In [1]:
```
#import libraries
import pandas as pd
import regex as re

import matplotlib.pyplot as plt
import plotly.express as px

from ydata_profiling import ProfileReport


from sklearn.cluster import MeanShift
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
import plotly.io as pio
pio.renderers.default = "notebook+pdf"
```

In [3]:
```
pip install -U kaleido
```

```
Requirement already satisfied: kaleido in /Users/shreyabaral/anaconda3/lib/python3.11/site-packages (0.2.1)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pd.set_option('display.max_colwidth', None) #show column without truncation
```

In [5]:
```
 pd.set_option('display.max_rows', None) #show rows without truncation
```

This report provides the analysis of credit card report published by London Borough of Barnet. We are given a dataset which is in CSV files. We will start by ,merging different csv file into pandas dataframe. We will drop the synonymous columns and the irrelavant column. Rename the columns if required and merge the csv files into a dataframe.

In [6]:
```
#reads the csv files into a dataframe
df1=pd.read_csv('Dataset/PCard 1617.csv')
df2=pd.read_csv('Dataset/PCard Transactions 15-16.csv')
df3=pd.read_csv('Dataset/Purchasing Card Data 2014 v1.csv')
```

In [7]:
```
#gets first five column of the dataframe
df1.head()
```

Out[7]:

	Service Area	Account Description	Creditor	Journal Date	Journal Reference	Total
0	Adults and Communities	Books-CDs-Audio-Video	AMAZON EU	05/12/2016	10510.0	45.00
1	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK MARKETPLACE	05/12/2016	10509.0	426.57
2	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK RETAIL AMAZO	06/12/2016	10524.0	121.38
3	Adults and Communities	Consumable Catering Supplies	WWW.ARGOS.CO.UK	01/03/2017	11667.0	78.94
4	Adults and Communities	CSG - IT	AMAZON UK MARKETPLACE	01/02/2017	10974.0	97.50

In [8]:
```
df2.head()
```

Out[8]:

	Service Area	Account Description	Creditor	Journal Date	Journal Reference	Total
0	Assurance	Miscellaneous Expenses	43033820 COSTA COFFEE	18/08/2015	5043.0	2
1	Children's Family Services	Miscellaneous Expenses	99 PLUS DISCOUNT MART	08/06/2015	4184.0	29.97
2	Children's Family Services	E19 - Learning Resources	99P STORES LTD	07/12/2015	6278.0	34.65
3	Children's Family Services	Equipment and Materials Purcha	99P STORES LTD	18/08/2015	5041.0	10.72
4	Children's Family Services	Subsistence	CHOPSTIX00000000000	21/05/2015	5750.0	33.7

In [9]:
```
df3.head()
```

Out[9]:

	Service Area	Account Description	Creditor	Transaction Date	JV Reference	JV Date	JV Value
0	Childrens Services	IT Services	123-REG.CO.UK	23/04/2014	93	20/05/2014	143.81
1	Childrens Services	Other Services	ACCESS EXPEDITIONS	03/04/2014	111	20/05/2014	6,000.00
2	Childrens Services	Equipment and Materials Repair	AFE SERVICELINE	02/04/2014	6	20/05/2014	309.38
3	Childrens Services	Equipment and Materials Repair	AFE SERVICELINE	02/04/2014	7	20/05/2014	218.76
4	Childrens Services	Building Repairs & Maintenance	ALLSOP & FRANCIS	15/04/2014	381	20/05/2014	306

In [10]:
```
#drop unnecessary columns
df3.drop(columns=['JV Reference','JV Date'],inplace=True)
df1.drop(columns=['Journal Reference'],inplace=True)
df2.drop(columns=['Journal Reference'],inplace=True)
```

In [11]:
```
#renaming the synonymous columns
df3.rename(columns={"Transaction Date": "Journal Date", "JV Value": "Total"},inplace=True)
```

In [12]:
```
df3.head()
```

Out[12]:

	Service Area	Account Description	Creditor	Journal Date	Total
0	Childrens Services	IT Services	123-REG.CO.UK	23/04/2014	143.81
1	Childrens Services	Other Services	ACCESS EXPEDITIONS	03/04/2014	6,000.00
2	Childrens Services	Equipment and Materials Repair	AFE SERVICELINE	02/04/2014	309.38

3	Childrens Services	Equipment and Materials Repair	AFE SERVICELINE	02/04/2014	218.76
4	Childrens Services	Building Repairs & Maintenance	ALLSOP & FRANCIS	15/04/2014	306

In [13]:
```
#combining datasets
df= pd.concat([df1,df2,df3],axis=0)
```

In [14]:
```
#rows and columns of the dataframe
df.shape
```

Out[14]:
```
(12589, 5)
```

In [15]:
```
#finding the duolicate values
df.duplicated().sum()
```

Out[15]:
```
726
```

In [16]:
```
#drop duplicate values
df.drop_duplicates(inplace=True)
```

In [17]:
```
#percentage of null values
df.isna().mean()
```

Out[17]:
```
Service Area           0.000084
Account Description    0.000169
Creditor               0.000169
Journal Date           0.000169
Total                  0.000000
dtype: float64
```

In [18]:
```
#number of null values in each column
df.isna().sum()
```

Out[18]:
```
Service Area           1
Account Description    2
Creditor               2
Journal Date           2
Total                  0
dtype: int64
```

In [19]:
```
#dropping null values
df.dropna(inplace=True)
```

In [20]:
```
#information about the count , datatype and memory usuage
df.info()
```

Out[20]:
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11861 entries, 0 to 4141
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Service Area         11861 non-null  object
 1   Account Description  11861 non-null  object
 2   Creditor             11861 non-null  object
 3   Journal Date         11861 non-null  object
 4   Total                11861 non-null  object
dtypes: object(5)
memory usage: 556.0+ KB
```

In [21]:
```
#rows and columns after initial data cleaning
df.shape
```

Out[21]:
```
(11861, 5)
```

From the df.info() we found out, all the columns in our dataframe are categorical. We will be converting the datetime and total columns to datetime and numeric columns respectively. This will give us the flexibility to perform different calculations and analyse the behaviour of our data.

In [22]:
```
#converting total column into numeric
def convert_total(i):
    output= re.sub(r'[,]',"",i)
    return output

df['Total'] = df['Total'].apply(lambda x: convert_total(x))
df['Total']=pd.to_numeric(df['Total'])
```

In [23]:
```
#verifying the datatype of total column
df['Total'].dtype
```

Out[23]:
```
dtype('float64')
```

In [24]:
```
#statistical summary of our data
df.describe()
```

Out[24]:

	Total
count	11861.000000
mean	100.880743
std	394.758945
min	-4707.000000
25%	10.000000

50%	28.570000
75%	92.870000
max	15340.800000

From the statistical summary of our data , we can see that we have total of 11861 rows. The maximum value id 15340 while minimum value is -4707.Our mean is 100.88, std is 394 , median is 28, 25% is 10 and 75% is 92.87.

This summary shows that data is widely ranged with significant presence of outliers.This also concludes that mean and std is highly influenced by the presence of outliers.

In [25]:
```
#sample of our data
df.sample(5)
```

Out[25]:

	Service Area	Account Description	Creditor	Journal Date	Total
3292	Children's Family Services	Other Transfer Payments to Soc	PABULUM CATERING	31/01/2017	20.00
1394	Children's Family Services	Other Services	EBUYER (UK) LTD	08/07/2015	989.97
2464	Children's Family Services	Miscellaneous Expenses	WWW.CIMAGLOBAL.COM	18/11/2016	108.00
554	Children's Family Services	Other Services	AMAZON UK MARKETPLACE	26/10/2015	39.98
2159	Children's Family Services	Food Costs	WAITROSE 191	13/05/2016	6.33

Data Understanding

In [26]:
```
#getting number of unique elements in our columns
for col in df.columns:
    print(col, df[col].nunique())
    print('-----------')
```

```
Service Area 24
-----------
Account Description 67
-----------
Creditor 1936
-----------
Journal Date 739
-----------
Total 5880
-----------
```

From the above output, we can see that we have total of 24 different service areas and 67 different account.

Feature Engineering

Feature engineering is the process of extracting the required information from the data. We will extract quarter and year from the journal date column and store it in a new column in our dataframe

In [27]:
```
#converting the object value to datetime datatype
df['Journal Date'] = pd.to_datetime(df['Journal Date'], format='%d/%m/%Y')
```

In [28]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11861 entries, 0 to 4141
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Service Area         11861 non-null  object
 1   Account Description  11861 non-null  object
 2   Creditor             11861 non-null  object
 3   Journal Date         11861 non-null  datetime64[ns]
 4   Total                11861 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(3)
memory usage: 556.0+ KB
```

In [29]:
```
#creating new column quarter which stores the quarter values
df['Quarter'] = df['Journal Date'].dt.quarter
```

In [30]:
```
df['Quarter'].value_counts(dropna=False)
```

Out[30]:
```
4    3283
1    2971
3    2918
2    2689
Name: Quarter, dtype: int64
```

In [31]:
```
#creating new column year which stores the year values
df['Year'] = df['Journal Date'].dt.year
```

In [32]:
```
df.head()
```

Out[32]:

	Service Area	Account Description	Creditor	Journal Date	Total	Quarter	Year
0	Adults and Communities	Books-CDs-Audio-Video	AMAZON EU	2016-12-05	45.00	4	2016

1	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK MARKETPLACE	2016-12-05	426.57	4	2016
2	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK RETAIL AMAZO	2016-12-06	121.38	4	2016
3	Adults and Communities	Consumable Catering Supplies	WWW.ARGOS.CO.UK	2017-03-01	78.94	1	2017
4	Adults and Communities	CSG - IT	AMAZON UK MARKETPLACE	2017-02-01	97.50	1	2017

Pandas Profiling

Pandas profiling is the open source library provided by pandas for quick and easy way to get insights into structure of our data.

In [33]:
```
profile = ProfileReport(df, title="Profiling Report")
profile.to_notebook_iframe()
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Profiling Report

Overview

Overview | Alerts 2 | Reproduction

Dataset statistics

Number of variables	7
Number of observations	11861
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	1
Duplicate rows (%)	< 0.1%
Total size in memory	741.3 KiB
Average record size in memory	64.0 B

Variable types

Categorical	3
Text	2
DateTime	1
Numeric	1

From the pandas profiling we can see that we have imbalanced dataset. Majority of the columns from service area falls under the same category. Some important keywords from our account description are equipment , material , costs while from the creditors are amazon,sainsbury etc. Furthermore it helped us to see the first few and last few columns of our data , missing values and Duplicate rows also the value_counts of each columns.

Task 1

In [34]:
```
service_area_summary= df.groupby(['Year','Quarter','Service Area']).agg(Transaction_Count=('Total', 'count'), Average_Total=('Total', 'mean')
```

Summary table of transaction count, average total, maximum , minimum and total sum per each service area per quarter per each year

In [35]:
```
service_area_summary
```

Out[35]:

			Transaction_Count	Average_Total	Maximum	Minimum	Total_Sum
Year	**Quarter**	**Service Area**					
2014	**2**	**Adults and Communities**	15	252.833333	815.50	20.00	3792.50
		CSG Managed Budget	20	1608.367000	7800.00	-44.99	32167.34
		Childrens Services	875	74.514103	6000.00	-500.00	65199.84
		Control Accounts	8	23.838750	83.31	3.06	190.71

		Deputy Chief Operating Officer	39	40.544615	354.00	2.15	1581.24
		Governance	3	2207.800000	6388.20	75.20	6623.40
		Internal Audit & CAFT	2	203.600000	403.20	4.00	407.20
		NSCSO	1	10.000000	10.00	10.00	10.00
		Public Health	2	-1.175000	10.95	-13.30	-2.35
		Strategic Commissioning Board	1	244.000000	244.00	244.00	244.00
		Street Scene	11	57.280000	117.60	4.20	630.08
	3	Adults and Communities	8	321.306250	840.00	124.00	2570.45
		CSG Managed Budget	12	2045.750000	8058.00	173.00	24549.00
		Children's Service DSG	30	96.591667	449.28	1.49	2897.75
		Childrens Services	320	73.205781	2439.16	-433.91	23425.85
		Commercial	9	304.783333	1008.00	-450.00	2743.05
		Deputy Chief Operating Officer	49	35.270000	312.50	2.25	1728.23
		Education	60	130.974167	830.10	0.50	7858.45
		Family Services	455	66.499099	989.29	-133.20	30257.09
		Governance	3	392.320000	480.03	252.00	1176.96
		Internal Audit & CAFT	7	27.564286	56.07	11.40	192.95
		NSCSO	2	222.750000	300.00	145.50	445.50
		Street Scene	16	35.664375	123.29	2.99	570.63
	4	Adults and Communities	17	118.195882	300.00	20.00	2009.33
		Assurance	3	35.113333	89.94	4.00	105.34
		CSG Managed Budget	3	3187.666667	4707.00	201.00	9563.00
		Children's Education & Skills	60	131.156000	648.00	-180.79	7869.36
		Children's Family Services	536	62.107425	989.05	-179.99	33289.58
		Children's Service DSG	32	162.927812	500.00	2.52	5213.69
		Childrens Services	20	63.205500	259.83	-32.10	1264.11
		Commissioning	42	73.700238	640.00	1.19	3095.41
		Customer Support Group	16	322.266875	3567.00	-4707.00	5156.27
		Deputy Chief Operating Officer	24	35.371250	460.00	2.15	848.91
		Education	35	141.619714	500.00	2.70	4956.69
		Family Services	273	64.203077	890.95	-50.00	17527.44
		Governance	1	53.940000	53.94	53.94	53.94
		Internal Audit & CAFT	2	58.000000	99.00	17.00	116.00
		Parking & Infrastructure	2	46.410000	51.02	41.80	92.82
		Street Scene	12	110.962500	400.00	9.62	1331.55
		Streetscene	19	56.486316	527.52	-527.52	1073.24
2015	1	Adults and Communities	11	183.441818	1000.00	18.67	2017.86
		Assurance	4	16.862500	27.90	6.95	67.45
		Children's Education & Skills	76	125.552105	730.44	-66.55	9541.96
		Children's Family Services	586	67.477918	876.43	-94.50	39542.06
		Children's Service DSG	15	202.047333	520.00	21.59	3030.71
		Commissioning	28	183.551429	1335.16	2.00	5139.44
		Customer Support Group	14	1350.069286	4752.00	50.00	18900.97
		Parking & Infrastructure	1	28.430000	28.43	28.43	28.43
		Regional Enterprise	1	60.000000	60.00	60.00	60.00
		Streetscene	29	133.580345	717.95	-7.14	3873.83
	2	Adults and Communities	8	148.425000	420.00	16.67	1187.40
		Assurance	51	57.854902	1276.92	1.17	2950.60
		Children's Education & Skills	83	136.632410	987.47	-112.00	11340.49
		Children's Family Services	572	74.127360	1240.86	-971.70	42400.85
		Children's Service DSG	12	128.605833	584.00	11.30	1543.27
		Commissioning	11	252.390000	1740.00	30.00	2776.29
		Customer Support Group	10	3218.600000	15340.80	86.40	32186.00
		Streetscene	22	125.762727	652.50	2.99	2766.78
	3	Adults and Communities	14	108.816429	354.00	-16.22	1523.43
		Assurance	38	58.875789	660.50	0.84	2237.28
		Children's Education & Skills	75	132.805600	495.00	-19.50	9960.42
		Children's Family Services	737	76.211316	2262.91	-751.75	56167.74

		Children's Service DSG	20	138.824500	400.00	8.89	2776.49
		Commissioning	17	187.755882	1310.00	13.00	3191.85
		Customer Support Group	13	1539.356154	6955.20	8.24	20011.63
		Streetscene	38	182.196842	2295.60	2.49	6923.48
	4	Adults and Communities	19	159.948947	1391.04	10.00	3039.03
		Assurance	30	70.333667	280.50	4.40	2110.01
		Children's Education & Skills	86	129.250581	489.70	-301.35	11115.55
		Children's Family Services	804	58.544689	1954.80	-65.83	47069.93
		Children's Service DSG	59	76.790169	749.17	-50.69	4530.62
		Commissioning	39	217.144615	3984.00	-27.87	8468.64
		Customer Support Group	11	1084.781818	5418.00	-178.80	11932.60
		Parking & Infrastructure	1	159.670000	159.67	159.67	159.67
		Regional Enterprise	1	1645.000000	1645.00	1645.00	1645.00
		Streetscene	30	146.193667	1098.00	6.01	4385.81
2016	1	Adults and Communities	23	107.326957	499.00	15.09	2468.52
		Assurance	29	29.285517	284.00	0.37	849.28
		Children's Education & Skills	54	138.177593	485.91	0.24	7461.59
		Children's Family Services	854	55.152436	850.60	-537.60	47100.18
		Children's Service DSG	30	131.910000	506.47	1.45	3957.30
		Commissioning	59	139.792034	1200.00	-235.93	8247.73
		Customer Support Group	9	1830.313333	5918.40	97.92	16472.82
		Regional Enterprise	1	60.000000	60.00	60.00	60.00
		Streetscene	19	180.113158	652.50	20.00	3422.15
	2	Adults and Communities	25	146.368800	1200.00	3.29	3659.22
		Assurance	59	161.161695	4342.20	-49.99	9508.54
		Children's Education & Skills	10	48.172000	252.00	-137.10	481.72
		Children's Family Services	743	71.553015	1695.22	-444.98	53163.89
		Children's Service DSG	24	139.278750	788.34	1.25	3342.69
		Commissioning	43	171.402326	1910.40	-780.00	7370.30
		Customer Support Group	10	2898.878000	11487.00	159.00	28988.78
		Parking & Infrastructure	1	500.000000	500.00	500.00	500.00
		Public Health	1	4.550000	4.55	4.55	4.55
		Streetscene	17	224.377059	652.50	11.45	3814.41
	3	Adults and Communities	37	182.227027	3028.20	4.73	6742.40
		Assurance	36	-4.206111	370.00	-1315.20	-151.42
		Children's Education & Skills	15	148.609333	833.33	-79.00	2229.14
		Children's Family Services	784	68.877117	1604.76	-437.50	53999.66
		Children's Service DSG	19	137.795789	1500.00	4.90	2618.12
		Commissioning	54	186.358148	3554.56	-1184.85	10063.34
		Customer Support Group	7	1885.285714	6069.00	-174.00	13197.00
		Parking & Infrastructure	1	76.250000	76.25	76.25	76.25
		Regional Enterprise	1	12.000000	12.00	12.00	12.00
		Streetscene	41	102.771951	249.98	-583.12	4213.65
	4	Adults and Communities	44	206.766818	1670.30	-15.97	9097.74
		Assurance	52	34.285192	399.60	-3.50	1782.83
		Children's Education & Skills	6	208.985000	500.00	9.27	1253.91
		Children's Family Services	871	76.400138	1569.07	-93.98	66544.52
		Children's Service DSG	24	66.637500	351.58	-29.97	1599.30
		Commissioning	52	140.654615	1200.00	-39.60	7314.04
		Customer Support Group	10	1591.130000	6762.00	65.00	15911.30
		HRA	1	289.940000	289.94	289.94	289.94
		Parking & Infrastructure	2	1784.125000	2773.25	795.00	3568.25
		Regional Enterprise	1	226.000000	226.00	226.00	226.00
		Streetscene	43	91.536977	278.99	-5.99	3936.09
2017	1	Adults and Communities	51	169.931961	3569.03	2.00	8666.53
		Assurance	38	30.793947	253.35	0.89	1170.17
		Children's Education & Skills	2	58.050000	68.15	47.95	116.10
		Children's Family Services	940	75.605351	1350.00	-368.00	71069.03

		Children's Service DSG	10	16.833000	48.81	3.80	168.33
		Commissioning	38	61.351316	1782.00	-500.00	2331.35
		Customer Support Group	10	2155.400000	7968.00	-300.00	21554.00
		Parking & Infrastructure	4	58.670000	109.99	11.78	234.68
		Regional Enterprise	1	226.000000	226.00	226.00	226.00
		Streetscene	34	148.336471	866.00	-65.00	5043.44
	2	Adults and Communities	1	79.000000	79.00	79.00	79.00
		Children's Family Services	8	121.751250	660.00	13.75	974.01
		Streetscene	1	86.000000	86.00	86.00	86.00

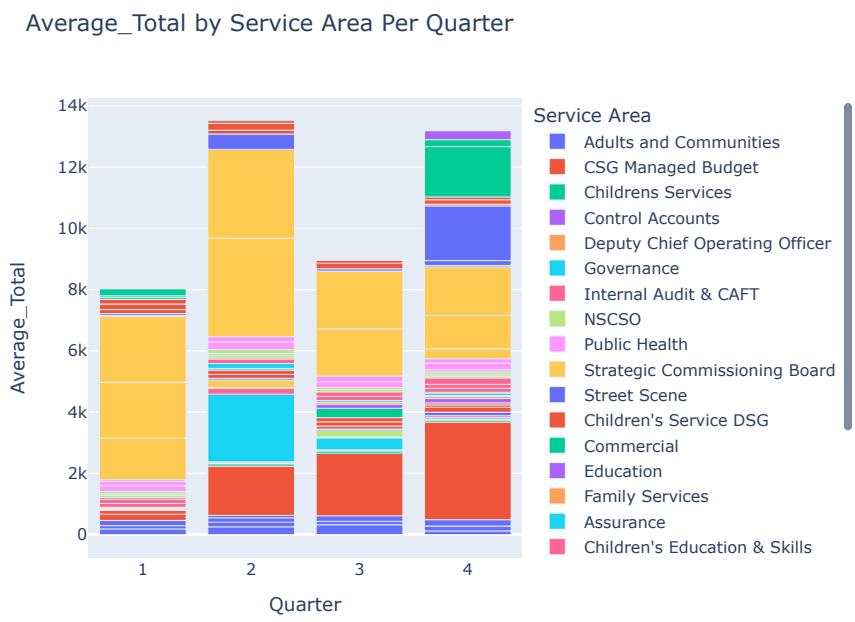
Service area summary is divided into 4 parts based on year i.e 2014,2015,2016,and 2017. Further , it store the statistical summary : transaction count, average total, maximum, minimum and total sum of each service area based on different quarters. This statistical summary is granular which can be helpful to compare and understand the spending patterns of each service area. This can also be helful for budgeting , financial as well as to understand the trends within each service area.

In [36]:
```
service_area_summary[:5]
```

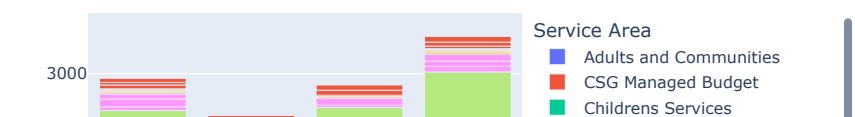
Out[36]:

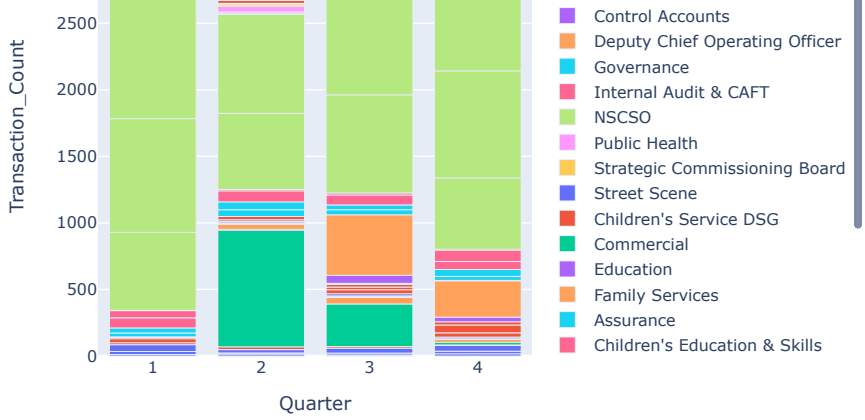
			Transaction_Count	Average_Total	Maximum	Minimum	Total_Sum
Year	Quarter	Service Area					
2014	2	Adults and Communities	15	252.833333	815.50	20.00	3792.50
		CSG Managed Budget	20	1608.367000	7800.00	-44.99	32167.34
		Childrens Services	875	74.514103	6000.00	-500.00	65199.84
		Control Accounts	8	23.838750	83.31	3.06	190.71
		Deputy Chief Operating Officer	39	40.544615	354.00	2.15	1581.24

In [37]:
```
service_area_summary_reset=service_area_summary.reset_index() #reset the index from grouped df

fig = px.bar(service_area_summary_reset, x='Quarter', y='Average_Total', color='Service Area',
             title='Average_Total by Service Area Per Quarter',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})  # Sort values in descending order
fig.show()
```



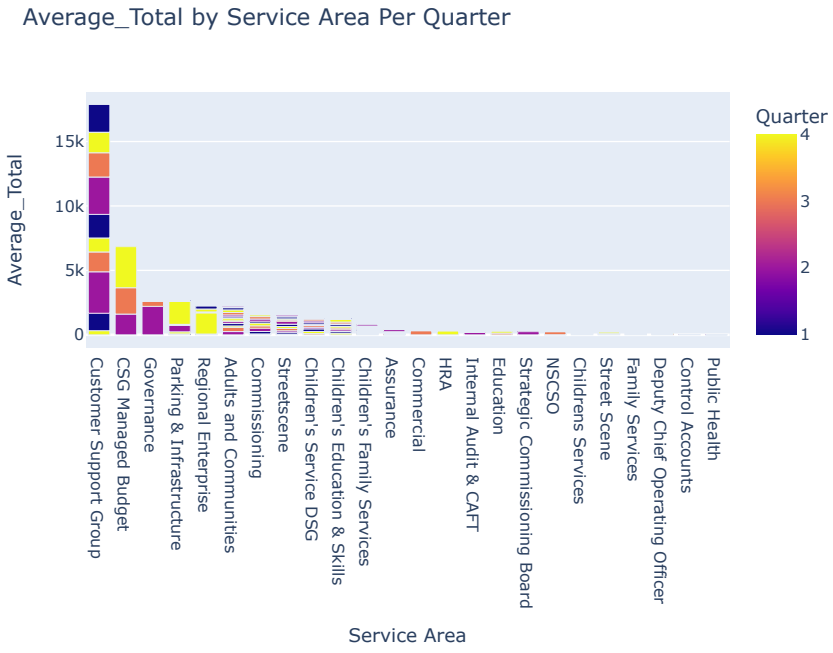
In [38]:
```
service_area_summary_reset=service_area_summary.reset_index() #reset the index from grouped df

fig = px.bar(service_area_summary_reset, x='Quarter', y='Transaction_Count', color='Service Area',
             title='Transaction Count by Service Area Per Quarter',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})  # Sort values in descending order
fig.show()
```





If we look at the summary per year , transaction count slightly increase from 2014 to 2015 to 2016 but it drastically dropped in the year 2017. Similar, pattern can be seen in the average total as well. There is no drastic difference of average total if first 3 years while , average total dropped to less tha 4k from more tha. 14k from year 2016 to 2017.

In [39]:
```
service_area_summary_reset=service_area_summary.reset_index() #reset the index from grouped df

fig = px.bar(service_area_summary_reset, x='Service Area', y='Average_Total', color='Quarter',
             title='Average_Total by Service Area Per Quarter',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})  # Sort values in descending order
fig.show()
```

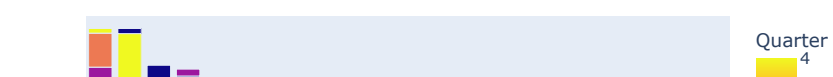


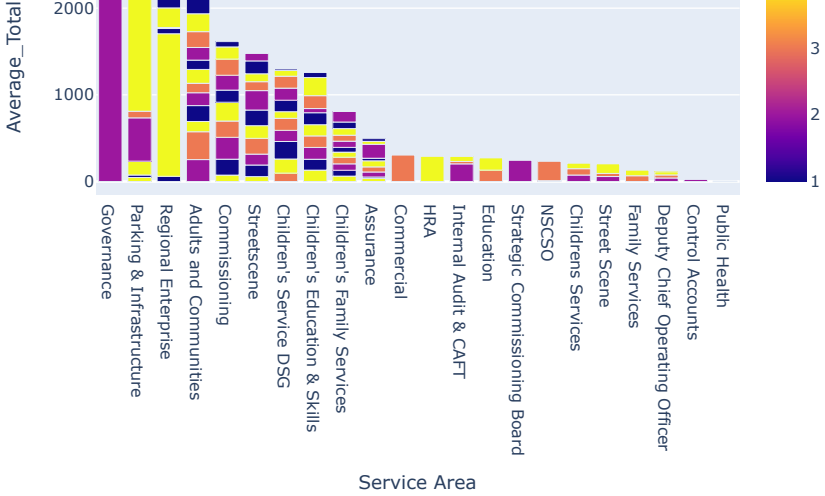
From the above diagram we can see that , our diagram is skewed as Customer Service Support has more tha 15k average total while for most of other values it is less than 5k.

Since , our data is skewed , we are dropping first 2 service area to get a closer look on other values.

In [40]:
```
#dropping Customer Support Group and CSG Manageemnt Budget to remove the biasness
service_area_remake= service_area_summary_reset[(service_area_summary_reset['Service Area']!='Customer Support Group') & (service_area_summa
```

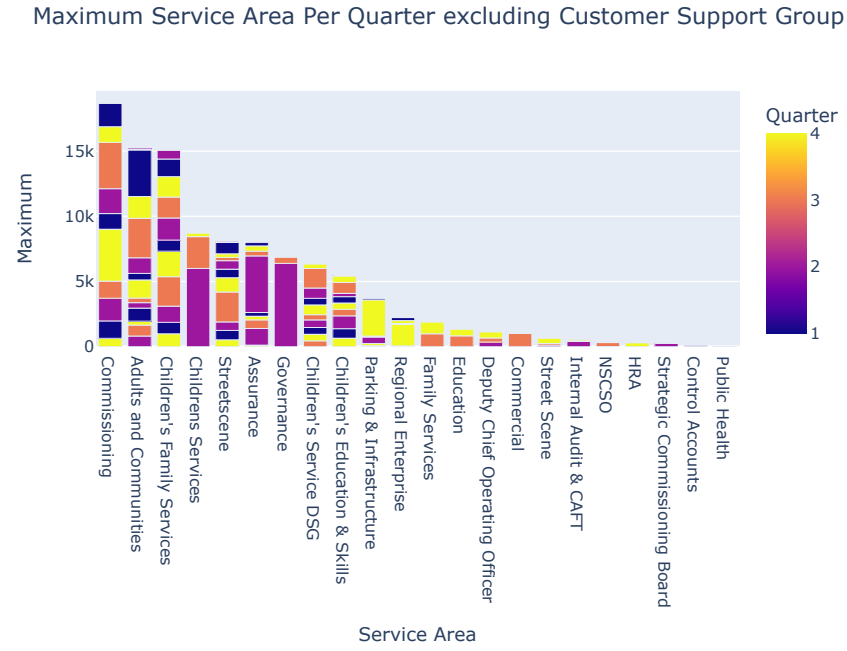
In [41]:
```
service_area_remake_reset=service_area_remake.reset_index()

df['Quarter'] = service_area_remake_reset['Quarter'].astype(str)

fig = px.bar(service_area_remake_reset, x='Service Area', y='Average_Total', color='Quarter',
             title='Transaction Total by Service Area Per Quarter excluding Customer Support Group and CSG Manageemnt Budget',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```



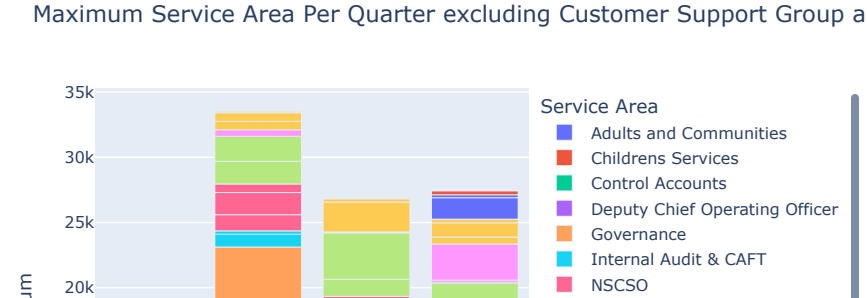


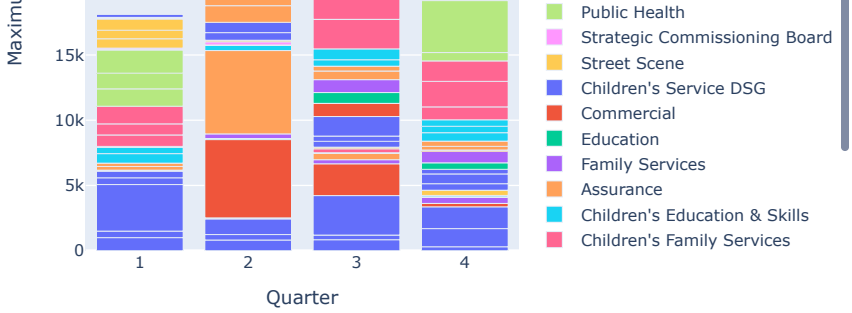
From this bar diagram we can see that average total max for all other values is 2500. It is also visible from the bar diagram that significant transaction happed in quarter 4 and quarter 2.

In [42]:
```
service_area_remake_reset=service_area_remake.reset_index()

df['Quarter'] = service_area_remake_reset['Quarter'].astype(str)

fig = px.bar(service_area_remake_reset, x='Service Area', y='Maximum', color='Quarter',
             title='Maximum Service Area Per Quarter excluding Customer Support Group and CSG Manageemnt Budget',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```



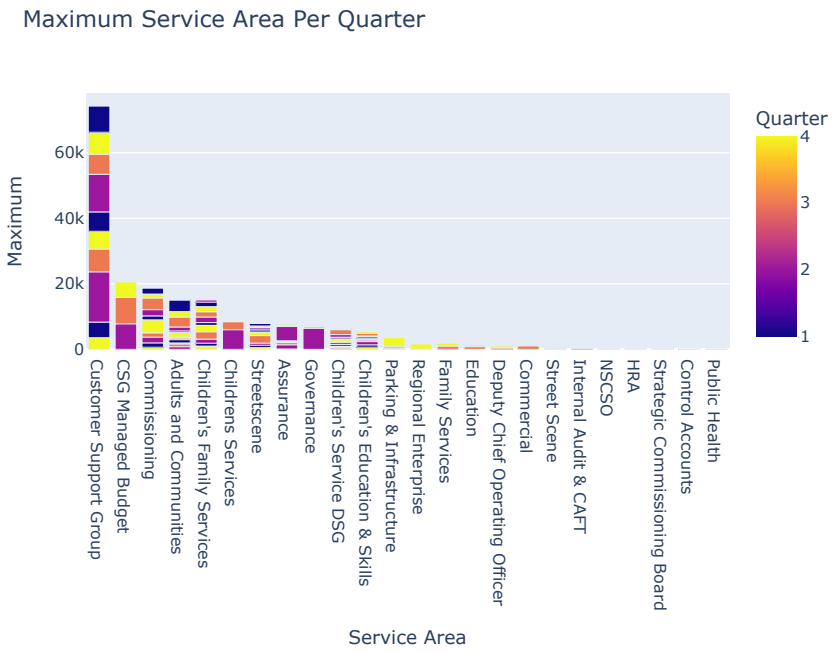
In [43]:
```
service_area_remake_reset=service_area_remake.reset_index()

df['Quarter'] = service_area_remake_reset['Quarter'].astype(str)

fig = px.bar(service_area_remake_reset, x='Quarter', y='Maximum', color='Service Area',
             title='Maximum Service Area Per Quarter excluding Customer Support Group and CSG Manageemnt Budget',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.show()
```





From this bar diagram gives the maximum value as per service area as per quarter. maximum values is above 30k under the quarter 2.

In [44]:
```
df['Quarter'] = service_area_summary_reset['Quarter'].astype(str)

fig = px.bar(service_area_summary_reset, x='Service Area', y='Maximum', color='Quarter',
             title='Maximum Service Area Per Quarter ',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.update_layout(xaxis={'categoryorder':'total descending'})  # Sort bars by total transaction count
fig.show()
```



In [45]:
```
summary_table_quarter  = df.groupby(['Service Area','Quarter']).agg(Transaction_Count=('Total', 'count'), Average_Total=('Total', 'mean'),Ma
```

In [46]:
```
summary_table_quarter_df= pd.DataFrame(summary_table_quarter)
```

Statistical summary table per service area per quarter

In [47]:
```
summary_table_quarter_df
```

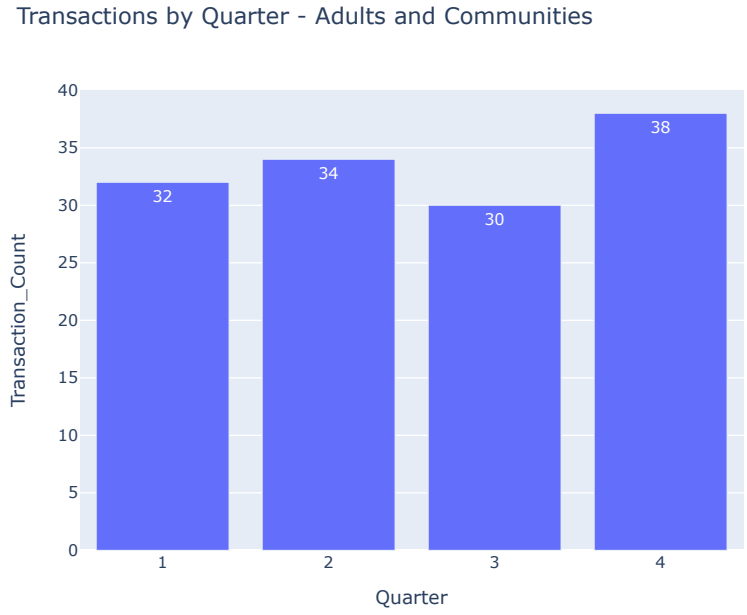
Out[47]:

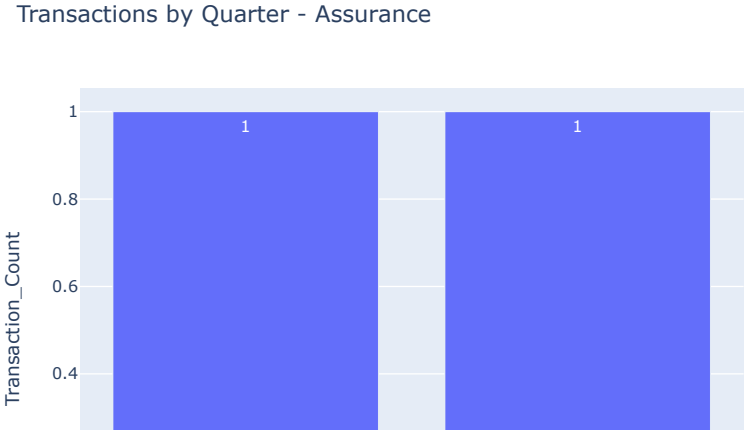
Service Area	Quarter	Transaction_Count	Average_Total	Maximum	Minimum	Total_Sum
Adults and Communities	**1**	32	277.007500	1670.30	2.00	8864.24
	2	34	189.918235	3569.03	-16.22	6457.22
	3	30	130.414333	3028.20	-15.97	3912.43
	4	38	58.614474	830.00	4.99	2227.35
Assurance	**2**	1	2.000000	2.00	2.00	2.00
	3	1	5.830000	5.83	5.83	5.83
Children's Education & Skills	**1**	3	256.690000	374.49	173.58	770.07
	2	2	18.655000	25.34	11.97	37.31
	4	6	294.831667	500.00	7.19	1768.99
Children's Family Services	**1**	22	53.101364	235.70	-10.64	1168.23
	2	26	112.133846	751.75	-9.25	2915.48
	3	26	51.562308	341.33	-751.75	1340.62
	4	22	32.122727	107.95	2.99	706.70
Children's Service DSG	**3**	1	480.000000	480.00	480.00	480.00

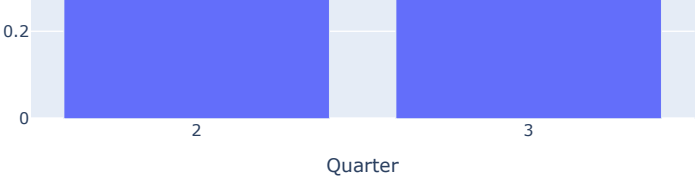
Childrens Services	**1**	22	44.190455	349.73	-23.40	972.19
	2	25	296.201600	6000.00	0.99	7405.04
	3	26	84.180769	600.00	-6.90	2188.70
	4	30	72.387667	500.00	1.90	2171.63
Commissioning	**1**	1	141.250000	141.25	141.25	141.25
	3	1	114.330000	114.33	114.33	114.33
	4	10	114.330000	114.33	114.33	1143.30
Control Accounts	**1**	4	9.062500	15.99	3.99	36.25
	4	3	30.173333	83.31	3.06	90.52
Customer Support Group	**3**	1	114.000000	114.00	114.00	114.00
Deputy Chief Operating Officer	**1**	1	10.000000	10.00	10.00	10.00
	4	1	10.000000	10.00	10.00	10.00
Governance	**1**	1	6388.200000	6388.20	6388.20	6388.20
Street Scene	**3**	3	26.633333	63.72	4.20	79.90
	4	1	100.000000	100.00	100.00	100.00

The table gives the comprehensive summary of transaction for each service area across four different quarters. This table is also a base for creating visual representation. This can help the auditor to identify the trends, visualize the anomalies and understand the underlying pattern of data. We have the visual representation of transaction count,average total and maximum for each individual service areas across the quarters. Users can get more information of the data, by hovering over the graph.

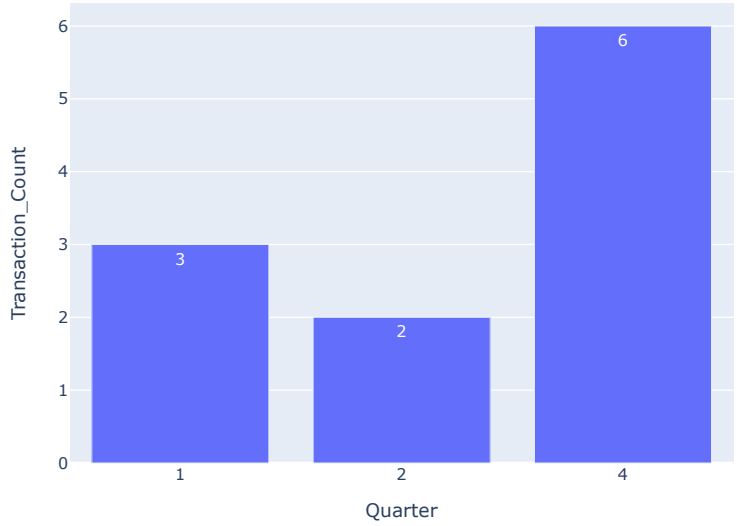
In [48]:
```
summary = summary_table_quarter_df.reset_index()

# Visualize transaction count by quarter for each service area
for service_area in summary['Service Area'].unique():
    service_area_data = summary[summary['Service Area'] == service_area]
    fig = px.bar(service_area_data, x='Quarter', y='Transaction_Count',text_auto=True,
             title=f'Transactions by Quarter - {service_area}',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
    fig.show()
```



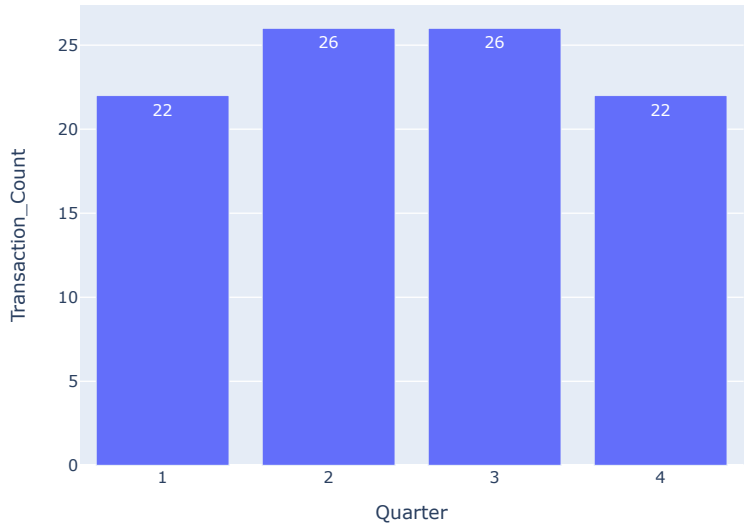




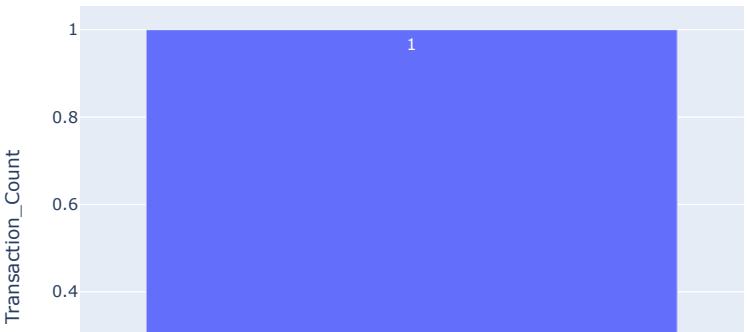
Transactions by Quarter - Children's Education & Skills

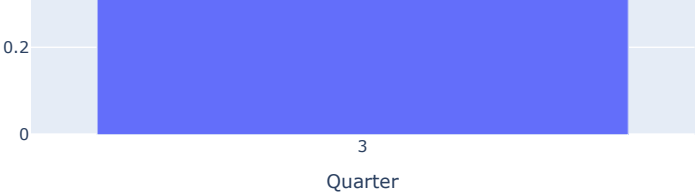


Transactions by Quarter - Children's Family Services

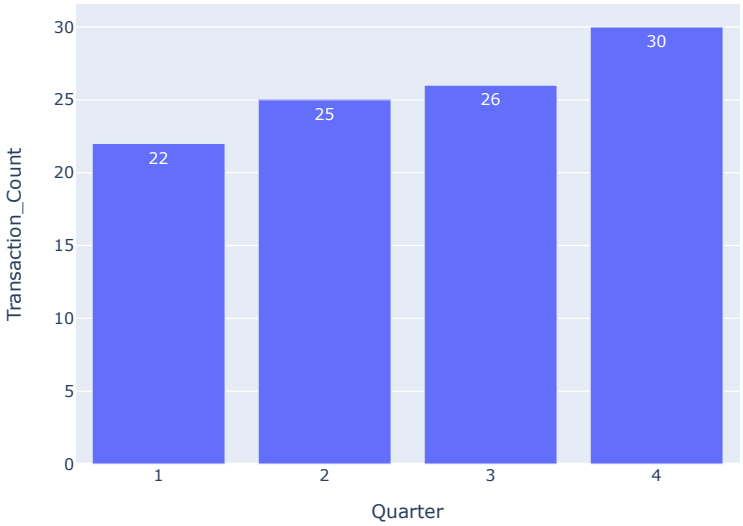


Transactions by Quarter - Children's Service DSG

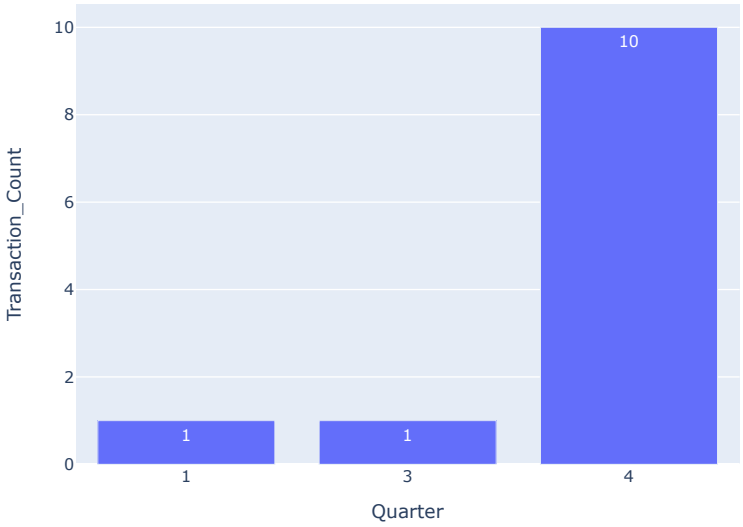




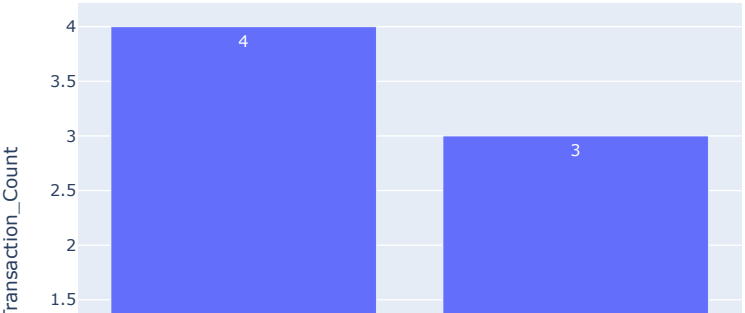
Transactions by Quarter - Childrens Services



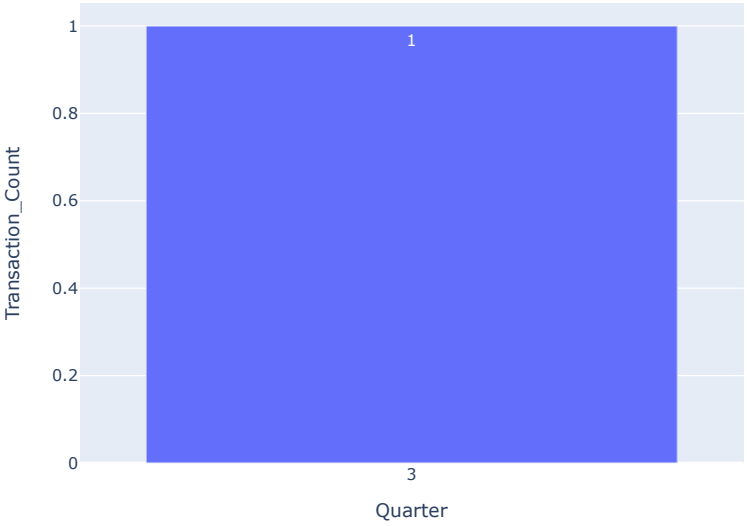
Transactions by Quarter - Commissioning



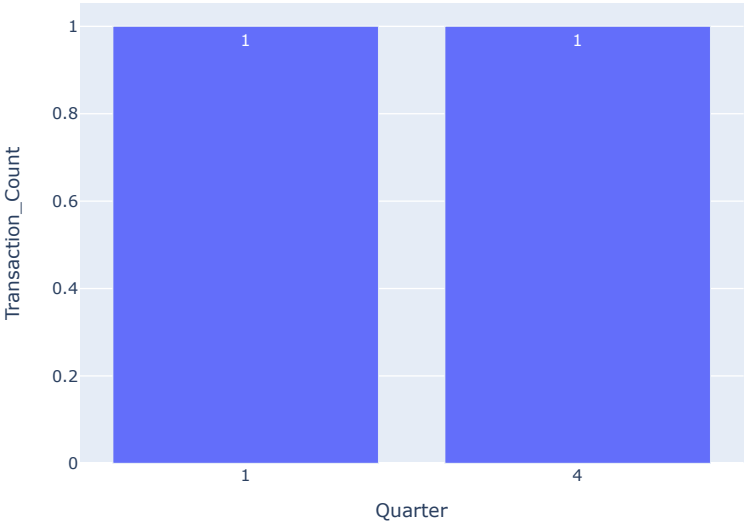
Transactions by Quarter - Control Accounts



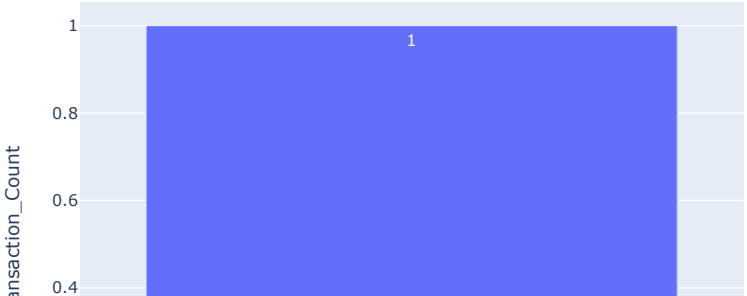
Transactions by Quarter - Customer Support Group



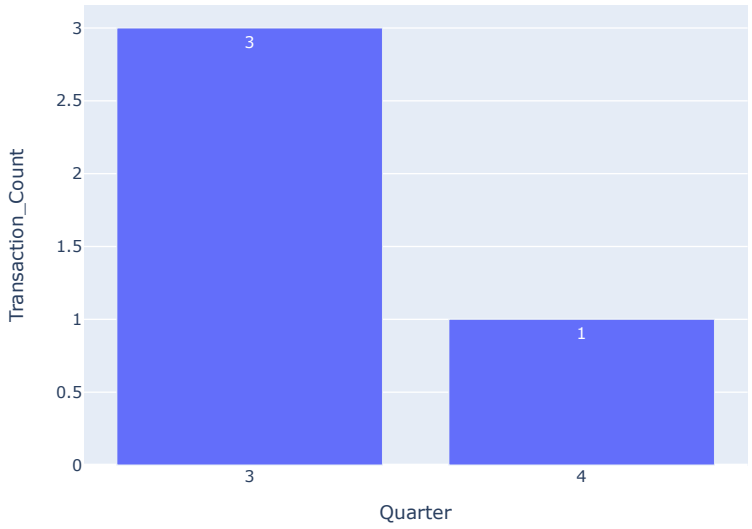
Transactions by Quarter - Deputy Chief Operating Officer



Transactions by Quarter - Governance

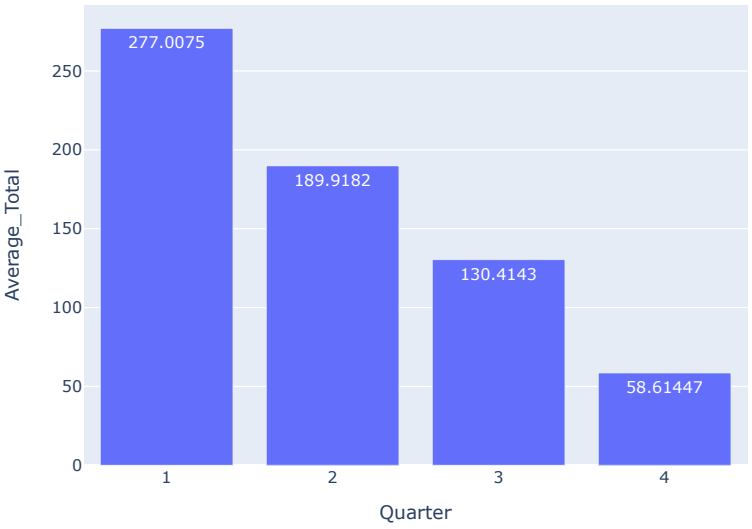


Transactions by Quarter - Street Scene

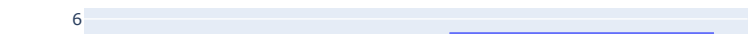


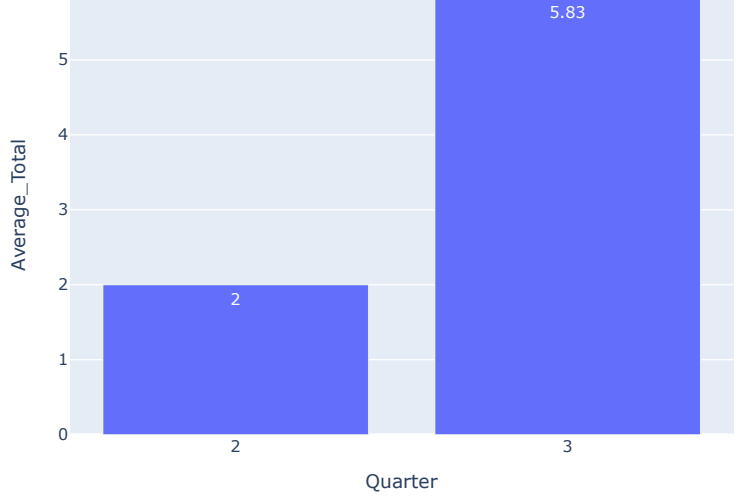
In [49]:
```
# Group by 'Service Area' and 'Quarter', and calculate statistics
summary = summary_table_quarter_df.reset_index()

# Visualize transactions by quarter for each service area
for service_area in summary['Service Area'].unique():
    service_area_data = summary[summary['Service Area'] == service_area]
    fig = px.bar(service_area_data, x='Quarter', y='Average_Total',text_auto=True,
             title=f'Transactions by Average - {service_area}',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
    fig.show()
```

Transactions by Average - Adults and Communities

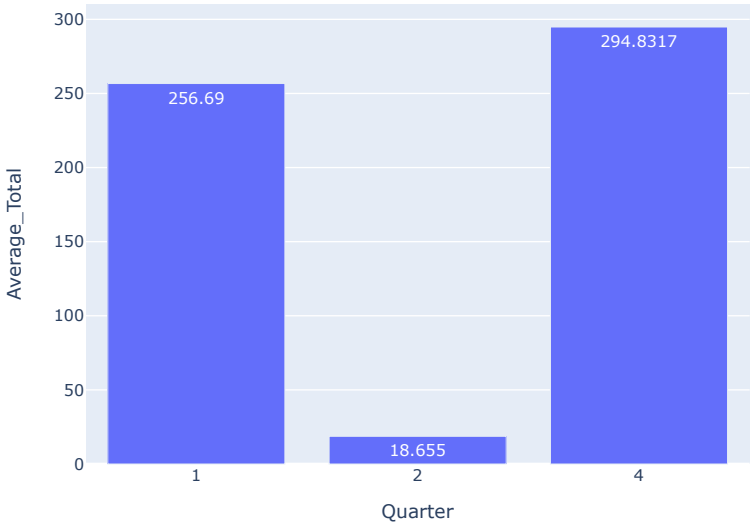


Transactions by Average - Assurance

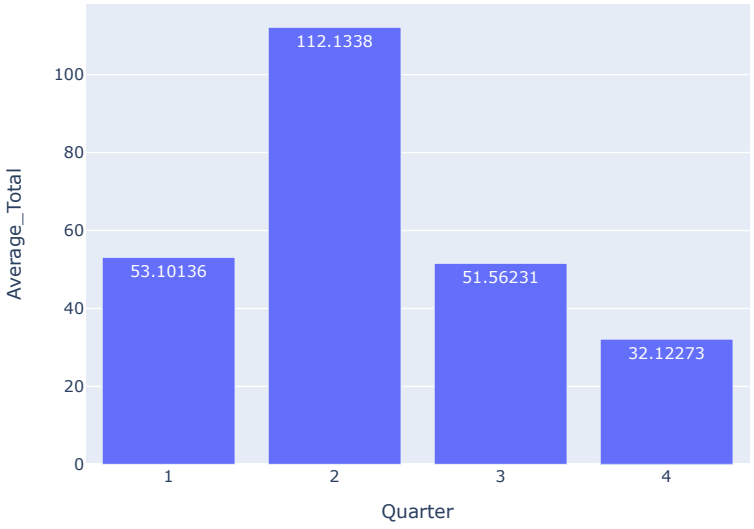




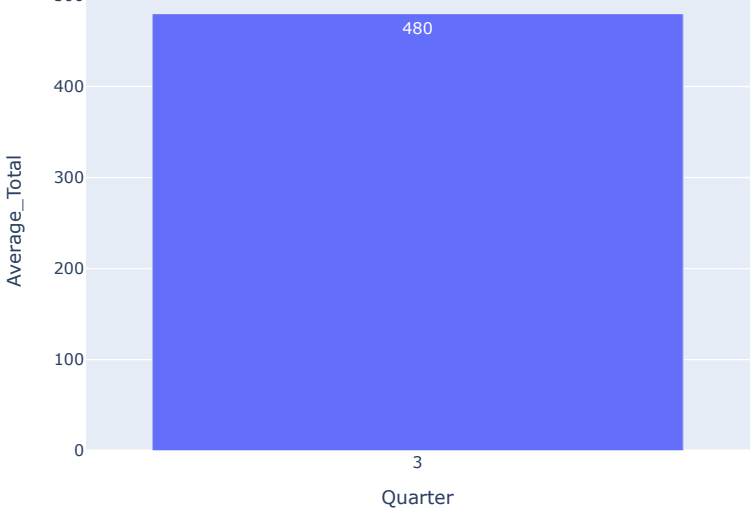
Transactions by Average - Children's Education & Skills



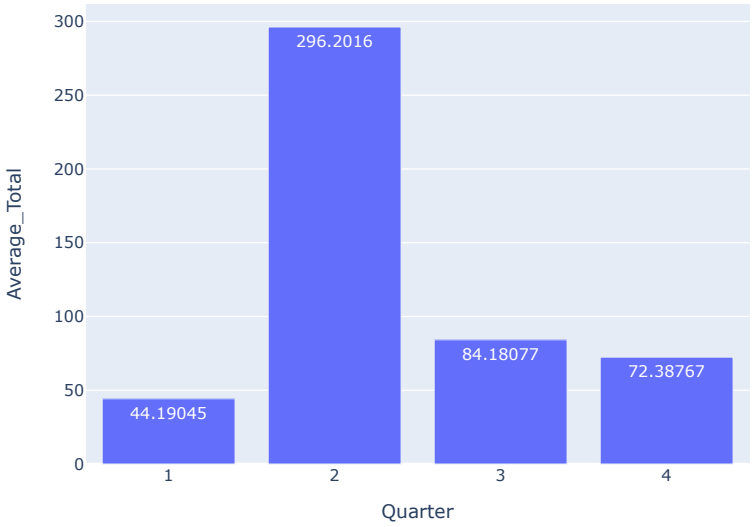
Transactions by Average - Children's Family Services



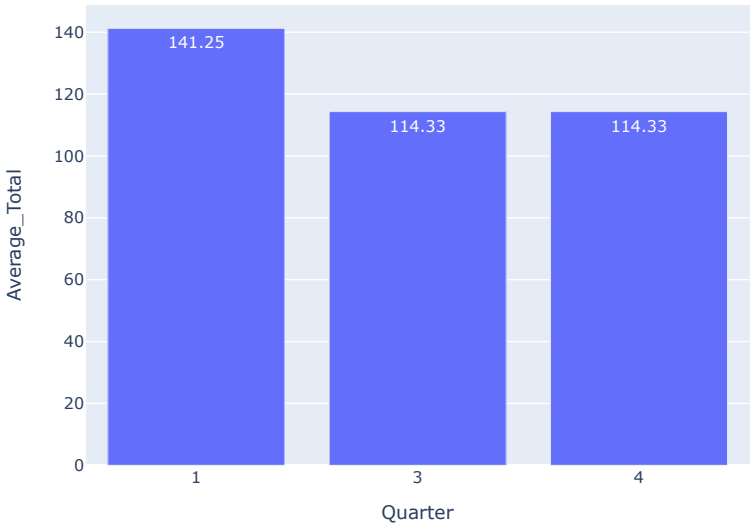
Transactions by Average - Children's Service DSG



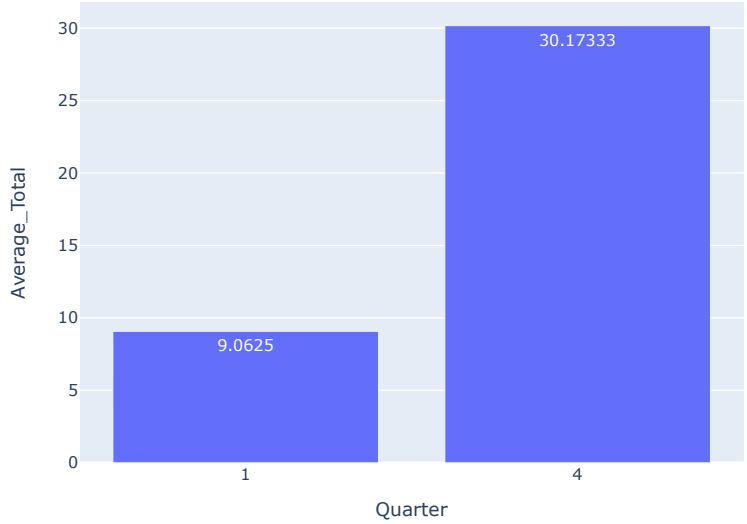
Transactions by Average - Childrens Services



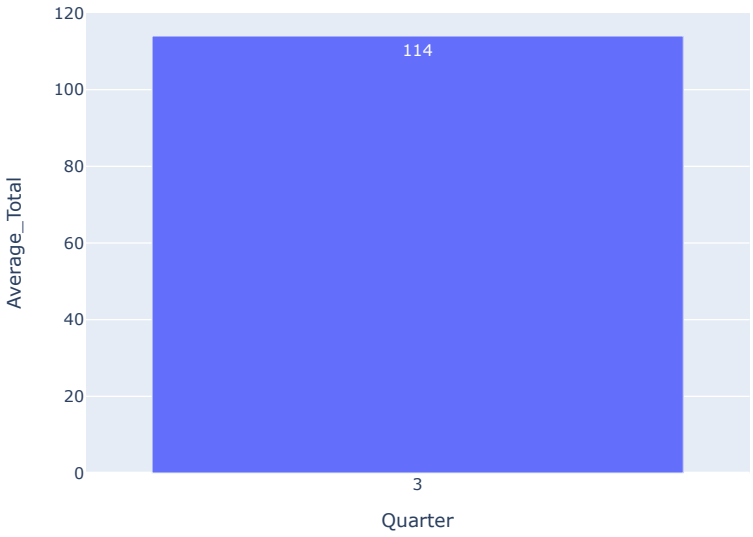
Transactions by Average - Commissioning



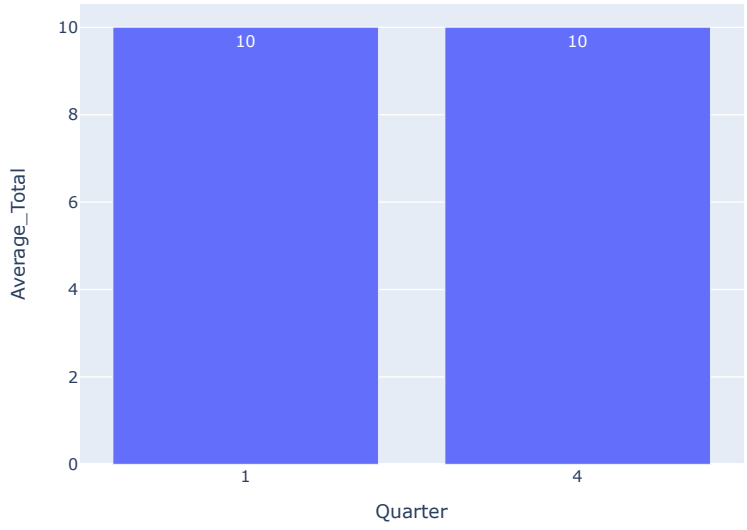
Transactions by Average - Control Accounts



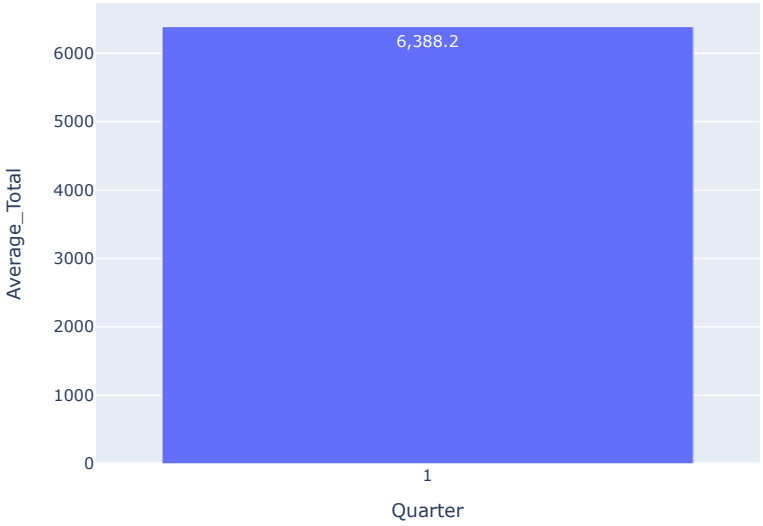
Transactions by Average - Customer Support Group



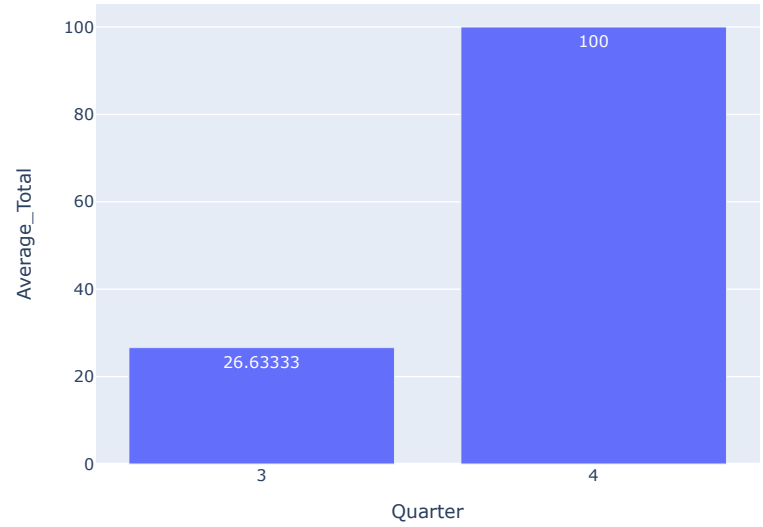
Transactions by Average - Deputy Chief Operating Officer



Transactions by Average - Governance

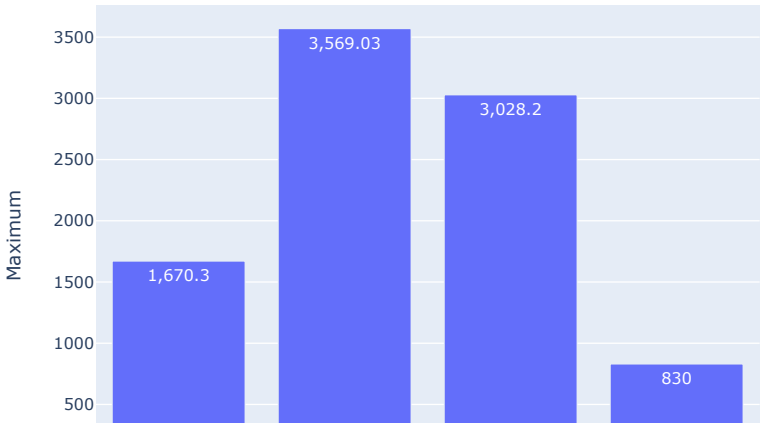


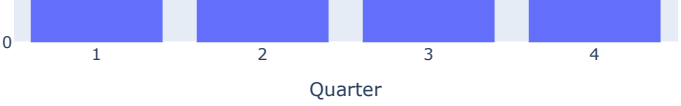
Transactions by Average - Street Scene



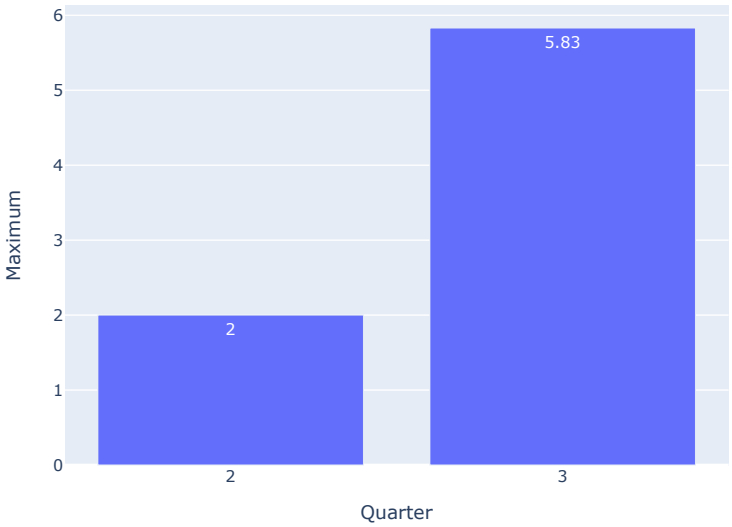
In [50]:
```
# Group by 'Service Area' and 'Quarter', and calculate statistics
summary = summary_table_quarter_df.reset_index()

# Visualize transactions by quarter for each service area
for service_area in summary['Service Area'].unique():
    service_area_data = summary[summary['Service Area'] == service_area]
    fig = px.bar(service_area_data, x='Quarter', y='Maximum',
             title=f'Maximum Transactions by service_area per quarter for  - {service_area}',
             barmode='group',text_auto=True)
    fig.show()
```

Maximum Transactions by service_area per quarter for - Adults and Com

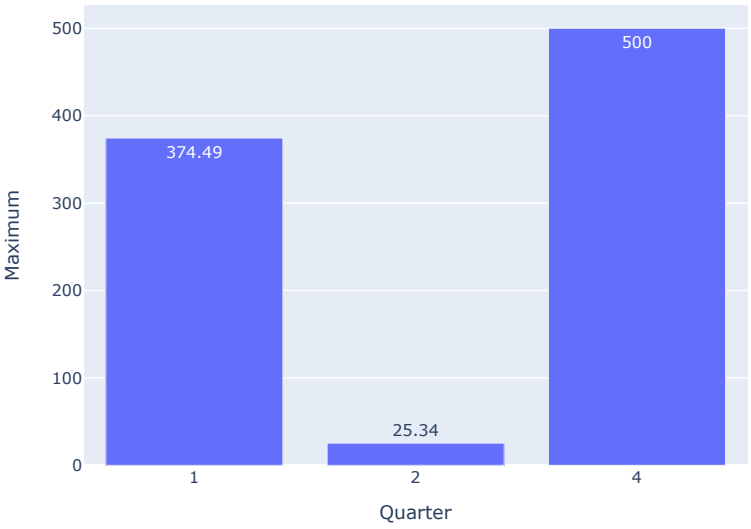




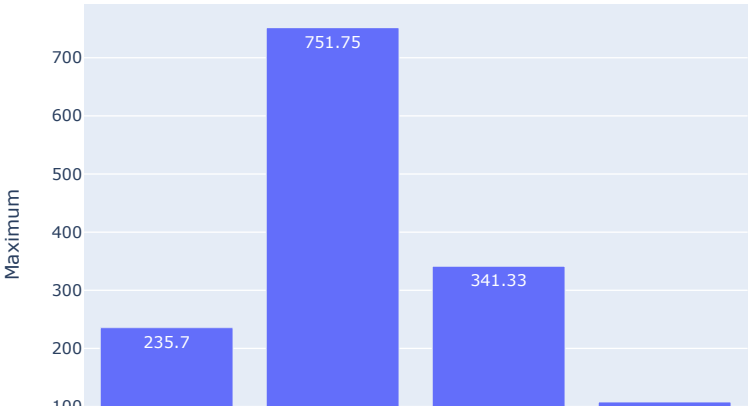
Maximum Transactions by service_area per quarter for - Assurance

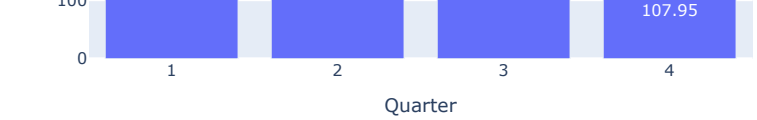


Maximum Transactions by service_area per quarter for - Children's Educ

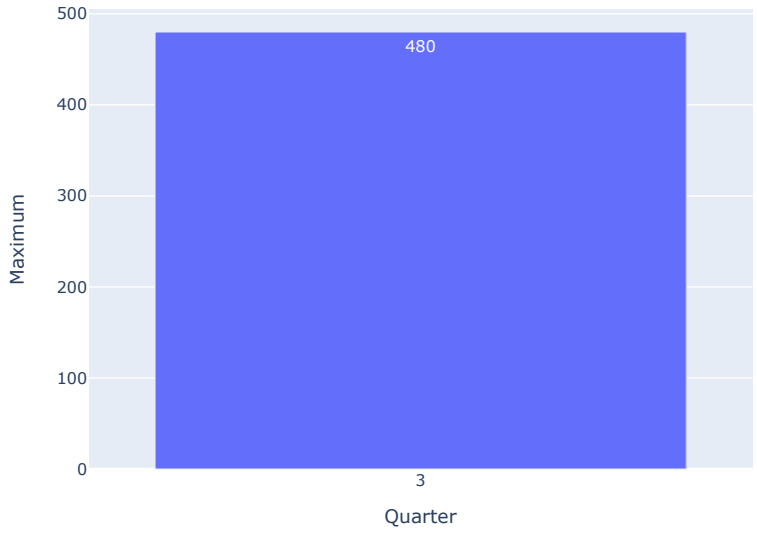


Maximum Transactions by service_area per quarter for - Children's Fami

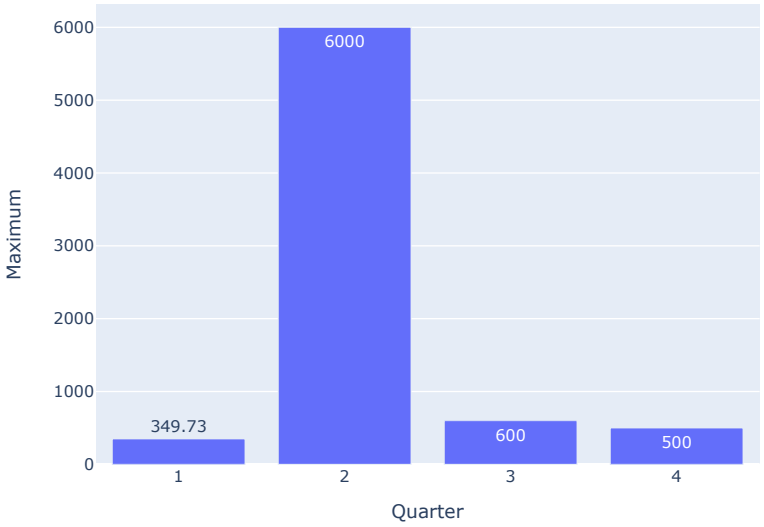




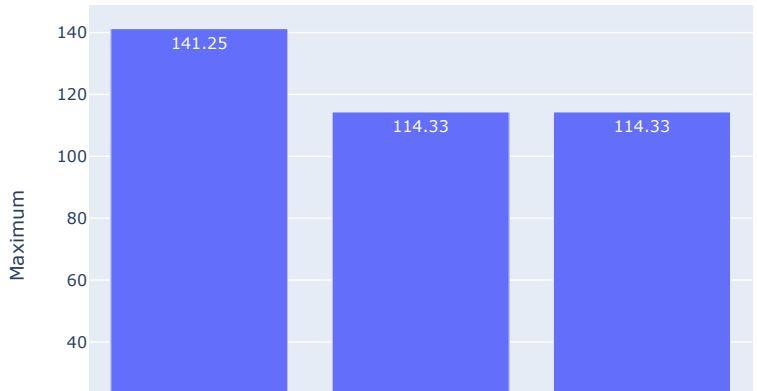
Maximum Transactions by service_area per quarter for - Children's Servi

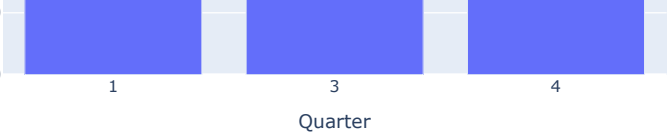


Maximum Transactions by service_area per quarter for - Childrens Servi

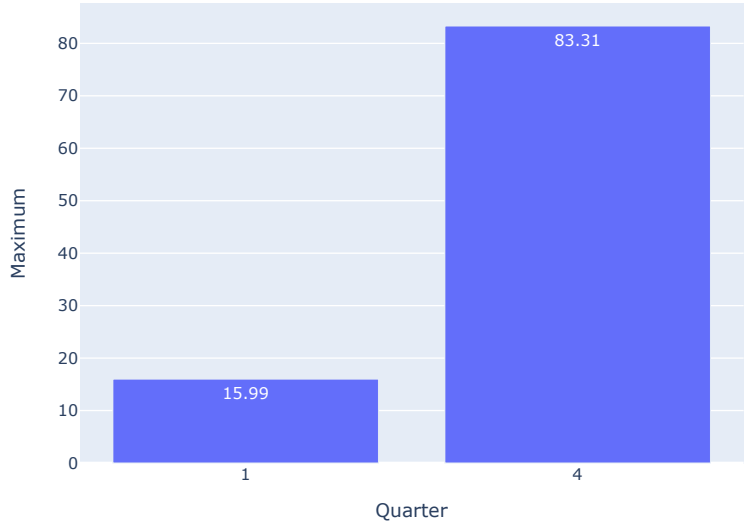


Maximum Transactions by service_area per quarter for - Commissioning

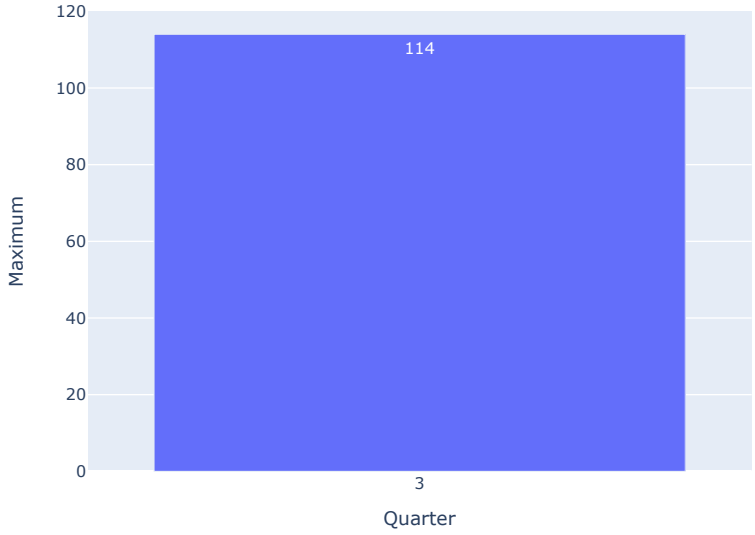




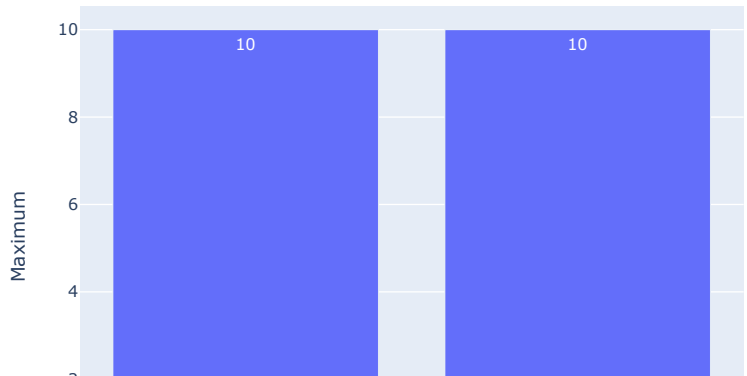
Maximum Transactions by service_area per quarter for - Control Accounl

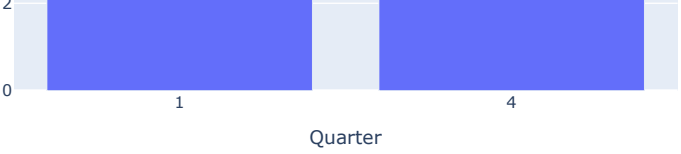


Maximum Transactions by service_area per quarter for - Customer Supp

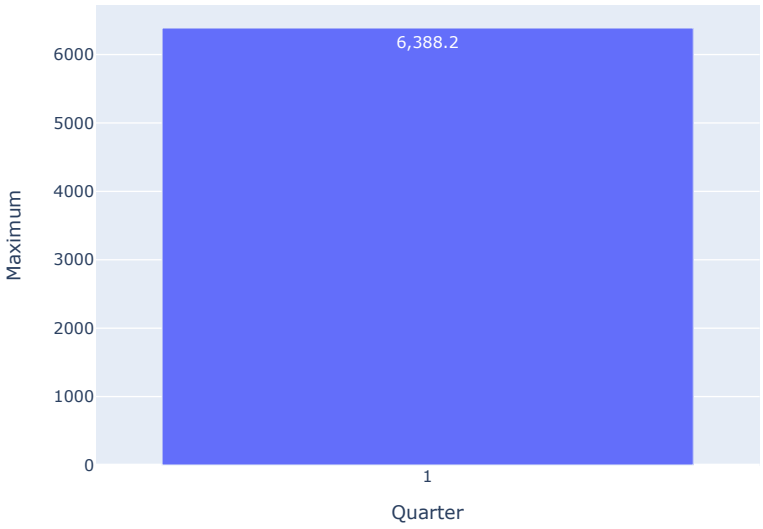


Maximum Transactions by service_area per quarter for - Deputy Chief O

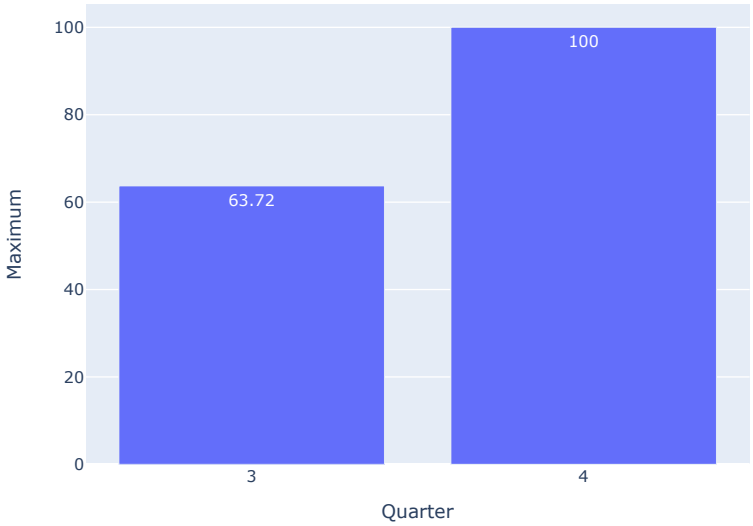




Maximum Transactions by service_area per quarter for - Governance



Maximum Transactions by service_area per quarter for - Street Scene



Task 2

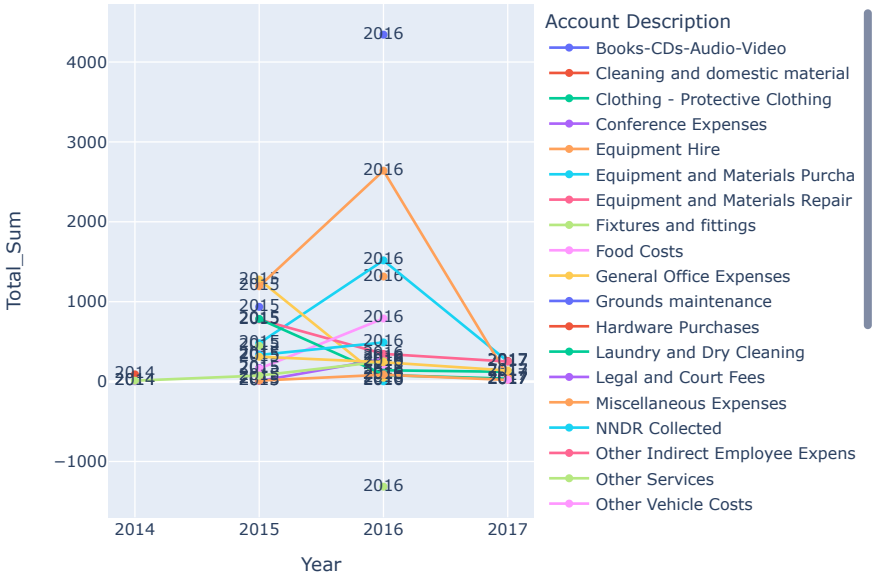
Data Preparation

In [51]:
```
#group together data by service area , accound description and year
dfs= df.groupby(['Service Area','Account Description','Year']).agg(Total_Sum=('Total','sum'))
```

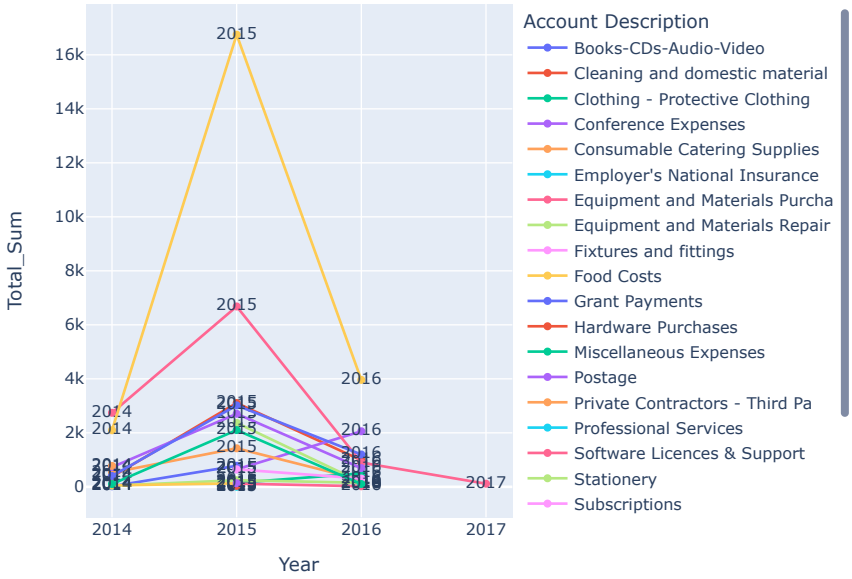
In [52]:
```
#reset the index of grouped data
dfsn= dfs.reset_index()
```

In [53]:
```
#function to visualize the spending behaviour
def view_spike(service_name):
    return(px.line(dfsn[dfsn['Service Area']==service_name],x='Year',y='Total_Sum',text="Year",color='Account Description'))
```

In [54]:
```
view_spike('Assurance')
```



In [55]:
```
view_spike("Children's Education & Skills")
```



The attempt to understand the spending behaviour trends based on service area and account description from the chart proved to be vague and obscure. Alternatively , z-score analysis is performed to understand the spending behaviour of service area and account.

In [56]:
```
#grouping data based on both quarter and year
Acc_spike= df.groupby(['Service Area','Account Description','Quarter','Year']).agg(Total_Expense=('Total', 'sum')).sort_values(by='Total_Expe
```

In [57]:
```
Acc_spike['Total_Expense']=Acc_spike['Total_Expense'].astype(int)
```

In [58]:
```
#reset the index of grouped dataframe
Acc_spike.reset_index()

#calculate mean and standard deviation to determine z score
total_exp_mean = Acc_spike['Total_Expense'].mean()
total_exp_std = Acc_spike['Total_Expense'].std()

#calculate the z score
Acc_spike['z_score'] = (Acc_spike['Total_Expense'] - int(total_exp_mean))/int(total_exp_std)

#determining our threshold value
th = 1.5

#creating the column to store the z-value
Acc_spike['spike'] = Acc_spike['z_score'].abs() > th

#creating the df to display the spiked data
spikes = Acc_spike[Acc_spike['spike']]
```

```
#display the dataframe
spikes
```

Out[58]:

				Total_Expense	z_score	spike
Service Area	**Account Description**	**Quarter**	**Year**			
Governance	**Other Services**	**1**	**2014**	6388	6.189441	True
Childrens Services	**Other Services**	**2**	**2014**	6000	5.787785	True
Adults and Communities	**Other Agencies - Third Party P**	**1**	**2016**	4125	3.846791	True
	Electricity	**2**	**2017**	3569	3.271222	True
	Rents	**3**	**2016**	3028	2.711180	True
	Other Agencies - Third Party P	**1**	**2017**	2354	2.013458	True
Children's Family Services	**Equipment and Materials Purcha**	**2**	**2015**	1994	1.640787	True

Z-score tells us how much standard deviation far is data from the mean of the distribution. Positive z-score shows value is above the mean while the negative score shows data is below the mean value.

In our data, z-score can tell how much significant our data deviates from the mean value which also indicates spike or permanent increase in spending behaviour. From the above diagram we can tell that , Service Area : Governance and Childrens Service under other services in 2014 quarter 2 has z score of 5.84 and 5.46 respectively which shows significant increase from the mean value indicating spike. Similarly , we can see spike in adults and communities in the year 2016 and 2017 suggesting noteworthy outliers.

Task 3

In [59]:
```
#groupe data based on creditor and account description
Acc_creditor = df.groupby(['Creditor', 'Account Description']).size().reset_index(name='Count')

print('Shape of df:', Acc_creditor.shape)
print('Shape with count equals to 1:', Acc_creditor[Acc_creditor['Count']==1].shape)
print('Shape with count less than 10:', Acc_creditor[Acc_creditor['Count']<5].shape)
print('Shape with count greater than 10:', Acc_creditor[Acc_creditor['Count']>5].shape)
```

```
Shape of df: (2875, 3)
Shape with count equals to 1: (1758, 3)
Shape with count less than 10: (2497, 3)
Shape with count greater than 10: (317, 3)
```

In [60]:
```
#display 10 data from Acc_creditor dataframe
Acc_creditor[:10]
```

Out[60]:

	Creditor	Account Description	Count
0	ARGOS	Other Transfer Payments to Soc	1
1	COFFEE REPUBLIC WOO	Food Costs	1
2	COSTCUTTER	Food Costs	1
3	H HARIA CHEMIST	Other Transfer Payments to Soc	1
4	LEWISS	Equipment and Materials Purcha	1
5	SAINSBURYS S/MKTS	Food Costs	1
6	SAVERS	Other Transfer Payments to Soc	1
7	STUDEN PHOTOCARD	Travelling Expenses	2
8	Sainsburys S/mkts	Food Costs	1
9	TESCO PFS 2473	Food Costs	2

In [61]:
```
#checks for duplicate entries
misclassified_creditors = Acc_creditor[Acc_creditor['Creditor'].duplicated(keep=False)]
misclassified_creditors.shape
```

Out[61]:
```
(1413, 3)
```

By executing above code we can identify instances where the same Creditor value appears in multiple Account description, which suggests the potential misclassification of transaction in our data. The duplicated function along with the parameter keep=False returns boolean value where true indicates the presence of more than one creditor value in our grouped dataframe.

In [62]:
```
misclassified_creditors[:20]
```

Out[62]:

	Creditor	Account Description	Count
15	123-REG.CO.UK	IT Services	1
16	123-REG.CO.UK	Subscriptions	1
22	99P STORES LTD	E19 - Learning Resources	1
23	99P STORES LTD	Equipment and Materials Purcha	2
25	A&Y LOCKSMITHS	Clothing - Protective Clothing	1
26	A&Y LOCKSMITHS	Miscellaneous Expenses	1
35	ABLE GROUP UK	Private Contractors - Third Pa	1

36	ABLE GROUP UK	Professional Services	1
38	ACCESS EXPEDITIONS	Equipment and Materials Purcha	1
39	ACCESS EXPEDITIONS	Other Services	1
50	AFE SERVICELINE	Equipment and Materials Purcha	1
51	AFE SERVICELINE	Equipment and Materials Repair	10
52	AFE SERVICELINE	Private Contractors - Third Pa	2
54	AFFINITY WATER LTD	Equipment and Materials Purcha	1
55	AFFINITY WATER LTD	Miscellaneous Expenses	1
56	AFFINITY WATER LTD	Water Services	1
58	ALDI	Food Costs	1
59	ALDI	Training	12
61	ALEXANDRA PALACE	Other Services	1
62	ALEXANDRA PALACE	Venue Hire	1

In [63]:
```
df[df['Creditor']=='STUDEN PHOTOCARD'] #total sample data
```

Out[63]:

	Service Area	Account Description	Creditor	Journal Date	Total	Quarter	Year
107	Childrens Services	Travelling Expenses	STUDEN PHOTOCARD	2014-04-02	10.0	4	2014
256	Childrens Services	Travelling Expenses	STUDEN PHOTOCARD	2014-04-24	10.0	NaN	2014
257	Childrens Services	Travelling Expenses	STUDEN PHOTOCARD	2014-04-21	10.0	NaN	2014
999	Childrens Services	Travelling Expenses	STUDEN PHOTOCARD	2014-06-16	10.0	NaN	2014
1336	Childrens Services	Other Transfer Payments to Soc	STUDEN PHOTOCARD	2014-07-16	10.0	NaN	2014
1964	Family Services	Travelling Expenses	STUDEN PHOTOCARD	2014-09-30	10.0	NaN	2014
2825	Children's Family Services	Travelling Expenses	STUDEN PHOTOCARD	2014-11-12	10.0	NaN	2014
3175	Children's Family Services	Travelling Expenses	STUDEN PHOTOCARD	2014-12-15	10.0	NaN	2014
3556	Children's Family Services	Other Transfer Payments to Soc	STUDEN PHOTOCARD	2015-01-09	10.0	NaN	2015

In [64]:
```
misclassified_creditors[misclassified_creditors['Creditor']=='STUDEN PHOTOCARD'] #misclassified
```

Out[64]:

	Creditor	Account Description	Count
2052	STUDEN PHOTOCARD	Other Transfer Payments to Soc	2
2053	STUDEN PHOTOCARD	Travelling Expenses	7

Task 4

There are different clustering techniques which can be used to cluster our data to understand the underlying pattern . One of the most common clustering technique is KMeans. KMeans require the number of clusters to be defined prior while the another technique mean shift clustering doesnot require to specify the number of clusters. The major challenge with KMeans is accurately defining the number of clusters hence I chose mean shift clustering technique. Mean shift clustering is the density based techniques which can identify the number of clusters with irregular shapes.

In [65]:
```
scaler = StandardScaler()
clustering_df = df.groupby('Service Area').agg(Transaction_Count=('Total', 'count')).reset_index()
clustering_df_scaled = scaler.fit_transform(clustering_df[['Transaction_Count']])

# Applying Mean Shift clustering
bandwidths = [0.05,0.08]
for bandwidth in bandwidths:
    meanshift = MeanShift(bandwidth=bandwidth)
    clustering_df['Cluster'] = meanshift.fit_predict(clustering_df_scaled)

# Results
clustering_df
```

Out[65]:

	Service Area	Transaction_Count	Cluster
0	Adults and Communities	273	1
1	Assurance	340	1
2	CSG Managed Budget	35	0
3	Children's Education & Skills	467	1
4	Children's Family Services	7435	2
5	Children's Service DSG	275	1
6	Childrens Services	1215	3
7	Commercial	9	0
8	Commissioning	383	1
9	Control Accounts	8	0
10	Customer Support Group	110	0
11	Deputy Chief Operating Officer	112	0

12	Education	95	0
13	Family Services	728	4
14	Governance	7	0
15	HRA	1	0
16	Internal Audit & CAFT	11	0
17	NSCSO	3	0
18	Parking & Infrastructure	12	0
19	Public Health	3	0
20	Regional Enterprise	6	0
21	Strategic Commissioning Board	1	0
22	Street Scene	39	0
23	Streetscene	293	1

In [66]:
```
clustering_df['Cluster'].value_counts()
```

Out[66]:
```
0    15
1     6
2     1
3     1
4     1
Name: Cluster, dtype: int64
```

In [67]:
```
clustering_df.loc[clustering_df['Cluster'].isin([3, 4]), 'Cluster'] = 2
```

In [68]:
```
#values in each cluster
clustering_df['Cluster'].value_counts()
```

Out[68]:
```
0    15
1     6
2     3
Name: Cluster, dtype: int64
```

Our algorithm success fully classified our service area into 5 clusters. Since the last 3 clusters had only 1 value in each of them i merged all three of them for simplicity. We have total of three cluster where , cluster 1 has 15 service area, 2 has 6 and 3 has 3 service areas. Service area with similar transactional behaviour are clustered together.The details of each cluster is shown below with visual representation.

In [69]:
```
cluster1= clustering_df[clustering_df['Cluster']==0].sort_values(by='Transaction_Count', ascending=False)
cluster1
```

Out[69]:

	Service Area	Transaction_Count	Cluster
11	Deputy Chief Operating Officer	112	0
10	Customer Support Group	110	0
12	Education	95	0
22	Street Scene	39	0
2	CSG Managed Budget	35	0
18	Parking & Infrastructure	12	0
16	Internal Audit & CAFT	11	0
7	Commercial	9	0
9	Control Accounts	8	0
14	Governance	7	0
20	Regional Enterprise	6	0
17	NSCSO	3	0
19	Public Health	3	0
15	HRA	1	0
21	Strategic Commissioning Board	1	0

In [70]:
```
cluster2= clustering_df[clustering_df['Cluster']==1].sort_values(by='Transaction_Count', ascending=False)
cluster2
```

Out[70]:

	Service Area	Transaction_Count	Cluster
3	Children's Education & Skills	467	1
8	Commissioning	383	1
1	Assurance	340	1
23	Streetscene	293	1
5	Children's Service DSG	275	1
0	Adults and Communities	273	1

In [71]:
```
cluster3= clustering_df[clustering_df['Cluster']==2].sort_values(by='Transaction_Count', ascending=False)
cluster3.reset_index()
```

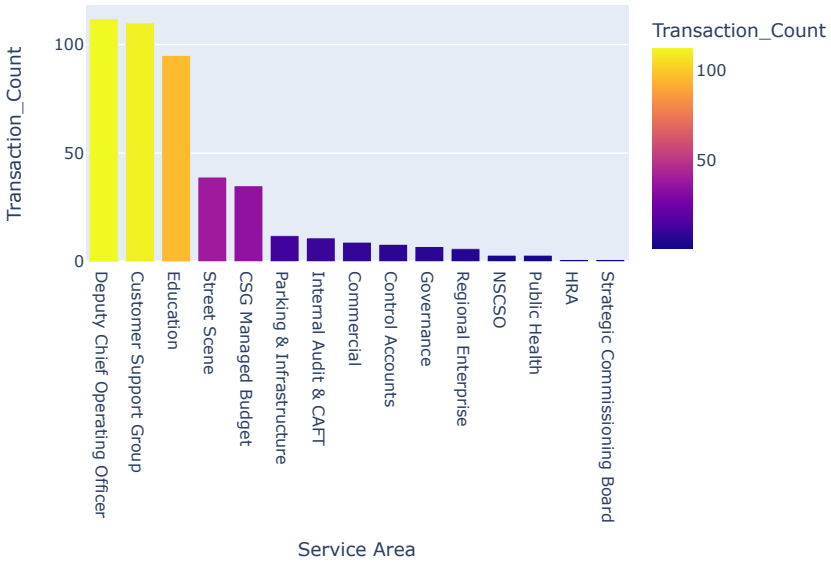
Out[71]:

	index	Service Area	Transaction_Count	Cluster

0	4	Children's Family Services	7435	2
1	6	Childrens Services	1215	2
2	13	Family Services	728	2

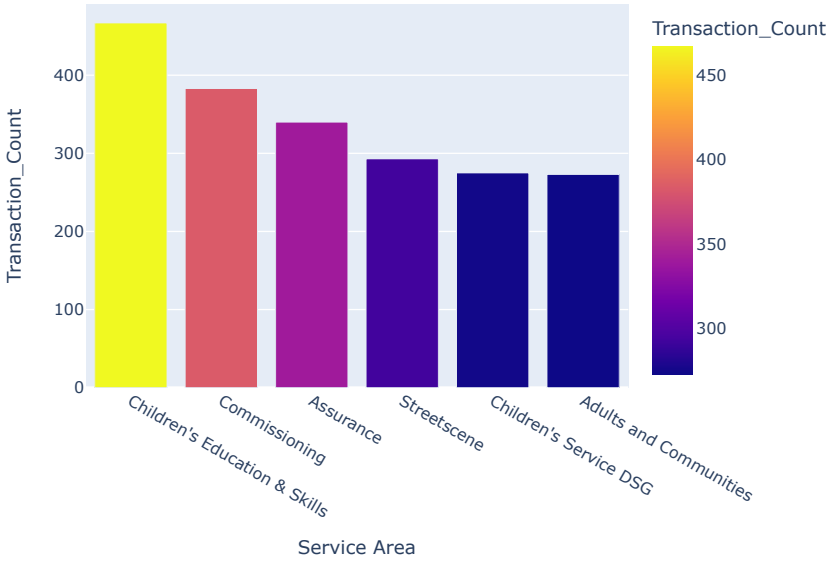
In [72]:
```
fig = px.bar(cluster1.reset_index(), x='Service Area', y='Transaction_Count', color='Transaction_Count',
             title='Cluster 1',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area'})
fig.show()
```

Cluster 1



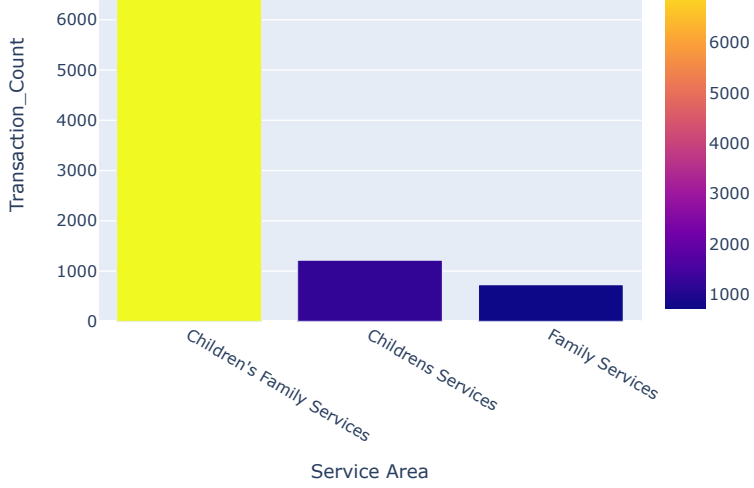
In [73]:
```
fig = px.bar(cluster2.reset_index(), x='Service Area', y='Transaction_Count', color='Transaction_Count',
             title='Cluster 2',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area'})
fig.show()
```

Cluster 2



In [74]:
```
fig = px.bar(cluster3.reset_index(), x='Service Area', y='Transaction_Count', color='Transaction_Count',
             title='Cluster 3',
             labels={'Average Total': 'Average_Total', 'Service Area': 'Service Area', 'Quarter': 'Quarter'})
fig.show()
```

Cluster 3



Task 5

In [75]:
```
df.head()
```

Out[75]:

	Service Area	Account Description	Creditor	Journal Date	Total	Quarter	Year
0	Adults and Communities	Books-CDs-Audio-Video	AMAZON EU	2016-12-05	45.00	2	2016
1	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK MARKETPLACE	2016-12-05	426.57	2	2016
2	Adults and Communities	Books-CDs-Audio-Video	AMAZON UK RETAIL AMAZO	2016-12-06	121.38	2	2016
3	Adults and Communities	Consumable Catering Supplies	WWW.ARGOS.CO.UK	2017-03-01	78.94	2	2017
4	Adults and Communities	CSG - IT	AMAZON UK MARKETPLACE	2017-02-01	97.50	2	2017

In [76]:
```
#Anomaly detection based on each service area
df['Service Area'].unique() #gives us the name of each service area
```

Out[76]:
```
array(['Adults and Communities', 'Assurance',
       "Children's Education & Skills", "Children's Family Services",
       "Children's Service DSG", 'Commissioning',
       'Customer Support Group', 'HRA', 'Parking & Infrastructure',
       'Public Health', 'Regional Enterprise', 'Streetscene',
       'Childrens Services', 'Control Accounts', 'Street Scene',
       'Governance', 'Deputy Chief Operating Officer',
       'Internal Audit & CAFT', 'NSCSO', 'CSG Managed Budget',
       'Strategic Commissioning Board', 'Family Services', 'Education',
       'Commercial'], dtype=object)
```

IQR stands for Interquartile range. It is the range between the quartile1(Q1) and quartile3(Q3)and is commonly used to identify the outliers or anomalies in the data. It is also way to understand the distribution of our data , by diving it into four parts.

Outliers in this case are those values which are below(q1 - (1.5 *iqr)) i.e lower_whisker and above (q3 + (1.5* iqr)) i.e upper_whisker.

"Calculate_whiskers" function takes the dataframe and columnname and returns the value of upper whisker and lower whisker

In [77]:
```
#determining the outliers using IQR technique
def calculate_whiskers(df,colname):
    q1 = df[colname].quantile(.25)
    q3 = df[colname].quantile(.75)
    iqr = q3 - q1
    w_multiplier = 1.5 * iqr
    lower_whisker = q1 - w_multiplier
    upper_whisker = q3 + w_multiplier
    return upper_whisker, lower_whisker
```

"find_outliers" function takes the upper whisker and lower whisker: and returns the dataframe of outliers.

In [78]:
```
def find_outliers(df):
    uw,lw = calculate_whiskers(df, 'Total')
    outliers_df = df[(df['Total'] < lw) | (df['Total'] > uw)]
    return outliers_df
```

We are implementing the for loop to pass the value of each service area from our dataframe. outliers_df_final is the final dataframe which is created by concatenating outlier values from each service area

In [79]:
```
outliers_df_final = pd.DataFrame()
for service_area in df['Service Area'].unique():
    service_area_df = df[df['Service Area'] == service_area]
    outliers_df_final = pd.concat([outliers_df_final, find_outliers(service_area_df)])
```

In [80]:
```
outliers_df_final.shape #total number of outlier values in our dataframe
```

Out[80]:
```
(1180, 7)
```

In [81]:
```
def get_samples(service_area):
    if len(service_area) >= 10:
        return service_area.sample(n=10)
    else:
        return service_area

# Apply get_samples on grouped service area
select_val = outliers_df_final.groupby('Service Area', group_keys=False).apply(get_samples)

if len(select_val) > 150:
    select_val = select_val.sample(n=150)

# Create a new DataFrame with the selected values
selected_df = pd.DataFrame(select_val)

# Reset index of the DataFrame
selected_df.reset_index(drop=True, inplace=True)

# Display the new DataFrame
selected_df[['Service Area','Account Description','Creditor','Journal Date','Total']].sort_values(by='Service Area')
```

Out[81]:

	Service Area	Account Description	Creditor	Journal Date	Total
0	Adults and Communities	Training	THE ADULT LEARNING	2016-09-19	930.00
1	Adults and Communities	Training	WWW.ADASS.ORG.UK	2016-02-04	499.00
2	Adults and Communities	Training	WWW.GOVKNOW.COM	2017-02-09	300.00
3	Adults and Communities	Other Agencies - Third Party P	HOLIDAY INNS	2016-11-22	1625.00
4	Adults and Communities	Training	WWW.ADASS.ORG.UK	2017-03-15	499.00
5	Adults and Communities	Training	EB DELIVERING INTEGRA	2015-08-28	354.00
6	Adults and Communities	Training	PREMIER INN	2016-11-07	489.05
7	Adults and Communities	Other Vehicle Costs	LBBARNET PAYENET	2014-09-04	534.45
8	Adults and Communities	Training	EB ENHANCED HEALTH IN	2016-12-01	480.00
9	Adults and Communities	Training	WWW.ADASS.ORG.UK	2016-05-30	1200.00
19	Assurance	Fixtures and fittings	MARQUEE CARPETS LIMITEWALTHAM CROSS	2016-08-19	-1315.20
18	Assurance	Other Vehicle Costs	HEARNS COACHES	2016-04-27	535.50
17	Assurance	Vehicle Running Costs	WHITE ROSE MOTORS SOUT	2015-12-22	278.40
15	Assurance	Pool Transport Charges	TRAINLINE.COM	2015-06-30	78.34
16	Assurance	Equipment and Materials Purcha	AMAZON UK MARKETPLACE	2016-11-02	117.89
13	Assurance	Miscellaneous Expenses	D H C LTD	2016-06-08	170.85
12	Assurance	Miscellaneous Expenses	ROYAL MAIL	2016-12-19	392.40
11	Assurance	Training	WWW.WESTMINSTER-BRIEFI	2016-12-09	399.60
10	Assurance	Training	PREMIER INN	2016-10-27	90.40
14	Assurance	Training	AMAZON UK RETAIL	2015-05-14	99.00
20	CSG Managed Budget	Legal and Court Fees	HMCOURTS-SERVICE.G	2014-09-23	8058.00
27	Children's Education & Skills	Books-CDs-Audio-Video	SP DOWN SYNDROME E	2015-02-10	730.44
26	Children's Education & Skills	Travelling Expenses	LBBARNET PAYENET	2015-01-16	610.80
25	Children's Education & Skills	Food Costs	COMPASS SERVICES UK	2014-11-28	648.00
23	Children's Education & Skills	Training	PENTAGON	2015-06-18	987.47
22	Children's Education & Skills	Conference Expenses	THE NOKE HOTEL	2016-09-08	833.33
21	Children's Education & Skills	Conference Expenses	THE NOKE HOTEL	2016-09-08	725.00
24	Children's Education & Skills	Food Costs	THE GRAPEVINE	2015-12-08	-301.35
34	Children's Family Services	Equipment and Materials Repair	DOMESTIC & GENERAL	2017-01-23	213.88
37	Children's Family Services	Other Transfer Payments to Soc	LOVE2REWARD.CO.UK	2015-11-12	506.00
36	Children's Family Services	Equipment and Materials Purcha	HOBBYCRAFT LTD	2016-11-25	200.00
35	Children's Family Services	Equipment and Materials Purcha	AMAZON UK MARKETPLACE	2017-02-22	299.00
33	Children's Family Services	Other Transfer Payments to Soc	AO RETAIL LIMITED	2016-12-13	407.98
31	Children's Family Services	Telephones Calls	BT PAY BY PHONE	2016-10-10	265.54
30	Children's Family Services	Advertising	FUNDING SOLUTIONS FOR	2016-11-10	660.00
29	Children's Family Services	Private Contractors - Third Pa	H A S CONSULTANTS	2015-01-14	206.83
28	Children's Family Services	Food Costs	TESCO STORES 6440	2017-03-02	191.96
32	Children's Family Services	Education CFR Administrative S	VODAFONE	2016-01-26	216.65
46	Children's Service DSG	Equipment and Materials Purcha	AA MEDIA	2015-10-23	480.00
47	Children's Service DSG	Conference Expenses	EB PRE-SCHOOL LEARNIN	2016-04-28	360.00
45	Children's Service DSG	Equipment and Materials Purcha	WWW.POSTURITE.CO.UK	2016-04-11	788.34
44	Children's Service DSG	Books-CDs-Audio-Video	OXFORDUNIVERSITYPR	2014-08-14	449.28
43	Children's Service DSG	Equipment and Materials Purcha	OXFORDUNIVERSITYPR	2014-09-22	449.28
42	Children's Service DSG	Books-CDs-Audio-Video	WP-THE BRITISH ASS	2014-11-21	375.00

41	Children's Service DSG	Travelling Expenses	METRO RADIO CARS	2016-04-11	360.00
40	Children's Service DSG	Books-CDs-Audio-Video	WP-THE BRITISH ASS	2014-12-11	500.00
39	Children's Service DSG	Training	PAYATRADER	2015-01-08	520.00
38	Children's Service DSG	Equipment and Materials Purcha	G AND S SMIRTHWAITE LT	2016-03-01	359.00
56	Childrens Services	Equipment and Materials Purcha	GLS EDUCATIONAL	2014-07-23	459.05
55	Childrens Services	Food Costs	ASDA HOME DELIVERY	2014-06-04	232.32
54	Childrens Services	Consumable Catering Supplies	REYNARDS UK LTD	2014-05-26	258.53
53	Childrens Services	Other Services	ACCESS EXPEDITIONS	2014-04-03	6000.00
48	Childrens Services	Food Costs	JS ONLINE GROCERY	2014-10-14	220.64
51	Childrens Services	Food Costs	SAINSBURYS S/MKT	2014-07-07	175.56
50	Childrens Services	Food Costs	ASDA HOME DELIVERY	2014-06-24	186.12
49	Childrens Services	Training	WWW.NASEN.ORG.UK	2014-04-10	581.90
57	Childrens Services	Food Costs	JS ONLINE GROCERY	2014-06-24	297.47
52	Childrens Services	Food Costs	ASDA STORES 7134	2014-07-22	192.23
65	Commissioning	Training	INCOME OFFICE (2)	2015-08-06	350.00
64	Commissioning	Training	EB DELIVERING INTEGRA	2015-12-23	378.00
67	Commissioning	Travelling Expenses	EASYJET ENRP	2015-02-20	507.88
66	Commissioning	Miscellaneous Expenses	HIREITAILL.COM LTD	2017-03-03	1782.00
63	Commissioning	Professional Services	CENTREMAPS	2016-01-21	720.00
59	Commissioning	General Office Expenses	WWW.PANELWAREHOUSE.COM	2016-10-18	717.30
61	Commissioning	Building Repairs & Maintenance	MET PARKING SERVICES L	2015-06-02	1740.00
60	Commissioning	Equipment and Materials Purcha	DARTS TROPHIES	2014-12-04	345.30
62	Commissioning	Advertising	NTH LONDON & ESSEX NEW	2016-10-12	600.00
58	Commissioning	Equipment and Materials Purcha	FACEBOOK TQU8X8NK52	2016-05-26	535.41
68	Control Accounts	Other Transfer Payments to Soc	Amazon *Mktplce EU-	2014-04-07	83.31
69	Control Accounts	Miscellaneous Expenses	ARGOS RETAIL GROUP	2014-06-13	63.94
78	Customer Support Group	Fees and Charges	HMCOURTS-SERVICE.G	2016-01-20	5918.40
77	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2016-05-26	11088.00
76	Customer Support Group	Fees and Charges	HMCOURTS-SERVICE.G	2015-11-17	5418.00
75	Customer Support Group	Fees and Charges	HMCOURTS-SERVICE.G	2015-07-21	6955.20
79	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2017-01-30	7968.00
73	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2016-07-20	5097.00
72	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2016-06-23	11487.00
71	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2014-11-06	-4707.00
70	Customer Support Group	Legal and Court Fees	HMCOURTS-SERVICE.G	2016-08-19	6069.00
74	Customer Support Group	Fees and Charges	HMCOURTS-SERVICE.G	2015-08-18	5781.00
89	Deputy Chief Operating Officer	IT Services	ADOBE SYSTEMS SOFTW	2014-10-06	114.33
88	Deputy Chief Operating Officer	Equipment and Materials Purcha	INSPIRED FRAMES	2014-07-11	182.50
87	Deputy Chief Operating Officer	Grounds maintenance	PINKS SPIRES	2014-06-03	340.00
86	Deputy Chief Operating Officer	Travelling Expenses	EUROSTAR INTERNATIO	2014-10-10	460.00
85	Deputy Chief Operating Officer	IT Services	ADOBE SYSTEMS SOFTW	2014-09-04	114.33
83	Deputy Chief Operating Officer	Stationery	CARTRIDGE WORLD TUF	2014-05-19	109.99
82	Deputy Chief Operating Officer	Equipment and Materials Purcha	INSPIRED FRAMES	2014-07-11	312.50
81	Deputy Chief Operating Officer	IT Services	ADOBE SYSTEMS SOFTW	2014-08-01	114.33
80	Deputy Chief Operating Officer	Advertising	TYPOFONDERIE	2014-04-24	123.69
84	Deputy Chief Operating Officer	IT Services	DRI SOPHOS SOFTWARE	2014-08-21	160.50
90	Education	Books-CDs-Audio-Video	PEARSON ED LTD	2014-09-25	830.10
97	Family Services	Food Costs	TESCO STORES 644	2014-09-09	195.36
100	Family Services	Food Costs	SAINSBURYS S/MKT	2014-10-20	157.94
99	Family Services	Food Costs	SAINSBURYS S/MKT	2014-09-29	163.89
98	Family Services	Food Costs	SAINSBURYS S/MKT	2014-10-13	182.78
96	Family Services	Equipment and Materials Purcha	JS ONLINE GROCERY	2014-09-30	154.49
91	Family Services	Equipment and Materials Purcha	FLOOR FASHION LTD	2014-09-30	250.00
94	Family Services	Training	SKILLS TRAINING	2014-09-25	576.00
93	Family Services	Postage	M4L LIMITED	2014-09-24	325.00
92	Family Services	Private Contractors - Third Pa	TICKETMASTER UK	2014-10-27	850.00
95	Family Services	Equipment and Materials Purcha	Amazon EU	2014-10-15	396.99
101	Governance	Other Services	BETTER LIFE HEALTH	2014-04-24	6388.20

102	Internal Audit & CAFT	Equipment and Materials Purcha	CANFORD AUDIO PLC	2014-06-19	403.20
103	Internal Audit & CAFT	Private Contractors - Third Pa	EB TENANCY FRAUD FO	2014-10-30	99.00
104	Parking & Infrastructure	Equipment and Materials Purcha	WWW.MIDLANDPALLETTRUCK	2016-12-16	795.00
105	Parking & Infrastructure	Miscellaneous Expenses	WWW.OPUSENERGY.COM	2016-10-10	2773.25
106	Regional Enterprise	Professional Services	J W RUDDOCK &SONS LTD	2015-12-01	1645.00
108	Street Scene	Vehicle Running Costs	POST OFFICE COUNTER	2014-10-29	287.50
107	Street Scene	Vehicle Running Costs	POST OFFICE COUNTER	2014-10-29	400.00
109	Streetscene	Equipment and Materials Purcha	WWW.TOOLSTATION.COM	2016-06-30	583.12
110	Streetscene	Vehicle Running Costs	WWW.DVLA.GOV.UK	2016-03-21	652.50
111	Streetscene	Postage	POST OFFICE COUNTER	2015-02-02	565.00
112	Streetscene	Equipment and Materials Purcha	N & N SIGNS LTD	2017-03-01	590.00
113	Streetscene	Other Services	BRITISH STANDARDS	2015-02-16	717.95
114	Streetscene	Vehicle Running Costs	WWW.DVLA.GOV.UK	2015-07-13	652.50
115	Streetscene	Vehicle Running Costs	DVLA VEHICLE TAX	2016-06-21	652.50
116	Streetscene	Equipment and Materials Purcha	ALPHA PNEUMATIC SUPPLI	2017-02-01	866.00
117	Streetscene	Vehicle Running Costs	WWW.DVLA.GOV.UK	2015-06-22	652.50
118	Streetscene	Building Repairs & Maintenance	DISCOUNT FLOORING -	2014-12-04	527.52

In [82]:
```
#extracting only few hundred outliers from the entire dataframe
selected_df.shape
```

Out[82]:
```
(119, 7)
```

In [83]:
```
#count of service area in our selected dataframe for anomaly detection
selected_df['Service Area'].nunique()
```

Out[83]:
```
19
```

This document provides the comprehensive summary of quarterly transaction of each summary data including the detailed visual representation. Using the z-score document has analysed the instances of spikes and permanent increase in transaction behaviour. By understanding how creditors are classified, document has successfully classified the misclassified creditors. The clustering technique is implemented to classify the similar service area into a same cluster based on their transaction pattern.This can help in resource allocation and budget planning. Finally , the outlier detection using the IQR technique can be the starting point to investigate the potential irregularities and suspicious activities.

In [ ]: