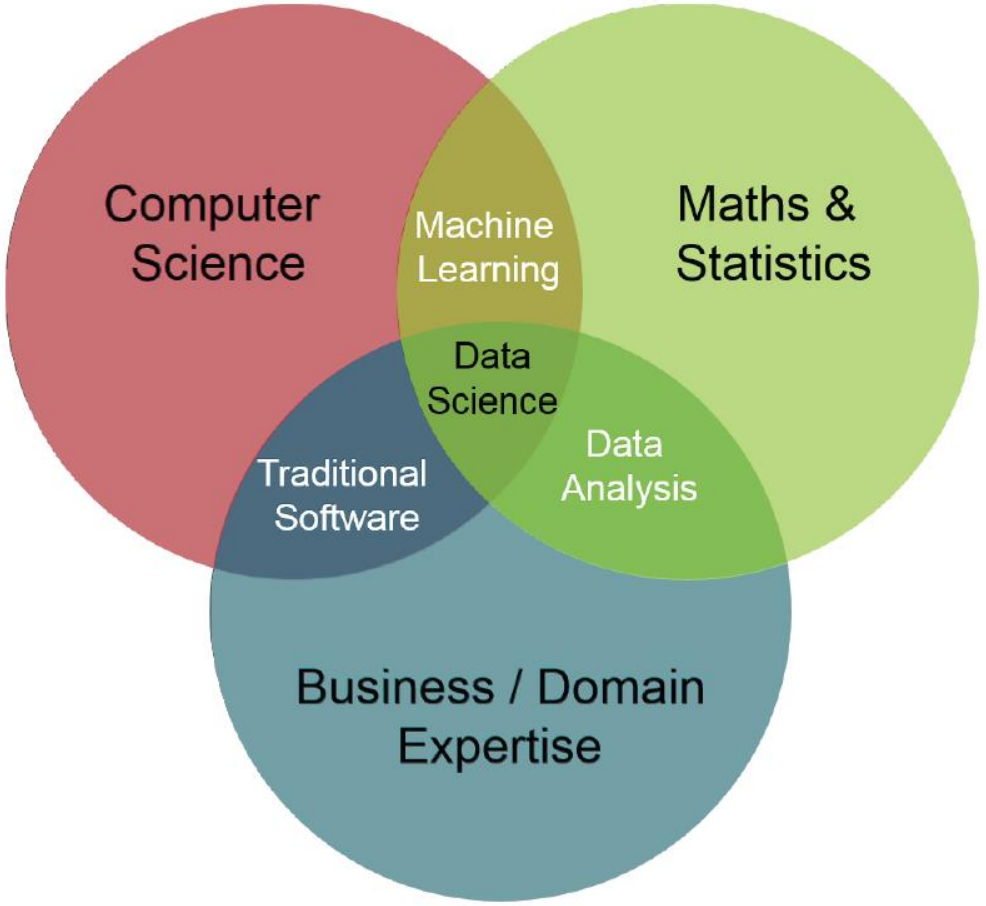
Recap

Programmierkurs 2 Data Science WS23/24

Leonard Traeger
M. Sc. Information Systems
leonard.traeger@fh-dortmund.de

What is Data Science? (Recap)



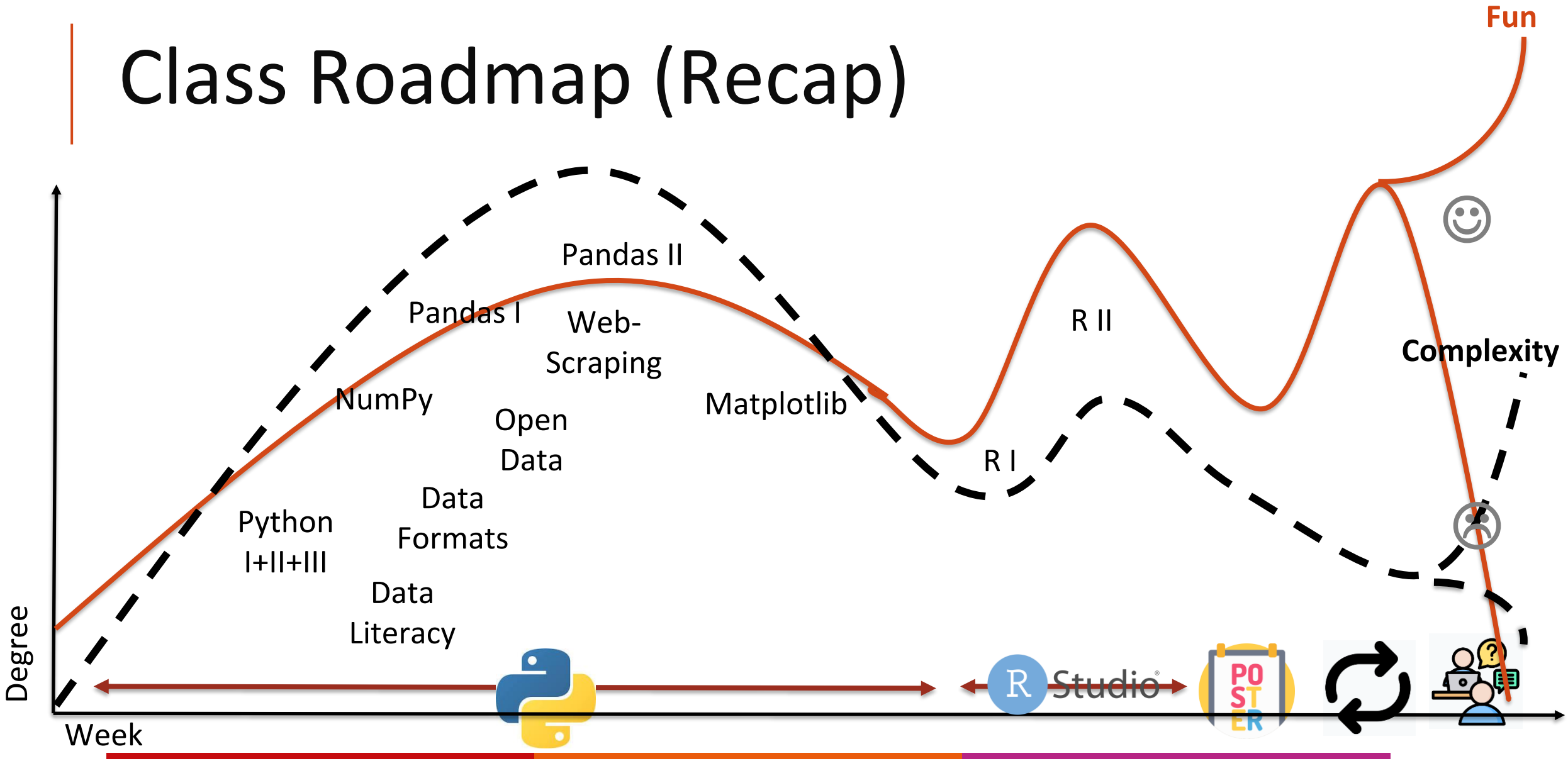
Data Science (Recap)

"Data Science beschäftigt sich mit einer **zweckorientierten Datenanalyse** und der **systematischen Generierung** von **Entscheidungshilfen** und -grundlagen, um **Wettbewerbsvorteile** erzielen zu können."

"In der Wissenschaft beschäftigt sich Data Science mit **unterschiedlichen Bereichen** und kann daher verschiedene akademische Hintergründe haben: *Informatik, Statistik, Mathematik, Natur- oder Wirtschaftswissenschaften, Machine Learnings, des statistischen Lernens, der Programmierung, der Datentechnik, der Mustererkennung, der Prognostik, der Modellierung von Unsicherheiten und der Datenlagerung*."

https://gi.de/themen/beitrag/data-literacy-und-data-science-education-digitale-kompetenzen-in-der-hochschulausbildung

Class Roadmap (Recap)



Overall Learning Goals (Recap)

By the end of this course you will be able to:

- **Discuss** Data Science and its current trends.
- **Explain** the fundamentals of typical data science applications.
- For a variety of data science life cycle frameworks, be able to **explain, compare** and **contrast**, and **discuss** ethics, limitations, and applicability.
- **Apply** Data Science techniques in **Python** to solve real problems.

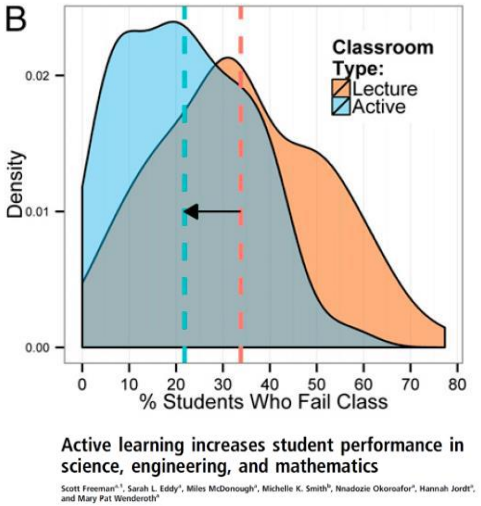
Week 17: Oral Exams (Recap)

- About lecture contents and project.
- Questions about both conceptual and coding problems.
- Imagine you are the expert providing consultancy to a potential customer ☺

Hierarchy of relevancy:

1. Slides including Training / Think-Pair-Share.
2. Your project documents.
3. Lab work.
4. Scripts and demos.
5. Books, articles, documentations
   (no readings are relevant if they are not covered in the slides).

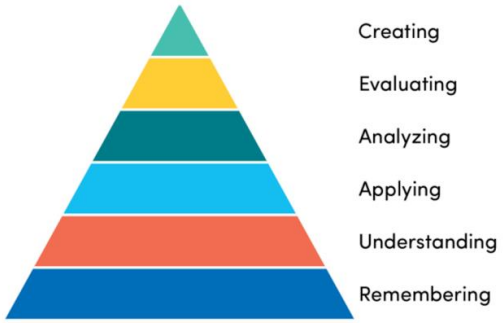
How to succeed in “Programmierkurs 2 Data Science“?



Active learning increases student performance in science, engineering, and mathematics

Scott Freeman[a,1], Sarah L. Eddy[a], Miles McDonough[a], Michelle K. Smith[b], Nnadozie Okoroafor[a], Hannah Jordt[a], and Mary Pat Wenderoth[a]

1. Follow each week‘s **learning goals** (in the beginning of the slides).



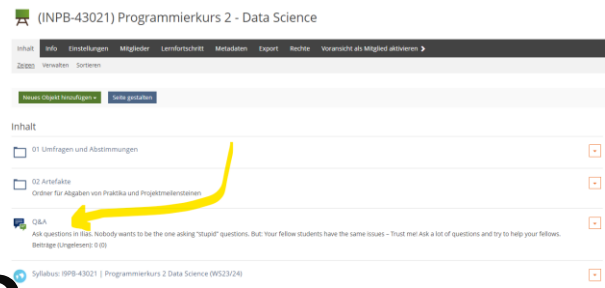
2. Participate in **Training** and **Think-Pair-Share**.

How to succeed in “Programmierkurs 2 Data Science“?

3. Lab Work “Praktikum”:

- Manifestation of conceptual and programming knowledge about frameworks and libraries.
- **Optional**.
- Split into two sections:
  - Lab I: Python I+II+III and NumPy
  - Lab II: Pandas and matplotlib
- Submit individually or (preferably) in **pairs of two** through Ilias.
- To be completed over three weeks.
- Each section contributes up to 8% (total 16%) of additional percentage points towards the final grade.

How to succeed in “Programmierkurs 2 Data Science“?

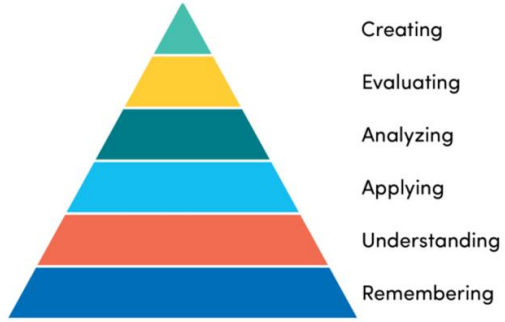


4. Ask questions in Ilias:

- Nobody wants to be the one asking “stupid” questions.
- But: Your fellow students have the same issues – Trust me!
- Ask a lot of questions and try to **help your fellows**.

A single question or comment related to conceptual frameworks, coding problems, team project, exam preparation, or anything (in your opinion) useful for the class contributes to **additional 0.66%** towards your final grade.

Oral Exam (15-20 Minutes)



40% Conceptual Knowledge	40% Coding	20% Data Science Project Life Cycle
6-8 minutes with **3 to 5** questions: • Name and describe relevant concepts and put into context	6-8 minutes with **Two** (simple&difficult) coding snippets: 1. Describe technicalities 2. Contextualize briefly and describe potential output 3. Categorize procedure into a data preprocessing phase	3-4 minutes with **1 to 2** questions: • Relate any question to a DS-Project Life Cycle
	Python I+II+III NumPy **Pandas I+II;** R I+II Praktikum I+II	Data Formats Open Tidy Data Data Literacy + Ethics Python II (first part) Projects

Oral Exam: bring Cheat-Sheets

Two A4 cheat-sheets with both sides printed.

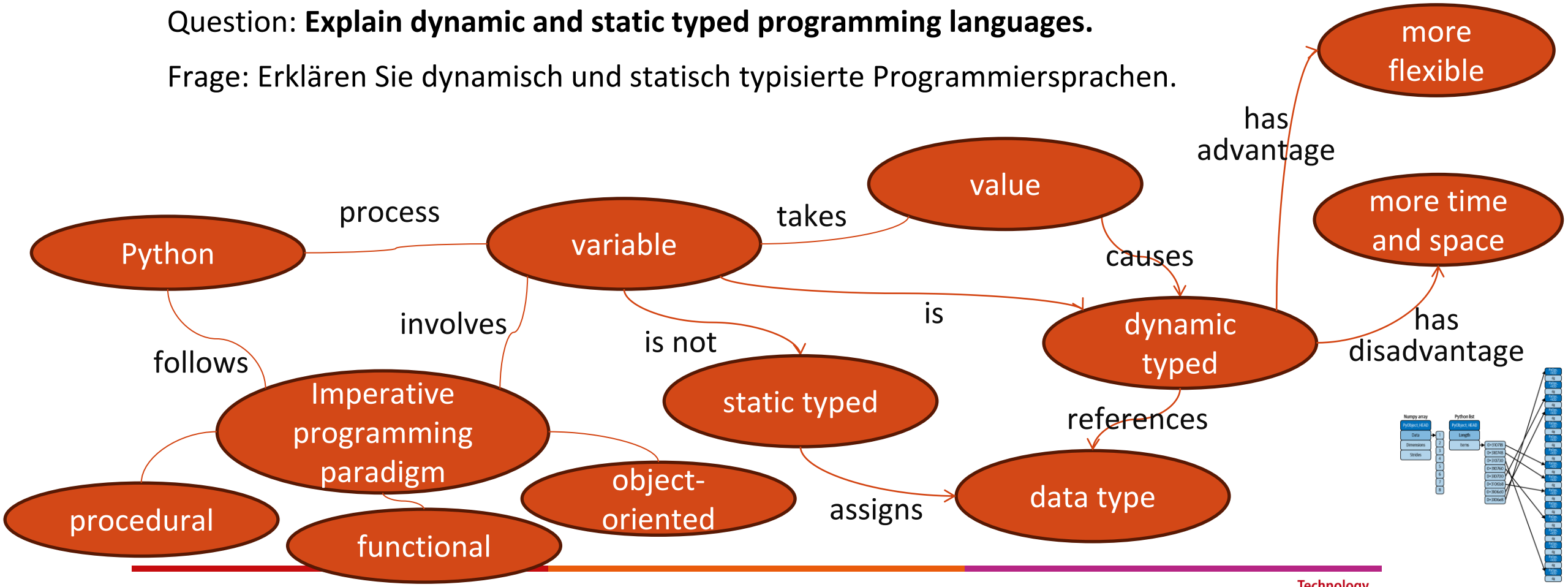
Can be anything you want ☺

- Pandas cheat-sheet: https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf
- R cheat-sheet: https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf
- *Personal Recommendation: Create own Mind-Map with* ***nodes, edges, and relationship.***

Example 1: Conceptual Knowledge

Question: **Explain dynamic and static typed programming languages.**

Frage: Erklären Sie dynamisch und statisch typisierte Programmiersprachen.



Variable and Datatypes (Recap)

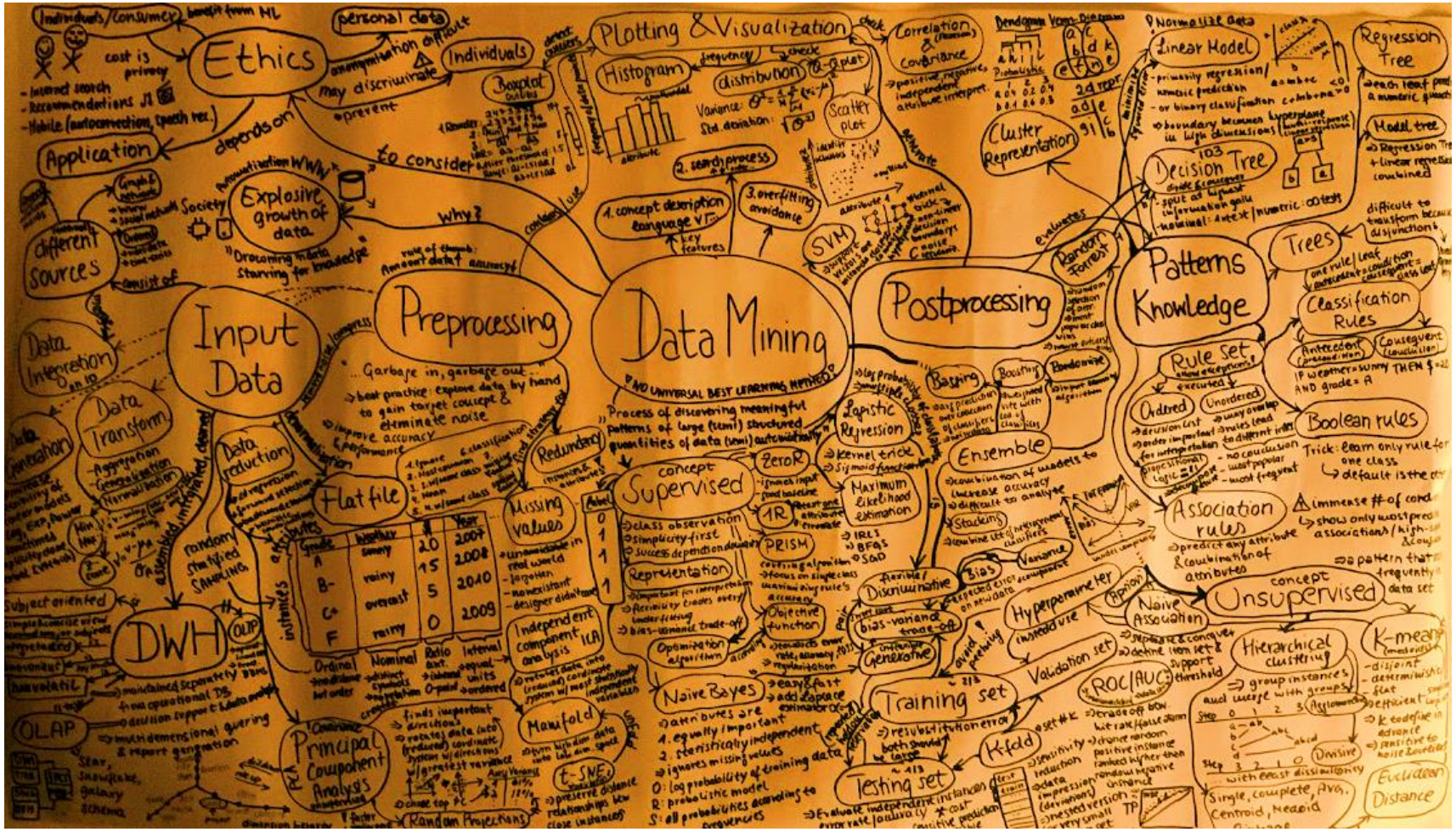
Assignment uses **dynamic referencing**.

- The **type/class** is determined from the **value**, not declared.
- **Type/class information** belongs to the **data**, not the name bound to that data.

```
x = 10000
```

- x is not just a “raw” integer.
- x is a pointer to a compound C structure, which contains several values.
- **Dynamic referencing** in Python is **more flexible** but also **more time** and **space** consuming than compared to raw C.

