Semesterarbeit des Moduls  
Data Collection, Integration and Pre-processing (CIP02)

ETL-Projektdokumentation

zum Thema  
Entwicklung von Sportgrossanlässen und Bruttoinlandprodukt sowie deren Korrelation von 1950-2019

Hochschule Luzern HSLU

Studiengang: MSc Applied Information and Data Science

Studenten: Régis Andréoli, Micha Käser  
Einreichdatum: 19.05.2021

Gruppennummer: 50

**Inhaltsverzeichnis**

[1 Ausgangslage 3](#_Toc72093959)

[2 Datenquellen 3](#_Toc72093960)

[3 Fragestellungen 4](#_Toc72093961)

[4 Tools 4](#_Toc72093962)

[5 ETL-Prozess 4](#_Toc72093963)

[6 Arbeitsteilung 4](#_Toc72093964)

[7 Lösungsschritte 5](#_Toc72093965)

[7.1 Extract 5](#_Toc72093966)

[7.1.1 A1 5](#_Toc72093967)

[7.1.2 B1 5](#_Toc72093968)

[7.1.3 B2 5](#_Toc72093969)

[7.1.4 B3 5](#_Toc72093970)

[7.1.5 C1 5](#_Toc72093971)

[7.1.6 C2 5](#_Toc72093972)

[7.2 Transform 5](#_Toc72093973)

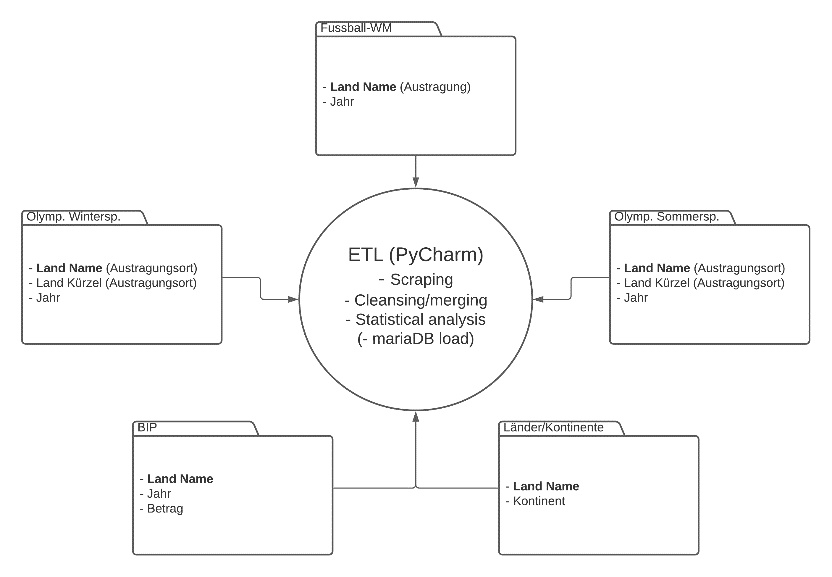
[7.3 Load 5](#_Toc72093974)

[8 Fragestellungen 5](#_Toc72093975)

[9 Reflexion 5](#_Toc72093976)

# Ausgangslage

Während des Masterstudiengangs Applied Information and Data Science erarbeiten die Studierenden innerhalb des Moduls CIP02 eine Semesterarbeit, welche die erlernten Extract-, Transform- und Load-Methoden der Vorlesung in die Praxis umsetzt. Ursprünglich zu dritt gestartet, brach ein Mitstudent das Studium ab – der knappen Zeit geschuldet setzen Régis Andréoli und Micha Käser das Projekt nun zu zweit um.

Als grosse Sportenthusiasten lag es nahe, ein entsprechendes Thema zu wählen. Damit auch eine gewisse Zeitachse (Time Series) gewährleistet ist, fiel die Wahl schlussendlich auf Sportgrossanlässe (Olympiaden/Fussballweltmeisterschaften) und deren Auswirkungen auf das Bruttoinlandprodukt (BIP) der jeweiligen austragenden Nation. Solche Anlässe sind gut dokumentiert und lassen sich viele Jahre zurückverfolgen, auch das BIP als wichtige wirtschaftliche Kennzahl lässt sich bis zur Industrialisierung nachvollziehen. Damit die Resultate verlässlich und aussagekräftig sind, werden die Daten allerdings erst ab der Nachkriegszeit (1950) bis 2019 ausgewertet.

# Datenquellen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Daten** | **Attribute** | **Link** |
| A1[[1]](#footnote-1) | BIP nach Land [[2]](#footnote-2) 1950-2019 | * Land Name * Jahr * Betrag | <https://febpwt.webhosting.rug.nl/Dmn/AggregateXs/PivotShow> |
| B1 | Fussball-Weltmeisterschaften 1930-2018 | * Land Name Austragungsort * Jahr | <https://www.fussball-wm-total.de/History/histehre.html> |
| B2 | Olympische Winterspiele 1924-2018 | * Land Name  Austragungsort * Land Kürzel * Jahr | <https://www.taschenhirn.de/sport/olympische-winterspiele/> |
| B3 | Olympische Sommerspiele 1896-2016 | * Land Name  Austragungsort * Land Kürzel * Jahr | <https://www.taschenhirn.de/sport/olympische-sommerspiele/> |
| C1 | Länder und Kontinente | * Land Name * Kontinent | [https://www.bernhard-gaul.de/wissen/staatenerde.php#uebneu](https://www.bernhard-gaul.de/wissen/staatenerde.php%23uebneu) |
| C2 | Ländercodes | * Land Name * Land Kürzel | https://www.laenderdaten.de/kuerzel/iso\_3166-1.aspx |

Die Tabellen der Sportgrossanlässe werden in einem Dataset zusammengefasst und nach Jahr sortiert. Die Korrelationsberechnung zum BIP werden auf Basis Jahr und Land durchgeführt.

# Fragestellungen

1. Welches Land war zwischen 1950-2019 am häufigsten Gastgeber eines Sportgrossanlasses (Sommer- und Winterolympiade, Fussballweltmeisterschaft)?
2. Welcher Kontinent verzeichnet den grössten BIP-Zuwachs in der Periode von 2000-2019?
3. In welcher Dekade wurden auf welchem Kontinent die meisten Sportgrossanlässe (Sommer- und Winterolympiade, Fussballweltmeisterschaft) durchgeführt?
4. Gibt es eine statistisch feststellbare, signifikante Korrelation zwischen der Durchführung von Sportgrossanlässen und der Veränderung des BIPs der Gastgebernation?

# Tools

|  |  |
| --- | --- |
| **Zweck** | **Tools** |
| Repository Code | GitHub |
| Repository Daten | GitHub/Switch Drive |
| Kommunikation | Slack, Zoom |
| Entwicklungsumgebung | Ubuntu (VM oder nativ), PyCharm, mariaDB, Microsoft Excel/LibreOffice Calc (manuelle Verunreinigung Daten) |
| Diagramme | Lucidchart |

# ETL-Prozess

Nach dem Scrapen der Daten müssen diese verunreinigt werden, da sie bereits eine sehr hohe Qualität aufweisen. Dafür werden in Microsoft Excel oder LibreOffice Calc BIP-Beträge in verschiedene Währungen umgerechnet, Strings mit Zahlen angereichert und unterschiedliche Sprachen verwendet.

Im nächsten Schritt werden diese Verunreinigungen mit PyCharm korrigiert bzw. bereinigt. Die sauberen Tabellen werden gemerged und die Fragestellungen in PyCharm ausgewertet und ausgerechnet. Zusätzlich sollen die sauberen Datasets in eine mariaDB-Datenbank eingelesen werden.

# Arbeitsteilung

|  |  |  |
| --- | --- | --- |
| **Wer** | **Datenquelle** | **Was** |
| Régis Andréoli | A1, B1, C1 | Datenquelle scrapen (B1, C1), CSV exportieren (A1) und Daten mit Python/Pandas bereinigen/anreichern |
| Micha Käser | B2, B3 | Datenquelle scrapen (B2, B3) und Daten mit Python/Pandas bereinigen/anreichern |
| Team | Alle | Daten mergen, joinen und gruppieren. Fragestellungen beantworten. Danach mariaDB aufsetzen und bereinigte Datasets einlesen |

# Lösungsschritte

Im folgenden Abschnitt wird die Projektarbeit nach dem Schema Extract, Transform und Load dokumentiert. Die Lösungswege werden erläutert und

## Extract

In der Extract Phase werden die benötigten Daten für die Problemstellung gesammelt. Die Daten wurden allesamt, bis auf ein CSV-File, mittels eines webscappers von den jeweiligen Webpages heruntergeladen

### A1 – a1\_rgdpna\_stage

Die A1-Daten sind das BIP pro Jahr pro Land. Die Länder-Spezifikation ist jeweils im ISO-3166 alpha-3 festgehalten.

Die Daten werden von einer Webpage manuel via eines CSV heruntergeladen. Der Ursprüngliche Versuch dieses CSV via Python herunterzuladen wurde eingestellt, als sich herausstellte, das das CSV dynamisch gebildet wird via einer Javascript Funktion im HTML Code. Allen Anscheind nach triggert der Button-Klick auf dem Browser eine Javascript -Funktion, die aus dem eigenen HTML Informationen herauszieht um die Funktion selbst zu vervollständigen. Danach wird der komplette Javascript Befehl ausgeführt. Die Daten sowie die Javascript Funktion sind beide in HTML Codes versteckt oder verteilt. Daher wird von der automation des downloads via Python abgesehen, zumal diese Datenquelle nicht der Scrapping Auflage unterliegt.

### B1 – b1\_web\_scrapper.py

Der B1 Web Scrapper ladet und erstellt aus einer Webseite eine List mit den Fussbalweltmeisterschaften und das jeweilige, dazughörige Jahr.

import requests  
from bs4 import BeautifulSoup as bs  
  
*#download html file and convert into soup object*raw\_html = requests.get(**"https://www.fussball-wm-total.de/History/histehre.html"**)  
soup\_html = bs(raw\_html.text, **"html.parser"**)  
  
*#get most distinctive html structure to target input*extract = soup\_html.find(**"table"**, width=**"430"**, border=**"1"**, cellpadding=**"10"**)  
  
*#loop through html lines, eliminate unwanted input values and store as csv*csv\_file = open(**"b1\_wm\_src.csv"**, **"w"**, encoding=**'utf-8'**)  
for p in extract.select(**"tr"**):  
 y = p.select(**"td"**)  
 try:  
 a = y[0].text  
 b = y[1].text  
 csv\_file.write(a + **","** + b + **"**\n**"**)  
 print(a + **","** + b + **"**\n**"**)  
 except:  
 1  
csv\_file.close()

Das Scrappen der WM-Fussballspiele war nicht ganz konventionell. Die webpage www.fussball-wm-total.de hat eine multiple, vernestete Struktur aus Table Tags. Es wurde generell wenig bis gar nicht mit Attributen gearbeitet und so war das Erkennen einer Struktur im HTML Code mehr ‘’Try and Error’’ als methodisch.

Die Bereinigung und das Abspeichern der Daten aus dem HTML Code im CSV-file erfolgt zeitgleich in einer Loop.

### B2

### B3

### C1 – c1\_web\_scrapper.py

Der B1 Web Scrapper ladet und erstellt aus einer Webseite eine Liste mit Ländernamen und das dazugehörige Kontinent.

import requests  
from bs4 import BeautifulSoup as bs  
  
*#download html file and convert into soup object*raw\_html = requests.get(**"https://www.bernhard-gaul.de/wissen/staatenerde.php#uebneu"**)  
soup\_html = bs(raw\_html.text, **"html.parser"**)  
  
*#load extracted text into csv file*csv\_file = open(**"c1\_country\_src.csv"**, **"w"**, encoding=**'utf-8'**)  
for p in soup\_html.select(**"tr"**):  
 y = p.select(**"td"**)  
 a = y[0].text  
 b = y[1].text  
 csv\_file.write(a + **","** + b + **"**\n**"**)  
 print(a + **","**)  
csv\_file.close()

### C2

## Transform

### A1

### B1 – b1\_cleaning.py

Folgenden Schritte werden für das Bereinigen der Daten unternommen:

1. Umlauten umschreiben
2. Korrigieren eines Spaltenfehlers durch ein Semicolon
3. Abfrage und Korrektur von Grossschreibung
4. Korrekturen von Datums

1.  
df = df.replace(**'ä'**, **'ae'**, regex=True)  
df = df.replace(**'ö'**, **'oe'**, regex=True)  
df = df.replace(**'ü'**, **'ue'**, regex=True)  
df = df.replace(**'Ä'**, **'Ae'**, regex=True)  
df = df.replace(**'Ö'**, **'Oe'**, regex=True)  
df = df.replace(**'Ü'**, **'Ue'**, regex=True)  
  
  
2.print(df[**'Land'**][14])  
df[**'Land'**][14] = **"USA"**df[**'Jahr'**][14] = **"1994"**df.count()  
  
*#second - search for duplicates*df.duplicated() *#no duplicates!*3.count = 0  
for p in df[**"Land"**]:  
 if df[**"Land"**][count][0].islower():  
 print(p)  
 df[**"Land"**][count] = (df[**"Land"**][count][0].upper() + df[**"Land"**][count][1:])  
 count += 1  
  
4.count = 0  
for p in df[**"Jahr"**]:  
 if len(df[**"Jahr"**][count]) > 4:  
 print(p)  
 df[**"Jahr"**][count] = df[**"Jahr"**][count][-4:]  
 count += 1

### B2

### B3

### C1

### C2

## Load

# Fragestellungen

# Reflexion

1. CSV-Export [↑](#footnote-ref-1)
2. Tabelle unter Quickstart Query «RGDPNA (1950-2019)» [↑](#footnote-ref-2)