VOL. 10 NO. 6,
NOVEMBER, 2025
E-ISSN 3027-1401
P-ISSN 3026-9148



African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Arts and Sociological Research (JASR)

Leveragıng Python Language For Data- Drıven Communıcatıon Research

***Shiraz Bello; & **Goodluck Matthew Anemikaye**
*Department of Public Relations & Publicity Social Science, Kocaeli University.
**Department of Science, Computer Engineering, Istanbul Ticaret University.
Corresponding Author: samadu.bello@istanbulticaret.edu.tr
DOI: https://doi.org/10.70382/ajasr.v10i6.082

Abstract

Python is an interpreted, high-level, general-purpose programming language that has emerged as a versatile and widely adopted tool across numerous disciplines, including machine learning, software development, and data analysis. Over more than three decades of continuous development, Python has been applied across diverse sectors, demonstrating substantial adaptability and scalability. In response to the growing dominance of algorithm-driven and data-intensive methodologies, communication scholars are increasingly encouraged to engage with programming languages-particularly Python-as integral components of contemporary research practice. The integration of Python into communication research represents a paradigmatic shift, necessitating that researchers with an interest in computational approaches adopt analytical frameworks and problem-solving logics traditionally associated with computer science. Owing to its relatively low learning curve, extensive open-source ecosystem, and wide availability of data-oriented libraries, Python remains accessible even to researchers with limited prior programming experience. This article advocates for the systematic adoption and application of Python within communication research, emphasizing its potential to enhance data analysis, visualization, and interpretative rigor. By expanding the methodological toolkit available to scholars, Python enables a broader and more nuanced approach to research interpretation in the evolving field of marketing communication.

Keyword: Communication Research, Python, Jupyter NoteBook, Data Analysis, Data Visualization

Introduction

Over the years, communication scholars have employed a range of methodological and analytical tools to examine data generated from communication research. As the volume, velocity, and variety of research data have increased, advances in digital technologies have transformed how data are collected, analyzed, and presented. Contemporary research environments now emphasize efficiency, transparency, and analytical rigor, prompting scholars to adopt innovative software tools and computational approaches to enhance data analysis (Neuendorf, 2017; Creswell & Plano Clark, 2018).

Currently, researchers have access to a wide array of data visualization and analytics platforms, including Power BI, Qlik, Looker, Sisense, and Tableau. These tools typically feature web-based interfaces and drag-and-drop functionalities that enable users to create interactive dashboards and reports with minimal technical complexity. Such platforms facilitate data cleaning, processing, and visualization, thereby improving usability and supporting clearer interpretation of complex datasets (Few, 2012; Knaflic, 2015). As a result, visual analytics has become an essential component of empirical research communication and knowledge dissemination.

Nonetheless, some communication scholars-particularly those with interests in computational methods and advanced analytics-prefer coding-based approaches to data analysis. For researchers with foundational programming skills, several analytical environments support flexible and reproducible workflows. Among these, the Python programming language has emerged as a prominent tool within the research analytics toolkit. Python is an interpreted, high-level, general-purpose programming language designed with an emphasis on readability and simplicity. Developed by Guido van Rossum and first released in 1991, Python supports multiple programming paradigms and has become central to data science and scientific computing (van Rossum & Drake, 2009).

In recent years, the growing integration of machine learning and algorithmic analysis into research practice has further elevated Python's relevance. Python-based libraries such as NumPy, pandas, scikit-learn, and TensorFlow enable efficient data manipulation, statistical modeling, and machine learning applications, reducing the rigor and complexity traditionally associated with large-scale data processing (McKinney, 2010; Pedregosa et al., 2011). Over the past three decades, Python has been widely applied across domains including machine learning, software development, application building, and data analytics, demonstrating its versatility and scalability.

These capabilities are equally transferable to communication research, where scholars increasingly engage with large datasets derived from social media platforms, digital content, and online user interactions. Consequently, there is a growing imperative for communication researchers to adopt computational and machine learning approaches in order to remain methodologically relevant. As contemporary communication data are increasingly shaped by algorithmic processes, venturing into machine learning and coding-based analytics represents a critical step toward advancing

empirical rigor and theoretical innovation in the field (Lewis & Westlund, 2015; Salganik, 2018).

Machine languages like Python is a non complicated language which is a perfect tool for data analysis. Pythons comes with command line interpreters which help to simplify and read datasets. With the use of Python command-line interpreter, student can easily check how operators or functions work (Downey et al, 2008). Python interpreter has built-in help module, which can significantly improves the process of understanding of different aspects of language. (Bogdanchikov et al, 2013) In order to understand programming languages, freshmen (communication researchers which have no experience in programming) need to learn how to think like computer scientist, and that requires a great effort and acomplete shift in their paradigm of thinking. (Jian et al,2011) . This is true. Moreso, with the availabilty of data code types, communication researchers could easily write and implement codes with the desired results and outcomes. Python also contains libraies which provide handy call of required functions to hlep run the data for the research,

Today, with the frequent application of programming language on datasets, it is important for commuunication researchers to learn at least one programming language, because it is relevant for a quick deciphering of the data in quantitive research. Also the data results obtained from the programming language like Python could serve as a verification tool for the Statistical Package for the Social Sciences (SPSS), which communication scholars are alrealry acquiante.

It is time to shift the boundaries of research interpretation tools used in data analyses in the field of marketing communication. Hence this article calls for the adoption and application of Python in communication research.

Objective of the study

The intention of this study moves to highlight the importance of Python Programmimng Language in communication research while it showscase the possible utilization of Python as an interpretation tool of communication research data.

Secondly, Communication scholars will be able to see how Python can easily read datasets and provide the expected results or outcome for the research work. To have a visual insight of this action, the article displays a case of Python scripts, written to analyze a dataset, which are short and less complex.

Communication Research

In conducting scientific investigations, researchers are required to make a range of fundamental decisions aimed at advancing knowledge and contributing meaningfully to existing scholarship (McLeod & Tichenor, 2003). This objective underpins all forms of research, including communication research. Historically, communication research has evolved substantially and has made significant contributions to academic literature by systematically examining how communication operates within society. Communication research can be defined as the structured investigation of communication processes, effects,

and dynamics across diverse contexts and channels. It seeks to explain how information is produced, transmitted, received, and interpreted, as well as how communication influences attitudes, behaviors, and social relationships (Craig, 1999; McQuail, 2010).

Quantitative communication research, in particular, is rooted in a highly conventionalized social scientific tradition. As with all scientific endeavors, quantitative research is shaped by social norms, institutional expectations, and professional standards. Understanding quantitative communication research therefore requires acknowledging its conventional nature-one that is reinforced through methodological textbooks, formal training of emerging scholars, criteria for tenure and promotion, evaluation mechanisms, and the peer-review process (Neuendorf, 2017). Within this framework, quantitative communication researchers often draw upon established methodological practices, replicating widely accepted analytical approaches to ensure rigor, reliability, and comparability of findings.

According to Mahmod (2011), communication researchers typically apply a reconstructed logic and follow a linear research process. This approach emphasizes the articulation of research questions and hypotheses, the precise operationalization and measurement of variables, and the empirical testing of relationships grounded in general causal explanations. These methodological principles align closely with contemporary computational data analysis environments. Notably, the Python programming language provides extensive statistical and analytical libraries that support core quantitative research practices.

Python-based libraries facilitate essential quantitative analyses commonly employed in communication research, including descriptive statistics (mean, median, and mode), correlation analysis, and regression modeling. Using concise and accessible coding scripts, researchers can efficiently import datasets, manipulate variables, and execute statistical analyses without the need for advanced programming expertise (McKinney, 2010; Van Rossum & Drake, 2009). This accessibility allows communication scholars to focus on theoretical interpretation and empirical rigor rather than technical complexity. Consequently, Python enhances the analytical capacity of quantitative communication research by promoting efficiency, transparency, and reproducibility in the research process.

THE ADOPTION OF PYTHON TO COMMUNICATION RESEARCH

As mentioned earlier, Python is a high-level, interpreted, and multipurpose programming language whose versatility extends across numerous fields of research, including communication studies. Its multipurpose nature allows researchers to integrate data collection, processing, analysis, and visualization within a single computational environment. The rationale for adopting Python as a data analysis tool in communication research is clearly articulated by McKinney (2013), who argues that:

> *"For data analysis and interactive, exploratory computing and data visualization, Python will inevitably draw comparisons with the many other domain-specific open source and commercial programming languages and tools in wide use, such as R, MATLAB, SAS, Stata, and others. In recent years, Python's improved library support (primarily pandas) has made it a strong*

alternative for data manipulation tasks. Combined with Python's strength in general-purpose programming, it is an excellent choice as a single language for building data-centric applications" (p. 16).

Beyond data analysis, Python is widely applied in areas such as game development, software engineering, language processing, graphical user interface (GUI) design, and image processing. However, the primary interest of communication research scholars lies in its capacity for scientific and numerical computation. Sabbith et al. (2019) note that Python's high processing speed, enhanced productivity, and extensive availability of scientific tools-particularly libraries such as Scientific Python and Numeric Python-have contributed to its widespread adoption in computational and data-intensive applications. These features have positioned Python as a central tool for the computation and processing of scientific research data.

In communication research, Python is especially valuable for exploratory data analysis, as it allows researchers to examine patterns, trends, and relationships within datasets using relatively simple and readable code. This accessibility enables accurate and efficient analysis while minimizing technical complexity. Moreover, Python facilitates the rapid creation of graphical representations-such as histograms and bar plots-which support effective data exploration and enhance the interpretability of research findings (McKinney, 2010; Hunter, 2007).

It is also important to note that Python supports multiple core phases of the data analysis process. These phases typically include data processing, data cleaning, descriptive and inferential analysis, and modeling through statistical or algorithmic techniques (VanderPlas, 2016). To support these analytical stages, several Python packages are particularly essential. NumPy provides efficient numerical computation, Pandas enables data manipulation and organization, Matplotlib supports data visualization, and Seaborn offers advanced statistical graphics built on Matplotlib (Harris et al., 2020; Waskom, 2021). Collectively, these tools make Python a robust and flexible analytical framework for quantitative communication research.

Python Overview

Python avails data research with any libraries like NumPy, Pandas, Seaborn, Matplotlib, Sklearn, etc (Nelli,2015). Following Nelli's list of libraries, Here are a brief explanation of the list:

NumPy: NumPy is a library written in Python, used for numerical analysis in Python. It stores the data in the form of nd-arrays (n-dimensional arrays).

Pandas: Pandas is mainly used for converting data into tabular form and hence, makes the data more structured and easily to read.

Matplotlib: Matplotlib is a data visualisation and graphical plotting package for Python and its numerical extension NumPy that runs on all platforms.

Seaborn: Seaborn is a Python data visualisation package based on matplotlib that is tightly connected with pandas data structures. The core component of Seaborn is visualisation, which aids in data exploration and comprehension.

Sklearn: Scikit-learn is the most useful library for machine learning in Python. It includes numerous useful tools for classification, regression, clustering, and dimensionality reduction.

METHODOLOGY

It is necessary to show how Python is used in a descriptive presentation using Covid Data set of Nigerians Living in Istanbul. This will be achieved by utilizing this Covid Data set in the following formats: Data Processing, Data Cleaning, Data Analysis/Description, Modeling and Algorithms.The Python Platform recommended for this procedure is the Jupyter Notebook.

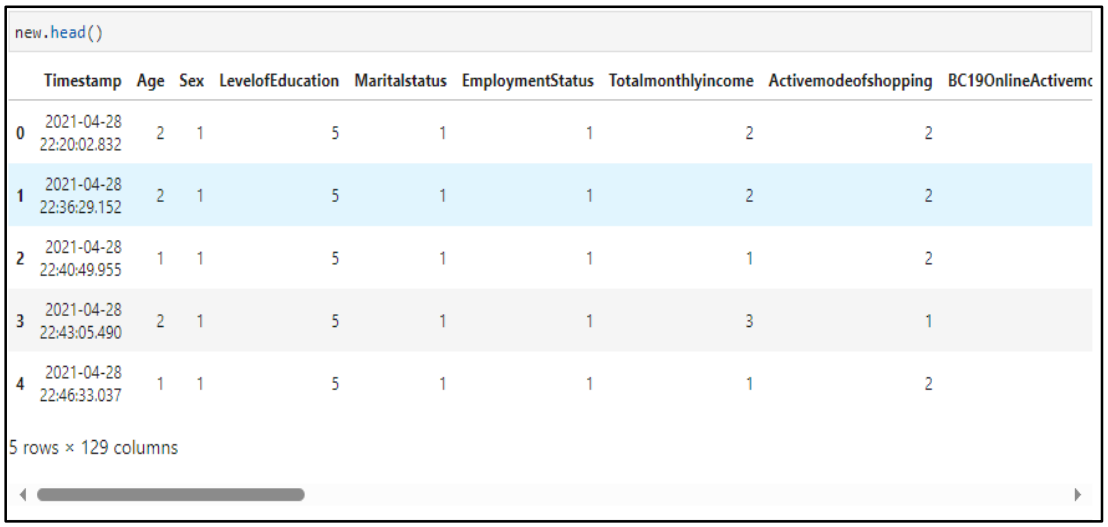
Data Processing:

Jupyter Notebook provides a dynamic and user-friendly computational environment that supports the entire data processing and analysis pipeline. One of its core functionalities is the seamless loading and integration of datasets, which enables users to efficiently import, manipulate, and manage data directly within the notebook interface. This foundational capability establishes the basis for a wide range of exploratory and analytical tasks, allowing researchers to transition smoothly from raw data to meaningful insights (Kluyver et al., 2016). The use of Jupyter Notebook has also transformed how students, researchers, and data scientists document and communicate analytical results. The ability to integrate executable code, narrative text, mathematical expressions, and visualizations within a single document reduces the burden of traditional report writing and enhances analytical transparency. As noted by Hunter, Cal, and San Luis (2020), consistent engagement with Jupyter Notebook improves users' capacity to produce clear, reproducible reports and to translate statistical analyses effectively through numerical and graphical summaries. This makes the platform particularly valuable for course assignments, academic publications, data journalism, and industry-based data science projects, where the creation of dynamic, "living" documents is increasingly essential.

Moreover, the interactive nature of the Jupyter environment enables step-by-step data exploration through the execution of discrete code cells. This structure allows users to inspect, transform, and evaluate data iteratively, thereby facilitating the identification of patterns, anomalies, and trends. Such an iterative and exploratory workflow is especially advantageous in empirical research, as it supports continuous refinement of analytical strategies and promotes deeper engagement with the data. Consequently, Jupyter Notebook enhances both the rigor and interpretability of data analysis across research contexts.



Figure 1: Reading the document file

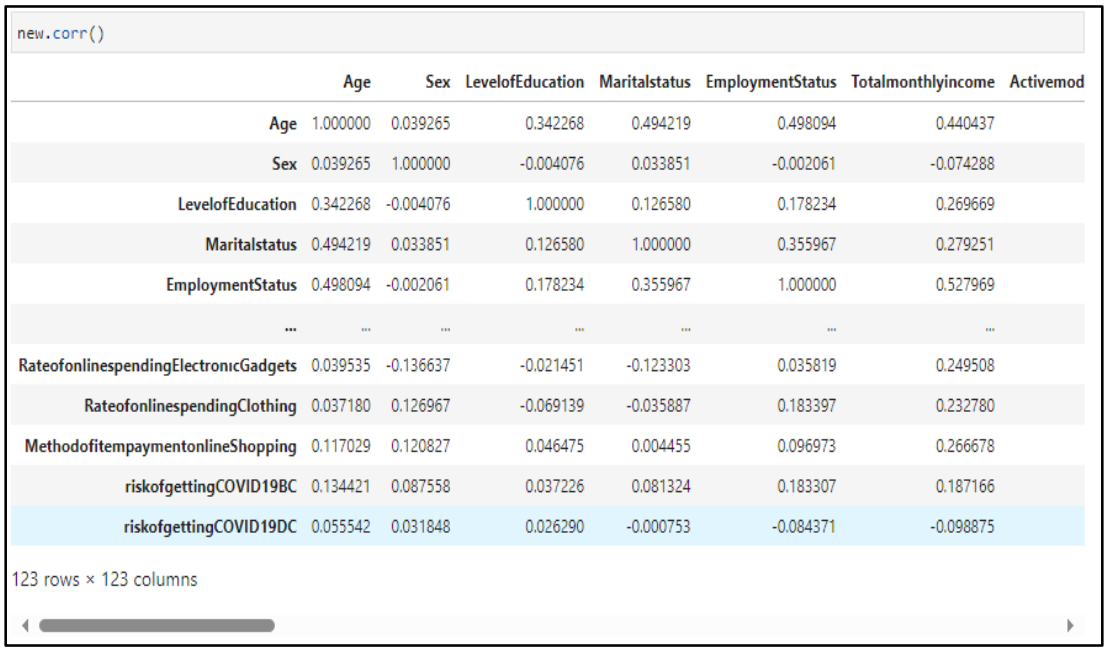


new.head()

	Timestamp	Age	Sex	LevelofEducation	Maritalstatus	EmploymentStatus	Totalmonthlyincome	Activemodeofshopping	BC19OnlineActivemc
0	2021-04-28 22:20:02.832	2	1	5	1	1	2	2	
1	2021-04-28 22:36:29.152	2	1	5	1	1	2	2	
2	2021-04-28 22:40:49.955	1	1	5	1	1	1	2	
3	2021-04-28 22:43:05.490	2	1	5	1	1	3	1	
4	2021-04-28 22:46:33.037	1	1	5	1	1	1	2	

5 rows × 129 columns

Figure 2: Processing the Dataset

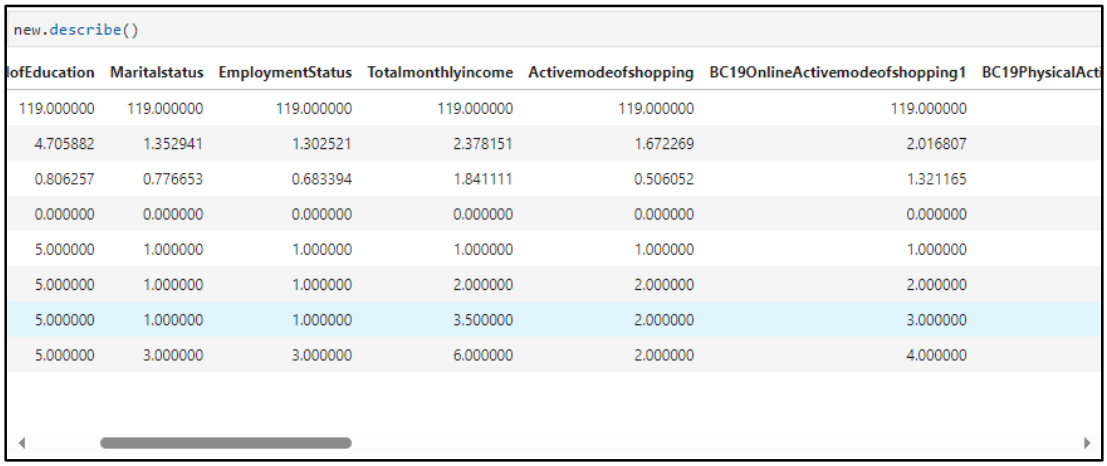


new.corr()

	Age	Sex	LevelofEducation	Maritalstatus	EmploymentStatus	Totalmonthlyincome	Activemod
Age	1.000000	0.039265	0.342268	0.494219	0.498094	0.440437	
Sex	0.039265	1.000000	-0.004076	0.033851	-0.002061	-0.074288	
LevelofEducation	0.342268	-0.004076	1.000000	0.126580	0.178234	0.269669	
Maritalstatus	0.494219	0.033851	0.126580	1.000000	0.355967	0.279251	
EmploymentStatus	0.498094	-0.002061	0.178234	0.355967	1.000000	0.527969	
...	...	...	...	...	...	...	
RateofonlinespendingElectronicGadgets	0.039535	-0.136637	-0.021451	-0.123303	0.035819	0.249508	
RateofonlinespendingClothing	0.037180	0.126967	-0.069139	-0.035887	0.183397	0.232780	
MethodofitempaymentonlineShopping	0.117029	0.120827	0.046475	0.004455	0.096973	0.266678	
riskofgettingCOVID19BC	0.134421	0.087558	0.037226	0.081324	0.183307	0.187166	
riskofgettingCOVID19DC	0.055542	0.031848	0.026290	-0.000753	-0.084371	-0.098875	

123 rows × 123 columns

Figure 3: Dataset Correlation



```
new.describe()
```

lofEducation	Maritalstatus	EmploymentStatus	Totalmonthlyincome	Activemodeofshopping	BC19OnlineActivemodeofshopping1	BC19PhysicalActi
119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	
4.705882	1.352941	1.302521	2.378151	1.672269	2.016807	
0.806257	0.776653	0.683394	1.841111	0.506052	1.321165	
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
5.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
5.000000	1.000000	1.000000	2.000000	2.000000	2.000000	
5.000000	1.000000	1.000000	3.500000	2.000000	3.000000	
5.000000	3.000000	3.000000	6.000000	2.000000	4.000000	

Figure 4: Dataset Description

Figure 1: Reading the document file

This figure illustrates the initial data import stage, where the dataset is loaded from an Excel file into the Python environment using the *pandas* library. This step ensures that the raw survey data are successfully accessed and prepared for subsequent analysis.

Figure 2: Processing the dataset

Figure 2 presents a preview of the dataset using descriptive rows and columns. It shows key socio-demographic and behavioral variables (e.g., age, sex, education level, employment status, income, and online shopping activities). Displaying the first few observations verifies data structure, variable coding, and completeness before statistical analysis.

Figure 3: Dataset correlation

This figure reports the Pearson correlation matrix among the study variables. The matrix highlights the strength and direction of linear relationships between socio-demographic factors, online shopping behaviors, and COVID-19–related risk perceptions. Such correlations provide an initial understanding of associations and guide further inferential analyses.

Figure 4: Dataset description

Figure 4 summarizes the descriptive statistics of selected variables, including count, mean, standard deviation, and minimum–maximum values. These statistics offer an overview of the central tendency and dispersion of the data, supporting an assessment of data distribution and suitability for subsequent quantitative analysis.

Data Cleaning:

On the Jupyter platform, crucial tasks of data cleaning and transformation, offering a robust and interactive environment for these fundamental data preparation activities is made available. In the realm of data cleaning, it provides a flexible workspace where you can employ

coding techniques to systematically address challenges such as missing values, outliers, and inconsistencies within the dataset. Mckinney (2017) believes that missing data occurs commonly in many data analysis applications. If the overall goal is to share their model with other collaborators, these scientists must perform cleaning steps to make it easier for others to understand their code. (Helen,2021) Below is a clear example of data cleaning using our sample dataset.

```
new.dtypes

Timestamp                                datetime64[ns]
Age                                              int64
Sex                                              int64
LevelofEducation                                 int64
Maritalstatus                                    int64
                                                 ...
RateofonlinespendingElectronicGadgets            int64
RateofonlinespendingClothing                     int64
MethodofitempaymentonlineShopping              float64
riskofgettingCOVID19BC                           int64
riskofgettingCOVID19DC                           int64
Length: 129, dtype: object
```

Figure 5: Data Type Identification

In [7]:
```
df.isna().describe()
```

Out[7]:

	Timestamp	Age	Sex	LevelofEducation	Maritalstatus	EmploymentStatus	Totalmonthlyincome	Activemodeofshopping	BC19OnlineActivemodeofshopp
count	119	119	119	119	119	119	119	119	
unique	1	1	1	1	1	1	1	1	
top	False	False	False	False	False	False	False	False	
freq	119	119	119	119	119	119	119	119	

4 rows × 129 columns

In [12]:
```
def replace_strings_with_0(column):
    return column.apply(lambda x: 0 if isinstance(x, str) else x)
```

In [24]:
```
#Clean Data to have non numerical replaced with 0
df = df.apply(replace_strings_with_0, axis=0)
```

In [25]:
```
# save cleaned data
file_path = 'output.xlsx'

# Save DataFrame to Excel file
df.to_excel(file_path, index=False)
```

In [26]:
```
#Replacing na with 0 part of data cleaning
df=df.fillna(value=0)
```

In [27]:
```
# checking for the cleanlines of data after cleaning it
df.isna().sum()
```

Figure 6: Data Replacement

Figure 5: Data Type Identification

Figure 5 presents the identification of variable data types in the dataset using the dtypes function. It shows that most variables are stored as numerical types (integers and floats), while the timestamp variable is recorded in datetime format. This step is essential for verifying correct variable encoding and ensuring compatibility with subsequent statistical and correlation analyses.

Figure 6: Data Replacement

Figure 6 illustrates the data cleaning and replacement process. Descriptive checks are first conducted to identify inconsistencies and non-numeric entries. Textual values are then systematically replaced or converted into appropriate numerical formats, missing or invalid entries are handled, and the cleaned dataset is exported to a new Excel file. This procedure ensures data consistency, reduces errors, and improves the reliability of further quantitative analyses.

Data Analysis/Description

By harnessing these visualization libraries, Jupyter facilitates the seamless integration of code-driven insights with graphical representations. This synthesis not only enhances the interpretability of complex datasets but also enables users to communicate findings effectively, making Jupyter a potent tool for data scientists, analysts, and researchers seeking to convey compelling narratives through their analyses.

```
import seaborn as sns

num_col = new._get_numeric_data().columns
describe_num_df = new.describe(include=['int64','float64'])
describe_num_df.reset_index(inplace=True)
# To remove any variable from plot
describe_num_df = describe_num_df[describe_num_df['index'] != 'count']
for i in num_col:
  if i in ['index']:
    continue
  sns.factorplot(x="index", y=i, data=describe_num_df)
  plt.show()
```

Figure 7: Data Cleaning



Figure 8: Data Visualization

```
import matplotlib.pyplot as plt
plt.matshow(new.corr())
plt.show()
```

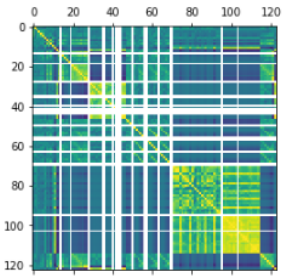


Figure 9: Modeling

```
corr =new.corr()
corr.style.background_gradient(cmap='coolwarm')
# 'RdBu_r', 'BrBG_r', & PuOr_r are other good diverging colormaps
```

```
C:\Users\Hp\anaconda3\lib\site-packages\pandas\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\Hp\anaconda3\lib\site-packages\pandas\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

	Age	Sex	LevelofEducation	Maritalstatus	EmploymentStatus	Totalmonthlyincome	Activemodeofshopping	BC19OnlineActivemodeofshopping1	BC19
Age	1.000000	0.039265	0.342268	0.494219	0.498094	0.440437	0.137928	0.128316	
Sex	0.039265	1.000000	-0.004076	0.033851	-0.002061	-0.074288	-0.116583	0.147943	
LevelofEducation	0.342268	-0.004076	1.000000	0.126580	0.178234	0.269669	0.281014	0.012636	
Maritalstatus	0.494219	0.033851	0.126580	1.000000	0.355967	0.279251	0.081176	0.093279	
EmploymentStatus	0.498094	-0.002061	0.178234	0.355967	1.000000	0.527969	0.068572	0.050638	
Totalmonthlyincome	0.440437	-0.074288	0.269669	0.279251	0.527969	1.000000	0.052282	0.220342	
Activemodeofshopping	0.137928	-0.116583	0.281014	0.081176	0.068572	0.052282	1.000000	-0.270553	
BC19OnlineActivemodeofshopping1	0.128316	0.147943	0.012636	0.093279	0.050638	0.220342	-0.270553	1.000000	
BC19PhysicalActivemodeofshopping2	0.057710	-0.015322	0.173448	0.009477	0.149647	0.123087	0.288590	0.047500	
DC19OnlineActivemodeofshopping1	0.102984	0.119803	-0.006206	0.000430	0.168760	0.238477	-0.299190	0.699288	
DC19PhysicalActivemodeofshopping2	0.064381	0.093287	0.177996	0.044795	-0.051666	0.037933	0.341390	0.140482	
Everusedanonlineshoppingplatform	-0.191811	-0.027835	0.096228	-0.132005	-0.153205	-0.253378	0.134931	-0.379760	
WhenstartOnlinetobuyandorderitems	0.094967	0.007259	-0.104094	0.020951	-0.006982	0.014552	-0.068901	-0.013057	
RateyourUseofOnlineshopsBC	0.170793	0.133750	0.042897	0.113633	0.142478	0.251937	-0.253515	0.571894	
V16	nan	nan	nan	nan	nan	nan	nan	nan	
ConvenienceBC	0.114103	-0.106593	0.070191	0.047299	0.201319	0.235505	-0.155554	0.332267	
VproductsBC	0.101932	0.061368	-0.076568	0.085375	0.026288	0.313041	-0.006455	0.389759	
BetterpricesBC	0.060515	-0.009319	0.037208	-0.005955	0.121580	0.203214	-0.016258	0.186167	

Figure 10: Data Diverging Colormaps

Figure 7: Data Cleaning

Figure 7 illustrates the data cleaning and preparation stage of the analysis. Descriptive statistics were computed for all numerical variables to identify missing values, data distributions, and potential anomalies. The exclusion of non-analytical indices ensured that only meaningful numeric variables were retained for subsequent analysis. This step was essential for improving data quality and ensuring the reliability of further statistical modeling and visualization procedures.

Figure 8: Data Visualization

Figure 8 presents an exploratory visualization of numerical variables using factor-based plots. These visualizations provide an initial overview of variable distributions and trends, allowing for the identification of patterns, variability, and potential outliers within the dataset. Such

exploratory visual analysis supports preliminary insights into consumer behavior and informs decisions regarding appropriate modeling techniques.

Figure 9: Modeling

Figure 9 displays a correlation matrix generated during the modeling phase. The matrix visualizes the strength and direction of relationships among the study variables, enabling the identification of positively and negatively correlated constructs. This step is critical for assessing multicollinearity, understanding inter-variable associations, and guiding the selection of variables for advanced statistical analyses such as regression or structural equation modeling.

Figure 10: Data Diverging Colormaps

Figure 10 shows a diverging colormap representation of the correlation matrix, enhancing interpretability through color gradients. Positive correlations are represented by warmer tones, while negative correlations are shown in cooler tones, allowing for rapid visual assessment of relational patterns among demographic, behavioral, and social media marketing variables. This visualization aids in highlighting key associations and supports the empirical interpretation of relationships within the dataset.

DISCUSSION AND CONCLUSION

In conclusion, the integration of Python into communication research represents a transformative development in contemporary research practice. In particular, the use of Jupyter Notebooks provides a powerful and flexible environment that enables researchers to seamlessly integrate code, data visualizations, and explanatory text within a single computational document. This integrated workflow significantly enhances research transparency and reproducibility, as all stages of the analytical process-from data preparation to interpretation-are explicitly documented and easily shared (Kluyver et al., 2016; McKinney, 2013).

The versatility of Jupyter Notebooks allows communication researchers to engage in exploratory and confirmatory data analysis with considerable flexibility. Researchers can efficiently test multiple statistical models, apply diverse visualization techniques, and perform data manipulation tasks in an iterative manner. Additionally, the platform's support for multiple programming languages broadens its applicability and accommodates researchers with varying technical backgrounds, thereby lowering barriers to computational analysis within the field of communication studies (VanderPlas, 2016).

Furthermore, the immediate feedback generated through the execution of code cells fosters an iterative and reflexive research process. This feature enables researchers to refine analytical strategies in real time and make evidence-based methodological decisions. The integration of executable code with narrative markdown text facilitates the construction of coherent analytical narratives, thereby improving the clarity and accessibility of research outputs for both technical and non-technical audiences (Hunter, Cal, & San Luis, 2020).

As communication research increasingly relies on large-scale, complex, and digitally mediated datasets, the adoption of Python and Jupyter Notebooks constitutes a critical step toward modernizing research workflows. By enhancing efficiency, transparency, and reproducibility, these tools align closely with contemporary standards of open and computational science. Ultimately, embracing Python-based analytical environments contributes to the methodological advancement of communication research and supports its continued evolution in the digital age.

REFERENCES

Bogdanchikov, A., Zhaparov, M., & Suliyev, R. (2013). Python programming language and its application in scientific research. Procedia Computer Science, 22, 494–499. https://doi.org/10.1016/j.procs.2013.09.127

Craig, R. T. (1999). Communication theory as a field. Communication Theory, 9(2), 119–161.

Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and conducting mixed methods research* (3rd ed.). Sage

Eid, Mahmoud. (2011). Research Methods in Communication.

Downey, A. B., Elkner, J., & Meyers, C. (2008). How to think like a computer scientist: Learning with Python. Green Tea Press.Fabio Nelli. 2015. Python data analytics: Data analysis and science using PANDAs, Matplotlib and the Python Programming Language. Apress.

Few, S. (2012). *Show me the numbers: Designing tables and graphs to enlighten*. Analytics Press

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585(7825), 357–362.

Helen Dong (2021) A qualitative study of cleaning in Jupyter notebooks. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2021). Association for Computing Machinery, New York, NY, USA, 1663–1665. https://doi.org/10.1145/3468264.3473490

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

Hunter Glanz, Cal Poly, San Luis Obispo,(2020) CA.Using Jupyter to Boost Your Data Science Workflow. SAS Global Forum. Paper 4732

Jian, H., Sandnes, F. E., Huang, Y. P., & Huang, Y. M. (2011). Teaching beginners programming using Python: Experiences and evaluation. International Journal of Engineering Education, 27(6), 1378–1387.

Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. (2016). Jupyter Notebooks-a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas (pp. 87–90). IOS Press.

Knaflic, S. N. (2015). *Storytelling with data: A data visualization guide for business professionals*. Wiley.

Lewis, S. C., & Westlund, O. (2015). Big data and journalism: Epistemology, expertise, economics, and ethics. Digital Journalism, 3(3), 447–466.

Mahmod, O. (2011). Research methods in communication studies. University Press.

McQuail, D. (2010). McQuail's mass communication theory (6th ed.). Sage.

Mckinney(2013) Python for Data Analysis.

McKinney (2017) Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter 3rd Edition

McLeod, Douglas M. & Tichenor, Phillip J. (2003) "The logic of social and behavioral science." In Guido H. Stempel,David H. Weaver and G. Cleveland Wilhoit (Eds.), Mass communication research and theory (pp. 91-110). Boston: Allyn and Bacon

McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.

Neuendorf, K. A. (2017). The content analysis guidebook (2nd ed.). Sage.

Sabbith, A., Chandrasekaran, K., & Karthik, S. (2019). Python programming for scientific computing and data analysis. *International Journal of Computer Applications*, 182(44), 1–6.

Salganik, M. J. (2018). Bit by bit: Social research in the digital age. Princeton University Press.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Van Rossum, G., & Drake, F. L. (2009). Python 3 reference manual. CreateSpace.

VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.

Waskom, M. L. (2021). Seaborn: Statistical data visualization. Journal of Open Source Software, 6(60), 3021.