Will a Customer Accept the Coupon?

Context

Imagine driving through town and a coupon is delivered to your cell phone for a restaraunt near where you are driving. Would you accept that coupon and take a short detour to the restaraunt? Would you accept the coupon but use it on a sunbsequent trip? Would you ignore the coupon entirely? What if the coupon was for a bar instead of a restaraunt? What about a coffee house? Would you accept a bar coupon with a minor passenger in the car? What about if it was just you and your partner in the car? Would weather impact the rate of acceptance? What about the time of day?

Obviously, proximity to the business is a factor on whether the coupon is delivered to the driver or not, but what are the factors that determine whether a driver accepts the coupon once it is delivered to them? How would you determine whether a driver is likely to accept a coupon?

Overview

The goal of this project is to use what you know about visualizations and probability distributions to distinguish between customers who accepted a driving coupon versus those that did not.

Data

This data comes to us from the UCI Machine Learning repository and was collected via a survey on Amazon Mechanical Turk. The survey describes different driving scenarios including the destination, current time, weather, passenger, etc., and then ask the person whether he will accept the coupon if he is the driver. Answers that the user will drive there ‘right away’ or ‘later before the coupon expires’ are labeled as ‘Y = 1’ and answers ‘no, I do not want the coupon’ are labeled as ‘Y = 0’. There are five different types of coupons -- less expensive restaurants (under $20), coffee houses, carry out & take away, bar, and more expensive restaurants (\$20 - $50).

Deliverables

Your final product should be a brief report that highlights the differences between customers who did and did not accept the coupons. To explore the data you will utilize your knowledge of plotting, statistical summaries, and visualization using Python. You will publish your findings in a public facing github repository as your first portfolio piece.

Data Description

The attributes of this data set include:

1. User attributes
    - Gender: male, female
    - Age: below 21, 21 to 25, 26 to 30, etc.
    - Marital Status: single, married partner, unmarried partner, or widowed
    - Number of children: 0, 1, or more than 1
    - Education: high school, bachelors degree, associates degree, or graduate degree
    - Occupation: architecture & engineering, business & financial, etc.
    - Annual income: less than $12500, \$12500 - $24999, \$25000 - $37499, etc.
    - Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she buys takeaway food: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she goes to a coffee house: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she eats at a restaurant with average expense less than $20 per person: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8 or greater than 8

1. Contextual attributes
    - Driving destination: home, work, or no urgent destination

- Location of user, coupon and destination: we provide a map to show the geographical location of the user, destination, and the venue, and we mark the distance between each two places with time of driving. The user can see whether the venue is in the same direction as the destination.
- Weather: sunny, rainy, or snowy
- Temperature: 30F, 55F, or 80F
- Time: 10AM, 2PM, or 6PM
- Passenger: alone, partner, kid(s), or friend(s)

1. Coupon attributes
   - time before it expires: 2 hours or one day

In [1]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

Problems

Use the prompts below to get started with your data analysis.

1. Read in the `coupons.csv` file.

In [2]:
```
data = pd.read_csv('data/coupons.csv')
```

In [3]:
```
data.head(1000)
```

Out[3]:

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CoffeeHouse	CarryAway	RestaurantLessThan20	Restaurant20To50	toCoupon_GEQ
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	never	NaN	4~8	1~3	
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	never	NaN	4~8	1~3	
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	never	NaN	4~8	1~3	
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	never	NaN	4~8	1~3	
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	never	NaN	4~8	1~3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Female	31	Married partner	...	never	1~3	never	never	
996	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Female	31	Married partner	...	never	1~3	never	never	
997	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Female	31	Married partner	...	never	1~3	never	never	
998	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	31	Married partner	...	never	1~3	never	never	
999	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Female	31	Married partner	...	never	1~3	never	never	

1000 rows × 26 columns

1. Investigate the dataset for missing or problematic data.

In [4]:

```
data.isnull().sum()
```

Out[4]:
```
destination                0
passanger                  0
weather                    0
temperature                0
time                       0
coupon                     0
expiration                 0
gender                     0
age                        0
maritalStatus              0
has_children               0
education                  0
occupation                 0
income                     0
car                    12576
Bar                      107
CoffeeHouse              217
CarryAway                151
RestaurantLessThan20     130
Restaurant20To50         189
toCoupon_GEQ5min           0
toCoupon_GEQ15min          0
toCoupon_GEQ25min          0
direction_same             0
direction_opp              0
Y                          0
dtype: int64
```

1. Decide what to do about your missing data -- drop, replace, other...

In [5]:
```
data["CarryAway"].fillna("never", inplace=True)
data["Bar"].fillna("never", inplace=True)
data["RestaurantLessThan20"].fillna("never", inplace=True)
data["Restaurant20To50"].fillna("never", inplace=True)
data["CoffeeHouse"].fillna("never", inplace=True)
data["car"].fillna("do not drive", inplace=True)
```

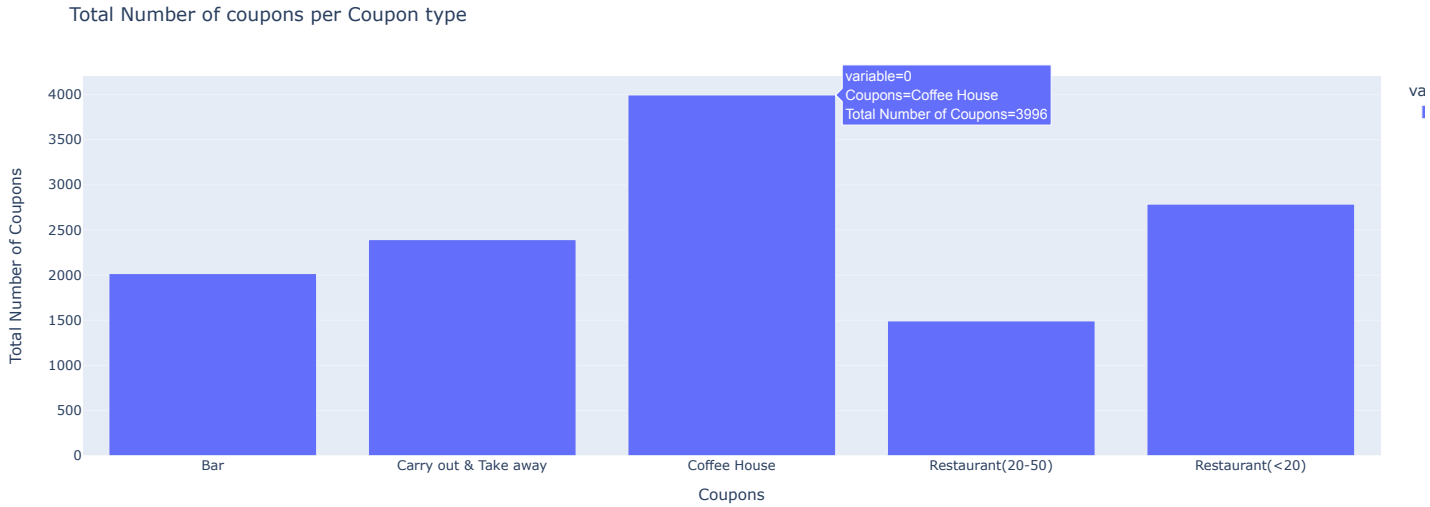
1. What proportion of the total observations chose to accept the coupon?

In [6]:
```
data.query('Y == 1').shape[0]/data.shape[0]
```

Out[6]:
```
0.5684326710816777
```

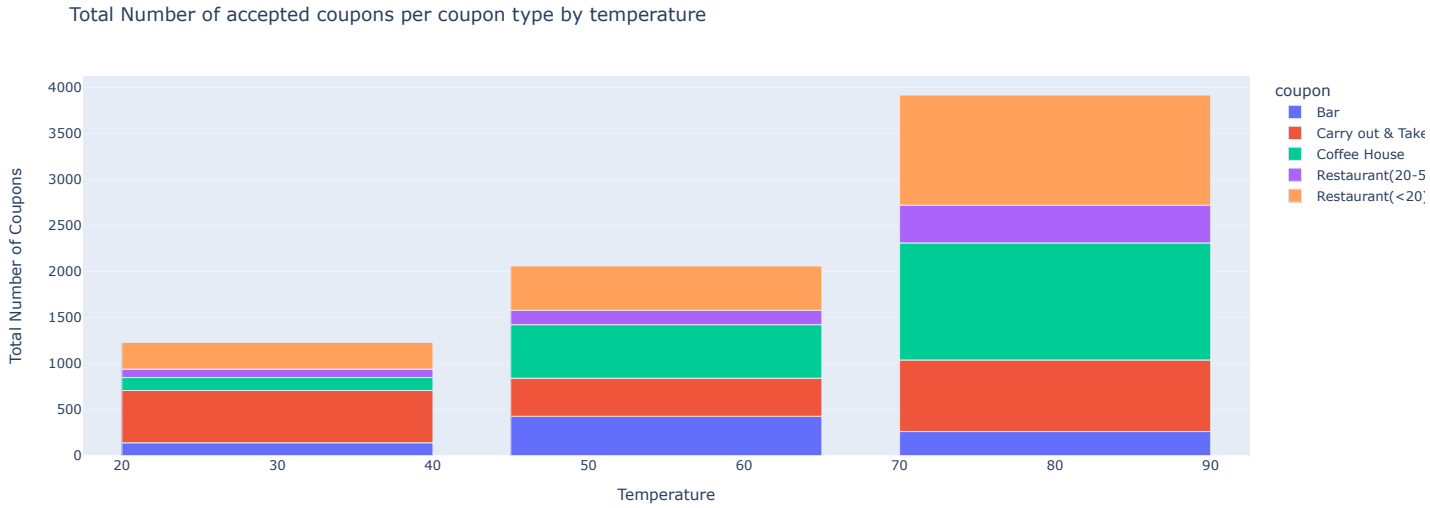
1. Use a bar plot to visualize the `coupon` column.

In [7]:
```
ds = data.groupby('coupon').size();
ds.columns = {'Coupons', 'total count'}
px.bar(ds, labels={'coupon': 'Coupons', 'value':'Total Number of Coupons'}, title="Total Number of coupons per Coupon type")
```

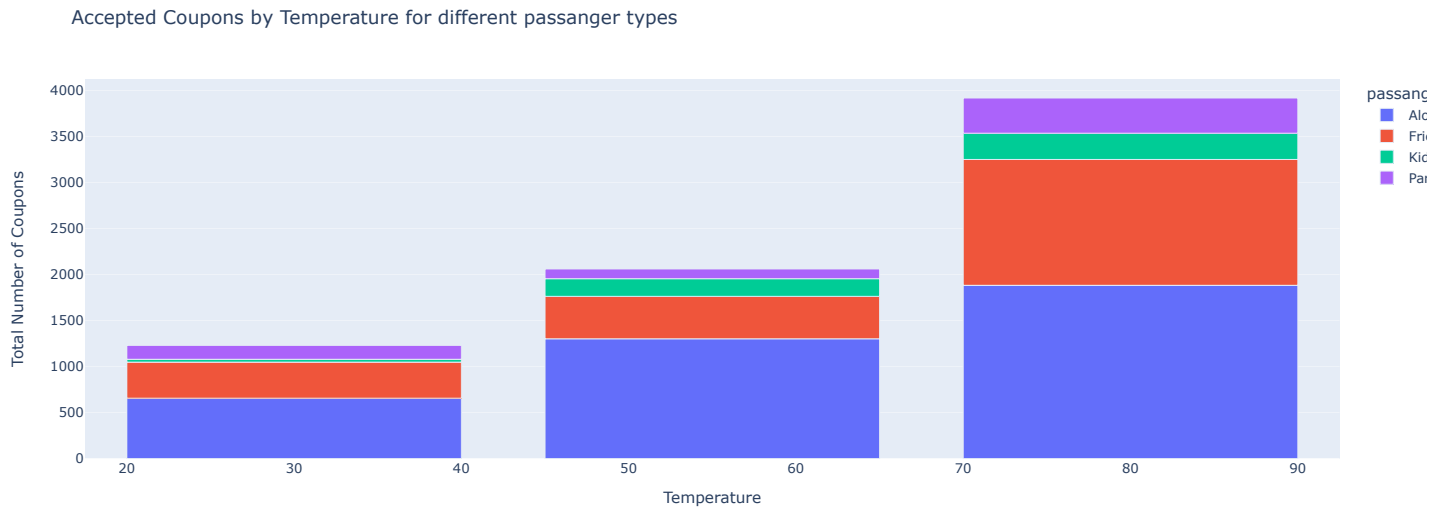


1. Use a histogram to visualize the temperature column.

In [8]:
```
dm = data.groupby(['temperature', 'coupon']).sum()
dm = dm.reset_index()
px.bar(dm, x='temperature', y='Y', color='coupon', labels={'temperature': 'Temperature', 'Y':'Total Number of Coupons'}, title="Total Number of accepted coupons per cou
```

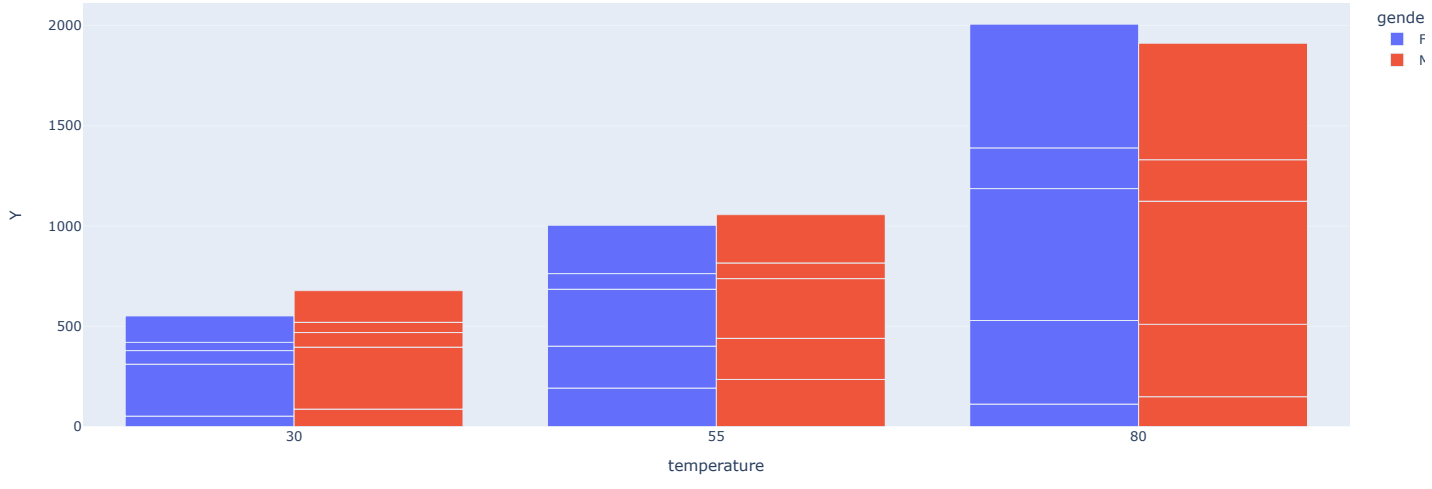


In [9]:
```
dl = data.groupby(['temperature', 'passanger']).sum()
dl = dl.reset_index()
px.bar(dl, x='temperature', y='Y', color='passanger', labels={'temperature': 'Temperature', 'Y':'Total Number of Coupons'}, title="Accepted Coupons by Temperature for d
```



In [10]:

```
grouped_df = data.groupby(['temperature', 'gender', 'coupon'], as_index="false").agg(
    {"Y": "sum"}
)
grouped_df.reset_index()
fig = px.bar(
    data_frame=grouped_df.reset_index(),
    x='temperature',
    y='Y',
    color='gender',
    barmode="group"
)
fig.show()
```



In [ ]:

Investigating the Bar Coupons

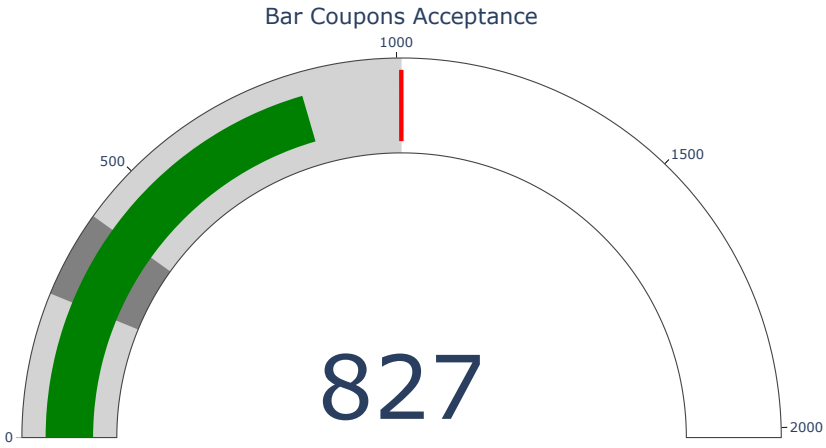
Now, we will lead you through an exploration of just the bar related coupons.

1. Create a new `DataFrame` that contains just the bar coupons.

In [11]:
```
dBar = data.query('coupon == "Bar"')
```

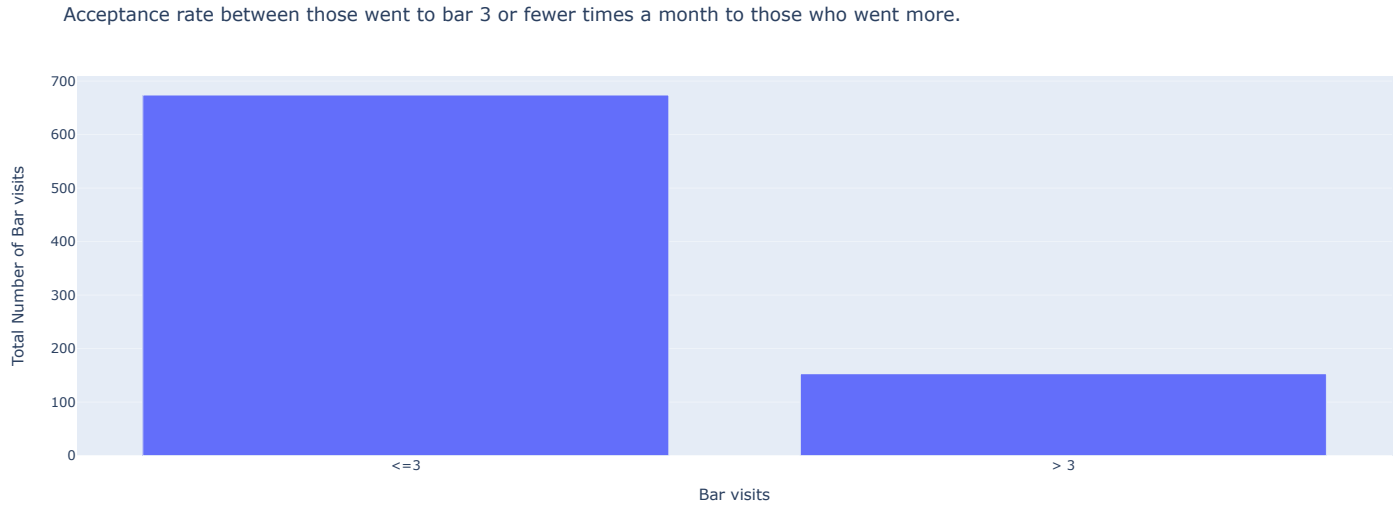
1. What proportion of bar coupons were accepted?

In [12]:
```
fig = go.Figure(go.Indicator(
    domain = {'x': [0, 1], 'y': [0, 1]},
    value = dBar.query('Y == 1').shape[0],
    mode = "gauge+number",
    title = {'text': "Bar Coupons Acceptance"},
    gauge = {'axis': {'range': [None, (dBar.shape[0])]},
             'steps' : [
                 {'range': [0, dBar.shape[0]/2], 'color': "lightgray"},
                 {'range': [250, 400], 'color': "gray"}],
             'threshold' : {'line': {'color': "red", 'width': 4}, 'thickness': 0.75, 'value': 1008}}))

fig.show()
```



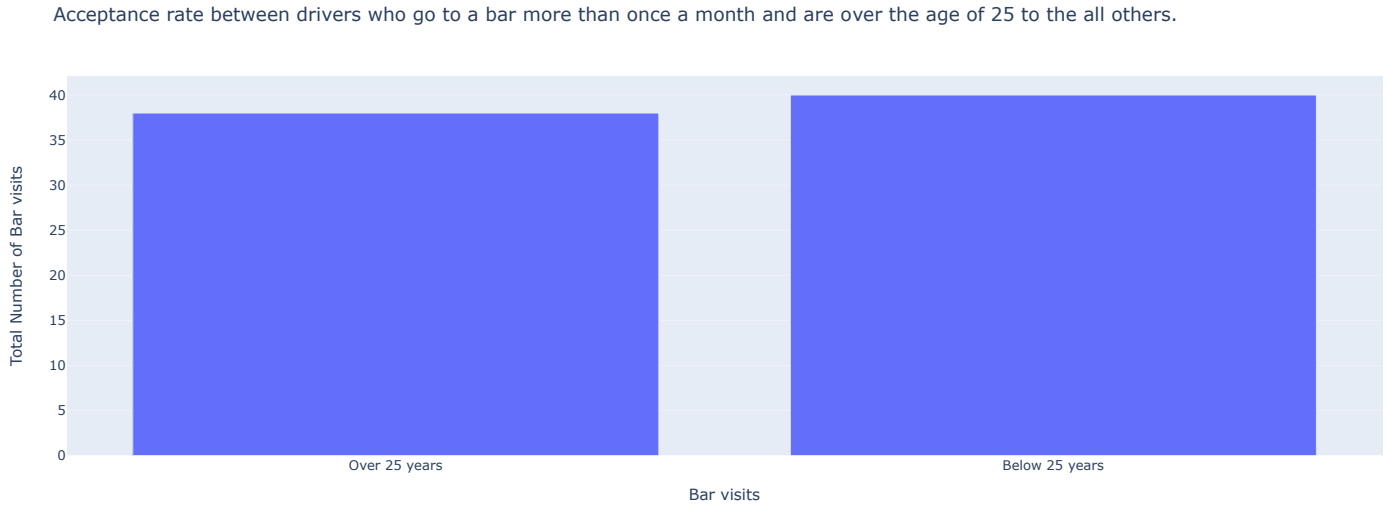
1. Compare the acceptance rate between those who went to a bar 3 or fewer times a month to those who went more.

In [13]:
```
dbar_totalCounts = dBar[['Bar', 'Y']].query('Y == 1').value_counts()
fewerthan3 = (dbar_totalCounts[0] + dbar_totalCounts[1] + dbar_totalCounts[2])
greaterthan3 = (dbar_totalCounts[3] + dbar_totalCounts[4])
x = [' <=3 ', ' > 3 ']
y = [fewerthan3, greaterthan3]
px.bar(x=x, y=y, labels={'x': 'Bar visits', 'y':'Total Number of Bar visits'}, title="Acceptance rate between those went to bar 3 or fewer times a month to those who we
```



1. Compare the acceptance rate between drivers who go to a bar more than once a month and are over the age of 25 to the all others. Is there a difference?

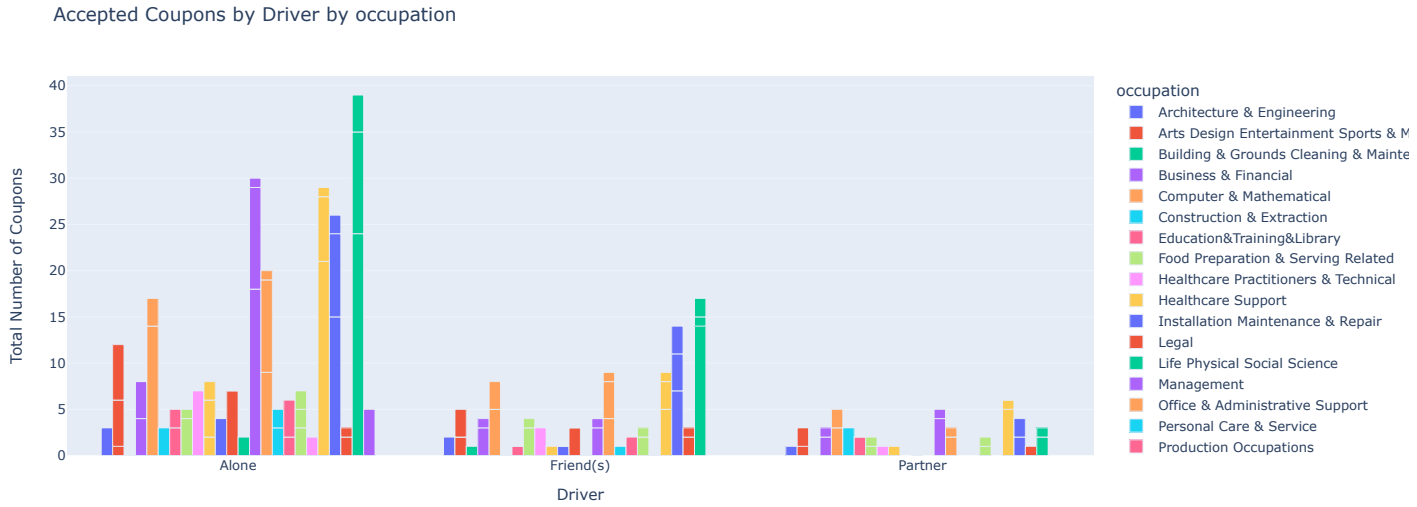
In [81]:
```
#data.groupby('age').sum()
dbar_over25_acceptance = dBar[['Bar', 'Y', 'age']].query('Y == 1 & age != "21" & age != "below21"').agg(
    {"Y": "sum"}
)

dbar_over25_all = (dBar[['Bar', 'Y', 'age']].query('age != "21" & age != "below21"')).shape[0]


dbar_less25_acceptance = dBar[['Bar', 'Y', 'age']].query('Y == 1 & (age == "21" | age != "below21")').agg(
    {"Y": "sum"}
)

dbar_less25_all = dBar[['Bar', 'Y', 'age']].query('(age == "21" | age != "below21")').shape[0]

dbar_over25_rate = dbar_over25_acceptance/dbar_over25_all
dbar_below25_rate = dbar_less25_acceptance/dbar_less25_all
#dbar_over25_rate -   dbar_below25_rate

xaxis = [ 'Over 25 years', 'Below 25 years' ]
yaxis = [int(dbar_over25_rate * 100), int(dbar_below25_rate * 100)]

px.bar(x=xaxis, y=yaxis,labels={'x': 'Bar visits', 'y':'Total Number of Bar visits'}, title="Acceptance rate between drivers who go to a bar more than once a month and
```



1. Use the same process to compare the acceptance rate between drivers who go to bars more than once a month and had passengers that were not a kid and had occupations other than farming, fishing, or forestry.

In [117…

```
dpassKid = dBar.query('passanger != "Kid(s)" & occupation != "farming" & occupation != "fishing" & occupation != "forestry"')
#More than one month
dpassKid = dpassKid.query('Bar != "never" & Bar != "less1"')
grouped_df2 = dpassKid.groupby(['Bar', 'passanger', 'occupation'], as_index="false").agg(
    {"Y": "sum"}
)
grouped_df2.reset_index()
fig = px.bar(
    data_frame=grouped_df2.reset_index(),
    x='passanger',
    y='Y',
    color='occupation',
    barmode="group",
    labels={'passanger': 'Driver', 'Y':'Total Number of Coupons'},
    title="Accepted Coupons by Driver by occupation"
)
fig.show()
```



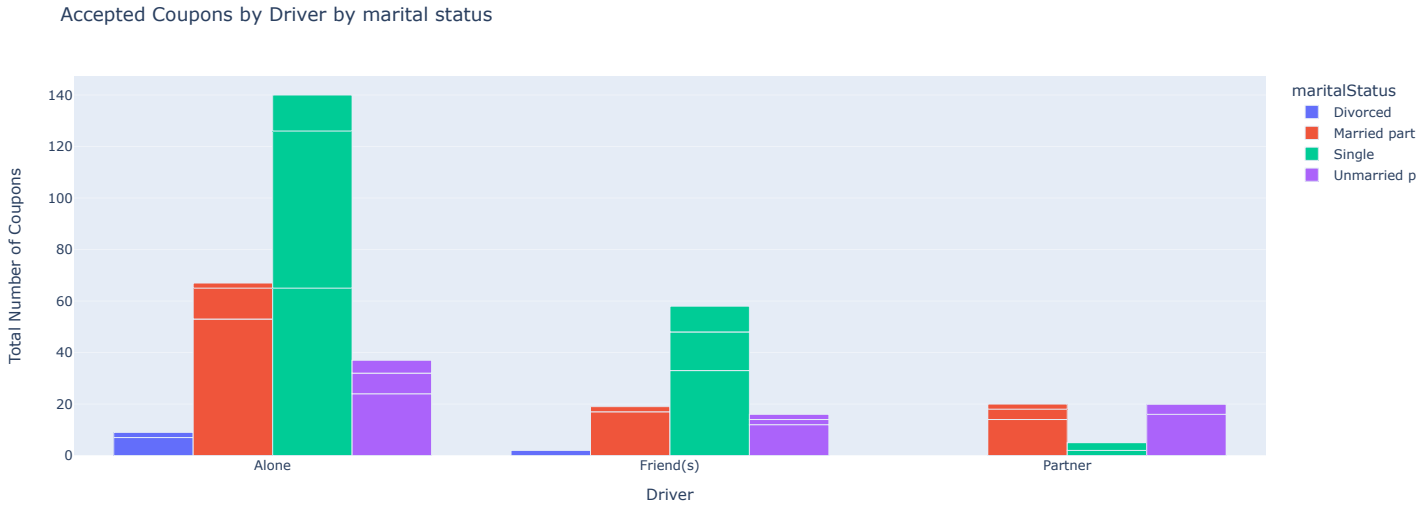
1. Compare the acceptance rates between those drivers who:

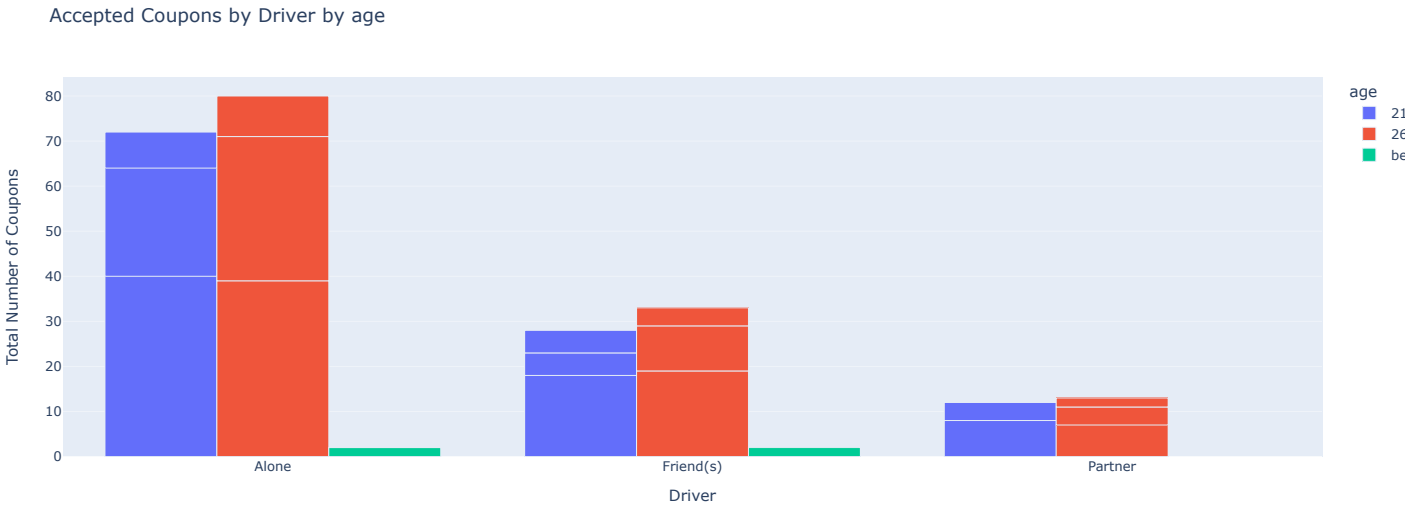
- go to bars more than once a month, had passengers that were not a kid, and were not widowed *OR*
- go to bars more than once a month and are under the age of 30 *OR*
- go to cheap restaurants more than 4 times a month and income is less than 50K.

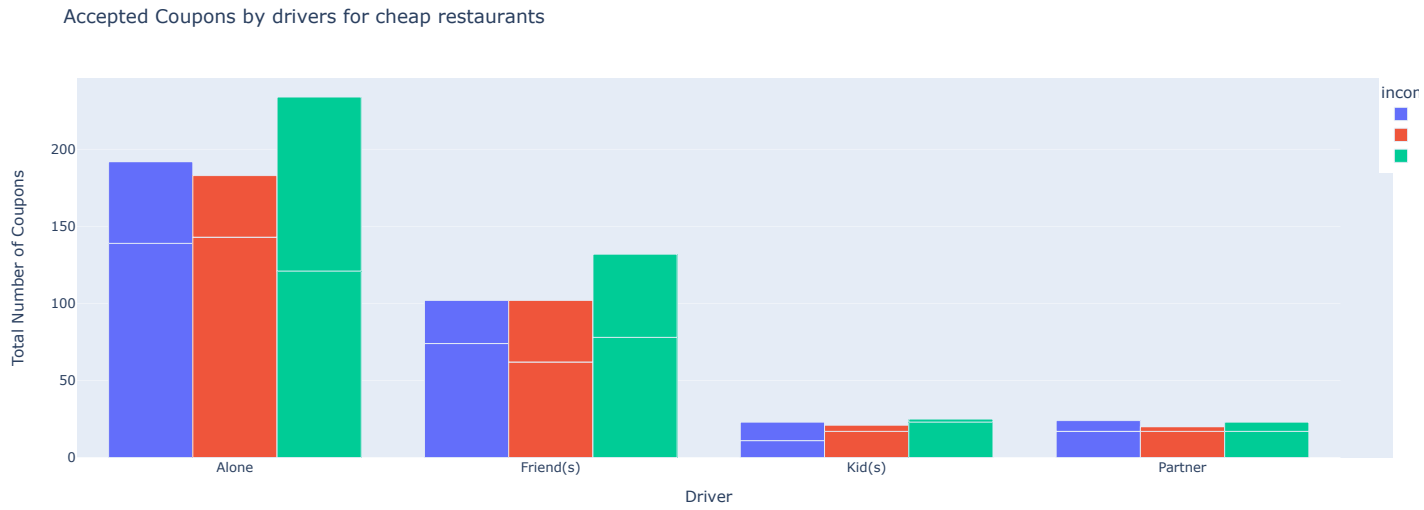
In [118…

```
df1 = dBar.query('Bar != "never" & Bar != "less1" & passanger != "Kid(s)" & maritalStatus != "widowed"')
df2 = dBar.query('Bar != "never" & Bar != "less1" & passanger != "Kid(s)" & (age == "below21" | age == "21" | age == "26")')
df3 = data.query('(RestaurantLessThan20 == "4~8" | RestaurantLessThan20 == "gt8") & (income == "$12500 - $24999"|income == "$25000 - $37499"|income == "$37500 - $49999"


grouped_df3 = df1.groupby(['Bar', 'passanger', 'maritalStatus'], as_index="false").agg(
    {"Y": "sum"}
)
grouped_df3.reset_index()
fig1 = px.bar(
    data_frame=grouped_df3.reset_index(),
    x='passanger',
    y='Y',
    color='maritalStatus',
    barmode="group",
    labels={'passanger': 'Driver', 'Y':'Total Number of Coupons'},
    title="Accepted Coupons by Driver by marital status"
)
fig1.show()
```

```
grouped_df4 = df2.groupby(['Bar', 'passanger', 'age'], as_index="false").agg(
    {"Y": "sum"}
)
grouped_df4.reset_index()
fig2 = px.bar(
    data_frame=grouped_df4.reset_index(),
    x='passanger',
    y='Y',
    color='age',
    barmode="group",
    labels={'passanger': 'Driver', 'Y':'Total Number of Coupons'},
    title="Accepted Coupons by Driver by age"
)
fig2.show()

grouped_df5 = df3.groupby(['RestaurantLessThan20', 'passanger', 'income'], as_index="false").agg(
    {"Y": "sum"}
)


grouped_df5.reset_index()
fig3 = px.bar(
    data_frame=grouped_df5.reset_index(),
    x='passanger',
    y='Y',
    color='income',
    barmode="group",
    labels={'passanger': 'Driver', 'Y':'Total Number of Coupons'},
    title="Accepted Coupons by drivers for cheap restaurants"
)
fig3.show()
```


Accepted Coupons by Driver by marital status
Total Number of Coupons
0
20
40
60
80
100
120
140
Alone
Friend(s)
Partner
Driver
maritalStatus
Divorced
Married part
Single
Unmarried p


Accepted Coupons by Driver by age
Total Number of Coupons
0
10
20
30
40
50
60
70
80
Alone
Friend(s)
Partner
Driver
age
21
26
be



1. Based on these observations, what do you hypothesize about drivers who accepted the bar coupons?

In [ ]:

Independent Investigation

Using the bar coupon example as motivation, you are to explore one of the other coupon groups and try to determine the characteristics of passengers who accept the coupons.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: