Econometrics

April 26, 2024

# 1 ECONOMETRICS FINAL PROJECT

The flipped Anscombe quartet

By: *Dante Schrantz, Luís Alvarez-Cascos and Miguel Díaz*

# 2 Question to be answered

**Are they all the same? We have been provided with four datasets, each with variables Y and X. That is, variables y1 and x1 form the first dataset, variables y2 and x2 form the second dataset, and so on. Using stata, analyse the relation between variables x and y with the techniques studied in class. That is, from basic statistics (mean, variance, correlation) to simple regression with any specification (linear-linear, log-log, etc.), or even with multiple regression(e.g., including polynomials of the variable x). Determine whether the four datasets are identical in the relation between x and y or not, and justify your answer with the statistical analysis you have carried out**

### 2.0.1 We connect our drive where we store the required files

[71]:
```
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/MyDrive/Colab\ Notebooks/
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
/content/drive/MyDrive/Colab Notebooks
```

# 3 Importing the required libraries

[72]:
```
import csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.metrics import mean_squared_error, mean_absolute_error
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

### 3.0.1 Selection of .xlsx and creation of dataframes

[73]:
```
df = pd.read_excel('final_project.xlsx')
df1 = df.iloc[:, [0, 1]]
df1.columns = ['y', 'x']

df2 = df.iloc[:, [2, 3]]
df2.columns = ['y', 'x']

df3 = df.iloc[:, [4,5]]
df3.columns = ['y','x']

df4 = df.iloc[:, [6,7]]
df4.columns = ['y', 'x']

df1.head()
df2.head()
df3.head()
df4.head()
```

[73]:

| | y | x |
|---|---|---|
| 0 | 8 | 6.58 |
| 1 | 8 | 5.76 |
| 2 | 8 | 7.71 |
| 3 | 8 | 8.84 |
| 4 | 8 | 8.47 |

### 3.0.2 Descriptive statistics of each dataset

[74]:
```
datasets = (df1,df2,df3,df4)

def describe_dataset(df, dataset_name):
    print(f"\nDataset: {dataset_name}")
    print(df.describe())
    print("Correlation:", df['y'].corr(df['x']))

dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

for df,name in zip(datasets,dataset_names):
  describe_dataset(df,name)
```

```
Dataset: Dataset 1
               y          x
count  11.000000  11.000000
mean    9.000000   7.500909
std     3.316625   2.031568
min     4.000000   4.260000
```

```
25%     6.500000   6.315000
50%     9.000000   7.580000
75%    11.500000   8.570000
max    14.000000  10.840000
Correlation: 0.8164205130526425

Dataset: Dataset 2
               y          x
count  11.000000  11.000000
mean    9.000000   7.500909
std     3.316625   2.031657
min     4.000000   3.100000
25%     6.500000   6.695000
50%     9.000000   8.140000
75%    11.500000   8.950000
max    14.000000   9.260000
Correlation: 0.816236487265412

Dataset: Dataset 3
               y          x
count  11.000000  11.000000
mean    9.000000   7.500000
std     3.316625   2.030424
min     4.000000   5.390000
25%     6.500000   6.250000
50%     9.000000   7.110000
75%    11.500000   7.980000
max    14.000000  12.740000
Correlation: 0.816286749614948

Dataset: Dataset 4
               y          x
count  11.000000  11.000000
mean    9.000000   7.500909
std     3.316625   2.030579
min     8.000000   5.250000
25%     8.000000   6.170000
50%     8.000000   7.040000
75%     8.000000   8.190000
max    19.000000  12.500000
Correlation: 0.8165214277339871
```

## 4 T-test for each data set

[75]:
```
datasets = (df1,df2,df3,df4)
dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

def run_t_test(df1, df2, dataset_name1, dataset_name2):
    t_stat_x, p_value_x = stats.ttest_ind(df1['x'], df2['x'])
    t_stat_y, p_value_y = stats.ttest_ind(df1['y'], df2['y'])
    print(f"\033[1mT-test results for 'x' between {dataset_name1} and␣
 ↪{dataset_name2}:\033[0m")
    print(f"t-statistic: {t_stat_x}, p-value: {p_value_x}")
    alpha = 0.05
    if p_value_x < alpha:
      print(f"We reject the null hypothesis; there is a significant difference␣
 ↪between the x values in {dataset_name1} and {dataset_name2}.\n")
    else:
      print(f"We fail to reject the null hypothesis; there is no significant␣
 ↪difference between the x values in {dataset_name1} and {dataset_name2}.\n")
    print(f"\033[1mT-test results for 'y' between {dataset_name1} and␣
 ↪{dataset_name2}:\033[0m")
    print(f"t-statistic: {t_stat_y}, p-value: {p_value_y}")
    if p_value_y < alpha:
      print(f"We reject the null hypothesis; there is a significant difference␣
 ↪between y values in {dataset_name1} and {dataset_name2}.\n")
    else:
      print(f"We fail to reject the null hypothesis; there is no significant␣
 ↪difference between the y values in {dataset_name1} and {dataset_name2}.\n\n")

for i in range(len(datasets)):
    for j in range(i+1, len(datasets)):
        run_t_test(datasets[i], datasets[j], dataset_names[i], dataset_names[j])
```

**T-test results for 'x' between Dataset 1 and Dataset 2:**
```
t-statistic: -1.5012019768382454e-07, p-value: 0.9999998817087112
We fail to reject the null hypothesis; there is no significant difference
between the x values in Dataset 1 and Dataset 2.
```

**T-test results for 'y' between Dataset 1 and Dataset 2:**
```
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference
between the y values in Dataset 1 and Dataset 2.
```

**T-test results for 'x' between Dataset 1 and Dataset 3:**
```
t-statistic: 0.0010498089087928985, p-value: 0.9991727747050596
We fail to reject the null hypothesis; there is no significant difference
between the x values in Dataset 1 and Dataset 3.
```

**T-test results for 'y' between Dataset 1 and Dataset 3:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference between the y values in Dataset 1 and Dataset 3.

**T-test results for 'x' between Dataset 1 and Dataset 4:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference between the x values in Dataset 1 and Dataset 4.

**T-test results for 'y' between Dataset 1 and Dataset 4:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference between the y values in Dataset 1 and Dataset 4.

**T-test results for 'x' between Dataset 2 and Dataset 3:**
t-statistic: 0.001049936136284265, p-value: 0.9991726744527587
We fail to reject the null hypothesis; there is no significant difference between the x values in Dataset 2 and Dataset 3.

**T-test results for 'y' between Dataset 2 and Dataset 3:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference between the y values in Dataset 2 and Dataset 3.

**T-test results for 'x' between Dataset 2 and Dataset 4:**
t-statistic: 1.5015676416033805e-07, p-value: 0.9999998816798977
We fail to reject the null hypothesis; there is no significant difference between the x values in Dataset 2 and Dataset 4.

**T-test results for 'y' between Dataset 2 and Dataset 4:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference between the y values in Dataset 2 and Dataset 4.

**T-test results for 'x' between Dataset 3 and Dataset 4:**
t-statistic: -0.0010500647779789261, p-value: 0.9991725730860987
We fail to reject the null hypothesis; there is no significant difference between the x values in Dataset 3 and Dataset 4.

**T-test results for 'y' between Dataset 3 and Dataset 4:**
t-statistic: 0.0, p-value: 1.0
We fail to reject the null hypothesis; there is no significant difference

between the y values in Dataset 3 and Dataset 4.

# 5 Regression Analysis

### 5.0.1 Linear Regression

[76]:
```
datasets = (df1, df2, df3, df4)
dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

def anova(df, dataset_name):
    model = ols('y ~ x', data=df).fit()
    anova_table = sm.stats.anova_lm(model, typ=2)
    print(f'\n\033[1mANOVA and Coefficient Table of Linear-Linear regression␣
 ↪for {dataset_name}:\033[0m\n')
    print(anova_table)
    print(model.summary().tables[0])
    print(model.summary().tables[1])
    print()

for df, name in zip(datasets, dataset_names):
    anova(df, name)
```

```
ANOVA and Coefficient Table of Linear-Linear regression for Dataset 1:

            sum_sq   df          F   PR(>F)
x         73.31967  1.0  17.989943  0.00217
Residual  36.68033  9.0        NaN      NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.667
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                     17.99
Date:                Fri, 26 Apr 2024   Prob (F-statistic):            0.00217
Time:                        09:50:29   Log-Likelihood:                -22.232
No. Observations:                  11   AIC:                             48.46
Df Residuals:                       9   BIC:                             49.26
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.9975      2.434     -0.410      0.692      -6.505       4.510
x              1.3328      0.314      4.241      0.002       0.622       2.044
```

```
==============================================================================
```

**ANOVA and Coefficient Table of Linear-Linear regression for Dataset 2:**

```
            sum_sq   df          F    PR(>F)
x         73.28662  1.0  17.965646  0.002179
Residual  36.71338  9.0        NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.666
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                     17.97
Date:                Fri, 26 Apr 2024   Prob (F-statistic):            0.00218
Time:                        09:50:29   Log-Likelihood:                -22.237
No. Observations:                  11   AIC:                             48.47
Df Residuals:                       9   BIC:                             49.27
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.9948      2.435     -0.408      0.692      -6.504       4.514
x              1.3325      0.314      4.239      0.002       0.621       2.044
==============================================================================
```

**ANOVA and Coefficient Table of Linear-Linear regression for Dataset 3:**

```
             sum_sq   df          F    PR(>F)
x         73.295646  1.0  17.972277  0.002176
Residual  36.704354  9.0        NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.666
Model:                            OLS   Adj. R-squared:                  0.629
Method:                 Least Squares   F-statistic:                     17.97
Date:                Fri, 26 Apr 2024   Prob (F-statistic):            0.00218
Time:                        09:50:29   Log-Likelihood:                -22.236
No. Observations:                  11   AIC:                             48.47
Df Residuals:                       9   BIC:                             49.27
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
```

```
Intercept     -1.0003      2.436     -0.411      0.691      -6.511       4.511
x              1.3334      0.315      4.239      0.002       0.622       2.045
==============================================================================
```

**ANOVA and Coefficient Table of Linear-Linear regression for Dataset 4:**

```
             sum_sq   df          F    PR(>F)
x         73.337797  1.0  18.003287  0.002165
Residual  36.662203  9.0        NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.667
Model:                            OLS   Adj. R-squared:                  0.630
Method:                 Least Squares   F-statistic:                     18.00
Date:                Fri, 26 Apr 2024   Prob (F-statistic):            0.00216
Time:                        09:50:29   Log-Likelihood:                -22.230
No. Observations:                  11   AIC:                             48.46
Df Residuals:                       9   BIC:                             49.25
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.0036      2.435     -0.412      0.690      -6.512       4.505
x              1.3337      0.314      4.243      0.002       0.623       2.045
==============================================================================
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
```

### 5.0.2 Logarithmic Regression

[77]:
```python
datasets = (df1, df2, df3, df4)
dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

def anova(df, dataset_name):
    df['log_x'] = np.log(df['x'])
    df['log_y'] = np.log(df['y'])
    model = ols('log_y ~ log_x', data=df).fit()
    anova_table = sm.stats.anova_lm(model, typ=2)

    print(f'\n\033[1mANOVA and Coefficient Table of Log-Log Regression for␣
 ↪{dataset_name}:\033[0m\n')
    print(anova_table)
    print(model.summary().tables[0])
    print(model.summary().tables[1])
    print()

for df, name in zip(datasets, dataset_names):
    anova(df, name)
```

**ANOVA and Coefficient Table of Log-Log Regression for Dataset 1:**

```
            sum_sq   df          F   PR(>F)
log_x     1.167225  1.0  21.527713  0.00122
Residual  0.487977  9.0        NaN      NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                  log_y   R-squared:                       0.705
Model:                            OLS   Adj. R-squared:                  0.672
Method:                 Least Squares   F-statistic:                     21.53
Date:                Fri, 26 Apr 2024   Prob (F-statistic):            0.00122
Time:                        09:50:29   Log-Likelihood:                 1.5263
No. Observations:                  11   AIC:                            0.9474
Df Residuals:                       9   BIC:                             1.743
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

```
------------------------------------------------------------------------------
Intercept     -0.2006      0.507     -0.396      0.701      -1.347       0.945
log_x          1.1765      0.254      4.640      0.001       0.603       1.750
==============================================================================
```

**ANOVA and Coefficient Table of Log-Log Regression for Dataset 2:**

```
            sum_sq   df          F    PR(>F)
log_x     1.291891  1.0  32.002954  0.000311
Residual  0.363311  9.0        NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                  log_y   R-squared:                       0.781
Model:                            OLS   Adj. R-squared:                  0.756
Method:                 Least Squares   F-statistic:                     32.00
Date:                Fri, 26 Apr 2024   Prob (F-statistic):           0.000311
Time:                        09:50:29   Log-Likelihood:                 3.1488
No. Observations:                  11   AIC:                            -2.298
Df Residuals:                       9   BIC:                            -1.502
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0777      0.367      0.212      0.837      -0.753       0.909
log_x          1.0408      0.184      5.657      0.000       0.625       1.457
==============================================================================
```

**ANOVA and Coefficient Table of Log-Log Regression for Dataset 3:**

```
            sum_sq   df          F    PR(>F)
log_x     1.235225  1.0  26.470539  0.000607
Residual  0.419977  9.0        NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                  log_y   R-squared:                       0.746
Model:                            OLS   Adj. R-squared:                  0.718
Method:                 Least Squares   F-statistic:                     26.47
Date:                Fri, 26 Apr 2024   Prob (F-statistic):           0.000607
Time:                        09:50:29   Log-Likelihood:                 2.3517
No. Observations:                  11   AIC:                           -0.7033
Df Residuals:                       9   BIC:                           0.09249
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.7954      0.572     -1.391      0.198      -2.089       0.498
log_x          1.4710      0.286      5.145      0.001       0.824       2.118
==============================================================================
```

**ANOVA and Coefficient Table of Log-Log Regression for Dataset 4:**

```
            sum_sq   df         F    PR(>F)
log_x     0.356676  1.0  9.922214  0.011738
Residual  0.323525  9.0       NaN       NaN
                            OLS Regression Results
==============================================================================
Dep. Variable:                  log_y   R-squared:                       0.524
Model:                            OLS   Adj. R-squared:                  0.472
Method:                 Least Squares   F-statistic:                     9.922
Date:                Fri, 26 Apr 2024   Prob (F-statistic):             0.0117
Time:                        09:50:29   Log-Likelihood:                 3.7867
No. Observations:                  11   AIC:                            -3.573
Df Residuals:                       9   BIC:                            -2.778
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6419      0.485      1.324      0.218      -0.455       1.738
log_x          0.7636      0.242      3.150      0.012       0.215       1.312
==============================================================================
```

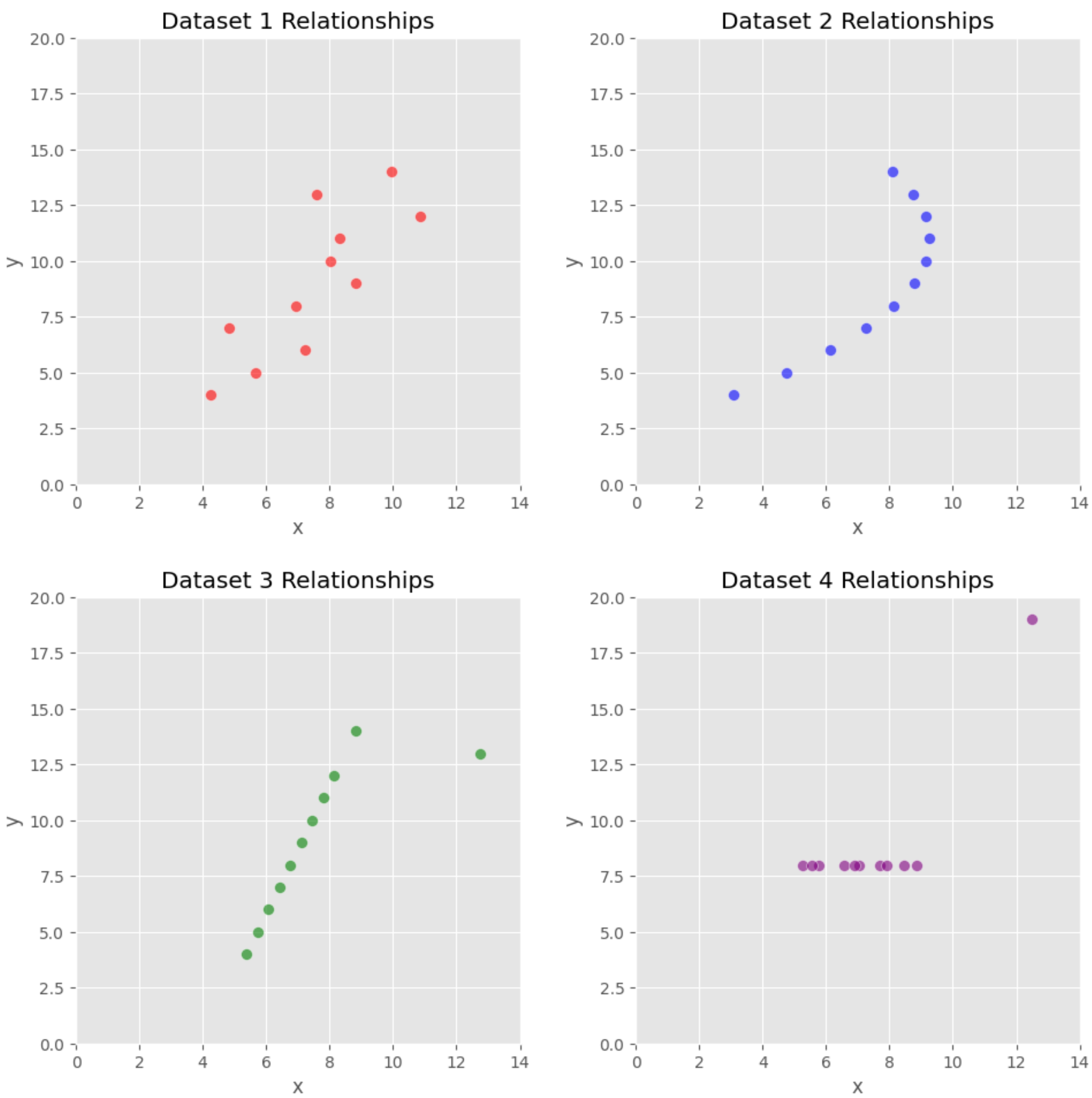
```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 … continuing "
```

# 6 Data Visualization

### 6.0.1 Standard Visualization

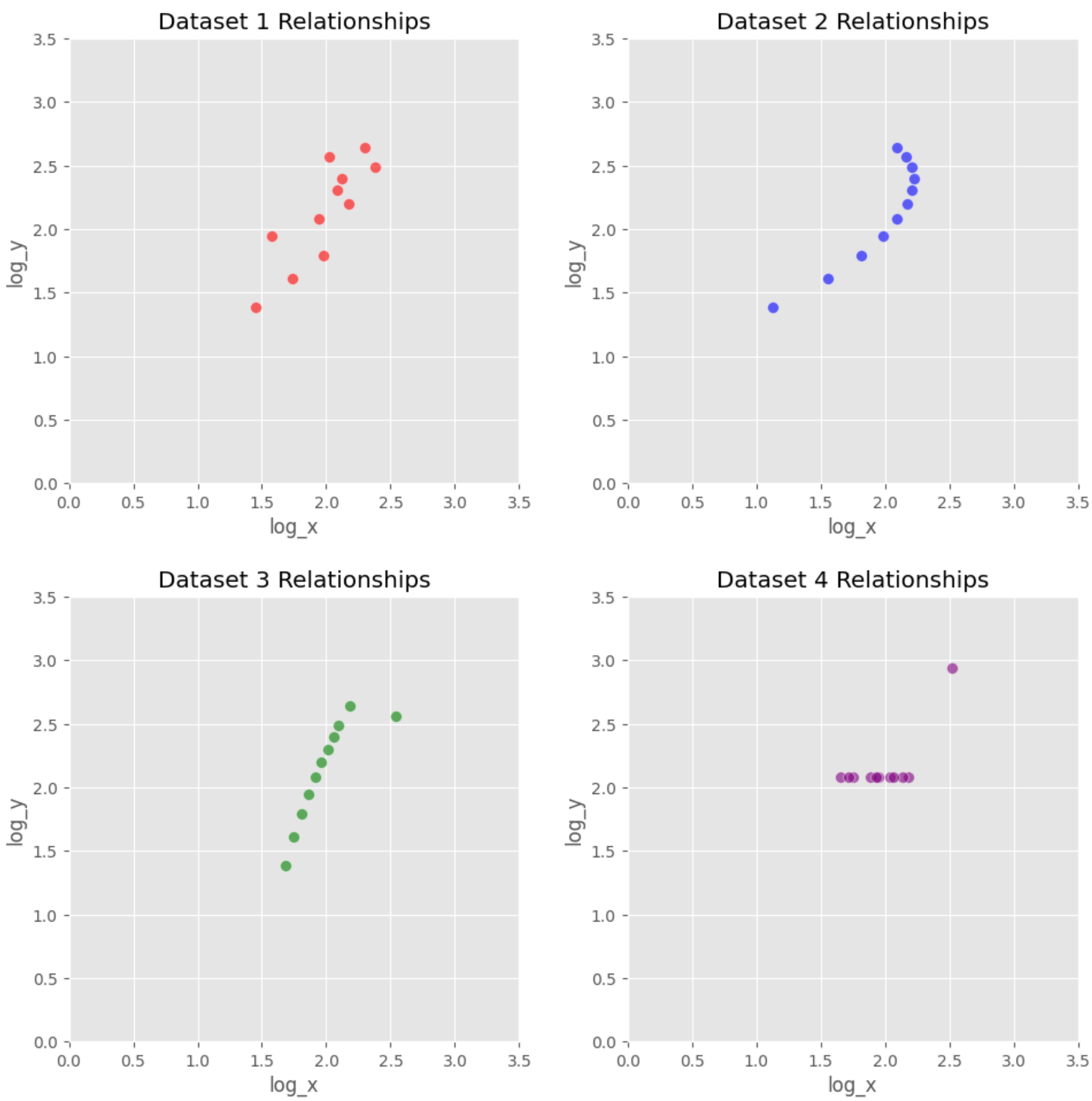
[78]:
```
colors = ['red', 'blue', 'green', 'purple']
plt.style.use('ggplot')

datasets = (df1, df2, df3, df4)
dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

fig, axs = plt.subplots(2, 2, figsize=(10, 10))
axs = axs.ravel()

for i, (df, name) in enumerate(zip(datasets, dataset_names)):
    axs[i].scatter(df['x'], df['y'], color=colors[i], alpha=0.6,␣
 ↪edgecolors='w', s=50)
    axs[i].set_xlabel("x")
    axs[i].set_ylabel("y")
    axs[i].set_title(f"{name} Relationships")
    axs[i].set_xlim(0, 14)
    axs[i].set_ylim(0, 20)


plt.tight_layout(pad=2)
plt.show()
```




### 6.0.2 Logarithmic Visualization

[79]:
```
colors = ['red', 'blue', 'green', 'purple']
plt.style.use('ggplot')

datasets = (df1, df2, df3, df4)
dataset_names = ("Dataset 1", "Dataset 2", "Dataset 3", "Dataset 4")

fig, axs = plt.subplots(2, 2, figsize=(10, 10))
axs = axs.ravel()

for i, (df, name) in enumerate(zip(datasets, dataset_names)):
    df['log_x'] = np.log(df['x'])
```

```
    df['log_y'] = np.log(df['y'])
    axs[i].scatter(df['log_x'], df['log_y'], color=colors[i], alpha=0.6, edgecolors='w', s=50)
    axs[i].set_xlabel("log_x")
    axs[i].set_ylabel("log_y")
    axs[i].set_title(f"{name} Relationships")
    axs[i].set_xlim(0, 3.5)
    axs[i].set_ylim(0, 3.5)

plt.tight_layout(pad=2)
plt.show()
```



# 7 Conclusion, Answering the main question: Are they all the same

In order to gain a better understanding of each data set we must create descriptive statistics between each dataset and compare the results. After comparing the results of each individual dataset it is important to formulate T-Tests and create Analysis of Variation (ANOVA) tables to dig deeper into the data. We have also decided to manipulate the data in a way to attempt to get a better understanding of the data. We have done so by applying logarithmic regression to the datasets. Lets start by analyzing the count, mean and standard deviation.

## 7.1 Analyzing the Statistical Data

The *statistical analysis* reveals some bizarre similarities among the four datasets:

**Count, Mean, Standard Deviation and Correlation Coefficient**: When looking at the mean, we get the same mean across all datasets for X (7.50) and Y (11.00) which would indicate that the datasets are the same. The count for each set is 11 observations. The standard deviation for Y across all datasets is ~3.316 and for X it is ~2.030 with the last 2 datasets being ~2.031. This being said we also have nearly identical correlation values at ~0.816, but when we make sure to not round we can see that there is a slight variation in actual correlation but the difference is miniscule.

**Datasets 1, 2, and 3** have the same minimum X value (4), maximum X value (14), and mean X value (9), which explains why the standard deviation of X is identical for these datasets. The minimum, maximum, and mean Y values are also nearly identical, contributing to the similar standard deviations for Y.

**Dataset 4** has a significantly different minimum X value (8) and maximum X value (19), yet the mean X value is the same (9) due to the dataset's concentration of X values around 8. The similar mean and standard deviation values for Y across all datasets suggests symmetry around the mean, but this dataset's spread in the X values is not reflected in the mean or standard deviation due to the concentration of values.

The **correlation coefficient** indicates the strength and direction of the linear relationship between X and Y, not the shape of the relationship or the distribution of data points along the axis. This explains why the correlation coefficient is similar across datasets even though the scatter plots suggest different relationships.

Despite the basic statistics and correlation being nearly identical, the relationships in the datasets are not the same. This comes to show how descriptive statistics can help one to understand the data but it does not paint the entire picture. As much as numbers don't lie, they can be misleading at times and can create false realities for those interpreting the data. The main statistical properties are almost equivalent. If one makes the mistake to round the data, they will be unable to see that there is slight variation in the data. Anscombe's quartet was created to confuse statisticians and prove that as much as descriptive statistics can be helpful one always needs to graph the data to get a full interpretation.

While basic statistics and the correlation coefficient can help us draw conclusions from datasets, it is very important to have a visual representation for the data. These simple descriptive statistics techniques can definitely help to gain a better understanding of the data, but it is clear that it can be misleading and create false conclusions.

> This issue shows the importance of visualizing data and utilizing more tools than basic descriptive statistics to analyze data.

**T-test Interpretation**: As shown in the results of the T-tests, we fail to reject the null hypothesis. There is no mean statistical evidence that indicates a difference in datasets. Why is that? A t-test compares the mean of the dataset and as we mentioned in the first point, all of the means both on the X and Y axis are the same at ~7.5 and ~9 respectively. This being said, even the T-tests are incapable of finding a difference between the datasets. As much as a T-test can be an indicator of difference, it lacks robustness as it is only comparing the mean of the datasets.

---

We have decided to use regression tools in order to gain a better understanding of the data:

**Linear Regression**:

R-squared and intercept: By conducting the linear regression, we conclude that although the four datasets share the same slope ~ 1.33 and intercept ~ 0.99. The R-squared not only varies little between the four of them but it provides a 66% of regression data certainty. The same happens when looking at the adjusted r-squared from dataset 4 where it becomes just slightly larger at ~ 0.630. With what we previously stated, it can be concluded that there is almost no variation between the four datasets and that they give very similar values. This is most definitely not the case and will be discussed in the graphical analysis

**Logarithmic Regression**:

The logarithmic regression in this case has more to talk about. We have decided to take the Log-Log approach which applies logarithms to both sides of the dataset. Applying logs to both sides will create percentage changes between each data point and attempt to remove the linearity that we have observed originally. This will hopefully help us to better prove that despite the original descriptive statistics and linear regression that the data is in fact different.

Examining the R-squared of the four data sets, it is observed that data sets 1, 2 and 3 are 70.5%, 78.1 and 74.6%, with data set 3 being the most descriptive of all. The r-squared for dataset 4 is 47.2%. This indicates that after transforming the data by logarithms we have a more descriptive and predictive model. Such a low value of prediction for dataset 4 indicates that the model has become less predictive and we should not transform the data. For the adjusted R-Squared also see that there is variation among the four Datasets following the same order of prediction explained previously for the case of the unadjusted R-Squared.

Although the results obtained in the case of linear regression maintain a high degree of similarity, this does not hold for the case of logarithmic regression, where the R-squared results have different levels of prediction. As stated earlier we tried to use logarithms to normalize the data which more or less worked. The data points became more concentrated and we were able to see that the relationships had changed. Taking logarithms also removed the linearity of the graphs and turned them a bit more into exponential graphs which was our initial intentions. This is explained by the relationship between the X and Y variables in each model. It makes sense that the linear regression gives very similar data with little deviation while in the case of the logarithmic regression the dispersion is explained by a relational change between the variables X and Y in each model.

### 7.2 Analyzing the graphs

**Dataset 1** The first dataset shows a positive correlation between X and Y because when X increases Y also increases. We do note however that there is not an extremely strong correlation as the points on the graph tend to have a decent amount of space for each other. For example when X~7.5 Y takes the values of ~13 and ~6 indicating that there is a large amount of variance in the possible values of Y at the same point in x.

**Dataset 2** This second dataset is quite interesting. At first we note that the relationship between X and Y is positive. For the beginning of the graph we would say that Y is almost exponentially corelated to X until we reach the 5th point on the graph. This is when things start to change and Y starts to take several values for points in X. The graph almost looks like a hairpin turn as the points go back onto themselves. The depency on X for the values of Y is not very clear in this graph yet we do see that as X increases Y also increases.

**Dataset 3** This dataset appears to have the strongest linear relationship between all of the graphs. The slope on this graph appears to be around 3 with a perfectly linear relationship and a negative intercept. This being said there is one point that is an outlier at around (12.5, 12.5) which does throw off the data. If we were to not pay attention to this outlier we would have a perfectly linear relationship within the data.

**Dataset 4** This dataset has no relationship between x and y. No matter the value of X, Y is always 8. This creates the horizontal line that we see on the graph. Like in graph 3 we do have an outlier at (12.5, 19). This being said it is the only outlier in the graph so we can continue with our initial assumption that this graph is horizontal.

---

### 7.3 Conclusion

After analyzing the data in various ways it is clear the Anscombe's quartet was created to show how basic descriptive statistics are thorough ways to analyze data but they are not the full picture. Although the datasets are different, the descriptive statistics done for each dataset demonstrate similarities for the mean, standard deviation, and correlation coefficient. These statistics indicate similarities between the datasets yet this is not the case. Again when we run the linear regression we get similarities between the datasets which in theory should not be the case. These descriptive statistics illustrate an incorrect situation. Although this is the case, when applying logarithms to the variables we get more variance between the datasets means but it is not enough to say confidently that the datasets are different. Throughout the entire statistical analysis we conclude that the data is the same but the most vital part in analyzing data is the visualization. Continuing with our analysis the data visualization is the final step to understand the data. The visualization magnifies the differences between the datasets. After observing the visualization we can confidently conclude that even though all of the descriptive statistics indicate similarities, the datasets are different. There is no similarity in the behavior of any dataset and they could not be any more different from each other. This differences would be almost impossible to see had it not been for our final step of visualization. Once again it is clear that Anscombe's quartet was created to indicate the importance of visualization and how descriptive statistics can many times be misleading.

## 8 Conversion to PDF

[80]:
```
!jupyter-nbconvert --to PDF "/content/drive/MyDrive/Colab Notebooks/
↪Econometrics.ipynb"
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab
Notebooks/Econometrics.ipynb to PDF
[NbConvertApp] Support files will be in Econometrics_files/
[NbConvertApp] Making directory ./Econometrics_files
[NbConvertApp] Making directory ./Econometrics_files
[NbConvertApp] Writing 77898 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 168746 bytes to /content/drive/MyDrive/Colab
Notebooks/Econometrics.pdf
```