Sales Call center data analysis

August 21, 2023

[33]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[34]:
```
file_path = r"C:\Users\kazit\Downloads\PROJECT_REPORT_WTD_7.31_-_8.6.xlsx"


sheet_name = "CALLS"  # read frist sheet

df = pd.read_excel(file_path, sheet_name=sheet_name)
df.head(10)
```

[34]:

	CALL ID	TIMESTAMP	CAMPAIGN
0	300000069666396	Mon, 31 Jul 2023 08:01:10	eb_c2c
1	300000069666407	Mon, 31 Jul 2023 08:03:01	eb_so
2	300000069666424	Mon, 31 Jul 2023 08:05:17	eb_ho
3	300000069666430	Mon, 31 Jul 2023 08:06:14	eb_web
4	300000069666438	Mon, 31 Jul 2023 08:06:20	eb_web
5	300000069666474	Mon, 31 Jul 2023 08:09:31	eb_hit
6	300000069666489	Mon, 31 Jul 2023 08:10:12	eb_so
7	300000069666402	Mon, 31 Jul 2023 08:10:16	_Transfer_Home_Closer
8	300000069666561	Mon, 31 Jul 2023 08:11:04	eb_hit
9	300000069666504	Mon, 31 Jul 2023 08:12:05	_Transfer_Auto_Closer

	CALL TYPE	AGENT	AGENT NAME
0	Inbound	eb.us-patriciatiernan	Patricia Tiernan
1	Inbound	eb.us-amandamahone	Amanda Mahone
2	Inbound	eb.us-amandamahone	Amanda Mahone
3	Inbound	eb.us-kimberlymontgomery	Kimberly Montgomery
4	Inbound	eb.us-kevinsavala	Kevin Savala
5	Inbound	eb.us-kimberlymontgomery	Kimberly Montgomery
6	Inbound	eb.us-patriciatiernan	Patricia Tiernan
7	Queue Callback	eb.us-ratoriusthomas	Ratorius Thomas
8	Inbound	eb.us-windehayden	Winde Hayden
9	Inbound	eb.us-patriciatiernan	Patricia Tiernan

	DISPOSITION	DNIS	ANI	RING TIME	TALK TIME

```
0                      NaN  1.970441e+10  12672497346  00:00:01  00:01:58
1                      NaN  1.800957e+10  14122454681  00:00:01  00:00:36
2          Not Interested   1.800893e+10  19563427936  00:00:01  00:14:08
3                      NaN  1.800895e+10  15862643650  00:00:01  00:00:28
4  Hung Up collecting info  1.800895e+10  19135796251  00:00:02  00:15:33
5                     Sale  1.800215e+10  18163721760  00:00:01  00:20:39
6    No Coverage Available  1.800925e+10  19169408128  00:00:01  00:01:32
7                     Sale  1.706527e+10  18667047276  00:00:00  00:09:03
8                     Sale  1.888587e+10  12245955727  00:00:02  00:21:33
9       Scheduled Callback  1.202984e+10  12029104323  00:00:02  00:10:19

   Column1 HOLD TIME PARK TIME AFTER CALL WORK TIME  TRANSFERS  \
0     True  00:00:20  00:00:00                  NaN        1.0
1     True  00:00:00  00:00:00                  NaN        1.0
2     True  00:00:00  00:00:00             00:00:13        NaN
3     True  00:00:00  00:00:00                  NaN        1.0
4     True  00:00:00  00:00:00             00:00:31        NaN
5     True  00:00:00  00:00:00             00:00:36        NaN
6     True  00:00:00  00:00:00             00:00:15        NaN
7     True  00:00:00  00:00:00             00:00:38        NaN
8     True  00:00:00  00:00:00             00:00:00        NaN
9     True  00:00:00  00:00:00             00:00:16        NaN

                    SKILL
0  Closers - Auto and Home
1           Closers - Auto
2           Closers - Auto
3           Closers - Auto
4           Closers - Auto
5           Closers - Auto
6           Closers - Auto
7   Closer Outbound - Home
8           Closers - Auto
9   Closer Outbound - Auto
```

0.1 DataSet Cleaning

[35]:
```
df.columns
```

[35]:
```
Index(['CALL ID', 'TIMESTAMP', 'CAMPAIGN', 'CALL TYPE', 'AGENT', 'AGENT NAME',
       'DISPOSITION', 'DNIS', 'ANI', 'RING TIME', 'TALK TIME', 'Column1',
       'HOLD TIME', 'PARK TIME', 'AFTER CALL WORK TIME', 'TRANSFERS', 'SKILL'],
      dtype='object')
```

[36]:
```
df.dtypes
```

[36]:
```
CALL ID                   int64
TIMESTAMP                object
CAMPAIGN                 object
CALL TYPE                object
AGENT                    object
AGENT NAME               object
DISPOSITION              object
DNIS                    float64
ANI                      object
RING TIME                object
TALK TIME                object
Column1                    bool
HOLD TIME                object
PARK TIME                object
AFTER CALL WORK TIME     object
TRANSFERS               float64
SKILL                    object
dtype: object
```

[37]:
```
df.isnull().sum()
```

[37]:
```
CALL ID                     0
TIMESTAMP                   0
CAMPAIGN                    0
CALL TYPE                   0
AGENT                       0
AGENT NAME                  0
DISPOSITION               584
DNIS                       85
ANI                        85
RING TIME                   0
TALK TIME                   0
Column1                     0
HOLD TIME                  35
PARK TIME                  35
AFTER CALL WORK TIME      584
TRANSFERS               12390
SKILL                       0
dtype: int64
```

[38]:
```
# drop_null_value
drop_null_value = ['DISPOSITION', 'DNIS', 'ANI', 'HOLD TIME', 'PARK TIME',␣
 ↪'AFTER CALL WORK TIME']

df = df.dropna(subset=drop_null_value)
```

[39]:
```
#Fill null values in 'TRANSFERS' column with False & '1' encoding True
df['TRANSFERS'] = df['TRANSFERS'].replace(True, 1)
df['TRANSFERS'].fillna(False, inplace=True)
```

[40]:
```
df.isnull().sum() #chek aging
```

[40]:
```
CALL ID                 0
TIMESTAMP               0
CAMPAIGN                0
CALL TYPE               0
AGENT                   0
AGENT NAME              0
DISPOSITION             0
DNIS                    0
ANI                     0
RING TIME               0
TALK TIME               0
Column1                 0
HOLD TIME               0
PARK TIME               0
AFTER CALL WORK TIME    0
TRANSFERS               0
SKILL                   0
dtype: int64
```

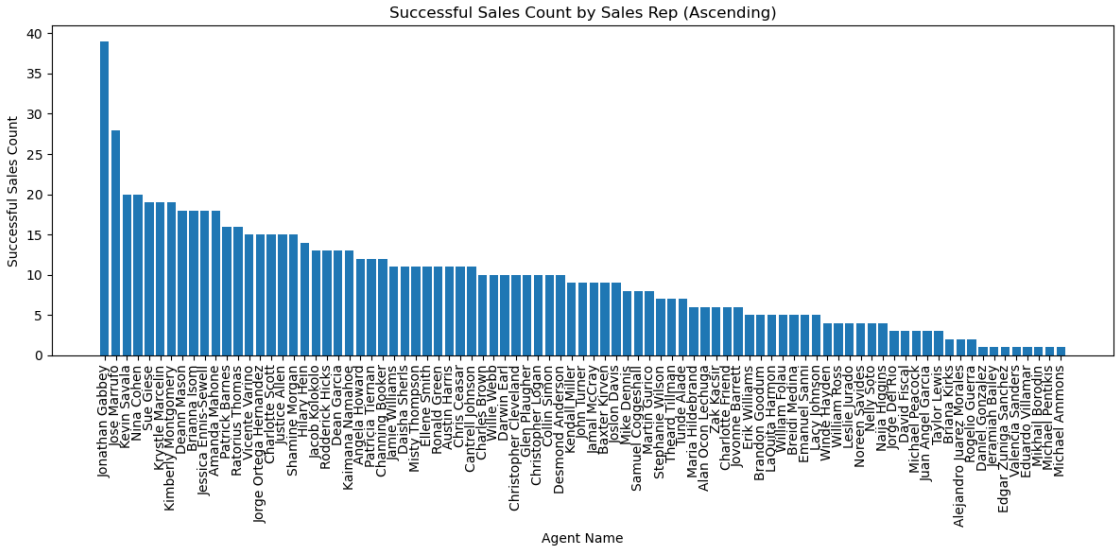
0.1.1 Conversion rate of sales reps all calls and calls that last over 10 seconds.

[41]:
```
# Convert 'TALK TIME' column to timedelta
df['TALK TIME'] = pd.to_timedelta(df['TALK TIME'].apply(str))

df['TALK TIME'] = df['TALK TIME'].dt.total_seconds()

filtered_df = df[(df['TALK TIME'] > 10)]

# Calculate total number of calls
total_calls = len(filtered_df)

if total_calls == 0: #chek have data
    print("No calls meet the criteria.")
else:

    successful_sales = len(filtered_df[filtered_df['DISPOSITION'] == 'Sale'])

    # Calculate conversion rate
    conversion_rate = (successful_sales / total_calls) * 100

    print("Total Calls:", total_calls)
```

```
    print("Successful Sales:", successful_sales)
    print("Conversion Rate:", conversion_rate, "%")
```

```
Total Calls: 10778
Successful Sales: 817
Conversion Rate: 7.580256077194285 %
```

[42]:
```
# Group the data by 'AGENT NAME' and count successful sales
agent_sales_counts = df[df['DISPOSITION'] == 'Sale'].groupby('AGENT NAME').
 ↪size()


sorted_agent_sales_counts = agent_sales_counts.sort_values(ascending=False)

# Plot a bar chart
plt.figure(figsize=(12, 6))
plt.bar(sorted_agent_sales_counts.index, sorted_agent_sales_counts.values)
plt.xticks(rotation=90)
plt.xlabel('Agent Name')
plt.ylabel('Successful Sales Count')
plt.title('Successful Sales Count by Sales Rep (Ascending)')
plt.tight_layout()
plt.show()
```



[43]:
```
#Top ten Agent

T10=sorted_agent_sales_counts[:11]
T10
```

[43]:
```
AGENT NAME
Jonathan Gabbey          39
Jose Marrufo             28
Kevin Savala             20
Nina Cohen               20
Sue Giese                19
Krystle Marcelin         19
Kimberly Montgomery      19
Deann Mason              18
Brianna Isom             18
Jessica Ennis-Sewell     18
Amanda Mahone            18
dtype: int64
```

[44]:
```
#last_ten Agent
L10=sorted_agent_sales_counts[-11:]
L10
```

[44]:
```
AGENT NAME
Briana Kirks                2
Alejandro Juarez Morales    2
Rogelio Guerra              2
Daniel Gonzalez             1
Jeramiah Bailey             1
Edgar Zuniga Sanchez        1
Valencia Sanders            1
Eduardo Villamar            1
Mikhail Borodin             1
Michael Pentikis            1
Michael Ammons              1
dtype: int64
```

0.1.2 What Is the conversion of each type of transfer? (10 sec vs deals)

There is no description of the calls that have been transferred in the call transfer column, so the rows of that description have been dropped. Therefore, the transfer call data set is now missing. But I have analyzed the transfer calls have call type inbound and outbound and in the skill column they have Closer Outbound - Auto or Closer Auto.

0.1.3 Talk time for each agent

[45]:
```
# Calculate talk time for each agent
talk_time_per_agent = df.groupby('AGENT NAME')['TALK TIME'].sum()
talk_time_per_agent = talk_time_per_agent.sort_values(ascending=False)
print(talk_time_per_agent)
```

```
AGENT NAME
Jonathan Gabbey      116202.0
```

```
Misty Thompson       110218.0
Krystle Marcelin     108302.0
Channing Booker      104144.0
Darwin Earl           99575.0
                        …
Michael Pentikis      12796.0
Briana Kirks          10814.0
Victoria Cowans        9802.0
Stephanie Starkey      8290.0
Jose Garcia             198.0
Name: TALK TIME, Length: 93, dtype: float64
```

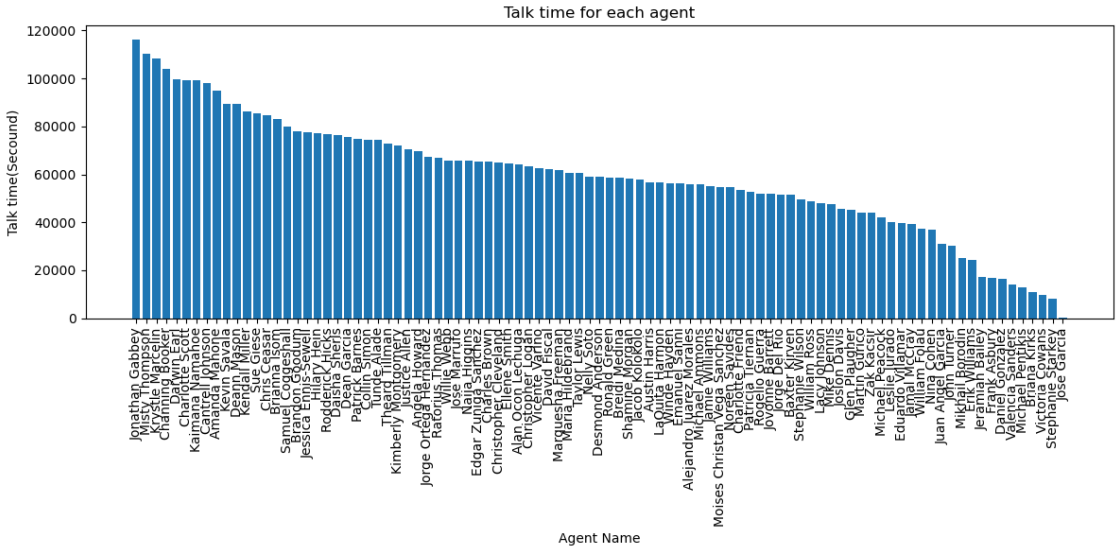
[46]:
```
#Top ten Agent
top10= talk_time_per_agent[:11]
top10
```

[46]:
```
AGENT NAME
Jonathan Gabbey     116202.0
Misty Thompson      110218.0
Krystle Marcelin    108302.0
Channing Booker     104144.0
Darwin Earl          99575.0
Charlotte Scott      99207.0
Kaimana Namahoe      99113.0
Cantrell Johnson     97936.0
Amanda Mahone        94992.0
Kevin Savala         89332.0
Deann Mason          89266.0
Name: TALK TIME, dtype: float64
```

[47]:
```
#last_ten Agent
last10= talk_time_per_agent[-11:]
last10
```

[47]:
```
AGENT NAME
Mikhail Borodin      25121.0
Erik Williams        24435.0
Jeramiah Bailey      17227.0
Frank Asbury         16957.0
Daniel Gonzalez      16309.0
Valencia Sanders     14143.0
Michael Pentikis     12796.0
Briana Kirks         10814.0
Victoria Cowans       9802.0
Stephanie Starkey     8290.0
Jose Garcia            198.0
Name: TALK TIME, dtype: float64
```

[48]:
```
# Plot a bar chart
plt.figure(figsize=(12, 6))
plt.bar(talk_time_per_agent.index, talk_time_per_agent.values)
plt.xticks(rotation=90)
plt.xlabel('Agent Name')
plt.ylabel('Talk time(Secound)')
plt.title('Talk time for each agent')
plt.tight_layout()
plt.show()
```



[49]:
```
file_path =  r"C:\Users\kazit\Downloads\PROJECT_REPORT_WTD_7.31_-_8.6.xlsx"


sheet_name = "SALES "  # read SALES sheet

sal = pd.read_excel(file_path, sheet_name=sheet_name)
```

[50]:
```
sal.head(50)
```

[50]:

	List ID	Policy	Agent Code	\
0	DAD BATCH 77	Monthly Plan	3358.0	
1	EB_SO	Monthly Plan	3277.0	
2	EB_SO	Monthly Plan	3397.0	
3	EB_SO	Monthly Plan	3277.0	
4	EB_HIT	Monthly Plan	3277.0	
5	NaN	Monthly Plan	2022.0	
6	EB_C2C	Monthly Plan	3366.0	
7	DAD BATCH 78	Monthly Plan	3082.0	

```
8               EB_SO-R  Monthly Plan      3277.0
9              EB_WEB-R  Monthly Plan      3366.0
10         DAD BATCH 78  Monthly Plan      2022.0
11          Remarketing  Monthly Plan      3316.0
12          Remarketing  Monthly Plan      3316.0
13                90294  Monthly Plan      3331.0
14  DAD TIER 2 BATCH 13  Monthly Plan      3082.0
15               621207  Monthly Plan      3331.0
16         DAD BATCH 78  Monthly Plan      3397.0
17              EB_SO-R  Monthly Plan      3082.0
18   CS_CUSTOMERSERVICE  Monthly Plan      3084.0
19      REMARKETING TSM  Monthly Plan      3153.0
20               EB_WEB   1-Year Plan      3193.0
21          Remarketing  Monthly Plan      3316.0
22       Remarketing_IB  Monthly Plan      3082.0
23         DAD BATCH 78  Monthly Plan      3323.0
24         DAD BATCH 78  Monthly Plan      3082.0
25         DAD BATCH 70  Monthly Plan      3331.0
26         DAD BATCH 78  Monthly Plan      3397.0
27         DAD BATCH 69  Monthly Plan      3366.0
28         DAD BATCH 78  Monthly Plan      3297.0
29               EB_HIT  Monthly Plan      3331.0
30                EB_SO  Monthly Plan      3082.0
31         DAD BATCH 78  Monthly Plan      3331.0
32                EB_SO  Monthly Plan      3297.0
33    STEP AUTO BATCH 4  Monthly Plan      3082.0
34                EB_SO  Monthly Plan      3257.0
35        STEP BATCH 17  Monthly Plan      3277.0
36              EB_SO-R  Monthly Plan      3277.0
37         DAD BATCH 78  Monthly Plan      3082.0
38                EB_SO  Monthly Plan      3257.0
39         DAD BATCH 77  Monthly Plan      3331.0
40               EB_WEB  Monthly Plan      3257.0
41               EB_WEB  Monthly Plan      3257.0
42               EB_WEB  Monthly Plan      3257.0
43               EB_WEB  Monthly Plan      3257.0
44       Remarketing_IB  Monthly Plan      3323.0
45       Remarketing_IB  Monthly Plan      3323.0
46       Remarketing_IB  Monthly Plan      3084.0
47         DAD BATCH 78  Monthly Plan      3082.0
48                EB_SO  Monthly Plan      3331.0
49                eb_so  Monthly Plan      3153.0

                              Screener              Disposition  \
0   eb.mex-yurenrafaelquineterovasquez  POST-DATE SALE COMPLETE
1                                  NaN             SALE COMPLETE
2                                  NaN             SALE COMPLETE
```

3	NaN		SALE COMPLETE
4	NaN		SALE COMPLETE
5	scardoza@everythingbreaks.com	POST-DATE	SALE COMPLETE
6	NaN		SALE COMPLETE
7	eb.mex-davidalvarezrivera		SALE COMPLETE
8	eb.mex-edgarmojica		SALE COMPLETE
9	eb.mex-marlonnavarrete		SALE COMPLETE
10	eb.mex-MiguelAngelMartinezLara		SALE COMPLETE
11	NaN		LEAD
12	NaN		LEAD
13	eb.mex-eduardomedinaganzalez		SALE COMPLETE
14	eb.mex-angelgomez		SALE COMPLETE
15	eb.mex-EfrainFigueroaValenzuela		SALE COMPLETE
16	eb.mex-CristianRamosRojas		SALE COMPLETE
17	eb.mex-jonathanchavez		SALE COMPLETE
18	NaN		SALE COMPLETE
19	NaN		SALE COMPLETE
20	NaN		SALE COMPLETE
21	NaN		LEAD
22	eb.mex-juanreyes		SALE COMPLETE
23	eb.mex-LilianaRojasGallardo		SALE COMPLETE
24	eb.mex-JorgeTorresLopez		SALE COMPLETE
25	eb.mex-FernandoRafaelPerezGonzalez		SALE COMPLETE
26	eb.mex-JoseAntonioMarinezCaro		SALE COMPLETE
27	eb.mex-erikcuevas		SALE COMPLETE
28	eb.mex-AdolfoMalagonGonzÃ¡lez		SALE COMPLETE
29	NaN		SALE COMPLETE
30	NaN		SALE COMPLETE
31	eb.mex-BryanIsmaelJaucheGarcia		SALE COMPLETE
32	NaN		SALE COMPLETE
33	eb.mex-AlexMartinezBarrales		SALE COMPLETE
34	eb.mex-juanreyes		SALE COMPLETE
35	eb.mex-ramsescabrera		SALE COMPLETE
36	eb.mex-vidalpaz		SALE COMPLETE
37	eb.mex-HectorManuelRomeroSalazar		SALE COMPLETE
38	NaN		SALE COMPLETE
39	eb.mex-IvanMorrillonMeza		SALE COMPLETE
40	NaN		SALE COMPLETE
41	NaN		SALE COMPLETE
42	NaN		SALE COMPLETE
43	NaN		SALE COMPLETE
44	eb.mex-vidalpaz		LEAD
45	eb.mex-vidalpaz		LEAD
46	eb.mex-pedroraymundolujano		SALE COMPLETE
47	eb.mex-americamichelriverasota		SALE COMPLETE
48	NaN		SALE COMPLETE
49	jmartinez@everythingbreaks.com		SALE COMPLETE

	Contract Number	Order #	WP	Phone	Status	Product	\
0	NaN	100062984	2023-08-04	(817) 630-5683	Active	Technology	
1	NaN	100063293	2023-08-05	(813) 376-8177	Active	Smartphones	
2	NaN	100063282	2023-08-05	(918) 271-2256	Active	Smartphones	
3	NaN	100063280	2023-08-05	(202) 631-7710	Active	Smartphones	
4	NaN	100063272	2023-08-05	(859) 245-4626	Active	Smartphones	
5	NaN	100063258	2023-08-05	(414) 213-8507	Active	Smartphones	
6	NaN	100063239	2023-08-04	(202) 729-0060	Active	Smartphones	
7	NaN	100063229	2023-08-04	(405) 762-3422	Active	Smartphones	
8	NaN	100063223	2023-08-04	(252) 214-4884	Active	Smartphones	
9	NaN	100063220	2023-08-04	(262) 278-9117	Active	Smartphones	
10	NaN	100063197	2023-08-04	(832) 245-3048	Active	Smartphones	
11	NaN	100063164	2023-08-04	(972) 824-0011	Active	Smartphones	
12	NaN	100063163	2023-08-04	(972) 824-0011	Active	Smartphones	
13	NaN	100063162	2023-08-04	(718) 737-6055	Active	Smartphones	
14	NaN	100063160	2023-08-04	(313) 299-1255	Active	Smartphones	
15	NaN	100063133	2023-08-04	(470) 546-6497	Active	Smartphones	
16	NaN	100063110	2023-08-04	(585) 491-0678	Active	Smartphones	
17	NaN	100063098	2023-08-04	(910) 748-7711	Active	Smartphones	
18	NaN	100063094	2023-08-04	(254) 422-9614	Active	Smartphones	
19	NaN	100063090	2023-08-04	(470) 598-2256	Active	Smartphones	
20	NaN	100063076	2023-08-04	(267) 410-2103	Active	Smartphones	
21	NaN	100063073	2023-08-04	(972) 824-0011	Active	Smartphones	
22	NaN	100063067	2023-08-04	(330) 840-1707	Active	Smartphones	
23	NaN	100063059	2023-08-04	(917) 885-5269	Active	Smartphones	
24	NaN	100063031	2023-08-04	(231) 557-8142	Active	Smartphones	
25	NaN	100063019	2023-08-04	(518) 390-0210	Active	Smartphones	
26	NaN	100063006	2023-08-04	(773) 503-8960	Active	Smartphones	
27	NaN	100062936	2023-08-03	(248) 946-7331	Active	Smartphones	
28	NaN	100062923	2023-08-03	(757) 284-9952	Active	Smartphones	
29	NaN	100062921	2023-08-03	(757) 258-0151	Active	Smartphones	
30	NaN	100062917	2023-08-03	(470) 602-5531	Active	Smartphones	
31	NaN	100062907	2023-08-03	(347) 586-1553	Active	Smartphones	
32	NaN	100062904	2023-08-03	(850) 363-9797	Active	Smartphones	
33	NaN	100062887	2023-08-03	(678) 403-8135	Active	Smartphones	
34	NaN	100062871	2023-08-03	(601) 397-1507	Active	Smartphones	
35	NaN	100062866	2023-08-03	(970) 702-0162	Active	Smartphones	
36	NaN	100062857	2023-08-03	(832) 996-0251	Active	Smartphones	
37	NaN	100062855	2023-08-03	(910) 470-3377	Active	Smartphones	
38	NaN	100062812	2023-08-03	(330) 396-8977	Active	Smartphones	
39	NaN	100062810	2023-08-03	(984) 233-1834	Active	Smartphones	
40	NaN	100062782	2023-08-03	(863) 855-8299	Active	Smartphones	
41	NaN	100062780	2023-08-03	(863) 855-8299	Active	Smartphones	
42	NaN	100062779	2023-08-03	(863) 855-8299	Active	Smartphones	
43	NaN	100062778	2023-08-03	(863) 855-8299	Active	Smartphones	
44	NaN	100062773	2023-08-03	(443) 564-3358	Active	Smartphones	

45	NaN	100062772	2023-08-03	(443) 564-3358	Active	Smartphones
46	NaN	100062765	2023-08-03	(920) 772-4213	Active	Smartphones
47	NaN	100062749	2023-08-03	(248) 819-0390	Active	Smartphones
48	NaN	100062746	2023-08-03	(602) 243-7661	Active	Smartphones
49	NaN	100062725	2023-08-03	(708) 254-5131	Active	Smartphones

	Grand Total	Alternate Phone	Third Agent Code	Unnamed: 14	Unnamed: 15
0	23.95	NaN	NaN	NaN	NaN
1	9.41	NaN	'+18136501930	NaN	NaN
2	9.28	NaN	'+18009252862	NaN	NaN
3	9.24	NaN	'+18003808503	NaN	NaN
4	9.41	NaN	'+18003744456	NaN	NaN
5	9.32	NaN	NaN	NaN	NaN
6	9.39	NaN	'+19704405110	NaN	NaN
7	9.28	NaN	NaN	NaN	NaN
8	9.39	NaN	NaN	NaN	NaN
9	9.32	NaN	NaN	NaN	NaN
10	9.44	NaN	NaN	NaN	NaN
11	9.44	NaN	NaN	NaN	NaN
12	9.44	NaN	NaN	NaN	NaN
13	7.99	NaN	NaN	NaN	NaN
14	9.41	NaN	NaN	NaN	NaN
15	8.35	NaN	NaN	NaN	NaN
16	9.24	NaN	NaN	NaN	NaN
17	9.39	NaN	NaN	NaN	NaN
18	9.44	(254) 229-6063	'+18667047276	NaN	NaN
19	8.61	NaN	'+14705982256	NaN	NaN
20	94.21	NaN	'+18008950842	NaN	NaN
21	9.44	NaN	NaN	NaN	NaN
22	9.37	NaN	NaN	NaN	NaN
23	9.24	NaN	NaN	NaN	NaN
24	9.39	NaN	NaN	NaN	NaN
25	9.24	NaN	NaN	NaN	NaN
26	9.44	NaN	NaN	NaN	NaN
27	9.41	NaN	NaN	NaN	NaN
28	9.24	NaN	NaN	NaN	NaN
29	8.35	(757) 813-5691	'+18002149375	NaN	NaN
30	9.24	NaN	'+18009235838	NaN	NaN
31	8.52	NaN	NaN	NaN	NaN
32	9.41	NaN	'+18009235838	NaN	NaN
33	9.24	NaN	NaN	NaN	NaN
34	8.61	NaN	NaN	NaN	NaN
35	8.25	NaN	NaN	NaN	NaN
36	8.55	NaN	NaN	NaN	NaN
37	9.39	NaN	NaN	NaN	NaN
38	8.48	NaN	NaN	NaN	NaN
39	8.50	NaN	NaN	NaN	NaN

```
40         8.52  (863) 651-4533    '+18008950842          NaN          NaN
41         8.52  (863) 651-4533    '+18008950842          NaN          NaN
42         8.52  (863) 651-4533    '+18008950842          NaN          NaN
43         8.52  (863) 651-4533    '+18008950842          NaN          NaN
44         8.52             NaN              NaN          NaN          NaN
45         8.52             NaN              NaN          NaN          NaN
46         9.32             NaN              NaN          NaN          NaN
47         9.41             NaN              NaN          NaN          NaN
48         8.58  (602) 550-2476    '+18002816692          NaN          NaN
49         9.44             NaN              NaN          NaN          NaN
```

[51]:
```
sal.isnull().sum()
```

[51]:
```
List ID              14
Policy                0
Agent Code            5
Screener            464
Disposition           0
Contract Number     142
Order #               0
WP                    0
Phone                 0
Status                0
Product               0
Grand Total           0
Alternate Phone    1223
Third Agent Code    928
Unnamed: 14        1365
Unnamed: 15        1364
dtype: int64
```

[52]:
```
sal.columns
```

[52]:
```
Index(['List ID', 'Policy', 'Agent Code', 'Screener', 'Disposition',
       'Contract Number', 'Order #', 'WP', 'Phone', 'Status', 'Product',
       'Grand Total', 'Alternate Phone', 'Third Agent Code', 'Unnamed: 14',
       'Unnamed: 15'],
      dtype='object')
```

[53]:
```
sal.drop(['Unnamed: 14','Unnamed: 15'], axis=1, inplace=True) #drop some␣
 ↪coloumns
```

[54]:
```
sal.columns
```

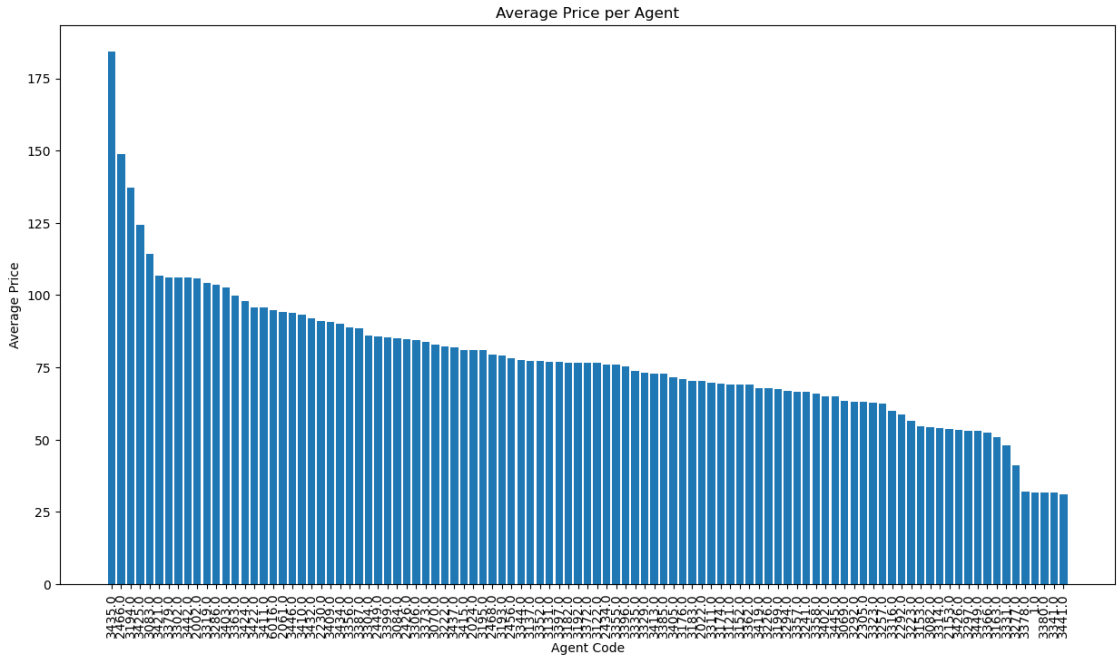
[54]:
```
Index(['List ID', 'Policy', 'Agent Code', 'Screener', 'Disposition',
       'Contract Number', 'Order #', 'WP', 'Phone', 'Status', 'Product',
       'Grand Total', 'Alternate Phone', 'Third Agent Code'],
```

```
      dtype='object')
```

[55]:
```
# Drop rows with null values in specified columns
drop_null_value = ['Agent Code']


sal = sal.dropna(subset=drop_null_value)
```

[56]:
```
avg_price_per_agent = sal.groupby('Agent Code')['Grand Total'].mean()

x_values = range(len(avg_price_per_agent))
avg_price_per_agent = avg_price_per_agent.sort_values(ascending=False)
print(avg_price_per_agent)

# Create a bar chart
plt.figure(figsize=(15, 8))
plt.bar(x_values, avg_price_per_agent.values)
plt.xticks(x_values, avg_price_per_agent.index)  # Use Agent Codes as tick␣
 ↪labels
plt.xlabel('Agent Code')
plt.ylabel('Average Price')
plt.xticks(rotation=90)
plt.title('Average Price per Agent')
plt.show()
```

```
Agent Code
3435.0    184.173333
2466.0    148.876429
3194.0    137.060000
3425.0    124.426250
3083.0    114.106667
             …
3378.0     31.970000
1.0        31.750000
3380.0     31.710000
3341.0     31.670000
3441.0     31.080000
Name: Grand Total, Length: 101, dtype: float64
```



[57]:
```
#Top ten Agent
top_ten = avg_price_per_agent.iloc[:10]
print(top_ten)
```

```
Agent Code
3435.0    184.173333
2466.0    148.876429
3194.0    137.060000
3425.0    124.426250
3083.0    114.106667
3421.0    106.870000
3379.0    106.120000
3302.0    106.120000
3442.0    105.995000
2002.0    105.870000
Name: Grand Total, dtype: float64
```

[58]:
```
#last_ten Agent
last_ten = avg_price_per_agent.iloc[-11:]
print(last_ten)
```

```
Agent Code
3297.0    53.027333
3449.0    52.925000
3366.0    52.320769
3163.0    50.920000
```

```
3331.0    47.963023
3277.0    41.230714
3378.0    31.970000
1.0       31.750000
3380.0    31.710000
3341.0    31.670000
3441.0    31.080000
Name: Grand Total, dtype: float64
```

[ ]: