Data Science Department
Survey Methodology

Team_6

جامعة الإسكندرية
ALEXANDRIA UNIVERSITY
كلية الحاسبات وعلوم البيانات

SURVEY METHODOLOGY

Phase 4: Collect the Data

ABSTRACT

This report outlines the pre-analysis phase of survey data collected as part of a research project. The objective of this phase is to obtain a clean dataset that is free from careless responses, missing values and inconsistencies while being well-framed and easy to read.

Phase _ 4

Contents: -

Introduction

Phase Four: As part of the data collection process, it is essential to ensure that the data obtained is clean and free from any errors or inconsistencies. In this phase, we will be focusing on the pre-analysis of the collected data with the aim of obtaining a clean dataset that is wellframed, easy to read and understand. This will involve reviewing the survey responses to identify any careless answers, missing values or inconsistencies.

Another important aspect of data cleaning is identifying and dealing with missing values. Missing values can occur when a participant fails to answer a question or if there was an error in the data collection process. It is crucial to handle missing values appropriately to ensure that the analysis is accurate and unbiased.

Finally, the data set needs to be well-framed, which means that it is presented in a way that is easy to read and understand. Well-framed data helps to ensure that the insights gained from the analysis are clear and relevant to the research questions at hand.

1 - Cleaning Dataset

Introduction

Cleaning a dataset is an essential step in Our Survey. It involves identifying and correcting or removing incomplete, inaccurate, or irrelevant data that can negatively affect the accuracy and reliability of

Our conclusions or predictions. Cleaning The dataset involved a wide range of tasks such as handling missing values, removing duplicates, correcting syntax errors, standardizing formats, and converting data types.

The methodology for cleaning a dataset typically involves several steps. We Started By exploring the dataset to understand its structure, content, and quality. Then, we started applying various techniques to clean the data. For example, removing duplicates, fill in missing values, correct syntax errors, standardize formats, and convert data types. It's important to document each cleaning step and keep track of the changes made to the original dataset. Overall, the methodology for cleaning a dataset requires attention to detail, creativity, and domain knowledge. It's essential for ensuring that the data is accurate, reliable, and suitable for further analysis or modeling.

```
In [128]: # check for duplicates
          print(df.duplicated().sum())

          # drop duplicates
          df.drop_duplicates(inplace=True)

          0

In [129]: df = df.rename(columns={'What is your Student ID?': 'Student_ID'})
          df = df.set_index('Student_ID')
          df
```

Student_ID											
		Month	twice	Recommendations	year	notice	new idea				
2.000000e+00	2	Last Month	Once or twice	Friends Recommendations	I don't go to the Cinema	The age of adaline	Because it is an amazing film	Quality of Story	Comedy;Drama;Romance;Horror;Family	No	2
2.022146e+10	2	Last Month	Less than once	Friends Recommendations	I don't go to the Cinema	The swimmers	Real movie	Quality of Story	Comedy	I don't think	2
2.022000e+03	3	Last Month	Once or twice	Internet Sites	Once or Twice a year	interstellar / A walk to remember / silent voice	Because they have a variety of special feelingss	Quality of Story	Comedy;Drama;Romance;Sci-fi or Fantasy	Psychological /	2

250 rows × 63 columns

```
In [134]: df = df.dropna(axis=1)
          df
```

		Month	twice	Recommendations	year	notice				
2.000000e+00	2	Last Month	Once or twice	Friends Recommendations	I don't go to the Cinema	The age of adaline	Quality of Story	Comedy;Drama;Romance;Horror;Family	Yes	Internet Sites
2.022146e+10	2	Last Month	Less than once	Friends Recommendations	I don't go to the Cinema	The swimmers	Quality of Story	Comedy	Yes	Internet Sites
2.022000e+03	3	Last Month	Once or twice	Internet Sites	Once or Twice a year	interstellar / A walk to remember / silent voice	Quality of Story	Comedy;Drama;Romance;Sci-fi or Fantasy	Yes	Critical Reviews;Friends' Opinions (who alread...

250 rows × 35 columns

2 - Non-Carelessness Responds

Introduction

Non-carelessness responds refer to the way one responds to a situation or problem with an attitude of responsibility, accountability, and attentiveness. It involves taking time to assess the situation, gathering information, and carefully considering possible courses of action before making a decision or responding.

Methodology

To identify non-carelessness answered surveys, various techniques were employed. These techniques included the implementation of attention checks questions strategically placed within the survey to assess the respondent's attentiveness.

Technique Used in Our Survey

We designed attention check questions which include questions that ensure the respondent's attentiveness and the integrity of their responses.

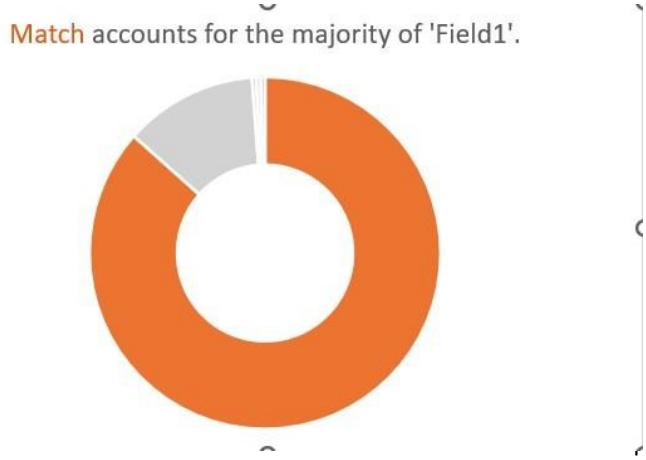
Our attention check question from the questionnaire is:

Most of the movies you wanted to watch was because of... *

- ◯ Trailers
- ◯ Social Media (Viral Videos)
- ◯ Internet Sites
- ◯ Friends Recommendations
- ◯ Other:

In Conclusion, we found that from the total number of respondents received, which is 250, there are 31 carelessness-answered surveys which means that there is a total number of non-carelessness responses of 219.

Field1	Count of Field1
Match	219
No Match	31
31	1
219	1
Comparison Between Col (C and Q)	1
(blank)	
Grand Total	**253**



```
In [ ]: data = df[df["comparison Between Col (C and Q)"].str.contains("No Match") == False]
        data
```

3 - No Missing Values

Firstly, We Used a Pre-Test Form to identify Any Needed Edits In The Questions, Which Appears in The First 70 Records Of The Respondents . Our Data Has No Other Missing Values Instead Of The Values Of The Pre-Test Form.

Secondly, we filled some of missing values by mean, mode, median.

4 - Well Framed Dataset

how the data was framed

We Used the Power of Python and Pandas Library to Make Sure That Our Data Is Well Framed and Easy to Understand. We Also Divided Some Columns Which Represented Checkbox Questions in Our Form to Get the Number of Times Each Box Was Selected. We also Renamed the Columns Which were Named by The Questions in The Form. so, We Tried to Make it Brief and Describes the Data.

Technique Used

Renaming Columns:

```
In [7]: new=df.rename(columns = {'When was the last time you watched a movie?':'Last_Watched',
                                 'How often do you watch movies (In a Month)?':'Frequent_Watch',
                                       'Most of the movies you wanted to watch was because of...':'Reason_of_Watch',
                                 'How often do you go to the Cinema?':'Frequent_Go_Cinema',
                                'What do you like most about your favorite movie?':'Factors',
                                "Do you usually research a new movie before watching it?":"Search_Before_Watch",
                                'Which form of Movie Advertising do you find most effective?':'Ads_effect',
                                 'How often does a movie disappoint your expectations?':'Usually_Disappoint',
                                'When going to the Cinema, you usually go..':'Going_Cinema',
                                'Do the Seats and Snacks of the Cinema usually have an effect on how much you enjoy a movie?'
                                 :'Seats and Snacks of the Cinema effect on  enjoying a movie?'}, inplace = True)

        # After renaming the columns
        print(df.columns)
```

```
Index(['day of week', 'Last_Watched', 'Frequent_Watch', 'Reason_of_Watch',
       'Frequent_Go_Cinema', 'What is your favorite movie?', 'Factors',
       'What's the genre of your favorite movie?', 'Search_Before_Watch',
       'If Yes, Which of the following sources do you use?', 'Ads_effect',
       'Usually_Disappoint',
       'When do you usually decide which movie you are going to see?',
       'Going_Cinema',
       'What do you usually do if you arrive to the Cinema and your movie has been sold out?',
       'Seats and Snacks of the Cinema effect on  enjoying a movie?',
       'What is your Age?', 'Please select your Gender.',
       'Please let us know if you have any comments or suggestions on how we can improve this survey.',
```

Dividing Checkbox (best day, best time):

```
In [75]: df[['best day(c)','best time(c)']] = df.Going_Cinema.str.split("-",expand=True)
         df
```

Out[75]:

hy ng ne ad of a?	setup watching home	usuall way to watch movies at home	favorite Streaming Platform	Age	Major	Gender	Critical Reviews	Friends' Opinions (who already watched it)	Internet Sites	TV Entertainment Programs	best day(c)	best time(c)
aN	NaN	NaN	NaN	19	Data Science	Female	0	0	0	0	Weekend	Evening
aN	NaN	NaN	NaN	20	Data Science	Male	0	1	0	0	Weekend	Evening
aN	NaN	NaN	NaN	20	Data Science	Male	0	0	1	0	Weekend	Evening
aN	NaN	NaN	NaN	20	Data Science	Male	0	0	1	0	Weekend	Evening
aN	NaN	NaN	NaN	20	Data Science	Female	0	0	0	0	Weekend	Evening
...	...	...	...	...	...	...	...	...	...	...	...	...
aN	NaN	NaN	NaN	20	Data Science	Female	0	0	0	0	Weekday	Evening
aN	NaN	NaN	NaN	20	Data Science	Female	0	0	0	1	Weekday	Evening
aN	NaN	NaN	NaN	20	Data Science	Female	0	0	1	0	Weekday	Afternoon
aN	NaN	NaN	NaN	19	Data Science	Female	0	0	0	0	Weekend	Afternoon

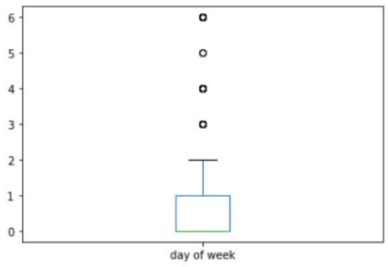
Dividing Checkbox (favorite genre):

```
In [79]: df['Action'] = df['fav_genre'].str.contains('Action').astype(int)
         df['Adventure'] = df['fav_genre'].str.contains('adventure').astype(int)
         df['Comedy'] = df['fav_genre'].str.contains('Comedy').astype(int)
         df['Romance'] = df['fav_genre'].str.contains('Romance').astype(int)
         df['Horror'] = df['fav_genre'].str.contains('horror').astype(int)
         df['Sci-fi or Fantasy'] = df['fav_genre'].str.contains('Sci-fi or Fantasy').astype(int)
         df['Family'] = df['fav_genre'].str.contains('Family').astype(int)
         df['Action'] = df['fav_genre'].str.contains('Action').astype(int)

         df
```

nder	Critical Reviews	Friends' Opinions (who already watched it)	Internet Sites	TV Entertainment Programs	best day(c)	best time(c)	fav_genre	Action	Adventure	Comedy	Romance	Horror	Sci-fi or Fantasy	Family
nale	0	0	0	0	Weekend	Evening	Action ; adventure ; Romance ;horror; drama	1	1	0	1	1	0	0
Male	0	1	0	0	Weekend	Evening	Comedy	0	0	1	0	0	0	0
Male	0	0	1	0	Weekend	Evening	Adventure	0	0	0	0	0	0	0
Male	0	0	1	0	Weekend	Evening	Action;Adventure;Comedy;Horror	1	0	1	0	0	0	0
nale	0	0	0	0	Weekend	Evening	Action;Adventure;Horror;Sci-fi or Fantasy	1	0	0	0	0	1	0

```
In [11]: df.boxplot(column =['day of week'], grid = False)
Out[11]: <AxesSubplot:>
```




```
In [12]: # Position of the Outlier
         import numpy as np
         #More than 2 is considered an outlier
         print(np.where(df['day of week']>2))

         (array([  0,   1,   2, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215,
                216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
                229, 230, 231, 232, 233, 234, 235, 249], dtype=int64),)
```

```
In [13]: # IQR
         Q1 = np.percentile(df['day of week'], 25,
         interpolation = 'midpoint')

         Q3 = np.percentile(df['day of week'], 75,
         interpolation = 'midpoint')
         IQR = Q3 - Q1
         IQR
Out[13]: 1.0

In [14]: # Above Upper bound
         upper=Q3+1.5*IQR
         upper_array=np.array(df['day of week']>=upper)
         print('Upper Bound:',upper)
         print(upper_array)

         #Below Lower bound
         lower=Q1-1.5*IQR
         lower_array=np.array(df['day of week']<=lower)
         print('Lower Bound',lower)
         print(lower_array)

         Upper Bound: 2.5
         [ True  True  True False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
          False False False False False False False False False False False False
```

```
In [ ]: # Split the DataFrame into two separate DataFrames (Cinema, Home)
        Cinema = df.loc[df['Watching_Place'] == 'In the Cinema']
        # Select rows that watching movies in the cinema
        Home = df.loc[df['Watching_Place'] != 'In the Cinema']
        # Select rows that watching movies at home

        Cinema.to_csv(r'C:\Users\N\Downloads\Cinema.csv', index=False)
        Home.to_csv(r'C:\Users\N\Downloads\Home.csv', index=False)
```

5 - Pilot Testing

Identifying Issues Before Full Implementation, we have already performed it on the first 70 responses to check if we refine question wording or instructions, to ensure that the survey is understandable and relevant to the target population and improve the quality and validity of the survey instrument.

6 - Validity and Reliability

Cronbach's alpha

To check reliability, we have used Cronbach's alpha (A measure of internal consistency reliability that assesses the degree to

internal consistency reliability that assesses the degree to which a set of survey questions are measuring the same construct. questions are measuring the same construct, It ranges from 0 to 1, with higher values indicating greater internal

consistency).

To achieve highest internal consistency reliability (Cronbach's alpha), we have used corr() function to get highest correlated features.

```
In [70]: pd.set_option('display.max_columns', None)
         survey_data.corr()
```

Out[70]:

	Action	Adventure	Comedy	Drama	Romance	Horror	Sci-fi or Fantasy	Musical	Family	Cast	Genre	Director	Studio	Reviews
Action	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Adventure	NaN	1.000000	-0.017951	0.013493	-0.020907	0.152647	0.413418	0.071261	-0.005582	0.190332	0.240156	0.017686	0.007616	0.129879
Comedy	NaN	-0.017951	1.000000	0.266977	0.183957	-0.075421	0.133969	0.162504	0.393446	0.118911	0.022784	0.021891	-0.030241	0.206500
Drama	NaN	0.013493	0.266977	1.000000	0.464268	-0.016211	0.102604	0.089087	0.207955	0.127521	0.102023	0.126759	-0.080297	0.137845
Romance	NaN	-0.020907	0.183957	0.464268	1.000000	0.012115	0.076350	0.122984	0.158298	0.166907	0.051761	0.002518	-0.103102	0.082255
Horror	NaN	0.152647	-0.075421	-0.016211	0.012115	1.000000	0.158578	0.032806	-0.133517	0.127592	0.076511	0.085586	0.157448	-0.003073
Sci-fi or Fantasy	NaN	0.413418	0.133969	0.102604	0.076350	0.158578	1.000000	0.129004	0.075233	0.102495	0.178154	0.045449	0.004978	0.128348
Musical	NaN	0.071261	0.162504	0.089087	0.122984	0.032806	0.129004	1.000000	0.330091	0.061668	-0.053846	0.078914	0.093913	0.071016
Family	NaN	-0.005582	0.393446	0.207955	0.158298	-0.133517	0.075233	0.330091	1.000000	0.120793	0.107695	0.023881	0.131099	0.123695
Cast	NaN	0.190332	0.118911	0.127521	0.166907	0.127592	0.102495	0.061668	0.120793	1.000000	0.374169	0.396997	0.226817	0.243248
Genre	NaN	0.240156	0.022784	0.102023	0.051761	0.076511	0.178154	-0.053846	0.107695	0.374169	1.000000	0.012822	0.026461	0.151730
Director	NaN	0.017686	0.021891	0.126759	0.002518	0.085586	0.045449	0.078914	0.023881	0.396997	0.012822	1.000000	0.536775	-0.023810
Studio	NaN	0.007616	-0.030241	-0.080297	-0.103102	0.157448	0.004978	0.093913	0.131099	0.226817	0.026461	0.536775	1.000000	0.086033
Reviews	NaN	0.129879	0.206500	0.137845	0.082255	-0.003073	0.128348	0.071016	0.123695	0.243248	0.151730	-0.023810	0.086033	1.000000
Quality of Story	NaN	0.195241	-0.009073	0.030776	0.094780	0.068128	0.151282	0.004284	0.001906	0.306945	0.323870	0.122754	0.111876	0.323163

Then, we implement Cronbach's alpha code as follow :

```
In [55]: import pingouin as pg

In [67]: # Select columns containing the survey questions
         survey_questions = survey_data[['Action', 'Adventure', 'Sci-fi or Fantasy', 'Romance', 'Drama']]

         # Calculate Cronbach's alpha using the alpha() function from the pingouin library
         alpha = pg.cronbach_alpha(survey_questions)

         # Print the value of Cronbach's alpha
         print("Cronbach's alpha:", alpha)

         Cronbach's alpha: (0.4323728235814256, array([0.322, 0.529]))
```

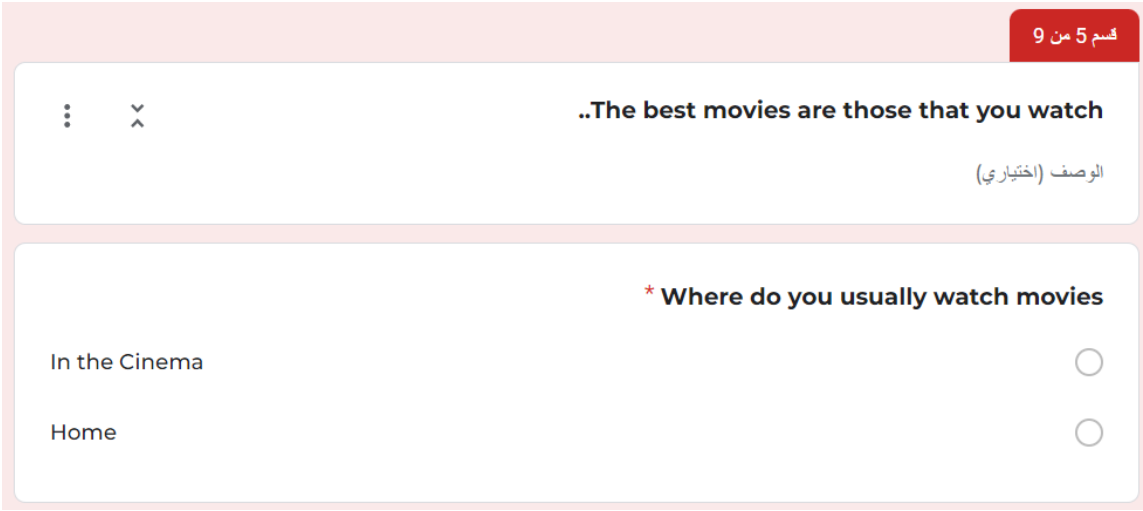
Conclusion : Highest value of alpha we have achieved is : 0.43 which is quite acceptable.

Test- Retest

"Same test – same sample – on different time"

To check validity of our responses, we have applied this technique on sample size = 30 and we compared their responses on different time for same person (same id) and we compared their responses on different time for same person (same id).

we consider our question to check validity



we **conclude** that only 2 responses of our 30-sample have answered different answers between each one he had answered it, so we have achieved validity

Conclusion And Recommendations

Overall, this report emphasizes the importance of cleaning data And Mentions the importance of this step in Our Survey.

``Thank you and hope we have made our report clear. ``