Pandas COVID19 Practice - Solutions - Unibs 2021

In this exercize, we will anlyze some public data provided by the "Dipartimento della Protezione Civile". To inform the citzens and give the reached datas, useful only to comunicate and informate, Dipartimento della Protezione Civile has elaborated a dashboard reachable to the URL http://arcg.is/C1unv (desktop version) and http://arcg.is/081a51 (mobile version) and give to everyone, under the licence CC-BY-4.0, the following infos updated dialy at 18.30:

- National evolution data
- json data
- Regional data
- Provincial data
- Summaries
- Areas
- Notes
- Contracts data DPC furnitures
- Metrics

See https://github.com/pcm-dpc/COVID-19

```
import matplotlib.pyplot as plt
```

Import pandas package under name `pd` and print version

```
hint: import ... as, pd.__version__
```

```
import pandas as pd
```

```
print(pd.__version__)
```

```
    1.1.5
```

Base

Download latest COVID19 csv of italian regions from official Italian Civil Protection github and show it

CSV data at: https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-regioni/dpc-covid19-ita-regioni-latest.csv

```
hint: pd.read_csv, sep=","
```

```
url = "https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-regioni/dpc-covid19-ita-regioni-latest.csv"
covid_regions_latest = pd.read_csv(url, sep=",")
```

```
covid_regions_latest
```

	data	stato	codice_regione	denominazione_regione	lat	long	ricoverati_con_sintomi	terapia_inte
0	2021-03-31T17:00:00	ITA	13	Abruzzo	42.351222	13.398438	607	
1	2021-03-31T17:00:00	ITA	17	Basilicata	40.639471	15.805148	170	
2	2021-03-31T17:00:00	ITA	18	Calabria	38.905976	16.594402	389	
3	2021-03-31T17:00:00	ITA	15	Campania	40.839566	14.250850	1587	
4	2021-03-31T17:00:00	ITA	8	Emilia-Romagna	44.494367	11.341721	3427	
5	2021-03-31T17:00:00	ITA	6	Friuli Venezia Giulia	45.649435	13.768136	664	
6	2021-03-31T17:00:00	ITA	12	Lazio	41.892770	12.483667	3044	
7	2021-03-31T17:00:00	ITA	7	Liguria	44.411493	8.932699	642	
8	2021-03-31T17:00:00	ITA	3	Lombardia	45.466794	9.190347	7033	
9	2021-03-31T17:00:00	ITA	11	Marche	43.616760	13.518875	803	
10	2021-03-31T17:00:00	ITA	14	Molise	41.557748	14.659161	63	

11	2021-03-31T17:00:00	ITA	21	P.A. Bolzano	46.499335	11.356624	90
12	2021-03-31T17:00:00	ITA	22	P.A. Trento	46.068935	11.121231	201
13	2021-03-31T17:00:00	ITA	1	Piemonte	45.073274	7.680687	3873
14	2021-03-31T17:00:00	ITA	16	Puglia	41.125596	16.867367	1840
15	2021-03-31T17:00:00	ITA	20	Sardegna	39.215312	9.110616	222
16	2021-03-31T17:00:00	ITA	19	Sicilia	38.115697	13.362357	891

Sort columns and show their types

```
hint: df.sort_index, axis=1, inplace=True, df.dtypes
```

```
covid_regions_latest.sort_index(axis=1, inplace=True)
```

```
covid_regions_latest.dtypes
```

```
    casi_da_screening                           float64
    casi_da_sospetto_diagnostico                float64
    casi_testati                                  int64
    codice_nuts_1                                object
    codice_nuts_2                                object
    codice_regione                                int64
    data                                         object
    deceduti                                      int64
    denominazione_regione                        object
    dimessi_guariti                               int64
```

```
ingressi_terapia_intensiva                    int64
isolamento_domiciliare                        int64
lat                                         float64
long                                        float64
note                                         object
note_casi                                    object
note_test                                   float64
nuovi_positivi                                int64
ricoverati_con_sintomi                        int64
stato                                        object
tamponi                                       int64
tamponi_test_antigenico_rapido                int64
tamponi_test_molecolare                       int64
terapia_intensiva                             int64
totale_casi                                   int64
totale_ospedalizzati                          int64
totale_positivi                               int64
totale_positivi_test_antigenico_rapido        int64
totale_positivi_test_molecolare               int64
variazione_totale_positivi                    int64
dtype: object
```

Print first and last five rows of the data

```
hint: df.head, df.tail
```

```
covid_regions_latest.head()
```

	casi_da_screening	casi_da_sospetto_diagnostico	casi_testati	codice_nuts_1	codice_nuts_2	codice_regione	
0	NaN	NaN	576469	ITF	ITF1	13	202 31T17
1	NaN	NaN	168149	ITF	ITF5	17	202 31T17
2	NaN	NaN	627407	ITF	ITF6	18	202 31T17

```
covid_regions_latest.tail()
```

casi_da_screening	casi_da_sospetto_diagnostico	casi_testati	codice_nuts_1	codice_nuts_2	codice_regione

Print synthetic statistical description of the dataframe (count, min, max, mean, etc.)

```
hint: df.describe
```

3111

```
covid_regions_latest.describe()
```

	casi_da_screening	casi_da_sospetto_diagnostico	casi_testati	codice_regione	deceduti	dimessi_guariti	in
count	0.0	0.0	2.100000e+01	21.000000	21.000000	21.000000	
mean	NaN	NaN	1.093538e+06	11.857143	5206.952381	138716.428571	
std	NaN	NaN	1.015114e+06	6.428730	6808.769063	144930.365204	
min	NaN	NaN	5.151400e+04	1.000000	425.000000	7971.000000	
25%	NaN	NaN	3.608490e+05	7.000000	1234.000000	37086.000000	
50%	NaN	NaN	6.100430e+05	12.000000	3307.000000	78350.000000	
75%	NaN	NaN	1.558375e+06	17.000000	5363.000000	227752.000000	
max	NaN	NaN	3.511485e+06	22.000000	30735.000000	608894.000000	

Count elements for each column

```
hint: df.count
```

```
covid_regions_latest.count()
```

```
casi_da_screening                          0
casi_da_sospetto_diagnostico               0
casi_testati                              21
codice_nuts_1                             21
codice_nuts_2                             21
codice_regione                            21
data                                      21
deceduti                                  21
denominazione_regione                     21
dimessi_guariti                           21
ingressi_terapia_intensiva                21
isolamento_domiciliare                    21
lat                                       21
long                                      21
note                                       7
note_casi                                  2
note_test                                  0
nuovi_positivi                            21
ricoverati_con_sintomi                    21
stato                                     21
tamponi                                   21
tamponi_test_antigenico_rapido            21
tamponi_test_molecolare                   21
terapia_intensiva                         21
totale_casi                               21
totale_ospedalizzati                      21
totale_positivi                           21
totale_positivi_test_antigenico_rapido    21
totale_positivi_test_molecolare           21
variazione_totale_positivi                21
dtype: int64
```

Select only "totale_positivi" and "nuovi_positivi" columns

```
hint: df[]
```

```
covid_regions_latest[["totale_positivi", "nuovi_positivi"]]
```

	totale_positivi	nuovi_positivi
0	10132	314
1	4774	149
2	10325	347
3	93117	2016
4	72435	1490
5	15197	644
6	51051	1800
7	7095	383
8	95855	3943
9	9367	807
10	866	17
11	686	120
12	2863	187
13	35059	2298
14	46857	1962
15	14397	444
16	19920	2904
17	28107	1538
18	4806	162
19	902	62
20	38697	2317

Create the new column "precedenti_positivi" columns using the formula $precedenti\_positivi = totale\_positivi - nuovi\_positivi$ and show it

```
hint: df[] = df[] - df[]
```

```
covid_regions_latest["precedenti_positivi"] = (
    covid_regions_latest.totale_positivi - covid_regions_latest.nuovi_positivi
)
covid_regions_latest.precedenti_positivi
```

```
0      9818
1      4625
2      9978
3     91101
4     70945
5     14553
6     49251
7      6712
8     91912
9      8560
10      849
11      566
12     2676
13    32761
14    44895
15    13953
16    17016
17    26569
18     4644
19      840
20    36380
Name: precedenti_positivi, dtype: int64
```

Select only rows from 5 to 7

```
hint: df.loc[]
```

```
covid_regions_latest.loc[5:7]
```

	casi_da_screening	casi_da_sospetto_diagnostico	casi_testati	codice_nuts_1	codice_nuts_2	codice_regione	
5	NaN	NaN	580139	ITH	ITH4	6	202 31T17
6	NaN	NaN	3421823	ITI	ITI4	12	202 31T17
7	NaN	NaN	517132	ITC	ITC3	7	202 31T17

Select only "totale_positivi" and "nuovi_positivi" columns and only rows from 5 to 7

```
hint: df[], df.loc[]
```

```
covid_regions_latest[["totale_positivi", "nuovi_positivi"]].loc[5:7]
```

	totale_positivi	nuovi_positivi
5	15197	644
6	51051	1800
7	7095	383

Set "denominazione_regione" as index and show it

```
hint: df.set_index, inplace=True
```

```
covid_regions_latest.set_index("denominazione_regione", inplace=True)
covid_regions_latest
```

g	casi_da_sospetto_diagnostico	casi_testati	codice_nuts_1	codice_nuts_2	codice_regione	data	deceduti	dimes
N	NaN	576469	ITF	ITF1	13	2021-03-31T17:00:00	2136	
N	NaN	168149	ITF	ITF5	17	2021-03-31T17:00:00	443	
N	NaN	627407	ITF	ITF6	18	2021-03-31T17:00:00	819	
N	NaN	2438913	ITF	ITF3	15	2021-03-31T17:00:00	5363	
N	NaN	1679293	ITH	ITH5	8	2021-03-31T17:00:00	11917	
N	NaN	580139	ITH	ITH4	6	2021-03-31T17:00:00	3307	
N	NaN	3421823	ITI	ITI4	12	2021-03-31T17:00:00	6644	
N	NaN	517132	ITC	ITC3	7	2021-03-31T17:00:00	3879	
N	NaN	3511485	ITC	ITC4	3	2021-03-31T17:00:00	30735	
N	NaN	610043	ITI	ITI3	11	2021-03-31T17:00:00	2621	
N	NaN	166401	ITF	ITF2	14	2021-03-	438	

31T17:00:00

N	NaN	360849	ITH	ITH1	21	2021-03- 31T17:00:00	1126

Plot horizontal bars of "tamponi_test_antigenico_rapido" and "tampone_test_molecolare" columns for each region

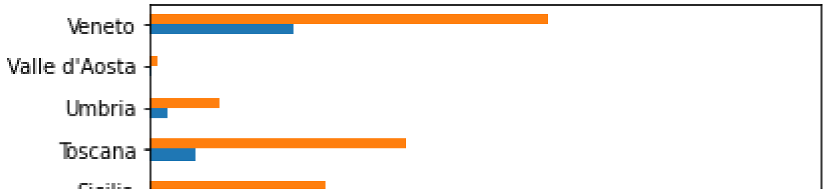
31T17:00:00

```
hint: df[].plot.barh, figsize=()
```

- Figure size of (6,8)

```
covid_regions_latest[
    ["tamponi_test_antigenico_rapido", "tamponi_test_molecolare"]
].plot.barh(figsize=(6, 8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337f3bb6d0>
```

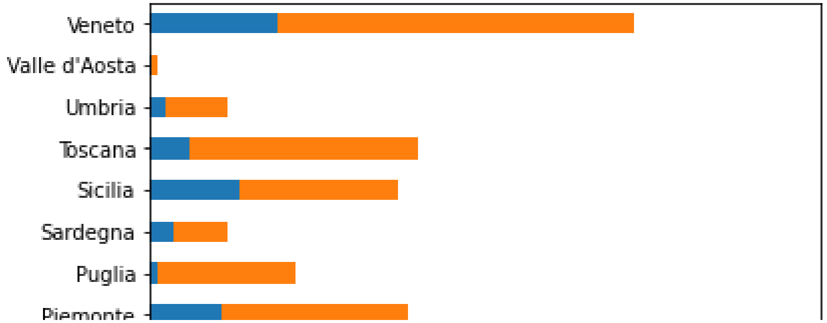


Plot horizontal bars of "tamponi_test_antigenico_rapido" and "tampone_test_molecolare" stacked together

```
hint: stacked=True
```

- Figure size of (6,8)

```
ax = covid_regions_latest[
    ["tamponi_test_antigenico_rapido", "tamponi_test_molecolare"]
].plot.barh(figsize=(6, 8), stacked=True)
```



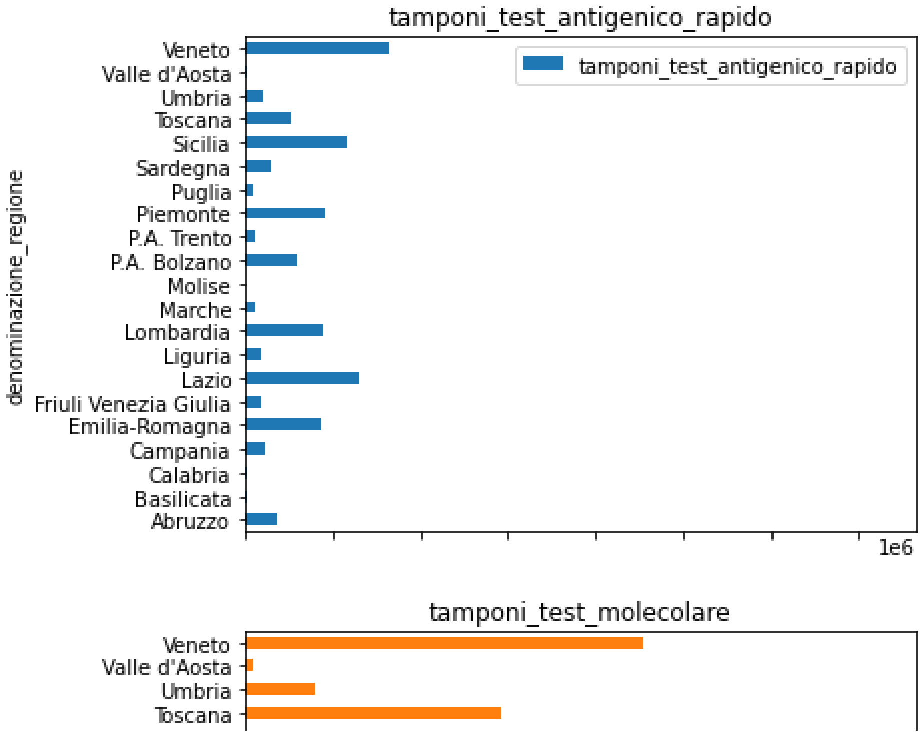
Plot horizontal bars of "tamponi_test_antigenico_rapido" and "tampone_test_molecolare" columns for each region in different subplots

hint: df[].plot.barh, figsize=()

- Figure size of (6,10)

```
covid_regions_latest[
    ["tamponi_test_antigenico_rapido", "tamponi_test_molecolare"]
].plot.barh(figsize=(6, 10), subplots=True)
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f337ed1de90>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f337ec06ad0>],
      dtype=object)
```



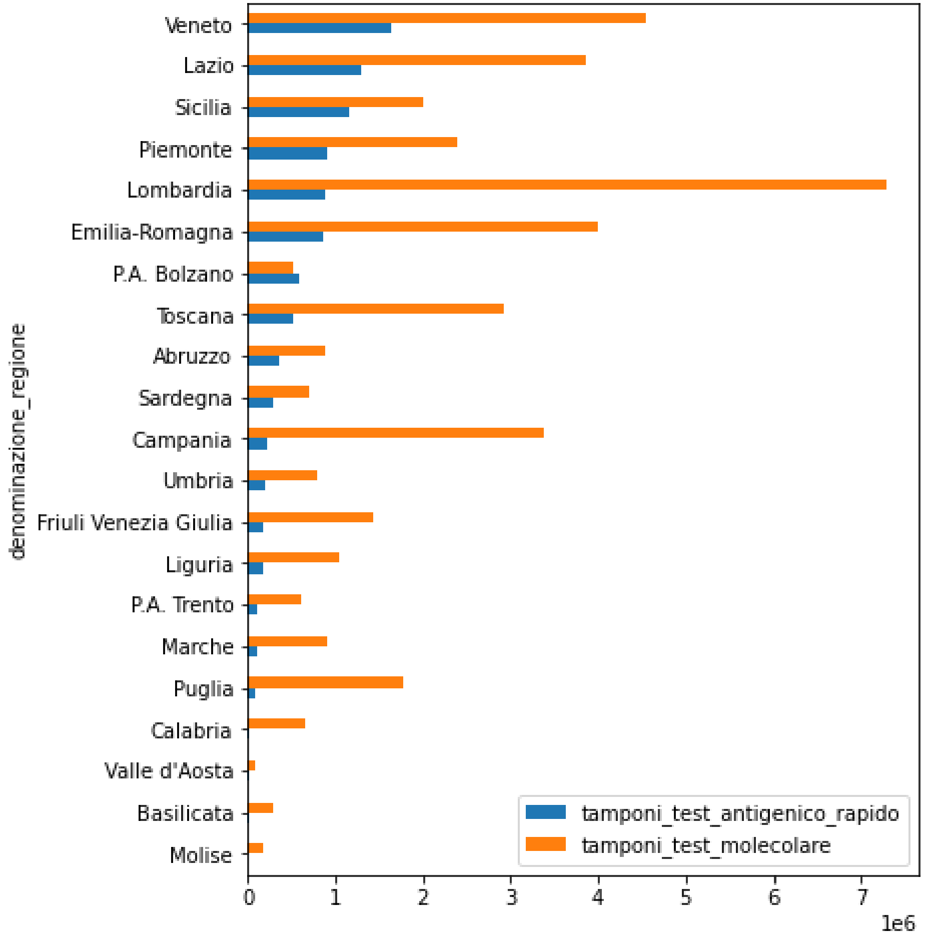
Plot horizontal bars of "tamponi_test_antigenico_rapido" and "tampone_test_molecolare" columns for each region sorting by "tampone_test_antigenico_rapido" column

```
hint: df.sort_values
```

- Figure size of (6,8)

```
covid_regions_latest[
    ["tamponi_test_antigenico_rapido", "tamponi_test_molecolare"]
].sort_values("tamponi_test_antigenico_rapido").plot.barh(figsize=(6, 8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337eb9ced0>
```



Intermediate

Plot histograms of "totale_ospedalizzati", "terapia_intensiva", "ricoverati_con_sintomi", "nuovi_positivi" in different subplots

```
hint: df[].plot.hist, bins=, alpha=
```

- Figure size of (10, 6)
- 10 Bins
- Alpha of 0.75

```
covid_regions_latest[
    [
        "totale_ospedalizzati",
        "terapia_intensiva",
        "ricoverati_con_sintomi",
        "nuovi_positivi",
    ]
].plot.hist(figsize=(10, 6), bins=10, alpha=0.5, subplots=True)
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f337e992d90>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f337e94fed0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f337e99df90>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f337e8c85d0>]
```

Plot in pie charts the number of "tamponi_test_molecolare" with percentage for each region with exploded slice for *Lombardia* region

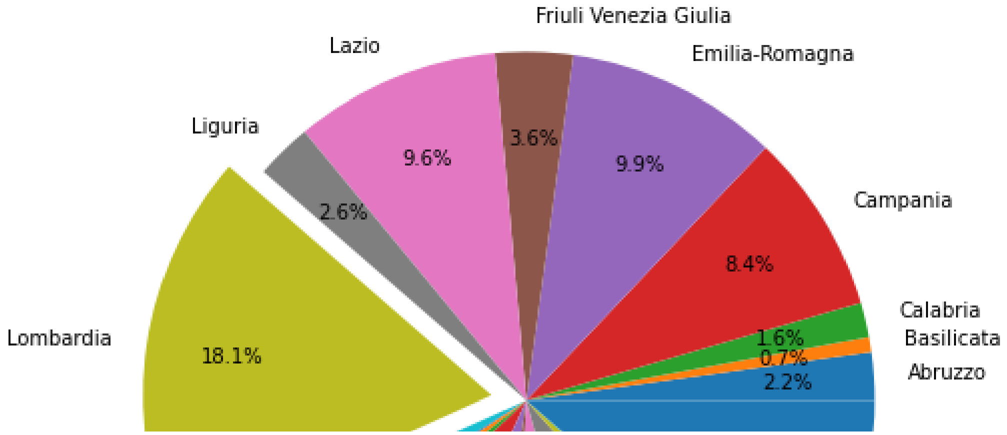
```
hint: df.index, df[].plot.pie, figsize=, autopct="%1.1f%%, pctdistance=, explode=, ylabel=",
```

- Figure size of (8,8)
- Explode of 0.1 for Lombardia region
- Distance of the percentage of 0.75

```
explode = [
    0.1 if region == "Lombardia" else 0.0 for region in covid_regions_latest.index
]

covid_regions_latest.tamponi_test_molecolare.plot.pie(
    figsize=(8, 8), autopct="%1.1f%%", pctdistance=0.75, explode=explode, ylabel=""
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337e748e50>
```



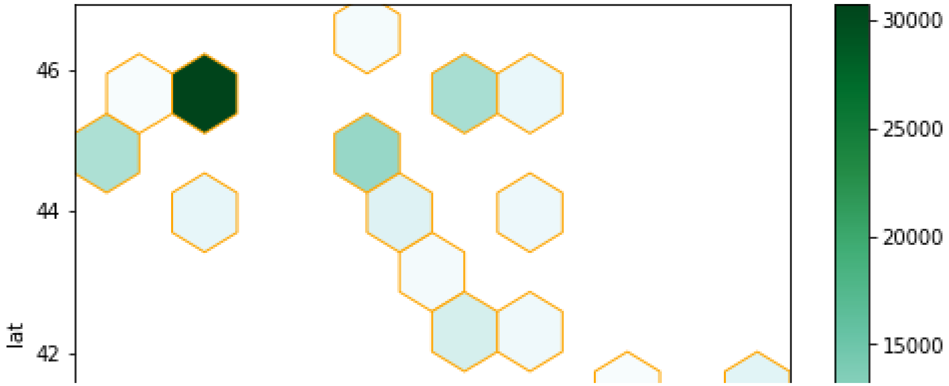
Plot hexbin plot of "deceduti" by ("long", "lat) coordinates of regions

```
hint: df.plot.hexbin, x=, y=, C=, edgecolor=, gridsize=
```

- Figure size of (8, 6)
- Gridsize of 10

```
covid_regions_latest.plot.hexbin(
    figsize=(8, 6), x="long", y="lat", C="deceduti", edgecolor="orange", gridsize=10
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337e9e3b10>
```



Plot scatterplots of "terapia_intensiva" and "ingressi_terapia_intensiva" by ("long", "lat") coordinates of regions both in the same plot

```
hint: df.plot.scatter, x=, y=, color=, alpha=, s=, label=, ax=, ax.legend
```

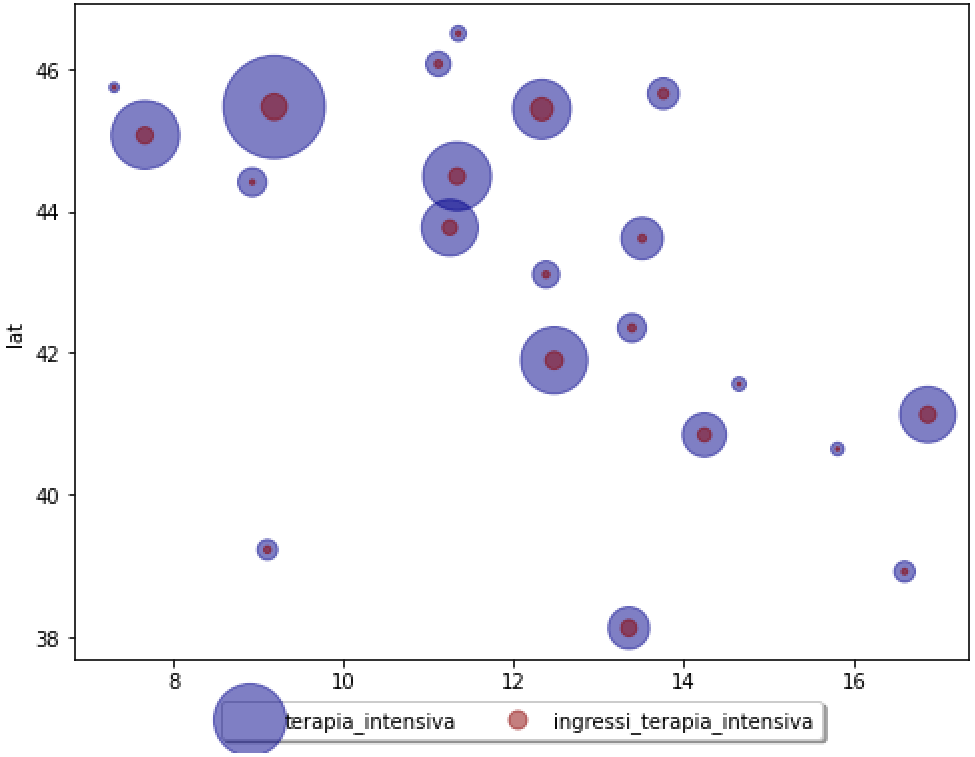
- Figure size of (8, 6)
- Alpha of 0.5
- Legend at the bottom

```
ZOOM = 3
ax = covid_regions_latest.plot.scatter(
    x="long",
    y="lat",
    color="DarkBlue",
    alpha=0.5,
    s=covid_regions_latest.terapia_intensiva * ZOOM,
    label="terapia_intensiva",
)

covid_regions_latest.plot.scatter(
    figsize=(8, 6),
    x="long",
    y="lat",
    c="DarkRed",
    alpha=0.5,
    s=covid_regions_latest.ingressi_terapia_intensiva * ZOOM,
    label="ingressi_terapia_intensiva",
    ax=ax,
)

ax.legend(
    loc="upper center", bbox_to_anchor=(0.5, -0.05), fancybox=True, shadow=True, ncol=5
)
```

```
<matplotlib.legend.Legend at 0x7f337eae0d90>
```



Group the regions by color and plot bars of the mean value of "nuovi_positivi" column

```
hint: pd.Series, df.groupby().mean().plot.bar(), title=, rot=,
```

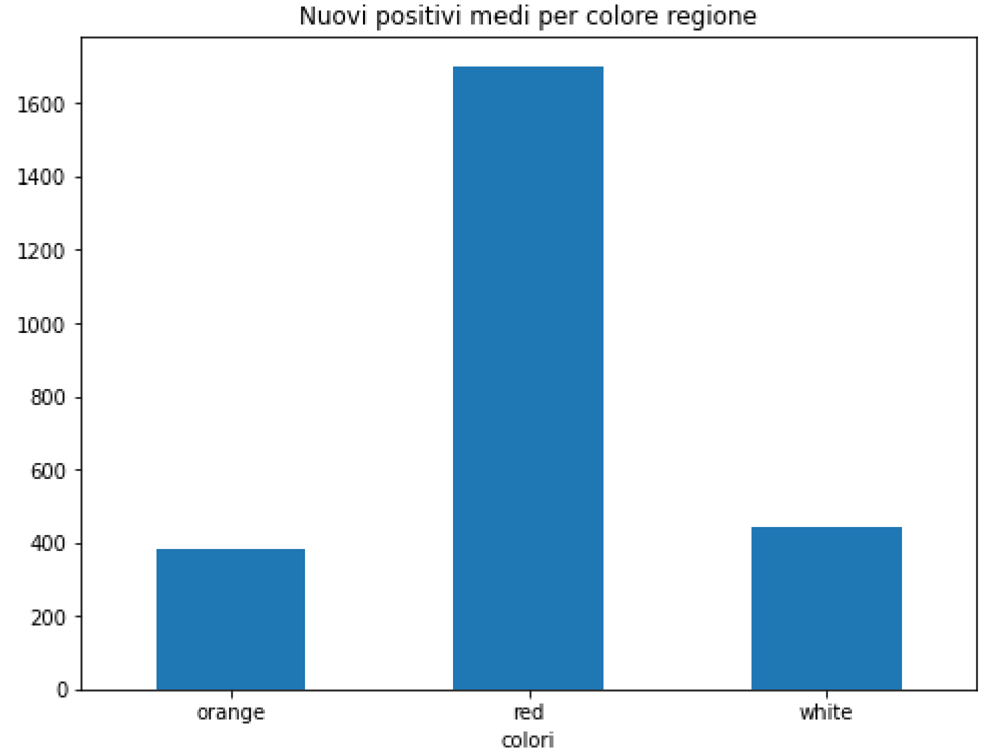
- Figure size (8, 6)

regions_colors = { "Abruzzo": "orange", "Basilicata": "orange", "Calabria": "orange", "Campania": "red", "Emilia-Romagna": "red", "Friuli Venezia Giulia": "red", "Lazio": "red", "Liguria": "orange", "Lombardia": "red", "Marche": "red", "Molise": "red", "P.A. Bolzano": "orange", "P.A. Trento": "red", "Piemonte": "red", "Puglia": "red", "Sardegna": "white", "Sicilia": "red", "Toscana": "orange", "Umbria": "orange", "Valle d'Aosta": "orange", "Veneto": "red", }

```
regions_colors = {
    "Abruzzo": "orange",
    "Basilicata": "orange",
    "Calabria": "orange",
    "Campania": "red",
    "Emilia-Romagna": "red",
    "Friuli Venezia Giulia": "red",
    "Lazio": "red",
    "Liguria": "orange",
    "Lombardia": "red",
    "Marche": "red",
    "Molise": "red",
    "P.A. Bolzano": "orange",
    "P.A. Trento": "red",
    "Piemonte": "red",
    "Puglia": "red",
    "Sardegna": "white",
    "Sicilia": "red",
    "Toscana": "orange",
    "Umbria": "orange",
    "Valle d'Aosta": "orange",
    "Veneto": "red",
}
covid_regions_latest["colori"] = pd.Series(regions_colors)
```

```
covid_regions_latest.groupby("colori")["nuovi_positivi"].mean().plot.bar(
    figsize=(8, 6), title="Nuovi positivi medi per colore regione", rot=0
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337d533ed0>
```



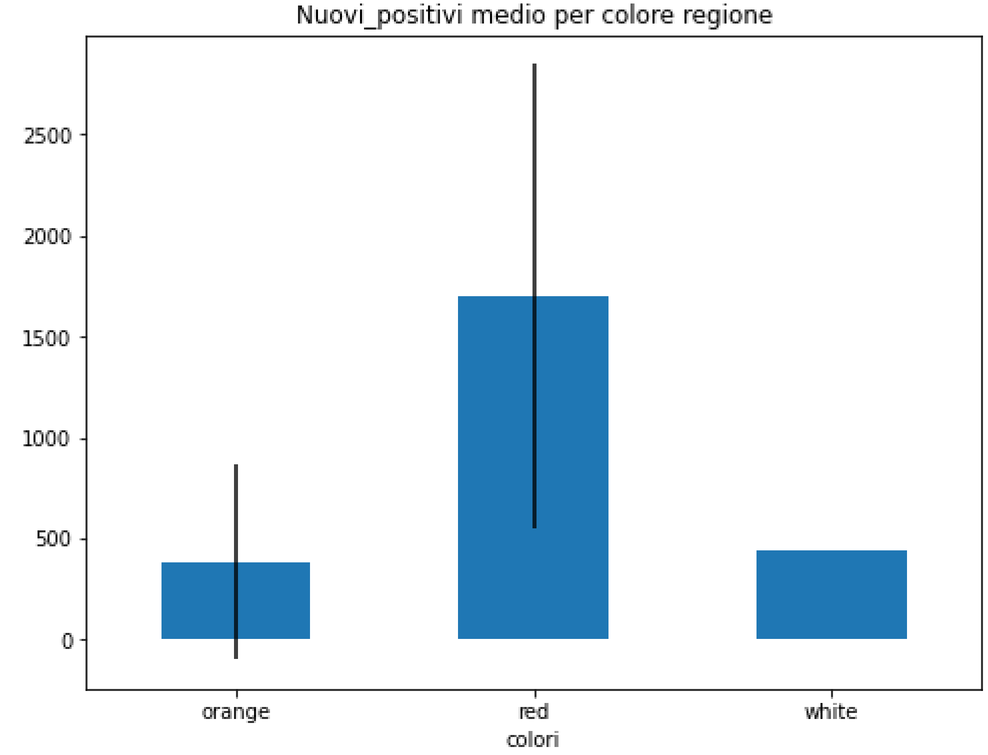
Group the regions by color and plot bars of mean *and error* of "nuovi_positivi" column

- Figure size (8, 6)

```
hint: df.groupby().std(), yerr=, rot=,
```

```
COLUMN = "nuovi_positivi"
COLUMN = "nuovi_positivi"
yerr = covid_regions_latest.groupby("colori")[COLUMN].std()
covid_regions_latest.groupby("colori")[COLUMN].mean().plot.bar(
    figsize=(8, 6),
    title=f"{COLUMN.capitalize()} medio per colore regione",
    yerr=yerr,
    rot=0,
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f337d50dfd0>
```

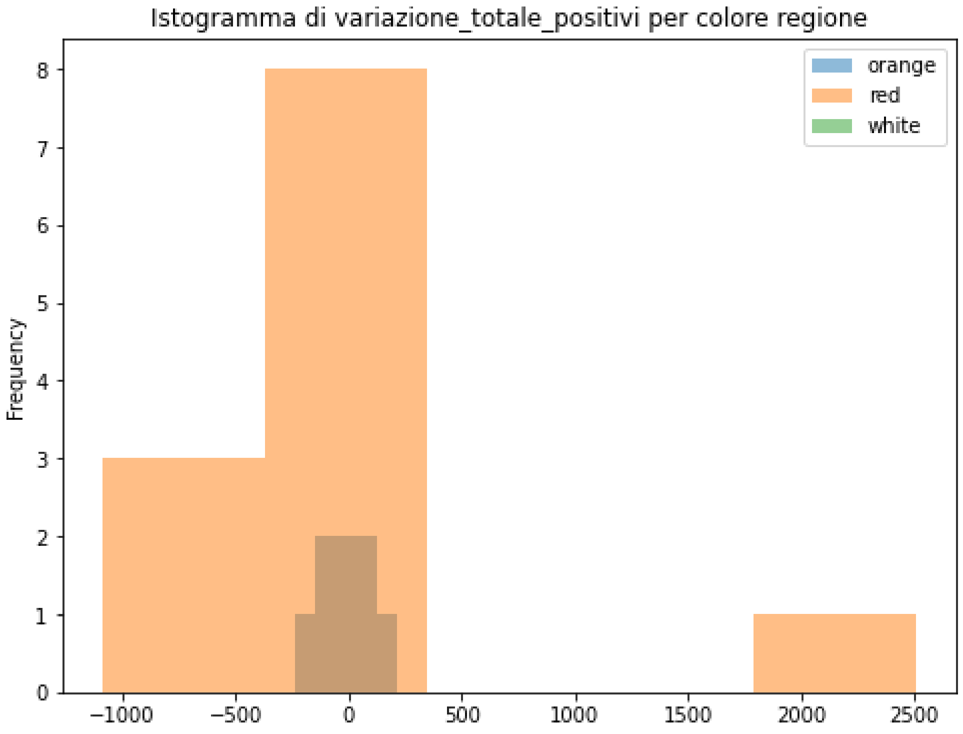


Group the regions by color and plot a hist of "variazione_totale_positivi" using 5 bins

- Figure size (8, 6)

```
hint: pd.Series, df.groupby().plot.hist(), legend=,
```

```
covid_regions_latest.groupby("colori")["variazione_totale_positivi"].plot.hist(
    figsize=(8, 6),
    title="Istogramma di variazione_totale_positivi per colore regione",
    alpha=0.5,
    rot=0,
    bins=5,
    legend=True,
)
plt.show()
```



```
## Thanks to Matteo Olivato m.olivato@unibs.it
```