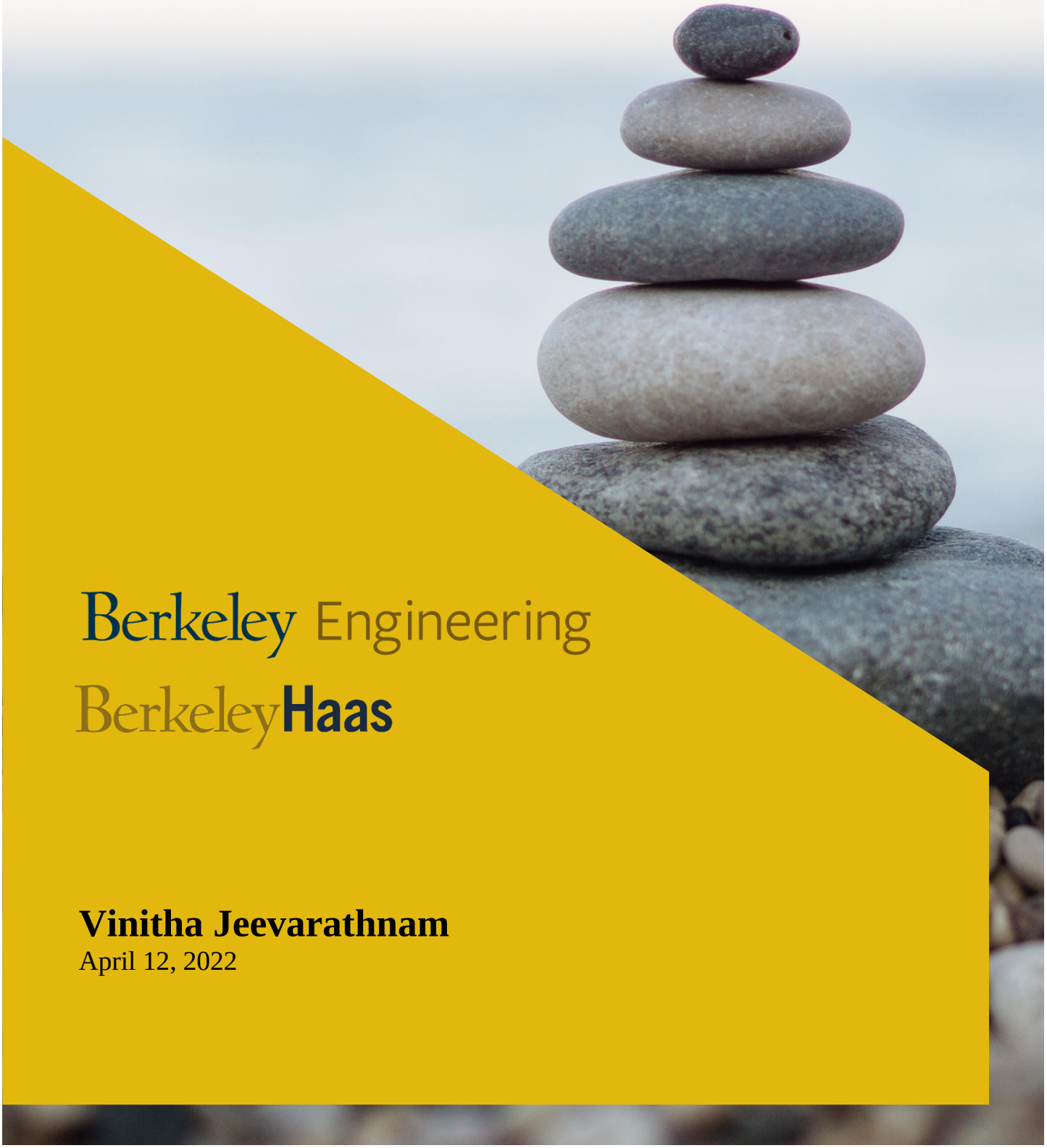
Module 5 – Practical Application 1



Berkeley Engineering

BerkeleyHaas

Vinitha Jeevarathnam

April 12, 2022

Table of Contents

1. Import Libraries

The following python libraries are used in this practical application,

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import plotly.express as px
```

2. Data Loading and Validation

- Load the data from coupon.csv

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

- Analyzed the data with head(), info(), and describe() to understand the data frame, data type, stats, and missing data.

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

Out[4]:

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CoffeeHouse	CarryAway	RestaurantLessTha
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	never	NaN	
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	never	NaN	
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	never	NaN	
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	never	NaN	
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	never	NaN	

5 rows × 26 columns

In [5]:
```
data.info() #Checking the Dtype of columns in the DataFrame
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64
 4   time                  12684 non-null  object
 5   coupon                12684 non-null  object
 6   expiration            12684 non-null  object
 7   gender                12684 non-null  object
 8   age                   12684 non-null  object
 9   maritalStatus         12684 non-null  object
 10  has_children          12684 non-null  int64
 11  education             12684 non-null  object
 12  occupation            12684 non-null  object
 13  income                12684 non-null  object
 14  car                   108 non-null    object
 15  Bar                   12577 non-null  object
 16  CoffeeHouse           12467 non-null  object
 17  CarryAway             12533 non-null  object
 18  RestaurantLessThan20  12554 non-null  object
 19  Restaurant20To50      12495 non-null  object
 20  toCoupon_GEQ5min      12684 non-null  int64
 21  toCoupon_GEQ15min     12684 non-null  int64
 22  toCoupon_GEQ25min     12684 non-null  int64
 23  direction_same        12684 non-null  int64
 24  direction_opp         12684 non-null  int64
 25  Y                     12684 non-null  int64
dtypes: int64(8), object(18)
memory usage: 2.5+ MB
```

In [6]:
```
data.describe()
```

Out[6]:

	temperature	has_children	toCoupon_GEQ5min	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same	direction_opp	Y
count	12684.000000	12684.000000	12684.0	12684.000000	12684.000000	12684.000000	12684.000000	12684.000000
mean	63.301798	0.414144	1.0	0.561495	0.119126	0.214759	0.785241	0.568433
std	19.154486	0.492593	0.0	0.496224	0.323950	0.410671	0.410671	0.495314
min	30.000000	0.000000	1.0	0.000000	0.000000	0.000000	0.000000	0.000000
25%	55.000000	0.000000	1.0	0.000000	0.000000	0.000000	1.000000	0.000000
50%	80.000000	0.000000	1.0	1.000000	0.000000	0.000000	1.000000	1.000000
75%	80.000000	1.000000	1.0	1.000000	0.000000	0.000000	1.000000	1.000000
max	80.000000	1.000000	1.0	1.000000	1.000000	1.000000	1.000000	1.000000

- Converting data frame datatypes,

In [7]:
```
data.convert_dtypes().dtypes #Converting DF columns to standard Dtypes
```

Out[7]:
```
destination             string
passanger               string
weather                 string
temperature              Int64
time                    string
coupon                  string
expiration              string
gender                  string
age                     string
maritalStatus           string
has_children             Int64
education               string
occupation              string
income                  string
car                     string
Bar                     string
CoffeeHouse             string
CarryAway               string
RestaurantLessThan20    string
Restaurant20To50        string
toCoupon_GEQ5min         Int64
toCoupon_GEQ15min        Int64
toCoupon_GEQ25min        Int64
direction_same           Int64
direction_opp            Int64
Y                        Int64
dtype: object
```

- Since age is mostly numeric and shows as a string, the following logic is applied to convert to int64

In [8]:
```
#Convert age from string to int64
data['age'].replace({'50plus':'50', 'below21':'20'}, inplace=True) #Replacing 50plus with 50 and below 21 to 20
data['age'] = data['age'].astype(np.int64)
```

- Checking for Null or NaN values

```
In [9]: data.isnull().sum() #Checking null values per column in the DF
```

```
Out[9]: 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
```

- Checking for any special characters

```
In [10]: #Checked ALL columns in the DF for special characters etc. For example,
         data['destination'].sort_values().unique()
```

```
Out[10]: array(['Home', 'No Urgent Place', 'Work'], dtype=object)
```

- Renaming columns

```
In [11]: #Standardized all column names to lower case and reassigned to a new DF
         data1 = data.rename(columns = str.lower)
```

- Dropping column(s) and ‘NaN’ values

```
In [12]: #Since we are missing 85% of the car information, we can drop this column and assigning it to a new DF
         data2 = data1.drop(columns = ['car'])
```

```
In [13]: #Checked for common NaN values in bar, coffeehouse, carryaway, restaurantlessthan20 and restaurant20to50 and dropped the rows
         data3 = data2.dropna(subset=['bar', 'coffeehouse', 'carryaway', 'restaurantlessthan20', 'restaurant20to50'], how='all')
         #Reduced DF from 12684 to 12642 rows
```

```
In [14]: #Filling NaN values with "No Data"
         data4 = data3.replace(to_replace = np.nan, value='No Data')
```

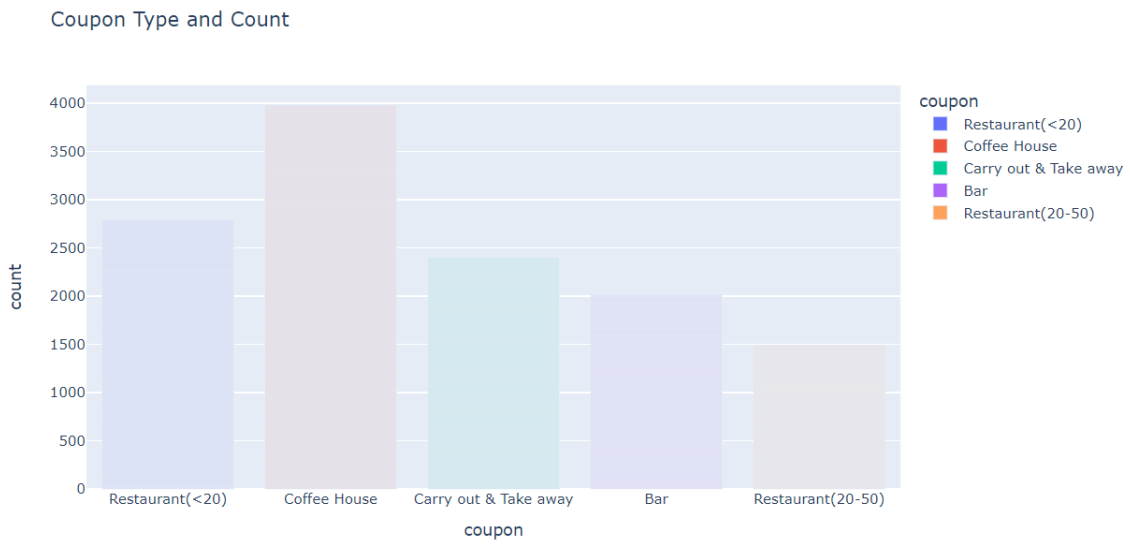
3. Data Analysis and Visualization - Coupons

```
In [18]: sns.countplot(data=data4, x='y')
         plt.title('Coupons Not Accepted vs Accetped')
         plt.xlabel('Coupon Acceptance: 0-No, 1-Yes')
         #The no. of coupons accepted is higher than not accepted

Out[18]: Text(0.5, 0, 'Coupon Acceptance: 0-No, 1-Yes')
```

In the entire data set, the coupon acceptance rate is higher than the coupon not accepted.

```
In [15]: px.bar(data4, x='coupon', title='Coupon Type and Count', color='coupon')
         #Coffee House coupons are the most popular coupon among the other categories
```



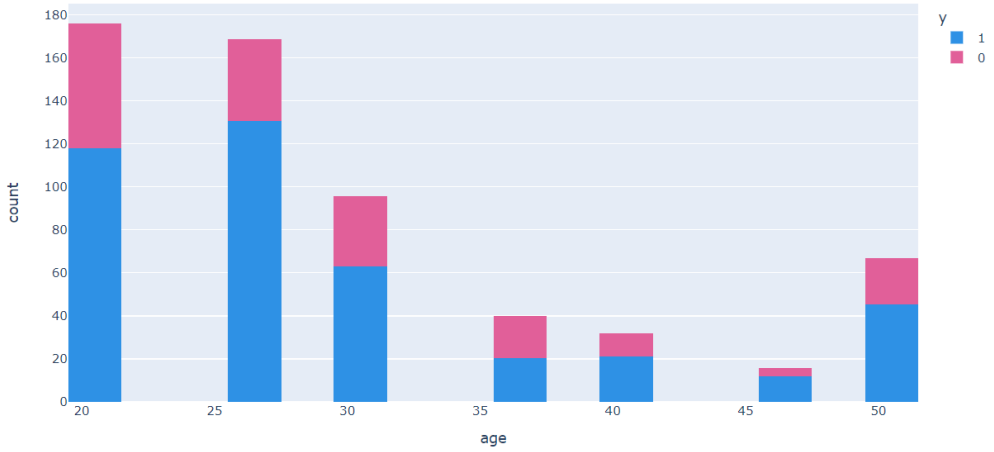
Coffee house coupons are more popular than other coupon categories.

4. Data Analysis and Visualization - Bar Coupon Acceptance

Here are some summary statistics of the acceptance rate of bar coupons among a few data points,

- 40.9% of drivers accepted a bar coupon
- Drivers who go to bar 1-3 times accept more bar coupons than those who go to bar frequently
- 35.5% of drivers over age 25 who goes to a bar at least once a month will accept a bar coupon.
- The probability of bar coupon acceptance among this group is higher than the younger counterparts (age less than 25)
- Acceptance of bar coupons among those who frequent the bar by age,

```
In [22]: px.histogram(data4_bar.query("bar == ['1~3','4~8','gt8']"), x='age', color='y', color_discrete_sequence=px.colors.qualitative.Dar
```



The above chart confirms that younger drivers (under 25) go to bars more often than the other groups. But, when it comes to the bar coupon acceptance rate, driver of age over 25 tends to accept more bar coupons than their younger counterparts. Though the chart trend goes down between ages 30 and under 50, the trend picks back up with older people (over age 50) as they frequent bars and accept more bar coupons.

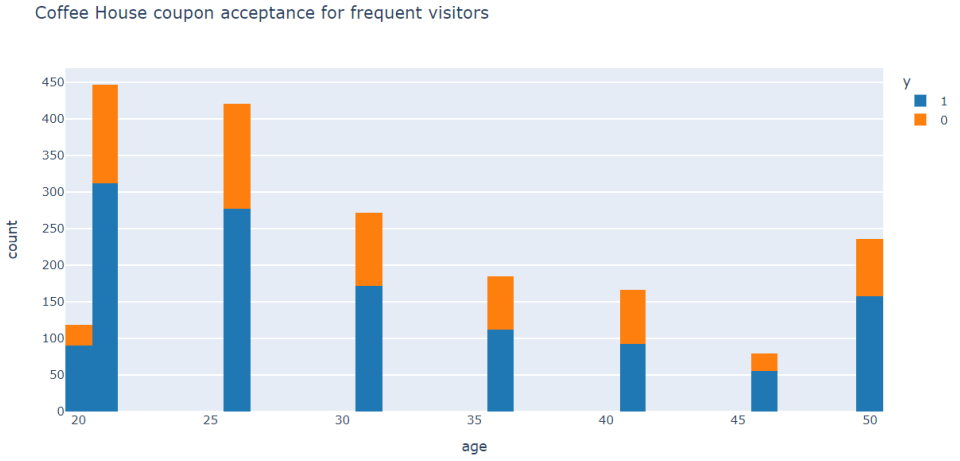
- Though younger drivers tend to go to bars more often, only those who are between 26 and 27 ages tend to accept more bar coupons than all other groups. The bar-goers accept only about 18.9% of cheap restaurant coupons and an income below $50K doesn't influence the acceptance of the cheap restaurant coupons.

5. Independent Analysis – Coffee House Coupon Acceptance

Since Coffeehouse has the highest coupon count, I am choosing this group to understand the coupon acceptance rate.
Here are the summary statistics of the coffee house coupon acceptance analysis,

- There is a 49.8% likelihood that coffee house coupons will be accepted by drivers
- Drivers who go to coffee houses 1-3 times accept more coffee house coupons than those who go to coffee houses frequently
- Drivers who are above age 25 tend to accept more coffee house coupons than younger drivers
- Analysis of coffee house coupon acceptance rate for those visited at least once a month and over age 25

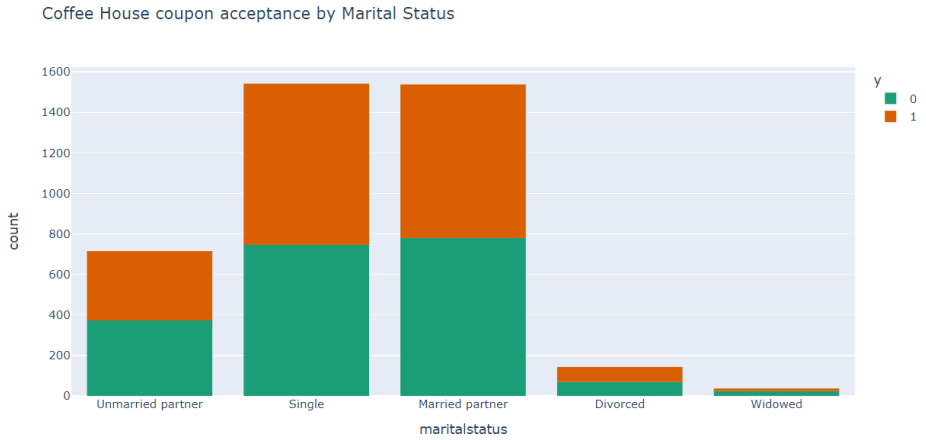
```
In [36]: px.histogram(data4_coffee.query("coffeehouse == ['1~3','4~8','gt8']"), x='age', color='y', color_discrete_sequence=px.colors.qual
```



The above chart confirms that younger people (under 25 age) go to bars more often than people of age over 25. But the chart also sheds the light that the younger drivers accept more coffee house coupons than others. Though the coupon acceptance trend goes down as the population gets older, it trends back up with older people (over age 50) who frequent coffee houses and accepts more coffee house coupons.

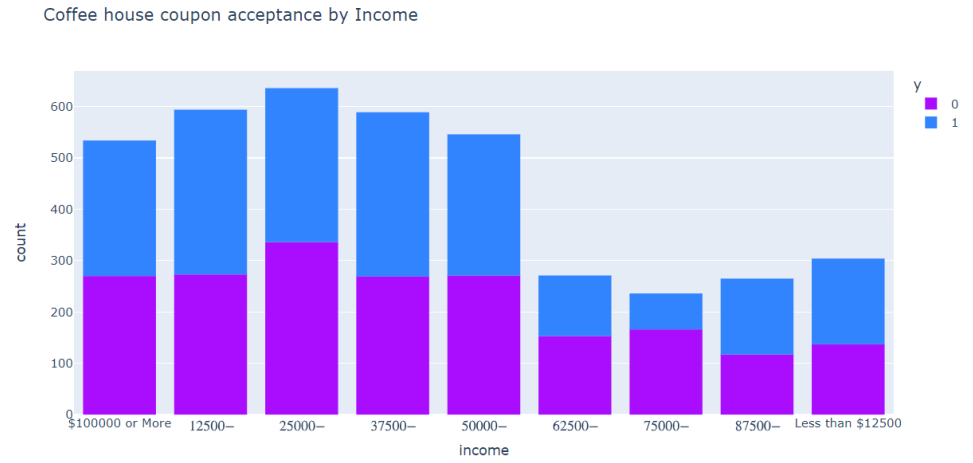
- Histogram charts to show coffee house coupon acceptance by Marital Status, Income, Education, and Gender

```
In [37]: #Coffee house coupon acceptance by Martial status
px.histogram(data4_coffee, x='maritalstatus', color='y', color_discrete_sequence=px.colors.qualitative.Dark2, title='Coffee House
```



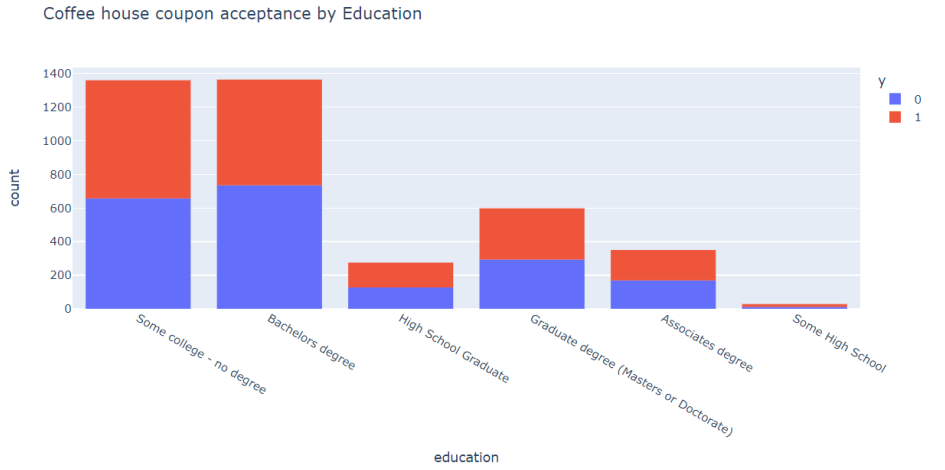
Single drivers accept more coffee house coupons followed by married partner drivers. Drivers with widowed marital status drivers visit and accept fewer coffee houses and coupons.

```
In [38]: #Coffee house coupon acceptance by Income
px.histogram(data4_coffee.sort_values(by='income'), x='income', color='y', color_discrete_sequence=px.colors.qualitative.Alphabet
```



Drivers making income between less than $12.5K and $50K tend to accept more coffee house coupons. Those making income between $51K and $99K tend to accept fewer coffee house coupons. But, the trend switches back for those who make an income of $100K or more to accept more coffee house coupons.

```
In [39]: #Coffee house coupon acceptance by Education
px.histogram(data4_coffee, x='education', color='y', color_discrete_sequence=px.colors.qualitative.Plotly, title='Coffee house co
```



Drivers with no college degree accepted more coffee house coupons than others. People in high school tend to visit and accept fewer coffee houses and coupons.

```
In [40]: #Coffee house coupon acceptance by Gender
px.histogram(data4_coffee, x='gender', color='y', color_discrete_sequence=px.colors.qualitative.Vivid, title='Coffee house coupon
```

Female drivers tend to go to coffee houses more than male drivers. Though the female drivers tend to reject more coffee house coupons, they still have a high coupon acceptance rate when compared to their male counterparts.

6. Correlation Analysis of Coffee House Data Frame

```
In [44]: data4_coffee_corr = data4_coffee.corr()
```

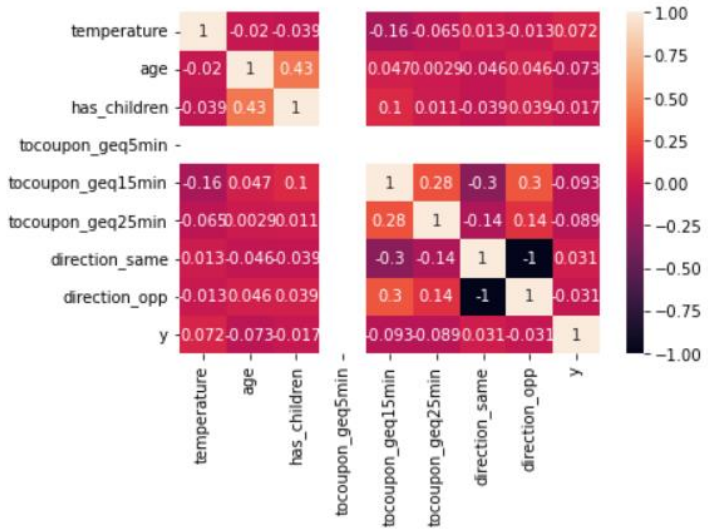
```
In [45]: data4_coffee_corr
```

Out[45]:

	temperature	age	has_children	tocoupon_geq5min	tocoupon_geq15min	tocoupon_geq25min	direction_same	direction_opp	
temperature	1.000000	-0.020453	-0.039430	NaN	-0.160117	-0.065253	0.013308	-0.013308	0.07220
age	-0.020453	1.000000	0.431939	NaN	0.046667	0.002879	-0.046051	0.046051	-0.07309
has_children	-0.039430	0.431939	1.000000	NaN	0.100791	0.011209	-0.039268	0.039268	-0.01730
tocoupon_geq5min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
tocoupon_geq15min	-0.160117	0.046667	0.100791	NaN	1.000000	0.279001	-0.295472	0.295472	-0.09332
tocoupon_geq25min	-0.065253	0.002879	0.011209	NaN	0.279001	1.000000	-0.140252	0.140252	-0.08940
direction_same	0.013308	-0.046051	-0.039268	NaN	-0.295472	-0.140252	1.000000	-1.000000	0.03067
direction_opp	-0.013308	0.046051	0.039268	NaN	0.295472	0.140252	-1.000000	1.000000	-0.03067
y	0.072209	-0.073097	-0.017303	NaN	-0.093320	-0.089406	0.030670	-0.030670	1.00000

```
In [46]: sns.heatmap(data4_coffee_corr, annot=True)
```

```
Out[46]: <AxesSubplot:>
```



There is a medium correlation between drivers' age and children when it comes to the correlation analysis of the coffee house data.