San Francisco Public Library patron dataset

Hide Code

In [1]:
```
# this will only work if you first 'conda install geopandas' through Anaconda
 prompt. pip install works too. Require to run Anaconda prompt as administrato
r (that's right click at Anaconda prompt )
import geopandas as gpd
import descartes
```

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
```

Dataset

Dataset is loaded. Here's the initial inspection of the head (with first few rows), description, and info that would indicate data types

In [76]:
```
df=pd.read_csv("data/Library_Usage.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	Patron Type Code	Patron Type Definition	Total Checkouts	Total Renewals	Age Range	Home Library Code	Home Library Definition	Circulation Active Month	Circu
0	0	ADULT	1092	761	60 to 64 years	M6	Mission	July	2016
1	0	ADULT	0	0	20 to 24 years	P1	Park	None	None
2	0	ADULT	31	22	25 to 34 years	S7	Sunset	April	2016
3	0	ADULT	0	0	45 to 54 years	P1	Park	None	None
4	0	ADULT	0	0	25 to 34 years	X	Main Library	None	None

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 423448 entries, 0 to 423447
Data columns (total 15 columns):
Patron Type Code                423448 non-null int64
Patron Type Definition          423448 non-null object
Total Checkouts                 423448 non-null int64
Total Renewals                  423448 non-null int64
Age Range                       423233 non-null object
Home Library Code               423408 non-null object
Home Library Definition         423448 non-null object
Circulation Active Month        423448 non-null object
Circulation Active Year         423448 non-null object
Notice Preference Code          423448 non-null object
Notice Preference Definition    423448 non-null object
Provided Email Address          423448 non-null bool
Year Patron Registered          423448 non-null int64
Outside of County               423448 non-null bool
Supervisor District             313138 non-null float64
dtypes: bool(2), float64(1), int64(4), object(8)
memory usage: 42.8+ MB
```

In [6]:
```
df.describe()
# only five numerical values. Patron Type Code, Supervisor District are techni
cally factorial. Year Patron is interval, but can be used to calculate how man
y years a patron has a library card.
```

Out[6]:

	Patron Type Code	Total Checkouts	Total Renewals	Year Patron Registered	Supervisor District
count	423448.000000	423448.000000	423448.000000	423448.000000	313138.000000
mean	1.036765	161.982097	59.657327	2010.348917	6.288240
std	4.188198	453.703678	225.009917	4.357374	3.123634
min	0.000000	0.000000	0.000000	2003.000000	1.000000
25%	0.000000	2.000000	0.000000	2007.000000	4.000000
50%	0.000000	19.000000	2.000000	2012.000000	6.000000
75%	1.000000	113.000000	27.000000	2014.000000	9.000000
max	104.000000	35907.000000	8965.000000	2016.000000	11.000000

Check for missing values

In [7]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[7]:
```
Supervisor District           110310
Age Range                        215
Home Library Code                 40
Outside of County                  0
Year Patron Registered             0
Provided Email Address             0
Notice Preference Definition       0
Notice Preference Code             0
Circulation Active Year            0
Circulation Active Month           0
Home Library Definition            0
Total Renewals                     0
Total Checkouts                    0
Patron Type Definition             0
Patron Type Code                   0
dtype: int64
```

Clearly, three variables - Supervisor District, Age Range, and Home Library Code - have missing values. Supervisor District is missing in approx. 25% of the dataset, so these records definitely could not be imputated. It is worth looking into why such the number is so big. Age Range, Home Library County is small enough, so it can be considered to be imputated. We will do it if we are using these for prediction

"Supervisor District" is an automatically populated fields and will be left blank for users who are outside of country. That will explain high volume of null values in this particular field.
From literature review, I found out that San Francisco Public Library (SFPL) considers equity and social justice as their service priority. They welcome patrons without fixed address - individuals who are homeless - to use their facilities.If the patron record without supervisor district indeed signifies that these are records from vulnerable population, it may be interesting to to run a Z-test to examine some of numerical variables, and/or chi-test to examine categorical variables.
As a library professional, I really admire how SFPL deems serving the vulnerable population as part of their mandate. It also reminds me that, while it is important for libraries to show its values by using numbers and statistics - such as usage stats like total checkouts - part of its true value in the society is in providing services to those who are in need. The weakness of this particular dataset is that it only reflects the usage of those patrons who borrow materials form the library.

Initial Cleanup

That include removable variables with less values, such as 'Circulation Active Month'

We are also going to combine Total Checkouts and Total Renewals to create total_cko, as these are both circulation activities. And then, we are going to calculate the year_registered and average cko, since it is important to take number of years a patron has an account before evaluating the usage.

In sum, three new variables will be created here, and one will be dropped

To be created:

- total_cko
- avg_cko
- years_registered

To be dropped

- Circulation Active Month

In [8]:
```
df=df.drop('Circulation Active Month', axis=1)
df['total_cko']=df['Total Checkouts']+df['Total Renewals']
df['years_registered']=2016-df['Year Patron Registered']
df['avg_cko']=df['total_cko']/(2016-df['Year Patron Registered'])
df['avg_cko'].loc[df['Year Patron Registered']==2016]=df['total_cko']
df.info()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py:190: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  self._setitem_with_indexer(indexer, value)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 423448 entries, 0 to 423447
Data columns (total 17 columns):
Patron Type Code                423448 non-null int64
Patron Type Definition          423448 non-null object
Total Checkouts                 423448 non-null int64
Total Renewals                  423448 non-null int64
Age Range                       423233 non-null object
Home Library Code               423408 non-null object
Home Library Definition         423448 non-null object
Circulation Active Year         423448 non-null object
Notice Preference Code          423448 non-null object
Notice Preference Definition    423448 non-null object
Provided Email Address          423448 non-null bool
Year Patron Registered          423448 non-null int64
Outside of County               423448 non-null bool
Supervisor District             313138 non-null float64
total_cko                       423448 non-null int64
years_registered                423448 non-null int64
avg_cko                         423448 non-null float64
dtypes: bool(2), float64(2), int64(6), object(7)
memory usage: 49.3+ MB
```

In [9]:
```
# inspection data after the initial clean-up
df.head()
```

Out[9]:

	Patron Type Code	Patron Type Definition	Total Checkouts	Total Renewals	Age Range	Home Library Code	Home Library Definition	Circulation Active Year	I Prefe
0	0	ADULT	1092	761	60 to 64 years	M6	Mission	2016	p
1	0	ADULT	0	0	20 to 24 years	P1	Park	None	z
2	0	ADULT	31	22	25 to 34 years	S7	Sunset	2016	z
3	0	ADULT	0	0	45 to 54 years	P1	Park	None	a
4	0	ADULT	0	0	25 to 34 years	X	Main Library	None	z

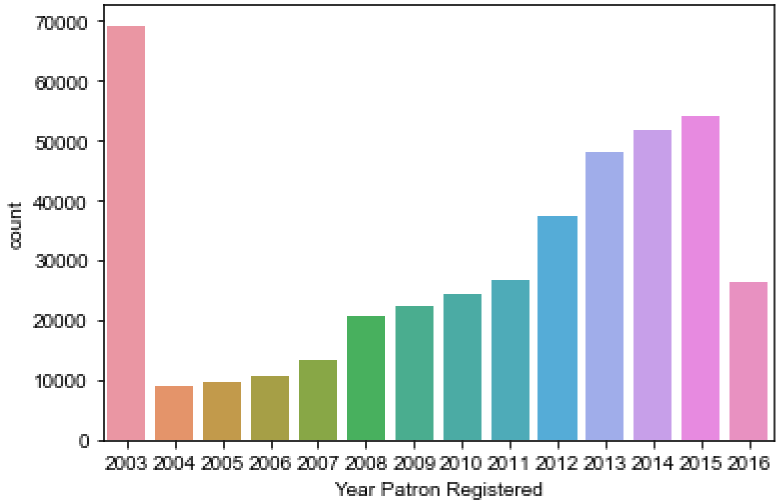
Dependent Variables

Categorical Variables

Year Patron Registered

The year "2003" is likely the year records got imported from old system to new system, so it is not surprised that this category would have a high count. 2016 data is likely to be incomplete, and hence does not reflect the trend. From this count plot, it looks like there is an increase in membership over the span of ten years. This is likely to be due to population growth, opening of new branches to cover a bigger geographical area, and, possibly, growing interest in using the library.

In [10]:
```
sns.countplot(x='Year Patron Registered',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
```

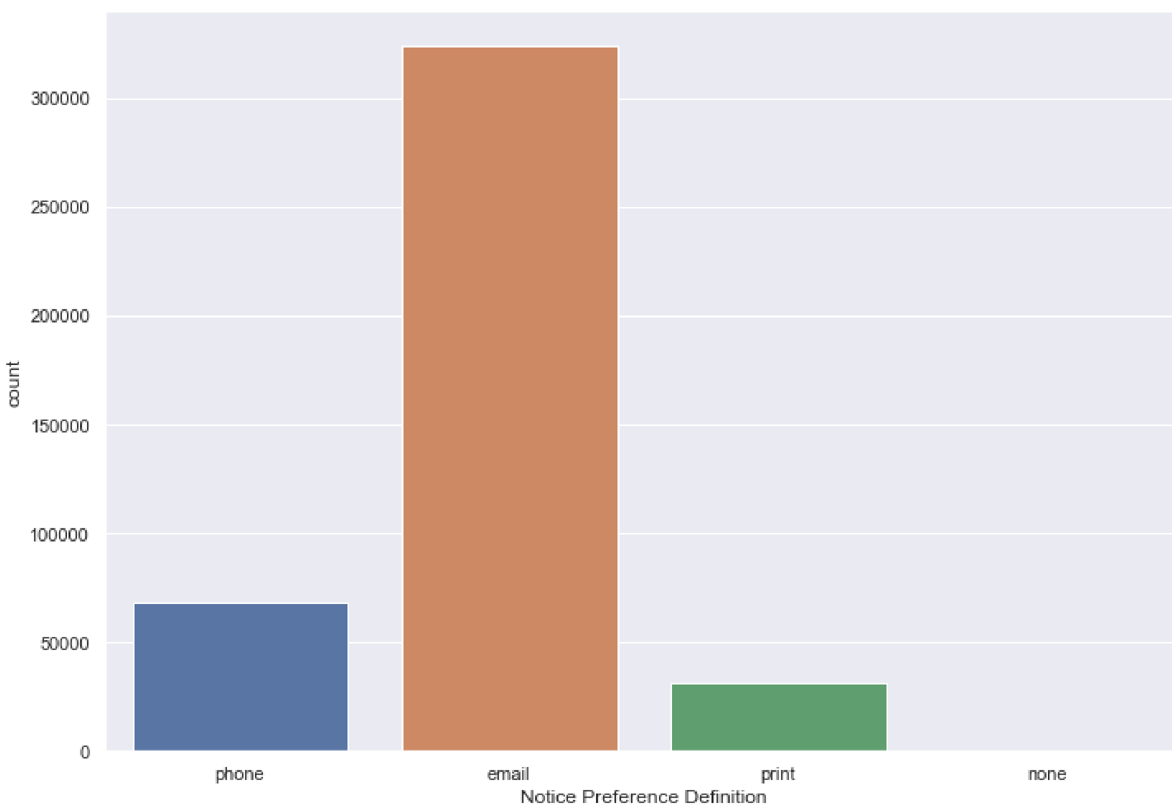


Notice Preference Definition/Provided Email Address

Not much surprised here. Most patrons prefer to receive notifications (such as when their holds are ready for pickup, when a book is overdue) through email. Those who do not provide email opted to receive notification via phone (second preferred option) and mail (print).

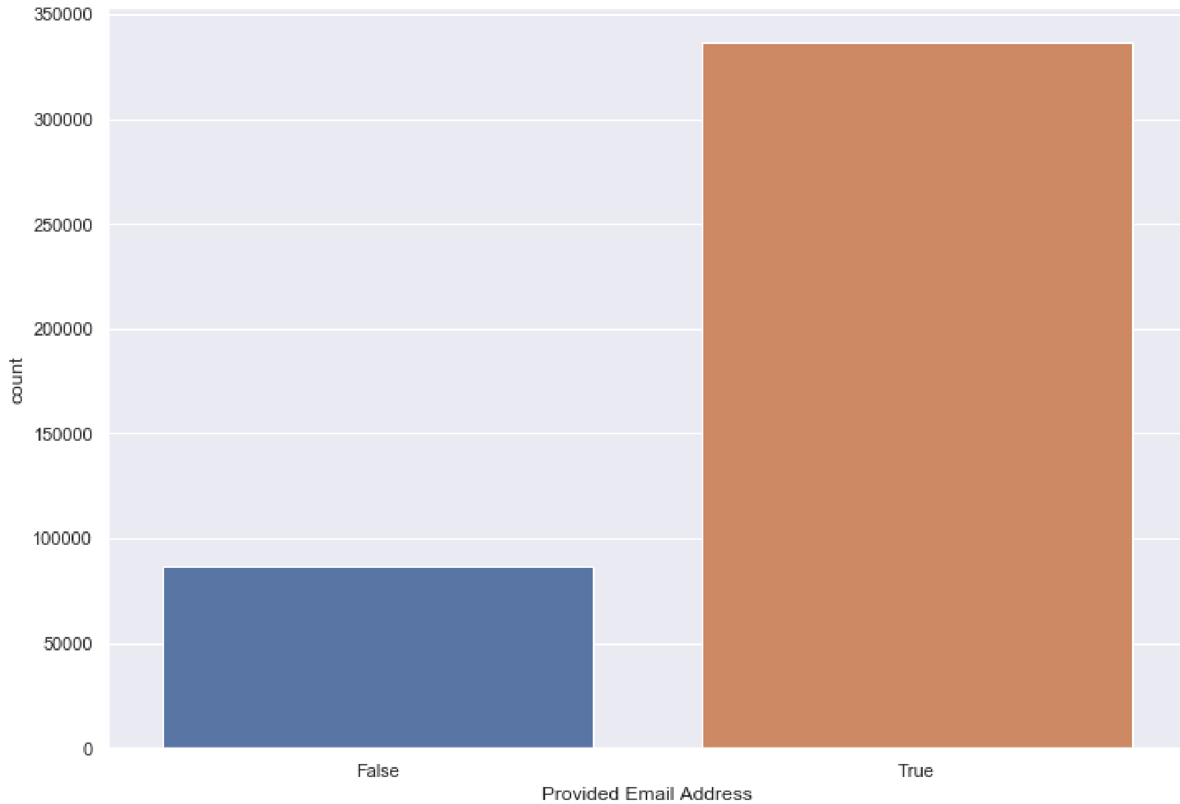
In [11]:
```
sns.countplot(x='Notice Preference Definition',data=df)
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x154abba57b8>

