????2

September 23, 2019

[51]:

```
#Exercise 1
import pandas as pd
df = pd.read_csv('https://raw.githubusercontent.com/nickeubank/
 ↪practicaldatascience/master/Example_Data/Datasaurus.txt', delimiter='\t')
```

[52]:

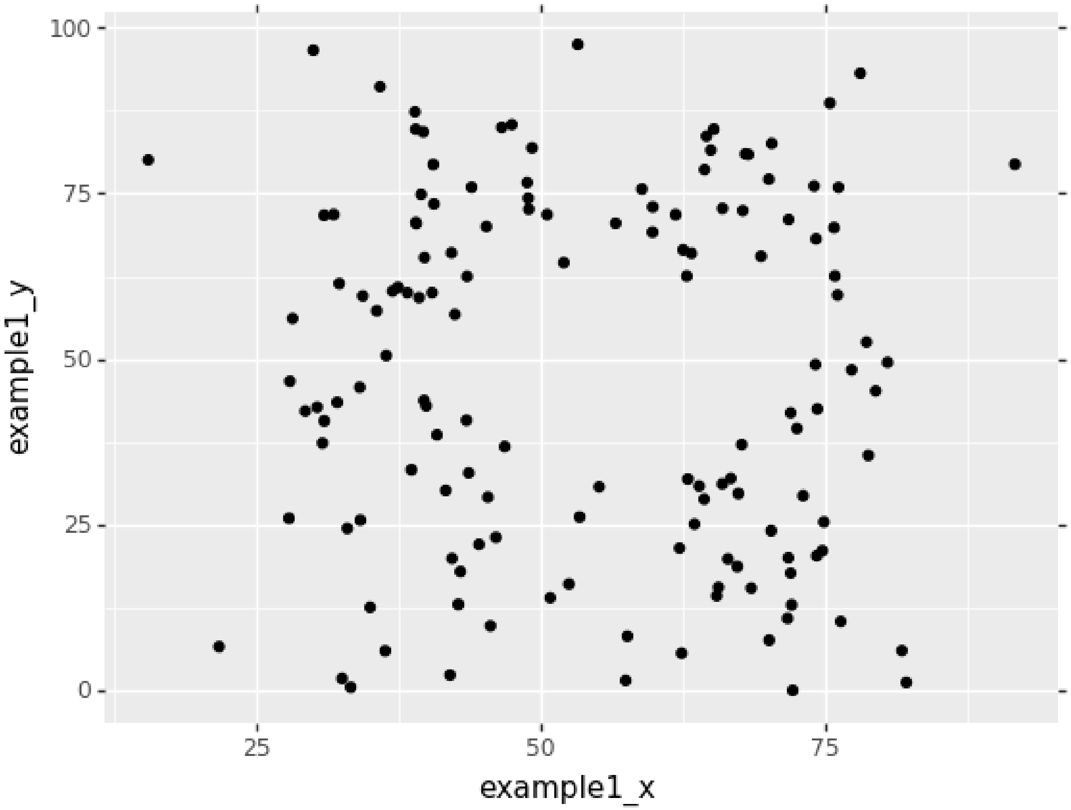
```
#Exercise2
import numpy as np
for i in range(13):
    x = df['example{}_x'.format(i+1)]
    y = df['example{}_y'.format(i+1)]
    c = pd.concat([x,y],axis=1)
    cr = c.corr()
    print('Example Dataset: {},\nMean x: {},\nMean y: {},\nStd Dev x: {},\nStd␣
 ↪Dev y: {},\nCorrelation: {}'.format(i+1,x.mean(),y.mean(),x.std(),y.std(),cr.
 ↪iloc[0,1]))
```

```
Example Dataset: 1,
Mean x: 54.26609978429576,
Mean y: 47.83472062494366,
Std Dev x: 16.769824954043756,
Std Dev y: 26.939743419267103,
Correlation: -0.06412835216739829
Example Dataset: 2,
Mean x: 54.268730022394344,
Mean y: 47.83082315530281,
Std Dev x: 16.7692394934544,
Std Dev y: 26.935726689918788,
Correlation: -0.06858639424107664
Example Dataset: 3,
Mean x: 54.26731970598594,
Mean y: 47.8377172672535,
Std Dev x: 16.760012659806083,
Std Dev y: 26.930036087838204,
Correlation: -0.0683433564802556
Example Dataset: 4,
Mean x: 54.26327323943664,
Mean y: 47.832252816901374,
Std Dev x: 16.765142039116785,
Std Dev y: 26.935403486939123,
Correlation: -0.06447185270095167
Example Dataset: 5,
Mean x: 54.26030345169013,
Mean y: 47.83982920901408,
Std Dev x: 16.767735488473797,
Std Dev y: 26.930191518533462,
Correlation: -0.06034144199921762
Example Dataset: 6,
Mean x: 54.261441783169026,
Mean y: 47.83025191366196,
Std Dev x: 16.76589790389934,
Std Dev y: 26.93987622043797,
Correlation: -0.06171483797263012
Example Dataset: 7,
Mean x: 54.26880527950701,
Mean y: 47.83545020401409,
Std Dev x: 16.766704015934764,
Std Dev y: 26.93999796141102,
Correlation: -0.06850422049412323
Example Dataset: 8,
Mean x: 54.26784882366198,
Mean y: 47.83589633112676,
Std Dev x: 16.76675894771805,
Std Dev y: 26.936104931679974,
Correlation: -0.06897973535951203
Example Dataset: 9,
Mean x: 54.26588178542257,
Mean y: 47.83149565232396,
Std Dev x: 16.768852670828498,
Std Dev y: 26.938608070871844,
Correlation: -0.06860920641825635
Example Dataset: 10,
Mean x: 54.26734110478872,
Mean y: 47.83954522535209,
Std Dev x: 16.768959216194457,
Std Dev y: 26.930274688088435,
Correlation: -0.06296110022065422
Example Dataset: 11,
Mean x: 54.2699272309155,
Mean y: 47.836987988408474,
Std Dev x: 16.76995861132538,
Std Dev y: 26.937683806980512,
Correlation: -0.06944556959350363
Example Dataset: 12,
Mean x: 54.26691630119717,
Mean y: 47.831601987971844,
Std Dev x: 16.76999961757302,
Std Dev y: 26.937901927731804,
Correlation: -0.06657523020460904
Example Dataset: 13,
Mean x: 54.26015033415493,
Mean y: 47.83971727945072,
Std Dev x: 16.76995769550748,
Std Dev y: 26.930001687162342,
Correlation: -0.06558333729297575
```

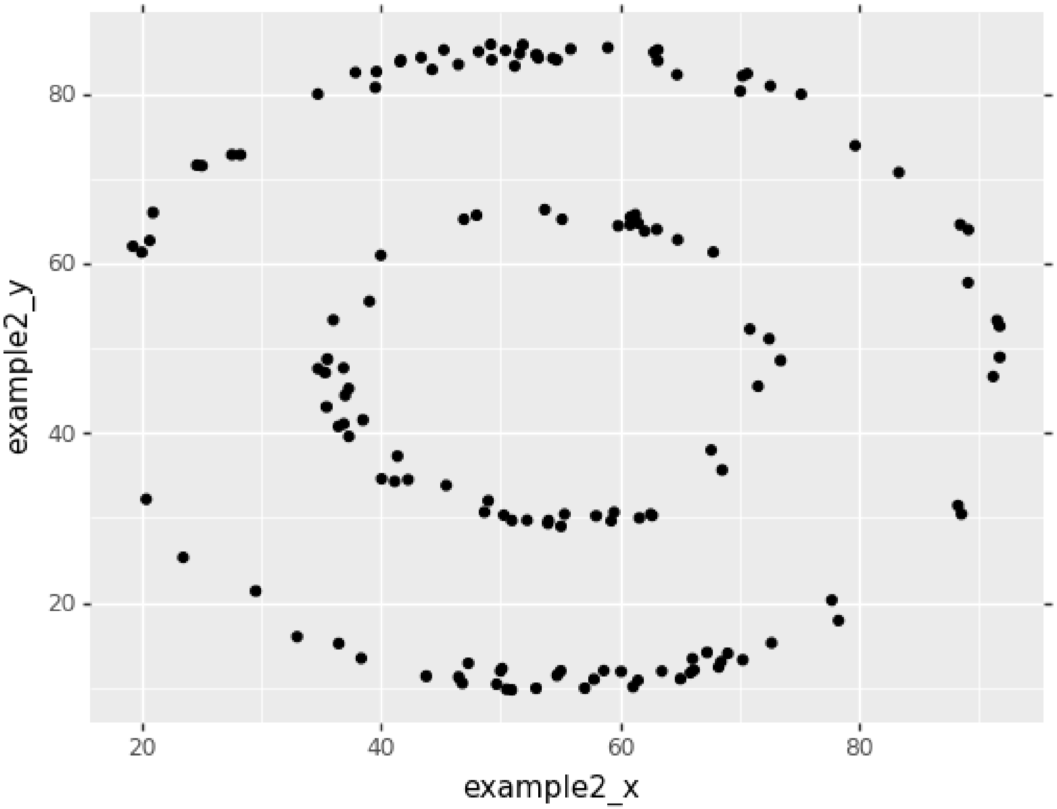
Exerciese3 Different datasets have really similar output, which suggests they have highly identical data distribution.

[53]:

```python
#Exercise4
import warnings
warnings.filterwarnings('ignore', module='plotnine')
from plotnine import *
for i in range(13):
    x = df['example{}_x'.format(i+1)]
    y = df['example{}_y'.format(i+1)]
    p = (ggplot(df,aes(x='example{}_x'.format(i+1), y='example{}_y'.
 ↪format(i+1)))+
        geom_point())
    print(p)
```

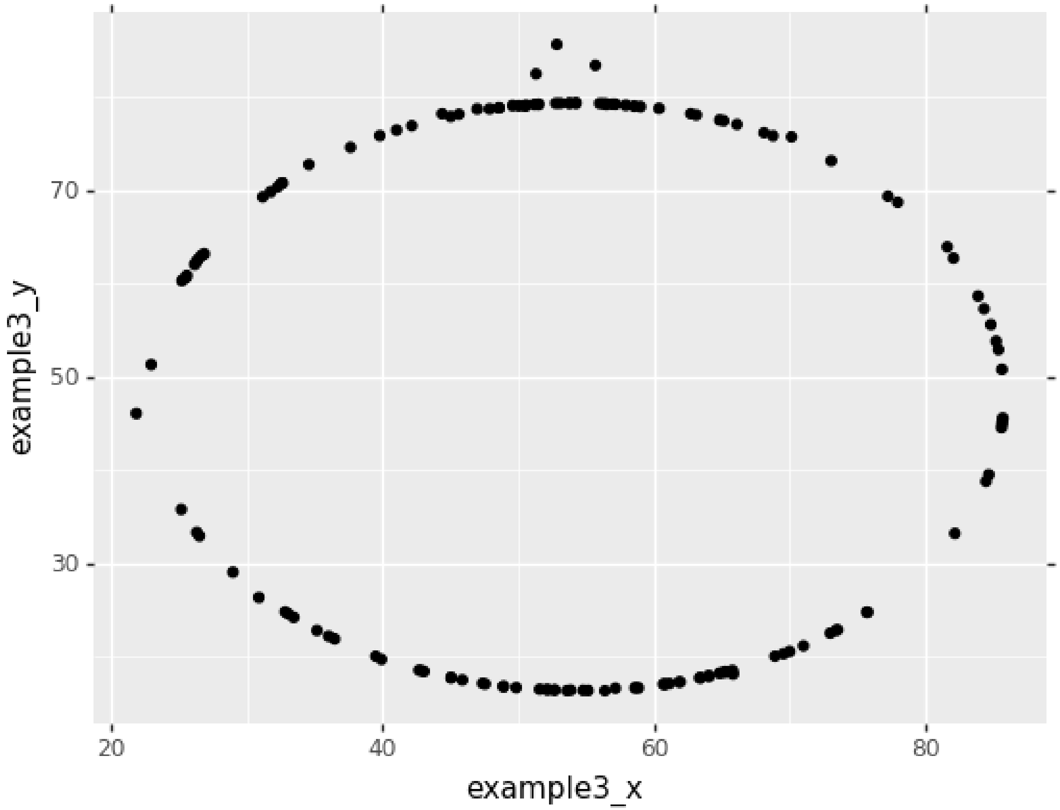


```
<ggplot: (7551622506)>
```

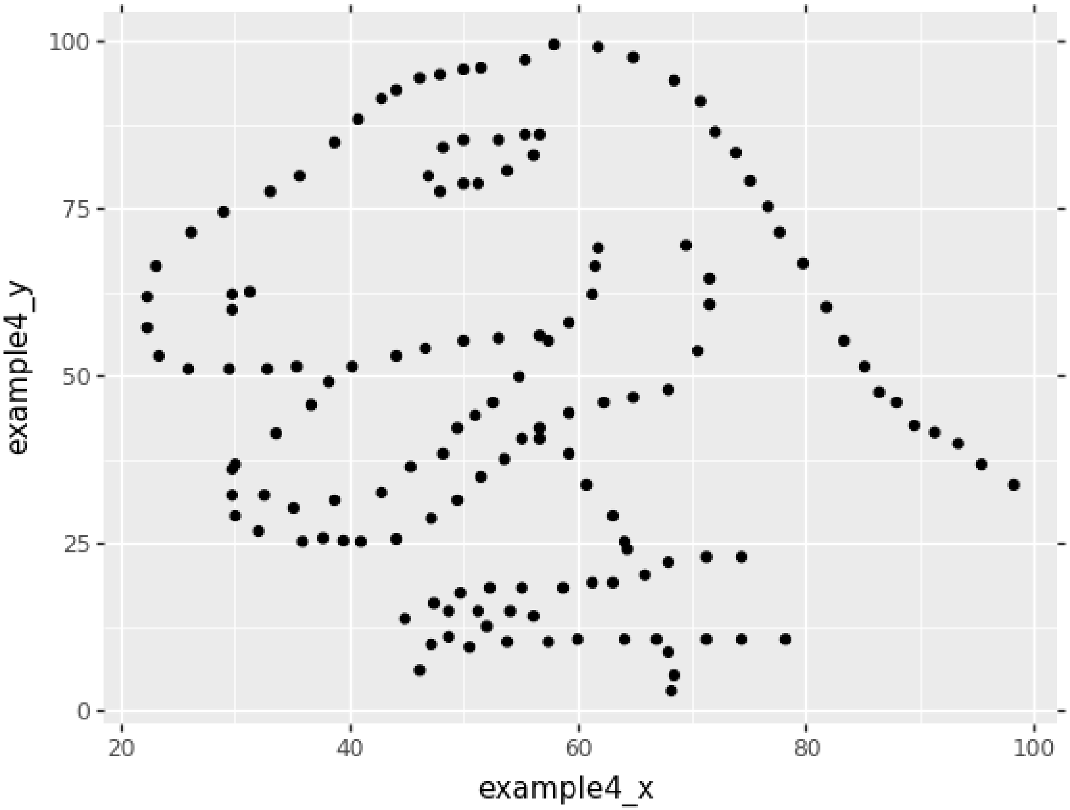


example2_y
80
60
40
20
20
40
60
80
example2_x

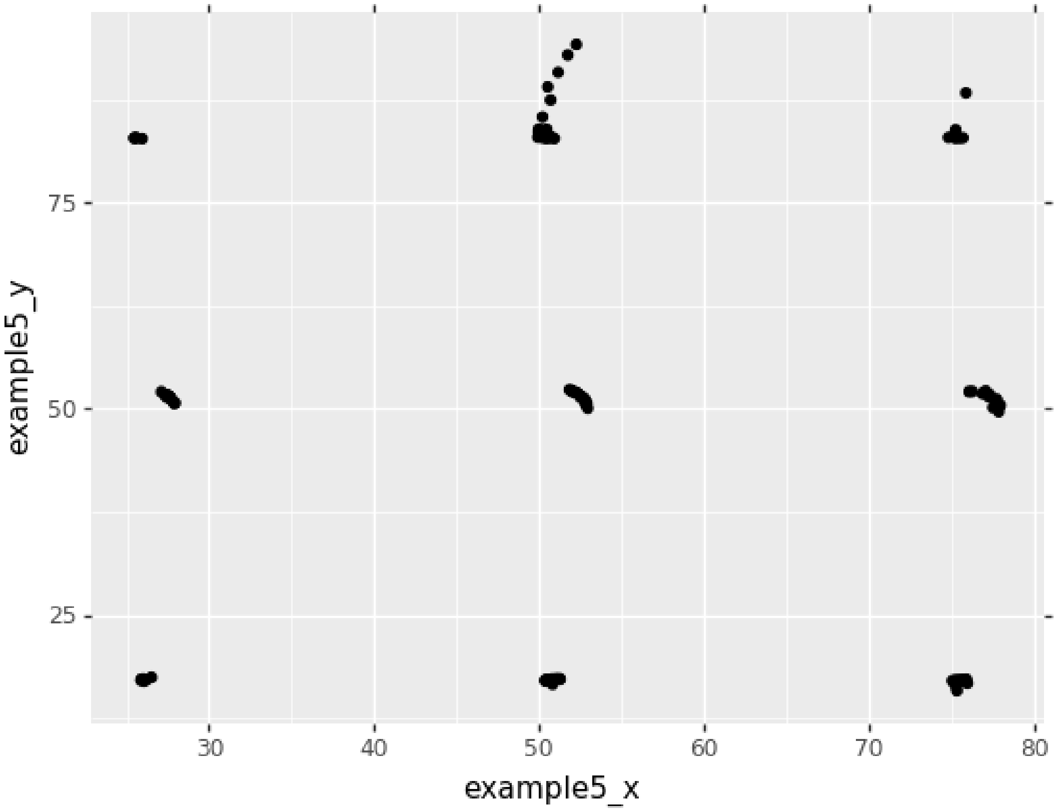
```
<ggplot: (7551071296)>
```


70
50
30
example3_y
20
40
60
80
example3_x

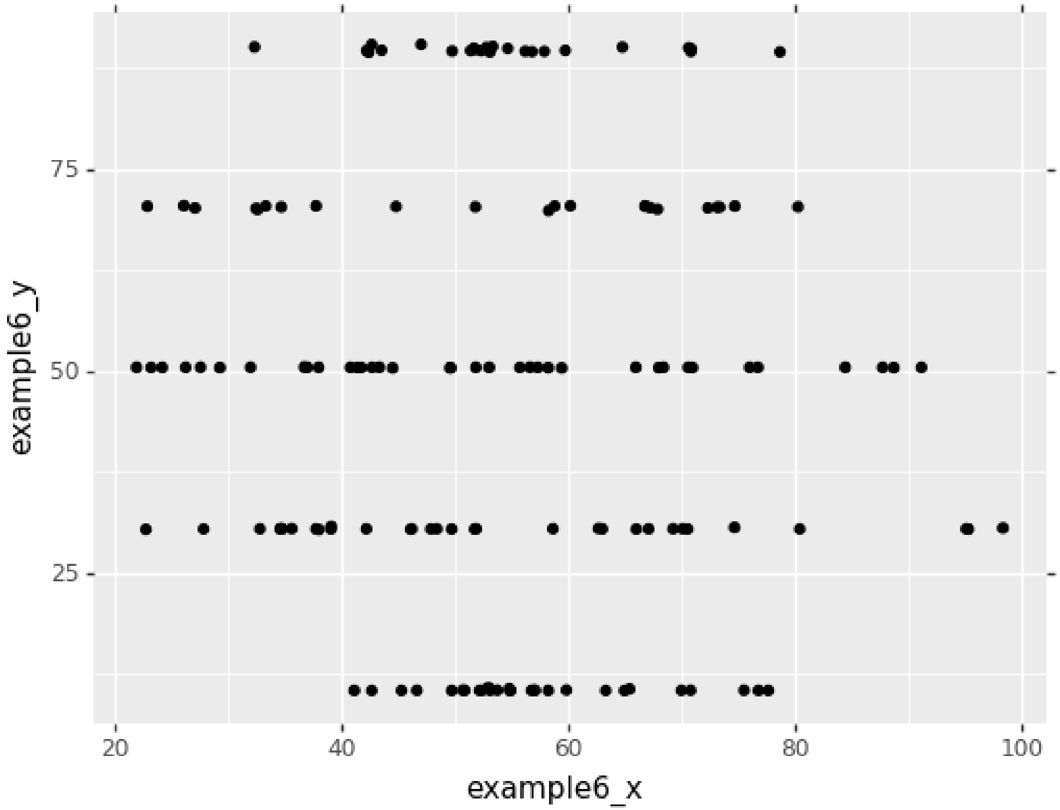
```
<ggplot: (7551144403)>
```


100
75
50
25
0
example4_y
20
40
60
80
100
example4_x

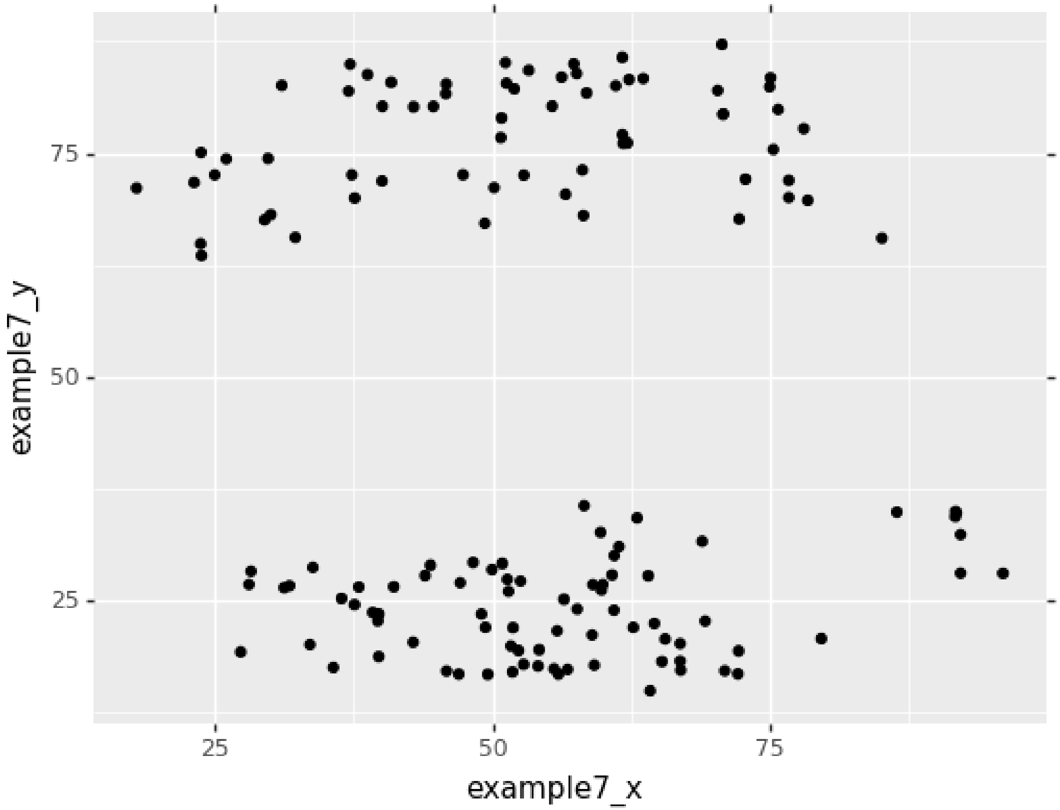
```
<ggplot: (7552030952)>
```


75
50
25
example5_y
30
40
50
60
70
80
example5_x

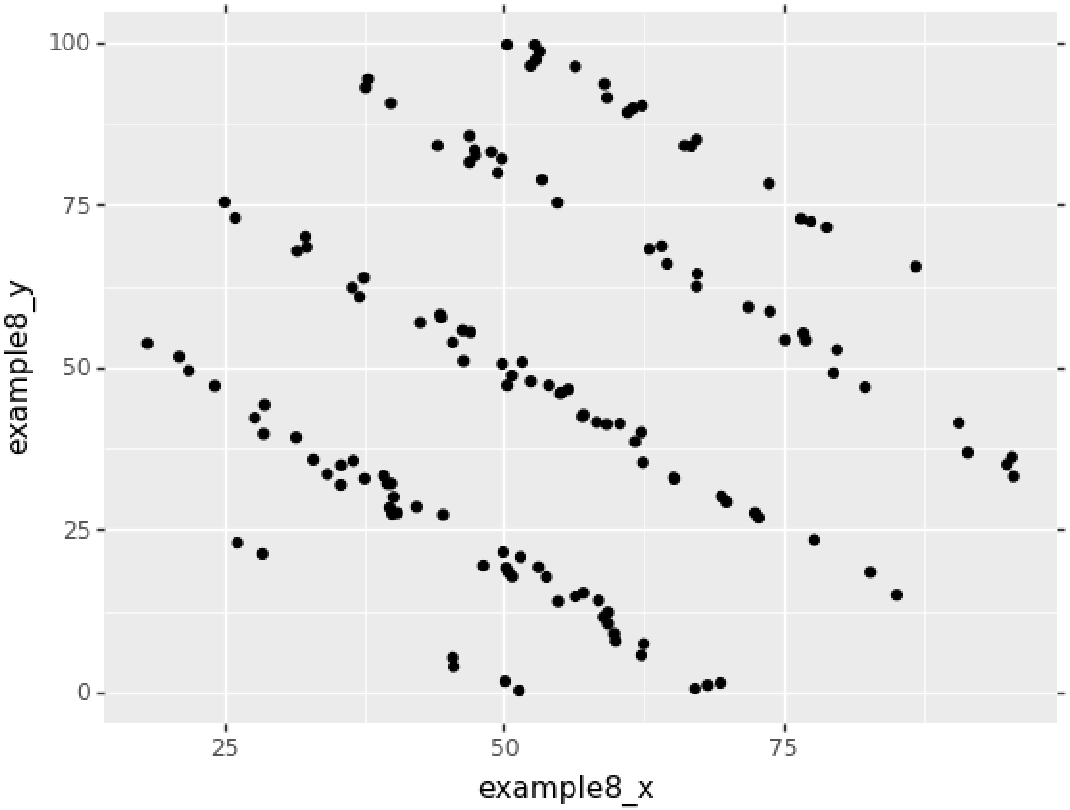
```
<ggplot: (7550969366)>
```


75
50
25
example6_y
20
40
60
80
100
example6_x

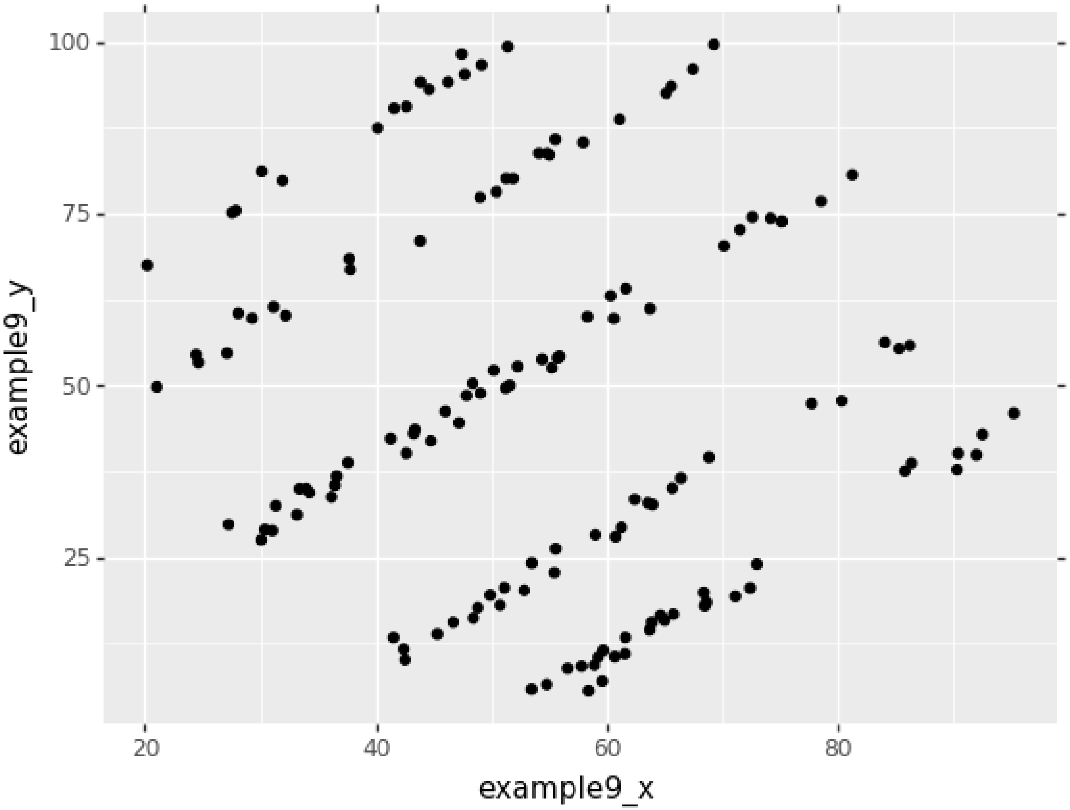
```
<ggplot: (-9223372029304070677)>
```


75
50
25
example7_y
25
50
75
example7_x

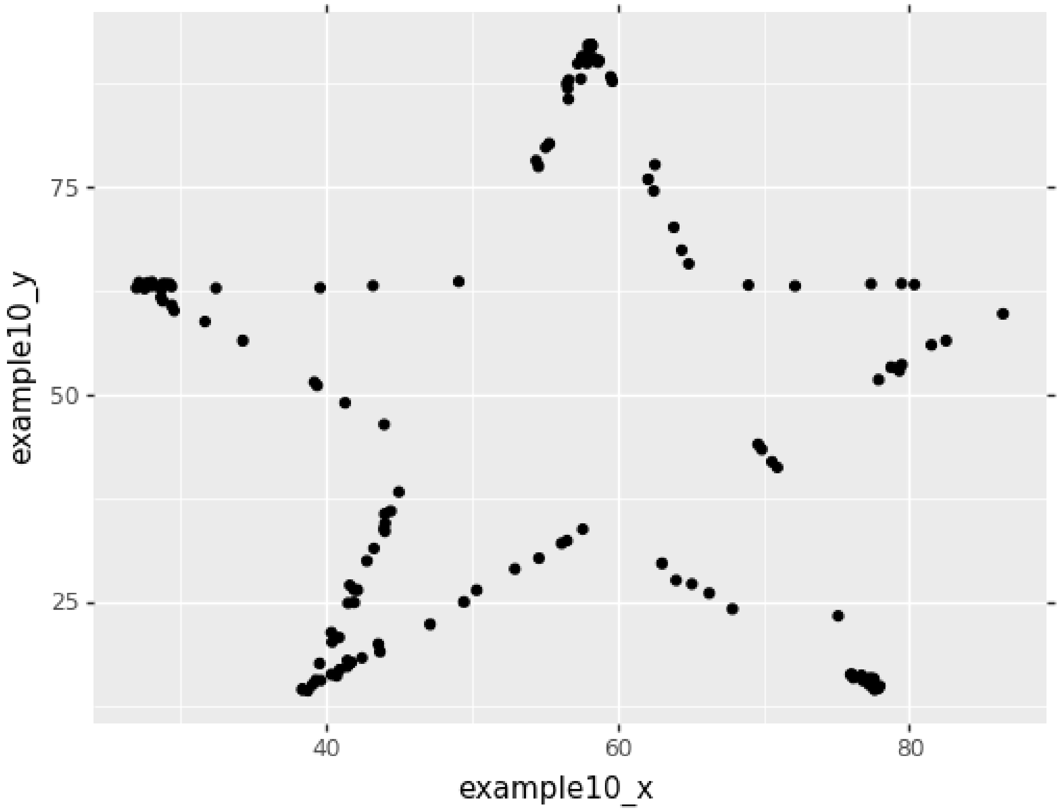
```
<ggplot: (7551821730)>
```


100
75
50
25
0
example8_y
25
50
75
example8_x

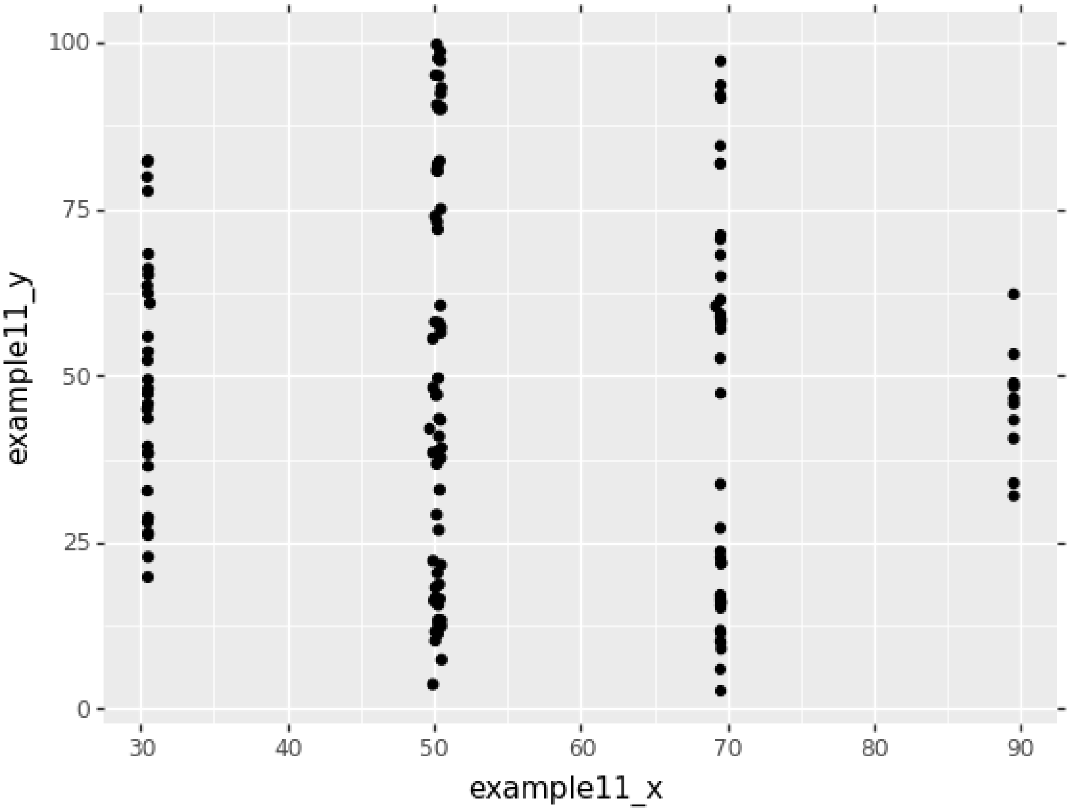
```
<ggplot: (7551047972)>
```


100
75
50
25
example9_y
20
40
60
80
example9_x

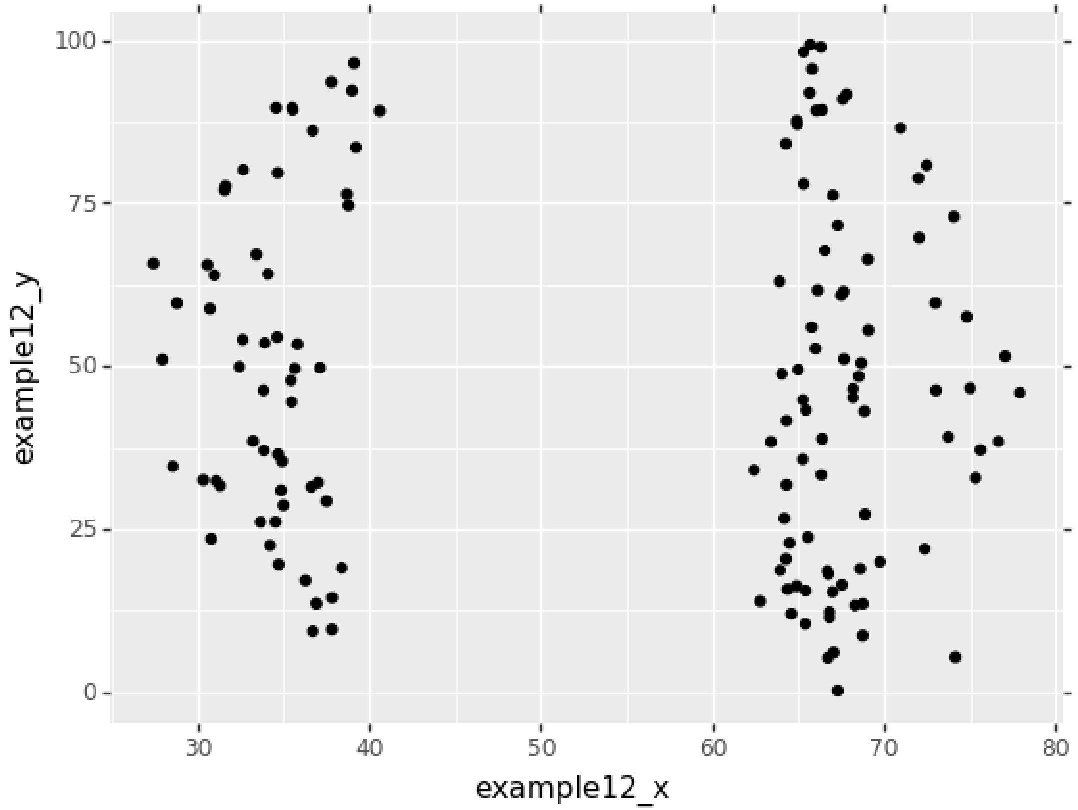
```
<ggplot: (-9223372029304070908)>
```


75
50
25
40
60
80
example10_y
example10_x

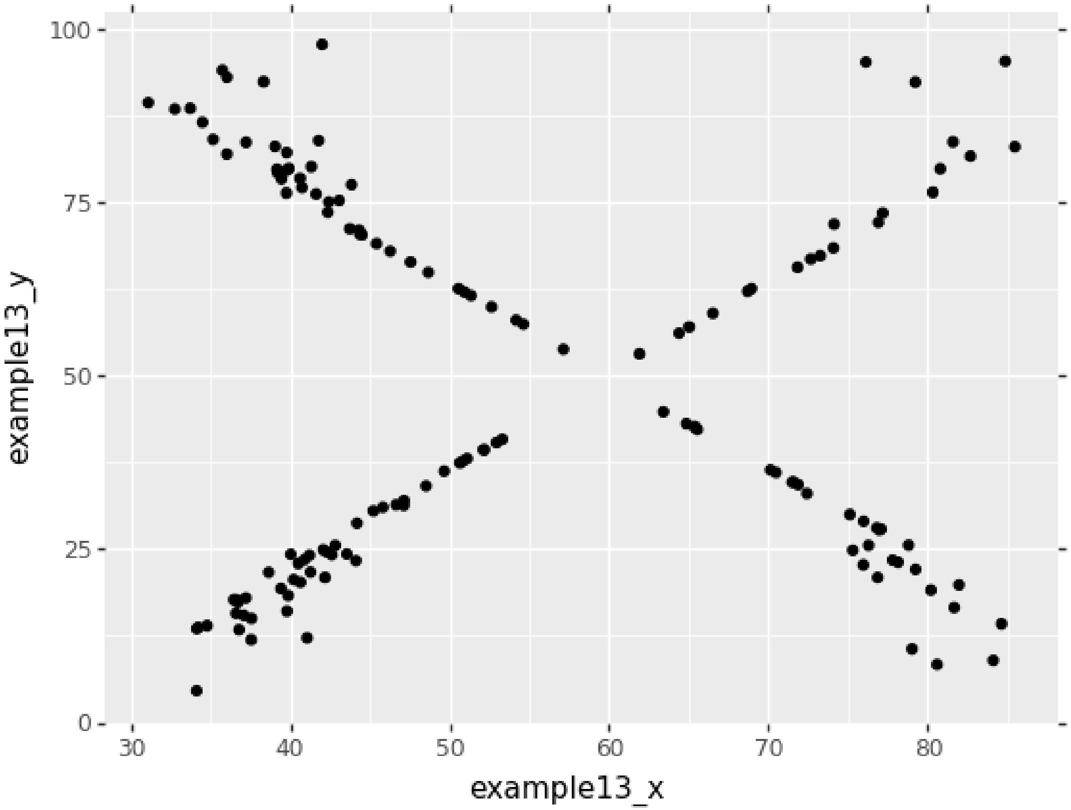
```
<ggplot: (7550958575)>
```


100
75
50
25
0
example11_y
30
40
50
60
70
80
90
example11_x

```
<ggplot: (-9223372029303431010)>
```


100
75
50
25
0
example12_y
30
40
50
60
70
80
example12_x

```
<ggplot: (-9223372029302988242)>
```



```
<ggplot: (7551043034)>
```

#Exercise 5 I thought these datasets must look almost the same on the plot. However, these plots show us that data with different distribution can have highly identical statistics.

[ ]: