In [174]:
```
import pandas as pd
```

In [175]:
```
box_placement = pd.read_excel('../data/Data Analyst Assessment Test.xlsx', sheet_
box_placement.head()
```

Out[175]:

	client_name	distance_to_water_body	enumerator_comment	expected_harvest_date	farmer_in_lis
0	A	0-5KM	Done	2022-05-27	
1	A	More_than_5_KM	successfuly done	2022-05-27	c
2	A	More_than_5_KM	ok	2022-06-08	c
3	A	More_than_5_KM	ok	2022-06-06	c
4	A	More_than_5_KM	ok	2022-06-16	c

5 rows × 81 columns

In [176]:
```
box_placement.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 81 columns):
 #   Column                                                Non-Null
Count  Dtype
---  ------                                                --------
------  -----
 0   client_name                                           2356 non
-null   object
 1   distance_to_water_body                                2297 non
-null   object
 2   enumerator_comment                                    2356 non
-null   object
 3   expected_harvest_date                                 2297 non
-null   datetime64[ns]
 4   farmer_in_list_or_control                             2356 non
-null   object
 5   farmer_state_district                                 2356 non
-null   object
 6   field_irrigated                                       2297 non
-null   object
 7   insured_crop                                          2356 non
-null   object
 8   intercropping                                         2356 non
-null   object
 9   other_factors_that_affected_germination               106 non-
null    object
 10  planting_date                                         2356 non
-null   datetime64[ns]
 11  success_box1                                          2356 non
-null   object
 12  success_box2                                          2297 non
-null   object
 13  ward_lga_subcounty_camp                               2356 non
-null   object
 14  box1_crop_condition                                   2297 non
-null   object
 15  box1_crop_stands_per_square_meter                     982 non-
null    float64
 16  box1_dim_8_by_5                                       2297 non
-null   object
 17  box1_length                                           2297 non
-null   float64
 18  box1_mode_of_planting                                 2244 non
-null   object
 19  box1_other_issues_occurrence_date                     29 non-n
ull     datetime64[ns]
 20  box1_other_problems                                   19 non-n
ull     object
 21  box1_problem                                          1176 non
-null   object
 22  box1_secondary_issues                                 1250 non
-null   object
 23  box1_width                                            2297 non
-null   float64
```

```
 24  box2_crop_condition                                                 2292 non
-null   object
 25  box2_crop_stands_per_square_meter                                   987 non-
null    float64
 26  box2_dim_8_by_5                                                     2292 non
-null   object
 27  box2_length                                                         2292 non
-null   float64
 28  box2_mode_of_planting                                               2242 non
-null   object
 29  box2_other_issues_occurrence_date                                   26 non-n
ull     datetime64[ns]
 30  box2_other_problems                                                 16 non-n
ull     object
 31  box2_problem                                                        1153 non
-null   object
 32  box2_secondary_issues                                               1212 non
-null   object
 33  box2_width                                                          2292 non
-null   float64
 34  @case_id                                                            2356 non
-null   object
 35  timeEnd                                                             2356 non
-null   object
 36  timeStart                                                           2356 non
-null   object
 37  type_of_pests_or_diseases_current_crop_stage                        28 non-n
ull     object
 38  what_steps_did_the_farmer_take_to_mitigate_the_pests_or_diseases  28 non-n
ull     object
 39  why_unable_to_place_box1                                            59 non-n
ull     object
 40  type_of_irrigation_system                                           1528 non
-null   object
 41  box1_rows                                                           1262 non
-null   float64
 42  measurement_box1_row1                                               1262 non
-null   float64
 43  measurement_box1_row2                                               1261 non
-null   float64
 44  measurement_box1_row3                                               1251 non
-null   float64
 45  box2_rows                                                           1255 non
-null   float64
 46  measurement_box2_row1                                               1255 non
-null   float64
 47  measurement_box2_row2                                               1253 non
-null   float64
 48  measurement_box2_row3                                               1240 non
-null   float64
 49  box1_drought_occurence_date                                         217 non-
null    datetime64[ns]
 50  box1_locust_infestation_date                                        82 non-n
ull     datetime64[ns]
 51  box2_drought_occurence_date                                         209 non-
null    datetime64[ns]
 52  box2_locust_infestation_date                                        83 non-n
```

```
ull     datetime64[ns]
 53  box1_other_pest_disease_occurence_date                 166 non-
null    datetime64[ns]
 54  box1_pests_or_diseases_mitigation                      148 non-
null    object
 55  box1_type_of_pests_or_diseases                         148 non-
null    object
 56  box2_other_pest_disease_occurence_date                 153 non-
null    datetime64[ns]
 57  box2_pests_and_diseases_mitigation                     136 non-
null    object
 58  box2_type_of_pests_or_diseases                         136 non-
null    object
 59  how_is_the_other_crop_planted                          230 non-
null    object
 60  other_crops_names                                      230 non-
null    object
 61  box1_animal_encroachment_date                          55 non-n
ull     datetime64[ns]
 62  box2_animal_encroachment_date                          56 non-n
ull     datetime64[ns]
 63  causes_of_weeds_box2                                   99 non-n
ull     object
 64  weeds_mitigation_box2                                  99 non-n
ull     object
 65  other_crops_not_listed                                 9 non-nu
ll      object
 66  causes_of_weeds_box1                                   106 non-
null    object
 67  weeds_mitigation_box1                                  106 non-
null    object
 68  why_unable_to_place_box2                               5 non-nu
ll      object
 69  username                                               2356 non
-null   object
 70  box_placement_comment                                  2356 non
-null   object
 71  herbicide_applied_box1                                 3 non-nu
ll      object
 72  box1_cause_of_flood                                    6 non-nu
ll      object
 73  box1_flood_occurence_date                              6 non-nu
ll      datetime64[ns]
 74  box2_flood_occurence_date                              4 non-nu
ll      datetime64[ns]
 75  other_causes_of_weeds_box1                             2 non-nu
ll      object
 76  other_causes_of_weeds_box2                             0 non-nu
ll      float64
 77  latitude                                               2356 non
-null   float64
 78  longitude                                              2356 non
-null   float64
 79  altitude                                               2356 non
-null   float64
 80  accuracy                                               2356 non
-null   float64
```

```
dtypes: datetime64[ns](14), float64(19), object(48)
memory usage: 1.5+ MB
```

In [177]:
```
wet_harvest = pd.read_excel('../data/Data Analyst Assessment Test.xlsx', sheet_na
wet_harvest.head()
```

Out[177]:

	@case_id	box1_harvest_possible	box2_harvest_possible	enumerator_comment	farmer_ve
0	9d1a878b-ea56-423a-83a5-dce4eee21302	yes	yes	ok	farmer_ve
1	ebf9955c-d21e-434a-80d7-658a2cecd0bf	yes	yes	ok	farmer_ve
2	eb5f0cf1-0814-48af-94be-62b5dc2524a9	yes	yes	ok	farmer_ve
3	79880c35-bb12-4089-8f06-3927b6c44875	yes	yes	ok	farmer_ve
	5d8d55c6-				

In [178]:
```
wet_harvest.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2234 entries, 0 to 2233
Data columns (total 24 columns):
 #   Column                              Non-Null Count  Dtype  
---  ------                              --------------  -----  
 0   @case_id                            2234 non-null   object 
 1   box1_harvest_possible               2137 non-null   object 
 2   box2_harvest_possible               2135 non-null   object 
 3   enumerator_comment                  2234 non-null   object 
 4   farmer_verified                     2234 non-null   object 
 5   box1_wet_weight                     2131 non-null   float64
 6   box1_wet_weight_confirmation        2131 non-null   float64
 7   box2_wet_weight                     2129 non-null   float64
 8   box2_wet_weight_confirmation        2129 non-null   float64
 9   timeEnd                             2234 non-null   object 
 10  timeStart                           2234 non-null   object 
 11  cannot_proceed_with_wet_harvest     48 non-null     object 
 12  box_1_non_compliance_confirmation   5 non-null      object 
 13  why_unable_to_capture_box1_weight   6 non-null      object 
 14  box_2_non_compliance_confirmation   5 non-null      object 
 15  why_unable_to_capture_box2_weight   6 non-null      object 
 16  why_no_crop_survived_in_box1        1 non-null      object 
 17  why_no_crop_survived_in_box2        1 non-null      object 
 18  username                            2234 non-null   object 
 19  wet_harvest_comment                 2234 non-null   object 
 20  latitude                            2134 non-null   float64
 21  longitude                           2134 non-null   float64
 22  altitude                            2134 non-null   float64
 23  accuracy                            2134 non-null   float64
dtypes: float64(8), object(16)
memory usage: 419.0+ KB
```

In [179]:
```
dry_harvest = pd.read_excel('../data/Data Analyst Assessment Test.xlsx', sheet_na
dry_harvest.head()
```

Out[179]:

	@case_id	did_the_farmer_keep_the_crops_in_separate_bags	enumerator_comment	farmer_ve
0	9d1a878b-ea56-423a-83a5-dce4eee21302	yes	ok	farmer_ve
1	ebf9955c-d21e-434a-80d7-658a2cecd0bf	yes	ok	farmer_ve
2	eb5f0cf1-0814-48af-94be-62b5dc2524a9	yes	ok	farmer_ve
3	79880c35-bb12-4089-8f06-3927b6c44875	yes	ok	farmer_ve
4	5d8d55c6-9514-495f-a629-21fd17b92c77	yes	ok	farmer_ve

In [180]:
```
dry_harvest.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2222 entries, 0 to 2221
Data columns (total 18 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   @case_id                                            2222 non-null   object
 1   did_the_farmer_keep_the_crops_in_separate_bags      2102 non-null   object
 2   enumerator_comment                                  2222 non-null   object
 3   farmer_verified                                     2222 non-null   object
 4   box1_dry_weight                                     2096 non-null   float6
4
 5   box1_dry_weight_confirmation                        2096 non-null   float6
4
 6   box2_dry_weight                                     2093 non-null   float6
4
 7   box2_dry_weight_confirmation                        2093 non-null   float6
4
 8   was_anything_added_or_removed_from_the_harvest_bag  2098 non-null   object
 9   timeEnd                                             2222 non-null   object
 10  timeStart                                           2222 non-null   object
 11  cannot_proceed_with_dry_harvest                     120 non-null    object
 12  username                                            2222 non-null   object
 13  dry_harvest_comment                                 2222 non-null   object
 14  latitude                                            2102 non-null   float6
4
 15  longitude                                           2102 non-null   float6
4
 16  altitude                                            2102 non-null   float6
4
 17  accuracy                                            2102 non-null   float6
4
dtypes: float64(8), object(10)
memory usage: 312.6+ KB
```

Now, I am going to place the '@case_id' feature at the beginning of the box_pacement dataframe and attach the name of the dataframes to this feature for the 3 dataframe so I can identify the beginning of the each dataframe when using the info() method after joining the 3 dataframes

In [181]:
```
cols = ['@case_id'] + [col for col in box_placement.columns if col != '@case_id']
box_placement = box_placement[cols]
```

In [182]:
```
box_placement.rename({'@case_id':'box_plc_@case_id'},axis=1, inplace=True)
```

In [183]:
```
wet_harvest.rename({'@case_id':'wet_@case_id'},axis=1, inplace=True)
```

In [184]:
```
dry_harvest.rename({'@case_id':'dry_@case_id'},axis=1, inplace=True)
```

Task 1: Merging the 3 Dataframes

In [185]:
```
'box_plc_@case_id' in  box_placement.columns,'wet_@case_id' in  wet_harvest.colum
```

Out[185]:
```
(True, True, True)
```

In [186]:
```
data = box_placement.merge(wet_harvest,left_on = 'box_plc_@case_id',right_on='wet
       merge(dry_harvest, left_on='box_plc_@case_id', right_on='dry_@case_id', s
```

In [187]:
```
name_dict = {'timeEnd':'timeEnd_dry','timeStart':'timeStart_dry',
             'latitude':'latitude_dry','longitude':'longitude_dry',
             'altitude':'altitude_dry','accuracy':'accuracy_dry','username':'user
data = data.rename(name_dict, axis=1)
```

In [188]:
```
sum(data['username_box_plc'] != data['username_wet']),sum(data['username_box_plc'
```

Out[188]:
```
(0, 0)
```

In [189]:
```
sum(~((data['box_plc_@case_id']==data['wet_@case_id']) | (data['box_plc_@case_id'
```

Out[189]:
```
0
```

Type *Markdown* and LaTeX: α^2

In [190]:
```
# data = data.drop(['wet_@case_id','dry_@case_id','username_wet','username_dry'],
```

In [191]:
```
data.info(verbose=1)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2187 entries, 0 to 2186
Data columns (total 123 columns):
 #    Column                                                  Dtype
---   ------                                                  -----
 0    box_plc_@case_id                                        objec
t
 1    client_name                                             objec
t
 2    distance_to_water_body                                  objec
t
 3    enumerator_comment_box_plc                              objec
t
 4    expected_harvest_date                                   datet
ime64[ns]
 5    farmer_in_list_or_control                               objec
t
 6    farmer_state_district                                   objec
t
```

Task 2: Checking for Incorrect Data

Dimension

In [192]:
```
len(data)
```

Out[192]:
```
2187
```

In [193]:
```
data['box1_dim_8_by_5'].value_counts(dropna=False)
```

Out[193]:
```
yes    2150
NaN      37
Name: box1_dim_8_by_5, dtype: int64
```

In [194]:
```
data['box1_length'].value_counts(dropna=False)
```

Out[194]:
```
8.0    2150
NaN      37
Name: box1_length, dtype: int64
```

In [195]:
```
data['box1_width'].value_counts(dropna=False)
```

Out[195]:
```
5.0    2150
NaN      37
Name: box1_width, dtype: int64
```

In [196]:
```
sum(data['box1_dim_8_by_5'].isna().index != data['box1_length'].isna().index)
```

Out[196]:
```
0
```

In [197]:
```
sum(data['box1_dim_8_by_5'].isna().index != data['box1_width'].isna().index)
```

Out[197]:
```
0
```

In total, there are 37 records without values for box1 dimensions. when the is no value for 'box1_dim_8_by_5', there is no vlaues for 'box1_length' or 'box1_width'

In [198]:
```
print('The following enumerators regisetred no value for box1 dimensions at least
data[data['box1_dim_8_by_5'].isna()]['username_box_plc'].value_counts().\
reset_index().rename({'index':'enumerator','username_box_plc':'count'},axis=1)
```

```
The following enumerators regisetred no value for box1 dimensions at least once
```

Out[198]:

	enumerator	count
0	nig024	11
1	nig175	9
2	nig137	5
3	nig035	5
4	nig186	3
5	nig033	3
6	nig021	1

In [199]:
```
data['box2_dim_8_by_5'].value_counts(dropna=False)
```

Out[199]:
```
yes    2147
NaN      40
Name: box2_dim_8_by_5, dtype: int64
```

In [200]:
```
data['box2_length'].value_counts(dropna=False)
```

Out[200]:
```
8.0    2147
NaN      40
Name: box2_length, dtype: int64
```

In [201]:
```
data['box2_width'].value_counts(dropna=False)
```

Out[201]:
```
5.0    2147
NaN      40
Name: box2_width, dtype: int64
```

In [202]:
```
sum(data['box2_dim_8_by_5'].isna().index != data['box2_length'].isna().index)
```

Out[202]:
```
0
```

In [203]:
```
sum(data['box2_dim_8_by_5'].isna().index != data['box2_width'].isna().index)
```

Out[203]:
```
0
```

In total, there are 40 records without values for box2 dimensions. when the is no value for

'box2_dim_8_by_5', there is no vlaues for 'box2_length' or 'box2_width'

In [204]:
```
print('The following enumerators regisetred no value for box2 dimensions at least
data[data['box2_dim_8_by_5'].isna()]['username_box_plc'].value_counts().\
reset_index().rename({'index':'enumerator','username_box_plc':'count'},axis=1)
```

```
The following enumerators regisetred no value for box2 dimensions at least once
```

Out[204]:

	enumerator	count
0	nig024	11
1	nig175	9
2	nig137	7
3	nig035	5
4	nig186	3
5	nig033	3
6	nig118	1
7	nig021	1

Except for missing values, things are normal with the dimensions columns

Zero Yields

The relevant columns are : ul

- "wet_box1_wet_weight".
- "wet_box1_wet_weight_confirmation".
- "wet_box2_wet_weight".
- "wet_box2_wet_weight_confirmation".
- "dry_box1_dry_weight".
- "dry_box2_dry_weight_confirmation".
- "dry_box2_dry_weight".
- "dry_box2_dry_weight_confirmation".

wet harvest

In [205]:
```
sum(data['box1_wet_weight'] != data['box1_wet_weight_confirmation'])
```

Out[205]: 87

In [206]:
```
data[(data['box1_wet_weight'] != data['box1_wet_weight_confirmation']) & (data['b
```

Out[206]:

	box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There are 87 times where the wieght of box1 in wet harvesting is null

In [207]:
```
sum(data['box2_wet_weight'] != data['box2_wet_weight_confirmation'])
```

Out[207]:
```
90
```

In [208]:
```
data[(data['box2_wet_weight'] != data['box2_wet_weight_confirmation']) & (data['b
```

Out[208]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There are 90 times where the wieght of box2 in wet harvesting is null

In [209]:
```
wet_box1_null = data['box1_wet_weight'].isna()
wet_box2_null = data['box2_wet_weight'].isna()
print('The follwoing enumerator registered null for their box1 or box2 wet harves
data[wet_box1_null & wet_box2_null]['username_box_plc'].value_counts().\
    reset_index().rename({'index':'enumerator','username_box_plc':'count'},axis=1
```

```
The follwoing enumerator registered null for their box1 or box2 wet harvest wei
ghts at least once
```

Out[209]:

	enumerator	count
0	nig168	21
1	nig024	11
2	nig175	9
3	nig137	7
4	nig114	6
5	nig148	5
6	nig035	5
7	nig033	3
8	nig118	3
9	nig186	3
10	nig187	2
11	nig188	2
12	nig022	2
13	nig103	2
14	nig174	1
15	nig178	1
16	nig109	1
17	nig021	1

dry harvest

In [210]:
```
sum(data['box1_dry_weight'] != data['box1_dry_weight_confirmation'])
```

Out[210]: 91

In [211]:
```
data[(data['box1_dry_weight'] != data['box1_dry_weight_confirmation']) & (data['b
```

Out[211]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There are 91 times where the wieght of box1 in dry harvesting is null

In [212]:
```
sum(data['box2_dry_weight'] != data['box2_dry_weight_confirmation'])
```

Out[212]: 94

In [213]:
```
data[(data['box2_dry_weight'] != data['box2_dry_weight_confirmation']) & (data['b
```

Out[213]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There are 94 times where the wieght of box2 in dry harvesting is null

In [214]:
```
dry_box1_null = data['box1_dry_weight'].isna()
dry_box2_null = data['box2_dry_weight'].isna()
print('The follwoing enumerators registered null for their box1 or box2 dry harve
data[dry_box1_null & dry_box2_null]['username_box_plc'].value_counts().\
    reset_index().rename({'index':'enumerator','username_box_plc':'count'},axis=1
```

```
The follwoing enumerators registered null for their box1 or box2 dry harvest we
ights at least once
```

Out[214]:

	enumerator	count
0	nig168	21
1	nig024	11
2	nig175	9
3	nig137	8
4	nig114	6
5	nig035	5
6	nig148	5
7	nig187	3
8	nig033	3
9	nig174	3
10	nig186	3
11	nig118	3
12	nig022	2
13	nig103	2
14	nig188	2
15	nig109	1
16	nig178	1
17	nig021	1

Non compliant dataset:

- Box dimension not captured yet there is yield data captured.
- Zero wet weight yield but greater than zero dry weight.
- Harvest crop mixed with other crops.

- Box dimension not captured yet there is yield data captured.

In [215]:
```
dim_cond1 = data['box1_length'].isna()
dim_cond2 = data['box1_width'].isna()
dim_cond3 = data['box2_length'].isna()
dim_cond4 = data['box2_width'].isna()
# dim_cond = (cond1 | cond2 | cond3 | cond4)

weight_cond1 = data['box1_wet_weight'].notna()
weight_cond2 = data['box2_wet_weight'].notna()
weight_cond3 = data['box1_dry_weight'].notna()
weight_cond4 = data['box2_dry_weight'].notna()
```

In [216]:
```
data[(dim_cond1|dim_cond2) & (weight_cond1|weight_cond3)]
```

Out[216]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

In [217]:
```
data[(dim_cond3|dim_cond4) & (weight_cond2|weight_cond4)]
```

Out[217]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There is no record where the box dimension not captured yet there is yield data captured

- Zero wet weight yield but greater than zero dry weight.

In [218]:
```
wet_zero_yield_cond = (data['box1_wet_weight']==0) | (data['box2_wet_weight']==0
dry_non_zero_yield_cond = (data['box1_dry_weight']>0) | (data['box2_dry_weight']>

data[wet_zero_yield_cond & dry_non_zero_yield_cond]
```

Out[218]:

box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected_

0 rows × 123 columns

There is no record where there is Zero wet weight yield but greater than zero dry weight.

- Harvest crop mixed with other crops

In [219]:
```
data['did_the_farmer_keep_the_crops_in_separate_bags'].value_counts(dropna=False)
```

Out[219]:
```
yes    2098
NaN      85
no        4
Name: did_the_farmer_keep_the_crops_in_separate_bags, dtype: int64
```

In [220]:
```
data['was_anything_added_or_removed_from_the_harvest_bag'].value_counts(dropna=Fa
```

Out[220]:
```
no     2098
NaN      89
Name: was_anything_added_or_removed_from_the_harvest_bag, dtype: int64
```

In [221]:
```
FarmerDidNotKeepCropsSeparate_cond = data['did_the_farmer_keep_the_crops_in_separ
ThingsWeresAddedOrRemovedfromBag_cond = data['was_anything_added_or_removed_from_
```

In [222]:
```
data[FarmerDidNotKeepCropsSeparate_cond | ThingsWeresAddedOrRemovedfromBag_cond].
```

Out[222]:

	box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expec
81	8b667feb-a64a-4f49-a155-0c7c4eca15ed	A	0-5KM	Done	
82	8fe16021-b971-4f3c-9cce-e1881d9c8e02	A	0-5KM	Done	
83	f596d5da-c4ad-4c99-86df-c46d876c17db	A	0-5KM	Done	
94	c503d820-3202-439c-8ebf-283b37fe2503	A	0-5KM	successful	
123	b530577e-23b0-4957-9eda-8a3fc5c0639b	A	More_than_5_KM	seed failure	

5 rows × 123 columns

In [223]:
```
len(data[FarmerDidNotKeepCropsSeparate_cond | ThingsWeresAddedOrRemovedfromBag_cc
```

Out[223]:
```
89
```

There are 89 records where there is no confirmation of not mixing the crops or adding somethings to them.

In [224]:
```
print("""The following enumerators regisetred not keeping the harvest bag without
crops, adding something to or removing something from it at least once.""")
data[FarmerDidNotKeepCropsSeparate_cond | ThingsWeresAddedOrRemovedfromBag_cond][
value_counts().reset_index().\
rename({'index':'enumerator','username_box_plc':'count'},axis=1)
```

```
The following enumerators regisetred not keeping the harvest bag without mixing
it with other
crops, adding something to or removing something from it at least once.
```

Out[224]:

	enumerator	count
0	nig168	21
1	nig024	11
2	nig175	9
3	nig137	8
4	nig114	6
5	nig035	5
6	nig148	5
7	nig187	3
8	nig033	3
9	nig174	3
10	nig186	3
11	nig118	3
12	nig022	2
13	nig103	2
14	nig188	2
15	nig109	1
16	nig178	1
17	nig021	1

Task 3: spatial distribution of data points on a map

Evaluating the Performance of Enumerators

Criteria

- "was_anything_added_or_removed_from_the_harvest_bag"
- Registering box1_wet_weight
- Registering box2_wet_weight
- Registering box1_dry_weight
- Registering box2_dry_weight

In [225]:
```
data['was_anything_added_or_removed_from_the_harvest_bag'].value_counts(dropna=Fa
```

Out[225]:
```
no     2098
NaN      89
Name: was_anything_added_or_removed_from_the_harvest_bag, dtype: int64
```

In [226]:
```
def evaluating_bag(val):
    if val == 'no':
        return 1
    else:
        0

data['harvest_bag_score'] = data['was_anything_added_or_removed_from_the_harvest_
                            apply(lambda x: evaluating_bag(x))
```

In [227]:
```
data['box1_wet_weight'].value_counts(dropna=False)
```

Out[227]:
```
NaN       87
1.900      7
1.760      6
1.810      6
1.610      6
          ..
13.180     1
4.435      1
4.525      1
1.515      1
7.600      1
Name: box1_wet_weight, Length: 1314, dtype: int64
```

In [228]:
```
data['box2_wet_weight'].value_counts(dropna=False)
```

Out[228]:
```
NaN       90
1.555      6
2.835      6
1.960      5
4.925      5
          ..
1.315      1
0.300      1
11.420     1
11.470     1
7.345      1
Name: box2_wet_weight, Length: 1308, dtype: int64
```

In [229]:
```
data['box1_dry_weight'].value_counts(dropna=False)
```

Out[229]:
```
NaN       91
2.000      8
3.000      7
3.280      6
1.895      6
          ..
14.620     1
16.890     1
7.735      1
4.950      1
5.795      1
Name: box1_dry_weight, Length: 1281, dtype: int64
```

In [230]:
```
data['box2_dry_weight'].value_counts(dropna=False)
```

Out[230]:
```
NaN      94
1.525     7
3.100     7
2.000     6
8.000     6
         ..
1.630     1
1.205     1
1.290     1
8.455     1
6.425     1
Name: box2_dry_weight, Length: 1245, dtype: int64
```

In [231]:
```
def evaluating_box(val):
    if val:
        return 1
    else:
        0

data['box1_wet_score'] = data['box1_wet_weight'].apply(lambda x: evaluating_box(x
data['box2_wet_score'] = data['box2_wet_weight'].apply(lambda x: evaluating_box(x
data['box1_dry_score'] = data['box1_dry_weight'].apply(lambda x: evaluating_box(x
data['box2_dry_score'] = data['box2_dry_weight'].apply(lambda x: evaluating_box(x
```

I will evaluate the performance of the enumerator for each season separately. Since the mixing happens only in the dry season, the score of the dry season will include the harvest_bag_score

In [232]:
```
data['enumerator_wet_score'] = data['box1_wet_score'] + data['box2_wet_score']

data['enumerator_dry_score'] = (
    data['box1_dry_score'] + data['box2_dry_score'] + data['harvest_bag_score'])
```

Evaluating the Coordinates

Criteria latitude_box_plc == latitude_wet == latitude_dry

In [233]:
```
cond1 = data['latitude_box_plc'] != data['latitude_wet']
cond2 = data['latitude_box_plc'] != data['latitude_dry']
cond3 = data['latitude_wet'] != data['latitude_dry']
len(data[cond1 & cond2]), len(data[cond3])
```

Out[233]: (2135, 2048)

In [234]:
```
2135/len(data), 2048/len(data)
```

Out[234]: (0.9762231367169639, 0.9364426154549611)

In more than 90% of the cases, the coordinates of box_placement are different from those of wet_harvest which are also different from those of dry_harvest.

In [235]:
```
wet_data_eval = data[['username_wet','latitude_wet','longitude_wet','enumerator_w
wet_data_eval = wet_data_eval.groupby(['username_wet','latitude_wet','longitude_w
                        agg({'enumerator_wet_score':'sum'}).\
                        reset_index()
wet_data_eval
```

Out[235]:

	username_wet	latitude_wet	longitude_wet	enumerator_wet_score
0	nig020	9.891854	9.865496	2
1	nig020	9.891857	9.865372	2
2	nig020	9.891892	9.865480	2
3	nig020	9.891938	9.865529	2
4	nig020	9.892006	9.865446	2
...	...	...	...	...
1985	nig190	11.878021	7.643446	2
1986	nig190	11.878037	7.643543	2
1987	nig190	11.878064	7.643559	2
1988	nig190	11.878084	7.643659	2
1989	nig190	11.878110	7.643607	2

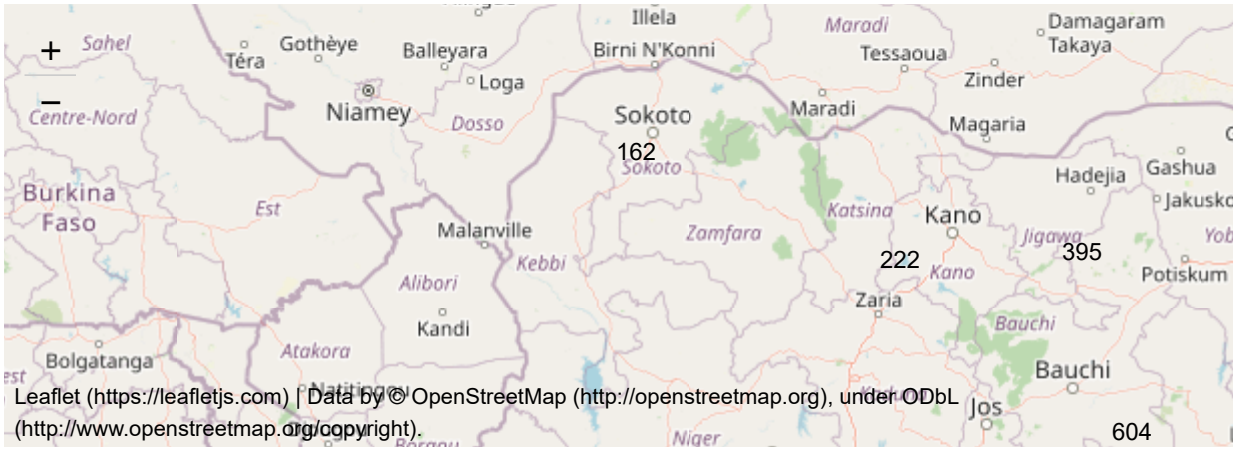
1990 rows × 4 columns

In [236]:
```
import folium
from folium import plugins
```

Wet Map

In [268]:
```
latitude = wet_data_eval['latitude_wet'].mean()
longitude = wet_data_eval['longitude_wet'].mean()

# let's start again with a clean copy of the map of San Francisco
wet_map = folium.Map(location = [latitude, longitude], height='60%',zoom_start =6

# instantiate a mark cluster object for the incidents in the dataframe
farms = plugins.MarkerCluster().add_to(wet_map)

# loop through the dataframe and add each data point to the mark cluster
for (index,row) in  wet_data_eval.iterrows():
    folium.Marker(location = [row['latitude_wet'], row['longitude_wet']],
                  icon=None,
                  popup='Enumerator: '+row['username_wet']+'\n'+'Score: '+ str(rc
                 ).add_to(farms)

# display map
wet_map
```

Out[268]:



In [238]:
```
dry_data_eval = data[['username_dry','latitude_dry','longitude_dry','enumerator_d
dry_data_eval = dry_data_eval.groupby(['username_dry','latitude_dry','longitude_d
                        agg({'enumerator_dry_score':'sum'}).\
                        reset_index()
dry_data_eval.head()
```

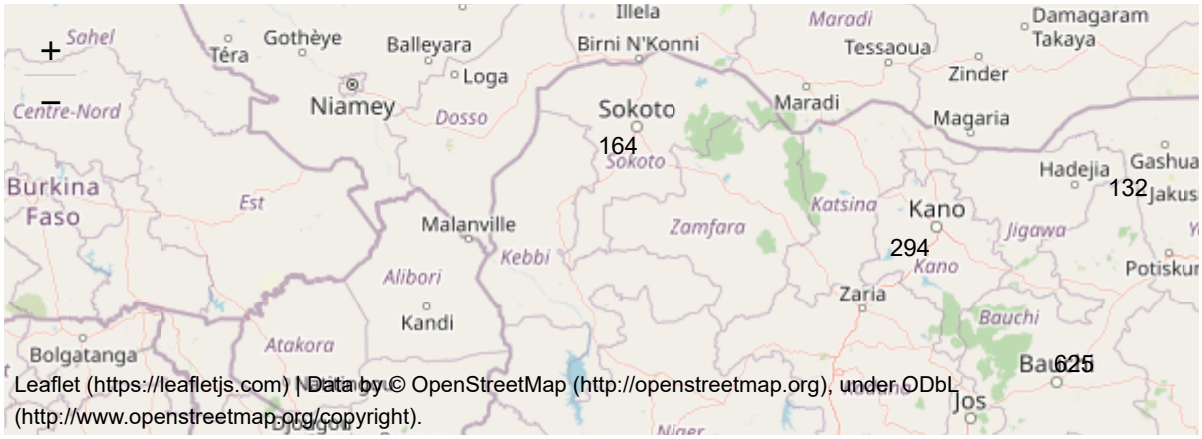
Out[238]:

	username_dry	latitude_dry	longitude_dry	enumerator_dry_score
0	nig020	9.891840	9.865432	3.0
1	nig020	9.891868	9.865470	3.0
2	nig020	9.891923	9.865427	3.0
3	nig020	9.891954	9.865426	3.0
4	nig020	9.891967	9.865486	3.0

Dry Map

In [269]:
```
latitude = dry_data_eval['latitude_dry'].mean()
longitude = dry_data_eval['longitude_dry'].mean()

# let's start again with a clean copy of the map of San Francisco
dry_map = folium.Map(location = [latitude, longitude], height='60%',zoom_start =6

# instantiate a mark cluster object for the incidents in the dataframe
farms = plugins.MarkerCluster().add_to(dry_map)

# loop through the dataframe and add each data point to the mark cluster
for (index,row) in  dry_data_eval.iterrows():
    folium.Marker(location = [row['latitude_dry'], row['longitude_dry']],
                  icon=None,
                  popup='Enumerator: '+row['username_dry']+'\n'+'Score: '+ str(rc
                 ).add_to(farms)

# display map
dry_map
```

Out[269]:



Task 4: average yield in Mt/ha using dry weight

The weight of the yields 'box1_dry_weight', 'box1_wet_weight','box2_dry_weight' and 'box2_wet_weight' are given in kilos while the dimensions of the box are 5x8 which mean 40 squared meter.

I will divied by 40 to get the yield per meter-squared then multiply by 10000 to get the yield per hectare, then divide by 1000 to get the weight in tons

In [244]:
```
data['box1_dry_weight_ton_mt/ha'] = (data['box1_dry_weight']/40)*10
data['box2_dry_weight_ton_mt/ha'] = (data['box2_dry_weight']/40)*10
data['box1_wet_weight_ton_mt/ha'] = (data['box1_wet_weight']/40)*10
data['box2_wet_weight_ton_mt/ha'] = (data['box2_wet_weight']/40)*10
```

Box1 Average Yield

In [245]:
```
box1_avg = data['box1_dry_weight_ton_mt/ha'].mean()
print(f'Box1 Avg.= {box1_avg} ton per Mt/Ha')
```

```
Box1 Avg.= 1.0414358301526718 ton per Mt/Ha
```

In [246]:
```
box2_avg = data['box2_dry_weight_ton_mt/ha'].mean()
print(f'Box2 Avg.= {box2_avg} ton per Mt/Ha')
```

```
Box2 Avg.= 1.045463210702341 ton per Mt/Ha
```

Task 5: finding outliers in yields

In [247]:
```
weight_stats = data[['box1_wet_weight','box2_wet_weight','box1_dry_weight','box2_
                describe(percentiles = [0.25,0.50,0.75])
weight_stats
```

Out[247]:

	box1_wet_weight	box2_wet_weight	box1_dry_weight	box2_dry_weight
count	2100.000000	2097.000000	2096.000000	2093.000000
mean	4.866418	4.837582	4.165743	4.181853
std	3.444603	4.035753	3.238921	3.860188
min	0.055000	0.045000	0.050000	0.045000
25%	2.255000	2.050000	1.800000	1.690000
50%	4.087500	3.980000	3.325000	3.290000
75%	6.691250	6.695000	5.661250	5.615000
max	22.000000	93.900000	22.000000	92.900000

In [248]:
```
box1_wet_iqr = weight_stats.loc['75%','box1_wet_weight']-weight_stats.loc['25%',
box1_wet_upper_limit = weight_stats.loc['75%','box1_wet_weight'] + 1.5*box1_wet_i
box1_wet_lower_limit = weight_stats.loc['25%','box1_wet_weight'] - 1.5*box1_wet_i

box2_wet_iqr = weight_stats.loc['75%','box2_wet_weight']-weight_stats.loc['25%',
box2_wet_upper_limit = weight_stats.loc['75%','box2_wet_weight'] + 1.5*box2_wet_i
box2_wet_lower_limit = weight_stats.loc['25%','box2_wet_weight'] - 1.5*box2_wet_i

box1_dry_iqr = weight_stats.loc['75%','box1_dry_weight']-weight_stats.loc['25%',
box1_dry_upper_limit = weight_stats.loc['75%','box1_dry_weight'] + 1.5*box1_dry_i
box1_dry_lower_limit = weight_stats.loc['25%','box1_dry_weight'] - 1.5*box1_dry_i

box2_dry_iqr = weight_stats.loc['75%','box2_dry_weight']-weight_stats.loc['25%',
box2_dry_upper_limit = weight_stats.loc['75%','box2_dry_weight'] + 1.5*box2_dry_i
box2_dry_lower_limit = weight_stats.loc['25%','box2_dry_weight'] - 1.5*box2_dry_i



def weight_outlier(val,lower_limit,upper_limit):
    if val < lower_limit or val > upper_limit:
        return 1
    else:
        return 0

data['box1_wet_outlier'] = data['box1_wet_weight'].apply(
                                lambda x: weight_outlier(x,box1_wet_lower_limit,b

data['box2_wet_outlier'] = data['box2_wet_weight'].apply(
                                lambda x: weight_outlier(x,box2_wet_lower_limit,b

data['box1_dry_outlier'] = data['box1_dry_weight'].apply(
                                lambda x: weight_outlier(x,box1_dry_lower_limit,b

data['box2_dry_outlier'] = data['box2_dry_weight'].apply(
                                lambda x: weight_outlier(x,box2_dry_lower_limit,b
```

In [249]:
```
# shwoing the filtered data
box1_wet_condition = data['box1_wet_outlier'] == 0
box2_wet_condition = data['box2_wet_outlier'] == 0
box1_dry_condition = data['box1_dry_outlier'] == 0
box2_dry_condition = data['box2_dry_outlier'] == 0


filtered_data = data[box1_wet_condition & box2_wet_condition & box1_dry_condition
filtered_data.head()
```

Out[249]:

	box_plc_@case_id	client_name	distance_to_water_body	enumerator_comment_box_plc	expected
0	36311aa7-29d1-4650-a230-830bcd565d72	A	More_than_5_KM	successfuly done	
1	0a9e2724-59e1-48cd-ab48-856dce787fec	A	More_than_5_KM	ok	
2	da9f7363-2415-415e-8cef-d6bb18ed2bc0	A	More_than_5_KM	ok	
3	9a14fb60-c3a6-4794-9f92-e84953db4b32	A	More_than_5_KM	ok	
4	2fe51289-da17-4214-a7fd-ee7e064dfed3	A	More_than_5_KM	ok	

5 rows × 138 columns

In [250]:
```
print(len(filtered_data))
```

```
2081
```

There are 2081 records without outliers in the box1 and box2 outliers for both wet and sry seasons

In [253]:
```
filtered_box1_wet_avg = filtered_data['box1_wet_weight_ton_mt/ha'].mean()
filtered_box2_wet_avg = filtered_data['box2_wet_weight_ton_mt/ha'].mean()
filtered_box1_dry_avg = filtered_data['box1_dry_weight_ton_mt/ha'].mean()
filtered_box2_dry_avg = filtered_data['box2_dry_weight_ton_mt/ha'].mean()
```

In [254]:
```
print(f'Wet-season Box1 Avg: {filtered_box1_wet_avg.round(2)} ton per Mt/Ha')
print(f'Wet-season Box2 Avg: {filtered_box2_wet_avg.round(2)} ton per Mt/Ha')
print(f'Dry-season Box1 Avg: {filtered_box1_dry_avg.round(2)} ton per Mt/Ha')
print(f'Dry-season Box2 Avg: {filtered_box2_dry_avg.round(2)} ton per Mt/Ha')
```

```
Wet-season Box1 Avg: 1.09 ton per Mt/Ha
Wet-season Box2 Avg: 1.07 ton per Mt/Ha
Dry-season Box1 Avg: 0.92 ton per Mt/Ha
Dry-season Box2 Avg: 0.91 ton per Mt/Ha
```

In [256]:
```
box1_wet_outlier_condition = data['box1_wet_outlier'] == 1
box2_wet_outlier_condition = data['box2_wet_outlier'] == 1
box1_dry_outlier_condition = data['box1_dry_outlier'] == 1
box2_dry_outlier_condition = data['box2_dry_outlier'] == 1
```

Enumerators with Suspecious Data

In [263]:
```
data[box1_wet_outlier_condition]['username_box_plc'].value_counts()
```

Out[263]:
```
nig045    27
nig184    25
nig027     8
nig148     5
nig114     2
nig168     1
nig049     1
nig029     1
nig020     1
nig033     1
Name: username_box_plc, dtype: int64
```

In [264]:
```
data[box2_wet_outlier_condition]['username_box_plc'].value_counts()
```

Out[264]:
```
nig045    25
nig184    24
nig027    13
nig168     3
nig148     2
nig137     1
nig029     1
nig020     1
nig033     1
Name: username_box_plc, dtype: int64
```

In [265]:
```
data[box1_dry_outlier_condition]['username_box_plc'].value_counts()
```

Out[265]:
```
nig184    30
nig045    26
nig027    14
nig168     5
nig148     3
nig114     2
nig137     1
nig030     1
nig020     1
nig033     1
Name: username_box_plc, dtype: int64
```

In [266]:
```
data[box2_dry_outlier_condition]['username_box_plc'].value_counts()
```

Out[266]:
```
nig184    30
nig045    27
nig027    20
nig168     5
nig030     2
nig114     1
nig029     1
nig020     1
nig033     1
Name: username_box_plc, dtype: int64
```

It seems that enumerators nig184, nig045 and nig027 have submitted multtple recors with outlier data

Task 6: indentifying the major problems affecting crops

In [151]:
```
box1_problems = data['box1_problem'].value_counts().to_frame().reset_index().\
 rename({'index':'problem','box1_problem':'count'},axis=1)
box1_problems
```

Out[151]:

	problem	count
0	late_planting	328
1	poor_germination	257
2	drought	76
3	other_pest_and_disease	65
4	other_pest_and_disease late_planting	49
5	poor_germination late_planting	33
6	drought poor_germination locust_infestation	33
7	weeds	27
8	drought poor_germination	20
9	drought late_planting	18
10	others	17
11	drought poor_germination late_planting	16
12	weeds late_planting	15
13	poor_germination weeds late_planting	15
14	poor_germination weeds	10
15	animal_cattle_encroachment	8
16	animal_cattle_encroachment late_planting	8
17	poor_germination animal_cattle_encroachment la...	8
18	poor_germination locust_infestation other_pest...	7
19	locust_infestation late_planting	7
20	locust_infestation	6
21	poor_germination animal_cattle_encroachment we...	5
22	late_planting others	5
23	drought locust_infestation other_pest_and_dise...	5
24	drought other_pest_and_disease	4
25	drought poor_germination locust_infestation ot...	4
26	poor_germination locust_infestation other_pest...	3
27	animal_cattle_encroachment weeds late_planting	3
28	drought animal_cattle_encroachment late_planting	2
29	animal_cattle_encroachment weeds	2
30	drought poor_germination locust_infestation ot...	2
31	drought other_pest_and_disease animal_cattle_e...	2
32	poor_germination animal_cattle_encroachment	2

	problem	count
33	poor_germination locust_infestation other_pest...	2
34	drought others	2
35	other_pest_and_disease weeds late_planting	2
36	poor_germination locust_infestation	2
37	poor_germination flood animal_cattle_encroachm...	2
38	poor_germination locust_infestation weeds late...	1
39	drought poor_germination other_pest_and_diseas...	1
40	locust_infestation other_pest_and_disease	1
41	drought poor_germination locust_infestation ot...	1
42	drought other_pest_and_disease animal_cattle_e...	1
43	locust_infestation other_pest_and_disease anim...	1
44	drought weeds late_planting	1
45	locust_infestation other_pest_and_disease late...	1
46	poor_germination locust_infestation other_pest...	1
47	weeds late_planting others	1
48	locust_infestation other_pest_and_disease anim...	1
49	drought animal_cattle_encroachment weeds late_...	1
50	flood other_pest_and_disease animal_cattle_enc...	1
51	poor_germination animal_cattle_encroachment weeds	1
52	drought poor_germination other_pest_and_disease	1
53	drought locust_infestation	1
54	poor_germination others	1
55	drought locust_infestation other_pest_and_dise...	1
56	drought poor_germination other_pest_and_diseas...	1
57	flood	1
58	drought animal_cattle_encroachment	1
59	poor_germination other_pest_and_disease	1

It can be seen that the main problems of box1 are:

- Late planning.
- Poor germination.
- Drought
- Other pest and disease.

Task 7: Report

General Observations:

- The dataset includes 20187 rows and 138 columns.
- There missing values in the multiple columns.
- There are not units for the measurements of weights.

Yields:

- The average yield is around 1.08 ton per Mt/Ha for wet-season, and 0.92 ton per Mt/Ha for dry season.

Erroneous Data:

- The main cause for the erroneous data is the missing values null.
- I filtered out the missing values for the 'box1_wet_weight', 'box2_wet_weight','box1_dry_weight' and 'box2_dry_weight'

Major Factors Affecting Crops:

- Late planning.
- Poor germination.
- Drought.
- Other pest and disease.

Suspecious Data:

- Enumerators nig184, nig045 and nig027 have submitted multtple recors with outlier data