

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

%matplotlib inline
```

```
In [2]: total_years = [1860, 1870, 1880, 1890, 1900, 1910, 1920]
total_years_string = [str(x) for x in total_years]
year_dict = dict() ; tehama_dict = dict()
```

```
In [3]: # DICTIONARY CREATION

for year in total_years:
    year_dict['ca2_'+ str(year)] = pd.read_excel(
        '/Users/Champagnesashimi/Desktop/ICPSR2896_CA2.xlsx', sheet_name=
    str(year))
    tehama_dict['ca2_'+ str(year)+'_tehama'] = year_dict['ca2_'+ str(
        year)].loc[year_dict['ca2_'+ str(year)][['name']] == 'TEHAMA']
```

```
In [4]: # CREATE THE DATAFRAME

pop_comp_col = ["EQUIPVAL", "Families", "totpop", "urbpop", "farmval",
                 "farms", "farmval_div_farms", "ruralpop_div_farms", "EQU
IPVAL_div_farm", "percent_urb",
                 'improved_acres', 'unimproved',
                 'livestock_val', 'agout', 'livestock_div_farm',
                 'agout_div_farm', 'livestock_val_improved', 'livestock_val
_div_acre',
                 'agout_div_acre', 'farm02', 'farm39', 'farm1019', 'farm2049'
, 'farm5099',
                 'farm100', 'farm500', 'farm1000']

Ta_pop_comp = pd.DataFrame(index=total_years, columns=pop_comp_col, dtype=
int)

pd.options.mode.chained_assignment = None
```

```
In [5]: # MISSING DATA / RENAMING DATA
```

```
tehama_dict["ca2_1900_tehama"].rename(columns={'farmequi':'equipval'},inplace=True)
tehama_dict["ca2_1910_tehama"].rename(columns={'farmequi':'equipval'},inplace=True)
tehama_dict["ca2_1920_tehama"].rename(columns={'farmequi':'equipval'},inplace=True)
tehama_dict["ca2_1910_tehama"].rename(columns={'acimpman':'acimp'},inplace=True)
tehama_dict["ca2_1870_tehama"].rename(columns={'acunioth':'acunimp'},inplace=True)
tehama_dict["ca2_1890_tehama"].rename(columns={'farm09':'farm02'},inplace=True)
tehama_dict["ca2_1900_tehama"].rename(columns={'farm12':'farm02'},inplace=True)

tehama_dict['ca2_1910_tehama']['equipval'] = 1258232
tehama_dict['ca2_1870_tehama']['families'] = 1509
tehama_dict['ca2_1880_tehama']['families'] = 1589
tehama_dict['ca2_1910_tehama']['farmval'] = 13764171
tehama_dict['ca2_1860_tehama']['farms'] = 289

tehama_dict['ca2_1910_tehama']['livstock'] = 3632891
tehama_dict['ca2_1900_tehama']['acunimp'] = 523924
tehama_dict['ca2_1910_tehama']['acunimp'] = 623948
tehama_dict['ca2_1920_tehama']['acunimp'] = 805724

tehama_dict['ca2_1860_tehama']['farmout'] = 757389
tehama_dict['ca2_1910_tehama']['farmout'] = 2048248
tehama_dict['ca2_1920_tehama']['farmout'] = 2284721

tehama_dict['ca2_1860_tehama']['farm02'] = 0
tehama_dict['ca2_1870_tehama']['farm39'] = 3
tehama_dict['ca2_1890_tehama']['farm39'] = 4
```

```
In [6]: urb_li = ["urb860", "urb870", "urb880", "urb890", "urb900", "urb910", "urb920"]

farm_li = ['farm02','farm39','farm1019','farm2049','farm5099','farm100','farm500','farm1000']
```

```
In [7]: # FUNCTIONS (Still need to put in a Class)
```

```
def DataWrangler(Ta,excel):
    for year,val in zip(total_years,tehama_dict.keys()):
        Ta_pop_comp[Ta][year] = tehama_dict[val][excel]

def DataWranglerDiv(Ta,excel_num,excel_denom):
    for year,val in zip(total_years,tehama_dict.keys()):
        Ta_pop_comp[Ta][year] = tehama_dict[val][excel_num]/tehama_dict[val][excel_denom]

def addToOrig_Dataset(li_of_vals,new_name):
    for val,urb in zip(tehama_dict.keys(),li_of_vals):
        tehama_dict[val][new_name] = tehama_dict[val][urb]
```

```
In [8]: # POPULATION CALLS
```

```
addToOrig_Dataset(urb_li,'g_urb')

DataWrangler('EQUIPVAL','equipval')
DataWrangler('Families','families')
DataWrangler('totpop','totpop')
DataWrangler('farmval','farmval')
DataWrangler('farms','farms')
DataWrangler('urbpop','g_urb')

DataWranglerDiv('farmval_div_farms','farmval','farms')
DataWranglerDiv('EQUIPVAL_div_farm','equipval','farms')
DataWranglerDiv('ruralpop_div_farms','g_urb','farms')
DataWranglerDiv('percent_urb','g_urb','totpop')
DataWranglerDiv('livestock_div_farm','livestock','farms')
DataWranglerDiv('agout_div_farm','livestock','acimp')
DataWranglerDiv('livestock_val_div_acre','livestock','acunimp')
DataWranglerDiv('agout_div_acre','farmout','acimp')
```

```
# FARM CALLS
```

```
DataWrangler('livestock_val','livstock')
DataWrangler('improved_acres','acimp')
DataWrangler('unimproved','acunimp')
DataWrangler('agout','farmout')
DataWrangler('farm02','farm02')
DataWrangler('farm39','farm39')
DataWrangler('farm1019','farm1019')
DataWrangler('farm2049','farm2049')
DataWrangler('farm5099','farm5099')
DataWrangler('farm100','farm100')
DataWrangler('farm500','farm500')
DataWrangler('farm1000','farm1000')
```

```
In [9]: pd.options.display.max_columns = None
Ta_pop_comp
```

Out[9]:

	EQUIPVAL	Families	totpop	urbpop	farmval	farms	farmval_div_farms	rural
1860	96053.0	789.0	4044.0	0.0	946343.0	289.0	3274.543253	0.000
1870	78340.0	1509.0	3587.0	0.0	1353815.0	360.0	3760.597222	0.000
1880	215972.0	1589.0	9301.0	0.0	5221823.0	636.0	8210.413522	0.000
1890	238450.0	1984.0	9916.0	2608.0	11671130.0	781.0	14943.828425	3.339
1900	440020.0	2517.0	10996.0	2750.0	11720120.0	1055.0	11109.118483	2.606
1910	1258232.0	2697.0	11401.0	3530.0	13764171.0	1006.0	13682.078529	3.508
1920	1544013.0	3585.0	12882.0	3104.0	34960408.0	1414.0	24724.475248	2.195

```
In [10]: # VISUALIZATION (still needs to be put in Class)
```

```
def scatter_plot(X,Y,title,X_label,Y_label):
    fig, ax = plt.subplots(figsize=(12,6))
    ax.scatter(X,Y,data=Ta_pop_comp)

    plt.xticks(size=14) ; plt.yticks(size=14)
    plt.xlabel(X_label,size=12); plt.ylabel(Y_label,size=12)
    plt.title(title,size=16)

    for year,txt in zip(total_years,total_years_string):
        ax.annotate(txt, (Ta_pop_comp[X][year], Ta_pop_comp[Y][year]),size=12)

def reg_plot_yearX(Y,title,Y_label):
    fig, ax = plt.subplots(figsize=(12,6))
    ax.plot(Ta_pop_comp.index, Ta_pop_comp[Y])

    plt.xticks(size=14) ; plt.yticks(size=14)
    plt.xlabel("Years",size=12); plt.ylabel(Y_label,size=12)
    plt.title(title,size=16) ; plt.grid(); plt.show()
```

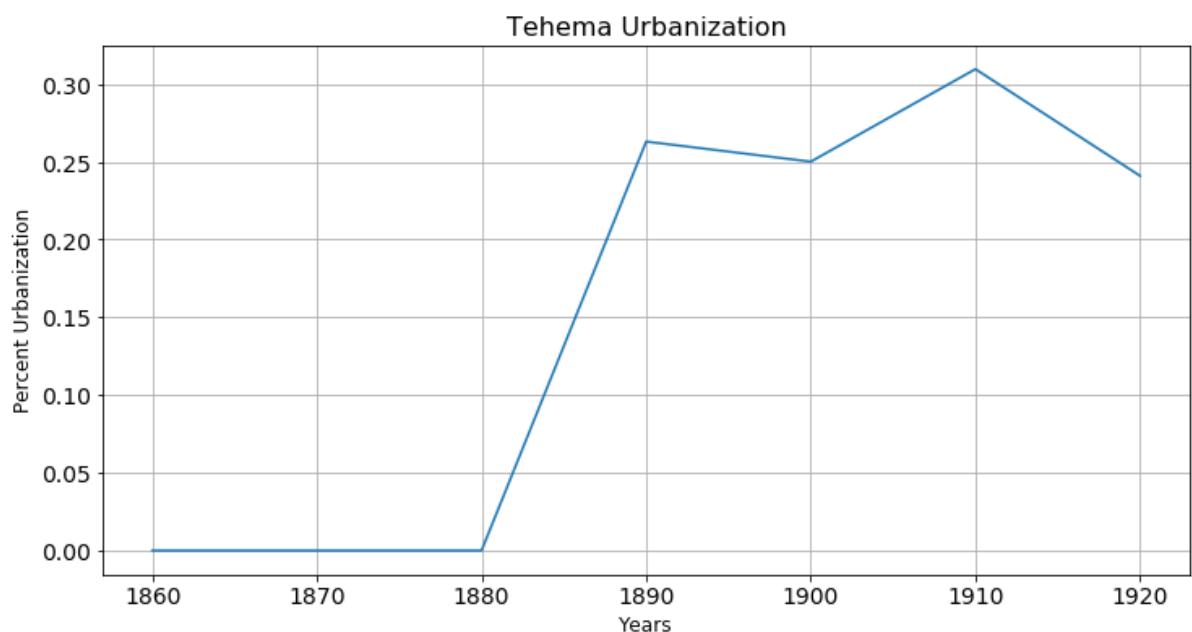
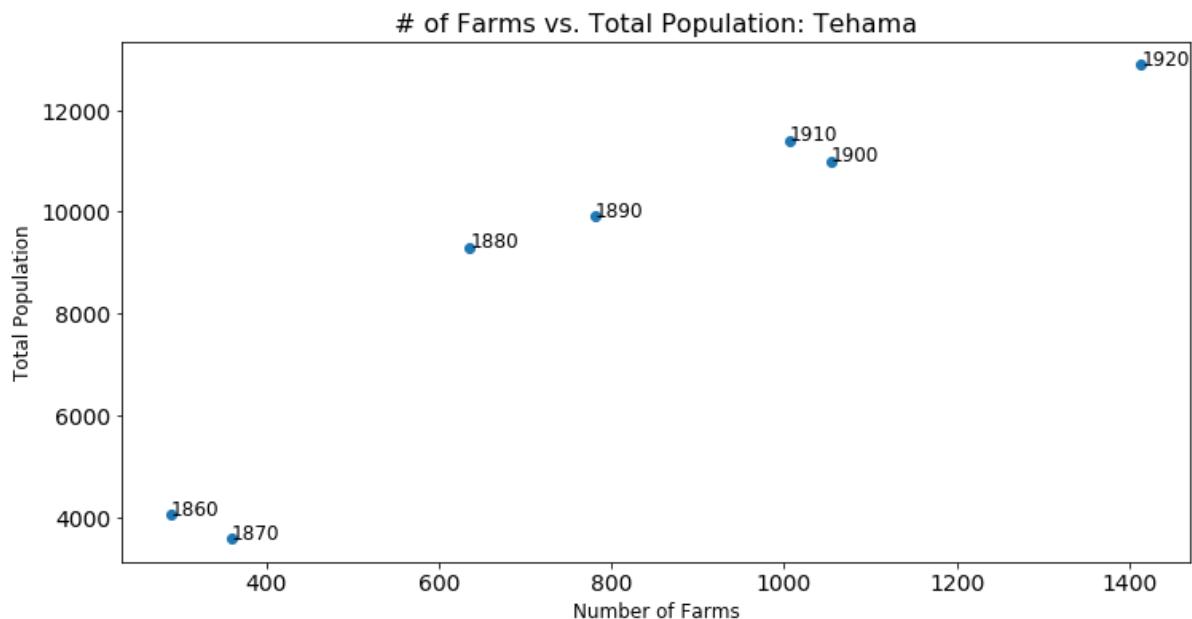
In [11]: Ta_pop_comp

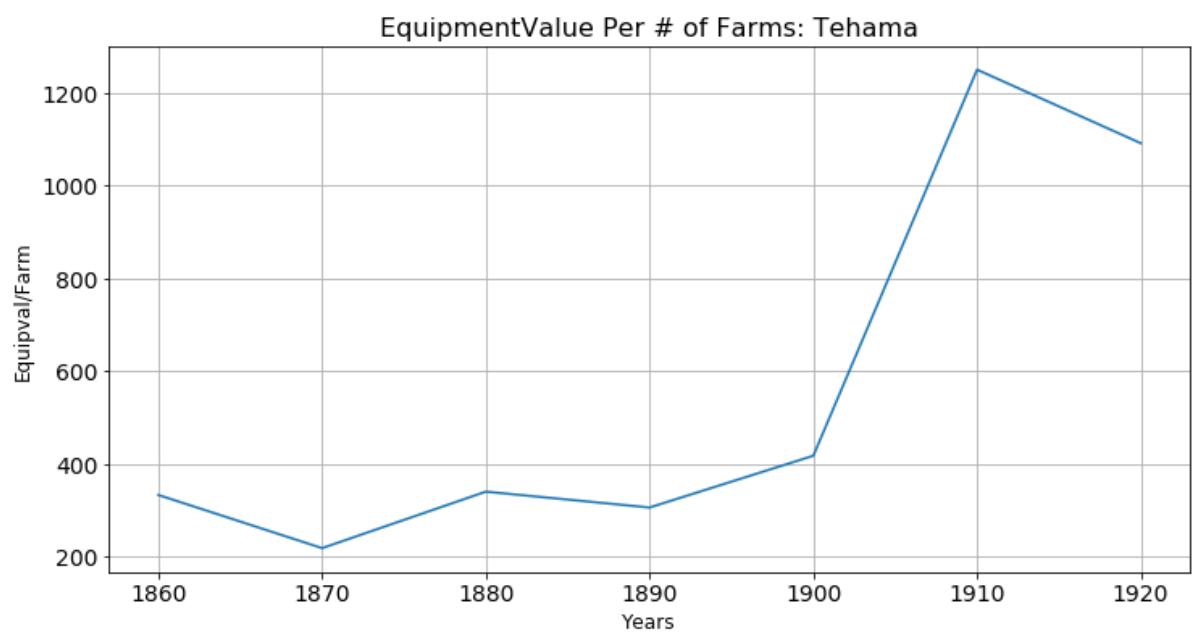
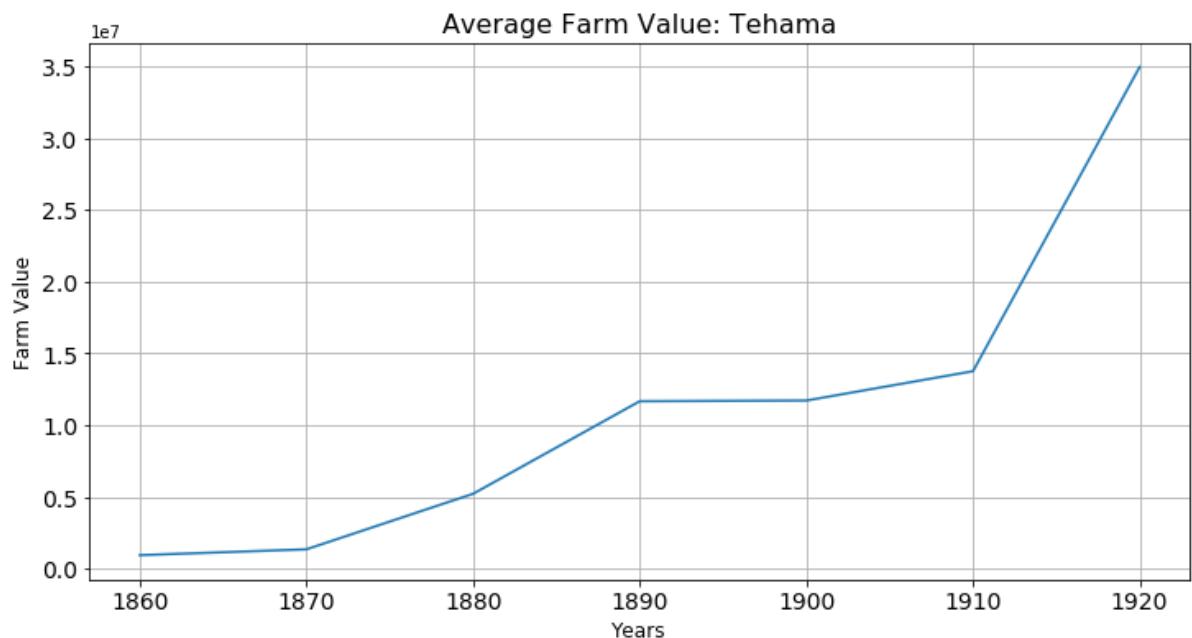
Out[11]:

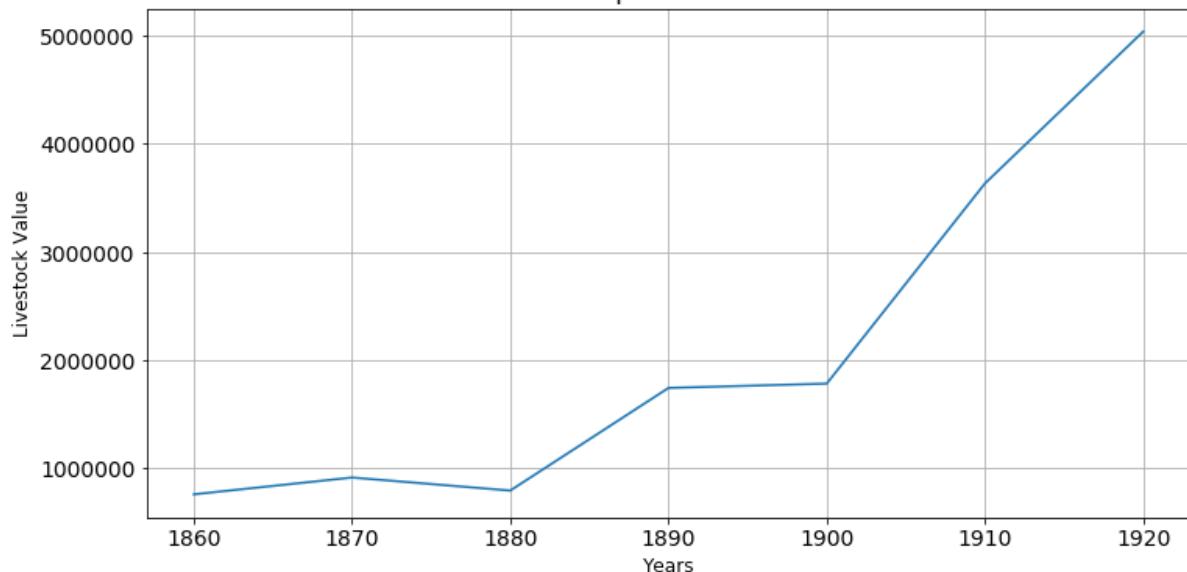
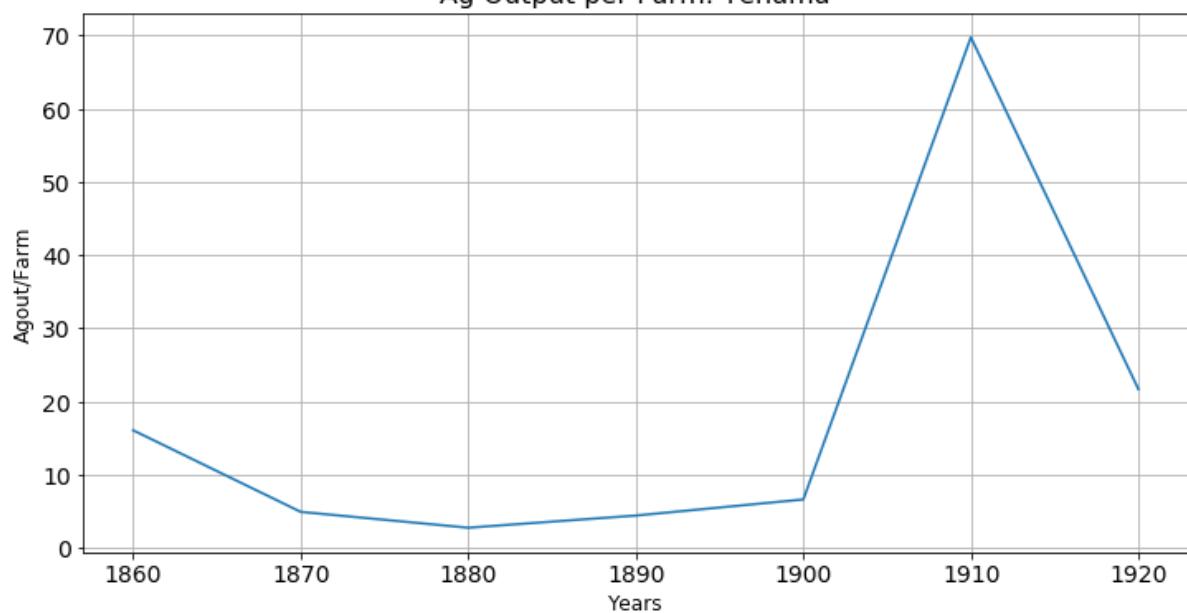
	EQUIPVAL	Families	totpop	urbpop	farmval	farms	farmval_div_farms	rurali
1860	96053.0	789.0	4044.0	0.0	946343.0	289.0	3274.543253	0.000
1870	78340.0	1509.0	3587.0	0.0	1353815.0	360.0	3760.597222	0.000
1880	215972.0	1589.0	9301.0	0.0	5221823.0	636.0	8210.413522	0.000
1890	238450.0	1984.0	9916.0	2608.0	11671130.0	781.0	14943.828425	3.339
1900	440020.0	2517.0	10996.0	2750.0	11720120.0	1055.0	11109.118483	2.606
1910	1258232.0	2697.0	11401.0	3530.0	13764171.0	1006.0	13682.078529	3.508
1920	1544013.0	3585.0	12882.0	3104.0	34960408.0	1414.0	24724.475248	2.195

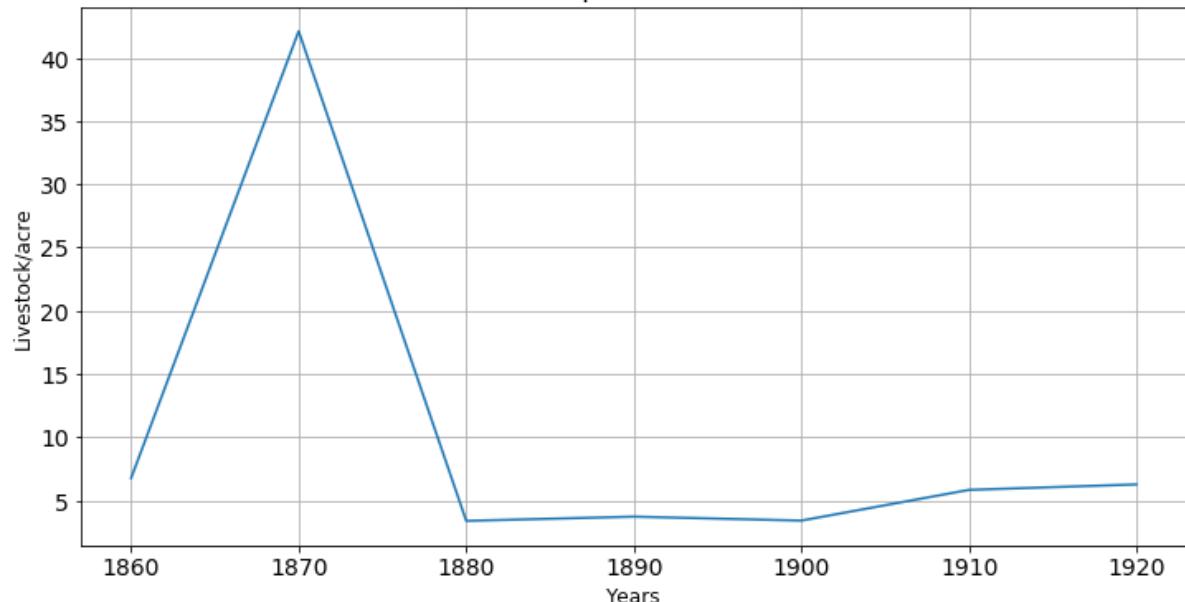
```
In [12]: # VISUALIZATION FUNCTION CLASS
```

```
scatter_plot('farms','totpop','# of Farms vs. Total Population: Tehama',
'Number of Farms','Total Population')
reg_plot_yearX('percent_urb','Tehema Urbanization','Percent Urbanizatio
n')
reg_plot_yearX('farmval','Average Farm Value: Tehama','Farm Value')
reg_plot_yearX('EQUIPVAL_div_farm','EquipmentValue Per # of Farms: Teham
a','Equipval/Farm')
reg_plot_yearX('livestock_val','Livestock per Farm: Tehama','Livestock V
alue')
reg_plot_yearX('agout_div_farm','Ag Output per Farm: Tehama','Agout/Far
m')
reg_plot_yearX('livestock_val_div_acre','Livestock per Acre: Tehama','Li
vestock/acre')
reg_plot_yearX('agout_div_acre','Ag Output per Acre: Tehama','Agout/Acr
e')
```





Livestock per Farm: Tehama**Ag Output per Farm: Tehama**

Livestock per Acre: Tehama**Ag Output per Acre: Tehama**