Final Report

January 6, 2024

1 COVID-19 and Commerce

2 Overview

Video Presentation

We are interested in answering how consumer trends were affected by pre- and post-COVID rates (measured in both COVID infection and COVID mortality rates). To answer this, we collected and organized commerce data from the US Census COVID-19 data from Our World In Data. Afterwards, we cleaned the data and honed in on specific variables to create graphs that visualized the relationships. In the end, our results showed that there were actually little to no correlations (indicated by low correlation coefficient) between COVID-19 and commerce trends with any of our intended variables. Ultimately, our findings contradicted our initial hypotheses that there is a significant relationship between the COVID-19 rates and consumer trends; however, we did identify a statistically significant positive relationship between COVID mortality rates and oil/gasoline consumership.

3 Names

- Siddhant Joshi
- Robin Edejer
- Judy Kim
- Allen Chen
- Sarah Yassein

Research Question

What can the state of the COVID-19 pandemic in the United States tell us about pre- and post-COVID-19 consumer trends for Americans?

Can the relationship between these two variables reveal something about the following industries: e-commerce, hobbies and recreation, electronics and appliances, health and personal care, and oil/gasoline?

3.1 Background & Prior Work

Background and Motivation:

The COVID-19 pandemic has had a profound global impact since its initial outbreak in early 2019. Subsequent lockdowns and travel restrictions implemented by countries worldwide greatly limited people's options for various aspects of daily life, including entertainment, socializing, and shopping. However, despite these hurdles, everyday lifestyle processes cannot simply come to a halt. Business psychologist Jagdish Sheth explains that since consumership is bound by location and time and the consumer can no longer go to the store, e-commerce emerged as a means of bringing the store to the consumer [4]. In response to the declining consumer activity caused by the pandemic, various industries have had to adapt and find new ways to navigate this unprecedented situation.

Given this trend, we wish to dive deep into analysis at the level of various industries to further explore this phenomenon. We will be examining time series data over the last decade from multiple industries within the retail sector as a proxy of the consumership of their American base to gain nuanced understanding of the significance in commerce trends. Additionally, with these seemingly significant changes in American consumer habits, we seek to discover underlying patterns that could offer insight into extrapolations of our economic future.

Current Literature:

Present research has explored various aspects of American commerce over the course of the pandemic. One study fits a regression model on the case and death rates across the country as predictors with Amazon's revenue statistics as the outcome [1]. Other psychology research reasons that general anticipation, whether fearful or hopeful, plays an undeniable role in consumer habits [5]. From 2020 - 2022, Americans spent 1.7 trillion USD, which was a 55% increase from the two years leading up to the pandemic [2]. Furthermore, approximately half of that spending was put toward groceries, clothing, and electronics [2][3]. U.S. Census data also report a general increase in consumership with a characteristic spike around 2020 [6][7]. However, some other studies conflict with those reported earlier, claiming a meager 3.5% increase in the proportion of online sales in America from 2019 to 2020 [3]. These staggering and contrasting studies explore different metrics to come to their evaluations of the state of commerce throughout the pandemic. We believe this is a perfect invitation to detailed discussion on the somewhat-agreed upon drivers of these trends, by testing these relationships with fitting linear regression models.

[1] Abdelrhim, Mansour and Elsayed, Abdalla, The Effect of COVID-19 Spread on the E-Commerce Market: The Case of the 5 Largest E-Commerce Companies in the World (June 7, 2020). Available at SSRN: https://ssrn.com/abstract=3621166 or http://dx.doi.org/10.2139/ssrn.3621166

[2] Koetsier, J. (2022, March 15). Pandemic Digital Spend: $1.7 Trillion. Forbes. Retrieved May 2, 2023, https://www.forbes.com/sites/johnkoetsier/2022/03/15/pandemic-digital-spend-17-trillion/?sh=11e9579e5035

[3] Sayyida, S., Hartini, S., Gunawan, S., & Husin, S. N. (2021). The Impact of the Covid-19 Pandemic on Retail Consumer Behavior. APTISI Transactions on Management (ATM), 5(1), 79–88. https://doi.org/10.33050/atm.v5i1.1497

[4] Sheth, Jagdish (2020), Impact of Covid-19 on consumer behavior: Will the old habits return or die?, Journal of Business Research, Volume 117, 2020, Pages 280-283, ISSN 0148-2963, https://doi.org/10.1016/j.jbusres.2020.05.059.

[5] Tasnim, R., Islam, M. S., Sujan, M. S. H., Sikder, M. T., Potenza, M. N., & Van Osch, L. (2020). Development and initial psychometric properties of the Panic Buying Scale during the COVID-19 pandemic. International Journal of Mental Health and Addiction, 1-14. doi: 10.1007/s11469-020-00316-3

[6] U.S. Census Bureau. (2022, April 14). E-commerce Sales Surged During Pandemic. Re-

trieved May 2, 2023, https://www.census.gov/library/stories/2022/04/ecommerce-sales-surged-during-pandemic.html

[7] U.S. Census Bureau. (n.d.). E-Commerce Retail Sales. Retrieved May 2, 2023, https://www.census.gov/retail/ecommerce.html

4 Hypothesis

Our team predicts that American consumer spending in the industries mentioned previously did alter as a consequence of the COVID-19 pandemic from 2019 - 2022 due to the lockdown and uncertainty in response. Since Americans were advised to limit their time outside their homes and in public spaces while exercising caution when interacting with others, we can expect to see a positive increase in e-commerce, electronics, recreation-related spending as well as in health and personal care products. In terms of a linear regression model, this can be represented in the following pair of statistical hypotheses. We plan to use a standard $\alpha = 0.05$ as our decision criterion for statistical significance.

Hypothesis 1: There is no statistically significant difference between consumer trends and the state of the pandemic. - $H_0 = 0$ - Model: Retail sales $= \beta_0$

Hypothesis 2: There is statistically significant difference between consumer trends and the state of the pandemic. - $H_a \neq 0$ - Model: Retail sales $= \beta_0 + \beta_1(new\ cases) + \beta_2(new\ deaths)$

5 Dataset(s)

Dataset Name: Estimated Monthly Sales for Retail and Food Services, by Kind of Business

Link to the dataset: https://www.census.gov/retail/sales.html

Number of observations: 10,200 rows

These datasets document the monthly food and retail sales in the U.S. While the sales are, we have decided to only include data from 2013 to 2023. The dataset also organizes those sales into various business categories, such as “electronics and appliance”, “oil/gasoline”, and “e-commerce”. These data would be useful for tracking consumer behavior over time from 2013 - 2022.

Dataset Name: Our World in Data: COVID-19 Data

Link to the dataset: https://github.com/owid/covid-19-data

Number of observations: 1080 rows condensed into 36 (monthly)

This dataset contains a daily documentation COVID-19 death and infection rates in the United States from Jan. 2019 to Jan. 2022. The data are organized on a daily basis and aggregated on a monthly basis. The data would be useful for capturing information on COVID-19 spread over time.

6 Setup

[1]:
```
# Import necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf

# Reading in all the datasets for the 10 year span of interest
frs2013 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2013.
↪csv')
frs2014 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2014.
↪csv')
frs2015 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2015.
↪csv')
frs2016 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2016.
↪csv')
frs2017 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2017.
↪csv')
frs2018 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2018.
↪csv')
frs2019 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2019.
↪csv')
frs2020 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2020.
↪csv')
frs2021 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2021.
↪csv')
frs2022 = pd.read_csv(r'C:
↪\Users\axjos\Downloads\drive-download-20230601T055937Z-001\foodretailsales2022.
↪csv')

# Reading in COVID-19 dataset
covid = pd.read_csv(r'C:\Users\axjos\OneDrive\Desktop\monthly-covid.csv')

# Creating list of datasets
```

```
data_list = [frs2013, frs2014, frs2015, frs2016, frs2017, frs2018, frs2019,␣
↪frs2020, frs2021, frs2022]
```

7 Data Cleaning

7.0.1 Defining Helper Function for Data Cleaning

[2]:
```
# Helper functions for data cleaning

# Function to convert numbers into float-types and stripping commas
def standardize_num(str_num):
    try:
        if ',' in str_num:
            str_num = str_num.replace(',', '')
            output = float(str_num)
        elif 'S' in str_num:
            output = np.nan
        else:
            output = float(str_num)
    except:
        output = float(str_num)

    return output
```

7.0.2 Conducting the Data Cleaning

Some of the priliminary cleaning steps outlined below include: - Renaming columns to month format - Dropping whitespace rows and columns - Cleaning out NaN values by dropping rows that contain any - Standardizing the numerical values such that they are all floats - Resetting indices after the above alterations

[3]:
```
for i, df in enumerate(data_list):

    year = str(2013 + i)
    # Renaming columns (120 x 16)
    df = df.set_axis(['TOSS','Business','Jan', 'Feb', 'Mar',
                      'Apr', 'May', 'Jun','Jul', 'Aug', 'Sep',
                      'Oct', 'Nov', 'Dec', 'Total', 'TOSS'], axis = 'columns')

    # Dropping first and last columns since those had no important/useful␣
↪information (120 x 14)
    df = df.drop(labels = 'TOSS', axis = 1)

    # Drops spacer rows at the top of the data frame
    df = df.drop(df.index[0:5])

    # Drops rows where all values are NaN
```

```
    df = df.dropna(axis = 0, how = 'all')

    # Resetting the index to start from 0 and deleting the extra column that␣
↪gets made in the process
    df = df.reset_index(drop = True)

    # Change all dollar values to floats with standardize function
    for col in df.columns:
        if col != 'Business':
            df[col] = df[col].apply(standardize_num)

    # Re-assigning cleaned data frome to list element
    data_list[i] = df

# Redefining data frame variables with cleaned data
frs2013 = data_list[0]
frs2014 = data_list[1]
frs2015 = data_list[2]
frs2016 = data_list[3]
frs2017 = data_list[4]
frs2018 = data_list[5]
frs2019 = data_list[6]
frs2020 = data_list[7]
frs2021 = data_list[8]
frs2022 = data_list[9]

frs2020.head()
```

[3]:

```
                                            Business       Jan       Feb  \
0              Retail and food services sales, total  480837.0  479584.0
1  Retail sales and food services excl motor vehi…  389176.0  383785.0
2  Retail sales and food services excl gasoline s…  441317.0  442772.0
3  Retail sales and food services excl motor vehi…  349656.0  346973.0
4                                Retail sales, total  417123.0  413923.0

        Mar       Apr       May       Jun       Jul       Aug       Sep  \
0  476288.0  408280.0  505991.0  533784.0  549955.0  546744.0  531870.0
1  395892.0  340622.0  401654.0  422852.0  437223.0  434515.0  422554.0
2  442180.0  381601.0  474055.0  497621.0  511283.0  508286.0  494981.0
3  361784.0  313943.0  369718.0  386689.0  398551.0  396057.0  385665.0
4  426258.0  376867.0  461785.0  479784.0  492689.0  487325.0  474038.0

        Oct       Nov       Dec      Total
0  554129.0  544421.0  611417.0  6223300.0
1  443179.0  444677.0  498976.0  5015105.0
2  516401.0  510532.0  575413.0  5796442.0
3  405451.0  410788.0  462972.0  4588247.0
```

```
4  493601.0  490941.0  557696.0  5572030.0
```

Now, we need to rename and aggregate rows into relevant categories

[4]:
```
# For loop for some extra cleaning, such as renaming and aggregating rows
for c, df in enumerate(data_list):
    # Renaming rows that do not need to be aggregated
    df = df.rename(index = {0:'Retail and Food total',
                                      4: 'Total retail sales',
                                      7: 'Motor vehicle and parts dealers',
                                      19: 'Electronics and appliance stores',
                                      26: 'Food and beverage stores',
                                      32: 'Gasoline stations',
                                      33: 'Clothing and accessories stores',
                                      41: 'Hobbies and recreation',
                                      45: 'General merchandise stores',
                                      53: 'Office supplies and gift stores',
                                      57: 'E-commerce and other non-store␣
 ↪retailers',
                                      59: 'Oil/Gasoline',
                                      60: 'Food services and drinking places'})
        # Aggregating the remaining 4 categories
    df.loc['Furniture and interior furnishing'] = df.loc[[14, 18]].sum()
    df.loc['Home Improvement'] = df.loc[[17, 22, 24, 25]].sum()
    df.loc['Health and personal care'] = df.loc[[30, 31]].sum()
    df.loc['Misc. store retailers'] = df.loc[[52, 56]].sum()

    # Deleting all remaining redundant rows (17 x 14)
    for i, j in df.iterrows():
        if type(i) != str:
            df.drop(i, inplace = True)

    # Resetting index and replacing old 'Business' column with index values
    df = df.reset_index()
    df = df.drop(['Business'], axis = 1)
    df.rename(columns = {'index' : 'Business'}, inplace = True)

    # Re-assigning cleaned data frome to list element
    data_list[c] = df


# Redefining data frame variables with cleaned data
frs2013 = data_list[0]
frs2014 = data_list[1]
frs2015 = data_list[2]
frs2016 = data_list[3]
frs2017 = data_list[4]
```

```
frs2018 = data_list[5]
frs2019 = data_list[6]
frs2020 = data_list[7]
frs2021 = data_list[8]
frs2022 = data_list[9]

frs2020.head()
```

8 Data Analysis & Results (EDA)

Carry out EDA on your dataset(s); Describe in this section.

[6]:
```
covid.describe()
```

[6]:

	new_cases	new_deaths
count	3.600000e+01	36.000000
mean	2.761436e+06	30135.916667
std	3.456615e+06	23853.680062
min	8.000000e+00	0.000000
25%	1.099556e+06	12550.000000
50%	1.845286e+06	23186.500000
75%	3.280654e+06	43478.750000
max	2.044476e+07	100119.000000

[7]:
```
frs2013.describe()
```

[7]:

	Jan	Feb	Mar	Apr	\
count	17.000000	17.000000	17.000000	17.000000	
mean	65796.764706	65404.647059	73569.352941	71272.000000	
std	110720.654987	110403.597346	124471.438288	120653.112337	
min	2498.000000	2422.000000	2532.000000	2465.000000	
25%	9254.000000	8935.000000	9899.000000	9274.000000	
50%	36423.000000	33101.000000	34765.000000	33682.000000	
75%	46649.000000	48549.000000	54255.000000	50126.000000	
max	375502.000000	373991.000000	421747.000000	408554.000000	

	May	Jun	Jul	Aug	\
count	17.000000	17.000000	17.000000	17.000000	
mean	76256.941176	72293.294118	73741.705882	75946.823529	
std	129326.624145	122699.793423	125104.727189	128804.002150	
min	2199.000000	1784.000000	1900.000000	1976.000000	
25%	9939.000000	9537.000000	10127.000000	10698.000000	
50%	37016.000000	31877.000000	32804.000000	33850.000000	
75%	54974.000000	53259.000000	52101.000000	55115.000000	
max	437198.000000	414798.000000	422495.000000	435164.000000	

```
                 Sep            Oct            Nov            Dec  \
count      17.000000      17.000000      17.000000      17.000000
mean    69211.705882   72690.352941   74142.941176   83759.882353
std    117062.675592  122726.223266  125268.409351  141493.108397
min      2234.000000    3016.000000    2713.000000    3324.000000
25%      9881.000000   10235.000000   10791.000000   12766.000000
50%     32656.000000   35201.000000   39676.000000   43165.000000
75%     50066.000000   53106.000000   54757.000000   58193.000000
max    396363.000000  415882.000000  424012.000000  477212.000000

              Total
count  1.700000e+01
mean   8.740864e+05
std    1.478423e+06
min    3.309300e+04
25%    1.221720e+05
50%    4.299880e+05
75%    6.414920e+05
max    5.002918e+06
```

8.0.1 Defining helper functions for EDA

[8]:
```python
# Normalizes the values of the dataframe that gets passed into it
def normalize_frs(df):
    temp = df.copy()
    temp = temp.drop(['Business', 'Total'], axis = 1)
    temp = temp.div(temp.sum(axis = 1), axis = 0)
    cols = ['Retail and Food total', 'Total retail sales', 'Motor vehicle and␣
 ↪parts dealers',
            'Electronics and appliance stores', 'Food and beverage stores',␣
 ↪'Gasoline stations',
            'Clothing and accessories stores', 'Hobbies and recreation',␣
 ↪'General merchandise stores',
            'Office supplies and gift stores', 'E-commerce and other non-store␣
 ↪retailers',
            'Oil/Gasoline', 'Food services and drinking places', 'Furniture and␣
 ↪interior furnishing',
            'Home Improvement', 'Health and personal care', 'Misc. store␣
 ↪retailers']
    temp.insert(0, 'Business', cols)
    return temp

def normalize_total(df):
    temp = df.copy()
    temp = temp.drop(['Year'], axis = 1)
    temp = temp.div(temp.sum(axis = 0), axis = 1)
```

```python
    cols = ['2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020',␣
 ↪'2021', '2022']
    temp.insert(0, 'Year', cols)
    return temp

def normalize_covid(df):
    temp = df.copy()
    temp = temp.drop(['date'], axis = 1)
    temp = temp.div(temp.sum(axis = 0), axis = 1)
    temp.reset_index(inplace = True)
    temp['index'] += 13
    temp = temp.rename(columns = {'index' : 'Months since Jan. 2019'})
    return temp

# Transforms a list of dataframes into graph-friendly, month-by-month format␣
 ↪and concatenates them together
def month_by_month_concat(df_list):
    monthly_dfs = []

    # Loops through each df in df_list, transforms it into graph-friendly␣
 ↪format, and appends it to monthly_dfs
    for df in df_list:
        temp = pd.DataFrame()
        temp['Business'] = df['Business']

        col_list = df.columns.to_list()

        for i, col in enumerate(col_list):
            if i != 0 and i != 13:
                temp[col] = df[col]

        temp.set_index('Business', inplace = True)
        temp = temp.T
        temp = temp.rename_axis(None, axis = 1)
        temp = temp.reset_index()
        temp.rename(columns = {'index' : 'month'}, inplace = True)
        monthly_dfs.append(temp)

    # Concatenates the data frames in monthly_dfs
    concat_data = pd.concat(monthly_dfs)

    return concat_data
```

8.0.2 Creating a summary dataframe of annual totals across industries

[9]:
```python
# Creating dataframe of annual totals per industry
totals = pd.DataFrame()
totals['Business'] = frs2013['Business']
for i, df in enumerate(data_list):
    totals[str(2013 + i)] = df['Total']

# Transforming 'totals' into a graphing-friendly format
totals.set_index('Business', inplace = True)
totals = totals.T
totals = totals.rename_axis(None, axis = 1)
totals = totals.reset_index()
totals.rename(columns = {'index' : 'Year'}, inplace = True)
totals.head()
```

[9]:
```
   Year  Retail and Food total  Total retail sales  \
0  2013              5002918.0           4459183.0
1  2014              5217688.0           4640561.0
2  2015              5350925.0           4725985.0
3  2016              5506177.0           4848422.0
4  2017              5732863.0           5040214.0

   Motor vehicle and parts dealers  Electronics and appliance stores  \
0                         962182.0                          103515.0
1                        1027015.0                          102851.0
2                        1103965.0                          100186.0
3                        1152379.0                           93967.0
4                        1185696.0                           92424.0

   Food and beverage stores  Gasoline stations  \
0                  641492.0           553708.0
1                  670428.0           546956.0
2                  687304.0           455594.0
3                  702045.0           435917.0
4                  728843.0           476152.0

   Clothing and accessories stores  Hobbies and recreation  \
0                         244950.0                 82752.0
1                         250888.0                 83591.0
2                         256583.0                 85471.0
3                         260684.0                 86309.0
4                         261396.0                 83896.0

   General merchandise stores  Office supplies and gift stores  \
0                    653688.0                          33093.0
1                    670849.0                          32682.0
```

```
2                    680499.0                          31962.0
3                    683202.0                          31115.0
4                    691891.0                          30061.0

   E-commerce and other non-store retailers  Oil/Gasoline  \
0                                  429988.0       38655.0
1                                  464194.0       41666.0
2                                  498375.0       32850.0
3                                  544573.0       27588.0
4                                  608194.0       31433.0

   Food services and drinking places  Furniture and interior furnishing  \
0                           543735.0                           122172.0
1                           577127.0                           127774.0
2                           624940.0                           136458.0
3                           657755.0                           141780.0
4                           692649.0                           145185.0

   Home Improvement  Health and personal care  Misc. store retailers
0          347161.0                  513016.0               127261.0
1          364878.0                  540859.0               130935.0
2          377976.0                  564706.0               134612.0
3          394166.0                  583608.0               136985.0
4          386792.0                  592920.0               139921.0
```

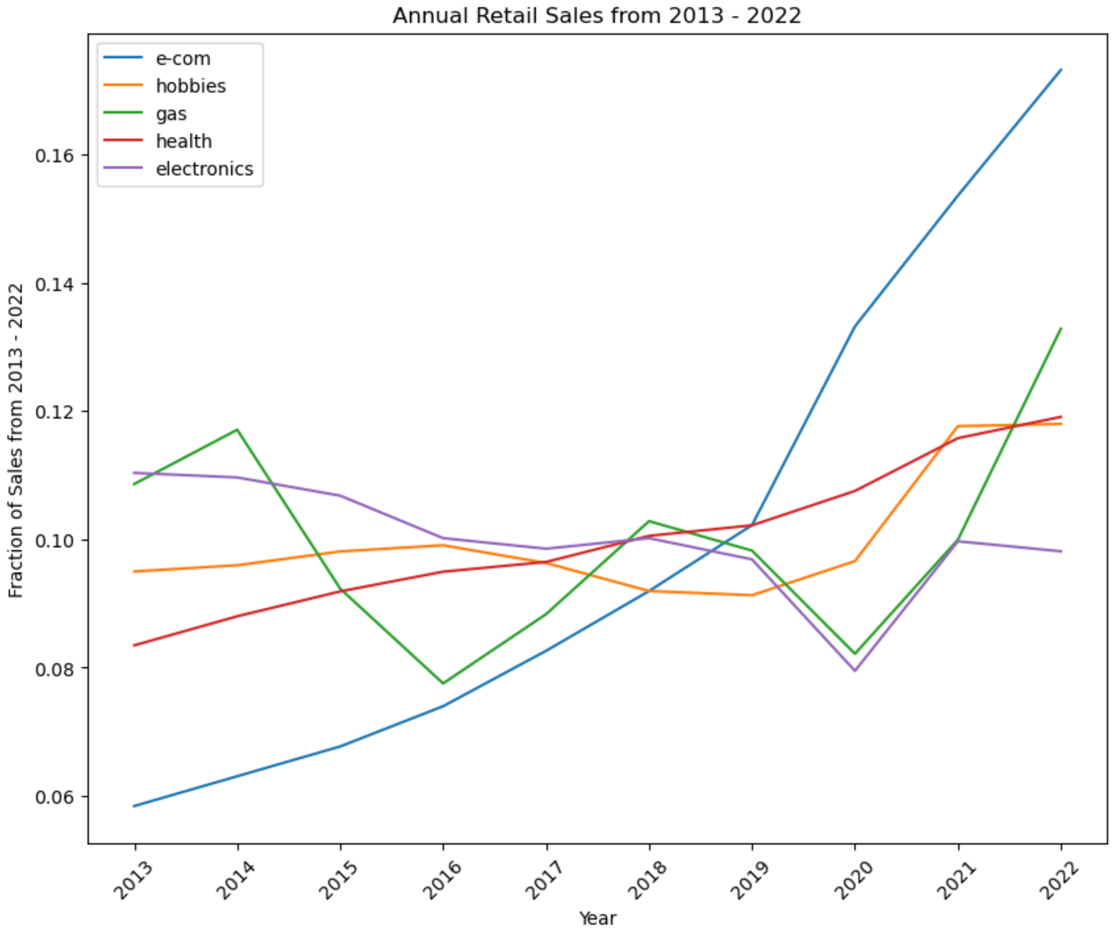
Normalizing the data and creating some initial plots to map retail sales trends over the interval of interest (2013 - 2022)

[15]:
```python
# Normalization
total_norm = normalize_total(totals)

# Plotting variables of interest
fig = plt.figure(figsize = (10, 8))
ax = plt.axes()
ax1 = sns.lineplot(data = total_norm, x = 'Year', y = 'E-commerce and other␣
 ↪non-store retailers', label = 'e-com')
ax2 = sns.lineplot(data = total_norm, x = 'Year', y = 'Hobbies and recreation',␣
 ↪label = 'hobbies')
ax3 = sns.lineplot(data = total_norm, x = 'Year', y = 'Oil/Gasoline', label =␣
 ↪'gas')
ax4 = sns.lineplot(data = total_norm, x = 'Year', y = 'Health and personal␣
 ↪care', label = 'health')
ax5 = sns.lineplot(data = total_norm, x = 'Year', y = 'Electronics and␣
 ↪appliance stores', label = 'electronics')

plt.xlabel('Year')
plt.ylabel('Fraction of Sales from 2013 - 2022')
```

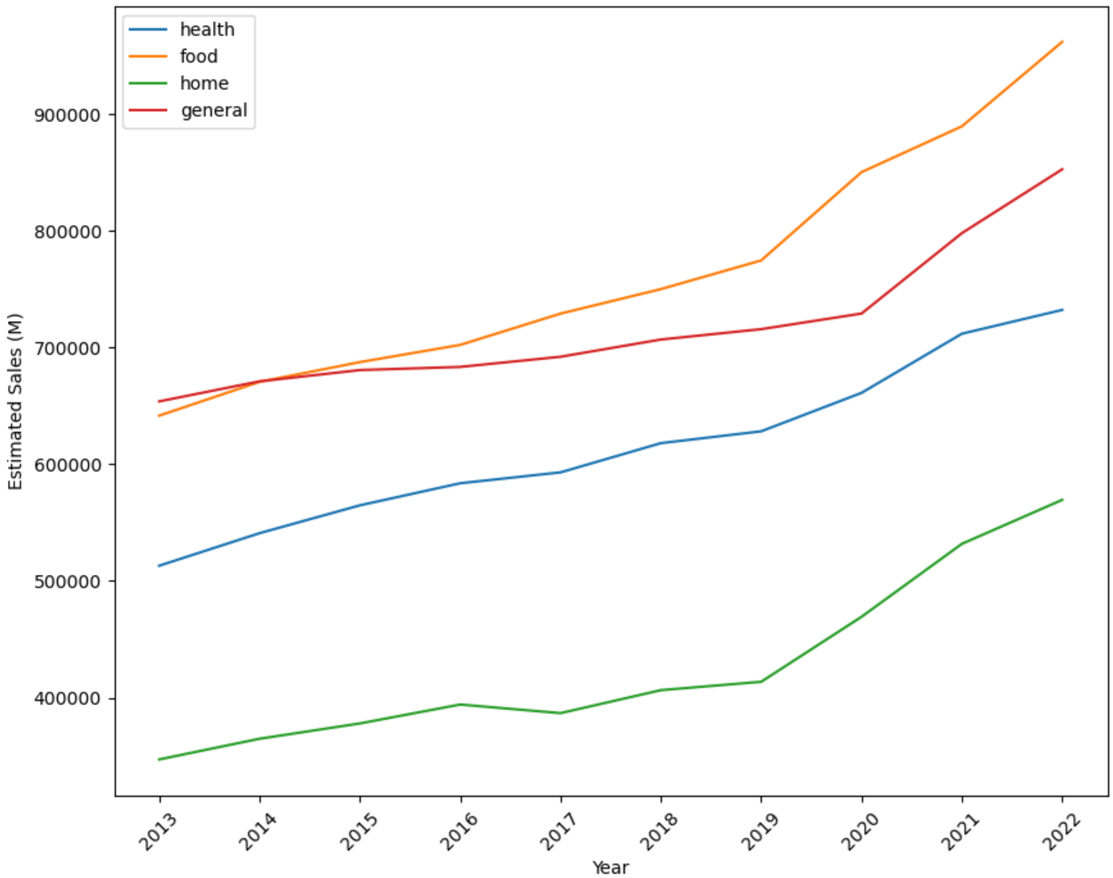
```
plt.title('Annual Retail Sales from 2013 - 2022')
plt.xticks(rotation = 45)
plt.show()
```



Some more plots! These four industries had steady increases, even into the pandemic (2020)

[11]:
```
fig = plt.figure(figsize = (10, 8))
ax = plt.axes()
ax1 = sns.lineplot(data = totals, x = 'Year', y = 'Health and personal care',␣
  ↪label = 'health')
ax2 = sns.lineplot(data = totals, x = 'Year', y = 'Food and beverage stores',␣
  ↪label = 'food')
ax3 = sns.lineplot(data = totals, x = 'Year', y = 'Home Improvement', label =␣
  ↪'home')
ax4 = sns.lineplot(data = totals, x = 'Year', y = 'General merchandise stores',␣
  ↪label = 'general')
```

```
plt.ylabel('Estimated Sales (M)')
plt.xticks(rotation = 45)
plt.legend()
plt.show()
```



8.0.3 Observations

After plotting and visualizing each of these industries, we notice the largest changes in the following industries:

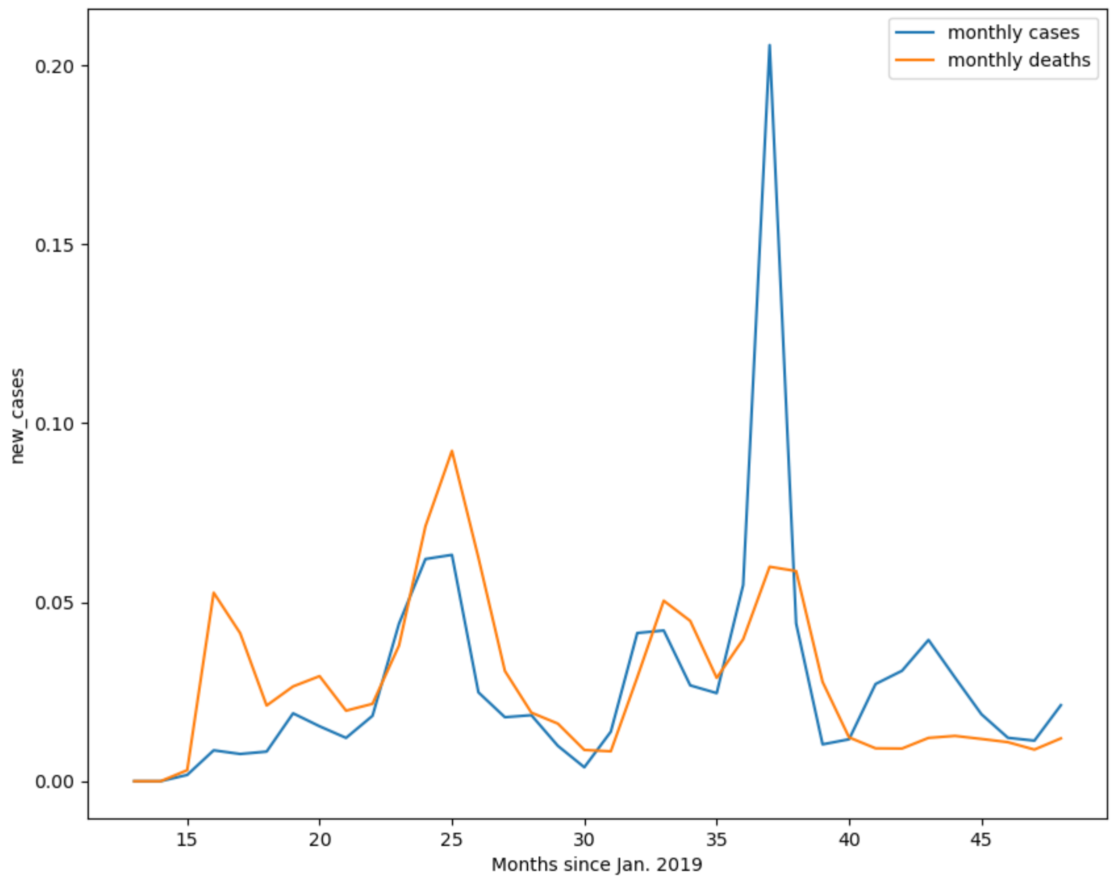
- Hobbies and recreation spiked upward in 2020
- E-commerce spiked upward in 2019
- Electronics and appliance stores dropped in 2019
- Gas dipped in 2019 and then rose back up in 2020
- Health has a slight increase in 2020

Many of these sudden changes in the regular trend of sales seem to occur from 2019 - 2022. To explore this further, we isolated those years and broke them down month-by-month. Additionally, we introduced a COVID-19 infections and deaths data set containing information on the number of monthly infections from Jan. 2020 - Dec. 2022. By plotting our data against the COVID-19 data, we hope to explore some correlations between the two events.

8.0.4 Preparing COVID-19 Infection Count Data

[12]:
```
# Prepare COVID-19 data
covid_norm = normalize_covid(covid)

# Plot the data
fig = plt.figure(figsize = (10, 8))
ax = plt.axes()
ax1 = sns.lineplot(data = covid_norm, x = 'Months since Jan. 2019', y =␣
↪'new_cases', label = 'monthly cases')
ax1 = sns.lineplot(data = covid_norm, x = 'Months since Jan. 2019', y =␣
↪'new_deaths', label = 'monthly deaths')
```



8.0.5 Concatenating month-by-month data of retail sales from 2019 - 2022

[16]:
```
# Normalizing each year's dataframe
norm_list = []
for df in [frs2019, frs2020, frs2021, frs2022]:
    norm_list.append(normalize_frs(df))
```

[17]:
```
# Concatenating normalized dataframes for 2019 - 2022
concat_19_22 = month_by_month_concat(norm_list)
concat_19_22 = concat_19_22.reset_index(drop = True)
concat_19_22 = concat_19_22.rename_axis('Months since Jan 2019', axis = 1)
concat_19_22
```

[17]:
```
Months since Jan 2019 month  Retail and Food total  Total retail sales  \
0                       Jan               0.073798            0.073895
1                       Feb               0.071462            0.071201
2                       Mar               0.083017            0.082695
3                       Apr               0.081877            0.081881
4                       May               0.087936            0.088012
5                       Jun               0.083267            0.082937
6                       Jul               0.085472            0.085426
7                       Aug               0.087674            0.087558
8                       Sep               0.079750            0.079354
9                       Oct               0.084507            0.084302
10                      Nov               0.086146            0.086571
11                      Dec               0.095094            0.096168
12                      Jan               0.077264            0.074860
13                      Feb               0.077063            0.074286
14                      Mar               0.076533            0.076500
15                      Apr               0.065605            0.067635
16                      May               0.081306            0.082876
17                      Jun               0.085772            0.086106
18                      Jul               0.088370            0.088422
19                      Aug               0.087854            0.087459
20                      Sep               0.085464            0.085075
21                      Oct               0.089041            0.088585
22                      Nov               0.087481            0.088108
23                      Dec               0.098246            0.100088
24                      Jan               0.070067            0.070724
25                      Feb               0.066446            0.067002
26                      Mar               0.085208            0.085889
27                      Apr               0.084677            0.084977
28                      May               0.086869            0.086649
29                      Jun               0.085822            0.085464
30                      Jul               0.085631            0.084656
31                      Aug               0.085041            0.084348
32                      Sep               0.081707            0.081123
```

33	Oct	0.085500	0.084872
34	Nov	0.087697	0.088152
35	Dec	0.095336	0.096145
36	Jan	0.072329	0.072857
37	Feb	0.071531	0.071585
38	Mar	0.084187	0.084361
39	Apr	0.084188	0.084100
40	May	0.087073	0.086910
41	Jun	0.085935	0.085939
42	Jul	0.084932	0.084556
43	Aug	0.086586	0.086470
44	Sep	0.081870	0.081461
45	Oct	0.084649	0.084167
46	Nov	0.084846	0.085300
47	Dec	0.091875	0.092294

Months since Jan 2019	Motor vehicle and parts dealers \
0	0.071268
1	0.072725
2	0.089418
3	0.083748
4	0.090520
5	0.084657
6	0.088249
7	0.092429
8	0.080099
9	0.083277
10	0.080971
11	0.082639
12	0.075866
13	0.079291
14	0.066542
15	0.055999
16	0.086358
17	0.091816
18	0.093306
19	0.092890
20	0.090479
21	0.091831
22	0.082556
23	0.093065
24	0.067821
25	0.066943
26	0.095764
27	0.093904
28	0.092938
29	0.089373

```
30                                          0.087231
31                                          0.083545
32                                          0.080354
33                                          0.081210
34                                          0.078252
35                                          0.082666
36                                          0.074544
37                                          0.076785
38                                          0.091782
39                                          0.089836
40                                          0.086357
41                                          0.086562
42                                          0.083377
43                                          0.088822
44                                          0.081916
45                                          0.082828
46                                          0.076818
47                                          0.080373

Months since Jan 2019  Electronics and appliance stores  \
0                                              0.079349
1                                              0.072768
2                                              0.079713
3                                              0.072020
4                                              0.078942
5                                              0.077423
6                                              0.079415
7                                              0.083124
8                                              0.078557
9                                              0.078051
10                                             0.101492
11                                             0.119145
12                                             0.093566
13                                             0.088040
14                                             0.076264
15                                             0.039701
16                                             0.051544
17                                             0.073420
18                                             0.085062
19                                             0.088805
20                                             0.083828
21                                             0.089086
22                                             0.107596
23                                             0.123087
24                                             0.072109
25                                             0.064677
26                                             0.083295
```

27	0.077830
28	0.079616
29	0.083840
30	0.084257
31	0.083573
32	0.080343
33	0.084012
34	0.095176
35	0.111270
36	0.075223
37	0.071899
38	0.085359
39	0.080666
40	0.080623
41	0.079515
42	0.081177
43	0.084599
44	0.082328
45	0.079645
46	0.091672
47	0.107294

Months since Jan 2019	Food and beverage stores	Gasoline stations	\
0	0.081435	0.071073	
1	0.073735	0.069206	
2	0.082360	0.081974	
3	0.080827	0.086273	
4	0.086244	0.092353	
5	0.083551	0.087465	
6	0.085945	0.090513	
7	0.086572	0.089750	
8	0.080605	0.083541	
9	0.083518	0.086686	
10	0.085364	0.081015	
11	0.089842	0.080150	
12	0.075264	0.092583	
13	0.072036	0.086239	
14	0.094047	0.079905	
15	0.082175	0.062501	
16	0.088356	0.074816	
17	0.082820	0.084719	
18	0.086564	0.090597	
19	0.083412	0.090096	
20	0.080193	0.086420	
21	0.082903	0.088385	
22	0.082905	0.079392	
23	0.089325	0.084347	

24	0.079255	0.064393
25	0.072839	0.062602
26	0.080440	0.079725
27	0.079269	0.078133
28	0.084783	0.084466
29	0.083187	0.087568
30	0.085983	0.091161
31	0.084912	0.090651
32	0.083058	0.089160
33	0.086265	0.093813
34	0.086074	0.089557
35	0.093936	0.088773
36	0.079020	0.064468
37	0.073153	0.065126
38	0.080597	0.084648
39	0.080959	0.085544
40	0.084491	0.096272
41	0.083931	0.099524
42	0.085964	0.097882
43	0.084445	0.089778
44	0.082572	0.081575
45	0.085350	0.085431
46	0.086452	0.078135
47	0.093066	0.071616

Months since Jan 2019	Clothing and accessories stores \
0	0.060099
1	0.066872
2	0.082086
3	0.079941
4	0.085910
5	0.078578
6	0.080924
7	0.088728
8	0.072792
9	0.079636
10	0.093520
11	0.130914
12	0.083616
13	0.095446
14	0.054582
15	0.013748
16	0.043182
17	0.080624
18	0.087428
19	0.092683
20	0.091514

21	0.097709
22	0.103644
23	0.155823
24	0.052503
25	0.054739
26	0.080763
27	0.079015
28	0.088129
29	0.083825
30	0.085879
31	0.085288
32	0.076886
33	0.082786
34	0.097505
35	0.132681
36	0.057862
37	0.065445
38	0.082316
39	0.082940
40	0.085963
41	0.080218
42	0.080877
43	0.084596
44	0.076737
45	0.080497
46	0.093027
47	0.129523

Months since Jan 2019	Hobbies and recreation	General merchandise stores	\
0	0.070030	0.072712	
1	0.065050	0.070719	
2	0.078745	0.081996	
3	0.077449	0.079606	
4	0.080807	0.085002	
5	0.082668	0.082436	
6	0.082215	0.082233	
7	0.094249	0.086545	
8	0.078267	0.076431	
9	0.079562	0.082257	
10	0.091345	0.092148	
11	0.119613	0.107915	
12	0.068672	0.073467	
13	0.065760	0.074935	
14	0.061732	0.082760	
15	0.040759	0.074408	
16	0.074031	0.087289	
17	0.097524	0.082948	

18	0.093424	0.083784
19	0.094541	0.084259
20	0.089526	0.079015
21	0.087506	0.085104
22	0.096763	0.088586
23	0.129762	0.103445
24	0.069917	0.073798
25	0.058658	0.066799
26	0.086786	0.081315
27	0.081391	0.081187
28	0.080571	0.086749
29	0.085557	0.081781
30	0.082132	0.083703
31	0.088708	0.084659
32	0.078083	0.078545
33	0.077927	0.086291
34	0.092397	0.089699
35	0.117871	0.105474
36	0.065508	0.071841
37	0.064457	0.069549
38	0.079205	0.077949
39	0.079224	0.081744
40	0.079779	0.085697
41	0.084020	0.083765
42	0.082921	0.083968
43	0.091919	0.084927
44	0.080897	0.080374
45	0.079458	0.085535
46	0.091754	0.089370
47	0.120860	0.105282

Months since Jan 2019	Office supplies and gift stores \
0	0.069442
1	0.066261
2	0.073706
3	0.074349
4	0.086058
5	0.081523
6	0.089814
7	0.096108
8	0.080677
9	0.099492
10	0.079357
11	0.103215
12	0.086426
13	0.082907
14	0.068580

15	0.034646
16	0.052535
17	0.072057
18	0.089527
19	0.097822
20	0.096271
21	0.118894
22	0.084541
23	0.115794
24	0.061820
25	0.058600
26	0.074869
27	0.074767
28	0.077478
29	0.083545
30	0.088900
31	0.095204
32	0.093171
33	0.107643
34	0.079376
35	0.104626
36	0.058465
37	0.060765
38	0.074183
39	0.073352
40	0.082042
41	0.081148
42	0.088751
43	0.099070
44	0.090988
45	0.109805
46	0.075589
47	0.105843

Months since Jan 2019	E-commerce and other non-store retailers	Oil/Gasoline	\
0	0.075831	0.133528	
1	0.070524	0.114420	
2	0.075775	0.107269	
3	0.076969	0.074888	
4	0.082469	0.062816	
5	0.076594	0.051060	
6	0.081638	0.053091	
7	0.081382	0.055780	
8	0.077588	0.058211	
9	0.086228	0.076032	
10	0.095370	0.095998	
11	0.119632	0.116908	

12	0.064330	0.144015
13	0.061716	0.120754
14	0.070368	0.101871
15	0.078195	0.069545
16	0.084206	0.058324
17	0.082259	0.052680
18	0.085413	0.053399
19	0.084557	0.054185
20	0.082730	0.062669
21	0.089156	0.075018
22	0.101618	0.084630
23	0.115452	0.122909
24	0.074564	0.111399
25	0.069992	0.119496
26	0.083502	0.101305
27	0.082212	0.070095
28	0.079531	0.059017
29	0.080994	0.057302
30	0.077150	0.054040
31	0.081092	0.055081
32	0.077869	0.062447
33	0.084473	0.082719
34	0.101005	0.105747
35	0.107617	0.121352
36	0.076754	0.113360
37	0.072573	0.111076
38	0.082195	0.104138
39	0.079505	0.073338
40	0.082470	0.071837
41	0.079281	0.065956
42	0.079699	0.052502
43	0.083556	0.059546
44	0.079441	0.061937
45	0.083906	0.083873
46	0.096240	0.089838
47	0.104381	0.112599

Months since Jan 2019	Food services and drinking places \
0	0.073116
1	0.073287
2	0.085266
3	0.081851
4	0.087410
5	0.085572
6	0.085792
7	0.088481
8	0.082516

9	0.085939
10	0.083178
11	0.087593
12	0.097830
13	0.100820
14	0.076819
15	0.048233
16	0.067877
17	0.082915
18	0.087930
19	0.091236
20	0.088799
21	0.092938
22	0.082116
23	0.082487
24	0.064984
25	0.062145
26	0.079945
27	0.082357
28	0.088575
29	0.088590
30	0.093170
31	0.090399
32	0.086219
33	0.090354
34	0.084179
35	0.089082
36	0.068487
37	0.071131
38	0.082921
39	0.084826
40	0.088259
41	0.085903
42	0.087665
43	0.087431
44	0.084842
45	0.088151
46	0.081549
47	0.088835

Months since Jan 2019	Furniture and interior furnishing	Home Improvement	\
0	0.073429	0.071606	
1	0.071186	0.064796	
2	0.082145	0.080461	
3	0.078893	0.090730	
4	0.084873	0.099743	
5	0.079337	0.090063	

6	0.083858	0.090462
7	0.087469	0.087290
8	0.082285	0.080710
9	0.084409	0.087759
10	0.093390	0.079784
11	0.098727	0.076595
12	0.090773	0.063576
13	0.088847	0.061795
14	0.073331	0.075477
15	0.032396	0.084035
16	0.064568	0.099571
17	0.090411	0.098949
18	0.097534	0.092731
19	0.102670	0.086549
20	0.101830	0.086833
21	0.083457	0.088135
22	0.083942	0.081094
23	0.090241	0.081254
24	0.073756	0.064648
25	0.070057	0.061393
26	0.087768	0.089468
27	0.084304	0.096740
28	0.084198	0.096654
29	0.082540	0.094038
30	0.084688	0.086182
31	0.085094	0.083132
32	0.084198	0.081654
33	0.084631	0.082810
34	0.089838	0.082790
35	0.088928	0.080491
36	0.072695	0.066118
37	0.074741	0.065609
38	0.087880	0.086560
39	0.085232	0.089888
40	0.084935	0.097846
41	0.082790	0.094491
42	0.082316	0.085444
43	0.087243	0.088676
44	0.084163	0.083927
45	0.084135	0.084830
46	0.087214	0.079754
47	0.086655	0.076858

Months since Jan 2019	Health and personal care	Misc. store retailers
0	0.084110	0.070220
1	0.078260	0.068762
2	0.083422	0.077138

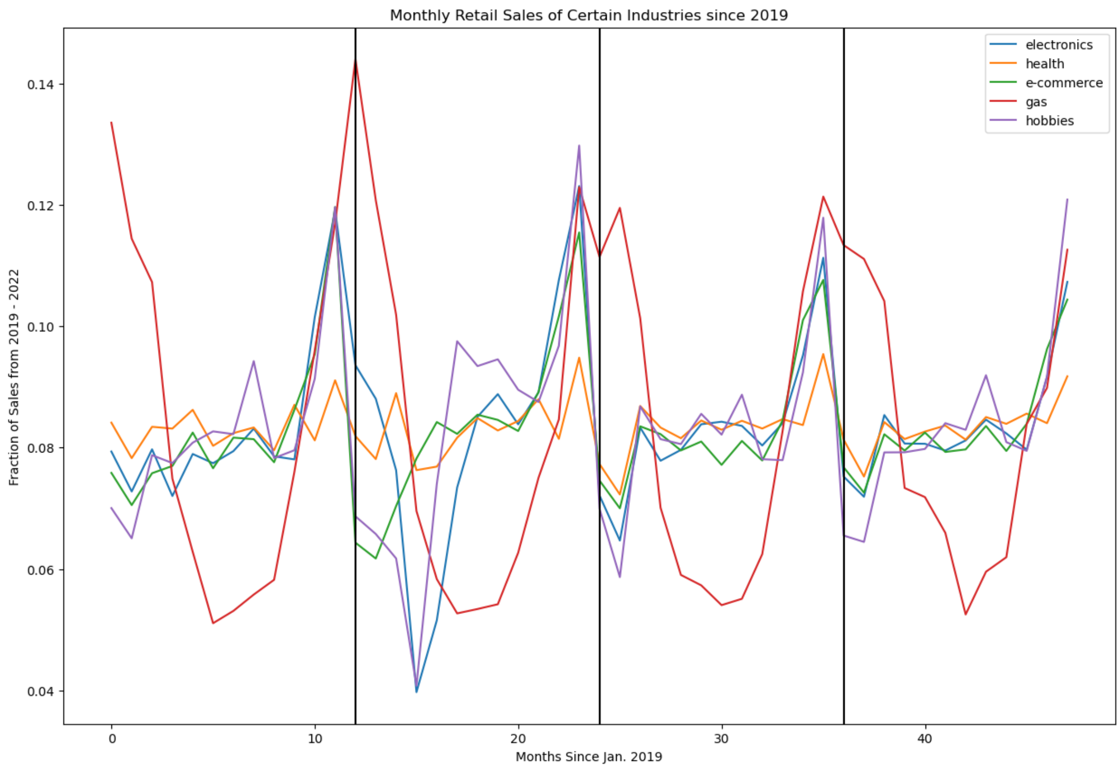
3	0.083129	0.080714
4	0.086218	0.092359
5	0.080289	0.084130
6	0.082379	0.084716
7	0.083311	0.088857
8	0.079564	0.081826
9	0.087054	0.092027
10	0.081185	0.087932
11	0.091080	0.091321
12	0.081889	0.081546
13	0.078102	0.081256
14	0.088974	0.071983
15	0.076277	0.052268
16	0.076863	0.072722
17	0.081629	0.085693
18	0.084856	0.089756
19	0.082819	0.090993
20	0.084373	0.088049
21	0.087968	0.095429
22	0.081438	0.089383
23	0.094813	0.100923
24	0.077289	0.065420
25	0.072270	0.062956
26	0.086848	0.080789
27	0.083298	0.082835
28	0.081540	0.084474
29	0.084458	0.086245
30	0.082939	0.087692
31	0.084413	0.087445
32	0.083137	0.085805
33	0.084690	0.092065
34	0.083709	0.089898
35	0.095410	0.094376
36	0.081308	0.068774
37	0.075226	0.070094
38	0.084200	0.082074
39	0.081402	0.083616
40	0.082637	0.090315
41	0.083626	0.089907
42	0.081304	0.087080
43	0.085029	0.088671
44	0.083913	0.083861
45	0.085597	0.086806
46	0.084015	0.082050
47	0.091742	0.086752

8.0.6 Plots of Normalized Month-by-Month data from 2019 - 2022

(vertical lines to partition years)

[18]:
```
# Plot of Monthly Retail Sales of Industries since 2019
fig = plt.figure(figsize = (15, 10))
ax = plt.axes()
ax1 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y =␣
 ↪'Electronics and appliance stores', label = 'electronics')
ax2 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Health and␣
 ↪personal care', label = 'health')
ax3 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'E-commerce␣
 ↪and other non-store retailers', label = 'e-commerce')
ax4 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Oil/
 ↪Gasoline', label = 'gas')
ax5 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Hobbies␣
 ↪and recreation', label = 'hobbies')


plt.xlabel('Months Since Jan. 2019')
plt.ylabel('Fraction of Sales from 2019 - 2022')
plt.title('Monthly Retail Sales of Certain Industries since 2019')
plt.axvline(12, color = 'black')
plt.axvline(24, color = 'black')
plt.axvline(36, color = 'black')
plt.legend()
plt.show()
```

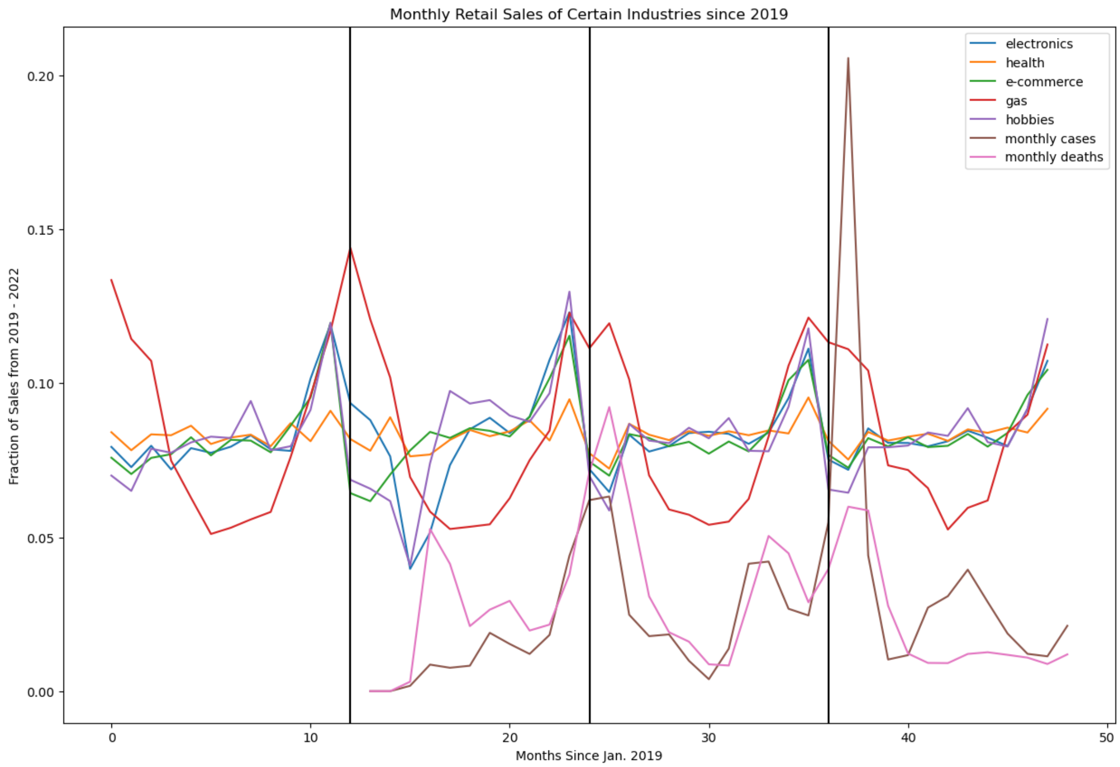


8.0.7 Overlaying COVID-19 Infection Rates and Retail Sales (2019 - 2022)

[19]:
```
fig = plt.figure(figsize = (15, 10))
ax = plt.axes()
ax1 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y =␣
 ↪'Electronics and appliance stores', label = 'electronics')
ax2 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Health and␣
 ↪personal care', label = 'health')
ax3 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'E-commerce␣
 ↪and other non-store retailers', label = 'e-commerce')
ax4 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Oil/
 ↪Gasoline', label = 'gas')
ax5 = sns.lineplot(data = concat_19_22, x = concat_19_22.index, y = 'Hobbies␣
 ↪and recreation', label = 'hobbies')
ax6 = sns.lineplot(data = covid_norm, x = 'Months since Jan. 2019', y =␣
 ↪'new_cases', label = 'monthly cases')
ax7 = sns.lineplot(data = covid_norm, x = 'Months since Jan. 2019', y =␣
 ↪'new_deaths', label = 'monthly deaths')

plt.xlabel('Months Since Jan. 2019')
plt.ylabel('Fraction of Sales from 2019 - 2022')
plt.title('Monthly Retail Sales of Certain Industries since 2019')
```

```
plt.axvline(12, color = 'black')
plt.axvline(24, color = 'black')
plt.axvline(36, color = 'black')
plt.legend()
plt.show()
```



8.1 Fitting Linear Regressions

Lastly, we fit our multiple linear regression models for each of the industries in which we are interested.

The models will be of the form: $retail\ sales = \beta_0 + \beta_1(new\ cases) + \beta_2(new\ deaths)$

The industries we plan on modeling include: - Hobbies and recreation - e-Commerce - Electronics and appliances - Oil/Gasoline - Health and personal care

[20]:
```
# Creating dataframe containing COVID-19 statistics, time period, and␣
↪industries of interest

reg_df = concat_19_22[['Health and personal care', 'Hobbies and recreation',␣
↪'E-commerce and other non-store retailers', 'Oil/Gasoline', 'Electronics and␣
↪appliance stores']]
reg_df = reg_df[12:]
reg_df = reg_df.rename(columns = {'Hobbies and recreation' : 'hobbies_rec',
```

```
                                    'E-commerce and other non-store retailers' :␣
    ↪'e_comm',
                                    'Oil/Gasoline' : 'oil_gas',
                                    'Electronics and appliance stores' :␣
    ↪'elec_app',
                                    'Health and personal care' : 'health'})
reg_df['new_cases'] = covid_norm['new_cases']
reg_df['new_deaths'] = covid_norm['new_deaths']
reg_df = reg_df.reset_index(drop = True)
reg_df = reg_df.rename_axis('Months since Jan 2020', axis = 1)
reg_df.head()
```

[20]:
```
Months since Jan 2020     health  hobbies_rec    e_comm   oil_gas  elec_app  \
0                       0.081889     0.068672  0.064330  0.144015  0.093566
1                       0.078102     0.065760  0.061716  0.120754  0.088040
2                       0.088974     0.061732  0.070368  0.101871  0.076264
3                       0.076277     0.040759  0.078195  0.069545  0.039701
4                       0.076863     0.074031  0.084206  0.058324  0.051544

Months since Jan 2020  new_cases  new_deaths
0                       0.063226    0.092285
1                       0.024817    0.062430
2                       0.017871    0.030784
3                       0.018430    0.019143
4                       0.009875    0.016077
```

[24]:
```
# Fitting the linear regression models and viewing summary statisitics

health_model = smf.ols('health ~ new_cases + new_deaths', data = reg_df).fit()
hobbies_model = smf.ols('hobbies_rec ~ new_cases + new_deaths', data = reg_df).
    ↪fit()
e_comm_model = smf.ols('e_comm ~ new_cases + new_deaths', data = reg_df).fit()
oil_gas_model = smf.ols('oil_gas ~ new_cases + new_deaths', data = reg_df).fit()
elec_app_model = smf.ols('elec_app ~ new_cases + new_deaths', data = reg_df).
    ↪fit()

print(health_model.summary(), hobbies_model.summary(), e_comm_model.summary(),␣
    ↪oil_gas_model.summary(), elec_app_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 health   R-squared:                       0.063
Model:                            OLS   Adj. R-squared:                 -0.026
Method:                 Least Squares   F-statistic:                    0.7062
Date:                Mon, 12 Jun 2023   Prob (F-statistic):              0.505
Time:                        16:58:09   Log-Likelihood:                 93.087
No. Observations:                  24   AIC:                            -180.2
```

```
Df Residuals:                      21   BIC:                            -176.6
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0850      0.002     47.679      0.000       0.081       0.089
new_cases     -0.0156      0.033     -0.473      0.641      -0.084       0.053
new_deaths    -0.0393      0.058     -0.681      0.503      -0.159       0.081
==============================================================================
Omnibus:                        3.527   Durbin-Watson:                   2.007
Prob(Omnibus):                  0.171   Jarque-Bera (JB):                1.855
Skew:                           0.578   Prob(JB):                        0.396
Kurtosis:                       3.719   Cond. No.                         55.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.                            OLS Regression Results
==============================================================================
Dep. Variable:            hobbies_rec   R-squared:                       0.073
Model:                            OLS   Adj. R-squared:                 -0.015
Method:                 Least Squares   F-statistic:                    0.8265
Date:                Mon, 12 Jun 2023   Prob (F-statistic):              0.451
Time:                        16:58:09   Log-Likelihood:                 63.098
No. Observations:                  24   AIC:                            -120.2
Df Residuals:                      21   BIC:                            -116.7
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0893      0.006     14.365      0.000       0.076       0.102
new_cases      0.0221      0.115      0.192      0.850      -0.217       0.262
new_deaths    -0.2385      0.201     -1.184      0.250      -0.657       0.180
==============================================================================
Omnibus:                        5.164   Durbin-Watson:                   1.192
Prob(Omnibus):                  0.076   Jarque-Bera (JB):                3.982
Skew:                           0.357   Prob(JB):                        0.137
Kurtosis:                       4.863   Cond. No.                         55.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.                            OLS Regression Results
==============================================================================
Dep. Variable:                 e_comm   R-squared:                       0.157
Model:                            OLS   Adj. R-squared:                  0.077
```

```
Method:                 Least Squares   F-statistic:                     1.956
Date:                Mon, 12 Jun 2023   Prob (F-statistic):              0.166
Time:                        16:58:09   Log-Likelihood:                 73.379
No. Observations:                  24   AIC:                            -140.8
Df Residuals:                      21   BIC:                            -137.2
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0894      0.004     22.063      0.000       0.081       0.098
new_cases      0.0181      0.075      0.241      0.812      -0.138       0.174
new_deaths    -0.2361      0.131     -1.799      0.086      -0.509       0.037
==============================================================================
Omnibus:                       12.572   Durbin-Watson:                   0.814
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               11.056
Skew:                           1.464   Prob(JB):                      0.00397
Kurtosis:                       4.576   Cond. No.                         55.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.                            OLS Regression Results
==============================================================================
Dep. Variable:                oil_gas   R-squared:                       0.452
Model:                            OLS   Adj. R-squared:                  0.400
Method:                 Least Squares   F-statistic:                     8.663
Date:                Mon, 12 Jun 2023   Prob (F-statistic):            0.00181
Time:                        16:58:09   Log-Likelihood:                 58.928
No. Observations:                  24   AIC:                            -111.9
Df Residuals:                      21   BIC:                            -108.3
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0596      0.007      8.057      0.000       0.044       0.075
new_cases     -0.0228      0.137     -0.166      0.869      -0.308       0.262
new_deaths     0.8704      0.240      3.632      0.002       0.372       1.369
==============================================================================
Omnibus:                        2.325   Durbin-Watson:                   0.501
Prob(Omnibus):                  0.313   Jarque-Bera (JB):                1.278
Skew:                           0.556   Prob(JB):                        0.528
Kurtosis:                       3.207   Cond. No.                         55.9
==============================================================================

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

```
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:               elec_app   R-squared:                       0.012
Model:                            OLS   Adj. R-squared:                 -0.082
Method:                 Least Squares   F-statistic:                    0.1284
Date:                Mon, 12 Jun 2023   Prob (F-statistic):              0.880
Time:                        16:58:09   Log-Likelihood:                 63.988
No. Observations:                  24   AIC:                            -122.0
Df Residuals:                      21   BIC:                            -118.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0814      0.006     13.594      0.000       0.069       0.094
new_cases     -0.0259      0.111     -0.233      0.818      -0.257       0.205
new_deaths     0.0981      0.194      0.505      0.619      -0.306       0.502
==============================================================================
Omnibus:                        3.078   Durbin-Watson:                   0.924
Prob(Omnibus):                  0.215   Jarque-Bera (JB):                1.597
Skew:                          -0.185   Prob(JB):                        0.450
Kurtosis:                       4.208   Cond. No.                         55.9
==============================================================================

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

[23]:
```python
# Generating p-values for the coefficients of each model to determine␣
↪statistical significance

model_list = [e_comm_model, hobbies_model, elec_app_model, health_model,␣
↪oil_gas_model]
model_names = ['e_comm_model', 'hobbies_model', 'elec_app_model',␣
↪'health_model', 'oil_gas_model']


for i, model in enumerate(model_list):
    print(model_names[i])
    print(model.pvalues[1:])
    print('')
```

```
e_comm_model
new_cases     0.812019
new_deaths    0.086423
dtype: float64
```

```
hobbies_model
new_cases     0.849540
new_deaths    0.249665
dtype: float64

elec_app_model
new_cases     0.817759
new_deaths    0.618532
dtype: float64

health_model
new_cases     0.640887
new_deaths    0.503022
dtype: float64

oil_gas_model
new_cases     0.869467
new_deaths    0.001562
dtype: float64
```

9 Ethics & Privacy

When considering biases and privacy issues with the data proposed, a large proportion of our actual consumer data are obtained from publicly available reports published by large companies who are inclined to publish and release data that puts them in a positive light to promote their brand and impress their stock investors. It is often the case that our data is collected by a third party, meaning that we were not directly involved in its collection process and unable to ensure the same procedures across all of the data collection. This inconsistency, in combination with the notion that our data comes from large corporate companies, proves problematic for allowing equitable analysis because our compiled data might be presented in a way that promotes the perceived profitability of the corporation's returns being analyzed.

In addition to the main issue of not fully knowing how objective our collected data is, another concern would be the lack of substantial data on small business commerce, meaning our data would be more representative of specifically trends among major corporations. Lastly, our research question focuses on specific industries within the U.S., yet it is inevitable that there are other major global events and external factors that may further influence consumer behaviors beyond the pandemic itself. With this in mind, the conclusions we draw from our analyses would not be telling a complete story.

In order to address our concerns, we will acknowledge the biases and present our results as a rough estimate of observed consumer behavior over time rather than a detailed reflection of customer purchases – since smaller businesses and specialty businesses are not included in our analysis of retail companies. Finally, we plan to weigh the data in each category in proportion to its volume with other categories so as to dampen the strength of outliers in raw counts.

10 Conclusion & Discussion

In conclusion, we were unable to reject the null hypothesis for any coefficients in our multiple linear regression models, except for the relationship between COVID-19 death rates and oil/gasoline sales. This is because the p -values for every other relationship was well-above our threshold of $\alpha = 0.05$. Therefore, we found that the presence of new COVID-19 cases from 2020 to 2022 did not have a significant impact on American spending habits across the industries we examined.

From our models, COVID-19 infection rates positively affected American consumer spending on hobbies and recreation, and e-commerce, while having a negative effect on electronics, oil and gas, and health and personal care. Similarly, COVID-19 death rates positively affected American consumer spending on oil and gas and electronics, and negatively on hobbies and recreation, e-commerce and health and personal care. Given that death rates and infection rates are oppositely correlated with one another for every industry except health products, we can only claim that American consumership of health and personal care products were negatively correlated with COVID-19 spread.

However, it is important to acknowledge the limitations of our analysis. While we did find the presence of a significant impact, our analysis is still limited by shortcomings in both time and experience. Due to our lack of experience in data analysis, we had to reshape our project multiple times throughout the quarter, thus restricting how in-depth we were able to get with our analysis in order to meet our deadline. In addition to this, the scale of our analysis is also limited by how we decided to only look at data from American spending, thus our results cannot be generalized to other regions or countries around the world. Also, the data is a generalization of the entire U.S., meaning that we cannot apply this data to a specific region of the country or outside of the U.S. as different factors may come into play in each sector.

It is also entirely plausible that our focused industries bore little significance in explaining a connection between American spending and COVID-19. Various industries could have been affected by other global factors present simultaneously with COVID-19, such as an existing computer chip shortage or economic sanctions on Russia. These could have resulted in a decline in spending seen in their respective industries. Holidays too might've had some impact, potentially explaining the cyclical nature of American spending.

Despite these limitations, our research provides insights into COVID-19 mortality and infection rates in relation to American consumer spending habits. Further exploration and analysis are necessary to gain a more comprehensive understanding of the complex dynamics at play and to account for the diverse factors influencing consumer behavior during the pandemic.