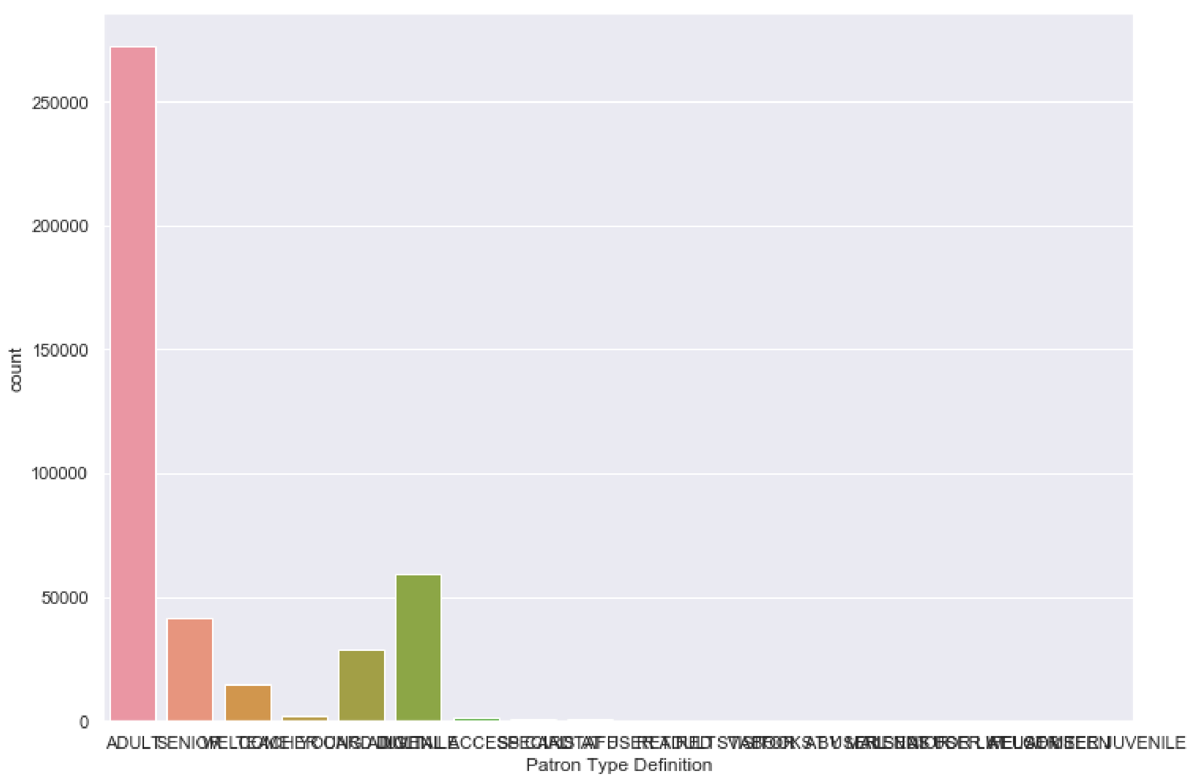


In [12]:
```
sns.countplot(x='Provided Email Address',data=df)
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x154abbfc048>



In [13]:
```
sns.countplot(x='Patron Type Definition',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
```



In [14]:
```
df.groupby(['Patron Type Definition'])['avg_cko'].agg(['count','median','min',
'max'])
```

Out[14]:

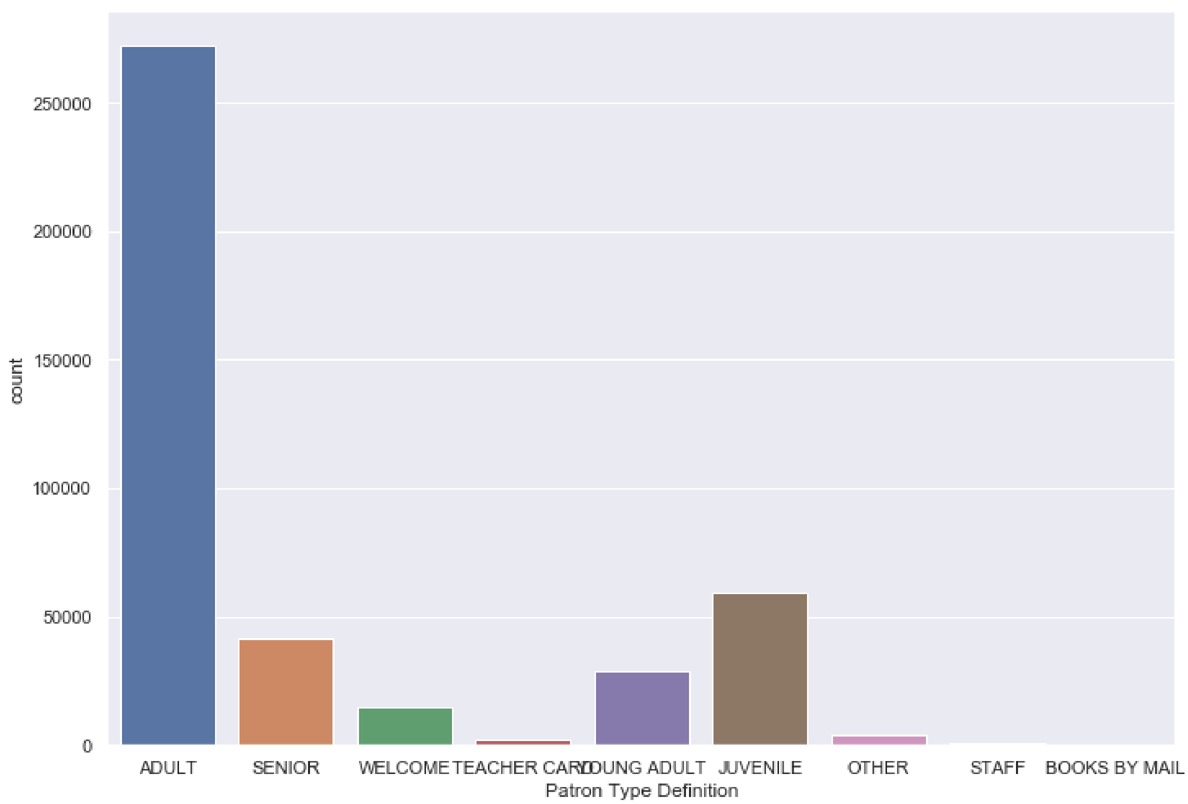
	count	median	min	max
Patron Type Definition				
ADULT	272251	5.166667	0.0	2771.000000
AT USER ADULT	349	4.333333	0.0	686.500000
AT USER JUVENILE	47	6.666667	0.0	337.500000
AT USER SENIOR	66	9.182692	0.0	335.538462
AT USER TEEN	44	2.750000	0.0	209.307692
AT USER WELCOME	45	0.111111	0.0	123.769231
BOOKS BY MAIL	95	20.000000	0.0	524.461538
DIGITAL ACCESS CARD	1744	0.000000	0.0	491.000000
FRIENDS FOR LIFE	40	34.192308	0.0	838.600000
JUVENILE	59208	16.000000	0.0	1413.000000
RETIRED STAFF	157	77.692308	0.0	628.153846
SENIOR	41619	8.400000	0.0	2567.100000
SPECIAL	977	15.923077	0.0	1564.000000
STAFF	862	85.250000	0.0	1299.000000
TEACHER CARD	1782	20.250000	0.0	947.000000
VISITOR	415	3.500000	0.0	164.000000
WELCOME	14931	0.250000	0.0	325.153846
YOUNG ADULT	28816	8.384615	0.0	1178.000000

In [15]:

```
# Reduce the number of categories
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER ADU
LT","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER JUV
ENILE","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER SEN
IOR","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER WEL
COME","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER TEE
N","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("AT USER TEE
N","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("FRIENDS FOR
LIFE","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("DIGITAL ACC
ESS CARD","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("RETIRED STA
FF","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("VISITOR","O
THER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("BOOK BY MAI
L","OTHER")
df['Patron Type Definition']=df['Patron Type Definition'].replace("SPECIAL","O
THER")
```

In [16]:

```
sns.countplot(x='Patron Type Definition',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
```



In [17]:
```
df.groupby(['Patron Type Definition'])['avg_cko'].agg(['count','median','min',
'max'])
```

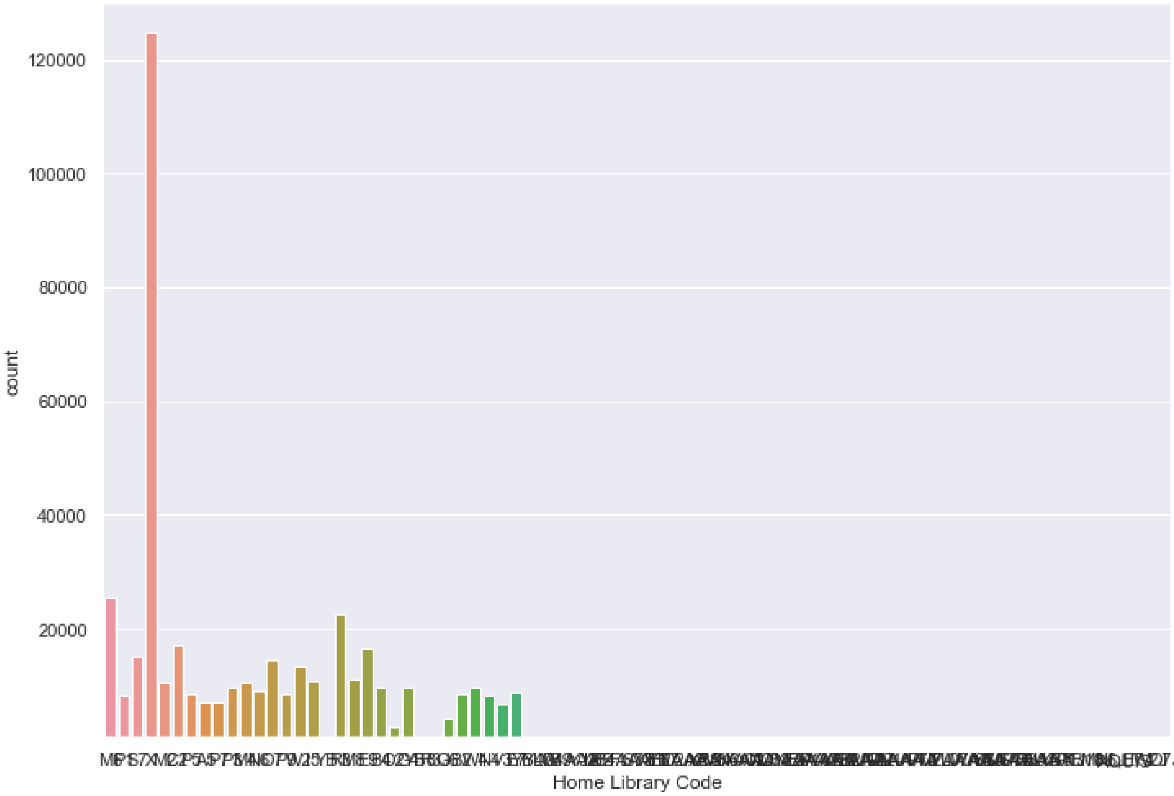
Out[17]:

	count	median	min	max
Patron Type Definition				
ADULT	272251	5.166667	0.0	2771.000000
BOOKS BY MAIL	95	20.000000	0.0	524.461538
JUVENILE	59208	16.000000	0.0	1413.000000
OTHER	3884	0.500000	0.0	1564.000000
SENIOR	41619	8.400000	0.0	2567.100000
STAFF	862	85.250000	0.0	1299.000000
TEACHER CARD	1782	20.250000	0.0	947.000000
WELCOME	14931	0.250000	0.0	325.153846
YOUNG ADULT	28816	8.384615	0.0	1178.000000

Home Library Code/Home Library Definition

Most of the traffic seems to be from Main Library, which is likely to be the biggest branch, with the most programming activities there. It is also more likely to be in downtown, making it the centre of actions. These factors are all likely boost circulaton activites.

In [18]:
```
sns.countplot(x='Home Library Code',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
plt.ylim(1000,130000);
```



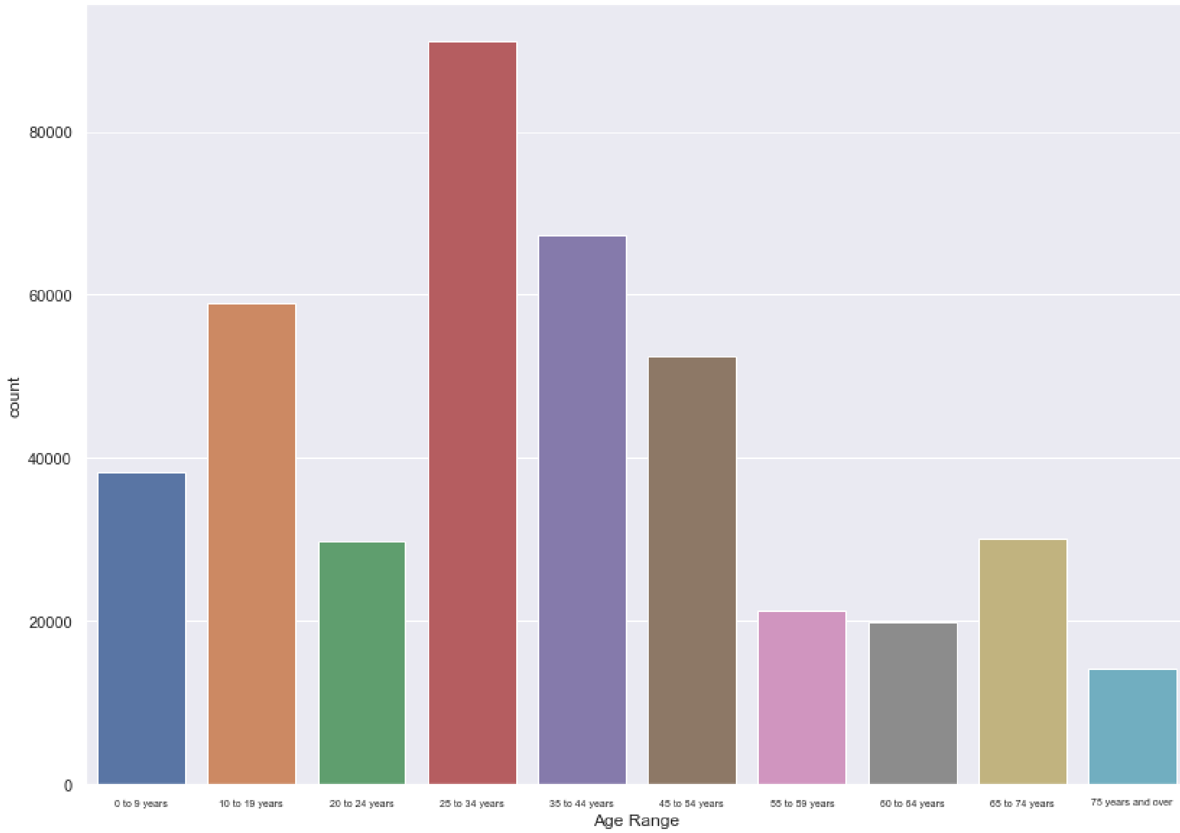
In [19]:

```
# Try to reduce the number of Home Library by combining Branch Mobile

df['Home Library Definition']=df['Home Library Definition'].replace("Branch Bo
okmobile (Excelsior)","Bookmobile")
df['Home Library Definition']=df['Home Library Definition'].replace("Branch Bo
okmobile (Marina)","Bookmobile")
df['Home Library Definition']=df['Home Library Definition'].replace("Branch Bo
okmobile (Sunset)","Bookmobile")
df['Home Library Definition']=df['Home Library Definition'].replace("Branch Bo
okmobile (West Portal)","Bookmobile")
df['Home Library Definition']=df['Home Library Definition'].replace("Childre
n's Bookmobile","Bookmobile")


df.groupby(['Home Library Definition'])['avg_cko'].agg(['count','median','min'
,'max'])
```

Out[19]:

	count	median	min	max
Home Library Definition				
Anza	7183	12.400000	0.0	986.692308
Bayview/Linda Brooks-Burton	8417	2.333333	0.0	1321.153846
Bernal Heights	9630	7.895833	0.0	843.714286
Bookmobile	968	6.519231	0.0	670.000000
Chinatown	17140	17.094017	0.0	1245.333333
Eureka Valley/Harvey Milk Memorial	8708	8.076923	0.0	1801.900000
Excelsior	16706	6.160256	0.0	1352.200000
Glen Park	9811	8.076923	0.0	986.307692
Golden Gate Valley	4381	6.000000	0.0	976.000000
Ingleside	10738	7.500000	0.0	900.500000
Library on Wheels	782	7.000000	0.0	480.461538
Main Library	124814	2.714286	0.0	2567.100000
Marina	10631	5.500000	0.0	1538.000000
Merced	10502	7.800000	0.0	1138.400000
Mission	25443	5.625000	0.0	866.923077
Mission Bay	11271	4.666667	0.0	1005.000000
Noe Valley/Sally Brunn	8399	10.000000	0.0	968.000000
North Beach	9162	6.088462	0.0	1282.666667
Ocean View	2914	6.125000	0.0	715.714286
Ortega	14456	16.923077	0.0	1115.230769
Park	8271	7.900000	0.0	948.230769
Parkside	9744	10.000000	0.0	1438.000000
Portola	8659	11.923077	0.0	1029.000000
Potrero	7196	6.000000	0.0	1156.000000
Presidio	8652	6.250000	0.0	965.200000
Richmond	22475	10.769231	0.0	1490.000000
Sunset	15020	11.428571	0.0	1805.500000
Unknown	1498	12.076923	0.0	1146.000000
Visitacion Valley	6833	6.916667	0.0	2771.000000
West Portal	13338	10.384615	0.0	1337.000000
Western Addition	9706	6.875000	0.0	956.500000

In [20]:
```
ax=sns.countplot(x='Age Range',order=('0 to 9 years','10 to 19 years','20 to 2
4 years','25 to 34 years','35 to 44 years','45 to 54 years', '55 to 59 years',
'60 to 64 years','65 to 74 years','75 years and over'),data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
ax.set_xticklabels(ax.get_xticklabels(), fontsize=7)
plt.tight_layout()
plt.show()
```

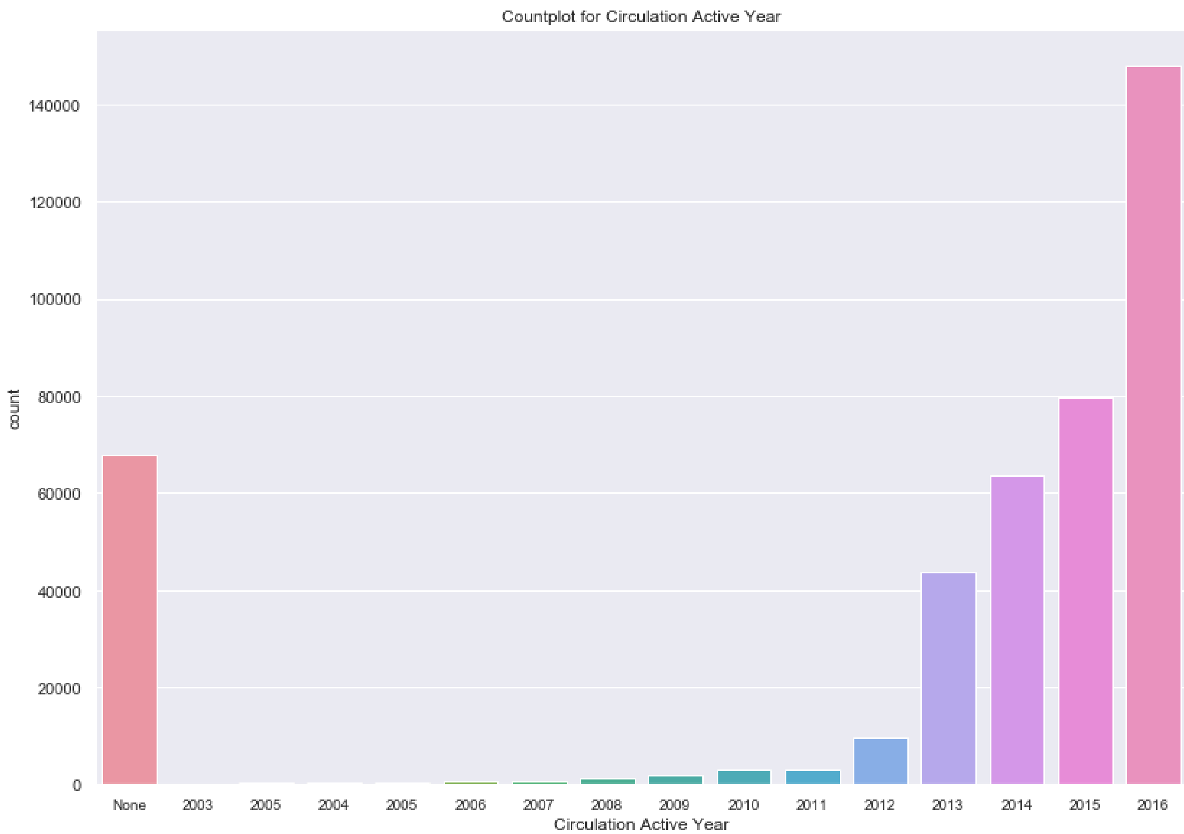


Age range

At the first glance, it looks like the data is not normally distributed. However, note that the intervals are not the same: the width varies from 5 years to 10 years. Because of this, this is not a histrogram that could be used to determine whether there is any skewness in the data.

If we redistribute "25 to 34 years","35 to 44 year", "45 to 54 years" so that each of these category would have an interval of ten years, it looks like the data would peak in the newly category of "30 to 39 years". This presumption is based on the likelihood that both "25 to 34 years" and "35 to 44 years" could be equally split into two parts and redistributed. Based on our literature review, we know that the median age of San Francisco is 38.9 year. This seems to be aligned with our findings with the library data

In [22]:
```
ax=sns.countplot(x="Circulation Active Year",hue_order='Supervisor District',o
rder=('None','2003','2005','2004','2005','2006','2007','2008','2009','2010','2
011','2012','2013','2014','2015','2016'),data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
ax.set_xticklabels(ax.get_xticklabels(), fontsize=10)
ax.set_title('Countplot for Circulation Active Year')
plt.tight_layout()
plt.show()
```

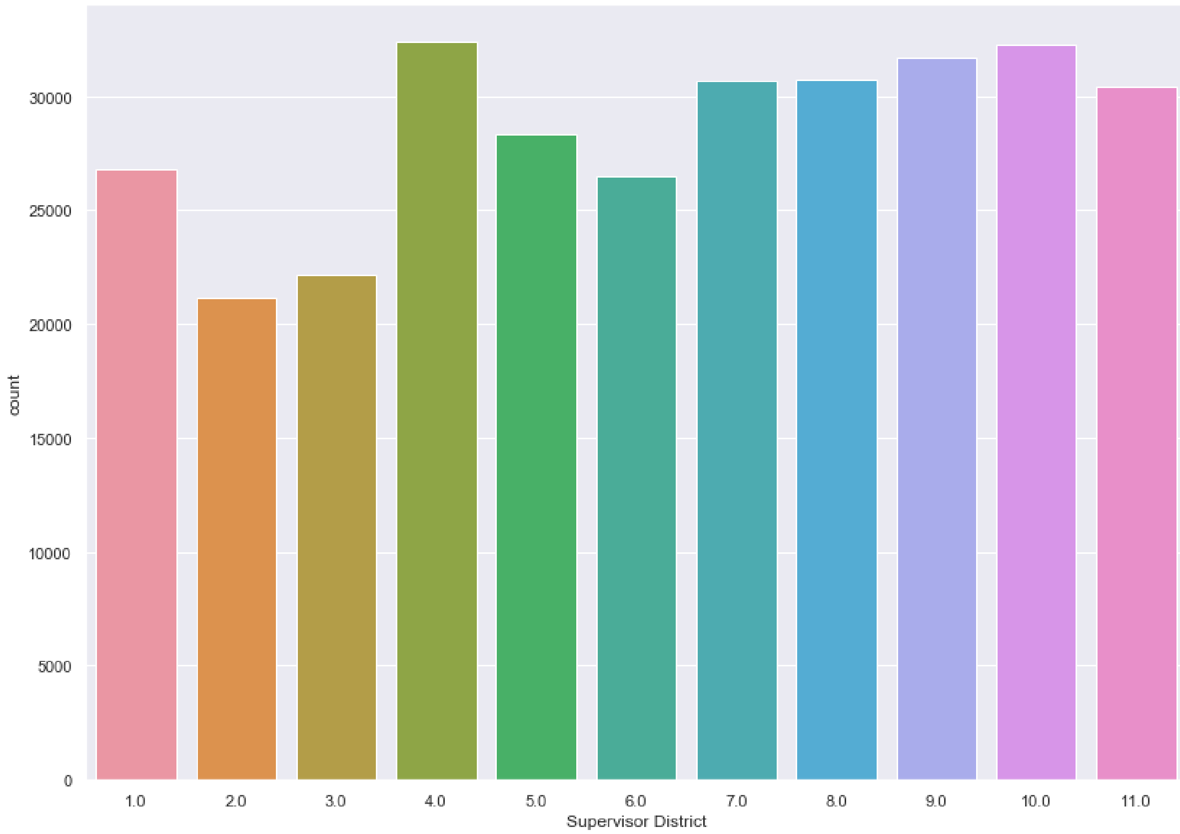


Circulation Last Year

This is interesting to see that most of the users have used the library in the past one year ("2016") or perhaps ("2015"). This indicates that SFPL has a lot of regular users.

Supervisor District

In [23]:
```
sns.countplot(x='Supervisor District',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
ax.set_xticklabels(ax.get_xticklabels(), fontsize=7)
plt.tight_layout()
plt.show()
```

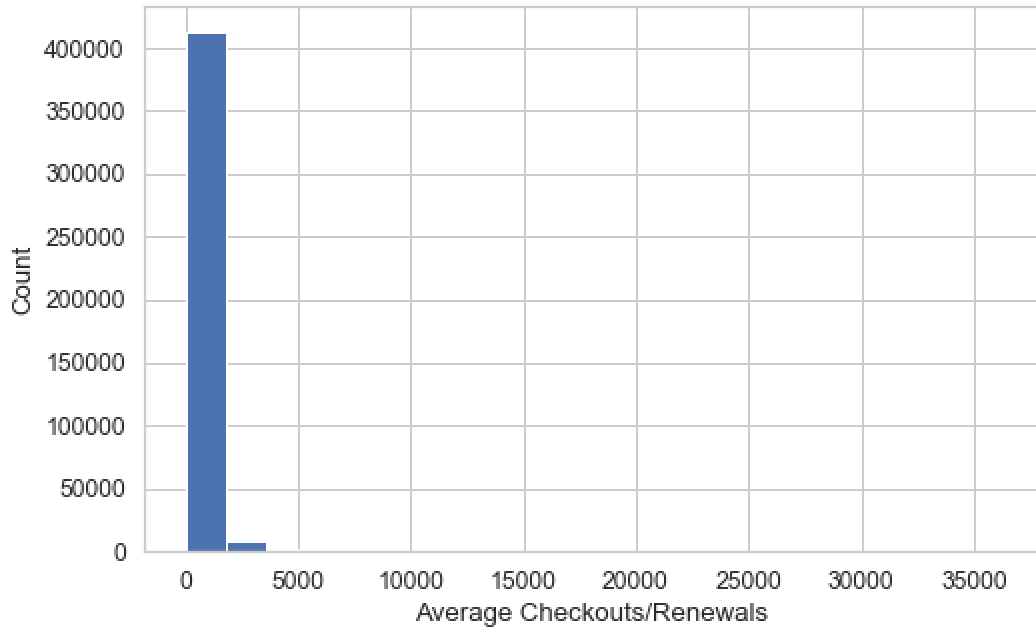


With the exception of District two and District three, the data seems to be uniformly distributed. Further analysis will be done in a later section when combining with average checkouts (ckos) and also with the use of GeoPandas.

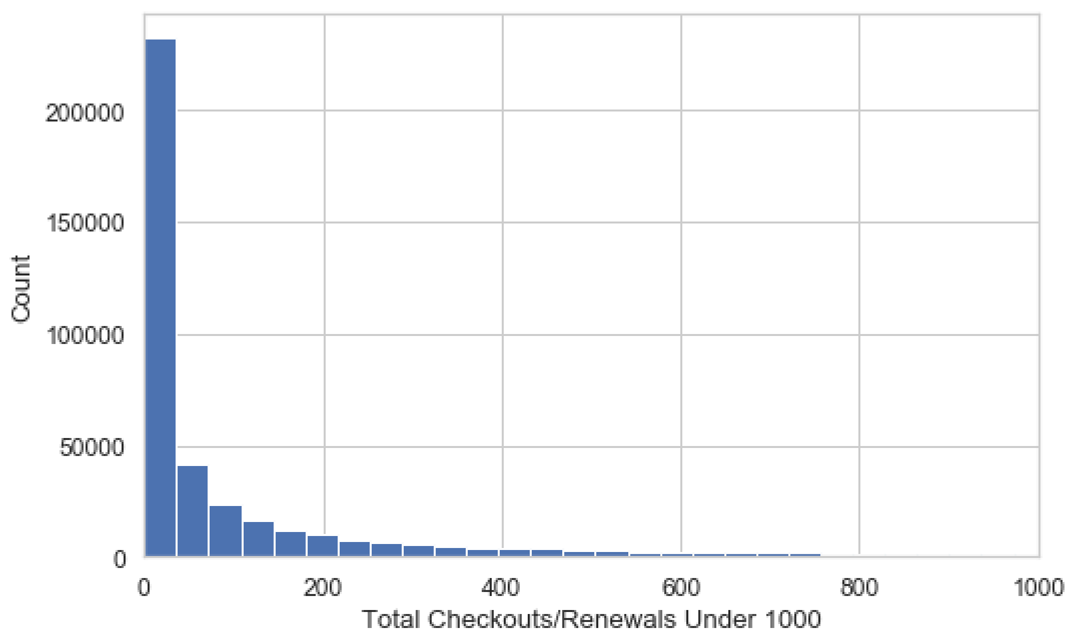
Numercial Variables

Total Checkouts/Average Checkouts

In [52]:
```
sns.set(style='whitegrid', palette="deep", font_scale=1.1, rc={"figure.figsiz
e": [8, 5]})

sns.distplot(df['total_cko'], norm_hist=False, kde=False, bins=20, hist_kws={
"alpha": 1}).set(xlabel='Average Checkouts/Renewals', ylabel='Count');
```

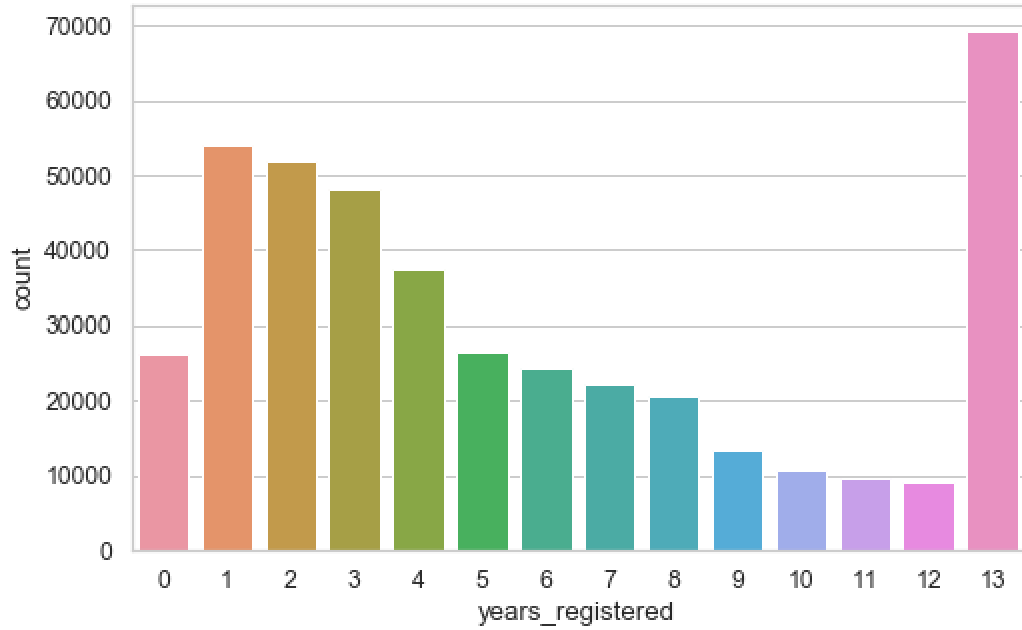


In [51]:
```
sns.set(style='whitegrid', palette="deep", font_scale=1.1, rc={"figure.figsiz
e": [8, 5]})

sns.distplot(df['total_cko'], norm_hist=False, kde=False, bins=1000, hist_kws=
{"alpha": 1}).set(xlabel='Total Checkouts/Renewals Under 1000', ylabel='Count'
)
plt.xlim(0,1000);
```

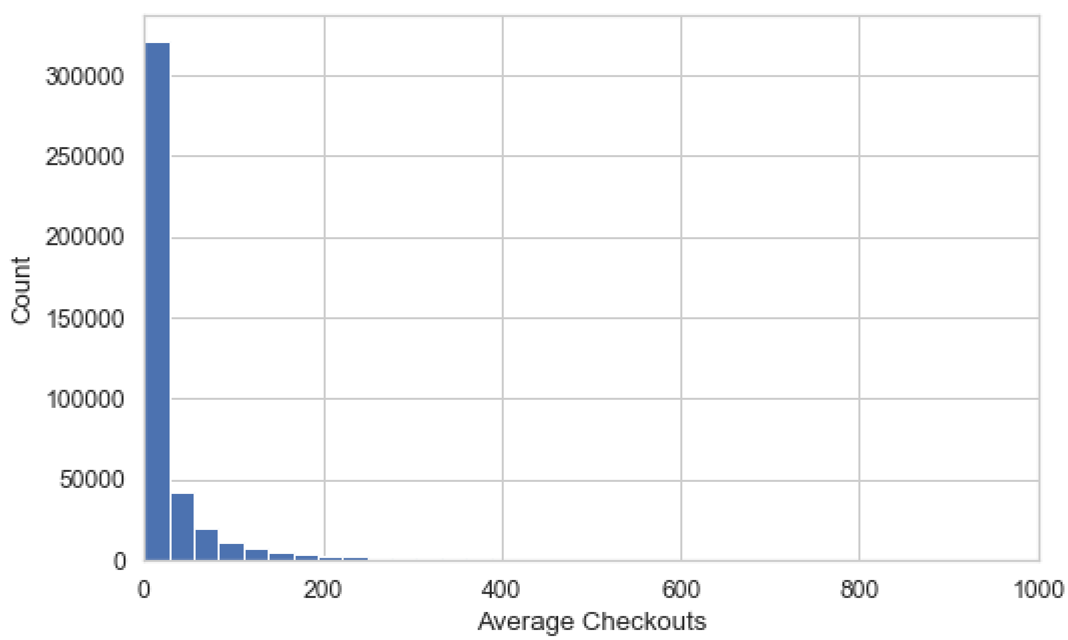


Taking a closer look at the circulation activities below 1000, it appears to be a good idea to label "frequent users" as those with activities under 50

In [26]:
```
sns.countplot(x='years_registered',data=df)
sns.set(rc={'figure.figsize':(11.7,8.27)})
```



In [53]:
```
sns.set(style='whitegrid', palette="deep", font_scale=1.1, rc={"figure.figsiz
e": [8, 5]})

