Modern data analytics

COVID-19 analysis

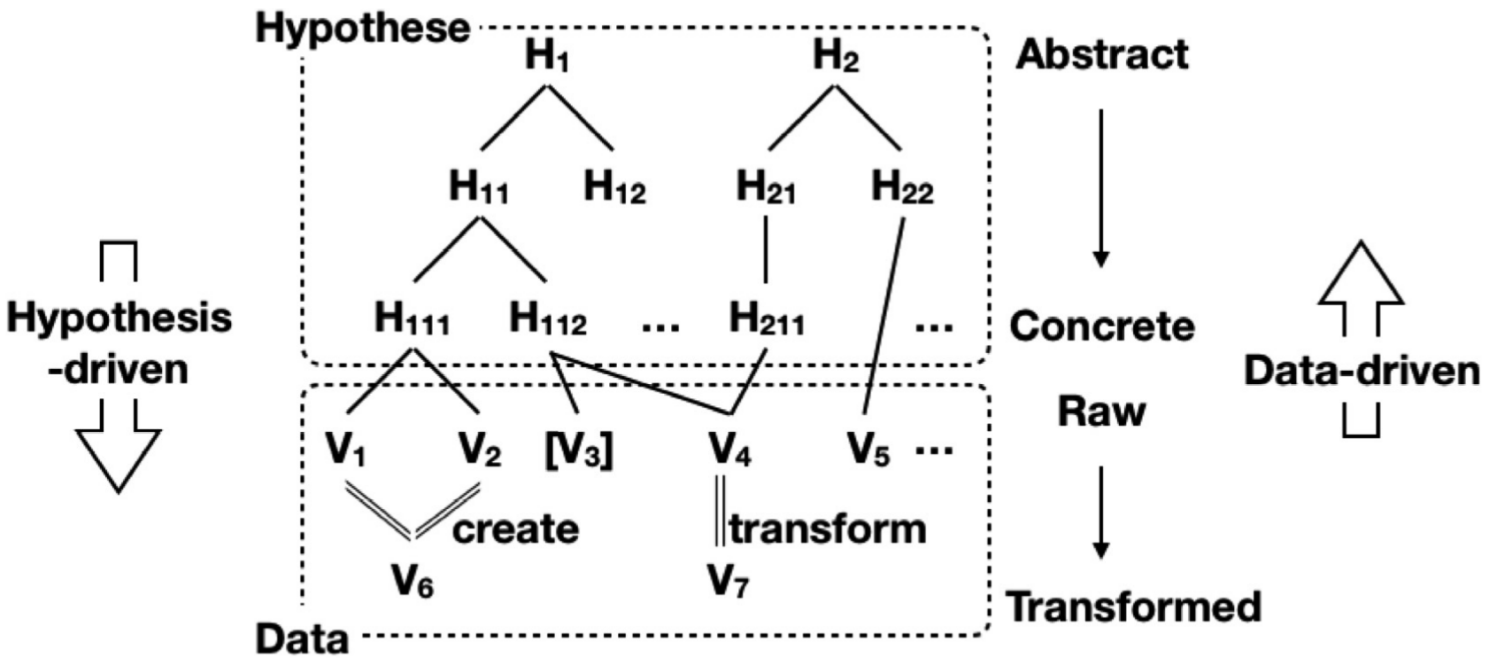
Sebastiaan Van den Broeck

KUL

August 13, 2022

Outline

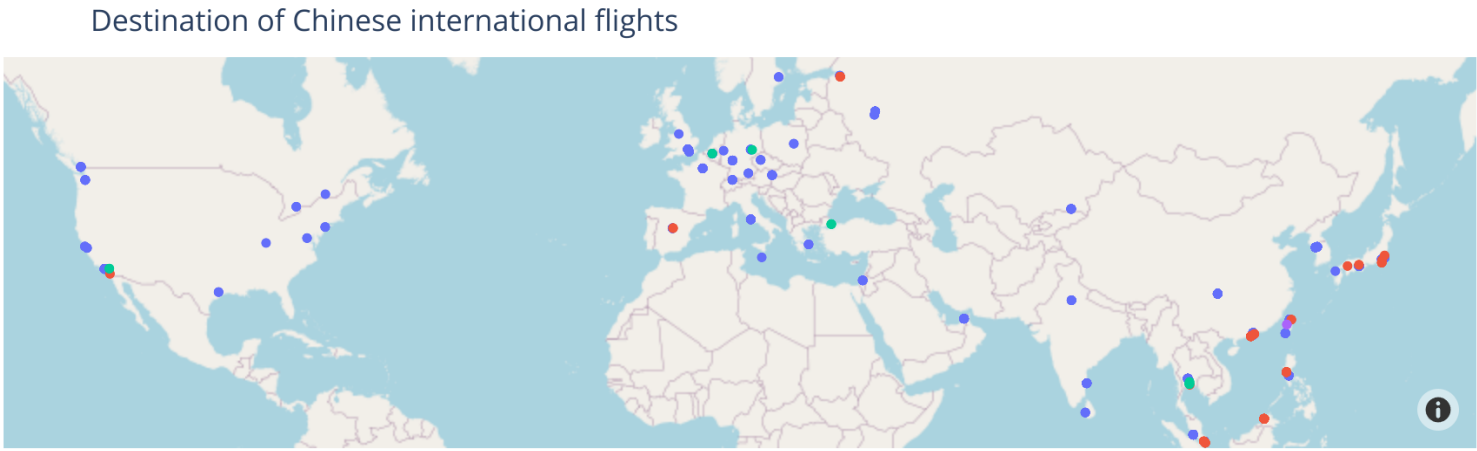
Methodology



Data sources

1. The COVID-19 dataset (The New York Times, 2021)
2. Economical information (Bureau of Economic Analysis, 2022)
3. The COVID-19 OpenSky dataset (Strohmeier et al., 2021)
4. The CORD-19 dataset (Lucy et al., 2021)

Graph mining



Graph mining

- Degree

  $c_D(i) = \sum_j^N x_{ij}$

  where i, j are nodes and x is the adjacency matrix.

- Betweenness

  $c_B(i) = \sum_{s,t \in i} \frac{\sigma(s,t|i)}{\sigma(s,t)}$

  where i is a node, s and t are source and target nodes, $\sigma(s,t)$ is the number of shortest paths between the source and target and $\sigma(s,t|v)$ is the number of shortest paths passing through the node v.

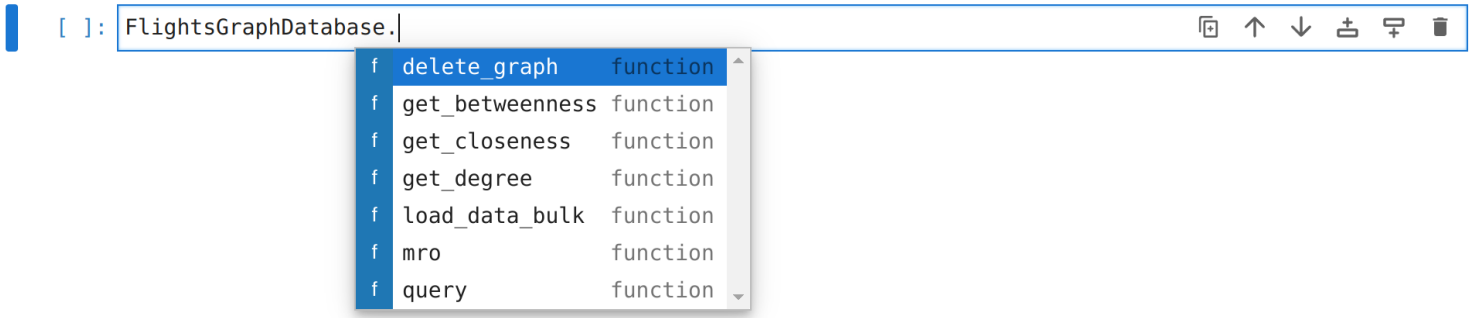
- Closeness

  $c_C(i) = \left[\sum_j^N d(i,j)\right]^{-1}$

  where d is a distance metric.

(Brandes, 2008; Opsahl et al., 2010)

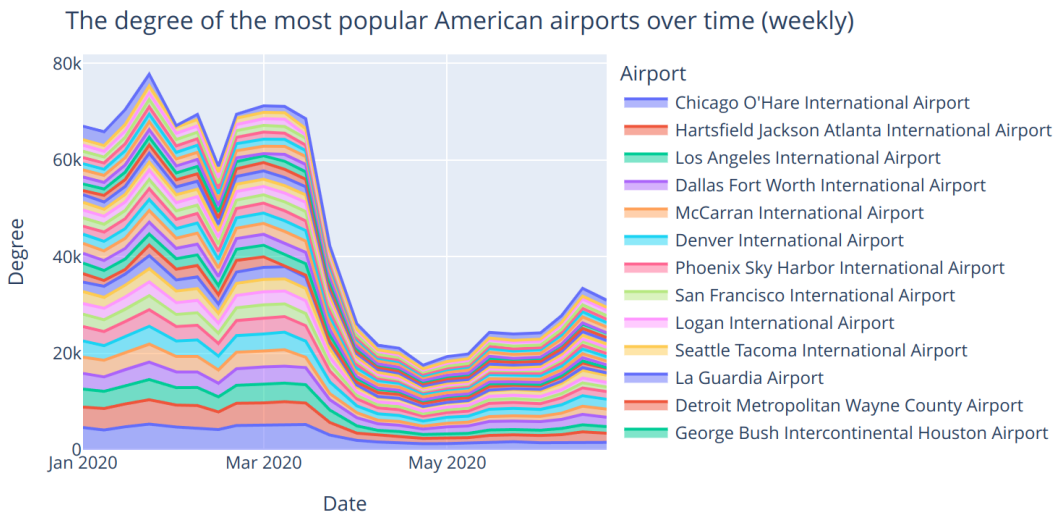
Graph mining - the code




```python
class FlightsGraphDatabase:
    def __init__(self, url, user, password):
        self.graph = Graph(url, auth=(user, password))


    def query(self, query) -> pd.DataFrame:
        """Runs a Cypher query on the graph and returns the results as a
        Pandas DataFrame."""

        return self.graph.run(query).to_data_frame()
```

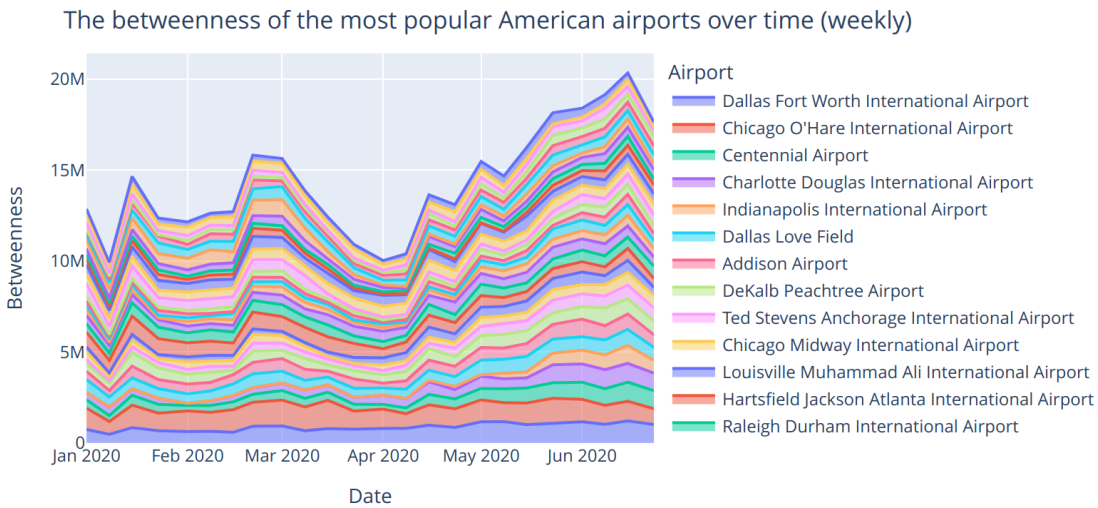
Graph mining - degree centrality

```python
def get_degree(self) -> pd.Dataframe:
    # First, create a projection
    self.query("CALL gds.graph.drop('flights', false)")
    self.query("CALL gds.graph.project('flights', 'Airport', 'FLIGHT') YIELD *")
    # Then, calculate the centrality for each node
    result = self.query("""CALL gds.degree.stream('flights')
                           YIELD nodeId, score MATCH (n:Airport)
                           WHERE ID(n) = nodeId RETURN n.name, n.code, score""")
    # Finally, sorting the results by degree
    result = result.sort_values("score", ascending=False)

    return result
```



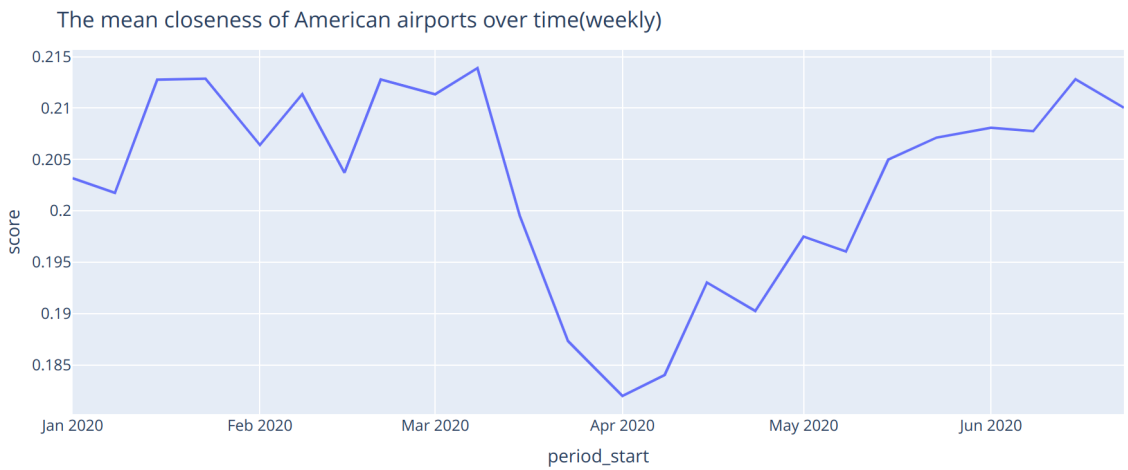
Graph mining - betweenness centrality

```python
def get_betweenness(self) -> pd.DataFrame:
    # First, create a projection
    self.query("CALL gds.graph.drop('flights', false)")
    self.query("CALL gds.graph.project('flights', 'Airport', 'FLIGHT') YIELD *")
    # Then, calculate the centrality for each node
    result = self.query("""CALL gds.betweenness.stream('flights')
                           YIELD nodeId, score MATCH (n:Airport)
                           WHERE ID(n) = nodeId RETURN n.name, n.code, score""")
    # Finally, sorting the results by degree
    result = result.sort_values("score", ascending=False)

    return result
```



Graph mining - closeness centrality

```python
def get_closeness(self) -> pd.DataFrame:
    # First, create a projection
    self.query("CALL gds.graph.drop('flights', false)")
    self.query("CALL gds.graph.project('flights', 'Airport', 'FLIGHT') YIELD *")
    # Then, calculate the centrality for each node
    result = self.query("""CALL gds.beta.closeness.stream('flights')
                           YIELD nodeId, score MATCH (n:Airport)
                           WHERE ID(n) = nodeId RETURN n.name, n.code, score""")
    # Finally, sorting the results by degree
    result = result.sort_values("score", ascending=False)

    return result
```



Text mining - preprocessing

```
def preprocess_text(self, series: pd.Series) -> pd.Series:
    """Applies some preprocessing steps to a collection of documents."""
    # Remove stopwords
    series = [gensim.parsing.preprocessing.remove_stopwords(i)
              for i in series]
    # Stem
    series = gensim.parsing.porter.PorterStemmer().stem_documents(series)
    # Remove numeric
    series = [gensim.parsing.preprocessing.strip_numeric(i)
              for i in series]
    # Remove punctuation
    series = [gensim.parsing.preprocessing.strip_punctuation(i)
              for i in series]
    # Remove special characters
    series = [re.sub("\\W+", " ", i) for i in series]
    # Remove short words
    series = pd.Series([gensim.parsing.preprocessing.strip_short(i)
                        for i in series])

    return series
```

Text mining - latent dirichet allocation

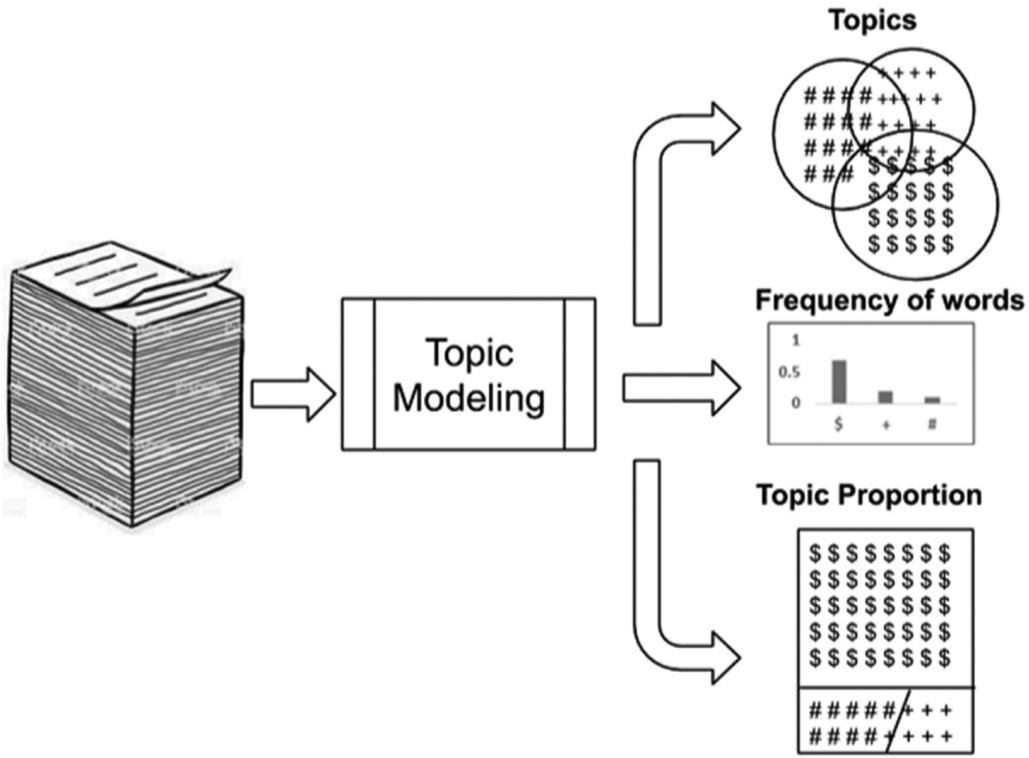
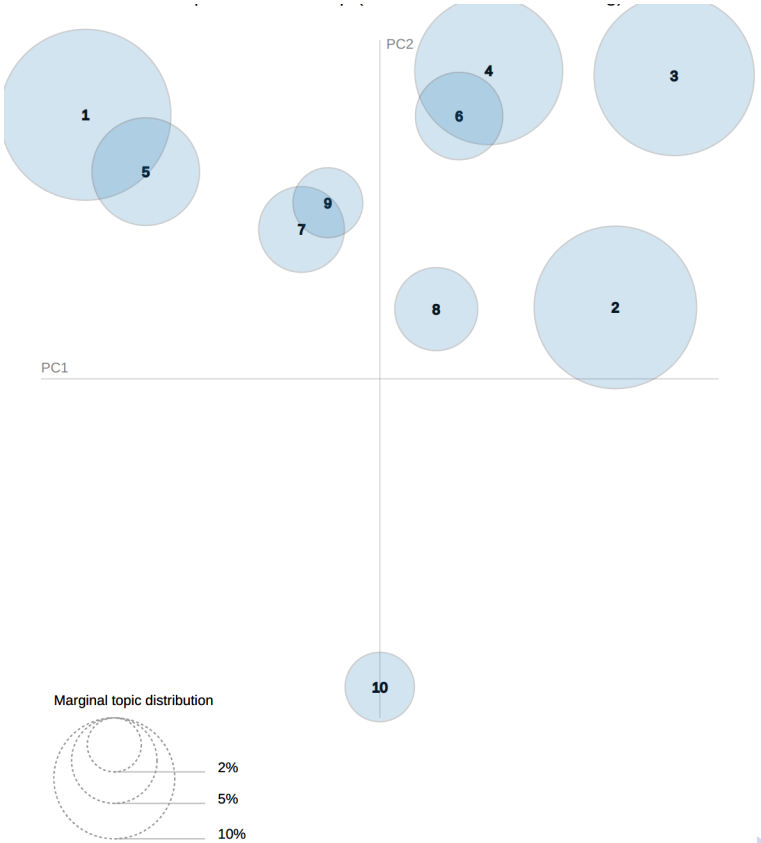


Figure: The area of topic modelling is concerned with making inferences from a huge collection of documents in terms of topics, frequency of words and topic proportions. Taken from Chauchan & Shah (2022)

Text mining - before the pandemic

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
cell	health	patient	cov	cell	infect	immun	vaccin	calv	der
protein	diseas	respiratori	detect	protein	respiratori	vaccin	viru	protein	die
viru	develop	studi	viru	activ	viral	cell	infect	viru	und
viral	model	infect	sequenc	lung	group	respons	influenza	structur	model
infect	public	associ	human	express	studi	protein	studi	studi	ein
activ	emerg	children	respiratori	induc	viru	antigen	result	result	network
express	data	clinic	virus	airwai	associ	antibodi	ibv	effect	von
rna	infecti	influenza	pcr	result	virus	specif	virus	test	mit
virus	studi	case	infect	increas	caus	express	protect	differ	bei
host	risk	hospit	sars	mice	patient	infect	effect	domain	cell



Text mining - during pandemic

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
health	patient	covid	test	patient	cov	protein	health	mask	der
covid	covid	model	covid	cell	sars	cov	studi	effect	die
studi	hospit	vaccin	model	covid	covid	cell	mental	covid	und
research	risk	case	data	diseas	infect	sars	depress	cov	text
data	studi	studi	method	treatment	patient	ace	associ	sars	studi
care	clinic	infect	detect	immun	antibodi	viral	anxieti	aerosol	infect
pandem	sever	number	perform	activ	respiratori	bind	women	patient	activ
provid	associ	effect	result	studi	coronaviru	covid	social	studi	gene
develop	group	data	studi	clinic	sever	viru	covid	infect	effect
social	patients	time	sampl	associ	studi	drug	effect	risk	model

Sources

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. Social Networks, 30(2), 136–145. https://doi.org/10.1016/j.socnet.2007.11.001

Bureau of Economic Analysis. (2022). Annual GDP per state [Dataset]. https://apps.bea.gov/regional/downloadzip.cfm

Chauhan, U., & Shah, A. (2022). Topic Modeling Using Latent Dirichlet allocation: A Survey. ACM Computing Surveys, 54(7), 1–35. https://doi.org/10.1145/3462478

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey. Multimedia Tools and Applications, 78(11), 15169–15211. https://doi.org/10.1007/s11042-018-6894-4

Klein, D. J. (2010). Centrality measure in graphs. Journal of Mathematical Chemistry, 47(4), 1209–1223. https://doi.org/10.1007/s10910-009-9635-0

Sources

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, et al.. 2020. CORD-19: The COVID-19 Open Research Dataset. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Online. Association for Computational Linguistics.

Manríquez, R., Guerrero-Nancuante, C., Martínez, F., & Taramasco, C. (2021). Spread of Epidemic Disease on Edge-Weighted Graphs from a Database: A Case Study of COVID-19. International Journal of Environmental Research and Public Health, 18(9), 4432. https://doi.org/10.3390/ijerph18094432

Sources

Matsumuro, M., & Miwa, K. (2019). Model for Data Analysis Process and Its Relationship to the Hypothesis-Driven and Data-Driven Research Approaches. In A. Coy, Y. Hayashi, & M. Chang (Eds.), Intelligent Tutoring Systems (Vol. 11528, pp. 123–132). Springer International Publishing. https://doi.org/10.1007/978-3-030-22244-4_16

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245–251. https://doi.org/10.1016/j.socnet.2010.03.006

Rehman, S. U., Khan, A. U., & Fong, S. (2012). Graph mining: A survey of graph mining techniques. Seventh International Conference on Digital Information Management (ICDIM 2012), 88–92. https://doi.org/10.1109/ICDIM.2012.6360146

Relman, D. A. (2020). To stop the next pandemic, we need to unravel the origins of COVID-19. Proceedings of the National Academy of Sciences, 117(47), 29246–29248. https://doi.org/10.1073/pnas.2021133117

Sources

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 63–70. https://doi.org/10.3115/v1/W14-3110

Strohmeier, M., Olive, X., Lübbe, J., Schäfer, M., & Lenders, V. (2021). Crowdsourced air traffic data from the OpenSky Network 2019–2020. Earth System Science Data, 13(2), 357–366. https://doi.org/10.5194/essd-13-357-2021

Stroustrup, B. (1988). What is object-oriented programming? IEEE Software, 5(3), 10–20. https://doi.org/10.1109/52.2020

The New York Times. (2021). Coronavirus (Covid-19) Data in the United States. Retrieved from https://github.com/nytimes/covid-19-data.