CWR 2022 Data Analysis

February 8, 2023

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

[ ]:
```
data = pd.read_csv("CWR 2022 Survey Data.csv")
data = data.rename(columns={"property_mgmt": "Property Management",
                                            "recycling_location\n": "Recycling␣
 ↪Location"})
```

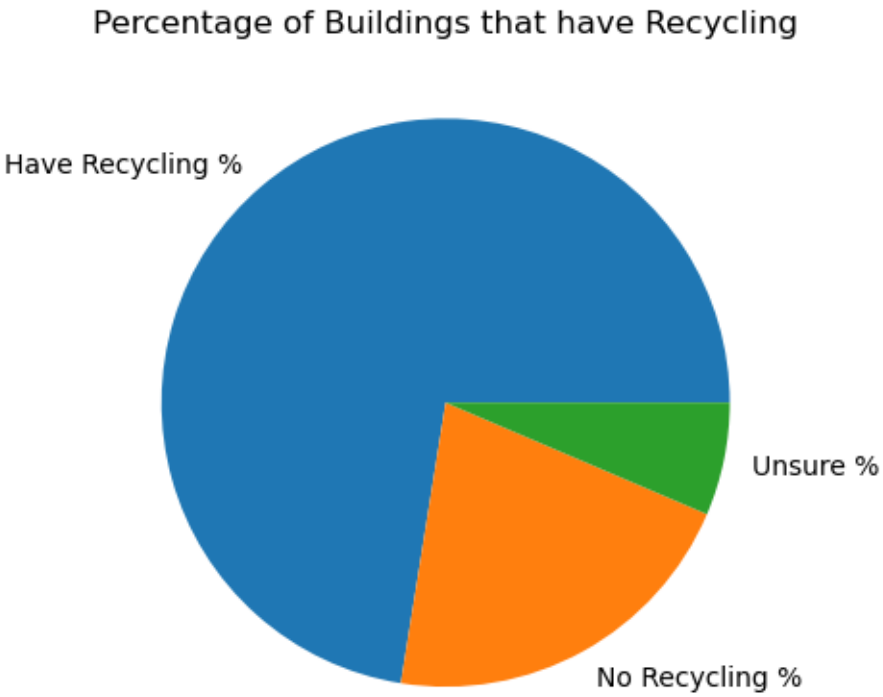
[226]:
```
#Checking the percentage of buildings that have recycling

recycling_df = data["build_have_recycling"].to_frame()
recycling_df = recycling_df.dropna()
yes_recycling = recycling_df["build_have_recycling"] == "Yes"
no_recycling = recycling_df["build_have_recycling"] == "No"
num_yes = len(recycling_df[yes_recycling])
num_no = len(recycling_df[no_recycling])
num_unsure = len(recycling_df) - num_yes - num_no
total_recycling = len(recycling_df)
percent_yes = (num_yes / total_recycling) * 100
percent_no = (num_no / total_recycling) * 100
percent_unsure = (num_unsure / total_recycling) * 100

print("Percentage of buildings that have recycling")
print("Yes: ", percent_yes, "%")
print("No: ", percent_no, "%")
print("Unsure: ", percent_unsure, "%")
```

```
Percentage of buildings that have recycling
Yes:  72.47706422018348 %
No:  21.100917431192663 %
Unsure:  6.422018348623854 %
```

[45]:
```
#Visualization of buildings that have recycling

per_recycle = [percent_yes, percent_no, percent_unsure]
plt.pie(per_recycle,labels={"Have Recycling %", "Unsure %", "No Recycling %"})
```

```
plt.title("Percentage of Buildings that have Recycling")
```

[45]: Text(0.5, 1.0, 'Percentage of Buildings that have Recycling')

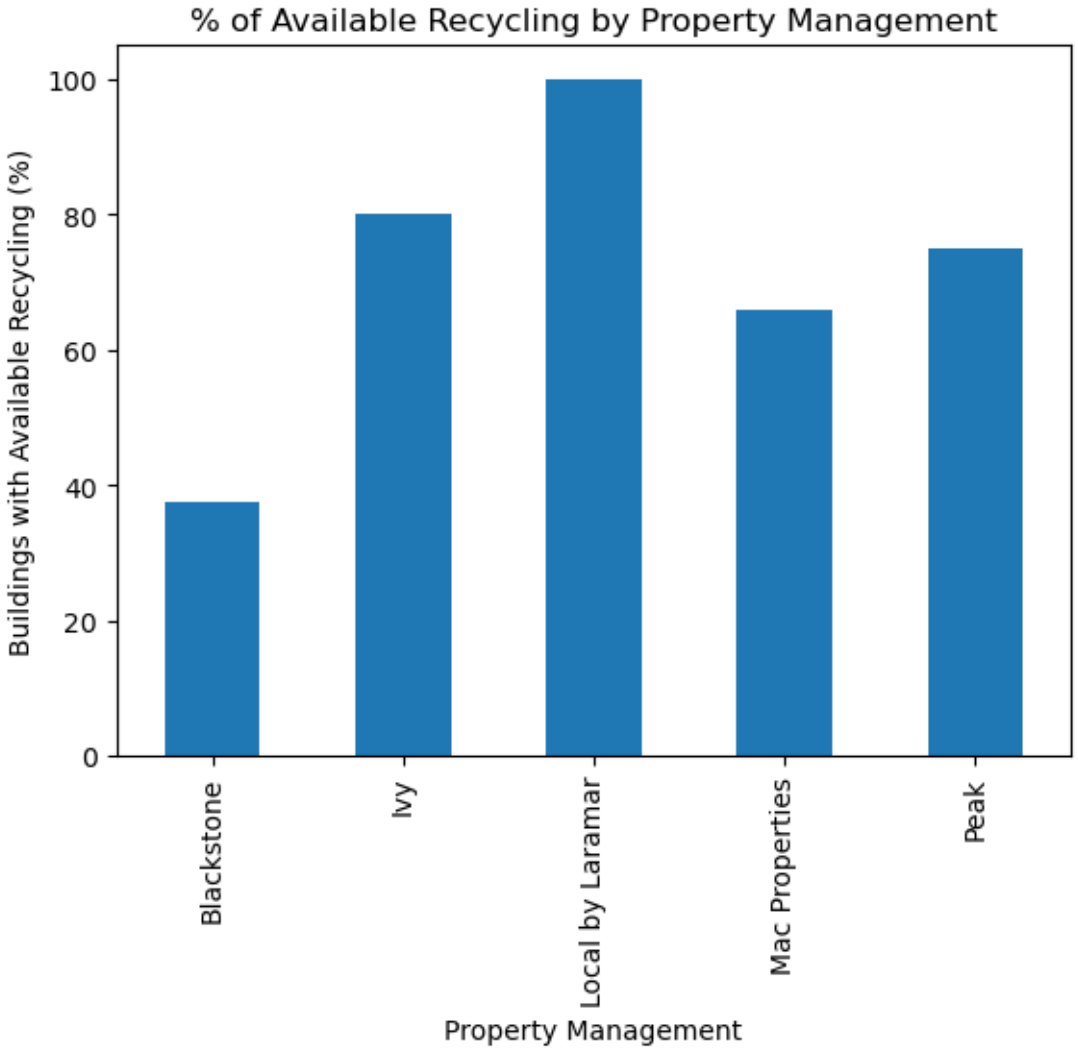


[38]:
```
#Percentage of available recycling by property management
#Percentage of people who answered "Yes" to having available recycling
#grouped by property management

no_mgmt = data[data["Property Management"] != "Other:"]
no_mgmt = no_mgmt.dropna(subset=["Property Management"])
no_mgmt = no_mgmt[no_mgmt["build_have_recycling"] == "Yes"]
ppt = no_mgmt.groupby(["Property Management"])
ppt = ppt.size().to_frame()
mgmt = data.groupby("Property Management")
mgmt = mgmt["Property Management"].size().to_frame()
mgmt = mgmt.drop(index="Other:")
mgmt["Yes"] = ppt[0]
mgmt["Percentage"] = (mgmt["Yes"] / mgmt["Property Management"]) * 100
```

[43]:
```
#Visualization of available recycling by property management
mgmt_plot = mgmt["Percentage"].plot(kind="bar", ylabel="Buildings with␣
 ↪Available Recycling (%)")
```

```
plt.title("% of Available Recycling by Property Management")
mgmt_plot
```

[43]: <AxesSubplot:title={'center':'% of Available Recycling by Property Management'}, xlabel='Property Management', ylabel='Buildings with Available Recycling (%)'>



[119]:
```
#Percentage of recycling that gets picked up regularly
#Deleted all answers that were left unanswered

pick_up_total = data.dropna(subset=["Recycling Picked Up Regularly?"])
pick_up_total = pick_up_total["Recycling Picked Up Regularly?"]
yes_pick_up = data[data["Recycling Picked Up Regularly?"] == "Yes"]
y_pu_total = len(yes_pick_up["Recycling Picked Up Regularly?"])
reg_percent = (y_pu_total / len(pick_up_total)) * 100
```

```
print("% of recycling that gets picked up regularly: ", reg_percent)
```

```
% of recycling that gets picked up regularly:  53.84615384615385
```

[227]:
```
#Property management that picks up their recycling regularly

regular_pu = data.groupby(["Property Management", "Recycling Picked Up␣
 ↪Regularly?"]).size().to_frame()

print("Whether recycling is picked up regularly by property management")
regular_pu
```

```
Whether recycling is picked up regularly by property management
```

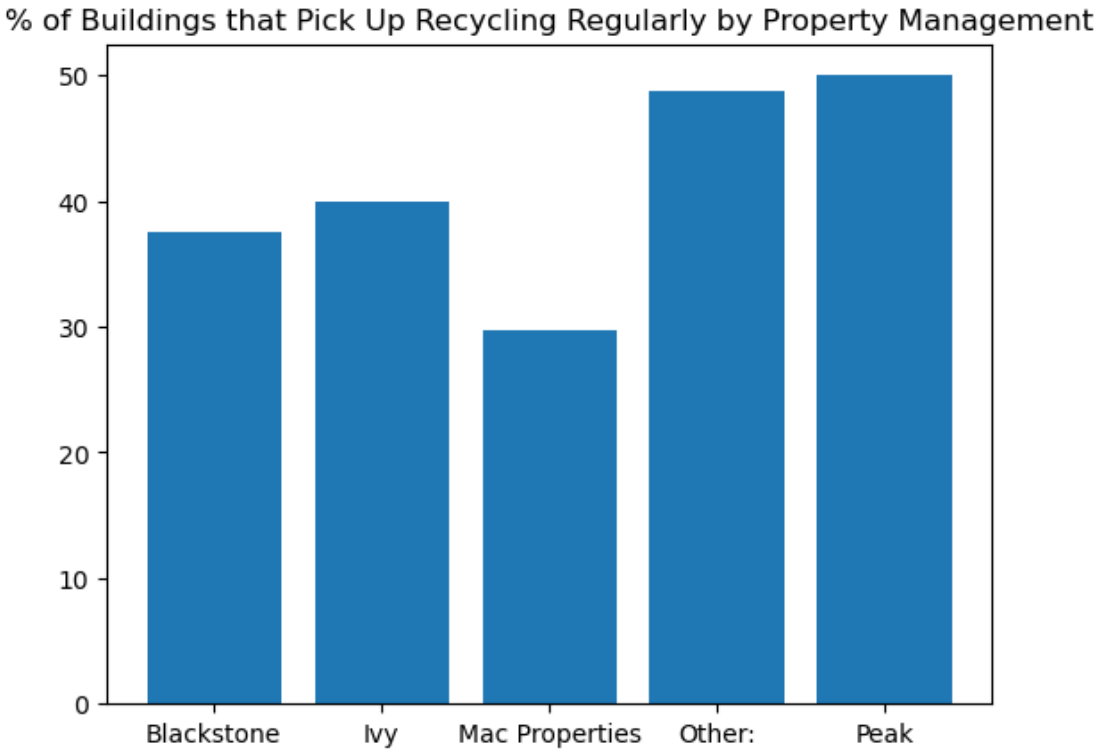
[227]:

		0
Property Management	Recycling Picked Up Regularly?	
Blackstone	Yes	3
Ivy	I'm not sure	4
	Yes	4
Local by Laramar	I'm not sure	1
Mac Properties	I'm not sure	16
	No	1
	Yes	14
Other:	I'm not sure	12
	No	1
	Yes	19
Peak	I'm not sure	1
	Yes	2

[111]:
```
#Visualization: percentage of buildings that for certain pick up recycling␣
 ↪regularly by property management

mgmt_count = data["Property Management"].value_counts().to_dict()
prop_recycle = data[data["Recycling Picked Up Regularly?"] == "Yes"]
prop_recycle = prop_recycle.groupby("Property Management").size().to_dict()

percent_recycle = {}

for prop, num in prop_recycle.items():
    percent_recycle[prop] = (num / mgmt_count[prop]) * 100

mgmt_names = list(percent_recycle.keys())
mgmt_per = list(percent_recycle.values())

plt.bar(range(len(percent_recycle)), mgmt_per, tick_label=mgmt_names)
```

```
plt.title("% of Buildings that Pick Up Recycling Regularly by Property␣
 ↪Management")
```

[111]:
```
Text(0.5, 1.0, '% of Buildings that Pick Up Recycling Regularly by Property
Management')
```



[228]:
```
#Frequency of pick up

freq = data.dropna(subset=["Picked Up By Week"])
freq = freq["Picked Up By Week"]
freq_replace = {"twice a week": "2/week", "every week?": "1/week",
                "once a week" : "1/week", "1": "1/week", "2": "2/week", "2-3":␣
 ↪"2-3/week"}
freq = freq.replace(freq_replace)
total_freq = len(freq)
freq = freq.value_counts().to_frame()
freq = (freq / total_freq) * 100
freq = freq.rename(columns={"Picked Up By Week": "Percentage"})

print("Frequency of Recycling Pick Up")
freq
```

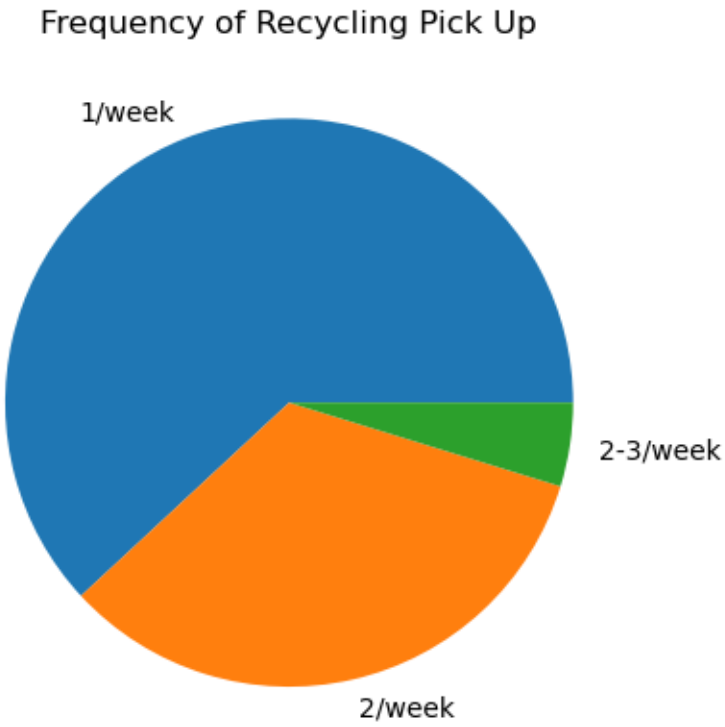
```
Frequency of Recycling Pick Up
```

[228]:

	Percentage
1/week	61.904762
2/week	33.333333
2-3/week	4.761905

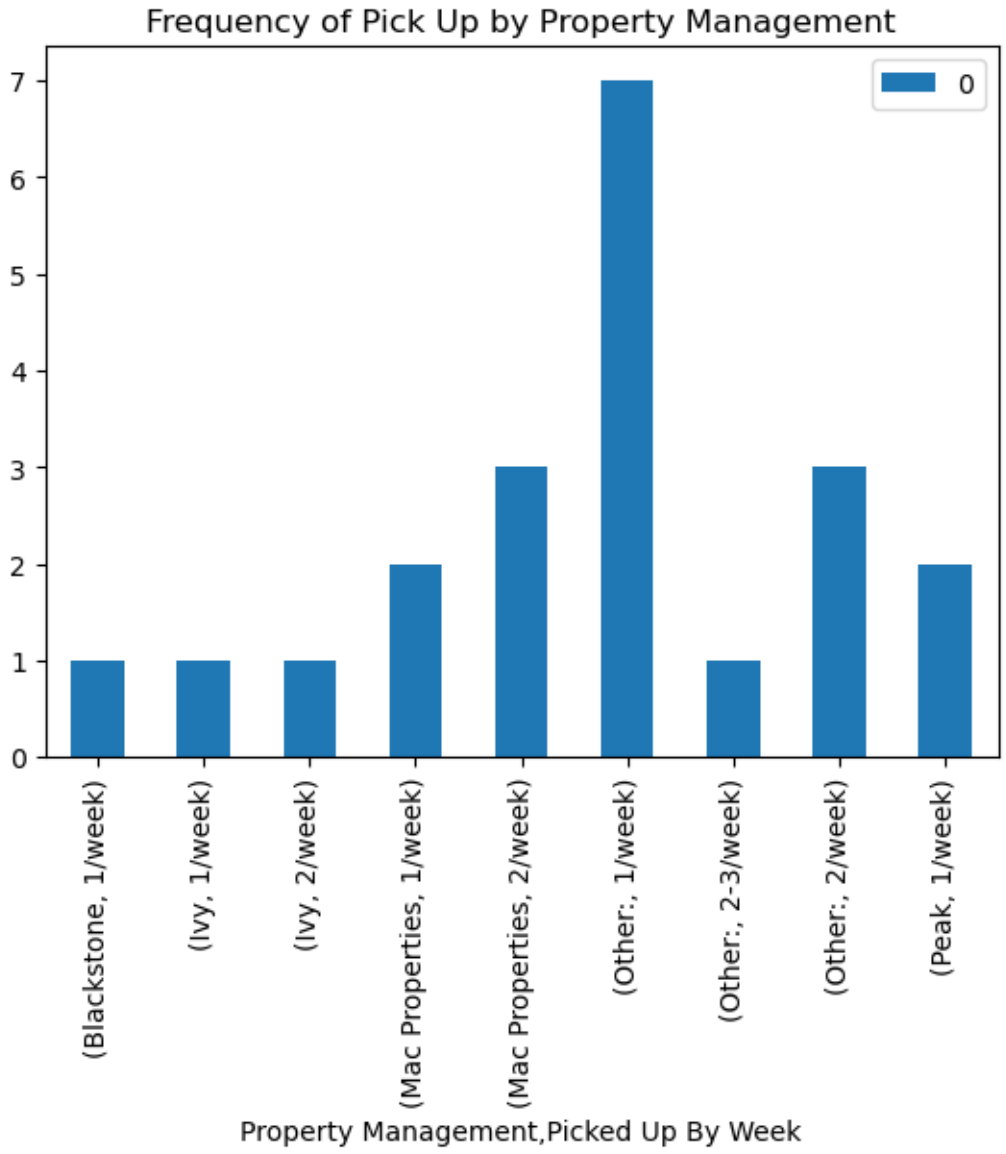
[116]:
```
#Visualization of frequency of pick up

plt.pie(freq["Percentage"], labels=["1/week", "2/week", "2-3/week"])
plt.title("Frequency of Recycling Pick Up")
```

[116]:
```
Text(0.5, 1.0, 'Frequency of Recycling Pick Up')
```



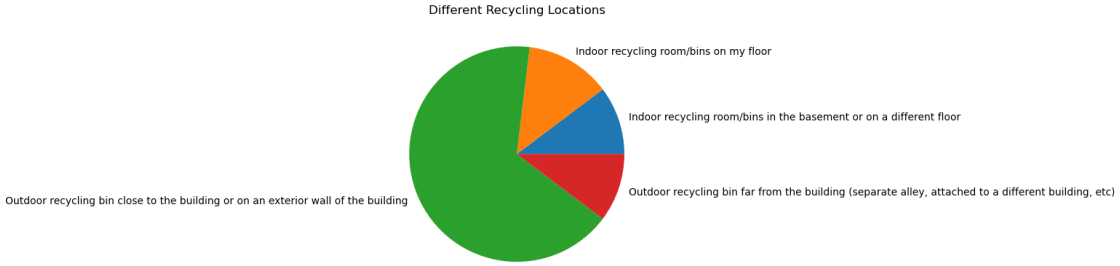
[148]:
```
#Frequency of pick up by property management

freq_ppt = data.dropna(subset=["Picked Up By Week"])
freq_ppt = freq_ppt.replace(freq_replace)
freq_ppt = freq_ppt.groupby(["Property Management", "Picked Up By Week"]).
 ↪size().to_frame()

freq_ppt.plot(kind="bar")
plt.title("Frequency of Pick Up by Property Management")
```

[148]: Text(0.5, 1.0, 'Frequency of Pick Up by Property Management')



[167]:
```
#Recycling locations

loc = data.dropna(subset=["Recycling Location"])
loc = loc["Recycling Location"].to_frame()
loc = loc.groupby("Recycling Location").size().to_frame()

locations = list(loc.index)
plt.pie(loc[0], labels=locations)
plt.title("Different Recycling Locations")
```

[167]: Text(0.5, 1.0, 'Different Recycling Locations')



[190]:
```
#Recycling location by property management

location = data.copy()
location = location.rename(columns={"property_mgmt": "Property Management",
                                    "recycling_location\n": "Recycling␣
 ↪Location"})
location = location.groupby(["Property Management", "Recycling Location"])
loc_by_mgmt = location.size().to_frame()

print("Recycling location by property management")
loc_by_mgmt
```

```
Recycling location by property management
```

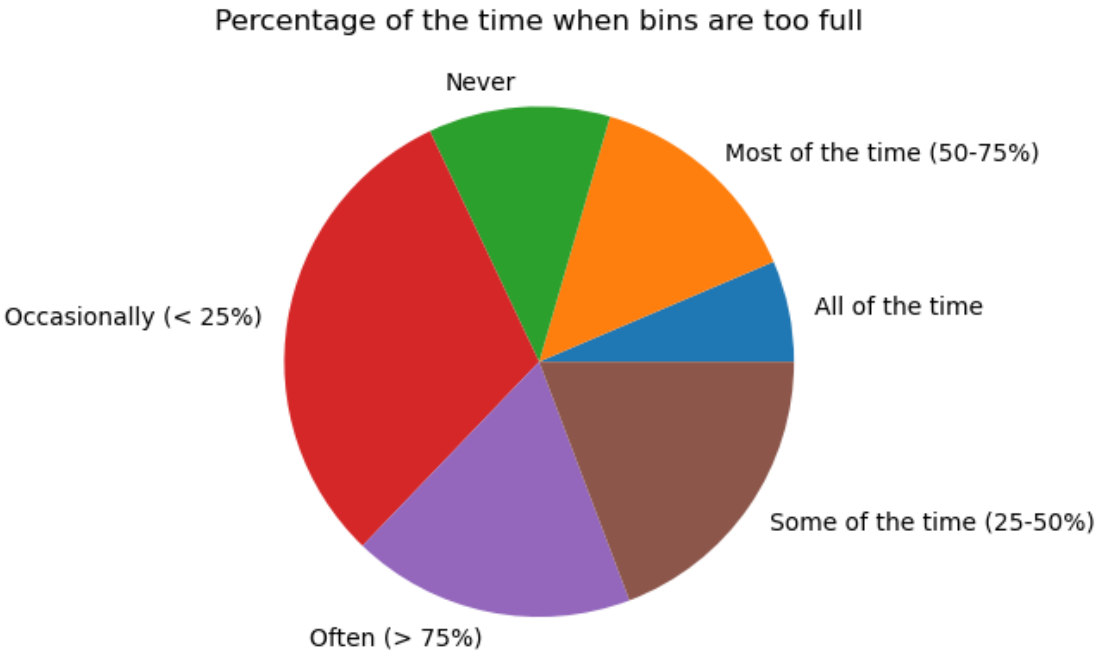
[190]:

Property Management	Recycling Location	0
Blackstone	Outdoor recycling bin close to the building or …	2
	Outdoor recycling bin far from the building (se…	1
Ivy	Outdoor recycling bin close to the building or …	7
	Outdoor recycling bin far from the building (se…	1
Local by Laramar	Indoor recycling room/bins in the basement or o…	1
Mac Properties	Indoor recycling room/bins on my floor	5
	Outdoor recycling bin close to the building or …	24
	Outdoor recycling bin far from the building (se…	2
Other:	Indoor recycling room/bins in the basement or o…	7
	Indoor recycling room/bins on my floor	5
	Outdoor recycling bin close to the building or …	16
	Outdoor recycling bin far from the building (se…	4
Peak	Outdoor recycling bin close to the building or …	3

[194]:
```
#Bins too full and their frequency

bins = data.dropna(subset=["Recycling Bins Too Full"])
total_bins = len(bins)
```

```
bins = bins.groupby("Recycling Bins Too Full").size().to_frame()
bins = (bins / total_bins) * 100

plt.pie(bins[0], labels=list(bins.index))
plt.title("Percentage of the time when bins are too full")
```

[194]: Text(0.5, 1.0, 'Percentage of the time when bins are too full')



[259]:
```
#Whether property management have enough recycling bins

enough_bins = data.groupby(["Property Management", "Enough Recycling Bins?"])
enough_bins = enough_bins.size().to_frame()

print("Whether every property management has enough recycling bins or not")
enough_bins
```

Whether every property management has enough recycling bins or not

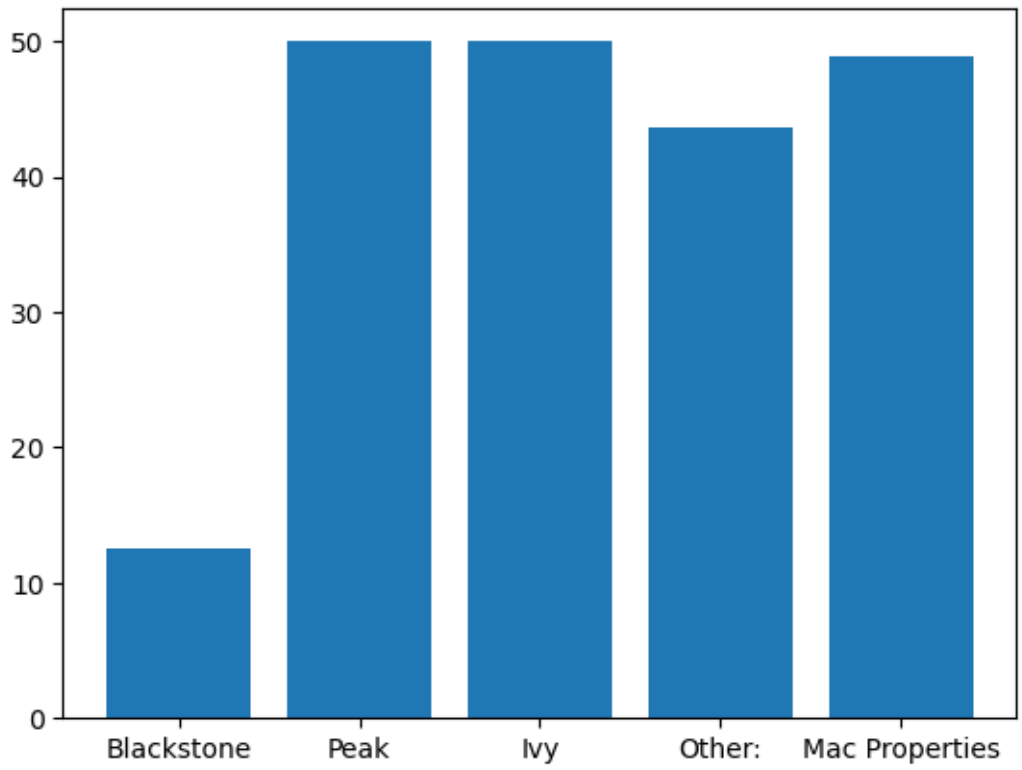
[259]:

		0
Property Management	Enough Recycling Bins?	
Blackstone	No	2
	Yes	1
Ivy	I don't know	1
	No	2
	Yes	5

```
Local by Laramar    I don't know            1
Mac Properties      I don't know            1
                    No                      7
                    Yes                    23
Other:              I don't know            5
                    No                     10
                    Yes                    17
Peak                I don't know            1
                    Yes                     2
```

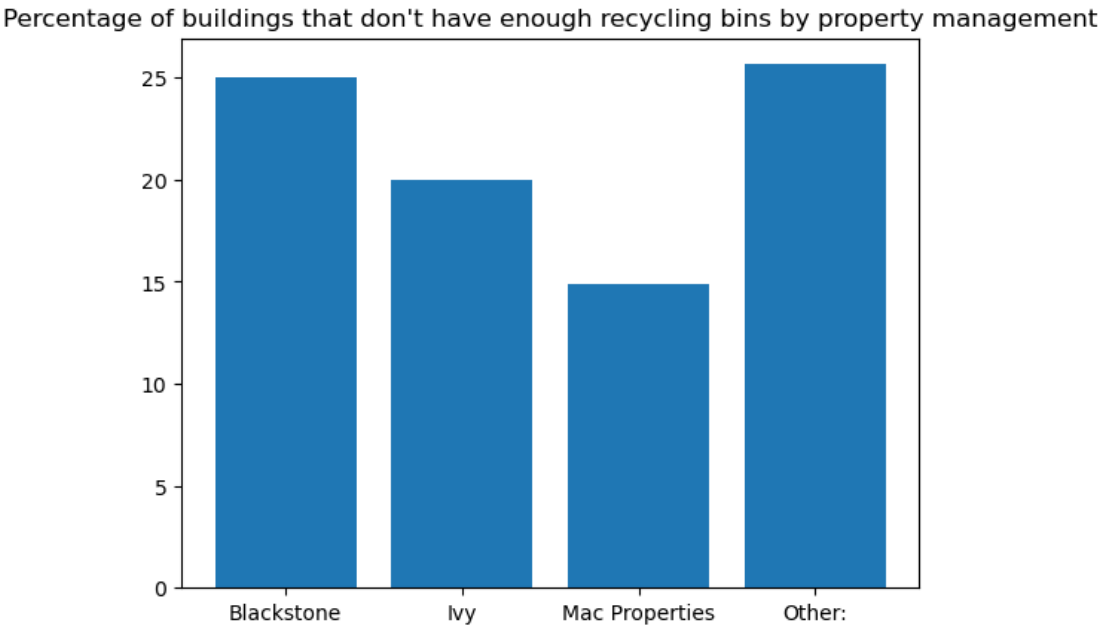
[191]:
```
#Percentage of properties that have enough recycling bins sorted grouped by␣
 ↪property management
#People who answered "Yes" to "Enough Recycling Bins?"

yes_df = data[data["Enough Recycling Bins?"] == "Yes"]
yes_df = yes_df.groupby("Property Management").size().sort_values().to_dict()

yes_dict = {}
for prop, val in yes_df.items():
    yes_dict[prop] = (val / mgmt_count[prop]) * 100

yes_names = list(yes_dict.keys())
yes_per = list(yes_dict.values())

plt.bar(range(len(yes_dict)), yes_per, tick_label=yes_names)
plt.title("Percentage of buildings that have enough recycling bins by property␣
 ↪management")
```

[191]:
```
<BarContainer object of 5 artists>
```



[193]:
```
#Percentage of properties that have enough recycling bins sorted grouped by␣
 ↪property management
#People who answered "No" to "Enough Recycling Bins?"

no_df = data[data["Enough Recycling Bins?"] == "No"]
no_df = no_df.groupby("Property Management").size().sort_values().to_dict()

no_dict = {}
for prop, val in no_df.items():
    no_dict[prop] = (val / mgmt_count[prop]) * 100

no_names = list(no_dict.keys())
no_per = list(no_dict.values())

plt.bar(range(len(no_dict)), no_per, tick_label=no_names)
plt.title("Percentage of buildings that don't have enough recycling bins by␣
 ↪property management")
```

[193]:
```
Text(0.5, 1.0, "Percentage of buildings that don't have enough recycling bins by
property management")
```



[229]:
```
#Recycling too full by property management

too_full = data.groupby(["Property Management", "Recycling Bins Too Full"]).
 ↪size().to_frame()

print("Property management and whether their reycling is too full")
too_full
```

```
Property management and whether their reycling is too full
```

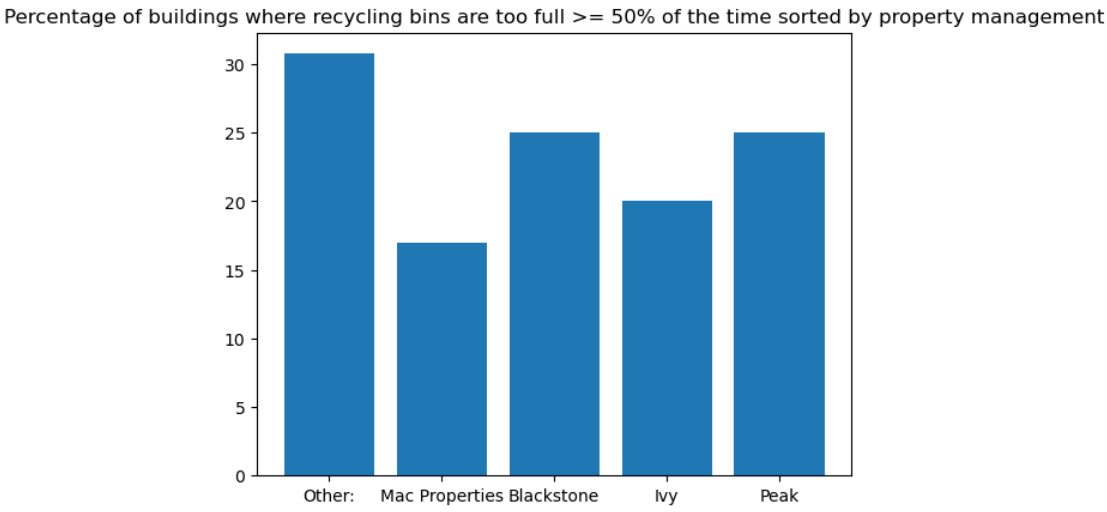
[229]:

		0
Property Management	Recycling Bins Too Full	
Blackstone	Most of the time (50-75%)	2
	Some of the time (25-50%)	1
Ivy	Most of the time (50-75%)	2
	Never	2
	Occasionally (< 25%)	1
	Some of the time (25-50%)	3
Local by Laramar	Occasionally (< 25%)	1
Mac Properties	All of the time	4
	Most of the time (50-75%)	3
	Never	4
	Occasionally (< 25%)	10
	Often (> 75%)	5
	Some of the time (25-50%)	5
Other:	All of the time	1

```
                        Most of the time (50-75%)   4
                        Never                       2
                        Occasionally (< 25%)       12
                        Often (> 75%)               8
                        Some of the time (25-50%)   5
Peak                    Never                       1
                        Often (> 75%)               1
                        Some of the time (25-50%)   1
```

[211]:
```
#Visualization of the percentage of the buildings where recycling bins
#are too full most/often (>= 50%) sorted by property management

most = data["Recycling Bins Too Full"] == "Most of the time (50-75%)"
often = data["Recycling Bins Too Full"] == "Often (> 75%)"
too_full = data[most | often]

too_full = too_full["Property Management"].value_counts().to_dict()

too_full_dict = {}
for prop, val in too_full.items():
    too_full_dict[prop] = (val / mgmt_count[prop]) * 100

too_full_names = list(too_full_dict.keys())
too_full_per = list(too_full_dict.values())

plt.bar(range(len(too_full_dict)), too_full_per, tick_label=too_full_names)
plt.title("Percentage of buildings where recycling bins are too full >= 50% of␣
 ↪the time sorted by property management")
```

[211]:
```
Text(0.5, 1.0, 'Percentage of buildings where recycling bins are too full >= 50%
of the time sorted by property management')
```



[272]:
```
#Building type and whether they have recycling or not

type_recycling = data.groupby(["building_type", "build_have_recycling"]).size().
 ↪to_frame()

print("Type of building and whether they have recycling or not")
type_recycling
```

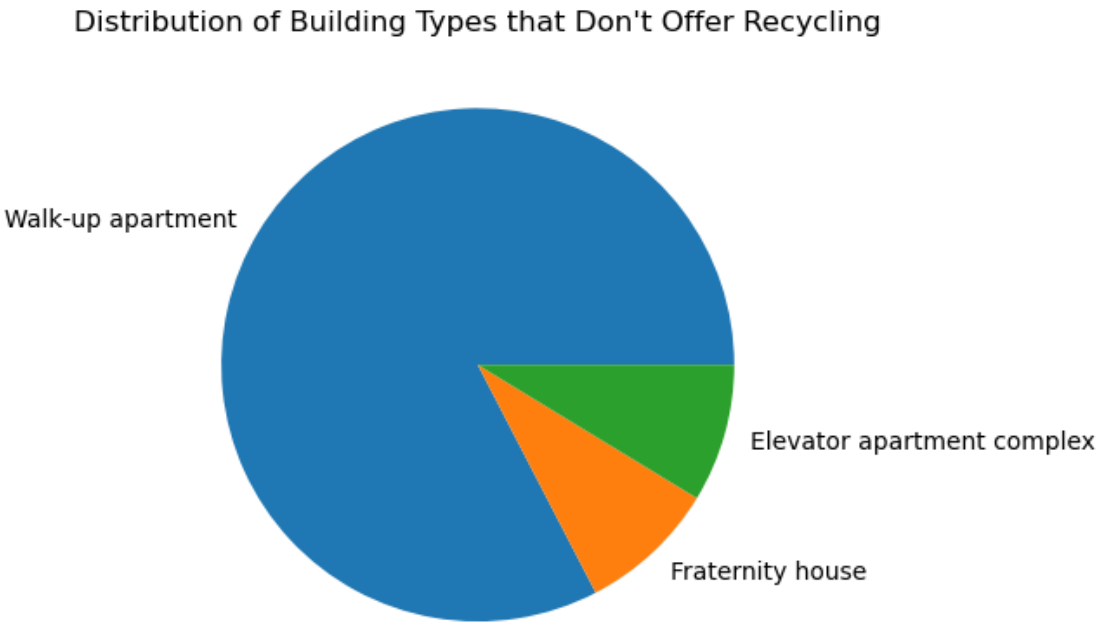
```
Type of building and whether they have recycling or not
```

[272]:

		0
building_type	build_have_recycling	
Elevator apartment complex	No	2
	Yes	20
Fraternity house	No	2
Single-residence home	Yes	1
Walk-up apartment	I don't know	7
	No	19
	Yes	58

[225]:
```
#Visualization of the distribution of building types that don't have recycling

recycle_build = data[data["build_have_recycling"] == "No"]
recycle_build = recycle_build["building_type"].value_counts().to_dict()

build_type = list(recycle_build.keys())
no_recycle_val = list(recycle_build.values())

plt.pie(no_recycle_val, labels=build_type)
plt.title("Distribution of Building Types that Don't Offer Recycling")
```

[225]:
```
Text(0.5, 1.0, "Distribution of Building Types that Don't Offer Recycling")
```



[274]:
```
#Building type and where they offer their recycling

rec_loc = data.groupby(["building_type", "Recycling Location"]).size().
 ↪to_frame()

print("Building type and where they offer recycling")
rec_loc
```

```
Building type and where they offer recycling
```

[274]:
```
0
building_type              Recycling Location
Elevator apartment complex Indoor recycling room/bins in the basement or o…
7
                           Indoor recycling room/bins on my floor
10
                           Outdoor recycling bin close to the building or …
2
                           Outdoor recycling bin far from the building (se…
1
Single-residence home      Outdoor recycling bin close to the building or …
1
Walk-up apartment          Indoor recycling room/bins in the basement or o…
1
```

```
                          Outdoor recycling bin close to the building or …
49
                          Outdoor recycling bin far from the building (se…
7
```

[ ]: