Notebook

# SENTIMENT ANALYSIS OF RESPONSES FROM BENEFICIARIES TO LIVELIHOOD SUBPROJECTS IN ZAMBIA[¶](#Xfa0bd3982aee6e165c37358e347ae931f94dcc2)

**Author:** Nathan Namatama  
**Institution:** Leibniz Institute of Ecological Urban and Regional Development and Technical University of Dresden  
**Year:** 2025  
**Related Publication:** The effects and impacts of livelihood activities and unplanned human settlement growth on greenspace and wetland landscapes in Zambia: *A case of the three areas of the Pilot Programme for Climate Resilience (PPCR)*

## Purpose of the Analysis[¶](#Purpose-of-the-Analysis)

The analysis is conducted on the data collection that was conducted in Zambia from 23rd July 2024 to 22nd September 2024. The primary data was collected using ArcGIS Survey123 application in an offline mode mostly in certain areas that did not have access to internet while those that ahd internet an online mode was used. The respondents were interviewed using a semi structured question and the responses were recorede in the application as they were responding to the questions.

The analysis is done in the framework of *systems thinking* of looking at *deep leverage points* in the ***governance for transformation*** of Social Ecologocal Systems so as to attain sustainable transfomation. The analysis is done in both qualitative and quantitatives (descriptions) to produce graphs and tables that are visualised within the jupyterlab notebook.

## Table of Contents[¶](#Table-of-Contents)

1. [Importing Libraries](#X74e63f2495d7c5ffb07c54b3aaf45f5cdb47b8a)
2. [Reading the Excel Table](#X2fb98c00f948b87b9b5cf7fde3b043095f45cbd)
3. [Missing values](#X15d2a8909405fc3a2a3002f109055d11b9f0aff)
4. [Deleting Columnns](#Xa8eaff489a091986ef86ff33b659ed2aa57f8bb)
5. [Renaming Columns](#Xdaaa7d4e2d53f55b7de282473ce9891898905f0)
6. [Selection of Likert Scale Columns](#X9ba728fa3cbe8ccaef3988d67a4dd7bd7138533)
7. [The Number of Responses](#Xea081d9a982cae653adfa1a90dd733f8bb4eecd)
8. [Defined or Responses with Choices](#Xa861aaa20dc8dcab36f9c86995e9cde275267ac)
9. [Description Statistics](#Xb63a2db521ab2f47d37d53ee78c252f279d851a)
10. [Reasons](#X3a6864d3553c0e5b6d0a7841152ba99470d60b9)
11. [Specific Variables](#X41e2ce28de3d28ad12805bb21bb04dcc259b2a5)
12. [Converting the Notebook](#X892bfffef340f7a4b7a78f2a4187307a68e2b40)

## 1. Importing Libraries[¶](#X74e63f2495d7c5ffb07c54b3aaf45f5cdb47b8a)

The liberarries that are needed for conducting the analysis are installed and downloaded. They are as follows:

* ***NLTK:*** For reading text and understanding it in a way that a human can do it
* ***Re:*** For text manipulation and pattern matching
* ***Pandas:*** For converting tables into a format understandable by the computer as well as visualisation
* ***Numpy:*** For conducting statistical culculations
* ***Matplotlib:*** For visualisation
* ***Seabron:*** For visualisation
* ***Io:*** For reading/writing binary and text data efficiently
* ***Csv:*** For reading the csv files
* ***Unicodeddata:*** For interacting with and analyzing Unicode characters
* ***String:*** For language analysis, user input, or file processing
* ***Plotly:*** For visualisation
* ***Plot\_Likert:*** For visualisation of likert scales
* ***%matplotlib line:*** For visualisation within the jupyterlab notebook
* ***Nbconvert:*** For converting to HTML format
* ***WordCloud:*** For creating a word cloud
* ***Bigrams:*** For making words into pairs
* ***Trigrams:*** For making three words sequencies
* ***GridSpec*** For ploting a graph in a specific grid

In [212]:

import nltk  
import re  
from collections import Counter  
from nltk.probability import FreqDist  
from nltk.corpus import stopwords  
from nltk.stem import WordNetLemmatizer  
from nltk.tokenize import sent\_tokenize  
from nltk.tokenize import word\_tokenize  
from nltk import sent\_tokenize, word\_tokenize, pos\_tag  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from matplotlib.gridspec import GridSpec  
import seaborn as sns  
import io  
from io import StringIO  
import csv  
import unicodedata  
import string  
import plotly  
import plotly.express as px  
import plot\_likert  
from sklearn.model\_selection import train\_test\_split  
%matplotlib inline  
import nbconvert  
from nbconvert import HTMLExporter  
import nbformat  
from wordcloud import WordCloud  
from nltk import bigrams  
from nltk import trigrams  
import subprocess  
import shutil  
import pypandoc

In [213]:

nltk.download('punkt\_tab')  
nltk.download('punkt')  
nltk.download('stopwords')  
nltk.download('wordnet')

[nltk\_data] Downloading package punkt\_tab to  
[nltk\_data] C:\Users\nazin\AppData\Roaming\nltk\_data...  
[nltk\_data] Package punkt\_tab is already up-to-date!  
[nltk\_data] Downloading package punkt to  
[nltk\_data] C:\Users\nazin\AppData\Roaming\nltk\_data...  
[nltk\_data] Package punkt is already up-to-date!  
[nltk\_data] Downloading package stopwords to  
[nltk\_data] C:\Users\nazin\AppData\Roaming\nltk\_data...  
[nltk\_data] Package stopwords is already up-to-date!  
[nltk\_data] Downloading package wordnet to  
[nltk\_data] C:\Users\nazin\AppData\Roaming\nltk\_data...  
[nltk\_data] Package wordnet is already up-to-date!

Out[213]:

True

## 2. Reading the Excel Table[¶](#X2fb98c00f948b87b9b5cf7fde3b043095f45cbd)

The csv file is converted to a padas dataframe  
The dataframe table is displyed with **ALL** columns and rows with cutting any

In [214]:

df = pd.read\_csv(r"D:\DataAnalysis\Social\_Survey\_Questionnaire\_for\_Beneficiaries\_0.csv")  
pd.set\_option('display.max\_colwidth', None)  
pd.set\_option('display.max\_rows', None)   
pd.set\_option('display.max\_columns', None)

In [215]:

type(df)

Out[215]:

pandas.core.frame.DataFrame

In [216]:

#df

In [217]:

df.shape

Out[217]:

(150, 120)

## 3. Missing values[¶](#X15d2a8909405fc3a2a3002f109055d11b9f0aff)

The Pandas Dataframe is checked for mising values

In [218]:

#df.isnull().sum()

### 3.1 Dropping all the Missing Values[¶](#Xfa0b0327b7a9873d13e6700446611ccbf2f083a)

In [219]:

#df.isnull().sum().sort\_values(ascending=False)

In [220]:

#df.dropna(inplace=True)

## 4. Deleting Columnns[¶](#Xa8eaff489a091986ef86ff33b659ed2aa57f8bb)

There are 120 columns and 85 columns were removed from the pandas dataframe so as to focus on specific columns that have questtions in regards to livelihood and landscape transfromation

In [221]:

df1=df.drop(df.columns[[1,2,3,4,5,6,7,8,9,10,11,12,13,14,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,116,117,118,119]], axis = 1)

In [222]:

#df1

## 5. Renaming Columns[¶](#Xdaaa7d4e2d53f55b7de282473ce9891898905f0)

The headings of the columns were renamed for easy inclusion in the code

In [223]:

df1.columns

Out[223]:

Index(['ObjectID', '6. Ward Name', '7. Sub Project Name',  
 '8. Type of Livelihood', '9. Size (Lima)', '12. Name of main project',  
 '13. Do you represent other beneficiaries?',  
 '14. How many beneficiaries do you represent?',  
 '45. Are there cultural practices that hinder the sustainable management of Forests, Wetlands, National Game Parks and Biodiversity?',  
 '46. What reasons can you give for your answer above?',  
 '47. Do you think some cultural practices can be changed?',  
 '48. Do you consider cultural aspects when formulating the livelihood projects?',  
 '49. What reasons can you give for your answer above?',  
 '50. What is the main purpose of landscapes (Forests, Water Bodies, Wetlandsâ¦ etc.) in your livelihood?',  
 '51. Do you think there is need to measure indicators when managing landscapes?',  
 '52. How is your connection to nature like?',  
 '53. What reasons can you give for your answer above?',  
 '54. How long have you worked on this livelihood project?',  
 '55. Does your livelihood depend on the natural resources for a living?',  
 '56. Do you consider changing your livelihood strategy in future?',  
 '57. Do you think it is easier to change your livelihood practices?',  
 '58. Are you able to give reasons for your answer above in your ability to change your livelihood practices?',  
 '59. Have the ecosystem services reduced from the inception of the project in your ward?',  
 '60. Has the deforestation increased in the ward?',  
 '61. Do you think protected areas are a hindrance to your livelihoods?',  
 '62. Are there new livelihood projects that you think of that have never been implemented?',  
 '63. Do you think the livelihood subprojects are contributing to the sustainability of landscapes?',  
 '64. Which livelihood is a major contributor to landscape transformation?',  
 '65. What reasons can you give for your answer above?',  
 '66. Which type of landscape do you depend on much for a livelihood?',  
 '67. What reasons can you give for your answer above?', 'Specify:.4',  
 'Specify:.5', 'Specify:.6', 'Specify:.7'],  
 dtype='object')

In [224]:

df2=df1.rename(columns= {'1. Do you agree to take part in the above study?': 'Part\_study',  
 '2. Do you know that your participation is voluntary and you are free to withdraw anytime?':'Participation\_Voluntary',   
 '3. Do you give permission to the data that emerges to be used by the researchers only in an anonymised form?': 'Anonymised\_Form',   
 '5. Date': 'Date',   
 '6. Ward Name': 'Ward\_Name',   
 '7. Sub Project Name': 'Sub\_Project\_name',   
 '8. Type of Livelihood': 'Livelihood',   
 '9. Size (Lima)': 'Size',   
 '12. Name of main project': 'Name\_Main\_Project',  
 '13. Do you represent other beneficiaries?': 'Representing\_Others',  
 '14. How many beneficiaries do you represent?': 'Number\_Beneficiaries',  
 '45. Are there cultural practices that hinder the sustainable management of Forests, Wetlands, National Game Parks and Biodiversity?': 'Cultural\_Practices\_Hinder',   
 '46. What reasons can you give for your answer above?': 'Cultural\_Practices\_Hinder\_Reason',   
 '47. Do you think some cultural practices can be changed?': 'Cultural\_Practices\_Changed',   
 '48. Do you consider cultural aspects when formulating the livelihood projects?': 'Cultural\_Aspects\_Considered',  
 '49. What reasons can you give for your answer above?': 'Cultural\_Aspects\_Considered\_Reasons',   
 '50. What is the main purpose of landscapes (Forests, Water Bodies, Wetlandsâ¦ etc.) in your livelihood?': 'Purpose\_Landscape',   
 '51. Do you think there is need to measure indicators when managing landscapes?': 'Measure\_Indicators',   
 '52. How is your connection to nature like?': 'Connection\_Nature',   
 '53. What reasons can you give for your answer above?': 'Connection\_Nature\_Reasons',  
 '54. How long have you worked on this livelihood project?': 'Range\_Years',  
 '55. Does your livelihood depend on the natural resources for a living?': 'Livilihood\_Depenedent',  
 '56. Do you consider changing your livelihood strategy in future?': 'Change\_Livelihood',  
 '57. Do you think it is easier to change your livelihood practices?': 'Change\_Livelihood\_Easy',  
 '58. Are you able to give reasons for your answer above in your ability to change your livelihood practices?': 'Change\_Livelihood\_Easy\_Reasons',  
 '59. Have the ecosystem services reduced from the inception of the project in your ward?': 'Ecosystem\_Services\_Reduced',  
 '60. Has the deforestation increased in the ward?': 'Deforestaion\_Increased',  
 '61. Do you think protected areas are a hindrance to your livelihoods?': 'Protected\_Areas\_Hinderarnce\_Livelihood',  
 '62. Are there new livelihood projects that you think of that have never been implemented?': 'New\_Livelihood\_Projects',  
 '63. Do you think the livelihood subprojects are contributing to the sustainability of landscapes?': 'Subprojects\_Sustainability\_Contribution',  
 '64. Which livelihood is a major contributor to landscape transformation?': 'Contributor\_Landscape\_Transformation',  
 '65. What reasons can you give for your answer above?': 'Contributor\_Landscape\_Transformation\_Reasons',  
 '66. Which type of landscape do you depend on much for a livelihood?': 'Landscape\_Depended\_Livelihood',  
 '67. What reasons can you give for your answer above?': 'Landscape\_Depeneded\_Livelihood\_Reasons',  
 'Specify:.4': 'Purpose\_Landscape\_Specific',  
 'Specify:.5': 'Connection\_Nature\_Specific',  
 'Specify:.6': 'Contributor\_Landscape\_Transformation\_Specific',  
 'Specify:.7': 'Landscape\_Depended\_Livelihood\_Specific'})

In [225]:

#df2

## 6. Selection of Likert Scale Columns[¶](#X9ba728fa3cbe8ccaef3988d67a4dd7bd7138533)

The columns that had likert scale responses were group in a single dataframe and they are 12 in number the columns were looking at the following questions as numbered in the questionaire:

* '45. Are there cultural practices that hinder the sustainable management of Forests, Wetlands, National Game Parks and Biodiversity?
* '47. Do you think some cultural practices can be changed?',
* '48. Do you consider cultural aspects when formulating the livelihood projects?',
* '51. Do you think there is need to measure indicators when managing landscapes?',
* '55. Does your livelihood depend on the natural resources for a living?',
* '56. Do you consider changing your livelihood strategy in future?',
* '57. Do you think it is easier to change your livelihood practices?',
* '59. Have the ecosystem services reduced from the inception of the project in your ward?',
* '60. Has the deforestation increased in the ward?',
* '61. Do you think protected areas are a hindrance to your livelihoods?',
* '62. Are there new livelihood projects that you think of that have never been implemented?',
* '63. Do you think the livelihood subprojects are contributing to the sustainability of landscapes?'

In [226]:

df3=df2.drop(df2.columns[[0,1,2,3,4,5,6,7,9,12,13,15,16,17,21,27,28,29,30,31,32,33,34]], axis = 1)

In [227]:

#df3

## 7. The Number of Responses[¶](#Xea081d9a982cae653adfa1a90dd733f8bb4eecd)

The number of responses were counted for each column that had a likert scale and the NaN indicates no response

In [228]:

all\_counts = df3.apply(pd.Series.value\_counts, dropna=False)

In [229]:

all\_counts

Out[229]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cultural\_Practices\_Hinder | Cultural\_Practices\_Changed | Cultural\_Aspects\_Considered | Measure\_Indicators | Livilihood\_Depenedent | Change\_Livelihood | Change\_Livelihood\_Easy | Ecosystem\_Services\_Reduced | Deforestaion\_Increased | Protected\_Areas\_Hinderarnce\_Livelihood | New\_Livelihood\_Projects | Subprojects\_Sustainability\_Contribution |
| Agree\_Likert | 20 | 29 | 19 | 28 | 35 | 39 | 46 | 39 | 16 | 14.0 | 58 | 37 |
| Disagree\_Likert | 22 | 22 | 20 | 4 | 25 | 37 | 24 | 13 | 32 | 19.0 | 15 | 4 |
| Strongly\_Agree\_Likert | 21 | 29 | 22 | 81 | 44 | 38 | 49 | 72 | 72 | 15.0 | 36 | 81 |
| Strongly\_Disagree\_Likert | 70 | 41 | 66 | 22 | 32 | 20 | 18 | 16 | 17 | 83.0 | 19 | 15 |
| Undecided\_Likert | 4 | 9 | 7 | 7 | 4 | 5 | 4 | 3 | 1 | NaN | 10 | 3 |
| NaN | 13 | 20 | 16 | 8 | 10 | 11 | 9 | 7 | 12 | 19.0 | 12 | 10 |

### 7.1 Transposing the Dataframe Table[¶](#Xf3320a41859681c7ed5f5eb96f1119bd6b7cc1b)

The columns and the rows were interchanged so that they can be easily presented on the graph

In [230]:

all\_counts1 = all\_counts.head().T

In [231]:

all\_counts1

Out[231]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree\_Likert | Disagree\_Likert | Strongly\_Agree\_Likert | Strongly\_Disagree\_Likert | Undecided\_Likert |
| Cultural\_Practices\_Hinder | 20.0 | 22.0 | 21.0 | 70.0 | 4.0 |
| Cultural\_Practices\_Changed | 29.0 | 22.0 | 29.0 | 41.0 | 9.0 |
| Cultural\_Aspects\_Considered | 19.0 | 20.0 | 22.0 | 66.0 | 7.0 |
| Measure\_Indicators | 28.0 | 4.0 | 81.0 | 22.0 | 7.0 |
| Livilihood\_Depenedent | 35.0 | 25.0 | 44.0 | 32.0 | 4.0 |
| Change\_Livelihood | 39.0 | 37.0 | 38.0 | 20.0 | 5.0 |
| Change\_Livelihood\_Easy | 46.0 | 24.0 | 49.0 | 18.0 | 4.0 |
| Ecosystem\_Services\_Reduced | 39.0 | 13.0 | 72.0 | 16.0 | 3.0 |
| Deforestaion\_Increased | 16.0 | 32.0 | 72.0 | 17.0 | 1.0 |
| Protected\_Areas\_Hinderarnce\_Livelihood | 14.0 | 19.0 | 15.0 | 83.0 | NaN |
| New\_Livelihood\_Projects | 58.0 | 15.0 | 36.0 | 19.0 | 10.0 |
| Subprojects\_Sustainability\_Contribution | 37.0 | 4.0 | 81.0 | 15.0 | 3.0 |

### 7.2 Changing the Order of Columns[¶](#X03fb2ec53eeccfc21fe5ddab8ee3951e7e1dd35)

The order of columns was changed so that they can be easily analysed

In [232]:

all\_counts2 = all\_counts1.iloc[:, [3, 1, 4, 0, 2]]

In [233]:

all\_counts2

Out[233]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly\_Disagree\_Likert | Disagree\_Likert | Undecided\_Likert | Agree\_Likert | Strongly\_Agree\_Likert |
| Cultural\_Practices\_Hinder | 70.0 | 22.0 | 4.0 | 20.0 | 21.0 |
| Cultural\_Practices\_Changed | 41.0 | 22.0 | 9.0 | 29.0 | 29.0 |
| Cultural\_Aspects\_Considered | 66.0 | 20.0 | 7.0 | 19.0 | 22.0 |
| Measure\_Indicators | 22.0 | 4.0 | 7.0 | 28.0 | 81.0 |
| Livilihood\_Depenedent | 32.0 | 25.0 | 4.0 | 35.0 | 44.0 |
| Change\_Livelihood | 20.0 | 37.0 | 5.0 | 39.0 | 38.0 |
| Change\_Livelihood\_Easy | 18.0 | 24.0 | 4.0 | 46.0 | 49.0 |
| Ecosystem\_Services\_Reduced | 16.0 | 13.0 | 3.0 | 39.0 | 72.0 |
| Deforestaion\_Increased | 17.0 | 32.0 | 1.0 | 16.0 | 72.0 |
| Protected\_Areas\_Hinderarnce\_Livelihood | 83.0 | 19.0 | NaN | 14.0 | 15.0 |
| New\_Livelihood\_Projects | 19.0 | 15.0 | 10.0 | 58.0 | 36.0 |
| Subprojects\_Sustainability\_Contribution | 15.0 | 4.0 | 3.0 | 37.0 | 81.0 |

### 7.3 Visualising the Results[¶](#X8272e72cb2e380535fd41b1f7a1515f0b143676)

The results were visualised in form of number of responses

In [234]:

%matplotlib inline  
plot\_likert.plot\_counts(all\_counts2, plot\_likert.scales.agree, plot\_percentage=False, bar\_labels=True, bar\_labels\_color="snow", colors=plot\_likert.colors.default\_with\_darker\_neutral)  
plt.title("Figure 1: The Total Number of Responses to Variables on Sustainable Transformation", fontsize=14)  
plt.show()

C:\Users\nazin\AppData\Local\anaconda3\envs\NLTK\_Py\_3\_12\Lib\site-packages\plot\_likert\plot\_likert.py:101: FutureWarning: parameter `plot\_percentage` for `plot\_likert.likert\_counts` is deprecated, set it to None and use `compute\_percentages` instead  
 warn(

![No description has been provided for this image](data:image/png;base64;base64,)

### 7.4 Converting the Responses to Percentages[¶](#Xefa40285bfb0f47da390debce0d2d32a9e002d2)

The responses were converted to percentages

In [235]:

all\_counts3 = df3.apply(lambda col: col.value\_counts(normalize=True, dropna=False).round(2))

In [236]:

all\_counts3

Out[236]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cultural\_Practices\_Hinder | Cultural\_Practices\_Changed | Cultural\_Aspects\_Considered | Measure\_Indicators | Livilihood\_Depenedent | Change\_Livelihood | Change\_Livelihood\_Easy | Ecosystem\_Services\_Reduced | Deforestaion\_Increased | Protected\_Areas\_Hinderarnce\_Livelihood | New\_Livelihood\_Projects | Subprojects\_Sustainability\_Contribution |
| Agree\_Likert | 0.13 | 0.19 | 0.13 | 0.19 | 0.23 | 0.26 | 0.31 | 0.26 | 0.11 | 0.09 | 0.39 | 0.25 |
| Disagree\_Likert | 0.15 | 0.15 | 0.13 | 0.03 | 0.17 | 0.25 | 0.16 | 0.09 | 0.21 | 0.13 | 0.10 | 0.03 |
| Strongly\_Agree\_Likert | 0.14 | 0.19 | 0.15 | 0.54 | 0.29 | 0.25 | 0.33 | 0.48 | 0.48 | 0.10 | 0.24 | 0.54 |
| Strongly\_Disagree\_Likert | 0.47 | 0.27 | 0.44 | 0.15 | 0.21 | 0.13 | 0.12 | 0.11 | 0.11 | 0.55 | 0.13 | 0.10 |
| Undecided\_Likert | 0.03 | 0.06 | 0.05 | 0.05 | 0.03 | 0.03 | 0.03 | 0.02 | 0.01 | NaN | 0.07 | 0.02 |
| NaN | 0.09 | 0.13 | 0.11 | 0.05 | 0.07 | 0.07 | 0.06 | 0.05 | 0.08 | 0.13 | 0.08 | 0.07 |

#### 7.4.1 Transposing the Dataframe Table[¶](#X6d7ad4a1251ea2c07023f86e2bee92ee0bcacce)

Interchanging the rows and columns

In [237]:

all\_counts4 = all\_counts3.head().T

In [238]:

all\_counts4

Out[238]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree\_Likert | Disagree\_Likert | Strongly\_Agree\_Likert | Strongly\_Disagree\_Likert | Undecided\_Likert |
| Cultural\_Practices\_Hinder | 0.13 | 0.15 | 0.14 | 0.47 | 0.03 |
| Cultural\_Practices\_Changed | 0.19 | 0.15 | 0.19 | 0.27 | 0.06 |
| Cultural\_Aspects\_Considered | 0.13 | 0.13 | 0.15 | 0.44 | 0.05 |
| Measure\_Indicators | 0.19 | 0.03 | 0.54 | 0.15 | 0.05 |
| Livilihood\_Depenedent | 0.23 | 0.17 | 0.29 | 0.21 | 0.03 |
| Change\_Livelihood | 0.26 | 0.25 | 0.25 | 0.13 | 0.03 |
| Change\_Livelihood\_Easy | 0.31 | 0.16 | 0.33 | 0.12 | 0.03 |
| Ecosystem\_Services\_Reduced | 0.26 | 0.09 | 0.48 | 0.11 | 0.02 |
| Deforestaion\_Increased | 0.11 | 0.21 | 0.48 | 0.11 | 0.01 |
| Protected\_Areas\_Hinderarnce\_Livelihood | 0.09 | 0.13 | 0.10 | 0.55 | NaN |
| New\_Livelihood\_Projects | 0.39 | 0.10 | 0.24 | 0.13 | 0.07 |
| Subprojects\_Sustainability\_Contribution | 0.25 | 0.03 | 0.54 | 0.10 | 0.02 |

#### 7.4.2 Changing the Order of Columns[¶](#X9c1ccf24773fca8eac31dd63b12902dd1db819e)

The order of columns was changed

In [239]:

all\_counts5 = all\_counts4.iloc[:, [3, 1, 4, 0, 2]]

In [240]:

all\_counts5

Out[240]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly\_Disagree\_Likert | Disagree\_Likert | Undecided\_Likert | Agree\_Likert | Strongly\_Agree\_Likert |
| Cultural\_Practices\_Hinder | 0.47 | 0.15 | 0.03 | 0.13 | 0.14 |
| Cultural\_Practices\_Changed | 0.27 | 0.15 | 0.06 | 0.19 | 0.19 |
| Cultural\_Aspects\_Considered | 0.44 | 0.13 | 0.05 | 0.13 | 0.15 |
| Measure\_Indicators | 0.15 | 0.03 | 0.05 | 0.19 | 0.54 |
| Livilihood\_Depenedent | 0.21 | 0.17 | 0.03 | 0.23 | 0.29 |
| Change\_Livelihood | 0.13 | 0.25 | 0.03 | 0.26 | 0.25 |
| Change\_Livelihood\_Easy | 0.12 | 0.16 | 0.03 | 0.31 | 0.33 |
| Ecosystem\_Services\_Reduced | 0.11 | 0.09 | 0.02 | 0.26 | 0.48 |
| Deforestaion\_Increased | 0.11 | 0.21 | 0.01 | 0.11 | 0.48 |
| Protected\_Areas\_Hinderarnce\_Livelihood | 0.55 | 0.13 | NaN | 0.09 | 0.10 |
| New\_Livelihood\_Projects | 0.13 | 0.10 | 0.07 | 0.39 | 0.24 |
| Subprojects\_Sustainability\_Contribution | 0.10 | 0.03 | 0.02 | 0.25 | 0.54 |

#### 7.4.3 Visualising the Results[¶](#Xcc18225e0433827156db47968ceaf2be67777db)

The results are visualised as pecentages

In [241]:

%matplotlib inline  
plot\_likert.plot\_counts(all\_counts5, plot\_likert.scales.agree, plot\_percentage=True, figsize=(16, 7), bar\_labels=True, bar\_labels\_color="snow", colors=plot\_likert.colors.default\_with\_darker\_neutral)  
plt.title("Figure 2: The Percentage Number of Responses to Variables on Sustainable Transformation", fontsize=18)  
plt.savefig("Likertscale.jpg")  
plt.savefig("Likertscale1.png", dpi=300)  
plt.show()

C:\Users\nazin\AppData\Local\anaconda3\envs\NLTK\_Py\_3\_12\Lib\site-packages\plot\_likert\plot\_likert.py:101: FutureWarning: parameter `plot\_percentage` for `plot\_likert.likert\_counts` is deprecated, set it to None and use `compute\_percentages` instead  
 warn(

![No description has been provided for this image](data:image/png;base64;base64,)

[[Back to top](#Purpose-of-the-Analysis)]

## 8. Defined or Responses with Choices[¶](#Xa861aaa20dc8dcab36f9c86995e9cde275267ac)

The responses that had choices other than those with a likert scale were grouped in a pandas dataframe looking the following question:

1. '50. What is the main purpose of landscapes (Forests, Water Bodies, Wetlandsâ€¦ etc.) in your livelihood?': 'Purpose\_Landscape',
2. '52. How is your connection to nature like?': 'Connection\_Nature',
3. '54. How long have you worked on this livelihood project?': 'Range\_Years',
4. '64. Which livelihood is a major contributor to landscape transformation?': 'Contributor\_Landscape\_Transformation',
5. '66. Which type of landscape do you depend on much for a livelihood?': 'Landscape\_Depended\_Livelihood'

In [242]:

df4=df2.drop(df2.columns[[0,1,2,3,4,5,6,7,8,9,10,11,12,14,16,18,19,20,21,22,23,24,25,26,28,30,31,32,33,34]], axis = 1)

In [243]:

#df4

### 8.1 The Purpose of Landscape to People[¶](#Xa5b5976d1823f8213d0515898b3a942e013cc64)

In [244]:

PL = df4['Purpose\_Landscape'].value\_counts(dropna=False)  
PL\_P = (df4['Purpose\_Landscape'].value\_counts(normalize=True, dropna=False).round(2))

In [245]:

PL\_P

Out[245]:

Purpose\_Landscape  
Nature\_Protection\_Purpose 0.62  
NaN 0.22  
Source\_Income\_Purpose 0.11  
Nature\_Protection\_Purpose,Other\_Purpose 0.01  
Nature\_Protection\_Purpose,Ancestral\_Shrines\_Purpose 0.01  
Source\_Income\_Purpose,Nature\_Protection\_Purpose 0.01  
Nature\_Protection\_Purpose,Source\_Income\_Purpose 0.01  
Source\_Income\_Purpose,Other\_Purpose 0.01  
No\_Idea\_Purpose 0.01  
Other\_Purpose 0.01  
Name: proportion, dtype: float64

In [246]:

df\_PL = pd.DataFrame(PL)

In [247]:

df\_PL

Out[247]:

|  |  |
| --- | --- |
|  | count |
| Purpose\_Landscape |  |
| Nature\_Protection\_Purpose | 93 |
| NaN | 33 |
| Source\_Income\_Purpose | 16 |
| Nature\_Protection\_Purpose,Other\_Purpose | 2 |
| Nature\_Protection\_Purpose,Ancestral\_Shrines\_Purpose | 1 |
| Source\_Income\_Purpose,Nature\_Protection\_Purpose | 1 |
| Nature\_Protection\_Purpose,Source\_Income\_Purpose | 1 |
| Source\_Income\_Purpose,Other\_Purpose | 1 |
| No\_Idea\_Purpose | 1 |
| Other\_Purpose | 1 |

In [248]:

ax = sns.countplot(df4["Purpose\_Landscape"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 3: Number of Responses on the Purpose of the Landscape", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 8.2 The Connection of Nature to People[¶](#X61ec01fc450542671236ec1aab383b2e6d8612c)

In [249]:

CN = df4['Connection\_Nature'].value\_counts(dropna=False)  
CN\_P = (df4['Connection\_Nature'].value\_counts(normalize=True, dropna=False).round(2))

In [250]:

CN\_P

Out[250]:

Connection\_Nature  
Material\_Connection 0.82  
NaN 0.09  
Other\_Connection 0.04  
Experiential\_Connection 0.01  
Philosophical\_Connection,Psychological\_Connection 0.01  
Psychological\_Connection,Material\_Connection 0.01  
Psychological\_Connection 0.01  
Philosophical\_Connection 0.01  
Experiential\_Connection,Material\_Connection 0.01  
Name: proportion, dtype: float64

In [251]:

df\_CN = pd.DataFrame(CN)

In [252]:

df\_CN

Out[252]:

|  |  |
| --- | --- |
|  | count |
| Connection\_Nature |  |
| Material\_Connection | 123 |
| NaN | 13 |
| Other\_Connection | 6 |
| Experiential\_Connection | 2 |
| Philosophical\_Connection,Psychological\_Connection | 2 |
| Psychological\_Connection,Material\_Connection | 1 |
| Psychological\_Connection | 1 |
| Philosophical\_Connection | 1 |
| Experiential\_Connection,Material\_Connection | 1 |

In [253]:

ax = sns.countplot(df4["Connection\_Nature"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 4: The Number of Responses to Connection to Nature", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [254]:

agreement\_levels = ["Material\_Connection", "Other\_Connection"]  
CN\_R = df2[df2["Connection\_Nature"].isin(agreement\_levels)]  
CN\_R1 = CN\_R.drop(CN\_R.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CN\_R1grouped = CN\_R1.groupby('Name\_Main\_Project')['Connection\_Nature']  
#CN\_R1

In [255]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CN\_R1.iterrows():  
 CN\_R1\_filter\_sentence = []  
 CN\_R1\_sentence = row["Connection\_Nature\_Reasons"]  
 if pd.isnull(CN\_R1\_sentence):  
 continue  
 CN\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CN\_R1\_sentence)  
 CN\_R1\_words = nltk.word\_tokenize(CN\_R1\_sentence\_cleaned)  
 CN\_R1\_words = [lemmatizer.lemmatize(w) for w in CN\_R1\_words if w.lower() not in stop\_words]  
 CN\_R1\_filter\_sentence.extend(CN\_R1\_words)  
 print(CN\_R1\_filter\_sentence)

['use', 'income']  
['source', 'income', 'future', 'generation', 'see']  
['Source', 'income']  
['source', 'income']  
['Thats', 'get', 'income', 'well', 'example', 'cattle', 'use', 'cultivating']  
['game', 'park', 'help', 'u', 'source', 'income', 'form', 'meat', 'animal', 'tree', 'rain', 'come', 'protected']  
['need', 'keep', 'fish', 'instance', 'fish', 'pond', 'get', 'extinct']  
['one', 'look', 'daily', 'basis']  
['source', 'livelihood']  
['need', 'protect', 'nature', 'instance', 'cutting', 'tree', 'along', 'river', 'lead', 'drying', 'destruction', 'animal', 'biodiversity', 'river']  
['answer']  
['purpose', 'future', 'meet', 'need']  
['source', 'income']  
['tree', 'help', 'u', 'bring', 'rainfall']  
['Source', 'income']  
['harvest', 'right', 'time', 'prevent', 'management', 'natural', 'resource', 'properly']  
['taking', 'care']  
['Source', 'income']  
['Source', 'income']  
['bring', 'development', 'tourism']  
['income']  
['natural', 'resource', 'like', 'tree', 'give', 'shade', 'medicine', 'also', 'get', 'fresh', 'air']  
['Like', 'water', 'source', 'life', 'tree', 'source', 'fresh', 'water']  
['source', 'income']  
['source', 'income']  
['ZAWA', 'Officers', 'one', 'connected', 'look']  
['source', 'income', 'tourism']  
['answer']  
['help', 'u', 'source', 'food', 'well', 'water']  
['tree', 'protected', 'lead', 'rainfall', 'area']  
['source', 'livelihood']  
['Somehow', 'protect']  
['help', 'lot', 'thing', 'air', 'breath']  
['protect', 'u', 'instance', 'tree', 'protect', 'wind', 'bring', 'fresh', 'air', 'well', 'prevent', 'river', 'drying']  
['Thats', 'get', 'honey', 'bee', 'hive', 'u', 'lot', 'money']  
['Thats', 'get', 'free', 'air', 'traditional', 'medicine', 'livelihood', 'depend', 'natural', 'resource']  
['instance', 'fish', 'caught', 'brings', 'income', 'source', 'food', 'well', 'tree', 'bring', 'rainfall', 'indirectly']  
[]

In [256]:

CN\_R1["Connection\_Nature\_Reasons"] = CN\_R1["Connection\_Nature\_Reasons"].fillna("")   
CN\_R1["Connection\_Nature\_Reasons"] = CN\_R1["Connection\_Nature\_Reasons"].astype(str)  
CN\_R1\_Text = " ".join(CN\_R1["Connection\_Nature\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CN\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Connection Nature", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [257]:

CN\_R1\_bigrams\_list = list(CN\_R1\_filter\_sentence)  
print(CN\_R1\_bigrams\_list)  
#CN\_R1\_bigram\_counts = Counter(zip(bigrams\_list, CN\_R1\_bigrams\_list[1:]))  
#print(CN\_R1\_bigram\_counts)   
#CN\_R1\_bigrams = pd.DataFrame(CN\_R1\_bigram\_counts.most\_common(7),  
#columns = ['Word', 'Frequency'])  
#print(CN\_R1\_bigrams)

[]

### 8.3 The Length of Existence of the Livelihood Project[¶](#X4e917b02fd15a67fda1c3fc8f486ea6ad71ca87)

In [258]:

RY\_grouped = df2.groupby('Name\_Main\_Project')['Range\_Years'].value\_counts(dropna=False)

In [259]:

RY\_grouped

Out[259]:

Name\_Main\_Project Range\_Years   
EbA\_CENTRAL\_MUCHINGA\_LUAPULA 3to4Years\_Long 2  
 NaN 1  
Ecosystem Conservation\_NORTH\_WESTERN 3to4Years\_Long 8  
 NaN 2  
 7to8Years\_Long 1  
PIN\_WESTERN 3to4Years\_Long 5  
 NaN 1  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN 3to4Years\_Long 12  
 5to6Years\_Long 9  
 Lessthan2Years\_Long 6  
 Greaterthan9Years\_Long 2  
 7to8Years\_Long 1  
SCReBS\_WESTERN 5to6Years\_Long 5  
 7to8Years\_Long 5  
 Greaterthan9Years\_Long 3  
 3to4Years\_Long 2  
SCRiKA\_LS 5to6Years\_Long 31  
 3to4Years\_Long 6  
 NaN 5  
 7to8Years\_Long 2  
TRALARD\_LNM 3to4Years\_Long 28  
 5to6Years\_Long 5  
 Lessthan2Years\_Long 3  
 Greaterthan9Years\_Long 2  
 NaN 2  
Name: count, dtype: int64

In [260]:

RY\_grouped1 = pd.DataFrame(RY\_grouped)

In [261]:

RY\_grouped1

Out[261]:

|  |  |  |
| --- | --- | --- |
|  |  | count |
| Name\_Main\_Project | Range\_Years |  |
| EbA\_CENTRAL\_MUCHINGA\_LUAPULA | 3to4Years\_Long | 2 |
|  | NaN | 1 |
| Ecosystem Conservation\_NORTH\_WESTERN | 3to4Years\_Long | 8 |
|  | NaN | 2 |
|  | 7to8Years\_Long | 1 |
| PIN\_WESTERN | 3to4Years\_Long | 5 |
|  | NaN | 1 |
| SCRALA\_SOUTHERN\_WESTERN\_NORTHEN | 3to4Years\_Long | 12 |
|  | 5to6Years\_Long | 9 |
|  | Lessthan2Years\_Long | 6 |
|  | Greaterthan9Years\_Long | 2 |
|  | 7to8Years\_Long | 1 |
| SCReBS\_WESTERN | 5to6Years\_Long | 5 |
|  | 7to8Years\_Long | 5 |
|  | Greaterthan9Years\_Long | 3 |
|  | 3to4Years\_Long | 2 |
| SCRiKA\_LS | 5to6Years\_Long | 31 |
|  | 3to4Years\_Long | 6 |
|  | NaN | 5 |
|  | 7to8Years\_Long | 2 |
| TRALARD\_LNM | 3to4Years\_Long | 28 |
|  | 5to6Years\_Long | 5 |
|  | Lessthan2Years\_Long | 3 |
|  | Greaterthan9Years\_Long | 2 |
|  | NaN | 2 |

In [262]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Lessthan2Years\_Long", "3to4Years\_Long", "5to6Years\_Long", "7to8Years\_Long", "Greaterthan9Years\_Long", "NaN"]  
ax = sns.barplot(data = RY\_grouped1, x="count", y="Name\_Main\_Project", hue="Range\_Years", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 5b: The Number of Livelihood Projects in a Particular Range of Year", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [263]:

LP = df4['Range\_Years'].value\_counts(dropna=False)  
LP\_P = (df4['Range\_Years'].value\_counts(normalize=True, dropna=False).round(2))

In [264]:

LP\_P

Out[264]:

Range\_Years  
3to4Years\_Long 0.42  
5to6Years\_Long 0.33  
NaN 0.08  
Lessthan2Years\_Long 0.06  
7to8Years\_Long 0.06  
Greaterthan9Years\_Long 0.05  
Name: proportion, dtype: float64

In [265]:

df\_LP = pd.DataFrame(LP)

In [266]:

df\_LP

Out[266]:

|  |  |
| --- | --- |
|  | count |
| Range\_Years |  |
| 3to4Years\_Long | 63 |
| 5to6Years\_Long | 50 |
| NaN | 12 |
| Lessthan2Years\_Long | 9 |
| 7to8Years\_Long | 9 |
| Greaterthan9Years\_Long | 7 |

In [267]:

ax = sns.countplot(df4["Range\_Years"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 5b: The Number of Livelihood Projects in a Particular Range of Year", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 8.4 Major Contributor to Landscape Transformation[¶](#X98038732099cf94d20950b33cc1cf073f0f5ab0)

In [268]:

CLT\_grouped = df2.groupby('Name\_Main\_Project')['Contributor\_Landscape\_Transformation'].value\_counts(dropna=False)

In [269]:

CLT\_grouped

Out[269]:

Name\_Main\_Project Contributor\_Landscape\_Transformation  
EbA\_CENTRAL\_MUCHINGA\_LUAPULA Land\_Agriculture 3  
Ecosystem Conservation\_NORTH\_WESTERN Land\_Agriculture 3  
 Uncontrolled\_Fires 3  
 NaN 3  
 Uncontrolled\_Fires,Land\_Agriculture 1  
 Wood\_Extraction 1  
PIN\_WESTERN Uncontrolled\_Fires 3  
 Wood\_Extraction 2  
 NaN 1  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN Uncontrolled\_Fires 19  
 Wood\_Extraction 7  
 Land\_Agriculture 2  
 NaN 2  
SCReBS\_WESTERN Uncontrolled\_Fires 7  
 Wood\_Extraction 5  
 Land\_Agriculture 2  
 Other 1  
SCRiKA\_LS Uncontrolled\_Fires 13  
 Land\_Agriculture 12  
 Wood\_Extraction 11  
 NaN 6  
 Uncontrolled\_Grazing 2  
TRALARD\_LNM Land\_Agriculture 22  
 Wood\_Extraction 13  
 Uncontrolled\_Fires 4  
 NaN 1  
Name: count, dtype: int64

In [270]:

CLT\_grouped1 = pd.DataFrame(CLT\_grouped)

In [271]:

CLT\_grouped1

Out[271]:

|  |  |  |
| --- | --- | --- |
|  |  | count |
| Name\_Main\_Project | Contributor\_Landscape\_Transformation |  |
| EbA\_CENTRAL\_MUCHINGA\_LUAPULA | Land\_Agriculture | 3 |
| Ecosystem Conservation\_NORTH\_WESTERN | Land\_Agriculture | 3 |
|  | Uncontrolled\_Fires | 3 |
|  | NaN | 3 |
|  | Uncontrolled\_Fires,Land\_Agriculture | 1 |
|  | Wood\_Extraction | 1 |
| PIN\_WESTERN | Uncontrolled\_Fires | 3 |
|  | Wood\_Extraction | 2 |
|  | NaN | 1 |
| SCRALA\_SOUTHERN\_WESTERN\_NORTHEN | Uncontrolled\_Fires | 19 |
|  | Wood\_Extraction | 7 |
|  | Land\_Agriculture | 2 |
|  | NaN | 2 |
| SCReBS\_WESTERN | Uncontrolled\_Fires | 7 |
|  | Wood\_Extraction | 5 |
|  | Land\_Agriculture | 2 |
|  | Other | 1 |
| SCRiKA\_LS | Uncontrolled\_Fires | 13 |
|  | Land\_Agriculture | 12 |
|  | Wood\_Extraction | 11 |
|  | NaN | 6 |
|  | Uncontrolled\_Grazing | 2 |
| TRALARD\_LNM | Land\_Agriculture | 22 |
|  | Wood\_Extraction | 13 |
|  | Uncontrolled\_Fires | 4 |
|  | NaN | 1 |

In [272]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Land\_Agriculture", "Wood\_Extraction", "Uncontrolled\_Fires", "Uncontrolled\_Grazing ", "Other", "NaN"]  
ax = sns.barplot(data = CLT\_grouped1, x="count", y="Name\_Main\_Project", hue="Contributor\_Landscape\_Transformation", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 6a: Major Contributor to Landscape Transformation", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [273]:

CLT = df4['Contributor\_Landscape\_Transformation'].value\_counts(dropna=False)  
CLT\_P = (df4['Contributor\_Landscape\_Transformation'].value\_counts(normalize=True, dropna=False).round(2))

In [274]:

CLT\_P

Out[274]:

Contributor\_Landscape\_Transformation  
Uncontrolled\_Fires 0.33  
Land\_Agriculture 0.29  
Wood\_Extraction 0.26  
NaN 0.09  
Uncontrolled\_Grazing 0.01  
Uncontrolled\_Fires,Land\_Agriculture 0.01  
Other 0.01  
Name: proportion, dtype: float64

In [275]:

df\_CLT = pd.DataFrame(CLT)

In [276]:

df\_CLT

Out[276]:

|  |  |
| --- | --- |
|  | count |
| Contributor\_Landscape\_Transformation |  |
| Uncontrolled\_Fires | 49 |
| Land\_Agriculture | 44 |
| Wood\_Extraction | 39 |
| NaN | 14 |
| Uncontrolled\_Grazing | 2 |
| Uncontrolled\_Fires,Land\_Agriculture | 1 |
| Other | 1 |

In [277]:

ax = sns.countplot(df4["Contributor\_Landscape\_Transformation"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 6b: Major Contributor to Landscape Transformation", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [278]:

agreement\_levels = ["Wood\_Extration", "Land\_Agriculture","Uncontrolled\_Fires"]  
CLT\_R = df2[df2["Contributor\_Landscape\_Transformation"].isin(agreement\_levels)]  
CLT\_R1 = CN\_R.drop(CLT\_R.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,29,30,31,32,33,34]], axis = 1)  
CLT\_R1grouped = CLT\_R1.groupby('Name\_Main\_Project')['Contributor\_Landscape\_Transformation']  
#CLT\_R1

In [279]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CLT\_R1.iterrows():  
 CLT\_R1\_filter\_sentence = []  
 CLT\_R1\_sentence = row["Contributor\_Landscape\_Transformation\_Reasons"]  
 if pd.isnull(CLT\_R1\_sentence):  
 continue  
 CLT\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CLT\_R1\_sentence)  
 CLT\_R1\_words = nltk.word\_tokenize(CLT\_R1\_sentence\_cleaned)  
 CLT\_R1\_words = [lemmatizer.lemmatize(w) for w in CLT\_R1\_words if w.lower() not in stop\_words]  
 CLT\_R1\_filter\_sentence.extend(CLT\_R1\_words)  
 print(CLT\_R1\_filter\_sentence)

['requires', 'huge', 'land', 'others', 'like', 'making', 'charcoal']  
['burning', 'charcoal', 'give', 'smoke', 'destroys', 'ozone', 'layer', 'well', 'brings', 'acidic', 'rain']  
['agriculture', 'activity', 'food']  
['electricity', 'thus', 'huge', 'demand', 'energy']  
['charcoal', 'purpose', 'bad', 'land', 'left', 'bare', 'unlike', 'agriculture', 'cutting', 'replaced', 'plant']  
['cultivation', 'cassava', 'requires', 'always', 'barren', 'land', 'never', 'cultivated', 'grow', 'well', 'crop', 'bean', 'vegetable', 'maize', 'reduce', 'deforestation']  
['burn', 'tree', 'shoot']  
['Chitemene', 'system', 'cultivation', 'lead', 'deforestation', 'others']  
['source', 'food', 'people', 'area']  
['Thats', 'get', 'livelihood']  
['source', 'income']  
['time', 'cut', 'tree', 'even', 'want', 'cultivate', 'well', 'burining', 'kill', 'animal', 'necesary', 'making', 'soil', 'fertile']  
['Buring', 'destroy', 'product', 'crop', 'soil']  
['cutting', 'treed', 'change', 'landscape']  
['cutting', 'tress', 'destroys', 'much', 'regrowth']  
['u', 'income', 'huge', 'area', 'tree', 'cut']  
['commercial', 'agriculture', 'activity', 'clear', 'huge', 'chuck', 'land']  
['Source', 'income']  
['forming', 'og', 'food', 'crop']  
['source', 'income']  
['people', 'plant', 'huge', 'area', 'land', 'livelihood']  
['animal', 'food', 'well', 'tree', 'would', 'dry']  
['income', 'food', 'crop']  
['farming', 'activity', 'brings', 'income']  
['soil', 'get', 'degraded', 'movebto', 'another', 'portion', 'land']  
['source', 'income']  
['people', 'burn', 'agriculture', 'area', 'looking', 'rat', 'addition', 'burn', 'food', 'crop', 'due', 'search', 'rat']  
['cutting', 'tree', 'destroy', 'change', 'landscape', 'livelihood']  
['tree', 'challenge', 'growung', 'burned']  
['people', 'cutting', 'tree', 'charcoal', 'destroy', 'difficult', 'regeneration']  
['Source', 'income']  
['tree', 'dry', 'burnt']  
['fire', 'cause', 'lot', 'damage', 'biodiversity', 'well', 'plant']  
['tree', 'get', 'burn', 'reducing', 'regeneration', 'well', 'fertility', 'soil']  
['tree', 'get', 'destroyed', 'burnt', 'difficult', 'regenerate']  
['cutting', 'tree', 'charcoal', 'prevents', 'regeneration']  
['cutting', 'tree', 'charcoal', 'requires', 'huge', 'land', 'compared', 'farm', 'one', 'partition', 'land']  
['small', 'biodiversity', 'destroyed', 'fire']  
['Burning', 'destroys', 'tree']  
['fire', 'burn', 'tree', 'lead', 'dry']  
['people', 'cutting', 'huge', 'chuck', 'land', 'cultivation', 'crop']  
['burning', 'lot', 'thing', 'like', 'snake', 'house', 'important', 'biodiversity', 'destroyed']  
['source', 'livelihood']  
['Thats', 'source', 'food', 'crop']  
['people', 'cutting', 'tree', 'anyhow']  
['cutting', 'tree', 'charcoal', 'finish', 'tree', 'others']  
['burning', 'destroys', 'fertility', 'soil']  
['reason', 'told', 'start', 'conservation', 'farming', 'entail', 'farming', 'locality']  
['Chitemene', 'system', 'burning', 'disallowed']  
['main', 'purpose', 'livelihood']  
['fire', 'destroys', 'lot', 'thing', 'air', 'breathe', 'soil', 'fertility', 'small', 'animal', 'plant']  
['land', 'cultivated', 'done', 'big', 'land', 'transforms', 'landscape']  
['fire', 'destroys', 'flower', 'production', 'honey', 'reduced']  
['people', 'make', 'charcoal', 'cut', 'tree', 'fresh', 'cut', 'huge', 'area']  
['destroys', 'lot', 'biodiversity', 'egg', 'bird', 'snake']  
['livelihood', 'depends', 'activity', 'charcoal', 'burning']  
['animal', 'always', 'grazing', 'vegetation', 'room', 'given', 'plant', 'sprout']  
['tree', 'cut', 'charcoal', 'stem', 'dy', 'replacement']  
['Fire', 'destroys', 'everything', 'others', 'even', 'biodiversity', 'get', 'killed']  
['fire', 'cut', 'across', 'huge', 'area', 'kill', 'everything', 'way']  
['land', 'used', 'agriculture', 'purpose', 'cutting', 'done', 'large', 'scale']  
['Burning', 'destroys', 'life', 'everything']  
['fire', 'destroys', 'almost', 'everything']  
['cause', 'soil', 'erossion']  
['agriculture', 'activity', 'uprooting', 'tree', 'thus', 'distruction', 'environment']  
['burning', 'cover', 'huge', 'area', 'kill', 'biodiversity', 'way']  
['making', 'charcoal', 'mainly', 'focus', 'big', 'tree', 'make', 'desert', 'area']  
['destroys', 'tree', 'everyone', 'cutting', 'tree', 'energy']  
['tree', 'cut', 'charcoal', 'take', 'time', 'grow', 'thus', 'causing', 'climate', 'change', 'turn', 'affecting', 'main', 'livelihood', 'agriculture']  
['cut', 'tree', 'rainfall', 'reduce', 'lead', 'animal', 'dying', 'thirst']  
['clear', 'law', 'prevents', 'people', 'scared', 'burning', 'bush']  
['huge', 'land', 'cleared', 'garden']  
['continuous', 'process', 'cutting', 'tree', 'charcoal', 'thus', 'destroys', 'landscape']  
['rainfall', 'reduce', 'would', 'much', 'wind', 'well', 'climate', 'change']  
['tree', 'reduced', 'going', 'problem', 'rainfall']  
['lot', 'famers', 'thus', 'major', 'contributor']  
['Source', 'livelihood', 'crop']  
['farmer']  
['burn', 'bush', 'destroys', 'tree', 'well', 'animal']  
['burning', 'make', 'tree', 'dry', 'well', 'young', 'animal', 'get', 'killed']  
['fire', 'distroys', 'almost', 'everything', 'even', 'bird', 'forest', 'destroyed']  
['tradition', 'cultivation', 'survive', 'tradition', 'burning', 'destroys', 'lot', 'thing', 'hence', 'major', 'contributor']  
['burning', 'bush', 'destroys', 'environment']  
['animal', 'depend', 'grass', 'thus', 'burnt', 'animal', 'would', 'die']  
['depend', 'farming', 'livelihood']  
['charcoal', 'people', 'burn', 'big', 'tree', 'destroys', 'habitat', 'animal']  
['Anyone', 'burn', 'bush', 'would', 'burn', 'forest', 'fire', 'guard', 'prevent', 'forest', 'burnt']  
['knowledge', 'protect', 'environment', 'people', 'thing', 'without', 'knowledge', 'protect', 'environment', 'thus', 'need', 'protect']  
['grass', 'grazing', 'animal', 'would', 'destroyed']  
['destroys', 'food', 'elephant', 'depends', 'burnt']  
['Fire', 'destroys', 'habitat', 'animal']  
['animal', 'feed', 'grass', 'grass', 'burnt', 'animal', 'come', 'community', 'disturb']  
['burn', 'destroys', 'habitat', 'animal']  
['Cutting', 'tree', 'come', 'strong', 'wind']  
['Thats', 'food', 'come']  
['Cutting', 'tree', 'make', 'gardenmatema', 'meaning', 'tree', 'replaced']  
['Fire', 'burn', 'everything', 'term', 'life']  
['fire', 'destroys', 'grazing', 'grass', 'animal']  
['People', 'always', 'burning', 'bush', 'without', 'control']  
['destroys', 'life']  
['Fire', 'burn', 'whole', 'area']  
['Burning', 'kill', 'lot', 'thing', 'biodiversity']  
['tree', 'cut', 'lot', 'thus', 'distructs', 'lot']  
['tree', 'cut', 'grow']  
['cutting', 'tree', 'brings', 'drought']  
['tree', 'cut', 'area', 'possing', 'treat', 'strong', 'wind']  
['rain', 'water', 'flow', 'due', 'lack', 'tree', 'block']  
['lead', 'climate', 'due', 'lack', 'tree']  
['animal', 'graz', 'thus', 'make', 'move', 'long', 'distance', 'find', 'pasture']  
['rainfall', 'back', 'green', 'vegetation', 'thus', 'burnt', 'less', 'rainfall']  
['tree', 'get', 'burnt', 'dry', 'thats', 'reason', 'less', 'rainfall']  
['use', 'Chitemene', 'system', 'cutting', 'tree', 'charcoal']  
['tree', 'cut', 'made', 'charcoal', 'trunk', 'stem', 'grow']  
['destroys', 'everything', 'way']  
['destroys', 'everything', 'path']

In [280]:

CLT\_R1["Contributor\_Landscape\_Transformation\_Reasons"] = CLT\_R1["Contributor\_Landscape\_Transformation\_Reasons"].fillna("")   
CLT\_R1["Contributor\_Landscape\_Transformation\_Reasons"] = CLT\_R1["Contributor\_Landscape\_Transformation\_Reasons"].astype(str)  
CLT\_R1\_Text = " ".join(CLT\_R1["Contributor\_Landscape\_Transformation\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CLT\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Contributor Landscape Transformation", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 8.5 The Landscape Depended on for a Livelihood[¶](#X5e17772f7d67d15b446dfefc4f2efb81be64c79)

In [281]:

LDL\_grouped = df2.groupby('Name\_Main\_Project')['Landscape\_Depended\_Livelihood'].value\_counts(dropna=False)

In [282]:

LDL\_grouped

Out[282]:

Name\_Main\_Project Landscape\_Depended\_Livelihood   
EbA\_CENTRAL\_MUCHINGA\_LUAPULA Agriculture\_Areas\_Dependent 3  
Ecosystem Conservation\_NORTH\_WESTERN NaN 5  
 Agriculture\_Areas\_Dependent 3  
 Forest\_Dependent 2  
 Wetlands\_Dependent 1  
PIN\_WESTERN Agriculture\_Areas\_Dependent 5  
 NaN 1  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN Agriculture\_Areas\_Dependent 21  
 Wetlands\_Dependent 6  
 Forest\_Dependent 2  
 NaN 1  
SCReBS\_WESTERN Agriculture\_Areas\_Dependent 10  
 Wetlands\_Dependent 5  
SCRiKA\_LS Agriculture\_Areas\_Dependent 30  
 Forest\_Dependent 4  
 Wetlands\_Dependent 4  
 NaN 4  
 Wetlands\_Dependent,Forest\_Dependent 2  
TRALARD\_LNM Agriculture\_Areas\_Dependent 20  
 Wetlands\_Dependent 9  
 Forest\_Dependent 5  
 Wetlands\_Dependent,Forest\_Dependent 2  
 Agriculture\_Areas\_Dependent,Wetlands\_Dependent 1  
 Forest\_Dependent,Wetlands\_Dependent 1  
 Wetlands\_Dependent,Agriculture\_Areas\_Dependent 1  
 NaN 1  
Name: count, dtype: int64

In [283]:

LDL\_grouped1 = pd.DataFrame(LDL\_grouped)

In [284]:

LDL\_grouped1

Out[284]:

|  |  |  |
| --- | --- | --- |
|  |  | count |
| Name\_Main\_Project | Landscape\_Depended\_Livelihood |  |
| EbA\_CENTRAL\_MUCHINGA\_LUAPULA | Agriculture\_Areas\_Dependent | 3 |
| Ecosystem Conservation\_NORTH\_WESTERN | NaN | 5 |
|  | Agriculture\_Areas\_Dependent | 3 |
|  | Forest\_Dependent | 2 |
|  | Wetlands\_Dependent | 1 |
| PIN\_WESTERN | Agriculture\_Areas\_Dependent | 5 |
|  | NaN | 1 |
| SCRALA\_SOUTHERN\_WESTERN\_NORTHEN | Agriculture\_Areas\_Dependent | 21 |
|  | Wetlands\_Dependent | 6 |
|  | Forest\_Dependent | 2 |
|  | NaN | 1 |
| SCReBS\_WESTERN | Agriculture\_Areas\_Dependent | 10 |
|  | Wetlands\_Dependent | 5 |
| SCRiKA\_LS | Agriculture\_Areas\_Dependent | 30 |
|  | Forest\_Dependent | 4 |
|  | Wetlands\_Dependent | 4 |
|  | NaN | 4 |
|  | Wetlands\_Dependent,Forest\_Dependent | 2 |
| TRALARD\_LNM | Agriculture\_Areas\_Dependent | 20 |
|  | Wetlands\_Dependent | 9 |
|  | Forest\_Dependent | 5 |
|  | Wetlands\_Dependent,Forest\_Dependent | 2 |
|  | Agriculture\_Areas\_Dependent,Wetlands\_Dependent | 1 |
|  | Forest\_Dependent,Wetlands\_Dependent | 1 |
|  | Wetlands\_Dependent,Agriculture\_Areas\_Dependent | 1 |
|  | NaN | 1 |

In [285]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Agriculture\_Areas\_Dependent", "Wetlands\_Dependent", "Forest\_Dependent", "NaN"]  
ax = sns.barplot(data = LDL\_grouped1, x="count", y="Name\_Main\_Project", hue="Landscape\_Depended\_Livelihood", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 7a: Landscape Dependent on by Livelihoods", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [286]:

LDL = df4['Landscape\_Depended\_Livelihood'].value\_counts(dropna=False)  
LDL\_P = (df4['Landscape\_Depended\_Livelihood'].value\_counts(normalize=True, dropna=False).round(2))

In [287]:

LDL\_P

Out[287]:

Landscape\_Depended\_Livelihood  
Agriculture\_Areas\_Dependent 0.61  
Wetlands\_Dependent 0.17  
Forest\_Dependent 0.09  
NaN 0.09  
Wetlands\_Dependent,Forest\_Dependent 0.03  
Agriculture\_Areas\_Dependent,Wetlands\_Dependent 0.01  
Forest\_Dependent,Wetlands\_Dependent 0.01  
Wetlands\_Dependent,Agriculture\_Areas\_Dependent 0.01  
Name: proportion, dtype: float64

In [288]:

df\_LDL = pd.DataFrame(LDL)

In [289]:

df\_LDL

Out[289]:

|  |  |
| --- | --- |
|  | count |
| Landscape\_Depended\_Livelihood |  |
| Agriculture\_Areas\_Dependent | 92 |
| Wetlands\_Dependent | 25 |
| Forest\_Dependent | 13 |
| NaN | 13 |
| Wetlands\_Dependent,Forest\_Dependent | 4 |
| Agriculture\_Areas\_Dependent,Wetlands\_Dependent | 1 |
| Forest\_Dependent,Wetlands\_Dependent | 1 |
| Wetlands\_Dependent,Agriculture\_Areas\_Dependent | 1 |

In [290]:

ax = sns.countplot(df4["Landscape\_Depended\_Livelihood"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 7b: Landscape Dependent on by Livelihoods", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [291]:

agreement\_levels = ["Agriculture\_Areas\_Dependent", "Wetlands\_Dependent","Forest\_Dependent"]  
LDL\_R = df2[df2["Landscape\_Depended\_Livelihood"].isin(agreement\_levels)]  
LDL\_R1 = LDL\_R.drop(LDL\_R.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,31,32,33,34]], axis = 1)  
LDL\_R1grouped = LDL\_R1.groupby('Name\_Main\_Project')['Landscape\_Depended\_Livelihood']  
#LDL\_R1

In [292]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in LDL\_R1.iterrows():  
 LDL\_R1\_filter\_sentence = []  
 LDL\_R1\_sentence = row["Landscape\_Depeneded\_Livelihood\_Reasons"]  
 if pd.isnull(LDL\_R1\_sentence):  
 continue  
 LDL\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',LDL\_R1\_sentence)  
 LDL\_R1\_words = nltk.word\_tokenize(LDL\_R1\_sentence\_cleaned)  
 LDL\_R1\_words = [lemmatizer.lemmatize(w) for w in LDL\_R1\_words if w.lower() not in stop\_words]  
 LDL\_R1\_filter\_sentence.extend(LDL\_R1\_words)  
 print(LDL\_R1\_filter\_sentence)

['cultivates', 'large', 'area', 'land', 'crop', 'sold', 'livelihood']  
['catching', 'fish', 'source', 'income', 'area']  
['Food', 'main', 'source', 'livelihood', 'thus', 'without', 'way', 'earn', 'living']  
['source', 'income', 'well', 'food', 'consumption']  
['farming', 'help', 'sourcing', 'food']  
['Thats', 'source', 'income']  
['source', 'livelihood']  
['income', 'come', 'living']  
['help', 'u', 'cultivate', 'cassava', 'maize', 'much', 'fishing', 'activity']  
['farming', 'obtain', 'food', 'consumption', 'household', 'level']  
['food', 'security', 'come']  
['tree', 'bring', 'rainfall', 'area', 'compared', 'area']  
['source', 'income']  
['majority', 'people', 'cultivate', 'land', 'livelihood', 'depending', 'buying', 'shop']  
['less', 'rainfall', 'help', 'u', 'source', 'water']  
['Thats', 'whats', 'common', 'within', 'area']  
['source', 'traditional', 'medicine']  
['provides', 'water', 'drinking', 'water', 'life']  
['Water', 'source', 'life']  
['source', 'food', 'crop', 'income']  
['use', 'cultivation']  
['Source', 'income']  
['mostly', 'farmer']  
['Thats', 'source', 'income', 'livelihood']  
['lot', 'activity', 'come', 'water']  
['food', 'crop']  
['food', 'crop', 'come']  
['Thats', 'source', 'food', 'crop', 'well', 'income']  
['Thats', 'grow', 'crop', 'livelihood']  
['livelihood', 'farming', 'main', 'stay']  
['provides', 'resource', 'people', 'term', 'wood']  
['obtain', 'food', 'crop', 'assist', 'u', 'livelihood']  
['Water', 'life', 'used', 'water', 'environment', 'tree', 'grow', 'prevent', 'developing', 'desert']  
['source', 'crop', 'food']  
['Thats', 'grow', 'crop', 'livelihood']  
['Source', 'life']  
['portion', 'land', 'cultivating', 'one', 'area', 'thus', 'cutting', 'tree']  
['Thats', 'get', 'food', 'product', 'livelihood']  
['Water', 'life']  
['Water', 'life']  
['Water', 'life']  
['answer']  
['crop', 'income', 'come']  
['Thats', 'get', 'food', 'eat']  
['Source', 'livelihood']  
['Life', 'water']  
['Water', 'life', 'evryone', 'drink']  
['income', 'food', 'crop', 'come']  
['tree', 'rain', 'much', 'well', 'soil', 'fertile']  
['Water', 'life']  
['Thats', 'get', 'food', 'crop', 'survival']  
['every', 'individual', 'depend', 'agriculture', 'land']  
['Water', 'life', 'thus', 'water', 'plant', 'dry']  
['source', 'mushroom', 'catapilars', 'forest', 'protected', 'well', 'working', 'honarary', 'officer']  
['Water', 'life', 'used', 'every', 'situation', 'cultivation', 'watering', 'garden']  
['brings', 'u', 'food', 'well', 'income']  
['flood', 'cattle', 'go', 'forest', 'area', 'graz']  
['Thats', 'get', 'food']  
['farmer']  
['farm', 'product']  
['Thats', 'get', 'food', 'crop']  
['Thats', 'get', 'food', 'crop']  
['get', 'food', 'crop', 'livelihood', 'dependent']  
['livelihood', 'based', 'farming']  
['source', 'source', 'livelihood']  
['get', 'food', 'crop', 'livehoods']  
['area', 'farming', 'found', 'forest', 'area']  
['cattle', 'graze']  
['cultivate', 'source', 'food', 'crop', 'game', 'park', 'help', 'depend', 'ZAWA', 'Officers', 'give', 'resource']  
['farmer', 'nature']  
['Thats', 'source', 'food', 'crop']  
['Thats', 'food', 'come']  
['Thats', 'get', 'food', 'eating', 'livelihood']  
['cultivate', 'get', 'crop']  
['food', 'income', 'come', 'help', 'u']  
['Thats', 'get', 'maize', 'staple', 'food', 'farming']  
['keeping', 'bird', 'gardening', 'thus', 'forest', 'protected', 'well', 'animal']  
['found']  
['Source', 'food', 'crop']  
['wetland', 'dry', 'thus', 'depend', 'agriculture', 'food', 'crop']  
['Thats', 'get', 'crop', 'food']  
['food', 'gotten', 'borehole', 'sank']  
['farmer']  
['Thats', 'animal', 'feed']  
['Thats', 'get', 'income']  
['Everything', 'come', 'agriculture']  
['farm', 'crop']  
['Water', 'source', 'life', 'animal', 'cattle']  
['farmer', 'nature']  
['farmer', 'nature', 'southern', 'province']  
['Thats', 'get', 'income']  
['Water', 'life', 'cattle', 'drink', 'water', 'wetland']  
['Water', 'life', 'domesticated', 'animal', 'need', 'water', 'depend', 'animal']  
['Thats', 'get', 'food', 'crop']  
['Thats', 'get', 'food', 'crop', 'income']  
['agriculture', 'get', 'food', 'crop', 'income']  
['major', 'activity', 'around', 'area']  
['food', 'crop']  
['Thats', 'get', 'food', 'living']  
['Thats', 'get', 'food', 'crop']  
['farmer']  
['Thats', 'get', 'food']  
['farming']  
['farmer']  
['Thats', 'food', 'come']  
['farmer']  
['Thats', 'animal', 'graze']  
['flood', 'go', 'leave', 'moisture', 'thats', 'help', 'people', 'grow', 'crop']  
['Thats', 'grow', 'food', 'crop']  
['wetland', 'cultivate', 'rice', 'well', 'get', 'water', 'watering', 'garden']  
['get', 'food', 'eating']  
['Thats', 'grow', 'crop']  
['Thats', 'get', 'food']  
['Thats', 'food', 'crop', 'come']  
['Thats', 'food', 'crop', 'come']  
['Thats', 'get', 'food', 'crop']  
['Thats', 'get', 'food', 'crop']  
['farmer', 'rice']  
['get', 'food', 'crop']  
['cultivate', 'flood', 'plain', 'animal', 'graze']  
['Thats', 'get', 'food']  
['farmer']  
['get', 'crop']  
['farmer']  
['Thats', 'get', 'food', 'money']  
['food', 'crop', 'found']  
['Thats', 'plant', 'maize', 'rice']

In [293]:

LDL\_R1["Landscape\_Depeneded\_Livelihood\_Reasons"] = LDL\_R1["Landscape\_Depeneded\_Livelihood\_Reasons"].fillna("")   
LDL\_R1["Landscape\_Depeneded\_Livelihood\_Reasons"] = LDL\_R1["Landscape\_Depeneded\_Livelihood\_Reasons"].astype(str)  
LDL\_R1\_Text = " ".join(LDL\_R1["Landscape\_Depeneded\_Livelihood\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(LDL\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Landscape Depeneded Livelihood", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

## 8.6 Explanation to Choices[¶](#X1acb2d2144870e216542e1c30cbda5a532ca0aa)

It gives an explanation to the choice selected that is not among the choices that were provided in the list of the questionnaire

In [294]:

df5=df2.drop(df2.columns[[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]], axis = 1)

In [295]:

#df5

In [296]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in df5.iterrows():  
 PL\_RS1\_filter\_sentence = []  
 PL\_RS1\_sentence = row["Purpose\_Landscape\_Specific"]  
 if pd.isnull(PL\_RS1\_sentence):  
 continue  
 PL\_RS1\_sentence\_cleaned = re.sub(r'[^\w\s]','',PL\_RS1\_sentence)  
 PL\_RS1\_words = nltk.word\_tokenize(PL\_RS1\_sentence\_cleaned)  
 PL\_RS1\_words = [lemmatizer.lemmatize(w) for w in PL\_RS1\_words if w.lower() not in stop\_words]  
 PL\_RS1\_filter\_sentence.extend(PL\_RS1\_words)  
 print(PL\_RS1\_filter\_sentence)

['protection', 'nature', 'others', 'important']  
['fire', 'burn', 'biodiversity', 'allow', 'environmental', 'process']  
['tourist', 'bring', 'income', 'term', 'viewing', 'animal']  
['source', 'income', 'tourism']

In [297]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in df5.iterrows():  
 CN\_RS1\_filter\_sentence = []  
 CN\_RS1\_sentence = row["Connection\_Nature\_Specific"]  
 if pd.isnull(CN\_RS1\_sentence):  
 continue  
 CN\_RS1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CN\_RS1\_sentence)  
 CN\_RS1\_words = nltk.word\_tokenize(CN\_RS1\_sentence\_cleaned)  
 CN\_RS1\_words = [lemmatizer.lemmatize(w) for w in CN\_RS1\_words if w.lower() not in stop\_words]  
 CN\_RS1\_filter\_sentence.extend(CN\_RS1\_words)  
 print(CN\_RS1\_filter\_sentence)

['Taking', 'care', 'nature', 'without', 'destroying']  
['Taking', 'care', 'nature']  
['Source', 'good', 'air', 'food', 'product']  
['related']  
['Protection', 'tree']  
['Taking', 'care', 'animal']

In [298]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in df5.iterrows():  
 CLT\_RS1\_filter\_sentence = []  
 CLT\_RS1\_sentence = row["Contributor\_Landscape\_Transformation\_Specific"]  
 if pd.isnull(CLT\_RS1\_sentence):  
 continue  
 CLT\_RS1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CLT\_RS1\_sentence)  
 CLT\_RS1\_words = nltk.word\_tokenize(CLT\_RS1\_sentence\_cleaned)  
 CLT\_RS1\_words = [lemmatizer.lemmatize(w) for w in CLT\_RS1\_words if w.lower() not in stop\_words]  
 CLT\_RS1\_filter\_sentence.extend(CLT\_RS1\_words)  
 print(CLT\_RS1\_filter\_sentence)

['Cutting', 'tree', 'sale']

In [299]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in df5.iterrows():  
 LDL\_RS1\_filter\_sentence = []  
 LDL\_RS1\_sentence = row["Landscape\_Depended\_Livelihood\_Specific"]  
 if pd.isnull(LDL\_RS1\_sentence):  
 continue  
 LDL\_RS1\_sentence\_cleaned = re.sub(r'[^\w\s]','',LDL\_RS1\_sentence)  
 LDL\_RS1\_words = nltk.word\_tokenize(LDL\_RS1\_sentence\_cleaned)  
 LDL\_RS1\_words = [lemmatizer.lemmatize(w) for w in LDL\_RS1\_words if w.lower() not in stop\_words]  
 LDL\_RS1\_filter\_sentence.extend(LDL\_RS1\_words)  
 print(LDL\_RS1\_filter\_sentence)

## 9. Description Statistics[¶](#Xb63a2db521ab2f47d37d53ee78c252f279d851a)

It gives a background to the study sites

In [300]:

df6=df2.drop(df2.columns[[0,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)

In [301]:

#df6

[[Back to top](#Purpose-of-the-Analysis)]

### 9.1 Main project Types[¶](#Xaf4c9e8cea389c0188e23c3ef2daea2f8544d3a)

This shows the number of questionaires that were administered to each project type

In [302]:

MP = df6['Name\_Main\_Project'].value\_counts(dropna=False)

In [303]:

MP

Out[303]:

Name\_Main\_Project  
SCRiKA\_LS 44  
TRALARD\_LNM 40  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN 30  
SCReBS\_WESTERN 15  
Ecosystem Conservation\_NORTH\_WESTERN 11  
PIN\_WESTERN 6  
EbA\_CENTRAL\_MUCHINGA\_LUAPULA 3  
NaN 1  
Name: count, dtype: int64

In [304]:

ax = sns.countplot(df6["Name\_Main\_Project"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 8: The Total Number of Respondents in each of the Main Project", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 9.2 The Wards[¶](#X2356ac38f682f08b1da47d39f5569e41b9f61ac)

The wards that had respodents and their number

In [305]:

WN = df6['Ward\_Name'].value\_counts(dropna=False)

In [306]:

WN

Out[306]:

Ward\_Name  
Namwala Central ward 17  
Omba ward 15  
Mbila ward 14  
Yeta ward 11  
Mwanambuyu ward 11  
Lulimala ward 11  
Isamba ward 10  
Kalobolelwa ward 9  
Ntonga ward 8  
Kalanga ward 8  
Moofwe ward 7  
Chitimbwa ward 7  
Makuya ward 6  
Nachikufu ward 5  
Luubwe ward 4  
Ntambu ward 4  
NaN 2  
Silunga ward 1  
Name: count, dtype: int64

In [307]:

ax = sns.countplot(df6["Ward\_Name"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 9: Number of Respondents in each Ward", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 9.3 Number of beneficiaries[¶](#X9b30bb981613628f0f1edc9bf4d8ac7135a04a5)

It shows the number of households that benefited from the interviewees

In [308]:

NB = df6['Number\_Beneficiaries'].value\_counts(dropna=False)

In [309]:

NB

Out[309]:

Number\_Beneficiaries  
Greaterthan40People\_Many 40  
10to20People\_Many 25  
NaN 25  
20to30People\_Many 24  
30to40People\_Many 23  
Lessthan10People\_Many 13  
Name: count, dtype: int64

In [310]:

ax = sns.countplot(df6["Number\_Beneficiaries"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 10: Number of household beneficiaries in each Cohort Category", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 9.4 Size of Landscape[¶](#Xb653e65a9df4c81a5a82f0afd9e6725855ed1a6)

In [311]:

SIZE = df6['Size'].value\_counts(dropna=False)

In [312]:

SIZE

Out[312]:

Size  
Lessthan4Lima\_Size 48  
Greaterthan16Lima\_Size 40  
NotApplicable\_Size 31  
4to8Lima\_Size 13  
NaN 10  
8to12Lima\_Size 5  
12to16Lima\_Size 3  
Name: count, dtype: int64

In [313]:

ax = sns.countplot(df6["Size"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 11: Number of beneficiaries in each Land Size Category", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 9.5 Represing Others[¶](#X308b8f647ce8a44903fcd973aca8ca6c597fb0d)

It looks at those representing one house and more

In [314]:

RO = df6['Representing\_Others'].value\_counts(dropna=False)

In [315]:

RO

Out[315]:

Representing\_Others  
yes 125  
no 23  
NaN 2  
Name: count, dtype: int64

In [316]:

ax = sns.countplot(df6["Representing\_Others"])  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
 ax.set\_title("Figure 12: Number of beneficiaries Representing Others", fontsize=14)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

[[Back to top](#Purpose-of-the-Analysis)]

## 10. Reasons[¶](#X3a6864d3553c0e5b6d0a7841152ba99470d60b9)

The pandas dataframe depicts columns for the reasons to the responses to the likert scales

In [317]:

df7=df2.drop(df2.columns[[0,1,2,3,4,5,6,7,8,10,11,13,14,15,17,18,19,20,22,23,24,25,26,27,29,31,32,33,34]], axis = 1)

In [318]:

#df7

### 10.1 Reasons for Cultural Practices Hidering[¶](#X83acbafc673b76b8ee0a0b9168f46bf6e5d3851)

The reasons for cultural practices that hinder transfromation

In [319]:

lemmatizer = WordNetLemmatizer()  
CPH = df7['Cultural\_Practices\_Hinder\_Reason'].str.lower().str.cat(sep=' ')  
CPH\_words = nltk.tokenize.word\_tokenize(CPH)  
CPH\_filtered\_tokens = [word for word in CPH\_words if len(CPH\_words) >= 8]  
CPH\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in CPH\_filtered\_tokens]  
CPH\_token\_counts = Counter(CPH\_lemmatized\_words)  
CPH\_columns = pd.DataFrame(CPH\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
#print(CPH\_columns)

In [320]:

bigrams\_list = list(bigrams(CPH\_filtered\_tokens))  
#print(bigrams\_list)  
CPH\_bigram\_counts = Counter(zip(bigrams\_list, bigrams\_list[1:]))  
#print(bigram\_counts)   
CPH\_bigrams = pd.DataFrame(CPH\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CPH\_bigrams)

Word Frequency  
0 ((there, is), (is, nothing)) 73  
1 ((is, nothing), (nothing, there)) 55  
2 ((nothing, there), (there, is)) 52  
3 ((protect, the), (the, environment)) 6  
4 ((to, protect), (protect, the)) 6  
5 ((there, is), (is, no)) 5  
6 ((there, are), (are, no)) 4  
7 ((the, environment), (environment, there)) 4  
8 ((is, nothing), (nothing, the)) 4  
9 ((customary, practices), (practices, that)) 3  
10 ((as, well), (well, as)) 3  
11 ((are, no), (no, cultural)) 3  
12 ((nothing, there), (there, are)) 3  
13 ((destroys, the), (the, environment)) 3  
14 ((cutting, down), (down, of)) 3  
15 ((environment, there), (there, is)) 3  
16 ((is, nothing), (nothing, i)) 3  
17 ((``, ''), ('', malende)) 3  
18 (('', malende), (malende, '')) 3  
19 ((malende, ''), ('', '')) 3  
20 ((is, no), (no, cultural)) 2  
21 ((there, are), (are, rules)) 2  
22 ((there, is), (is, a)) 2  
23 ((that, the), (the, bush)) 2  
24 ((it, is), (is, not)) 2  
25 ((are, customary), (customary, practices)) 2  
26 ((a, customary), (customary, practice)) 2  
27 ((that, destroys), (destroys, the)) 2  
28 ((down, of), (of, trees)) 2  
29 ((along, the), (the, river)) 2  
30 ((a, long), (long, time)) 2  
31 ((but, at), (at, the)) 2  
32 ((at, the), (the, moment)) 2  
33 ((no, cultural), (cultural, practices)) 2  
34 ((the, natural), (natural, resources)) 2  
35 ((there, are), (are, places)) 2  
36 ((nothing, i), (i, have)) 2  
37 ((is, nothing), (nothing, our)) 2  
38 ((us, to), (to, protect)) 2  
39 ((of, the), (the, environment)) 2  
40 ((the, environment), (environment, the)) 2  
41 ((on, how), (how, to)) 2  
42 ((how, to), (to, protect)) 2  
43 ((trees, there), (there, is)) 2  
44 ((cut, down), (down, trees)) 2  
45 ((called, ``), (``, '')) 2  
46 (('', ''), ('', that)) 2  
47 ((trees, but), (but, the)) 2  
48 ((of, the), (the, trees)) 2  
49 ((it, has), (has, never)) 1  
50 ((has, never), (never, happed)) 1  
51 ((never, happed), (happed, before)) 1  
52 ((happed, before), (before, in)) 1  
53 ((before, in), (in, his)) 1  
54 ((in, his), (his, life)) 1  
55 ((his, life), (life, time)) 1  
56 ((life, time), (time, there)) 1  
57 ((time, there), (there, is)) 1  
58 ((there, is), (is, need)) 1  
59 ((is, need), (need, to)) 1  
60 ((need, to), (to, harvest)) 1  
61 ((to, harvest), (harvest, trees)) 1  
62 ((harvest, trees), (trees, when)) 1  
63 ((trees, when), (when, they)) 1  
64 ((when, they), (they, have)) 1  
65 ((they, have), (have, fully)) 1  
66 ((have, fully), (fully, grown)) 1

In [321]:

trigrams\_list = list(trigrams(CPH\_filtered\_tokens))  
#print(bigrams\_list)  
CPH\_trigram\_counts = Counter(zip(trigrams\_list, trigrams\_list[1:]))  
#print(bigram\_counts)   
CPH\_trigrams = pd.DataFrame(CPH\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CPH\_trigrams)

Word \  
0 ((there, is, nothing), (is, nothing, there))   
1 ((is, nothing, there), (nothing, there, is))   
2 ((nothing, there, is), (there, is, nothing))   
3 ((to, protect, the), (protect, the, environment))   
4 ((there, is, nothing), (is, nothing, the))   
5 ((there, are, no), (are, no, cultural))   
6 ((is, nothing, there), (nothing, there, are))   
7 ((the, environment, there), (environment, there, is))   
8 ((environment, there, is), (there, is, nothing))   
9 ((there, is, nothing), (is, nothing, i))   
10 ((``, '', malende), ('', malende, ''))   
11 (('', malende, ''), (malende, '', ''))   
12 ((nothing, there, is), (there, is, no))   
13 ((there, is, no), (is, no, cultural))   
14 ((are, customary, practices), (customary, practices, that))   
15 ((that, destroys, the), (destroys, the, environment))   
16 ((destroys, the, environment), (the, environment, there))   
17 ((cutting, down, of), (down, of, trees))   
18 ((but, at, the), (at, the, moment))   
19 ((are, no, cultural), (no, cultural, practices))   
20 ((is, nothing, i), (nothing, i, have))   
21 ((there, is, nothing), (is, nothing, our))   
22 ((us, to, protect), (to, protect, the))   
23 ((on, how, to), (how, to, protect))   
24 ((how, to, protect), (to, protect, the))   
25 ((it, has, never), (has, never, happed))   
26 ((has, never, happed), (never, happed, before))   
27 ((never, happed, before), (happed, before, in))   
28 ((happed, before, in), (before, in, his))   
29 ((before, in, his), (in, his, life))   
30 ((in, his, life), (his, life, time))   
31 ((his, life, time), (life, time, there))   
32 ((life, time, there), (time, there, is))   
33 ((time, there, is), (there, is, need))   
34 ((there, is, need), (is, need, to))   
35 ((is, need, to), (need, to, harvest))   
36 ((need, to, harvest), (to, harvest, trees))   
37 ((to, harvest, trees), (harvest, trees, when))   
38 ((harvest, trees, when), (trees, when, they))   
39 ((trees, when, they), (when, they, have))   
40 ((when, they, have), (they, have, fully))   
41 ((they, have, fully), (have, fully, grown))   
42 ((have, fully, grown), (fully, grown, the))   
43 ((fully, grown, the), (grown, the, same))   
44 ((grown, the, same), (the, same, applies))   
45 ((the, same, applies), (same, applies, to))   
46 ((same, applies, to), (applies, to, biodiversity))   
47 ((applies, to, biodiversity), (to, biodiversity, there))   
48 ((to, biodiversity, there), (biodiversity, there, are))   
49 ((biodiversity, there, are), (there, are, no))   
50 ((there, are, no), (are, no, customary))   
51 ((are, no, customary), (no, customary, practices))   
52 ((no, customary, practices), (customary, practices, that))   
53 ((customary, practices, that), (practices, that, hinder))   
54 ((practices, that, hinder), (that, hinder, sustainable))   
55 ((that, hinder, sustainable), (hinder, sustainable, management))   
56 ((hinder, sustainable, management), (sustainable, management, there))   
57 ((sustainable, management, there), (management, there, are))   
58 ((management, there, are), (there, are, crop))   
59 ((there, are, crop), (are, crop, rotations))   
60 ((are, crop, rotations), (crop, rotations, made))   
61 ((crop, rotations, made), (rotations, made, on))   
62 ((rotations, made, on), (made, on, the))   
63 ((made, on, the), (on, the, land))   
64 ((on, the, land), (the, land, when))   
65 ((the, land, when), (land, when, cultivating))   
66 ((land, when, cultivating), (when, cultivating, to))   
  
 Frequency   
0 54   
1 52   
2 47   
3 5   
4 4   
5 3   
6 3   
7 3   
8 3   
9 3   
10 3   
11 3   
12 2   
13 2   
14 2   
15 2   
16 2   
17 2   
18 2   
19 2   
20 2   
21 2   
22 2   
23 2   
24 2   
25 1   
26 1   
27 1   
28 1   
29 1   
30 1   
31 1   
32 1   
33 1   
34 1   
35 1   
36 1   
37 1   
38 1   
39 1   
40 1   
41 1   
42 1   
43 1   
44 1   
45 1   
46 1   
47 1   
48 1   
49 1   
50 1   
51 1   
52 1   
53 1   
54 1   
55 1   
56 1   
57 1   
58 1   
59 1   
60 1   
61 1   
62 1   
63 1   
64 1   
65 1   
66 1

In [322]:

lemmatizer = WordNetLemmatizer()  
CAC = df7['Cultural\_Aspects\_Considered\_Reasons'].str.lower().str.cat(sep=' ')  
CAC\_words = nltk.tokenize.word\_tokenize(CAC)  
CAC\_filtered\_tokens = [word for word in CAC\_words if len(CAC\_words) >= 4]  
CAC\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in CAC\_filtered\_tokens]  
CAC\_token\_counts = Counter(CAC\_lemmatized\_words)  
CAC\_columns = pd.DataFrame(CAC\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CAC\_columns)

Word Frequency  
0 the 73  
1 not 52  
2 is 51  
3 we 47  
4 there 45  
5 nothing 39  
6 do 36  
7 to 23  
8 are 23  
9 it 17  
10 project 15  
11 that 13  
12 of 13  
13 cultural 12  
14 have 11  
15 can 11  
16 included 11  
17 they 11  
18 no 10  
19 in 10  
20 a 10  
21 because 10  
22 traditional 8  
23 aspect 7  
24 follow 7  
25 practice 7  
26 and 6  
27 be 6  
28 if 6  
29 our 6  
30 for 5  
31 from 5  
32 include 5  
33 considered 5  
34 what 4  
35 so 4  
36 chief 4  
37 give 4  
38 or 4  
39 answer 4  
40 authority 4  
41 at 3  
42 time 3  
43 land 3  
44 only 3  
45 government 3  
46 u 3  
47 when 3  
48 law 3  
49 place 3  
50 would 3  
51 well 3  
52 customary 3  
53 destroyed 3  
54 done 3  
55 but 3  
56 people 3  
57 change 3  
58 community 3  
59 history 2  
60 need 2  
61 come 2  
62 with 2  
63 say 2  
64 , 2  
65 tree 2  
66 cut 2

In [323]:

CAC\_bigrams\_list = list(bigrams(CAC\_filtered\_tokens))  
#print(bigrams\_list)  
CAC\_bigram\_counts = Counter(zip(CAC\_bigrams\_list, CAC\_bigrams\_list[1:]))  
#print(bigram\_counts)   
CAC\_bigrams = pd.DataFrame(CAC\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CAC\_bigrams)

Word Frequency  
0 ((there, is), (is, nothing)) 33  
1 ((we, do), (do, not)) 26  
2 ((is, nothing), (nothing, we)) 9  
3 ((nothing, we), (we, do)) 9  
4 ((is, nothing), (nothing, there)) 8  
5 ((nothing, there), (there, is)) 7  
6 ((do, not), (not, do)) 7  
7 ((do, not), (not, include)) 5  
8 ((they, are), (are, not)) 5  
9 ((not, do), (do, it)) 5  
10 ((the, cultural), (cultural, aspects)) 5  
11 ((are, not), (not, included)) 4  
12 ((there, are), (are, no)) 3  
13 ((is, nothing), (nothing, the)) 3  
14 ((the, traditional), (traditional, authority)) 3  
15 ((cultural, aspects), (aspects, are)) 3  
16 ((not, included), (included, we)) 3  
17 ((included, we), (we, do)) 3  
18 ((that, there), (there, is)) 2  
19 ((we, only), (only, follow)) 2  
20 ((we, have), (have, to)) 2  
21 ((has, to), (to, be)) 2  
22 ((is, nothing), (nothing, at)) 2  
23 ((nothing, at), (at, the)) 2  
24 ((at, the), (the, moment)) 2  
25 ((we, follow), (follow, the)) 2  
26 ((follow, the), (the, laws)) 2  
27 ((the, project), (project, can)) 2  
28 ((if, there), (there, are)) 2  
29 ((are, no), (no, customary)) 2  
30 ((no, customary), (customary, practices)) 2  
31 ((if, it), (it, is)) 2  
32 ((it, is), (is, not)) 2  
33 ((not, work), (work, well)) 2  
34 ((not, do), (do, that)) 2  
35 ((is, nothing), (nothing, no)) 2  
36 ((nothing, no), (no, answer)) 2  
37 ((do, it), (it, we)) 2  
38 ((do, not), (not, have)) 2  
39 ((we, always), (always, follow)) 2  
40 ((not, include), (include, there)) 2  
41 ((include, there), (there, is)) 2  
42 ((is, nothing), (nothing, they)) 2  
43 ((can, not), (not, be)) 2  
44 ((do, not), (not, the)) 2  
45 ((are, considered), (considered, because)) 2  
46 ((in, the), (the, community)) 2  
47 ((is, nothing), (nothing, included)) 2  
48 ((nothing, is), (is, considered)) 2  
49 ((considered, there), (there, is)) 2  
50 ((do, not), (not, not)) 2  
51 ((do, not), (not, we)) 2  
52 ((aspects, are), (are, not)) 2  
53 ((there, is), (is, no)) 1  
54 ((is, no), (no, cultural)) 1  
55 ((no, cultural), (cultural, history)) 1  
56 ((cultural, history), (history, in)) 1  
57 ((history, in), (in, the)) 1  
58 ((in, the), (the, area)) 1  
59 ((the, area), (area, there)) 1  
60 ((area, there), (there, is)) 1  
61 ((there, is), (is, need)) 1  
62 ((is, need), (need, to)) 1  
63 ((need, to), (to, have)) 1  
64 ((to, have), (have, power)) 1  
65 ((have, power), (power, in)) 1  
66 ((power, in), (in, what)) 1

In [324]:

CAC\_trigrams\_list = list(trigrams(CAC\_filtered\_tokens))  
#print(bigrams\_list)  
CAC\_trigram\_counts = Counter(zip(CAC\_trigrams\_list, CAC\_trigrams\_list[1:]))  
#print(bigram\_counts)   
CAC\_trigrams = pd.DataFrame(CAC\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CAC\_trigrams)

Word Frequency  
0 ((there, is, nothing), (is, nothing, we)) 9  
1 ((nothing, we, do), (we, do, not)) 9  
2 ((is, nothing, we), (nothing, we, do)) 8  
3 ((there, is, nothing), (is, nothing, there)) 7  
4 ((is, nothing, there), (nothing, there, is)) 7  
5 ((nothing, there, is), (there, is, nothing)) 7  
6 ((we, do, not), (do, not, do)) 7  
7 ((we, do, not), (do, not, include)) 5  
8 ((do, not, do), (not, do, it)) 5  
9 ((there, is, nothing), (is, nothing, the)) 3  
10 ((the, cultural, aspects), (cultural, aspects, are)) 3  
11 ((not, included, we), (included, we, do)) 3  
12 ((included, we, do), (we, do, not)) 3  
13 ((there, is, nothing), (is, nothing, at)) 2  
14 ((is, nothing, at), (nothing, at, the)) 2  
15 ((nothing, at, the), (at, the, moment)) 2  
16 ((they, are, not), (are, not, included)) 2  
17 ((if, there, are), (there, are, no)) 2  
18 ((there, are, no), (are, no, customary)) 2  
19 ((are, no, customary), (no, customary, practices)) 2  
20 ((if, it, is), (it, is, not)) 2  
21 ((do, not, do), (not, do, that)) 2  
22 ((there, is, nothing), (is, nothing, no)) 2  
23 ((is, nothing, no), (nothing, no, answer)) 2  
24 ((not, do, it), (do, it, we)) 2  
25 ((we, do, not), (do, not, have)) 2  
26 ((do, not, include), (not, include, there)) 2  
27 ((not, include, there), (include, there, is)) 2  
28 ((include, there, is), (there, is, nothing)) 2  
29 ((there, is, nothing), (is, nothing, they)) 2  
30 ((we, do, not), (do, not, the)) 2  
31 ((there, is, nothing), (is, nothing, included)) 2  
32 ((considered, there, is), (there, is, nothing)) 2  
33 ((we, do, not), (do, not, not)) 2  
34 ((we, do, not), (do, not, we)) 2  
35 ((cultural, aspects, are), (aspects, are, not)) 2  
36 ((aspects, are, not), (are, not, included)) 2  
37 ((are, not, included), (not, included, we)) 2  
38 ((there, is, no), (is, no, cultural)) 1  
39 ((is, no, cultural), (no, cultural, history)) 1  
40 ((no, cultural, history), (cultural, history, in)) 1  
41 ((cultural, history, in), (history, in, the)) 1  
42 ((history, in, the), (in, the, area)) 1  
43 ((in, the, area), (the, area, there)) 1  
44 ((the, area, there), (area, there, is)) 1  
45 ((area, there, is), (there, is, need)) 1  
46 ((there, is, need), (is, need, to)) 1  
47 ((is, need, to), (need, to, have)) 1  
48 ((need, to, have), (to, have, power)) 1  
49 ((to, have, power), (have, power, in)) 1  
50 ((have, power, in), (power, in, what)) 1  
51 ((power, in, what), (in, what, is)) 1  
52 ((in, what, is), (what, is, being)) 1  
53 ((what, is, being), (is, being, formulated)) 1  
54 ((is, being, formulated), (being, formulated, so)) 1  
55 ((being, formulated, so), (formulated, so, that)) 1  
56 ((formulated, so, that), (so, that, there)) 1  
57 ((so, that, there), (that, there, is)) 1  
58 ((that, there, is), (there, is, ownership)) 1  
59 ((there, is, ownership), (is, ownership, there)) 1  
60 ((is, ownership, there), (ownership, there, are)) 1  
61 ((ownership, there, are), (there, are, no)) 1  
62 ((there, are, no), (are, no, cultural)) 1  
63 ((are, no, cultural), (no, cultural, aspects)) 1  
64 ((no, cultural, aspects), (cultural, aspects, at)) 1  
65 ((cultural, aspects, at), (aspects, at, the)) 1  
66 ((aspects, at, the), (at, the, time)) 1

In [325]:

lemmatizer = WordNetLemmatizer()  
CoN = df7['Connection\_Nature\_Reasons'].str.lower().str.cat(sep=' ')  
CoN\_words = nltk.tokenize.word\_tokenize(CoN)  
CoN\_filtered\_tokens = [word for word in CoN\_words if len(CoN\_words) >= 4]  
CoN\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in CoN\_filtered\_tokens]  
CoN\_token\_counts = Counter(CoN\_lemmatized\_words)  
CoN\_columns = pd.DataFrame(CoN\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
#print(CoN\_columns)

In [326]:

CoN\_bigrams\_list = list(bigrams(CoN\_filtered\_tokens))  
#print(bigrams\_list)  
CoN\_bigram\_counts = Counter(zip(CoN\_bigrams\_list, CoN\_bigrams\_list[1:]))  
#print(bigram\_counts)   
CoN\_bigrams = pd.DataFrame(CoN\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CoN\_bigrams)

Word Frequency  
0 ((source, of), (of, income)) 11  
1 ((a, source), (source, of)) 9  
2 ((as, well), (well, as)) 5  
3 ((they, are), (are, a)) 5  
4 ((are, a), (a, source)) 5  
5 ((of, income), (income, they)) 3  
6 ((the, natural), (natural, resources)) 3  
7 ((the, animals), (animals, can)) 2  
8 ((animals, can), (can, be)) 2  
9 ((can, be), (be, used)) 2  
10 ((the, future), (future, generation)) 2  
11 ((income, they), (they, are)) 2  
12 ((future, generation), (generation, to)) 2  
13 ((them, source), (source, of)) 2  
14 ((it, is), (is, a)) 2  
15 ((is, a), (a, source)) 2  
16 ((there, is), (is, need)) 2  
17 ((is, need), (need, to)) 2  
18 ((are, the), (the, ones)) 2  
19 ((look, after), (after, them)) 2  
20 ((source, of), (of, livelihood)) 2  
21 ((of, income), (income, the)) 2  
22 ((help, us), (us, in)) 2  
23 ((are, source), (source, of)) 2  
24 ((source, of), (of, food)) 2  
25 ((of, food), (food, as)) 2  
26 ((food, as), (as, well)) 2  
27 ((a, lot), (lot, of)) 2  
28 ((thats, where), (where, we)) 2  
29 ((where, we), (we, get)) 2  
30 ((be, used), (used, by)) 1  
31 ((used, by), (by, the)) 1  
32 ((by, the), (the, future)) 1  
33 ((future, generation), (generation, as)) 1  
34 ((generation, as), (as, well)) 1  
35 ((as, well), (well, how)) 1  
36 ((well, how), (how, the)) 1  
37 ((how, the), (the, animals)) 1  
38 ((be, used), (used, as)) 1  
39 ((used, as), (as, an)) 1  
40 ((as, an), (an, example)) 1  
41 ((an, example), (example, on)) 1  
42 ((example, on), (on, how)) 1  
43 ((on, how), (how, people)) 1  
44 ((how, people), (people, should)) 1  
45 ((people, should), (should, lead)) 1  
46 ((should, lead), (lead, their)) 1  
47 ((lead, their), (their, life)) 1  
48 ((their, life), (life, through)) 1  
49 ((life, through), (through, experiments)) 1  
50 ((through, experiments), (experiments, he)) 1  
51 ((experiments, he), (he, gains)) 1  
52 ((he, gains), (gains, knowledge)) 1  
53 ((gains, knowledge), (knowledge, on)) 1  
54 ((knowledge, on), (on, the)) 1  
55 ((on, the), (the, management)) 1  
56 ((the, management), (management, of)) 1  
57 ((management, of), (of, forest)) 1  
58 ((of, forest), (forest, to)) 1  
59 ((forest, to), (to, use)) 1  
60 ((to, use), (use, some)) 1  
61 ((use, some), (some, of)) 1  
62 ((some, of), (of, them)) 1  
63 ((of, them), (them, for)) 1  
64 ((them, for), (for, income)) 1  
65 ((for, income), (income, they)) 1  
66 ((they, are), (are, important)) 1

In [327]:

CoN\_trigrams\_list = list(trigrams(CoN\_filtered\_tokens))  
#print(bigrams\_list)  
CoN\_trigram\_counts = Counter(zip(CoN\_trigrams\_list, CoN\_trigrams\_list[1:]))  
#print(bigram\_counts)   
CoN\_trigrams = pd.DataFrame(CoN\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CoN\_trigrams)

Word Frequency  
0 ((a, source, of), (source, of, income)) 6  
1 ((they, are, a), (are, a, source)) 5  
2 ((are, a, source), (a, source, of)) 5  
3 ((source, of, income), (of, income, they)) 3  
4 ((the, animals, can), (animals, can, be)) 2  
5 ((animals, can, be), (can, be, used)) 2  
6 ((them, source, of), (source, of, income)) 2  
7 ((it, is, a), (is, a, source)) 2  
8 ((is, a, source), (a, source, of)) 2  
9 ((there, is, need), (is, need, to)) 2  
10 ((a, source, of), (source, of, livelihood)) 2  
11 ((source, of, income), (of, income, the)) 2  
12 ((source, of, food), (of, food, as)) 2  
13 ((of, food, as), (food, as, well)) 2  
14 ((food, as, well), (as, well, as)) 2  
15 ((thats, where, we), (where, we, get)) 2  
16 ((can, be, used), (be, used, by)) 1  
17 ((be, used, by), (used, by, the)) 1  
18 ((used, by, the), (by, the, future)) 1  
19 ((by, the, future), (the, future, generation)) 1  
20 ((the, future, generation), (future, generation, as)) 1  
21 ((future, generation, as), (generation, as, well)) 1  
22 ((generation, as, well), (as, well, how)) 1  
23 ((as, well, how), (well, how, the)) 1  
24 ((well, how, the), (how, the, animals)) 1  
25 ((how, the, animals), (the, animals, can)) 1  
26 ((can, be, used), (be, used, as)) 1  
27 ((be, used, as), (used, as, an)) 1  
28 ((used, as, an), (as, an, example)) 1  
29 ((as, an, example), (an, example, on)) 1  
30 ((an, example, on), (example, on, how)) 1  
31 ((example, on, how), (on, how, people)) 1  
32 ((on, how, people), (how, people, should)) 1  
33 ((how, people, should), (people, should, lead)) 1  
34 ((people, should, lead), (should, lead, their)) 1  
35 ((should, lead, their), (lead, their, life)) 1  
36 ((lead, their, life), (their, life, through)) 1  
37 ((their, life, through), (life, through, experiments)) 1  
38 ((life, through, experiments), (through, experiments, he)) 1  
39 ((through, experiments, he), (experiments, he, gains)) 1  
40 ((experiments, he, gains), (he, gains, knowledge)) 1  
41 ((he, gains, knowledge), (gains, knowledge, on)) 1  
42 ((gains, knowledge, on), (knowledge, on, the)) 1  
43 ((knowledge, on, the), (on, the, management)) 1  
44 ((on, the, management), (the, management, of)) 1  
45 ((the, management, of), (management, of, forest)) 1  
46 ((management, of, forest), (of, forest, to)) 1  
47 ((of, forest, to), (forest, to, use)) 1  
48 ((forest, to, use), (to, use, some)) 1  
49 ((to, use, some), (use, some, of)) 1  
50 ((use, some, of), (some, of, them)) 1  
51 ((some, of, them), (of, them, for)) 1  
52 ((of, them, for), (them, for, income)) 1  
53 ((them, for, income), (for, income, they)) 1  
54 ((for, income, they), (income, they, are)) 1  
55 ((income, they, are), (they, are, important)) 1  
56 ((they, are, important), (are, important, in)) 1  
57 ((are, important, in), (important, in, our)) 1  
58 ((important, in, our), (in, our, lifes)) 1  
59 ((in, our, lifes), (our, lifes, like)) 1  
60 ((our, lifes, like), (lifes, like, trees)) 1  
61 ((lifes, like, trees), (like, trees, they)) 1  
62 ((like, trees, they), (trees, they, provide)) 1  
63 ((trees, they, provide), (they, provide, home)) 1  
64 ((they, provide, home), (provide, home, for)) 1  
65 ((provide, home, for), (home, for, animals)) 1  
66 ((home, for, animals), (for, animals, as)) 1

In [328]:

lemmatizer = WordNetLemmatizer()  
CLE = df7['Change\_Livelihood\_Easy\_Reasons'].str.lower().str.cat(sep=' ')  
CLE\_words = nltk.tokenize.word\_tokenize(CLE)  
CLE\_filtered\_tokens = [word for word in CLE\_words if len(CLE\_words) >= 4]  
CLE\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in CLE\_filtered\_tokens]  
CLE\_token\_counts = Counter(CLE\_lemmatized\_words)  
CLE\_columns = pd.DataFrame(CLE\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
#print(CLE\_columns)

In [329]:

CLE\_bigrams\_list = list(bigrams(CLE\_filtered\_tokens))  
#print(bigrams\_list)  
CLE\_bigram\_counts = Counter(zip(CLE\_bigrams\_list, CLE\_bigrams\_list[1:]))  
#print(bigram\_counts)   
CLE\_bigrams = pd.DataFrame(CLE\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CLE\_bigrams)

Word Frequency  
0 ((as, long), (long, as)) 14  
1 ((it, is), (is, a)) 7  
2 ((to, change), (change, because)) 6  
3 ((long, as), (as, there)) 6  
4 ((need, to), (to, change)) 5  
5 ((we, can), (can, change)) 5  
6 ((we, depend), (depend, on)) 5  
7 ((we, do), (do, not)) 5  
8 ((there, is), (is, need)) 4  
9 ((the, natural), (natural, resources)) 4  
10 ((as, well), (well, as)) 4  
11 ((it, is), (is, easier)) 4  
12 ((if, there), (there, are)) 4  
13 ((as, there), (there, is)) 4  
14 ((do, not), (not, have)) 4  
15 ((is, a), (a, challenge)) 4  
16 ((of, climate), (climate, change)) 4  
17 ((thus, it), (it, is)) 4  
18 ((can, not), (not, change)) 4  
19 ((if, there), (there, is)) 3  
20 ((can, change), (change, the)) 3  
21 ((is, need), (need, to)) 3  
22 ((long, as), (as, i)) 3  
23 ((as, i), (i, have)) 3  
24 ((source, of), (of, income)) 3  
25 ((that, we), (we, do)) 3  
26 ((is, not), (not, easy)) 3  
27 ((i, do), (do, not)) 3  
28 ((for, our), (our, livelihoods)) 3  
29 ((it, can), (can, be)) 3  
30 ((long, as), (as, we)) 3  
31 ((it, is), (is, easy)) 3  
32 ((difficult, to), (to, change)) 3  
33 ((change, it), (it, is)) 3  
34 ((to, change), (change, to)) 2  
35 ((change, is), (is, easier)) 2  
36 ((is, easier), (easier, because)) 2  
37 ((to, change), (change, if)) 2  
38 ((change, if), (if, there)) 2  
39 ((can, be), (be, changed)) 2  
40 ((from, natural), (natural, resources)) 2  
41 ((not, depend), (depend, on)) 2  
42 ((depend, on), (on, the)) 2  
43 ((on, the), (the, natural)) 2  
44 ((have, money), (money, for)) 2  
45 ((other, livelihoods), (livelihoods, we)) 2  
46 ((a, source), (source, of)) 2  
47 ((the, world), (world, is)) 2  
48 ((we, need), (need, to)) 2  
49 ((there, are), (are, some)) 2  
50 ((thus, changing), (changing, is)) 2  
51 ((do, not), (not, use)) 2  
52 ((use, natural), (natural, resources)) 2  
53 ((can, not), (not, be)) 2  
54 ((so, that), (that, the)) 2  
55 ((i, have), (have, some)) 2  
56 ((change, because), (because, of)) 2  
57 ((to, climate), (climate, change)) 2  
58 ((change, no), (no, answer)) 2  
59 ((the, livelihoods), (livelihoods, we)) 2  
60 ((depend, on), (on, them)) 2  
61 ((because, the), (the, livelihood)) 2  
62 ((resources, we), (we, can)) 2  
63 ((that, we), (we, can)) 2  
64 ((we, can), (can, do)) 2  
65 ((so, that), (that, we)) 2  
66 ((us, act), (act, in)) 2

In [330]:

CLE\_trigrams\_list = list(trigrams(CLE\_filtered\_tokens))  
#print(bigrams\_list)  
CLE\_trigram\_counts = Counter(zip(CLE\_trigrams\_list, CLE\_trigrams\_list[1:]))  
#print(bigram\_counts)   
CLE\_trigrams = pd.DataFrame(CLE\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CLE\_trigrams)

Word Frequency  
0 ((as, long, as), (long, as, there)) 6  
1 ((long, as, there), (as, there, is)) 4  
2 ((it, is, a), (is, a, challenge)) 4  
3 ((we, can, change), (can, change, the)) 3  
4 ((there, is, need), (is, need, to)) 3  
5 ((is, need, to), (need, to, change)) 3  
6 ((as, long, as), (long, as, i)) 3  
7 ((long, as, i), (as, i, have)) 3  
8 ((we, do, not), (do, not, have)) 3  
9 ((as, long, as), (long, as, we)) 3  
10 ((need, to, change), (to, change, to)) 2  
11 ((change, is, easier), (is, easier, because)) 2  
12 ((to, change, if), (change, if, there)) 2  
13 ((change, if, there), (if, there, is)) 2  
14 ((on, the, natural), (the, natural, resources)) 2  
15 ((a, source, of), (source, of, income)) 2  
16 ((to, change, because), (change, because, of)) 2  
17 ((resources, we, can), (we, can, change)) 2  
18 ((do, not, have), (not, have, money)) 2  
19 ((is, a, challenge), (a, challenge, to)) 2  
20 ((a, challenge, to), (challenge, to, change)) 2  
21 ((we, only, depend), (only, depend, on)) 2  
22 ((livelihood, as, long), (as, long, as)) 2  
23 ((long, as, we), (as, we, have)) 2  
24 ((thus, i, can), (i, can, not)) 2  
25 ((it, as, long), (as, long, as)) 2  
26 ((long, as, there), (as, there, are)) 2  
27 ((thus, it, is), (it, is, easy)) 2  
28 ((it, is, easy), (is, easy, to)) 2  
29 ((is, easy, to), (easy, to, change)) 2  
30 ((it, is, not), (is, not, easy)) 2  
31 ((is, not, easy), (not, easy, but)) 2  
32 ((not, easy, but), (easy, but, it)) 2  
33 ((easy, but, it), (but, it, is)) 2  
34 ((because, of, climate), (of, climate, change)) 2  
35 ((thus, it, is), (it, is, a)) 2  
36 ((change, as, long), (as, long, as)) 2  
37 ((given, to, us), (to, us, by)) 2  
38 ((it, is, easier), (is, easier, as)) 2  
39 ((is, easier, as), (easier, as, long)) 2  
40 ((easier, as, long), (as, long, as)) 2  
41 ((as, long, as), (long, as, you)) 2  
42 ((if, the, ses), (the, ses, are)) 1  
43 ((the, ses, are), (ses, are, taken)) 1  
44 ((ses, are, taken), (are, taken, care)) 1  
45 ((are, taken, care), (taken, care, of)) 1  
46 ((taken, care, of), (care, of, they)) 1  
47 ((care, of, they), (of, they, can)) 1  
48 ((of, they, can), (they, can, increase)) 1  
49 ((they, can, increase), (can, increase, in)) 1  
50 ((can, increase, in), (increase, in, number)) 1  
51 ((increase, in, number), (in, number, and)) 1  
52 ((in, number, and), (number, and, bring)) 1  
53 ((number, and, bring), (and, bring, income)) 1  
54 ((and, bring, income), (bring, income, the)) 1  
55 ((bring, income, the), (income, the, ses)) 1  
56 ((income, the, ses), (the, ses, can)) 1  
57 ((the, ses, can), (ses, can, be)) 1  
58 ((ses, can, be), (can, be, depleted)) 1  
59 ((can, be, depleted), (be, depleted, thus)) 1  
60 ((be, depleted, thus), (depleted, thus, there)) 1  
61 ((depleted, thus, there), (thus, there, need)) 1  
62 ((thus, there, need), (there, need, to)) 1  
63 ((there, need, to), (need, to, change)) 1  
64 ((to, change, to), (change, to, agriculture)) 1  
65 ((change, to, agriculture), (to, agriculture, like)) 1  
66 ((to, agriculture, like), (agriculture, like, goat)) 1

In [331]:

lemmatizer = WordNetLemmatizer()  
CLT = df7['Contributor\_Landscape\_Transformation\_Reasons'].str.lower().str.cat(sep=' ')  
CLT\_words = nltk.tokenize.word\_tokenize(CLT)  
CLT\_filtered\_tokens = [word for word in CLT\_words if len(CLT\_words) >= 4]  
CLT\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in CLT\_filtered\_tokens]  
CLT\_token\_counts = Counter(CLT\_lemmatized\_words)  
CLT\_columns = pd.DataFrame(CLT\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
#print(CLT\_columns)

In [332]:

CLT\_bigrams\_list = list(bigrams(CLT\_filtered\_tokens))  
#print(bigrams\_list)  
CLT\_bigram\_counts = Counter(zip(CLT\_bigrams\_list, CLT\_bigrams\_list[1:]))  
#print(bigram\_counts)   
CLT\_bigrams = pd.DataFrame(CLT\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CLT\_bigrams)

Word Frequency  
0 ((as, well), (well, as)) 8  
1 ((a, lot), (lot, of)) 6  
2 ((source, of), (of, income)) 5  
3 ((there, is), (is, no)) 4  
4 ((is, the), (the, source)) 4  
5 ((the, source), (source, of)) 4  
6 ((the, cutting), (cutting, down)) 4  
7 ((a, huge), (huge, area)) 4  
8 ((trees, for), (for, charcoal)) 4  
9 ((lot, of), (of, things)) 4  
10 ((the, fire), (fire, destroys)) 4  
11 ((it, destroys), (destroys, the)) 4  
12 ((when, the), (the, trees)) 4  
13 ((the, trees), (trees, are)) 4  
14 ((trees, are), (are, cut)) 4  
15 ((than, the), (the, others)) 3  
16 ((the, burning), (burning, of)) 3  
17 ((it, is), (is, a)) 3  
18 ((it, is), (is, the)) 3  
19 ((cutting, down), (down, of)) 3  
20 ((that, is), (is, the)) 3  
21 ((well, as), (as, the)) 3  
22 ((the, trees), (trees, get)) 3  
23 ((cutting, trees), (trees, for)) 3  
24 ((are, a), (a, lot)) 3  
25 ((destroys, a), (a, lot)) 3  
26 ((fire, destroys), (destroys, the)) 3  
27 ((destroys, the), (the, habitat)) 3  
28 ((the, trees), (trees, have)) 2  
29 ((requires, a), (a, huge)) 2  
30 ((a, huge), (huge, land)) 2  
31 ((the, agriculture), (agriculture, activities)) 2  
32 ((that, destroys), (destroys, the)) 2  
33 ((it, is), (is, for)) 2  
34 ((for, agriculture), (agriculture, purposes)) 2  
35 ((brings, income), (income, the)) 2  
36 ((the, land), (land, is)) 2  
37 ((the, trees), (trees, that)) 2  
38 ((is, a), (a, source)) 2  
39 ((a, source), (source, of)) 2  
40 ((source, of), (of, food)) 2  
41 ((in, the), (the, area)) 2  
42 ((of, income), (income, some)) 2  
43 ((the, soil), (soil, the)) 2  
44 ((change, the), (the, landscape)) 2  
45 ((the, landscape), (landscape, the)) 2  
46 ((huge, chucks), (chucks, of)) 2  
47 ((of, land), (land, for)) 2  
48 ((for, a), (a, livelihood)) 2  
49 ((a, livelihood), (livelihood, the)) 2  
50 ((the, animals), (animals, will)) 2  
51 ((animals, will), (will, not)) 2  
52 ((will, not), (not, have)) 2  
53 ((cutting, tree), (tree, for)) 2  
54 ((tree, for), (for, charcoal)) 2  
55 ((it, is), (is, difficult)) 2  
56 ((is, difficult), (difficult, for)) 2  
57 ((the, trees), (trees, dry)) 2  
58 ((when, they), (they, are)) 2  
59 ((they, are), (are, burnt)) 2  
60 ((the, fertility), (fertility, of)) 2  
61 ((fertility, of), (of, the)) 2  
62 ((of, the), (the, soil)) 2  
63 ((the, cutting), (cutting, of)) 2  
64 ((cutting, of), (of, trees)) 2  
65 ((of, trees), (trees, for)) 2  
66 ((burning, destroys), (destroys, the)) 2

In [333]:

CLT\_trigrams\_list = list(trigrams(CLT\_filtered\_tokens))  
#print(bigrams\_list)  
CLT\_trigram\_counts = Counter(zip(CLT\_trigrams\_list, CLT\_trigrams\_list[1:]))  
#print(bigram\_counts)   
CLT\_trigrams = pd.DataFrame(CLT\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(CLT\_trigrams)

Word Frequency  
0 ((is, the, source), (the, source, of)) 4  
1 ((the, cutting, down), (cutting, down, of)) 3  
2 ((as, well, as), (well, as, the)) 3  
3 ((a, lot, of), (lot, of, things)) 3  
4 ((destroys, a, lot), (a, lot, of)) 3  
5 ((when, the, trees), (the, trees, are)) 3  
6 ((the, trees, are), (trees, are, cut)) 3  
7 ((requires, a, huge), (a, huge, land)) 2  
8 ((it, is, a), (is, a, source)) 2  
9 ((is, a, source), (a, source, of)) 2  
10 ((the, source, of), (source, of, income)) 2  
11 ((source, of, income), (of, income, some)) 2  
12 ((that, is, the), (is, the, source)) 2  
13 ((for, a, livelihood), (a, livelihood, the)) 2  
14 ((the, animals, will), (animals, will, not)) 2  
15 ((animals, will, not), (will, not, have)) 2  
16 ((cutting, tree, for), (tree, for, charcoal)) 2  
17 ((it, is, difficult), (is, difficult, for)) 2  
18 ((when, they, are), (they, are, burnt)) 2  
19 ((the, fertility, of), (fertility, of, the)) 2  
20 ((fertility, of, the), (of, the, soil)) 2  
21 ((the, cutting, of), (cutting, of, trees)) 2  
22 ((cutting, of, trees), (of, trees, for)) 2  
23 ((of, trees, for), (trees, for, charcoal)) 2  
24 ((cutting, trees, for), (trees, for, charcoal)) 2  
25 ((there, are, a), (are, a, lot)) 2  
26 ((are, a, lot), (a, lot, of)) 2  
27 ((the, burning, of), (burning, of, the)) 2  
28 ((the, fire, destroys), (fire, destroys, the)) 2  
29 ((trees, are, cut), (are, cut, for)) 2  
30 ((are, cut, for), (cut, for, charcoal)) 2  
31 ((a, huge, area), (huge, area, and)) 2  
32 ((huge, area, and), (area, and, kills)) 2  
33 ((the, rainfall, will), (rainfall, will, reduce)) 2  
34 ((to, protect, the), (protect, the, environment)) 2  
35 ((destroys, the, habitat), (the, habitat, of)) 2  
36 ((it, destroys, everything), (destroys, everything, on)) 2  
37 ((destroys, everything, on), (everything, on, its)) 2  
38 ((the, trees, have), (trees, have, been)) 1  
39 ((trees, have, been), (have, been, depleted)) 1  
40 ((have, been, depleted), (been, depleted, and)) 1  
41 ((been, depleted, and), (depleted, and, it)) 1  
42 ((depleted, and, it), (and, it, has)) 1  
43 ((and, it, has), (it, has, caused)) 1  
44 ((it, has, caused), (has, caused, reduced)) 1  
45 ((has, caused, reduced), (caused, reduced, rainfall)) 1  
46 ((caused, reduced, rainfall), (reduced, rainfall, because)) 1  
47 ((reduced, rainfall, because), (rainfall, because, i)) 1  
48 ((rainfall, because, i), (because, i, have)) 1  
49 ((because, i, have), (i, have, to)) 1  
50 ((i, have, to), (have, to, clear)) 1  
51 ((have, to, clear), (to, clear, the)) 1  
52 ((to, clear, the), (clear, the, land)) 1  
53 ((clear, the, land), (the, land, for)) 1  
54 ((the, land, for), (land, for, agriculture)) 1  
55 ((land, for, agriculture), (for, agriculture, to)) 1  
56 ((for, agriculture, to), (agriculture, to, plant)) 1  
57 ((agriculture, to, plant), (to, plant, crops)) 1  
58 ((to, plant, crops), (plant, crops, hence)) 1  
59 ((plant, crops, hence), (crops, hence, the)) 1  
60 ((crops, hence, the), (hence, the, landscape)) 1  
61 ((hence, the, landscape), (the, landscape, change)) 1  
62 ((the, landscape, change), (landscape, change, there)) 1  
63 ((landscape, change, there), (change, there, would)) 1  
64 ((change, there, would), (there, would, a)) 1  
65 ((there, would, a), (would, a, reduction)) 1  
66 ((would, a, reduction), (a, reduction, in)) 1

In [334]:

lemmatizer = WordNetLemmatizer()  
LDL = df7['Landscape\_Depeneded\_Livelihood\_Reasons'].str.lower().str.cat(sep=' ')  
LDL\_words = nltk.tokenize.word\_tokenize(LDL)  
LDL\_filtered\_tokens = [word for word in LDL\_words if len(LDL\_words) >= 4]  
LDL\_lemmatized\_words = [lemmatizer.lemmatize(word) for word in LDL\_filtered\_tokens]  
LDL\_token\_counts = Counter(LDL\_lemmatized\_words)  
LDL\_columns = pd.DataFrame(LDL\_token\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
#print(LDL\_columns)

In [335]:

LDL\_bigrams\_list = list(bigrams(LDL\_filtered\_tokens))  
#print(bigrams\_list)  
LDL\_bigram\_counts = Counter(zip(LDL\_bigrams\_list, LDL\_bigrams\_list[1:]))  
#print(bigram\_counts)   
LDL\_bigrams = pd.DataFrame(LDL\_bigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(LDL\_bigrams)

Word Frequency  
0 ((where, we), (we, get)) 17  
1 ((thats, where), (where, we)) 16  
2 ((we, get), (get, food)) 16  
3 ((a, source), (source, of)) 12  
4 ((water, is), (is, life)) 11  
5 ((thats, were), (were, we)) 10  
6 ((were, we), (we, get)) 9  
7 ((we, are), (are, farmers)) 9  
8 ((as, well), (well, as)) 7  
9 ((get, food), (food, crops)) 7  
10 ((source, of), (of, income)) 6  
11 ((is, a), (a, source)) 6  
12 ((we, get), (get, our)) 6  
13 ((food, crops), (crops, thats)) 6  
14 ((our, food), (food, crops)) 6  
15 ((is, the), (the, source)) 5  
16 ((the, source), (source, of)) 5  
17 ((source, of), (of, food)) 5  
18 ((of, food), (food, crops)) 5  
19 ((that, is), (is, where)) 5  
20 ((is, where), (where, we)) 5  
21 ((crops, thats), (thats, where)) 5  
22 ((source, of), (of, livelihood)) 4  
23 ((it, is), (is, where)) 4  
24 ((food, crops), (crops, and)) 4  
25 ((crops, and), (and, income)) 4  
26 ((get, our), (our, food)) 4  
27 ((are, farmers), (farmers, thats)) 4  
28 ((it, is), (is, a)) 3  
29 ((is, life), (life, water)) 3  
30 ((life, water), (water, is)) 3  
31 ((source, of), (of, life)) 3  
32 ((for, our), (our, livelihoods)) 3  
33 ((food, crops), (crops, come)) 3  
34 ((crops, come), (come, from)) 3  
35 ((were, we), (we, grow)) 3  
36 ((is, life), (life, and)) 3  
37 ((crops, for), (for, our)) 3  
38 ((comes, from), (from, thats)) 3  
39 ((in, the), (the, forest)) 3  
40 ((where, we), (we, cultivate)) 3  
41 ((we, depend), (depend, on)) 3  
42 ((are, farmers), (farmers, by)) 3  
43 ((farmers, by), (by, nature)) 3  
44 ((crops, thats), (thats, were)) 3  
45 ((from, thats), (thats, where)) 3  
46 ((get, food), (food, for)) 3  
47 ((that, is), (is, were)) 3  
48 ((food, crops), (crops, we)) 3  
49 ((farmers, thats), (thats, where)) 3  
50 ((for, a), (a, livelihood)) 2  
51 ((a, livelihood), (livelihood, the)) 2  
52 ((there, is), (is, no)) 2  
53 ((that, is), (is, a)) 2  
54 ((food, for), (for, consumption)) 2  
55 ((thats, a), (a, source)) 2  
56 ((income, comes), (comes, from)) 2  
57 ((for, a), (a, living)) 2  
58 ((a, living), (living, it)) 2  
59 ((it, helps), (helps, us)) 2  
60 ((from, farming), (farming, we)) 2  
61 ((we, obtain), (obtain, food)) 2  
62 ((where, our), (our, food)) 2  
63 ((if, there), (there, is)) 2  
64 ((as, a), (a, source)) 2  
65 ((they, are), (are, a)) 2  
66 ((are, a), (a, source)) 2

In [336]:

LDL\_trigrams\_list = list(trigrams(LDL\_filtered\_tokens))  
#print(bigrams\_list)  
LDL\_trigram\_counts = Counter(zip(LDL\_trigrams\_list, LDL\_trigrams\_list[1:]))  
#print(bigram\_counts)   
LDL\_trigrams = pd.DataFrame(LDL\_trigram\_counts.most\_common(67),  
columns = ['Word', 'Frequency'])  
print(LDL\_trigrams)

Word Frequency  
0 ((thats, where, we), (where, we, get)) 14  
1 ((where, we, get), (we, get, food)) 11  
2 ((we, get, food), (get, food, crops)) 7  
3 ((is, a, source), (a, source, of)) 6  
4 ((thats, were, we), (were, we, get)) 6  
5 ((is, the, source), (the, source, of)) 5  
6 ((source, of, food), (of, food, crops)) 5  
7 ((were, we, get), (we, get, food)) 4  
8 ((where, we, get), (we, get, our)) 4  
9 ((we, get, our), (get, our, food)) 4  
10 ((food, crops, thats), (crops, thats, where)) 4  
11 ((crops, thats, where), (thats, where, we)) 4  
12 ((we, are, farmers), (are, farmers, thats)) 4  
13 ((a, source, of), (source, of, income)) 3  
14 ((it, is, a), (is, a, source)) 3  
15 ((water, is, life), (is, life, water)) 3  
16 ((is, life, water), (life, water, is)) 3  
17 ((a, source, of), (source, of, food)) 3  
18 ((food, crops, and), (crops, and, income)) 3  
19 ((food, crops, come), (crops, come, from)) 3  
20 ((thats, were, we), (were, we, grow)) 3  
21 ((water, is, life), (is, life, and)) 3  
22 ((that, is, where), (is, where, we)) 3  
23 ((we, are, farmers), (are, farmers, by)) 3  
24 ((are, farmers, by), (farmers, by, nature)) 3  
25 ((we, get, food), (get, food, for)) 3  
26 ((get, our, food), (our, food, crops)) 3  
27 ((for, a, livelihood), (a, livelihood, the)) 2  
28 ((the, source, of), (source, of, income)) 2  
29 ((that, is, a), (is, a, source)) 2  
30 ((thats, a, source), (a, source, of)) 2  
31 ((for, a, living), (a, living, it)) 2  
32 ((as, a, source), (a, source, of)) 2  
33 ((they, are, a), (are, a, source)) 2  
34 ((are, a, source), (a, source, of)) 2  
35 ((source, of, traditional), (of, traditional, medicine)) 2  
36 ((crops, come, from), (come, from, thats)) 2  
37 ((as, well, as), (well, as, income)) 2  
38 ((life, and, it), (and, it, is)) 2  
39 ((were, we, grow), (we, grow, crops)) 2  
40 ((life, water, is), (water, is, life)) 2  
41 ((as, well, as), (well, as, the)) 2  
42 ((is, where, we), (where, we, farm)) 2  
43 ((that, where, we), (where, we, get)) 2  
44 ((is, where, we), (where, we, cultivate)) 2  
45 ((we, depend, on), (depend, on, the)) 2  
46 ((food, crops, thats), (crops, thats, were)) 2  
47 ((comes, from, thats), (from, thats, where)) 2  
48 ((from, thats, where), (thats, where, we)) 2  
49 ((get, food, for), (food, for, eating)) 2  
50 ((that, is, were), (is, were, we)) 2  
51 ((is, were, we), (were, we, get)) 2  
52 ((animals, graze, in), (graze, in, the)) 2  
53 ((are, farmers, thats), (farmers, thats, were)) 2  
54 ((were, we, get), (we, get, our)) 2  
55 ((our, food, crops), (food, crops, thats)) 2  
56 ((get, food, crops), (food, crops, and)) 2  
57 ((crops, thats, were), (thats, were, we)) 2  
58 ((get, food, crops), (food, crops, we)) 2  
59 ((food, crops, we), (crops, we, are)) 2  
60 ((crops, we, are), (we, are, farmers)) 2  
61 ((are, farmers, thats), (farmers, thats, where)) 2  
62 ((farmers, thats, where), (thats, where, we)) 2  
63 ((from, we, are), (we, are, farmers)) 2  
64 ((he, cultivates, a), (cultivates, a, large)) 1  
65 ((cultivates, a, large), (a, large, area)) 1  
66 ((a, large, area), (large, area, of)) 1

[[Back to top](#Purpose-of-the-Analysis)]

## 11. Specific Variables[¶](#X41e2ce28de3d28ad12805bb21bb04dcc259b2a5)

### 11.1 Main Project and Cultural Practices[¶](#X585ebeff30804a13ebe7270211f64ca7006a3c8)

The responses are grouped in line with the main projects There is also the grouping of responses of those that agreed and storngly agreed in one dataframe as well as those whose responses were strongly disagreed and disagreed  
The text responses in form of reasons are tokenized and lammentized

In [337]:

CPH\_grouped = df2.groupby('Name\_Main\_Project')['Cultural\_Practices\_Hinder'].value\_counts(dropna=False)

In [338]:

CPH\_grouped

Out[338]:

Name\_Main\_Project Cultural\_Practices\_Hinder  
EbA\_CENTRAL\_MUCHINGA\_LUAPULA Strongly\_Agree\_Likert 2  
 Strongly\_Disagree\_Likert 1  
Ecosystem Conservation\_NORTH\_WESTERN Strongly\_Disagree\_Likert 5  
 Strongly\_Agree\_Likert 2  
 NaN 2  
 Agree\_Likert 1  
 Disagree\_Likert 1  
PIN\_WESTERN Strongly\_Disagree\_Likert 5  
 NaN 1  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN Disagree\_Likert 11  
 Strongly\_Disagree\_Likert 9  
 Agree\_Likert 5  
 Strongly\_Agree\_Likert 2  
 Undecided\_Likert 2  
 NaN 1  
SCReBS\_WESTERN Strongly\_Disagree\_Likert 14  
 Agree\_Likert 1  
SCRiKA\_LS Strongly\_Disagree\_Likert 20  
 Disagree\_Likert 6  
 NaN 6  
 Agree\_Likert 5  
 Strongly\_Agree\_Likert 5  
 Undecided\_Likert 2  
TRALARD\_LNM Strongly\_Disagree\_Likert 16  
 Strongly\_Agree\_Likert 10  
 Agree\_Likert 8  
 Disagree\_Likert 4  
 NaN 2  
Name: count, dtype: int64

In [339]:

CPH\_grouped1 = pd.DataFrame(CPH\_grouped)

In [340]:

CPH\_grouped1

Out[340]:

|  |  |  |
| --- | --- | --- |
|  |  | count |
| Name\_Main\_Project | Cultural\_Practices\_Hinder |  |
| EbA\_CENTRAL\_MUCHINGA\_LUAPULA | Strongly\_Agree\_Likert | 2 |
|  | Strongly\_Disagree\_Likert | 1 |
| Ecosystem Conservation\_NORTH\_WESTERN | Strongly\_Disagree\_Likert | 5 |
|  | Strongly\_Agree\_Likert | 2 |
|  | NaN | 2 |
|  | Agree\_Likert | 1 |
|  | Disagree\_Likert | 1 |
| PIN\_WESTERN | Strongly\_Disagree\_Likert | 5 |
|  | NaN | 1 |
| SCRALA\_SOUTHERN\_WESTERN\_NORTHEN | Disagree\_Likert | 11 |
|  | Strongly\_Disagree\_Likert | 9 |
|  | Agree\_Likert | 5 |
|  | Strongly\_Agree\_Likert | 2 |
|  | Undecided\_Likert | 2 |
|  | NaN | 1 |
| SCReBS\_WESTERN | Strongly\_Disagree\_Likert | 14 |
|  | Agree\_Likert | 1 |
| SCRiKA\_LS | Strongly\_Disagree\_Likert | 20 |
|  | Disagree\_Likert | 6 |
|  | NaN | 6 |
|  | Agree\_Likert | 5 |
|  | Strongly\_Agree\_Likert | 5 |
|  | Undecided\_Likert | 2 |
| TRALARD\_LNM | Strongly\_Disagree\_Likert | 16 |
|  | Strongly\_Agree\_Likert | 10 |
|  | Agree\_Likert | 8 |
|  | Disagree\_Likert | 4 |
|  | NaN | 2 |

In [341]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert", "NaN"]  
ax = sns.barplot(data = CPH\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Practices\_Hinder", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 13: Number of Responses on cultural Practices Hindering Sustainable Management of SES in the Main Projects", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [342]:

agreement\_levels = ["Agree\_Likert", "Strongly\_Agree\_Likert"]  
CPH\_R = df2[df2["Cultural\_Practices\_Hinder"].isin(agreement\_levels)]

In [343]:

CPH\_R1 = CPH\_R.drop(CPH\_R.columns[[0,1,2,3,4,6,7,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CPH\_R1grouped = CPH\_R1.groupby('Name\_Main\_Project')['Cultural\_Practices\_Hinder']  
#CPH\_R1

In [344]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CPH\_R1.iterrows():  
 CPH\_R1\_filter\_sentence = []  
 CPH\_R1\_sentence = row["Cultural\_Practices\_Hinder\_Reason"]  
 if pd.isnull(CPH\_R1\_sentence):  
 continue  
 CPH\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CPH\_R1\_sentence)  
 CPH\_R1\_words = nltk.word\_tokenize(CPH\_R1\_sentence\_cleaned)  
 CPH\_R1\_words = [lemmatizer.lemmatize(w) for w in CPH\_R1\_words if w.lower() not in stop\_words]  
 CPH\_R1\_filter\_sentence.extend(CPH\_R1\_words)  
 print(CPH\_R1\_filter\_sentence)

['need', 'harvest', 'tree', 'fully', 'grown', 'applies', 'biodiversity']  
['crop', 'rotation', 'made', 'land', 'cultivating', 'ensure', 'fertility', 'soil']  
['allowed', 'cultivate', 'area', '3', 'year', 'shifting', 'another', 'area']  
['rule', 'chief', 'ensure', 'harvesting', 'period', 'followed', 'well', 'period', 'burning', 'bush']  
['norm', 'bush', 'burned', 'dry', 'bush', 'completely', 'burn', 'also', 'support', 'quick', 'regermination', 'vegetation', 'tree']  
['nothing']  
['never', 'heard', 'learnt']  
['cutting', 'tree', 'along', 'river', 'allowed']  
['act', 'late', 'burning', 'destroys', 'environment', 'people', 'community', 'would', 'want', 'catch', 'Catapilars', 'consumption', 'sale']  
['long', 'time', 'ago', 'people', 'poor', 'management', 'natural', 'moment', 'practice', 'maintain', 'biodiversity']  
['cultural', 'practice', 'future', 'generation', 'know', 'anything']  
['customary', 'practice', 'promote', 'early', 'burning', 'people', 'tend', 'burn', 'late']  
['unregulated', 'allocation', 'parcel', 'land', 'forest']  
['Ba', 'chipupila', 'customary', 'practice', 'protecting', 'natural', 'resource']  
['place', 'grave', 'yard', 'protected', 'deforestation']  
['land', 'natural', 'resource', 'located', 'owned', 'traditional', 'authority']  
['Chitemene', 'system', 'destroys', 'environment']  
['experienced', 'cultural', 'practice', 'interfering', 'forest', 'management']  
['old', 'parent', 'taught', 'u', 'protect', 'environment', 'like', 'customary', 'practice']  
['rule', 'made', 'customary', 'practice', 'followed']  
['place', 'allow', 'cutting', 'tree', 'well', 'cutting', 'fruit', 'bearing', 'tree']  
['protection', 'environment', 'resource', 'depleted']  
['cutting', 'tree', 'anyhow']  
['teach', 'u', 'protect', 'environment', 'preventing', 'Chitemene', 'system']  
['always', 'teach', 'people', 'community', 'protect', 'environment']  
['normally', 'give', 'rule', 'protect', 'environment']  
['traditional', 'method', 'harvesting', 'poaching', 'burning', 'cutting', 'tree']  
['people', 'cut', 'tree', 'thus', 'leading', 'wild', 'animal', 'lacking', 'sleep', 'shelter']  
['practice', 'called', 'Malende', 'protect', 'certain', 'area', 'prohibit', 'tree', 'cut']  
['Hynas', 'eat', 'livestock', 'kill', 'conflict', 'ZAWA', 'Officers']  
['ownership', 'land', 'Chief', 'river', 'give', 'power', 'destroy', 'area', 'giving', 'cultivation', 'activity']  
['cultural', 'practice', 'prevent', 'rain', 'falling', 'Malende', 'disturbed']  
['bad', 'fishing', 'method', 'well', 'people', 'settling', 'game', 'park']  
['lack', 'support', 'WDCS', 'CRB', 'local', 'community', 'protect', 'environment', 'license', 'given', 'investor', 'cut', 'tree', 'community', 'benefit', 'sale', 'tree', 'tradition', 'authority', 'benefit']  
['people', 'still', 'want', 'lead', 'life', 'hunting', 'use', 'mosquito', 'net', 'catching', 'fish']  
['traditional', 'leader', 'prohibit', 'people', 'making', 'decision', 'sell', 'land', 'people', 'lead', 'destruction', 'environment', 'even', 'people', 'agree', 'headman', 'headman', 'say', 'land', 'sell', 'want']  
['cultural', 'activity', 'getting', 'root', 'tree', 'medicine', 'destroy', 'tree']  
['culture', 'someone', 'live', 'well', 'need', 'cultivate', 'huge', 'parcel', 'land', 'thus', 'leading', 'cutting', 'tree']  
['Chiefs', 'asking', 'people', 'stop', 'living', 'along', 'river', 'bank', 'using', 'mosquito', 'net', 'fish', 'community', 'adhearing']

In [345]:

CPH\_R1["Cultural\_Practices\_Hinder\_Reason"] = CPH\_R1["Cultural\_Practices\_Hinder\_Reason"].fillna("")   
CPH\_R1["Cultural\_Practices\_Hinder\_Reason"] = CPH\_R1["Cultural\_Practices\_Hinder\_Reason"].astype(str)  
CPH\_R1\_Text = " ".join(CPH\_R1["Cultural\_Practices\_Hinder\_Reason"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CPH\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Cultural Practices Hinder", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [346]:

agreement\_levels = ["Disagree\_Likert", "Strongly\_Disagree\_Likert"]  
CPH\_R2 = df2[df2["Cultural\_Practices\_Hinder"].isin(agreement\_levels)]

In [347]:

CPH\_R3 = CPH\_R2.drop(CPH\_R2.columns[[0,1,2,3,4,6,7,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CPH\_R3grouped = CPH\_R3.groupby('Name\_Main\_Project')['Cultural\_Practices\_Hinder']  
#CPH\_R3

In [348]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CPH\_R3.iterrows():  
 CPH\_R3\_filter\_sentence = []  
 CPH\_R3\_sentence = row["Cultural\_Practices\_Hinder\_Reason"]  
 if pd.isnull(CPH\_R3\_sentence):  
 continue  
 CPH\_R3\_sentence\_cleaned = re.sub(r'[^\w\s]','',CPH\_R3\_sentence)  
 CPH\_R3\_words = nltk.word\_tokenize(CPH\_R3\_sentence\_cleaned)  
 CPH\_R3\_words = [lemmatizer.lemmatize(w) for w in CPH\_R3\_words if w.lower() not in stop\_words]  
 CPH\_R3\_filter\_sentence.extend(CPH\_R3\_words)  
 print(CPH\_R3\_filter\_sentence)

['never', 'happed', 'life', 'time']  
['customary', 'practice', 'hinder', 'sustainable', 'management']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['cultural', 'history', 'within', 'community', 'except', 'governmnet']  
['cultural', 'norm']  
['nothing']  
['nothing']  
['nothing']  
['customary', 'practice', 'protect', 'environment', 'like', 'way', 'ZAWA', 'protects', 'biodiversity']  
['chitemene', 'system', 'customary', 'practice', 'destroys', 'environment']  
['cultural', 'practice', 'burn', 'bush', 'certain', 'period', 'harvesting']  
['cultural', 'practice']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothinhg']  
['traditional', 'practice', 'government', 'guide', 'u']  
['nothing']  
['nothing']  
['nothing', 'SES', 'looked', 'game', 'park', 'officer', 'officer', 'mandated']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['traditional', 'way', 'management', 'stopped', 'government', 'taken', 'management', 'forest', 'biodiversity']  
['nothing']  
['long', 'time', 'malende', 'used', 'protect', 'tree', 'moment', 'taking', 'place']  
['nothing']  
['nothing']  
['nothing']  
['heard', 'anything', 'like']  
['nothing', 'know', 'hinder']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['thing']  
['nothing']  
['nothing']  
['nothing']  
['Thee', 'nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['hill', 'called', 'Omba', 'release', 'smoke', 'indicating', 'particular', 'year', 'would', 'rainfall', 'evergthing', 'would', 'fine', 'regard', 'rainfall']  
['following', 'law']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['nothing']

In [349]:

CPH\_R3["Cultural\_Practices\_Hinder\_Reason"] = CPH\_R3["Cultural\_Practices\_Hinder\_Reason"].fillna("")   
CPH\_R3["Cultural\_Practices\_Hinder\_Reason"] = CPH\_R3["Cultural\_Practices\_Hinder\_Reason"].astype(str)  
CPH\_R3\_Text = " ".join(CPH\_R3["Cultural\_Practices\_Hinder\_Reason"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CPH\_R3\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Cultural Practices Hinder", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.2 Main Project and Cultural Practices Changed[¶](#X412a565bacb1bb2b582a0d0bbbd500f6db562dc)

In [350]:

CPC\_grouped = df2.groupby('Name\_Main\_Project')['Cultural\_Practices\_Changed'].value\_counts(dropna=False)

In [351]:

CPC\_grouped

Out[351]:

Name\_Main\_Project Cultural\_Practices\_Changed  
EbA\_CENTRAL\_MUCHINGA\_LUAPULA Strongly\_Agree\_Likert 1  
 Strongly\_Disagree\_Likert 1  
 Undecided\_Likert 1  
Ecosystem Conservation\_NORTH\_WESTERN NaN 5  
 Strongly\_Agree\_Likert 4  
 Strongly\_Disagree\_Likert 2  
PIN\_WESTERN Strongly\_Disagree\_Likert 4  
 Strongly\_Agree\_Likert 1  
 NaN 1  
SCRALA\_SOUTHERN\_WESTERN\_NORTHEN Agree\_Likert 11  
 Disagree\_Likert 7  
 Strongly\_Disagree\_Likert 4  
 Undecided\_Likert 4  
 Strongly\_Agree\_Likert 3  
 NaN 1  
SCReBS\_WESTERN Strongly\_Disagree\_Likert 6  
 Strongly\_Agree\_Likert 5  
 Agree\_Likert 4  
SCRiKA\_LS Agree\_Likert 11  
 Strongly\_Agree\_Likert 10  
 NaN 10  
 Strongly\_Disagree\_Likert 6  
 Disagree\_Likert 5  
 Undecided\_Likert 2  
TRALARD\_LNM Strongly\_Disagree\_Likert 18  
 Disagree\_Likert 10  
 Strongly\_Agree\_Likert 5  
 Agree\_Likert 3  
 Undecided\_Likert 2  
 NaN 2  
Name: count, dtype: int64

In [352]:

CPC\_grouped1 = pd.DataFrame(CPC\_grouped)

In [353]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = CPC\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Practices\_Changed", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 14: Number of Responses on Willingnes to Change Cultural Practices for Sustainable Management of SES in the Main Projects", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.3 Main project and Cultural Aspects Considered[¶](#X04afd0462544a54832a8b2d4aae596ff4ef9878)

In [354]:

CAC\_grouped = df2.groupby('Name\_Main\_Project')['Cultural\_Aspects\_Considered'].value\_counts(dropna=False)

In [355]:

CAC\_grouped1 = pd.DataFrame(CAC\_grouped)

In [356]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = CAC\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Aspects\_Considered", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 15: Number of Responses on if Cultural Aspects are Considered for Sustainable Management of SES in the Main Projects", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [357]:

agreement\_levels = ["Agree\_Likert", "Strongly\_Agree\_Likert"]  
CAC\_R = df2[df2["Cultural\_Aspects\_Considered"].isin(agreement\_levels)]  
CAC\_R1 = CAC\_R.drop(CPH\_R.columns[[0,1,2,3,4,6,7,8,9,10,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CAC\_R1grouped = CAC\_R1.groupby('Name\_Main\_Project')['Cultural\_Aspects\_Considered']  
#CAC\_R1

In [358]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CAC\_R1.iterrows():  
 CAC\_R1\_filter\_sentence = []  
 CAC\_R1\_sentence = row["Cultural\_Aspects\_Considered\_Reasons"]  
 if pd.isnull(CAC\_R1\_sentence):  
 continue  
 CAC\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CAC\_R1\_sentence)  
 CAC\_R1\_words = nltk.word\_tokenize(CAC\_R1\_sentence\_cleaned)  
 CAC\_R1\_words = [lemmatizer.lemmatize(w) for w in CAC\_R1\_words if w.lower() not in stop\_words]  
 CAC\_R1\_filter\_sentence.extend(CAC\_R1\_words)  
 print(CAC\_R1\_filter\_sentence)

['need', 'power', 'formulated', 'ownership']  
['time', 'getting', 'land', 'project', 'come', 'regulation']  
['rule', 'along', 'river', 'bank', 'tree', 'need', 'cut']  
['sometimes', 'give', 'Chiefs', 'harvest', 'collect', 'produce', 'accounted']  
['nothing', 'thing', 'ended', 'long', 'time', 'ago', 'Chiefs', 'would', 'talk', 'spirit', 'protect', 'land']  
['included']  
['customary', 'practice', 'natural', 'resource', 'get', 'destroyed']  
['answer']  
['answer']  
['done', 'product', 'project', 'work', 'well']  
['follow', 'customary', 'practice', 'people', 'burn', 'late', 'inorder']  
['livelihood', 'improve']  
['follow', 'historical', 'practice', 'forefather']  
['cultural', 'practice', 'spririal', 'rite', 'done', 'traditional', 'authority']  
['accessing', 'land', 'traditional', 'authority', 'also', 'give', 'instruction', 'put', 'livelihood', 'close', 'water', 'source', 'resource']  
['project', 'operate', 'without', 'consulting', 'traditional', 'authority']  
['answer']  
['Sometimes', 'discus', 'locally', 'animal', 'going', 'looked']  
['always', 'follow', 'law', 'taught', 'u']  
['destroy', 'environment', 'customary', 'practice']  
['changed', 'would', 'like', 'environment', 'protected', 'generation', 'see', 'future']  
['Previously', 'people', 'used', 'cut', 'tree', 'anyhow', 'change']  
['cultural', 'aspect', 'considered', 'within', 'CFMG']  
['Even', 'Bible', 'say', 'forget', 'root']  
['always', 'follow', 'told']  
['consider', 'project', 'would', 'work', 'well', 'instance', 'Livingstone', 'community', 'destroyed', 'cultural', 'activity', 'northwestern', 'came', 'different', 'region']  
['Borehole', 'sank', 'near', 'grave', 'yard', 'funeral', 'community', 'hold', 'meeting', 'village']  
['group', 'mission', 'like', 'group', 'give', 'money', 'orphan', 'profit', 'make']  
[]  
[]  
['answer']  
['considered', 'thatched', 'roof', 'using', 'pole', 'cattle', 'craw', 'cultural', 'aspect', 'difficult', 'change', 'unless', 'people', 'enough', 'money']  
['project', 'bettering', 'life']  
['taught']  
['traditional', 'leader', 'accept', 'certain', 'project', 'done', 'specific', 'place']  
['included', 'induna']  
['indunas', 'send', 'representative', 'learn', 'accept', 'project', 'community']

In [359]:

CAC\_R1["Cultural\_Aspects\_Considered\_Reasons"] = CAC\_R1["Cultural\_Aspects\_Considered\_Reasons"].fillna("")   
CAC\_R1["Cultural\_Aspects\_Considered\_Reasons"] = CAC\_R1["Cultural\_Aspects\_Considered\_Reasons"].astype(str)  
CAC\_R1\_Text = " ".join(CAC\_R1["Cultural\_Aspects\_Considered\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CAC\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Cultural Aspects Considered", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [360]:

agreement\_levels = ["Disagree\_Likert", "Strongly\_Disagree\_Likert"]  
CAC\_R2 = df2[df2["Cultural\_Aspects\_Considered"].isin(agreement\_levels)]  
CAC\_R3 = CAC\_R2.drop(CPH\_R2.columns[[0,1,2,3,4,6,7,8,9,10,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CAC\_R3grouped = CAC\_R3.groupby('Name\_Main\_Project')['Cultural\_Aspects\_Considered']  
#CAC\_R3

In [361]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CAC\_R3.iterrows():  
 CAC\_R3\_filter\_sentence = []  
 CAC\_R3\_sentence = row["Cultural\_Aspects\_Considered\_Reasons"]  
 if pd.isnull(CAC\_R3\_sentence):  
 continue  
 CAC\_R3\_sentence\_cleaned = re.sub(r'[^\w\s]','',CAC\_R3\_sentence)  
 CAC\_R3\_words = nltk.word\_tokenize(CAC\_R3\_sentence\_cleaned)  
 CAC\_R3\_words = [lemmatizer.lemmatize(w) for w in CAC\_R3\_words if w.lower() not in stop\_words]  
 CAC\_R3\_filter\_sentence.extend(CAC\_R3\_words)  
 print(CAC\_R3\_filter\_sentence)

['cultural', 'history', 'area']  
['cultural', 'aspect']  
['follow', 'chief', 'government', 'say']  
['nothing']  
['nothing']  
['nothing']  
['nothing']  
['plan', 'given', 'u', 'TRALARD', 'consideration', 'cultural', 'history', 'formulating', 'project']  
['nothing']  
['moment', 'follow', 'law', 'government', 'cultural', 'norm']  
['nothing']  
['ask', 'headman', 'give', 'u', 'place', 'keep', 'goat', 'place', 'acceptable', 'traditional', 'law']  
['nothing', 'moment', 'traditional', 'leader', 'spritual', 'activity', 'project']  
['include', 'cultural', 'practice']  
['project', 'move', 'well', 'headman', 'project', 'otherwise', 'included', 'might', 'problem']  
['available']  
['consider']  
['nothing']  
['nothing']  
[]  
['nothing']  
[]  
['unneccesary']  
['anything']  
[]  
['nothing']  
['include']  
[]  
['nothing']  
['include', 'aspect']  
['nothing']  
['nothing', 'included']  
['Nothing', 'considered']  
['nothing']  
[]  
['Nothing', 'considered']  
['nothing']  
['included']  
['include']  
['nothing', 'included']  
['nothing']  
['included']  
['Nothing']  
['Nothing']  
['nothing']  
['control', 'Chief', 'make', 'change']  
['nothing', 'included']  
['taken', 'consideration']  
['project', 'destroyed', 'culture']  
['nothing']  
[]  
['nothing']  
['project', 'associated', 'cultural', 'aspect']  
['Thee', 'nothing']  
['nothing']  
[]  
['considered']  
['nothing']  
['nothing']  
['nothing']  
['project', 'come', 'BRE']  
['part', 'project']  
[]  
[]  
['Nothing']  
[]  
['nothing']  
['follow', 'teaching']  
['cultural', 'aspect', 'included']  
['cultural', 'aspect', 'included']  
[]  
[]  
['included']  
['include']  
['nothing']  
['nothing']  
[]

In [362]:

CAC\_R3["Cultural\_Aspects\_Considered\_Reasons"] = CAC\_R3["Cultural\_Aspects\_Considered\_Reasons"].fillna("")   
CAC\_R3["Cultural\_Aspects\_Considered\_Reasons"] = CAC\_R3["Cultural\_Aspects\_Considered\_Reasons"].astype(str)  
CAC\_R3\_Text = " ".join(CAC\_R3["Cultural\_Aspects\_Considered\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CAC\_R3\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Cultural Aspects Considered", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.4 Main Project and Measure of Indicators[¶](#X7f8a2c24b50fe6d2fc4599a9cee4c8ad1df1d50)

In [363]:

MI\_grouped = df2.groupby('Name\_Main\_Project')['Measure\_Indicators'].value\_counts(dropna=False)

In [364]:

MI\_grouped1 = pd.DataFrame(MI\_grouped)

In [365]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = MI\_grouped1, x="count", y="Name\_Main\_Project", hue="Measure\_Indicators", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 16: Number of Responses on if Measurement of Indicators is important for Sustainable Management of SES in the Main Projects", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.5 Main Project and Livelihood Dependent[¶](#Xf0d1ed621b1e66bb6071f3f668e6f3b2ee29739)

In [366]:

LD\_grouped = df2.groupby('Name\_Main\_Project')['Livilihood\_Depenedent'].value\_counts(dropna=False)

In [367]:

LD\_grouped1 = pd.DataFrame(LD\_grouped)

In [368]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = LD\_grouped1, x="count", y="Name\_Main\_Project", hue="Livilihood\_Depenedent", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 17: Number of Responses in each of the main Project on if Livelihoods Depend on SES", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.6 Main Project and Livelihood Changed[¶](#Xf19c96fe02efdb424098909cca21c9c9a6a41d6)

In [369]:

CL\_grouped = df2.groupby('Name\_Main\_Project')['Change\_Livelihood'].value\_counts(dropna=False)

In [370]:

CL\_grouped1 = pd.DataFrame(CL\_grouped)

In [371]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = CL\_grouped1, x="count", y="Name\_Main\_Project", hue="Change\_Livelihood", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 18: Number of Responses in each of the main Project on if Livelihoods can be Changed", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.7 Main Project and Livelihood Changing Easy[¶](#Xee36b2194a5c27c2009b74491df4682b68d5247)

In [372]:

CLE\_grouped = df2.groupby('Name\_Main\_Project')['Change\_Livelihood\_Easy'].value\_counts(dropna=False)

In [373]:

CLE\_grouped1 = pd.DataFrame(CLE\_grouped)

In [374]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = CLE\_grouped1, x="count", y="Name\_Main\_Project", hue="Change\_Livelihood\_Easy", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 19: Number of Responses in each of the main Project on if Livelihoods can be Changed Easily", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [375]:

agreement\_levels = ["Agree\_Likert", "Strongly\_Agree\_Likert"]  
CLE\_R = df2[df2["Change\_Livelihood\_Easy"].isin(agreement\_levels)]  
CLE\_R1 = CLE\_R.drop(CLE\_R.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CLE\_R1grouped = CLE\_R1.groupby('Name\_Main\_Project')['Change\_Livelihood\_Easy']  
#CLE\_R1

In [376]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CLE\_R1.iterrows():  
 CLE\_R1\_filter\_sentence = []  
 CLE\_R1\_sentence = row["Change\_Livelihood\_Easy\_Reasons"]  
 if pd.isnull(CLE\_R1\_sentence):  
 continue  
 CLE\_R1\_sentence\_cleaned = re.sub(r'[^\w\s]','',CLE\_R1\_sentence)  
 CLE\_R1\_words = nltk.word\_tokenize(CLE\_R1\_sentence\_cleaned)  
 CLE\_R1\_words = [lemmatizer.lemmatize(w) for w in CLE\_R1\_words if w.lower() not in stop\_words]  
 CLE\_R1\_filter\_sentence.extend(CLE\_R1\_words)  
 print(CLE\_R1\_filter\_sentence)

['SES', 'taken', 'care', 'increase', 'number', 'bring', 'income']  
['SeS', 'depleted', 'thus', 'need', 'change', 'agriculture', 'like', 'goat', 'rearing']  
['Change', 'easier', 'one', 'decides', 'change']  
['thing', 'like', 'climate', 'change', 'affecting', 'u', 'thus', 'thought', 'change']  
['new', 'improvement', 'like', 'cooking', 'stove', 'change', 'way', 'livelihood']  
['destruction', 'SES']  
['need', 'change', 'cutting', 'activity']  
['need', 'change', 'activity', 'destroy', 'environment']  
['changed', 'learning']  
['kept', 'well', 'project', 'like', 'chicken', 'goat', 'would', 'help', 'depend', 'natural', 'resource']  
['fish', 'pond', 'project']  
['also', 'improve', 'livelihood', 'future']  
['long', 'money', 'livelihood']  
['want', 'venture', 'farming', 'reason', 'shifted', 'place', 'located']  
['sub', 'project', 'bee', 'keeping', 'disturbed', 'late', 'burning', 'early', 'burning', 'disturb', 'fire', 'much']  
['need', 'change', 'instance', 'depend', 'water', 'Lulimala', 'river', 'dry', 'get', 'water']  
['natural', 'resource', 'protected', 'accessed']  
['use', 'according', 'accepted', 'norm']  
['Yes', 'normally', 'use', 'natural', 'resource', 'somehow']  
['environment', 'protected', 'business']  
['easier', 'change', 'experience', 'natural', 'resource', 'change', 'accordance', 'climate', 'change']  
['change', 'slowly', 'due', 'low', 'performance', 'group']  
['depend']  
['would', 'difficult', 'time', 'would', 'change', 'gradually']  
['need', 'change', 'world', 'developing', 'hence', 'move']  
['keeping', 'goat', 'help', 'protecting', 'environment', 'livelihood', 'depend']  
['source', 'income']  
['government', 'support', 'u', 'turn', 'protect', 'environment']  
['people', 'teaching', 'new', 'thing', 'learn']  
['thing', 'difficult', 'look', 'instance', 'honey', 'bee', 'mushroom', 'may', 'difficult', 'find']  
['protected', 'resource', 'change']  
['long', 'help', 'government']  
['farming']  
['project', 'help', 'u', 'improve', 'livelihood']  
['change', 'way', 'depending', 'cutting', 'tree', 'focus', 'producing', 'honey', 'livelihood']  
['knowledge', 'make', 'u', 'act', 'certain', 'way', 'currently', 'future', 'might', 'new', 'knowledge', 'would', 'make', 'u', 'act', 'different', 'way', 'initial', 'one']  
['earning', 'natural', 'resource', 'develop']  
['change', 'easier', 'depent', 'people', 'easily', 'changed']  
['livelihood', 'plan', 'small', 'livestock', 'CFMG', 'well', 'garden']  
['source', 'earning', 'living']  
['management', 'forest', 'capacity', 'buildiing']  
['climate', 'change', 'force', 'change', 'livelihood']  
['depend', 'domecticated', 'animal']  
['long', 'support', 'somewhere']  
['resource', 'change', 'livelihood']  
['long', 'depend', 'livelihood']  
['enough', 'water', 'inland', 'depeneding', 'river', 'shore', 'cultivation', 'purpose', 'livelihood', 'would', 'change']  
['Thing', 'evolving', 'thus', 'stagnant']  
['long', 'helped', 'altrenative', 'livelihood']  
['trying', 'shift', 'make', 'garden', 'livelihood']  
['way', 'thing', 'climate', 'change', 'call', 'change']  
['long', 'different', 'alternative', 'livelihood']  
['Yes', 'depending', 'catle', 'looking', 'thus', 'easy', 'change']  
['profit', 'one', 'livelihood', 'change', 'another', 'type', 'livelihood']  
['easy', 'important', 'change', 'climate', 'change', 'change', 'without', 'taking', 'alternative']  
['livelihood', 'depends', 'farming', 'main', 'activity']  
['Changing', 'difficult', 'get', 'used', 'fine']  
['always', 'depending', 'farming', 'thus', 'change', 'would', 'lead', 'u', 'access', 'currently']  
['always', 'focused', 'agriculture', 'thus', 'little', 'bit', 'difficult', 'change']  
['answer']  
['would', 'prefer', 'shift', 'gardening', 'activity']  
['change', 'lead', 'better', 'life', 'easy']  
['guideline', 'assist', 'changing', 'livelihood']  
['know', 'livelihood', 'would', 'change', 'might', 'worse', 'current', 'one']  
['changed', 'dependency', 'climate', 'change']  
['long', 'advantage', 'disadvantage', 'well', 'technology', 'allow', 'done']  
['depend', 'forest', 'wetland']  
['changing', 'better', 'one']  
['Life', 'hard', 'due', 'high', 'cost', 'commodity']  
['problem', 'boreholes']  
['easy', 'good', 'thing']  
['way', 'live', 'adapt', 'environment', 'like', 'climate', 'change', 'thus', 'adapted', 'challenge']  
['find', 'someone', 'assist', 'u', 'change']  
['knowledge', 'use', 'thing', 'given', 'u', 'government', 'live', 'good', 'life']  
['law', 'ask', 'u', 'change']  
['long', 'help', 'somewhere']  
['easier', 'long', 'time', 'process']  
['long', 'use', 'change']  
['limited', 'responsibility', 'thus', 'easy', 'change']  
['period', 'climate', 'change', 'call', 'different', 'way', 'thing']  
['empowered', 'easy']  
['use', 'knowledge', 'adquately', 'easier']  
['long', 'need', 'change', 'well', 'climate', 'change']  
['easier', 'long', 'commitment']  
['person', 'make', 'decision', 'looking', 'back', 'done', 'make', 'corrective', 'measure']  
['long', 'committed']  
['difficult', 'long', 'follow', 'taught', 'change']  
['long', 'capacity']

In [377]:

CLE\_R1["Change\_Livelihood\_Easy\_Reasons"] = CLE\_R1["Change\_Livelihood\_Easy\_Reasons"].fillna("")   
CLE\_R1["Change\_Livelihood\_Easy\_Reasons"] = CLE\_R1["Change\_Livelihood\_Easy\_Reasons"].astype(str)  
CLE\_R1\_Text = " ".join(CLE\_R1["Change\_Livelihood\_Easy\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CLE\_R1\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Change Livelihood Easy", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

In [378]:

agreement\_levels = ["Disagree\_Likert", "Strongly\_Disagree\_Likert"]  
CLE\_R2 = df2[df2["Change\_Livelihood\_Easy"].isin(agreement\_levels)]  
CLE\_R3 = CLE\_R2.drop(CLE\_R2.columns[[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,23,24,25,26,27,28,29,30,31,32,33,34]], axis = 1)  
CLE\_R3grouped = CLE\_R3.groupby('Name\_Main\_Project')['Change\_Livelihood\_Easy']  
#CLE\_R3

In [379]:

lemmatizer=WordNetLemmatizer()  
stop\_words = set(stopwords.words('english'))  
for index, row in CLE\_R3.iterrows():  
 CLE\_R3\_filter\_sentence = []  
 CLE\_R3\_sentence = row["Change\_Livelihood\_Easy\_Reasons"]  
 if pd.isnull(CLE\_R3\_sentence):  
 continue  
 CLE\_R3\_sentence\_cleaned = re.sub(r'[^\w\s]','',CLE\_R3\_sentence)  
 CLE\_R3\_words = nltk.word\_tokenize(CLE\_R3\_sentence\_cleaned)  
 CLE\_R3\_words = [lemmatizer.lemmatize(w) for w in CLE\_R3\_words if w.lower() not in stop\_words]  
 CLE\_R3\_filter\_sentence.extend(CLE\_R3\_words)  
 print(CLE\_R3\_filter\_sentence)

['livelihood', 'would', 'improve']  
['everything', 'use', 'come', 'natural', 'resource', 'like', 'tree', 'building', 'animal', 'protein']  
['source', 'income']  
['assist', 'adequate', 'water', 'tree', 'cut', 'well', 'future', 'general', 'see', 'natural', 'resource']  
['world', 'becoming', 'mordenised', 'thus', 'need', 'adapt', 'current', 'status']  
['Thats', 'income', 'come']  
['depend', 'natural', 'resource', 'thus', 'changing', 'easy']  
['use', 'natural', 'resource']  
['depend', 'agriculture', 'natural', 'resource', 'like', 'forest']  
['mostly', 'use', 'goat', 'pig', 'livelihood']  
['livelihood', 'engage', 'keeping', 'goat', 'fish', 'farming']  
['livelihood', 'like', 'keeping', 'goat', 'chicken', 'gardening', 'hiring', 'wedding', 'dress']  
['use', 'domesticated', 'animal']  
['use']  
['thing', 'learning', 'already']  
['money', 'protection', 'environment', 'yet', 'money', 'carbon', 'trade', 'waiting', 'long', 'time']  
['used']  
['changing', 'another', 'lifestyle', 'mean', 'starting', 'new', 'life']  
['currently', 'drought', 'thus', 'difficult', 'change']  
['natural', 'resource', 'given', 'u', 'God', 'depend', 'root', 'tree', 'medicine', 'fruit']  
['one', 'activity', 'farming', 'thus', 'changing', 'difficult']  
['livelihood', 'based', 'farming']  
['always', 'use', 'product', 'forest']  
['livelihood', 'difficult', 'moment', 'climate', 'change']  
['problem', 'thus', 'change']  
['person', 'change', 'unless', 'person', 'shown']  
['manage', 'live', 'without', 'depending', 'forest', 'like', 'craw', 'use', 'tree']  
['starting', 'point', 'difficult', 'change', 'need', 'finance']  
['livelihood', 'shifting']  
['livelihood']  
['sure', 'future', 'thus', 'cannaot', 'change']  
['challenge']  
['firewood', 'depend', 'tree', 'thus', 'challenge', 'change', 'Agriculture', 'mean', 'cutting', 'tree', 'building', 'house', 'depends', 'tree', 'Also', 'piggery', 'project', 'done', 'market', 'thus', 'project', 'effective']  
['difficult', 'money', 'pig', 'keeping', 'challenge', 'looking', 'die']  
['person', 'leading', 'better', 'life', 'change']  
['old', 'age', 'thus', 'change', 'livelihood']  
['continue', 'trying']

In [380]:

CLE\_R3["Change\_Livelihood\_Easy\_Reasons"] = CLE\_R3["Change\_Livelihood\_Easy\_Reasons"].fillna("")   
CLE\_R3["Change\_Livelihood\_Easy\_Reasons"] = CLE\_R3["Change\_Livelihood\_Easy\_Reasons"].astype(str)  
CLE\_R3\_Text = " ".join(CLE\_R1["Change\_Livelihood\_Easy\_Reasons"])  
wordcloud = WordCloud(background\_color = "white", width = 1000, height = 400).generate(CLE\_R3\_Text)  
plt.figure(figsize=(20, 10))  
plt.imshow(wordcloud, interpolation="bilinear")  
plt.title("Figure 20: Change Livelihood Easy", loc="left", fontsize=20, pad=20)  
plt.axis("off")   
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.8 Main Project and Ecosystem Service Reduction[¶](#Xc71f2041e0473ad50f2b3073dfab4648c659ad4)

In [381]:

ESR\_grouped = df2.groupby('Name\_Main\_Project')['Ecosystem\_Services\_Reduced'].value\_counts(dropna=False)

In [382]:

ESR\_grouped1 = pd.DataFrame(ESR\_grouped)

In [383]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = ESR\_grouped1, x="count", y="Name\_Main\_Project", hue="Ecosystem\_Services\_Reduced", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 20: Number of Responses in each of the main Project on if Ecosystem services have Reduced", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.9 Main project and Deforestation Increase[¶](#Xe304be4e595b140172e863b01a73713bc0ca87f)

In [384]:

DI\_grouped = df2.groupby('Name\_Main\_Project')['Deforestaion\_Increased'].value\_counts(dropna=False)

In [385]:

DI\_grouped1 = pd.DataFrame(DI\_grouped)

In [386]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = DI\_grouped1, x="count", y="Name\_Main\_Project", hue="Deforestaion\_Increased", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 21: Number of Responses in each of the main Project on if Deforestation has Increased", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.10 Main project and Protected Areas as a Hinderance[¶](#X4a0521fe89d860a4b6011efed3efc8feb293a0f)

In [387]:

PAHL\_grouped = df2.groupby('Name\_Main\_Project')['Protected\_Areas\_Hinderarnce\_Livelihood'].value\_counts(dropna=False)

In [388]:

PAHL\_grouped1 = pd.DataFrame(PAHL\_grouped)

In [389]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = PAHL\_grouped1, x="count", y="Name\_Main\_Project", hue="Protected\_Areas\_Hinderarnce\_Livelihood", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 22: Number of Responses in each of the main Project on if Protected Areas are a Hinderance to Livelihood", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.11 Main Project and New Livelihood Projects[¶](#X7f5c52c88463dd3f3e314bd682cf04825ab5971)

In [390]:

NLP\_grouped = df2.groupby('Name\_Main\_Project')['New\_Livelihood\_Projects'].value\_counts(dropna=False)

In [391]:

NLP\_grouped1 = pd.DataFrame(NLP\_grouped)

In [392]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = NLP\_grouped1, x="count", y="Name\_Main\_Project", hue="New\_Livelihood\_Projects", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 23: Number of Responses in each of the main Project on if some Livelihood Subprojects not Implemented", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()

![No description has been provided for this image](data:image/png;base64;base64,)

### 11.12 Main Project and Sustainability of Subprojects[¶](#Xb53b1dbcad267823d8b2b2db9a9269011099864)

In [393]:

SSC\_grouped = df2.groupby('Name\_Main\_Project')['Subprojects\_Sustainability\_Contribution'].value\_counts(dropna=False)

In [394]:

SSC\_grouped1 = pd.DataFrame(SSC\_grouped)

In [395]:

plt.figure(figsize=(8.7, 8.27))  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax = sns.barplot(data = SSC\_grouped1, x="count", y="Name\_Main\_Project", hue="Subprojects\_Sustainability\_Contribution", hue\_order=hue\_order, legend=True)  
ax.set\_title("Figure 24: Number of Responses in each of the main Project on if Subprojects Contribute to Sustainability", fontsize=14)  
plt.legend(title="KEY")  
for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", padding=3)  
plt.show()  
#plt.saveax("charts.png", dpi=300)

![No description has been provided for this image](data:image/png;base64;base64,)

[[Back to top](#Purpose-of-the-Analysis)]

In [ ]:

fig = plt.figure(figsize=(30, 60))  
gs = GridSpec(6, 6, figure=fig)  
ax1 = fig.add\_subplot(gs[0, :3]) # Row 0, columns 0-2  
ax2 = fig.add\_subplot(gs[0, 3:]) # Row 0, columns 3-6  
ax3 = fig.add\_subplot(gs[1, :3]) # Row 1, columns 0-1  
ax4 = fig.add\_subplot(gs[1, 3:]) # Row 1, columns 2-4  
ax5 = fig.add\_subplot(gs[2, :3]) # Row 1, columns 5-6  
ax6 = fig.add\_subplot(gs[2, 3:]) # Row 2, columns 0-2  
ax7 = fig.add\_subplot(gs[3, :3]) # Row 2, columns 3-6  
ax8 = fig.add\_subplot(gs[3, 3:]) # Row 3, columns 0-3  
ax9 = fig.add\_subplot(gs[4, :3]) # Row 3, columns 4-6  
ax10 = fig.add\_subplot(gs[4, 3:]) # Row 4, columns 0-1  
ax11 = fig.add\_subplot(gs[5, :3]) # Row 4, columns 2-4  
ax12 = fig.add\_subplot(gs[5, 3:]) # Row 4, columns 5-6  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert", "NaN"]  
ax1 = sns.barplot(data = CPH\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Practices\_Hinder", hue\_order=hue\_order, legend=True, ax=ax1)  
ax1.set\_title("Figure 2: Number of Responses on cultural Practices Hindering Sustainable Management of SES in the Main Projects", fontsize=14)  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax2 = sns.barplot(data = CPC\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Practices\_Changed", hue\_order=hue\_order, legend=True, ax=ax2)  
ax2.set\_title("Figure 3: Number of Responses on Willingnes to Change Cultural Practices for Sustainable Management of SES in the Main Projects", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax3 = sns.barplot(data = CAC\_grouped1, x="count", y="Name\_Main\_Project", hue="Cultural\_Aspects\_Considered", hue\_order=hue\_order, legend=True, ax=ax3)  
ax3.set\_title("Figure 4: Number of Responses on if Cultural Aspects are Considered for Sustainable Management of SES ", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax4 = sns.barplot(data = MI\_grouped1, x="count", y="Name\_Main\_Project", hue="Measure\_Indicators", hue\_order=hue\_order, legend=True, ax=ax4)  
ax4.set\_title("Figure 5: Number of Responses on if Measurement of Indicators is important for Sustainable Management of SES", fontsize=14)  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax5 = sns.barplot(data = LD\_grouped1, x="count", y="Name\_Main\_Project", hue="Livilihood\_Depenedent", hue\_order=hue\_order, legend=True, ax=ax5)  
ax5.set\_title("Figure 6: Number of Responses in each of the main Project on if Livelihoods Depend on SES", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax6 = sns.barplot(data = CL\_grouped1, x="count", y="Name\_Main\_Project", hue="Change\_Livelihood", hue\_order=hue\_order, legend=True, ax=ax6)  
ax6.set\_title("Figure 7: Number of Responses in each of the main Project on if Livelihoods can be Changed", fontsize=14)  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax7 = sns.barplot(data = CLE\_grouped1, x="count", y="Name\_Main\_Project", hue="Change\_Livelihood\_Easy", hue\_order=hue\_order, legend=True, ax=ax7)  
ax7.set\_title("Figure 8: Number of Responses in each of the main Project on if Livelihoods can be Changed Easily", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax8 = sns.barplot(data = ESR\_grouped1, x="count", y="Name\_Main\_Project", hue="Ecosystem\_Services\_Reduced", hue\_order=hue\_order, legend=True, ax=ax8)  
ax8.set\_title("Figure 9: Number of Responses in each of the main Project on if Ecosystem services have Reduced", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax9 = sns.barplot(data = DI\_grouped1, x="count", y="Name\_Main\_Project", hue="Deforestaion\_Increased", hue\_order=hue\_order, legend=True, ax=ax9)  
ax9.set\_title("Figure 10: Number of Responses in each of the main Project on if Deforestation has Increased", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax10 = sns.barplot(data = PAHL\_grouped1, x="count", y="Name\_Main\_Project", hue="Protected\_Areas\_Hinderarnce\_Livelihood", hue\_order=hue\_order, legend=True, ax=ax10)  
ax10.set\_title("Figure 11: Number of Responses in each of the main Project on if Protected Areas are a Hinderance to Livelihood", fontsize=14)  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax11 = sns.barplot(data = NLP\_grouped1, x="count", y="Name\_Main\_Project", hue="New\_Livelihood\_Projects", hue\_order=hue\_order, legend=True, ax=ax11)  
ax11.set\_title("Figure 12: Number of Responses in each of the main Project on if some Livelihood Subprojects not Implemented", fontsize=14)  
  
  
hue\_order = ["Strongly\_Disagree\_Likert", "Disagree\_Likert", "Undecided\_Likert", "Agree\_Likert", "Strongly\_Agree\_Likert"]  
ax12 = sns.barplot(data = SSC\_grouped1, x="count", y="Name\_Main\_Project", hue="Subprojects\_Sustainability\_Contribution", hue\_order=hue\_order, legend=True, ax=ax12)  
ax12.set\_title("Figure 13: Number of Responses in each of the main Project on if Subprojects Contribute to Sustainability", fontsize=14)  
  
plt.tight\_layout()  
plt.savefig("12charts.png", dpi=300)  
plt.savefig("12graph.jpg")  
plt.show()

## 12. Converting the Notebook[¶](#X892bfffef340f7a4b7a78f2a4187307a68e2b40)

In [ ]:

with open('02\_Landscape\_Transformation\_Livelihood.ipynb', 'r') as f:  
 notebook = nbformat.read(f, as\_version=4)  
  
# Initialize the HTML Exporter  
html\_exporter = HTMLExporter()  
(body, resources) = html\_exporter.from\_notebook\_node(notebook)  
  
# Save the HTML output  
with open('02\_Landscape\_Transformation\_Livelihood.html', 'w') as f:  
 f.write(body)  
  
print("Conversion to HTML completed!")

[[Back to top](#Purpose-of-the-Analysis)]

## 13. Converting to Word Document[¶](#X6ff35b1cddadc4c47f7792bf000cb01f040956e)

In [210]:

#!pandoc 02\_Landscape\_Transformation\_Livelihood.md -o output.docx  
!pandoc 02\_Landscape\_Transformation\_Livelihood.html -o D:/DataAnalysis/LandscapeTransformationLivelihood.docx

UTF-8 decoding error in 02\_Landscape\_Transformation\_Livelihood.html at byte offset 243172 (95).  
The input must be a UTF-8 encoded text.

In [209]:

#subprocess.run(["pandoc", "02\_Landscape\_Transformation\_Livelihood.md", "-o", "02\_Landscape\_Transformation\_Livelihood.docx"])  
#subprocess.run(["pandoc", "02\_Landscape\_Transformation\_Livelihood.md", "-o", "C:/Users/nazin/Data\_Analysis/02\_Landscape\_Transformation.docx"])  
subprocess.run(["pandoc", r"02\_Landscape\_Transformation\_Livelihood.html", "-o", r"C:/Users/nazin/Data\_Analysis/02\_Landscape\_Transformation.docx"], capture\_output=True, text=True)  
print("Conversion successful! File saved as 02\_Landscape\_Transformation\_Livelihood.docx")

Conversion successful! File saved as 02\_Landscape\_Transformation\_Livelihood.docx

In [207]:

print(shutil.which("pandoc"))

C:\Users\nazin\AppData\Local\anaconda3\envs\NLTK\_Py\_3\_12\Scripts\pandoc.EXE

In [ ]: