Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

geoinformation engineering
ikg eth zurich

Spring Semester 2019

The future of logistics: how will electric trucks shape the future energy demand and distribution of Switzerland?

Final Report for
227-1671-10L Semester Project
MSc. Energy Science and Technology (MEST)

Supervisors:
Prof. Dr. Martin Raubal
Dominik Bucher
Dr. Rene Buffat
Henry Martin

Date: July 11th 2019

This report is written by:
Keyuan Yin (17-945-957)

This page is intentionally left blank.

Abstract

Driven by climate change, carbon emissions from transportation sectors have to be reduced. Many studies have covered electric cars. Nevertheless, more than half of the ground logistics stems from road transportations in Switzerland. This study assessed the technological feasibility of replacing internal combustion engine (ICE) trucks with currently available battery electric trucks (curb weight above $3.5\ \mathrm{t}$) by analyzing acquired data of transportations carried out by Switzerland's trucks.

By routing planning, constructing elevation profiles, and using analytical energy models with different efficiencies and utilization schemes, total energy demand patterns on annual and each postal code's basis have been derived.

A full deployment of battery electric trucks (BET) will only pose 2.15% to 2.4% of Swiss total electricity end-consumption, and estimated carbon emissions can be reduced by 1.9% of total Switzerland's carbon emissions. However, with current infrastructures on local municipalities with loose populations, local grids may need major optimizations and upgrades to meet future charging demands. Our study suggests that 71.24% to 81.77% of the Swiss ICE truck fleet can be replaced by electric trucks with currently available solutions. To achieve an electrification of 85% the truck fleet, a minimal battery capacity of $258\ \mathrm{kW\,h}$ should be deployed.

This page is intentionally left blank.

Acknowledgment

I would like to thank Prof. Dr. Martin Raubal's group for providing this topic and supporting my study for opening this topic as a semester thesis. Specifically, I would like to thank my three supervisors, Dominik Bucher, Dr. Rene Buffat, and Henry Martin, for their constant and timely supports throughout the entire project.

I would also like to thank my current working company, Landis+Gyr, that I have been working with for the last two and half months. While working, the related knowledge in electric vehicles and smart grids did play an essential role for me to comprehend the electrification of the logistics sector in Switzerland.

In the last, I would like to express my thanks to Frederic Schuh and Zhe Zhu from Daimler with their insights into electric trucks and loads of interesting and useful information.

This page is intentionally left blank.

Contents

List of Figures

List of Tables

This page is intentionally left blank.

List of Terms

B

BET

Battery electric truck.

C

COP

Conference of the Parties, is the governing body of an international convention.

D

DEM

Digital elevation model.

DFI

Federal Department of Home Affairs (Département fédéral de l'intérieur in French).

DSO

Distribution system operator.

E

EPAL

European Pallet Association.

ETH

Eidgenössische Technische Hochschule Zürich, Swiss Federal Institute of Technology in Zurich in English.

EV

Electric vehicle.

F

FEON

Federal Office for the Environment.

FSO

Federal Statistical Office (Office fédéral de la statistique in French; Bundesamt für Statistik in German).

G

GTE

Normal commercial transportation survey for trucks (above 3.5t).

I

ICE

Internal combustion engine.

IEA

International Energy Agency.

L

Li

Lithium, is a chemical element with atomic number 3.

LWE

Light weight transportation vehicle.

M

MCV

Manufacturing commercial vehicle.

O

OSRM

Open Source Routing Machine.

S

SFOE

Swiss Federal Office of Energy.

SOC

State of charge.

T

TFC

Total final consumption.

V

V2G

Vehicle-to-grid.

This page is intentionally left blank.

Chapter 1

Introduction

This chapter describes the necessity to change the energy source of the transportation sector, currently available technologies, and past studies that are relevant to our research.

1.1 Background information

《 At the Paris climate conference (COP21) in December 2015, 195 countries adopted the first-ever universal, legally binding global climate deal.》 (European Commission 2016). The success of the Paris agreement indicated more evidence on climate change, urging more effective responses from different parties. Schmidt & Huenteler (2016) also pointed out that Paris paradigm leads to further coordinations between climate policies, energy policies, and industry policies. Hence, we see changes in the logistics sector to be impending.

《 On 28 November 2018, the Commission presented its long-term strategic vision for a prosperous, modern, competitive and climate-neutral economy by 2050 in line with the Paris Agreement objective to keep the global temperature increase to well below 2°C and pursue efforts to keep it to 1.5 °C.》 (European Commission 2018*a*).

To achieve the above-mentioned target, it is every industrial sector's responsibility to take effective actions. In the logistics sector, more strict emission standards have been released to limit CO_2 emissions within heavy-duty trucks. The EU has proposed that heavy duty vehicles in 2025 should have 15% lower carbon emissions than those produced in 2019 (European Commission 2018*b*).

It is inevitable that the amount of internal combustion engine(ICE) trucks will be reduced in order to meet the future emission and climate target worldwide. With such an external environment, Switzerland must act as well. Switzerland Energy Strategy 2050 has proposed focuses on increasing the energy efficiency by electrifying the heating and transport sectors and on increasing the share of renewables by deploying solar and seasonal storages(Swiss Federal Office of Energy (SFOE) 2018).

1.2 Current energy consumption and truck transportations of Switzerland

《Petroleum and other fuels are the main sources of energy in Switzerland (50.6%), followed by electricity (25%), gas (13.5%) and wood (4.4%) (Swiss Federal Statistical

Office (FSO) & Swiss Federal Office of Energy (SFOE) 2016).⟫

In order to meet the above-mentioned climate target, a change of energy sources for transportation sectors is necessary.

As suggested by Swiss Federal Statistical Office (FSO) (2018), Switzerland's ground transportations can be split into road and railway transportations. In these two categories, 17.22 billion ton-kilometers were from the road and 10.07 billion ton-kilometers were from the railway. Trucks (curb weight above 3.5 tonnes) and vans (curb weight below 3.5 tonnes) were two groups contributing 17.22 billion ton-kilometers in 2017. Most of trucks and vans have diesel as their energy sources, and they are referred as internal combustion engine trucks (ICE trucks). In 2017, Switzerland had around 54,000 trucks and 363,000 vans. Although the fleet size of trucks was much smaller, trucks could carry much higher weights of cargo compared to vans. Trucks in 2017 covered 2.2 billion kilometers. With an assumption that a truck carried 5 tonnes of cargo on average, 54,000 trucks contributed more than half of Switzerland's ton-kilometers on road transportations or 40% of Switzerland's ton-kilometers on ground transportations. Hence, a fully-electrified truck fleet in Switzerland can contribute to the reduction of both EU-wide heavy-duty truck emissions and Switzerland's carbon emission significantly.

1.3 Current electric truck development

With the above-mentioned amount of transportations contributed by trucks in Switzerland, it is also essential to discuss currently available technologies on electric trucks to investigate their potentials to be deployed in Switzerland.

Hydrogen fuel cell trucks, battery electric trucks, and other industrial solutions of electric trucks had been proposed. However, hydrogen fuel cell trucks are currently constrained by hydrogen's production rate and cost, which ranges from 12 to 16 $ per kg, and other industrial solutions, such as Siemens's eHighway project, are still at the pilot stage (Siemens 2019*a*, California Fuel Cell Partnership 2019). At the current stage of development, the Li-based battery group is still considered as the most promising green power source for the transportation sector as it has a high specific energy (up to 2000 $\mathrm{W\,h\,kg^{-1}}$) and a high specific power (Van den Bossche et al. 2006).

Figure 1.1 shows that the Li-ion battery group has the largest ranges of specific power and specific energy when it is compared with other currently existing technologies. Furthermore, as suggested by Gerssen-Gondelach & Faaij (2012), the learning rate of Li-ion battery is around 16% to 17%. Such a learning rate indicates that a 16% to 17% cost reduction is expected when the output amount of Li-ion battery has been doubled. Gerssen-Gondelach & Faaij (2012) also estimated that beyond 2025, the specific energy and specific power of Li-ion battery would develop to 250 $\mathrm{W\,h\,kg^{-1}}$ and 500 $\mathrm{W\,kg^{-1}}$ respectively, with an efficiency of 95% and a prolonged lifetime of 12 years.

Such a trend of battery development also accelerates the development of battery electric trucks, in short BET (Earl et al. 2018, Market Study Report, LLC 2019, QY Research 2019).

Currently, traditional truck producers, such as Daimler, DAF, and Scania, have been disclosing their most updated information on electric truck developments with various loading capacities for different purposes (Daimler 2017, Scania 2018, DAF 2019). Battery companies and also passenger-focused EV companies, like Tesla and BYD, also

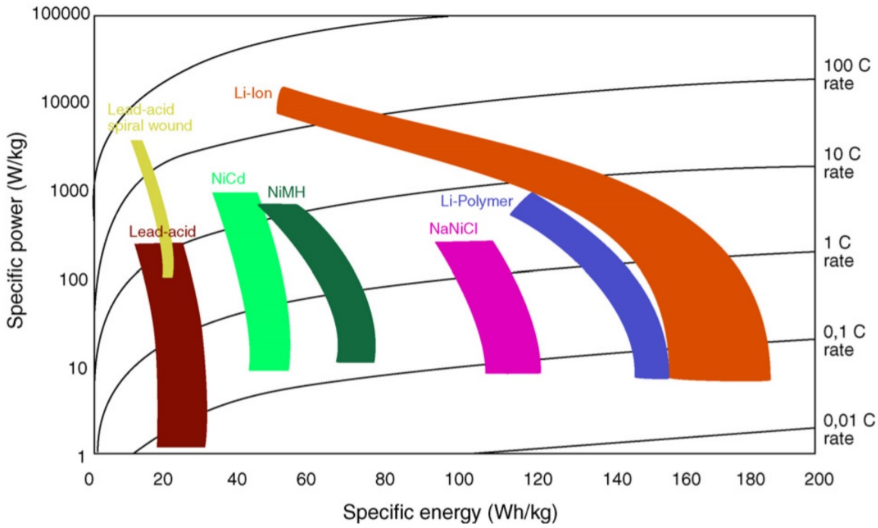


Figure 1.1: Ragone plot of different battery technologies (Van den Bossche et al. 2006). The Li-ion battery group has the largest ranges of specific power and specific energy.

released their products in commercial logistics sectors (Tesla 2019, BYD 2018*b*). Therefore, with currently available technologies and available information, it is possible to assess the feasibility of deploying electric trucks in Switzerland.

1.4 Relevant studies

Recently, several studies have discussed the possibility of electrifying truck transportations of Switzerland. Liimatainen et al. (2019) analyzed the energy demand of the Swiss truck transportation in 2016 with the simplified energy model under several electrification scenarios, such as different charging powers and different specific energy density of batteries, without spatial analyses. They concluded that roughly 71% of trucks that were registered in Switzerland in 2016 could be replaced by electric trucks. On the contrary, Çabukoglu et al. (2018) concluded that before going fully electrification of trucks of the Swiss logistics sector, three conditions have to be met: an allowance to exceed current maximum permissible weight regulations, a high-capacity grid access for charging at the home-base (at least 50kW), and a supporting intra-day energy infrastructure. However, the study assumed that the volume of battery packs in BET is the same as the volume of fuel tanks in ICE trucks. Such an assumption may not be suitable because simpler power train structures of BET should be able to provide more spaces for battery packs.

Going beyond the scope of Switzerland, McLane & Mullaney (2019) has illustrated how Shenzhen, a megacity in China, has made 60,000 commercial transportation vehicles deployed within the recent 3 years by analyzing the political supports behind. Yang et al. (2018) found that light-duty plug-in electric trucks exhibit the best performance in terms of cost saving and greenhouse gas emission reduction via analyzing life cycle assessments on diesel, plug-in electric, and battery-swap electric commercial vehicles.

Many studies and real cases indicate that the electrification of commercial vehicles is possible. Compared to previous studies, what we have planned is to apply an analytical energy model to routing-machine-generated elevation profiles and travel distances for the obtained data. On the BET side, we select commercially available solutions of nowadays to match the transportation demands for different drivers. Such a bottom-up approach should mimic a realistic scenario of truck transportations in Switzerland in terms of energy needs; therefore, it provides more insights into the potential of electrifying Swiss logistics sectors.

1.5 Structure of the report

This report is organized as follows:

- Chapter 2 describes the data, the pre-processing for the data, and the validation process of the data. Then, an analytical energy model will be presented. Current BET solutions will also be introduced. Furthermore, different BET efficiency schemes of battery electric trucks will be discussed. Lastly, methods in route planning and elevation profile retrieving will be presented.

- Chapter 3 will present results from the perspective of the potential impact of the total energy demand in Switzerland on both national and local levels brought by the full deployment of BET, and from the perspective of the percentage of ICE trucks can be replaced by BETs.

- Chapter 4 will give conclusions that can be drawn from this study and some outlooks based on our findings.

Chapter 2

Methodology

In this chapter, the used data and its associated handling procedures will be described. An energy model for transportations will also be introduced. Based on the energy model, the total energy demand of truck-based transportations can be derived, and different scenarios will be discussed.

2.1 Data description

Provided by Territory and Environment Division of Federal Statistical Office of Federal Department of Home Affairs(DFI-FSO), we have used 4 sets of data:

- Domestic light commercial vehicle transportation (curb weight below 3.5t) data from 2013 (***LWE13***). Selected important items with descriptions can be found in Table 2.1.

- Domestic normal commercial transportation data from 2013, 2014, and 2017 (***GTE13***, ***GTE14***, and ***GTE17***) from domestically registered trucks that are heavier than 3.5t. Selected important items with descriptions can be found in Table 2.2.

Selected item name	Description
OID	ID of the vehicle.
CURB_WEIGHT	Net weight of the curb.
LOADING_CAPACITY	Maximum loading weight of the vehicle.
LADEN_WEIGHT	Weight of cargo loaded in the vehicle.
HAULED_LOAD	Weight of cargo hauled at the back of the vehicle.
Quelle_NPA	Postal code of transportation origin.
Ziel_NPA	Postal code of transportation destination.
Nr. of entries	14191

Table 2.1: Selected items from ***LWE13*** (Light commercial vehicle transportation data)

It should be noted that 4 sets of data are records for domestically registered vehicles, thus cross-border transportations for domestically registered vehicles were also recorded.

For ***GTE13*** and ***GTE14***, descriptions of surveying methods were provided by Bundesamt für Statistik BFS (2015). The documentation described that each week, roughly

Selected item name	Description
ernr	Id of the vehicle, connecting 3 files together.
	transport.csv
tonCH	Weight of cargo excl. pallets/packagins in kg.
fromPlz	Postal code of origin of a transportation.
toPlz	Postal code of destination of a transportation.
transportKmCH	Distance travelled domestically in km.
transportKm	Total distance travelled in km.
	week.csv
week	The week of the vehicle being surveyed.
	journey.csv
journeyTonkmTot	Ton-kilometer of the entire journey.

Table 2.2: Selected items from ***GTE*** (normal commercial transportation data)

170 surveys were sent out to drivers for recording one week's transportations. Therefore, in the entire year, 8,500 surveys were sent out in total. However, DFI only took 53% of the total surveys that were sent out because of poor quality, denial, or not responding. It was also stated that the amount of transportation trucks heavier than 3.5 t in Switzerland had been staying around 57,000.

GTE17 was covered in another report (Swiss Federal Statistical Office (FSO) 2018). ***GTE17***'s contents are very similar to ***GTE13***'s and ***GTE14***'s except suggesting a truck fleet size of nearly 54,000. Hence, the assumptions and conditions applied to ***GTE13*** and ***GTE14***, in our studies, are also applied to ***GTE17***. All 3 datasets have roughly 100,000 transportation entries, with 4,000 to 5,000 drivers/vehicles being recorded. As mentioned before, each of the drivers was required to fill the questionnaire for his/her entire week's transportation.

Information in ***GTE13***, ***GTE14***, and ***GTE17*** that includes an ID for each driver, secondary ID for each transportation of the same driver, the origin of transportation (in postal codes), the destination of transportation (in postal codes), the distance covered in each transportation, the ton-kilometer recorded by drivers for each journey, the weight of transported goods excl. packaging and pallets, the time survey being conducted (in week, from 1 to 52), etc. is not available within a single file but was split into **'transport.csv'**, **'journeych.csv'**, and **'week.csv'**. A small selection of used items in our study can be found in Table 2.2. Our methods will mainly take records of **'transport.csv'** in order to stand on the conservative side. Such an approach is conservative because it considers every transportation as an individual journey without optimizing routes for several transportations into one aggregated journey. **'journeych.csv'** is mainly used for validation purposes as it contains ton-kilometer information.

2.2 Data pre-processing

Our study mainly focused on trucks' commercial transportations in 2013, 2014, and 2017 (***GTE13, GTE14, GTE17***) because of following constraints:

- The available elevation profile for routing calculation and energy demand calculation only covers Switzerland and small parts of its neighboring territories.

Therefore, route's elevation changes cannot be processed with foreign countries. Instead, for foreign transportations, we apply the simplified flat surface energy model (illustrated in section 2.4) with distance reported by the driver within the dataset.

- The survey was conducted with geo-locations determined by postal codes. Coordinations of postal codes within Switzerland are well-defined via Swiss Post database; however, postal codes of foreign countries either have the problem of unable to find an official source or have the obstacle to obtain coordinates in a systematic way, leading to a manual search of many foreign codes.

- ***LWE13*** contains data with less granularity. The documentation of ***LWE13*** also contains fewer details compared to ***GTE13***, ***GTE14***, and ***GTE17***. For instance, the length of the survey to a driver is only one survey day with the exact date unknown, giving a higher uncertainty. Albeit ***LWE13***'s lack of granularity, ***LWE13*** provided us with very good supporting materials helping us building assumptions on trucks' transportations, such as determining ton-kilometers of trucks etc.

For ***GTE13***, ***GTE14***, and ***GTE17***, pre-processing of the data is necessary for further energy-related calculations. Since only Switzerland's elevation profile is available as before-mentioned, the elevation profile of generated routes can only be obtained within Switzerland's realm. Therefore, pre-processing was conducted to split **'transport.csv'** into domestic and foreign transportations. Followed by the first step, additional pre-processing is appending the **'week.csv'**'s time information to **'transport.csv'** and **'journeych.csv'** by matching drivers' IDs.

After the integration of different subsets (**'week.csv'**, **'transport.csv'**, **'journeych.csv'**) into one aggregated file for each (***GTE13***, ***GTE14***, and ***GTE17***), wrong entries of postal codes in destinations and origins need to be adjusted. For instance, some drivers filled the postal codes that do not exist. Wrong entries will not be moved out from the existing datasets but will be referred to the geographical center of Switzerland. Since wrong entries of postal codes are rare in each year's survey (< 30 entries), the influence of referring them to the geographical center of Switzerland can be ignored. With all pre-processing steps conducted, our data is clean enough to be used for further research.

2.3 Preliminary studies and validation of the data

Before calculating routes, distances, and elevation changes based on origins and destinations provided in the dataset, we carried out several preliminary studies and validation studies for the data provided.

The first preliminary study was based on each transportation entry for the distance traveled. Histograms and associated box-plots can be obtained as shown in Figure 2.1. Since ***GTE13*** and ***GTE14*** have very similar distributions as ***GTE17***, the same plots for ***GTE13*** and ***GTE14*** are only provided in Appendix A. From the distribution of transportation distances, we can very clearly observe that more than 75% of the transportation activities are shorter than $100\mathrm{km}$ in terms of travel distances.

Ton-kilometer is a parameter simply by multiplying a transportation's travel distance in km with this transportation's cargo weight in t. Swiss Federal Statistical Office

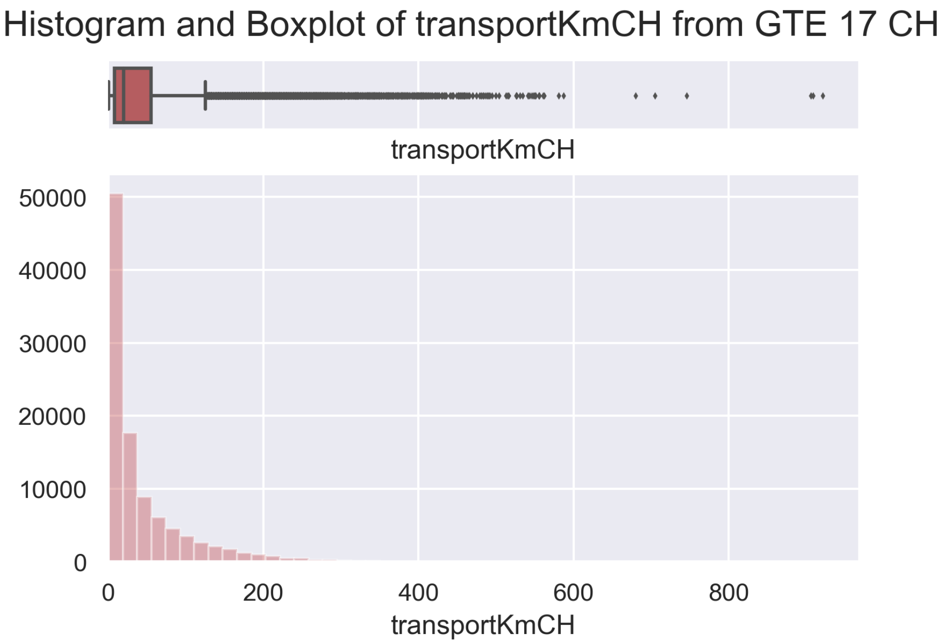


Figure 2.1: Histogram and box-plot for distribution of domestic ***GTE17*** transportation distances (provided by drivers).

	GTE13	GTE14	GTE17
ton-kilometer from trucks in billion ("transport.csv")	11.47	10.54	9.49
ton-kilometer from trucks in billion ("journey.csv")	10.21	10.01	9.15
ton-kilometer from all transportation vehicles in billion	NA	17.54	17.22
Percentage of ton-kilometer from trucks in %	NA	57-60%	53-55%

Table 2.3: Ton-kilometers of road transportations in 2013, 2014,and 2017 in Switzerland. Swiss Federal Statistical Office (FSO) (2016, 2018) provided the data of ton-kilometer from all vehicles in billion.

(FSO) (2016) and Swiss Federal Statistical Office (FSO) (2018) suggest that 17.54 billion and 17.22 billion ton-kilometers occurred through the road transportation in 2014 and 2017 respectively, including vehicles both above and below 3.5 t.

By multiplying the kilometers with the loading weights in the datasets, we get the total ton-kilometer for all vehicles in our datasets. Swiss Federal Statistical Office (FSO) (2015, 2018) have provided the total registered number of trucks in Switzerland. From the sum of ton-kilometers in our datasets, we can derive the ton-kilometers for the entire truck fleet in Switzerland by applying the ratio of the amount of surveyed vehicles to the amount of all registered vehicles. Table 2.3 also suggests that there is a decreasing trend of road transportations coming from trucks in the period from 2013 to 2017. We see that 53% to 60% of the road transportations were carried out by trucks. It should also be noted that considering each transportation as a single journey yields higher ton-kilometers, indicating a more conservative estimation of the total energy demand of the truck transportation sector. Light vehicles are also an important part of Switzerland's logistics sector. In 2015, light vehicles had total driving distances of 4.1 billion km, which is almost twice as the total driving distances of trucks (2.2 billion km). By using ***LWE13*** dataset, we can have a rough idea on the average loading weights from light vehicles. When considering the weight of trailers, the average loading weight of each light vehicle for each transportation was 3 t. If only the loading weight inside the vehicle is considered, the average loading weight of each light vehicle for each transportation was 1 t. Therefore, total ton-kilometers contributed by light vehicles can vary from 4.1 billion ton-kilometers to 12.3 billion ton-kilometers. The total ton-kilometers therefore for 2013 can vary from 15.57 ton-kilometers to 23.77 ton-kilometers. When comparing this range to total ton-kilometers of ***GTE14*** and ***GTE17***, we can see that both of them locate within the range that we calculated from ***LWE13*** and ***GTE13***. Therefore, we can validate that our data and the assumption of the total number of vehicles for each year is good enough for more detailed analyses with energy models and route calculations.

2.4 Energy model

During this study, an analytical solution for truck's energy demand has been used with an assumption that all transportations have a constant velocity of 70 $\mathrm{km\,h^{-1}}$. Suggested by Earl et al. (2018), for trucks, two main forces that work against the forward motion are the aerodynamic drag force and the tyre rolling resistance. The analytical solution to the total force a truck is facing during moving forward can be described by Equation 2.1. The graphical illustration of this analytical energy model is shown in Figure 2.2.

$$F_{RL-flat} = \frac{1}{2} \cdot \rho \cdot v^2 \cdot H \cdot W \cdot C_D + m \cdot g \cdot C_{RR} \tag{2.1}$$

where:
$F_{RL-flat}$ = total resistance forces faced by the truck on flat surface in N
ρ = 1.2 $\mathrm{kg\,m^{-3}}$, the density of air
v = truck speed in $\mathrm{m\,s^{-1}}$
$H\&W$ = height and width of truck front surface in m
m = gross vehicle weight in kg
C_{RR} = rolling resistence coefficient, determined by tyre type
C_D = drag coefficient, determined by truck's aerodynamic characteristics

However, for most of the trucks and other types of vehicles, transportations do not solely happen on flat roads. An additional equation is necessary to calculate energy demand on inclined roads, which include both upslopes and downslopes. A previous master thesis that was hosted at Geoinformation chair at ETH Zurich focusing on optimization of e-Bike operational ranges suggested adding the gravitational factor and the angle of inclination into the energy model calculation (Haumann 2017). Accordingly, the force resistance equation will include the gravitational component and the angle of inclination as shown in Equation 2.2 and Figure 2.3.

$$F_{RL-inclined} = \frac{1}{2} \cdot \rho \cdot v^2 \cdot H \cdot W \cdot C_D + m \cdot g \cdot C_{RR} \cdot cos(\alpha) + m \cdot g \cdot sin(\alpha) \tag{2.2}$$

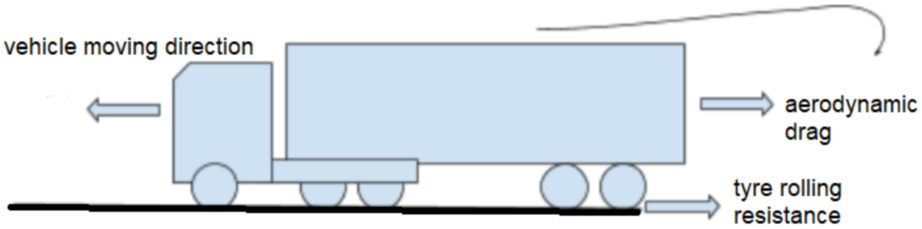


Figure 2.2: Flat surface truck analytical energy model includes main resistive forces to forward motion from the simplified road load equation, modified from Earl et al. (2018).

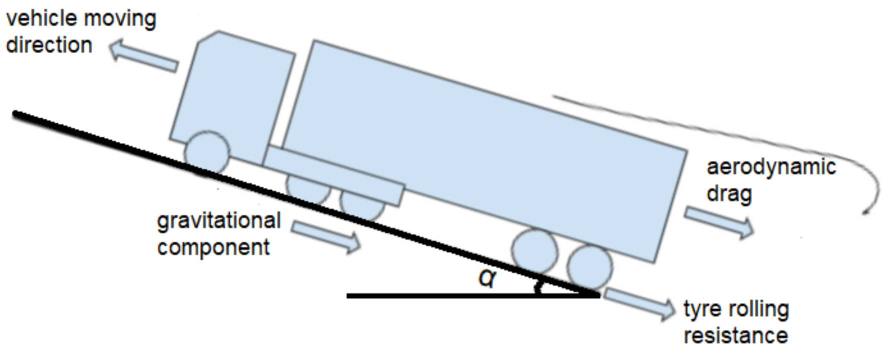


Figure 2.3: Inclined surface truck analytical energy model includes main resistive forces to forward motion from the flat surface equation, and gravitational component, modified from Earl et al. (2018). α indicates the angle of inclination.

where: $F_{RL-inclined}$ = total resistance forces faced by the truck on inclined surface in N
α = angle of inclination

$$E = F_{RL} \cdot distance \tag{2.3}$$

where: E = energy needed to move the truck for a certain distance in J

With the resistance of moving in both flat and inclined conditions has been defined, an energy demand can be derived by simply applying Equation 2.3 when truck is moving in constant velocity, which is $70\mathrm{km\,h^{-1}}$ through our study. As alpha can be negative when going downslope and yield a negative resistance force, it is possible to generate electricity for vehicle's further power consumptions. In our studies, if negative energy, standing for an energy gain from the transportation, is obtained through one transportation, we consider the energy consumed for such a transportation is $0\ \mathrm{kW\,h}$. The reason for having such an assumption is that ⟪ vehicle-to-grid (V2G) plays a limited role in improving the penetration of renewables in the literature, most likely due to excessive battery degradation which results in a relatively high cost of providing V2G power⟫ (Richardson 2013). Such a simplification also allows easier applying different efficiencies of the energy demand as the efficiency calculation is always a division with a number between 0 and 1. This kind of calculation makes sense when the number is positive, meaning costing more energy. However, when the number is negative, the calculation that stands for lower efficiency actually generates more energy. Therefore, to stand on the conservative side, taking potential energy gain as zero would produce less optimistic but more realistic results.

It should also be noted that electric vehicles can retrieve kinetic energy while braking. Such a feature is called regenerative braking. The efficiency of regenerative braking for electric trucks is ill-defined. Besides, as our model and assumption will be based on constant moving electric trucks, regenerative braking does not play a role in the energy consumption calculations.

2.4.1 Cargo-to-pallet weight ratio

The actual loading weights, which are applied to our energy model, were derived from the multiplication of the recorded loading weights in the datasets by 1.05 as the recorded loading weights that were entered by the drivers did not consider the weights of pallets or other packages. Standard European pallet, as specified by the European Pallet Association (EPAL), is called EUR-pallet. A normal EUR/EPAL-pallet (EPAL 1) is approximately 25kg in weight, and it has the maximum load capacity of 1500kg (European Pallet Association e.V. 2019). Therefore, at the maximum load conditions, the cargo-to-pallet weight ratio is 1.67%. However, the maximum load is not always the case in reality. To consider a safety margin, we consider a 5% cargo-to-pallet weight ratio.

2.5 Current battery electric truck solutions

Within the scope of this research, BETs that are produced by BYD, a Chinese leading EV and battery company, are considered as the potential replacements of current ICE trucks. The reasons are the following:

- BYD has a comprehensive fleet of commercial full-electrical vehicles with loading capabilities ranging from 3.5 t to 44 t (BYD 2018*a*).
- BYD's products have detailed technology parameters and information.
- BYD's products are currently already in the market, and they can be reckoned as the current commercially available options for us to directly apply to the logistics sector.

We chose BYD's BET fleet as the replacement of current ICE trucks from datasets ***GTE13***, ***GTE14***, and ***GTE17***. Since in these datasets, all vehicles that are heavier than 3.5 t had been considered; it would not be reasonable to consider all vehicles having the same load capacity. Therefore, for drivers with their maximum loading weights no heavier than 2890kg, BYD 7.5 TON MCV is considered as their BET replacements. For drivers with their maximum loading weights no heavier than 4615kg, BYD 11 TON

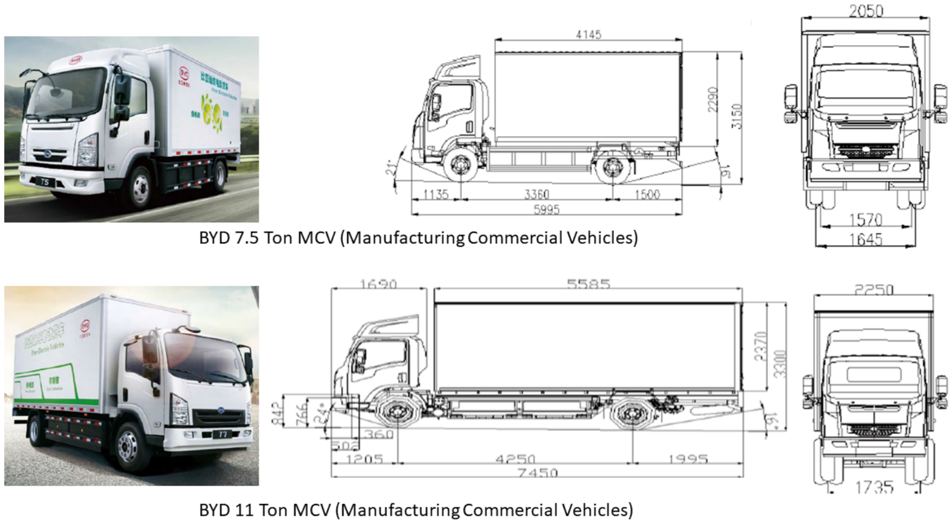


Figure 2.4: BYD's 7.5 TON MCV and 1 TON MCV dimension and appearance, modified from BYD (2018*a*).

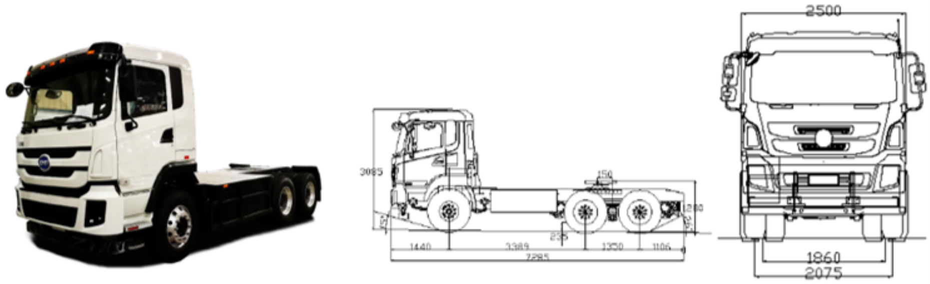


Figure 2.5: BYD's Q3M Dimension and Appearance, modified from BYD (2018*a*).

MCV is considered as replacements. The appearances and dimension parameters of BYD 7.5 TON MCV and 11 TON MCV are shown in Figure 2.4. With the maximum loads heavier than 4615kg, BYD Q3M is considered as the replacement (see Figure 2.5). BYD Q3M has the maximum loading capacity of roughly 37 t. Three BETs' technical parameters can be found in Table 2.4 . The most important pieces of information obtained from the technical parameters will be the height and width of the truck front, the battery capacity, and the charging time in hours. A minimal 1.5h of charging time does not allow recharging during the transportations. Hence, based on the current recharging times, we assume that the recharging activities would only happen before transportations start and no recharging activities would happen during the transportations.

By applying the analytical energy model to ***GTE13***, ***GTE14***, and ***GTE17*** with BYD trucks' parameters, each transportation's theoretical energy demand can be obtained. Furthermore, after grouping the total energy demand by each driver's ID in the dataset, the amount of energy demand for each truck on a daily basis can be obtained. From the supply side, BYD trucks' battery capacities will be compared to the demand side, which is each driver/vehicle's daily transportation energy demand.

L×W×H (mm)	Curb Weight (kg)	GCWR(Kg)	Top Speed(km/h)	Range(Km)	BatteryCapacity (kWh)	Charging Time (h)
5995×2050×3150	4300	7320	100	>250	145	AC4h/DC1.5h
6630×2250×2360	5885	10695	100	>180	148.5	AC4h/DC2h
7285×2500×3085	10500	47627	90	>160	207	AC2.6h

Table 2.4: Technical parameters of BYD 7.5 TON MCV (row 1), 11 TON MCV (row 2), and BYD Q3M (row 3)

2.6 Efficiency of battery electric trucks

Our energy model does not consider any efficiency loss, which happens from EV's charging stations to the final power delivered to the wheels. In another word, our energy model is the theoretical minimum amount of energy in need to move the BET forwards. The general efficiency of BET from the charging station to the rotating wheels on the roads needs to be defined to provide more precise and accurate suggestions.

Following sections provide several available sources on general efficiencies and their decomposed values. All different efficiency schemes will be considered as different scenarios for calculating BETs' realistic energy consumptions.

2.6.1 Efficiency scheme 1

Earl et al. (2018) suggests a general efficiency of 85% as shown in Table 2.5. In the table, it also shows the comparison with current ICE trucks in terms of the general drive-train efficiency, which can be helpful for us to determine the total CO_2 emission after the total energy demand is derived.

Efficiency	Diesel ICE	BET
AC/DC rectification	-	95%
Battery charging/running	-	95%
DC/AC inversion	-	95%
Engine operation	39% 46% (best in class)	95%
Transmission	95%	99%
Total drive-train	37%	85%

Table 2.5: Drive-train efficiency comparison between ICE and BET, modified from Earl et al. (2018).

2.6.2 Efficiency scheme 2

The second suggestion of the general electric truck's drive-train efficiency was kindly given by Frederic Schuh and Zhe Zhu from Daimler (Schuh & Zhu 2019). The values are presented in Table 2.6.

2.6.3 Efficiency scheme 3

An organization, Transport and Environment, published its research on general efficiencies for three promising carbon emission-free transport technologies, including

Efficiency	General electric trucks
AC/DC conversion	98%
Battery charging/running	95%
DC/AC inversion	98%
Motor	92%
Transmission (Gearbox)	96-97%
Total drive-train	81%

Table 2.6: Drivetrain efficiency scheme 2 (Schuh & Zhu 2019)

Efficiency	Battery Electric Vehicle
Inversion AC/DC	95%
Battery charging/running	95%
DC/AC inversion	95%
Engine efficiency	90%
Total drive-train	77%

Table 2.7: Drive-train efficiency scheme 3, modified from transportenvironment.org (2017)

EVs, hydrogen fuel cell vehicles, and power-to-liquid vehicles (transportenvironment.org 2017). The decomposed efficiency of battery electric vehicle can be found in Table 2.7.

2.6.4 State of Charge (SOC) and battery using behavior

The state of charge (SOC) is the ratio between the difference of the rated capacity and the charge balance on the one hand and the rated capacity on the other hand. The state of charge is 1 when the state of charge full is reached and 0 after a net discharge of the rated capacity (Sauer et al. 1999).

It is suggested by Miles (2018) that lithium batteries should never be entirely discharged, as such a behavior drastically shortens their life. The number of cycles before the capacity reduced to 85% can be doubled if discharging activity normally happens from 85% to 25% compared to discharging activity that normally happens from 100% to 25% (Miles 2018). Therefore, in our studies, we will consider not only different general efficiency schemes, but also will we consider average battery using behaviors. We will mainly consider 3 scenarios: recharging at 30% battery capacity left, recharging at 20% battery capacity left, and recharging at 10% battery capacity left. Suggestions will primarily focus on 30% battery capacity drainage rate to stay on the conservative side.

2.7 Routing and elevation profile

In the scope of this study, transport distances that were reported by drivers will not be used as they omitted the details of elevation changes, which are important to determine the energy demand using our analytical energy model. We chose to run routes calculation for all domestic transportations. Limiting the scope to domestic transportations is due to that the elevation profile is only available within Switzerland, and coordinates of postal codes are also only available within Switzerland. Open Source Routing Machine (OSRM) is the routing machine of our choice.

Since the dataset provides only the original and destination postal codes of any transportations, our routing needs additional supports to obtain coordinates of origins and destinations. We refer to postal codes published by Swiss Post via a repository (Swiss Post 2015, Gambazzi 2015). The coordinates normally refer to a pick-up location or a post office within a certain postal code area. One flaw of this repository is that there are more than one coordinates for one postal code in some areas. To avoid the ambiguity, we calculate the average of several coordinates in one postal code area, and we use the averaged coordinates for route calculations. It should be noted that some coordinates may not be connected with roads. Thanks to that OSRM has the option of car mode: if the origin and destination of one transportation are not next to a road, OSRM will directly allocate the origin and destination to the closest road connections. Therefore, for each of the route calculation, the transportation can be assumed approximately happen in a post office or pick-up location of a postal code area or the nearest road to a post office or pick-up location.

The mechanism of OSRM is presented here. The required input for one route calculation is the longitude and latitude coordinates of both the origin and the destination. When OSRM receives one query, it will return various pieces of information, which includes an encoded polyline string, the total distance, and the details of waypoints. We only extracted the information of the encoded polyline and the distance for our further calculation of the energy demand. The encoded polyline follows the Google polyline encoder scheme (Google Developers 2018), and the decoded information can be converted into a list of coordinations indicating the waypoints. Figure 2.6 shows the example of the returned strings from an OSRM query.

```
...     "{"code":"Ok","routes":
        [{"geometry":"fq`Huids@dl@gm@f^zY{MhdD~rArfMdo@nmCktBbuKin@b_Bu`BxxAbEhcCij@`eA~OtuCzhAd}DltBdeArfDpbE`bA`hOflBhgCh_AhyDpq
        FzuHaFjkFzQlbCrdB~tF_u@nfEfVh_I{Lf{E|
        {CnhMl`DtoCvvA~oHpyDvqI`~C~WtcCt}@diCgNrqExqAxqEfb@brB|yAtxAraDptBhi@faAndDhtAl_@hdC`fDnbBoOv[ua@rh@~h@v{Asc@zi@mkCh|@ca@mR
        eo@iS|C","legs":[{"steps":
        [],"distance":127002.7,"duration":5299.5,"summary":"","weight":5300.5}],"distance":127002.7,"duration":5299.5,"weight_name"
        :"routability","weight":5300.5}],"waypoints":[{"hint":"wravgP___38nAAAALAAAABsAAAASAAAAJwAAACwAAAAbAAAAEgAAAMWEAACwaoIA9-
        zTAsxqggDj7NMCAgAvDrYxLwo=","name":"Neue Rohr-Strasse","location":[8.546992,47.443191]},
        {"hint":"HEXCgyzwyYMrAAAAIAAAAAAAAAAAAAAAKwAAACAAAAAAAAAAAAAAAAMWEAAAllHIApRzMArWUcgDdHMwCAACvFLYxLwo=","name":"Tannackerstr
        asse","location":[7.509029,46.931109]}]}" == $0
```

Figure 2.6: Example of the returned value of OSRM. After the first red circle is the encoded polyline string which can be encoded with Google polyline encoding scheme, and after the second red circle is the distance in m.

For each of the transportations recorded in datasets ***GTE13***, ***GTE14***, and ***GTE17***, a route of a transportation can be obtained through the before-mentioned processes. Since Switzerland has a great diversity of landscapes; transportations in flat roads are not expected. Consequentially, using a flat surface energy model is not appropriate. We have obtained the permission to use the 25m digital elevation model (DEM) provided by Swisstopo (2019) within the domain of ETH for academic purposes.

After decoding the geometry from OSRM server, a list of geographical coordinates of waypoints can be obtained. By sending each waypoint of a route into the online-hosted DEM server, an elevation profile can be obtained in the unit of m. Calculating the distance between each waypoints within a route can also decompose the entire routes into several parts. Each part has an origin and a destination with elevations. Therefore, concatenating several parts into an entire route can give us a rough elevation changing profile for one transportation. Figure 2.7 shows the elevation profile that is obtained by using OSRM and 25m DEM with only postal codes of origin and destination are available from our datasets.

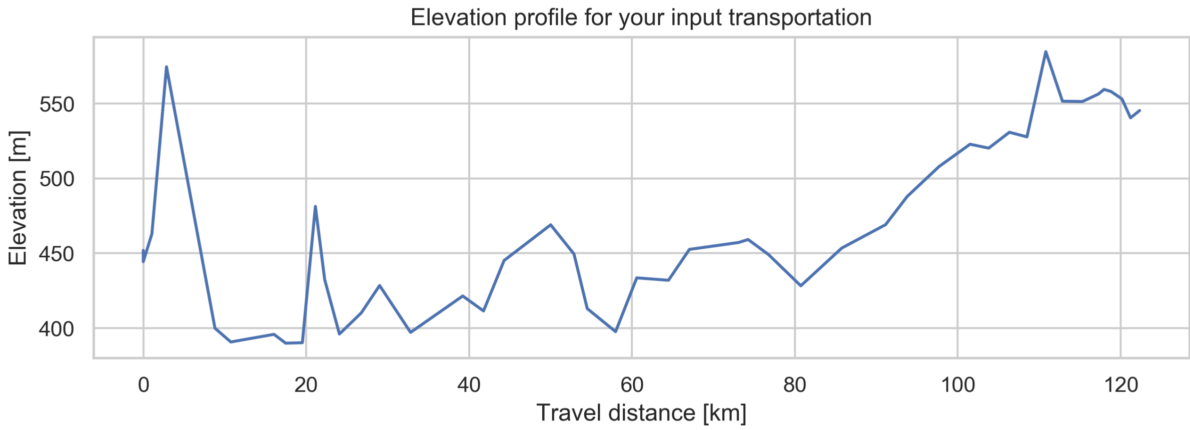


Figure 2.7: Elevation profile derived by using OSRM and DEM. This profile shows the elevation change from Seebach, Zurich (8052) to Bern (3002).

To validate whether Figure 2.7 produces the right profile, we use Graphhopper map because Graphhopper map provides the elevation profile of a route on each route request. By inputting postal codes of origin and destination within Graphhopper, a route can be calculated with elevation profile displayed as well. The elevation profile from Graphhopper is shown in Figure 2.8.

Figure 2.8: Elevation profile obtained from Graphhopper for the route that has the same origin and destination postal codes as Figure 2.7.

Graphhopper's elevation profile has a higher sampling rate than our routing machine giving more precise locations of global maximum and minimum elevations along the route. However, Figure 2.7 and Figure 2.8 do show very similar trends of change. Hence, with our methodology, an elevation profile of a route within Switzerland can be produced correctly. Consequentially, with the correct elevation profile and routing distances, we can more precisely estimate the energy demand of each transportation by summing up the energy demands on decomposed sections of one transportation. Figure 2.9 shows comprehensive steps on how the energy demand of one transportation can be obtained with only the cargo weight, and the postal codes of origin and destination as inputs.

By applying the route calculation, the elevation profile extraction, our analytical energy model, and different using behaviors and general efficiencies for all the data pieces in the datasets, we can calculate the entire energy demand for sampled vehicles under different scenarios. By referring to the total number of trucks in Switzerland, we can derive the total energy demand yearly from domestically registered trucks.

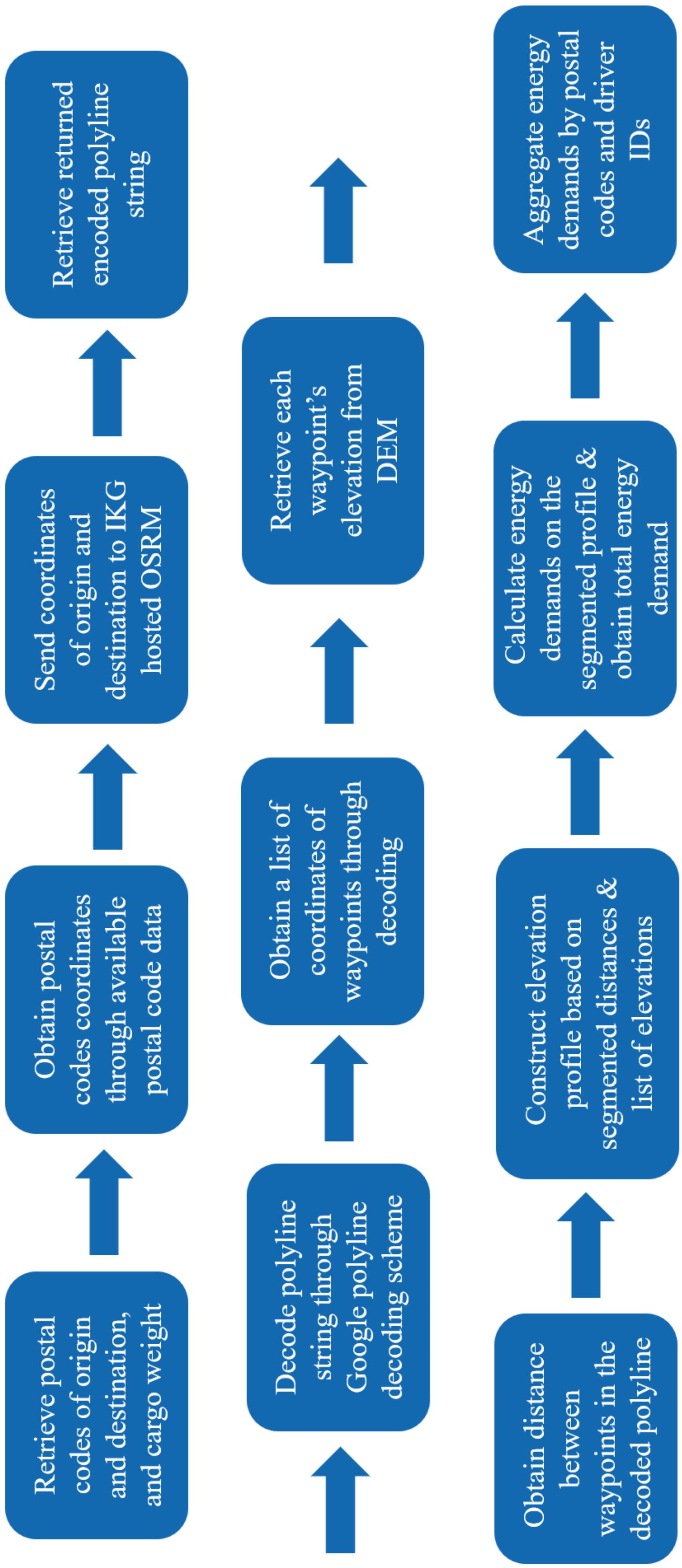


Figure 2.9: Flowchart of obtaining transportations' theoretical energy demands.

Chapter 3

Results and discussions

In this chapter, results that consider different perspectives and scenarios will be presented. On the one hand, the total energy demand that comes from the potential BET deployment on national and postal code levels will be presented. The carbon avoidance effect will also be addressed. On the other hand, we will also discuss different scenarios that may give the reader several clues on the capability of current BETs if they are placed into Switzerland's logistic sector immediately, and how future's BETs should be to replace a certain ratio of current ICE trucks.

3.1 Switzerland's energy demand coming from trucks

In this section, three types of results will be discussed:

- The consumption of energy from BETs if all trucks being replaced by BYD's current BET solutions and the ratio of BET energy demand to Switzerland's total electricity end-consumption.
- The potential amount of CO_2 can be avoided from the full deployment of BETs.
- On the postal code level, how much energy would be needed for each of the postal code areas under the full deployment. Furthermore, the potential impact brought by charging activities will be discussed.

3.1.1 Total energy consumption from truck transportation

By assuming that all current ICE trucks will be replaced by BYD's available BETs, the total energy demand from the truck transportation of domestically registered vehicles in Switzerland can be derived. As mentioned in chapter 2, general efficiency of BETs can vary between 77% to 85% according to available studies. In addition, we added a 90% general efficiency assumption in this subsection to give the reader a rough idea of how much of the energy consumption can be reduced if BET's general efficiency can increase to 90%. The derived total energy demand under each general efficiency level has been multiplied by a factor of 1.05 in order to compensate the charging and storage losses (transportenvironment.org 2017).

It should be noted that according to Swiss Federal Statistical Office (FSO) (2015, 2016, 2018), the number of registered vehicles (heavier than 3.5t) had been decreasing. Therefore, in our 3 sets of data, we applied different numbers of vehicles into our

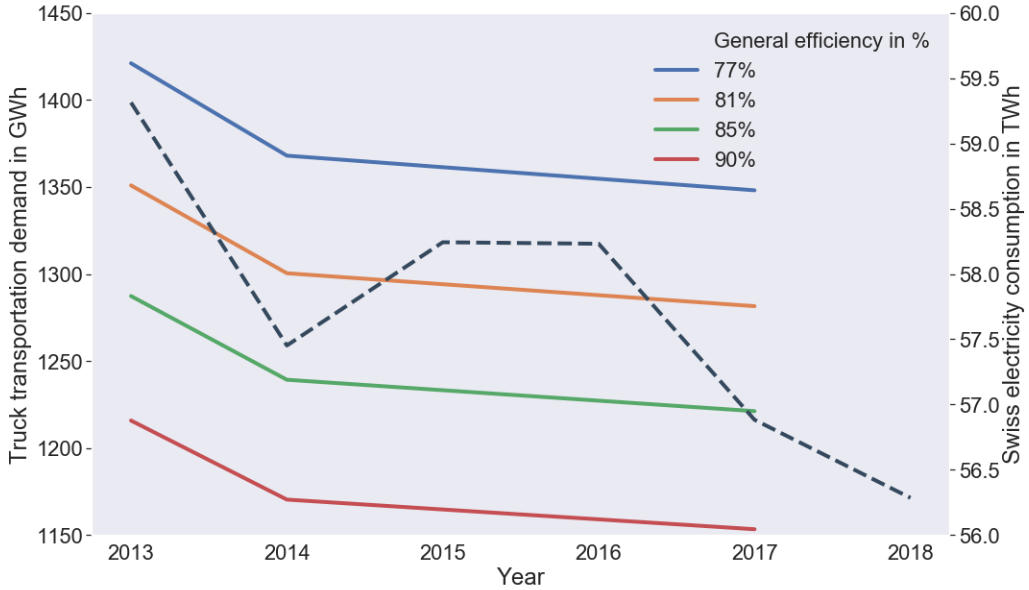


Figure 3.1: The total energy demand of truck transportation assuming BETs are entirely deployed; the dashed line and secondary axis is the total electricity consumption in Switzerland annually, data derived from International Energy Agency (IEA) (2019) and Swissgrid (2019).

total energy demand estimation. For ***GTE13*** and ***GTE14***, we applied 57,000 vehicles as the total number of registered vehicles as suggested by Swiss Federal Statistical Office (FSO) (2015) that 8,500 vehicles that they sent out forms to comprise of 15% of the total number of vehicles. For ***GTE17***, we applied 54,000 vehicles as the total number of registered vehicles in accordance with Swiss Federal Statistical Office (FSO) (2018).

By taking all above-mentioned conditions, we have Figure 3.1 suggesting that both truck transportation's total energy consumption and Switzerland's end-consumption on electricity are decreasing. Such a result does not surprise because Switzerland's 2050 energy strategy already suggests that by 2050, the total energy demand will be reduced to 40% of 2008's energy consumption level and by 2035, the total electricity consumption will be reduced to 87% of 2000's electricity consumption level Swiss Federal Office of Energy (SFOE) (2018).

Figure 3.1 also allows us to calculate the ratio of the potential BET energy consumption to the total electricity end-consumption. By considering a 77% general efficiency of BETs, the annual energy demand from truck transportations was calculated to be roughly $1348\mathrm{GW\,h}$ in 2017, which is the latest year of our dataset. In 2017, Switzerland consumed 56.88 $\mathrm{TW\,h}$ of electricity in terms of end-consumption. Therefore, the ratio of the potential BET energy consumption to the total electricity end-consumption in this case can be calculated by $\frac{1.348\mathrm{TW\,h}}{56.88\mathrm{TW\,h}} \cdot 100\% = 2.37\%$. For 2014, the annual energy demand of BETs was calculated to be approximately $1368\mathrm{GW\,h}$, and the annual end-consumption of electricity in Switzerland was $57.45\mathrm{TW\,h}$. The ratio can be calculated as $\frac{1.368\mathrm{TW\,h}}{57.45\mathrm{TW\,h}} \cdot 100\% = 2.38\%$. Lastly, in 2013, which is the earliest year of our dataset, truck transportation's annual energy demand was calculated to be approximately $1421\mathrm{GW\,h}$, and the annual end-consumption of electricity in Switzerland was $59.313\mathrm{TW\,h}$. The ratio in 2013, by considering the general efficiency of BET to be 77%, is $\frac{1.421\mathrm{TW\,h}}{59.31\mathrm{TW\,h}} \cdot 100\% = 2.40\%$. We can see that although general demands for electricity and energy have been decreasing over the past 6 years, the ratio between the energy

Year	Electricity end-consumption (in TWh)	Energy demand from BET (in GWh)			
		77%	81%	85%	90%
2017	56.88	1,348	1,282	1,221	1,153
Ratio		2.37%	2.25%	2.15%	2.03%
2014	57.45	1,368	1,300	1,239	1,170
Ratio		2.38%	2.26%	2.16%	2.04%
2013	59.31	1,421	1,351	1,287	1,216
Ratio		2.40%	2.28%	2.17%	2.05%

Table 3.1: Energy demands of BETs under different general efficiencies and ratio to Switzerland's electricity consumption

demand of BETs and the electricity consumption from end-consumers has been staying around 2.4% when considering the general efficiency of 77% for BETs.

Table 3.1 expands the calculations of the previous paragraph into different general efficiency levels. It is clear to see that under the scenario that BETs can reach 90% of total efficiency, the ratio of the potential BET energy demand to the national annual electricity end-consumption will be approximately 2%. As of current general efficiencies (77%-85%), the lower bound and upper bound of the ratio of the BET energy demand to the total electricity end-consumption can be defined as 2.15% and 2.4%.

3.1.2 Potential amount of CO_2 avoided

As shown in Table 2.5, BETs have higher efficiency. Therefore, for transporting the same amount of goods each year, BETs require less secondary energy. Consequentially, BETs should produce less CO_2 than ICE trucks do.

To calculate the amount of CO_2 avoided by the full deployment of BETs, Switzerland's electricity mix is necessary for us to calculate the carbon intensity per $\mathrm{kW\,h}$ electricity delivered to the end-consumer. According to Federal Office for the Environment (FEON) (2018), the carbon intensity of delivery energy mix is 81.6g $CO_2/\mathrm{kW\,h}$[1]. By applying obtained carbon intensity to energy demand under different scenarios, CO_2 emissions caused by BETs can be defined as shown in Table 3.2. From Table 3.2, we can see that if all transportations in 2017 could be carried out by BETs, roughly 0.1 Mt of CO_2 would be produced under the 77% general efficiency scheme.

Year		General efficiency of BET			
		77%	81%	85%	90%
2017	Electricity demand ($\mathrm{GW\,h}$)	1,348	1,281	1,221	1,153
	CO2 production (Mt)	0.110	0.104	0.099	0.094
2014	Electricity demand ($\mathrm{GW\,h}$)	1,368	1,300	1,239	1,170
	CO2 production (Mt)	0.112	0.106	0.10	0.096
2013	Electricity demand ($\mathrm{GW\,h}$)	1,421	1,351	1,287	1,216
	CO2 production (Mt)	0.116	0.110	0.105	0.992

Table 3.2: Potential CO2 production from BETs' energy demands

[1]Such a figure takes consideration of electricity mix of imported electricity, which may not be produced from carbon-neutral sources.

For current ICE trucks and their associated CO_2 production due to transportation activities, we firstly calculated the energy demand based on the ICE efficiency that is presented in Table 2.5. With an average fleet efficiency of 37%, we can obtain the total energy demand for ICE trucks in $\mathrm{kW\,h}$ by taking the theoretical energy demand calculated from our energy model. Common diesel fuel has 9.5 $\mathrm{kW\,h}$ per liter (Zittel & Wurster 2006, Glenn Elert and Various Authors 2019). As a result, the annual diesel demand can be obtained in liters. Suggested by ecoscore.be (2019), a liter of diesel can emit 2640 g CO_2 after combustion. The total CO_2 caused by truck transportations then can be derived as shown in Table 3.3. In 2017, there were roughly 0.78 Mt CO_2 produced because of transportations carried out by diesel trucks.

	2017	2014	2013
Theoritical energy demand (GW h)	1,038	1,053	1,094
ICE energy demand (37%, in GW h)	2,805.44	2,846.86	2,957.32
Diesel in need (l)	295,309,786	299,669,104	311,296,871
CO2 production (kg)	779,617,836	791,126,434	821,823,738
CO2 production (Mt)	0.779	0.791	0.822

Table 3.3: Switzerland's annual diesel demand and CO2 production from diesel trucks

Compared to the potential CO_2 brought by BETs at a general efficiency of 77%, diesel trucks can produce 0.67 Mt additional CO_2 per year as of 2017. A full deployment of BETs can reduce the CO_2 produced by truck transportations by 86% ($\frac{0.67}{0.78} \cdot 100\%$). According to World Bank (2019*b*), Switzerland has an annual CO_2 emission of 35.05 Mt. A reduction of 0.67 Mt of CO_2 in truck transportations stands for a 1.9% reduction of Switzerland's total CO_2 production if all ICE trucks can be replaced by BETs.

However, even with a very high projected carbon price, which is assumed to be 100$ per ton in 2030 given by World Bank (2017), 0.67 Mt of CO_2 does not pose a high financial return. By considering 100% per ton of CO_2 as the future price, 0.67 Mt can give a cost-reduction of 67 million $ per year. According to Refinitiv (2018), the projected carbon price will not change dramatically, staying at between 20 to 25 €. On average, each year we should expect a cost reduction of $22.5€{\cdot}0.67Mt{=}15.08mn€$ solely coming from the carbon avoidance by operating BETs.

3.1.3 Energy demand based on locations

In this subsection, ***GTE13***, ***GTE14***, and ***GTE17*** are all combined into one dataset for visualization. By combining the datasets, the coverage of transportations is more comprehensive, covering more land areas of Switzerland.

Figure 3.2 shows the energy demands on each postal code area on an annual basis. It should be noted that because current technology's charging time (as shown in Table 2.6) is longer than a typical driver's rest time, this section and the next section assume every charging activity happens before the transportation starts. In Figure 3.2, it shows 5 major cities in Switzerland along with the energy demand on the local level. It is very clear to see that transportation hot areas (as indicated by red colors) are very closely linked to either nearby districts of major cities or in the middle of major cities' connections. Two exceptions can be found on the Eastern part of Switzerland. By comparing the geographical information, we can find that hot areas are St.

Gallen and Chur. Chur is the connecting Alpine City between north and south. And the A3/A13 motorway connects Chur as an international trans-alpine route with the European highway network on the axis between Germany and Italy (Chur City Info 2019). Such a location makes Chur a very important logistics center. Similar situations also apply to St. Gallen, which is a population center in East Switzerland and connects with the A1 motorway easily connecting all major routes to Switzerland, Germany, and Italy.

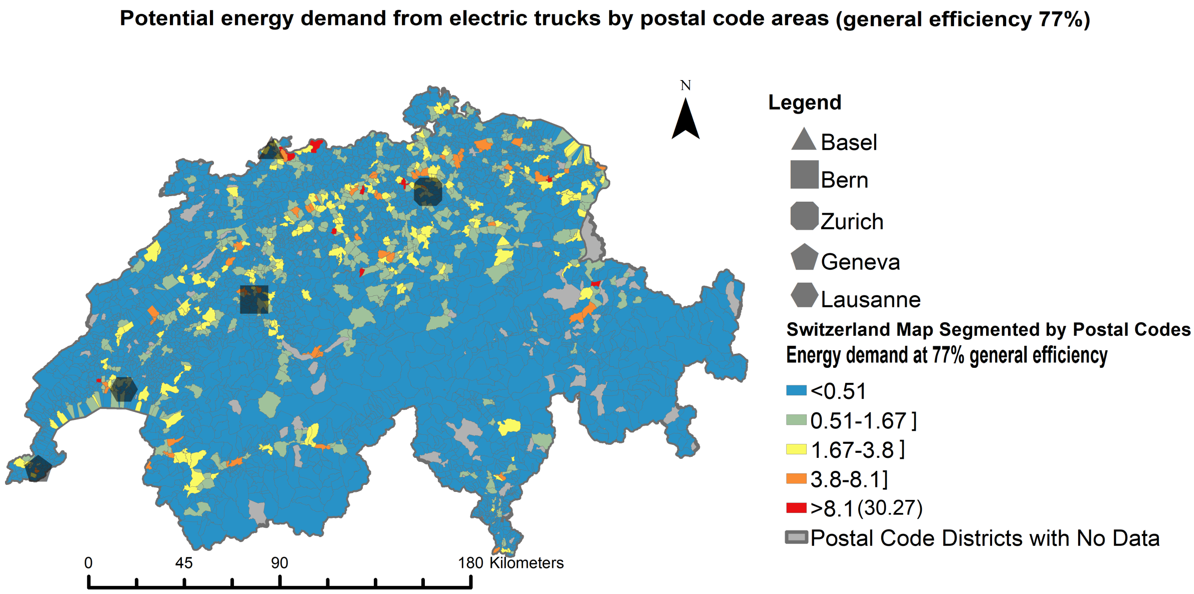


Figure 3.2: Energy demands on each postal code area for truck transportation assuming that BETs are entirely deployed. The classification method is natural breaks (Jenks 1967). Transparent symbols in different shapes are five major cities in Switzerland (from left: Geneva, Lausanne, Bern, Basel, and Zurich).

3.1.4 Energy demand's impact on local level

With the energy demand annually categorized by postal code areas, it is possible to discuss the potential impact of BETs charging activities that can bring into the grid. On the national level, an additional energy demand of 2.15% to 2.4% from BETs should not have a heavy impact on the supply side when considering the general electricity end-consumption on average is going down. Nevertheless, it is possible that on the local level, the potential energy demand from BETs can have a large impact on local communities due to the stresses caused on the transmission and distribution systems. However, a McKinsey's study on EV market suggests that EVs are unlikely to create a power-demand crisis but could reshape the load curve (Engel et al. 2018). Nevertheless, a large number of distribution system operators (DSO) should also be considered. According to IEA (2018), 650 DSOs are currently in the electricity market of Switzerland. How to coordinate the transmission and distribution systems can be a big challenge with upcoming deployments of EVs and BETs.

We have obtained the population data based on the municipality level with all postal codes mapping to the municipalities (Swiss Federal Statistical Office (FSO) 2019*b*,*a*). Each postal code may correspond to one or several municipalities, and vice versa. Therefore, we can map the population in each postal code area.

The annual average personal electricity consumption can be estimated as 7,520 $\mathrm{kW\,h}$, including both residential and industrial electricity consumptions in Switzerland (World Bank 2019*a*). By normalizing the potential energy demand from BETs on each postal code area with the population, the obtained result can be compared to the average electricity consumption per capita. In such a method, a very first impression on how BETs' energy demands can impact on certain local communities can be delivered. In this study, we consider below 10% of the personal average electricity consumption as very little stress to the transmission line on the local level. Between 10% to 30% is considered to be acceptable. Above 30% can be considered as not suitable to directly build charging stations on existing networks because of the potential high increment of the load. In such a sense, we can produce the plots under different general efficiencies.

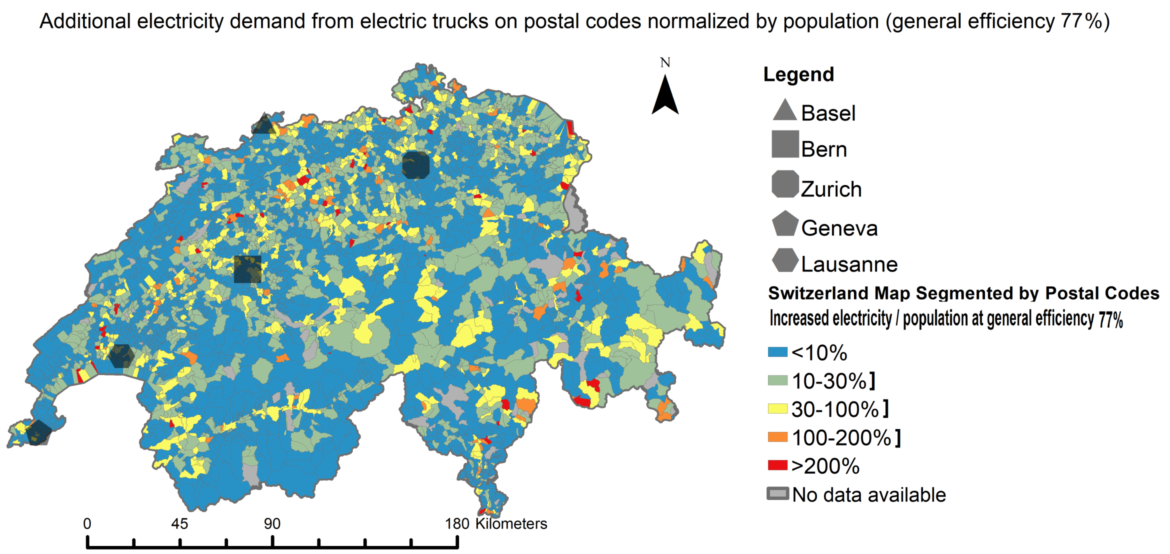


Figure 3.3: Switzerland map of the ratio between BETs energy demand per capita and the average personal electricity consumption on each postal code area.

As Figure 3.3 shows us, in the middle part of Switzerland, many postal code areas will be expecting highly increased energy consumptions if BETs completely replace current ICE trucks. Moreover, high demand areas do not show a continuity pattern, making the reinforcement of transmission infrastructures even more difficult. Therefore, from a perspective of transmission safety, we do not recommend going fully electric in the logistics sector at current stages. Nonetheless, as penetrations of EV and BET getting more invasive, an overall upgrade of transmission and distributions can be expected as many current companies are already involving in researches and developments of EV-related infrastructures. For instance, Siemens has proposed a general solution from the upstream grid stabilization to the end-consumer billing system helping the impending deployments of EVs (Siemens 2019*b*). With more and more regulations and technologies in favor of EVs, it is optimistic to say that currently concluded potential stresses given by BETs can be resolved with the help of fast technology developments and effective policies accelerating the EV development.

3.2 Substitutability of battery electric trucks

The substitutability of BETs in our study is defined as the ratio of current ICE trucks that can be replaced by commercially available BYD solutions.

In the previous chapter, each transportation's energy demands under different general efficiencies have been calculated under the assumption that recharging activities happen before transportations start. In the following sections, we primarily consider each vehicle operates 5 days a week[2] and drivers normally prefer to charge at off-work times during the night. The number of vehicles successfully participated in each year's survey is also known by sorting out unique IDs. Therefore, grouping each driver's energy demand via his/her ID yields each vehicle's weekly energy demand. By comparing each vehicle's average daily demand with current BYD solutions' battery capacities, we can calculate the percentage of vehicles that can be substituted by BYD BETs. We also take different average SOCs at recharging (or drainage rate) into consideration.

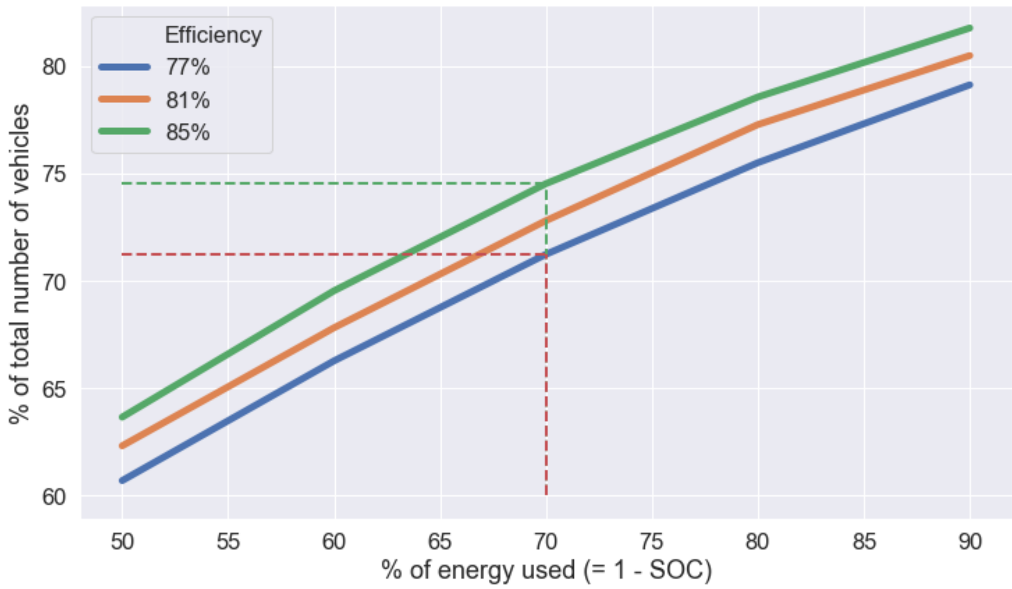


Figure 3.4: Percentage of vehicles can be replaced by BETs under different general efficiencies and different SOCs (derived from 2017's dataset). The range of the substitutability of BETs in our study can be defined as 71.24% to 81.77%.

Figure 3.4 shows very clearly that when considering a normal BET will be re-charged after 70% usage of the battery, BYD BETs can replace approximately 74.52% of current ICE trucks under a general efficiency of 85%. If considering re-charges normally happen after an 80% drainage of the battery, BYD BETs can replace 75.50% to 78.56% of current ICE trucks. By taking the range of percentages of the energy used to be 70% and 90% and considering 3 scenarios of general efficiencies (77%, 81%, and 85%), the substitutability of BETs can be defined as 71.24% to 81.77%.

[2]In the dataset, the amount of transportations distributes evenly across Monday through Friday around 20,000 entries per day with only around 3,000 entries occurred during weekends.

It is also interesting to see that under the assumptions of average drainage rate of 70% and 5 working days in one week, 71.24% to 74.52% of current ICE trucks can be replaced by the BETs; however, when we sum up the total energy demand required by trucks that can be replaced by current BETs, the energy demands from the replaceable group only counts for 29.15% to 33.29% of the total energy demand (shown in Table 3.4). Therefore, 25.48% to 28.76% of trucks are responsible for 66.71% to 70.85% of the total energy demand. Under the current scenario, if all drivers use diesel as the energy source, we can say that 25.48% to 28.76% of trucks are responsible for 66.71% to 70.85% of the total emissions.

	Effi. 77% SOC70	Effi. 81% SOC70	Effi. 85% SOC70
Can be replaced by BET	29.15%	31.04%	33.29%
Cannot be replaced by BET	70.85%	68.96%	66.71%

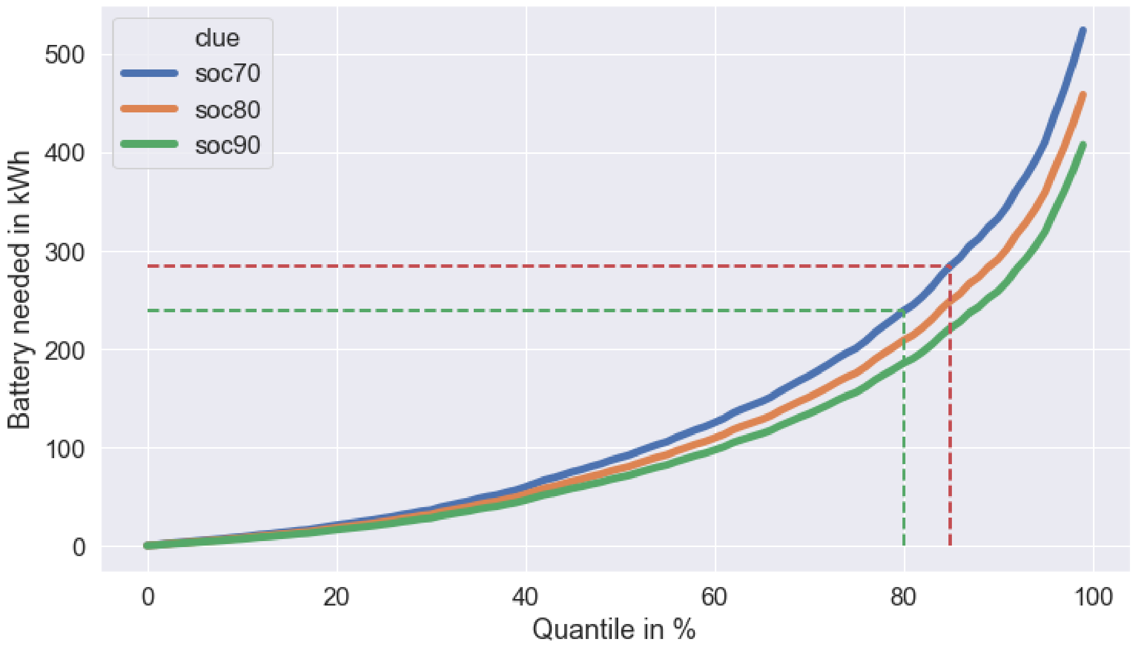
Table 3.4: Percentage of energy demand to total energy demand in the BYD BET-replaceable and the BYD BET-irreplaceable group (derived from 2017's dataset)

3.3 Future of BETs

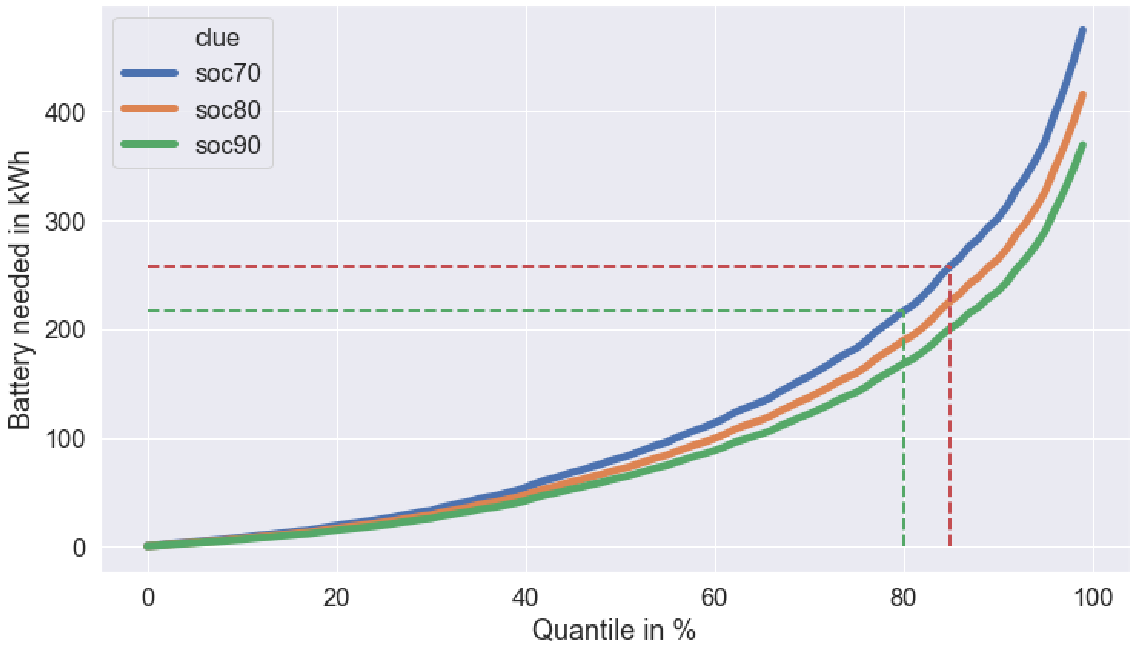
As shown in the last subsection, under the assumption that drivers will normally not drain the battery to below 30% of the total capacity, slightly over 70% of current vehicles can be replaced by the current exiting BYD fleet. Therefore, in this subsection, our main idea is to provide information on which level the battery capacity is needed to meet 80% or 85% electrification of the Swiss-registered trucks under the same assumptions as we have in section 3.2.

Figure 3.5 shows that a BET with a minimum battery capacity of 200$\mathrm{kW\,h}$ can already replace nearly 80% of current Switzerland's ICE fleet regardless of financial considerations under both 77% and 85% general efficiencies. To cover 80% trucks' energy demands at the general efficiency level of 77%, 239$\mathrm{kW\,h}$ battery capacity is needed under the average drainage rate of 70% (shown as SOC70) as shown in the dashed green line in Figure 3.5a. Under the same general efficiency, the battery will need to have a 285$\mathrm{kW\,h}$ minimum capacity, as shown in Figure 3.5a as the dashed red line, to replace 85% of current ICE trucks. Figure 3.5b shows the battery capacity requirements if the general efficiency can reach 85% on average: a minimum of 217$\mathrm{kW\,h}$ battery capacity will be required under the 70% drainage rate (shown as SOC70) in order to replace 80% of current ICE trucks, and a minimum of 258$\mathrm{kW\,h}$ battery capacity will be needed to replace 85% of current ICE trucks.

Under the currently available technology of Li-based batteries, it is not possible to go for 100% electric in the logistics sector. As Figure 3.5 indicates, to meet 100% electric, more than 500 $\mathrm{kW\,h}$ battery capacity may be required. As shown in Figure 1.1, the specific energy of the Li-ion battery varies from 60 to 180 $\mathrm{W\,h\,kg^{-1}}$. 500 $\mathrm{kW\,h}$ battery can weigh up to 2.78t. Therefore, the future of BET can be developed towards the deployment of the higher specific energy type of battery, better charging coordinations, and faster charging technologies. These technologies can be deployed so that charging activities can be applied within normal truck driver's rest time window or during the transportations.



(a) Battery capacity needed to meet higher replacement rate with BET's general efficiency at 77% derived from ***GTE17***.



(b) Battery capacity needed to meet higher replacement rate with BET's general efficiency at 85% derived from ***GTE17***.

Figure 3.5: Battery capacity needed to meet the higher replacement rate (from 2017's data).

This page is intentionally left blank.

Chapter 4

Conclusions and outlook

After processing the datasets and analyzing them by constructing routes and elevation profile and by applying the analytical energy models, the study has found the following conclusions by discussing from different perspectives:

- The electricity demand from the full deployment of electric trucks will count 2.15% to 2.4% for the total Switzerland's electricity end-consumption. By considering the general electricity end-consumption will go down, 2.15% to 2.4% of the total electricity consumption should not pose a large stress to transmission networks on the national level.

- On the local level, municipalities with a high transportation demand and a small amount of residents may expect large stresses if an immediate deployment of electric trucks happens. Grid reinforcement is recommended.

- A full deployment of BETs can avoid at least 1.9% of the CO_2 emission of Switzerland annually, indicating a cost reduction of 15.08mn€ incurred from ICE trucks' CO_2 emissions annually.

- 71.24% to 81.77% of current Swiss-registered ICE trucks can be replaced with current commercially available BETs under different using and general efficiency scenarios.

- 217 to 285$\mathrm{kW\,h}$ minimum battery capacity is needed if 80% to 85% of current ICE trucks need to be replaced by BETs under the condition that the general efficiency stays at our assumed range (77% to 85%).

At the current market, we have observed that many companies have started addressing the potential issues that will be brought by the deployments of EVs. We suggest a better understanding of the charging patterns from a temporal-based approach to better assess the potential impact of the BET deployment to effectively plan a deployment strategy as it is very likely to see that many commercial vehicles will be replaced by BETs or similar forms. Furthermore, a financial analysis is suggested to assess the financial feasibility of deploying BETs in Switzerland to provide precise and accurate suggestions for policy-makers to make the right decision.

This page is intentionally left blank.

Bibliography

Bundesamt für Statistik BFS (2015), 'Gütertransporterhebung gte 2008-2014: Erhebungsbericht'. in English: Goods transport survey GTE 2008-2014: Survey report.

BYD (2018*a*), 'Byd electric vehicles for every use', https://www.transportenvironment.org/sites/te/files/BYD%20electric%20vehicles%20for%20every%20use.pdf. Accessed: 30/05/19.

BYD (2018*b*), 'Byd truck official english website', https://en.byd.com/truck/#models. Accessed: 30/05/19.

Çabukoglu, E., Georges, G., Küng, L., Pareschi, G. & Boulouchos, K. (2018), 'Battery electric propulsion: An option for heavy-duty vehicles? results from a swiss case-study', *Transportation Research Part C: Emerging Technologies* **88**, 107–123.

California Fuel Cell Partnership (2019), 'Hydrogen - cost to refill', https://cafcp.org/content/cost-refill.

Chur City Info (2019), 'Getting to chur', https://chur.graubuenden.ch/en/info/getting-chur.

DAF (2019), 'Innovative electric trucks demonstrate the road to the future', https://www.daf.com/en/about-daf/innovation/electric-and-hybrid-trucks. Accessed: 30/05/19.

Daimler (2017), 'This is what winners look like: Mercedes-benz etruck scoops the european transport prize for sustainability 2018', https://bit.ly/2MywuCT. Accessed: 30/05/19.

Davlantes, M. (2015), 'How to decode polylines from google maps direction api in php', https://stackoverflow.com/questions/15380712/how-to-decode-polylines-from-google-maps-direction-api-in-php.

Earl, T., Mathieu, L., Cornelis, S., Kenny, S., Ambel, C. C. & Nix, J. (2018), Analysis of long haul battery electric trucks in eu marketplace and technology, economic, environmental, and policy perspectives., *in* '8th Commercial Vehicle Workshop, Graz, 17-18 May 2018'.

ecoscore.be (2019), 'How to calculate the co2 emission from the fuel consumption?', http://ecoscore.be/en/info/ecoscore/co2.

Engel, H., Hensley, R., Knupfer, S. & Sahdev, S. (2018), 'The potential impact of electric vehicles on global energy systems', https://www.mckinsey.

com/industries/automotive-and-assembly/our-insights/the-potential-impact-of-electric-vehicles-on-global-energy-systems.

European Commission (2016), 'Paris agreement', https://ec.europa.eu/clima/policies/international/negotiations/paris_en#tab-0-0. Accessed: 13/03/19.

European Commission (2018*a*), '2050 long-term strategy', https://ec.europa.eu/clima/policies/strategies/2050_en. Accessed: 13/03/19.

European Commission (2018*b*), 'Reducing co2 emissions from heavy-duty vehicles', https://ec.europa.eu/clima/policies/transport/vehicles/heavy_en. Accessed: 24/06/19.

European Pallet Association e.V. (2019), 'Epal euro pallet (epal 1)', https://www.epal-pallets.org/eu-en/load-carriers/epal-euro-pallet/. Accessed: 1/06/19.

Federal Office for the Environment (FEON) (2018), 'Climate change: Questions and answers', https://www.bafu.admin.ch/bafu/en/home/topics/climate/climate-change--questions-and-answers.html.

Gambazzi, L. (2015), 'Github repository - swiss-geolocation', https://github.com/gamba/swiss-geolocation.

Gerssen-Gondelach, S. J. & Faaij, A. P. (2012), 'Performance of batteries for electric vehicles on short and longer term', *Journal of power sources* **212**, 111–129.

Glenn Elert and Various Authors (2019), 'The physics factbook', https://hypertextbook.com/facts/2006/TatyanaNektalova.shtml.

Google Developers (2018), 'Encoded polyline algorithm format', https://developers.google.com/maps/documentation/utilities/polylinealgorithm.

Haumann, S. T. (2017), Optimizing the operation range of e-bikes in routing systems, Master's thesis, University of Vienna, ETH Zurich.

IEA (2018), Energy policies of iea countries switzerland 2018 review, Technical report, IEA.

International Energy Agency (IEA) (2019), 'Statistics - global energy data at your fingertips - switzerland total final consumption (TFC) by source', https://www.iea.org/statistics/?country=SWITLAND&year=2016&category=Energy%20consumption&indicator=TFCbySource&mode=chart&dataTable=BALANCES. Accessed: 2/06/19.

Jenks, G. F. (1967), 'The data model concept in statistical mapping', *International yearbook of cartography* **7**, 186–190.

Liimatainen, H., van Vliet, O. & Aplyn, D. (2019), 'The potential of electric trucks–an international commodity-level analysis', *Applied energy* **236**, 804–814.

Market Study Report, LLC (2019), Global battery electric truck (bet) market growth 2019-2024, Technical report, Market Study Report, LLC.

McLane, R. & Mullaney, D. (2019), 'Shenzhen: A city miles ahead'.

Miles, A. (2018), 'The secret life of an ev battery', https://cleantechnica.com/2018/08/26/the-secret-life-of-an-ev-battery/. Accessed: 1/06/19.

QY Research (2019), Global battery electric truck (bet) industry research report, growth trends and competitive analysis 2019-2025, Technical report, QY Research.

Refinitiv (2018), 'Will high european carbon prices last?', https://www.refinitiv.com/perspectives/market-insights/will-high-european-carbon-prices-last/.

Richardson, D. B. (2013), 'Electric vehicles and the electric grid: A review of modeling approaches, impacts, and renewable energy integration', *Renewable and Sustainable Energy Reviews* **19**, 247–254.

Sauer, D. U., Bopp, G., Jossen, A., Garche, J., Rothert, M. & Wollny, M. (1999), State of charge–what do we really speak about, *in* 'The 21st international telecommunications energy conference', pp. 6–9.

Scania (2018), 'Scania and northvolt partner for heavy vehicle electrification', https://www.scania.com/group/en/scania-and-northvolt-partner-for-heavy-vehicle-electrification-2/. Accessed: 30/05/19.

Schmidt, T. S. & Huenteler, J. (2016), 'Anticipating industry localization effects of clean technology deployment policies in developing countries', *Global Environmental Change* **38**, 8–20.

Schuh, F. & Zhu, Z. (2019), 'Skype Meeting on 29 May 2019 with Dominik Bucher', Skype Meeting - Private Communication.

Siemens (2019*a*), 'ehighway – electrification of road freight transport', https://new.siemens.com/global/en/products/mobility/road-solutions/electromobility/ehighway.html.

Siemens (2019*b*), 'Power to move: Fast-charging technology from siemens', https://new.siemens.com/global/en/products/mobility/road-solutions/electromobility/ecars-high-power-charging-infrastructure.html.

Swiss Federal Office of Energy (SFOE) (2018), 'Switzerland energy stategy 2050', https://www.bfe.admin.ch/bfe/en/home/policy/energy-strategy-2050.html. Accessed: 2/06/19.

Swiss Federal Statistical Office (FSO) (2019*a*), 'Regionalporträts 2019: Gemeinden - kennzahlen'.

Swiss Federal Statistical Office (FSO) (2019*b*), 'Umwandlung von plz in bfs-gemeindenummer'.

Swiss Federal Statistical Office (FSO) & Swiss Federal Office of Energy (SFOE) (2016), 'Energy – facts and figures', https://www.eda.admin.ch/content/dam/PRS-Web/bilder/infografiken/en/4.2.1_EN.jpg. Accessed: 06/05/19.

Swiss Federal Statistical Office (FSO) (2015), 'Gütertransporterhebung gte 2008-2014: Erhebungsbericht', https://www.bfs.admin.ch/bfs/de/home/statistiken/mobilitaet-verkehr/erhebungen/gte.html.

Swiss Federal Statistical Office (FSO) (2016), 'Güterverkehr in der schweiz 2015', https://www.bfs.admin.ch/bfs/de/home/statistiken/kataloge-datenbanken/publikationen.assetdetail.1365759.html.

Swiss Federal Statistical Office (FSO) (2018), 'Güterverkehr in der schweiz 2017', https://www.bfs.admin.ch/bfs/de/home/statistiken/kataloge-datenbanken/publikationen.assetdetail.6706830.html.

Swiss Post (2015), 'Customer center - all online services - locations and opening hours', https://places.post.ch/?lang=en&service=places.

Swissgrid (2019), 'Grid operation - grid data - production and consumption', https://www.swissgrid.ch/en/home/operation/grid-data/generation.html. Accessed: 2/06/19.

Swisstopo (2019), 'Swisstopo online shops - height models', https://shop.swisstopo.admin.ch/en/products/height_models.

Tesla (2019), 'Tesla semi', https://www.tesla.com/semi. Accessed: 30/05/19.

transportenvironment.org (2017), 'Cars: Battery electric most efficient by far', https://pbs.twimg.com/media/DGN8sTWWsAAepeU.jpg. Accessed: 30/05/19.

Van den Bossche, P., Vergels, F., Van Mierlo, J., Matheys, J. & Van Autenboer, W. (2006), 'Subat: An assessment of sustainable battery technology', *Journal of power sources* **162**(2), 913–919.

World Bank (2017), 'Results briefs - carbon pricing', https://https://www.worldbank.org/en/results/2017/12/01/carbon-pricing.

World Bank (2019*a*), 'Switzerland - electric power consumption (kwh per capita)', https://tradingeconomics.com/switzerland/electric-power-consumption-kwh-per-capita-wb-data.html.

World Bank (2019*b*), 'World bank database', http://datatopics.worldbank.org/world-development-indicators/.

Yang, L., Hao, C. & Chai, Y. (2018), 'Life cycle assessment of commercial delivery trucks: Diesel, plug-in electric, and battery-swap electric', *Sustainability* **10**(12), 4547.

Zittel, W. & Wurster, R. (2006), 'Advantages and disadvantages of hydrogen. hydrogen in the energy sector.', https://www.netinform.de/h2/start.aspx.

Appendix A

Additional figures

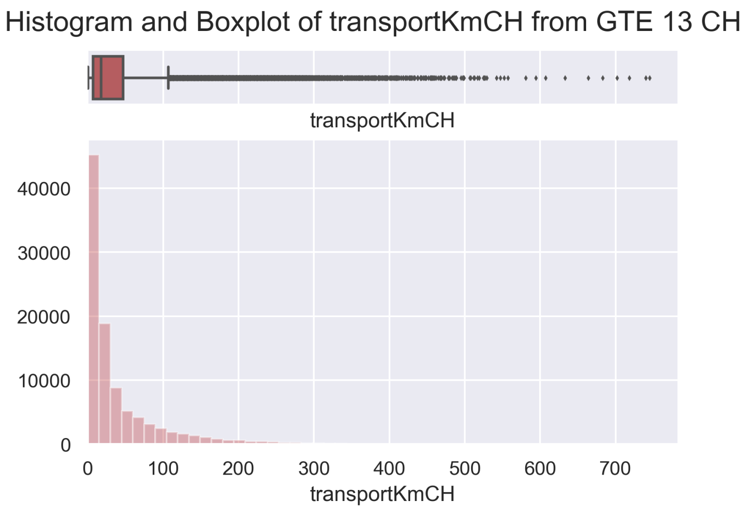


Figure A.1: Histogram and box-plot for distribution of domestic ***GTE13*** transportation distances (provided by drivers).

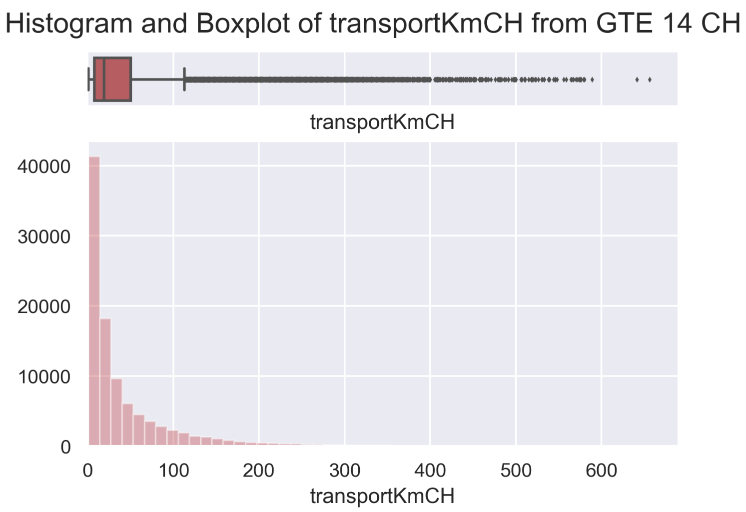


Figure A.2: Histogram and box-plot for distribution of domestic ***GTE14*** transportation distances (provided by drivers).

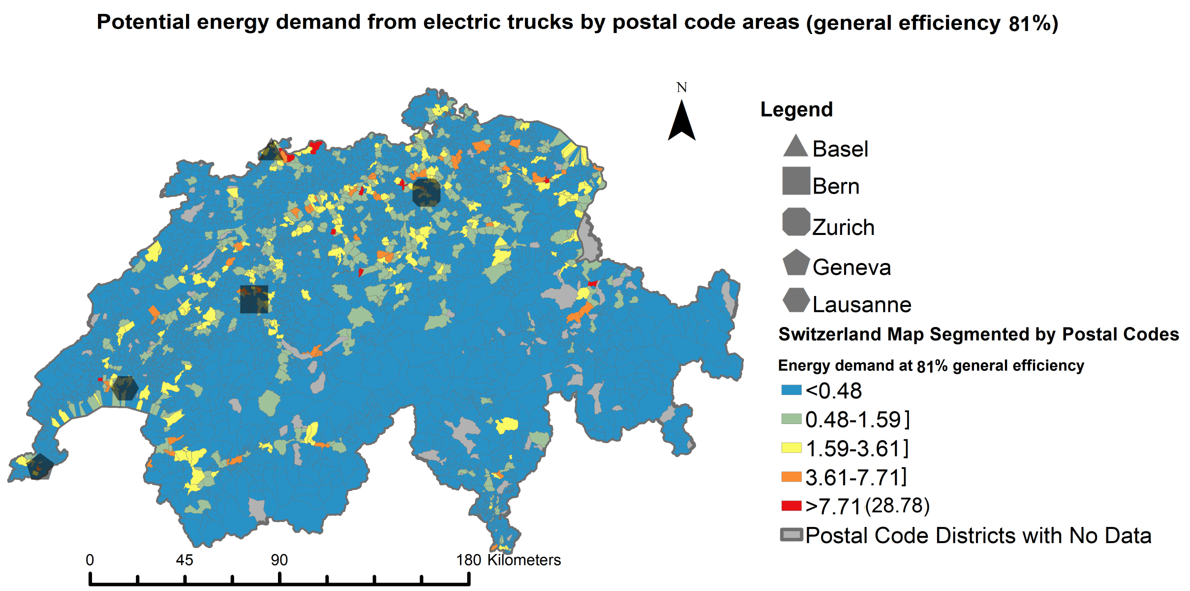


Figure A.3: Energy demands on each postal code area for truck transportation assuming that BETs are entirely deployed and that 81% of general efficiency of BETs. Classification method is natural breaks (Jenks 1967). Transparent symbols in different shapes are five major cities in Switzerland (from left: Geneva, Lausanne, Bern, Basel, and Zurich).

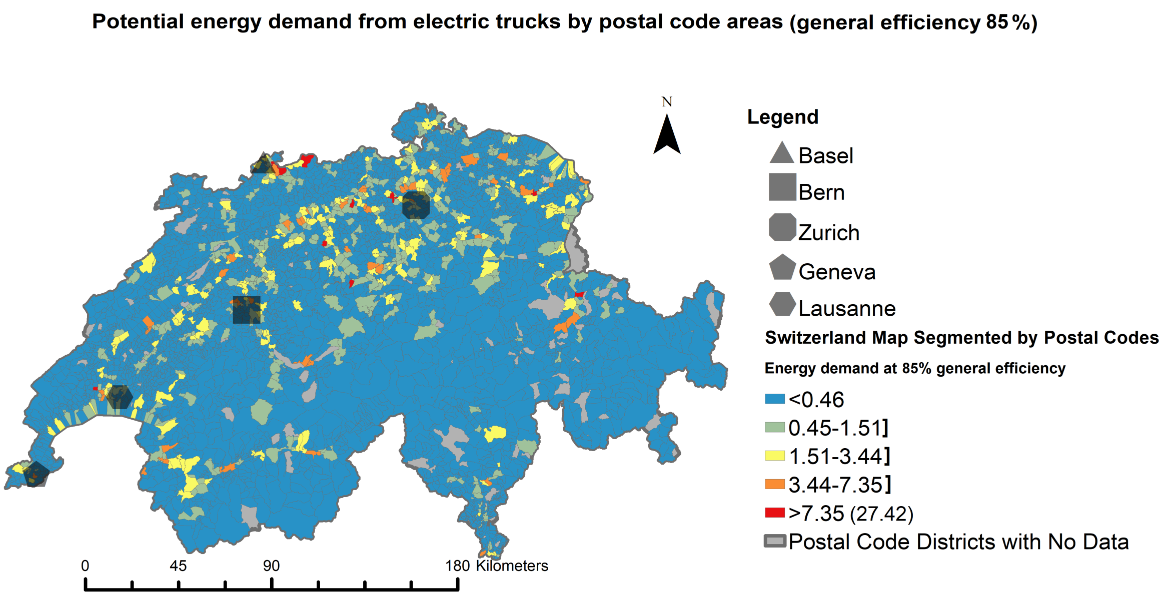


Figure A.4: Energy demands on each postal code area for truck transportation assuming that BETs are entirely deployed and that 85% of general efficiency of BETs. Classification method is natural breaks (Jenks 1967). Transparent symbols in different shapes are five major cities in Switzerland (from left: Geneva, Lausanne, Bern, Basel, and Zurich).

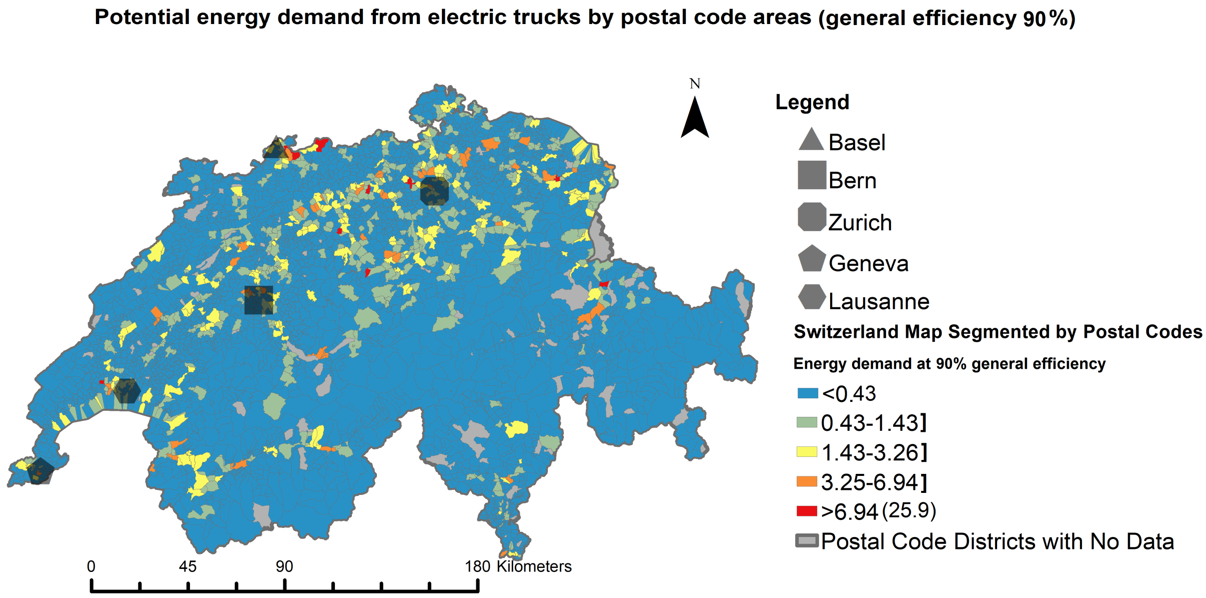


Figure A.5: Energy demands on each postal code area for truck transportation assuming that BETs are entirely deployed and that 90% of general efficiency of BETs. Classification method is natural breaks (Jenks 1967). Transparent symbols in different shapes are five major cities in Switzerland (from left: Geneva, Lausanne, Bern, Basel, and Zurich).

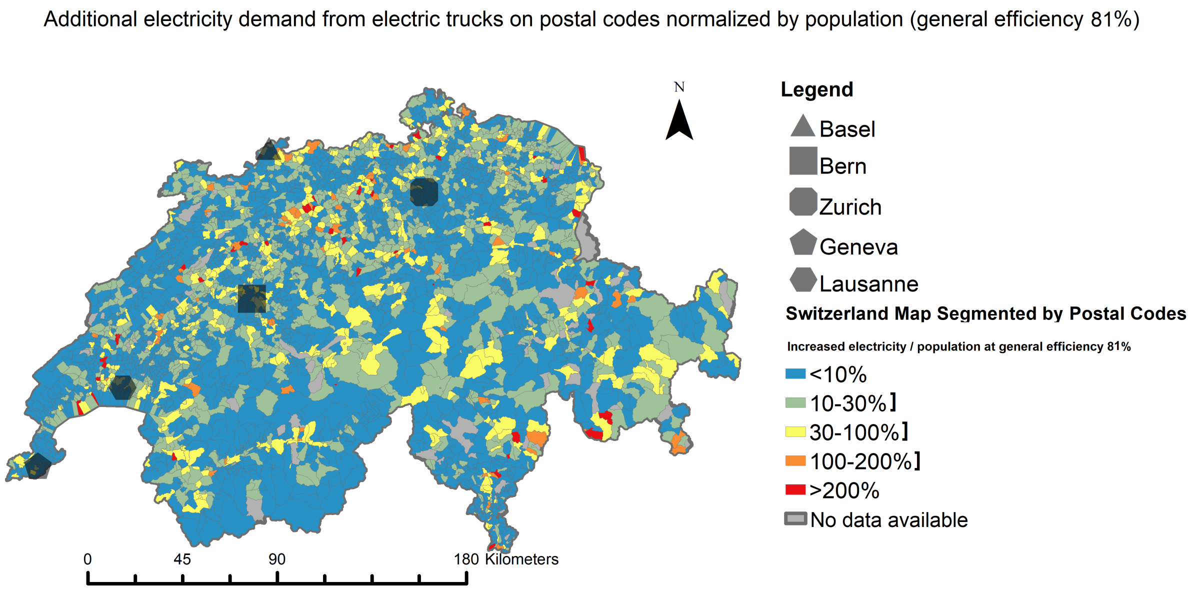


Figure A.6: Switzerland map of ratio between BET energy demand per capita and average personal electricity consumption on each postal code area under the general efficiency of 81%.

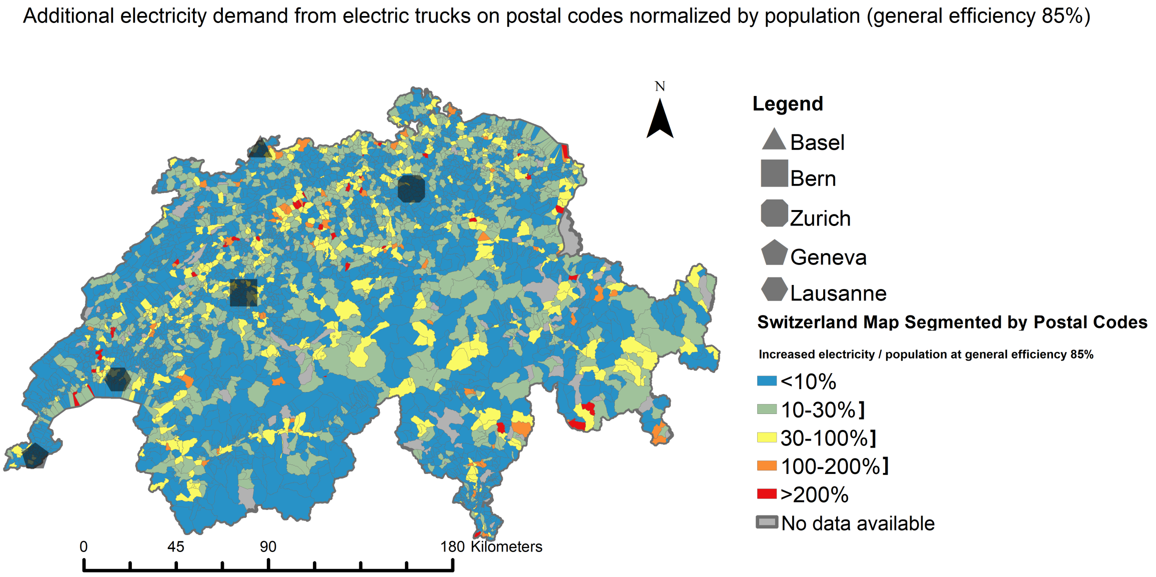


Figure A.7: Switzerland map of ratio between BET energy demand per capita and average personal electricity consumption on each postal code area under the general efficiency of 85%.

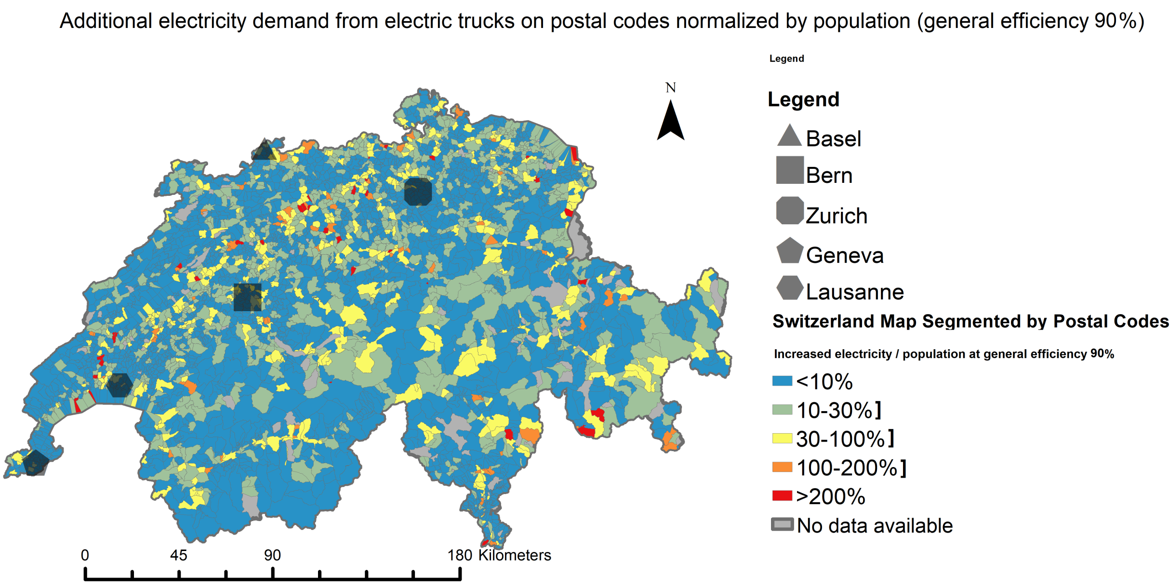


Figure A.8: Switzerland map of ratio between BET energy demand per capita and average personal electricity consumption on each postal code area under the general efficiency of 90%.

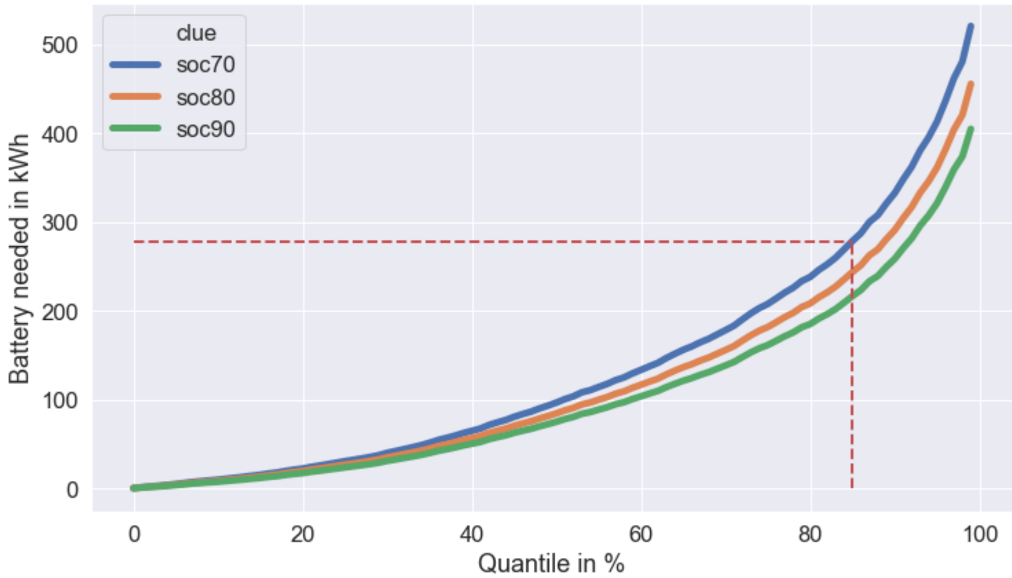


Figure A.9: Battery capacity needed to meet higher replacement rate with eTruck's general efficiency at 77% based on 2013's data.

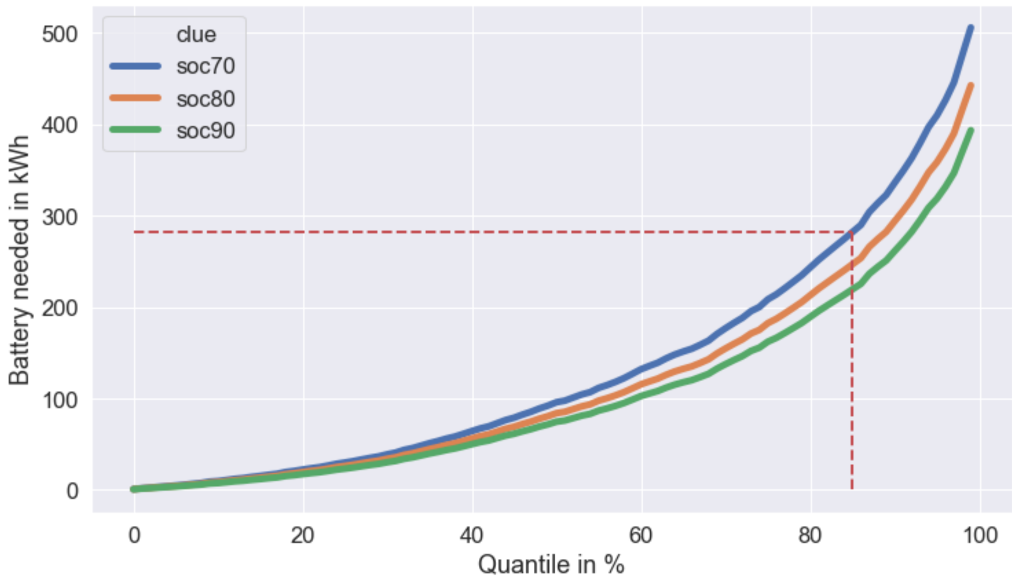


Figure A.10: Battery capacity needed to meet higher replacement rate with eTruck's general efficiency at 77% based on 2014's data.

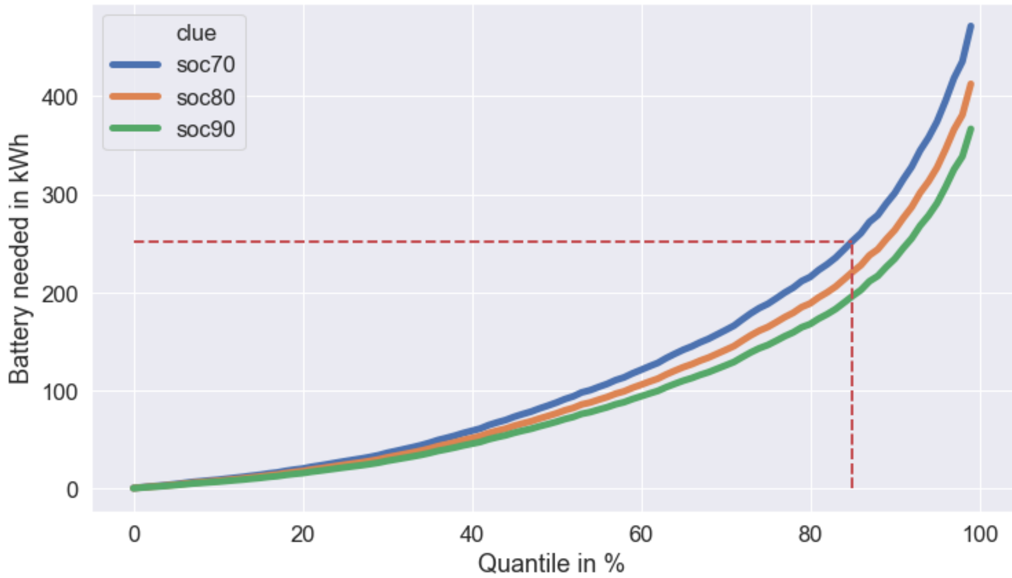


Figure A.11: Battery capacity needed to meet higher replacement rate with eTruck's general efficiency at 85% based on 2013's data.

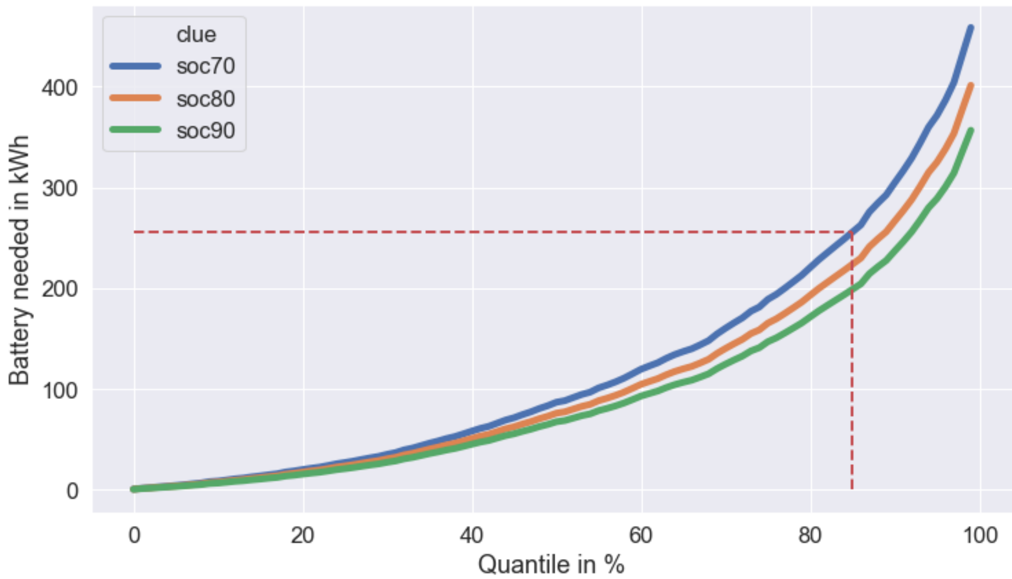


Figure A.12: Battery capacity needed to meet higher replacement rate with eTruck's general efficiency at 85% based on 2014's data.

Appendix B

Supporting codes

This part of appendix provides you with the walk-through from raw data processing, route and elevation profile construction, until the energy demand calculation of each transportation under different scenarios.

Python file that reads all the available files and temporarily stores them as dataframe:

```python
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Import available data in dataframe for further purposes
# Headers have been printed for easier use.

df_journeych13 = pd.read_csv("journeych13.csv", delimiter=';')
'''
86990 Rows
['ernr', 'journeyId', 'typeOfJourney', 'fromPlz', 'toPlz', 'fromNuts', 'toNuts',
'fromLand', 'toLand', 'numberOfDifferentGoods', 'numberOfDifferentDangerousGoods',
'numberOfDifferentFreight', 'journedistDomestic', 'journedistAbroad', 'journedisttotal', 'journeyTonCH',
'journeyTon', 'journeyTonkmCH', 'journeyTonkmTot', 'journeyLoading', 'journeyUnloading']
'''

df_transport13 = pd.read_csv("transport13.csv", delimiter=";")
'''
100013 Rows
['ernr', 'journeyId', 'weekday', 'typeOfGoodsNST2007', 'tonCH', 'CHDangerousGoods', 'typeOfCargo',
'fromNuts', 'toNuts', 'fromLand', 'toLand', 'fromPlz', 'toPlz', 'transportKm', 'transportKmCH']
'''

df_week13 = pd.read_csv("week13.csv", delimiter=";")
'''
4821 rows
['ernr', 'week', 'age', 'emission', 'maxPermissibleWeightCH', 'completeMaxPermissibleWeightCH',
'loadCapacityCH', 'lsvaRequired', 'nonResponse', 'typeOfTransport', 'vehicleKind', 'typeOfVhl',
'vehicleBrand', 'nAxes', 'cantonOfOrigin', 'trailerDevice', 'ramp', 'crane', 'winch', 'vhlKm', 'vhlKmCH', 'grossingFactor']
'''
```

```

df_journeych14 = pd.read_csv("journeych14.csv", delimiter=';')
'''
['ernr', 'journeyId', 'typeOfJourney', 'fromPlz', 'toPlz', 'fromNuts',
 'toNuts', 'fromLand', 'toLand', 'numberOfDifferentGoods', 'numberOfDifferentDangerousGoods',
 'numberOfDifferentFreight', 'journedistDomestic', 'journedistAbroad', 'journedisttotal',
 'journeyTonCH', 'journeyTon', 'journeyTonkmCH', 'journeyTonkmTot', 'journeyLoading', 'journeyUnloading']
'''


df_transport14 = pd.read_csv("transport14.csv", delimiter=";")
'''
['ernr', 'journeyId', 'weekday', 'typeOfGoodsNST2007', 'tonCH', 'CHDangerousGoods',
'typeOfCargo', 'fromNuts', 'toNuts', 'fromLand', 'toLand', 'fromPlz', 'toPlz', 'transportKm',
'transportKmCH', 'quelle_km', 'ziel_km']
'''


df_week14 = pd.read_csv("week14.csv", delimiter=";")
'''
['ernr', 'week', 'age', 'emission', 'maxPermissibleWeightCH', 'completeMaxPermissibleWeightCH',
'loadCapacityCH', 'lsvaRequired', 'nonResponse', 'typeOfTransport', 'vehicleKind', 'typeOfVhl',
'vehicleBrand', 'nAxes', 'cantonOfOrigin', 'trailerDevice', 'ramp', 'crane', 'winch', 'vhlKm',
'vhlKmCH', 'grossingFactor']
'''


df_journeych17 = pd.read_csv("journeych17.csv", delimiter=';')
'''
['ernr', 'journeyId', 'typeOfJourney', 'fromPlz', 'toPlz', 'fromNuts', 'toNuts',
'fromLand', 'toLand', 'numberOfDifferentGoods', 'numberOfDifferentDangerousGoods',
'numberOfDifferentFreight', 'journedistDomestic', 'journedistAbroad', 'journedisttotal',
'journeyTonCH', 'journeyTon', 'journeyTonkmCH', 'journeyTonkmTot', 'journeyLoading', 'journeyUnloading']
'''


df_transport17 = pd.read_csv("transport17.csv", delimiter=";")
'''
['ernr', 'journeyId', 'weekday', 'typeOfGoodsNST2007', 'tonCH', 'CHDangerousGoods',
'typeOfCargo', 'fromNuts', 'toNuts', 'fromLand', 'toLand', 'fromPlz', 'toPlz', 'transportKm', 'transportKmCH']
'''


df_week17 = pd.read_csv("week17.csv", delimiter=";")
'''
['ernr', 'week', 'age', 'emission', 'maxPermissibleWeightCH', 'completeMaxPermissibleWeightCH',
'loadCapacityCH', 'lsvaRequired', 'nonResponse', 'typeOfTransport', 'vehicleKind', 'typeOfVhl',
```

```
'vehicleBrand', 'nAxes', 'cantonOfOrigin', 'trailerDevice', 'ramp', 'crane', 'winch', 'vhlKm',
'vhlKmCH', 'grossingFactor']
'''

df_light_transport = pd.read_csv("lwe_transport_13.csv", delimiter=';')
'''
['OID', 'TYPE_OF_VEHICLE', 'TYPE_DESIGNATION', 'BODY_DESIGN_ID', 'TRAILER_HITCH',
'NUMBER_OF_AXLES', 'AUTO_LIFT', 'CRANE', 'WINCH', 'HEAVY_VEHICLE_FEE_CODE', 'EMISSION_CODE',
'CO2_EMISSION', 'AGE_OF_VEHICLE_IN_SURVEY_YEAR', 'CURB_WEIGHT', 'LOADING_CAPACITY', 'LADEN_WEIGHT',
'HAULED_LOAD', 'GROSS_VEHICLE_MASS', 'Kanton', 'STRATA_ID', 'survey_date', 'MAIN_USE_GOODS_TRANSPORT',
'MAIN_USE_SERVICE', 'MAIN_USE_EMPTY_JOURNEY', 'MAIN_USE_PASSENGER', 'MAIN_USE_RENTAL', 'MAIN_USE_PRIVATE',
'MAIN_USE_OTHER', 'Warengruppe', 'Warengewicht', 'Gefahrgutcode', 'Quelle_Land', 'Quelle_NPA', 'FromNuts',
'Ziel_Land', 'Ziel_NPA', 'ToNuts', 'Anzahl_Stopps', 'TRANSPORT_ENDS_NEXT_DAY_YES_NO', 'transportKm',
'transportKmCH', 'NOGA_2008', 'wh_tot_cal']
'''


df_light_vehicle = pd.read_csv("lwe_vehicle_13.csv", delimiter=";", encoding='latin-1')
'''
['OID', 'TYPE_OF_VEHICLE', 'TYPE_DESIGNATION', 'BODY_DESIGN_ID', 'TRAILER_HITCH', 'NUMBER_OF_AXLES',
'AUTO_LIFT', 'CRANE', 'WINCH', 'HEAVY_VEHICLE_FEE_CODE', 'EMISSION_CODE', 'CO2_EMISSION',
'AGE_OF_VEHICLE_IN_SURVEY_YEAR', 'CURB_WEIGHT', 'LOADING_CAPACITY', 'LADEN_WEIGHT', 'HAULED_LOAD',
'GROSS_VEHICLE_MASS', 'Kanton', 'STRATA_ID', 'survey_date', 'KM_ABROAD', 'KM_NIGHT', 'KM_TOTAL_CH',
'KM_TOTAL', 'TRAILER_YES_NO', 'TRAILER_CURB_WEIGHT', 'TRAILER_CAPACITY', 'MAIN_USE_GOODS_TRANSPORT',
'MAIN_USE_SERVICE', 'MAIN_USE_EMPTY_JOURNEY', 'MAIN_USE_PASSENGER', 'MAIN_USE_RENTAL', 'MAIN_USE_PRIVATE',
'MAIN_USE_OTHER', 'NOGA_2008', 'wh_tot_cal']
'''


df_cross_border = pd.read_csv("cross_border_14.csv", delimiter=";", encoding='latin-1')
'''
['OID', 'COUNTRY_REGISTRATION', 'VEHICLE_TYPE', 'AXLE_CONFIGURATION', 'EMISSION_CODE',
'UNLADEN_WEIGHT', 'GROSS_WEIGHT', 'TRANSPORT_MODE', 'BORDER_CROSSING_IN', 'BORDER_CROSSING_OUT',
'DIRECTION', 'COUNTRY_OF_LOADING', 'ORIGIN', 'CH_MUNICIPALITY_ORIGIN', 'COUNTRY_OF_UNLOADING',
'DESTINATION', 'CH_MUNICIPALITY_DESTINATION', 'TYPE_OF_GOOD', 'DANGEROUS_GOOD', 'TYPE_OF_CARGO',
'WEIGHT_OF_GOOD', 'KM_PERFORMANCE', 'WEIGHTING_FACTOR', 'DIVISOR']
'''

```

```
population_plz = pd.read_csv("population_plz.csv", delimiter=";", encoding='latin-1')
population_plz_sum = population_plz.groupby(['plz']).sum()
```

Python file that produces histograms and boxplots for domestic truck transportations' distance for ***GTE13***, ***GTE14***, and ***GTE14***. This file also produces csv file that can be read by ArcGIS or QGIS for determining the areas that have large amounts of transportation going outwards or inwards.

```
import matplotlib.pyplot as plt
import seaborn as sns

from dataframe_creation import *

swiss_plz = pd.read_csv("CH_PLZ.csv", delimiter=';',encoding='latin-1')
plz_count = swiss_plz[['PLZ', 'E', 'N']]
plz_count.to_csv("plz_brief.csv", index=False)



df_GTE13 = df_transport13[df_transport13['fromLand'].str.contains("CH")]
df_GTE13 = df_GTE13[df_GTE13['toLand'].str.contains("CH")]
df_GTE13 = df_GTE13[["ernr", "journeyId", 'fromPlz', 'toPlz', 'fromNuts', 'toNuts','tonCH', 'transportKmCH']].copy()
df_GTE13.name = 'GTE 13 CH'

df_GTE14 = df_transport14[df_transport14['fromLand'].str.contains("CH")]
df_GTE14 = df_GTE14[df_GTE14['toLand'].str.contains("CH")]
df_GTE14 = df_GTE14[["ernr", "journeyId", 'fromPlz', 'toPlz', 'fromNuts', 'toNuts','tonCH', 'transportKmCH']].copy()
df_GTE14.name = 'GTE 14 CH'

df_GTE17 = df_transport17[df_transport17['fromLand'].str.contains("CH")]
df_GTE17 = df_GTE17[df_GTE17['toLand'].str.contains("CH")]
df_GTE17 = df_GTE17[["ernr", "journeyId", 'fromPlz', 'toPlz', 'fromNuts', 'toNuts','tonCH', 'transportKmCH']].copy()
df_GTE17.name = 'GTE 17 CH'

#  print(np.average(df_GTE13["transportKmCH"]),np.average(df_GTE14["transportKmCH"]),np.average(df_GTE17["transportKmCH"]))
#  print(np.var(df_GTE13["tonCH"]), np.var(df_GTE14["tonCH"]), np.var(df_GTE17["tonCH"]))
#  print(np.average(df_GTE13["tonCH"]),np.average(df_GTE14["tonCH"]),np.average(df_GTE17["tonCH"]))
#  print(np.var(df_GTE13["tonCH"]),np.var(df_GTE14["tonCH"]),np.var(df_GTE17["tonCH"]))
#

def hist_box_plot(dataframe, index):
    sns.set(style="darkgrid")

    x = dataframe.iloc[:, 7]
    f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85)})

    sns.boxplot(x, ax=ax_box, fliersize=1, color="r")
    sns.distplot(x, ax=ax_hist, kde=False, color="r")

    ax_box.set(yticks=[])
```

```
    sns.despine(ax=ax_hist)
    sns.despine(ax=ax_box, left=True)
    plt.xlim(0, None)
    plt.ylim(0, None)
    plt.subplots_adjust(top=0.9)
    plt.suptitle("Histogram and Boxplot of {0} from {1}".format(list(
    dataframe.head(0))[index], dataframe.name), fontsize=16)
    plt.show()

hist_box_plot(df_GTE13, 7)
hist_box_plot(df_GTE14, 7)
hist_box_plot(df_GTE17, 7)

dist_13 = df_GTE13['transportKmCH'].value_counts(sort=False).sort_index()
dist_14 = df_GTE14['transportKmCH'].value_counts(sort=False).sort_index()
dist_17 = df_GTE17['transportKmCH'].value_counts(sort=False).sort_index()

#print(dist_14)

# ======================================
# weight_13 = df_GTE13['tonCH'].value_counts(sort=False).sort_index()
# weight_14 = df_GTE14['tonCH'].value_counts(sort=False).sort_index()
# weight_17 = df_GTE17['tonCH'].value_counts(sort=False).sort_index()
#
# ======================================
origin_13 = df_GTE13['fromPlz'].value_counts(sort=False).sort_index()
origin_14 = df_GTE14['fromPlz'].value_counts(sort=False).sort_index()
origin_17 = df_GTE17['fromPlz'].value_counts(sort=False).sort_index()

dest_13 = df_GTE13['toPlz'].value_counts(sort=False).sort_index()
dest_14 = df_GTE14['toPlz'].value_counts(sort=False).sort_index()
dest_17 = df_GTE17['toPlz'].value_counts(sort=False).sort_index()

uniq_ernr_13 = df_GTE13['ernr'].value_counts(sort=False).sort_index()
uniq_ernr_14 = df_GTE14['ernr'].value_counts(sort=False).sort_index()
uniq_ernr_17 = df_GTE17['ernr'].value_counts(sort=False).sort_index()

# ======================================
#
# weight_13.to_csv("freq_weight_13.csv", index=False)
# weight_14.to_csv("freq_weight_14.csv", index=False)
# weight_17.to_csv("freq_weight_17.csv", index=False)
#
# ======================================
origin_13.to_csv("freq_origin_13.csv", index=True)
origin_14.to_csv("freq_origin_14.csv", index=True)
origin_17.to_csv("freq_origin_17.csv", index=True)

dest_13.to_csv("freq_dest_13.csv", index=True)
dest_14.to_csv("freq_dest_14.csv", index=True)
dest_17.to_csv("freq_dest_17.csv", index=True)

uniq_ernr_13.to_csv("uniq_ernr_13.csv", index=True)
uniq_ernr_14.to_csv("uniq_ernr_14.csv", index=True)
uniq_ernr_17.to_csv("uniq_ernr_17.csv", index=True)

# ======================================
# weight_13 = pd.read_csv("freq_weight_13.csv")
```

```
# weight_14 = pd.read_csv("freq_weight_14.csv")
# weight_17 = pd.read_csv("freq_weight_17.csv")
# ======================================

origin_13 = pd.read_csv("freq_origin_13.csv", names = ['fromPLZ13','fromPLZ13_Count'])
origin_14 = pd.read_csv("freq_origin_14.csv", names = ['fromPLZ14','fromPLZ14_Count'])
origin_17 = pd.read_csv("freq_origin_17.csv", names = ['fromPLZ17','fromPLZ17_Count'])

dest_13 = pd.read_csv("freq_dest_13.csv", names = ['toPLZ13','toPLZ13_Count'])
dest_14 = pd.read_csv("freq_dest_14.csv", names = ['toPLZ14','toPLZ14_Count'])
dest_17 = pd.read_csv("freq_dest_17.csv", names = ['toPLZ17','toPLZ17_Count'])

uniq_ernr_13 = pd.read_csv("uniq_ernr_13.csv", names = ['ernr','ernr_Count'])
uniq_ernr_14 = pd.read_csv("uniq_ernr_14.csv", names = ['ernr','ernr_Count'])
uniq_ernr_17 = pd.read_csv("uniq_ernr_17.csv", names = ['ernr','ernr_Count'])

origin_13.to_csv("freq_origin_13.csv", index=False)
origin_14.to_csv("freq_origin_14.csv", index=False)
origin_17.to_csv("freq_origin_17.csv", index=False)

dest_13.to_csv("freq_dest_13.csv", index=False)
dest_14.to_csv("freq_dest_14.csv", index=False)
dest_17.to_csv("freq_dest_17.csv", index=False)

origin_13 = pd.read_csv("freq_origin_13.csv")
origin_14 = pd.read_csv("freq_origin_14.csv")
origin_17 = pd.read_csv("freq_origin_17.csv")

dest_13 = pd.read_csv("freq_dest_13.csv")
dest_14 = pd.read_csv("freq_dest_14.csv")
dest_17 = pd.read_csv("freq_dest_17.csv")

merge_ori_dest = plz_count.merge(origin_13, left_on='PLZ', right_on='fromPLZ13')
merge_ori_dest = merge_ori_dest.merge(origin_14, left_on='PLZ', right_on='fromPLZ14')
merge_ori_dest = merge_ori_dest.merge(origin_17, left_on='PLZ', right_on='fromPLZ17')
merge_ori_dest = merge_ori_dest.merge(dest_13, left_on='PLZ', right_on='toPLZ13')
merge_ori_dest = merge_ori_dest.merge(dest_14, left_on='PLZ', right_on='toPLZ14')
merge_ori_dest = merge_ori_dest.merge(dest_17, left_on='PLZ', right_on='toPLZ17')
merge_ori_dest = merge_ori_dest.merge(population_plz_sum, left_on='PLZ', right_on='plz')

merge_ori_dest.to_csv("merge_ori_dest.csv", index=False)
```

Python file that contains functions of energy model.

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np

# https://www.transportenvironment.org/sites/te/files/publications/20180725
    _T%26E_Battery_Electric_Trucks_EU_FINAL.pdf
def flat_truck_energy_demand(velo_m_s, truck_height, truck_width,
    truck_weight, goods_weight, distance_km):
     rho_air = 1.2
     v = velo_m_s
     g = 9.81
     crr = 0.0055
     cd = 0.6
     cd_tesla = 0.35
     area = truck_height * truck_width
    m = truck_weight + goods_weight
     f = 0.5 * rho_air * v * v * area * cd + m * g * crr
     # in kWh
     energy_per_100km = (f * 100 * 1000) / 1000 / 3600
     energy = (distance_km/100) * energy_per_100km
     return energy

def slope_truck_energy_demand(velo_m_s, truck_height, truck_width,
    truck_weight, goods_weight, distance_km, elev_gain_m):
     rho_air = 1.2
     v = velo_m_s
     g = 9.81
     crr = 0.0055
     cd = 0.6
     cd_tesla = 0.35
     area = truck_height * truck_width
    m = truck_weight + goods_weight
     dist_prop = (distance_km * 1000) / (velo_m_s * 3600)
     if distance_km == 0:
         alpha = 0
     else:
         alpha = np.arctan(elev_gain_m / (distance_km * 1000) )

     if alpha >= 0:
         f = 0.5 * rho_air * v * v * area * cd + m * g * crr * np.cos(alpha)
     + m * g * np.sin(alpha)
         energy_per_ride = (f * distance_km * 1000) / 1000 / 3600
     else:
         f = 0.5 * rho_air * v * v * area * cd + m * g * crr * np.cos(alpha)
     + m * g * np.sin(alpha)
         energy_per_ride = (f * distance_km * 1000) / 1000 / 3600 -
    dist_prop*(0.5 * m * velo_m_s * velo_m_s / (1000 * 3600)) * 0.7
     # in kWh

     return energy_per_ride
```

Python file that creates routes way-points information for all entries in ***GTE17*** (***GTE13*** and ***GTE14*** are handled with the same codes except variable names have been changed accordingly); part of the codes comes from Davlantes (2015):

```
import requests
import urllib
import geopy.distance
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np
```

```python

from data_extraction import df_GTE17

def decode_polyline(origin_lon, origin_lat, dest_lon, dest_lat):

    url_route = "http://ikgoeco.ethz.ch/osrm/route/v1/car/{0},{1};{2},{3}".format(
        origin_lon, origin_lat, dest_lon, dest_lat)

    response = requests.get(url_route).json()
    polyline_str = response["routes"][0]["geometry"]

    # original in: https://stackoverflow.com/questions/15380712/how-to-decode-polylines-from-google-maps-direction-api-in-php

    index, lat, lng = 0, 0, 0
    coordinates = []
    changes = {'latitude': 0, 'longitude': 0}
    while index < len(polyline_str):
        for unit in ['latitude', 'longitude']:
            shift, result = 0, 0

            while True:
                byte = ord(polyline_str[index]) - 63
                index+=1
                result |= (byte & 0x1f) << shift
                shift += 5
                if not byte >= 0x20:
                    break

            if (result & 1):
                changes[unit] = ~(result >> 1)
            else:
                changes[unit] = (result >> 1)

        lat += changes['latitude']
        lng += changes['longitude']

        coordinates.append((lat / 100000.0, lng / 100000.0))

    return coordinates

swiss_plz = pd.read_csv("post-codes.csv", encoding='latin-1')
plz_count = swiss_plz[['zip', 'lat', 'lng']]
plz_count_mean = plz_count.groupby(['zip']).mean()
plz_count_mean.to_csv("plz_count_mean.csv", index=True)
plz_count_mean = pd.read_csv("plz_count_mean.csv", encoding='latin-1')

dict_plz = plz_count_mean.set_index('zip').to_dict()


df_GTE17["origin_lat"] = np.nan
df_GTE17["origin_lon"] = np.nan
df_GTE17["dest_lat"] = np.nan
df_GTE17["dest_lon"] = np.nan
df_GTE17["polyline"] = None

counter = 0
```

```python
for idx_0 in range(0, len(df_GTE17)):
    origin_plz = df_GTE17.iloc[idx_0]["fromPlz"]
    dest_plz = df_GTE17.iloc[idx_0]["toPlz"]
    print(origin_plz, dest_plz)
    print(counter)
    df_GTE17["origin_lat"].iloc[idx_0] = dict_plz["lat"].get(int(origin_plz
    ), 46.818200)
    df_GTE17["origin_lon"].iloc[idx_0] = dict_plz["lng"].get(int(origin_plz
    ), 8.227500)

    df_GTE17["dest_lat"].iloc[idx_0] = dict_plz["lat"].get(int(dest_plz),
    46.81820)
    df_GTE17["dest_lon"].iloc[idx_0] = dict_plz["lng"].get(int(dest_plz),
    8.227500)

    df_GTE17["polyline"].iloc[idx_0] = decode_polyline(
            df_GTE17["origin_lon"].iloc[idx_0],
            df_GTE17["origin_lat"].iloc[idx_0],
            df_GTE17["dest_lon"].iloc[idx_0],
            df_GTE17["dest_lat"].iloc[idx_0])

    counter += 1

df_GTE17[["ernr","journeyId","origin_lat","origin_lon","dest_lat","dest_lon
    ", "tonCH", "polyline"]].to_csv("GTE17_routes.csv")
```

Python file that creates elevation for all routes calculated in ***GTE17*** (***GTE13*** and ***GTE14*** are handled with the same codes except variable names have been changed accordingly):

```python
import requests
import urllib
import geopy.distance
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np
from elevation_calculation import *

def get_distance(origin_lon, origin_lat, dest_lon, dest_lat):
    url_route = "http://ikgoeco.ethz.ch/osrm/route/v1/car/{0},{1};{2},{3}".
    format(
        origin_lon, origin_lat, dest_lon, dest_lat)

    response = requests.get(url_route).json()
    distance_km = response["routes"][0]["distance"]

    return float(distance_km / 1000)

def get_elevation(lat, lon):
    elev = read_elevation_file(lon, lat)
    if elev == 'Hours, longitude or latitude out of bounds.':
        elev = 555.555
        # random assign an elevation as the gain and loss will be equaled
        # after a true elevation is gained.
    return float(elev)


df_GTE17_addElev = pd.read_csv("GTE17_route.csv", encoding='latin-1')

df_GTE17_addElev["elev"] = np.empty
```

```
df_GTE17_addElev["net_elev_gain"] = np.nan
#df_GTE17_addElev["distance"] = np.nan

counter = 0
for idx_0 in range(0, len(df_GTE17_addElev)):
    lst_elev = []
    waypoint_lst = ([float(i) for i in (((df_GTE17_addElev["polyline"].iloc[idx_0].replace("[","").
                                         replace("(","").replace(")","").replace("]","").
    replace(",",""))).split())])

    for idx_1 in range(1, (int(len(waypoint_lst) / 2)) + 1):
        lst_elev.append(get_elevation(waypoint_lst[2 * idx_1 - 2], waypoint_lst[2 * idx_1 - 1]))
#         df_GTE17_addElev["distance"].iloc[idx_0] = get_distance(
#             df_GTE17_addElev["origin_lon"].iloc[idx_0],
#             df_GTE17_addElev["origin_lat"].iloc[idx_0],
#             df_GTE17_addElev["dest_lon"].iloc[idx_0],
#             df_GTE17_addElev["dest_lat"].iloc[idx_0])

    net_gain = 0
    for idx_2 in range(len(lst_elev) - 1):
        net_gain += (lst_elev[idx_2 + 1] - lst_elev[idx_2])

    print(net_gain)
    df_GTE17_addElev["elev"].iloc[idx_0] = lst_elev
    df_GTE17_addElev["net_elev_gain"].iloc[idx_0] = net_gain
    print(counter)
    counter += 1

df_GTE17_addElev.to_csv("GTE17_route_elev.csv")
```

Python file that calculates energy demand for each of the transportations in ***GTE17*** based on route and elevation profile created from previous python files(***GTE13*** and ***GTE14*** are handled with the same codes except variable names have been changed accordingly):

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np
import truck_energy_model as tem
import geopy.distance as distance

df_GTE17_addAveEner = pd.read_csv("GTE17_route_elev_dist_backup.csv", encoding='latin-1')

# Add weight of packages into consideration
df_GTE17_addAveEner["tonCH_Package"] = df_GTE17_addAveEner["tonCH"] * 1.05


# Prepare car category
df_GTE17_addAveEner["car_category"] = 0

df_GTE17_addAveEner.loc[df_GTE17_addAveEner['tonCH_Package'] <2890, ['car_category']] = 1
df_GTE17_addAveEner.loc[df_GTE17_addAveEner['tonCH_Package'] >4615, ['car_category']] = 3
df_GTE17_addAveEner.loc[df_GTE17_addAveEner['car_category'] == 0, ['car_category']] = 2
```

```

df_GTE17_car = df_GTE17_addAveEner.groupby(['ernr'])['car_category'].max().to_frame()
df_GTE17_car['ernr'] = df_GTE17_car.index

df_GTE17_addAveEner.drop('car_category', axis=1, inplace=True)

df_GTE17_addAveEner = df_GTE17_addAveEner.merge(df_GTE17_car, left_on=['ernr'], right_on=['ernr'])


# Average 70 km/hr
df_GTE17_addAveEner["average_energy_70"] = np.nan
# Average 60 km/hr
df_GTE17_addAveEner["average_energy_60"] = np.nan
# Average 45 km/hr
df_GTE17_addAveEner["average_energy_45"] = np.nan



counter = 0
for idx_0 in range(0, len(df_GTE17_addAveEner)):
    waypoint_lst = ([float(i) for i in (((df_GTE17_addAveEner["polyline"].iloc[idx_0].replace("[","").
                                                        replace("(","").replace(")","").replace("]","").
    replace(",",""))).split())])

    elev_lst = ([float(i) for i in (((df_GTE17_addAveEner["elev"].iloc[idx_0].replace("[","").
                                                        replace("(","").replace(")","").replace("]","").
    replace(",",""))).split())])

    elev_diff_lst = []
    for idx_0_1 in range(1, len(elev_lst)):
        elev_diff_lst.append(elev_lst[idx_0_1]-elev_lst[idx_0_1 -1])

    polyline_lat = []
    polyline_lon = []
    for idx_1 in range(1, (int(len(waypoint_lst) / 2)) + 1):
        polyline_lat.append(waypoint_lst[2 * idx_1 - 2])
        polyline_lon.append(waypoint_lst[2 * idx_1 - 1])

#    BYD 7.5 TON MCV
    if df_GTE17_addAveEner["car_category"].iloc[idx_0] == 1:
        truck_width = 2.050
        truck_height = 3.150
        truck_weight = 4300
#    BYD 11 TON MCV
    elif df_GTE17_addAveEner["car_category"].iloc[idx_0] == 2:
        truck_width = 2.500
        truck_height = 3.030
        truck_weight = 5885
#    BYD Q3M
    else:
        truck_width = 2.500
        truck_height = 3.085
```

```
        truck_weight = 10500


    length = len(polyline_lat)
    decompose_dist = []
#     decompose_dist_accumulated = [0]
#
    for idx_2 in range(1 , length):
        sect_dist = distance.distance((polyline_lat[idx_2], polyline_lon[
idx_2]),(polyline_lat[idx_2 -1], polyline_lon[idx_2 -1])).km
        decompose_dist.append(sect_dist)

    energy_total70 = 0
    for idx_3 in range(len(decompose_dist)):
        energy_total70 += tem.slope_truck_energy_demand(19.45, truck_height
, truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], decompose_dist[idx_3], elev_diff_lst[idx_3])
#     In case if transport happened in the same PLZ zone.
    if energy_total70 == 0:
        energy_total70 = tem.flat_truck_energy_demand(19.45, truck_height,
truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], df_GTE17_addAveEner["transportKmCH"].iloc[idx_0])

    df_GTE17_addAveEner["average_energy_70"].iloc[idx_0] = energy_total70
    print(energy_total70,"----------------", counter)

    energy_total60 = 0
    for idx_3 in range(len(decompose_dist)):
        energy_total60 += tem.slope_truck_energy_demand(16.67, truck_height
, truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], decompose_dist[idx_3], elev_diff_lst[idx_3])
#     In case if transport happened in the same PLZ zone.
    if energy_total60 == 0:
        energy_total60 = tem.flat_truck_energy_demand(16.67, truck_height,
truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], df_GTE17_addAveEner["transportKmCH"].iloc[idx_0])

    df_GTE17_addAveEner["average_energy_60"].iloc[idx_0] = energy_total60
    print(energy_total60,"----------------", counter)

    energy_total45 = 0
    for idx_3 in range(len(decompose_dist)):
        energy_total45 += tem.slope_truck_energy_demand(12.5, truck_height,
 truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], decompose_dist[idx_3], elev_diff_lst[idx_3])
#     In case if transport happened in the same PLZ zone.
    if energy_total45 == 0:
        energy_total45 = tem.flat_truck_energy_demand(12.5, truck_height,
truck_width, truck_weight,
                                                    df_GTE17_addAveEner["tonCH_Package"].
iloc[idx_0], df_GTE17_addAveEner["transportKmCH"].iloc[idx_0])

    df_GTE17_addAveEner["average_energy_45"].iloc[idx_0] = energy_total45
```

```
    print(energy_total45,"----------------", counter)


    counter += 1

# 5% weight is belonging to the packages

df_GTE17_addAveEner.to_csv("GTE17_route_elev_dist_energy.csv", index=False)
```

Python file that calculates energy demand based on different general efficiencies and also different battery recharging status (different SOC):

```
import pandas as pd
import numpy as np
from truck_energy_model import flat_truck_energy_demand

# Read file contain theoritical energy consumption of each transportation
df_17dom = pd.read_csv("GTE17_route_elev_dist_energy.csv")
#list(df_17dom)


# Read week.csv file to obtain the number of successful surveys each week
df_week17 = pd.read_csv("week17.csv", delimiter = ";")
df_week17.groupby(["week"]).count()

# Simplify the week survey data structure
df_week17 = df_week17[['ernr', 'week']]

# Merge the week of entry to the energy consumption dataframe and store it
df_17dom = df_17dom.merge(df_week17, left_on="ernr", right_on="ernr")
df_17dom.loc[df_17dom['average_energy_70'] <0, ['average_energy_70']] = 0
df_17dom.to_csv("GTE17week_route_elev_dist_energy.csv")


# Read foreign entries processed data
df_transport17_foreign_1 = pd.read_csv("transport17.csv", delimiter=';')
df_transport17_foreign_1 = df_transport17_foreign_1[~
    df_transport17_foreign_1['fromLand'].str.contains("CH")]
df_transport17_foreign_1 = df_transport17_foreign_1[~
    df_transport17_foreign_1['toLand'].str.contains("CH")]

df_transport17_foreign_2 = pd.read_csv("transport17.csv", delimiter=';')
df_transport17_foreign_2 = df_transport17_foreign_2[~
    df_transport17_foreign_2['fromLand'].str.contains("CH")]

df_transport17_foreign_3 = pd.read_csv("transport17.csv", delimiter=';')
df_transport17_foreign_3 = df_transport17_foreign_3[~
    df_transport17_foreign_3['toLand'].str.contains("CH")]

df_17for = pd.concat([df_transport17_foreign_1, df_transport17_foreign_2,
    df_transport17_foreign_3])


df_17for["car_category"] = 0
df_17for.loc[df_17for['tonCH'] <2890/1.05, ['car_category']] = 1
df_17for.loc[df_17for['tonCH'] >4615/1.05, ['car_category']] = 3
df_17for.loc[df_17for['car_category'] == 0, ['car_category']] = 2
df_17for_car = df_17for.groupby(['ernr'])['car_category'].max().to_frame()
df_17for_car['ernr'] = df_17for_car.index
```

```python
df_17for.drop('car_category', axis=1, inplace=True)
df_17for = df_17for.merge(df_17for_car, left_on=['ernr'], right_on=['ernr'])
# Create new column for foreign entry to calculate energy
# using flat energy model
df_17for["average_energy_70"] = np.nan


df_17for['average_energy_70'] =  df_17for.loc[df_17for['car_category'] == 1, ["average_energy_70"]]=flat_truck_energy_demand(19.444, 3.15, 2.05, 4300, df_17for["tonCH"]*1.05,df_17for["transportKmCH"])
df_17for['average_energy_70'] =  df_17for.loc[df_17for['car_category'] == 2, ["average_energy_70"]]=flat_truck_energy_demand(19.444, 2.36, 2.25, 5885, df_17for["tonCH"]*1.05,df_17for["transportKmCH"])
df_17for['average_energy_70'] =  df_17for.loc[df_17for['car_category'] == 3, ["average_energy_70"]]=flat_truck_energy_demand(19.444, 3.085, 2.5, 10500, df_17for["tonCH"]*1.05,df_17for["transportKmCH"])

# take the ratio of foreign transportation energy demand / domestic energy demand
foreignEnergyRatio = df_17for["average_energy_70"].sum() / (df_17for["average_energy_70"].sum()+df_17dom["average_energy_70"].sum())

# Sum the total energy demand by each week and store them in a new dataframe with each week as a row
df_17dom_week = df_17dom.groupby(["week"]).sum()

# Sum the energy demand for every PLZ
df_17dom_plzOri = df_17dom[["distance","fromPlz","origin_lat","origin_lon","average_energy_70", "tonCH", "transportKmCH"]].groupby(["fromPlz"]).sum()
df_17dom_plzOri["fromPlz"] = df_17dom_plzOri.index

#https://insideevs.com/news/332584/efficiency-compared-battery-electric-73-hydrogen-22-ice-13/
df_17dom_plzOri["estEnergy77EffikWh"] = ((df_17dom_plzOri["average_energy_70"] / (0.77 * (1-foreignEnergyRatio))) / len(df_week17)) * 52 * 54000
df_17dom_plzOri["estEnergy85EffikWh"] = ((df_17dom_plzOri["average_energy_70"] / (0.85 * (1-foreignEnergyRatio))) / len(df_week17)) * 52 * 54000
df_17dom_plzOri["estEnergy90EffikWh"] = ((df_17dom_plzOri["average_energy_70"] / (0.90 * (1-foreignEnergyRatio))) / len(df_week17)) * 52 * 54000
df_17dom_plzOri["estEnergy81EffikWh"] = ((df_17dom_plzOri["average_energy_70"] / (0.81 * (1-foreignEnergyRatio))) / len(df_week17)) * 52 * 54000

df_17dom_plzOri.to_csv("GTE17_annualEnergy_by_oriPLZ.csv", index=False)

# Sum the energy demand for each driver
df_17dom_driver = df_17dom[["distance","ernr","average_energy_70","tonCH"]].groupby(["ernr"]).sum()
df_17dom_driver["ernr"] = df_17dom_driver.index
df_17dom_driver["dailyDistAverage5"] = df_17dom_driver["distance"]/5
df_17dom_driver["dailyDistAverage7"] = df_17dom_driver["distance"]/7
df_17dom_driver["dailyEnerAverage5"] = df_17dom_driver["average_energy_70"]/5
```


yink@student.ethz.ch

```
df_17dom_driver["dailyEnerAverage7"] = df_17dom_driver["average_energy_70"]/7
df_17dom["car_category"] = 0
df_17dom.loc[df_17dom['tonCH'] <2890/1.05, ['car_category']] = 1
df_17dom.loc[df_17dom['tonCH'] >4615/1.05, ['car_category']] = 3
df_17dom.loc[df_17dom['car_category'] == 0, ['car_category']] = 2
df_17dom_car = df_17dom.groupby(['ernr'])['car_category'].max().to_frame()
df_17dom_car['ernr'] = df_17dom_car.index
df_17dom.drop('car_category', axis=1, inplace=True)
df_17dom = df_17dom.merge(df_17dom_car, left_on=['ernr'], right_on=['ernr'])
df_17dom_driver = df_17dom_driver.merge(df_17dom_car, left_on=['ernr'], right_on=['ernr'])
df_17dom_driver["battery_capacity"] = 0
df_17dom_driver.loc[df_17dom_driver['car_category'] ==1, ['battery_capacity']] = 145
df_17dom_driver.loc[df_17dom_driver['car_category'] ==2, ['battery_capacity']] = 148.5
df_17dom_driver.loc[df_17dom_driver['car_category'] ==3, ['battery_capacity']] = 207


df_17for_driver = df_17for[["average_energy_70","ernr","transportKm","tonCH"]].groupby(["ernr"]).sum()
df_17for_driver = df_17for_driver.merge(df_17for_car, left_on=['ernr'], right_on=['ernr'])
df_17for_driver["battery_capacity"] = 0
df_17for_driver.loc[df_17for_driver['car_category'] ==1, ['battery_capacity']] = 145
df_17for_driver.loc[df_17for_driver['car_category'] ==2, ['battery_capacity']] = 148.5
df_17for_driver.loc[df_17for_driver['car_category'] ==3, ['battery_capacity']] = 207
df_17for_driver["dailyDistAverage5"] = df_17for_driver["transportKm"]/5
df_17for_driver["dailyDistAverage7"] = df_17for_driver["transportKm"]/7
df_17for_driver["dailyEnerAverage5"] = df_17for_driver["average_energy_70"]/5
df_17for_driver["dailyEnerAverage7"] = df_17for_driver["average_energy_70"]/7
df_17for_driver.rename(columns={'transportKm':'distance'}, inplace=True)

df_17All_car = pd.concat([df_17dom_car, df_17for_car])
df_17All_car = df_17All_car.groupby(["ernr"]).max()
df_17All_car["ernr"] = df_17All_car.index

df_17All_driver = pd.concat([df_17dom_driver, df_17for_driver])
df_17All_driver = df_17All_driver[["average_energy_70","ernr","distance","tonCH","dailyDistAverage5","dailyDistAverage7", "dailyEnerAverage5", "dailyEnerAverage7"]].groupby(["ernr"]).sum()
df_17All_car.rename(columns={'car_category_x':'car_category'}, inplace=True)
df_17All_driver["ernr"] = df_17All_driver.index

df_17All_driver = df_17All_car.merge(df_17All_driver, left_on=["ernr"], right_on=["ernr"])

df_17All_driver["battery_capacity"] = 0
df_17All_driver.loc[df_17All_driver['car_category'] ==1, ['battery_capacity
```

```
']] = 145
df_17All_driver.loc[df_17All_driver['car_category'] ==2, ['battery_capacity']] = 148.5
df_17All_driver.loc[df_17All_driver['car_category'] ==3, ['battery_capacity']] = 207

# Consider each vehicle's demand on daily basis
df_17All_driver["dailyEnerAverage5_Eff77"] = df_17All_driver["dailyEnerAverage5"] / 0.77
df_17All_driver["dailyEnerAverage7_Eff77"] = df_17All_driver["dailyEnerAverage7"] / 0.77
df_17All_driver["dailyEnerAverage5_Eff85"] = df_17All_driver["dailyEnerAverage5"] / 0.85
df_17All_driver["dailyEnerAverage7_Eff85"] = df_17All_driver["dailyEnerAverage7"] / 0.85
df_17All_driver["dailyEnerAverage5_Eff81"] = df_17All_driver["dailyEnerAverage5"] / 0.81
df_17All_driver["dailyEnerAverage7_Eff81"] = df_17All_driver["dailyEnerAverage7"] / 0.81

df_17All_driver["dailyEnerAverage5_Eff77_soc90"] = df_17All_driver["dailyEnerAverage5"] / 0.77 / 0.9
df_17All_driver["dailyEnerAverage7_Eff77_soc90"] = df_17All_driver["dailyEnerAverage7"] / 0.77 / 0.9
df_17All_driver["dailyEnerAverage5_Eff85_soc90"] = df_17All_driver["dailyEnerAverage5"] / 0.85 / 0.9
df_17All_driver["dailyEnerAverage7_Eff85_soc90"] = df_17All_driver["dailyEnerAverage7"] / 0.85 / 0.9
df_17All_driver["dailyEnerAverage5_Eff81_soc90"] = df_17All_driver["dailyEnerAverage5"] / 0.81 / 0.9
df_17All_driver["dailyEnerAverage7_Eff81_soc90"] = df_17All_driver["dailyEnerAverage7"] / 0.81 / 0.9

df_17All_driver["dailyEnerAverage5_Eff77_soc80"] = df_17All_driver["dailyEnerAverage5"] / 0.77 / 0.8
df_17All_driver["dailyEnerAverage7_Eff77_soc80"] = df_17All_driver["dailyEnerAverage7"] / 0.77 / 0.8
df_17All_driver["dailyEnerAverage5_Eff85_soc80"] = df_17All_driver["dailyEnerAverage5"] / 0.85 / 0.8
df_17All_driver["dailyEnerAverage7_Eff85_soc80"] = df_17All_driver["dailyEnerAverage7"] / 0.85 / 0.8
df_17All_driver["dailyEnerAverage5_Eff81_soc80"] = df_17All_driver["dailyEnerAverage5"] / 0.81 / 0.8
df_17All_driver["dailyEnerAverage7_Eff81_soc80"] = df_17All_driver["dailyEnerAverage7"] / 0.81 / 0.8

df_17All_driver["dailyEnerAverage5_Eff77_soc70"] = df_17All_driver["dailyEnerAverage5"] / 0.77 / 0.7
df_17All_driver["dailyEnerAverage7_Eff77_soc70"] = df_17All_driver["dailyEnerAverage7"] / 0.77 / 0.7
df_17All_driver["dailyEnerAverage5_Eff85_soc70"] = df_17All_driver["dailyEnerAverage5"] / 0.85 / 0.7
df_17All_driver["dailyEnerAverage7_Eff85_soc70"] = df_17All_driver["dailyEnerAverage7"] / 0.85 / 0.7
df_17All_driver["dailyEnerAverage5_Eff81_soc70"] = df_17All_driver["dailyEnerAverage5"] / 0.81 / 0.7
df_17All_driver["dailyEnerAverage7_Eff81_soc70"] = df_17All_driver["dailyEnerAverage7"] / 0.81 / 0.7
```

```
df_17All_driver["dailyEnerAverage5_Eff77_soc60"] = df_17All_driver["dailyEnerAverage5"] / 0.77 / 0.6
df_17All_driver["dailyEnerAverage7_Eff77_soc60"] = df_17All_driver["dailyEnerAverage7"] / 0.77 / 0.6
df_17All_driver["dailyEnerAverage5_Eff85_soc60"] = df_17All_driver["dailyEnerAverage5"] / 0.85 / 0.6
df_17All_driver["dailyEnerAverage7_Eff85_soc60"] = df_17All_driver["dailyEnerAverage7"] / 0.85 / 0.6
df_17All_driver["dailyEnerAverage5_Eff81_soc60"] = df_17All_driver["dailyEnerAverage5"] / 0.81 / 0.6
df_17All_driver["dailyEnerAverage7_Eff81_soc60"] = df_17All_driver["dailyEnerAverage7"] / 0.81 / 0.6

df_17All_driver["dailyEnerAverage5_Eff77_soc50"] = df_17All_driver["dailyEnerAverage5"] / 0.77 / 0.5
df_17All_driver["dailyEnerAverage7_Eff77_soc50"] = df_17All_driver["dailyEnerAverage7"] / 0.77 / 0.5
df_17All_driver["dailyEnerAverage5_Eff85_soc50"] = df_17All_driver["dailyEnerAverage5"] / 0.85 / 0.5
df_17All_driver["dailyEnerAverage7_Eff85_soc50"] = df_17All_driver["dailyEnerAverage7"] / 0.85 / 0.5
df_17All_driver["dailyEnerAverage5_Eff81_soc50"] = df_17All_driver["dailyEnerAverage5"] / 0.81 / 0.5
df_17All_driver["dailyEnerAverage7_Eff81_soc50"] = df_17All_driver["dailyEnerAverage7"] / 0.81 / 0.5

# df_17All_driver["dailyEnerAverage5_Eff77_soc80_battery"].sum() / len(df_17All_driver)
df_17All_driver["dailyEnerAverage5_Eff77_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc90'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff77_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff81_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc90'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff81_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff85_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc90'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff85_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff77_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc80'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff77_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff85_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc80'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff85_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff81_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc80'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff81_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff77_soc70_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc70'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage5_Eff77_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff85_soc70_battery"] = 0
```

```
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc70'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff85_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff81_soc70_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc70'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff81_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff77_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc60'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff77_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff85_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc60'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff85_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff81_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc60'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff81_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff77_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc50'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff77_soc50_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff85_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc50'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff85_soc50_battery']] = 1
df_17All_driver["dailyEnerAverage5_Eff81_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc50'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage5_Eff81_soc50_battery']] = 1


df_17All_driver["dailyEnerAverage7_Eff77_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff77_soc90'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff77_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff81_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff81_soc90'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff81_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff85_soc90_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage5_Eff85_soc90'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff85_soc90_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff77_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff77_soc80'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff77_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff85_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff85_soc80'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff85_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff81_soc80_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff81_soc80'] <
    df_17All_driver["battery_capacity"], ['
    dailyEnerAverage7_Eff81_soc80_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff77_soc70_battery"] = 0
```


yink@student.ethz.ch

```
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff77_soc70'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff77_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff85_soc70_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff85_soc70'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff85_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff81_soc70_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff81_soc70'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff81_soc70_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff77_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff77_soc60'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff77_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff85_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff85_soc60'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff85_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff81_soc60_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff81_soc60'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff81_soc60_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff77_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff77_soc50'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff77_soc50_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff85_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff85_soc50'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff85_soc50_battery']] = 1
df_17All_driver["dailyEnerAverage7_Eff81_soc50_battery"] = 0
df_17All_driver.loc[df_17All_driver['dailyEnerAverage7_Eff81_soc50'] < df_17All_driver["battery_capacity"], ['dailyEnerAverage7_Eff81_soc50_battery']] = 1


# df_17All_driver["dailyEnerAverage5_Eff77"].quantile(0.55)

df_17All_driver.to_csv("df_17All_driver.csv", index = True)

# Read the week file again and group by week to form a new dataframe with each week as a row
df_17entry_week = pd.read_csv("week17.csv", delimiter = ";")
df_17entry_week = df_17entry_week.groupby(["week"]).count()

# Add additional column to calculate the entire energy demand of Swiss based sampled data
df_17dom_week["swissEnergyTheory"] = np.nan

# By taking the ratio of foreign energy demand/domestic energy demand
# Entire energy demand can be derived by domestic transportation only
df_17dom_week["swissEnergyTheory"] = (df_17dom_week["average_energy_70"] * (54000 / df_17entry_week["ernr"])) / (1 - foreignEnergyRatio)
# df_17All_driver["dailyEnerAverage5_Eff81_soc50_battery"].sum() / len(df_17All_driver)

# in GWh
```

```
df_17dom_week["swissEnergyTheory"].sum() / (1000000)

df_17total_week = df_17dom_week
# Considering different general efficiency
df_17total_week["swissEnergy77"] = df_17total_week["swissEnergyTheory"] /
    0.77
df_17total_week["swissEnergy85"] = df_17total_week["swissEnergyTheory"] /
    0.85
df_17total_week["swissEnergy90"] = df_17total_week["swissEnergyTheory"] /
    0.9
df_17total_week["swissEnergy81"] = df_17total_week["swissEnergyTheory"] /
    0.81

df_17total_week["swissEnergy77_Chargin"] = 1.05 * df_17total_week["
    swissEnergyTheory"] / 0.77
df_17total_week["swissEnergy85_Chargin"] = 1.05 * df_17total_week["
    swissEnergyTheory"] / 0.85
df_17total_week["swissEnergy90_Chargin"] = 1.05 * df_17total_week["
    swissEnergyTheory"] / 0.9
df_17total_week["swissEnergy81_Chargin"] = 1.05 * df_17total_week["
    swissEnergyTheory"] / 0.81

df_17total_week.to_csv("GTE17_domANDfor_energy_demand.csv", index = False)

df_17total_week["swissEnergy85"].sum() / (1000000)


# Validate the number with TonKilometer (upper bound, driving distance is
    defined by the driver)
df_17dom["tonKiloUpper"] = np.nan
df_17dom["tonKiloUpper"] = (df_17dom["transportKmCH"] * df_17dom["tonCH"])
    /(1000)

df_17for["tonKilo"] = np.nan
df_17for["tonKilo"] = (df_17for["transportKm"] * df_17for["tonCH"])/(1000)

print("Total tonKilometer is ", ( (df_17for["tonKilo"].sum() + df_17dom["
    tonKiloUpper"].sum()) / (len(df_week17)) ) * (52 * 54000))

# Validate the number with TonKilometer (lower bound, driving distance is
    defined by OSRM)
df_17dom["tonKiloLower"] = np.nan
df_17dom["tonKiloLower"] = (df_17dom["distance"] * df_17dom["tonCH"])
    /(1000)

df_17for["tonKilo"] = np.nan
df_17for["tonKilo"] = (df_17for["transportKm"] * df_17for["tonCH"])/(1000)

print("Total tonKilometer is ", ( (df_17for["tonKilo"].sum() + df_17dom["
    tonKiloLower"].sum()) / (len(df_week17)) ) * (52 * 54000))
# df_17All_driver["dailyEnerAverage5_Eff77_130"].quantile(0.8)
```

Above-provided codes should be sufficient to derive all preliminary results that can be directly used for further analysis, such as group by origin postal codes, destination postal codes, type of cargo, etc. Codes are also available to be found in the appendix share box hosting at polybox.

Polybox link: https://polybox.ethz.ch/index.php/s/H3K5Gn3lCsDCHOz

Appendix C

Plot-ready csv documents

CSV files that help plot figures in chapter 3 have been provided within the polybox shared folder. Files include derived population information under each postal code area, potential stresses that can be brought under each postal code area normalized by population, and total annual energy demand under each postal code area.

Polybox link: `https://polybox.ethz.ch/index.php/s/H3K5Gn3lCsDCHOz`

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

THE FUTURE OF LOGISTICS: HOW WILL ELECTRIC TRUCKS SHAPE THE FUTURE ENERGY DEMAND AND DISTRIBUTION OF SWITZERLAND?

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
YIN	KEYUAN

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zurich, 11 July 2019

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.