In [40]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

In [ ]:
```
```

In [41]:
```
# 1. Read in the coupons.csv file
df = pd.read_csv('data/coupons.csv')
```

In [42]:
```
df.head()
```

Out[42]:

	destination	passanger	weather	temperature	time	coupon	expiration	gen
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Fem
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Fem
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Fem
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Fem
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Fem

5 rows × 26 columns

In [43]:
```
# 2. Investigate the dataset for missing or problematic data.
for index in df.columns:
    print(df[index].value_counts())
    print("\n")
bad_data = df.isna().sum()
bad_data.plot(kind='bar')
plt.xlabel('Features')
plt.ylabel('Bad Data Count')
plt.title('UCI Features Bad Data Count')
plt.show()

print("\n\n")
print("Feature 'car' have too many error entries, remove column from calcula
print("\n\n")

df = df.drop( columns=['car'] )

bad_data = df.isna().sum()
bad_data.plot(kind='bar')
plt.xlabel('Features')
plt.ylabel('Bad Data Count')
plt.title('UCI Features Bad Data Count')
plt.show()
```

```
df_data = df.dropna().copy()
df_p = df_data.count()

df_p.plot(kind='bar')
plt.xlabel('Features')
plt.ylabel('Good Data Count')
plt.title('UCI Features Cleaned Up')
plt.show()

print(f'************************************************************')
print(f'* Q: Investigate and clean up problematic data             *')
print(f'************************************************************')
print(f'* A: Bar plot of count of NaN or Null values for each      *')
print(f'*    and found out that feature "car" is predominately     *')
print(f'     null values, and removed (drop) from the dataframe. *')
print(f'     The entire DataFrame data is then removed of NaN.     *')
print(f'************************************************************')
```

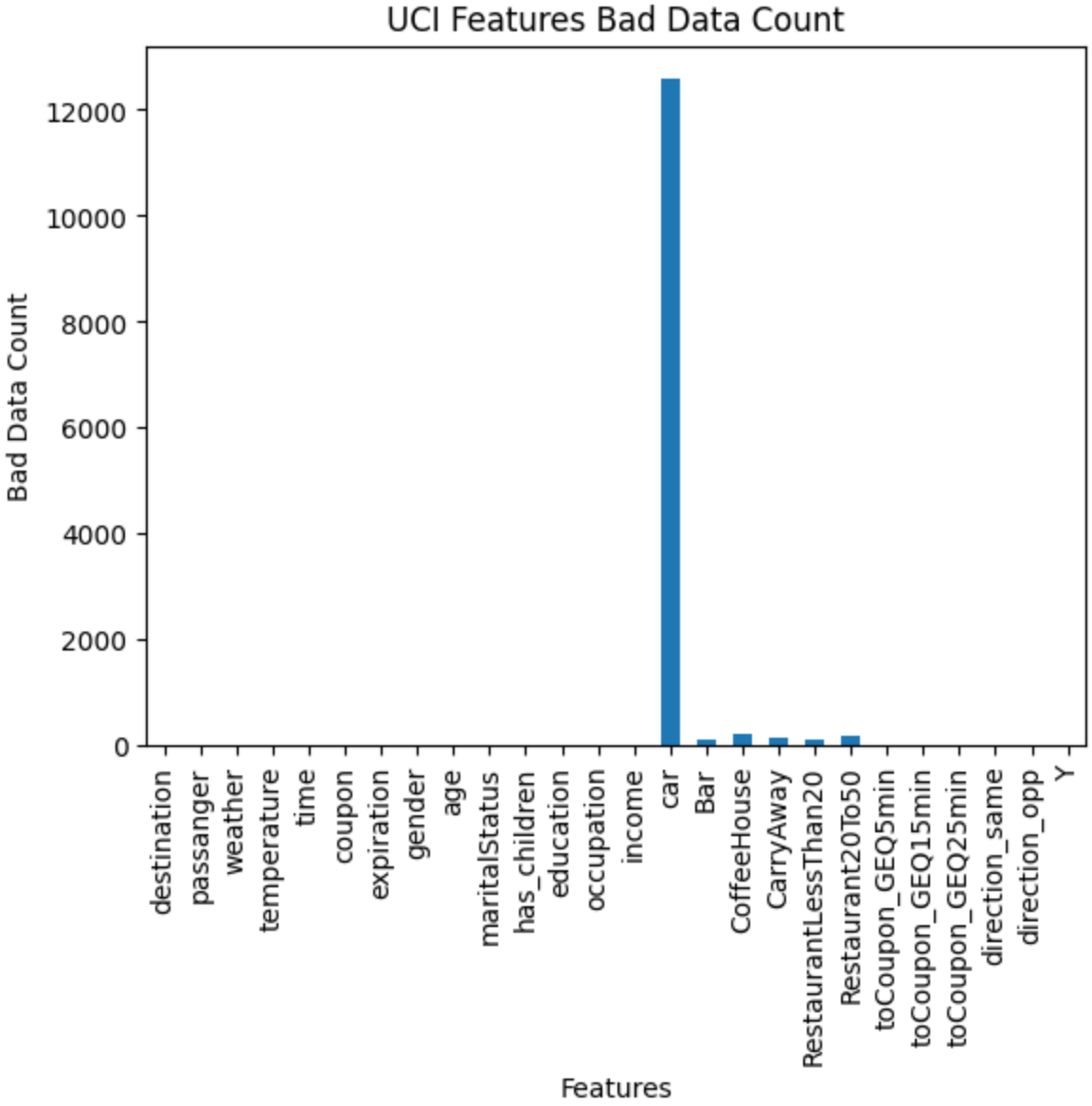
```
destination
No Urgent Place    6283
Home               3237
Work               3164
Name: count, dtype: int64


passanger
Alone        7305
Friend(s)    3298
Partner      1075
Kid(s)       1006
Name: count, dtype: int64


weather
Sunny    10069
Snowy     1405
Rainy     1210
Name: count, dtype: int64


temperature
80    6528
55    3840
30    2316
Name: count, dtype: int64


time
6PM     3230
7AM     3164
10AM    2275
2PM     2009
10PM    2006
Name: count, dtype: int64


coupon
Coffee House             3996
Restaurant(<20)          2786
Carry out & Take away    2393
Bar                      2017
Restaurant(20-50)        1492
Name: count, dtype: int64


expiration
1d    7091
2h    5593
Name: count, dtype: int64


gender
Female    6511
Male      6173
```

```
Name: count, dtype: int64


age
21         2653
26         2559
31         2039
50plus     1788
36         1319
41         1093
46          686
below21     547
Name: count, dtype: int64


maritalStatus
Married partner      5100
Single               4752
Unmarried partner    2186
Divorced              516
Widowed               130
Name: count, dtype: int64


has_children
0    7431
1    5253
Name: count, dtype: int64


education
Some college - no degree                 4351
Bachelors degree                         4335
Graduate degree (Masters or Doctorate)   1852
Associates degree                        1153
High School Graduate                      905
Some High School                           88
Name: count, dtype: int64


occupation
Unemployed                                 1870
Student                                    1584
Computer & Mathematical                    1408
Sales & Related                            1093
Education&Training&Library                  943
Management                                  838
Office & Administrative Support             639
Arts Design Entertainment Sports & Media    629
Business & Financial                        544
Retired                                     495
Food Preparation & Serving Related          298
Healthcare Practitioners & Technical        244
Healthcare Support                          242
Community & Social Services                 241
Legal                                       219
```

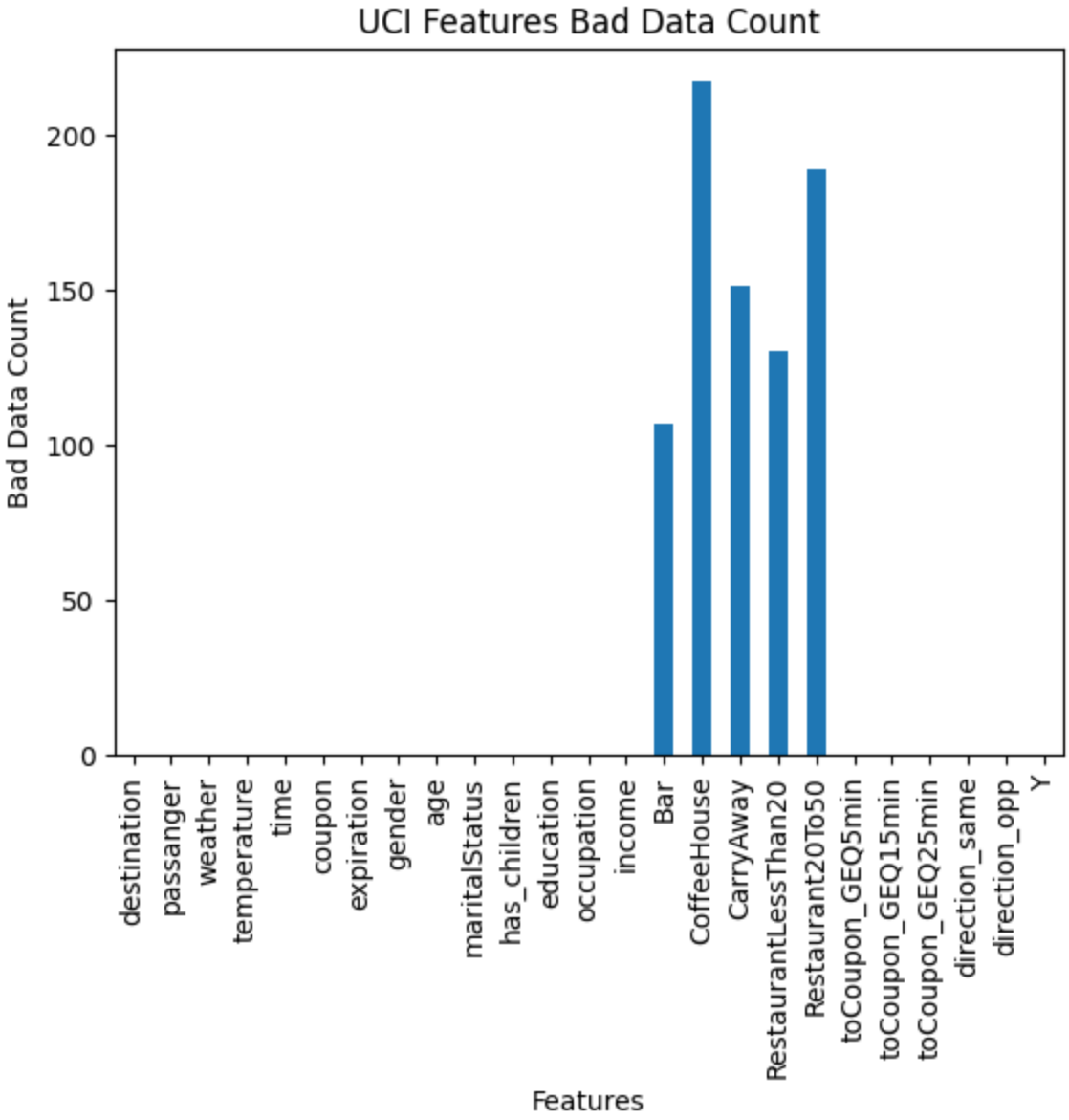
```
Transportation & Material Moving             218
Architecture & Engineering                   175
Personal Care & Service                      175
Protective Service                           175
Life Physical Social Science                 170
Construction & Extraction                    154
Installation Maintenance & Repair            133
Production Occupations                       110
Building & Grounds Cleaning & Maintenance     44
Farming Fishing & Forestry                    43
Name: count, dtype: int64


income
$25000 - $37499     2013
$12500 - $24999     1831
$37500 - $49999     1805
$100000 or More     1736
$50000 - $62499     1659
Less than $12500    1042
$87500 - $99999      895
$75000 - $87499      857
$62500 - $74999      846
Name: count, dtype: int64


car
Scooter and motorcycle                      22
Mazda5                                      22
do not drive                                22
crossover                                   21
Car that is too old to install Onstar :D    21
Name: count, dtype: int64


Bar
never    5197
less1    3482
1~3      2473
4~8      1076
gt8       349
Name: count, dtype: int64


CoffeeHouse
less1    3385
1~3      3225
never    2962
4~8      1784
gt8      1111
Name: count, dtype: int64


CarryAway
1~3      4672
4~8      4258
```

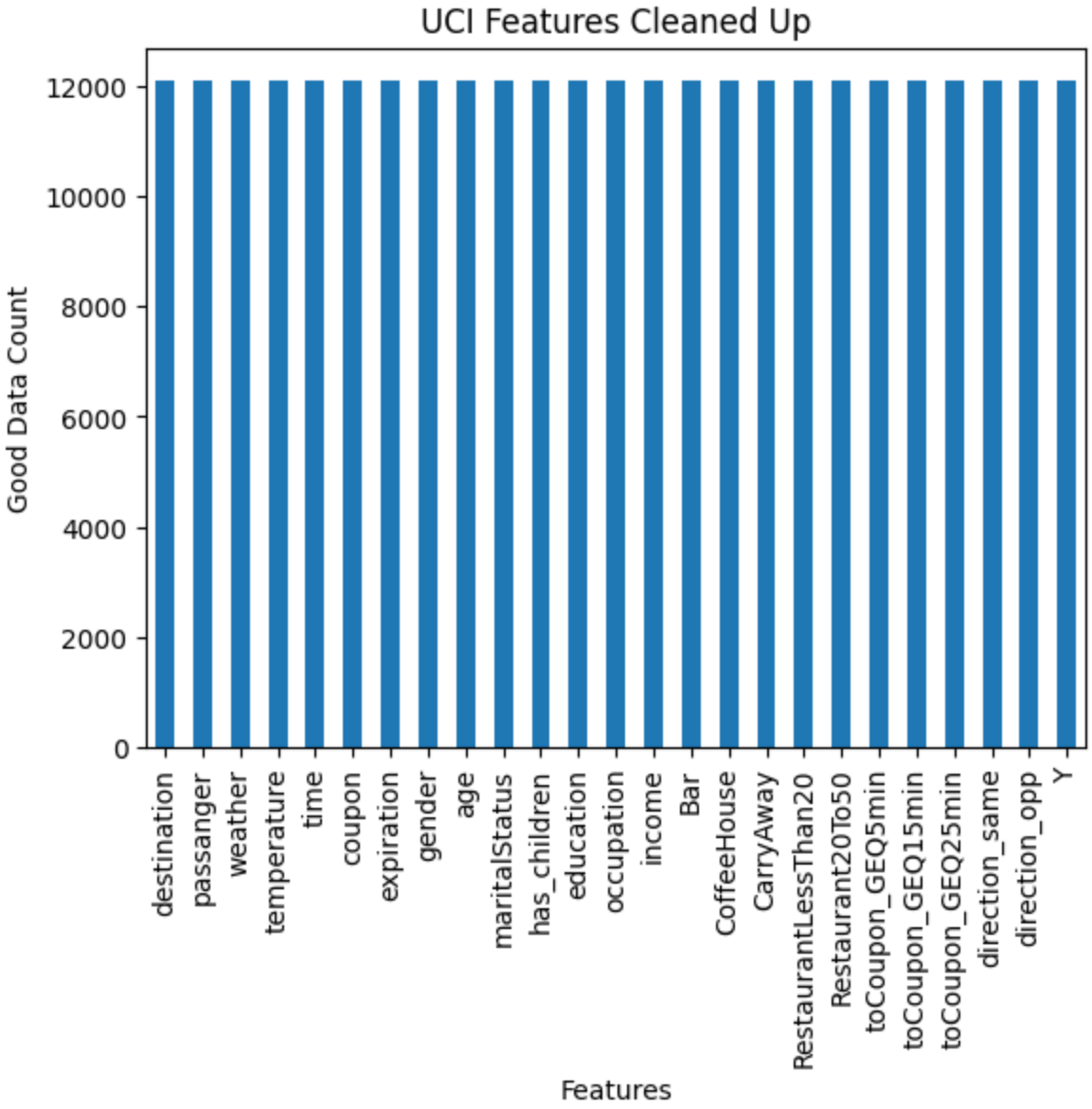
```
less1    1856
gt8      1594
never     153
Name: count, dtype: int64


RestaurantLessThan20
1~3      5376
4~8      3580
less1    2093
gt8      1285
never     220
Name: count, dtype: int64


Restaurant20To50
less1    6077
1~3      3290
never    2136
4~8       728
gt8       264
Name: count, dtype: int64


toCoupon_GEQ5min
1    12684
Name: count, dtype: int64


toCoupon_GEQ15min
1    7122
0    5562
Name: count, dtype: int64


toCoupon_GEQ25min
0    11173
1     1511
Name: count, dtype: int64


direction_same
0    9960
1    2724
Name: count, dtype: int64


direction_opp
1    9960
0    2724
Name: count, dtype: int64


Y
1    7210
0    5474
```

```
Name: count, dtype: int64
```




```
Feature 'car' have too many error entries, remove column from calculations
```


UCI Features Bad Data Count
Bad Data Count
0
50
100
150
200
destination
passanger
weather
temperature
time
coupon
expiration
gender
age
maritalStatus
has_children
education
occupation
income
Bar
CoffeeHouse
CarryAway
RestaurantLessThan20
Restaurant20To50
toCoupon_GEQ5min
toCoupon_GEQ15min
toCoupon_GEQ25min
direction_same
direction_opp
Y
Features




```
************************************************************
* Q: Investigate and clean up problematic data             *
************************************************************
* A: Bar plot of count of NaN or Null values for each      *
*    and found out that feature "car" is predominately     *
     null values, and removed (drop) from the dataframe.   *
     The entire DataFrame data is then removed of NaN.     *
************************************************************
```

In [44]:
```
# 3. Decide what to do about your missing data -- drop, replace, other...
print("\n\nQuestion #3: Missing Data Decision:  Drop 'car' column, and drop
```

```
Question #3: Missing Data Decision:  Drop 'car' column, and drop NA
```

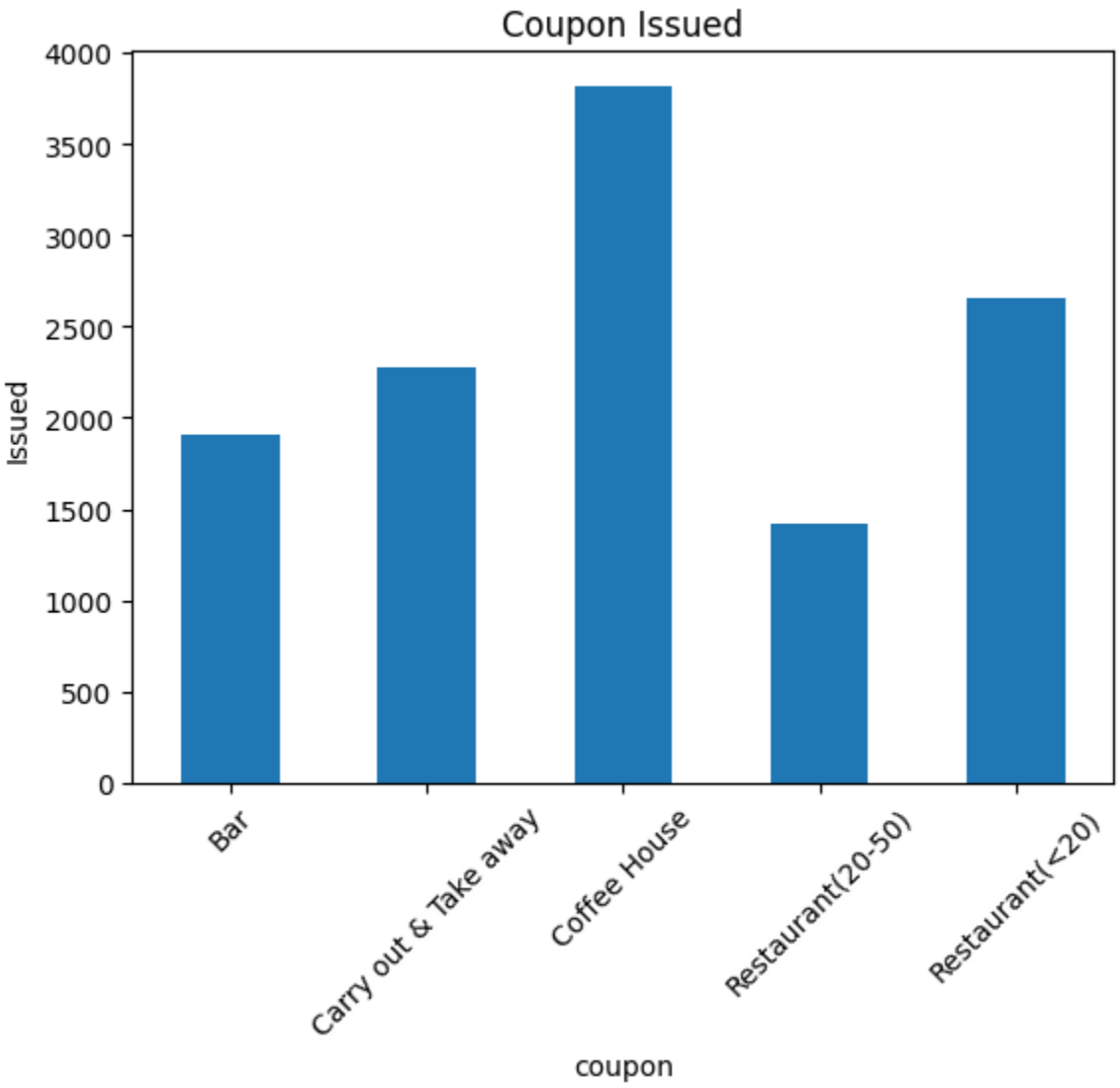
In [45]:
```
# 4. What proportion of the total observations chose to accept the coupon?
total_accept = df_data.groupby('Y')['Y'].count()

acceptance_rate = total_accept[1] / df_data.shape[0] * 100
```

```
print(f'*************************************************************************
print(f'* Q: What proportion of the total observations chose to accept the c
print(f'*************************************************************************
print(f'* A: count the total of coupon accepted and divide by total observat
print(f'*    Coupon Acceptance Rate: {acceptance_rate:.2f}%
print(f'*************************************************************************
```

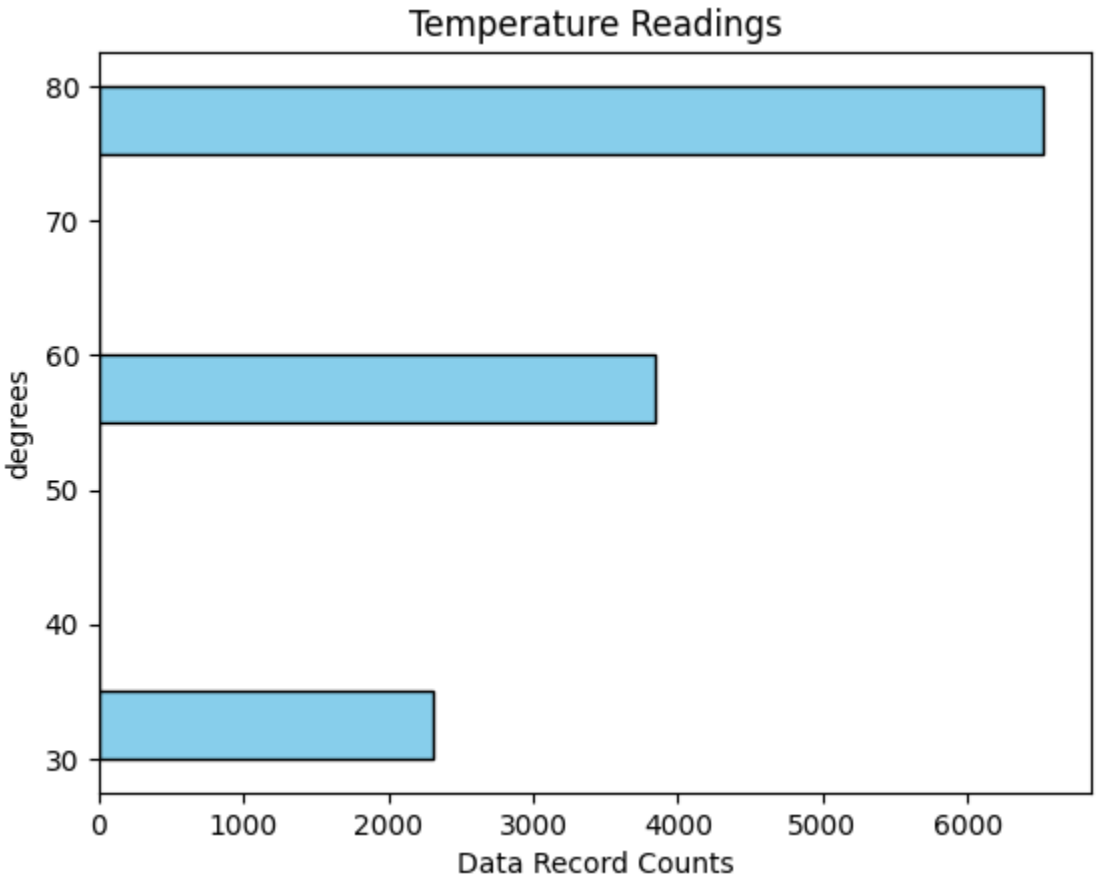
```
*************************************************************************
* Q: What proportion of the total observations chose to accept the coupon? *
*************************************************************************
* A: count the total of coupon accepted and divide by total observations.  *
*    Coupon Acceptance Rate: 56.93%                                       *
*************************************************************************
```

In [46]:
```
# 5. Use a bar plot to visualize the coupon column.
coupon_bar = df_data.groupby('coupon')['coupon'].count()
coupon_bar.plot(kind='bar')
plt.xticks(rotation=45)
plt.ylabel('Issued')
plt.title( 'Coupon Issued')
```

Out[46]: Text(0.5, 1.0, 'Coupon Issued')



In [47]:
```
# 6. Use a histogram to visualize the temperature column.
plt.hist(df['temperature'], bins=10, color='skyblue', edgecolor='black', ori
plt.xlabel('Data Record Counts')
plt.ylabel('degrees')
plt.title( 'Temperature Readings')
plt.show()
```



In [48]:
```
# Investigating the Bar Coupons
# 1. Create a new DataFrame that contains just the bar coupons.
df_bar = df_data[['Bar']]
print(f'************************************************')
print(f'*  Created a new DataFrame containing just the  *')
print(f'*  bar coupons.                                  *')
print(f'*  Bar dataFrame Type: {type(df_bar)}           *')
print(f'************************************************')
```

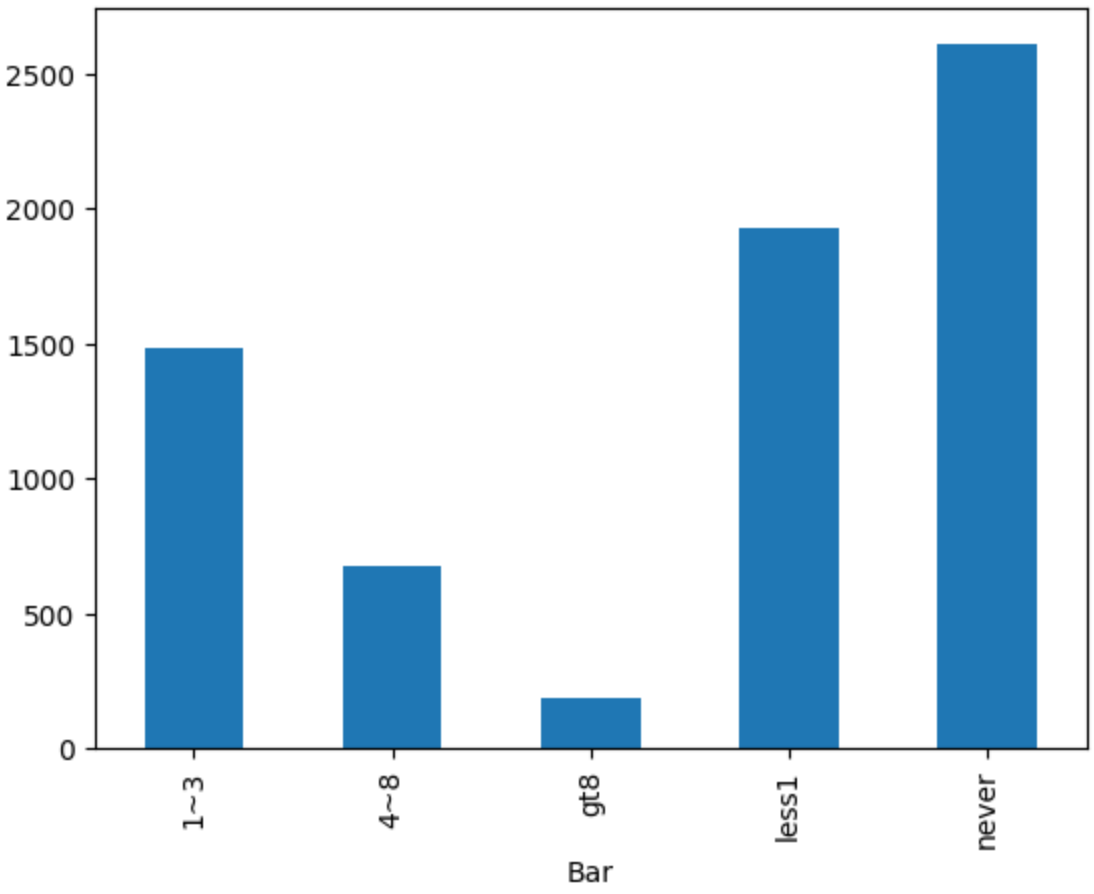
```
************************************************
*  Created a new DataFrame containing just the  *
*  bar coupons.                                  *
*  Bar dataFrame Type: <class 'pandas.core.frame.DataFrame'>           *
************************************************
```

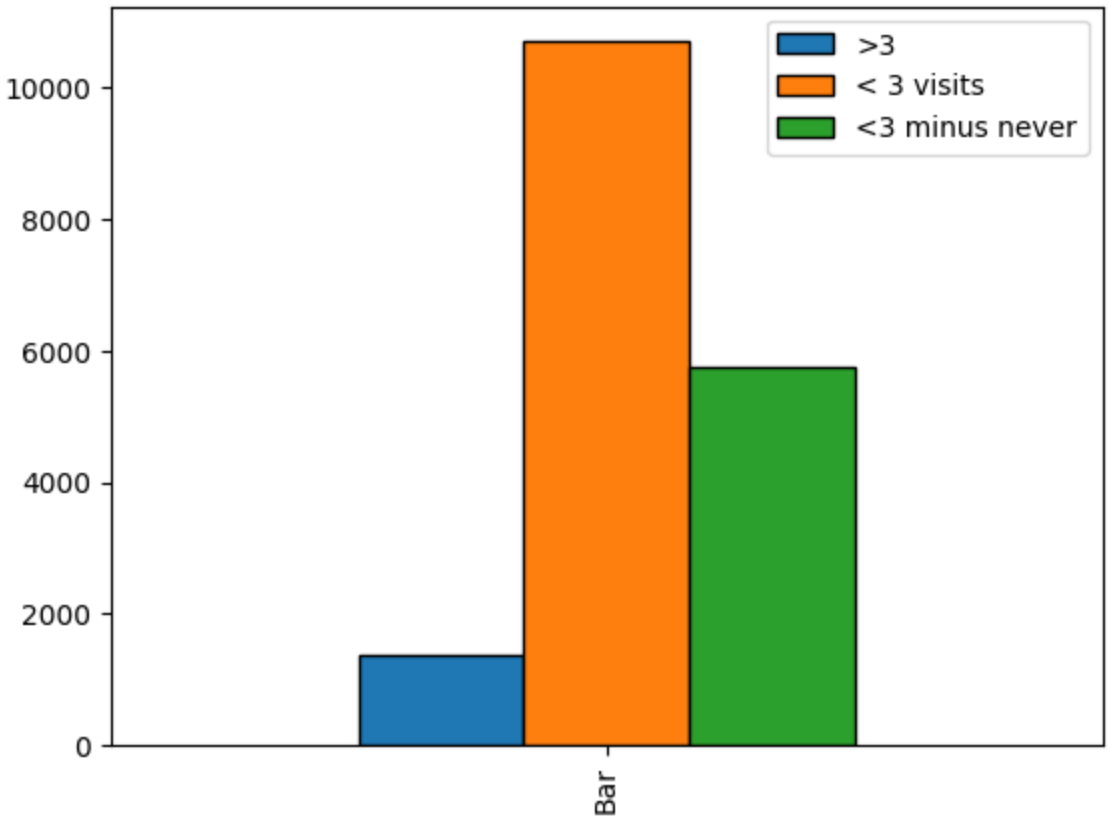
In [49]:
```
# 2. What proportion of bar coupons were accepted?
df_bar_acc = df_data[['Bar','Y']]
df_bar_never = df_bar_acc[ df_bar_acc['Y'] == 0 ]

df_bar_acceptance = (df_bar_acc.count() - df_bar_never.count() ) / df_bar_ac

print(f'************************************************************')
```

```
print(f'* Q: What proportion of bar coupons were accepted?         *')
print(f'* A: {df_bar_acceptance.iloc[0]:.2f}%
print(f'*************************************************************')
```

```
*************************************************************
* Q: What proportion of bar coupons were accepted?         *
* A: 56.93%                                                *
*************************************************************
```

In [50]:
```
# 3. Compare the acceptance rate between those who went to a bar 3 or fewer
df_bar_times = df_bar_acc.query('Y == 1').groupby('Bar')['Bar'].count()

df_bar_times.plot(kind='bar')
plt.show()

df_bar_gt_3  = df_bar[ (df_bar['Bar'] == 'gt8') | (df_bar['Bar'] == '4~8') ]
df_bar_lt_3  = df_bar[ (df_bar['Bar'] != 'gt8') & (df_bar['Bar'] != '4~8') ]
df_bar_lt_3_ = df_bar[ (df_bar['Bar'] != 'gt8') & (df_bar['Bar'] != '4~8') &
df_bar_total = df_bar['Bar'].count()

bar_acceptance_rate = pd.DataFrame({ '>3' : df_bar_gt_3, '< 3 visits' : df_b
bar_acceptance_rate.plot( kind='bar', edgecolor='black')
plt.show()

print(f'          Bar Coupon Visit Result Rates')
print(f'=======================================================')
print(f'> 3 visits     :  { 100* (df_bar_gt_3/(df_bar_total)) }')
print(f'<=3 visits     :  { 100* (df_bar_lt_3/(df_bar_total)) }')
print(f'<=3 minus never:  { 100* (df_bar_lt_3_/(df_bar_total))}')
```





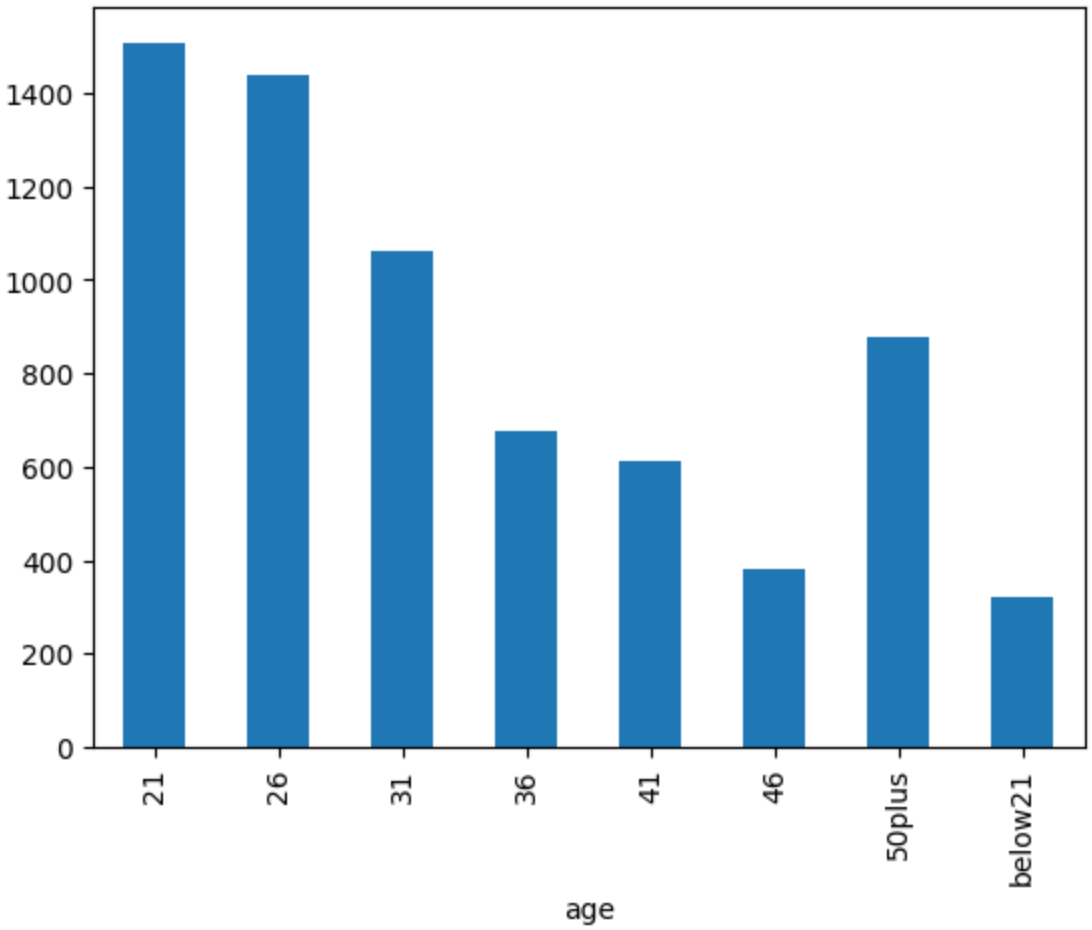

```
          Bar Coupon Visit Result Rates
=======================================================
> 3 visits     :  Bar    11.433066
dtype: float64
<=3 visits     :  Bar    88.566934
dtype: float64
<=3 minus never:  Bar    47.719182
dtype: float64
```

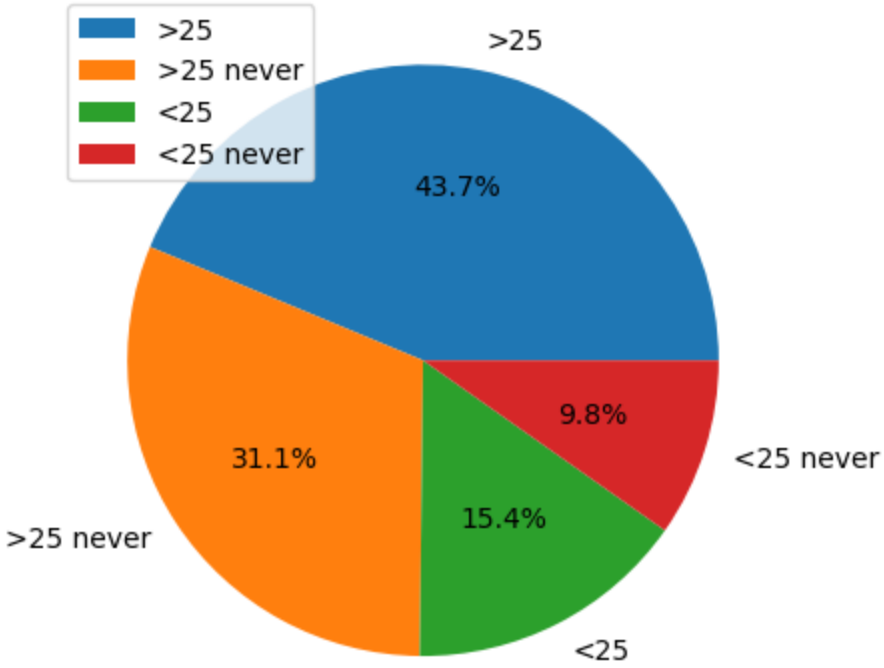
In [51]:
```
# 4. Compare the acceptance rate between drivers who go to a bar more than o
df_age = df_data.query('Y == 1').groupby('age')['age'].value_counts()
# print(df_age.head())
df_age.plot(kind='bar')

df_data_age = df_data.copy()
df_data_age['age'] = df_data_age['age'].replace('50plus' ,'50')
df_data_age['age'] = df_data_age['age'].replace('below21','20')
df_data_age['age'] = df_data_age['age'].astype('int64')

# print(df_data_age.info())
count_age25_over   = df_data_age[ (df_data_age['age'] >= 25) &(df_data_age['
count_age25_over_  = df_data_age[ (df_data_age['age'] >= 25) &(df_data_age['
count_age25_under  = df_data_age[ (df_data_age['age'] <  25) &(df_data_age['
count_age25_under_ = df_data_age[ (df_data_age['age'] <  25) &(df_data_age['
count_all         = df_data_age[ (df_data_age['Bar'] != 'never') ].count()['

print(f'*****************************************************')
print(f'* Q: Coupon Acceptance Rate based on Age 25:        *')
print(f'* A: >25: {count_age25_over}                                    *'
print(f'*    <25: {count_age25_under}                                     *
print(f'*    All: {count_all}                                  *')
```

```
print(f'*****************************************************')

values = [ count_age25_over, count_age25_over_, count_age25_under, count_age
labels = [ '>25', '>25 never', '<25','<25 never' ]
fig,ax = plt.subplots()
ax.pie( values, labels=labels,autopct=lambda p: f'{p:.1f}%' if p>6.0 else ''
ax.legend()
```

```
*****************************************************
* Q: Coupon Acceptance Rate based on Age 25:        *
* A: >25: 5282                                      *
*    <25: 1863                                      *
*    All: 7145                                      *
*****************************************************
```

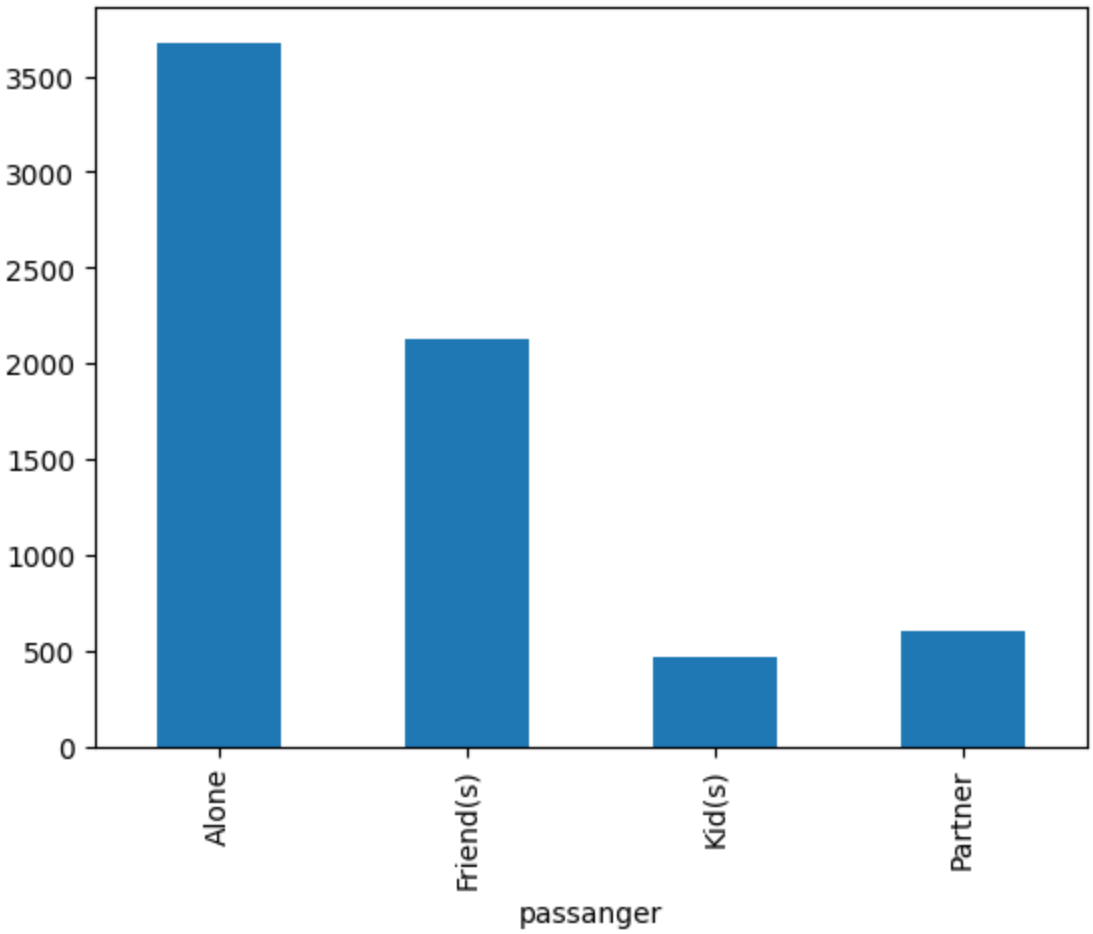
Out[51]: <matplotlib.legend.Legend at 0x7fd516d014f0>

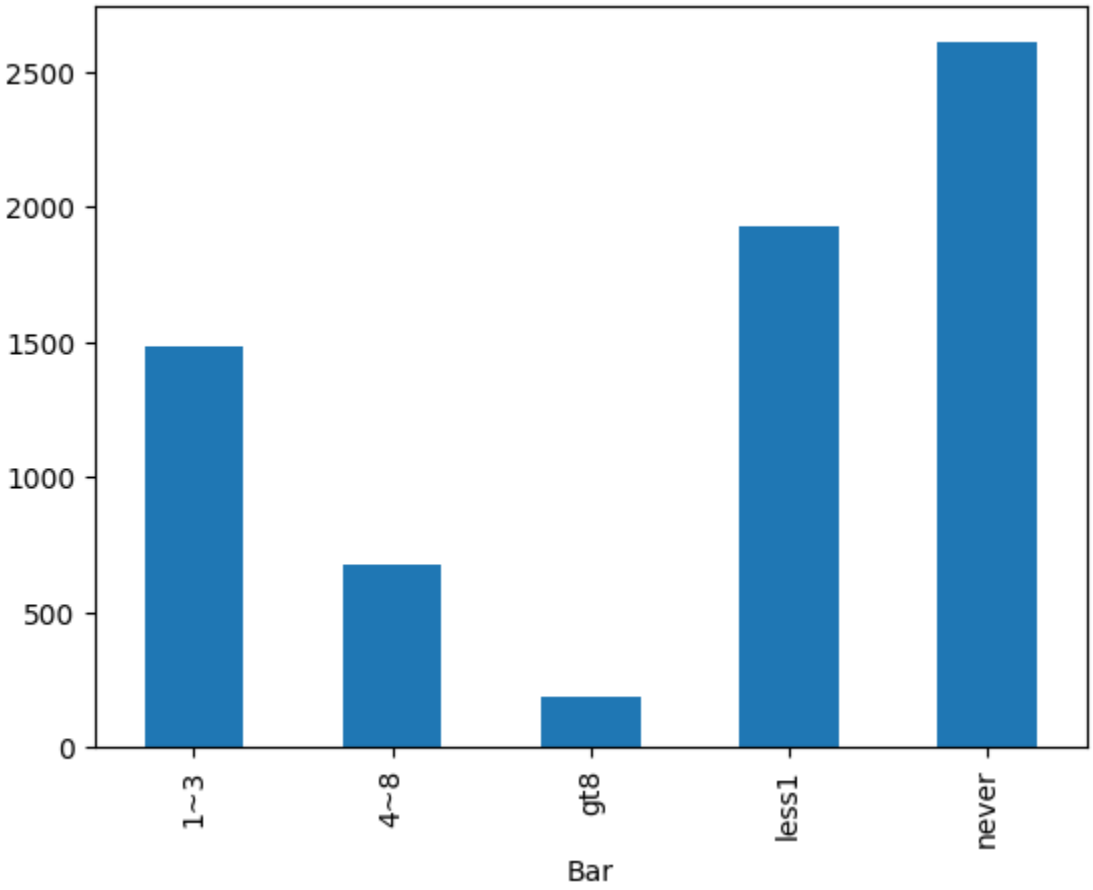


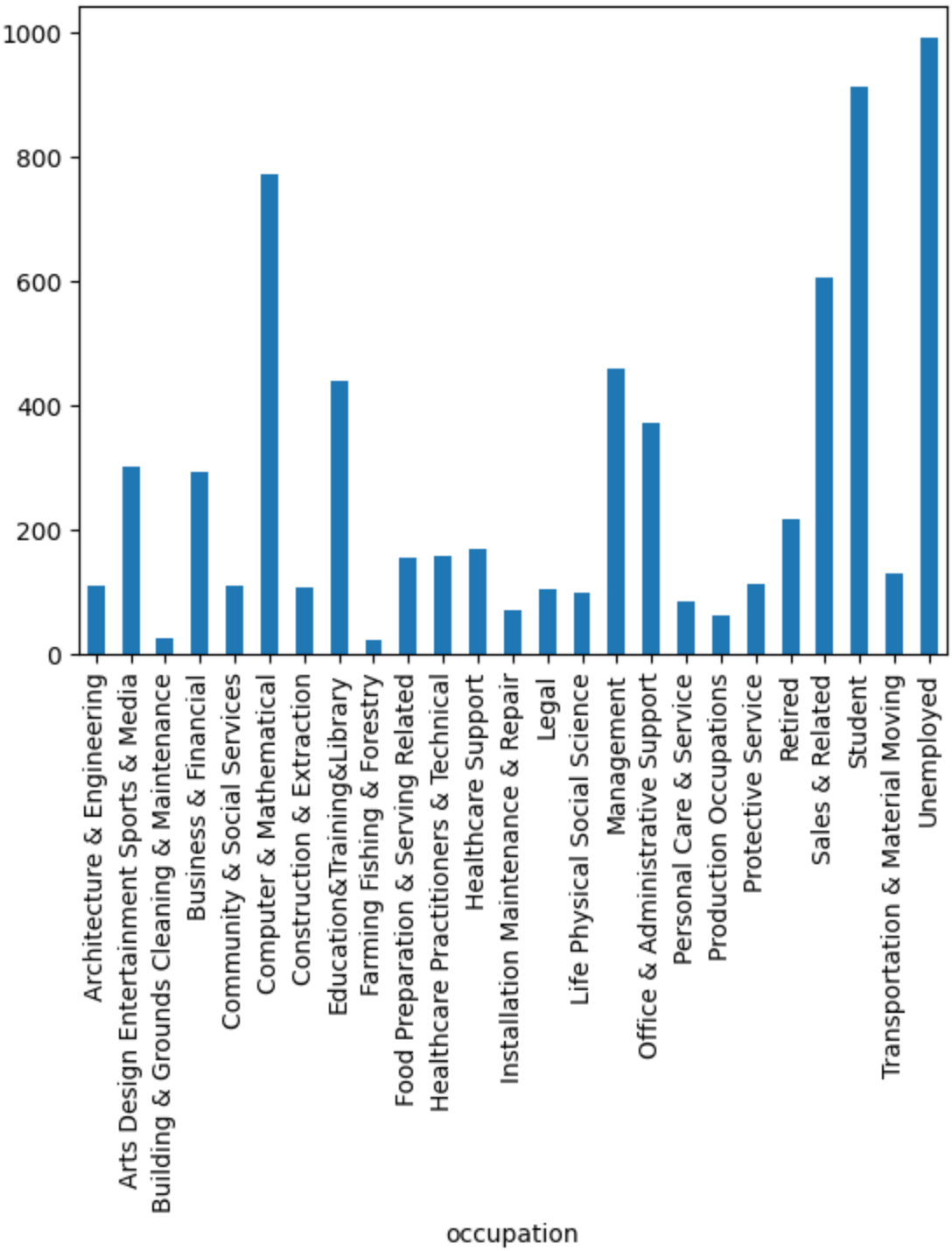


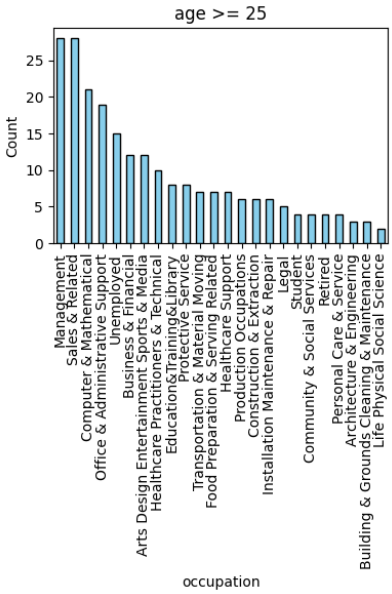
In [52]:
```
# 5. Use the same process to compare the acceptance rate between drivers who
# had occupations other than farming, fishing, or forestry.
occupation_list = ['farming', 'fishing', 'forestry']
passenger_criteria = ['Kid(s)']
bar_visits = ['never','less1']

df_data_age = df_data.copy()
df_data_age['age'] = df_data_age['age'].replace('50plus' ,'50')
df_data_age['age'] = df_data_age['age'].replace('below21','20')
df_data_age['age'] = df_data_age['age'].astype('int64')

df_passenger = df_data_age.query('Y == 1').groupby('passanger')['passanger']
df_passenger.plot(kind='bar')
plt.show()

df_b = df_data_age.query('Y == 1').groupby('Bar')['Bar'].value_counts()
df_b.plot(kind='bar')
plt.show()

df_occupation = df_data_age.query('Y == 1').groupby('occupation')['occupatio
df_occupation.plot(kind='bar')
plt.show()

agg_list = ['age','gender','occupation','passanger','Bar']
df_a = df_data_age.query('Y == 1').query('occupation not in @occupation_list
df_a = df_a.groupby('age')[ agg_list ].value_counts().reset_index()
df_a_over = df_a[ df_a['age'] >= 25 ]
df_a_under= df_a[ df_a['age'] <  25 ]

# Plot bar plots for Occupation, Passenger, and Bar
fig, axs = plt.subplots(1, 3, figsize=(12, 6))
for i, col in enumerate(['occupation', 'passanger', 'Bar']):
```

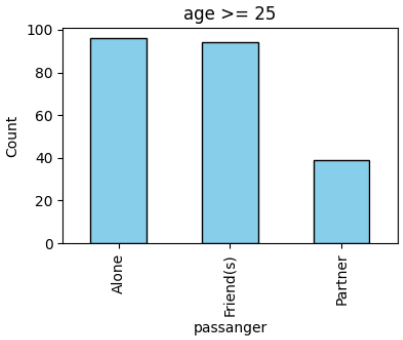
```
    df_a_over[col].value_counts().plot(kind='bar', ax=axs[i], color='skyblue
    axs[i].set_xlabel(col)
    axs[i].set_ylabel('Count')
    axs[i].set_title(f'age >= 25')
plt.tight_layout()
plt.show()

# Plot bar plots for Occupation, Passenger, and Bar
for gender in ['Male','Female']:
    df_g = df_a[ df_a['gender'] == gender ]
    fig, axs = plt.subplots(1, 2, figsize=(12, 6))
    for i, col in enumerate(['occupation', 'passanger']):
        df_over  = df_g[ df_g['age'] >= 25 ]
        if i == 1:
            df_over[col].value_counts().plot(kind='bar', ax=axs[i], edgecolo
        else:
            df_over[col].value_counts().plot(kind='pie', ax=axs[i], autopct=
        axs[i].set_ylabel('Count')
        axs[i].set_title(f'{gender}, age >= 25')
    plt.tight_layout()
    plt.show()

for gender in ['Male','Female']:
    df_g = df_a[ df_a['gender'] == gender ]
    fig, axs = plt.subplots(1, 2, figsize=(12, 6))
    for i, col in enumerate(['occupation', 'passanger']):
        df_under = df_g[ df_g['age'] < 25  ]
        if i == 1:
            df_under[col].value_counts().plot(kind='bar', ax=axs[i], edgecol
        else:
            df_under[col].value_counts().plot(kind='pie', ax=axs[i], autopct
        axs[i].set_xlabel(col)
        axs[i].set_ylabel('Count')
        axs[i].set_title(f'{gender}, age < 25')
    plt.tight_layout()
    plt.show()
```

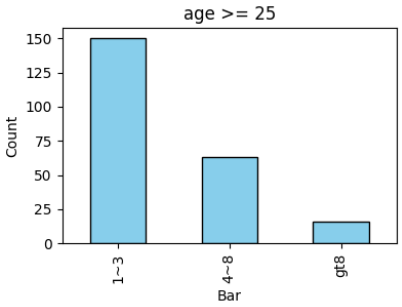

3500
3000
2500
2000
1500
1000
500
0
Alone
Friend(s)
Kid(s)
Partner
passanger

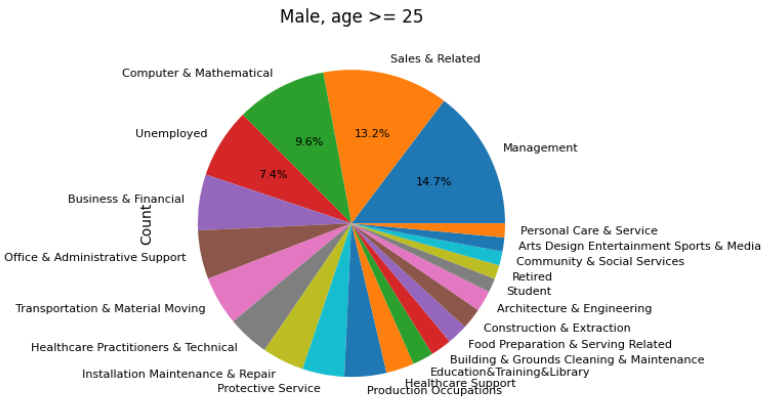

2500
2000
1500
1000
500
0
1~3
4~8
gt8
less1
never
Bar

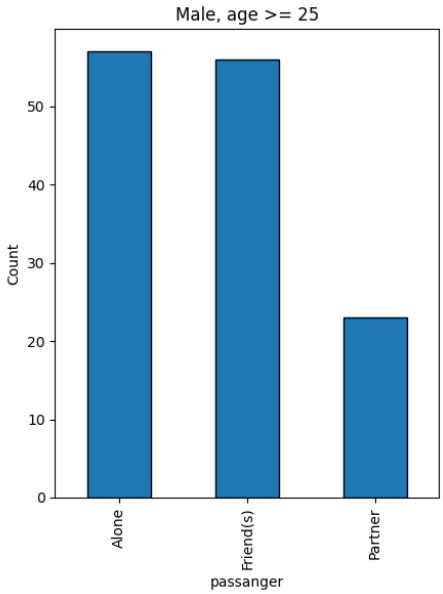

1000
800
600
400
200
0
Architecture & Engineering
Arts Design Entertainment Sports & Media
Building & Grounds Cleaning & Maintenance
Business & Financial
Community & Social Services
Computer & Mathematical
Construction & Extraction
Education&Training&Library
Farming Fishing & Forestry
Food Preparation & Serving Related
Healthcare Practitioners & Technical
Healthcare Support
Installation Maintenance & Repair
Legal
Life Physical Social Science
Management
Office & Administrative Support
Personal Care & Service
Production Occupations
Protective Service
Retired
Sales & Related
Student
Transportation & Material Moving
Unemployed
occupation

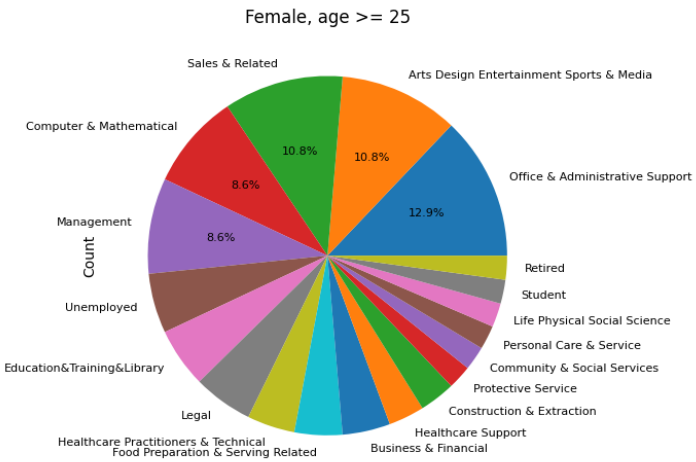

age >= 25
Count
occupation
Management
Sales & Related
Computer & Mathematical
Office & Administrative Support
Unemployed
Business & Financial
Arts Design Entertainment Sports & Media
Healthcare Practitioners & Technical
Education&Training&Library
Protective Service
Transportation & Material Moving
Food Preparation & Serving Related
Healthcare Support
Production Occupations
Construction & Extraction
Installation Maintenance & Repair
Legal
Student
Community & Social Services
Retired
Personal Care & Service
Architecture & Engineering
Building & Grounds Cleaning & Maintenance
Life Physical Social Science

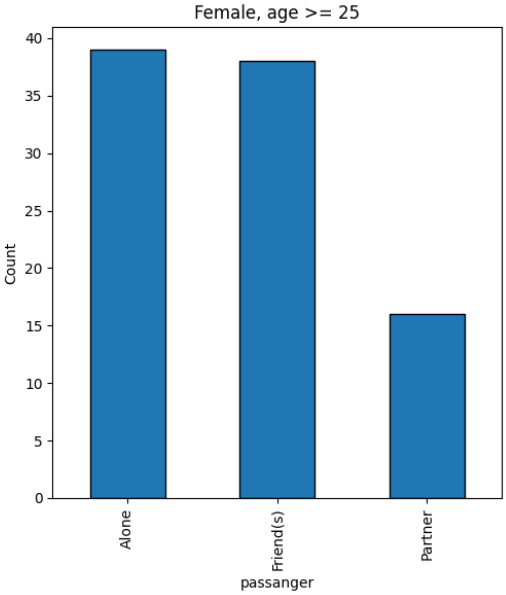

age >= 25
Count
passanger
Alone
Friend(s)
Partner

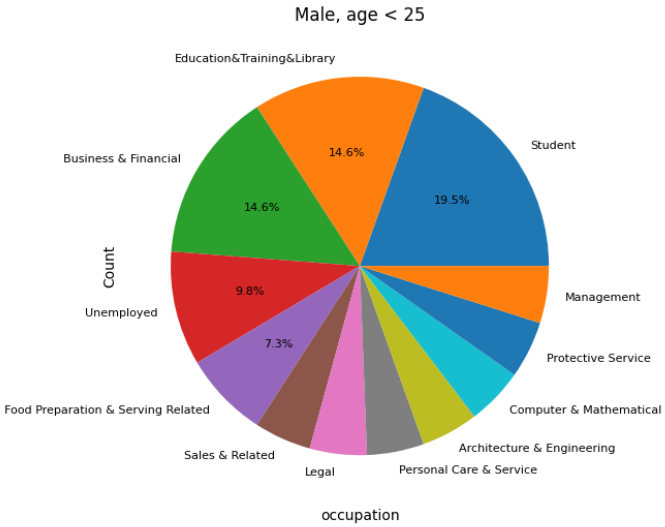

age >= 25
Count
Bar
1~3
4~8
gt8

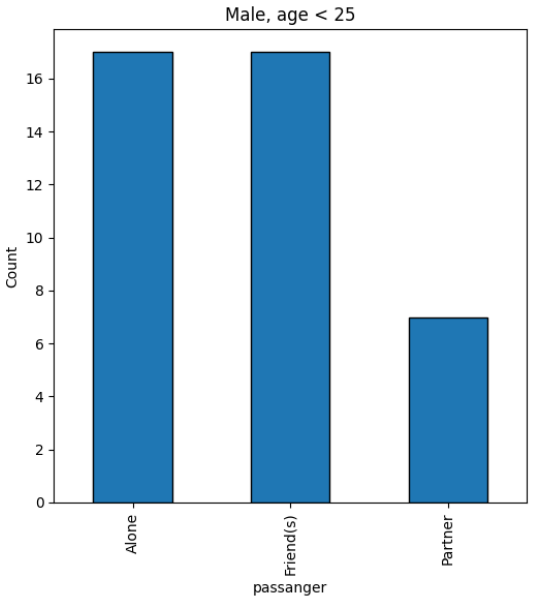

Male, age >= 25
Count
Sales & Related
Computer & Mathematical
Unemployed
Management
Business & Financial
Office & Administrative Support
Transportation & Material Moving
Healthcare Practitioners & Technical
Installation Maintenance & Repair
Protective Service
Healthcare Support
Production Occupations
Education&Training&Library
Building & Grounds Cleaning & Maintenance
Food Preparation & Serving Related
Construction & Extraction
Architecture & Engineering
Student
Retired
Community & Social Services
Arts Design Entertainment Sports & Media
Personal Care & Service
13.2%
9.6%
14.7%
7.4%

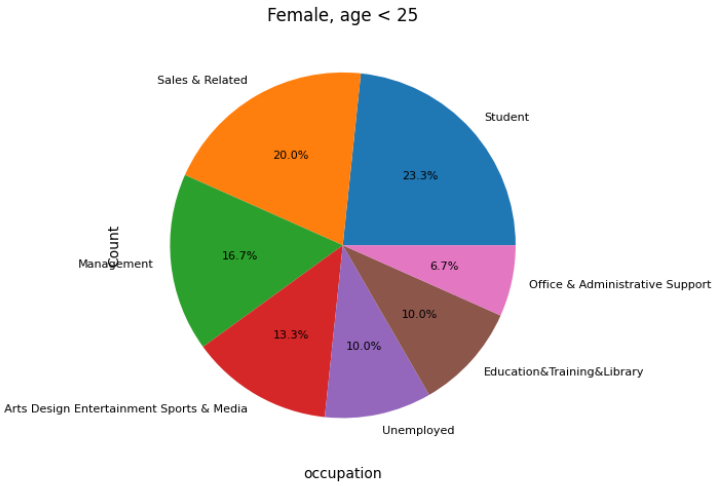

Male, age >= 25
Count
passanger
Alone
Friend(s)
Partner

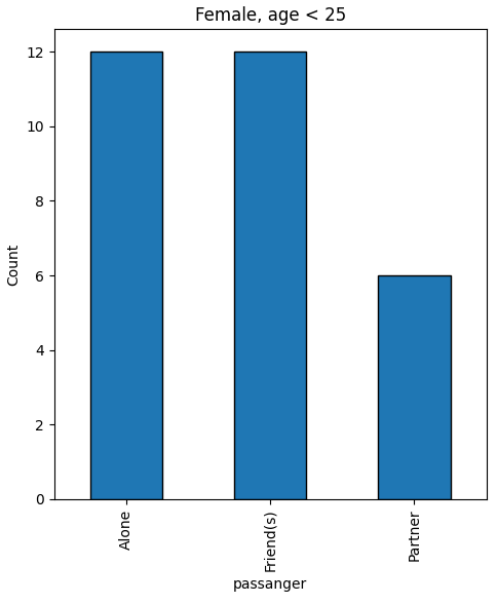

Female, age >= 25
Count
Sales & Related
Arts Design Entertainment Sports & Media
Computer & Mathematical
Office & Administrative Support
Management
Retired
Unemployed
Student
Life Physical Social Science
Education&Training&Library
Personal Care & Service
Community & Social Services
Legal
Protective Service
Construction & Extraction
Healthcare Practitioners & Technical
Healthcare Support
Food Preparation & Serving Related
Business & Financial
10.8%
10.8%
8.6%
12.9%
8.6%


Female, age >= 25
Count
passanger
Alone
Friend(s)
Partner





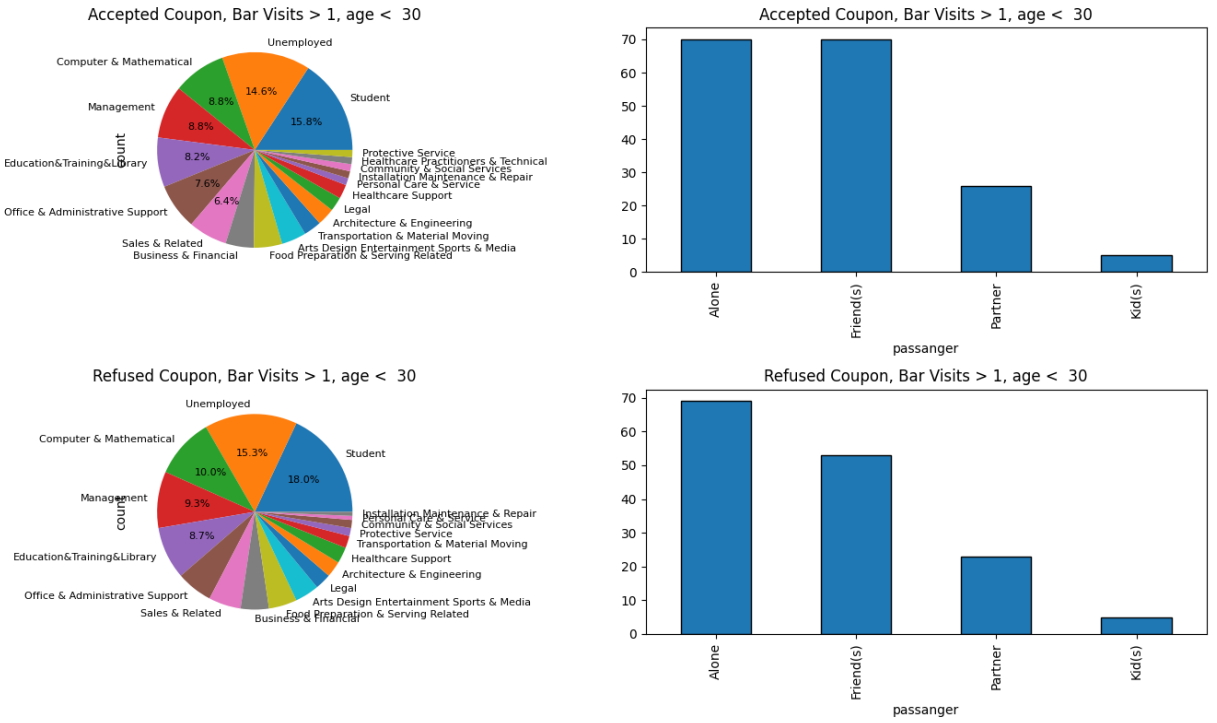




In [53]:
```
# 6. Compare the acceptance rates between those drivers who:
#    go to bars more than once a month, had passengers that were not a kid,
```

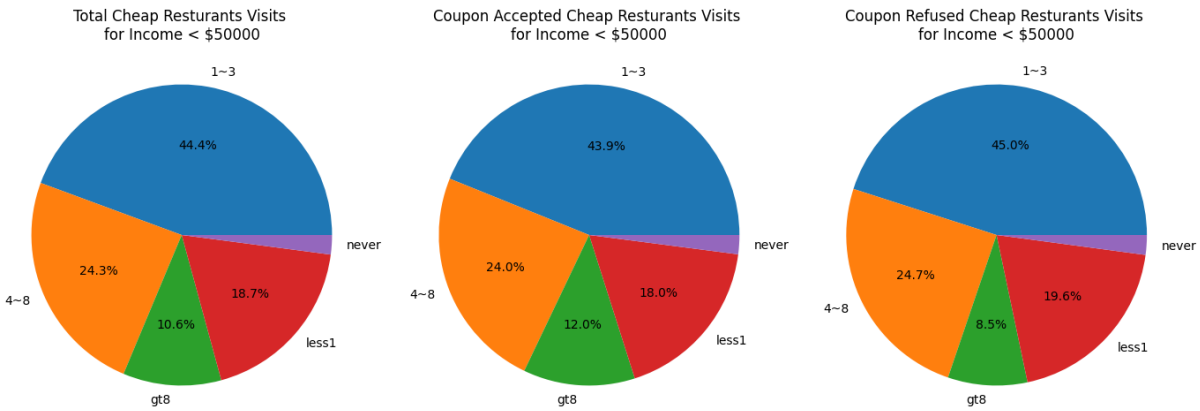
In [54]:
```
#    go to bars more than once a month and are under the age of 30 OR

occupation_list = []
passenger_criteria = []
marital_status = []

bar_visits = ['never','less1']

df_data_age = df_data.copy()
df_data_age['age'] = df_data_age['age'].replace('50plus' ,'50')
df_data_age['age'] = df_data_age['age'].replace('below21','20')
df_data_age['age'] = df_data_age['age'].astype('int64')

agg_list = ['age','gender','occupation','passanger','Bar','maritalStatus']
df_a1 = df_data_age.query('Bar not in @bar_visits')
df_a = df_a1[ df_a1['Y'] == 1 ]
```

```
df_n = df_a1[ df_a1['Y'] == 0 ]

df_a = df_a.groupby('age')[ agg_list ].value_counts().reset_index()
df_a_under= df_a[ df_a['age'] <  30 ]

df_n = df_n.groupby('age')[ agg_list ].value_counts().reset_index()
df_n_under= df_n[ df_n['age'] <  30 ]

# Plot bar plots for Occupation, Passenger, and Bar who accepted coupon offe
#
fig, axs = plt.subplots(2, 2, figsize=(14, 8))
for j, df_ in enumerate( [df_a_under, df_n_under] ):
    for i, col in enumerate(['occupation', 'passanger']):
        if i == 1:
            df_[col].value_counts().plot(kind='bar', ax=axs[j,i], edgecolor=
        else:
            df_[col].value_counts().plot(kind='pie', ax=axs[j,i], autopct=la
        if df_.equals(df_a_under):
            axs[j,i].set_title(f'Accepted Coupon, Bar Visits > 1, age <  30'
        else:
            axs[j,i].set_title(f'Refused Coupon, Bar Visits > 1, age <  30')
plt.tight_layout()
plt.show()
```



In [55]:
```
#    go to cheap restaurants more than 4 times a month and income is less th
df_data_age = df_data.copy()
df_data_age['age'] = df_data_age['age'].replace('50plus' ,'50')
df_data_age['age'] = df_data_age['age'].replace('below21','20')
df_data_age['age'] = df_data_age['age'].astype('int64')

rating_dim = ['gender', 'age', 'maritalStatus', 'has_children',
       'education', 'occupation', 'income', 'Bar', 'CoffeeHouse', 'CarryAway

group_criteria = ['RestaurantLessThan20']
```

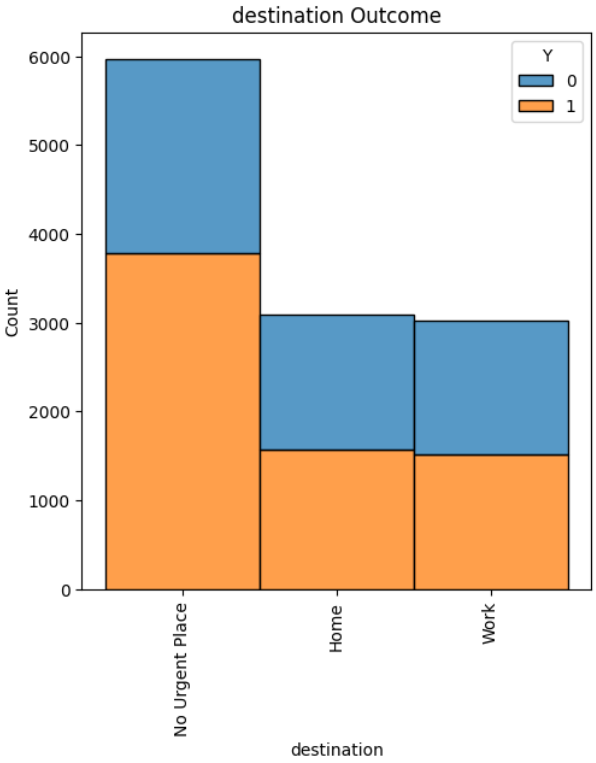
```
income_criteria = ['Less than $12500','$12500 - $24999','$25000 - $37499','$

df_q = df_data_age.query('income in @income_criteria').groupby(group_criteri

df_qa = df_data_age.query('income in @income_criteria').query('Y == 1').grou

df_na = df_data_age.query('income in @income_criteria').query('Y == 0').grou

fig, axs = plt.subplots(1, 3, figsize=(14, 8))
axs[0].pie(x=df_q['income'],autopct=lambda p: f'{p:.1f}%' if p>6.0 else '',l
axs[0].set_title('Total Cheap Resturants Visits \nfor Income < $50000')
axs[1].pie(x=df_qa['income'],autopct=lambda p: f'{p:.1f}%' if p>6.0 else '',
axs[1].set_title('Coupon Accepted Cheap Resturants Visits \nfor Income < $50
axs[2].pie(x=df_na['income'],autopct=lambda p: f'{p:.1f}%' if p>6.0 else '',
axs[2].set_title('Coupon Refused Cheap Resturants Visits \nfor Income < $500

plt.tight_layout()
plt.show()

#df_na_resetindex = df_na.reset_index()
#
#print(df_na_resetindex.info())
#print(df_na_resetindex['occupation'].head())
```

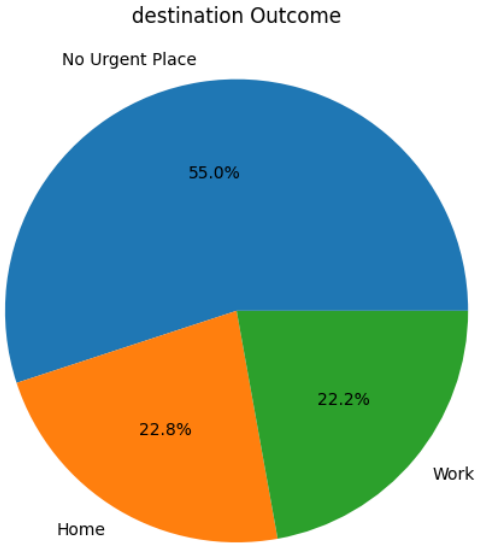


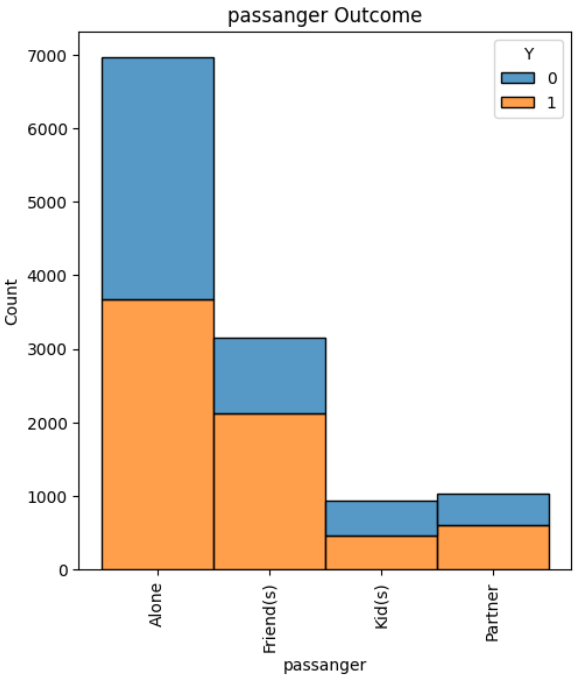
In [56]:
```
# 7. Based on these observations, what do you hypothesize about drivers who
```

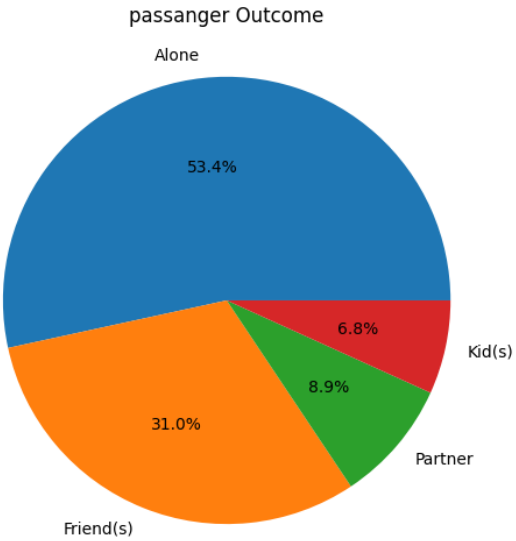
In [57]:
```
# Independent Investigation
#
# Cleanup Data
#   Remove 'Car' column
#   dropna()
# Perform subgroup counts()
#   Value Counts for coupons accepted / non-accepted
#
df = pd.read_csv('data/coupons.csv')
df = df.drop( columns=['car'] )

df['age'] = df['age'].replace('50plus' ,'50')
df['age'] = df['age'].replace('below21','20')
df['age'] = df['age'].astype('int64')
df = df.dropna()
```

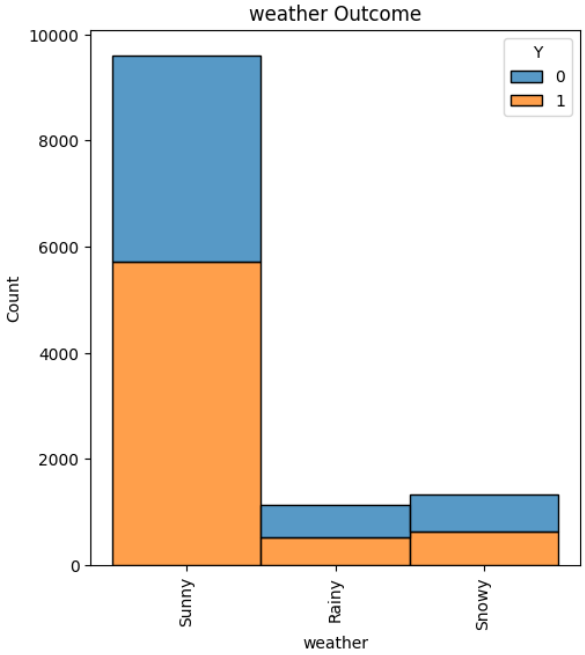
```
# Individual features % with positive outcome
outcome_field = 'Y'
for feature in df.columns:
    if feature != 'Y':
        total_counts = df[ feature ].count()
        percentage_good = df[ df['Y'] ==1 ][feature].value_counts() / total_
        percentage_bad  = 100 - percentage_good
        fig, axs = plt.subplots(1, 2, figsize=(12,6))
        sns.histplot(data=df, x=feature, bins= 100, hue=outcome_field, multi
        axs[0].set_title(f'{feature} Outcome')
        axs[0].set_xlabel(feature)
        axs[0].set_ylabel('Count')
        axs[0].tick_params(axis='x', rotation=90)  # Rotate x-axis labels

        # Plot pie chart
        axs[1].pie(percentage_good, labels=percentage_good.index, radius=0.5
        axs[1].set_title(f'{feature} Outcome')
        axs[1].axis('equal')  # Equal aspect ratio ensures that pie is drawn

        plt.show()

total_counts = len( df['Y'] == 1 )
occ_count = len(df[ df['Y'] == 1 ]['occupation'])
percentage = 100 * (occ_count / total_counts)

outcome_field = 'Y'
total_rcds = len( df )
for feature in df.columns:
    if feature != 'Y':
        percentage_good = df[df['Y']==1][feature].value_counts()
        percentage_impact = 100 * ( percentage_good / total_rcds )
        print(f'{feature}: {percentage_impact}%')


print(f'*********************************************************************')
print(f'* Investigation: Which Feature most effect Coupon Acceptance      *')
print(f'*********************************************************************')
print(f'* This section examines each feature of the dataset, explore its  *')
print(f'* effect on the acceptance of the coupon. A bar plot is using     *')
print(f'* value_count() to show outcome counts for each feature with the  *')
print(f'* pie char next to it to show the % of each.                      *')
print(f'*********************************************************************')
print(f'* Result:  unexpected outcome, as each each feature contributes   *')
print(f'* equally to the coupon acceptance outcome, since I filter for    *')
print(f'*  only the Y = 1 column, and all other columns have non-null data*')
print(f'*  and therefore all contributed to the outcome.  Need to flatten *')
print(f'*  each feature, using value_count() and find which of the single *')
print(f'*  most important value of each feature effect the outcome.       *')
print(f'*********************************************************************')
```

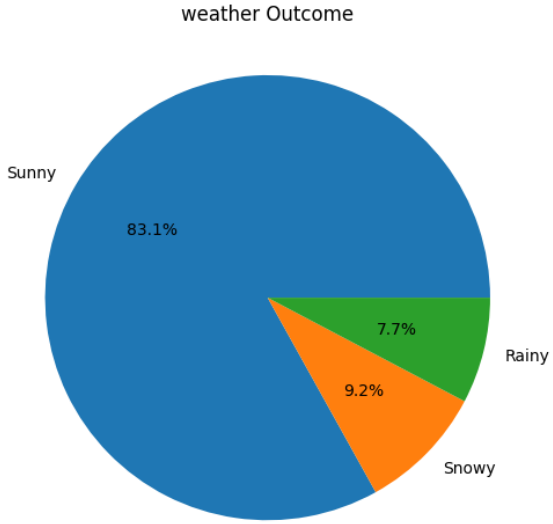

destination Outcome
Y
0
1
Count
6000
5000
4000
3000
2000
1000
0
No Urgent Place
Home
Work
destination

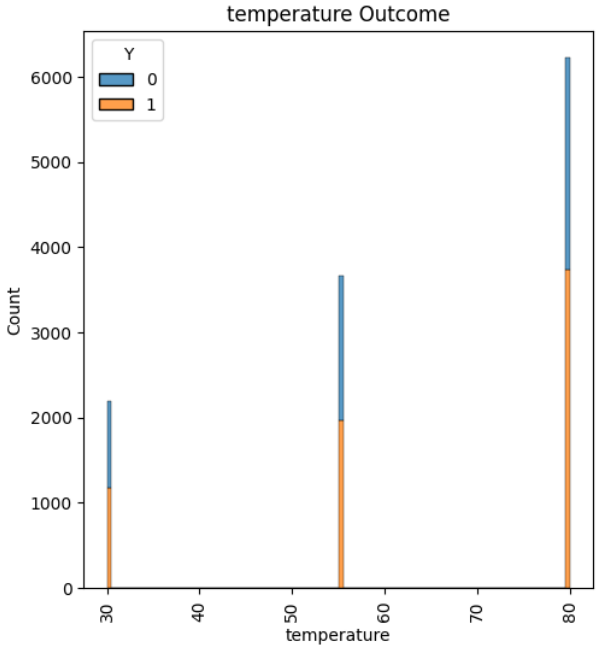

destination Outcome
No Urgent Place
55.0%
22.8%
22.2%
Work
Home

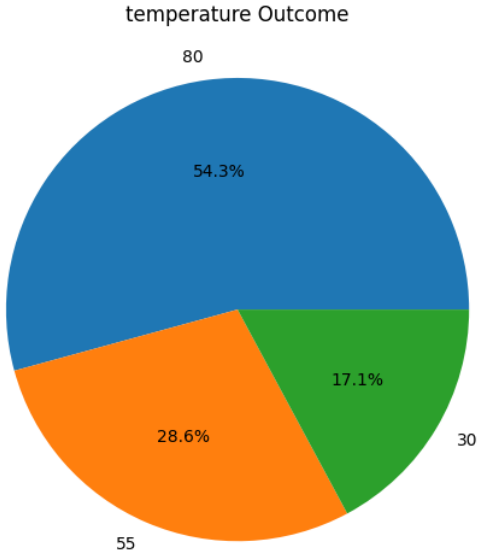

passanger Outcome
Y
0
1
Count
7000
6000
5000
4000
3000
2000
1000
0
Alone
Friend(s)
Kid(s)
Partner
passanger

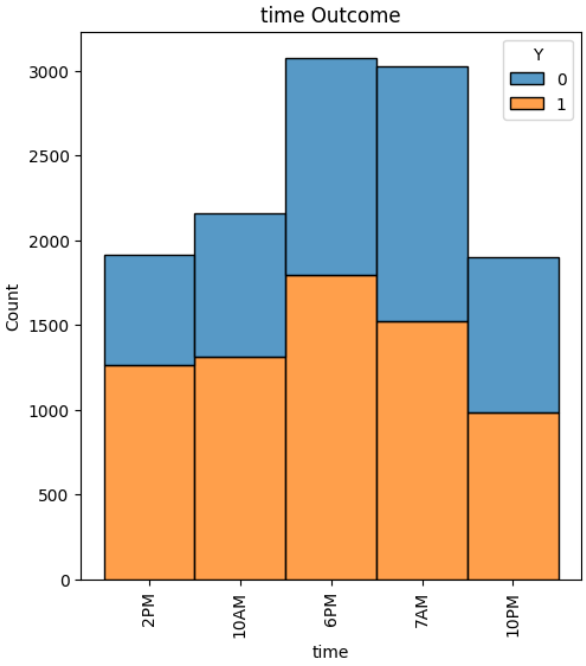

passanger Outcome
Alone
53.4%
6.8%
Kid(s)
8.9%
31.0%
Partner
Friend(s)

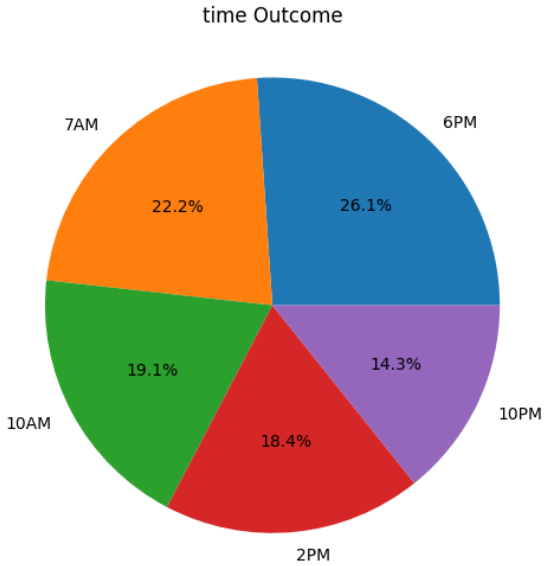

weather Outcome
Y
0
1
Count
10000
8000
6000
4000
2000
0
Sunny
Rainy
Snowy
weather

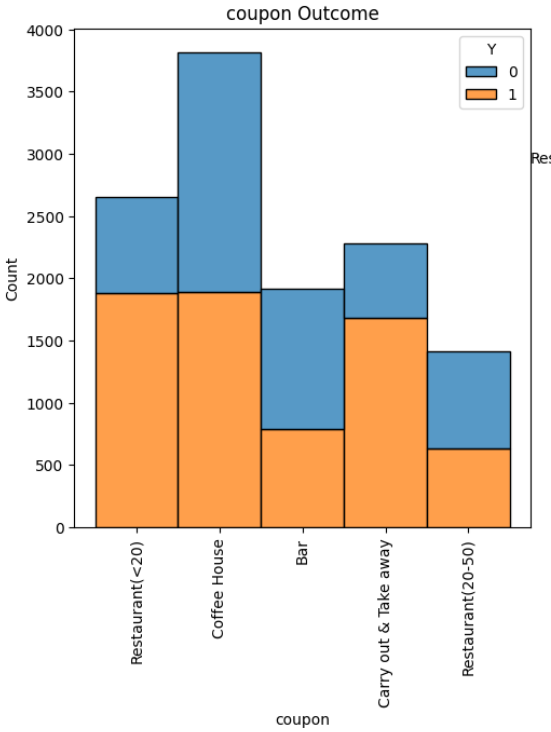

weather Outcome
Sunny
83.1%
7.7%
Rainy
9.2%
Snowy

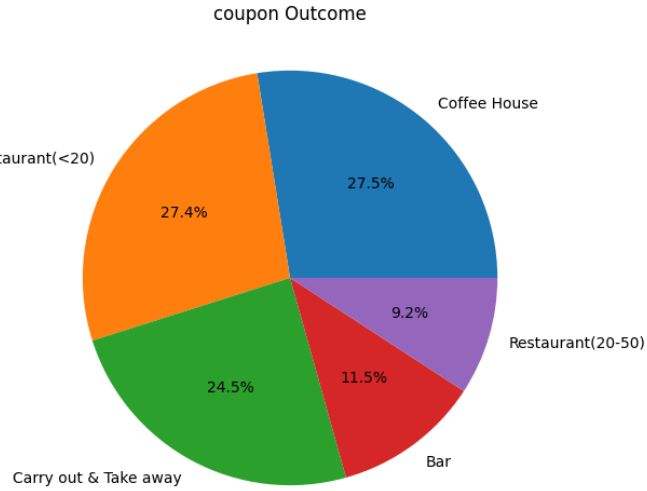

temperature Outcome
Y
0
1
Count
6000
5000
4000
3000
2000
1000
0
30
40
50
60
70
80
temperature

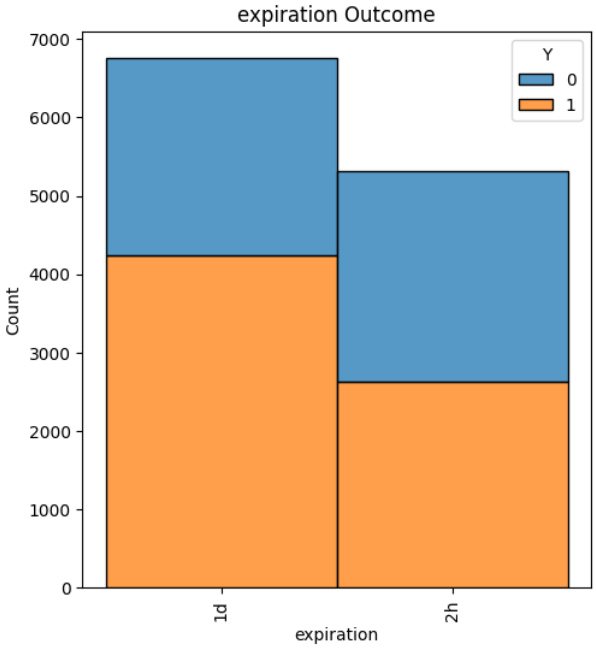

temperature Outcome
80
54.3%
17.1%
30
28.6%
55

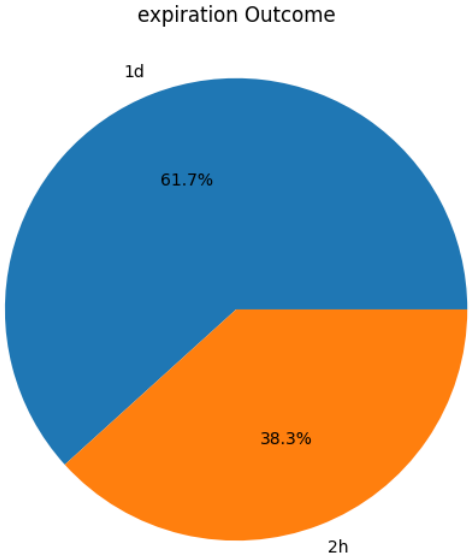

time Outcome
Y
0
1
Count
3000
2500
2000
1500
1000
500
0
2PM
10AM
6PM
7AM
10PM
time

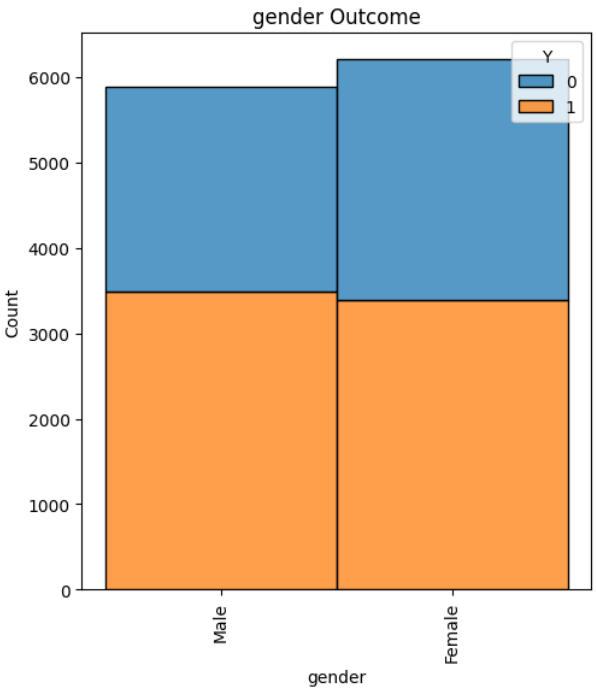

time Outcome
7AM
6PM
22.2%
26.1%
19.1%
14.3%
10AM
10PM
18.4%
2PM

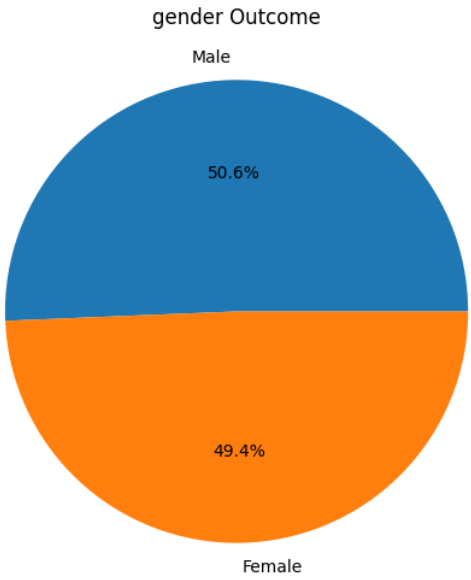

coupon Outcome
Y
0
1
Count
4000
3500
3000
2500
2000
1500
1000
500
0
Restaurant(<20)
Coffee House
Bar
Carry out & Take away
Restaurant(20-50)
coupon

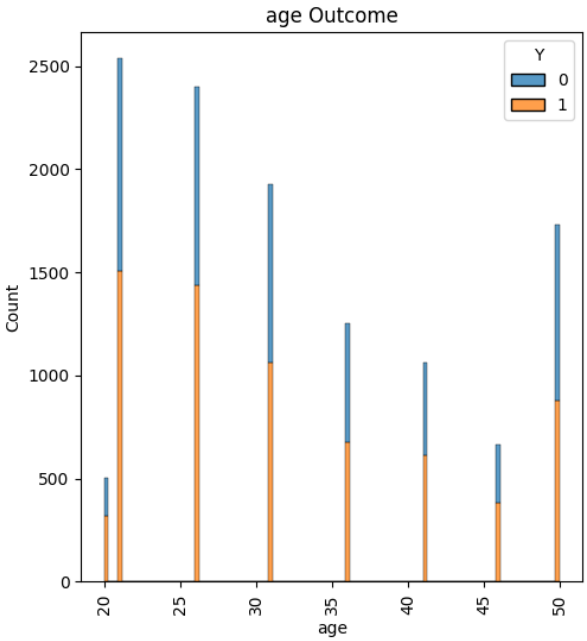

coupon Outcome
Coffee House
Restaurant(<20)
27.5%
27.4%
9.2%
Restaurant(20-50)
24.5%
11.5%
Bar
Carry out & Take away

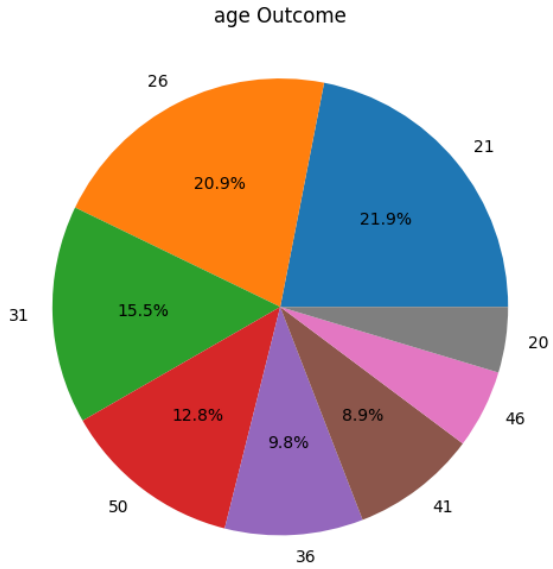

expiration Outcome
Y
0
1
Count
7000
6000
5000
4000
3000
2000
1000
0
1d
2h
expiration

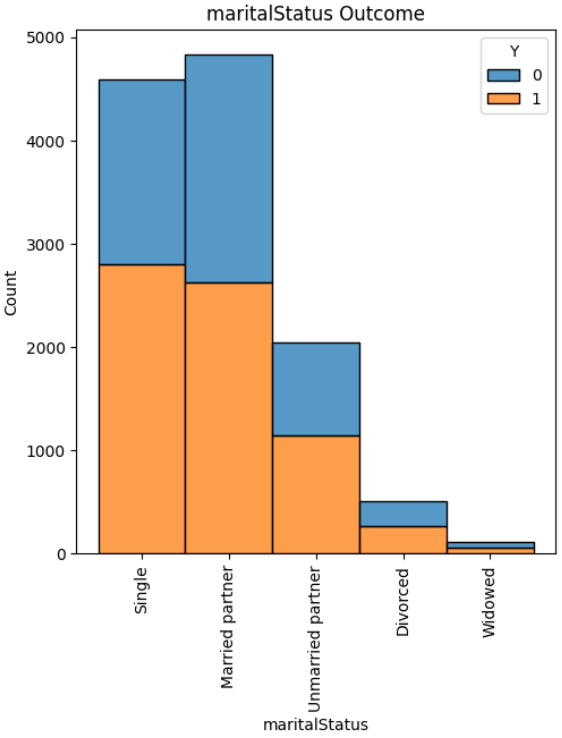

expiration Outcome
1d
61.7%
38.3%
2h

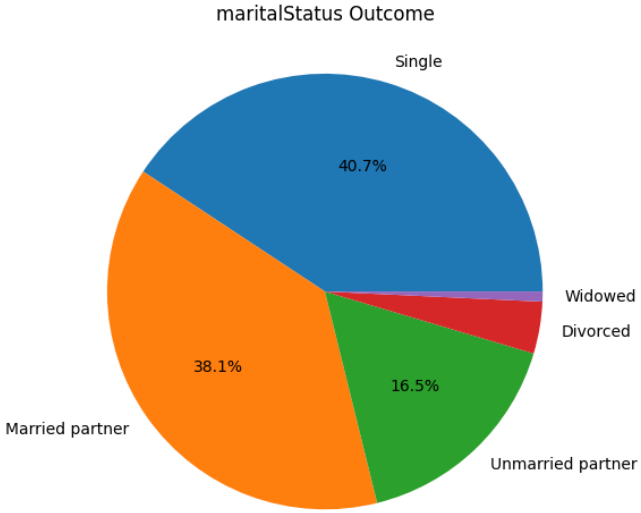

gender Outcome
Y
0
1
Count
6000
5000
4000
3000
2000
1000
0
Male
Female
gender

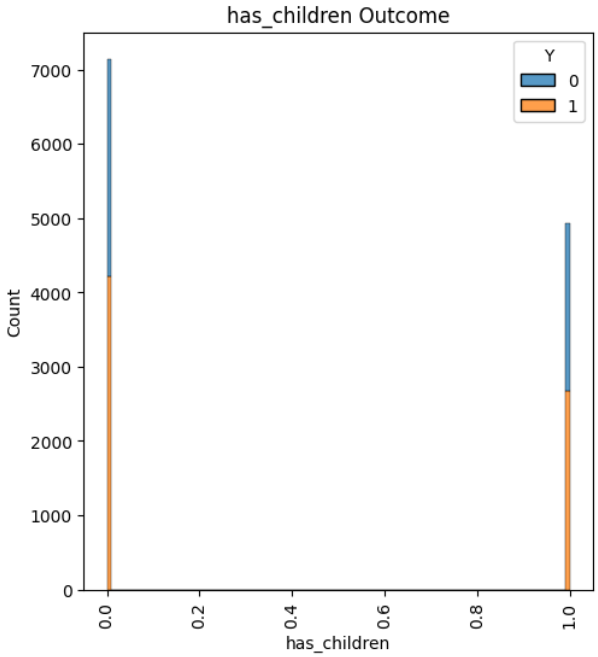

gender Outcome
Male
50.6%
49.4%
Female

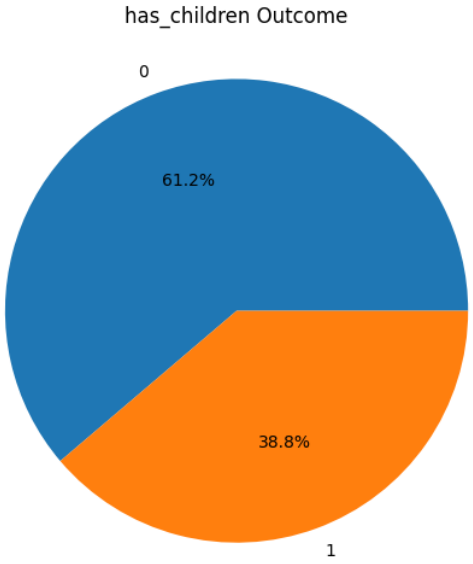

age Outcome
Y
0
1
Count
2500
2000
1500
1000
500
0
20
25
30
35
40
45
50
age

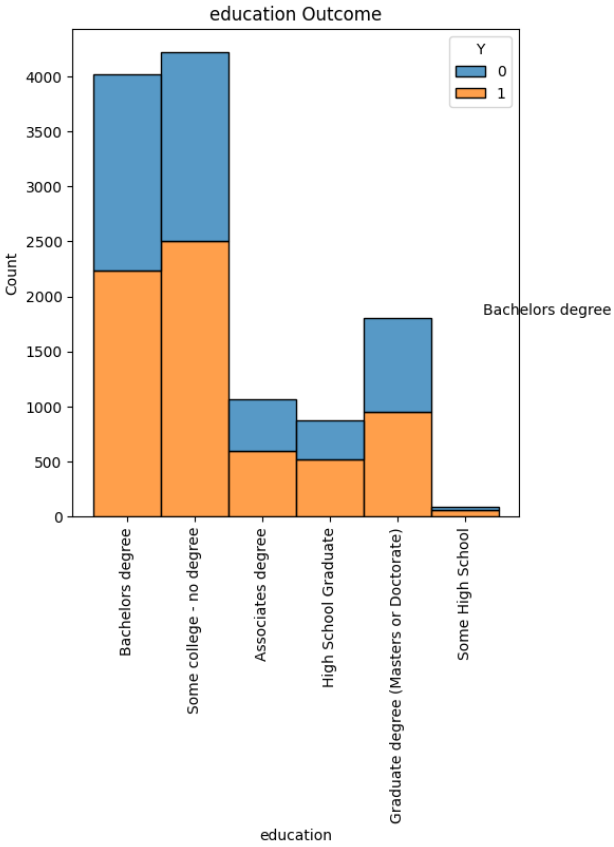

age Outcome
26
21
20.9%
21.9%
31
15.5%
20
46
12.8%
8.9%
9.8%
50
41
36

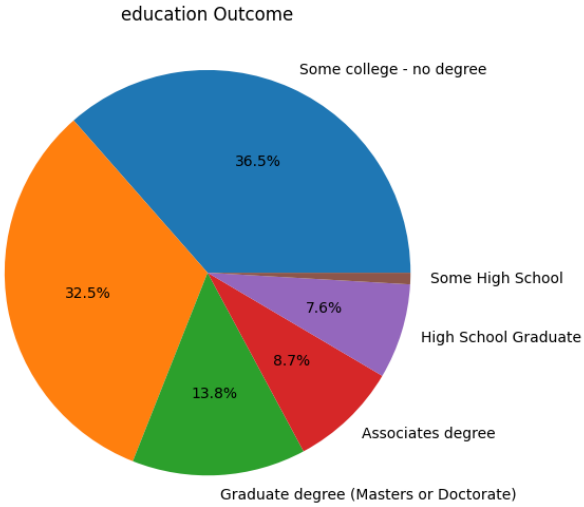

maritalStatus Outcome
Y
0
1
Count
5000
4000
3000
2000
1000
0
Single
Married partner
Unmarried partner
Divorced
Widowed
maritalStatus

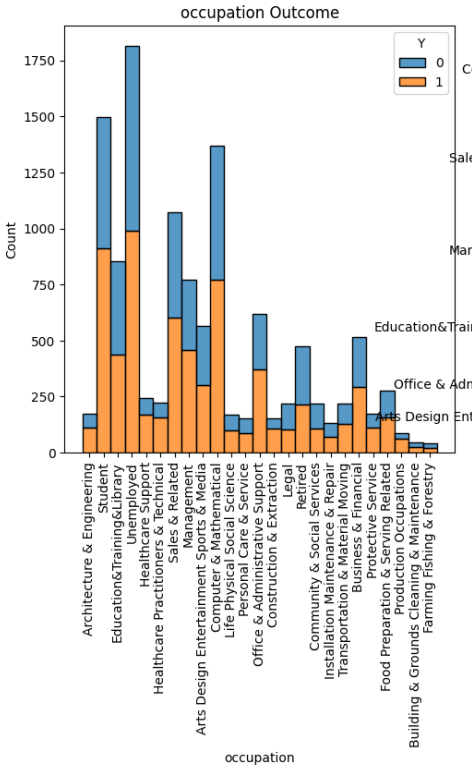

maritalStatus Outcome
Single
40.7%
Widowed
Divorced
38.1%
16.5%
Married partner
Unmarried partner

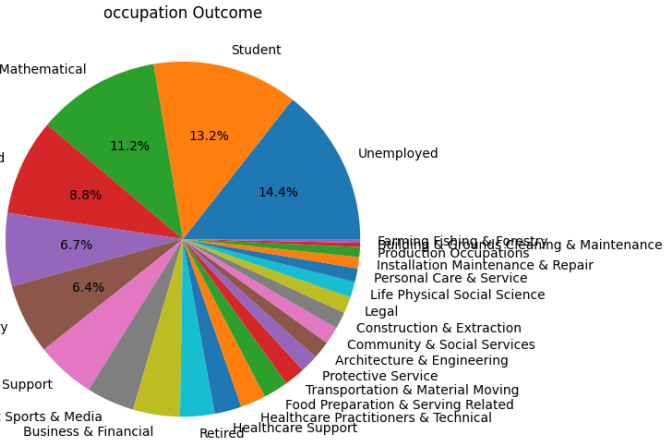

has_children Outcome
Y
0
1
Count
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
has_children

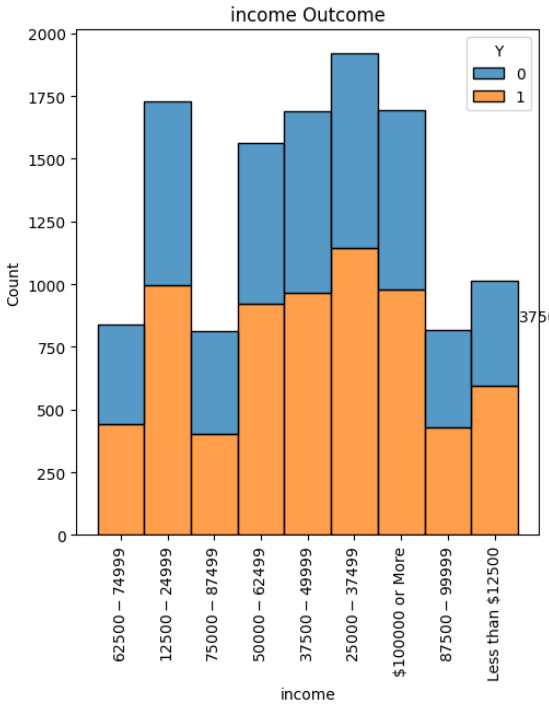

has_children Outcome
0
61.2%
38.8%
1

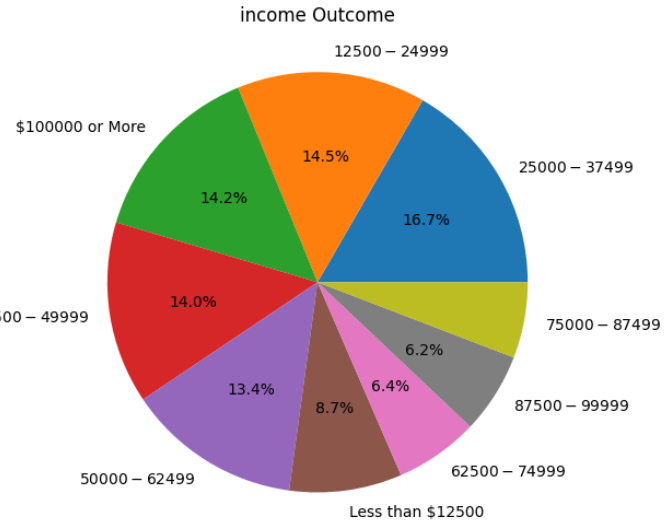

education Outcome
Y
0
1
Count
4000
3500
3000
2500
2000
1500
1000
500
0
Bachelors degree
Some college - no degree
Associates degree
High School Graduate
Graduate degree (Masters or Doctorate)
Some High School
education

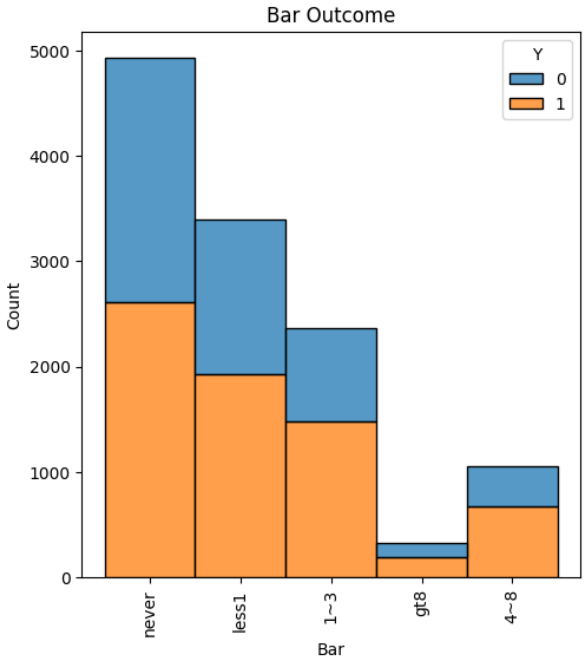

education Outcome
Some college - no degree
36.5%
Some High School
32.5%
7.6%
High School Graduate
8.7%
13.8%
Associates degree
Graduate degree (Masters or Doctorate)

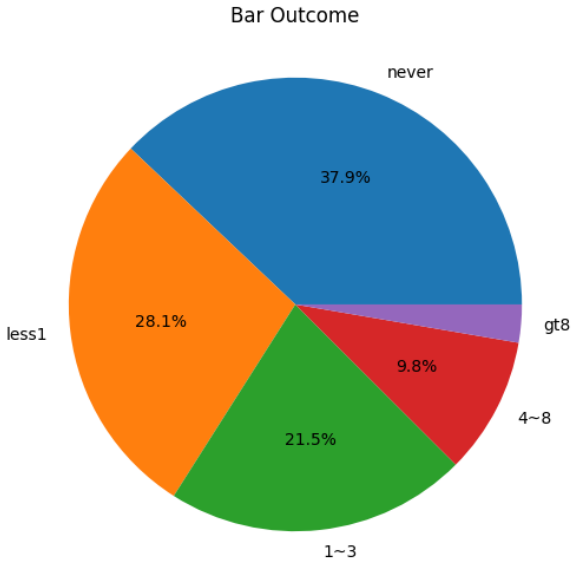

occupation Outcome
Y
0
1
Count
occupation
Architecture & Engineering
Student
Education&Training&Library
Unemployed
Healthcare Support
Healthcare Practitioners & Technical
Sales & Related
Management
Arts Design Entertainment Sports & Media
Computer & Mathematical
Life Physical Social Science
Personal Care & Service
Office & Administrative Support
Construction & Extraction
Legal
Retired
Community & Social Services
Installation Maintenance & Repair
Transportation & Material Moving
Business & Financial
Protective Service
Food Preparation & Serving Related
Production Occupations
Building & Grounds Cleaning & Maintenance
Farming Fishing & Forestry

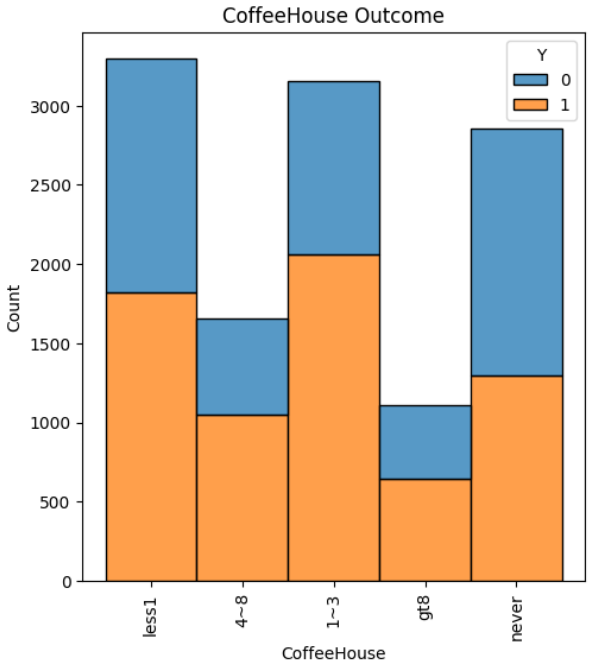

occupation Outcome
Student
Computer & Mathematical
Sales & Related
Management
Education&Training&Library
Office & Administrative Support
Arts Design Entertainment Sports & Media
Business & Financial
Retired
Healthcare Support
Unemployed
Farming Fishing & Forestry
Building & Grounds Cleaning & Maintenance
Production Occupations
Installation Maintenance & Repair
Personal Care & Service
Life Physical Social Science
Legal
Construction & Extraction
Community & Social Services
Architecture & Engineering
Protective Service
Transportation & Material Moving
Food Preparation & Serving Related
Healthcare Practitioners & Technical
11.2%
13.2%
14.4%
8.8%
6.7%
6.4%

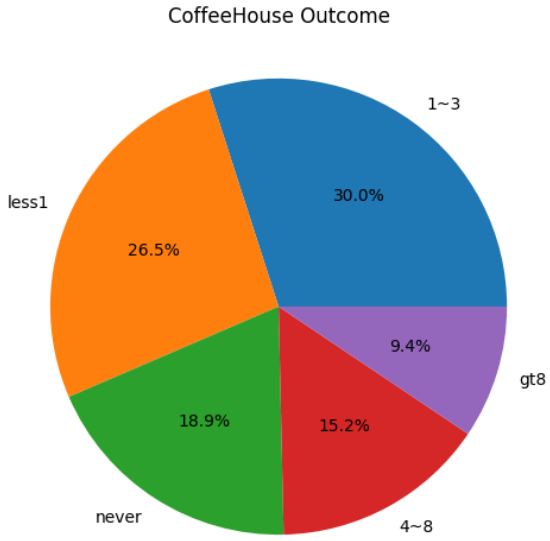

income Outcome
Y
0
1
Count
income
62500 – 74999
12500 – 24999
75000 – 87499
50000 – 62499
37500 – 49999
25000 – 37499
$100000 or More
87500 – 99999
Less than $12500

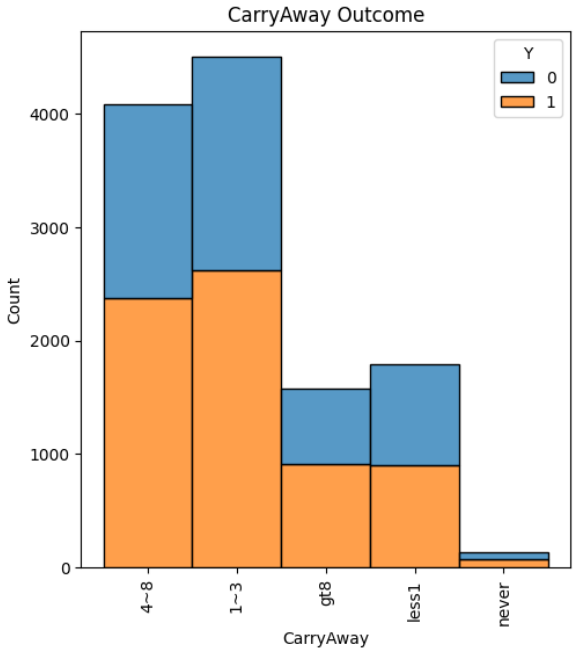

income Outcome
12500 – 24999
$100000 or More
25000 – 37499
37500 – 49999
75000 – 87499
87500 – 99999
50000 – 62499
62500 – 74999
Less than $12500
14.5%
14.2%
16.7%
14.0%
6.2%
13.4%
6.4%
8.7%

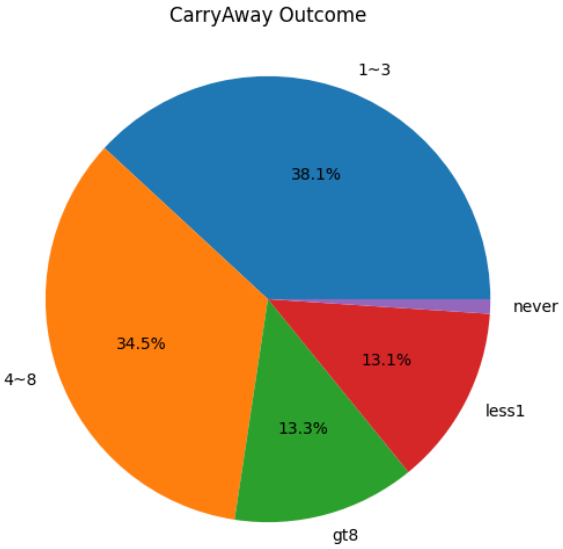

Bar Outcome
Y
0
1
Count
Bar
never
less1
1~3
gt8
4~8
0
1000
2000
3000
4000
5000

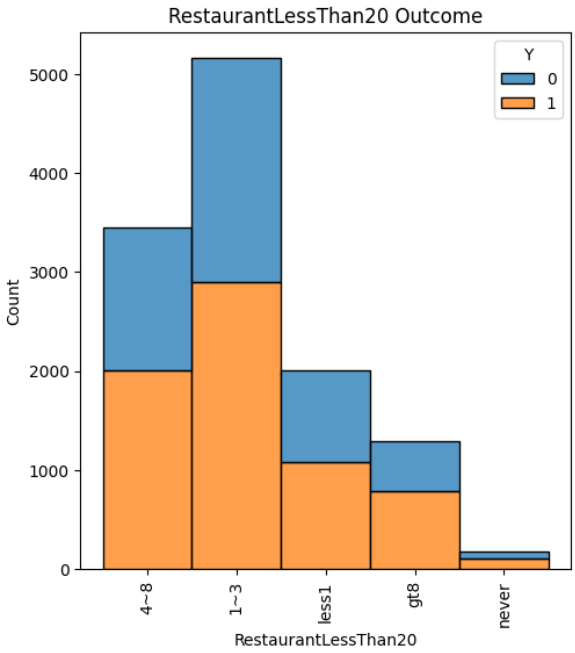

Bar Outcome
never
37.9%
less1
28.1%
gt8
9.8%
4~8
21.5%
1~3

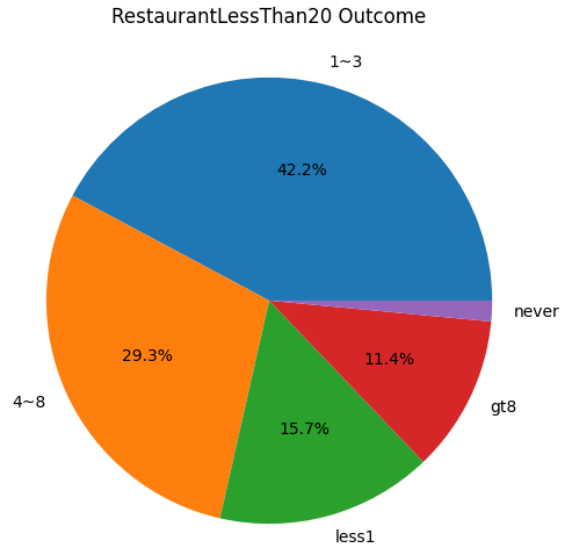

CoffeeHouse Outcome
Y
0
1
Count
CoffeeHouse
less1
4~8
1~3
gt8
never
0
500
1000
1500
2000
2500
3000

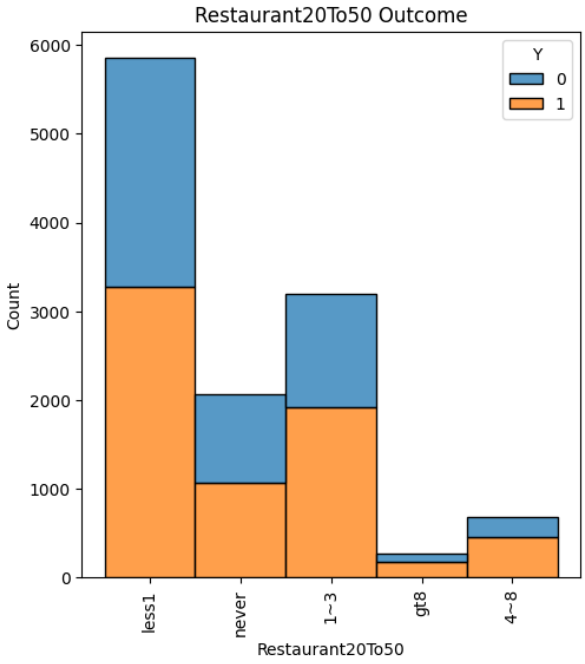

CoffeeHouse Outcome
1~3
30.0%
less1
26.5%
9.4%
gt8
18.9%
15.2%
never
4~8

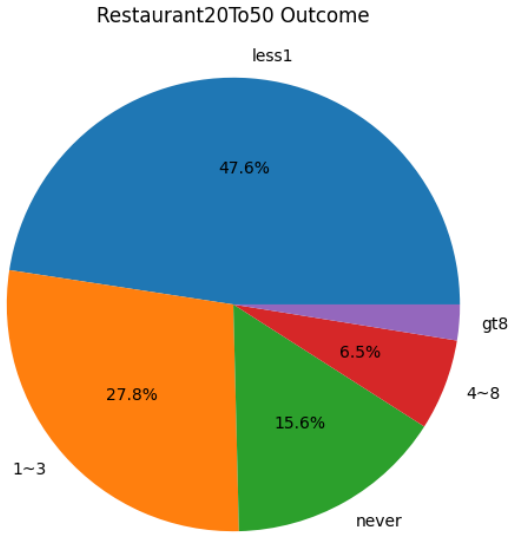

CarryAway Outcome
Y
0
1
Count
4000
3000
2000
1000
0
4~8
1~3
gt8
less1
never
CarryAway

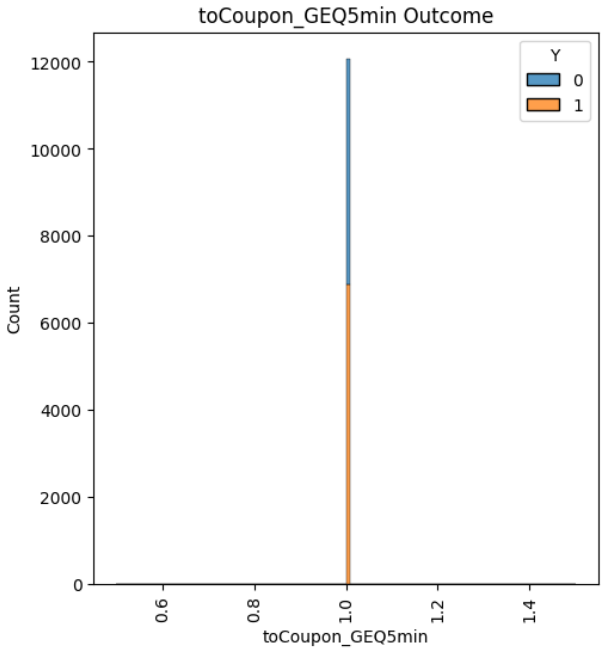

CarryAway Outcome
1~3
38.1%
never
34.5%
13.1%
4~8
less1
13.3%
gt8

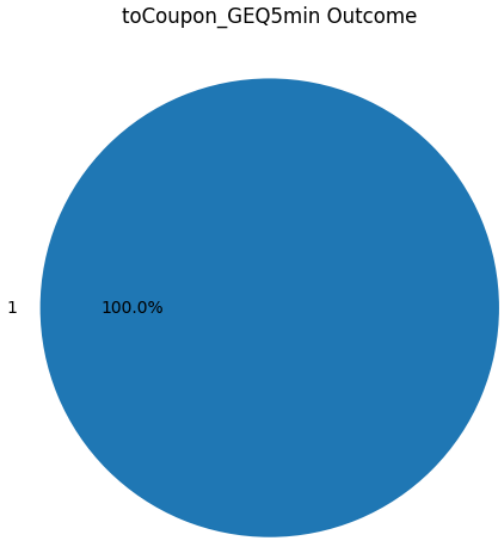

RestaurantLessThan20 Outcome
Y
0
1
Count
5000
4000
3000
2000
1000
0
4~8
1~3
less1
gt8
never
RestaurantLessThan20

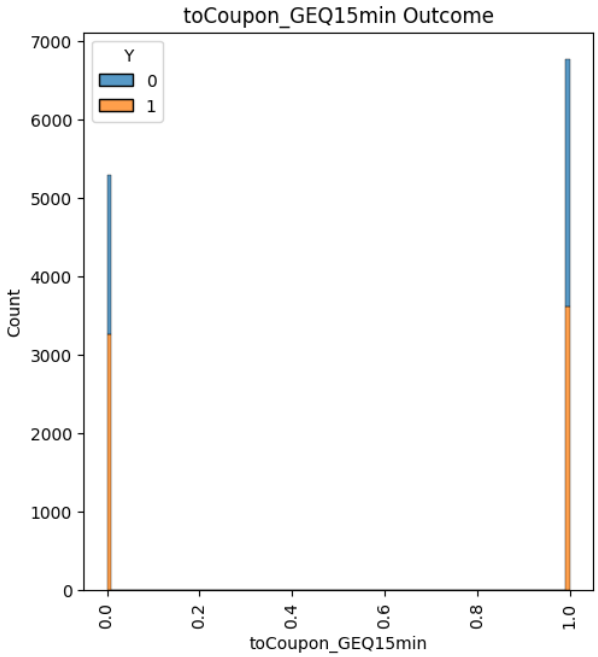

RestaurantLessThan20 Outcome
1~3
42.2%
never
29.3%
11.4%
4~8
gt8
15.7%
less1

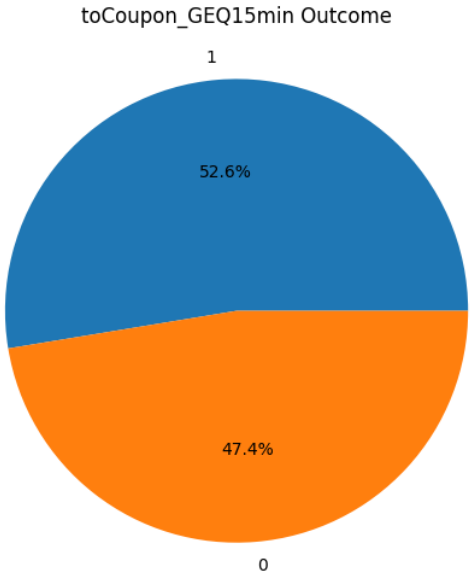

Restaurant20To50 Outcome
Y
0
1
Count
6000
5000
4000
3000
2000
1000
0
less1
never
1~3
gt8
4~8
Restaurant20To50

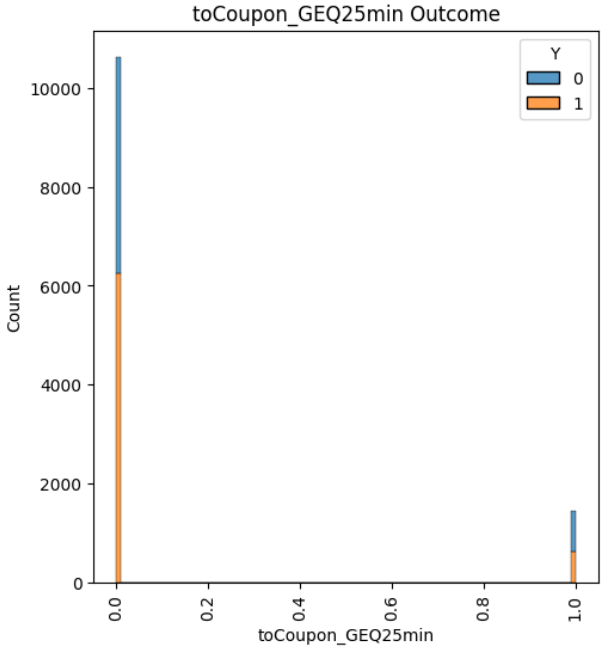

Restaurant20To50 Outcome
less1
47.6%
gt8
6.5%
4~8
27.8%
15.6%
1~3
never

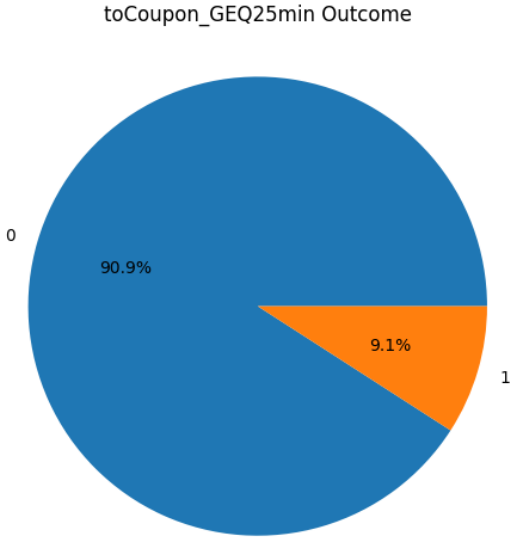

toCoupon_GEQ5min Outcome
Y
0
1
Count
12000
10000
8000
6000
4000
2000
0
0.6
0.8
1.0
1.2
1.4
toCoupon_GEQ5min

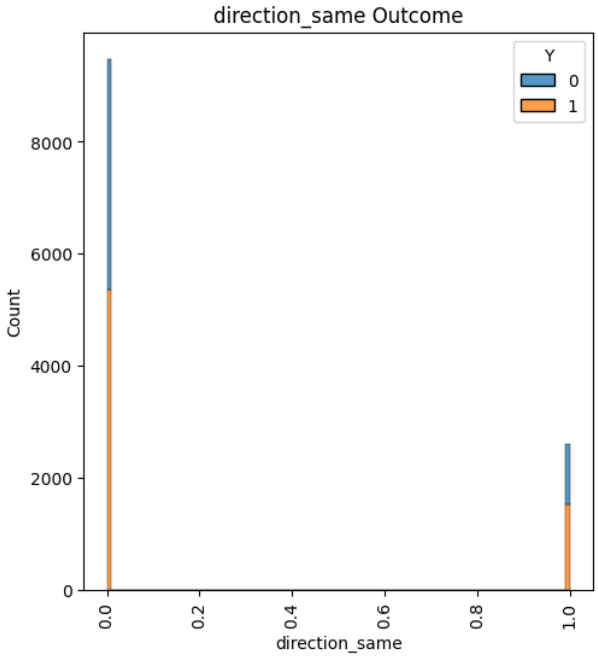

toCoupon_GEQ5min Outcome
1
100.0%

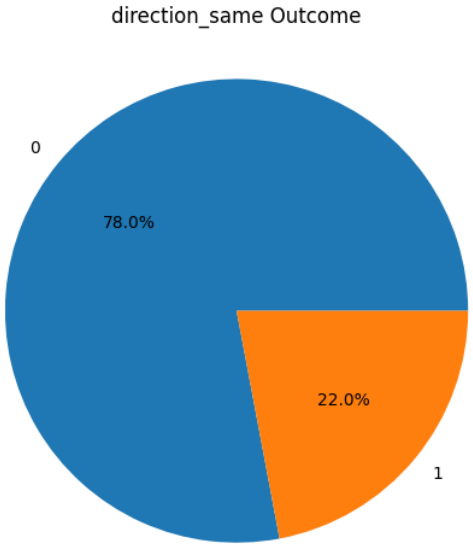

toCoupon_GEQ15min Outcome
Y
0
1
Count
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
toCoupon_GEQ15min

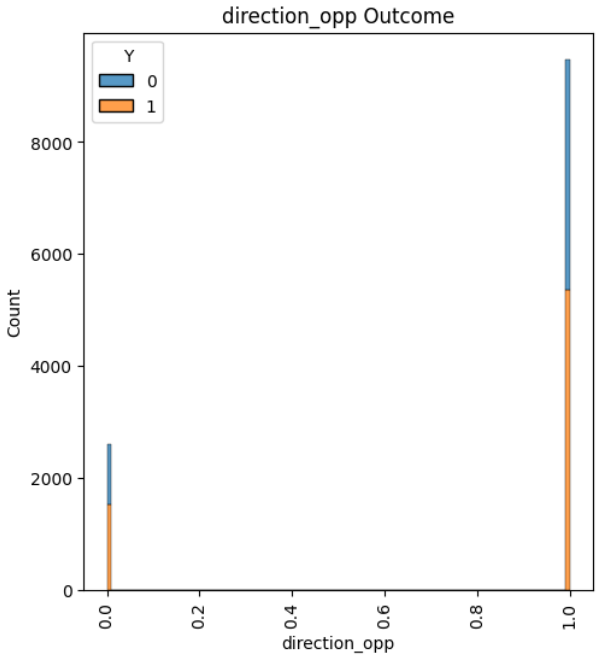

toCoupon_GEQ15min Outcome
1
52.6%
47.4%
0

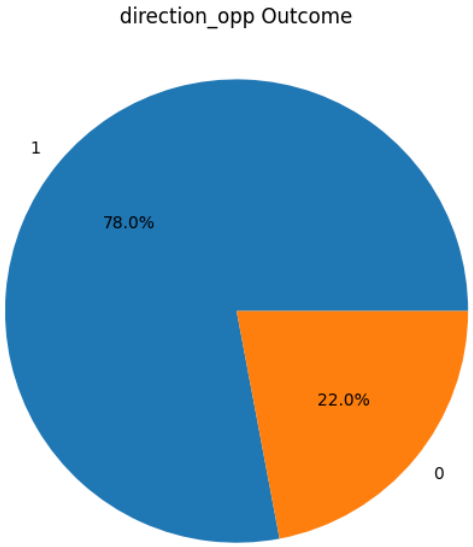

toCoupon_GEQ25min Outcome
Y
0
1
Count
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
toCoupon_GEQ25min


toCoupon_GEQ25min Outcome
0
90.9%
9.1%
1


direction_same Outcome
Y
0
1
8000
6000
4000
2000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
direction_same


direction_same Outcome
0
78.0%
22.0%
1


direction_opp Outcome
Y
0
1
8000
6000
4000
2000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
direction_opp


direction_opp Outcome
1
78.0%
22.0%
0

```
destination: destination
No Urgent Place    31.335375
Home               12.981207
Work               12.616938
Name: count, dtype: float64%
passanger: passanger
Alone        30.408146
Friend(s)    17.625631
Partner       5.041808
Kid(s)        3.857935
Name: count, dtype: float64%
weather: weather
Sunny    47.296962
Snowy     5.248779
Rainy     4.387780
Name: count, dtype: float64%
temperature: temperature
80    30.888319
55    16.284461
30     9.760742
Name: count, dtype: float64%
time: time
6PM     14.852223
7AM     12.616938
10AM    10.861826
2PM     10.481000
10PM     8.121533
Name: count, dtype: float64%
coupon: coupon
Coffee House             15.680106
Restaurant(<20)          15.572481
Carry out & Take away    13.924994
Bar                       6.523719
Restaurant(20-50)         5.232221
Name: count, dtype: float64%
expiration: expiration
1d    35.135359
2h    21.798162
Name: count, dtype: float64%
gender: gender
Male      28.835168
Female    28.098353
Name: count, dtype: float64%
age: age
21    12.476198
26    11.913238
31     8.800397
50     7.277092
36     5.588211
41     5.066645
46     3.170792
20     2.640947
Name: count, dtype: float64%
maritalStatus: maritalStatus
Single               23.180727
Married partner      21.715374
```

```
Unmarried partner     9.421310
Divorced              2.193890
Widowed               0.422220
Name: count, dtype: float64%
has_children: has_children
0    34.853879
1    22.079642
Name: count, dtype: float64%
education: education
Some college - no degree                   20.763308
Bachelors degree                           18.519745
Graduate degree (Masters or Doctorate)      7.881447
Associates degree                           4.934183
High School Graduate                        4.313271
Some High School                            0.521566
Name: count, dtype: float64%
occupation: occupation
Unemployed                                    8.196043
Student                                       7.542015
Computer & Mathematical                       6.382979
Sales & Related                               5.000414
Management                                    3.791705
Education&Training&Library                    3.634407
Office & Administrative Support               3.079725
Arts Design Entertainment Sports & Media      2.491928
Business & Financial                          2.433976
Retired                                       1.788228
Healthcare Support                            1.399122
Healthcare Practitioners & Technical          1.316334
Food Preparation & Serving Related            1.291498
Transportation & Material Moving              1.076248
Protective Service                            0.935508
Architecture & Engineering                    0.918950
Community & Social Services                   0.902393
Construction & Extraction                     0.877556
Legal                                         0.852720
Life Physical Social Science                  0.811325
Personal Care & Service                       0.703701
Installation Maintenance & Repair             0.587797
Production Occupations                        0.513288
Building & Grounds Cleaning & Maintenance     0.215250
Farming Fishing & Forestry                    0.190413
Name: count, dtype: float64%
income: income
$25000 - $37499     9.487540
$12500 - $24999     8.262273
$100000 or More     8.096697
$37500 - $49999     7.980793
$50000 - $62499     7.641361
Less than $12500    4.934183
$62500 - $74999     3.667522
$87500 - $99999     3.543340
$75000 - $87499     3.319811
Name: count, dtype: float64%
Bar: Bar
never    21.599470
```

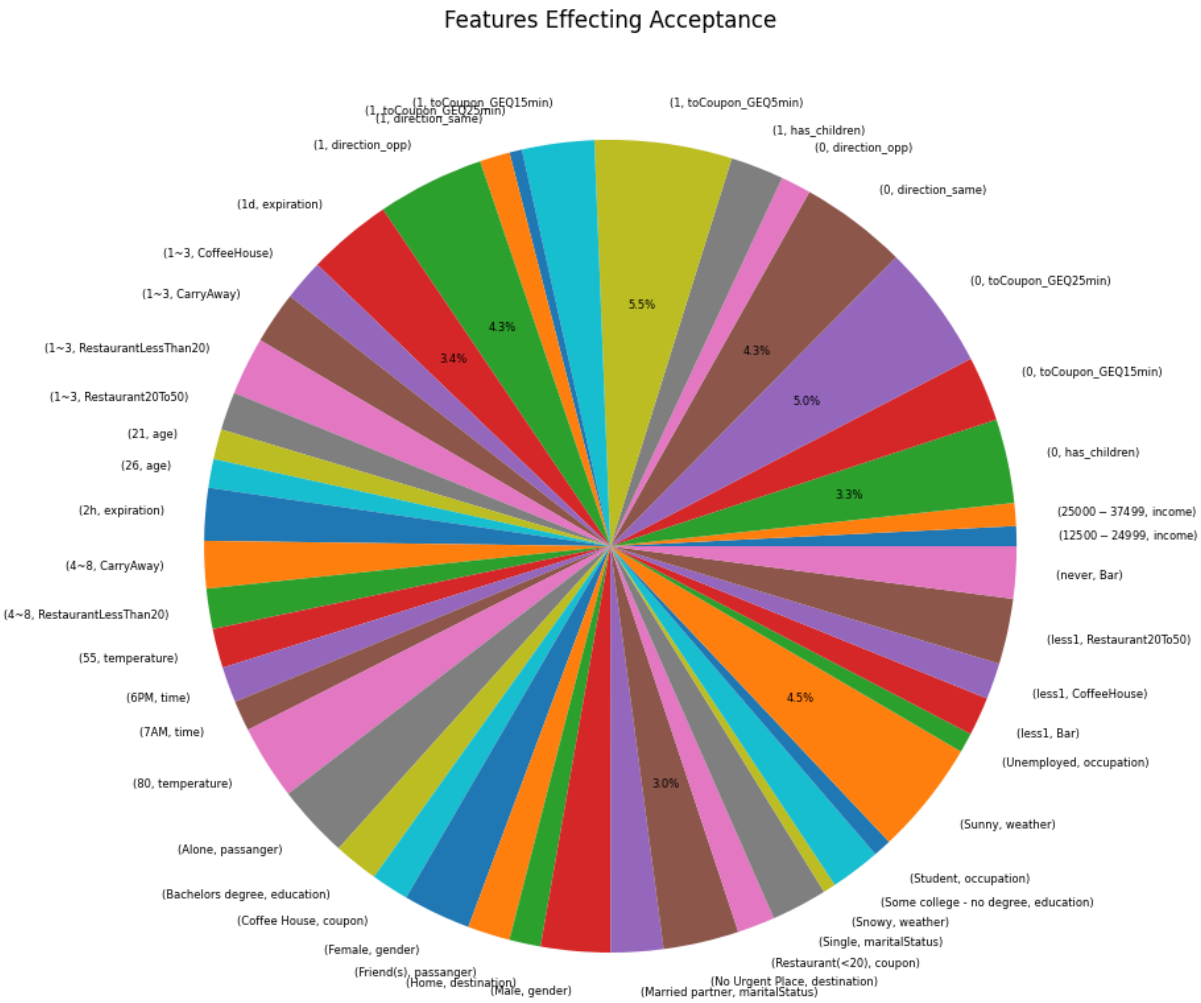
```
less1    15.969865
1~3      12.260949
4~8       5.563374
gt8       1.539863
Name: count, dtype: float64%
CoffeeHouse: CoffeeHouse
1~3      17.054392
less1    15.100588
never    10.745923
4~8       8.667936
gt8       5.364683
Name: count, dtype: float64%
CarryAway: CarryAway
1~3      21.715374
4~8      19.620830
gt8       7.558573
less1     7.442669
never     0.596076
Name: count, dtype: float64%
RestaurantLessThan20: RestaurantLessThan20
1~3      24.025168
4~8      16.657008
less1     8.932859
gt8       6.465767
never     0.852720
Name: count, dtype: float64%
Restaurant20To50: Restaurant20To50
less1    27.088335
1~3      15.837404
never     8.858349
4~8       3.700637
gt8       1.448795
Name: count, dtype: float64%
toCoupon_GEQ5min: toCoupon_GEQ5min
1    56.933521
Name: count, dtype: float64%
toCoupon_GEQ15min: toCoupon_GEQ15min
1    29.944532
0    26.988989
Name: count, dtype: float64%
toCoupon_GEQ25min: toCoupon_GEQ25min
0    51.750973
1     5.182548
Name: count, dtype: float64%
direction_same: direction_same
0    44.391092
1    12.542429
Name: count, dtype: float64%
direction_opp: direction_opp
1    44.391092
0    12.542429
Name: count, dtype: float64%
**********************************************************************
* Investigation: Which Feature most effect Coupon Acceptance      *
**********************************************************************
* This section examines each feature of the dataset, explore its  *
```

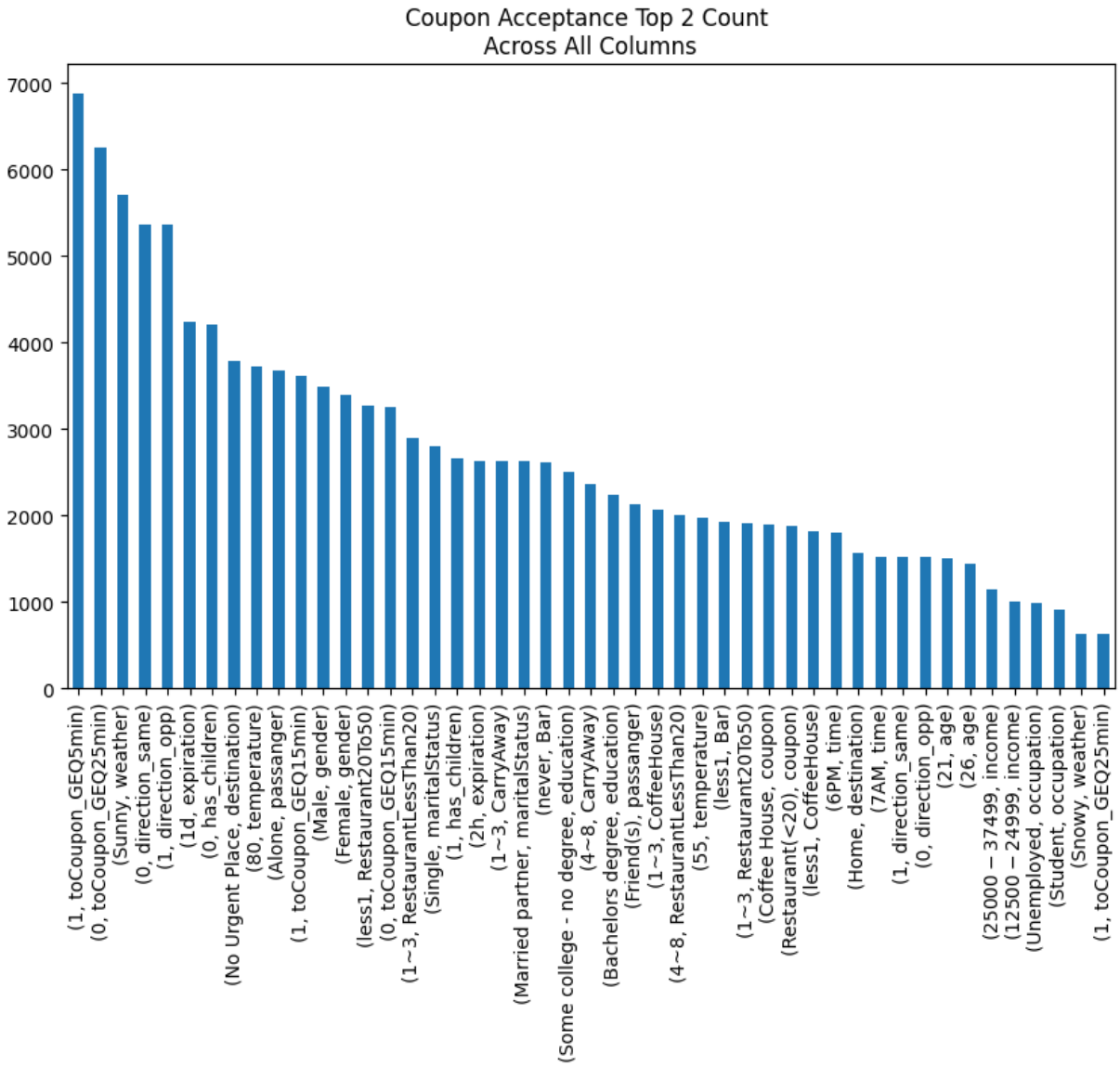
```
* effect on the acceptance of the coupon. A bar plot is using     *
* value_count() to show outcome counts for each feature with the  *
* pie char next to it to show the % of each.                      *
*******************************************************************
* Result:  unexpected outcome, as each each feature contributes   *
* equally to the coupon acceptance outcome, since I filter for    *
*  only the Y = 1 column, and all other columns have non-null data*
*  and therefore all contributed to the outcome.  Need to flatten *
*  each feature, using value_count() and find which of the single *
*  most important value of each feature effect the outcome.       *
*******************************************************************
```

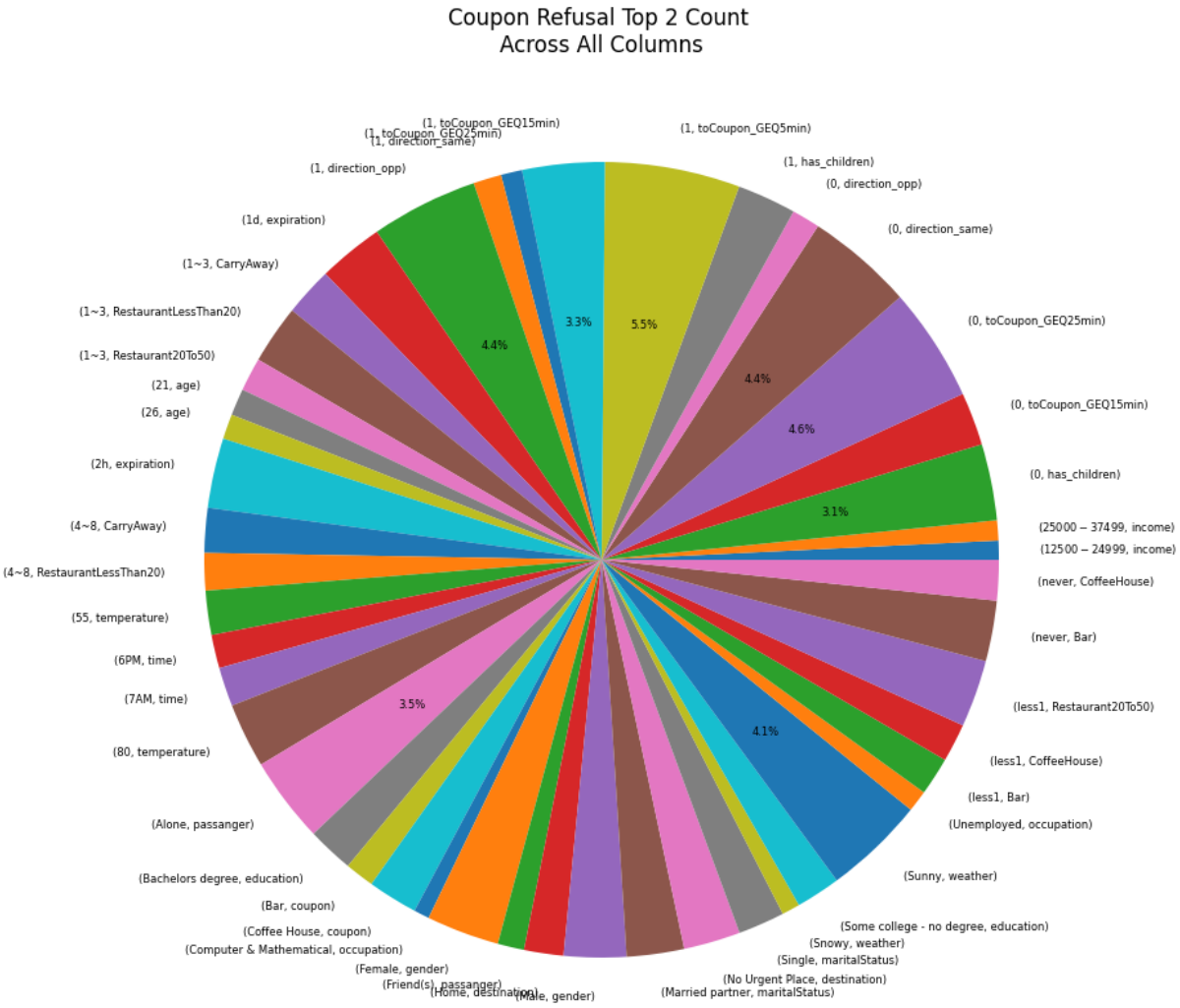
In [58]:
```
#
# Investigation: Which unique "features" are important to coupon acceptance
#
def explore_multi_index_features(focus_series, criteria_list, title, index0_
    result_series_list = []
    for keyword in criteria_list:
        # Search for the keyword in the first level of the MultiIndex
        filtered_series = focus_series[ (focus_series.index.get_level_values
                                    ~(focus_series.index.get_level_values(0).
        result_series_list.append( filtered_series )

    results = pd.concat( result_series_list ).sort_values(ascending=False)
    results.plot(kind='bar',figsize=(8,6))
    plt.title(title)
    plt.show()

def plot_bar(df,title):
    df.plot(kind='bar', figsize=(10,6))
    plt.title(title)
    plt.show()

def plot_pie(df,title):
    plt.figure(figsize=(14,10))
    df.plot(kind='pie',autopct=lambda p: f'{p:.1f}%' if p>3.0 else '',labeld
    plt.title( title )
    plt.show()

df = pd.read_csv('data/coupons.csv')

#  Clean out data
df = df.drop( columns=['car'] )

#  Clean out data, normalize data
df['age'] = df['age'].replace('50plus' ,'50')
df['age'] = df['age'].replace('below21','20')
df['age'] = df['age'].astype('int64')
df = df.dropna()

# Analyze top coupon acceptance feature criterias
# Prepare data fixing data type for operation
df_work = df[df['Y']==1].copy()
df_work = df_work.drop( columns=['Y'] )
df_work = df_work.applymap(str)
#####################################################
```

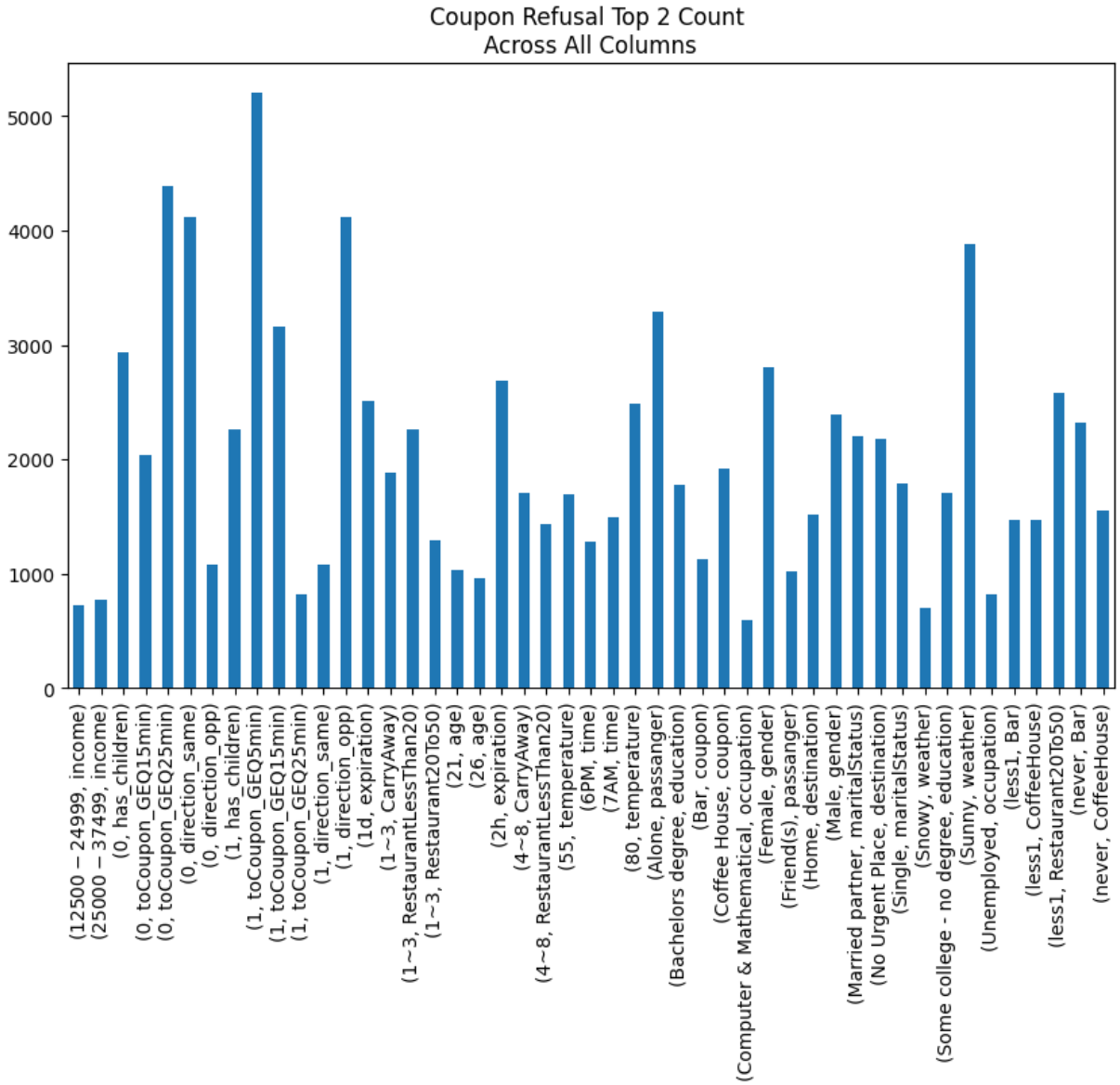
```
#   Individual features % with positive outcome   #
#####################################################
top2_value_counts_1 = df_work.apply(lambda x: x.value_counts().nlargest(2))
top_value_counts_1  = df_work.apply(lambda x: x.value_counts())

value_counts_1      = top2_value_counts_1.stack()   # flatten DataFrame inde
value_counts_1_all  = top_value_counts_1.stack()

# show each feature effecting coupon acceptance
plot_pie( value_counts_1, 'Features Effecting Acceptance' )

# sorted view of coupon acceptance for each feature
value_counts_sorted_1 = value_counts_1.sort_values(ascending=False)
plot_bar( value_counts_sorted_1, 'Coupon Acceptance Top 2 Count \nAcross All

#####################################################
#   Individual features % with negative outcome   #
#####################################################
df_work = df[df['Y']==0].copy()
df_work = df_work.drop( columns=['Y'] )
df_work = df_work.applymap(str)

top2_value_counts_0 = df_work.apply(lambda x: x.value_counts().nlargest(2))
top_value_counts_0  = df_work.apply(lambda x: x.value_counts())
value_counts_0      = top2_value_counts_0.stack()
value_counts_0_all  = top_value_counts_0.stack()

# show each feature effecting coupon refusal
plot_pie( value_counts_0,'Coupon Refusal Top 2 Count \nAcross All Columns' )

plot_bar( value_counts_0, 'Coupon Refusal Top 2 Count \nAcross All Columns')

# sorted view of coupon refusal for each feature
value_counts_0_sorted = value_counts_0.sort_values(ascending=False)
plot_bar( value_counts_0_sorted, 'Coupon Refusal Top 2 Count \nAcross All Co


#####################################################################
#   Analysis each features coupon acceptance / refusal, normalize   #
#   Per feature, positive means coupon acceptance, negative refusal #
#####################################################################
value_count_diff = value_counts_1_all - value_counts_0_all
value_diff = value_count_diff.sort_values(ascending=False)

# focus only on head, where features effecting positive coupon acceptance
# tail where negative coupon acceptance
# plot
focus_series = pd.concat( [value_diff.head(60) ] )
plot_bar( focus_series,"Coupon Acceptance / Refusal by Feature" )

focus_series = pd.concat( [value_diff.tail(60) ] )
plot_bar( focus_series,"Coupon Acceptance / Refusal by Feature" )

#############################################################
# Promotion Target: Background, Incoming, Weather         #
# Prior Visits                                            #
```

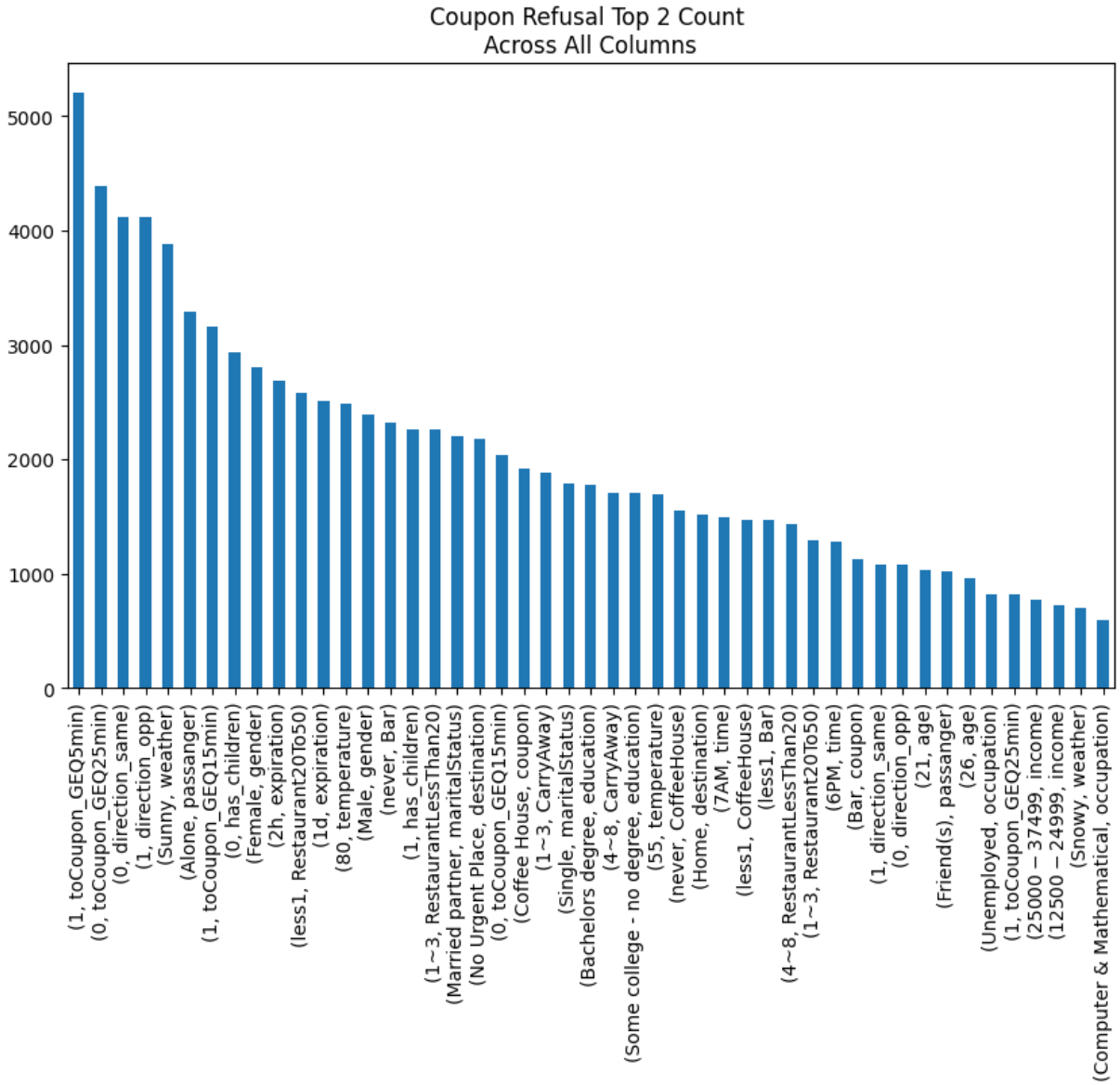
```
# Coupon Type                                                  #
###############################################################
###############################################################
# feature exploration                                          #
###############################################################
focus_series = value_diff

condition_criteria  = ['destination','weather','temperature','passanger','ti
background_criteria = ['maritalStatus','age','gender','has_children', 'occup
coupon_criteria     = ['coupon','toCoupon_GEQ5min','toCoupon_GEQ15min','toCo
exp_criteria        = ['expiration']
prior_visits        = ['Bar','CoffeeHouse','CarrayAway','ResturantLessThan20

explore_multi_index_features( focus_series, coupon_criteria,    'All Coupon
explore_multi_index_features( focus_series, coupon_criteria,    'Coupon Acce
explore_multi_index_features( focus_series, condition_criteria, 'Best Condit
explore_multi_index_features( focus_series, background_criteria,'Target Back
explore_multi_index_features( focus_series, prior_visits,       'Target Habi

print(f'*****************************************************************')
print(f'* Investigation 2: which (value) for the features most effect    *')
print(f'*    positive coupon acceptance                                 *')
print(f'*****************************************************************')
print(f'* 1.  Flatten all features for both positive & negative case     *')
print(f'* 2.  Subtract positive case count over negative cases           *')
print(f'* 3.  Sort the DataFrame, and examine top 10, and bottom 10      *')
print(f'* 4.  Found out results need to be based on certain criteria     *')
print(f'* 5.  Created criterias:                                         *')
print(f'*      a. based on the environment related conditions when the    *')
print(f'*         coupon was issued                                       *')
print(f'*      b. personal background, age, gender, income, etc...        *')
print(f'*      c. how soon coupon going to expire                         *')
print(f'*      d. prior visiting habits (bars, coffee house, cheap etc... *')
print(f'*      e. how far to redeem coupon                                *')
print(f'*****************************************************************')
print(f'* A1. Found need to future filter multi-index for 0/1 for some    *')
print(f'*     features                                                    *')
print(f'*****************************************************************')
print(f'* Findings: (for best acceptances)                               *')
print(f'* Conditions:  Sunny, No Urgent destination, with passanger      *')
print(f'*              afternoon to early evening                        *')
print(f'* Background: Male Single, Female Single, 21-26, <= 1 child      *')
print(f'* Habits: 1-3 Coffee House, 1-3 Bar visits                       *')
print(f'*****************************************************************')
```

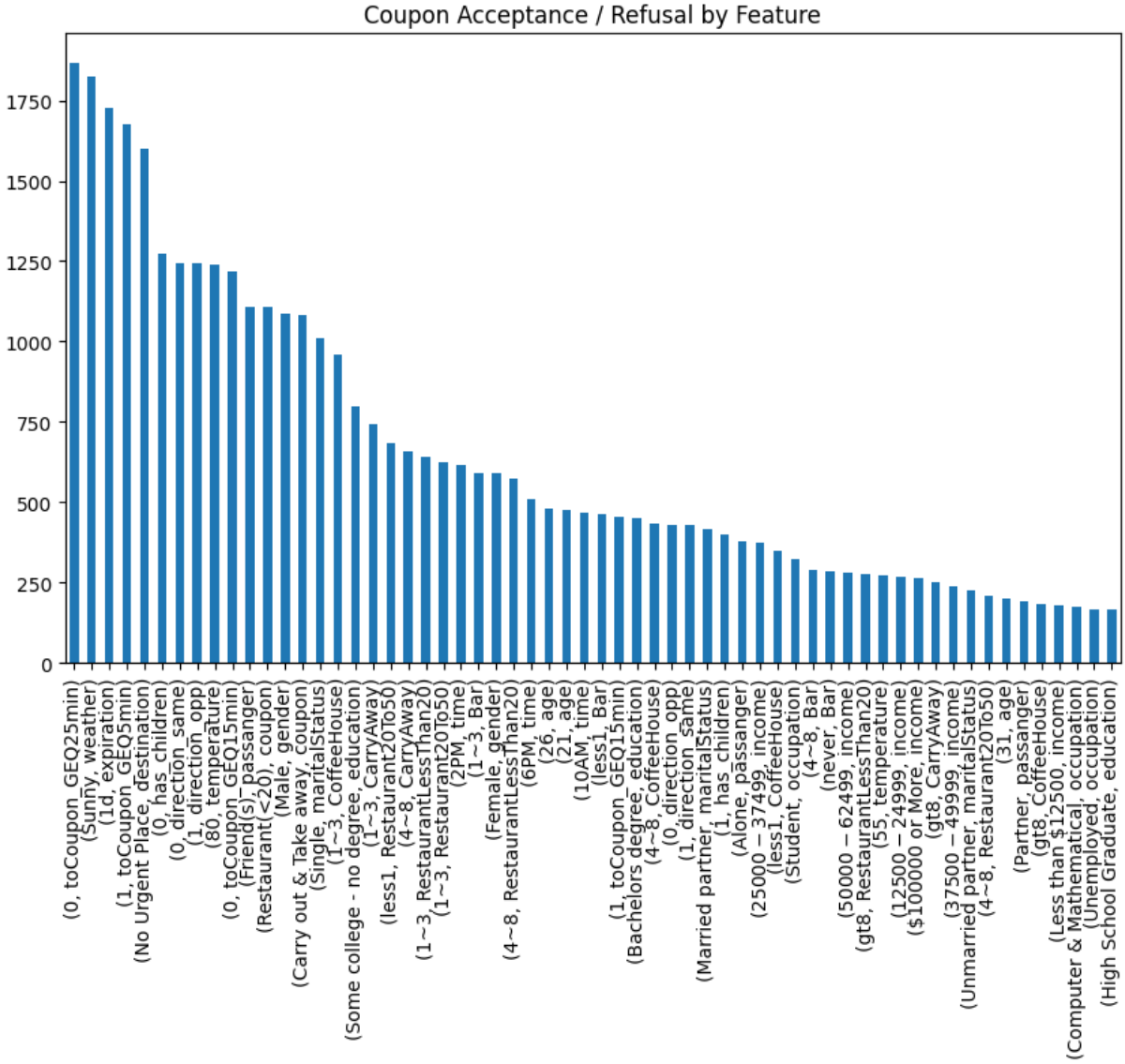

Features Effecting Acceptance
(1, toCoupon_GEQ15min)
(1, toCoupon_GEQ5min)
(1, toCoupon_GEQ25min)
(1, direction_same)
(1, has_children)
(1, direction_opp)
(0, direction_opp)
(1d, expiration)
(0, direction_same)
(1~3, CoffeeHouse)
(1~3, CarryAway)
(0, toCoupon_GEQ25min)
(1~3, RestaurantLessThan20)
(0, toCoupon_GEQ15min)
(1~3, Restaurant20To50)
(21, age)
(0, has_children)
(26, age)
(2h, expiration)
(25000 – 37499, income)
(12500 – 24999, income)
(4~8, CarryAway)
(never, Bar)
(4~8, RestaurantLessThan20)
(less1, Restaurant20To50)
(55, temperature)
(less1, CoffeeHouse)
(6PM, time)
(less1, Bar)
(7AM, time)
(Unemployed, occupation)
(80, temperature)
(Sunny, weather)
(Alone, passanger)
(Student, occupation)
(Bachelors degree, education)
(Some college - no degree, education)
(Coffee House, coupon)
(Snowy, weather)
(Female, gender)
(Single, maritalStatus)
(Friend(s), passanger)
(Restaurant(<20), coupon)
(Home, destination)
(No Urgent Place, destination)
(Male, gender)
(Married partner, maritalStatus)
5.5%
4.3%
4.3%
5.0%
3.4%
4.3%
3.3%
4.5%
3.0%

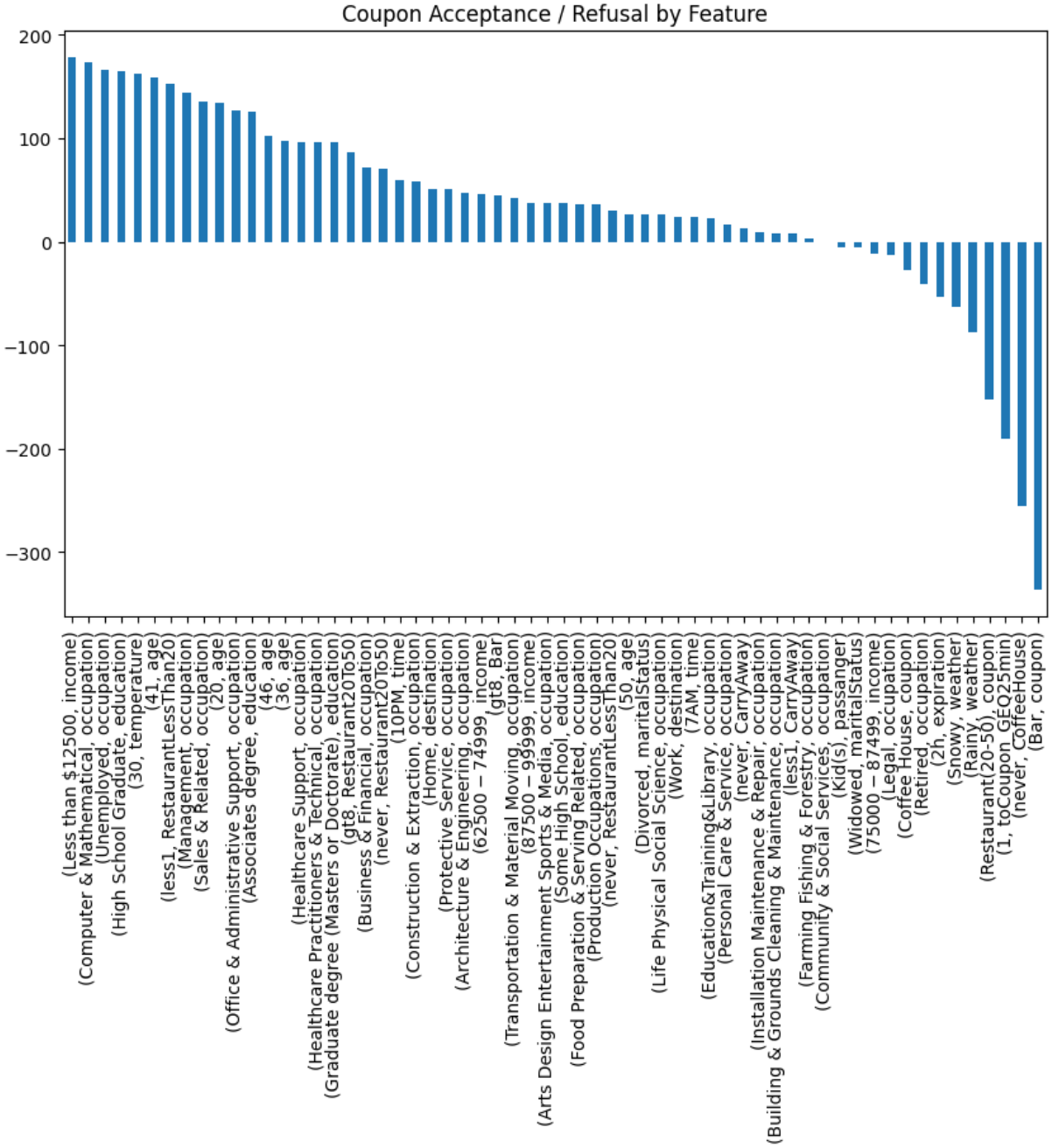

Coupon Acceptance Top 2 Count
Across All Columns
7000
6000
5000
4000
3000
2000
1000
0
(1, toCoupon_GEQ5min)
(0, toCoupon_GEQ25min)
(Sunny, weather)
(0, direction_same)
(1, direction_opp)
(1d, expiration)
(0, has_children)
(No Urgent Place, destination)
(80, temperature)
(Alone, passanger)
(1, toCoupon_GEQ15min)
(Male, gender)
(Female, gender)
(less1, Restaurant20To50)
(0, toCoupon_GEQ15min)
(1~3, RestaurantLessThan20)
(Single, maritalStatus)
(1, has_children)
(2h, expiration)
(1~3, CarryAway)
(Married partner, maritalStatus)
(never, Bar)
(Some college - no degree, education)
(4~8, CarryAway)
(Bachelors degree, education)
(Friend(s), passanger)
(1~3, CoffeeHouse)
(4~8, RestaurantLessThan20)
(55, temperature)
(less1, Bar)
(1~3, Restaurant20To50)
(Coffee House, coupon)
(Restaurant(<20), coupon)
(less1, CoffeeHouse)
(6PM, time)
(Home, destination)
(7AM, time)
(1, direction_same)
(0, direction_opp)
(21, age)
(26, age)
(25000 - 37499, income)
(12500 - 24999, income)
(Unemployed, occupation)
(Student, occupation)
(Snowy, weather)
(1, toCoupon_GEQ25min)

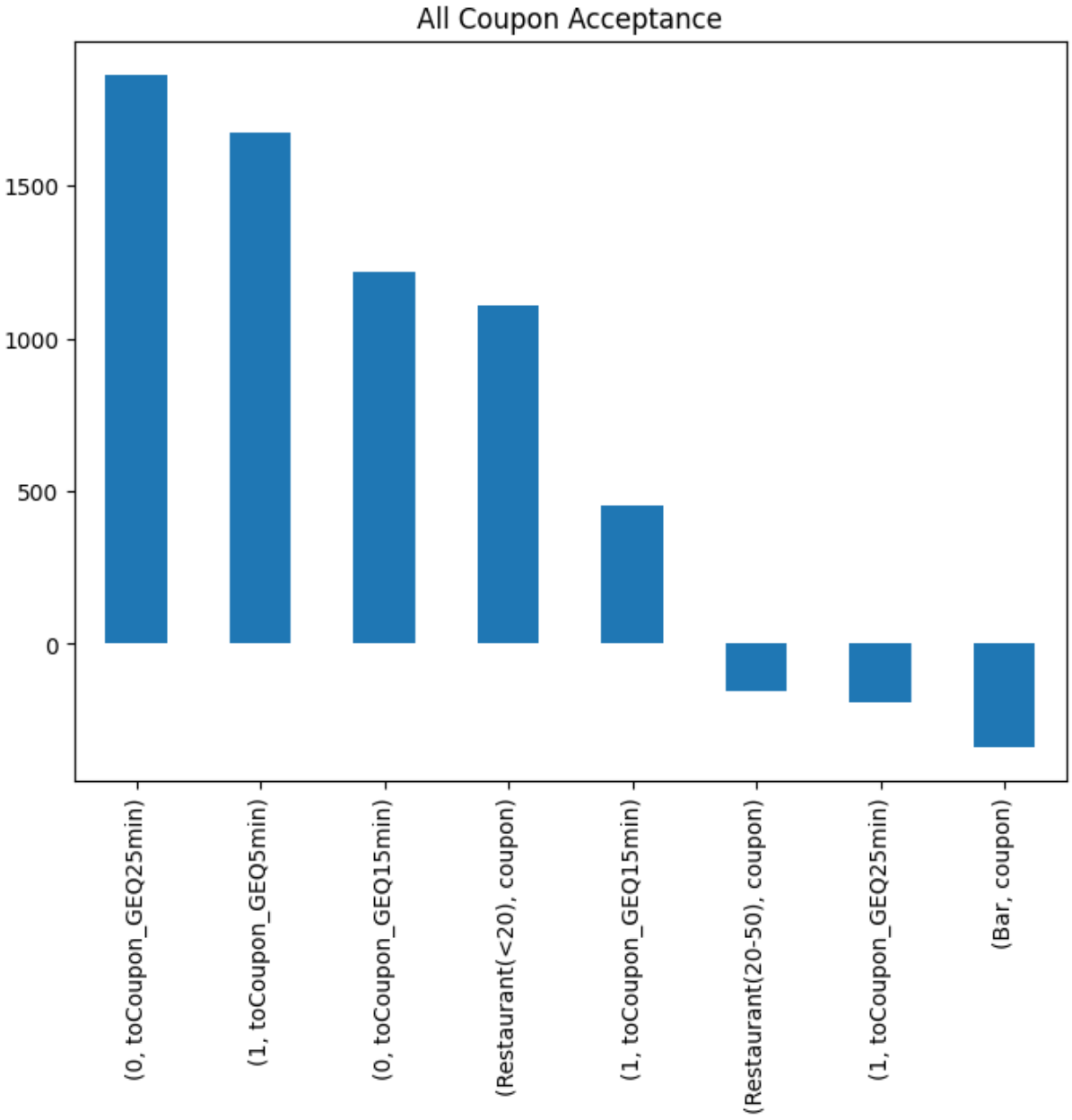

Coupon Refusal Top 2 Count
Across All Columns
(1, toCoupon_GEQ15min)
(1, toCoupon_GEQ25min)
(1, direction_same)
(1, toCoupon_GEQ5min)
(1, has_children)
(0, direction_opp)
(1, direction_opp)
(0, direction_same)
(1d, expiration)
(1~3, CarryAway)
(1~3, RestaurantLessThan20)
(1~3, Restaurant20To50)
(21, age)
(26, age)
(2h, expiration)
(4~8, CarryAway)
(4~8, RestaurantLessThan20)
(55, temperature)
(6PM, time)
(7AM, time)
(80, temperature)
(Alone, passanger)
(Bachelors degree, education)
(Bar, coupon)
(Coffee House, coupon)
(Computer & Mathematical, occupation)
(Female, gender)
(Friend(s), passanger)
(Home, destination)
(Male, gender)
(Married partner, maritalStatus)
(No Urgent Place, destination)
(Single, maritalStatus)
(Snowy, weather)
(Some college - no degree, education)
(Sunny, weather)
(Unemployed, occupation)
(less1, Bar)
(less1, CoffeeHouse)
(less1, Restaurant20To50)
(never, Bar)
(never, CoffeeHouse)
(12500 – 24999, income)
(25000 – 37499, income)
(0, has_children)
(0, toCoupon_GEQ15min)
(0, toCoupon_GEQ25min)
3.3%
5.5%
4.4%
4.4%
4.6%
3.1%
3.5%
4.1%

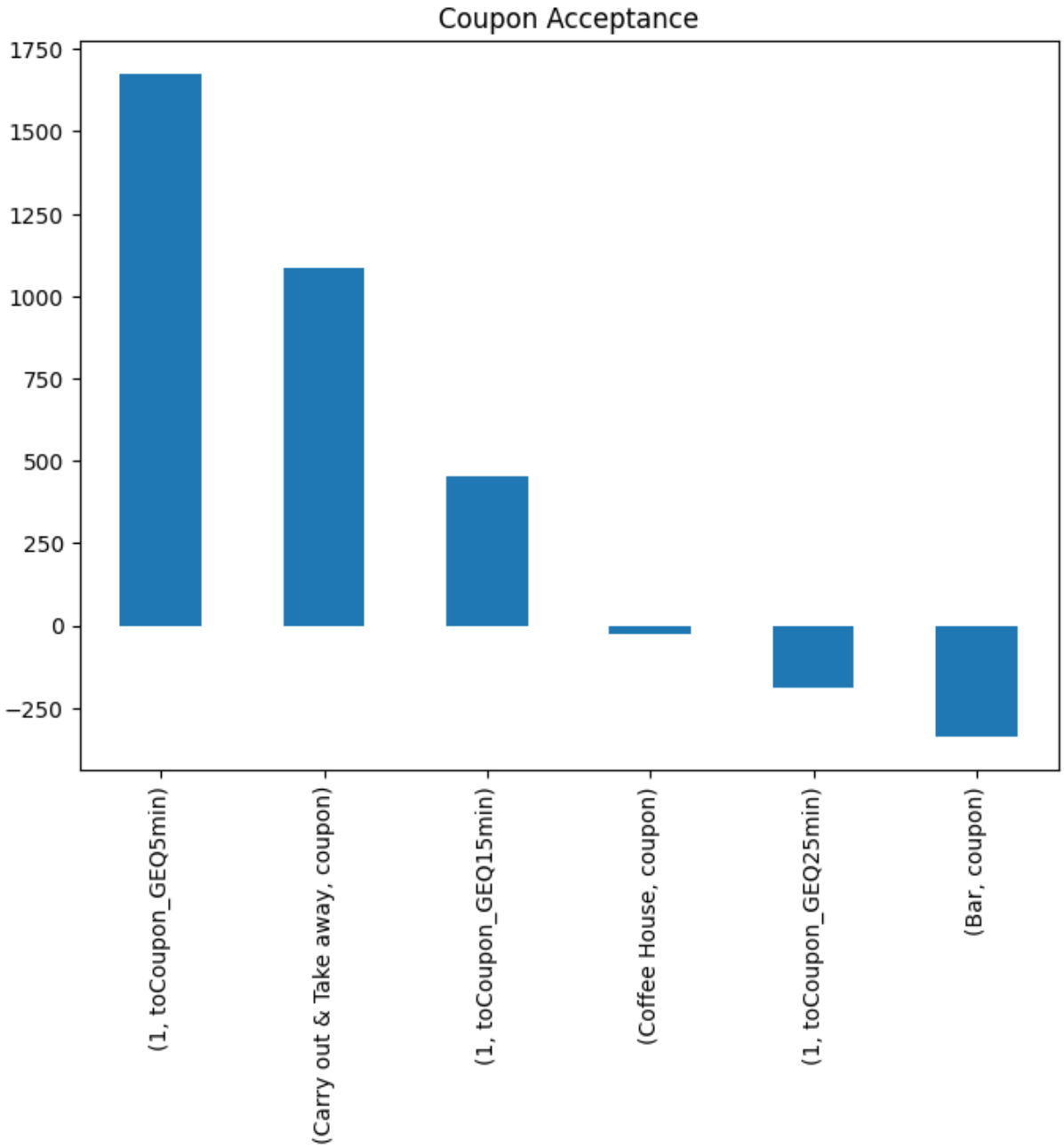

Coupon Refusal Top 2 Count
Across All Columns
5000
4000
3000
2000
1000
0
(12500 − 24999, income)
(25000 − 37499, income)
(0, has_children)
(0, toCoupon_GEQ15min)
(0, toCoupon_GEQ25min)
(0, direction_same)
(0, direction_opp)
(1, has_children)
(1, toCoupon_GEQ5min)
(1, toCoupon_GEQ15min)
(1, toCoupon_GEQ25min)
(1, direction_same)
(1, direction_opp)
(1d, expiration)
(1~3, CarryAway)
(1~3, RestaurantLessThan20)
(1~3, Restaurant20To50)
(21, age)
(26, age)
(2h, expiration)
(4~8, CarryAway)
(4~8, RestaurantLessThan20)
(55, temperature)
(6PM, time)
(7AM, time)
(80, temperature)
(Alone, passanger)
(Bachelors degree, education)
(Bar, coupon)
(Coffee House, coupon)
(Computer & Mathematical, occupation)
(Female, gender)
(Friend(s), passanger)
(Home, destination)
(Male, gender)
(Married partner, maritalStatus)
(No Urgent Place, destination)
(Single, maritalStatus)
(Snowy, weather)
(Some college - no degree, education)
(Sunny, weather)
(Unemployed, occupation)
(less1, Bar)
(less1, CoffeeHouse)
(less1, Restaurant20To50)
(never, Bar)
(never, CoffeeHouse)

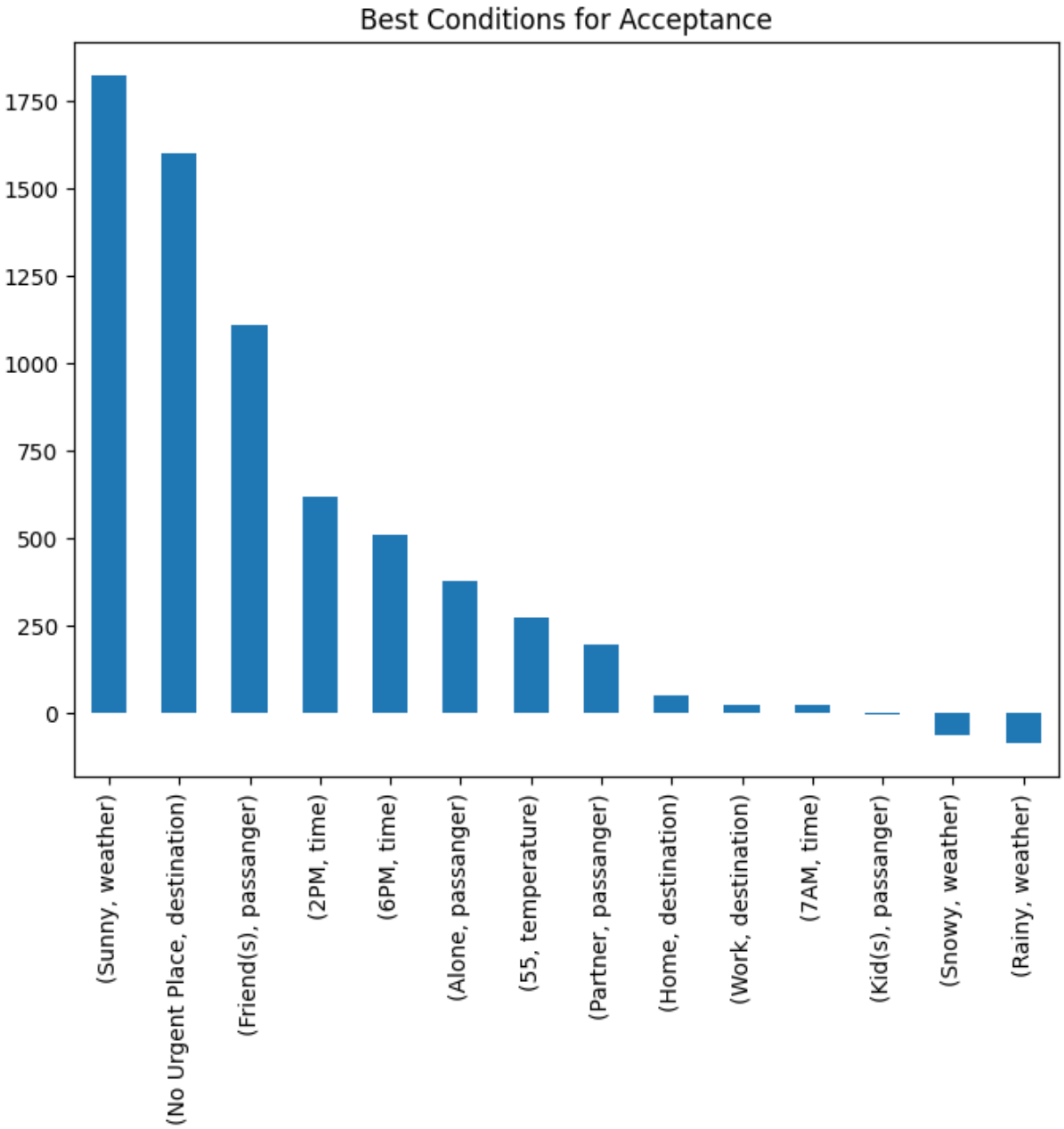

Coupon Refusal Top 2 Count
Across All Columns
5000
4000
3000
2000
1000
0
(1, toCoupon_GEQ5min)
(0, toCoupon_GEQ25min)
(0, direction_same)
(1, direction_opp)
(Sunny, weather)
(Alone, passanger)
(1, toCoupon_GEQ15min)
(0, has_children)
(Female, gender)
(2h, expiration)
(less1, Restaurant20To50)
(1d, expiration)
(80, temperature)
(Male, gender)
(never, Bar)
(1, has_children)
(1~3, RestaurantLessThan20)
(Married partner, maritalStatus)
(No Urgent Place, destination)
(0, toCoupon_GEQ15min)
(Coffee House, coupon)
(1~3, CarryAway)
(Single, maritalStatus)
(Bachelors degree, education)
(4~8, CarryAway)
(Some college - no degree, education)
(55, temperature)
(never, CoffeeHouse)
(Home, destination)
(7AM, time)
(less1, CoffeeHouse)
(less1, Bar)
(4~8, RestaurantLessThan20)
(1~3, Restaurant20To50)
(6PM, time)
(Bar, coupon)
(1, direction_same)
(0, direction_opp)
(21, age)
(Friend(s), passanger)
(26, age)
(Unemployed, occupation)
(1, toCoupon_GEQ25min)
(25000 - 37499, income)
(12500 - 24999, income)
(Snowy, weather)
(Computer & Mathematical, occupation)

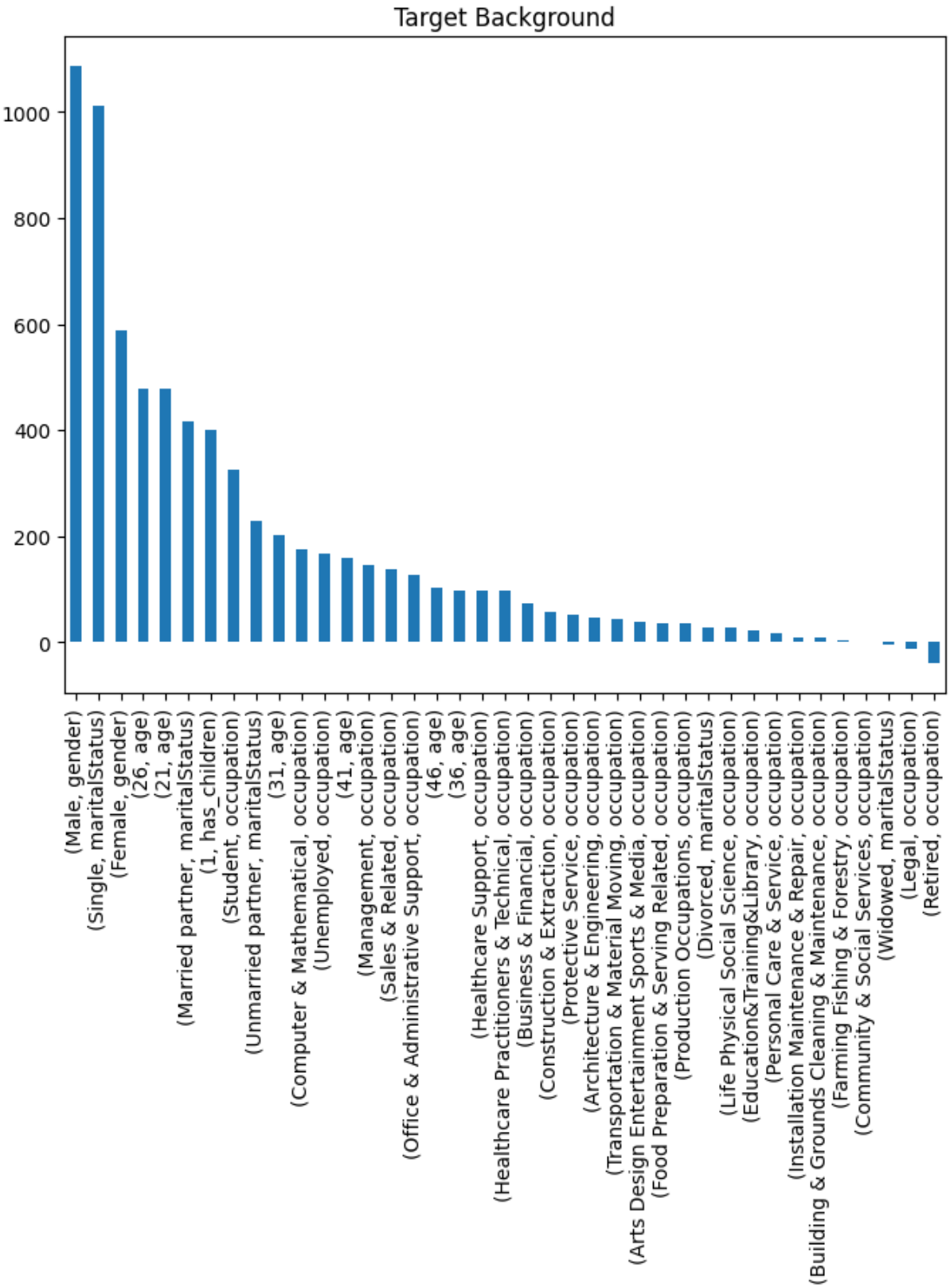

Coupon Acceptance / Refusal by Feature
1750
1500
1250
1000
750
500
250
0
(0, toCoupon_GEQ25min)
(Sunny, weather)
(1d, expiration)
(1, toCoupon_GEQ5min)
(No Urgent Place, destination)
(0, has_children)
(0, direction_same)
(1, direction_opp)
(80, temperature)
(0, toCoupon_GEQ15min)
(Friend(s), passanger)
(Restaurant(<20), coupon)
(Male, gender)
(Carry out & Take away, coupon)
(Single, maritalStatus)
(1~3, CoffeeHouse)
(Some college - no degree, education)
(1~3, CarryAway)
(less1, Restaurant20To50)
(4~8, CarryAway)
(1~3, RestaurantLessThan20)
(1~3, Restaurant20To50)
(2PM, time)
(1~3, Bar)
(Female, gender)
(4~8, RestaurantLessThan20)
(6PM, time)
(26, age)
(21, age)
(10AM, time)
(less1, Bar)
(1, toCoupon_GEQ15min)
(Bachelors degree, education)
(4~8, CoffeeHouse)
(0, direction_opp)
(1, direction_same)
(Married partner, maritalStatus)
(1, has_children)
(Alone, passanger)
(25000 – 37499, income)
(less1, CoffeeHouse)
(Student, occupation)
(4~8, Bar)
(never, Bar)
(50000 – 62499, income)
(gt8, RestaurantLessThan20)
(55, temperature)
(12500 – 24999, income)
($100000 or More, income)
(gt8, CarryAway)
(37500 – 49999, income)
(Unmarried partner, maritalStatus)
(4~8, Restaurant20To50)
(31, age)
(Partner, passanger)
(gt8, CoffeeHouse)
(Less than $12500, income)
(Computer & Mathematical, occupation)
(Unemployed, occupation)
(High School Graduate, education)

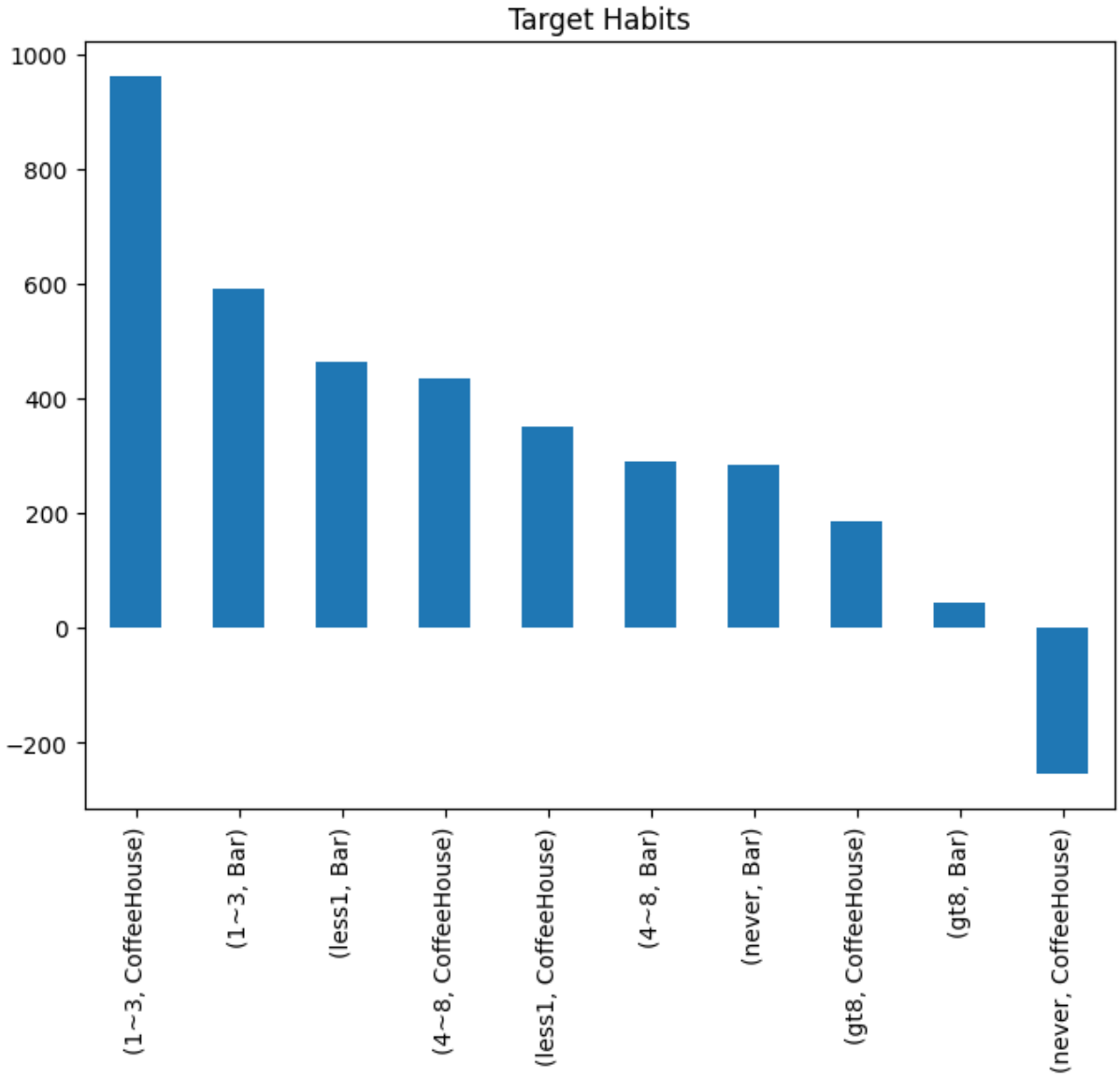

Coupon Acceptance / Refusal by Feature
200
100
0
−100
−200
−300
(Less than $12500, income)
(Computer & Mathematical, occupation)
(Unemployed, occupation)
(High School Graduate, education)
(30, temperature)
(41, age)
(less1, RestaurantLessThan20)
(Management, occupation)
(Sales & Related, occupation)
(20, age)
(Office & Administrative Support, occupation)
(Associates degree, education)
(46, age)
(36, age)
(Healthcare Support, occupation)
(Healthcare Practitioners & Technical, occupation)
(Graduate degree (Masters or Doctorate), education)
(gt8, Restaurant20To50)
(Business & Financial, occupation)
(never, Restaurant20To50)
(10PM, time)
(Construction & Extraction, occupation)
(Home, destination)
(Protective Service, occupation)
(Architecture & Engineering, occupation)
(62500 – 74999, income)
(gt8, Bar)
(Transportation & Material Moving, occupation)
(87500 – 99999, income)
(Arts Design Entertainment Sports & Media, occupation)
(Some High School, education)
(Food Preparation & Serving Related, occupation)
(Production Occupations, occupation)
(never, RestaurantLessThan20)
(50, age)
(Divorced, maritalStatus)
(Life Physical Social Science, occupation)
(Work, destination)
(7AM, time)
(Education&Training&Library, occupation)
(Personal Care & Service, occupation)
(never, CarryAway)
(Installation Maintenance & Repair, occupation)
(Building & Grounds Cleaning & Maintenance, occupation)
(less1, CarryAway)
(Farming Fishing & Forestry, occupation)
(Community & Social Services, occupation)
(Kid(s), passanger)
(Widowed, maritalStatus)
(75000 – 87499, income)
(Legal, occupation)
(Coffee House, coupon)
(Retired, occupation)
(2h, expiration)
(Snowy, weather)
(Rainy, weather)
(Restaurant(20-50), coupon)
(1, toCoupon_GEQ25min)
(never, CoffeeHouse)
(Bar, coupon)


All Coupon Acceptance
1500
1000
500
0
(0, toCoupon_GEQ25min)
(1, toCoupon_GEQ5min)
(0, toCoupon_GEQ15min)
(Restaurant(<20), coupon)
(1, toCoupon_GEQ15min)
(Restaurant(20-50), coupon)
(1, toCoupon_GEQ25min)
(Bar, coupon)


Coupon Acceptance
1750
1500
1250
1000
750
500
250
0
−250
(1, toCoupon_GEQ5min)
(Carry out & Take away, coupon)
(1, toCoupon_GEQ15min)
(Coffee House, coupon)
(1, toCoupon_GEQ25min)
(Bar, coupon)


Best Conditions for Acceptance
1750
1500
1250
1000
750
500
250
0
(Sunny, weather)
(No Urgent Place, destination)
(Friend(s), passanger)
(2PM, time)
(6PM, time)
(Alone, passanger)
(55, temperature)
(Partner, passanger)
(Home, destination)
(Work, destination)
(7AM, time)
(Kid(s), passanger)
(Snowy, weather)
(Rainy, weather)


Target Background
1000
800
600
400
200
0
(Male, gender)
(Single, maritalStatus)
(Female, gender)
(26, age)
(21, age)
(Married partner, maritalStatus)
(1, has_children)
(Student, occupation)
(Unmarried partner, maritalStatus)
(31, age)
(Computer & Mathematical, occupation)
(Unemployed, occupation)
(41, age)
(Management, occupation)
(Sales & Related, occupation)
(Office & Administrative Support, occupation)
(46, age)
(36, age)
(Healthcare Support, occupation)
(Healthcare Practitioners & Technical, occupation)
(Business & Financial, occupation)
(Construction & Extraction, occupation)
(Protective Service, occupation)
(Architecture & Engineering, occupation)
(Transportation & Material Moving, occupation)
(Arts Design Entertainment Sports & Media, occupation)
(Food Preparation & Serving Related, occupation)
(Production Occupations, occupation)
(Divorced, maritalStatus)
(Life Physical Social Science, occupation)
(Education&Training&Library, occupation)
(Personal Care & Service, occupation)
(Installation Maintenance & Repair, occupation)
(Building & Grounds Cleaning & Maintenance, occupation)
(Farming Fishing & Forestry, occupation)
(Community & Social Services, occupation)
(Widowed, maritalStatus)
(Legal, occupation)
(Retired, occupation)


Target Habits
1000
800
600
400
200
0
−200
(1~3, CoffeeHouse)
(1~3, Bar)
(less1, Bar)
(4~8, CoffeeHouse)
(less1, CoffeeHouse)
(4~8, Bar)
(never, Bar)
(gt8, CoffeeHouse)
(gt8, Bar)
(never, CoffeeHouse)

```
**********************************************************************
* Investigation 2: which (value) for the features most effect    *
*    positive coupon acceptance                                  *
**********************************************************************
* 1.  Flatten all features for both positive & negative case      *
* 2.  Subtract positive case count over negative cases            *
* 3.  Sort the DataFrame, and examine top 10, and bottom 10       *
* 4.  Found out results need to be based on certain criteria      *
* 5.  Created criterias:                                          *
*      a. based on the environment related conditions when the    *
*         coupon was issued                                       *
*      b. personal background, age, gender, income, etc...        *
*      c. how soon coupon going to expire                         *
*      d. prior visiting habits (bars, coffee house, cheap etc... *
*      e. how far to redeem coupon                                *
**********************************************************************
* A1. Found need to future filter multi-index for 0/1 for some    *
*     features                                                    *
**********************************************************************
* Findings: (for best acceptances)                                *
* Conditions:  Sunny, No Urgent destination, with passanger       *
*              afternoon to early evening                         *
* Background: Male Single, Female Single, 21-26, <= 1 child      *
* Habits: 1-3 Coffee House, 1-3 Bar visits                        *
**********************************************************************
```

In [ ]: