In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
HP = pd.read_csv("C:\\Users\\harsh\\Personal\\A1a,b\\NSSO.csv")
```

In [3]:
```
HP.describe()
```

Out[3]:

	Unnamed: 0	grp	Round_Centre	FSU_number	Round	Schedule_Number	Sample	Sector	state	State_Region	...	preparedswe
count	2041.000000	2.041000e+03	2041.0	2041.000000	2041.0	2041.0	2041.0	2041.000000	2041.0	2041.000000	...	2041.000
mean	85732.754532	7.362489e+31	1.0	73626.030867	68.0	10.0	1.0	1.187653	2.0	21.485546	...	15.321
std	22639.858394	1.162820e+31	0.0	11623.665973	0.0	0.0	0.0	0.390531	0.0	0.499914	...	39.726
min	38448.000000	4.940000e+31	1.0	49410.000000	68.0	10.0	1.0	1.000000	2.0	21.000000	...	0.000
25%	95876.000000	7.910000e+31	1.0	79115.000000	68.0	10.0	1.0	1.000000	2.0	21.000000	...	0.000
50%	96386.000000	7.920000e+31	1.0	79179.000000	68.0	10.0	1.0	1.000000	2.0	21.000000	...	0.000
75%	96896.000000	7.920000e+31	1.0	79243.000000	68.0	10.0	1.0	1.000000	2.0	22.000000	...	23.333
max	98126.000000	7.950000e+31	1.0	79488.000000	68.0	10.0	1.0	2.000000	2.0	22.000000	...	1300.000

8 rows × 383 columns

In [4]:
```
import pandas as pd

HP.columns = HP.columns.tolist()
print(HP.columns)
```

```
Index(['Unnamed: 0', 'grp', 'Round_Centre', 'FSU_number', 'Round',
       'Schedule_Number', 'Sample', 'Sector', 'state', 'State_Region',
       ...
       'pickle_v', 'sauce_jam_v', 'Othrprocessed_v', 'Beveragestotal_v',
       'foodtotal_v', 'foodtotal_q', 'state_1', 'Region', 'fruits_df_tt_v',
       'fv_tot'],
      dtype='object', length=384)
```

In [5]:
```
from sklearn.linear_model import LinearRegression
from sklearn import datasets, linear_model, metrics
```

In [6]:
```
list(HP)
```

Out[6]:
```
['Unnamed: 0',
 'grp',
 'Round_Centre',
 'FSU_number',
 'Round',
 'Schedule_Number',
 'Sample',
 'Sector',
 'state',
 'State_Region',
 'District',
 'Stratum_Number',
 'Sub_Stratum',
 'Schedule_type',
 'Sub_Round',
 'Sub_Sample',
 'FOD_Sub_Region',
 'Hamlet_Group_Sub_Block',
 'Second',
 'X_Stage_Stratum',
 'HHS_No',
 'Level',
 'Filler',
 'hhdsz',
 'NIC_2008',
 'NCO_2004',
 'HH_type',
 'Religion',
 'Social_Group',
 'Whether_owns_any_land',
 'Type_of_land_owned',
 'Land_Owned',
 'Land_Leased_in',
 'Otherwise_possessed',
 'Land_Leased_out',
 'Land_Total_possessed',
 'During_July_June_Cultivated',
 'During_July_June_Irrigated',
 'NSS',
 'NSC',
 'MLT',
 'land_tt',
 'Cooking_code',
 'Lighting_code',
```

```
'Dwelling_unit_code',
'Regular_salary_earner',
'Perform_Ceremony',
'Meals_seved_to_non_hhld_members',
'Possess_ration_card',
'Type_of_ration_card',
'MPCE_URP',
'MPCE_MRP',
'Person_Srl_No',
'Relation',
'Sex',
'Age',
'Marital_Status',
'Education',
'Days_Stayed_away',
'No_of_Meals_per_day',
'Meals_School',
'Meals_Employer',
'Meals_Others',
'Meals_Payment',
'Meals_At_Home',
'Item_Code',
'Source_Code',
'ricepds_q',
'riceos_q',
'ricetotal_q',
'chira_q',
'khoi_q',
'muri_q',
'ricepro_q',
'riceGT_q',
'Wheatpds_q',
'wheatos_q',
'wheattotal_q',
'maida_q',
'suji_q',
'sewai_q',
'bread_q',
'wheatp_q',
'wheatGT_q',
'jowarp_q',
'bajrap_q',
'maizep_q',
'barleyp_q',
```

```
'milletp_q',
'ragip_q',
'cerealot_q',
'cerealtot_q',
'cerealsub_q',
'cerealstt_q',
'arhar_q',
'gramdal_q',
'gramwholep_q',
'gramGT_q',
'moong_q',
'masur_q',
'urd_q',
'peasdal_q',
'khesari_q',
'otpulse_q',
'gramp_q',
'besan_q',
'pulsep_q',
'pulsestot_q',
'pulsestt_q',
'soyabean_q',
'milk_q',
'babyfood_q',
'milkcond_q',
'curd_q',
'ghee_q',
'butter_q',
'icecream_q',
'otmilkp_q',
'Milktotal_q',
'milkprott_q',
'vanas_q',
'musoil_q',
'gnoil_q',
'cocooil_q',
'edioilothr_q',
'edibletotal_q',
'ediblest_q',
'eggsno_q',
'fishprawn_q',
'goatmeat_q',
'beef_q',
'pork_q',
```

```
'chicken_q',
'othrbirds_q',
'nonvegtotal_q',
'emftt_q',
'potato_q',
'onion_q',
'tamato_q',
'brinjal_q',
'radish_q',
'carrot_q',
'palak_q',
'chillig_q',
'bhindi_q',
'parwal_q',
'cauli_q',
'cabbage_q',
'pumpkin_q',
'peas_q',
'fbeans_q',
'lemonno_q',
'otveg_q',
'vegtt_q',
'bananano_q',
'jackfruit_q',
'watermel_q',
'pineaplno_q',
'cocono_q',
'cocogno_q',
'guava_q',
'sighara_q',
'orangeno_q',
'papayar_q',
'mango_q',
'kharbooz_q',
'pears_q',
'berries_q',
'leechi_q',
'apple_q',
'grapes_q',
'otfruits_q',
'fruitstt_q',
'fruitt_total',
'cocodf_q',
'gnutdf_q',
```

```
'datesdf_q',
'cashewdf_q',
'walnutdf_q',
'otnutsdf_q',
'kishmish_q',
'otherdf_q',
'dryfruitstotal_q',
'dftt_q',
'sugarpds_q',
'sugaros_q',
'sugarst_q',
'gur_q',
'misri_q',
'honey_q',
'sugartotal_q',
'sugartt_q',
'salt_q',
'ginger_q',
'garlic_q',
'jeera_q',
'dhania_q',
'turnmeric_q',
'blackpepper_q',
'drychilly_q',
'tamarind_q',
'currypowder_q',
'oilseeds_q',
'spicesothr_q',
'spicetot_q',
'spicestotal_q',
'teacupno_q',
'tealeaf_q',
'teatotal_q',
'cofeeno_q',
'coffeepwdr_q',
'cofeetotal_q',
'ice_q',
'coldbvrg_q',
'juice_q',
'othrbevrg_q',
'bevergest_q',
'Biscuits_q',
'preparedsweet_q',
'pickle_q',
```

```
'sauce_jam_q',
'Othrprocessed_q',
'Beveragestotal_q',
'ricepds_v',
'riceos_v',
'ricetotal_v',
'chira_v',
'khoi_v',
'muri_v',
'ricepro_v',
'riceGT_v',
'Wheatpds_v',
'wheatos_v',
'wheattotal_v',
'maida_v',
'suji_v',
'sewai_v',
'bread_v',
'wheatp_v',
'wheatGT_v',
'jowarp_v',
'bajrap_v',
'maizep_v',
'barleyp_v',
'milletp_v',
'ragip_v',
'cerealot_v',
'cerealtot_v',
'cerealsub_v',
'cerealstt_v',
'arhar_v',
'gramdal_v',
'gramwholep_v',
'gramGT_v',
'moong_v',
'masur_v',
'urd_v',
'peasdal_v',
'khesari_v',
'otpulse_v',
'gramp_v',
'besan_v',
'pulsep_v',
'pulsestot_v',
```

```
'pulsestt_v',
'soyabean_v',
'milk_v',
'babyfood_v',
'milkcond_v',
'curd_v',
'ghee_v',
'butter_v',
'icecream_v',
'otmilkp_v',
'Milktotal_v',
'milkprott_v',
'vanas_v',
'musoil_v',
'gnoil_v',
'cocooil_v',
'edioilothr_v',
'edibletotal_v',
'ediblest_v',
'eggsno_v',
'fishprawn_v',
'goatmeat_v',
'beef_v',
'pork_v',
'chicken_v',
'othrbirds_v',
'nonvegtotal_v',
'emftt_v',
'potato_v',
'onion_v',
'tamato_v',
'brinjal_v',
'radish_v',
'carrot_v',
'palak_v',
'chillig_v',
'bhindi_v',
'parwal_v',
'cauli_v',
'cabbage_v',
'pumpkin_v',
'peas_v',
'fbeans_v',
'lemonno_v',
```

```
'otveg_v',
'vegtt_v',
'bananano_v',
'jackfruit_v',
'watermel_v',
'pineaplno_v',
'cocono_v',
'cocogno_v',
'guava_v',
'sighara_v',
'orangeno_v',
'papayar_v',
'mango_v',
'kharbooz_v',
'pears_v',
'berries_v',
'leechi_v',
'apple_v',
'grapes_v',
'otfruits_v',
'fruitstt_v',
'cocodf_v',
'gnutdf_v',
'datesdf_v',
'cashewdf_v',
'walnutdf_v',
'otnutsdf_v',
'kishmish_v',
'otherdf_v',
'dryfruitstotal_v',
'dftt_v',
'sugarpds_v',
'sugaros_v',
'sugarst_v',
'gur_v',
'misri_v',
'honey_v',
'sugartotal_v',
'sugartt_v',
'salt_v',
'ginger_v',
'garlic_v',
'jeera_v',
'dhania_v',
```

```
'turnmeric_v',
'blackpepper_v',
'drychilly_v',
'tamarind_v',
'currypowder_v',
'oilseeds_v',
'spicesothr_v',
'spicetot_v',
'spicestotal_v',
'teacupno_v',
'tealeaf_v',
'teatotal_v',
'cofeeno_v',
'coffeepwdr_v',
'cofeetotal_v',
'ice_v',
'coldbvrg_v',
'juice_v',
'othrbevrg_v',
'bevergest_v',
'Biscuits_v',
'preparedsweet_v',
'pickle_v',
'sauce_jam_v',
'Othrprocessed_v',
'Beveragestotal_v',
'foodtotal_v',
'foodtotal_q',
'state_1',
'Region',
'fruits_df_tt_v',
'fv_tot']
```

In [7]:
```
HP_new = HP[[ 'Land_Owned','Land_Leased_in','Land_Leased_out','Age','Meals_At_Home','ricepds_q','Wheatpds_v','pulsep_v','chicken_
```

In [8]:
```
HP_new['total_consumption'] = HP_new[['ricepds_q', 'Wheatpds_v', 'pulsep_v', 'chicken_v']].sum(axis=1)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_15172\1485983585.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  HP_new['total_consumption'] = HP_new[['ricepds_q', 'Wheatpds_v', 'pulsep_v', 'chicken_v']].sum(axis=1)
```

In [9]:
```
HP_new.drop(['ricepds_q', 'Wheatpds_v', 'pulsep_v', 'chicken_v'], axis=1, inplace=True)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_15172\4000381120.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  HP_new.drop(['ricepds_q', 'Wheatpds_v', 'pulsep_v', 'chicken_v'], axis=1, inplace=True)
```

In [10]:
```
HP_new=HP_new.dropna()
```

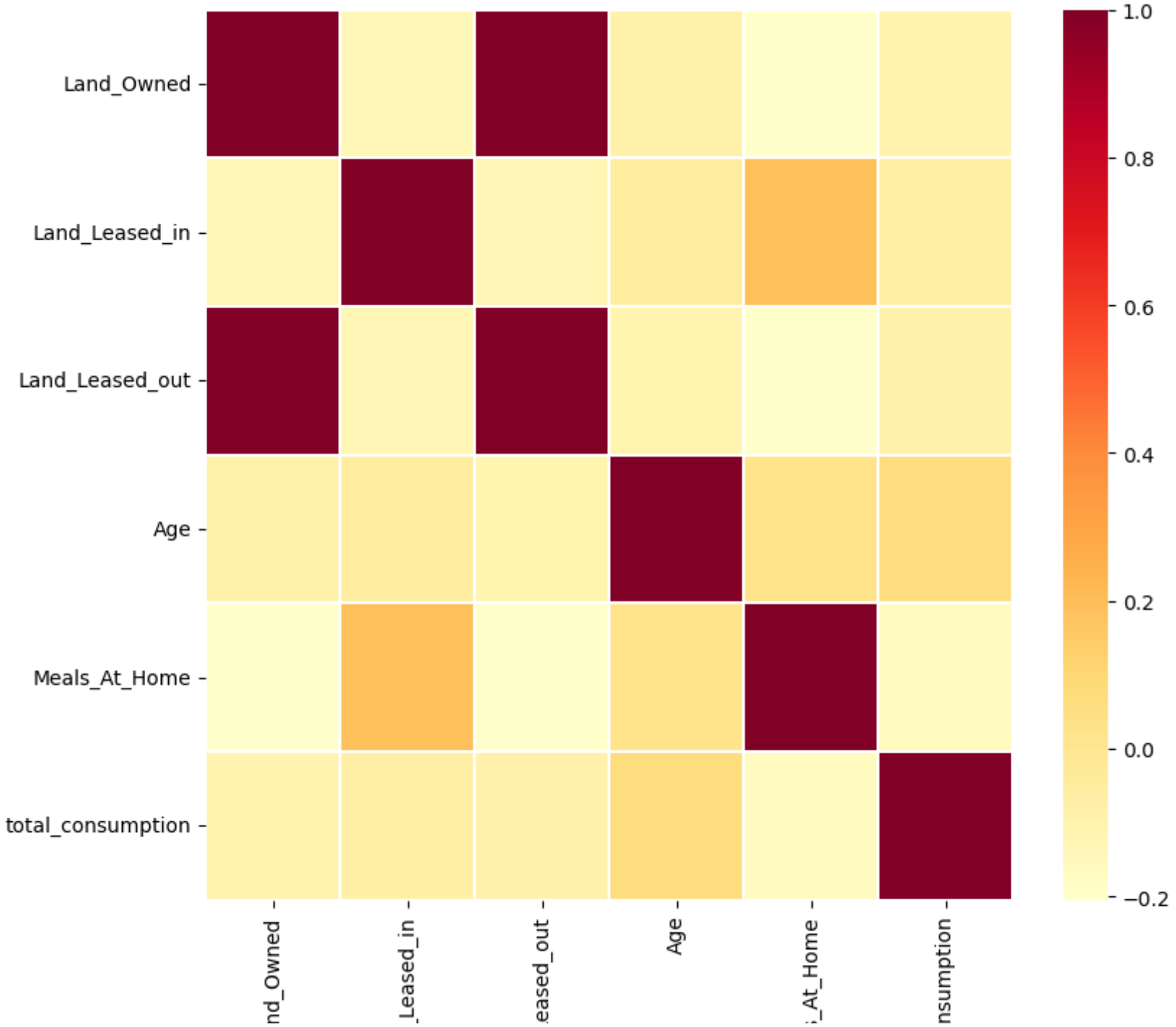
In [11]:
```
HP_new.isnull().any()
```

Out[11]:
```
Land_Owned          False
Land_Leased_in      False
Land_Leased_out     False
Age                 False
Meals_At_Home       False
total_consumption   False
dtype: bool
```

In [12]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:
```
corrmat = HP_new.corr()
f, ax = plt.subplots(figsize = (9, 8))
sns.heatmap(corrmat, ax = ax, cmap='YlOrRd', linewidths = 0.1)
```

Out[13]:
```
<Axes: >
```


Land_Owned
Land_Leased_in
Land_Leased_out
Age
Meals_At_Home
total_consumption
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

In [14]:
```
x=HP_new[['Land_Owned','Land_Leased_in','Land_Leased_out','Age']]
y=HP_new["total_consumption"]
```

In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [17]:
```
linreg=LinearRegression()
linreg.fit(x_train,y_train)
```

Out[17]:
```
▾ LinearRegression

LinearRegression()
```

In [18]:
```
y_pred=linreg.predict(x_test)
y_pred
```

Out[18]:
```
array([50.63322105, 48.13298294, 43.23789642, 48.85191464, 35.15244286,
       38.69667199, 23.70340548])
```

In [19]:
```
from sklearn.metrics import r2_score
```

In [20]:
```
R_squared= r2_score(y_test,y_pred)*100
print( "R_squared of the model is %.2f" %R_squared)
```

```
R_squared of the model is -25.66
```

In [21]:
```
import numpy as pd

def mape(y_test, pred):
    y_test, pred = np.array(y_test), np.array(pred)
    mape = np.mean(np.abs((y_test - pred) / y_test))
    return mape
```

In [22]:
```
mape(y_test,y_pred)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_15172\1099413660.py:5: RuntimeWarning: divide by zero encountered in divide
  mape = np.mean(np.abs((y_test - pred) / y_test))
```

Out[22]:
```
inf
```

In [23]:
```
def mape(y_test, pred):
    y_test, pred = np.array(y_test), np.array(pred)
    mask = y_test != 0  # Mask to exclude zero values
    mape = np.mean(np.abs((y_test - pred) / y_test)[mask])
    return mape
```

In [24]:
```
mape(y_test,y_pred)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_15172\3164413528.py:4: RuntimeWarning: divide by zero encountered in divide
  mape = np.mean(np.abs((y_test - pred) / y_test)[mask])
```

Out[24]:
```
0.33176995742223775
```