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 [3]:

```
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 [4]:

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

In [5]:

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

In [8]:

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

In [ ]:

```

```

In [9]:

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

displaying data by year

In [10]:

```
dropdown_year = widgets.Dropdown(options = unique_sorted_values_plus_ALL(data.ye
ar))
output_year = widgets.Output()

def dropdown_year_eventhandler(change):
    output_year.clear_output()
    with output_year:
        if (change.new == ALL):
            display(data)
        else:
            display(data[data.year == change.new])

dropdown_year.observe(dropdown_year_eventhandler, names='value')
display(dropdown_year)
display(output_year)
```

In [11]:

```
regions=data.sub_region_1.unique()
```

In [12]:

```
regions=regions[1:]
variabl=["Retail_Recreation","Grocery_Pharmacy","Parks","Transit_stations","Work
places","Residential"]
```

In [13]:

```
c1 = widgets.Dropdown(options = regions )
c2 = widgets.Dropdown(options = regions )
c3 = widgets.Dropdown(options = regions )
c4 = widgets.Dropdown(options = regions )
c5= widgets.Dropdown(options = variabl )
```

plot of data for different cities

In [14]:

```
display(c1)
display(c2)
display(c3)
display(c4)
display(c5)
btn = widgets.Button(description='plot')
output_plot = widgets.Output()

def btn_eventhandler(obj):
    output_plot.clear_output()
    with output_plot:
        bigcities = data[
        data["sub_region_1"].isin(
            [c1.value, c2.value, c3.value, c4.value]
        )]
        sns.catplot(
        x="sub_region_1",
        y=c5.value ,
        kind="box",
        data=bigcities,
        height=6,
        aspect=1.5,)
        plt.show()

btn.on_click(btn_eventhandler)
display(btn)
display(output_plot)
```

In [ ]:

```
```

In [15]:

```
london_data=data.loc[data['sub_region_1'] == 'Greater London']
greater=london_data[london_data['sub_region_2'].isnull()]
```

In [16]:

```
london_data
```

Out[16]:

	country_region_code	country_region	sub_region_1	sub_region_2	metro_area	iso_3166
36253	GB	United Kingdom	Greater London	NaN	NaN	
36254	GB	United Kingdom	Greater London	NaN	NaN	
36255	GB	United Kingdom	Greater London	NaN	NaN	
36256	GB	United Kingdom	Greater London	NaN	NaN	
36257	GB	United Kingdom	Greater London	NaN	NaN	
...	...	...	...	...	...	
14337	GB	United Kingdom	Greater London	Royal Borough of Kingston upon Thames	NaN	
14338	GB	United Kingdom	Greater London	Royal Borough of Kingston upon Thames	NaN	
14339	GB	United Kingdom	Greater London	Royal Borough of Kingston upon Thames	NaN	
14340	GB	United Kingdom	Greater London	Royal Borough of Kingston upon Thames	NaN	
14341	GB	United Kingdom	Greater London	Royal Borough of Kingston upon Thames	NaN	

26622 rows × 16 columns

In [21]:

```
# data=data[(data['sub_region_1']=='Greater London') & (data['sub_region_2'].isn
ull())]['sub_region_2'].replace(np.nan,'all',inplace=True)
```

In [22]:

```
# data[(data['sub_region_1']=='Greater London') & (data['sub_region_2'].isnull
())]['sub_region_2']
```

In [23]:

```
# data.loc[data['sub_region_1'] == 'Greater London']
```

In [17]:

```
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 [18]:

```
c1 = widgets.Dropdown(options = london_regions )
c2 = widgets.Dropdown(options = london_regions )
c3 = widgets.Dropdown(options = london_regions )
c4 = widgets.Dropdown(options = london_regions )
c5= widgets.Dropdown(options = variabl )
```

In [19]:

```
display(c1)
display(c2)
display(c3)
display(c4)
display(c5)
btn = widgets.Button(description='plot')
output_plot = widgets.Output()

def btn_eventhandler(obj):
    output_plot.clear_output()
    with output_plot:
        bigcities = data[
        data["sub_region_2"].isin(
            [c1.value, c2.value, c3.value, c4.value]
        )]
        sns.catplot(
        x="sub_region_2",
        y=c5.value ,
        kind="box",
        data=bigcities,
        height=6,
        aspect=1.5,)
        plt.show()

btn.on_click(btn_eventhandler)
display(btn)
display(output_plot)
```

In [20]:

```
london_long = pd.melt(
    london_data,
    id_vars=[ "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=london_data.columns[9:15],
).dropna()

london_long
```

Out[20]:

	sub_region_2	date	variable	value
321	City of London	2020-02-15	Retail_Recreation	-5.0
322	City of London	2020-02-16	Retail_Recreation	-1.0
323	City of London	2020-02-17	Retail_Recreation	-3.0
324	City of London	2020-02-18	Retail_Recreation	-2.0
325	City of London	2020-02-19	Retail_Recreation	-7.0
...	...	...	...	...
159727	Royal Borough of Kingston upon Thames	2022-04-03	Residential	0.0
159728	Royal Borough of Kingston upon Thames	2022-04-04	Residential	12.0
159729	Royal Borough of Kingston upon Thames	2022-04-05	Residential	10.0
159730	Royal Borough of Kingston upon Thames	2022-04-06	Residential	11.0
159731	Royal Borough of Kingston upon Thames	2022-04-07	Residential	11.0

152724 rows × 4 columns

In [21]:

```
london_regions=london_data.sub_region_2.unique()
london_regions=london_regions[1:]
```

In [23]:

```
c1 = widgets.Dropdown(options = london_regions )
c2 = widgets.Dropdown(options = london_regions )
c3 = widgets.Dropdown(options = london_regions )
c4 = widgets.Dropdown(options = london_regions )
display(c1)
display(c2)
display(c3)
display(c4)
btn = widgets.Button(description='plot')
output_plot = widgets.Output()
def btn_eventhandler(obj):
    output_plot.clear_output()
    with output_plot:
        sub = london_long[
        london_long["sub_region_2"].isin(
            [c1.value, c2.value, c3.value, c4.value]
        )]

        sns.catplot(
        x="sub_region_2",
        y="value",
        col="variable",
        col_wrap=2,
        kind="boxen",
        height=6,
        aspect=1.5,
        sharey=False,
        data=sub,
        )
        plt.show()



#         grid = sns.relplot(
#         x="date",
#         y="value",
#         hue="sub_region_2",
#         col="variable",
#         col_wrap=1,
#         height=6,
#         aspect=4,
#         linewidth=2,
#         ci=99,
#         seed=42,
#         facet_kws={"sharey": False, "sharex": True},
#         kind="line",
#         data=sub,)

#         grid.set(ylabel="Mean mobility change from baseline (%)")
#         grid.set_xticklabels(rotation=45)

#         # For each plot, draw a horizontal line at y = 0 representing the base
line
#         for ax in grid.axes.flat:
#             ax.axhline(color="gray", linestyle="--", lw=2)
```

```
btn.on_click(btn_eventhandler)
display(btn)
display(output_plot)
```

In [25]:

```
sub = london_long[
       london_long["sub_region_2"].isin(
           [c1.value, c2.value, c3.value, c4.value]
       )]
```

In [26]:

```
sub
```

Out[26]:

	sub_region_2	date	variable	value
321	City of London	2020-02-15	Retail_Recreation	-5.0
322	City of London	2020-02-16	Retail_Recreation	-1.0
323	City of London	2020-02-17	Retail_Recreation	-3.0
324	City of London	2020-02-18	Retail_Recreation	-2.0
325	City of London	2020-02-19	Retail_Recreation	-7.0
...	...	...	...	...
159242	London Borough of Tower Hamlets	2022-04-03	Residential	1.0
159243	London Borough of Tower Hamlets	2022-04-04	Residential	14.0
159244	London Borough of Tower Hamlets	2022-04-05	Residential	12.0
159245	London Borough of Tower Hamlets	2022-04-06	Residential	12.0
159246	London Borough of Tower Hamlets	2022-04-07	Residential	11.0

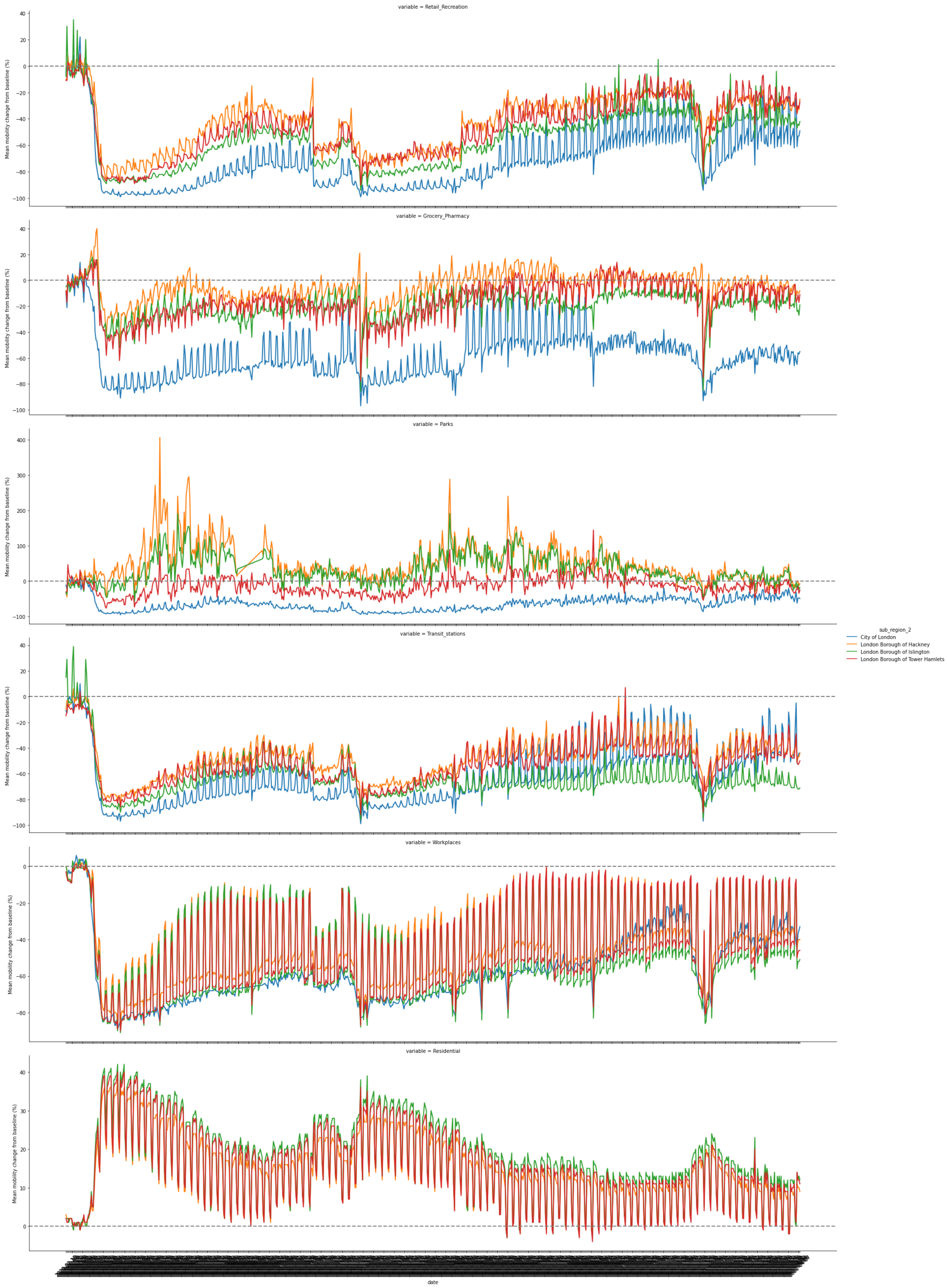
17697 rows × 4 columns

In [27]:

```
grid = sns.relplot(
    x="date",
    y="value",
    hue="sub_region_2",
    col="variable",
    col_wrap=1,
    height=6,
    aspect=4,
    linewidth=2,
    ci=99,
    seed=42,
    facet_kws={"sharey": False, "sharex": True},
    kind="line",
    data=sub,)

grid.set(ylabel="Mean mobility change from baseline (%)")
grid.set_xticklabels(rotation=45)

#          # For each plot, draw a horizontal line at y = 0 representing the base
line
for ax in grid.axes.flat:
    ax.axhline(color="gray", linestyle="--", lw=2)
```



bootstraping for tower hamlets

In [28]:

```
from scipy.stats import bootstrap
```

In [22]:

```
c=sub[(sub['sub_region_2']=='City of London')]
```

In [31]:

```
tower=london_data[london_data["sub_region_2"] == "London Borough of Tower Hamlet
s"]
```

In [47]:

```
b_d={"Retail_Recreation":(np.array(tower.Retail_Recreation),),
      "Grocery_Pharmacy":(np.array(tower.Grocery_Pharmacy),),
      "Parks":(np.array(tower.Parks),),
      "Transit_stations":(np.array(tower.Transit_stations),),
      "Workplaces":(np.array(tower.Workplaces),),
      "Residential":(np.array(tower.Workplaces),),}
```

In [60]:

```
for k,v in b_d.items():
    bootstrap_ci = bootstrap(v, np.median, confidence_level=0.95,
                         random_state=1, method='percentile')

    #view 95% boostrapped confidence interval
    print(f"tower hamlets , {k}. median {bootstrap_ci.confidence_interval}.")
```

```
tower hamlets , Retail_Recreation. median ConfidenceInterval(low=-4
8.0, high=-43.0).
tower hamlets , Grocery_Pharmacy. median ConfidenceInterval(low=-15.
0, high=-13.0).
tower hamlets , Parks. median ConfidenceInterval(low=-17.0, high=-1
3.0).
tower hamlets , Transit_stations. median ConfidenceInterval(low=-55.
0, high=-52.0).
tower hamlets , Workplaces. median ConfidenceInterval(low=-55.0, hig
h=-51.0).
tower hamlets , Residential. median ConfidenceInterval(low=-55.0, hi
gh=-51.0).
```

lockdowns analysis

In [61]:

```
data_long = pd.melt(
    data,
    id_vars=["country_region", "sub_region_1", "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 [62]:

```
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")
]
```

In [66]:

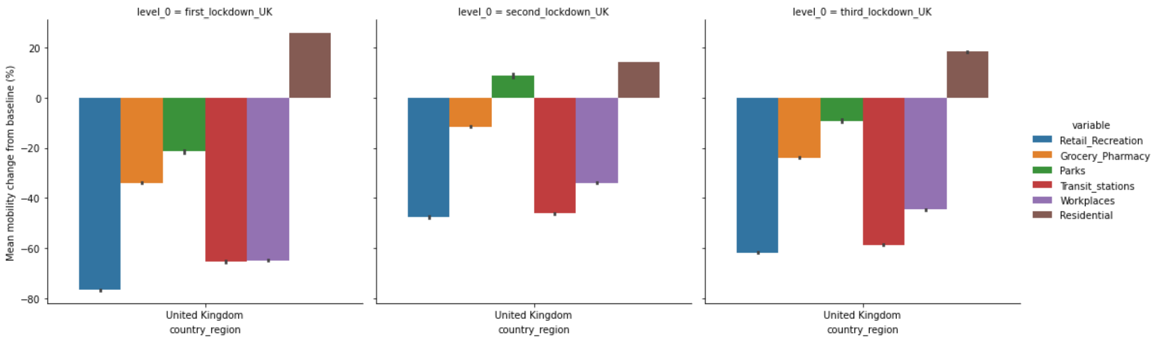
```
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

In [67]:

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

Out[67]:

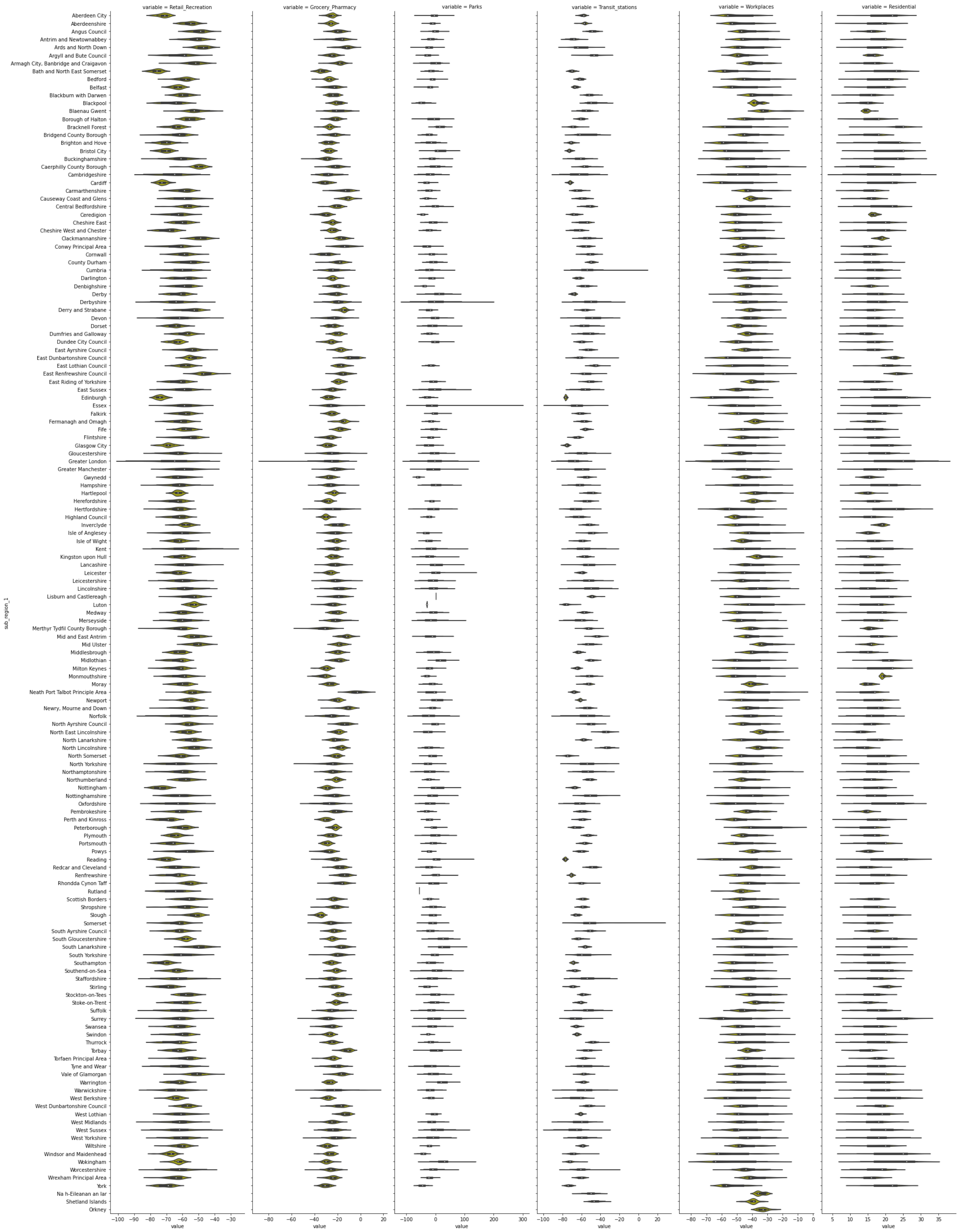
```
<seaborn.axisgrid.FacetGrid at 0x7fbdd51d9c10>
```



Third lockdown

In [78]:

```
sns.catplot(
    x="value",
    y="sub_region_1",
    col="variable",
    kind="violin",
    sharex=False,
    height=35,
    aspect=0.13,
    color="y",
    data=third_lockdown_UK,
);
```



In [68]:

```
third_lockdown_UK_mean = (
    third_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
third_lockdown_UK_mean
```

Out[68]:

	variable	sub_region_1	value
0	Grocery_Pharmacy	Aberdeen City	-25.333333
1	Grocery_Pharmacy	Aberdeenshire	-26.190476
2	Grocery_Pharmacy	Angus Council	-19.714286
3	Grocery_Pharmacy	Antrim and Newtownabbey	-18.523810
4	Grocery_Pharmacy	Ards and North Down	-13.285714
...	...	...	...
871	Workplaces	Windsor and Maidenhead	-54.952381
872	Workplaces	Wokingham	-55.714286
873	Workplaces	Worcestershire	-42.469388
874	Workplaces	Wrexham Principal Area	-38.428571
875	Workplaces	York	-53.904762

876 rows × 3 columns

In [69]:

```
third_lockdown_UK_mean_sorted = third_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
third_lockdown_UK_mean_sorted
```

Out[69]:

	sub_region_1	variable	value
737	Blaenau Gwent	Workplaces	-29.523810
803	Mid Ulster	Workplaces	-31.142857
816	North East Lincolnshire	Workplaces	-32.523810
825	Orkney	Workplaces	-32.866667
818	North Lincolnshire	Workplaces	-33.380952
...	...	...	...
25	Ceredigion	Grocery_Pharmacy	-31.142857
83	Monmouthshire	Grocery_Pharmacy	-32.095238
77	Merthyr Tydfil County Borough	Grocery_Pharmacy	-32.761905
7	Bath and North East Somerset	Grocery_Pharmacy	-34.095238
112	Slough	Grocery_Pharmacy	-35.523810

876 rows × 3 columns

In [91]:

```
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is East Renfrewshire Council wit
h value of -46.19047619047619
max change is for Retail_Recreation is Bath and North East Somerset
with value of -76.0


min change is for Grocery_Pharmacy is Neath Port Talbot Principle Ar
ea with value of -3.2857142857142856
max change is for Grocery_Pharmacy is Slough with value of -35.52380
9523809526


min change is for Parks is South Lanarkshire with value of 31.285714
285714285
max change is for Parks is Gwynedd with value of -59.714285714285715


min change is for Transit_stations is North Lincolnshire with value
of -33.285714285714285
max change is for Transit_stations is Reading with value of -77.1904
7619047619


min change is for Workplaces is Blaenau Gwent with value of -29.5238
09523809526
max change is for Workplaces is Edinburgh with value of -59.23809523
809524


min change is for Residential is Wokingham with value of 23.85714285
7142858
max change is for Residential is North East Lincolnshire with value
of 12.428571428571429
```

In [70]:

```
third_lockdown_UK_descriptive_stats = (
    third_lockdown_UK.groupby(["sub_region_1", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
third_lockdown_UK_descriptive_stats
```

Out[70]:

	sub_region_1	variable	min	max	mean	median	std
0	Aberdeen City	Grocery_Pharmacy	-33.0	-20.0	-25.333333	-24.0	2.921187
1	Aberdeen City	Parks	-51.0	42.0	-3.142857	-4.0	20.060622
2	Aberdeen City	Residential	11.0	24.0	19.809524	22.0	4.319943
3	Aberdeen City	Retail_Recreation	-79.0	-68.0	-72.047619	-71.0	3.138092
4	Aberdeen City	Transit_stations	-64.0	-55.0	-58.619048	-58.0	2.290768
...	...	...	...	...	...	...	...
871	York	Parks	-63.0	-22.0	-46.190476	-45.0	10.975514
872	York	Residential	13.0	25.0	20.666667	22.0	3.799123
873	York	Retail_Recreation	-78.0	-64.0	-70.571429	-69.0	4.307800
874	York	Transit_stations	-78.0	-69.0	-73.380952	-74.0	2.578298
875	York	Workplaces	-60.0	-41.0	-53.904762	-57.0	6.847662

876 rows × 7 columns

In [82]:

```
third_lockdown_UK_descriptive_stats[
    third_lockdown_UK_descriptive_stats["variable"] == "Retail_Recreation"
].sort_values(by="median")
```

Out[82]:

	sub_region_1	variable	min	max	mean	median	std
44	Bath and North East Somerset	Retail_Recreation	-82.0	-71.0	-76.000000	-76.0	2.898275
575	Nottingham	Retail_Recreation	-80.0	-68.0	-73.619048	-74.0	3.513918
284	Edinburgh	Retail_Recreation	-78.0	-69.0	-73.809524	-74.0	2.400397
126	Cardiff	Retail_Recreation	-77.0	-67.0	-72.523810	-73.0	2.441701
3	Aberdeen City	Retail_Recreation	-79.0	-68.0	-72.047619	-71.0	3.138092
...	...	...	...	...	...	...	...
114	Caerphilly County Borough	Retail_Recreation	-64.0	-46.0	-50.238095	-50.0	3.871754
167	Clackmannanshire	Retail_Recreation	-57.0	-42.0	-49.352941	-49.0	4.076475
15	Angus Council	Retail_Recreation	-58.0	-41.0	-49.904762	-49.0	4.380694
27	Ards and North Down	Retail_Recreation	-56.0	-41.0	-47.857143	-47.0	3.650832
266	East Renfrewshire Council	Retail_Recreation	-55.0	-35.0	-46.190476	-47.0	4.331501

148 rows × 7 columns

In [83]:

```
third_lockdown_UK_descriptive_stats[
    third_lockdown_UK_descriptive_stats["variable"] == "Parks"
].sort_values(by="max", ascending=False)
```

Out[83]:

	sub_region_1	variable	min	max	mean	median	std
288	Essex	Parks	-69.0	270.0	-3.511737	-12.0	45.749895
213	Derbyshire	Parks	-86.0	168.0	0.476744	-4.5	45.763114
330	Greater London	Parks	-94.0	131.0	-3.167376	-3.0	35.234568
412	Leicester	Parks	-24.0	110.0	5.666667	1.0	28.812035
628	Reading	Parks	-29.0	101.0	6.526316	1.0	28.083939
...	...	...	...	...	...	...	...
847	Windsor and Maidenhead	Parks	-63.0	-27.0	-43.666667	-42.0	9.640194
436	Luton	Parks	-30.0	-29.0	-29.333333	-29.0	0.577350
148	Ceredigion	Parks	-56.0	-33.0	-43.857143	-43.0	6.966245
342	Gwynedd	Parks	-71.0	-45.0	-59.714286	-61.0	6.827466
651	Rutland	Parks	-55.0	-55.0	-55.000000	-55.0	0.000000

135 rows × 7 columns

second lockdown

In [94]:

```
second_lockdown_UK_mean = (
    second_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
```

In [95]:

```
second_lockdown_UK_mean_sorted = second_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
```

In [96]:

```
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is Moray with value of -14.52380
9523809524
max change is for Retail_Recreation is Bath and North East Somerset
with value of -69.42857142857143


min change is for Grocery_Pharmacy is Neath Port Talbot Principle Ar
ea with value of 12.047619047619047
max change is for Grocery_Pharmacy is Slough with value of -24.95238
0952380953


min change is for Parks is Bridgend County Borough with value of 46.
833333333333336
max change is for Parks is Rutland with value of -51.0


min change is for Transit_stations is East Lothian Council with valu
e of -27.61904761904762
max change is for Transit_stations is Luton with value of -73.238095
23809524


min change is for Workplaces is Moray with value of -17.857142857142
858
max change is for Workplaces is Greater London with value of -46.789
54802259887


min change is for Residential is Wokingham with value of 19.47619047
6190474
max change is for Residential is Moray with value of 6.7777777777777
78
```

In [97]:

```
second_lockdown_UK_descriptive_stats = (
    second_lockdown_UK.groupby(["sub_region_1", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
second_lockdown_UK_descriptive_stats
```

Out[97]:

	sub_region_1	variable	min	max	mean	median	std
0	Aberdeen City	Grocery_Pharmacy	-15.0	-1.0	-7.238095	-7.0	3.192253
1	Aberdeen City	Parks	-28.0	57.0	3.809524	-2.0	21.720081
2	Aberdeen City	Residential	6.0	16.0	12.428571	14.0	3.295018
3	Aberdeen City	Retail_Recreation	-42.0	-32.0	-36.857143	-37.0	2.574601
4	Aberdeen City	Transit_stations	-46.0	-37.0	-40.571429	-40.0	2.693908
...	...	...	...	...	...	...	...
871	York	Parks	-58.0	-5.0	-28.809524	-24.0	15.028703
872	York	Residential	12.0	20.0	17.380952	18.0	2.479439
873	York	Retail_Recreation	-73.0	-53.0	-60.619048	-59.0	6.468974
874	York	Transit_stations	-75.0	-59.0	-64.761905	-63.0	4.657304
875	York	Workplaces	-49.0	-40.0	-45.857143	-47.0	2.868549

876 rows × 7 columns

first lockdown

In [99]:

```
first_lockdown_UK_mean = (
    first_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
first_lockdown_UK_mean_sorted = first_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is East Renfrewshire Council wit
h value of -64.4
max change is for Retail_Recreation is Bath and North East Somerset
with value of -86.38095238095238


min change is for Grocery_Pharmacy is East Renfrewshire Council with
value of -18.19047619047619
max change is for Grocery_Pharmacy is Monmouthshire with value of -4
5.333333333333336


min change is for Parks is South Gloucestershire with value of 5.0
max change is for Parks is Lisburn and Castlereagh with value of -7
5.33333333333333


min change is for Transit_stations is Scottish Borders with value of
-44.19047619047619
max change is for Transit_stations is Luton with value of -86.095238
0952381


min change is for Workplaces is North East Lincolnshire with value o
f -53.57142857142857
max change is for Workplaces is Edinburgh with value of -76.76190476
190476


min change is for Residential is Wokingham with value of 32.23076923
076923
max change is for Residential is Pembrokeshire with value of 18.0
```

In [100]:

```
first_lockdown_UK_descriptive_stats = (
    first_lockdown_UK.groupby(["sub_region_1", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
second_lockdown_UK_descriptive_stats
```

Out[100]:

	sub_region_1	variable	min	max	mean	median	std
0	Aberdeen City	Grocery_Pharmacy	-15.0	-1.0	-7.238095	-7.0	3.192253
1	Aberdeen City	Parks	-28.0	57.0	3.809524	-2.0	21.720081
2	Aberdeen City	Residential	6.0	16.0	12.428571	14.0	3.295018
3	Aberdeen City	Retail_Recreation	-42.0	-32.0	-36.857143	-37.0	2.574601
4	Aberdeen City	Transit_stations	-46.0	-37.0	-40.571429	-40.0	2.693908
...	...	...	...	...	...	...	...
871	York	Parks	-58.0	-5.0	-28.809524	-24.0	15.028703
872	York	Residential	12.0	20.0	17.380952	18.0	2.479439
873	York	Retail_Recreation	-73.0	-53.0	-60.619048	-59.0	6.468974
874	York	Transit_stations	-75.0	-59.0	-64.761905	-63.0	4.657304
875	York	Workplaces	-49.0	-40.0	-45.857143	-47.0	2.868549

876 rows × 7 columns

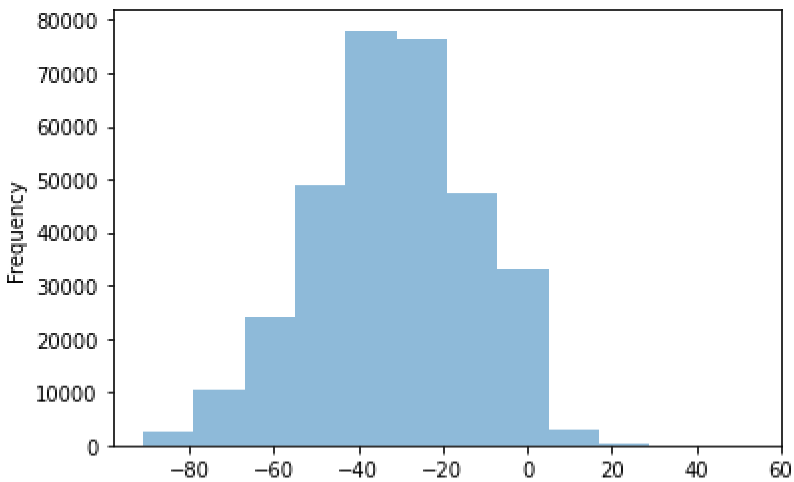
In [ ]:

```

```

In [92]:

```
ax = data["Workplaces"].plot.hist(bins=12, alpha=0.5)
```



In [ ]:

```

```

In [93]:

```
data.iloc[:, 9:15].mean()
```

Out[93]:

```
Retail_Recreation   -25.628914
Grocery_Pharmacy     -3.427401
Parks                30.829508
Transit_stations    -32.848772
Workplaces          -32.167015
Residential          10.469924
dtype: float64
```

In [95]:

```
data.iloc[:, 9:15].std()
```

Out[95]:

```
Retail_Recreation    27.519923
Grocery_Pharmacy     17.430261
Parks                55.238187
Transit_stations     24.075431
Workplaces           19.088629
Residential           7.068883
dtype: float64
```

In [101]:

```
UK_mean = data.groupby("sub_region_1")[
    [
        "Retail_Recreation",
        "Grocery_Pharmacy",
        "Parks",
        "Transit_stations",
        "Workplaces",
        "Residential",
    ]
].mean()

# Check the data in the DataFrame of mean mobility change we computed
UK_mean.head()
```

Out[101]:

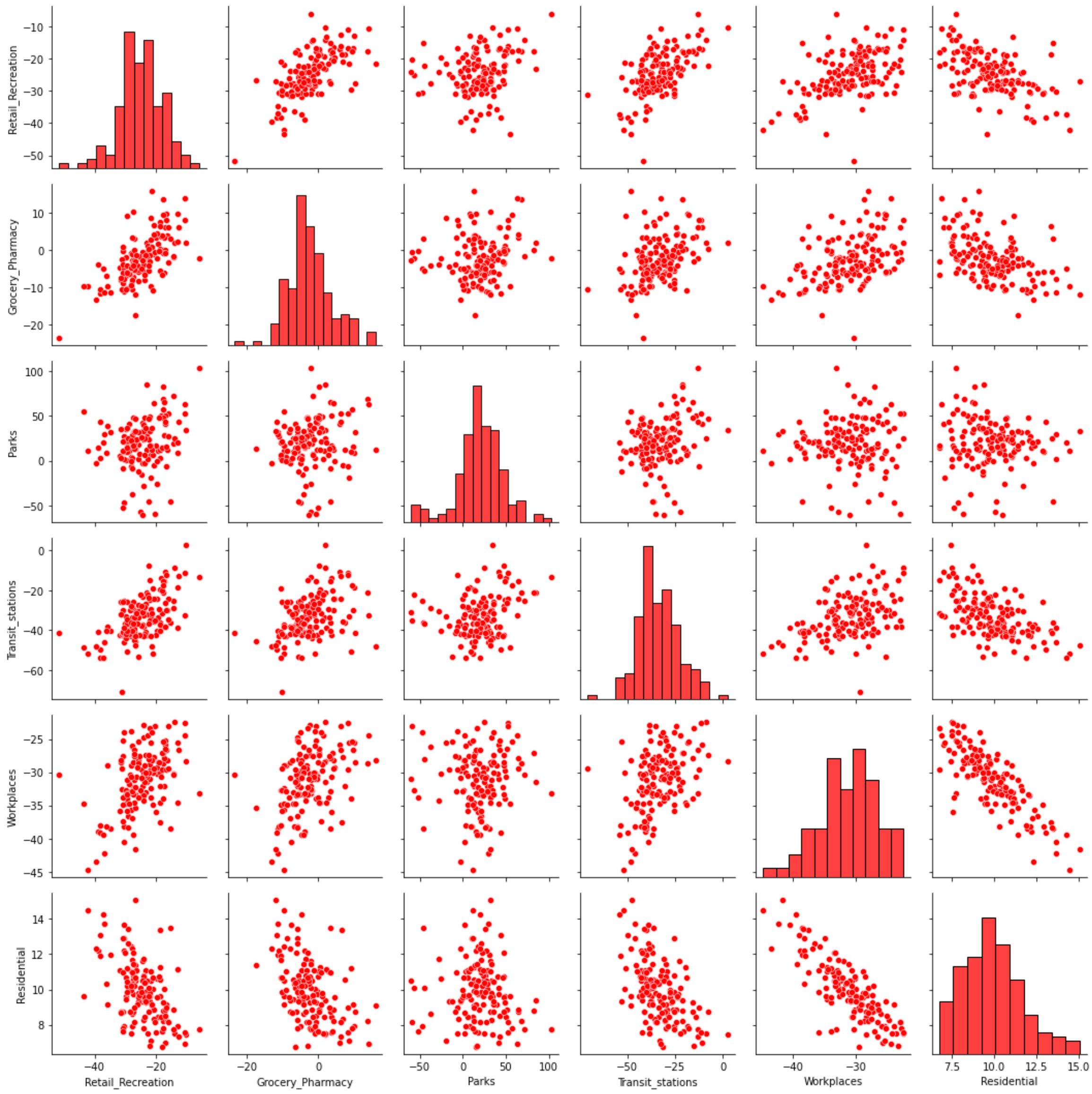
	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplaces
sub_region_1					
Aberdeen City	-38.800515	-3.892670	14.851802	-40.639640	-38.903475
Aberdeenshire	-17.288918	-6.057592	28.317881	-38.360158	-32.154440
Angus Council	-14.431398	-1.850923	18.748106	-29.073879	-30.151866
Antrim and Newtownabbey	-19.798153	-2.240106	-26.127660	-40.439314	-30.236808
Ards and North Down	-18.643799	7.485488	12.363515	-36.190588	-32.114543

In [102]:

```
grid = sns.PairGrid(UK_mean)
grid.map_diag(sns.histplot, color="r")
grid.map_offdiag(sns.scatterplot, color="r")
```

Out[102]:

```
<seaborn.axisgrid.PairGrid at 0x7fbdd4fc6890>
```



In [105]:

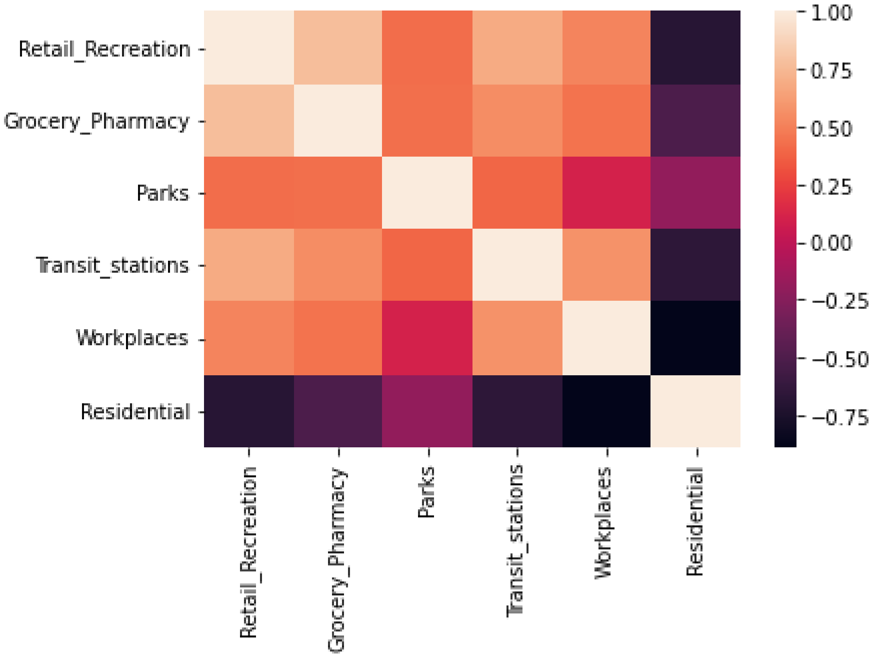
```
# Compute pairwise correlation between our six mobility categories
# using the original (non-aggregated) mobility_trends_UK DataFrame.
mobility_trends_UK_corr = data.iloc[:, 9:15].corr()
mobility_trends_UK_corr
```

Out[105]:

	Retail_Recreation	Grocery_Pharmacy	Parks	Transit_stations	Workplac
Retail_Recreation	1.000000	0.774223	0.422229	0.685147	0.5194
Grocery_Pharmacy	0.774223	1.000000	0.431409	0.555065	0.4404
Parks	0.422229	0.431409	1.000000	0.388513	0.1061
Transit_stations	0.685147	0.555065	0.388513	1.000000	0.5784
Workplaces	0.519476	0.440429	0.106138	0.578451	1.0000
Residential	-0.695228	-0.511430	-0.199375	-0.663171	-0.8887

In [106]:

```
# Plot a heatmap based on the correlation analysis
sns.heatmap(mobility_trends_UK_corr);
```



In [107]:

```
UK_NADrop = data.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.shape
```

Out[107]:

```
(242309, 16)
```

In [106]:

```
UK_NADrop=UK_NADrop.drop(['year', ], axis=1)
```

linear regression

In [122]:

```
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[variabl[i]], UK_NADrop[variabl[j]])
        print(model_outputs)
        print('\n')
```

```
for Retail_Recreation and Grocery_Pharmacy linear regression val is
LinregressResult(slope=0.4827166466083039, intercept=9.1791902544236
77, rvalue=0.7723367031159675, pvalue=0.0, stderr=0.0008065331968265
191, intercept_stderr=0.03024567938580045)


for Retail_Recreation and Parks linear regression val is
LinregressResult(slope=0.8116073658003042, intercept=53.115413835430
86, rvalue=0.4009359906770533, pvalue=0.0, stderr=0.0037673304605785
763, intercept_stderr=0.14127808960544078)


for Retail_Recreation and Transit_stations linear regression val is
LinregressResult(slope=0.6237201227403455, intercept=-16.83689763894
0602, rvalue=0.7096518046916471, pvalue=0.0, stderr=0.00125798326533
35256, intercept_stderr=0.04717544010053215)


for Retail_Recreation and Workplaces linear regression val is
LinregressResult(slope=0.3865795292388424, intercept=-21.67746334820
0114, rvalue=0.5377346771428168, pvalue=0.0, stderr=0.00123132923412
7156, intercept_stderr=0.0461758913090143)


for Retail_Recreation and Residential linear regression val is
LinregressResult(slope=-0.19069057119350036, intercept=5.62606074834
7025, rvalue=-0.7039972092744856, pvalue=0.0, stderr=0.0003908027797
442044, intercept_stderr=0.01465543591476649)


for Grocery_Pharmacy and Parks linear regression val is
LinregressResult(slope=1.3288082390797014, intercept=36.492835713147
1, rvalue=0.41027681690897455, pvalue=0.0, stderr=0.0060003783908343
15, intercept_stderr=0.10333442771969896)


for Grocery_Pharmacy and Transit_stations linear regression val is
LinregressResult(slope=0.8051364701299387, intercept=-30.34023696804
468, rvalue=0.5725463587416176, pvalue=0.0, stderr=0.002342189884874
7127, intercept_stderr=0.04033559812396155)


for Grocery_Pharmacy and Workplaces linear regression val is
LinregressResult(slope=0.5104625561587575, intercept=-30.00818439798
682, rvalue=0.44379119749567925, pvalue=0.0, stderr=0.00209398395886
89254, intercept_stderr=0.03606116480495219)


for Grocery_Pharmacy and Residential linear regression val is
LinregressResult(slope=-0.22497417336489223, intercept=9.82590788023
1666, rvalue=-0.5191107045696359, pvalue=0.0, stderr=0.0007525008014
315289, intercept_stderr=0.012959056014420772)


for Parks and Transit_stations linear regression val is
LinregressResult(slope=0.16207157466325198, intercept=-38.2443960028
621, rvalue=0.37327881627537934, pvalue=0.0, stderr=0.00081828968141
25932, intercept_stderr=0.05185828988271547)


for Parks and Workplaces linear regression val is
```

```
LinregressResult(slope=0.027120979886880822, intercept=-32.598458525
09002, rvalue=0.0763669603070138, pvalue=0.0, stderr=0.0007193608544
16196, intercept_stderr=0.04558877444744822)


for Parks and Residential linear regression val is
LinregressResult(slope=-0.02645210955839176, intercept=11.4316263195
8967, rvalue=-0.19768465877070676, pvalue=0.0, stderr=0.000266469898
60083776, intercept_stderr=0.016887263227864744)


for Transit_stations and Workplaces linear regression val is
LinregressResult(slope=0.4886704443238805, intercept=-15.57658189908
3553, rvalue=0.5974336882943647, pvalue=0.0, stderr=0.00133251933166
80141, intercept_stderr=0.05423700038145337)


for Transit_stations and Residential linear regression val is
LinregressResult(slope=-0.21044177995699906, intercept=3.62842775787
12016, rvalue=-0.6828387026551698, pvalue=0.0, stderr=0.000457395825
082221, intercept_stderr=0.018617198977823345)


for Workplaces and Residential linear regression val is
LinregressResult(slope=-0.33819117792270126, intercept=-0.1460163489
105284, rvalue=-0.8975833775370103, pvalue=0.0, stderr=0.00033743501
864197053, intercept_stderr=0.012553904891912146)
```

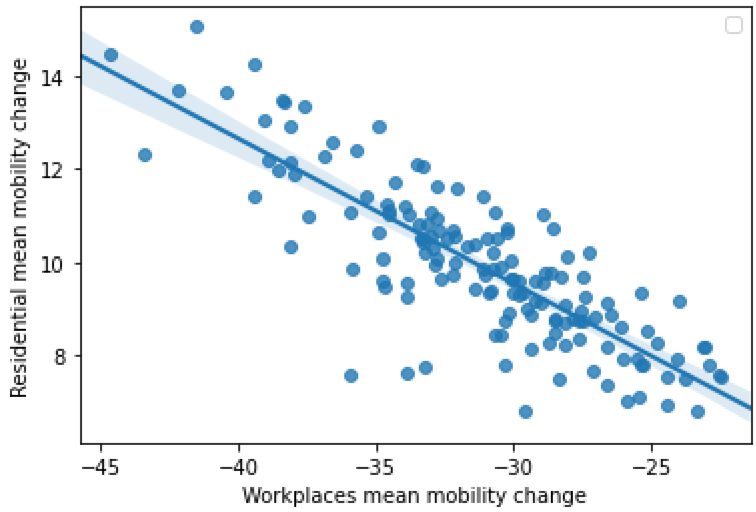
In [111]:

```
fig = sns.regplot(
    x="Workplaces",
    y="Residential",
#     label="r = {0:.3}, p-value < {1:.3}".format(r_value, 0.001),
    data=UK_mean,
)
fig.set(
    xlabel="Workplaces mean mobility change", ylabel="Residential mean mobility
 change"
)
fig.legend()
```

No handles with labels found to put in legend.

Out[111]:

<matplotlib.legend.Legend at 0x7fe264fc1950>



In [125]:

```
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[variabl[i]])
        Y = UK_NADrop[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:                       0.597
Model:                            OLS   Adj. R-squared:                  0.597
Method:                 Least Squares   F-statistic:                 3.582e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):               0.00
Time:                        05:06:20   Log-Likelihood:            -9.1877e+05
No. Observations:              242309   AIC:                         1.838e+06
Df Residuals:                  242307   BIC:                         1.838e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 9.1792      0.030    303.488      0.000       9.120       9.238
Retail_Recreation     0.4827      0.001    598.508      0.000       0.481       0.484
==============================================================================
Omnibus:                    21731.724   Durbin-Watson:                   0.560
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           152969.554
Skew:                          -0.055   Prob(JB):                         0.00
Kurtosis:                       6.891   Cond. No.                         52.1
==============================================================================

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:                       0.161
Model:                            OLS   Adj. R-squared:                  0.161
Method:                 Least Squares   F-statistic:                 4.641e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):               0.00
Time:                        05:06:20   Log-Likelihood:                      -
```

```
1.2923e+06
No. Observations:              242309   AIC:
2.585e+06
Df Residuals:                  242307   BIC:
2.585e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------
-----------------
const                53.1154      0.141    375.964      0.000      5
2.839      53.392
Retail_Recreation     0.8116      0.004    215.433      0.000
0.804      0.819
=====================================================================
==========
Omnibus:                    91903.841   Durbin-Watson:
0.345
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
536055.839
Skew:                           1.726   Prob(JB):
0.00
Kurtosis:                       9.417   Cond. No.
52.1
=====================================================================
==========

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:
0.504
Model:                            OLS   Adj. R-squared:
0.504
Method:                 Least Squares   F-statistic:
2.458e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:               -
1.0265e+06
No. Observations:              242309   AIC:
2.053e+06
Df Residuals:                  242307   BIC:
2.053e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------
-----------------
```

```
const                 -16.8369      0.047   -356.900      0.000     -1
6.929     -16.744
Retail_Recreation     0.6237      0.001    495.810      0.000
0.621       0.626
====================================================================
==========
Omnibus:                    76036.456   Durbin-Watson:
0.358
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
564892.212
Skew:                           1.312   Prob(JB):
0.00
Kurtosis:                      10.005   Cond. No.
52.1
====================================================================
==========

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:
0.289
Model:                            OLS   Adj. R-squared:
0.289
Method:                 Least Squares   F-statistic:
9.857e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:                -
1.0213e+06
No. Observations:              242309   AIC:
2.043e+06
Df Residuals:                  242307   BIC:
2.043e+06
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
=================
                        coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------
-----------------
const                 -21.6775      0.046   -469.454      0.000     -2
1.768     -21.587
Retail_Recreation     0.3866      0.001    313.953      0.000
0.384       0.389
====================================================================
==========
Omnibus:                      566.360   Durbin-Watson:
0.950
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
564.759
Skew:                          -0.112   Prob(JB):
2.31e-123
Kurtosis:                       2.924   Cond. No.
```

```
52.1
==============================================================================

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:                       0.496
Model:                            OLS   Adj. R-squared:                  0.496
Method:                 Least Squares   F-statistic:                 2.381e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):               0.00
Time:                        05:06:20   Log-Likelihood:            -7.4321e+05
No. Observations:              242309   AIC:                         1.486e+06
Df Residuals:                  242307   BIC:                         1.486e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 5.6261      0.015    383.889      0.000       5.597       5.655
Retail_Recreation    -0.1907      0.000   -487.946      0.000      -0.191      -0.190
==============================================================================
Omnibus:                     1779.179   Durbin-Watson:                   0.914
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1831.316
Skew:                           0.203   Prob(JB):                         0.00
Kurtosis:                       3.128   Cond. No.                         52.1
==============================================================================

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:
0.168
Model:                            OLS   Adj. R-squared:
0.168
Method:                 Least Squares   F-statistic:
4.904e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:             -
1.2912e+06
No. Observations:              242309   AIC:
2.582e+06
Df Residuals:                  242307   BIC:
2.582e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------
----------------
const               36.4928      0.103    353.153      0.000      3
6.290      36.695
Grocery_Pharmacy     1.3288      0.006    221.454      0.000
1.317       1.341
=====================================================================
==========
Omnibus:                    90161.841   Durbin-Watson:
0.393
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
528162.397
Skew:                           1.687   Prob(JB):
0.00
Kurtosis:                       9.398   Cond. No.
17.6
=====================================================================
==========

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:
0.328
Model:                            OLS   Adj. R-squared:
0.328
Method:                 Least Squares   F-statistic:
1.182e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:             -
1.0632e+06
No. Observations:              242309   AIC:
2.126e+06
Df Residuals:                  242307   BIC:
```

```
2.126e+06
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
================
                        coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
----------------
const               -30.3402      0.040   -752.195      0.000     -3
0.419     -30.261
Grocery_Pharmacy      0.8051      0.002    343.754      0.000
0.801       0.810
======================================================================
==========
Omnibus:                    56913.261   Durbin-Watson:
0.346
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
262322.426
Skew:                           1.077   Prob(JB):
0.00
Kurtosis:                       7.620   Cond. No.
17.6
======================================================================
==========

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:
0.197
Model:                            OLS   Adj. R-squared:
0.197
Method:                 Least Squares   F-statistic:
5.943e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:                -
1.0361e+06
No. Observations:              242309   AIC:
2.072e+06
Df Residuals:                  242307   BIC:
2.072e+06
Df Model:                           1
Covariance Type:            nonrobust
======================================================================
================
                        coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
----------------
const               -30.0082      0.036   -832.147      0.000     -3
0.079     -29.938
Grocery_Pharmacy      0.5105      0.002    243.776      0.000
0.506       0.515
```

```
==============================================================================
==========
Omnibus:                      576.822   Durbin-Watson:
0.854
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
462.978
Skew:                           0.019   Prob(JB):
2.92e-101
Kurtosis:                       2.789   Cond. No.
17.6
==============================================================================
==========

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:
0.269
Model:                            OLS   Adj. R-squared:
0.269
Method:                 Least Squares   F-statistic:
8.938e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:                    -
7.8808e+05
No. Observations:              242309   AIC:
1.576e+06
Df Residuals:                  242307   BIC:
1.576e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------
const                9.8259      0.013    758.227      0.000
9.801       9.851
Grocery_Pharmacy    -0.2250      0.001   -298.969      0.000      -
0.226      -0.223
==============================================================================
==========
Omnibus:                     2382.387   Durbin-Watson:
0.677
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
2454.882
Skew:                           0.243   Prob(JB):
0.00
Kurtosis:                       3.084   Cond. No.
17.6
==============================================================================
==========
```

```
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:
0.139
Model:                            OLS   Adj. R-squared:
0.139
Method:                 Least Squares   F-statistic:
3.923e+04
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:             -
1.0932e+06
No. Observations:              242309   AIC:
2.186e+06
Df Residuals:                  242307   BIC:
2.186e+06
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------
----------
const        -38.2444      0.052   -737.479      0.000     -38.346
-38.143
Parks          0.1621      0.001    198.061      0.000       0.160
0.164
====================================================================
==========
Omnibus:                    30914.879   Durbin-Watson:
0.229
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
83462.686
Skew:                           0.712   Prob(JB):
0.00
Kurtosis:                       5.498   Cond. No.
73.4
====================================================================
==========

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:
0.006
Model:                            OLS   Adj. R-squared:
0.006
```

```
Method:                 Least Squares   F-statistic:
1421.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
3.76e-310
Time:                        05:06:20   Log-Likelihood:                 -
1.0619e+06
No. Observations:              242309   AIC:
2.124e+06
Df Residuals:                  242307   BIC:
2.124e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
---------------------------------------------------------------------
----------
const        -32.5985      0.046   -715.055      0.000     -32.688
-32.509
Parks          0.0271      0.001     37.701      0.000       0.026
0.029
=====================================================================
==========
Omnibus:                     2928.012   Durbin-Watson:
0.620
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
2410.277
Skew:                          -0.177   Prob(JB):
0.00
Kurtosis:                       2.662   Cond. No.
73.4
=====================================================================
==========

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:
0.039
Model:                            OLS   Adj. R-squared:
0.039
Method:                 Least Squares   F-statistic:
9854.
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:                 -
8.2130e+05
No. Observations:              242309   AIC:
1.643e+06
Df Residuals:                  242307   BIC:
1.643e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
```

```
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
const         11.4316      0.017    676.938      0.000      11.399
11.465
Parks         -0.0265      0.000    -99.269      0.000      -0.027
-0.026
==============================================================================
==========
Omnibus:                    15578.451   Durbin-Watson:
0.411
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
18756.333
Skew:                           0.676   Prob(JB):
0.00
Kurtosis:                       3.178   Cond. No.
73.4
==============================================================================
==========

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:
0.357
Model:                            OLS   Adj. R-squared:
0.357
Method:                 Least Squares   F-statistic:
1.345e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:            -
1.0092e+06
No. Observations:              242309   AIC:
2.018e+06
Df Residuals:                  242307   BIC:
2.018e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------
const              -15.5766      0.054   -287.195      0.000     -1
5.683     -15.470
Transit_stations     0.4887      0.001    366.727      0.000
0.486      0.491
==============================================================================
==========
Omnibus:                     1697.591   Durbin-Watson:
0.749
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
2363.961
Skew:                          -0.089   Prob(JB):
0.00
Kurtosis:                       3.450   Cond. No.
69.8
====================================================================
==========

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:
0.466
Model:                            OLS   Adj. R-squared:
0.466
Method:                 Least Squares   F-statistic:
2.117e+05
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:              -
7.5006e+05
No. Observations:              242309   AIC:
1.500e+06
Df Residuals:                  242307   BIC:
1.500e+06
Df Model:                           1
Covariance Type:            nonrobust
====================================================================
================
                       coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------
----------------
const                3.6284      0.019    194.897      0.000
3.592       3.665
Transit_stations    -0.2104      0.000   -460.087      0.000      -
0.211      -0.210
====================================================================
==========
Omnibus:                    11305.604   Durbin-Watson:
0.532
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
14541.501
Skew:                           0.480   Prob(JB):
0.00
Kurtosis:                       3.719   Cond. No.
69.8
====================================================================
==========

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:
0.806
Model:                            OLS   Adj. R-squared:
0.806
Method:                 Least Squares   F-statistic:
1.004e+06
Date:                Fri, 30 Sep 2022   Prob (F-statistic):
0.00
Time:                        05:06:20   Log-Likelihood:                 -
6.2766e+05
No. Observations:              242309   AIC:
1.255e+06
Df Residuals:                  242307   BIC:
1.255e+06
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================
==========
                 coef    std err          t      P>|t|      [0.025
0.975]
---------------------------------------------------------------------
----------
const         -0.1460      0.013    -11.631      0.000      -0.171
-0.121
Workplaces    -0.3382      0.000  -1002.241      0.000      -0.339
-0.338
=====================================================================
==========
Omnibus:                     1024.582   Durbin-Watson:
0.467
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
1098.269
Skew:                          -0.132   Prob(JB):
3.27e-239
Kurtosis:                       3.199   Cond. No.
71.3
=====================================================================
==========

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

after lockdown improvements

In [126]:

```
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")

]
```

In [127]:

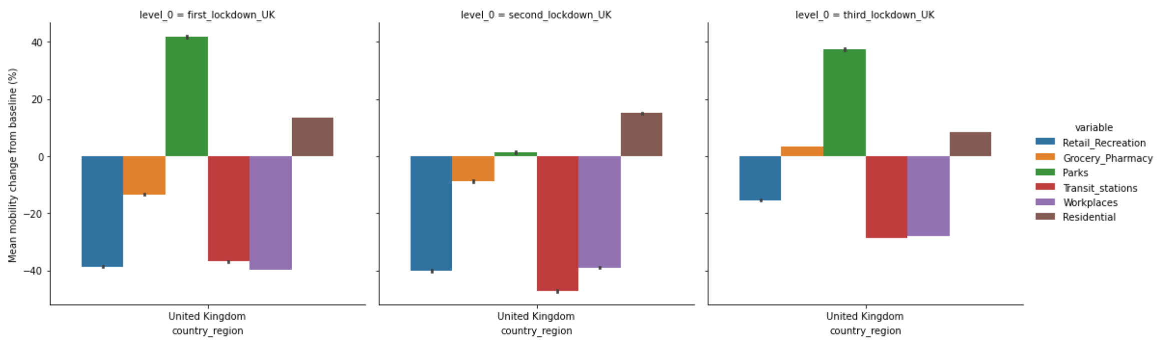
```
lockdowns_dataframes = [first_lockdown_UK, second_lockdown_UK, third_lockdown_UK
]
three_lockdowns_UK = pd.concat(
    lockdowns_dataframes,
    keys=["first_lockdown_UK", "second_lockdown_UK", "third_lockdown_UK"],
).reset_index()
```

In [128]:

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

Out[128]:

```
<seaborn.axisgrid.FacetGrid at 0x7fbdd4f9da10>
```



In [129]:

```
third_lockdown_UK_mean = (
    third_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
third_lockdown_UK_mean
```

Out[129]:

	variable	sub_region_1	value
0	Grocery_Pharmacy	Aberdeen City	4.121281
1	Grocery_Pharmacy	Aberdeenshire	0.391304
2	Grocery_Pharmacy	Angus Council	3.462243
3	Grocery_Pharmacy	Antrim and Newtownabbey	4.901602
4	Grocery_Pharmacy	Ards and North Down	16.707094
...	...	...	...
876	Workplaces	Windsor and Maidenhead	-37.267735
877	Workplaces	Wokingham	-37.970252
878	Workplaces	Worcestershire	-27.209546
879	Workplaces	Wrexham Principal Area	-24.196796
880	Workplaces	York	-34.645309

881 rows × 3 columns

In [130]:

```
third_lockdown_UK_mean_sorted = third_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
third_lockdown_UK_mean_sorted
```

Out[130]:

	sub_region_1	variable	value
741	Blackpool	Workplaces	-15.155606
796	Isle of Wight	Workplaces	-15.894737
802	Lincolnshire	Workplaces	-16.794622
845	Shropshire	Workplaces	-17.109840
810	Middlesbrough	Workplaces	-17.812357
...	...	...	...
56	Greater London	Grocery_Pharmacy	-6.331606
50	Falkirk	Grocery_Pharmacy	-6.496568
143	Wokingham	Grocery_Pharmacy	-6.510297
21	Cardiff	Grocery_Pharmacy	-7.908467
112	Slough	Grocery_Pharmacy	-15.302059

881 rows × 3 columns

In [131]:

```
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is Cornwall with value of 6.4576
659038901605
max change is for Retail_Recreation is Nottingham with value of -34.
87185354691076


min change is for Grocery_Pharmacy is Pembrokeshire with value of 2
8.94724770642202
max change is for Grocery_Pharmacy is Slough with value of -15.30205
9496567505


min change is for Parks is Cornwall with value of 129.50114416475972
max change is for Parks is West Dunbartonshire Council with value of
-56.0


min change is for Transit_stations is Argyll and Bute Council with v
alue of 17.217391304347824
max change is for Transit_stations is Clackmannanshire with value of
-53.08571428571429


min change is for Workplaces is Blackpool with value of -15.15560640
7322654
max change is for Workplaces is Cardiff with value of -41.0366132723
11214


min change is for Residential is East Renfrewshire Council with valu
e of 13.047923322683706
max change is for Residential is Fermanagh and Omagh with value of
4.432
```

In [132]:

```
third_lockdown_UK_descriptive_stats = (
    third_lockdown_UK.groupby(["sub_region_1", "variable"])["value"]
    .agg([min, max, np.mean, np.median, np.std])
    .reset_index()
)
third_lockdown_UK_descriptive_stats
```

Out[132]:

	sub_region_1	variable	min	max	mean	median	std
0	Aberdeen City	Grocery_Pharmacy	-89.0	48.0	4.121281	6.0	11.868022
1	Aberdeen City	Parks	-51.0	112.0	12.828375	8.0	30.650072
2	Aberdeen City	Residential	-1.0	24.0	10.084668	10.0	5.463781
3	Aberdeen City	Retail_Recreation	-93.0	-1.0	-29.384439	-22.0	19.662602
4	Aberdeen City	Transit_stations	-86.0	-18.0	-36.562929	-33.0	11.356960
...	...	...	...	...	...	...	...
876	York	Parks	-77.0	235.0	31.549425	24.0	49.753301
877	York	Residential	-1.0	23.0	9.187643	9.0	5.514496
878	York	Retail_Recreation	-92.0	33.0	-15.283753	-10.0	25.313757
879	York	Transit_stations	-92.0	43.0	-28.382151	-25.0	22.297570
880	York	Workplaces	-78.0	-2.0	-34.645309	-36.0	15.206682

881 rows × 7 columns

In [133]:

```
second_lockdown_UK_mean = (
    second_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
```

In [134]:

```
second_lockdown_UK_mean_sorted = second_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
```

In [135]:

```
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is Clackmannanshire with value o
f -22.2
max change is for Retail_Recreation is Nottingham with value of -58.
06976744186046


min change is for Grocery_Pharmacy is Neath Port Talbot Principle Ar
ea with value of 9.44186046511628
max change is for Grocery_Pharmacy is Slough with value of -26.90697
6744186046


min change is for Parks is Nottingham with value of 32.8604651162790
7
max change is for Parks is Inverclyde with value of -68.0


min change is for Transit_stations is Shropshire with value of -24.6
86046511627907
max change is for Transit_stations is Orkney with value of -71.0


min change is for Workplaces is Dumfries and Galloway with value of
-28.74418604651163
max change is for Workplaces is Edinburgh with value of -50.25581395
348837


min change is for Residential is Wokingham with value of 19.97674418
6046513
max change is for Residential is Dumfries and Galloway with value of
10.093023255813954
```

In [136]:

```
first_lockdown_UK_mean = (
    first_lockdown_UK.groupby(["variable", "sub_region_1"])["value"]
    .mean()
    .reset_index()
)
first_lockdown_UK_mean_sorted = first_lockdown_UK_mean.sort_values(
    by=[
        "variable",
        "value",
    ],
    ascending=False,
)[["sub_region_1", "variable", "value"]]
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_1} with value of {t_min.v
alue}')
    print(f'max change is for {i} is {t_max.sub_region_1} with value of {t_max.v
alue}')
    print('\n')
```

```
min change is for Retail_Recreation is Cornwall with value of -13.92
5373134328359
max change is for Retail_Recreation is Aberdeen City with value of -
56.62189054726368


min change is for Grocery_Pharmacy is Neath Port Talbot Principle Ar
ea with value of 3.368131868131868
max change is for Grocery_Pharmacy is Wokingham with value of -22.35
7142857142858


min change is for Parks is Cornwall with value of 118.33888888888889
max change is for Parks is West Dunbartonshire Council with value of
-63.25


min change is for Transit_stations is Argyll and Bute Council with v
alue of -8.126373626373626
max change is for Transit_stations is Reading with value of -64.7149
7584541063


min change is for Workplaces is North East Lincolnshire with value o
f -29.567164179104477
max change is for Workplaces is Edinburgh with value of -56.61835748
792271


min change is for Residential is Wokingham with value of 20.09493670
886076
max change is for Residential is Argyll and Bute Council with value
of 7.963414634146342
```

In [ ]: