

Will a Customer Accept the Coupon?

Context

Imagine driving through town and a coupon is delivered to your cell phone for a restaurant near where you are driving. Would you accept that coupon and take a short detour to the restaurant? Would you accept the coupon but use it on a subsequent trip? Would you ignore the coupon entirely? What if the coupon was for a bar instead of a restaurant? 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.

[illegible]

995	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Female	31
996	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Female	31
997	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Female	31
998	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	31
999	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Female	31

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

```
dfRejection = (data.shape[0] - data.query('Y == 1').shape[0]) / data.shape[0]
dfData = [ [dfAcceptance, 'Acceptance'], [dfRejection, 'Rejection']]
df = pd.DataFrame(dfData, columns=['Acceptance Rate', 'Acceptance'])
df = df.set_index('Acceptance')

px.bar(df, labels={'index': 'driver', 'value': 'Acceptance rate of coupons'}, title="Prop
```

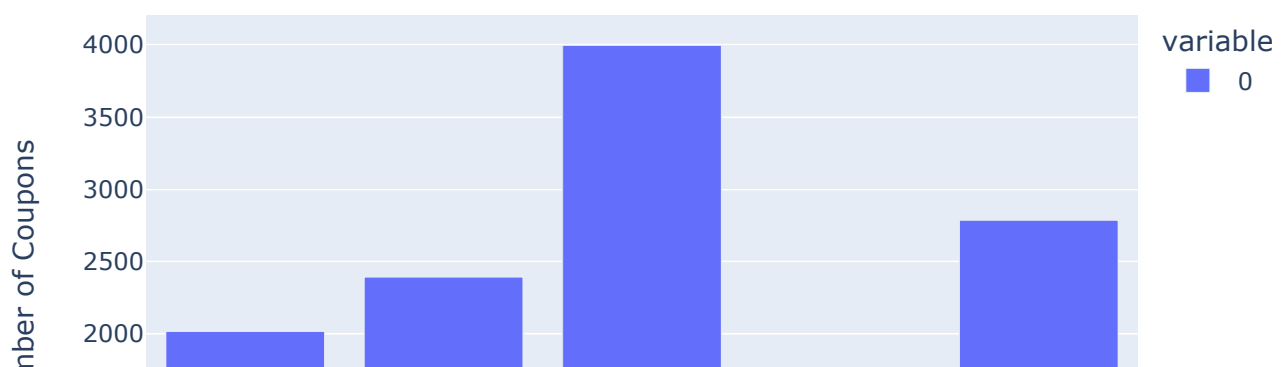
Proportion of total observations Accept or Reject a coupon

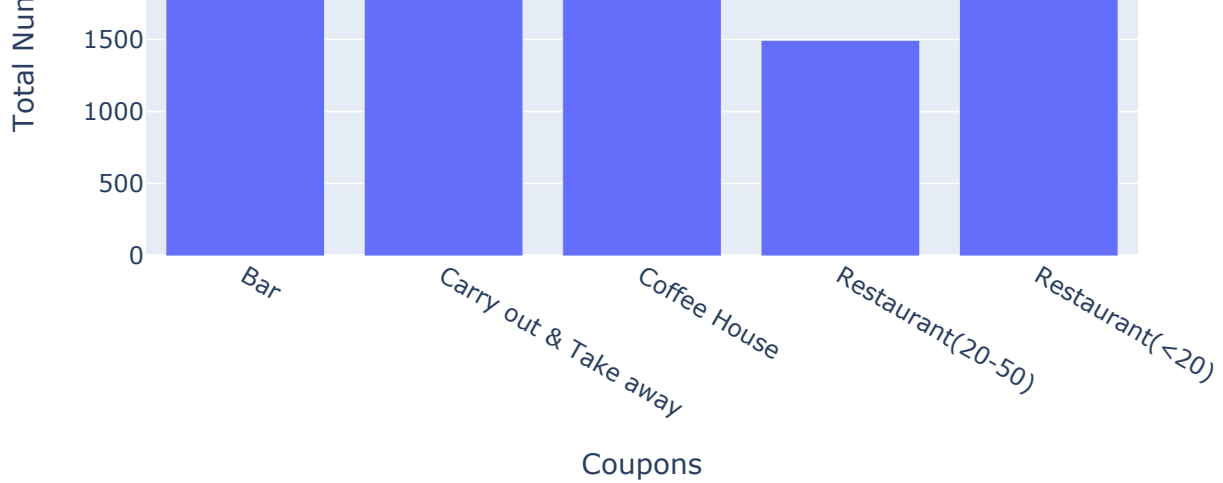


1. Use a bar plot to visualize the `coupon` column.

```
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
```

Total Number of coupons per Coupon type



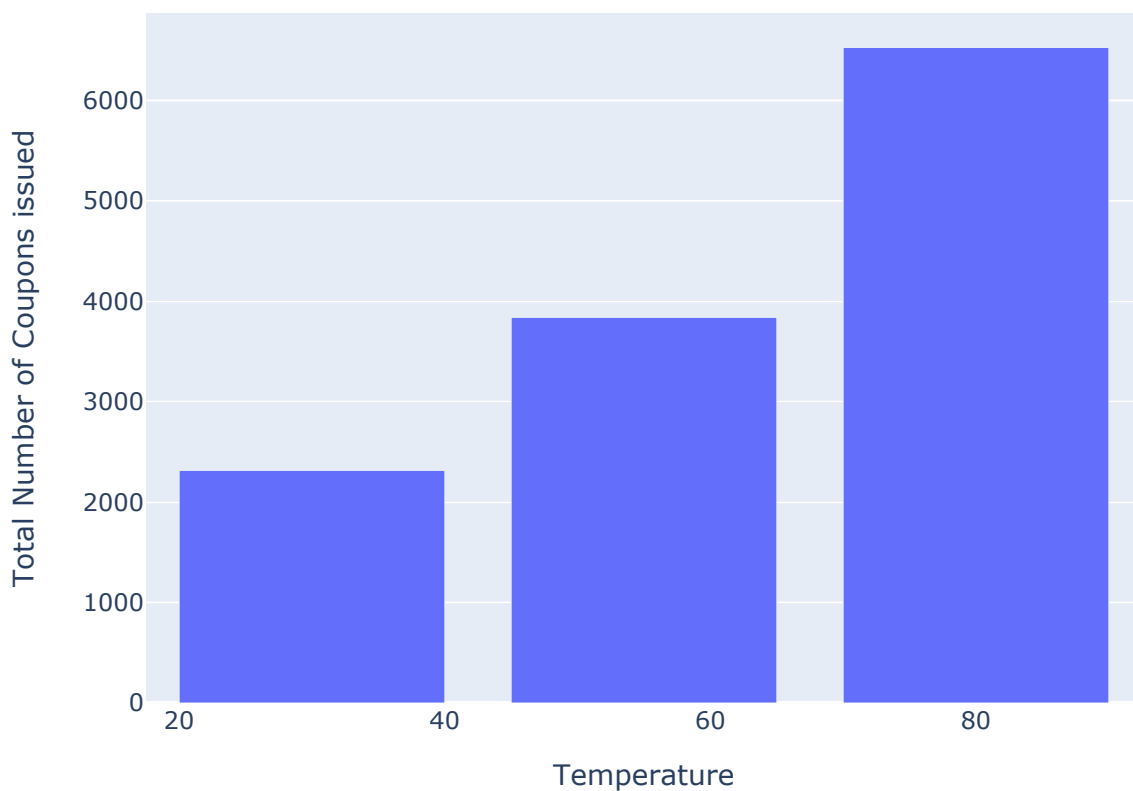


1. Use a histogram to visualize the temperature column.

```
In [8]: dm = data.groupby(['temperature']).sum()
dm = dm.reset_index()

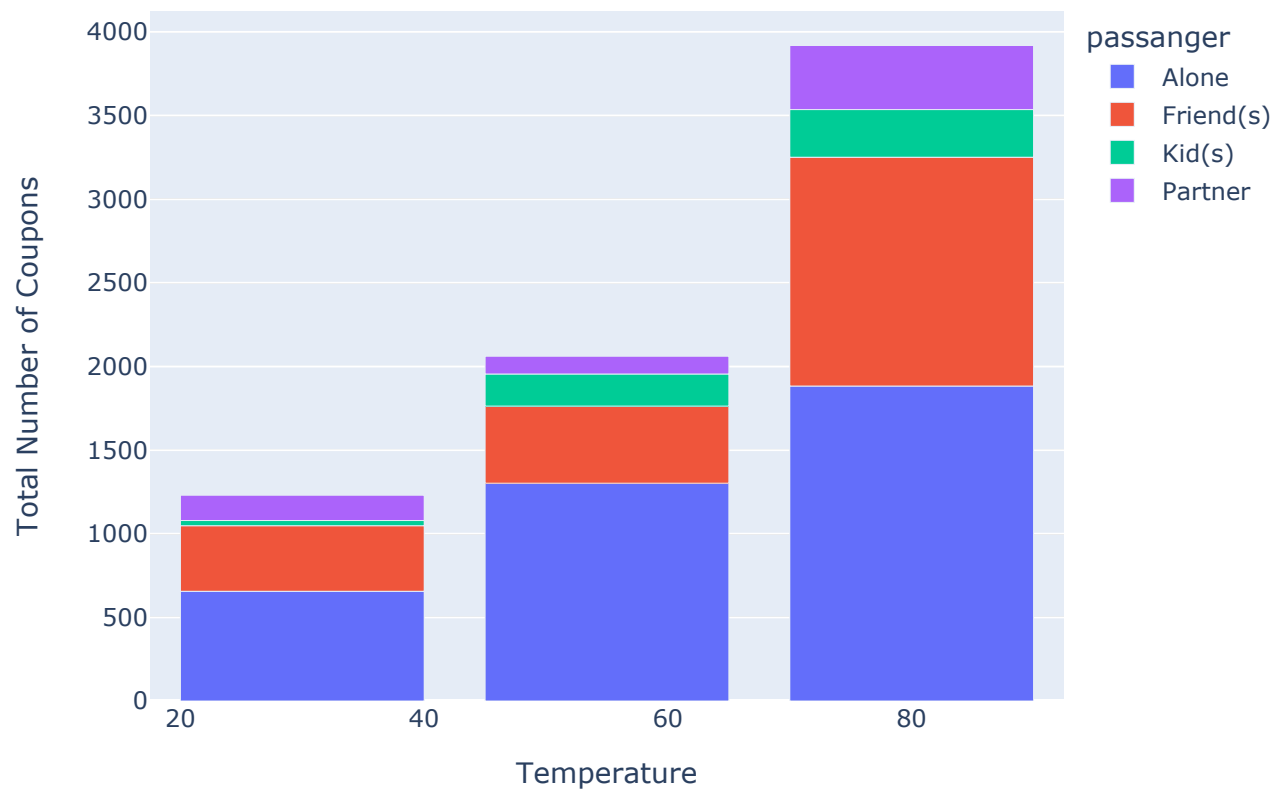
temp = data.groupby('temperature').size()
px.bar(dm, x='temperature', y=temp, labels={'temperature': 'Temperature', 'y': 'Total Num
```

Histogram by temperature

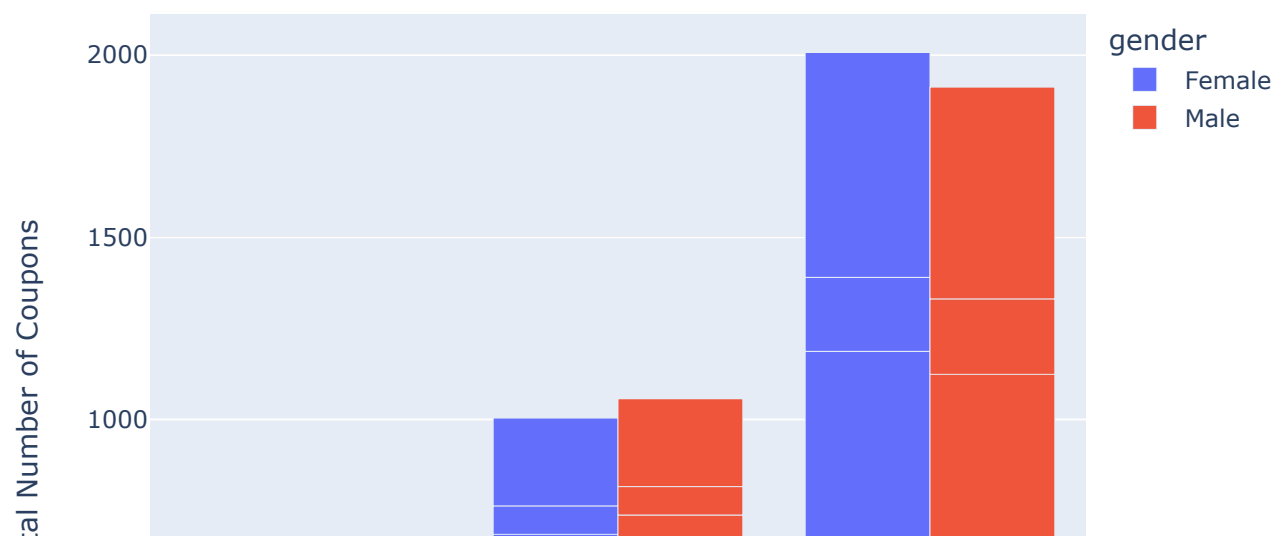


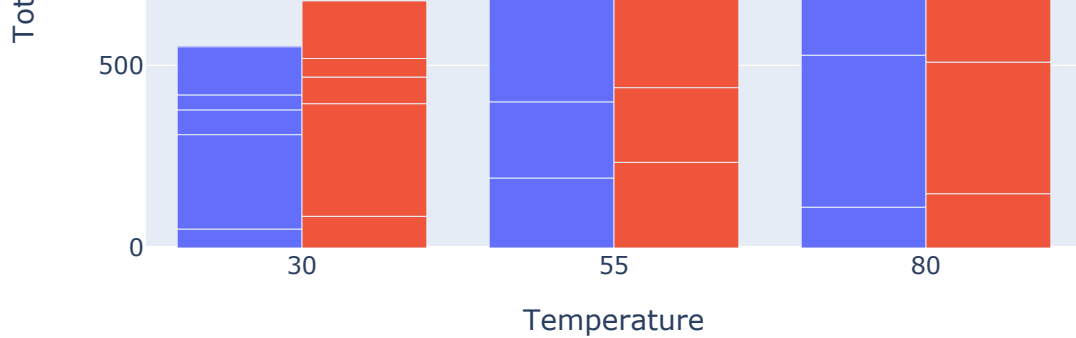
```
In [9]: dl = data.groupby(['temperature', 'passanger']).sum()
dl = dl.reset_index()
px.bar(dl, x='temperature', y='Y', color='passanger', labels={'temperature': 'Temperatur
```

AcceptedCoupons by Temperature for different passanger types



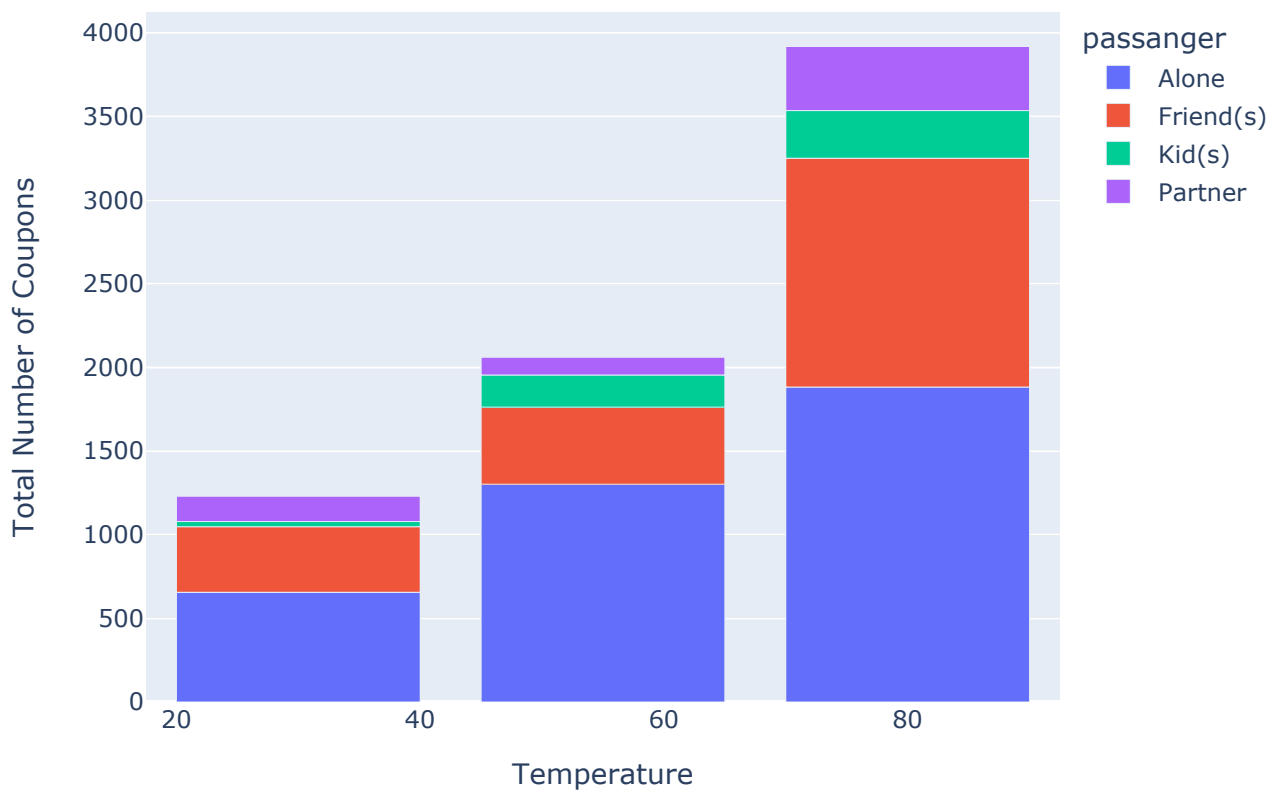
```
In [10]: grouped_df = data.groupby(['temperature', 'gender', 'coupon'], as_index=False).agg(
        {"Y": "sum"}
    )
    grouped_df.reset_index()
    fig = px.bar(
        data_frame=grouped_df.reset_index(),
        x='temperature',
        y='Y',
        color='gender',
        barmode="group",
        labels={'temperature': 'Temperature', 'Y': 'Total Number of Coupons'}
    )
    fig.show()
```





```
In [11]: dl = data.groupby(['temperature', 'passanger']).sum()
dl = dl.reset_index()
px.bar(dl, x='temperature', y='Y', color='passanger', labels={'temperature': 'Temperatur
```

Accepted Coupons by Temperature for different passanger types



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

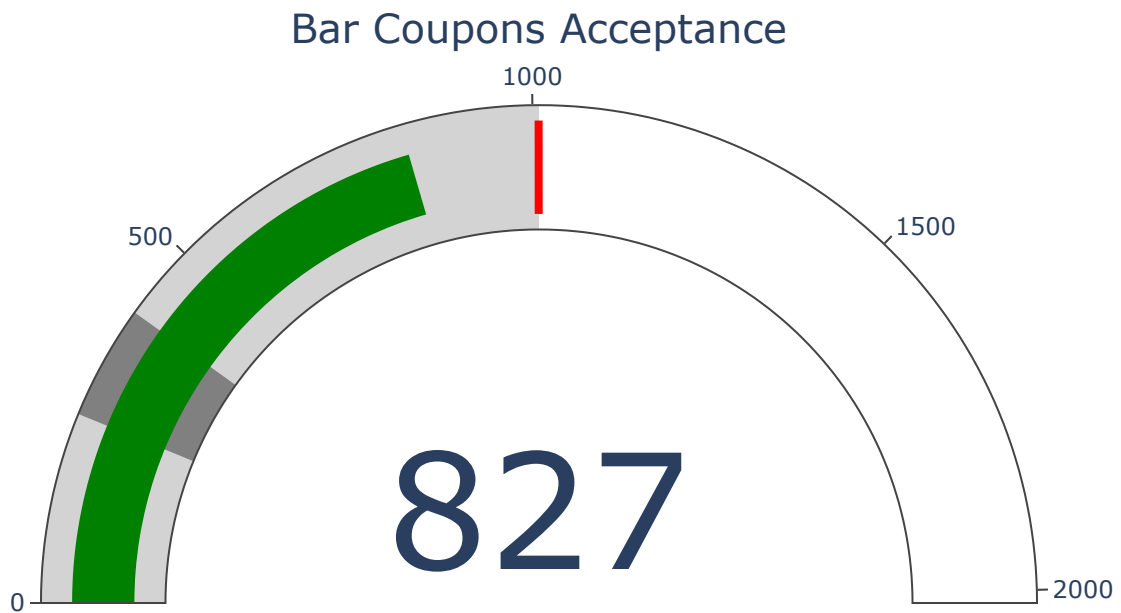
1. What proportion of bar coupons were accepted?

```
In [13]: 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, 'va
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.

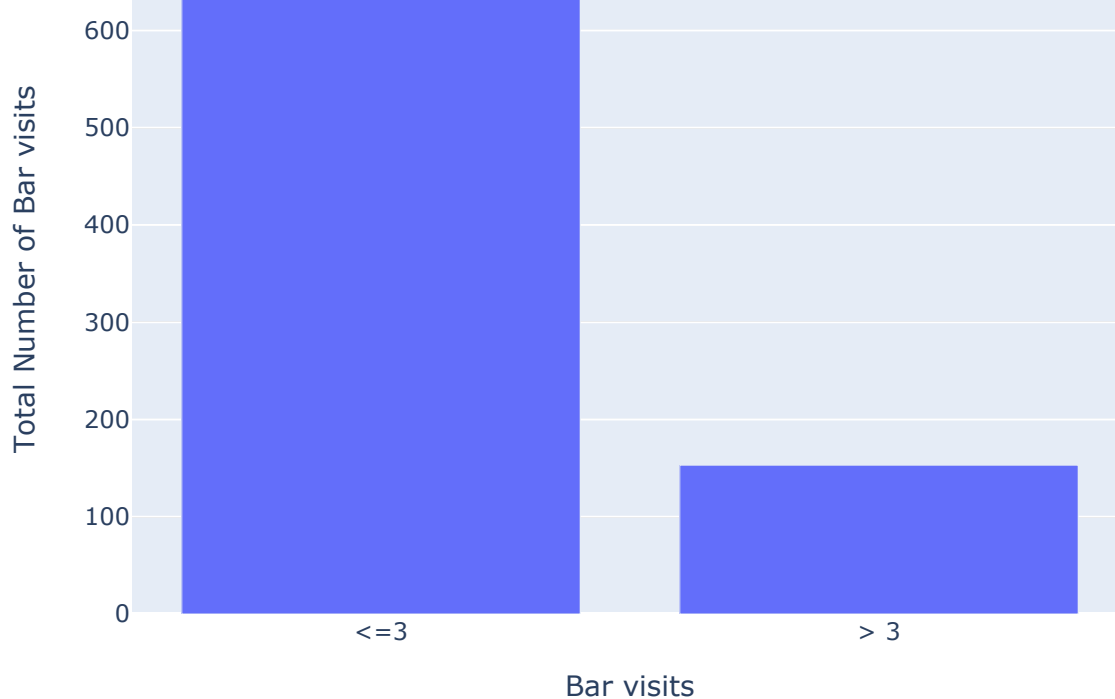
```

In [14]: 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="Ac

```

Acceptance rate between those went to bar 3 or fewer times a month to

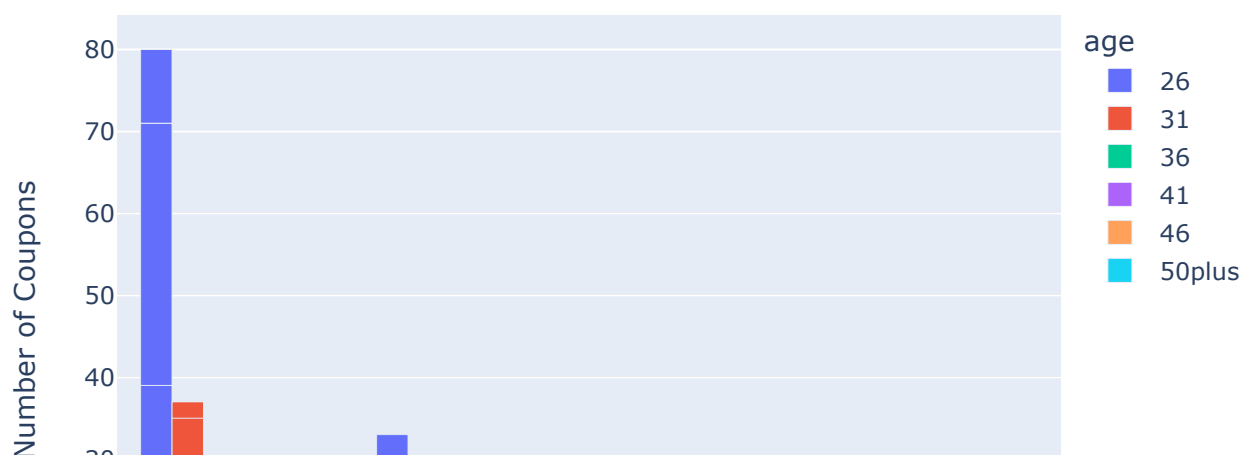


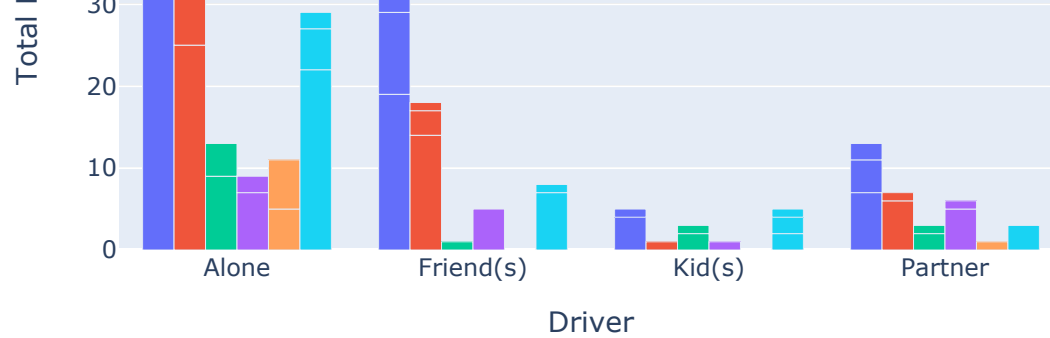


1. Compare the acceptance rate between drivers who go to a bar more than once a month and are over the age of 25 to the all others. Is there a difference?

```
In [83]: dbar_over25_acceptance = dBar.query('Y == 1 & age != "21" & age != "below21" & Bar != "1"')
dbar_over25_shape = dbar_over25_acceptance.shape[0]
dbar_over25_acceptance.groupby('age').sum()
dbar_over25_acceptance = dbar_over25_acceptance.groupby(['Bar', 'age', 'passanger'], as_index=False).sum()
dbar_over25_acceptance.reset_index()
fig = px.bar(
    data_frame=dbar_over25_acceptance.reset_index(),
    x='passanger',
    y='Y',
    color='age',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Coupon Acceptance difference between drivers who go to a bar more than once a month and are over the age of 25 to the all others."
)
fig.show()
```

Coupon Acceptance difference between drivers who go to a bar more than once a month and are over the age of 25 to the all others.





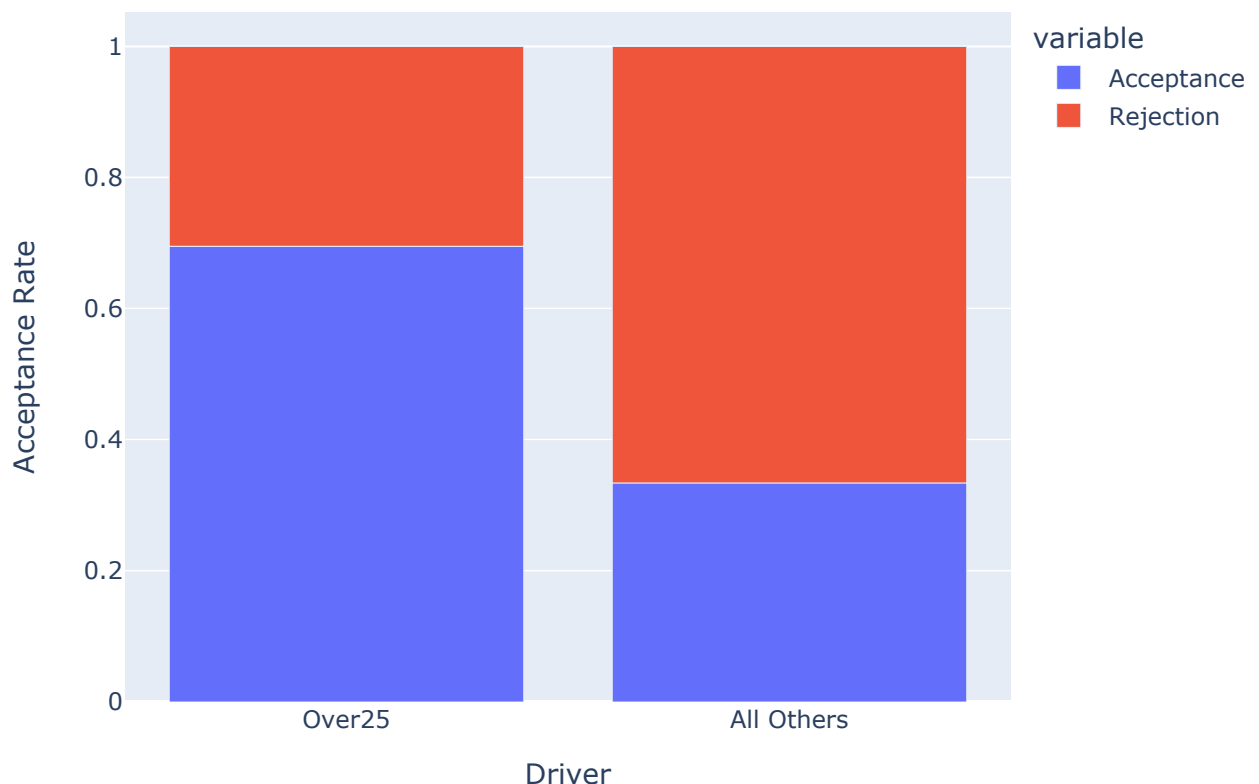
```
In [73]: #dbar_over25_acceptance['Y'].sum()
dbar_over25_allRows = dBar.query('age != "21" & age != "below21" & Bar != "less1" & Bar
dbar_over25_allRows.shape[0]
dbarOver25Acceptance = dbar_over25_allRows['Y'].sum()/dbar_over25_allRows.shape[0]
dbarOver25Rejection = (dbar_over25_allRows.shape[0] - dbar_over25_allRows['Y'].sum())/db

dbar_notover25_allRows = pd.concat([dBar,dbar_over25_allRows]).drop_duplicates(keep=False)
dbar_notover25_Acceptance = dbar_notover25_allRows['Y'].sum()/dbar_notover25_allRows.sha
dbarnotover25_Rejection = (dbar_notover25_allRows.shape[0] - dbar_notover25_allRows['Y']

dfData = [ [dbarOver25Acceptance,dbarOver25Rejection,'Over25' ], [dbar_notover25_Accepta
df = pd.DataFrame(dfData, columns=['Acceptance', 'Rejection', 'Driver'])
df = df.set_index('Driver')

px.bar(df, labels={'Driver': 'Driver', 'value':'Acceptance Rate'}, title="Acceptance rat
```

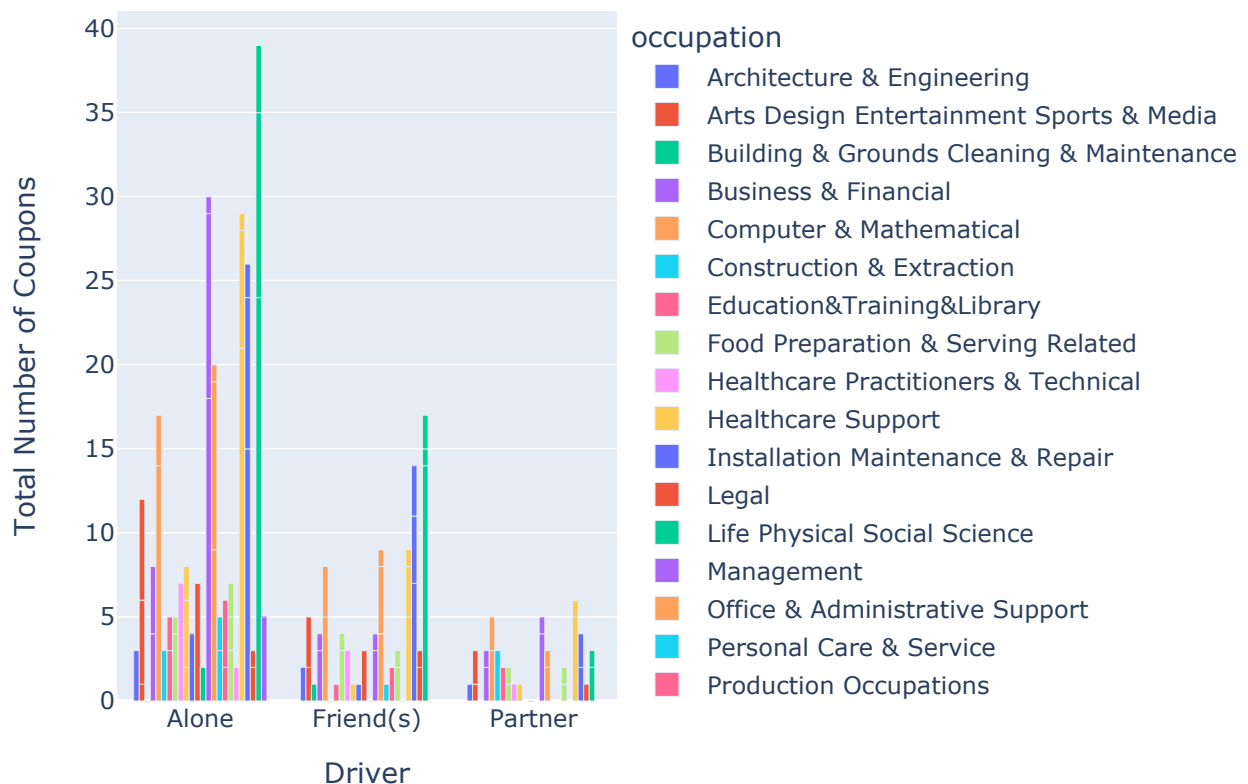
Acceptance rate between those who go to a bar more than once a month to the all others



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 [16]: dpassKid = dBar.query(' Bar != "less1" & Bar != "never" & passanger != "Kid(s)" & occupa
#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 and occupation"
)
fig.show()
```

Accepted Coupons by Driver and occupation



```
In [75]: dpasskid_allRows = dpassKid
dpasskidAcceptance = dpasskid_allRows['Y'].sum()/dpasskid_allRows.shape[0]
dpasskidRejection = (dpasskid_allRows.shape[0] - dpasskid_allRows['Y'].sum())/dpasskid_a

NotdPassKid_allRows = pd.concat([dBar, dpasskid_allRows]).drop_duplicates(keep=False)
NotPassKid_Acceptance = NotdPassKid_allRows['Y'].sum()/NotdPassKid_allRows.shape[0]
NotPassKid_Rejection = (NotdPassKid_allRows.shape[0] - NotdPassKid_allRows['Y'].sum())/N
```



```

        title="Accepted Coupons by Driver and 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 and age"
)
fig2.show()

grouped_df5 = df3.groupby(['RestaurantLessThan20', 'passanger', 'income'], as_index="fal
    {"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()

df1_allRows = df1
df1_Acceptance = df1_allRows['Y'].sum()/df1_allRows.shape[0]
df1_Rejection = (df1_allRows.shape[0] - df1_allRows['Y'].sum())/df1_allRows.shape[0]

#df2_allRows = df2
df2_allRows = df2
df2_Acceptance = df2_allRows['Y'].sum()/df2_allRows.shape[0]
df2_Rejection = (df2_allRows.shape[0] - df2_allRows['Y'].sum())/df2_allRows.shape[0]

#df3_allRows = df3
df3_allRows = df3
df3_Acceptance = df3_allRows['Y'].sum()/df3_allRows.shape[0]
df3_Rejection = (df3_allRows.shape[0] - df3_allRows['Y'].sum())/df3_allRows.shape[0]

dfdata1 = [[df1_Acceptance, df1_Rejection, 'Marital Status'], [df2_Acceptance, df2_Rejecti
df = pd.DataFrame(dfdata1, columns=['Acceptance', 'Rejection', 'Driver'])
df = df.set_index('Driver')

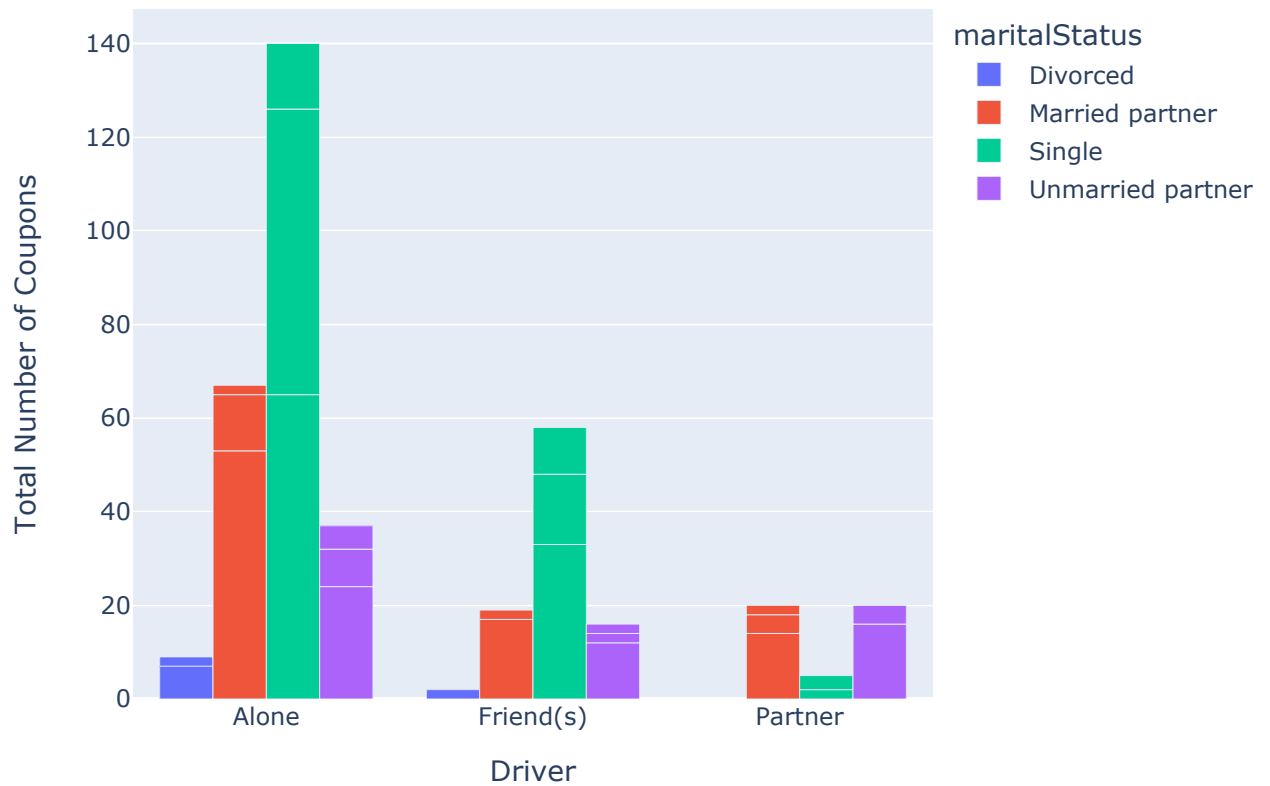
title1 = "Acceptance Rate between drivers"

fig = px.bar(df, labels={'Driver': 'Driver', 'value': 'Acceptance Rate'},
    title=title1)

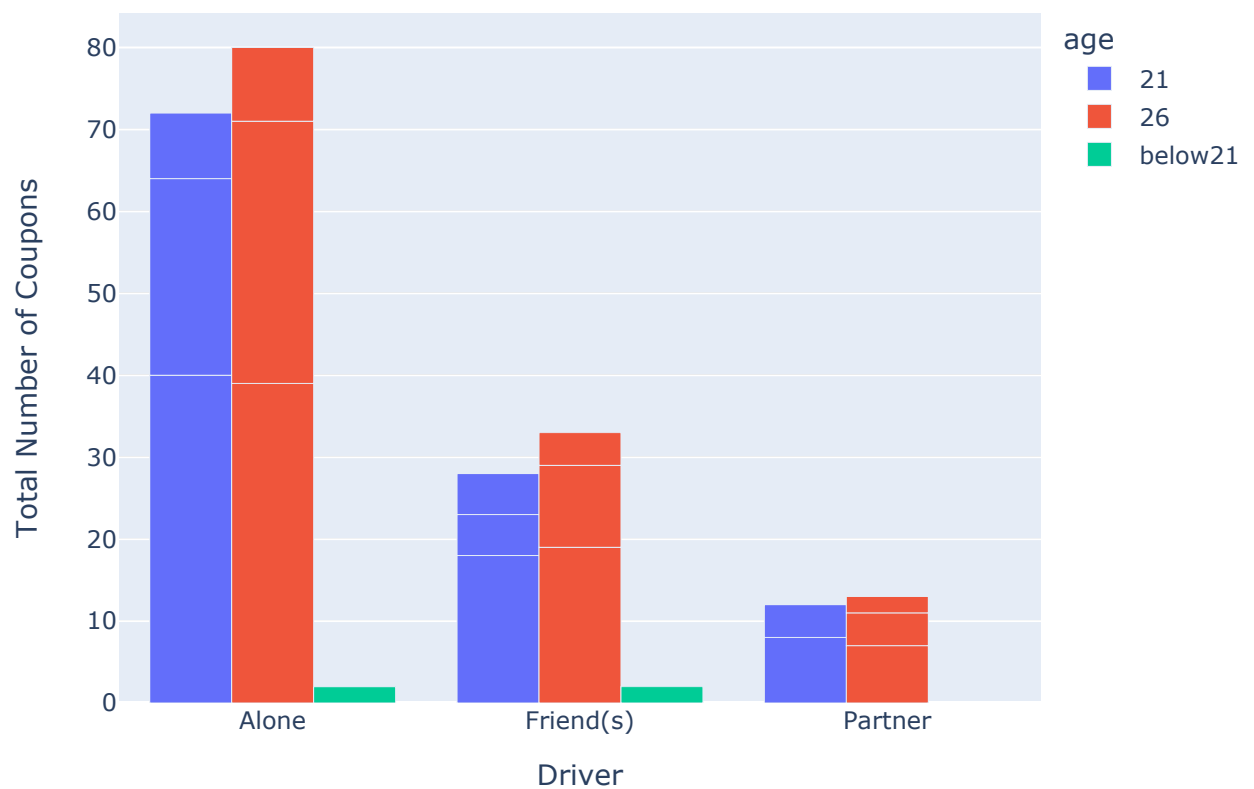
fig.show()

```

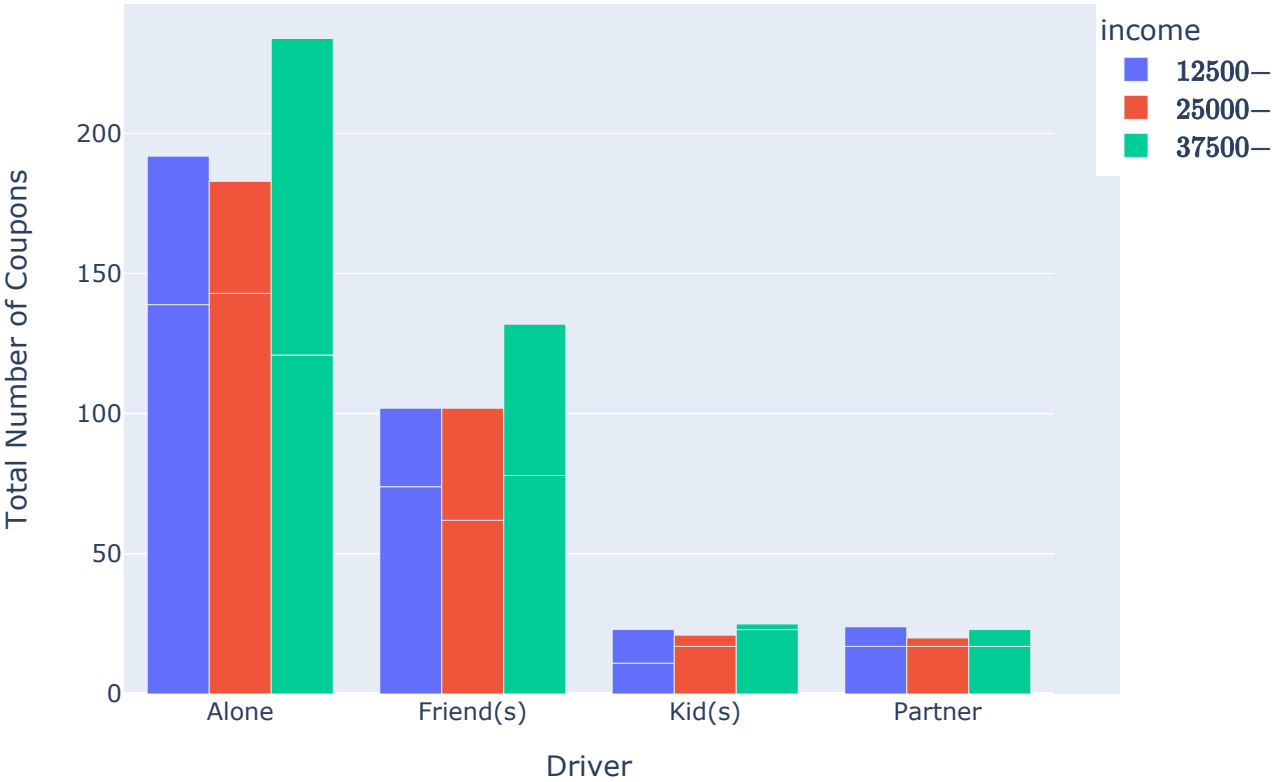
Accepted Coupons by Driver and marital status



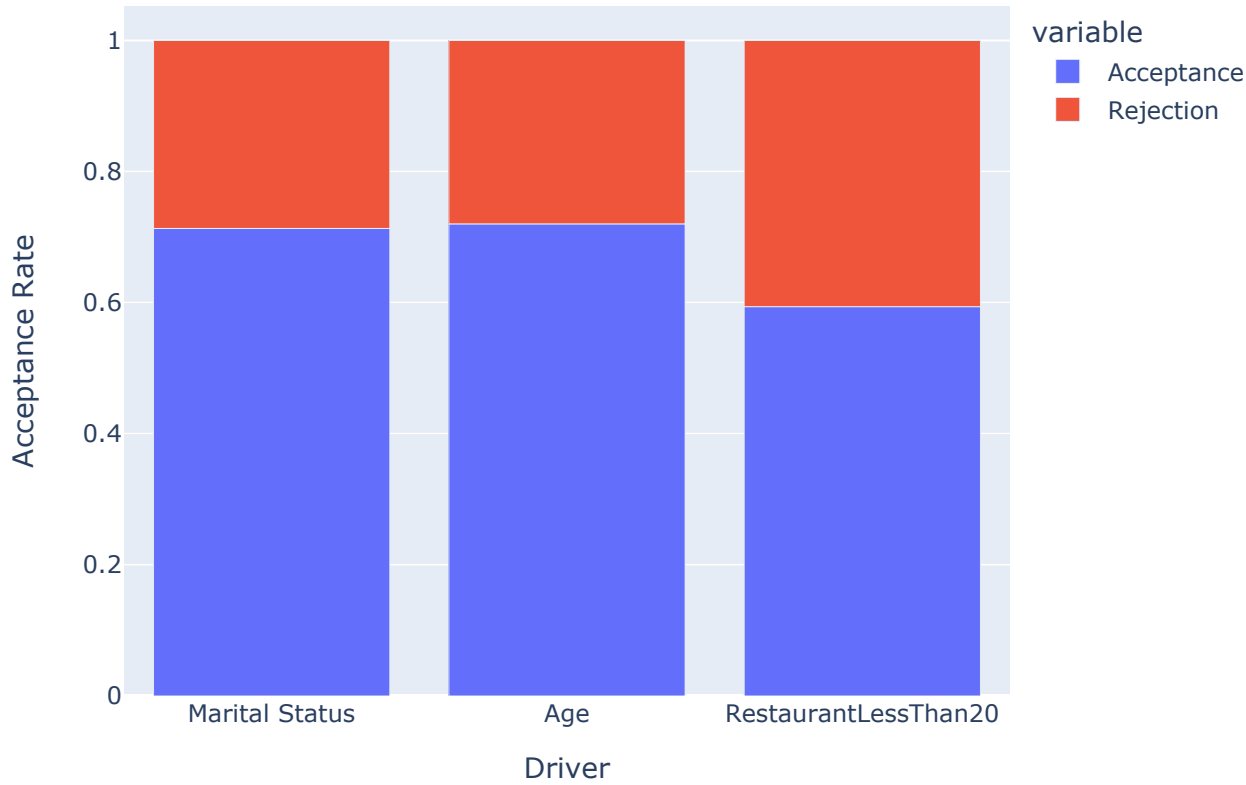
Accepted Coupons by Driver and age



Accepted Coupons by drivers for cheap restaurants



Acceptance Rate between drivers



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

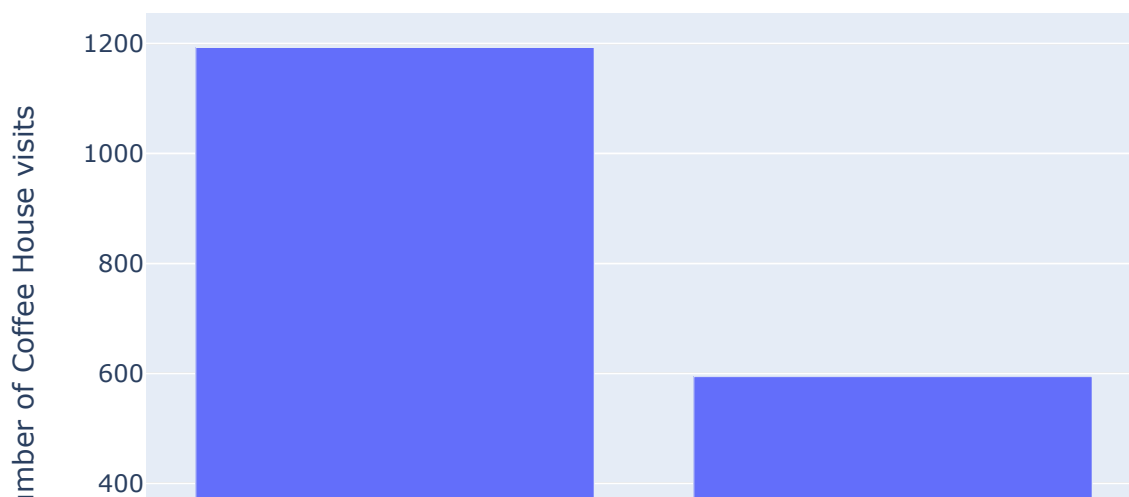
Overall Acceptance of bar coupons by drivers is around 41%. Drivers who tend to visit the bar less than 3 times a month accept more coupons than the drivers who visit more than 3 times. Drivers in the age group of 26 years accept more bar coupons compared to the other age groups over 25. The acceptance rate of coupons between drivers who go to a bar more than once a month and are over the age of 25 is greater than all others. However, drivers with kids and partners visit bar fewer times compared to the drivers driving alone or with friends. 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 accepted more number of coupons compared to all others. Drivers who drive alone tend to accept more coupons irrespective of the outside temperature. At lower temperatures, male drivers accept more coupons compared to female drivers who accept when the temperature outside is good. Number of coupons accepted by the drivers increases with increase in temperature. Drivers who drive alone and work at building, ground cleaning and maintenance accept coupons more number of times compared to the other occupation. Drivers with age 26 tend to accept more coupons compared to other age groups below 30. In terms of income, drivers whose income is more than 37500–49999 tend to visit more often and income around 25000–374999 visit less often.

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.

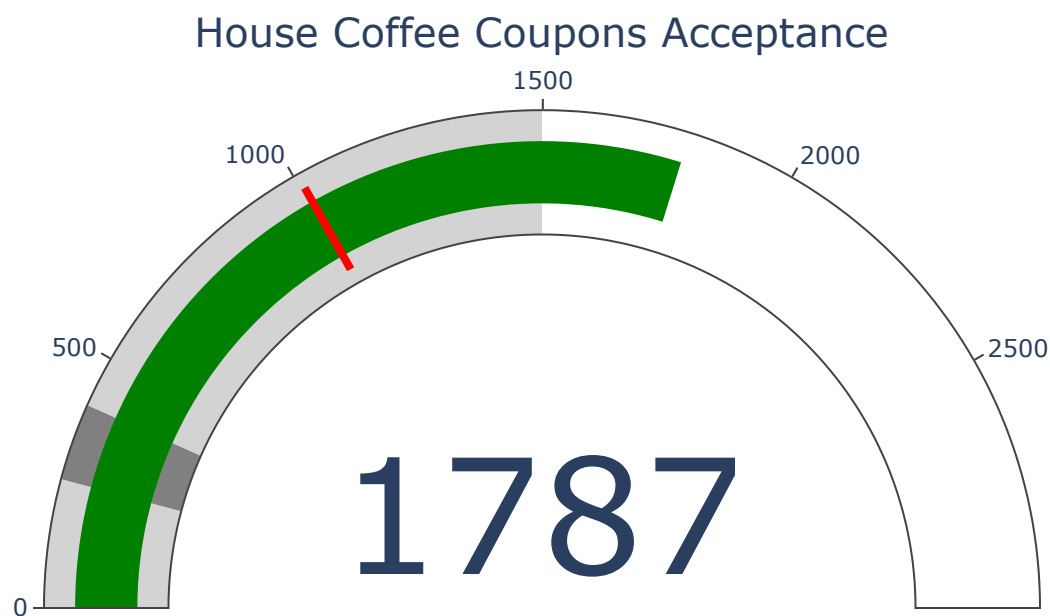
```
In [20]: dCoffeeHouse = data.query('coupon == "Coffee House" & CoffeeHouse != "never"')
#Compare the acceptance rate between those who went to a Coffee House 3 or fewer times a
dCoffeeHouse_totalCounts = dCoffeeHouse[['CoffeeHouse', 'Y']].query('Y == 1').value_counts()
dCoffeeHouse_totalCounts
fewerthan3 = (dCoffeeHouse_totalCounts[0] + dCoffeeHouse_totalCounts[1])
greaterthan3 = (dCoffeeHouse_totalCounts[2] + dCoffeeHouse_totalCounts[3])
x = ['<=3', '> 3']
y = [fewerthan3, greaterthan3]
px.bar(x=x, y=y, labels={'x': 'Coffee House visits', 'y': 'Total Number of Coffee House v
```

Acceptance rate between those went to Coffee House 3 or fewer times a





```
In [21]: fig = go.Figure(go.Indicator(
    domain = {'x': [0, 1], 'y': [0, 1]},
    value = dCoffeeHouse.query('Y == 1').shape[0],
    mode = "gauge+number",
    title = {'text': "House Coffee Coupons Acceptance"},
    gauge = {'axis': {'range': [None, (dCoffeeHouse.shape[0])]},
              'steps' : [
                  {'range': [0, dCoffeeHouse.shape[0]/2], 'color': "lightgray"},
                  {'range': [250, 400], 'color': "gray"}],
              'threshold' : {'line': {'color': "red", 'width': 4}, 'thickness': 0.75, 'value': dCoffeeHouse.shape[0]}
            ])
    fig.show()
```

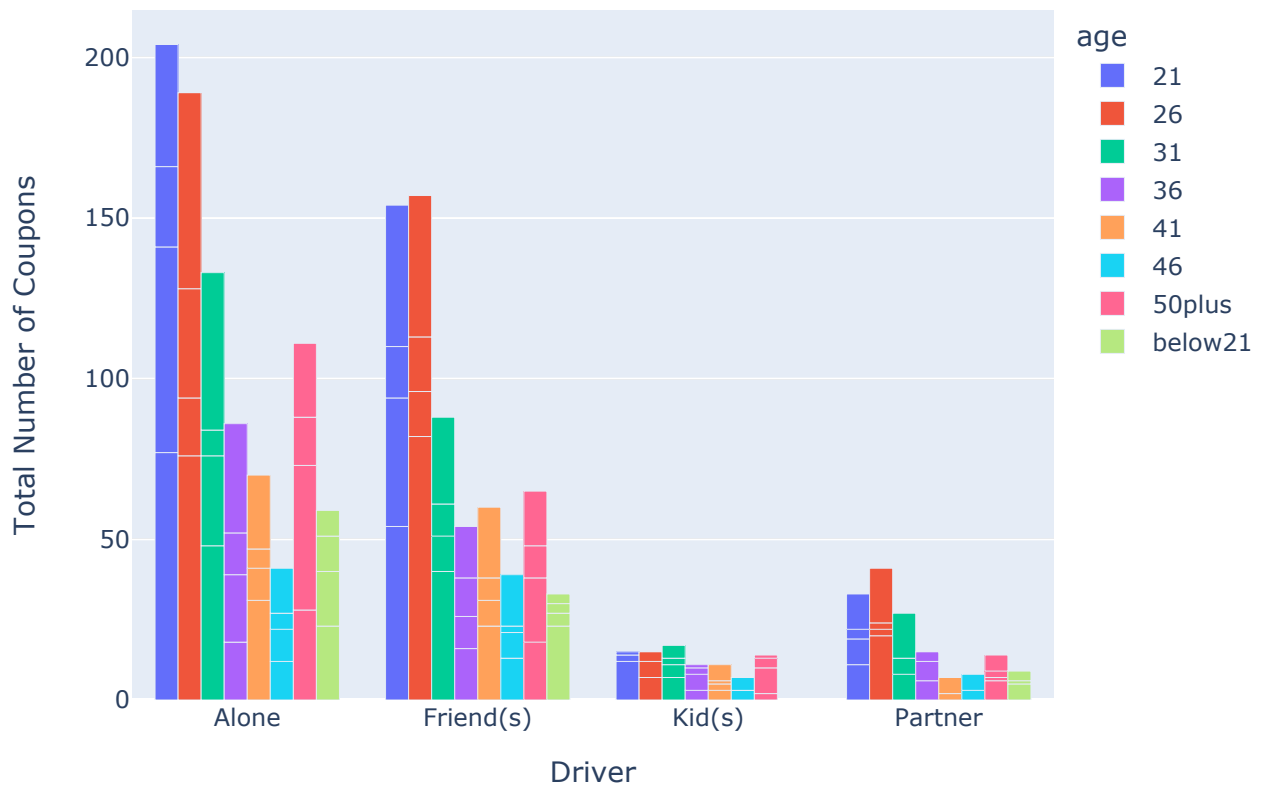


```
In [23]: ### Compare the acceptance rate between drivers who go to a Coffee House more than once

dCoffeeHouse_over25_acceptance = dCoffeeHouse.query('Y == 1')
dCoffeeHouse_over25_acceptance.groupby('age').sum()
dCoffeeHouse_over25_acceptance = dCoffeeHouse_over25_acceptance.groupby(['CoffeeHouse',
dCoffeeHouse_over25_acceptance.reset_index()
```

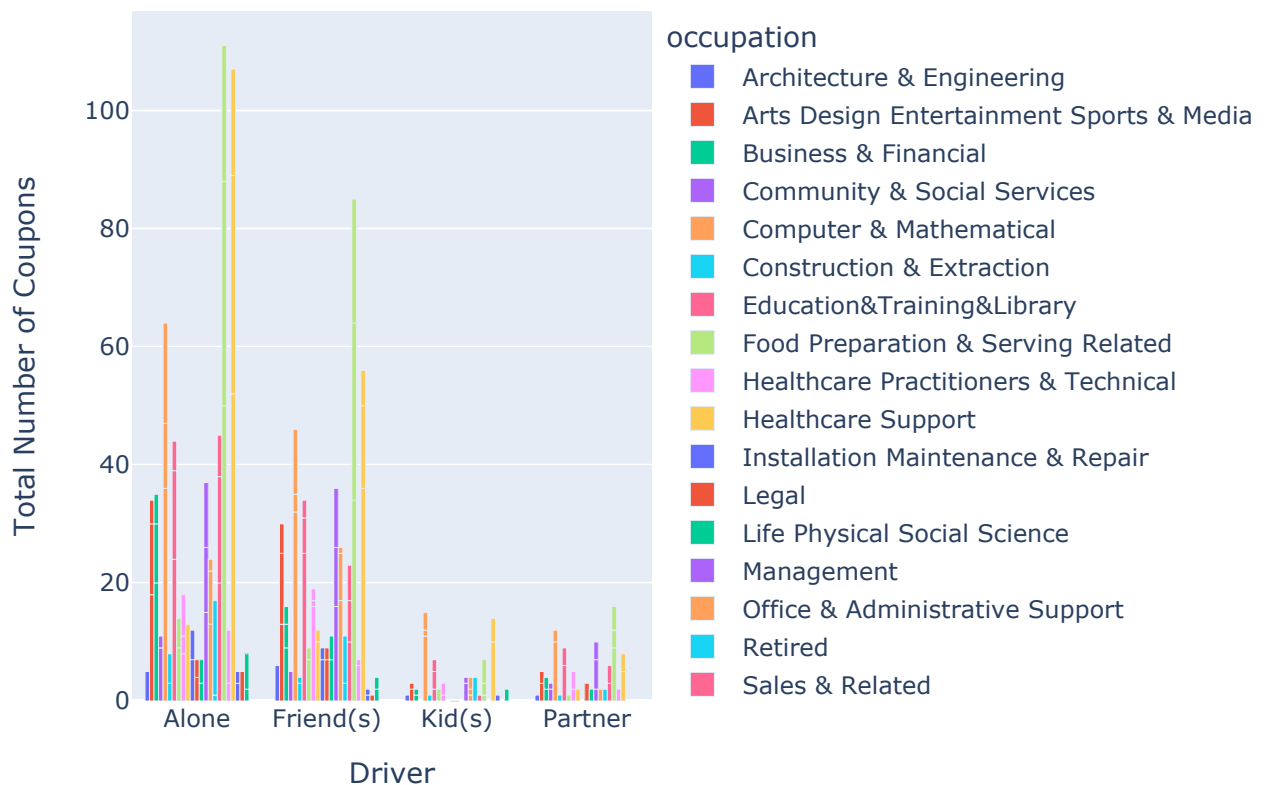
```
fig = px.bar(
    data_frame=dCoffeeHouse_over25_acceptance.reset_index(),
    x='passanger',
    y='Y',
    color='age',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Acceptance rate of drivers who go to a Coffee House more than once a month."
)
fig.show()
```

Acceptance rate of drivers who go to a Coffee House more than once a month



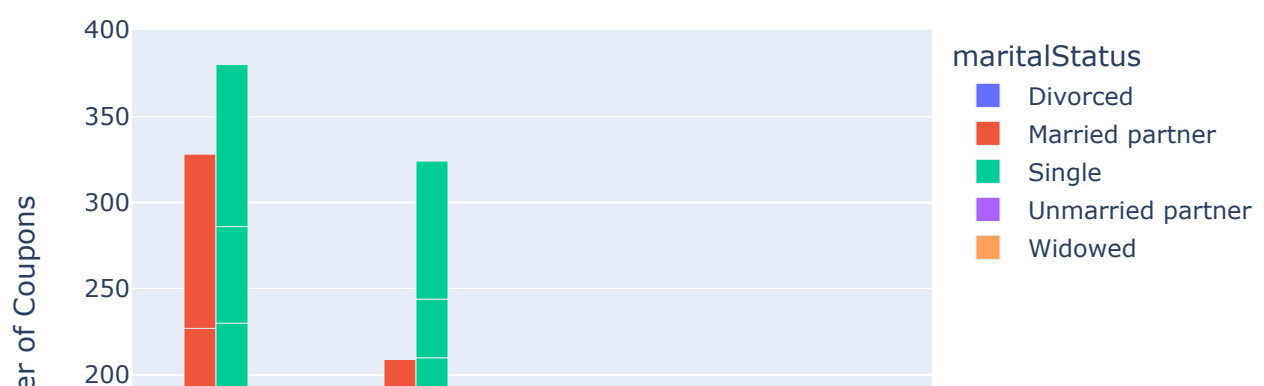
```
In [24]: #More than one month
dCoffeeHousebyOccupation = dCoffeeHouse.query('CoffeeHouse != "never" & CoffeeHouse != "
grouped_occupation = dCoffeeHousebyOccupation.groupby(['CoffeeHouse', 'passanger', 'occu
    {"Y": "sum"}
)
grouped_occupation.reset_index()
fig = px.bar(
    data_frame=grouped_occupation.reset_index(),
    x='passanger',
    y='Y',
    color='occupation',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Accepted Coupons for Coffee House by Driver and occupation"
)
fig.show()
```

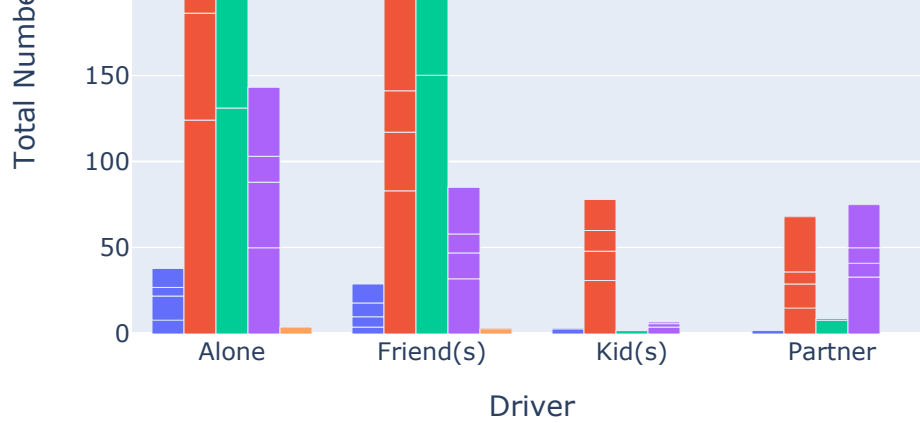
Accepted Coupons for Coffee House by Driver and occupation



```
In [25]: grouped_dfMaritalStatus = dCoffeeHouse.query('CoffeeHouse != "never"')
grouped_dfMaritalStatus = grouped_dfMaritalStatus.groupby(['CoffeeHouse', 'passanger', 'Y'])
grouped_dfMaritalStatus.reset_index()
fig1 = px.bar(
    data_frame=grouped_dfMaritalStatus.reset_index(),
    x='passanger',
    y='Y',
    color='maritalStatus',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Accepted Coupons for Coffee House by Driver and marital status"
)
fig1.show()
```

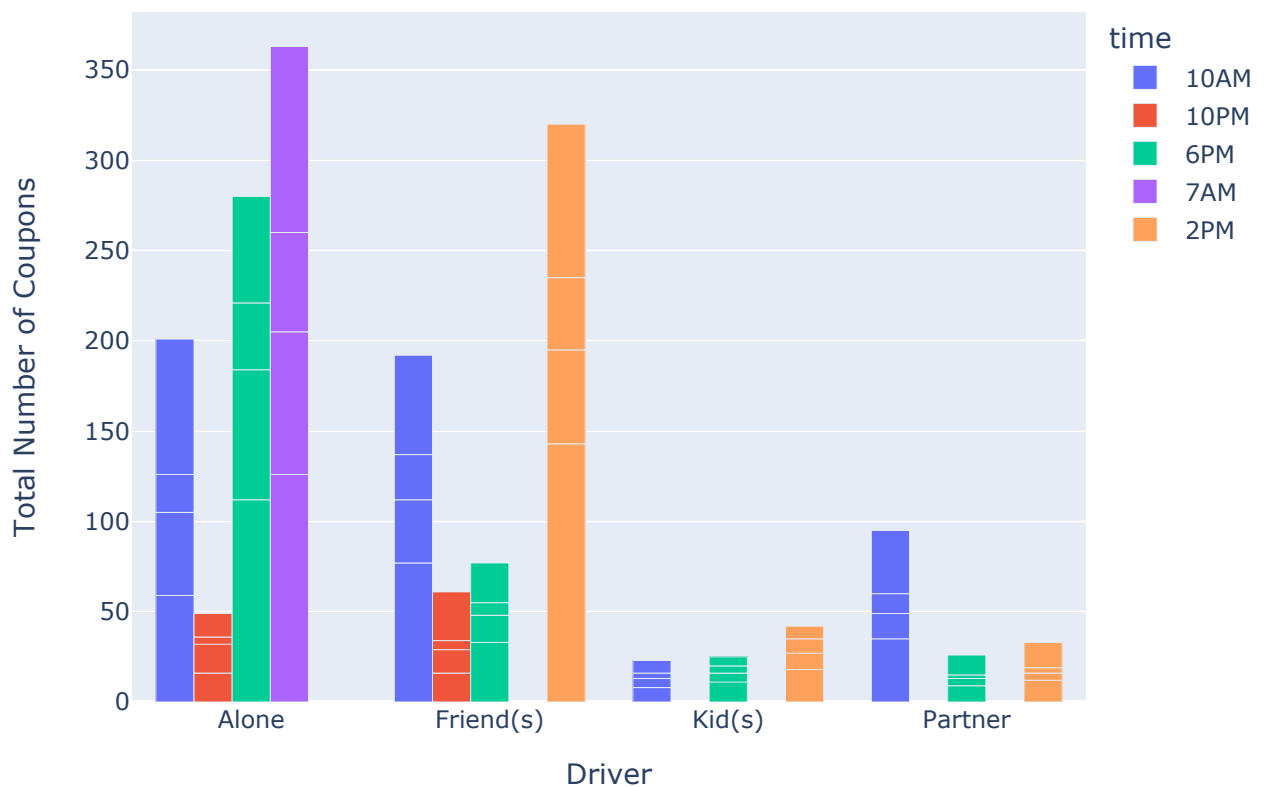
Accepted Coupons for Coffee House by Driver and marital status





```
In [26]: grouped_dfTime = dCoffeeHouse.query('CoffeeHouse != "never"')
grouped_dfTime = grouped_dfTime.groupby(['CoffeeHouse', 'passanger', 'time'], as_index="
{'Y': "sum"}
)
grouped_dfMaritalStatus.reset_index()
fig1 = px.bar(
    data_frame=grouped_dfTime.reset_index(),
    x='passanger',
    y='Y',
    color='time',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Accepted Coupons for Coffee House by Driver and time"
)
fig1.show()
```

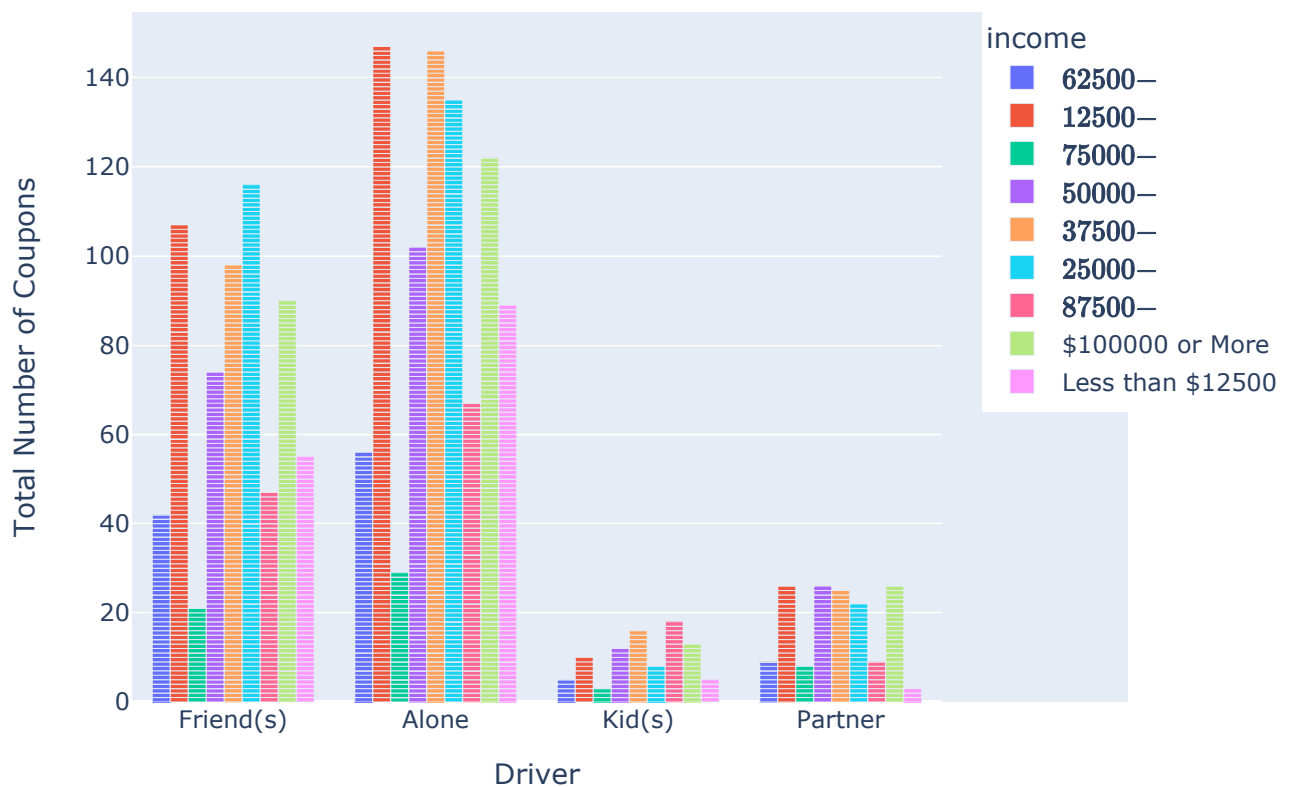
Accepted Coupons for Coffee House by Driver and time



```
In [27]: #dfCoffeeHouse3 = data.query('( (income == "$12500 - $24999"|income == "$25000 - $37499"

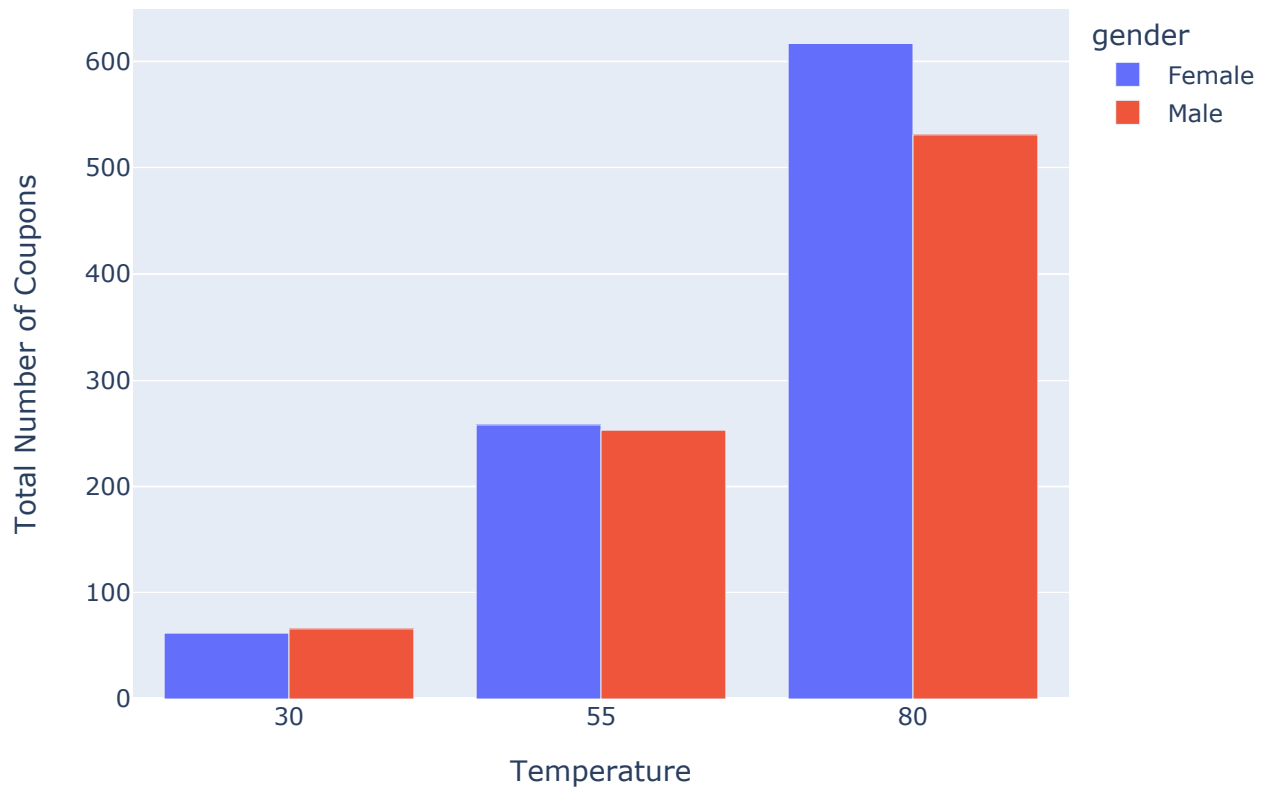
dCoffeeHouse.reset_index()
fig3 = px.bar(
    data_frame=dCoffeeHouse.reset_index(),
    x='passanger',
    y='Y',
    color='income',
    barmode="group",
    labels={'passanger': 'Driver', 'Y': 'Total Number of Coupons'},
    title="Accepted Coupons by drivers and income"
)
fig3.show()
```

Accepted Coupons by drivers and income



```
In [28]: grouped_df = dCoffeeHouse.groupby(['temperature', 'gender', 'coupon'], as_index="false")
        {"Y": "sum"}
        )
grouped_df.reset_index()
fig = px.bar(
    data_frame=grouped_df.reset_index(),
    x='temperature',
    y='Y',
    color='gender',
    barmode="group",
    labels={'temperature': 'Temperature', 'Y': 'Total Number of Coupons'},
    title='Male vs Female drivers acceptance of coupons by temeperature'
)
fig.show()
```

Male vs Female drivers acceptance of coupons by temeperature



In conclusion, drivers who visited the coffee house less than 3 times have accepted more number of coupons. The acceptance rate for coffee house coupons is over 50%. Drivers of age 21 and 26 have accepted more number of coupons compared to the other drivers. Food Preparation and serving related seem to accept more coffee house coupons. Drivers who are single and alone accepted the most number of coupons. Drivers driving alone at 7am have accepted more number of coffee house coupons. In terms of income, drivers with salary of 12500 and 37500 have accepted more number of coupons. Female drivers have accepted more number of coupons than male drivers.

In []:

In []:

In []: