Beispielprojekt Covid - Management of Scientific Data

Ein Vortrag von Dominic Wild

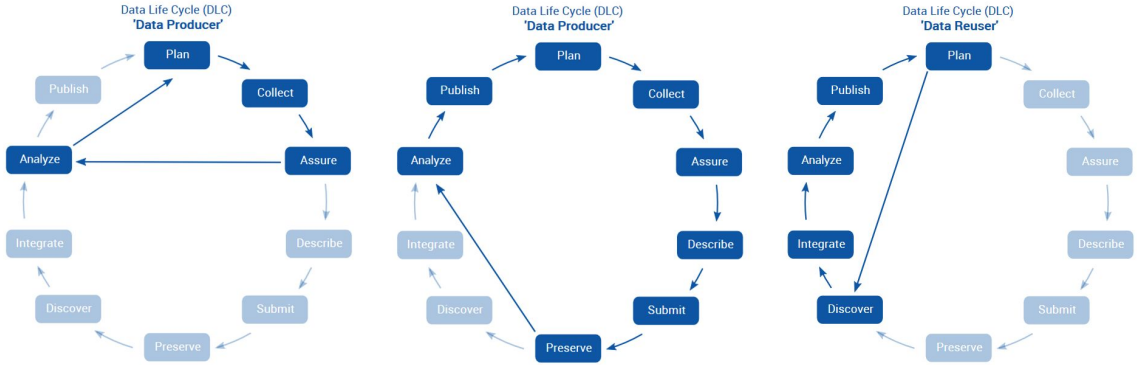
FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA
Fakultät für Mathematik
und Informatik

Forschungsfrage

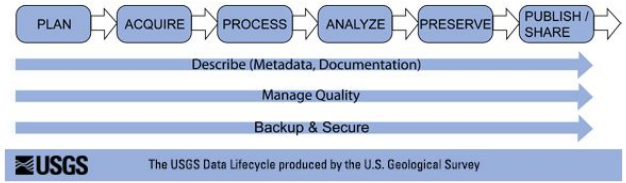
"Gibt es eine Korrelation zwischen der Positivitäts-Rate und der Anzahl abgeschlossener Tests in Deutschland zur Corona-Zeit?"

- dazu Datensatz: Archive of historical data on the testing volume for COVID-19 vom European Centre for Disease Prevention and Control
- Positivitäts-Rate = 100 x Anzahl neuer positiver Fälle/Anzahl abgeschlossener Tests pro Woche
  - zeigt besser, wie viele Leute wirklich krank sind
- In einer perfekten Welt:
  - es gibt genug Tests und Menschen testen sich dann, wenn sie sich krank fühlen
  - -> eine Korrelation sollte vorhanden sein
- Frage: War das in der Realität auch so?

Data Lifecycle



German Federation for Biological Data



United States Geological Survey

Data Lifecycle (meine Version)

Plan

- Forschungsfrage -> Anforderungen an Daten
- Data Management Plan
  - basierend auf Horizon Europe Template (auch empfohlen durch Uni Jena)
  - klärt viele Fragen bezüglich des Lifecycles
    - Datenbeschreibung
    - FAIR sicherstellen
    - Kosten
    - Datensicherheit und ethische Fragen
  - hilft dabei, nichts zu vergessen
  - besonders sinnvoll in großen Projekten
  - "living document"
    - manche Fragen können erst im Lauf des Projektes geklärt werden

Discover

- verschiedene Quellen bereits gegeben
- Datensatz vom "European Centre for Disease Prevention and Control" gefunden
- Beim Suchen verschiedene Dinge überprüfen:
  - Inhalt
  - Qualität (erster Eindruck)
  - Größe des Datasets
  - Provenance
  - sensitive Daten?
  - Verfügbarkeit / Lizenz
    - ECDC soll zitiert werden

