In [1]:

```
# Import Python libraries for visualisation and data analysis
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

# sns.set_theme()  # Apply the default Seaborn theme
%matplotlib inline

# Suppress warnings to avoid potential confusion
import warnings

# Libraries for statistical and scientific computing
import statsmodels.api as sm
from scipy import stats

warnings.filterwarnings("ignore")
import ipywidgets as widgets
from IPython.display import display
```

In [2]:

```
d1=pd.read_csv('2020.csv')
# d1.set_index('date',inplace=True)
d2=pd.read_csv('2021.csv')
# d2.set_index('date',inplace=True)
d3=pd.read_csv('2022.csv')

# d3.set_index('date',inplace=True)
```

In [86]:

```
def subperiod_mobility_trends(data, start_date, end_date):
    """
    Add your mobility data in `data`.

    This function selects a subperiod of the mobility data based on prespecified
start data and end date.
    """
    subdata= data[
        data["date"].isin(pd.date_range(start=start_date, end=end_date))
    ]
    return subdata


def rename_mobility_trends(data):
    """
    This function renames the column headings of the six mobility categories.
    """
    data = data.rename(
        columns={
            "retail_and_recreation_percent_change_from_baseline": "Retail_Recrea
tion",
            "grocery_and_pharmacy_percent_change_from_baseline": "Grocery_Pharma
cy",
            "parks_percent_change_from_baseline": "Parks",
            "transit_stations_percent_change_from_baseline": "Transit_stations",
            "workplaces_percent_change_from_baseline": "Workplaces",
            "residential_percent_change_from_baseline": "Residential",
        }
    )
    return data
```

In [87]:

```
d1=rename_mobility_trends(d1)
d2=rename_mobility_trends(d2)
d3=rename_mobility_trends(d3)
```

In [88]:

```
d1['year']='2020'
d2['year']='2021'
d3['year']='2022'
```

In [89]:

```
data=d1.append(d2)
data=data.append(d3)
```

In [90]:

```
ALL = 'ALL'
def unique_sorted_values_plus_ALL(array):
    unique = array.unique().tolist()
    unique.sort()
    unique.insert(0, ALL)
    return unique
```

In [91]:

```
london_data=data.loc[data['sub_region_1'] == 'Greater London']
greater=london_data[london_data['sub_region_2'].isnull()]
london_regions=london_data.sub_region_2.unique()
london_regions=london_regions[1:]
variabl=["Retail_Recreation","Grocery_Pharmacy","Parks","Transit_stations","Work
places","Residential"]
```

In [92]:

```
data_long = pd.melt(
    london_data,
    id_vars=["country_region", "sub_region_1","sub_region_2", "date"],
    # The columns 'date' and 'sub_region_1' are not needed for the box
    # plots below but we will need the two variables in subsequent tasks.
    value_vars=data.columns[9:15],
).dropna()
```

In [93]:

```
def ld(data_long):
    first_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2020-03-24")
        & (data_long["date"] <= "2020-04-13")
    ]

    second_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2020-11-05")
        & (data_long["date"] <= "2020-11-25")
    ]

    third_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2021-01-06")
        & (data_long["date"] <= "2021-01-26")
    ]
    return [first_lockdown_UK,second_lockdown_UK,third_lockdown_UK]
```

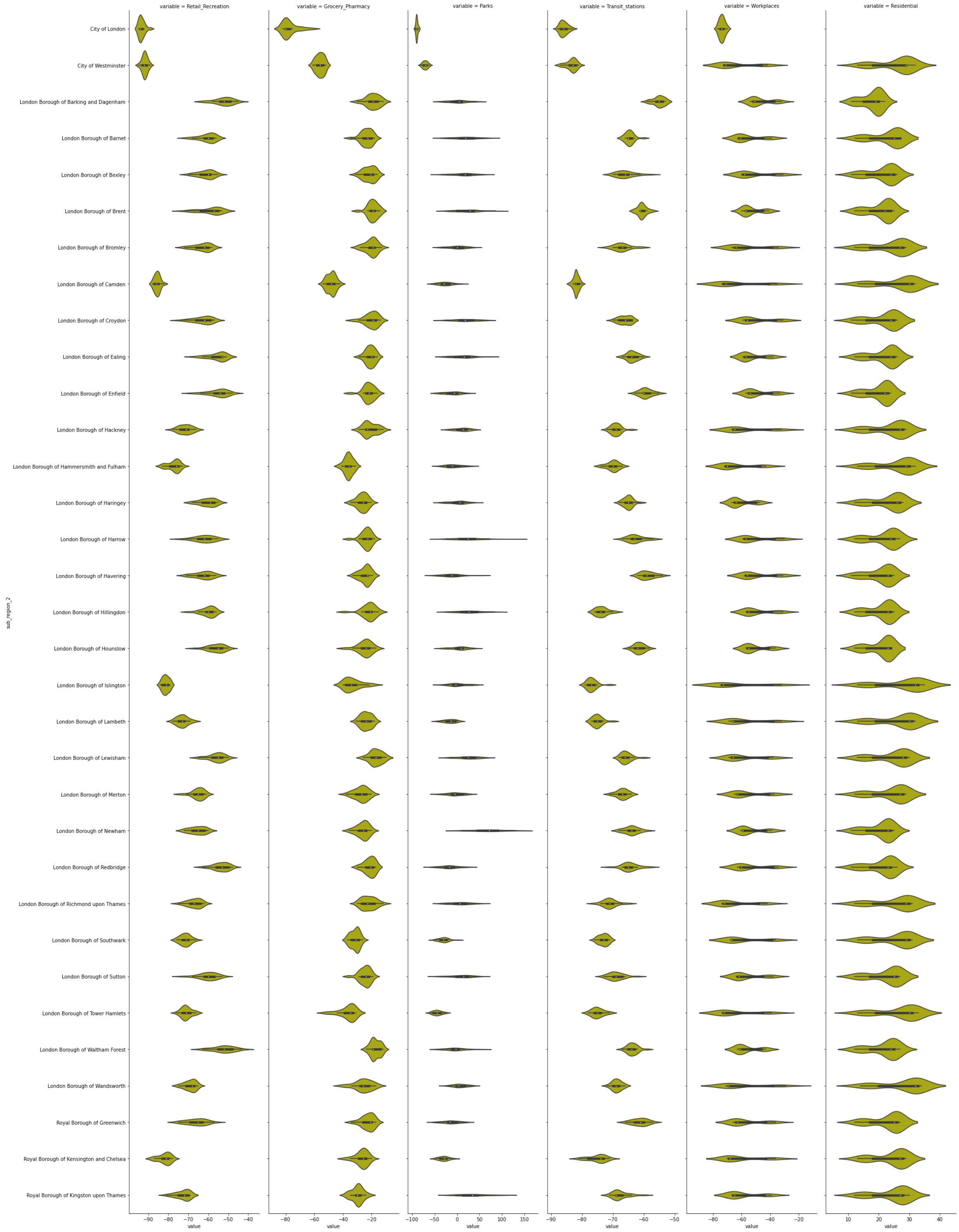
In [94]:

```
first_lockdown_UK,second_lockdown_UK,third_lockdown_UK=ld(data_long)
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

third lockdown

In [38]:

```
sns.catplot(
    x="value",
    y="sub_region_2",
    col="variable",
    kind="violin",
    sharex=False,
    height=35,
    aspect=0.13,
    color="y",
    data=third_lockdown_UK,
);
```



In [12]:

```
third_lockdown_UK_mean = (
    third_lockdown_UK.groupby(["variable", "sub_region_2"])["value"]
    .mean()
    .reset_index()
)
```

In [14]:

```
third_lockdown_UK_mean_sorted = third_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_2", "variable", "value"]]
third_lockdown_UK_mean_sorted
```

Out[14]:

	sub_region_2	variable	value
166	London Borough of Barking and Dagenham	Workplaces	-46.904762
174	London Borough of Enfield	Workplaces	-48.952381
180	London Borough of Hillingdon	Workplaces	-49.238095
179	London Borough of Havering	Workplaces	-49.523810
172	London Borough of Croydon	Workplaces	-49.904762
...	...	...	...
12	London Borough of Hammersmith and Fulham	Grocery_Pharmacy	-36.238095
27	London Borough of Tower Hamlets	Grocery_Pharmacy	-36.809524
7	London Borough of Camden	Grocery_Pharmacy	-48.285714
1	City of Westminster	Grocery_Pharmacy	-56.095238
0	City of London	Grocery_Pharmacy	-77.190476

197 rows × 3 columns

In [17]:

```
variabl=["Retail_Recreation","Grocery_Pharmacy","Parks","Transit_stations","Work
places","Residential"]
```

In [18]:

```
for i in variabl:
    t_min=third_lockdown_UK_mean_sorted[third_lockdown_UK_mean_sorted['variable'
]==i].iloc[0]
    t_max=third_lockdown_UK_mean_sorted[third_lockdown_UK_mean_sorted['variable'
]==i].iloc[-1]

    print(f'min change is for {i} is {t_min.sub_region_2} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_2} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is London Borough of Barking and
Dagenham with value of -52.333333333333336
max change is for Retail_Recreation is City of London with value of
-93.38095238095238


min change is for Grocery_Pharmacy is London Borough of Lewisham wit
h value of -17.142857142857142
max change is for Grocery_Pharmacy is City of London with value of -
77.19047619047619


min change is for Parks is London Borough of Newham with value of 6
9.42857142857143
max change is for Parks is City of London with value of -89.76190476
190476


min change is for Transit_stations is London Borough of Barking and
Dagenham with value of -55.19047619047619
max change is for Transit_stations is City of London with value of -
85.9047619047619


min change is for Workplaces is London Borough of Barking and Dagenh
am with value of -46.904761904761905
max change is for Workplaces is City of London with value of -73.8


min change is for Residential is London Borough of Wandsworth with v
alue of 27.857142857142858
max change is for Residential is London Borough of Barking and Dagen
ham with value of 17.904761904761905
```

In [19]:

```
third_lockdown_UK_descriptive_stats = (
    third_lockdown_UK.groupby(["sub_region_2", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
third_lockdown_UK_descriptive_stats
```

Out[19]:

	sub_region_2	variable	min	max	mean	median	std
0	City of London	Grocery_Pharmacy	-82.0	-62.0	-77.190476	-80.0	5.306779
1	City of London	Parks	-94.0	-85.0	-89.761905	-90.0	2.211442
2	City of London	Retail_Recreation	-95.0	-89.0	-93.380952	-94.0	1.532194
3	City of London	Transit_stations	-88.0	-83.0	-85.904762	-86.0	1.261141
4	City of London	Workplaces	-77.0	-70.0	-73.800000	-74.0	1.934647
...	...	...	...	...	...	...	...
192	Royal Borough of Kingston upon Thames	Parks	-5.0	95.0	34.916667	34.5	30.696338
193	Royal Borough of Kingston upon Thames	Residential	12.0	30.0	24.238095	27.0	6.032452
194	Royal Borough of Kingston upon Thames	Retail_Recreation	-81.0	-69.0	-72.857143	-72.0	3.539572
195	Royal Borough of Kingston upon Thames	Transit_stations	-71.0	-60.0	-67.619048	-69.0	2.635834
196	Royal Borough of Kingston upon Thames	Workplaces	-67.0	-39.0	-58.619048	-65.0	11.182469

197 rows × 7 columns

Second Lockdown

In [24]:

```
second_lockdown_UK_mean = (
    second_lockdown_UK.groupby(["variable", "sub_region_2"])["value"]
    .mean()
    .reset_index()
)

second_lockdown_UK_mean_sorted = second_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_2", "variable", "value"]]
```

Out[24]:

	sub_region_2	variable	value
166	London Borough of Barking and Dagenham	Workplaces	-33.857143
174	London Borough of Enfield	Workplaces	-37.285714
180	London Borough of Hillingdon	Workplaces	-37.523810
178	London Borough of Harrow	Workplaces	-37.714286
172	London Borough of Croydon	Workplaces	-37.904762
...	...	...	...
27	London Borough of Tower Hamlets	Grocery_Pharmacy	-23.761905
12	London Borough of Hammersmith and Fulham	Grocery_Pharmacy	-27.095238
7	London Borough of Camden	Grocery_Pharmacy	-37.523810
1	City of Westminster	Grocery_Pharmacy	-46.761905
0	City of London	Grocery_Pharmacy	-70.238095

197 rows × 3 columns

In [25]:

```
for i in variabl:
    t_min=second_lockdown_UK_mean_sorted[second_lockdown_UK_mean_sorted['variabl
e']==i].iloc[0]
    t_max=second_lockdown_UK_mean_sorted[second_lockdown_UK_mean_sorted['variabl
e']==i].iloc[-1]

    print(f'min change is for {i} is {t_min.sub_region_2} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_2} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is London Borough of Barking and
Dagenham with value of -38.857142857142854
max change is for Retail_Recreation is City of London with value of
-89.80952380952381


min change is for Grocery_Pharmacy is London Borough of Lewisham wit
h value of -3.9523809523809526
max change is for Grocery_Pharmacy is City of London with value of -
70.23809523809524


min change is for Parks is London Borough of Newham with value of 11
7.0952380952381
max change is for Parks is City of London with value of -84.80952380
952381


min change is for Transit_stations is London Borough of Barking and
Dagenham with value of -37.57142857142857
max change is for Transit_stations is City of London with value of -
78.80952380952381


min change is for Workplaces is London Borough of Barking and Dagenh
am with value of -33.857142857142854
max change is for Workplaces is City of London with value of -66.133
33333333334


min change is for Residential is London Borough of Islington with va
lue of 23.80952380952381
max change is for Residential is London Borough of Barking and Dagen
ham with value of 12.761904761904763
```

In [26]:

```
second_lockdown_UK_descriptive_stats = (
    second_lockdown_UK.groupby(["sub_region_2", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
second_lockdown_UK_descriptive_stats
```

Out[26]:

	sub_region_2	variable	min	max	mean	median	std
0	City of London	Grocery_Pharmacy	-75.0	-56.0	-70.238095	-72.0	5.347006
1	City of London	Parks	-89.0	-74.0	-84.809524	-85.0	3.444112
2	City of London	Retail_Recreation	-92.0	-85.0	-89.809524	-90.0	2.015417
3	City of London	Transit_stations	-81.0	-71.0	-78.809524	-80.0	2.522282
4	City of London	Workplaces	-69.0	-64.0	-66.133333	-66.0	1.302013
...	...	...	...	...	...	...	...
192	Royal Borough of Kingston upon Thames	Parks	-10.0	93.0	54.428571	55.0	33.069912
193	Royal Borough of Kingston upon Thames	Residential	11.0	23.0	19.523810	21.0	4.020187
194	Royal Borough of Kingston upon Thames	Retail_Recreation	-74.0	-57.0	-63.476190	-62.0	4.578417
195	Royal Borough of Kingston upon Thames	Transit_stations	-60.0	-44.0	-51.714286	-52.0	3.689754
196	Royal Borough of Kingston upon Thames	Workplaces	-53.0	-32.0	-47.809524	-52.0	7.040022

197 rows × 7 columns

first lockdown

In [27]:

```
first_lockdown_UK_mean = (
    first_lockdown_UK.groupby(["variable", "sub_region_2"])["value"]
    .mean()
    .reset_index()
)

first_lockdown_UK_mean_sorted = first_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_2", "variable", "value"]]
```

In [28]:

```
for i in variabl:
    t_min=first_lockdown_UK_mean_sorted[first_lockdown_UK_mean_sorted['variable'
]==i].iloc[0]
    t_max=first_lockdown_UK_mean_sorted[first_lockdown_UK_mean_sorted['variable'
]==i].iloc[-1]

    print(f'min change is for {i} is {t_min.sub_region_2} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_2} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is London Borough of Waltham For
est with value of -66.61904761904762
max change is for Retail_Recreation is City of London with value of
-96.0


min change is for Grocery_Pharmacy is London Borough of Lewisham wit
h value of -26.19047619047619
max change is for Grocery_Pharmacy is City of London with value of -
81.9047619047619


min change is for Parks is London Borough of Croydon with value of 4
0.142857142857146
max change is for Parks is City of London with value of -90.90476190
47619


min change is for Transit_stations is London Borough of Barking and
Dagenham with value of -61.904761904761905
max change is for Transit_stations is City of London with value of -
92.33333333333333


min change is for Workplaces is London Borough of Barking and Dagenh
am with value of -65.04761904761905
max change is for Workplaces is City of London with value of -84.428
57142857143


min change is for Residential is London Borough of Hammersmith and F
ulham with value of 37.266666666666666
max change is for Residential is London Borough of Barking and Dagen
ham with value of 25.571428571428573
```

In [29]:

```
first_lockdown_UK_descriptive_stats = (
    first_lockdown_UK.groupby(["sub_region_2", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
first_lockdown_UK_descriptive_stats
```

Out[29]:

	sub_region_2	variable	min	max	mean	median	std
0	City of London	Grocery_Pharmacy	-91.0	-72.0	-81.904762	-83.0	4.773937
1	City of London	Parks	-96.0	-87.0	-90.904762	-91.0	2.188716
2	City of London	Retail_Recreation	-99.0	-93.0	-96.000000	-96.0	1.378405
3	City of London	Transit_stations	-97.0	-87.0	-92.333333	-93.0	2.033060
4	City of London	Workplaces	-89.0	-80.0	-84.428571	-85.0	2.563480
...	...	...	...	...	...	...	...
192	Royal Borough of Kingston upon Thames	Parks	-38.0	41.0	13.687500	18.0	23.508066
193	Royal Borough of Kingston upon Thames	Residential	22.0	37.0	32.222222	34.0	4.917622
194	Royal Borough of Kingston upon Thames	Retail_Recreation	-91.0	-77.0	-82.619048	-81.0	3.667100
195	Royal Borough of Kingston upon Thames	Transit_stations	-84.0	-71.0	-77.476190	-78.0	3.010300
196	Royal Borough of Kingston upon Thames	Workplaces	-88.0	-61.0	-74.666667	-78.0	7.945649

197 rows × 7 columns

corelation

In [30]:

```
UK_mean = london_data.groupby("sub_region_2")[
    [
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
].mean()

# Check the data in the DataFrame of mean mobility change we computed
UK_mean
```

Out[30]:

sub_region_2	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplaces
City of London	-71.259259	-58.294723	-65.460746	-63.177522	-57.272727
City of Westminster	-61.560664	-31.286079	-35.431673	-56.229885	-48.742308
London Borough of Barking and Dagenham	-23.893997	-5.338442	54.286280	-34.256705	-36.652618
London Borough of Barnet	-29.389527	-7.459770	50.473615	-41.559387	-40.756066
London Borough of Bexley	-31.983397	-4.544061	36.230871	-43.371648	-37.819923
London Borough of Brent	-27.117497	-4.019157	67.255937	-38.916986	-38.439336
London Borough of Bromley	-29.655172	-3.939974	24.294889	-44.297573	-41.135377
London Borough of Camden	-59.791826	-30.435504	4.310567	-52.535121	-47.696154
London Borough of Croydon	-30.715198	-3.641124	43.848285	-44.825032	-36.247765
London Borough of Ealing	-26.604087	-5.389527	49.989446	-43.500639	-39.178799
London Borough of Enfield	-27.653895	-7.777778	25.773087	-42.277139	-35.895275
London Borough of Hackney	-42.409962	-5.667944	49.246702	-47.177522	-43.183908
London Borough of Hammersmith and Fulham	-47.515964	-19.461047	24.372032	-48.191571	-48.830769
London Borough of Haringey	-32.148148	-12.642401	32.320580	-43.575990	-46.706258
London Borough of Harrow	-28.797954	-2.694764	71.846358	-37.715198	-32.913155
London Borough of Havering	-29.049808	-6.095785	18.307388	-36.833972	-35.236271

sub_region_2	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplaces
London Borough of Hillingdon	-30.960409	-8.284802	60.963061	-56.014049	-36.523627
London Borough of Hounslow	-26.613027	-5.236271	34.843008	-37.243934	-36.002554
London Borough of Islington	-55.355045	-19.494253	32.201847	-64.200511	-50.503846
London Borough of Lambeth	-44.544061	-11.353768	25.678385	-51.174968	-42.699872
London Borough of Lewisham	-24.535121	-0.340996	50.795515	-41.329502	-44.185185
London Borough of Merton	-35.519796	-11.674330	23.606860	-45.126437	-41.200511
London Borough of Newham	-36.200511	-8.685824	147.387054	-42.398467	-37.397190
London Borough of Redbridge	-29.342273	-6.408685	28.374670	-44.058748	-42.378033
London Borough of Richmond upon Thames	-33.135550	-4.287179	29.488220	-46.877395	-50.096525
London Borough of Southwark	-43.363985	-15.542784	0.535121	-49.826309	-42.186462
London Borough of Sutton	-28.873533	-8.725415	39.916887	-48.143040	-41.744872
London Borough of Tower Hamlets	-47.049808	-15.140485	-13.584930	-52.744572	-47.791826
London Borough of Waltham Forest	-20.342273	-6.747126	25.879947	-43.535121	-42.888889
London Borough of Wandsworth	-38.323116	-13.988506	29.126943	-45.245211	-44.556833
Royal Borough of Greenwich	-30.939974	-5.644955	13.830931	-36.293742	-43.113665
Royal Borough of Kensington and Chelsea	-51.969349	-8.543590	-4.650396	-55.075351	-47.325611

	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplaces
sub_region_2					
Royal Borough of Kingston upon Thames	-36.174968	-13.084291	48.361635	-43.125160	-41.402564

city of Westminster

In [64]:

```
col=data[data['sub_region_2']=='City of Westminster']
UK_NADrop_col= col.dropna(
    subset=[
        "country_region",
        "sub_region_1",
        "date",
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
)

# Number of rows and columns in the DataFrame without NaNs
UK_NADrop_col.shape
```

Out[64]:

```
(768, 15)
```

In [47]:

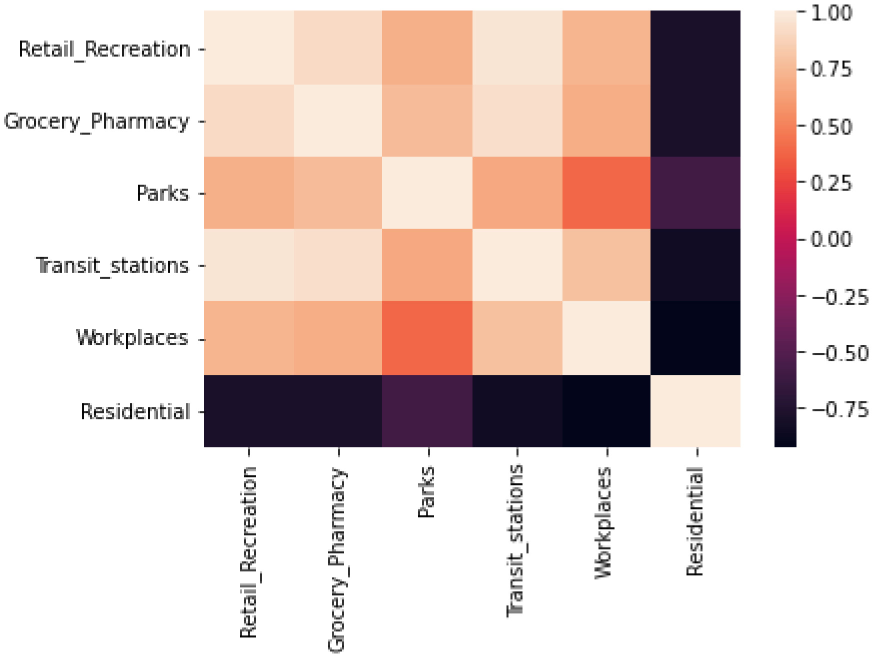
```
mobility_trends_UK_corr = col.iloc[:, 9:15].corr()
mobility_trends_UK_corr
```

Out[47]:

	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplac
Retail_Recreation	1.000000	0.917150	0.706955	0.969522	0.7257
Grocery_Pharmacy	0.917150	1.000000	0.764265	0.935028	0.6979
Parks	0.706955	0.764265	1.000000	0.675479	0.3891
Transit_stations	0.969522	0.935028	0.675479	1.000000	0.7886
Workplaces	0.725736	0.697965	0.389129	0.788607	1.0000
Residential	-0.794691	-0.792296	-0.587968	-0.834631	-0.9227

In [48]:

```
sns.heatmap(mobility_trends_UK_corr);
```



tower hamlets correlation

In [31]:

```
```

In [49]:

```
tower=data[data['sub_region_2']=='London Borough of Tower Hamlets']
```

In [50]:

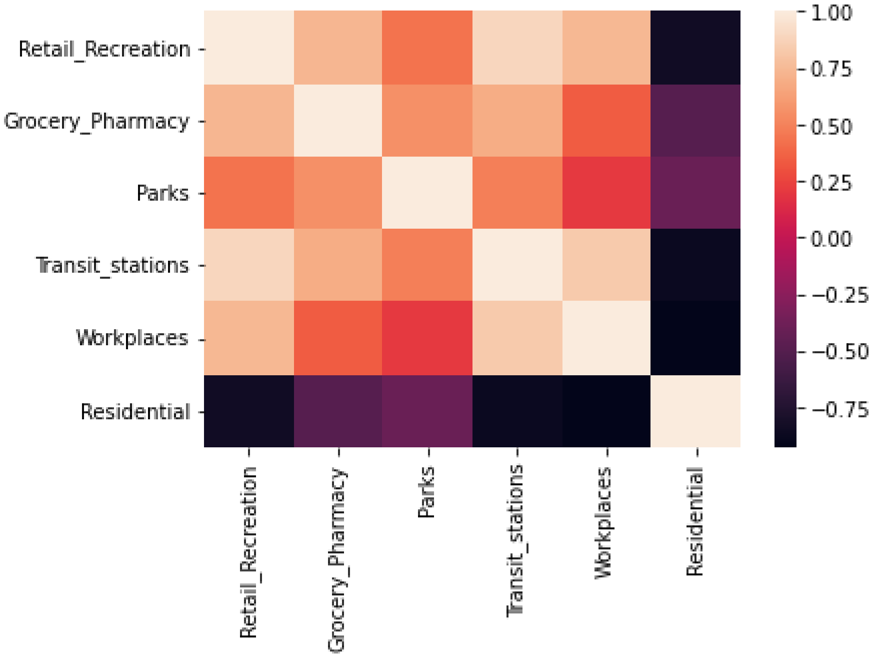
```
mobility_trends_UK_corr = tower.iloc[:, 9:15].corr()
mobility_trends_UK_corr
```

Out[50]:

	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplac
Retail_Recreation	1.000000	0.730188	0.430977	0.892447	0.738(
Grocery_Pharmacy	0.730188	1.000000	0.559293	0.684907	0.3437
Parks	0.430977	0.559293	1.000000	0.484571	0.2069
Transit_stations	0.892447	0.684907	0.484571	1.000000	0.831(
Workplaces	0.738090	0.343768	0.206967	0.831061	1.000(
Residential	-0.839647	-0.491185	-0.407533	-0.871011	-0.9269

In [51]:

```
sns.heatmap(mobility_trends_UK_corr);
```



Tower hamlets lockdown

In [34]:

```
bourgh=data_long.sub_region_2.unique()
```

In [35]:

```
tower=data_long[data_long["sub_region_2"] == "London Borough of Tower Hamlets"]
first_lockdown_UK,second_lockdown_UK,third_lockdown_UK=ld(tower)
```

In [36]:

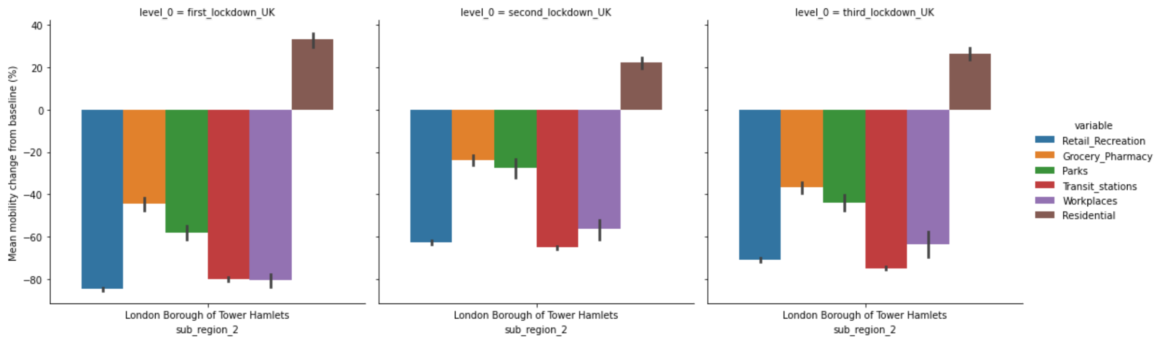
```
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

In [37]:

```
# Display the three lockdowns as a catplot multi-plot
grid = sns.catplot(
    kind="bar",
    x="sub_region_2",
    y="value",
    hue="variable",
    col="level_0",
    data=three_lockdowns_UK,
)
grid.set_ylabels("Mean mobility change from baseline (%)")
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x7fe5d5b5acd0>
```



In [ ]:

tower hamlets regression

In [59]:

```
UK_NADrop_tower= tower.dropna(
    subset=[
        "country_region",
        "sub_region_1",
        "date",
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
)

# Number of rows and columns in the DataFrame without NaNs
UK_NADrop_tower.shape
```

Out[59]:

```
(783, 15)
```

In [60]:

```
for i in range(len(variabl)):
    for j in range(i+1,len(variabl)):
        print(f'for {variabl[i]} and {variabl[j]} linear regression val is')
        model_outputs = stats.linregress(
        UK_NADrop_tower[variabl[i]], UK_NADrop_tower[variabl[j]])
        print(model_outputs)
        print('\n')
```

```
for Retail_Recreation and Grocery_Pharmacy linear regression val is
LinregressResult(slope=0.4913752304796176, intercept=7.9786251479809
88, rvalue=0.7301884264302877, pvalue=2.489914731174765e-131, stderr
=0.016452419982281415, intercept_stderr=0.8504464571029303)


for Retail_Recreation and Parks linear regression val is
LinregressResult(slope=0.5108280550466056, intercept=10.449432372818
583, rvalue=0.4309774138359571, pvalue=9.360402205387711e-37, stderr
=0.03827153271677052, intercept_stderr=1.9783040696705485)


for Retail_Recreation and Transit_stations linear regression val is
LinregressResult(slope=0.6863180602597102, intercept=-20.45343890170
1496, rvalue=0.8924467505711825, pvalue=3.4000461506203783e-272, std
err=0.012414842089434682, intercept_stderr=0.6417389345654179)


for Retail_Recreation and Workplaces linear regression val is
LinregressResult(slope=0.7964393940704284, intercept=-10.31950539265
059, rvalue=0.7380902766664937, pvalue=1.3284859833233748e-135, stde
rr=0.026051318474448738, intercept_stderr=1.3466256953879945)


for Retail_Recreation and Residential linear regression val is
LinregressResult(slope=-0.3927666400961281, intercept=-2.65200896697
49183, rvalue=-0.8396471056109742, pvalue=3.1079164173223936e-209, s
tderr=0.00909113158631897, intercept_stderr=0.46993212287116687)


for Grocery_Pharmacy and Parks linear regression val is
LinregressResult(slope=0.9851028223828072, intercept=1.3300050566387
984, rvalue=0.5592933409626778, pvalue=1.2172083328854827e-65, stder
r=0.05224624904916296, intercept_stderr=1.0919063294984426)


for Grocery_Pharmacy and Transit_stations linear regression val is
LinregressResult(slope=0.7827013724158757, intercept=-40.89409352491
672, rvalue=0.6849065129377235, pvalue=1.728495309583809e-109, stder
r=0.0297952408297318, intercept_stderr=0.6226975647629924)


for Grocery_Pharmacy and Workplaces linear regression val is
LinregressResult(slope=0.5512264026993599, intercept=-39.44599105491
582, rvalue=0.3437677072268708, pvalue=3.8615155368446857e-23, stder
r=0.05388033853352187, intercept_stderr=1.1260575400333726)


for Grocery_Pharmacy and Residential linear regression val is
LinregressResult(slope=-0.3414321864121227, intercept=10.65813720317
2778, rvalue=-0.4911848649633377, pvalue=8.67435819290633e-49, stder
r=0.02166604895256816, intercept_stderr=0.4528037211680231)


for Parks and Transit_stations linear regression val is
LinregressResult(slope=0.31439817557890154, intercept=-48.4734950272
8892, rvalue=0.4845706800290579, pvalue=2.4016722508394013e-47, stde
rr=0.02030871134691612, intercept_stderr=0.5845285023639575)


for Parks and Workplaces linear regression val is
```

```
LinregressResult(slope=0.1884190316644853, intercept=-45.23216699895
897, rvalue=0.2069673069181005, pvalue=5.049224118335105e-09, stderr
=0.03187064119920662, intercept_stderr=0.9173057734350116)


for Parks and Residential linear regression val is
LinregressResult(slope=-0.16083489853009653, intercept=13.6426554078
35713, rvalue=-0.4075329435194508, pvalue=1.0990431289872208e-32, st
derr=0.01289595291394231, intercept_stderr=0.371173331216186)


for Transit_stations and Workplaces linear regression val is
LinregressResult(slope=1.1660932164260875, intercept=13.713261488098
325, rvalue=0.8310614445293381, pvalue=3.541367564242635e-201, stder
r=0.02792482529041059, intercept_stderr=1.54296554630431)


for Transit_stations and Residential linear regression val is
LinregressResult(slope=-0.5298080007546805, intercept=-12.1169101189
87928, rvalue=-0.871011379210925, pvalue=2.700487871921114e-243, std
err=0.010692594091016586, intercept_stderr=0.5908113698645476)


for Workplaces and Residential linear regression val is
LinregressResult(slope=-0.40181764721463786, intercept=-3.3760129967
036576, rvalue=-0.9269033615459982, pvalue=0.0, stderr=0.00582167158
3430434, intercept_stderr=0.30902533361951584)
```

In [61]:

```
for i in range(len(variabl)):
    for j in range(i+1,len(variabl)):
        print(f'for {variabl[i]} and {variabl[j]} linear regression val is')
        X = sm.add_constant(UK_NADrop_tower[variabl[i]])
        Y = UK_NADrop_tower[variabl[j]]
        model = sm.OLS(Y, X)
        results = model.fit()

        print_model = results.summary()
        print(print_model)
        print('\n')
```

```
for Retail_Recreation and Grocery_Pharmacy linear regression val is
                            OLS Regression Results
==============================================================================
Dep. Variable:       Grocery_Pharmacy   R-squared:
0.533
Model:                            OLS   Adj. R-squared:
0.533
Method:                 Least Squares   F-statistic:
892.0
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
2.49e-131
Time:                        06:16:33   Log-Likelihood:
-2901.6
No. Observations:                 783   AIC:
5807.
Df Residuals:                     781   BIC:
5816.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|
[0.025      0.975]
-------------------------------------------------------------------------------------
const                 7.9786      0.850      9.382      0.000
6.309       9.648
Retail_Recreation     0.4914      0.016     29.866      0.000
0.459       0.524
==============================================================================
Omnibus:                       34.164   Durbin-Watson:
1.032
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
44.725
Skew:                          -0.416   Prob(JB):
1.94e-10
Kurtosis:                       3.823   Cond. No.
125.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Retail_Recreation and Parks linear regression val is
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Parks   R-squared:
0.186
Model:                            OLS   Adj. R-squared:
0.185
Method:                 Least Squares   F-statistic:
178.2
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
9.36e-37
Time:                        06:16:33   Log-Likelihood:
```

```
-3562.6
No. Observations:                 783   AIC:
7129.
Df Residuals:                     781   BIC:
7139.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------
-----------------
const                10.4494      1.978      5.282      0.000
6.566      14.333
Retail_Recreation     0.5108      0.038     13.347      0.000
0.436       0.586
=====================================================================
==========
Omnibus:                      227.407   Durbin-Watson:
0.614
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
837.526
Skew:                           1.339   Prob(JB):
1.36e-182
Kurtosis:                       7.301   Cond. No.
125.
=====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Retail_Recreation and Transit_stations linear regression val is
                            OLS Regression Results
=====================================================================
==========
Dep. Variable:       Transit_stations   R-squared:
0.796
Model:                            OLS   Adj. R-squared:
0.796
Method:                 Least Squares   F-statistic:
3056.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
3.40e-272
Time:                        06:16:33   Log-Likelihood:
-2681.1
No. Observations:                 783   AIC:
5366.
Df Residuals:                     781   BIC:
5376.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------
-----------------
```

```
const              -20.4534      0.642    -31.872      0.000     -2
1.713     -19.194
Retail_Recreation     0.6863      0.012     55.282      0.000
0.662       0.711
======================================================================
==========
Omnibus:                       19.442   Durbin-Watson:
0.390
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
20.513
Skew:                           0.360   Prob(JB):
3.51e-05
Kurtosis:                       3.333   Cond. No.
125.
======================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Retail_Recreation and Workplaces linear regression val is
                            OLS Regression Results
======================================================================
==========
Dep. Variable:             Workplaces   R-squared:
0.545
Model:                            OLS   Adj. R-squared:
0.544
Method:                 Least Squares   F-statistic:
934.6
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
1.33e-135
Time:                        06:16:33   Log-Likelihood:
-3261.4
No. Observations:                 783   AIC:
6527.
Df Residuals:                     781   BIC:
6536.
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
-----------------
const              -10.3195      1.347     -7.663      0.000     -1
2.963      -7.676
Retail_Recreation     0.7964      0.026     30.572      0.000
0.745       0.848
======================================================================
==========
Omnibus:                       83.921   Durbin-Watson:
1.262
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
65.395
Skew:                           0.612   Prob(JB):
6.30e-15
Kurtosis:                       2.290   Cond. No.
```

```
125.
==============================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Retail_Recreation and Residential linear regression val is
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:            Residential   R-squared:
0.705
Model:                            OLS   Adj. R-squared:
0.705
Method:                 Least Squares   F-statistic:
1867.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
3.11e-209
Time:                        06:16:33   Log-Likelihood:
-2437.1
No. Observations:                 783   AIC:
4878.
Df Residuals:                     781   BIC:
4888.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-----------------
const                -2.6520      0.470     -5.643      0.000      -
3.574      -1.730
Retail_Recreation    -0.3928      0.009    -43.203      0.000      -
0.411      -0.375
==============================================================================
==========
Omnibus:                       92.453   Durbin-Watson:
1.186
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
125.896
Skew:                          -0.982   Prob(JB):
4.59e-28
Kurtosis:                       2.971   Cond. No.
125.
==============================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Grocery_Pharmacy and Parks linear regression val is
                            OLS Regression Results
==============================================================================
==========
```

```
Dep. Variable:                  Parks   R-squared:
0.313
Model:                            OLS   Adj. R-squared:
0.312
Method:                 Least Squares   F-statistic:
355.5
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
1.22e-65
Time:                        06:16:33   Log-Likelihood:
-3496.2
No. Observations:                 783   AIC:
6996.
Df Residuals:                     781   BIC:
7006.
Df Model:                           1
Covariance Type:            nonrobust
===========================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------------
----------------
const                1.3300      1.092      1.218      0.224      -
0.813       3.473
Grocery_Pharmacy     0.9851      0.052     18.855      0.000
0.883       1.088
===========================================================================
==========
Omnibus:                      288.146   Durbin-Watson:
0.798
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
1632.067
Skew:                           1.562   Prob(JB):
0.00
Kurtosis:                       9.346   Cond. No.
30.4
===========================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Grocery_Pharmacy and Transit_stations linear regression val is
                            OLS Regression Results
===========================================================================
==========
Dep. Variable:       Transit_stations   R-squared:
0.469
Model:                            OLS   Adj. R-squared:
0.468
Method:                 Least Squares   F-statistic:
690.1
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
1.73e-109
Time:                        06:16:33   Log-Likelihood:
-3056.4
No. Observations:                 783   AIC:
6117.
Df Residuals:                     781   BIC:
```

```
6126.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const             -40.8941      0.623    -65.672      0.000     -42.116     -39.672
Grocery_Pharmacy    0.7827      0.030     26.269      0.000       0.724       0.841
==============================================================================
Omnibus:                      166.281   Durbin-Watson:                   0.916
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              287.308
Skew:                           1.315   Prob(JB):                     4.09e-63
Kurtosis:                       4.375   Cond. No.                         30.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


for Grocery_Pharmacy and Workplaces linear regression val is
                             OLS Regression Results
==============================================================================
Dep. Variable:             Workplaces   R-squared:                       0.118
Model:                            OLS   Adj. R-squared:                  0.117
Method:                 Least Squares   F-statistic:                     104.7
Date:                Fri, 30 Sep 2022   Prob (F-statistic):           3.86e-23
Time:                        06:16:33   Log-Likelihood:                -3520.3
No. Observations:                 783   AIC:                             7045.
Df Residuals:                     781   BIC:                             7054.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const             -39.4460      1.126    -35.030      0.000     -41.656     -37.236
Grocery_Pharmacy    0.5512      0.054     10.231      0.000       0.445       0.657
```

```
==============================================================================
==========
Omnibus:                      130.030   Durbin-Watson:
1.158
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
109.354
Skew:                           0.826   Prob(JB):
1.79e-24
Kurtosis:                       2.211   Cond. No.
30.4
==============================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Grocery_Pharmacy and Residential linear regression val is
                            OLS Regression Results
==============================================================================
==========
Dep. Variable:            Residential   R-squared:
0.241
Model:                            OLS   Adj. R-squared:
0.240
Method:                 Least Squares   F-statistic:
248.3
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
8.67e-49
Time:                        06:16:34   Log-Likelihood:
-2807.0
No. Observations:                 783   AIC:
5618.
Df Residuals:                     781   BIC:
5627.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------
const               10.6581      0.453     23.538      0.000
9.769      11.547
Grocery_Pharmacy    -0.3414      0.022    -15.759      0.000      -
0.384      -0.299
==============================================================================
==========
Omnibus:                       54.111   Durbin-Watson:
1.099
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
44.361
Skew:                          -0.499   Prob(JB):
2.33e-10
Kurtosis:                       2.397   Cond. No.
30.4
==============================================================================
==========
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Parks and Transit_stations linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:       Transit_stations   R-squared:
0.235
Model:                            OLS   Adj. R-squared:
0.234
Method:                 Least Squares   F-statistic:
239.7
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
2.40e-47
Time:                        06:16:34   Log-Likelihood:
-3199.5
No. Observations:                 783   AIC:
6403.
Df Residuals:                     781   BIC:
6412.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------
----------
const        -48.4735      0.585    -82.928      0.000     -49.621
-47.326
Parks          0.3144      0.020     15.481      0.000       0.275
0.354
====================================================================
==========
Omnibus:                       70.433   Durbin-Watson:
0.409
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
94.147
Skew:                           0.712   Prob(JB):
3.60e-21
Kurtosis:                       3.926   Cond. No.
32.7
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Parks and Workplaces linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:             Workplaces   R-squared:
0.043
Model:                            OLS   Adj. R-squared:
0.042
```

```
Method:                 Least Squares   F-statistic:
34.95
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
5.05e-09
Time:                        06:16:34   Log-Likelihood:
-3552.4
No. Observations:                 783   AIC:
7109.
Df Residuals:                     781   BIC:
7118.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
---------------------------------------------------------------------
----------
const        -45.2322      0.917    -49.310      0.000     -47.033
-43.431
Parks          0.1884      0.032      5.912      0.000       0.126
0.251
=====================================================================
==========
Omnibus:                       70.618   Durbin-Watson:
0.904
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
57.820
Skew:                           0.579   Prob(JB):
2.78e-13
Kurtosis:                       2.343   Cond. No.
32.7
=====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Parks and Residential linear regression val is
                            OLS Regression Results
=====================================================================
==========
Dep. Variable:            Residential   R-squared:
0.166
Model:                            OLS   Adj. R-squared:
0.165
Method:                 Least Squares   F-statistic:
155.5
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
1.10e-32
Time:                        06:16:34   Log-Likelihood:
-2844.0
No. Observations:                 783   AIC:
5692.
Df Residuals:                     781   BIC:
5701.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
```

```
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
----------
const         13.6427      0.371     36.755      0.000      12.914
14.371
Parks         -0.1608      0.013    -12.472      0.000      -0.186
-0.136
======================================================================
==========
Omnibus:                      39.213   Durbin-Watson:
0.744
Prob(Omnibus):                 0.000   Jarque-Bera (JB):
16.113
Skew:                          0.035   Prob(JB):
0.000317
Kurtosis:                      2.301   Cond. No.
32.7
======================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Transit_stations and Workplaces linear regression val is
                            OLS Regression Results
======================================================================
==========
Dep. Variable:             Workplaces   R-squared:
0.691
Model:                            OLS   Adj. R-squared:
0.690
Method:                 Least Squares   F-statistic:
1744.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
3.54e-201
Time:                        06:16:34   Log-Likelihood:
-3110.2
No. Observations:                 783   AIC:
6224.
Df Residuals:                     781   BIC:
6234.
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
----------------
const               13.7133      1.543      8.888      0.000      1
0.684      16.742
Transit_stations     1.1661      0.028     41.758      0.000
1.111      1.221
======================================================================
==========
Omnibus:                      66.502   Durbin-Watson:
1.253
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
82.572
Skew:                           0.795   Prob(JB):
1.17e-18
Kurtosis:                       3.055   Cond. No.
185.
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Transit_stations and Residential linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:            Residential   R-squared:
0.759
Model:                            OLS   Adj. R-squared:
0.758
Method:                 Least Squares   F-statistic:
2455.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
2.70e-243
Time:                        06:16:34   Log-Likelihood:
-2358.5
No. Observations:                 783   AIC:
4721.
Df Residuals:                     781   BIC:
4730.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------
----------------
const              -12.1169      0.591    -20.509      0.000     -1
3.277     -10.957
Transit_stations    -0.5298      0.011    -49.549      0.000      -
0.551     -0.509
====================================================================
==========
Omnibus:                       26.381   Durbin-Watson:
0.984
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
28.156
Skew:                          -0.448   Prob(JB):
7.69e-07
Kurtosis:                       3.249   Cond. No.
185.
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

```
for Workplaces and Residential linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:            Residential   R-squared:
0.859
Model:                            OLS   Adj. R-squared:
0.859
Method:                 Least Squares   F-statistic:
4764.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:16:34   Log-Likelihood:
-2147.7
No. Observations:                 783   AIC:
4299.
Df Residuals:                     781   BIC:
4309.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------
----------
const         -3.3760      0.309    -10.925      0.000      -3.983
-2.769
Workplaces    -0.4018      0.006    -69.021      0.000      -0.413
-0.390
====================================================================
==========
Omnibus:                        4.906   Durbin-Watson:
0.552
Prob(Omnibus):                  0.086   Jarque-Bera (JB):
4.748
Skew:                          -0.181   Prob(JB):
0.0931
Kurtosis:                       3.118   Cond. No.
122.
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

Tower hamlet regression against Southwark

In [79]:

```
tower=data[data['sub_region_2']=='London Borough of Tower Hamlets']
UK_NADrop_tower= tower.dropna(
    subset=[
        "country_region",
        "sub_region_1",
        "date",
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
)
UK_NADrop_tower.set_index('date',inplace=True)
# Number of rows and columns in the DataFrame without NaNs
UK_NADrop_tower.shape
```

Out[79]:

```
(783, 14)
```

In [80]:

```
Southwark=data[data['sub_region_2']=='London Borough of Southwark']
UK_NADrop_Southwark= Southwark.dropna(
    subset=[
        "country_region",
        "sub_region_1",
        "date",
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
)
UK_NADrop_Southwark.set_index('date',inplace=True)
# Number of rows and columns in the DataFrame without NaNs
UK_NADrop_Southwark.shape
```

Out[80]:

```
(783, 14)
```

In [81]:

```
for i in range(len(variabl)):
    print(f'for {variabl[i]} linear regression val is')
    X = sm.add_constant(UK_NADrop_tower[variabl[i]])
    Y = UK_NADrop_Southwark[variabl[i]]
    model = sm.OLS(Y, X)
    results = model.fit()

    print_model = results.summary()
    print(print_model)
    print('\n')
```

```
for Retail_Recreation linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:      Retail_Recreation   R-squared:
0.909
Model:                            OLS   Adj. R-squared:
0.909
Method:                 Least Squares   F-statistic:
7785.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:30:47   Log-Likelihood:
-2535.7
No. Observations:                 783   AIC:
5075.
Df Residuals:                     781   BIC:
5085.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------
-----------------
const                -0.5617      0.533     -1.054      0.292      -
1.608       0.484
Retail_Recreation     0.9097      0.010     88.232      0.000
0.889       0.930
====================================================================
==========
Omnibus:                       20.017   Durbin-Watson:
0.784
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
11.102
Skew:                           0.091   Prob(JB):
0.00388
Kurtosis:                       2.446   Cond. No.
125.
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Grocery_Pharmacy linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:       Grocery_Pharmacy   R-squared:
0.780
Model:                            OLS   Adj. R-squared:
0.779
Method:                 Least Squares   F-statistic:
2765.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
7.86e-259
Time:                        06:30:47   Log-Likelihood:
```

```
-2460.9
No. Observations:                 783   AIC:
4926.
Df Residuals:                     781   BIC:
4935.
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
----------------
const                -4.4548      0.291    -15.306      0.000      -
5.026      -3.883
Grocery_Pharmacy      0.7323      0.014     52.587      0.000
0.705       0.760
======================================================================
==========
Omnibus:                       33.240   Durbin-Watson:
1.378
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
83.892
Skew:                           0.155   Prob(JB):
6.07e-19
Kurtosis:                       4.573   Cond. No.
30.4
======================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Parks linear regression val is
                            OLS Regression Results
======================================================================
==========
Dep. Variable:                  Parks   R-squared:
0.853
Model:                            OLS   Adj. R-squared:
0.853
Method:                 Least Squares   F-statistic:
4534.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:30:47   Log-Likelihood:
-2897.3
No. Observations:                 783   AIC:
5799.
Df Residuals:                     781   BIC:
5808.
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
----------
```

```
const          13.1630      0.397     33.129      0.000      12.383
13.943
Parks           0.9295      0.014     67.336      0.000       0.902
0.957
====================================================================
==========
Omnibus:                      489.729   Durbin-Watson:
0.821
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
23725.153
Skew:                          -2.143   Prob(JB):
0.00
Kurtosis:                      29.624   Cond. No.
32.7
====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Transit_stations linear regression val is
                            OLS Regression Results
====================================================================
==========
Dep. Variable:       Transit_stations   R-squared:
0.927
Model:                            OLS   Adj. R-squared:
0.927
Method:                 Least Squares   F-statistic:
9894.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:30:47   Log-Likelihood:
-2375.6
No. Observations:                 783   AIC:
4755.
Df Residuals:                     781   BIC:
4765.
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------
----------------
const                7.5082      0.604     12.434      0.000
6.323       8.694
Transit_stations     1.0870      0.011     99.467      0.000
1.066       1.108
====================================================================
==========
Omnibus:                       12.228   Durbin-Watson:
0.605
Prob(Omnibus):                  0.002   Jarque-Bera (JB):
18.295
Skew:                           0.118   Prob(JB):
0.000106
Kurtosis:                       3.711   Cond. No.
```

```
185.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Workplaces linear regression val is
                            OLS Regression Results
==============================================================================
Dep. Variable:             Workplaces   R-squared:
0.980
Model:                            OLS   Adj. R-squared:
0.980
Method:                 Least Squares   F-statistic:
3.864e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:30:47   Log-Likelihood:
-2019.8
No. Observations:                 783   AIC:
4044.
Df Residuals:                     781   BIC:
4053.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
const          4.2652      0.262     16.250      0.000       3.750
4.780
Workplaces     0.9720      0.005    196.568      0.000       0.962
0.982
==============================================================================
Omnibus:                        1.974   Durbin-Watson:
0.690
Prob(Omnibus):                  0.373   Jarque-Bera (JB):
1.826
Skew:                           0.033   Prob(JB):
0.401
Kurtosis:                       2.773   Cond. No.
122.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.


for Residential linear regression val is
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:            Residential   R-squared:
0.991
Model:                            OLS   Adj. R-squared:
0.991
Method:                 Least Squares   F-statistic:
8.919e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        06:30:47   Log-Likelihood:
-1020.7
No. Observations:                 783   AIC:
2045.
Df Residuals:                     781   BIC:
2055.
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
===========
                  coef    std err          t      P>|t|      [0.025
0.975]
---------------------------------------------------------------------
-----------
const          7.9e-05      0.060      0.001      0.999      -0.117
0.117
Residential     0.9509      0.003    298.642      0.000       0.945
0.957
=====================================================================
==========
Omnibus:                      150.773   Durbin-Watson:
1.173
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
387.437
Skew:                           0.993   Prob(JB):
7.40e-85
Kurtosis:                       5.816   Cond. No.
35.1
=====================================================================
==========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

In [97]:

```
#after lockdown improvements
```

In [101]:

```
def ld(data_long):
    first_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2020-04-13")
        & (data_long["date"] <= "2020-11-05")
    ]

    second_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2020-11-25")
        & (data_long["date"] <= "2021-01-06")
    ]

    third_lockdown_UK = data_long[
        (data_long["country_region"] == "United Kingdom")
        & (data_long["date"] >= "2021-01-26")

    ]
    return [first_lockdown_UK,second_lockdown_UK,third_lockdown_UK]
```

In [102]:

```
first_lockdown_UK,second_lockdown_UK,third_lockdown_UK=ld(data_long)
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

In [103]:

```
third_lockdown_UK_mean = (
    third_lockdown_UK.groupby(["variable", "sub_region_2"])["value"]
    .mean()
    .reset_index()
)
```

In [104]:

```
third_lockdown_UK_mean_sorted = third_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_2", "variable", "value"]]
third_lockdown_UK_mean_sorted
```

Out[104]:

	sub_region_2	variable	value
178	London Borough of Harrow	Workplaces	-25.940503
181	London Borough of Hounslow	Workplaces	-30.592677
179	London Borough of Havering	Workplaces	-30.807780
186	London Borough of Newham	Workplaces	-31.354691
172	London Borough of Croydon	Workplaces	-32.185355
...	...	...	...
12	London Borough of Hammersmith and Fulham	Grocery_Pharmacy	-14.139588
18	London Borough of Islington	Grocery_Pharmacy	-16.615561
1	City of Westminster	Grocery_Pharmacy	-21.757437
7	London Borough of Camden	Grocery_Pharmacy	-23.745995
0	City of London	Grocery_Pharmacy	-54.526316

197 rows × 3 columns

In [105]:

```
for i in variabl:
    t_min=third_lockdown_UK_mean_sorted[third_lockdown_UK_mean_sorted['variable'
]==i].iloc[0]
    t_max=third_lockdown_UK_mean_sorted[third_lockdown_UK_mean_sorted['variable'
]==i].iloc[-1]

    print(f'min change is for {i} is {t_min.sub_region_2} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_2} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is London Borough of Waltham For
est with value of -11.219679633867276
max change is for Retail_Recreation is City of London with value of
-65.12356979405034


min change is for Grocery_Pharmacy is London Borough of Harrow with
value of 5.780320366132723
max change is for Grocery_Pharmacy is City of London with value of -
54.526315789473685


min change is for Parks is London Borough of Newham with value of 17
6.9862385321101
max change is for Parks is City of London with value of -61.57665903
89016


min change is for Transit_stations is Royal Borough of Greenwich wit
h value of -30.787185354691076
max change is for Transit_stations is London Borough of Islington wi
th value of -66.76201372997711


min change is for Workplaces is London Borough of Harrow with value
of -25.94050343249428
max change is for Workplaces is City of London with value of -51.626
198083067095


min change is for Residential is London Borough of Islington with va
lue of 13.933638443935926
max change is for Residential is London Borough of Barking and Dagen
ham with value of 7.345537757437071
```

In [106]:

```
tower=data_long[data_long["sub_region_2"] == "London Borough of Tower Hamlets"]
first_lockdown_UK,second_lockdown_UK,third_lockdown_UK=ld(tower)
```

In [107]:

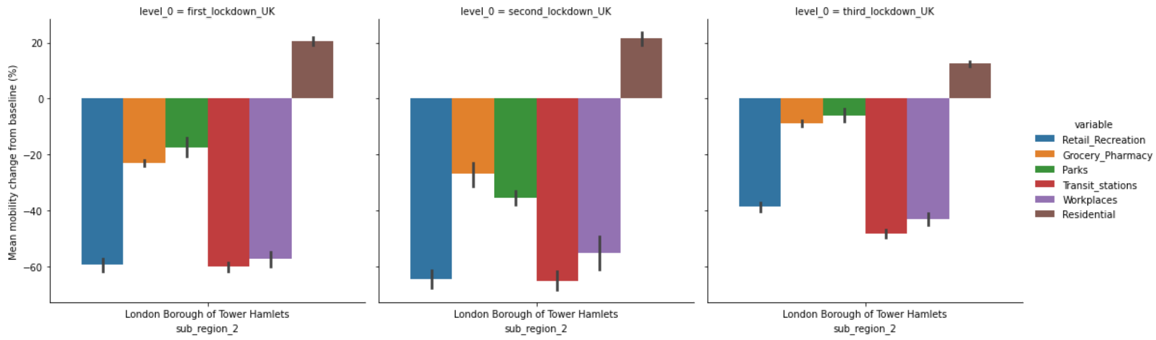
```
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

In [108]:

```
# Display the three lockdowns as a catplot multi-plot
grid = sns.catplot(
    kind="bar",
    x="sub_region_2",
    y="value",
    hue="variable",
    col="level_0",
    data=three_lockdowns_UK,
)
grid.set_ylabels("Mean mobility change from baseline (%)")
```

Out[108]:

```
<seaborn.axisgrid.FacetGrid at 0x7fe5c4bea590>
```



In [ ]: