Week 3: Identify Risk Factors for Infection

UPDATE
Thank you again for the previous analysis. We will next be publishing a public health advisory that warns of specific infection risk factors of which individuals should be aware. Please advise as to which population characteristics are associated with higher infection rates.

Your goal for this notebook will be to identify key potential demographic and economic risk factors for infection by comparing the infected and uninfected populations.

Imports

In [12]:
```
import cudf
import cuml
```

Load Data

Begin by loading the data you've received about week 3 of the outbreak into a cuDF data frame. The data is located at `./data/week3.csv`. For this notebook you will need all columns of the data.

In [13]:
```
# Load all columns from the week 3 outbreak data into a cuDF DataFrame
week3_df = cudf.read_csv('./data/week3.csv')
```

Calculate Infection Rates by Employment Code

Convert the `infected` column to type `float32`. For people who are not infected, the float32 `infected` value should be `0.0`, and for infected people it should be `1.0`.

In [14]:
```
# Convert the 'infected' column to float32, setting values based on infection statu
week3_df['infected'] = week3_df['infected'].astype('float32').replace({0: 0.0, 1: 1
```

Now, produce a list of employment types and their associated **rates** of infection, sorted from highest to lowest rate of infection.

NOTE: The infection **rate** for each employment type should be the percentage of total individuals within an employment type who are infected. Therefore, if employment type "X" has 1000 people, and 10 of them are infected, the infection **rate** would be .01. If employment type "Z" has 10,000 people, and 50 of them are infected, the infection rate

would be .005, and would be **lower** than for type "X", even though more people within that employment type were infected.

In [15]:
```
# Group by employment type and calculate the infection rate
infection_rates = (
    week3_df.groupby('employment')
    .agg({
        'infected': ['count', 'sum']  # Aggregate functions for 'infected' column
    })
)

# Rename columns for clarity
infection_rates.columns = ['total_individuals', 'infected_count']

# Calculate infection rate as the percentage of infected individuals
infection_rates['infection_rate'] = infection_rates['infected_count'] / infection_r

# Sort by infection rate from highest to lowest
sorted_infection_rates = infection_rates.sort_values(by='infection_rate', ascending

# Reset index for better readability
sorted_infection_rates = sorted_infection_rates.reset_index()

# Display the result
print(sorted_infection_rates)
```

	employment	total_individuals	infected_count	infection_rate
0	Q	3802602	48505.0	0.012756
1	I	1556575	16116.0	0.010354
2	V	10098466	76648.0	0.007590
3	P	3006149	18609.0	0.006190
4	Z	7161907	40498.0	0.005655
5	R, S, T	1669197	8997.0	0.005390
6	O	1843446	9741.0	0.005284
7	L	346470	1722.0	0.004970
8	G	3549465	17561.0	0.004948
9	N	1367137	6541.0	0.004784
10	M	2214336	10578.0	0.004777
11	K	1122406	5356.0	0.004772
12	X	181988	826.0	0.004539
13	J	1180372	4650.0	0.003939
14	C	2653753	10301.0	0.003882
15	A	305755	1178.0	0.003853
16	B, D, E	486785	1837.0	0.003774
17	H	1398342	4737.0	0.003388
18	F	2075628	6604.0	0.003182
19	U	12459115	2702.0	0.000217

Finally, read in the employment codes guide from `./data/code_guide.csv` to interpret which employment types are seeing the highest rates of infection.

In [16]:
```
code_guide_df = cudf.read_csv('./data/code_guide.csv')
```

```
# Display the contents of the DataFrame
print(code_guide_df)
```

```
      Code                                              Field
0        A                    Agriculture, forestry & fishing
1   B, D, E                    Mining, energy and water supply
2        C                                      Manufacturing
3        F                                       Construction
4        G     Wholesale, retail & repair of motor vehicles
5        H                                Transport & storage
6        I                     Accommodation & food services
7        J                       Information & communication
8        K                  Financial & insurance activities
9        L                             Real estate activities
10       M  Professional, scientific & technical activities
11       N                 Administrative & support services
12       O          Public admin & defence; social security
13       P                                       Education
14       Q            Human health & social work activities
15  R, S, T                                  Other services
16       U                                         Student
17       V                                         Retired
18       X                  Outside the UK or not specified
19       Y                               Pre-school child
20       Z                             Not formally employed
```

Calculate Infection Rates by Employment Code and Sex

We want to see if there is an effect of `sex` on infection rate, either in addition to `employment` or confounding it. Group by both `employment` and `sex` simultaneously to get the infection rate for the intersection of those categories.

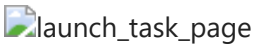
In [17]:
```
# Group by both 'employment' and 'sex' and calculate the infection rates
infection_rates_sex = (
    week3_df.groupby(['employment', 'sex'])
    .agg({
        'infected': ['count', 'sum']  # Count total individuals and infected indivi
    })
)

# Rename the columns for better readability
infection_rates_sex.columns = ['total_individuals', 'infected_count']

# Calculate the infection rate as the percentage of infected individuals
infection_rates_sex['infection_rate'] = infection_rates_sex['infected_count'] / inf

# Sort by infection rate from highest to lowest
infection_rates_sex = infection_rates_sex.sort_values(by='infection_rate', ascendin

# Reset the index for better readability
infection_rates_sex.reset_index(inplace=True)
```

```
# Display the resulting DataFrame
print(infection_rates_sex)
```

```
   employment sex  total_individuals  infected_count  infection_rate
0           I   f             819431         12344.0        0.015064
1           Q   f            2954725         44164.0        0.014947
2           V   f            5502736         59713.0        0.010852
3     B, D, E   f             115888           924.0        0.007973
4     R, S, T   f             913091          7075.0        0.007748
5           O   f             976302          7536.0        0.007719
6           K   f             494256          3792.0        0.007672
7           M   f             975240          7456.0        0.007645
8           J   f             322584          2466.0        0.007645
9           C   f             688196          5251.0        0.007630
10          Z   f            4382232         33430.0        0.007629
11          P   f            2169494         16453.0        0.007584
12          F   f             266193          2017.0        0.007577
13          G   f            1682827         12715.0        0.007556
14          A   f              82229           616.0        0.007491
15          X   f              77253           571.0        0.007391
16          N   f             633114          4678.0        0.007389
17          H   f             258211          1907.0        0.007385
18          L   f             191242          1381.0        0.007221
19          Q   m             847877          4341.0        0.005120
20          I   m             737144          3772.0        0.005117
21          V   m            4595730         16935.0        0.003685
22          G   m            1866638          4846.0        0.002596
23          P   m             836655          2156.0        0.002577
24          C   m            1965557          5050.0        0.002569
25          J   m             857788          2184.0        0.002546
26          O   m             867144          2205.0        0.002543
27          Z   m            2779675          7068.0        0.002543
28    R, S, T   m             756106          1922.0        0.002542
29          N   m             734023          1863.0        0.002538
30          F   m            1809435          4587.0        0.002535
31          M   m            1239096          3122.0        0.002520
32          A   m             223526           562.0        0.002514
33          K   m             628150          1564.0        0.002490
34          H   m            1140131          2830.0        0.002482
35    B, D, E   m             370897           913.0        0.002462
36          X   m             104735           255.0        0.002435
37          L   m             155228           341.0        0.002197
38          U   f            6073869          2001.0        0.000329
39          U   m            6385246           701.0        0.000110
```

Take the Assessment

After completing the work above, visit the *Launch Section* web page that you used to launch this Jupyter Lab. Scroll down below where you launched Jupyter Lab, and answer the question *Week 3 Assessment*. You can view your overall progress in the assessment by visiting the same *Launch Section* page and clicking on the link to the *Progress* page. On the *Progress* page, if you have successfully answered all the assessment questions, you can click on *Generate Certificate* to receive your certificate in the course.



Optional: Restart the Kernel

If you plan to continue work in other notebooks, please shutdown the kernel.

In [18]:
```
import IPython
app = IPython.Application.instance()
app.kernel.do_shutdown(True)
```

Out[18]:
```
{'status': 'ok', 'restart': True}
```