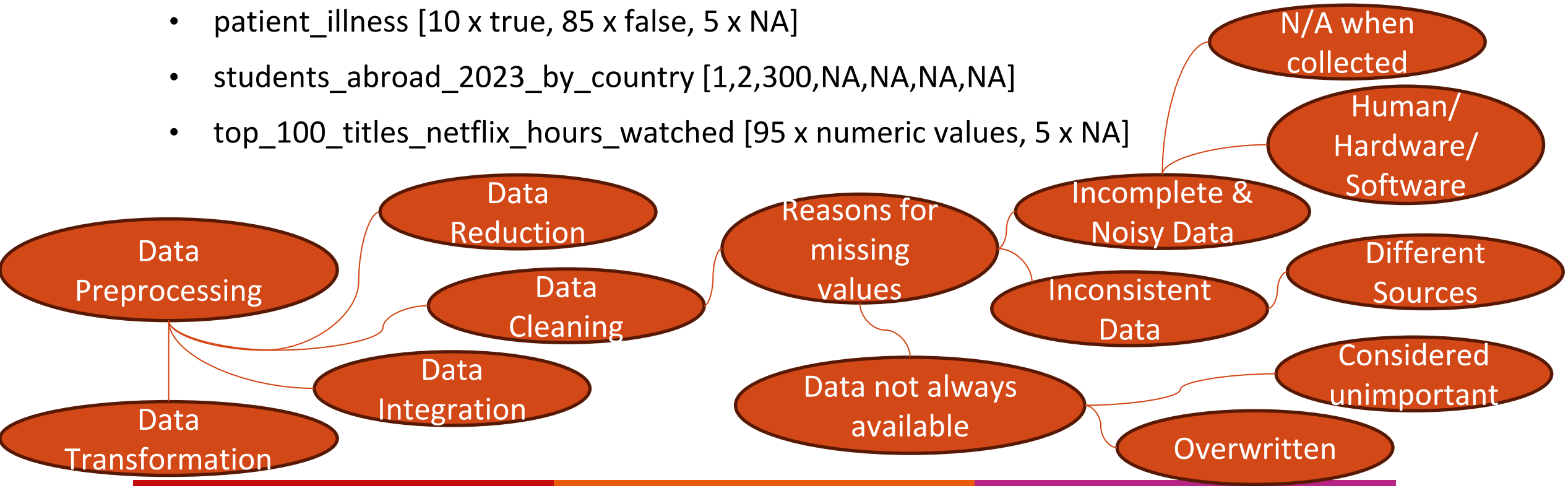
Example Mind-Map



Example 2: Conceptual Knowledge

Frage: Nennen Sie Gründe für fehlende Daten (NA) und schlagen Sie eine Strategie vor, um diese zu finden und sinnvoll zu ersetzen.

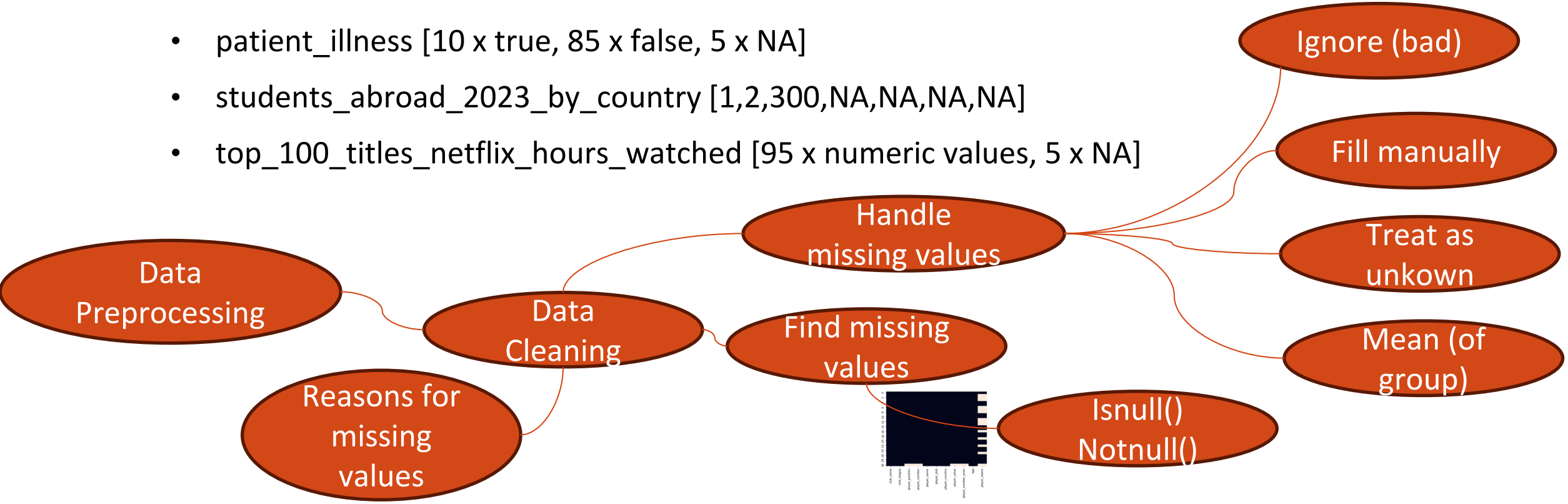
- patient_illness [10 x true, 85 x false, 5 x NA]
- students_abroad_2023_by_country [1,2,300,NA,NA,NA,NA]
- top_100_titles_netflix_hours_watched [95 x numeric values, 5 x NA]



Example 2: Conceptual Knowledge (cont.)

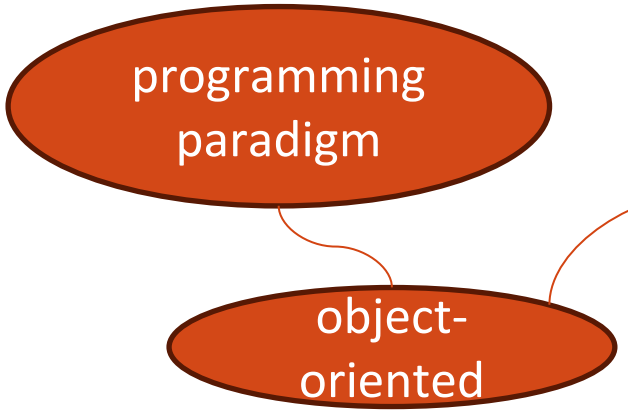
Frage: Nennen Sie Gründe für fehlende Daten (NA) und schlagen Sie eine Strategie vor, um diese zu finden und sinnvoll zu ersetzen

- patient_illness [10 x true, 85 x false, 5 x NA]
- students_abroad_2023_by_country [1,2,300,NA,NA,NA,NA]
- top_100_titles_netflix_hours_watched [95 x numeric values, 5 x NA]



Example 3: Conceptual Knowledge

Frage: Welche Eigenschaften und Unterschiede haben Python-Container (siehe List, Set, Dictionary,...)?



		Ordered	Changeable	Indexed	Duplicates
List	**[ ]**	Yes	Yes	Yes	Yes
Tuple	**( )**	Yes	No	Yes	Yes
Set	**{ }**	No	Yes	No	No
Dictionary	**{"_:_"}**	No	Yes	Yes	No

player_score[0,1,3,3]

tweet = {'#cat', '#cute', '#lol'}

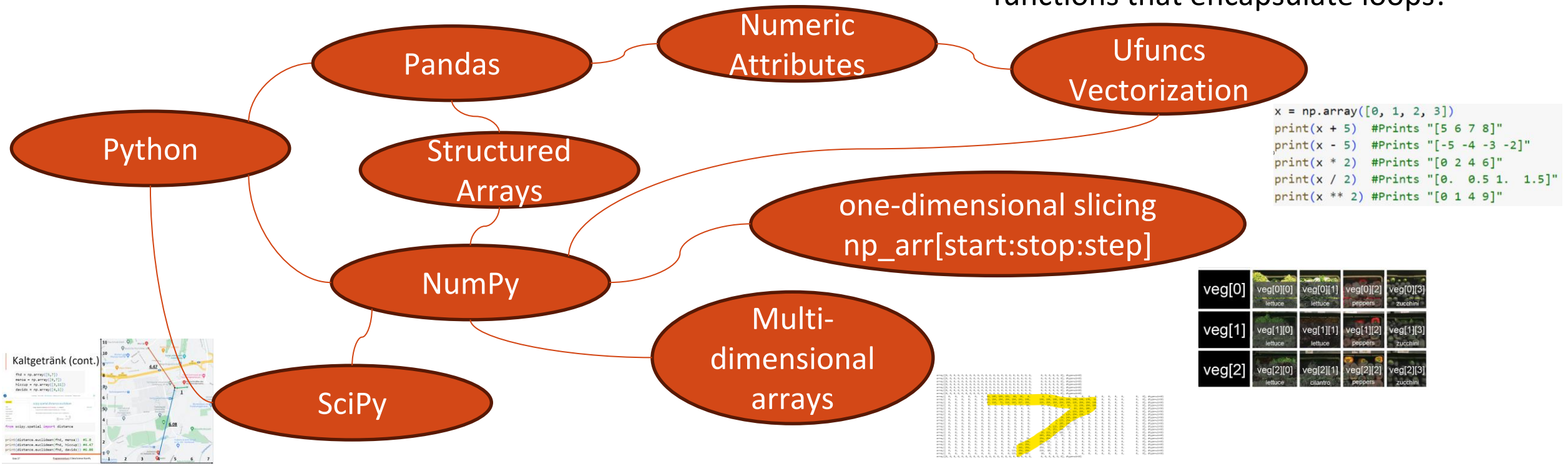
player_score(0,1,3,3)

eng_ger = {'I':'Ich', 'learn':'lernen'}

Example 4: Conceptual Knowledge

Frage: Was vereinfacht "Vectorization" in NumPy? Kann das Konzept auch auf numerische Attribute in Panda DataFrames übertragen werden?

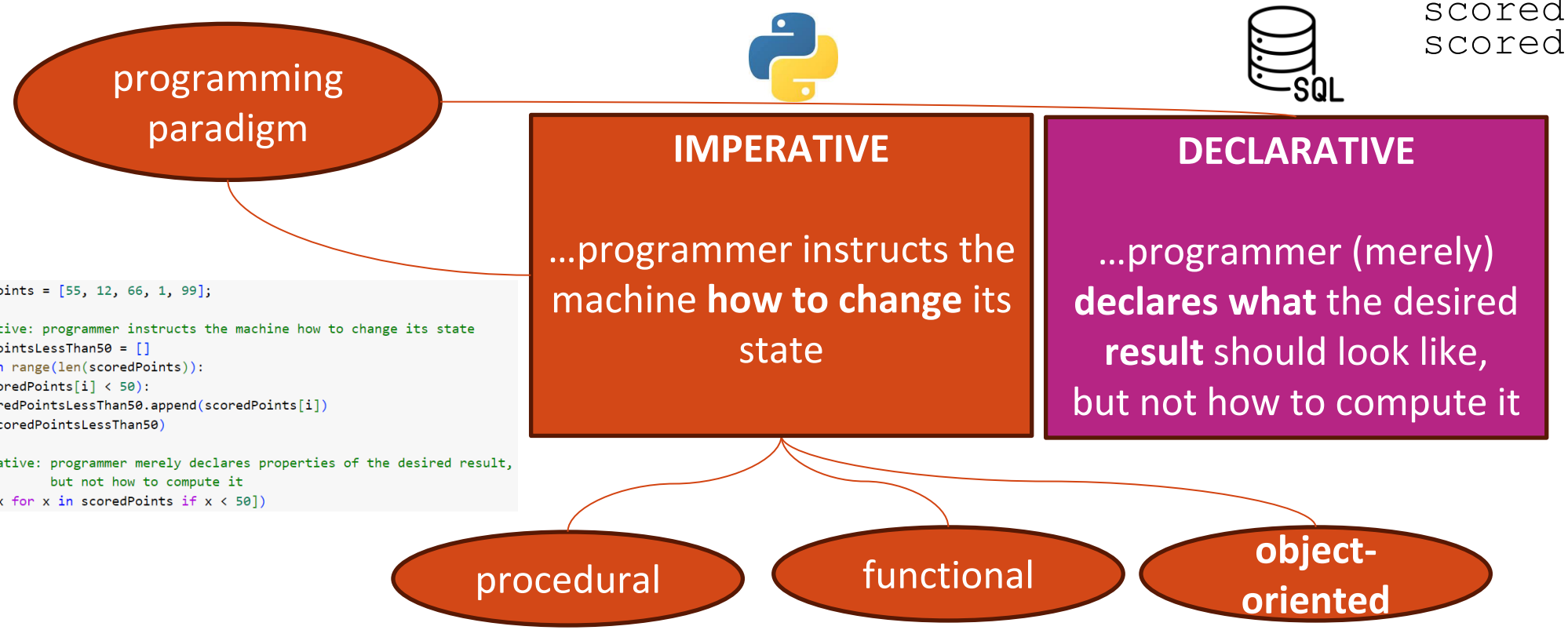
functions that encapsulate loops!



Example 5: Conceptual Knowledge

Frage: Ist Python eine imperative oder deklarative Sprache?

SQL: `SELECT * FROM scoredPoints WHERE scoredPoints < 50;`



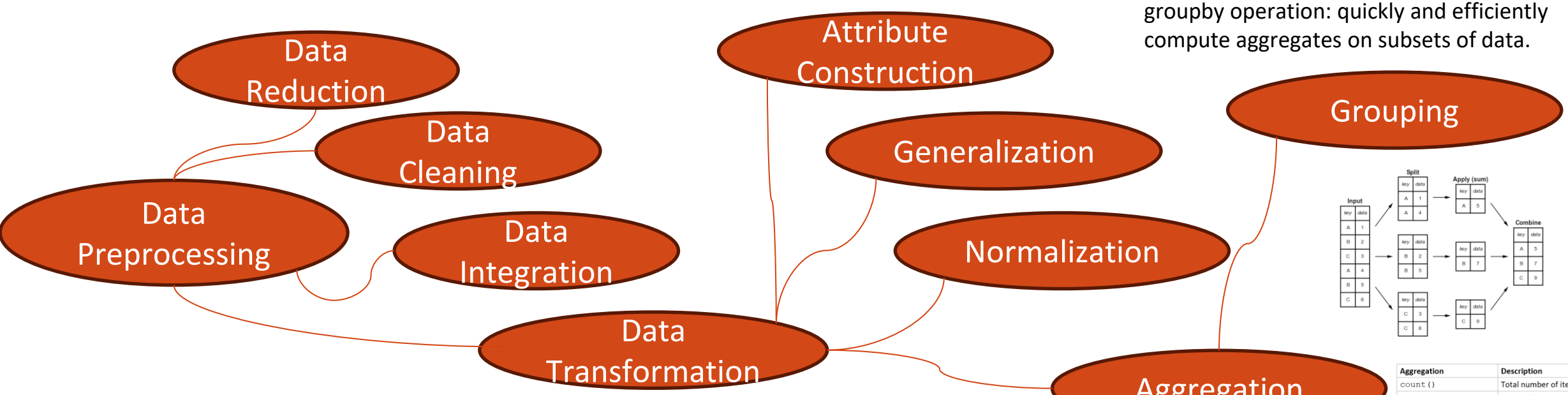

```python
scoredPoints = [55, 12, 66, 1, 99];

#imperative: programmer instructs the machine how to change its state
scoredPointsLessThan50 = []
for i in range(len(scoredPoints)):
  if(scoredPoints[i] < 50):
    scoredPointsLessThan50.append(scoredPoints[i])
print(scoredPointsLessThan50)

#declarative: programmer merely declares properties of the desired result,
#             but not how to compute it
print([x for x in scoredPoints if x < 50])
```

Example 6: Conceptual Knowledge

Frage: Welche Methode fasst das Konzept von **split-apply-combine** zusammen?



groupby operation: quickly and efficiently compute aggregates on subsets of data.

reduce the entire “array” to a single summarizing value!

Aggregation	Description
`count()`	Total number of items
`first(), last()`	First and last item
`mean(), median()`	Mean and median
`min(), max()`	Minimum and maximum
`std(), var()`	Standard deviation and variance
`mad()`	Mean absolute deviation
`prod()`	Product of all items
`sum()`	Sum of all items

Example 1: Coding

```
df_dsa[["Studienland", "Kontinent","Außenhandel_Saldo_Mrd_2015","Außenhandel_Umsatz_Mrd_2015"]].head()
```

	Studienland	Kontinent	Außenhandel_Saldo_Mrd_2015	Außenhandel_Umsatz_Mrd_2015
0	Östereich	Europa	20.9	95.5
1	Niederlande	Europa	-8.7	167.1
2	Vereinigtes Königsreich	Europa	50.6	127.4
3	Schweiz	Europa	7.0	91.2
4	Vereinigte Staaten	Nordamerika	53.5	173.9

```
def get_category_partner(saldo, umsatz):
  if((saldo > 0) & (umsatz > 50)):
    return "Profitabler Großpartner"
  elif((saldo < 0) & (umsatz > 50)):
    return "Defizitärer Großpartner"
  elif(saldo > 0):
    return "Profitabler Partner"
  elif(saldo < 0):
    return "Defizitärer Partner"
  else:
    return "Partner"
```

```
df_dsa[1:3].apply(lambda x:
                  get_category_partner(
                      x.Außenhandel_Saldo_Mrd_2015,
                      x.Außenhandel_Umsatz_Mrd_2015)
                  , axis=1)
```

1. Describe technicalities (in order what makes most sense):

- **DataFrame**
- Column indexing
- Row slicing via head()
- **Function** has two numerical inputs
- Function has four return cases and one default
- Function uniformly returns a string value
- **Apply** function onto sliced DataFrame
- Lambda resembles an unnamed (anonymous) function executed during run-time (necessary due to two parameter values)
- x resembles a row (axis=1) that passes two attribute values to the function (using the dot notation)

2. Contextualize and describe potential output:

- Is code executable without errors? **Yes**
- “A country receives a partnership category based on two numerical values on transit revenue”
- “Profitabler Großpartner”, “Defizitärer Großpartner”

3. Categorize to data preprocessing phase

- Data Transformation > Generalization / Attribute Construction

Example 2: Coding

```
import names
```

```
names = [names.get_full_name() for i in range(3)]
print(names)
```

```
['Irene Perkins', 'Shelley Upton', 'Kevin Moore']
```

```
def has_M(name):
  return "M" in name
```

```
def get_abbreviation(names):
  split_names = names.split(" ");
  return split_names[0][0] + "." + split_names[1][0] + "."
```

```
names.append('MarcoReus')
list(filter(has_M, map(get_abbreviation, names)))
```

1. Describe technicalities (in order what makes most sense):

- **List creation** via List Comprehension
- Iterate three times calling the names-library object with the method get_full_name()
- **First function** returns true if the passing value has an uppercase M in the string
- **Second function** split the inserted string into two strings delimited with a space character, then returns the first letter with a dot of each
- **Append** an additional item to the end of the list
- **Map** get_abbreviation function to names list
- **Filter** name abbreviations list that contain upper-case M

2. Contextualize and describe potential output:

- Is code executable without errors? **No; why?**
- We want to generate a list of abbreviations of names which contain an upper-case M
- “K.M.”, “M.R”

3. Categorize to data preprocessing phase

- None or
- Data Transformation > Attribute Construction

Example 3: Coding

```
df_top_100_clubs[["club_name","club_stadium_seats", "club_league"]].head()
```

	club_name	club_stadium_seats	club_league
0	Manchester City	55.017 Plätze	Premier League
1	FC Arsenal	60.704 Plätze	Premier League
2	FC Paris Saint-Germain	49.691 Plätze	Ligue 1
3	Real Madrid	81.044 Plätze	LaLiga
4	FC Chelsea	40.853 Plätze	Premier League

```
df_top_100_clubs["club_stadium_seats"] = df_top_100_clubs.club_stadium_seats.apply(
    lambda x: int(float(x.split(" ")[0])*1000))
```

```
df_top_100_clubs.groupby("club_league").club_stadium_seats.sum().sort_values(ascending=False)
```

1. Describe technicalities (in order what makes most sense):

- **DataFrame**
- Column indexing
- Row slicing via head()
- **Apply** function onto sliced DataFrame Series
- Lambda resembles an unnamed (anonymous) function executed during run-time
- x resembles an element of Series that is split with space element; the numeric string part is casted as float; multiplied times 1000; casted as int
- **Overwrite** old Series with new numeric Series
- **Group** DataFrame by clubs and sum seats per group using “split-apply-combine”; sort output descendingly

2. Contextualize and describe potential output:

- Is code executable without errors? **Yes**
- Premier League ~ 150K
  LaLiga ~80K
  Ligue 1 ~ 50K

3. Categorize to data preprocessing phase

- Data Cleaning > Smooth Noisy Data
- Data Transformation > Aggregation

Further Examples: Coding

Example 4:

```
df_bvb_player.apply(lambda x: "Young" if x.age > 21 else "Old", axis=1)
```

Example 5:

```
df_bvb_player["player_name","player_age"].player_position.map({'Torwart': 1, 'Abwehr':
   2, 'Mittelfeld': 3, 'Sturm':4, np.NaN:0})
```

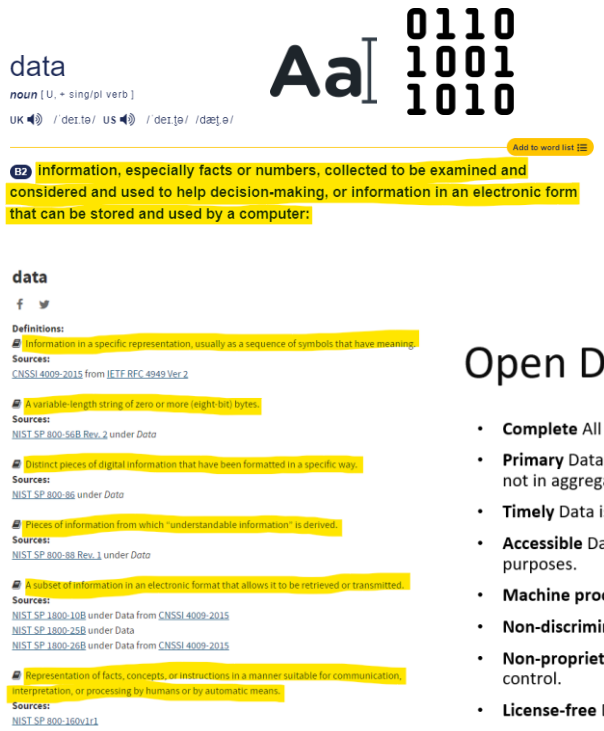
Example 6:

```
b = np.array([[1,2,3],[4,5,6]]); b[1,-3] = 42
```

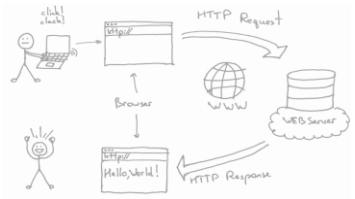
Example 1: Data Science Project Life Cycle

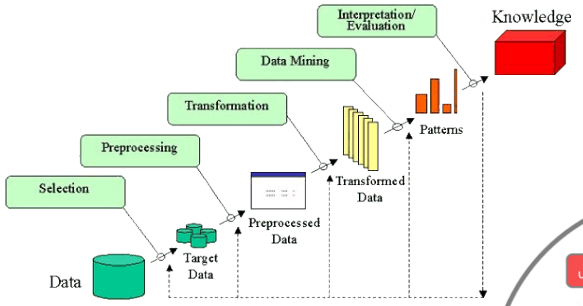
Question: **What is Open Data and at what phase would you use it in a DS Project?**

Frage: Was ist Open Data und in welcher Phase würden Sie es in einem DS Projekt verwenden?



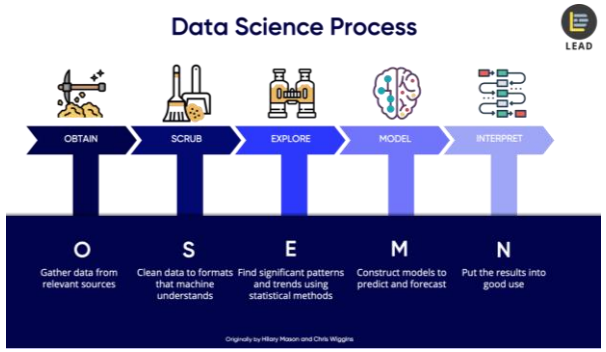
1. **CSV:** strings separated by commas and newlines.
   → Simple and most common in modelling relational data.
2. **JSON**: uses javascript syntax.
   → Tree-based and most common in data exchange.
3. **XML**: is a mark-up language (meta) and builds XHTML (language of the web).
   → Tree-based and most common in (web) structured documents.





Open Data Principles

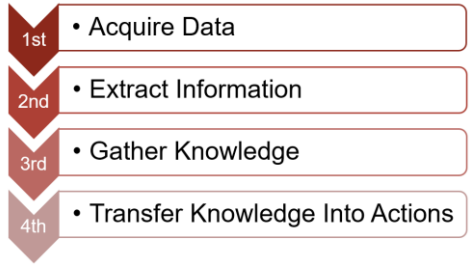
- **Complete** All public data is made available.
- **Primary** Data is as collected at the source, with the highest possible level of granularity, not in aggregate/modified forms.
- **Timely** Data is made available as quickly as necessary to preserve the value of the data.
- **Accessible** Data is available to the widest range of users for the widest range of purposes.
- **Machine processable** Data is reasonably structured to allow automated processing.
- **Non-discriminatory** Data is available to anyone, with no requirement of registration.
- **Non-proprietary** Data is available in a format over which no entity has exclusive control.
- **License-free** Data is not subject to any copyright, patent, ...

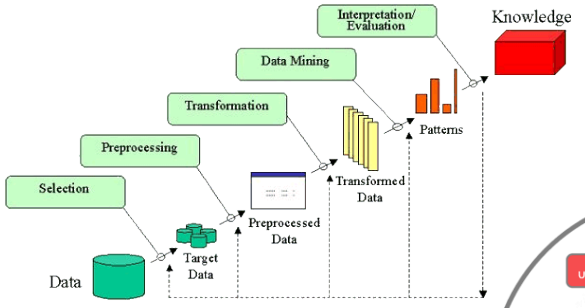


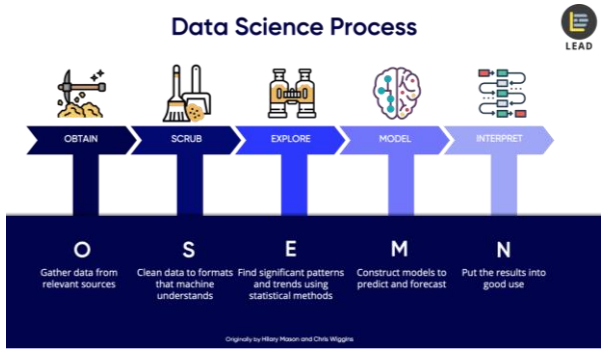
Example 2: Data Science Project Life Cycle

Question: **Would it make sense to switch to a different DS Project Life Cycle in a running project?**

Frage: Würde es im laufenden Projekt Sinn machen zu einem andere DS Project Life Cycle zu wechseln?







- Iterative set of data science components to plan and deliver a project.
- Differences in
  - Non-linearity versus linearity.
  - Several smaller steps versus larger comprehensive phases.
  - Data-centric versus business-understanding.

Every data science project and team are different;
you must adapt your own version.

Example 3: Data Science Project Life Cycle

Question: **What is Data Literacy and how can it be applied to DS Project planning?**

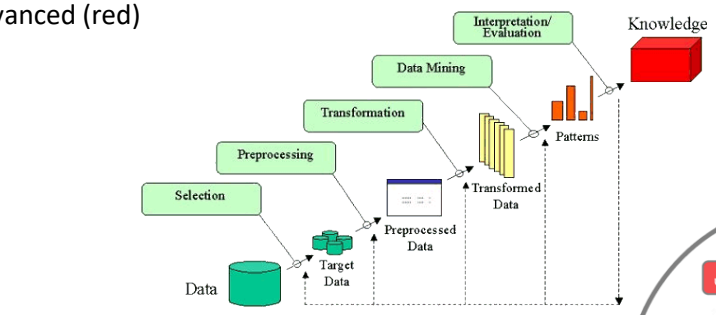
Frage: Was ist Data Literacy und wie kann das Konzept auf die Planung von DS-Projekten angewendet werden?

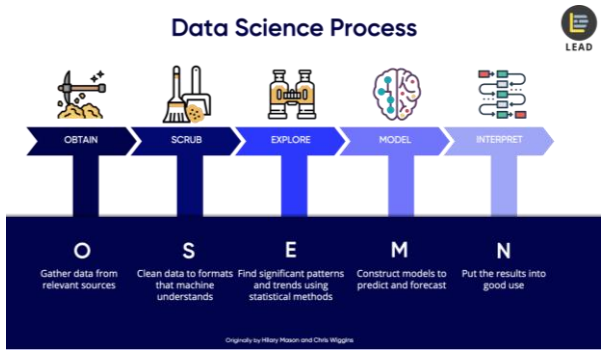
1. What do I want to do with data?
   - Data and its analysis are not an end in themselves (Selbstzweck).
   - *Target a concrete use case or application.*
2. What can I do with data?
   - The technical and methodological possibilities play a crucial role.
   - *Become aware of your capabilities.*
3. What am I allowed to do with data?
   - Legal regulations governing the use of data.
   - *Consider what you are allowed or at least not allowed to do.*
4. What should I do with data?
   - Data is a valuable resource which can create, beyond legally permitted actions, something good for society.
   - *Consider your personal and societal benefit of your application.*

Skill categorization: conceptional (blue), core (green), advanced (red)

Conceptual Framework	Introduction to Data
Data Collection	Data Discovery and Collection
	Evaluating and Ensuring Quality of Data and Sources
Data Management	Data Organization
	Data Manipulation
	Data Conversion
	Metadata Creation and Use
	Data Curation, Security and Re-Use
	Data Preservation
Data Evaluation	Data Tools
	Basic Data Analytics
	Data Interpretation (Understanding Data)
	Identifying Problems Using
	Data Visualization
	Presenting Data (Verbally)
	Data Driven Decisions Making (DDDM)
Data Application	Critical Thinking
	Data Culture
	Data Ethics
	Data Citation
	Data Sharing
	Evaluating Decisions based on Data

Abbildung 1: Data-Literacy-Kompetenzen nach Ridsdale et al.





Example 4: Data Science Project Life Cycle

Question: **What can you notice during an Initial Exploration of your dataset? Give examples.**

Frage: Was können Sie bei einer initialen Analyse Ihres Datensatzes entdecken? Nennen Sie Beispiele.

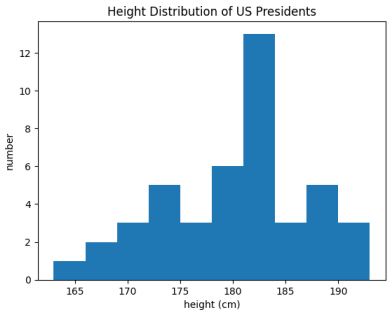
Take the **time** to **open** up your data **file** and have a lool

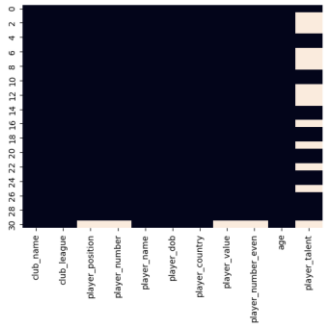
You might **be surprised** at what you find!

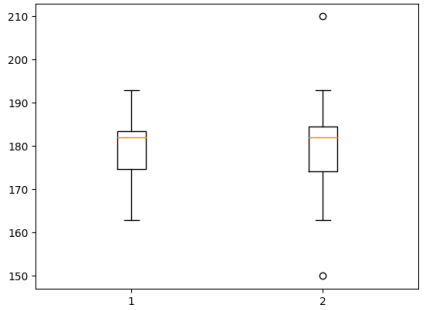
You may **notice obious issues** with the data, e.g.:

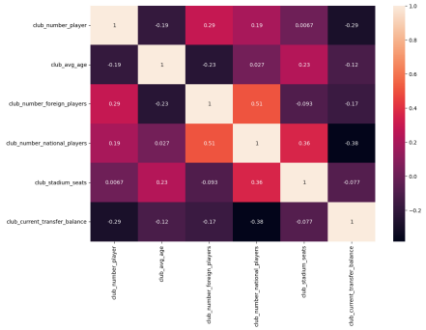
- Duplicate records
- Duplicate attributes
- Nonsensical values
- Useless attributes
- Incomplete data formatting during I/O ☺

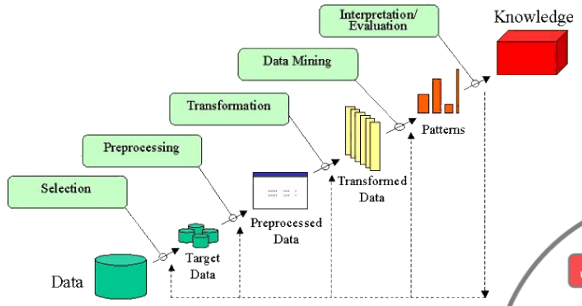
Too much data to inspect manually? Take a sample!

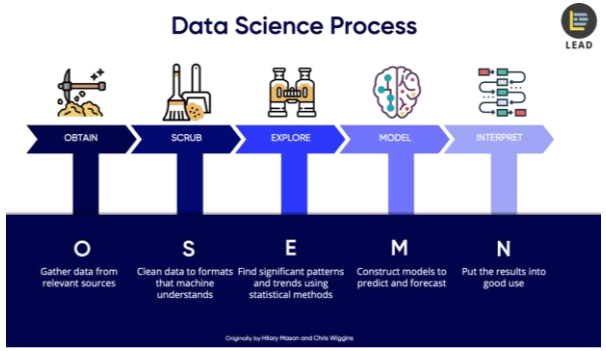








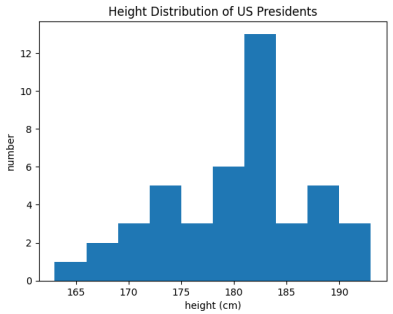


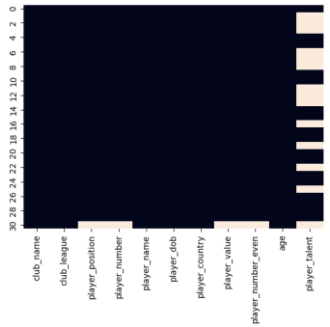


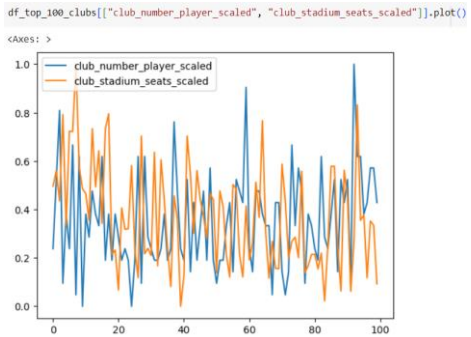
Example 5: Data Science Project Life Cycle

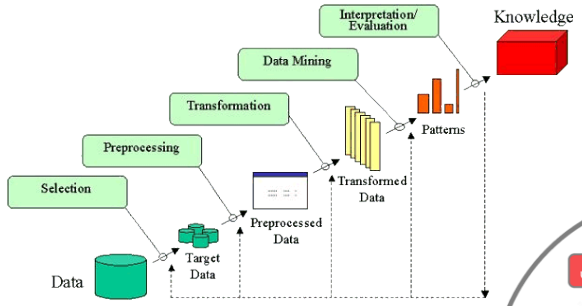
Question: **At which DS Project-Life-Cycle can you generate visualizations? Give examples.**

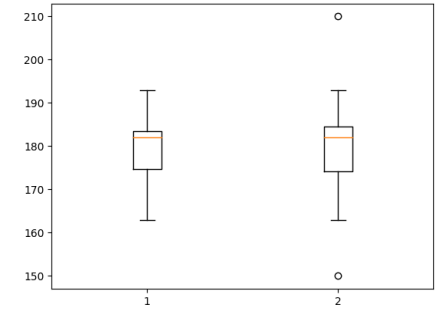
Frage: Zu welchen DS Projektphasen können Visualisierungen generiert werden? Nennen Sie Beispiele.

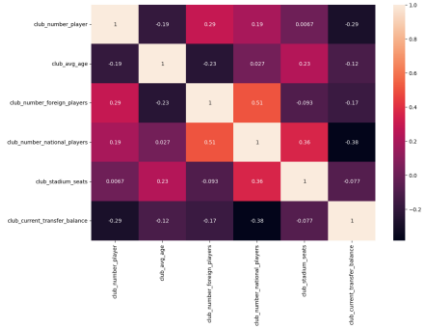


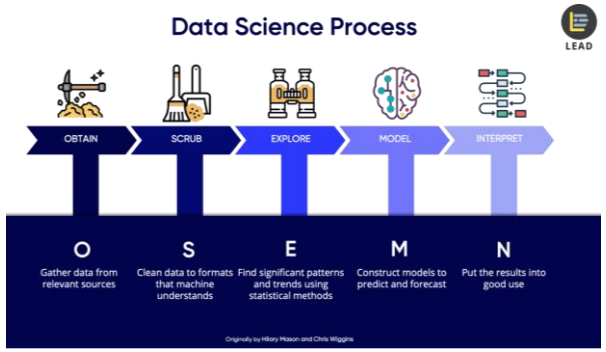












Logistics Oral Exam

- WebEx or in person still **to-be-defined** as I am looking for a second tutor...
- Grades for Praktikum II before 07.02.24
- Grades for projects most likely afterwards – sorry!

I will keep you posted ☺