**Daily Expenses**

## Assignment

Janne Janssen is a Dutch TU/e student living near Eindhoven. She would like to gain more insight in her income and expenses, so she decided to keep track of all her income and expenses for a period of (a bit longer than) three months. She divided these data into several categories, like food, transportation, study grant, other, etc.. The data can be found in an Excel file that is provided with this assignment. Janne heard that you are currently following the course Stochastics and Simulation: Theory and Applications, so she has high hopes that you can build a sophisticated mathematical model that would help her answer typical questions like:

- What exactly is Janne spending her money on, and how much do these expenses vary per category?
- Sometimes she earns a bit of extra money with her student assistantship (which, by the way, is categorised as “other”), a one-day job, or she gets some allowance from her parents. Is this enough to prevent her bank balance from becoming negative (for too long)?
- Assuming that the income and expenses do not change significantly, is her current financial situation sustainable? Or does she need an additional income to stay financially healthy?
- Would a compound Poisson process be a suitable mathematical model to describe the process of income and expenses? Can this model be used to make predictions for the short-term and long-term future?

Although we would like to give you as much freedom as possible in this assignment to answer these questions, we will give a couple of rules/guidelines and a list of requirements for the assignment.

- The programming part of your assignment consists of (at least) the following parts:
    1. a data analysis where you report all interesting features of the data. Discuss topics like outliers, summary statistics of the several categories income/expenses, distribution fitting of the relevant random variable(s), time between successive items (and the arrival process in general).
    2. A stochastic simulation model that captures the arrival process and the sizes of the various incomes/expenses. Implement multiple variations of the same model, or try multiple different models. This is up to you, as long as one of the models

consists of *one* compound Poisson process for the incomes and *one* compound Poisson process for the expenses.

- Your report should contain a scenario analysis where you create and analyse several what-if scenarios. These can be used to answer typical questions like “What would happen if the study grant would be stopped?”, “What would happen if the public transport fees are raised by 10%?”, etc.. Come up with at least three original scenarios and answer the corresponding questions using the developed simulation model.

Be sure to use all the topics taught in the simulation part of this course, including data analysis, distribution fitting, stochastic simulation, confidence intervals and discussion of number of runs. Admittedly, the data analysis part has not been discussed very extensively in this course, but in Canvas we will upload a Jupyter Notebook which is a quick guide to elementary data analytics and Pandas. Everything you need can be found in this notebook.

## The report

This Assignment will count towards 20% of the final grade of the course 2DMM30. Each group submission should at least contain:

- You should hand in a report in PDF format. The title page must contain: the course name and code, the group number and the name and student numbers of all involved group members.
- For each question, a well-written concise solution containing a simulation description, a result paragraph and a brief conclusion.
- Include 95% confidence intervals for all of your results whenever an expected value is asked.
- Your appendix should contain the full source code of your simulation program in a **text-searchable** format (so no screenshots). Document your code with sufficiently many comments and keep in mind that we have to interpret your code.
- Besides the source code in the appendix, the report should contain a chapter with a simulation description, where you describe (only the main parts of) your simulation in pseudo-code (so no actual Python code). This code should be based on the code we discussed during the lectures, so you are *not* allowed to strongly deviate from this. Even if you solution might be more elegant, it makes it impossible for us to check the correctness of your code. You are *not* allowed any Python library that has not been used during the lectures. If you still would like to use another library, please consult with the lecturer and you might get permission.
- You do *not* have to insert a description of the code for the data analysis part, but it should be included in the appendix.
- In the appendix, add a description of the workload distribution within your group: Who conducted which tasks?

## Using stochastic simulation to answer probabilistic questions

We stress that *simulation* has to be used to answer the main questions (not the data analysis). In particular, this means that *no analytical* solutions are allowed. As a rule of thumb, you can be sure that if you are *not* sampling random numbers, you are *not* using stochastic simulation. But also note that not every solution based on sampling random numbers is a proper stochastic simulation. If there is a stochastic process described in the question, then you have to simulate that exact process without using theoretical knowledge to simplify or modify this stochastic process, even if you know from theory that it would yield the correct result. We will give some examples below. In case of doubt, please consult the lecturer.

We will give some examples of what we mean:

**Question 1:**

Let $X \sim \exp(0.1)$. What is $P(X > 5)$?

**Possible answers:**

- The student gives the following calculation:

  $$P(X > 5) = 1 - (1 - e^{-0.1 \times 5}) = e^{-0.5}.$$

  Wrong answer: because we don't want a theoretical answer, but a Python program. You are always allowed to add this to your report to show that you verified the answer obtained by simulation, but it's not sufficient to only give this theoretical answer.

- The student writes the following Python code:

  ```
  1 - stats.expon(scale=1/0.1).cdf(5)
  ```

  Wrong answer: This is code that does the numerical computation of the exact answer, but no stochastic simulation is used. Note that no random numbers are sampled.

- The student writes the following Python code:

  ```
  mean(stats.expon(scale=1/0.1).rvs(1000000) > 5)
  ```

  Correct answer: Using simulation, sampling random numbers from an exponential distribution and using the law of large numbers, an estimate of the mean is given.

**Question 2:**

Let $Y$ be an exponentially distributed random variable with parameter $\lambda = 0.1$ and let $Z$ be a standard normally distributed random variable, independend of $Y$. Define the random variable $X$ as follows:

$$X = \begin{cases} Y & \text{with probability 0.2,} \\ Z & \text{with probability 0.8.} \end{cases}$$

What is $\mathbb{E}[X]$?

The student writes the following Python code:

```
n = 1000000
EY = np.mean(stats.expon(scale=1/0.1).rvs(n))
EZ = np.mean(stats.norm().rvs(n))
EX = 0.2 * EY + 0.8 * EZ
```

Wrong answer: This answer is is wrong because it simulates $\mathbb{E}[Y]$ and $\mathbb{E}[Z]$ and then determines $\mathbb{E}[X]$ by using the theoretical rule $\mathbb{E}[X] = 0.2\mathbb{E}[Y] + 0.8\mathbb{E}[Z]$. The correct answer involves simulating $X$ according to its definition, which means that first it has to be decided (randomly) whether $X$ is equal to $Y$ or equal to $Z$ and then sample from the right random variable.

Correct answer:

```
n = 1000000
X = np.zeros(n)
for i in range(n):
    u = rng.random()
    if u < 0.2:   # X = Y
        X[i] = stats.expon(scale=1/0.1).rvs()
    else:         # X = Z
        X[i] = stats.norm().rvs()
EX = np.mean(X)
```

Please note that this is not the most efficient way to program this, but it is arguably the easiest to understand.

**Question 3:**

Let $X \sim \exp(\lambda)$. Show that $\mathbb{E}[X] = 1/\lambda$.

**Possible solution:**

The most important thing to realise is that we have not specified any value for $\lambda$. In a simulation, we have to choose a numerical value, though. Unfortunately, picking one value for $\lambda$ and showing that the relation is true for that particular value, is not sufficient. So you should at least pick multiple values for $\lambda$. In general, there are two ways to do this:

1. Systematic: pick some carefully chosen values that (preferably) cover the range of possible values. For example, when varying a probability, you could choose $0.1, 0.5, 0.9$ (corresponding to a low, medium, high value). It's even better if you can use a for-loop to simulate for many different $p$-values between 0 and 1 on a finer grid and display the results in a plot, but this is only possible for relatively fast simulations.

2. Random: instead of selecting specified values, one can also randomly sample parameter values. Please note that this may require more values than just the three from the aforementioned example, to get reliable results.

## (Not) Using ChatGPT / Copilot / ...

As indicated above, you are not allowed to use AI tools such as ChatGPT and Copilot to develop your program and mathematical model. You are also not allowed to use these tools to generate text for your report. The only, very limited, usage of AI tools that we allow, is to help you debug your code, enhance the output or your program, and/or correct and improve your language.

The reason why we opted for this policy is that we find it very important that you learn how to do these tasks yourself and that you understand every step in the process. By letting ChatGPT generate the code for you, you might get correct results, but you can never be sure *and* you might not understand the steps taken to get to this solution. We will discuss a couple of examples of usage of ChatGPT / Copilot that is *not* allowed.

**Distribution fitting.** If you ask ChatGPT to fit suitable probability distributions, it might generate the following code.

```
# Fit several distributions and compare using the Kolmogorov-Smirnov test
candidate_distributions = ['norm', 'expon', 'gamma', 'lognorm', 'beta']
ks_results = {}

for dist_name in candidate_distributions:
    dist = getattr(stats, dist_name)
    params = dist.fit(expenses)
    ks_stat, ks_p = stats.kstest(expenses, dist_name, args=params)
    ks_results[dist_name] = {'ks_stat': ks_stat, 'ks_p': ks_p, 'params': params}

best_fit = max(ks_results.items(), key=lambda x: x[1]['ks_p'])
```

There are several reasons why we do not approve of this code:

- The simple reason that it is generated by AI tools, which is not allowed;
- It includes distributions in the test that do not make sense at all for the given dataset, something you could have known without even trying them first;
- It uses a general `fit` method instead of the manual implementation of the method-of-moments discussed during the lectures;
- Although arguably quite elegant, the last line is unnecessarily complicated, using a lambda function. We prefer simple solutions in this course. Also: you now let the script determine automatically what is the best distribution, but it does not explain *why* this is the best fit and how good or bad the second and third option were. Nor does it consider other criteria than the P-value of the Kolmogorov-Smirnov test.

**Vague interpretation.** Another warning about using AI tools to generate and interpret your results: when asked to interpret the distribution fit, the following text might be generated:

> This distribution suggests that your expenses are positively skewed, with most values clustered at the lower end and a long tail of higher expenses.

Although this might actually be correct, it still is by far not the best way to explain this feature of the data. What does it mean that the expenses are “positively skewed”? And that values are clustered at the lower end and that there is a long tail? Explain your findings in terminology that is understandable and easy to interpret for the reader.

**Mistakes that are difficult to spot.** The following code was generated by Copilot to create a time series plot of the income and expenses per day:

```
data['Day'] = data['Date'].dt.date
daily_data = data.groupby(['Day', 'Income/Expense'])['Amount'].sum().unstack().fillna(0)
daily_data.plot(kind='line', figsize=(12, 6), title='Daily Income and Expenses')
```

Admittedly, the generated plot looks very nice and the solution using `unstack()` and `fillna(0)` seems elegant. However, this code does not take into account that on some of the days there are no amounts (income or expenses) at all. As a consequence, `fillna(0)` does not apply to these days. Nevertheless, the plot suggests that there are positive values on this days. So despite the fact that this plot looks cool, it still is incorrect, but this very difficult to spot and it only became clear to me when I tried to understand the `unstack` and —fillna— commands.

**So is all usage of ChatGPT / Copilot not allowed?** It is disallowed to use these tools to generate parts of your code and to generate text for your report. However, we can think of two possible applications that do not violate this rule:

- You can use it to search for spelling errors and grammatical mistakes in your written text. Of course, we get into a gray area when you are considering to use ChatGPT to make your text better by, for example, asking it to make it more scientific. Tools like ChatGPT tend to either overdo it, or generate text that is too vague. Remember the “positively skewed” example above.
- You can AI tools it for inspiration. If you ask them to analyse the data, it will generate a lot of code that you’re not allowed to use, but it can still inspire you to include certain parts in your own data analysis. Of course, then you have to be careful not to just copy the ChatGPT code, but use the code that we provided as a basis to create your own code to perform that particular piece of data analysis.