Assure

- Qualitätsmerkmale überprüfen
  - Completeness: 100%, keine Woche fehlt, keine NaN-Einträge
  - Uniqueness: 100%, keine Wochen mehrfach im Dataset
  - Timeliness: historische Daten, kurz nach Ende des Erfassungszeitraums veröffentlicht
  - Validity: alle Datenpunkte haben korrekten Typ nach Import als Dataframe
  - Accuracy: schwer zu sagen, einfache plots sehen sinnvoll aus, Quelle ist verlässlich
  - Consistency: nur ein Datensatz genutzt -> dieser ist konsistent
- einige Preprocessing Schritte bereits benötigt zur Überprüfung

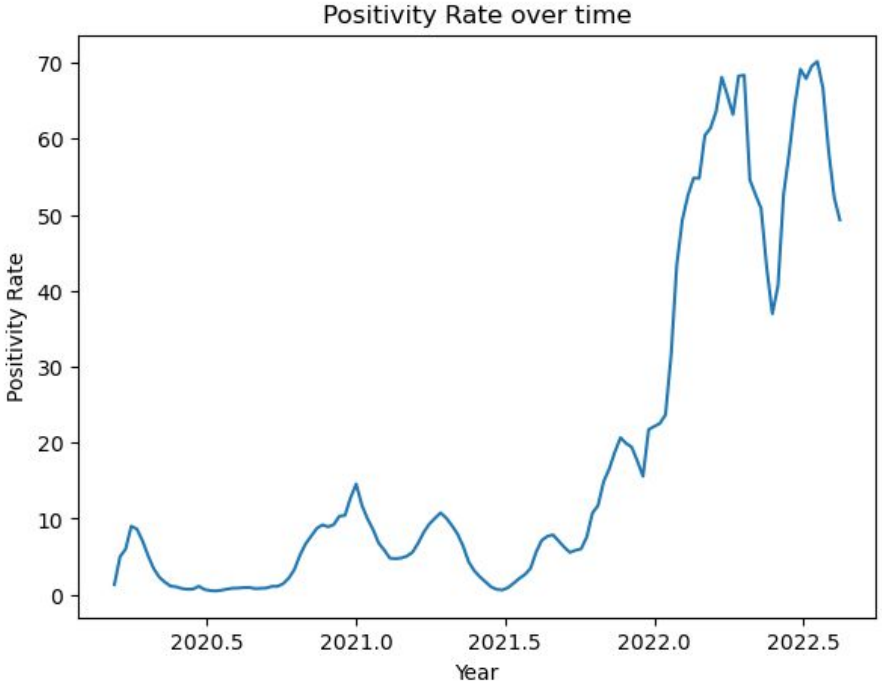
Process

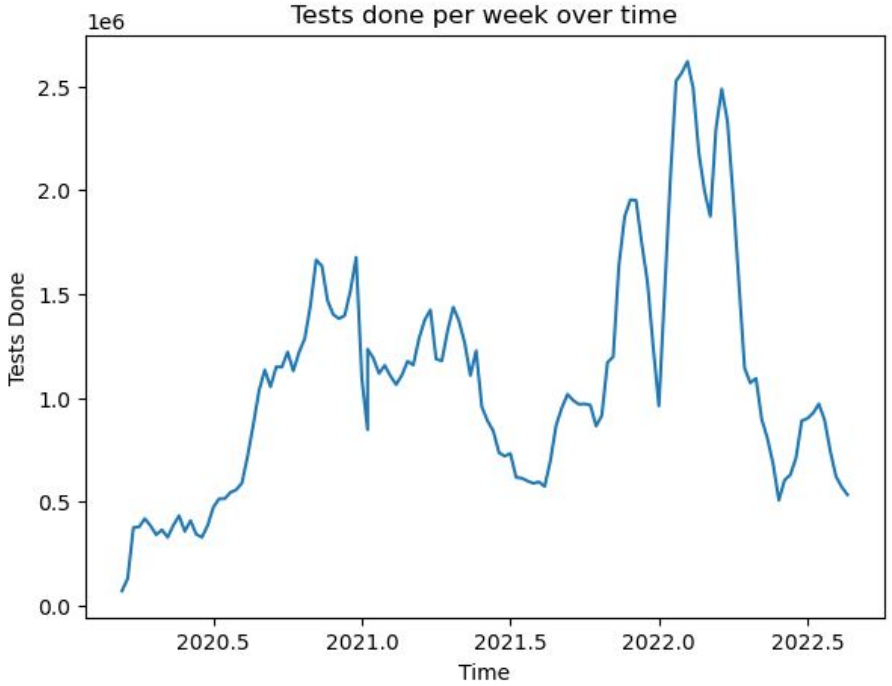
- Filtern der Daten in Deutschland auf level "national"
- Herausfiltern von unnötigen Spalten
- Herausfiltern von NaN-values
- Spalte year_week: string("2020-W35")
  - unpraktisch für statistische Analyse
  - daher Aufsplitten in 3 Spalten:
    - year
    - week
    - time
  - Spalte time = year + week/53
    - ermöglicht vergleichbare Daten (z.B. 2020.04 < 2020.36)
- Erstellen einer neuen Datei mit vorverarbeiteten Daten

Describe

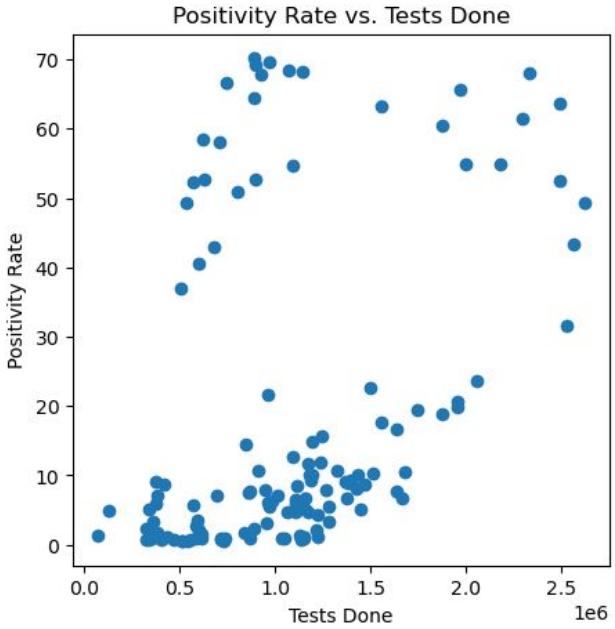
- Beschreibung der Verarbeitungsschritte
  - festgehalten in jupyter notebook
    - markdown + code
  - somit sogar reproduzierbar
  - dafür nötig: Abhängigkeiten beschreiben
    - Programmiersprache, Packages + Versionen, Installationsanleitung
- Beschreiben der Daten -> Metadaten
  - festgehalten in markdown-file und xml file
  - somit human- und machine-readable
  - Inhalt:
    - Kurzbeschreibung, Autor, Spalten, Zeitraum, Typ, Veröffentlichungsdatum, Identifier, Contributor, Nutzungsrechte, …

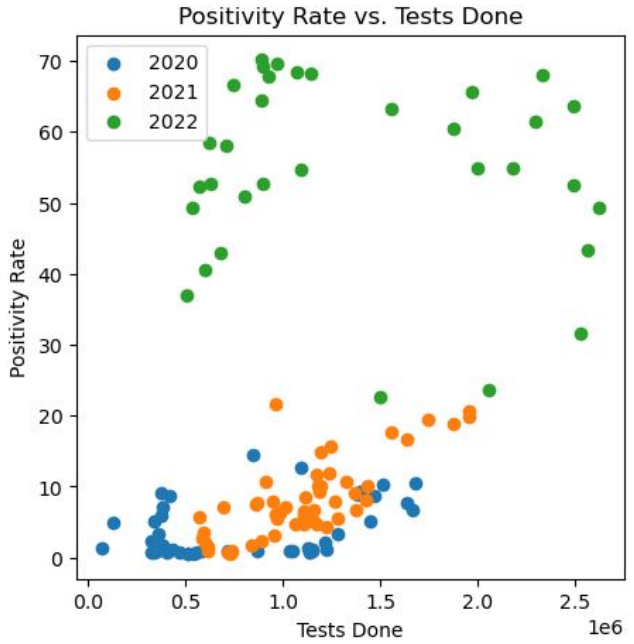
Analyze


Positivity Rate over time
Positivity Rate
0
10
20
30
40
50
60
70
2020.5
2021.0
2021.5
2022.0
2022.5
Year

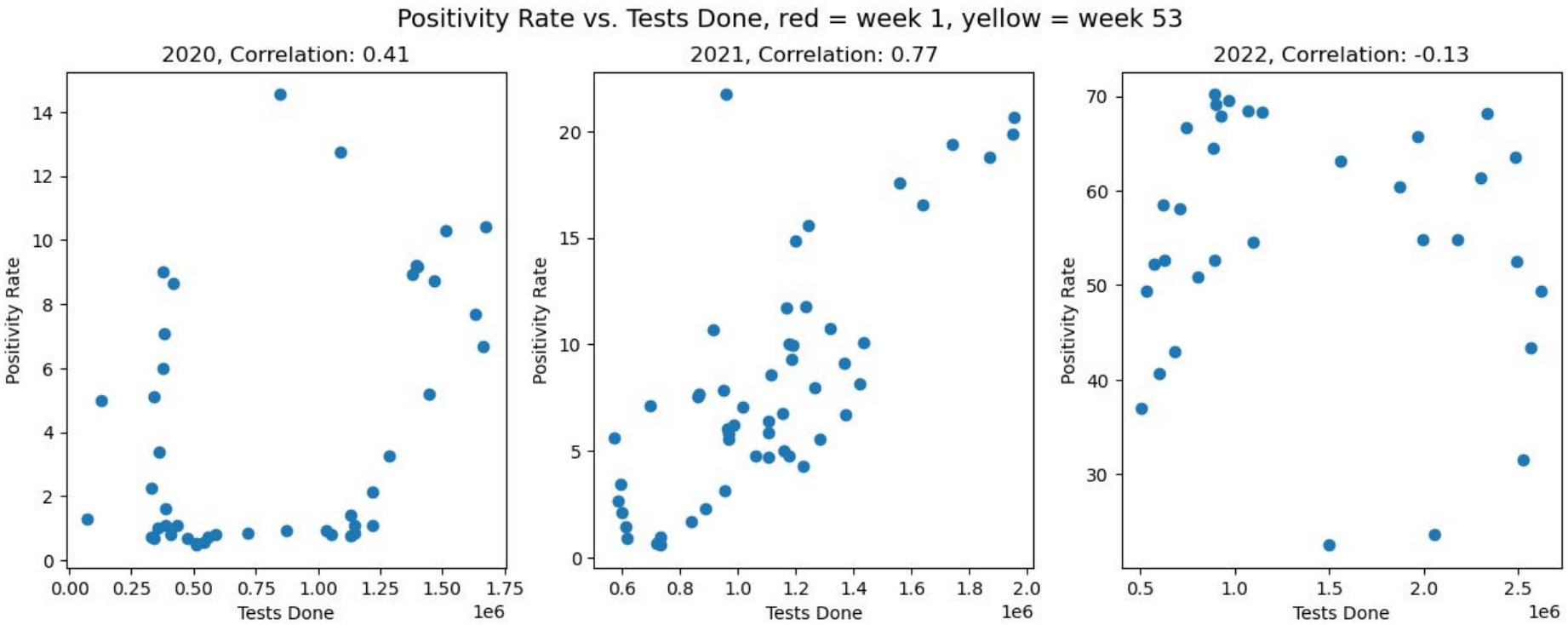

Tests done per week over time
1e6
Tests Done
0.0
0.5
1.0
1.5
2.0
2.5
2020.5
2021.0
2021.5
2022.0
2022.5
Time

Analyze

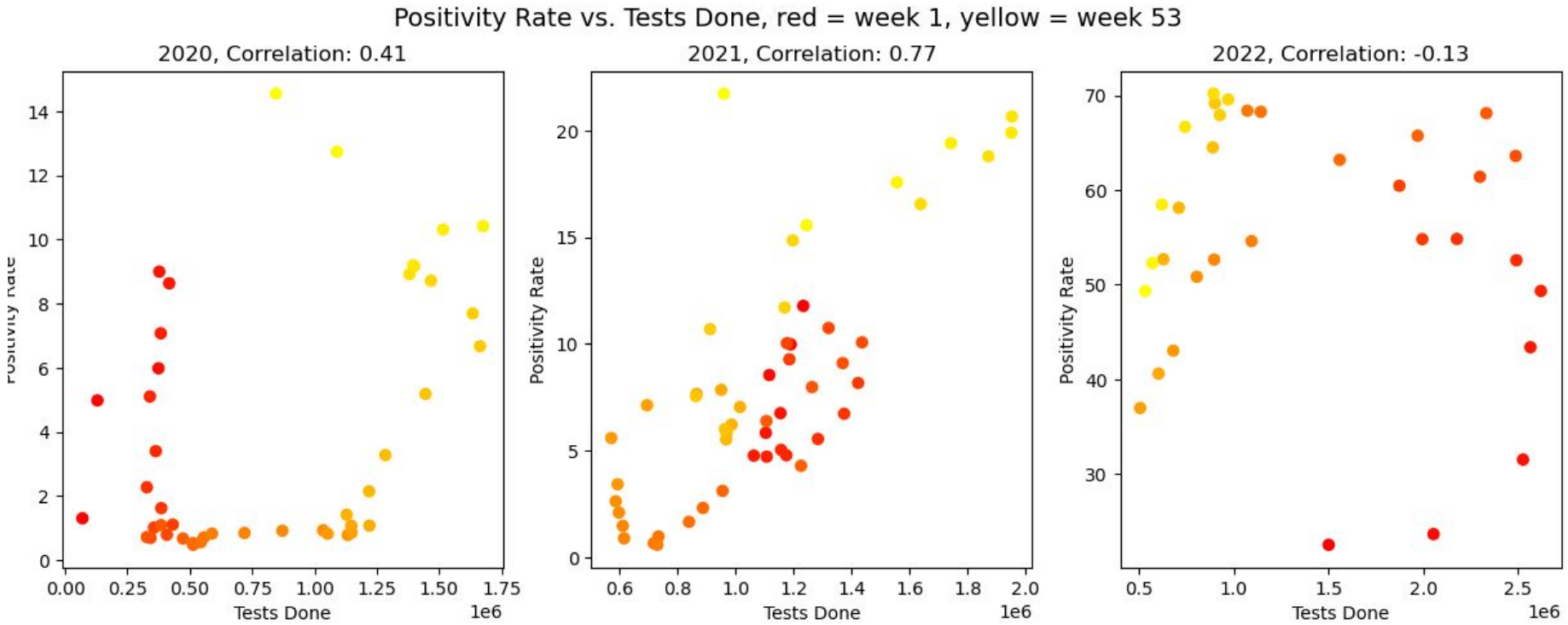

Positivity Rate vs. Tests Done
Positivity Rate
Tests Done
1e6
0
10
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
2.5


Positivity Rate vs. Tests Done
2020
2021
2022
Positivity Rate
Tests Done
1e6
0
10
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
2.5

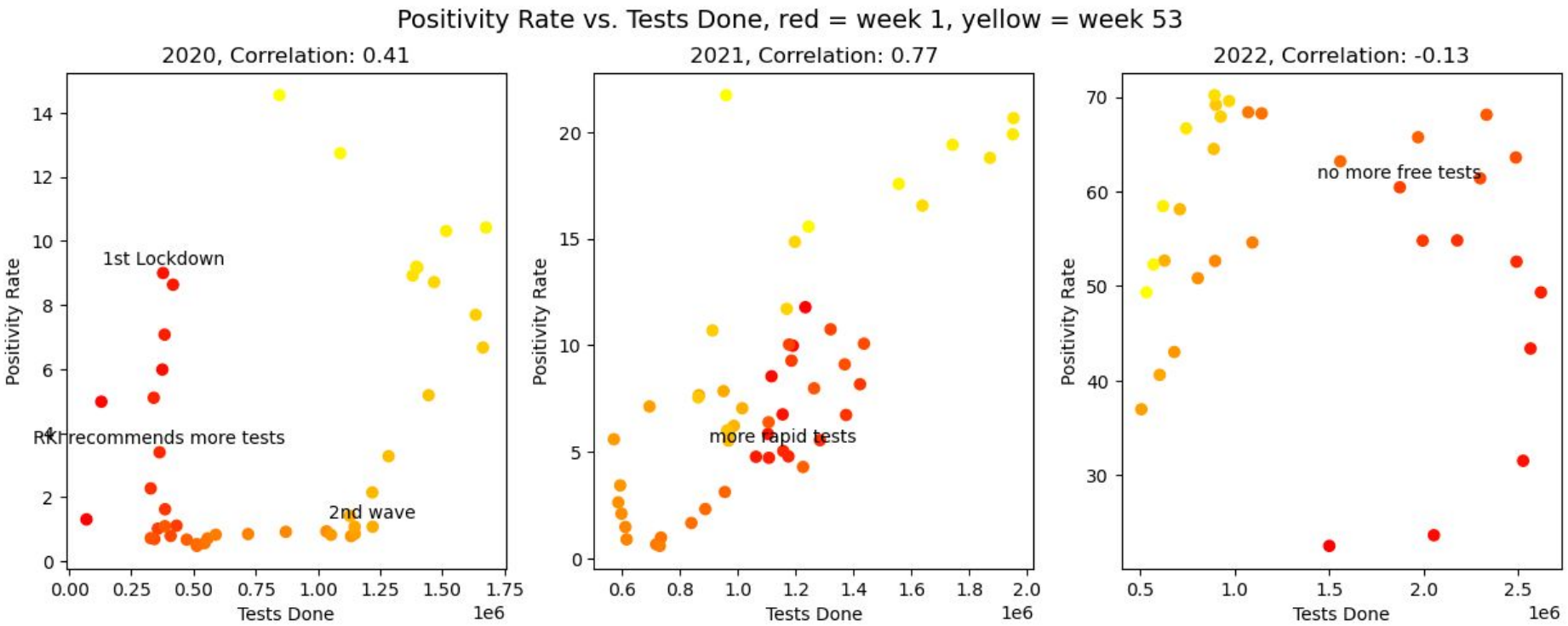
Analyze


Positivity Rate vs. Tests Done, red = week 1, yellow = week 53
2020, Correlation: 0.41
2021, Correlation: 0.77
2022, Correlation: -0.13
Positivity Rate
Tests Done
1e6

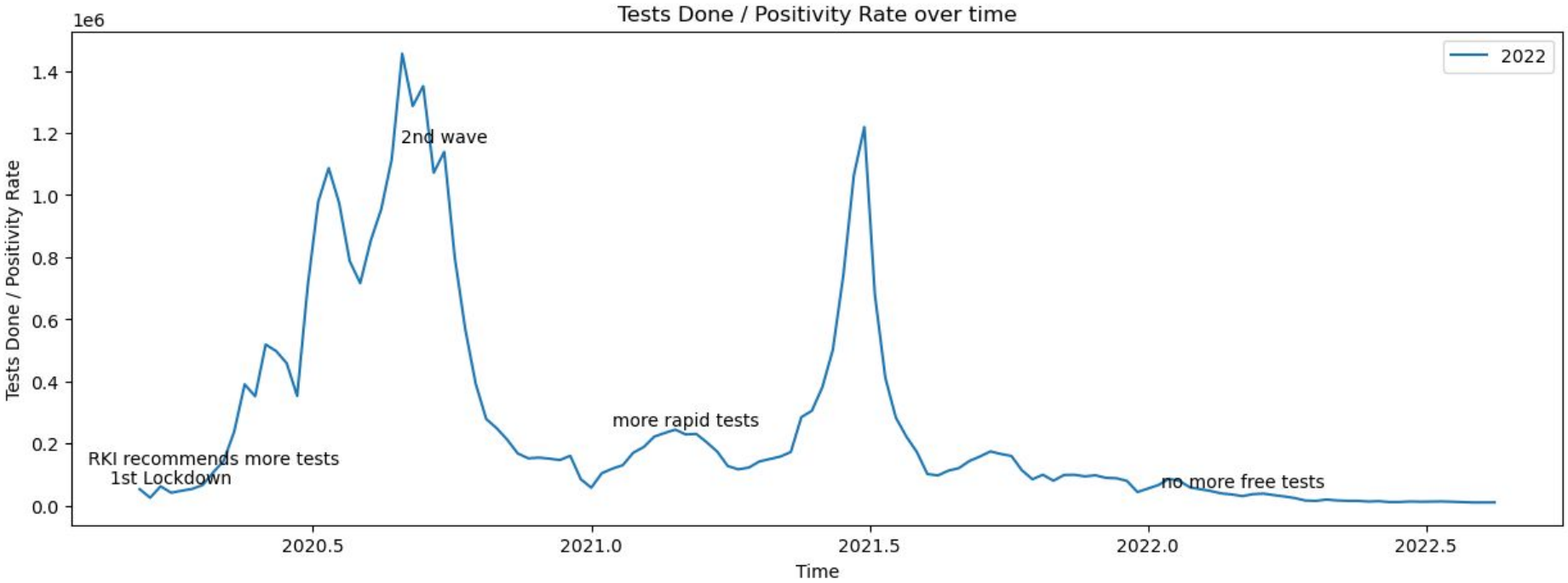
Analyze


Positivity Rate vs. Tests Done, red = week 1, yellow = week 53
2020, Correlation: 0.41
2021, Correlation: 0.77
2022, Correlation: -0.13
Positivity Rate
Tests Done
1e6

Analyze


Positivity Rate vs. Tests Done, red = week 1, yellow = week 53
2020, Correlation: 0.41
2021, Correlation: 0.77
2022, Correlation: -0.13
Positivity Rate
Tests Done
1e6
1st Lockdown
RKI recommends more tests
2nd wave
more rapid tests
no more free tests

Analyze


Tests Done / Positivity Rate over time
1e6
Tests Done / Positivity Rate
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2022
2nd wave
more rapid tests
RKI recommends more tests
1st Lockdown
no more free tests
2020.5
2021.0
2021.5
2022.0
2022.5
Time

Analyze

"Gibt es eine Korrelation zwischen der Positivitäts-Rate und der Anzahl abgeschlossener Tests in Deutschland zur Corona-Zeit?"

- Korrelation besteht in manchen Zeiträumen
- besonders Ende 2020 und 2021 das ganze Jahr über
- in anderen Zeiträumen kaum vorhanden
  - Gründe: fehlende Tests, keine kostenlosen Tests
- verschiedene Ereignisse / politische Beschlüsse spiegeln sich in Daten wieder

Publish and Preserve

- Ziel: Erfüllung der FAIR-Kriterien
- **F**indable, **A**ccessible, **I**nteroperable, **R**eusable
- Findable:
  - alles hochgeladen in öffentlichem GitHub Repository
  - GitHub sollte im Index verschiedener Suchmaschinen erfasst sein
    - zusätzlich möglich: Anfragen nach Indexing
  - alle Dateien im Repo haben einzigartige Identifier
  - maschinenlesbare Metadaten sind vorhanden
    - enthalten wichtige keywords
  - Synchronisation mit Research Data Repository (Zenodo)
    - Identifier auf Zenodo sind persistent!

Publish and Preserve

- Accessible:
  - GitHub/Zenodo sind vertrauenswürdig + sehr verbreitet
  - Herunterladen:
    - automatisch: git + ssh / https, GitHub for command line
    - manuell von Website
- Interoperable:
  - Daten als csv abgespeichert
    - sehr verbreitet, von vielen Tools unterstützt
  - Spaltennamen sind selbsterklärend und oft verwendet
  - maschinenlesbare Metadaten erstellt mit "dublin core generator"
    - -> verwenden somit typisches Vokabular + Format
  - Abhängigkeiten des Codes + Installation beschrieben

Publish and Preserve

- Reusable
  - Nutzung von Jupyter Notebooks für Preprocessing
    - reproduzierbar
    - nachvollziehbar
  - MIT non-copyleft license für Code, CC BY license für Daten
    - erlaubt (weitensgehend) freie Wiederverwendung
  - Datenqualität ist hoch
  - Datensatz wird sich nicht mehr ändern
    - -> Repository wird archiviert

Data Lifecycle

Vielen Dank für ihre Aufmerksamkeit!