Reproduction Package for an artificial business szenario

Sebastian Sippl
OTH-Regensburg

Thomas Brandl
OTH-Regensburg

February 19, 2022

Abstract

This paper deals with a reproduction package for an artificial business scenario. This scenario addresses an optimization problem witch is solved using a Microsoft (MS)-Excel-Plugin called *Solver*. This paper introduces this scenario and represents the results. After that a reproduction package will be prepared to ensure an easy way to reproduce the result from this scenario.

1 Introduction

The Book "Head First Data Analysis" [1] rolls out an artificial business scenario. A manufacturer of rubber ducks and rubber fish wants to maximize their profit. This is done by optimizing the product mix for the quantity of ducks and fish. The profit is based on some variables called constraints and decision variables. All following descriptions refer to the book of "Head First Data Analysis".

2 Variables & Optimization

Constraints are variables which can not be controlled and therefore limit the object you want to optimize. For the production of rubber ducks and fish these constraints can be the available time for the production, the quantity of rubber need for fish and ducks and the profit for one duck and one fish. Decision Variables on the other side are variables which can be controlled. This means the manufacturer can decide how many ducks and fish they will produce. So this are the parameters which can actively be changed.

To maximize or minimize something by changing variables is called an optimization problem. An optimization problem links the constraints, decision variables and the object you want to optimize together into a so called *objective function*. This function is described as

$$c_1 \cdot x_1 + c_2 \cdot x_2 = P \tag{1}$$

where c refers to the constraints, x to the decision variables and P to the object you want to optimize. Regarding to the scenario this function looks like

$$p_d \cdot x_d + p_f \cdot x_f = P \tag{2}$$

where p refers to the profit of one fish (f) and one duck (d) and x to the number of ducks and fish. P is the total profit.

3 Feasible Region & Excel Solver

As described, a constraint limits the profit. For example, the manufacturer has a production time for max. 400 ducks and max. 300 fish. Adding an other constraint restricts the profit again. The manufacturer has a quantity of rubber to produce only 400 ducks and no fish or 300 fish and no ducks. All this constraints together is shown in figure 1. Alone in this little feasible region (see figure 1, green area) there are tones of possible product mixes. A manual calculation of all possible product mixes isn't very efficient. The MS-Excel-Plugin *Solver* [2] helps to calculate the optimal product mix. This *Solver* takes all variables and the objective function to calculate the best product mix. The *Solver* calculates a product mix with the given variables that says to produce 400 ducks and 80 fish with a total profit of 2320$ [1, p. 96].

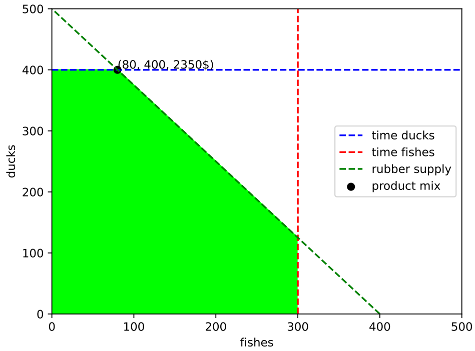

Figure 1: The feasible region.

This model doesn't consider what people will buy. So it is important to take care of sales data from the past. Figure 2 shows some historical sales data given in an MS-Excel-Table (XLS)-format. An assumption is that the soled ducks and fish are negatively connected. According to this, a new constraint can be added to the model that considers prediction of future sales from the given data.

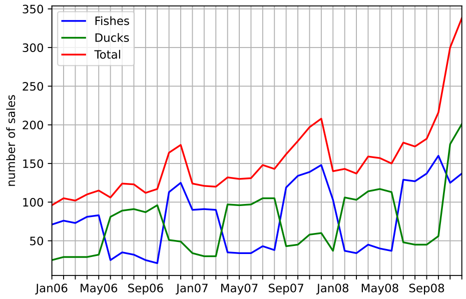

Figure 2: Historical sales data

4 Reproduction Package

The goal of this reproduction package is to ensure an easy way to replicate all results from the upper introduced scenario. Furthermore this package should avoid propriatary tools and it should automatically present the results.

5 Decoupling Dependency

In order to achieve broad availability and therefore good reproducibility of scientific experiments, it is necessary to avoid proprietary software as far as possible. There is an open source alternative for most common tools such as MS-Word, MS-Excel or Matlab which in most cases meets all requirements. Publishing the source code ensures long-term availability of software because there is no conventional manufacturer with a profit making intention of licensing the tools. This ensures that replications don't fail due to missing licenses.

To solve the upper introduced optimization problem the MS-Excel-Plugin *Solver* is used. The *Solver* adjusts the values of the decision variables to satisfy the limits on the constraints and produce the result you want [2]. A Python script that calculates every combination of the decision variables can be used to achieve the functionality of the MS-Excel *Solver*. Constraints can be adjusted via a command line interface to change the limits. Figure 3 shows the feasible region and the optimal product mix which is calculated by a Python script and represents the same result from the book "Head First Data Analysis" [1, p. 107]. Addionally, a prediction constraint (see chapter 3) is added wich says that not more than 150 ducks an 50 fish will be bought next month.

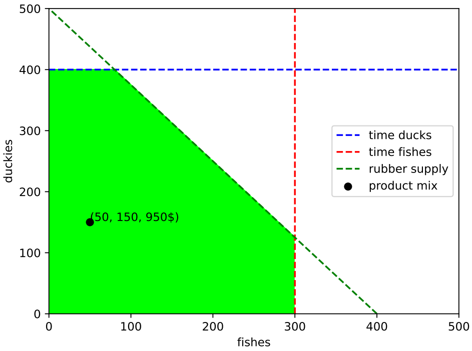

Figure 3: The feasible region using the python script.

The historical sales data that is givn in *XLS*-format leads to another dependency. A light-weighted data format is comma-separated-values (*CSV*), which is a common data exchange format that is widely supported by consumer, business and scientific applications. Furthermore, CSV files consist of plain text which can still be processed many years from now without a licensed tool. To document the results the open source program LaTeX is used which offers possibilities to automate the creation of documents. Figure 1 represents some results of the the book "Head First Data Analysis" and is created by a Python script that uses the open source library *Matplotlib* [4]. Figure 2 is generated from the same Python script based on the sales date stored in *CSV*-format. Figure 3 is generatet by the Python script which implements the *Solver's* functionality.

6 Docker

Docker is an open source platform for developing, shipping, and running applications. It provides the ability to package and run an application in a isolated environment called a *container*. Unlike virtual machines (*VM*), containers do not bundle a full operating system – only libraries and settings required to run the application. So containers can easily shared and the receiver gets the same container that works in the same way.

An *image* is a read-only template with instructions for creating a Docker *container*. To build an image a *Dockerfile* defines the steps which are needed to create the image. A *container* is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API. By default, a container is relatively well-isolated from other containers and its host machine [3].

In this reproduction package the container provides the hole experimental setup including the data in *CSV*-format, all Python source codes and the documentation based on LaTeX. During the build-process of the container, all required tools and libraries are installed and all artefacts are copied into the container. The Python scripts for generating the requiered data are executed via a shell script after the container has started. Finally, this document is generated automatically and copied from the container to the host PC. The container can also be started in interactive mode to retrace the functionality of the Python-Solver.

7 Github & Zenodo

In order to make the development of the project traceable, all changes are documented in a Git repository and uploaded to the website *GitHub.com*. In addition, it is possible to work on a project with several people at the same time. However, GitHub is not suitable for long-term documentation of research results because there is no guarantee that the repository will remain available for decades to come. For this reason, *Zenodo* is used. *Zenodo* [6] is financed by public funds from the EU and is there to keep research results available over the long term. With a clear Digital Object Identifier (*DOI*), other researchers can also refer to these publications and use them as a source in their work.

References

[1] Michael Milton, "Optimization: Take It to the Max," in *Head First Data Analysis*, O'Reilly Media Inc, 2009, pp. 75-109

[2] Microsoft, 2022, `https://support.microsoft.com/en-us/office/define-and-solve-a-problem-by-using-solver-5d1a388f-079d-43ac-a7eb-f63e45925040` (accessed on 03.02.2022)

[3] Docker Docs, 2021, `https://docs.docker.com/get-started/overview/` (accessed on 13.02.2022)

[4] The Matplotlib Development team, 2021 `https://matplotlib.org/` (accesed on 17.02.2022)

[5] GitHub, Inc., 2022, `https://github.com/` (accesed on 17.02.2022)

[6] CERN Data Centre Invenio., 2022, `https://zenodo.org/` (accessed on 17.02.2022)