrli", (0,100), dfp.index)


ax4.bar(x=dfp.index, height=100, width=.95, color=blues_d[0])
ax4.bar(x=dfp.index, height=dfp["hospitalizirani_neur_rel"]*100,␣
 ↪bottom=dfp["umrli_rel"]*100, width=.95, color=blues_d[1])
ax4.bar(x=dfp.index, height=dfp["intenzivni_neur_rel"]*100,␣
 ↪bottom=dfp["umrli_rel"]*100, width=.95, color=blues_d[3])
ax4.bar(x=dfp.index, height=dfp["umrli_rel"]*100, width=.95, color=[0,0,0])
off = .075
w=.82

ax4.bar(x=dfp.index+off, height=99, width=w, color=blues_d[0], linewidth=0)
ax4.bar(x=dfp.index+off, height=dfp["hospitalizirani_rel"]*100,␣
 ↪bottom=dfp["umrli_rel"]*100, width=w, color=blues_d[1])
ax4.bar(x=dfp.index+off, height=dfp["intenzivni_rel"]*100,␣
 ↪bottom=dfp["umrli_rel"]*100, width=w, color=blues_d[3])
ax4.bar(x=dfp.index, height=dfp["umrli_rel"]*100, width=.95, color=[0,0,0])


utils_plot1(ax4, False, "Stanje (relativno)", (0,100), dfp.index)
ax4.legend(["ostali", "Hospitalizirani", "Intenzivni", "Umrli"], loc="upper␣
 ↪left")
ax4.set_yticks([1,5,10,20, 50,  100])
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:3: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
  This is separate from the ipykernel package so we can avoid doing imports
until
```



1.5.2 Regije

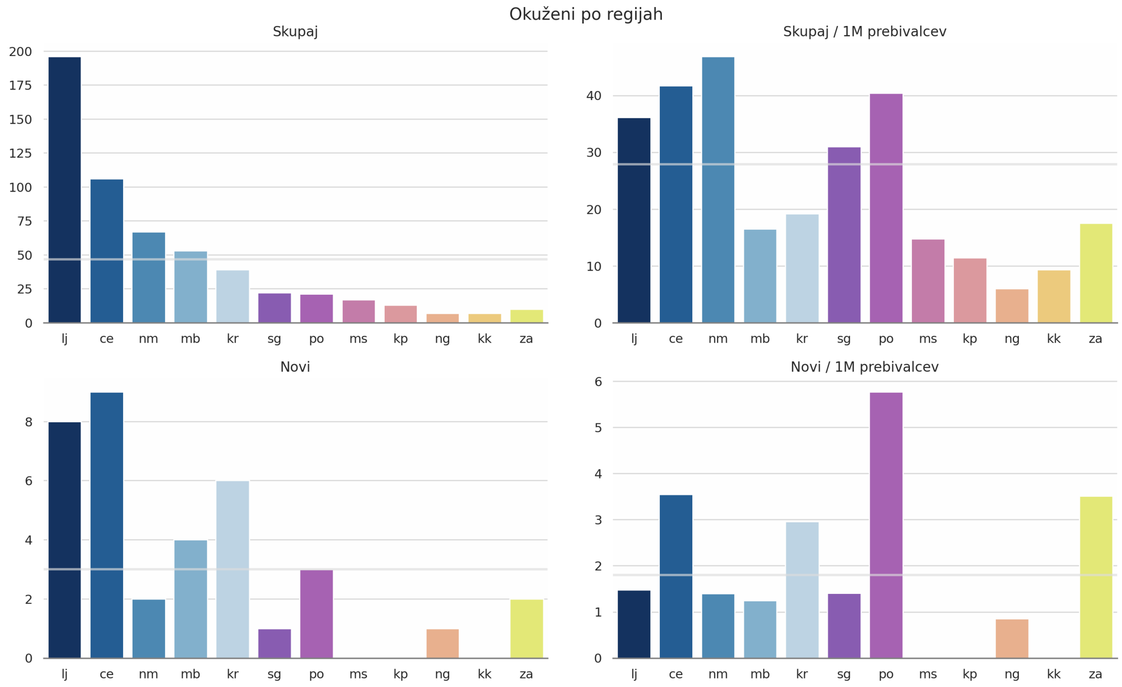
[106]:

```
# Regije
regije_populacija = np.array([542, 254, 143, 322, 203, 71, 52, 115, 114, 117,␣
 ↪75, 57])/100
regije_polno1 = ["Osrednjeslovenska", "Savinjska", "Jugovzhodna S.",␣
 ↪"Podravska", "Gorenjska",
                "Koroška", "Primorsko-notr.", "Pomurska", "Obala", "Goriška",␣
 ↪"Posavska", "Zasavska"]

regije = ["lj", "ce", "nm", "mb", "kr", "sg", "po", "ms", "kp", "ng", "kk",␣
 ↪"za"]
regije_polno = ["{} ({})".format(regije_polno1[i], regije[i]) for i in␣
 ↪range(len(regije))]

regije_rel = [r + "_rel" for r in regije]
regije_popul = [r + "_popul" for r in regije]
regije_popul_rel = [r + "_rel" for r in regije_popul]

regije_novi = [r + "_novi" for r in regije]
regije_novi_rel = [r + "_rel" for r in regije_novi]
regije_novi_popul = [r + "_popul" for r in regije_novi]
regije_novi_popul_rel = [r + "_rel" for r in regije_novi_popul]
```

```
df[regije_rel] = df[regije].div(df[regije].sum(axis=1), axis=0)*100
df[regije_popul] = df[regije]/regije_populacija
df[regije_popul_rel] = df[regije_popul].div(df[regije_popul].sum(axis=1),␣
 ↪axis=0)*100

df[regije_novi] = df[regije].diff()
df[regije_novi_rel] = df[regije_novi].div(df[regije_novi].sum(axis=1),␣
 ↪axis=0)*100
df[regije_novi_popul] = df[regije_novi]/regije_populacija
df[regije_novi_popul_rel] = df[regije_novi_popul].div(df[regije_novi_popul].
 ↪sum(axis=1), axis=0)*100
```

[107]:

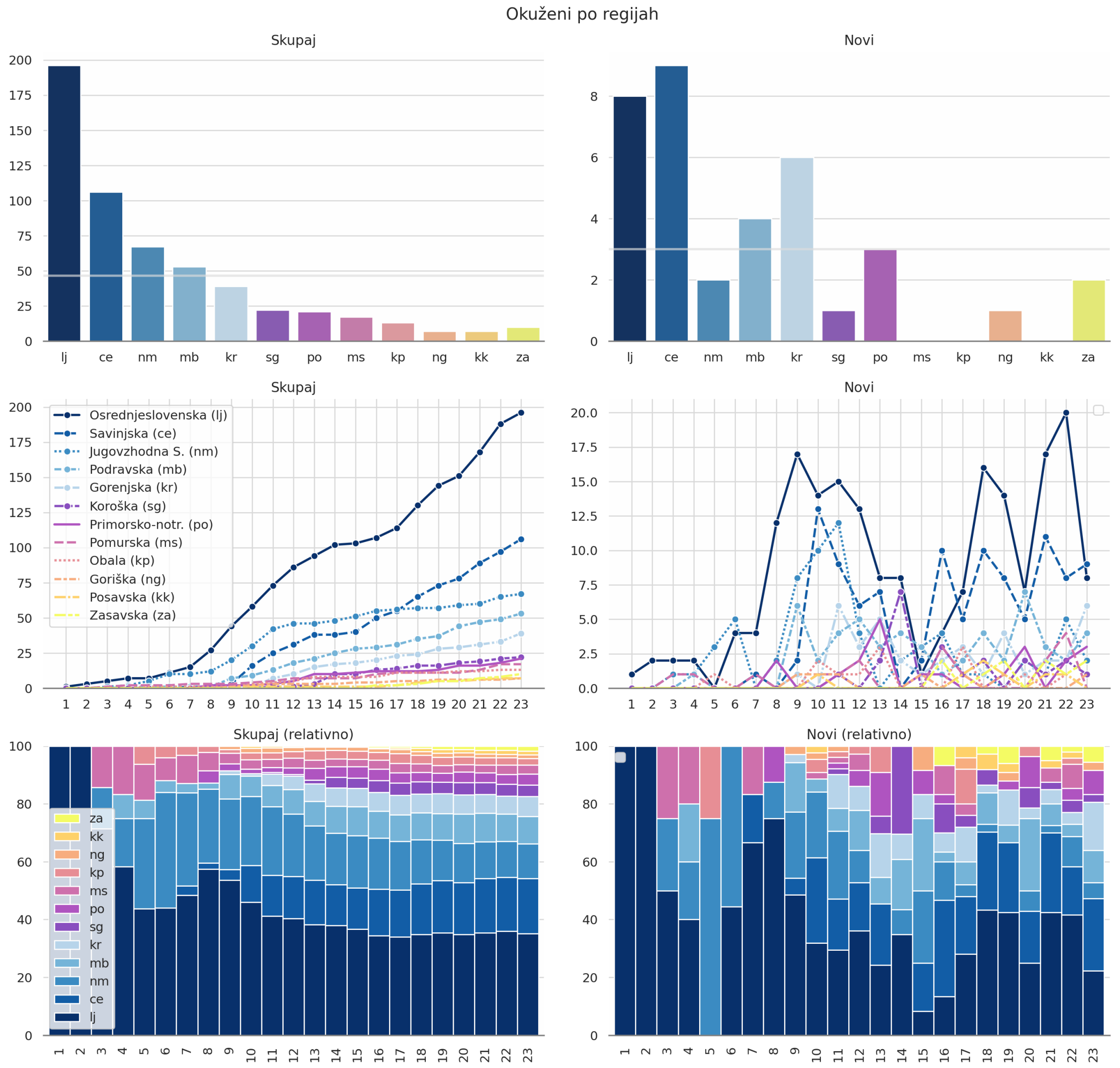
```
ind = 21

fig, ((ax5, ax6), (ax7, ax8)) = plt.subplots(nrows=2, ncols=2, figsize=(15, 9),␣
 ↪sharex="all")
fig.suptitle("Okuženi po regijah", y=0.96)

dfp = df[4:]

sns.barplot(ax=ax5, data=dfp[regije][-1:], palette=cmap)
ax5.axhline(dfp[regije].sum(axis=1).to_numpy()[-1]/12, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax5, False, "Skupaj", None, None)

sns.barplot(ax=ax6, data=dfp[regije_popul][-1:], palette=cmap)
ax6.axhline(dfp[regije].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax6, False, "Skupaj / 1M prebivalcev", None, None)

sns.barplot(ax=ax7, data=dfp[regije_novi][-1:], palette=cmap)
ax7.axhline(dfp[regije_novi].sum(axis=1).to_numpy()[-1]/12, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax7, False, "Novi", None, None)

sns.barplot(ax=ax8, data=dfp[regije_novi_popul][-1:], palette=cmap)
ax8.axhline(dfp[regije_novi].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax8, False, "Novi / 1M prebivalcev", None, None)
ax8.set_xticklabels(regije)
plt.show()
```



[108]:

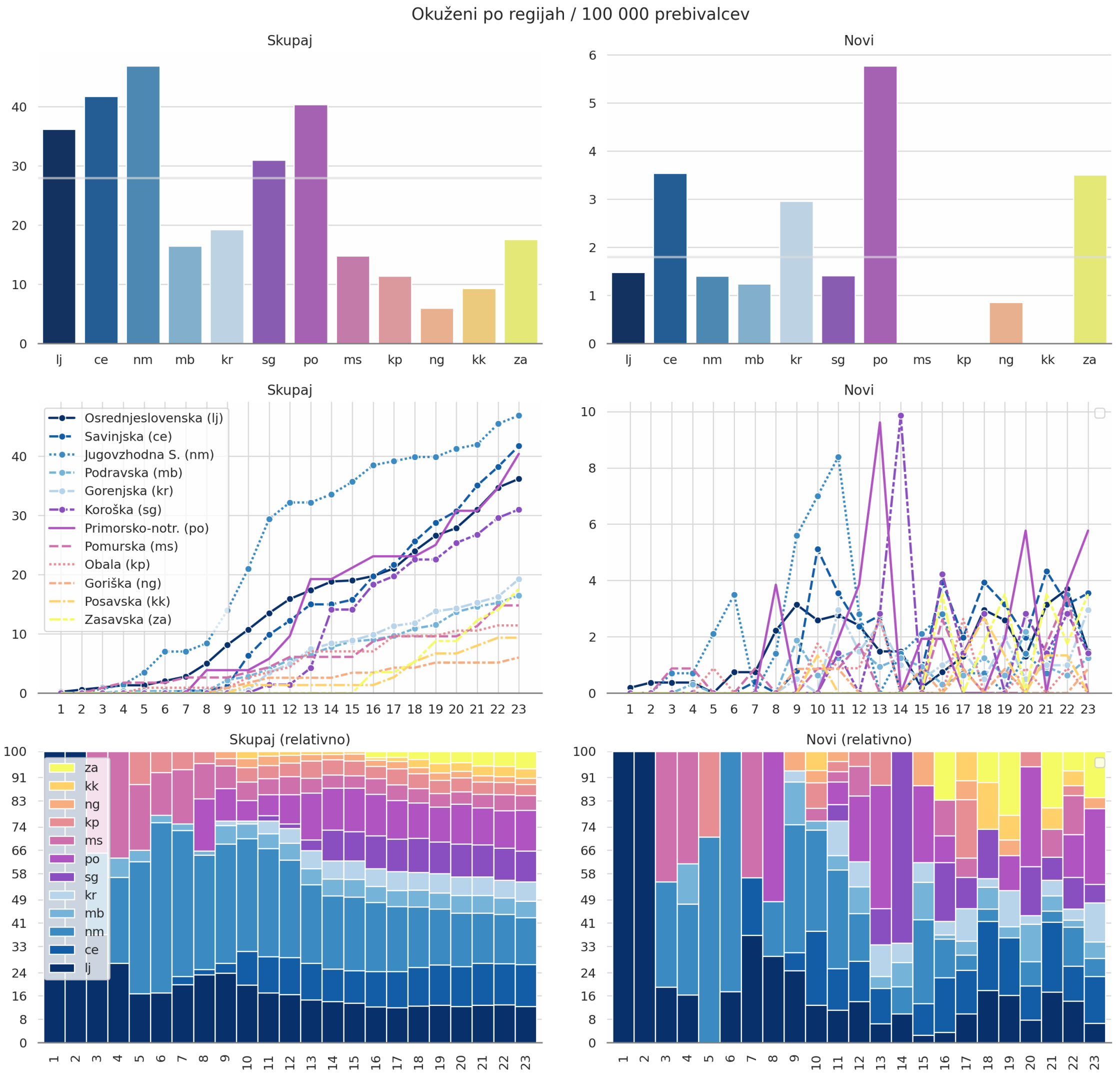
```
current_palette = sns.color_palette()
sns.set_palette(cmap)

fig, ((ax5, ax6), (ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=3, ncols=2, figsize=(15, 14))
fig.suptitle("Okuženi po regijah")

dfp = df[4:]

sns.barplot(ax=ax5, data=dfp[regije][-1:], palette=cmap)
ax5.axhline(dfp[regije].sum(axis=1).to_numpy()[-1]/12, color="#ddda", linewidth=2, marker="")
utils_plot1(ax5, False, "Skupaj", None, None)
ax5.set_xticklabels(regije)

sns.barplot(ax=ax6, data=dfp[regije_novi][-1:], palette=cmap)
ax6.axhline(dfp[regije_novi].sum(axis=1).to_numpy()[-1]/12, color="#ddda", linewidth=2, marker="")
utils_plot1(ax6, False, "Novi", None, None)
ax6.set_xticklabels(regije)


sns.lineplot(data=dfp[regije[:6]], ax=ax1)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije[6:]], marker="", ax=ax1)
```

```
ax1.legend(regije_polno)
leg = ax1.get_legend()
for i, h in enumerate(leg.legendHandles):
    h.set_color(cmap[i])
utils_plot1(ax1, False, "Skupaj", None, dfp.index)
sns.set_palette(cmap)


sns.lineplot(data=dfp[regije_novi[:6]], ax=ax2)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije_novi[6:]], marker="", ax=ax2)
ax2.legend("")
utils_plot1(ax2, False, "Novi", None, dfp.index)


dfp[regije_rel].plot(color=cmap, kind="bar", stacked=True, width=1,␣
 ↪ylim=(0,100), ax=ax3)
reverse_legend(ax3, regije)
utils_plot1(ax3, False, "Skupaj (relativno)", None, None)

dfp[regije_novi_rel].plot(color=cmap, kind="bar", stacked=True, width=1,␣
 ↪ylim=(0,100), ax=ax4)
ax4.legend("")
utils_plot1(ax4, False, "Novi (relativno)", None, None)

sns.set_palette(current_palette)
plt.subplots_adjust(top=0.94)
```



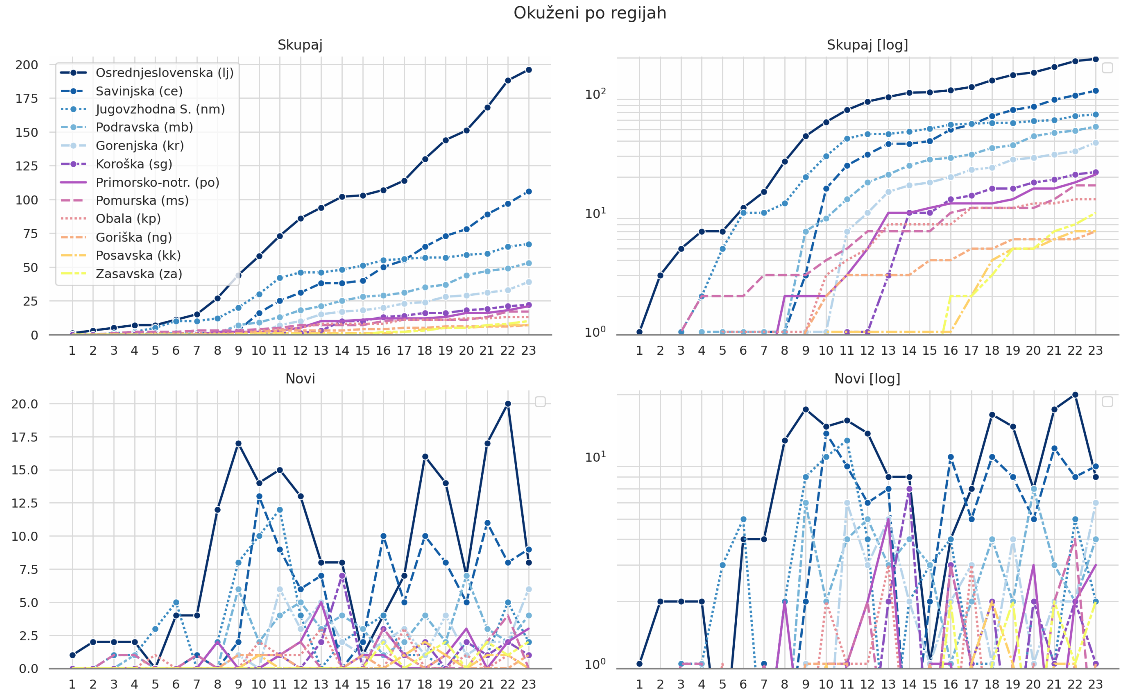
```
[109]: current_palette = sns.color_palette()
       sns.set_palette(cmap)

       fig, ((ax5, ax6),(ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=3, ncols=2,␣
        ↪figsize=(15, 14))
       fig.suptitle("Okuženi po regijah / 100 000 prebivalcev")

       dfp = df[4:]

       sns.barplot(ax=ax5, data=dfp[regije_popul][-1:], palette=cmap)
       ax5.axhline(dfp[regije].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
        ↪linewidth=2, marker="")
       utils_plot1(ax5, False, "Skupaj", None, None)
       ax5.set_xticklabels(regije)
```

```
sns.barplot(ax=ax6, data=dfp[regije_novi_popul][-1:], palette=cmap)
ax6.axhline(dfp[regije_novi].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax6, False, "Novi", None, None)
ax6.set_xticklabels(regije)


sns.lineplot(data=dfp[regije_popul[:6]], ax=ax1)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije_popul[6:]], marker="", ax=ax1)
ax1.legend(regije_polno)
leg = ax1.get_legend()
for i, h in enumerate(leg.legendHandles):
    h.set_color(cmap[i])
sns.set_palette(cmap)
utils_plot1(ax1, False, "Skupaj", None, dfp.index)

sns.lineplot(data=dfp[regije_novi_popul[:6]], ax=ax2)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije_novi_popul[6:]], marker="", ax=ax2)
ax2.legend("")
utils_plot1(ax2, False, "Novi", None, dfp.index)


dfp[regije_popul_rel].plot(color=cmap, kind="bar", stacked=True, width=1,␣
 ↪ylim=(0,100), ax=ax3)
reverse_legend(ax3, regije)
utils_plot1(ax3, False, "Skupaj (relativno)", None, None)
ax3.set_yticks([int(i/10) for i in range(0, 950, 83)] + [100])

dfp[regije_novi_popul_rel].plot(color=cmap, kind="bar", stacked=True, width=1,␣
 ↪ylim=(0,100), ax=ax4)
ax4.legend("")
utils_plot1(ax4, False, "Novi (relativno)", None, None)
ax4.set_yticks([int(i/10) for i in range(0, 950, 83)] + [100])

sns.set_palette(current_palette)
plt.subplots_adjust(top=0.94)
```



[110]:
```
current_palette = sns.color_palette()
sns.set_palette(cmap)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize=(15, 9),␣
 ↪sharex="all")
fig.suptitle("Okuženi po regijah")

dfp = df[4:]

sns.lineplot(data=dfp[regije[:6]], ax=ax1)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije[6:]], marker="", ax=ax1)

ax1.legend(regije_polno)
leg = ax1.get_legend()
```

```
for i, h in enumerate(leg.legendHandles):
    h.set_color(cmap[i])
utils_plot1(ax1, False, "Skupaj", None, dfp.index)
sns.set_palette(cmap)


sns.lineplot(data=dfp[regije[:6]].replace(0, 0.3), ax=ax2)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije[6:]].replace(0, 0.3), marker="", ax=ax2)

ax2.legend("")
utils_plot1(ax2, True, "Skupaj", None, dfp.index)
sns.set_palette(cmap)

sns.lineplot(data=dfp[regije_novi[:6]], ax=ax3)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije_novi[6:]], marker="", ax=ax3)
ax3.legend("")
utils_plot1(ax3, False, "Novi", None, dfp.index)

sns.set_palette(cmap)
sns.lineplot(data=dfp[regije_novi[:6]].replace(0, 0.3), ax=ax4)
sns.set_palette(cmap[6:,:])
sns.lineplot(data=dfp[regije_novi[6:]].replace(0, 0.3), marker="", ax=ax4)
ax4.legend("")
utils_plot1(ax4, True, "Novi", None, dfp.index)
sns.set_palette(current_palette)
```



1.5.3 Starost

[101]:

```
starosti_populacija = np.array([1., 2.11, 1.95, 2.53, 3.15, 3.01, 2.95, 2.22, 1.
 ↪39, 0.52])

starosti5 = ["0-15", "16-29", "30-49", "50-59", "60+"]
starosti10 = [str(i*10) for i in range(0, 10)]
starosti = cat((starosti5, starosti10))

starosti_rel = [r + "_rel" for r in starosti]
starosti_popul = [r + "_popul" for r in starosti10]

starosti5_novi = [r + "_novi" for r in starosti5]
starosti10_novi = [r + "_novi" for r in starosti10]
starosti_novi = cat((starosti5_novi, starosti10_novi))

starosti_novi_popul = [r + "_novi_popul" for r in starosti10]
starosti_novi_rel = [r + "_rel" for r in starosti_novi]

df[starosti_rel] = df[starosti].div(df[starosti].sum(axis=1), axis=0)*100
df[starosti_popul] = df[starosti10]/starosti_populacija

df[starosti_novi] = df[starosti].diff()
df[starosti_novi_rel] = df[starosti_novi].div(df[starosti_novi].sum(axis=1),␣
 ↪axis=0)*100
df[starosti_novi_popul] = df[starosti_novi[5:]]/starosti_populacija
```

[102]:

```
starosti10_skupine = ["0-24", "25-34", "35-54", "55-64", "65+"]

df["0-24"] = df["0"] + df["10"] + df["20"]
df["25-34"] = df["30"]
df["35-54"] = df["40"] + df["50"]
df["55-64"] = df["60"]
df["65+"] = df["70"] + df["80"] + df["90"]

starosti10_skupine_novi = [r + "_novi" for r in starosti10_skupine]
df[starosti10_skupine_novi] = df[starosti10_skupine].diff()
```

[103]:

```
avg_box = np.array([10, 22, 40, 55, 70])
avg_box1 = (np.array([5, 16, 30, 50, 60]) + avg_box)/2
avg_box2 = (np.array([15, 29, 49, 59, 90]) + avg_box)/2

avg_box10 = np.array([10*i for i in range(0, 10)])
```

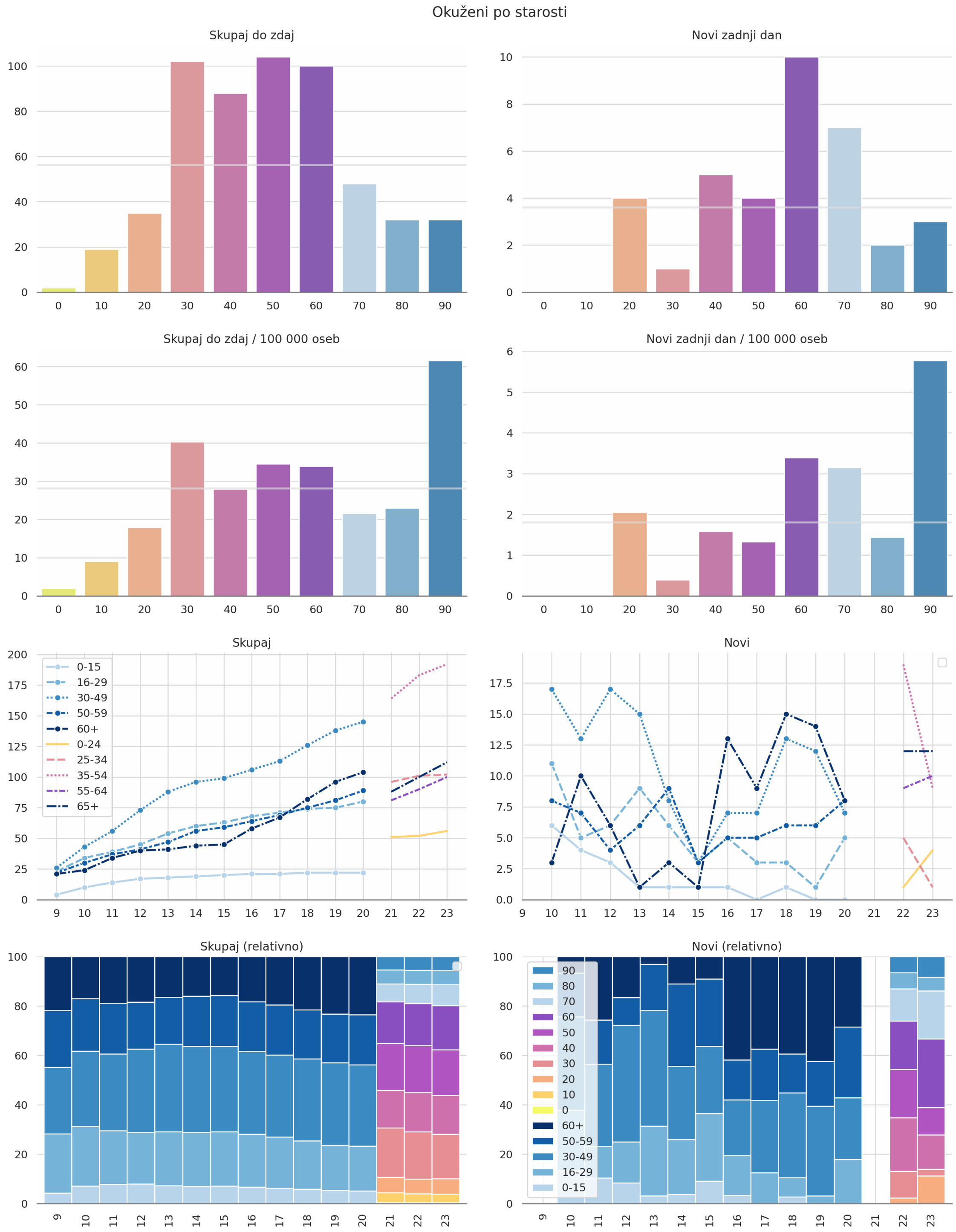
```
avg_box110 = np.array([0] + [10*i - 3 for i in range(1, 10)])
avg_box210 = np.array([10*i + 3 for i in range(0, 9)] + [95])

avg_starost = (df[starosti] * cat((avg_box, avg_box10))).sum(axis=1).
 ↪div(df[starosti].sum(axis=1), axis=0)
avg_starost1 = (df[starosti] * cat((avg_box1, avg_box110))).sum(axis=1).
 ↪div(df[starosti].sum(axis=1), axis=0)
avg_starost2 = (df[starosti] * cat((avg_box2, avg_box210))).sum(axis=1).
 ↪div(df[starosti].sum(axis=1), axis=0)
novi_avg_starost = (df[starosti_novi] * cat((avg_box, avg_box10))).sum(axis=1).
 ↪div(df[starosti_novi].sum(axis=1), axis=0)
novi_avg_starost1 = (df[starosti_novi] * cat((avg_box1, avg_box110))).
 ↪sum(axis=1).div(df[starosti_novi].sum(axis=1), axis=0)
novi_avg_starost2 = (df[starosti_novi] * cat((avg_box2, avg_box210))).
 ↪sum(axis=1).div(df[starosti_novi].sum(axis=1), axis=0)
```

[104]:

```
current_palette = sns.color_palette()
sns.set_palette(blues_dd)
fig, ((ax5, ax7), (ax6, ax8), (ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=4,␣
 ↪ncols=2, figsize=(15, 19))
fig.suptitle("Okuženi po starosti")



dfp = df[12:]

sns.barplot(ax=ax5, data=dfp[starosti[5:]][-1:], palette=cmap[::-1])
ax5.axhline(dfp[starosti].sum(axis=1).to_numpy()[-1]/10, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax5, False, "Skupaj do zdaj", None, None)
ax5.set_xticklabels(starosti10)

sns.barplot(ax=ax6, data=dfp[starosti_popul][-1:], palette=cmap[::-1])
ax6.axhline(dfp[starosti].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax6, False, "Skupaj do zdaj / 100 000 oseb", None, None)
ax6.set_xticklabels(starosti10)

sns.barplot(ax=ax7, data=dfp[starosti_novi[5:]][-1:], palette=cmap[::-1])
ax7.axhline(dfp[starosti_novi].sum(axis=1).to_numpy()[-1]/10, color="#ddda",␣
 ↪linewidth=2, marker="")
utils_plot1(ax7, False, "Novi zadnji dan", None, None)
ax7.set_xticklabels(starosti10)

sns.barplot(ax=ax8, data=dfp[starosti_novi_popul][-1:], palette=cmap[::-1])
ax8.axhline(dfp[starosti_novi].sum(axis=1).to_numpy()[-1]/20, color="#ddda",␣
 ↪linewidth=2, marker="")
```

```
utils_plot1(ax8, False, "Novi zadnji dan / 100 000 oseb", None, None)
ax8.set_xticklabels(starosti10)

'''
cmapr = cmap[::-1]
sns.lineplot(data=dfp[starosti5], ax=ax1)
sns.set_palette(cmapr[5:])
sns.lineplot(data=dfp[starosti10[5:]], ax=ax1, marker="")
sns.set_palette(cmapr)
sns.lineplot(data=dfp[starosti10[:5]], ax=ax1)

ax1.legend(starosti5)
utils_plot1(ax1, False, "Skupaj", None, dfp.index)

sns.set_palette(blues_dd)
sns.lineplot(data=dfp[starosti5_novi], ax=ax2)
sns.set_palette(cmapr[5:])
sns.lineplot(data=dfp[starosti10_novi[6:]], ax=ax2)
sns.set_palette(cmapr)
sns.lineplot(data=dfp[starosti10_novi[:6]], ax=ax2)
ax2.legend("")
utils_plot1(ax2, False, "Novi", None, dfp.index)'''

cmapr = cmap[::-1]
cmapr2 = np.stack((cmapr[1,:], cmapr[3,:], cmapr[4,:], cmapr[6,:], cmapr[-1,:]))

sns.lineplot(data=dfp[starosti5], ax=ax1)
sns.set_palette(cmapr2)
sns.lineplot(data=dfp[starosti10_skupine[:5]], ax=ax1, marker="")
ax1.legend(starosti5 + starosti10_skupine)
leg = ax1.get_legend()
for i, h in enumerate(leg.legendHandles[5:]):
    h.set_color(cmapr2[i])
utils_plot1(ax1, False, "Skupaj", None, dfp.index)

sns.set_palette(blues_dd)
sns.lineplot(data=dfp[starosti5_novi], ax=ax2)
sns.set_palette(cmapr2)
sns.lineplot(data=dfp[starosti10_skupine_novi], ax=ax2, marker="")
ax2.legend("")
utils_plot1(ax2, False, "Novi", None, dfp.index)



dfp[starosti_rel[:5]].plot(ax=ax3, kind="bar", stacked=True, width=1,␣
 ↪color=blues_d)
```

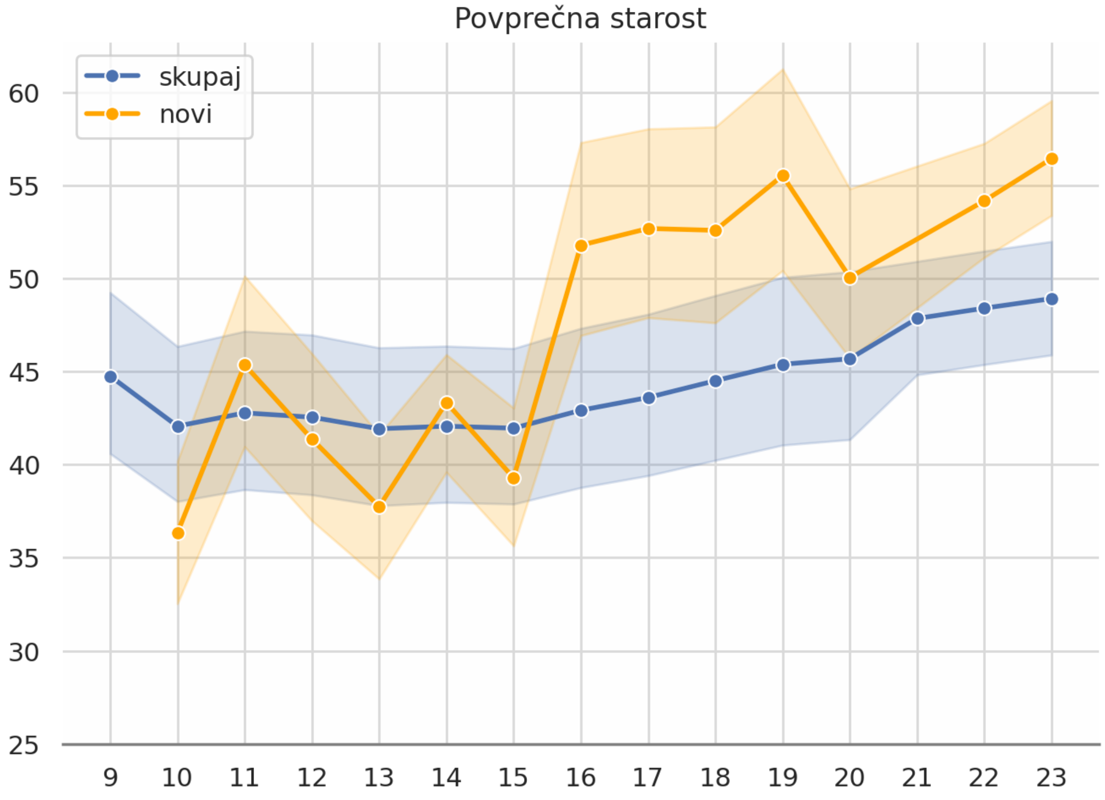
```
dfp[starosti_rel[5:]].plot(ax=ax3, kind="bar", stacked=True, width=1,␣
 ↪color=cmapr)
utils_plot1(ax3, False, "Skupaj (relativno)", (0, 100), None)
ax3.legend("")

dfp[starosti_novi_rel[:5]].plot(ax=ax4, kind="bar", stacked=True, width=1,␣
 ↪color=blues_d)
dfp[starosti_novi_rel[5:]].plot(ax=ax4, kind="bar", stacked=True, width=1,␣
 ↪color=cmapr)
ax4.legend("", loc="center left")
sns.set_palette(cmapr)
reverse_legend(ax4, starosti)
utils_plot1(ax4, False, "Novi (relativno)", (0, 100), None)

sns.set_palette(current_palette)
plt.subplots_adjust(top=0.95, hspace=.23)
```



[105]:

```
fig, (ax7, ax8) = plt.subplots(nrows=1, ncols=2, figsize=(15, 5), sharex=True)
sns.lineplot(data=[avg_starost,avg_starost1, avg_starost2], err_style="band",␣
↪ax=ax7, color="b")
```

```
sns.lineplot(data=[novi_avg_starost,novi_avg_starost1, novi_avg_starost2],␣
 ↪err_style="band", ax=ax7, color="orange")
#sns.lineplot(ax=ax7, x=cat((dni10, dni10, dni10)), y=cat([avg_starost10.
 ↪to_numpy(),avg_starost110.to_numpy(),avg_starost210.to_numpy()]), color="b",␣
 ↪err_style="bars")
ax7.legend(["skupaj", "novi"], loc="upper left")
utils_plot1(ax7, False, "Povprečna starost", 25, avg_starost.index[12:])
ax8.remove()
```


Povprečna starost
skupaj
novi