Assignment 2 Data Science

Jacob, Christian og Mia

March 2022

Introduktion

Givet datasættet Haircut-tip-amounts, der bl.a. indeholder data på hvor meget gifte og ikke gifte mennekser har tippet, samt hvilken kultur der tipper mest og oftest. På baggrund af datasættet vil vi undersøge hvornår på dagen, og hvilken dag, der tippes bedst. Vi vil også undersøge om det er gifte mennesker eller singler der giver de bedste tips. Vi vil ligeledes forsøge at besvare spørgsmålet om hviken kultur der tipper bedst og oftest.

Github repo som kan findes under mappen Assignment 2.

Cleaning datasættet

Vi startede med at hente datasættet ind i vores notebook og sætte det ind i et dataframe "df":

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

url = './Haircut_Tip_amounts.xlsx'
df = pd.read_excel(url)
```

	Tip amount	Time	Day	Culture	Married	Age	Unnamed: 6	Unnamed: 7	Culture Codes	Unnamed: 9
0	0	5.0	Fri	I	NM	30.0	NaN	NaN	I	Indian
1	1	2.0	Tue	E	NM	10.0	NaN	NaN	E	European
2	1	7.0	Tues	I	NM	35.0	NaN	NaN	B	African American
3	2	5.0	Mon	I	NM	35.0	NaN	NaN	W	American
4	2	12.0	Fri	M	NM	15.0	NaN	NaN	P	Phillipino

Vi kunne se at der var nogle tomme kollonner markeret med NaN ved navn 'Unnamed: 6' og 'Unnamed: 7'. Kulturkoderne var også kommet med, men dem var vi ikke interesserede i at arbejde med, men bare havde brug for at kende.

Først fjernede vi de tomme kolonner ved hjælp af følgende kode:

```
df = df.drop(['Unnamed: 6'], axis=1)
df = df.drop(['Unnamed: 7'], axis=1)
```

Datasættet var delt op i noget cleaned data og noget ikke cleaned. Det cleanede data startede fra linje 117 og det var kun det vi ville arbejde med, så vi benyttede os af iloc:

```
df = df.iloc[116:,]
```

Nu fjerner vi kulturkode kolonnen samt en anden tom kolonne:

```
df = df.drop(['Culture Codes'], axis=1)
df = df.drop(['Unnamed: 9'], axis=1)
```

Vi ender med et cleaned datasæt således:

	Tip amount	Time	Day	Culture	Married	Age
116	20	10.0	mon	P	NM	30.0
117	5	11.0	mon	W	NM	16.0
118	5	11.0	mon	E	M	40.0
119	3	11.0	Fri	M	M	40.0
120	6	11.0	Fri	W	NM	30.0
...	...	...	...	...	...	...
229	10	1.0	Sat	W	M	40.0
230	1	13.0	Sat	I	NM	10.0
231	5	14.0	Sat	W	M	36.0
232	3	15.0	Sat	W	NM	8.0
233	5	16.0	Fri	I	NM	35.0

118 rows × 6 columns

Det eneste der nu mangler er at dagene ikke er ens, eksempelvis Tues = Tue og mon = Mon. Vi starter med at lave alle dage til capslock:

```
df['Day'] = df['Day'].str.upper()
```

Så er mon rettet så de alle er ens. Tues er lidt en anden sag som vi løser ved at udskifte alle Tues med Tue:

```
df['Day'] = df['Day'].replace(['TUES'], 'TUE')
```

Vi har nu et cleaned datasæt som vi kan arbejde med. Kolonnerne har følgende typer:

```
df.dtypes
```

```
Tip amount     object
Time          float64
Day            object
Culture        object
Married        object
Age           float64
dtype: object
```

Tip amount står som default som et obejct, det laver vi om til en int så vi kan arbejde med den:

```
df['Tip amount'] = df['Tip amount'].astype(int)
```

```
df.dtypes
```

```
Tip amount      int64
Time          float64
Day            object
Culture        object
Married        object
Age           float64
dtype: object
```

Hvornår på dagen, og hvilken dag, tippes der bedst

For at finde ud af hvilken dag der bliver tippet bedst kan vi sortere dataframet efter dag og finde gennemsnittet af hver dag:

```
df.groupby('Day').mean()
```

Day	Tip amount	Time	Age
FRI	6.333333	12.266667	31.400000
MON	6.739130	14.173913	31.260870
SAT	5.744186	13.255814	31.813953
THUR	6.000000	12.142857	31.428571
TUE	9.200000	14.850000	27.900000
WED	8.300000	11.600000	27.600000

På ovenstående kan vi aflæse at om tirsdagen (**TUE**) har vi i gennemsnit den højeste tip amount, nemlig **9,2**.

For at undersøge hvornår på dagen det højeste tip gives, laver vi et nyt dataframe sorteret efter TUE:

```
dfTue = df.loc[df['Day'] == 'TUE']
```

Hvis vi udskriver den sorteret efter højeste tip:

```
dfTue.sort_values('Tip amount', ascending = False)
```

	Tip amount	Time	Day	Culture	Married	Age
185	40	16.0	TUE	P	NM	35.0
190	25	18.0	TUE	I	NM	30.0
225	20	15.0	TUE	P	NM	30.0
224	10	14.0	TUE	W	M	35.0
183	10	15.0	TUE	P	NM	25.0
188	10	17.0	TUE	W	M	40.0
189	8	18.0	TUE	W	NM	31.0
137	7	11.0	TUE	P	NM	30.0
192	5	19.0	TUE	W	NM	10.0
139	5	14.0	TUE	I	NM	30.0
140	5	14.0	TUE	W	M	35.0

Kan vi se at de højeste tips ligger på **40 og 25** og gives henholdsvis mellem kl. **16 og 18.**

Giver gifte mennesker eller singler de bedste tips?

Fra vores research kan vi se at ikke gifte mennesker tipper mere end gifte mennesker gør. Dette kunne begrundes af at singler ikke skal bruge penge på andre end dem selv, med mindre de selvfølgelig har børn. Derfor har de måske lidt ekstra penge som de kan tippe med. For at finde ud af dette er skal der først findes ud af hvor mange der er NM = ikke gift og hvor mange der er M = gift. dette er gjort med følgende kode:

```
dfNM = df.loc[df['Married'] == 'NM']
dfNM.count()
```

Der er 79 som ikke er gift som også kan ses på nedestående billede:

```
Tip amount    79
Time          79
Day           79
Culture       79
Married       79
Age           79
dtype: int64
```

Det samme er gjort for dem der er gifte med nedestående kode:

```
dfNM = df.loc[df['Married'] == 'M']
dfNM.count()
```

Hvor der er 39 som er gift hvilket næsten er halvdelen i forhold til ikke gifte:

```
Tip amount    39
Time          39
Day           39
Culture       39
Married       39
Age           39
dtype: int64
```

Næste trin er at finde ud af hvad gennemsnittet er med hvad tip amount er både for ikke gift men også for gifte, dette er gjort med nedestående kode:

```
dfNM.groupby('Married').mean()
```

På nedestående billede kan det ses at gennemsnittet for ikke gite er 7,02 kr pr. person

Married	Tip amount	Time	Age
NM	7.025316	13.670886	26.088608

Igen er det samme gjort for gifte med nedestående kode:

```
dfM.groupby('Married').mean()
```

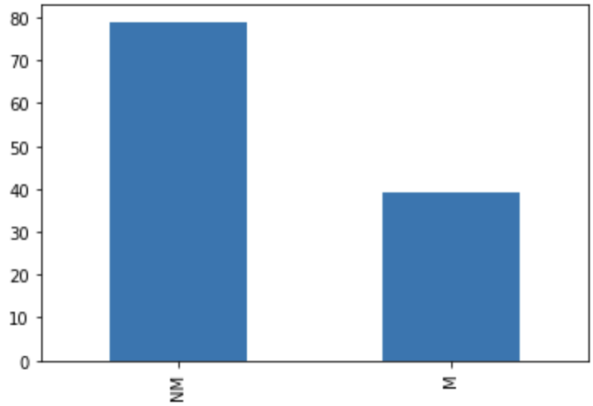
Hvor gennemsnittet er 6,43 kr pr person som også kan ses på billedet under

Married	Tip amount	Time	Age
M	6.435897	12.769231	39.769231

De sidste par trin er at få dataen mere visuelt hvilket er gjort med nedestående kode hvor der er lavet en sammenligning med gift og ikke gift så man kan se det med et diagram istedet for kun tal.

```
df['Married'].value_counts().plot(kind='bar')
```

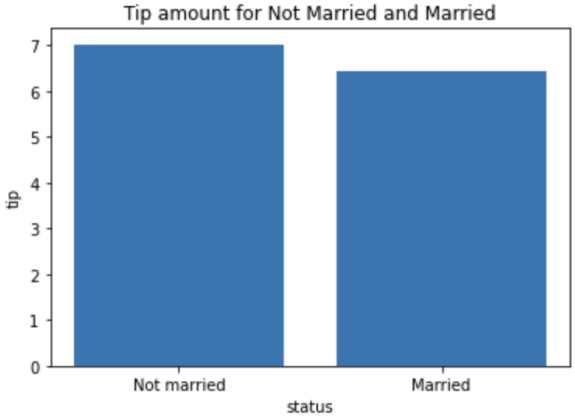
Oversteående kodde kommer til at se sådan ud rent visuelt.



Næste graf viser gennemsnittet af tips som er giver af gifte og ikke gifte og dette er igen for at få det mere visuelt istedet for kun tal og der kan rent faktisk ses en stor forskel selvom differencen er så lav.

```
status = ['Not married', 'Married']
plt.bar(status, tip)
plt.title('Tip amount for Not Married and Married')
plt.xlabel('status')
plt.ylabel('tip')
plt.show()
```

Dette afspejer forskellen med tip mellem gift og ikke gift



I artikelen Financial Benefits of Marriage vs. Being Single – What's Better?, beskrives der at når man bliver gift stiger ens rigdom, i hvert fald i Amerika. Dette strider mod vores teori, da vores resultater kan tolkes som det modsatte.

"*A 2005 study from The Ohio State University (OSU) found that people saw a sharp increase in their level of wealth after getting married.*"

Undersøgelsen er fra 2005 og derfor kan omstændighederne godt have ændret sig. Ligeledes beskriver artiklen Singler tjener mindre end folk i fast forhold: Her er årsagen, fra 2017, at singler tjener mindre og har større udgifter.

"*Singler har højere udgifter end folk i fast forhold, og de får statistisk set lavere løn..*"

Vi har derfor ikke nogen empiriske beviser som understøtter vores undersøgelser om at ikke gifte mennesker giver mere i tips.

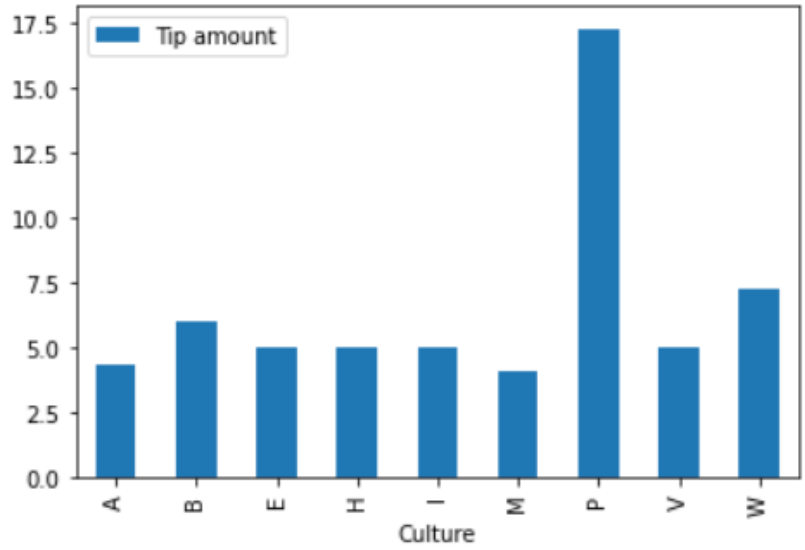
Hviken kultur tipper bedst?

For at besvare dette spørgsmål laver vi et ny dataframe på baggrund af 'Tip amount' og 'Culture' kolonnerne:

```
dfCult = df.loc[:, ['Tip amount', 'Culture']]
```

Denne plotter vi på baggrund af det gennemsnitlige tip amount pr. kultur:

```
dfCult.groupby('Culture').mean().plot(kind='bar')
```

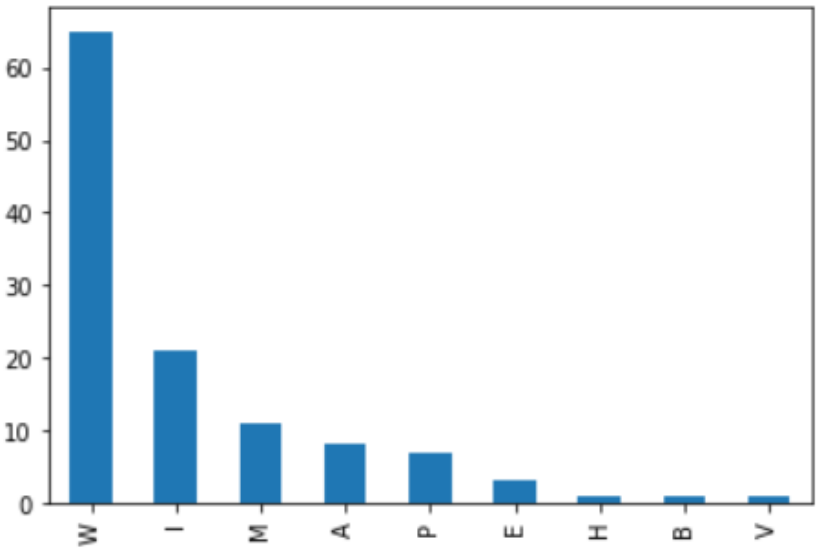


På ovenstående kan man se at P, som er Philipinerne, tipper bedst gennemsnitligt.

Hviken kultur tipper oftest?

Vi kan hurtigt finde ud af hvilken kultur der tipper oftest ved at sortere efter antallet af tips der er givet pr. kultur:

```
df['Culture'].value_counts().plot(kind='bar')
```



Her kan vi se at den amerikanske kultur tipper oftere end andre kulturer. Amerika, som er W, tipper bemærkelsesværdigt oftere end man gør i de andre kulturer, nemlig over 3 gange så meget som den indiske kultur. P, Philipinerne, som tipper med højere beløb tipper utroligt lidt i forhold til Amerikanerne.

Vores fund kan understøttes af artikelen Tipping culture around the world. How much should you tip for services?, der beskriver at amerikanerne tipper mere da det er en del af den ansattes løn.

"*In the United States and Canada, the tip is higher than in most other places, as it's seen as part of the employee's income.*"

Artiklen skriver også at man i europa ser anderledes på tipping, og dermed ikke tipper lige så ofte. Der skal være en bemærkelsesværdig service før man vælger at tippe her.

"*In France it is considered opulent for someone to leave a tip without good reason.*"

Mange steder i europa bliver et tip ikke forventet da lønnen på eksempel hoteller og restauranter er så god. MEn man bliver selvfølgelig heller ikke fornærmet over at få et tip.

"*Nordic countries also provide decent wages for workers in restaurants and hotels, so the tip is not expected, but not considered offensive.*"

Vi kan dog tale af erfaring at man ikke altid må modtage tips hvis man eksempelvis arbejder i et supermarked i Danmark.

Det etiske

Nogen ville mene at sammenligne tip amount med kultur eller gifte status er forkert, fordi det kan være stødende hvis man eksempelvis er nyskilt og derfor ikke har så mange penge at tippe med. Også, hvis man kommer fra en fattigere kultur kan man måske ikke ligge lige så mange penge som rige hvide amerikanere. Dataen handler jo om en bestemt frisør, så hvis mange af kunderne er flygtninge har de måske ikke ressourcer til at tippe. Det er samtidig ikke alle der er interesseret i at oplyse sin alder, derfor kunne man også diskutere etikken ved at alderen er en faktor her.