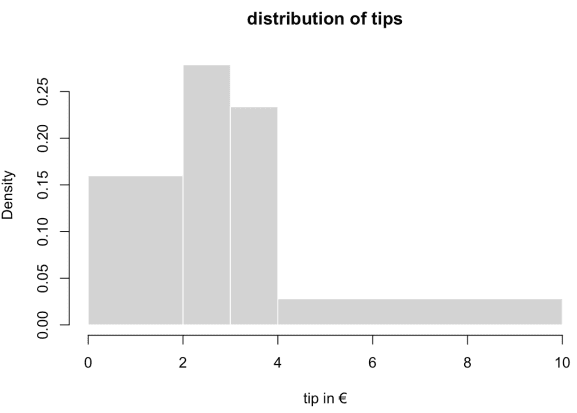


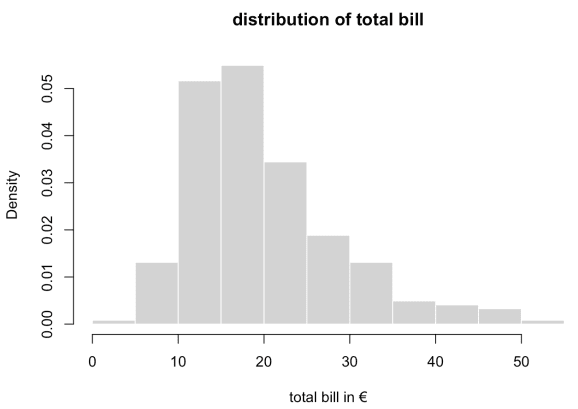
Based on these informations, how would you maximize your **tip** as a waiter working in this restaurant?

# 1 – Descriptive analysis

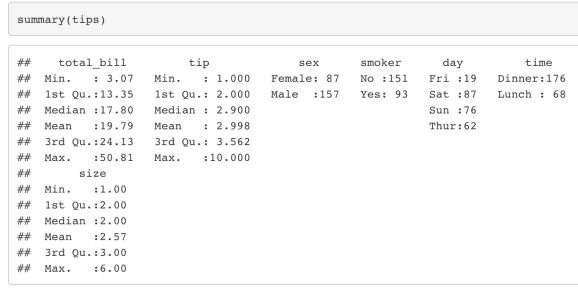
The tips (available in variable tip in tips) range from 0 to 10€, and are **mostly comprised between 2 and 4€**:



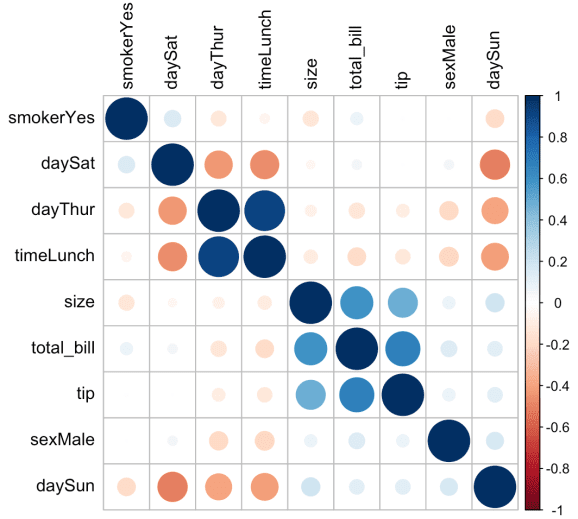
Another interesting information is the **amount of total bills**, which is comprised between 3 and 50€, and mostly between 10 and 20€:



Both distributions – of tips and total bill amounts – are **left-skewed**. We could fit a probability distribution to each one of them, such as lognormal or Weibull, but this would not be extremely informative. We would be able to derive some confidence intervals or things like the **probability of having a total bill higher than 40€** though. Generally, in addition to tip and total\_bill, we have the following raw information on the **marginal distributions of other variables**:



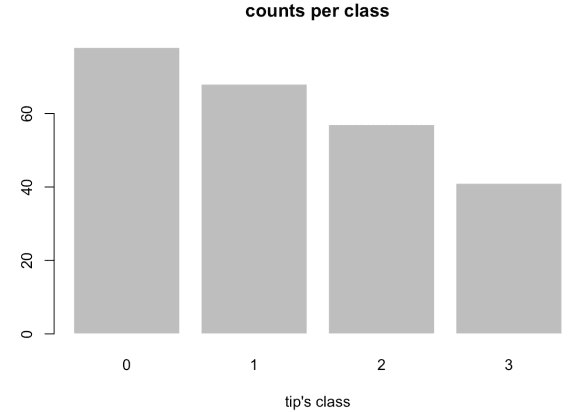
A transformation of tips dataset using a one-hot encoder allows to obtain a dataset with numerical columns at the expense of creating a larger dataset, and to **derive correlations**:



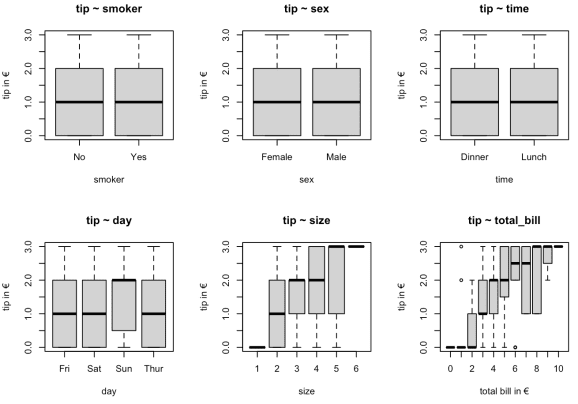
Some correlations mean nothing at all. For example, the correlation between daySat and dayThur or sexMale and timeLunch. The most interesting ones are those between tip and the other variables. Tips in € are more positively correlated with total bills amounts, and with the number of people dining at a table we will categorize our tips in **four classes**:

* **Class 0**: tip in a ]0; 2] € range – **Low**
* **Class 1**: tip in a ]2; 3] € range – **Medium**
* **Class 2**: tip in a ]3; 4] € range – **High**
* **Class 3**: tip in a ]4; 10] € range – **Very high**

We’ll hence be considering a **classification problem**: how to be in class 2 or 3 given the explanatory variables?



**Class 0**, **low tip** contains 78 observations. **Class 1**, **medium tip** contains 68 observations. **Class 2**, **high tip** contains 57 observations. **Class 3**, **very high tip** contains 41 observations. Below, as an **additional descriptive information related to these classes**, we present a distribution of tips (in four classes) as a function of explanatory variables **smoker**, **sex**, **time**, **day**, **size** and **total bill** (with the total bill being segmented according to its histogram breaks):



According to this figure, the fact that the table is reserved for smokers or not, doesn’t highly affect the **median tip**. The same remark holds for the **waiter’s sex** and the **time of the day** when the meals are served (dinner or lunch), which both don’t seem to have a substantial effect on median amounts of tips.

Conversely, **Sunday seems to be the best day for you to work** if you want to maximize your tip. The **number of people dining at a table, and total bills amounts are other influential explanatory variables for the tip**: the higher, the better. But unless you can choose the table you’ll be assigned to (you’re the boss, or his friend!), or are great at embellishing and advertising the menu, your influence on these variables – **size** and **total\_bill** – will be limited.

In section 2 of this post, we’ll study these effects more systematically by using a statistical learning procedure; a procedure designed for accurately classifying tips within the four classes we’ve just defined (low, medium, high, very high), given our explanatory variables.

# 2 – Encoding; cross-validation

**Import Python packages**

import requests

import nnetsauce as ns

import mlsauce as ms

import numpy as np

import pandas as pd

import querier as qr

from sklearn.model\_selection import cross\_val\_score

from sklearn.ensemble import RandomForestClassifier

from tqdm import tqdm

**Import tips**

url = '<https://github.com/thierrymoudiki/querier/tree/master/querier/tests/data/tips.csv>'

f = requests.get(url)

df = qr.select(pd.read\_html(f.text)[0],

'total\_bill, tip, sex, smoker, day, time, size')

**Create the response (for classification)**

# tips' classes = response variable

y\_int = np.asarray([0, 0, 2, 2, 2, 3, 0, 2, 0, 2, 0, 3, 0, 1, 2, 2, 0, 2, 2, 2, 3,

1, 1, 3, 2, 1, 0, 0, 3, 1, 0, 1, 1, 1, 2, 2, 0,

2, 1, 3, 1, 1, 2, 0, 3, 1, 3, 3, 1, 1, 1, 1, 3, 0, 3, 2, 1, 0, 0, 3, 2, 0, 0, 2, 1, 2, 1, 0, 1, 1, 0, 1, 2, 3,

1, 0, 2, 2, 1, 1, 1, 2, 0, 3, 1, 3, 0, 2, 3, 1, 1, 2, 0, 3, 2, 3, 2, 0, 1, 0, 1, 1, 1, 2, 3, 0, 3, 3, 2, 2, 1,

0, 2, 1, 2, 2, 3, 0, 0, 1, 1, 0, 1, 0, 1, 3, 0, 0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 1, 2, 3, 3, 3, 1, 0, 0, 0,

0, 0, 1, 0, 1, 0, 0, 3, 3, 2, 1, 0, 2, 1, 0, 0, 1, 2, 1, 3, 0, 0, 3, 2, 3, 2, 2, 2, 0, 0, 2, 2, 2, 3, 2, 3, 1,

3, 2, 0, 2, 2, 0, 3, 1, 1, 2, 0, 0, 3, 0, 0, 2, 1, 0, 1, 2, 2, 2, 1, 1, 1, 0, 3, 3, 1, 3, 0, 1, 0, 0, 2, 1, 2,

0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 2, 0, 1, 0, 0, 0, 3, 3, 0, 0, 0, 1])

**Obtain a distribution of scores, using encoding**

Here, we use corrtarget\_encoder to **convert categorical variables (containing character strings) to numerical variables**:

n\_cors = 15

n\_repeats = 10

scores\_rf = {k: [] for k in range(n\_cors)} # accuracy scores

for i, rho in enumerate(np.linspace(-0.9, 0.9, num=n\_cors)):

print("\n")

for j in range(n\_repeats):

# Use the encoder

df\_temp = ms.corrtarget\_encoder(df, target='tip',

rho=rho,

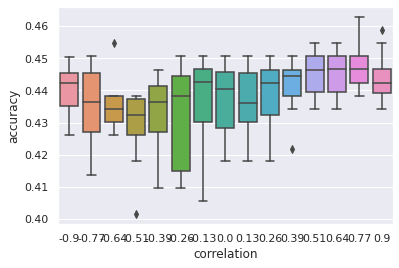
seed=i\*10+j\*10)[0]

X = qr.select(df\_temp, 'total\_bill, sex, smoker, day, time, size').values

regr = RandomForestClassifier(n\_estimators=250)

scores\_rf[i].append(cross\_val\_score(regr, X, y\_int, cv=3).mean())

From these accuracy scores scores\_rf, we obtain the following figure:



**Quite low accuracies… Why is that?** With that said, the best scores are still obtained for high correlations between response and pseudo response. In Part 3 of “Maximizing your tip as a waiter”, **here are the options that we’ll investigate**:

* Compare the correlation-based encoder with one-hot’s accuracy
* Further decorrelate the numerically encoded variables by using a new trick (summing different, independent pseudo targets instead of one currently)
* Consider the use a different dataset if classification results remain poor on tips. Maybe tips is just random?

Your remarks are welcome as usual, **stay tuned!**