5000 Filmes DataSet

In [2]:

```
import pandas as pd
import numpy as np
import os
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
# !pip install seaborn==0.9.0
print(sns.__version__)
```

```
0.11.1
```

In [4]:

```
#import sys
#!{sys.executable} -m pip install --user
```

In [5]:

```
#Lê o caminho atual: os.path.join(current_path,'ml-latest-small',"rating.csv" )
current_path = os.getcwd()

movies_db = pd.read_csv(os.path.join(current_path,'tmdb','tmdb_5000_movies.csv'))

credits_db = pd.read_csv(os.path.join(current_path,'tmdb','tmdb_5000_credits.csv'))
```

In [6]:

```
movies_db.head()
```

Out[6]:

	budget	genres	homepage	id	keywords	original
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	

In [7]:

```
movies_db.original_language.unique()
```

Out[7]:

```
array(['en', 'ja', 'fr', 'zh', 'es', 'de', 'hi', 'ru', 'ko', 'te', 'cn',
       'it', 'nl', 'ta', 'sv', 'th', 'da', 'xx', 'hu', 'cs', 'pt', 'is',
       'tr', 'nb', 'af', 'pl', 'he', 'ar', 'vi', 'ky', 'id', 'ro', 'fa',
       'no', 'sl', 'ps', 'el'], dtype=object)
```

In [8]:

```
# primeiro grau
# segundo grau
# terceiro grau
# 1 grau < 2 grau < 3 grau # categoria ordinalb
```

In [9]:

```
# budget => orçamento => quantitativa continuo
```

In [10]:

```
# quantidade de votos => 1, 2, 3, 4, não tem 2.5 voto.
# notas movielens => 0.5, 1, 1.5 ..., 5 não tem 2.7
```

In [11]:

```
movies_db.original_language
```

Out[11]:

```
0       en
1       en
2       en
3       en
4       en
        ..
4798    es
4799    en
4800    en
4801    en
4802    en
Name: original_language, Length: 4803, dtype: object
```

In [12]:

```
movies_db.original_language.value_counts()
```

Out[12]:

```
en    4505
fr      70
es      32
de      27
zh      27
hi      19
ja      16
it      14
cn      12
ru      11
ko      11
pt       9
da       7
sv       5
nl       4
fa       4
th       3
he       3
ro       2
cs       2
ta       2
id       2
ar       2
sl       1
el       1
nb       1
hu       1
ky       1
no       1
ps       1
te       1
af       1
pl       1
vi       1
tr       1
is       1
xx       1
Name: original_language, dtype: int64
```

In [13]:

```
movies_db.original_language.value_counts().index
```

Out[13]:

```
Index(['en', 'fr', 'es', 'de', 'zh', 'hi', 'ja', 'it', 'cn', 'ru', 'ko', 'p
t',
       'da', 'sv', 'nl', 'fa', 'th', 'he', 'ro', 'cs', 'ta', 'id', 'ar', 's
l',
       'el', 'nb', 'hu', 'ky', 'no', 'ps', 'te', 'af', 'pl', 'vi', 'tr', 'i
s',
       'xx'],
      dtype='object')
```

In [14]:

```
movies_db.original_language.value_counts().values
```

Out[14]:

```
array([4505,   70,   32,   27,   27,   19,   16,   14,   12,   11,   11,
          9,    7,    5,    4,    4,    3,    3,    2,    2,    2,    2,
          2,    1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
          1,    1,    1,    1], dtype=int64)
```

In [15]:

```
movies_db.original_language.value_counts().to_frame()
```

Out[15]:

	original_language
en	4505
fr	70
es	32
de	27
zh	27
hi	19
ja	16
it	14
cn	12
ru	11
ko	11
pt	9
da	7
sv	5
nl	4
fa	4
th	3
he	3
ro	2
cs	2
ta	2
id	2
ar	2
sl	1
el	1
nb	1
hu	1
ky	1
no	1
ps	1
te	1
af	1
pl	1
vi	1

	original_language
tr	1
is	1
xx	1

In [16]:

```
movies_db.original_language.value_counts().to_frame().reset_index()
```

Out[16]:

	index	original_language
0	en	4505
1	fr	70
2	es	32
3	de	27
4	zh	27
5	hi	19
6	ja	16
7	it	14
8	cn	12
9	ru	11
10	ko	11
11	pt	9
12	da	7
13	sv	5
14	nl	4
15	fa	4
16	th	3
17	he	3
18	ro	2
19	cs	2
20	ta	2
21	id	2
22	ar	2
23	sl	1
24	el	1
25	nb	1
26	hu	1
27	ky	1
28	no	1
29	ps	1
30	te	1
31	af	1
32	pl	1
33	vi	1

	index	original_language
34	tr	1
35	is	1
36	xx	1

In [17]:

```
contagem_de_lingua = movies_db.original_language.value_counts().to_frame().reset_index()
contagem_de_lingua.columns = ['original_language','total']
contagem_de_lingua.head()
```

Out[17]:

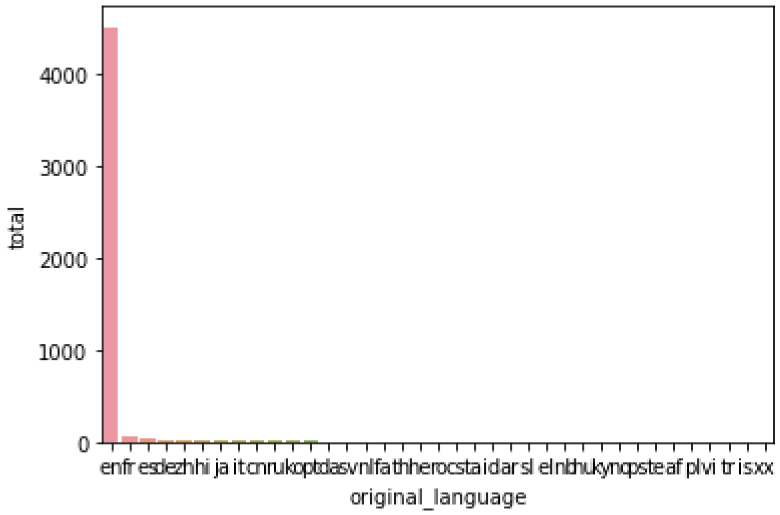
	original_language	total
0	en	4505
1	fr	70
2	es	32
3	de	27
4	zh	27

In [18]:

```
sns.barplot(x="original_language", y='total', data = contagem_de_lingua)
```

Out[18]:

```
<AxesSubplot:xlabel='original_language', ylabel='total'>
```

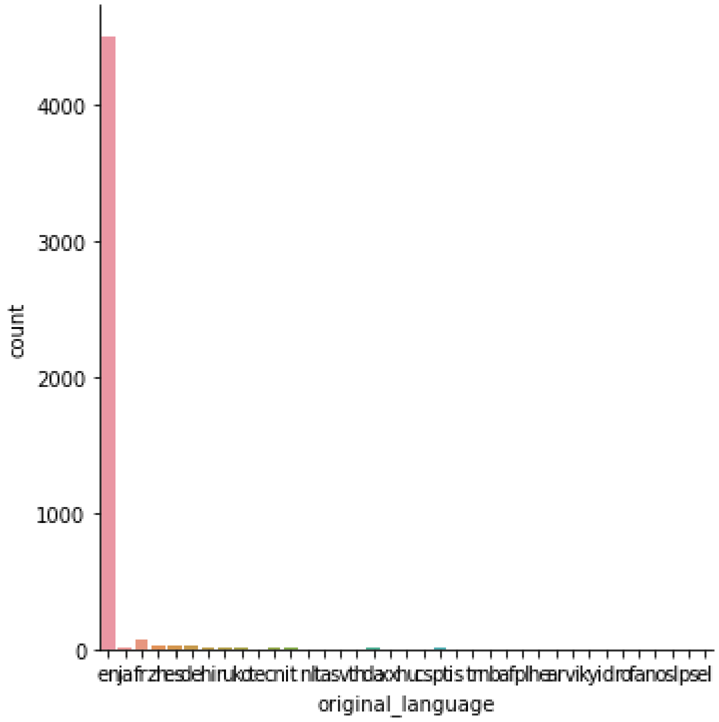


In [19]:

```
sns.catplot(x="original_language", kind="count", data=movies_db,)
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x20bf28527c8>
```

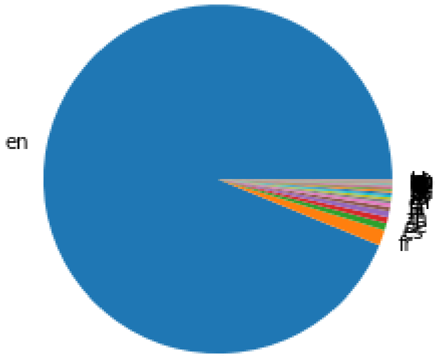


In [20]:

```
plt.pie(contagem_de_lingua['total'], labels = contagem_de_lingua['original_language'])
```

Out[20]:

```
([<matplotlib.patches.Wedge at 0x20bf29cb848>,
  <matplotlib.patches.Wedge at 0x20bf29d2488>,
  <matplotlib.patches.Wedge at 0x20bf29d2e48>,
  <matplotlib.patches.Wedge at 0x20bf29d78c8>,
  <matplotlib.patches.Wedge at 0x20bf29df388>,
  <matplotlib.patches.Wedge at 0x20bf29dfec8>,
  <matplotlib.patches.Wedge at 0x20bf29e6948>,
  <matplotlib.patches.Wedge at 0x20bf29ed388>,
  <matplotlib.patches.Wedge at 0x20bf29df348>,
  <matplotlib.patches.Wedge at 0x20bf29dfe88>,
  <matplotlib.patches.Wedge at 0x20bf29a8ec8>,
  <matplotlib.patches.Wedge at 0x20bf29fab88>,
  <matplotlib.patches.Wedge at 0x20bf2a01608>,
  <matplotlib.patches.Wedge at 0x20bf2a01f88>,
  <matplotlib.patches.Wedge at 0x20bf2a08ac8>,
  <matplotlib.patches.Wedge at 0x20bf2a0f548>,
  <matplotlib.patches.Wedge at 0x20bf2a0ff88>,
  <matplotlib.patches.Wedge at 0x20bf2a15a08>,
  <matplotlib.patches.Wedge at 0x20bf2a1a488>,
  <matplotlib.patches.Wedge at 0x20bf2a1aec8>,
  <matplotlib.patches.Wedge at 0x20bf2a22948>,
  <matplotlib.patches.Wedge at 0x20bf2a283c8>,
  <matplotlib.patches.Wedge at 0x20bf2a28e08>,
  <matplotlib.patches.Wedge at 0x20bf2a2f888>,
  <matplotlib.patches.Wedge at 0x20bf2a34308>,
  <matplotlib.patches.Wedge at 0x20bf2a34d48>,
  <matplotlib.patches.Wedge at 0x20bf2a3d7c8>,
  <matplotlib.patches.Wedge at 0x20bf2a43248>,
  <matplotlib.patches.Wedge at 0x20bf2a43c88>,
  <matplotlib.patches.Wedge at 0x20bf2a4a708>,
  <matplotlib.patches.Wedge at 0x20bf2a4f188>,
  <matplotlib.patches.Wedge at 0x20bf2a4fbc8>,
  <matplotlib.patches.Wedge at 0x20bf2a56648>,
  <matplotlib.patches.Wedge at 0x20bf2a56fc8>,
  <matplotlib.patches.Wedge at 0x20bf2a5db08>,
  <matplotlib.patches.Wedge at 0x20bf2a64588>,
  <matplotlib.patches.Wedge at 0x20bf2a64fc8>],
 [Text(-1.0791697536499925, 0.2130554923183512, 'en'),
  Text(1.0355355017029462, -0.3710339940124459, 'fr'),
  Text(1.0579676486019882, -0.3011718023181785, 'es'),
  Text(1.0687996606645356, -0.26012936274741094, 'de'),
  Text(1.077319105706255, -0.22222406260195313, 'zh'),
  Text(1.0835167978583342, -0.18971386021801853, 'hi'),
  Text(1.0875756432724297, -0.16486121484618815, 'ja'),
  Text(1.0906010773146022, -0.14348968659882622, 'it'),
  Text(1.092883487371409, -0.12492270822755745, 'cn'),
  Text(1.0946390911069936, -0.10846778425161549, 'ru'),
  Text(1.0960865535188649, -0.09270527058984593, 'ko'),
  Text(1.0972054830031333, -0.07835896928789601, 'pt'),
  Text(1.097965443340663, -0.06687215586282344, 'da'),
  Text(1.0984565010300316, -0.05825217030171998, 'sv'),
  Text(1.0987803851616647, -0.0517847968421653, 'nl'),
  Text(1.0990363161210686, -0.04603450713357274, 'fa'),
  Text(1.0992355702663055, -0.04100196411527794, 'th'),
  Text(1.0993880184234357, -0.03668766750546649, 'he'),
  Text(1.0995021239019234, -0.03309198596578415, 'ro'),
  Text(1.099584941078101, -0.03021518416739545, 'cs'),
  Text(1.0996602312343366, -0.027338175536150495, 'ta'),
  Text(1.099727993855245, -0.024460979766119193, 'id'),
  Text(1.0997882284769684, -0.02158361655264929, 'ar'),
  Text(1.0998284639438185, -0.01942549610642471, 'sl'),
  Text(1.0998529348820232, -0.01798670707495573, 'el'),
  Text(1.0998755236058106, -0.01654788726224571, 'nb'),
  Text(1.0998962300765243, -0.01510903913059323, 'hu'),
  Text(1.0999150542587282, -0.013670165142345335, 'ky'),
  Text(1.0999319961202083, -0.012231267759896247, 'no'),
  Text(1.0999470556319713, -0.01079234944567632, 'ps'),
  Text(1.099960232768245, -0.00935341266215563, 'te'),
  Text(1.0999715275064792, -0.007914459871831963, 'af'),
  Text(1.0999809398273452, -0.006475493537234394, 'pl'),
  Text(1.0999884697147349, -0.005036516120911278, 'vi'),
  Text(1.0999941171557621, -0.0035975300854338356, 'tr'),
  Text(1.0999978821407626, -0.0021585378933851127, 'is'),
  Text(1.0999997646632929, -0.0007195420073586872, 'xx')])
```



In [21]:

```
total_por_lingua_outros_filmes = movies_db.query("original_language != 'en'").original_lang
total_por_lingua_outros_filmes
```

Out[21]:

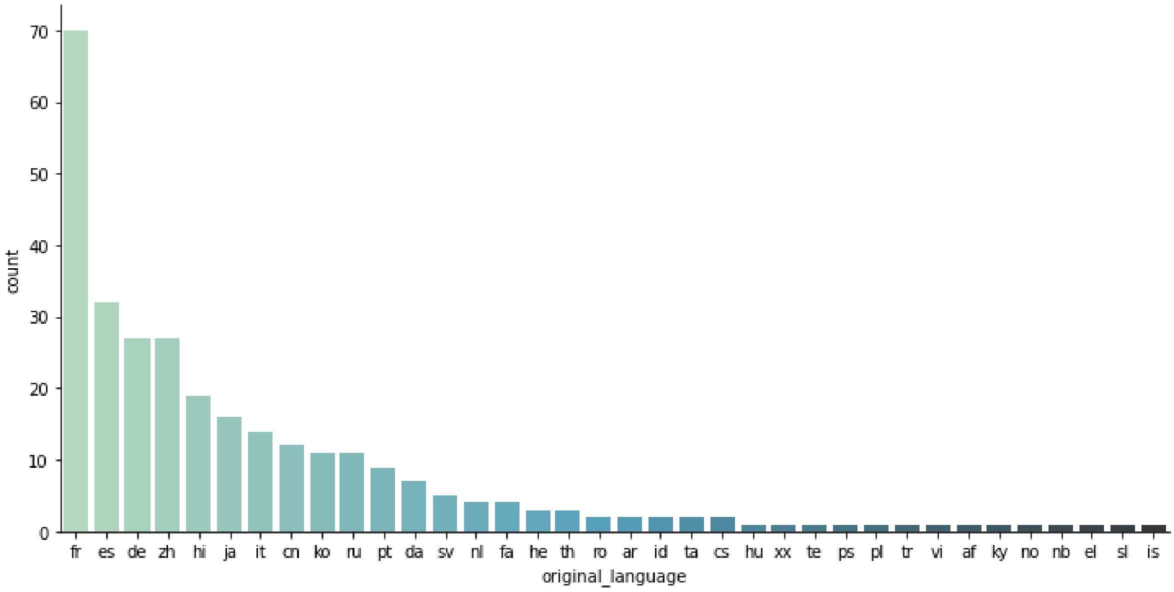
```
fr    70
es    32
de    27
zh    27
hi    19
ja    16
it    14
cn    12
ko    11
ru    11
pt     9
da     7
sv     5
nl     4
fa     4
he     3
th     3
ro     2
ar     2
id     2
ta     2
cs     2
hu     1
xx     1
te     1
ps     1
pl     1
tr     1
vi     1
af     1
ky     1
no     1
nb     1
el     1
sl     1
is     1
Name: original_language, dtype: int64
```

In [22]:

```
filmes_sem_lingua_original_em_ingles = movies_db.query("original_language != 'en'")
sns.catplot(x='original_language', kind='count', data=filmes_sem_lingua_original_em_ingles,
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x20bf119c648>
```



In [23]:

```
total_por_lingua = movies_db['original_language'].value_counts()
total_geral = total_por_lingua.sum()
print("total geral",total_geral)
total_de_ingles = total_por_lingua.loc["en"] # loc localiza 4505
total_do_resto = total_geral-total_de_ingles
print(total_de_ingles, total_do_resto)
```

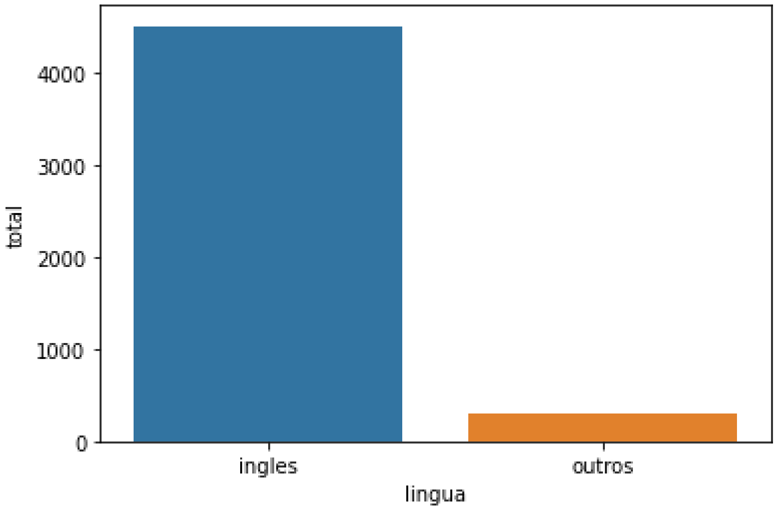
```
total geral 4803
4505 298
```

In [24]:

```
dados = {
    'lingua':['ingles', 'outros'],
    'total':[total_de_ingles, total_do_resto]
}
dados = pd.DataFrame(dados)
dados
sns.barplot(x="lingua", y="total", data = dados)
```

Out[24]:

```
<AxesSubplot:xlabel='lingua', ylabel='total'>
```



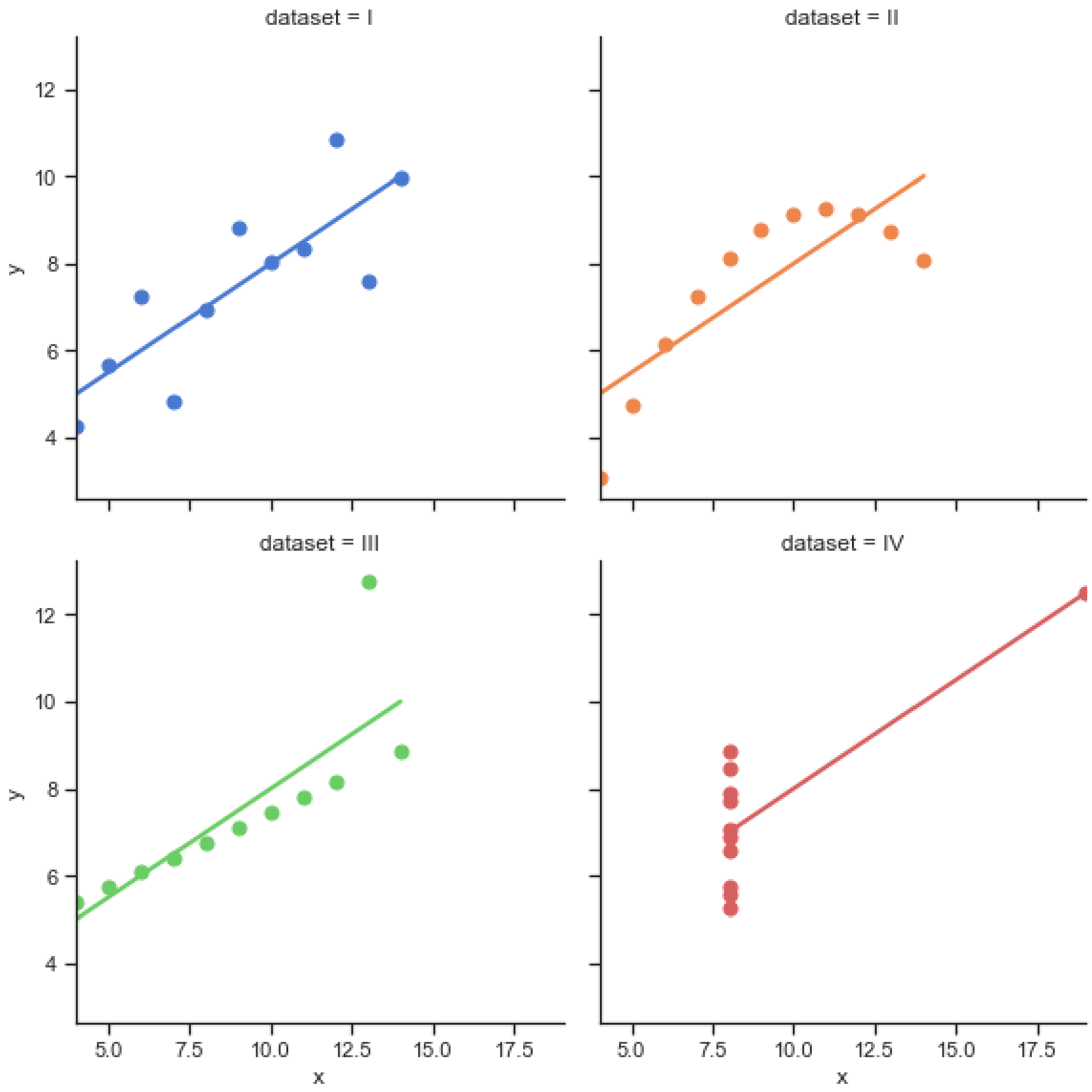
In [25]:

```
sns.set(style="ticks")
df = sns.load_dataset("anscombe")
print(df.head())
sns.lmplot(x="x", y="y", col="dataset", hue="dataset", data=df,
           col_wrap=2, ci=None, palette="muted", height=4,
           scatter_kws={"s": 50, "alpha": 1})
```

```
  dataset     x     y
0       I  10.0  8.04
1       I   8.0  6.95
2       I  13.0  7.58
3       I   9.0  8.81
4       I  11.0  8.33
```

Out[25]:

```
<seaborn.axisgrid.FacetGrid at 0x20bf2d31e08>
```


dataset = I
dataset = II
dataset = III
dataset = IV
y
y
x
x