sns.distplot(df['avg_cko'], norm_hist=False, kde=False, bins=100, hist_kws={"a
lpha": 1}).set(xlabel='Average Checkouts', ylabel='Count')
plt.xlim(0,1000);
```



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

Out[28]:

	Patron Type Code	Total Checkouts	Total Renewals	Year Patron Registered	Supervisor District	
count	423448.000000	423448.000000	423448.000000	423448.000000	313138.000000	42
mean	1.036765	161.982097	59.657327	2010.348917	6.288240	22
std	4.188198	453.703678	225.009917	4.357374	3.123634	61
min	0.000000	0.000000	0.000000	2003.000000	1.000000	0.0
25%	0.000000	2.000000	0.000000	2007.000000	4.000000	3.0
50%	0.000000	19.000000	2.000000	2012.000000	6.000000	26
75%	1.000000	113.000000	27.000000	2014.000000	9.000000	15
max	104.000000	35907.000000	8965.000000	2016.000000	11.000000	36

In [29]:
```
outliers=df[df['Total Checkouts']>10000]
print(outliers)
```

	Patron Type Code	Patron Type Definition	Total Checkouts	\
895	9	OTHER	18064	
2007	0	ADULT	10521	
3543	0	ADULT	12740	
20083	3	SENIOR	16060	
31334	0	ADULT	13784	
39620	0	ADULT	11086	
57244	0	ADULT	10906	
86671	3	SENIOR	11748	
117604	0	ADULT	11817	
120565	3	SENIOR	10108	
129328	3	SENIOR	12757	
138318	0	ADULT	10809	
145594	3	SENIOR	11871	
146589	0	ADULT	24093	
156774	0	ADULT	17308	
163847	3	SENIOR	25223	
180148	0	ADULT	10371	
200176	3	SENIOR	18397	
216249	0	ADULT	12950	
222872	3	SENIOR	14502	
227545	3	SENIOR	11147	
231752	3	SENIOR	11896	
237636	0	ADULT	15505	
255022	3	SENIOR	11102	
278357	3	SENIOR	11366	
288654	0	ADULT	15598	
290972	0	ADULT	12733	
293388	0	ADULT	35907	
294146	0	ADULT	10863	
330569	3	SENIOR	10637	
401347	5	STAFF	13362	

	Total Renewals	Age Range	Home Library Code	\
895	2268	60 to 64 years	X	
2007	621	60 to 64 years	X	
3543	2209	45 to 54 years	C2	
20083	66	65 to 74 years	C2	
31334	74	60 to 64 years	M4	
39620	1083	35 to 44 years	C2	
57244	1421	45 to 54 years	P1	
86671	963	65 to 74 years	X	
117604	2859	45 to 54 years	S7	
120565	59	65 to 74 years	X	
129328	500	65 to 74 years	R3	
138318	24	20 to 24 years	S7	
145594	50	65 to 74 years	M4	
146589	383	45 to 54 years	X	
156774	257	60 to 64 years	X	
163847	448	65 to 74 years	X	
180148	1345	55 to 59 years	S7	
200176	6	65 to 74 years	X	
216249	1548	60 to 64 years	O7	
222872	3517	65 to 74 years	E7	
227545	123	65 to 74 years	M6	
231752	1408	65 to 74 years	E7	
237636	294	60 to 64 years	X	

255022	2652	65 to 74 years	X
278357	62	65 to 74 years	X
288654	228	55 to 59 years	X
290972	60	45 to 54 years	X
293388	116	35 to 44 years	V3
294146	57	45 to 54 years	M6
330569	148	75 years and over	C2
401347	1926	0 to 9 years	X

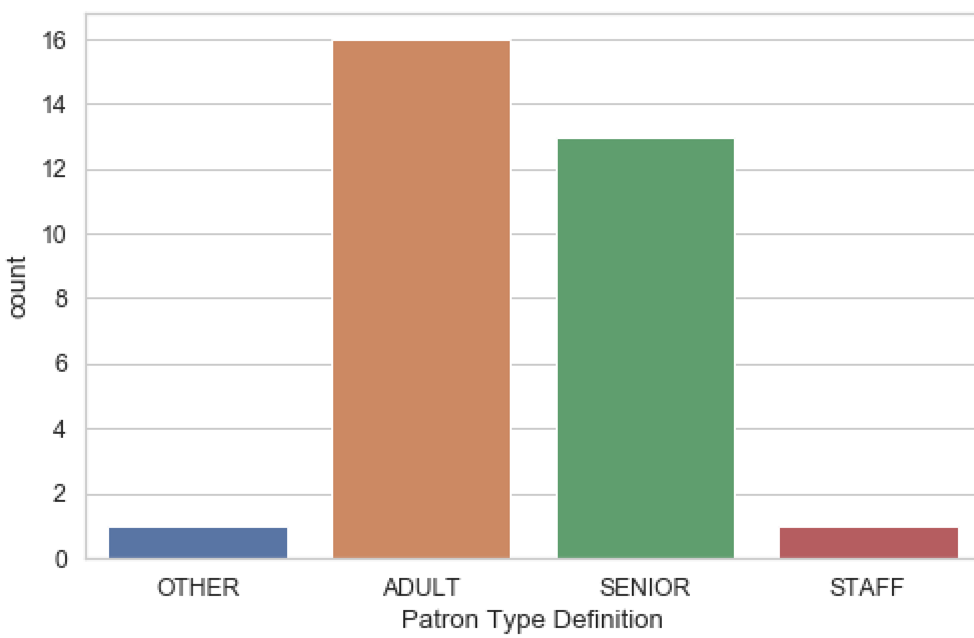
	Home Library Definition	Circulation Active Year \
895	Main Library	2016
2007	Main Library	2016
3543	Chinatown	2016
20083	Chinatown	2016
31334	Merced	2016
39620	Chinatown	2016
57244	Park	2016
86671	Main Library	2016
117604	Sunset	2016
120565	Main Library	2013
129328	Richmond	2016
138318	Sunset	2016
145594	Merced	2016
146589	Main Library	2016
156774	Main Library	2016
163847	Main Library	2016
180148	Sunset	2016
200176	Main Library	2016
216249	Ortega	2016
222872	Eureka Valley/Harvey Milk Memorial	2016
227545	Mission	2016
231752	Eureka Valley/Harvey Milk Memorial	2016
237636	Main Library	2016
255022	Main Library	2016
278357	Main Library	2016
288654	Main Library	2016
290972	Main Library	2016
293388	Visitacion Valley	2016
294146	Mission	2016
330569	Chinatown	2015
401347	Main Library	2016

	Notice Preference Code	Notice Preference Definition \
895	p	phone
2007	p	phone
3543	z	email
20083	a	print
31334	z	email
39620	p	phone
57244	z	email
86671	z	email
117604	z	email
120565	p	phone
129328	p	phone
138318	z	email
145594	z	email
146589	p	phone

```
156774                z                         email
163847                p                         phone
180148                z                         email
200176                z                         email
216249                z                         email
222872                z                         email
227545                a                         print
231752                z                         email
237636                p                         phone
255022                z                         email
278357                z                         email
288654                p                         phone
290972                z                         email
293388                p                         phone
294146                p                         phone
330569                p                         phone
401347                z                         email

        Provided Email Address  Year Patron Registered  Outside of County  \
895                      False                    2003              False
2007                     False                    2003              False
3543                      True                    2003              False
20083                    False                    2003              False
31334                     True                    2003              False
39620                    False                    2003              False
57244                     True                    2003              False
86671                     True                    2003              False
117604                    True                    2003              False
120565                   False                    2003              False
129328                   False                    2004              False
138318                    True                    2010              False
145594                    True                    2003              False
146589                    True                    2003              False
156774                    True                    2003              False
163847                   False                    2006              False
180148                    True                    2003              False
200176                    True                    2005              False
216249                    True                    2003              False
222872                    True                    2006              False
227545                   False                    2003              False
231752                    True                    2003              False
237636                   False                    2004              False
255022                    True                    2003              False
278357                    True                    2003              False
288654                   False                    2003              False
290972                    True                    2003              False
293388                   False                    2003              False
294146                   False                    2003              False
330569                   False                    2003              False
401347                    True                    2004               True

        Supervisor District  total_cko  years_registered      avg_cko
895                     NaN      20332                13  1564.000000
2007                    6.0      11142                13   857.076923
3543                    3.0      14949                13  1149.923077
20083                   3.0      16126                13  1240.461538
31334                  11.0      13858                13  1066.000000
```

```
39620          3.0   12169       13   936.076923
57244          8.0   12327       13   948.230769
86671          5.0   12711       13   977.769231
117604         4.0   14676       13  1128.923077
120565         6.0   10167       13   782.076923
129328         NaN   13257       12  1104.750000
138318         4.0   10833        6  1805.500000
145594         7.0   11921       13   917.000000
146589         4.0   24476       13  1882.769231
156774         6.0   17565       13  1351.153846
163847         9.0   25671       10  2567.100000
180148         4.0   11716       13   901.230769
200176         NaN   18403       11  1673.000000
216249         4.0   14498       13  1115.230769
222872         8.0   18019       10  1801.900000
227545         NaN   11270       13   866.923077
231752         8.0   13304       13  1023.384615
237636         6.0   15799       12  1316.583333
255022         NaN   13754       13  1058.000000
278357         8.0   11428       13   879.076923
288654        11.0   15826       13  1217.384615
290972         8.0   12793       13   984.076923
293388         NaN   36023       13  2771.000000
294146         6.0   10920       13   840.000000
330569         3.0   10785       13   829.615385
401347         NaN   15288       12  1274.000000
```

In [30]:
```
sns.countplot(x='Patron Type Definition',data=outliers)
plt.show()
```



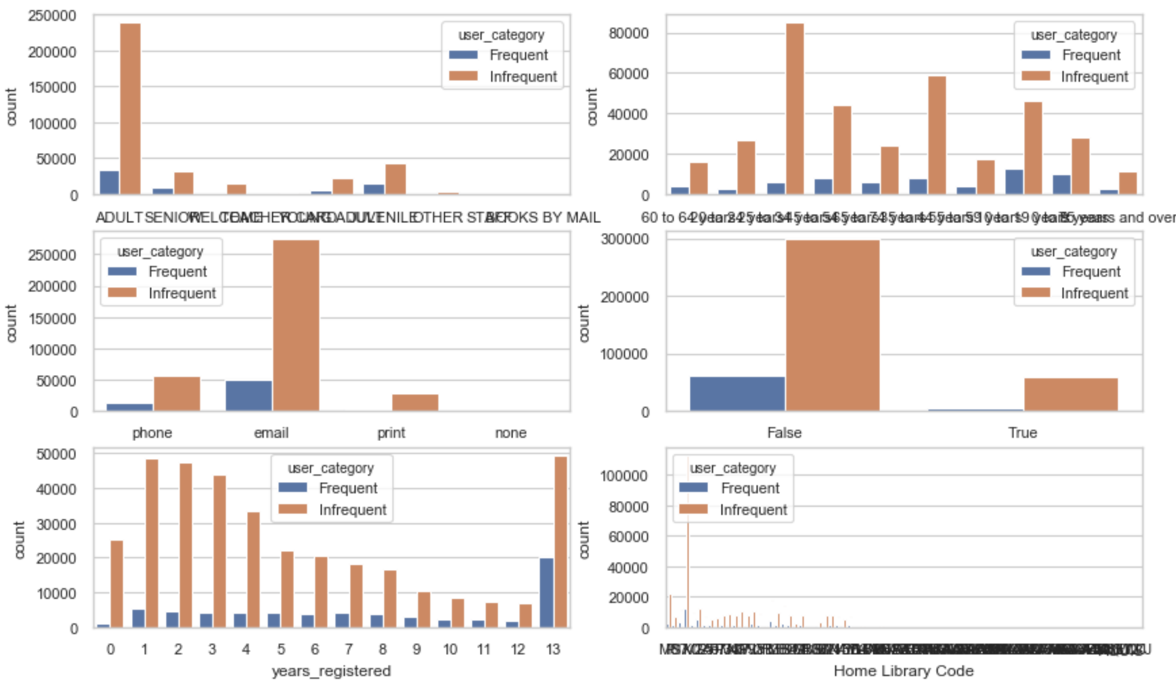
Dependent Variables

Create the label "Frequent" and "Infrequent" to use, so that we can test on using this dataset for prediction "Frequent" is defined as user with 50 times average circulation activities (avg_cko). Below are various countplots to explore the relationship between this new variable ("user_category") and the existing variables. Nothing jump out when inspecting these plots.

In [31]:
```
df['user_category']='Infrequent'
df['user_category'].loc[df['avg_cko']>50]='Frequent'
```

In [32]:
```
sns.set(style='whitegrid', rc={"grid.linewidth": 0.2})
sns.set_context("paper", font_scale=0.9)
fig, axes = plt.subplots(nrows=3, ncols=2,figsize=(10,6), dpi=100)
sns.countplot(x='Patron Type Definition',data=df,ax=axes[0][0],hue='user_categ
ory')
sns.countplot(x='Age Range',data=df,ax=axes[0][1],hue='user_category')
sns.countplot(x='Notice Preference Definition',data=df,ax=axes[1][0],hue='user
_category')
sns.countplot(x='Outside of County',data=df,ax=axes[1][1],hue='user_category')
sns.countplot(x='years_registered',data=df,ax=axes[2][0],hue='user_category')
sns.countplot(x='Home Library Code',data=df,ax=axes[2][1],hue='user_category')
```

Out[32]: `<matplotlib.axes._subplots.AxesSubplot at 0x154acb0df98>`



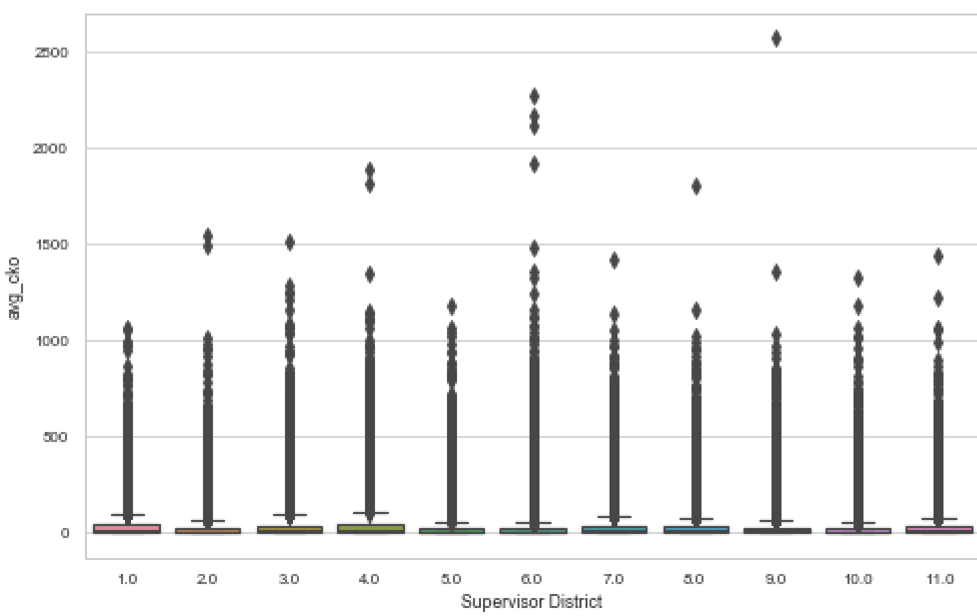
In [33]:
```
df.head()
```

Out[33]:

	Patron Type Code	Patron Type Definition	Total Checkouts	Total Renewals	Age Range	Home Library Code	Home Library Definition	Circulation Active Year	I Prefe
0	0	ADULT	1092	761	60 to 64 years	M6	Mission	2016	p
1	0	ADULT	0	0	20 to 24 years	P1	Park	None	z
2	0	ADULT	31	22	25 to 34 years	S7	Sunset	2016	z
3	0	ADULT	0	0	45 to 54 years	P1	Park	None	a
4	0	ADULT	0	0	25 to 34 years	X	Main Library	None	z

In [34]:
```
#sorted_nb = df.groupby(['Supervisor District'])['avg_cko'].median().sort_valu
es()
#sns.boxplot(x=df['Supervisor District'], y=df['avg_cko'], order=list(sorted_n
b.index))
sns.boxplot(x=df['Supervisor District'], y=df['avg_cko'])
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x154acab3470>



In [35]:
```
GB1=df.groupby(['Supervisor District'])['avg_cko'].agg(['count','median','min'
,'max'])
#G1=df.groupby(['Supervisor District'])['avg_cko'].median()
GB1.head(20)
```

Out[35]:

	count	median	min	max
Supervisor District				
1.0	26787	11.000000	0.0	1066.000000
2.0	21153	6.333333	0.0	1538.000000
3.0	22151	8.333333	0.0	1504.000000
4.0	32401	12.400000	0.0	1882.769231
5.0	28356	6.100000	0.0	1173.000000
6.0	26507	5.000000	0.0	2265.000000
7.0	30670	9.750000	0.0	1413.000000
8.0	30732	7.923077	0.0	1801.900000
9.0	31677	7.250000	0.0	2567.100000
10.0	32268	5.285714	0.0	1321.153846
11.0	30436	8.000000	0.0	1438.000000

In [36]:
```
gdf2 = gpd.read_file('data/Shapefile/Supervisor Districts as of April 2012/geo
_export_2012.shp')
```

In [37]:
```
GB2=pd.merge(gdf2,GB1,left_on="supervisor",right_on="Supervisor District",how=
"left")
GB2.head()
```

Out[37]:

	supname	supervisor	numbertext	supdist	geometry	count	median	m
0	Farrell	2.0	TWO	SUPERVISORIAL DISTRICT 2	POLYGON ((-122.41922 37.80845, -122.41921 37.8...	21153	6.333333	0
1	Mar	1.0	ONE	SUPERVISORIAL DISTRICT 1	POLYGON ((-122.49374 37.78761, -122.49367 37.7...	26787	11.000000	0
2	Tang	4.0	FOUR	SUPERVISORIAL DISTRICT 4	POLYGON ((-122.47485 37.76179, -122.47496 37.7...	32401	12.400000	0
3	Yee	7.0	SEVEN	SUPERVISORIAL DISTRICT 7	POLYGON ((-122.44854 37.75904, -122.44847 37.7...	30670	9.750000	0
4	Wiener	8.0	EIGHT	SUPERVISORIAL DISTRICT 8	POLYGON ((-122.42327 37.77206, -122.42325 37.7...	30732	7.923077	0

In [38]:
```
gdf = gpd.read_file('data/Shapefile/geo_export.shp')
print(gdf)
```

```
    supname  supervisor numbertext                   supdist  \
0   Farrell         2.0        TWO   SUPERVISORIAL DISTRICT 2
1       Mar         1.0        ONE   SUPERVISORIAL DISTRICT 1
2      Tang         4.0       FOUR   SUPERVISORIAL DISTRICT 4
3       Yee         7.0      SEVEN   SUPERVISORIAL DISTRICT 7
4    Wiener         8.0      EIGHT   SUPERVISORIAL DISTRICT 8
5    Avalos        11.0     ELEVEN  SUPERVISORIAL DISTRICT 11
6    Campos         9.0       NINE   SUPERVISORIAL DISTRICT 9
7     Cohen        10.0        TEN  SUPERVISORIAL DISTRICT 10
8       Kim         6.0        SIX   SUPERVISORIAL DISTRICT 6
9      Chiu         3.0      THREE   SUPERVISORIAL DISTRICT 3
10    Breed         5.0       FIVE   SUPERVISORIAL DISTRICT 5

                                             geometry
0   POLYGON ((-122.41922 37.80845, -122.41921 37.8...
1   POLYGON ((-122.49374 37.78761, -122.49367 37.7...
2   POLYGON ((-122.47485 37.76179, -122.47496 37.7...
3   POLYGON ((-122.44854 37.75904, -122.44847 37.7...
4   POLYGON ((-122.42327 37.77206, -122.42325 37.7...
5   POLYGON ((-122.42247 37.71789, -122.42249 37.7...
6   POLYGON ((-122.41093 37.76941, -122.41088 37.7...
7   MULTIPOLYGON (((-122.39905 37.76973, -122.3981...
8   MULTIPOLYGON (((-122.39382 37.79374, -122.3931...
9   POLYGON ((-122.39198 37.79387, -122.39218 37.7...
10  POLYGON ((-122.42157 37.78662, -122.42145 37.7...
```

In [39]:
```
# %matplotlib inline
#gdf.plot()
```

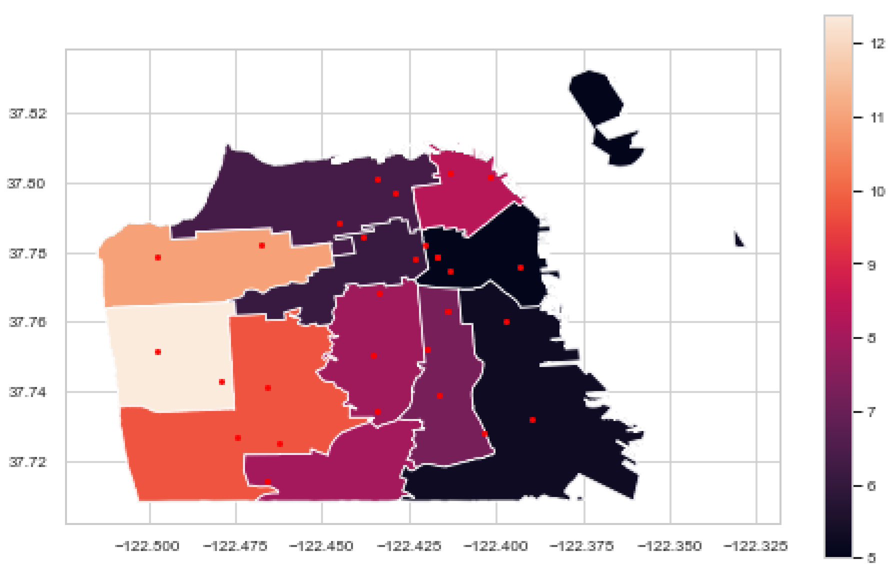
Median seems to be the most meaningful variables here, so we are plotting it with a bigger figsize. The red dot denoted the location of different branches. The other varaibles are also plotted below

In [40]:
```
sfpl_loc=pd.read_csv('data/sfpl_loc.csv')
sfpl_loc.head()
sfpl_loc=gpd.GeoDataFrame(sfpl_loc, geometry=gpd.points_from_xy(sfpl_loc.Longi
tude, sfpl_loc.Latitude))
```

In [73]:
```
#import matplotlib.pyplot as plt
f,ax=plt.subplots(1)
gdf.plot(ax=ax)
gdf3=gpd.GeoDataFrame(GB2, geometry="geometry")
#gdf.plot(column='supdist', cmap=None, legend=True, figsize=(20, 20))
gdf3.plot(ax=ax,column='median', cmap=None, legend=True, figsize=(20, 20))
sfpl_loc.plot(ax=ax, marker='o', color='red', markersize=5)
;

gdf3.head()
#gdf.plot(column='supdist', cmap=None, legend=True, figsize=(20, 20))
```

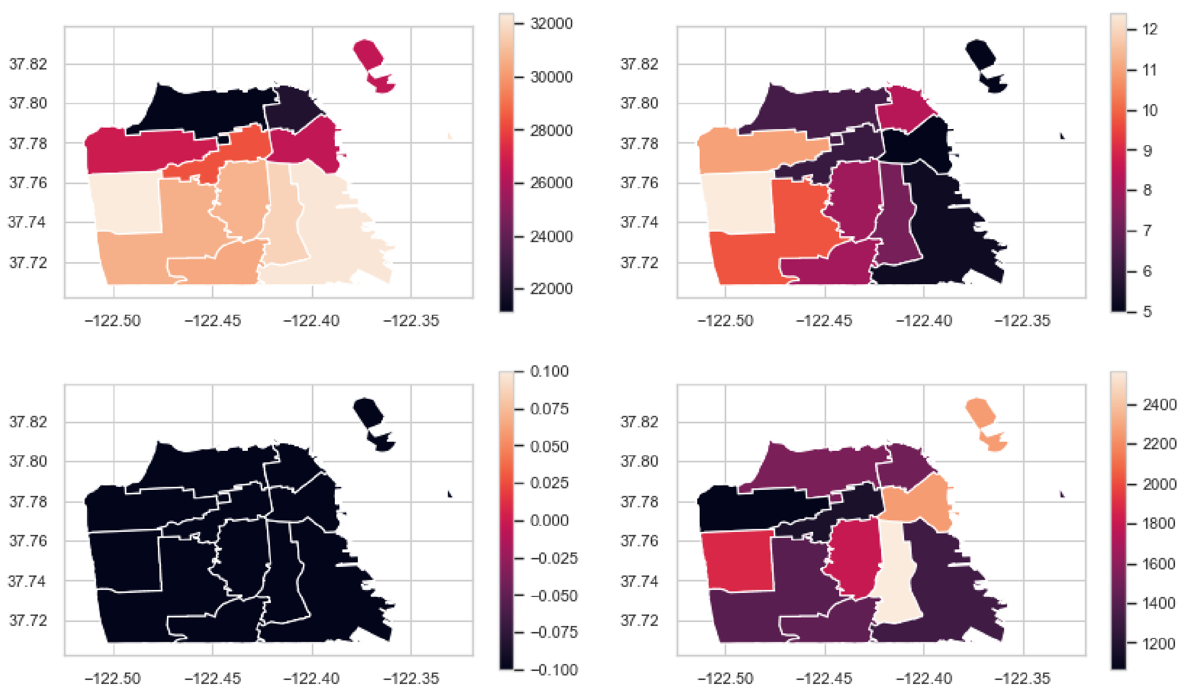
Out[73]:

	supname	supervisor	numbertext	supdist	geometry	count	median	m
0	Farrell	2.0	TWO	SUPERVISORIAL DISTRICT 2	POLYGON ((-122.41922 37.80845, -122.41921 37.8...	21153	6.333333	0
1	Mar	1.0	ONE	SUPERVISORIAL DISTRICT 1	POLYGON ((-122.49374 37.78761, -122.49367 37.7...	26787	11.000000	0
2	Tang	4.0	FOUR	SUPERVISORIAL DISTRICT 4	POLYGON ((-122.47485 37.76179, -122.47496 37.7...	32401	12.400000	0
3	Yee	7.0	SEVEN	SUPERVISORIAL DISTRICT 7	POLYGON ((-122.44854 37.75904, -122.44847 37.7...	30670	9.750000	0
4	Wiener	8.0	EIGHT	SUPERVISORIAL DISTRICT 8	POLYGON ((-122.42327 37.77206, -122.42325 37.7...	30732	7.923077	0



In [74]:
```
#import matplotlib.pyplot as plt

sns.set(style='whitegrid', rc={"grid.linewidth": 0.2})
sns.set_context("paper", font_scale=0.9)
fig, axes = plt.subplots(nrows=2, ncols=2,figsize=(10,6), dpi=100)
#sns.countplot(x='Patron Type Definition',data=df,ax=axes[0][0],hue='user_cate
gory')
#
gdf3=gpd.GeoDataFrame(GB2, geometry="geometry")

#gdf.plot(column='supdist', cmap=None, legend=True, figsize=(20, 20))
gdf3.plot(ax=axes[0][0],column='count', cmap=None, legend=True,figsize=(20, 20
))
sfpl_loc.plot(ax=ax, marker='o', color='red', markersize=5)
gdf3.plot(ax=axes[0][1],column='median', cmap=None, legend=True, figsize=(20,
20))
sfpl_loc.plot(ax=ax, marker='o', color='red', markersize=5)
gdf3.plot(ax=axes[1][0],column='min', cmap=None, legend=True, figsize=(20, 20
))
sfpl_loc.plot(ax=ax, marker='o', color='red', markersize=5)
gdf3.plot(ax=axes[1][1],column='max', cmap=None, legend=True, figsize=(20, 20
))
sfpl_loc.plot(ax=ax, marker='o', color='red', markersize=5)

#gdf.plot(column='supdist', cmap=None, legend=True, figsize=(20, 20))
```

Out[74]: <matplotlib.axes._subplots.AxesSubplot at 0x154ad215518>



In [68]:

<Figure size 576x360 with 0 Axes>

In [45]:
```
GB1=df.groupby(['Supervisor District','Outside of County'])['avg_cko'].agg(['c
ount','median','min','max'])

GB1.head(20)
```

Out[45]:

		count	median	min	max
Supervisor District	**Outside of County**				
1.0	**False**	26777	11.000000	0.000000	1066.000000
	True	10	11.833333	0.666667	549.000000
2.0	**False**	21135	6.333333	0.000000	1538.000000
	True	18	11.000000	0.000000	947.000000
3.0	**False**	22113	8.363636	0.000000	1504.000000
	True	38	4.250000	0.000000	351.000000
4.0	**False**	32382	12.400000	0.000000	1882.769231
	True	19	28.846154	0.000000	222.000000
5.0	**False**	28319	6.100000	0.000000	1173.000000
	True	37	5.333333	0.000000	553.923077
6.0	**False**	26434	5.000000	0.000000	2265.000000
	True	73	4.000000	0.000000	185.000000
7.0	**False**	30616	9.769231	0.000000	1413.000000
	True	54	4.000000	0.000000	337.615385
8.0	**False**	30709	8.000000	0.000000	1801.900000
	True	23	7.000000	0.000000	243.615385
9.0	**False**	31650	7.250000	0.000000	2567.100000
	True	27	14.000000	0.000000	334.000000
10.0	**False**	32254	5.285714	0.000000	1321.153846
	True	14	12.250000	0.000000	55.000000