CorrelationTests

June 16, 2021

1 Testing correlation between employee counts and # facilities

[63]:
```
import pandas as pd
from pandas import read_csv
import numpy as np
from sklearn.linear_model import LinearRegression
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
import matplotlib.pyplot as plt
import seaborn as sn


pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

[200]:
```
# Group A
groupA=pd.read_csv("CompleteSet_GroupA.csv")
groupA=groupA.drop("Unnamed: 0",axis=1)
#Strip all leading whitespace in Area column
groupA['Area'] = groupA['Area'].apply(lambda x: x.strip())

#Filter only for 2006, 2013 and 2018
groupA = groupA.loc[(groupA['Year'] == 2006) | (groupA['Year'] == 2013)|␣
↪(groupA['Year']==2018)]

#Remove total NZ row
groupA = groupA.loc[(groupA['Area'] != "Total NZ by Regional Council/
↪Statistical Area")]

#Remove total regions
groupA = groupA.loc[(groupA['ParentArea'] != "NewZealand")]

#Only a certain region
#groupA = groupA.loc[(groupA['ParentArea'] == "AucklandRegion")] #Only Auckland
groupA = groupA.loc[(groupA['ParentArea'] == "WaikatoRegion")] #Only Waikato
```

```
#fill in nans caused
groupA=groupA.fillna(0)
```

Useful websites * https://realpython.com/linear-regression-in-python/#simple-linear-regression-with-scikit-learn * https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

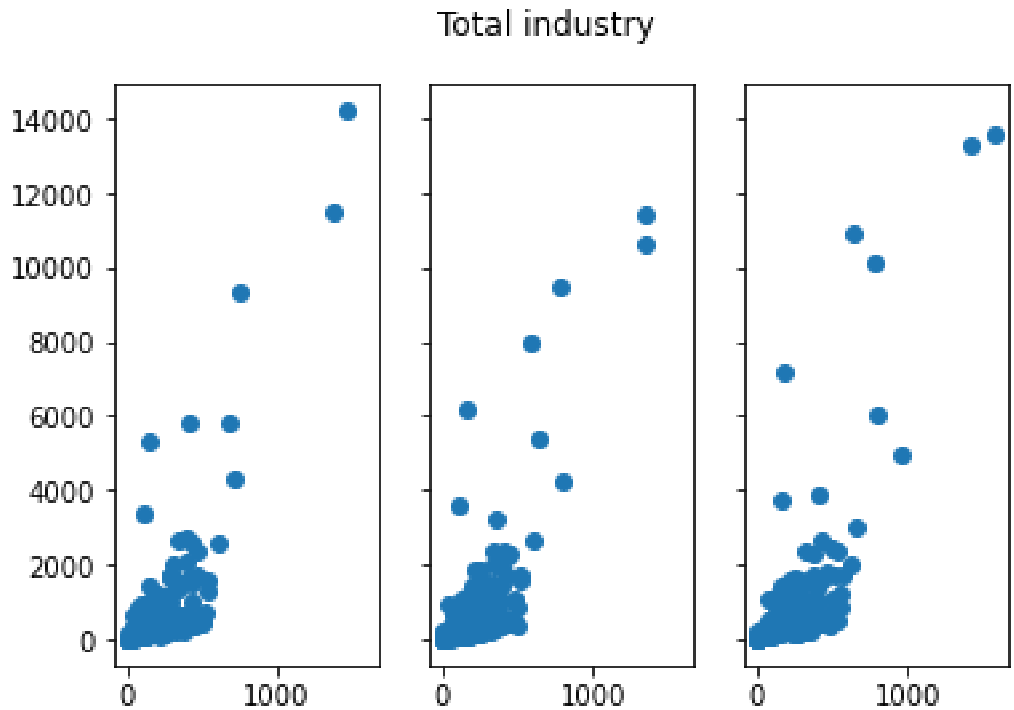
1.1 Total industry

1.1.1 Scatterplot

[201]:
```
totInd_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].
 ↪TotInd_GeogUnits.tolist())
totInd_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].TotInd_EmpCo.
 ↪tolist())

totInd_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].
 ↪TotInd_GeogUnits.tolist())
totInd_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].TotInd_EmpCo.
 ↪tolist())

totInd_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].
 ↪TotInd_GeogUnits.tolist())
totInd_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].TotInd_EmpCo.
 ↪tolist())
```

[202]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Total industry')
ax1.scatter(totInd_Fac_2006, totInd_Emp_2006)
ax2.scatter(totInd_Fac_2013, totInd_Emp_2013)
ax3.scatter(totInd_Fac_2018, totInd_Emp_2018)
```

[202]:
```
<matplotlib.collections.PathCollection at 0x11f7d7bb0>
```



1.1.2 Correlation tests

[203]:
```
# Covariance

print("2006")
covariance = np.cov([totInd_Fac_2006], [totInd_Emp_2006])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(totInd_Fac_2006, totInd_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(totInd_Fac_2006, totInd_Emp_2006)
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([totInd_Fac_2013], [totInd_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(totInd_Fac_2013, totInd_Emp_2013)
```

```
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(totInd_Fac_2013, totInd_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([totInd_Fac_2018], [totInd_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(totInd_Fac_2018, totInd_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(totInd_Fac_2018, totInd_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[  31841.18795528  205865.68528669]
 [ 205865.68528669 2298884.26547061]]
Pearsons correlation: 0.761
Spearmans correlation: 0.699
2013
[[  31034.03871259  189361.88276235]
 [ 189361.88276235 2090650.14460873]]
Pearsons correlation: 0.743
Spearmans correlation: 0.713
2018
[[  37175.63887486  247486.71485756]
 [ 247486.71485756 2936158.30066715]]
Pearsons correlation: 0.749
Spearmans correlation: 0.699
```

1.1.3 Linear regression

regressor - # Facilities ; predictor - employee count

[204]:
```
totInd_Fac_2006 = totInd_Fac_2006.reshape((-1, 1))
totInd_Fac_2013 = totInd_Fac_2013.reshape((-1, 1))
totInd_Fac_2018 = totInd_Fac_2018.reshape((-1, 1))
```

1.1.4 Create model

[205]:
```
model_2006 = LinearRegression().fit(totInd_Fac_2006, totInd_Emp_2006)
model_2013 = LinearRegression().fit(totInd_Fac_2013, totInd_Emp_2013)
model_2018 = LinearRegression().fit(totInd_Fac_2018, totInd_Emp_2018)
```

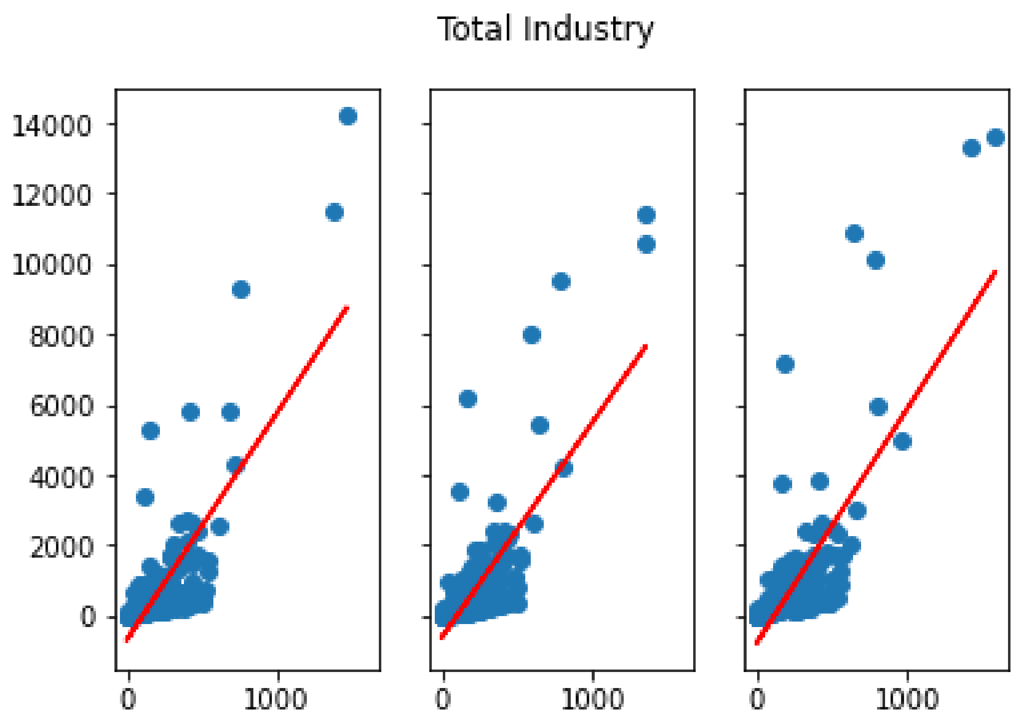
[206]:
```
print("2006")
r_sq = model_2006.score(totInd_Fac_2006, totInd_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(totInd_Fac_2013, totInd_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(totInd_Fac_2018, totInd_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.5789773237110525
intercept: -685.8496636982453
slope: [6.46538959]
2013
coefficient of determination: 0.5526694588315046
intercept: -610.4072182158287
slope: [6.10174797]
2018
coefficient of determination: 0.5611330831336114
intercept: -765.849120622487
slope: [6.65722829]
```

[207]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Total Industry')
ax1.scatter(totInd_Fac_2006, totInd_Emp_2006)
ax1.plot(totInd_Fac_2006,model_2006.coef_*totInd_Fac_2006+model_2006.
 ↪intercept_,'r')
ax2.scatter(totInd_Fac_2013, totInd_Emp_2013)
ax2.plot(totInd_Fac_2013,model_2013.coef_*totInd_Fac_2013+model_2013.
 ↪intercept_,'r')
ax3.scatter(totInd_Fac_2018, totInd_Emp_2018)
```

```
ax3.plot(totInd_Fac_2018,model_2018.coef_*totInd_Fac_2018+model_2018.
    ↪intercept_,'r')
```

[207]: [<matplotlib.lines.Line2D at 0x11f86fe20>]



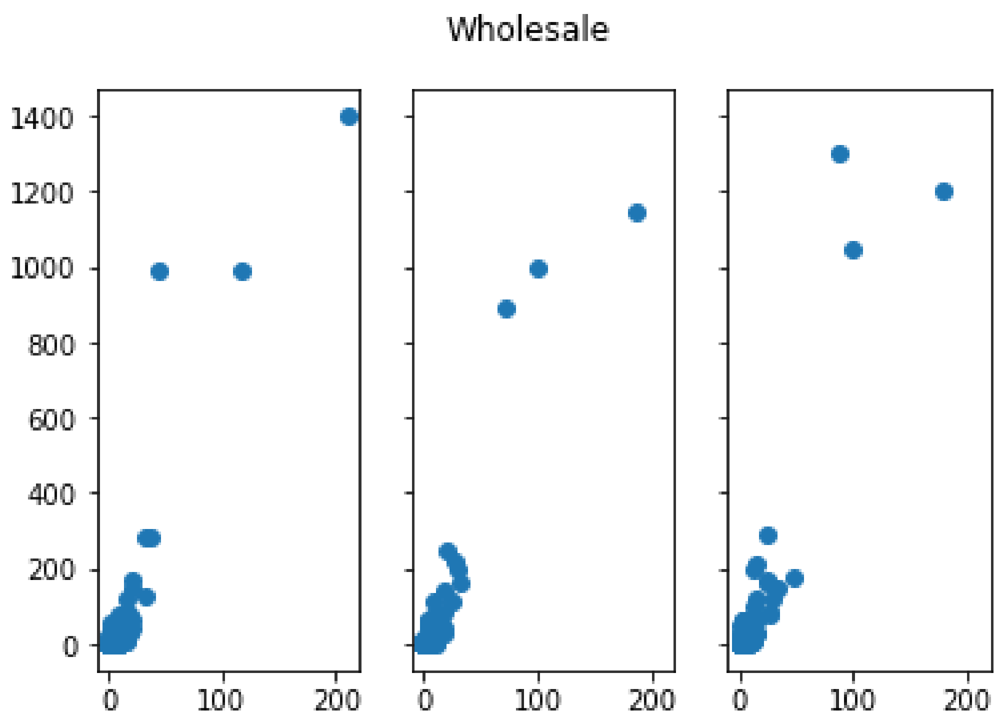
1.2 Wholesale (F)

1.2.1 Scatterplot

[208]:
```
whole_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].F_GeogUnits.
    ↪tolist())
whole_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].F_EmpCo.tolist())

whole_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].F_GeogUnits.
    ↪tolist())
whole_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].F_EmpCo.tolist())

whole_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].F_GeogUnits.
    ↪tolist())
whole_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].F_EmpCo.tolist())
```

[209]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Wholesale')
ax1.scatter(whole_Fac_2006, whole_Emp_2006)
ax2.scatter(whole_Fac_2013, whole_Emp_2013)
ax3.scatter(whole_Fac_2018, whole_Emp_2018)
```

[209]:
```
<matplotlib.collections.PathCollection at 0x11f9ca790>
```



1.2.2 Correlation tests

[210]:
```
# Covariance

print("2006")
covariance = np.cov([whole_Fac_2006], [whole_Emp_2006])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(whole_Fac_2006, whole_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(whole_Fac_2006, whole_Emp_2006)
```

```python
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([whole_Fac_2013], [whole_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(whole_Fac_2013, whole_Emp_2013)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(whole_Fac_2013, whole_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([whole_Fac_2018], [whole_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(whole_Fac_2018, whole_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(whole_Fac_2018, whole_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[  268.02408943  1963.02452218]
 [ 1963.02452218 17256.03844212]]
Pearsons correlation: 0.913
Spearmans correlation: 0.615
2013
[[  220.93896502  1632.59053372]
 [ 1632.59053372 13809.68285251]]
Pearsons correlation: 0.935
Spearmans correlation: 0.650
2018
[[  231.74675442  1889.68101334]
 [ 1889.68101334 18725.43142806]]
Pearsons correlation: 0.907
Spearmans correlation: 0.657
```

1.2.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.2.4 Create model

[211]:
```
whole_Fac_2006 = whole_Fac_2006.reshape((-1, 1))
whole_Fac_2013 = whole_Fac_2013.reshape((-1, 1))
whole_Fac_2018 = whole_Fac_2018.reshape((-1, 1))
```

[212]:
```
model_2006 = LinearRegression().fit(whole_Fac_2006, whole_Emp_2006)
model_2013 = LinearRegression().fit(whole_Fac_2013, whole_Emp_2013)
model_2018 = LinearRegression().fit(whole_Fac_2018, whole_Emp_2018)
```

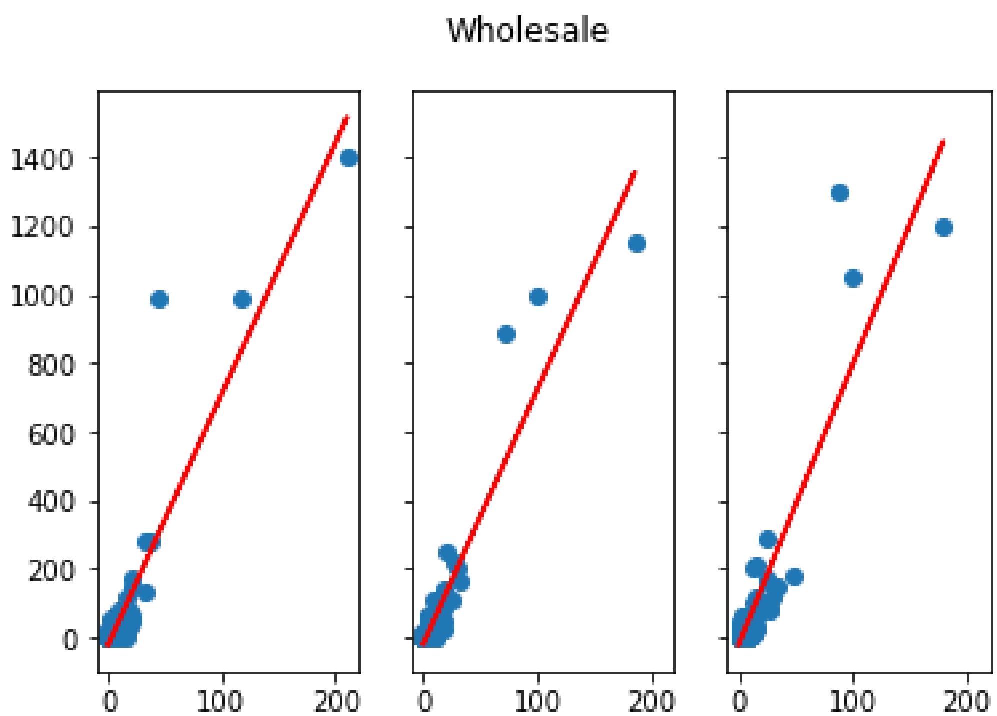
[213]:
```
print("2006")
r_sq = model_2006.score(whole_Fac_2006, whole_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(whole_Fac_2013, whole_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(whole_Fac_2018, whole_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.8331755577222226
intercept: -22.470242621807394
slope: [7.32406004]
2013
coefficient of determination: 0.8735716428349292
intercept: -18.63468150351642
slope: [7.38932824]
2018
coefficient of determination: 0.8228705242147732
intercept: -22.521397685695792
slope: [8.15407758]
```

[214]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Wholesale')
ax1.scatter(whole_Fac_2006, whole_Emp_2006)
```

```
ax1.plot(whole_Fac_2006,model_2006.coef_*whole_Fac_2006+model_2006.
 ↪intercept_,'r')
ax2.scatter(whole_Fac_2013, whole_Emp_2013)
ax2.plot(whole_Fac_2013,model_2013.coef_*whole_Fac_2013+model_2013.
 ↪intercept_,'r')
ax3.scatter(whole_Fac_2018, whole_Emp_2018)
ax3.plot(whole_Fac_2018,model_2018.coef_*whole_Fac_2018+model_2018.
 ↪intercept_,'r')
```

[214]: [<matplotlib.lines.Line2D at 0x11f321790>]



1.3 Retail (G)

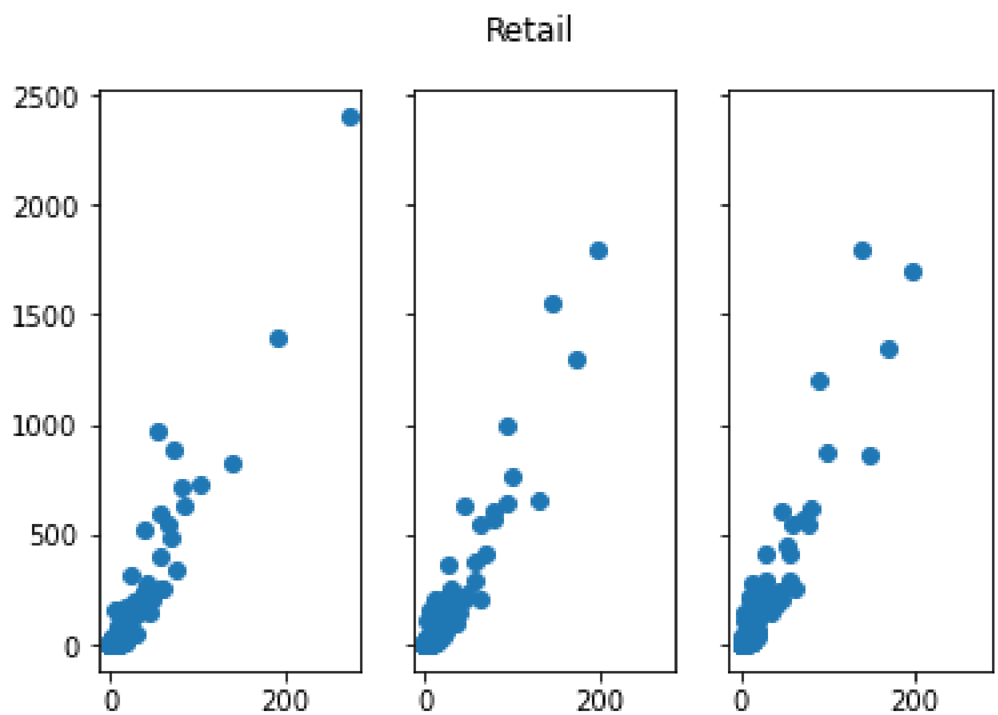
1.3.1 Scatterplot

[215]:
```
retail_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].G_GeogUnits.
 ↪tolist())
retail_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].G_EmpCo.
 ↪tolist())

retail_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].G_GeogUnits.
 ↪tolist())
```

```
retail_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].G_EmpCo.
 ↪tolist())

retail_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].G_GeogUnits.
 ↪tolist())
retail_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].G_EmpCo.
 ↪tolist())
```

[216]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Retail')
ax1.scatter(retail_Fac_2006, retail_Emp_2006)
ax2.scatter(retail_Fac_2013, retail_Emp_2013)
ax3.scatter(retail_Fac_2018, retail_Emp_2018)
```

[216]: <matplotlib.collections.PathCollection at 0x11fb76460>



1.3.2 Correlation tests

[217]:
```
# Covariance

print("2006")
covariance = np.cov([retail_Fac_2006], [retail_Emp_2006])
```

```
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(retail_Fac_2006, retail_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(retail_Fac_2006, retail_Emp_2006)
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([retail_Fac_2013], [retail_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(retail_Fac_2013, retail_Emp_2013)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(retail_Fac_2013, retail_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([retail_Fac_2018], [retail_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(retail_Fac_2018, retail_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(retail_Fac_2018, retail_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[  772.3841868   6125.39334656]
 [ 6125.39334656 53487.70968265]]
Pearsons correlation: 0.953
Spearmans correlation: 0.799
2013
[[  656.53377209  5315.5134872 ]
 [ 5315.5134872  47357.20800577]]
Pearsons correlation: 0.953
Spearmans correlation: 0.792
2018
[[  635.89965741  5497.13487198]
 [ 5497.13487198 53569.65510278]]
```

```
Pearsons correlation: 0.942
Spearmans correlation: 0.827
```

1.3.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.3.4 Create model

[218]:
```
retail_Fac_2006 = retail_Fac_2006.reshape((-1, 1))
retail_Fac_2013 = retail_Fac_2013.reshape((-1, 1))
retail_Fac_2018 = retail_Fac_2018.reshape((-1, 1))
```

[219]:
```
model_2006 = LinearRegression().fit(retail_Fac_2006, retail_Emp_2006)
model_2013 = LinearRegression().fit(retail_Fac_2013, retail_Emp_2013)
model_2018 = LinearRegression().fit(retail_Fac_2018, retail_Emp_2018)
```

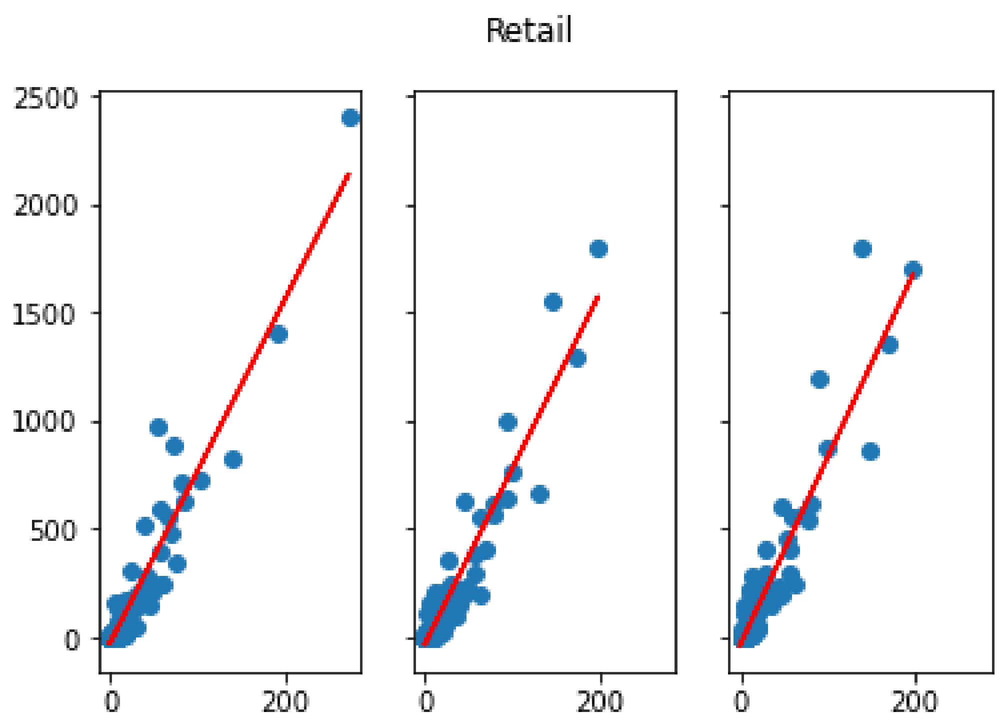
[220]:
```
print("2006")
r_sq = model_2006.score(retail_Fac_2006, retail_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(retail_Fac_2013, retail_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(retail_Fac_2018, retail_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.9081981002675357
intercept: -28.105392823729233
slope: [7.93050072]
2013
coefficient of determination: 0.9087559922825942
intercept: -31.096373513949487
slope: [8.0963291]
2018
```

```
coefficient of determination: 0.8870852206556968
intercept: -35.94444508243468
slope: [8.6446577]
```

[221]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Retail')
ax1.scatter(retail_Fac_2006, retail_Emp_2006)
ax1.plot(retail_Fac_2006,model_2006.coef_*retail_Fac_2006+model_2006.
 ↪intercept_,'r')
ax2.scatter(retail_Fac_2013, retail_Emp_2013)
ax2.plot(retail_Fac_2013,model_2013.coef_*retail_Fac_2013+model_2013.
 ↪intercept_,'r')
ax3.scatter(retail_Fac_2018, retail_Emp_2018)
ax3.plot(retail_Fac_2018,model_2018.coef_*retail_Fac_2018+model_2018.
 ↪intercept_,'r')
```

[221]:
```
[<matplotlib.lines.Line2D at 0x1201e6cd0>]
```



1.4 TransPostWare

[222]:
```
groupA['TransPostWare_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I4
groupA['TransPostWare_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']+g
```

1.4.1 Scatterplot

[223]:
```
tpw_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].
 ↪TransPostWare_GeogUnits.tolist())
tpw_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].
 ↪TransPostWare_EmpCo.tolist())

tpw_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].
 ↪TransPostWare_GeogUnits.tolist())
tpw_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].
 ↪TransPostWare_EmpCo.tolist())

tpw_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].
 ↪TransPostWare_GeogUnits.tolist())
tpw_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].
 ↪TransPostWare_EmpCo.tolist())
```

[224]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Transport Post Warehouse')
ax1.scatter(tpw_Fac_2006, tpw_Emp_2006)
ax2.scatter(tpw_Fac_2013, tpw_Emp_2013)
ax3.scatter(tpw_Fac_2018, tpw_Emp_2018)
```

[224]:
```
<matplotlib.collections.PathCollection at 0x12034dd30>
```

1.4.2 Correlation tests

[225]:
```python
# Covariance

print("2006")
covariance = np.cov([tpw_Fac_2006], [tpw_Emp_2006])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(tpw_Fac_2006, tpw_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(tpw_Fac_2006, tpw_Emp_2006)
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([tpw_Fac_2013], [tpw_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(tpw_Fac_2013, tpw_Emp_2013)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(tpw_Fac_2013, tpw_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([tpw_Fac_2018], [tpw_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(tpw_Fac_2018, tpw_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(tpw_Fac_2018, tpw_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[  18.87500902  124.89931482]
 [ 124.89931482 3664.7924991 ]]
Pearsons correlation: 0.475
Spearmans correlation: 0.625
```

```
2013
[[  17.41254959  100.73909124]
 [ 100.73909124 1694.22652362]]
Pearsons correlation: 0.587
Spearmans correlation: 0.618
2018
[[  15.76444284  108.73869455]
 [ 108.73869455 2541.60714028]]
Pearsons correlation: 0.543
Spearmans correlation: 0.660
```

1.4.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.4.4 Create model

[226]:
```
tpw_Fac_2006 = tpw_Fac_2006.reshape((-1, 1))
tpw_Fac_2013 = tpw_Fac_2013.reshape((-1, 1))
tpw_Fac_2018 = tpw_Fac_2018.reshape((-1, 1))
```

[227]:
```
model_2006 = LinearRegression().fit(tpw_Fac_2006, tpw_Emp_2006)
model_2013 = LinearRegression().fit(tpw_Fac_2013, tpw_Emp_2013)
model_2018 = LinearRegression().fit(tpw_Fac_2018, tpw_Emp_2018)
```

[228]:
```
print("2006")
r_sq = model_2006.score(tpw_Fac_2006, tpw_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(tpw_Fac_2013, tpw_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(tpw_Fac_2018, tpw_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.22551921658323792
intercept: -8.47599559804663
slope: [6.61717908]
2013
coefficient of determination: 0.34400298806816354
intercept: -7.439515377446401
slope: [5.78543026]
2018
coefficient of determination: 0.2951081350482555
intercept: -10.080087293090664
slope: [6.89771885]
```

[229]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Transport Post Warehouse')
ax1.scatter(tpw_Fac_2006, tpw_Emp_2006)
ax1.plot(tpw_Fac_2006,model_2006.coef_*tpw_Fac_2006+model_2006.intercept_,'r')
ax2.scatter(tpw_Fac_2013, tpw_Emp_2013)
ax2.plot(tpw_Fac_2013,model_2013.coef_*tpw_Fac_2013+model_2013.intercept_,'r')
ax3.scatter(tpw_Fac_2018, tpw_Emp_2018)
ax3.plot(tpw_Fac_2018,model_2018.coef_*tpw_Fac_2018+model_2018.intercept_,'r')
```

[229]:
```
[<matplotlib.lines.Line2D at 0x1204d5280>]
```

1.5 Transport

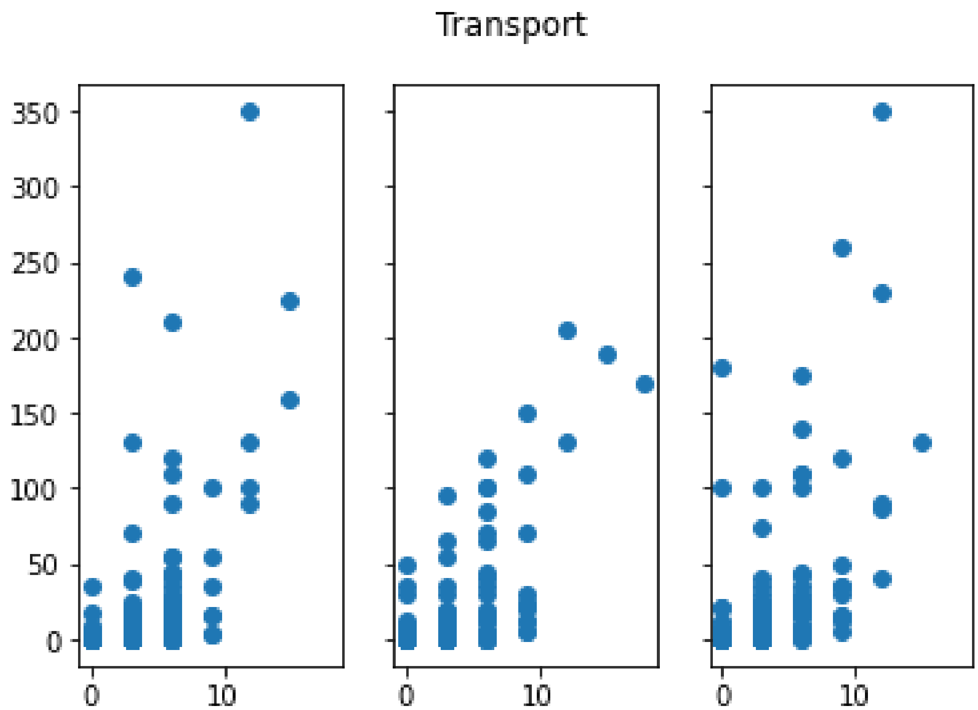
[230]:
```
groupA['Transport_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I481_G
groupA['Transport_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']
```

1.5.1 Scatterplot

[231]:
```
trans_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].
 ↪Transport_GeogUnits.tolist())
trans_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].Transport_EmpCo.
 ↪tolist())

trans_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].
 ↪Transport_GeogUnits.tolist())
trans_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].Transport_EmpCo.
 ↪tolist())

trans_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].
 ↪Transport_GeogUnits.tolist())
trans_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].Transport_EmpCo.
 ↪tolist())
```

[232]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Transport')
ax1.scatter(trans_Fac_2006, trans_Emp_2006)
ax2.scatter(trans_Fac_2013, trans_Emp_2013)
ax3.scatter(trans_Fac_2018, trans_Emp_2018)
```

[232]:
```
<matplotlib.collections.PathCollection at 0x120636f70>
```



1.5.2 Correlation tests

[233]:
```
# Covariance

print("2006")
covariance = np.cov([trans_Fac_2006], [trans_Emp_2006])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(trans_Fac_2006, trans_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(trans_Fac_2006, trans_Emp_2006)
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([trans_Fac_2013], [trans_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(trans_Fac_2013, trans_Emp_2013)
```

```python
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(trans_Fac_2013, trans_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([trans_Fac_2018], [trans_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(trans_Fac_2018, trans_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(trans_Fac_2018, trans_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[   9.34208439   65.6743599 ]
 [  65.6743599  1638.28589975]]
Pearsons correlation: 0.531
Spearmans correlation: 0.671
2013
[[  9.05842048  59.7077894 ]
 [ 59.7077894  928.67614497]]
Pearsons correlation: 0.651
Spearmans correlation: 0.626
2018
[[   9.16990624   66.28339344]
 [  66.28339344 1725.67578435]]
Pearsons correlation: 0.527
Spearmans correlation: 0.703
```

1.5.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.5.4 Create model

[234]:
```python
trans_Fac_2006 = trans_Fac_2006.reshape((-1, 1))
trans_Fac_2013 = trans_Fac_2013.reshape((-1, 1))
trans_Fac_2018 = trans_Fac_2018.reshape((-1, 1))
```

[235]:
```
model_2006 = LinearRegression().fit(trans_Fac_2006, trans_Emp_2006)
model_2013 = LinearRegression().fit(trans_Fac_2013, trans_Emp_2013)
model_2018 = LinearRegression().fit(trans_Fac_2018, trans_Emp_2018)
```

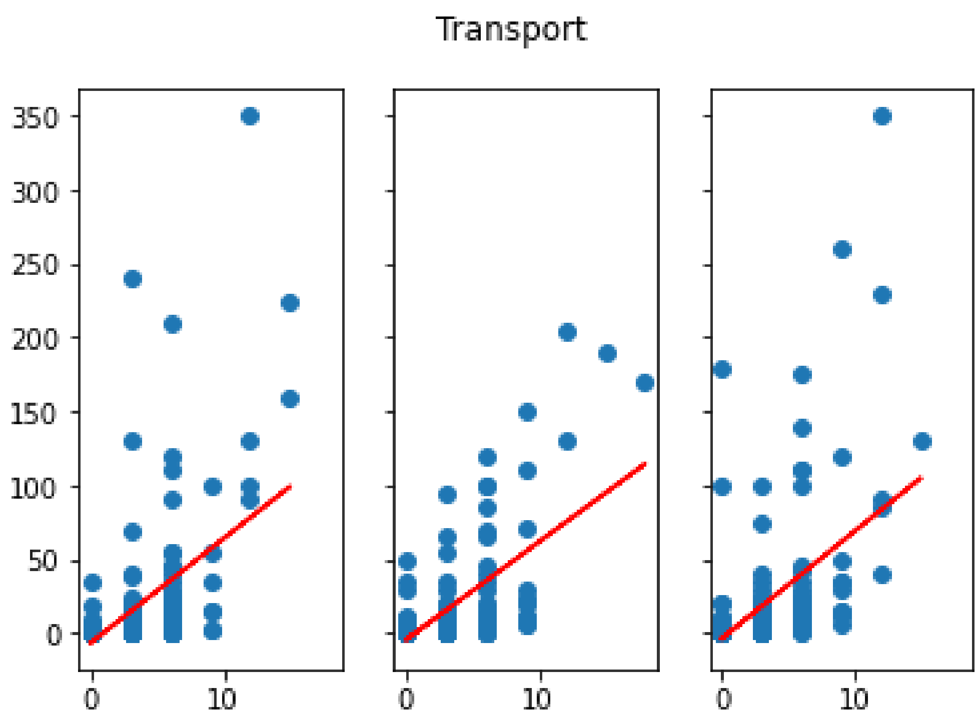
[236]:
```
print("2006")
r_sq = model_2006.score(trans_Fac_2006, trans_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(trans_Fac_2013, trans_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(trans_Fac_2018, trans_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.28181117965296165
intercept: -6.359505280711513
slope: [7.02994719]
2013
coefficient of determination: 0.4237846464444991
intercept: -4.276280902902183
slope: [6.59141287]
2018
coefficient of determination: 0.277642130836118
intercept: -3.6686959137805797
slope: [7.22836109]
```

[237]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Transport')
ax1.scatter(trans_Fac_2006, trans_Emp_2006)
ax1.plot(trans_Fac_2006,model_2006.coef_*trans_Fac_2006+model_2006.
 ↪intercept_,'r')
ax2.scatter(trans_Fac_2013, trans_Emp_2013)
ax2.plot(trans_Fac_2013,model_2013.coef_*trans_Fac_2013+model_2013.
 ↪intercept_,'r')
ax3.scatter(trans_Fac_2018, trans_Emp_2018)
```

```
ax3.plot(trans_Fac_2018,model_2018.coef_*trans_Fac_2018+model_2018.
↪intercept_,'r')
```

[237]: [<matplotlib.lines.Line2D at 0x1207ce4c0>]



1.6 Post and storage

[238]:
```
groupA['Storage_GeogUnits']=groupA['I51_GeogUnits']+groupA['I53_GeogUnits']
groupA['Storage_EmpCo']=groupA['I51_EmpCo']+groupA['I53_EmpCo']
```

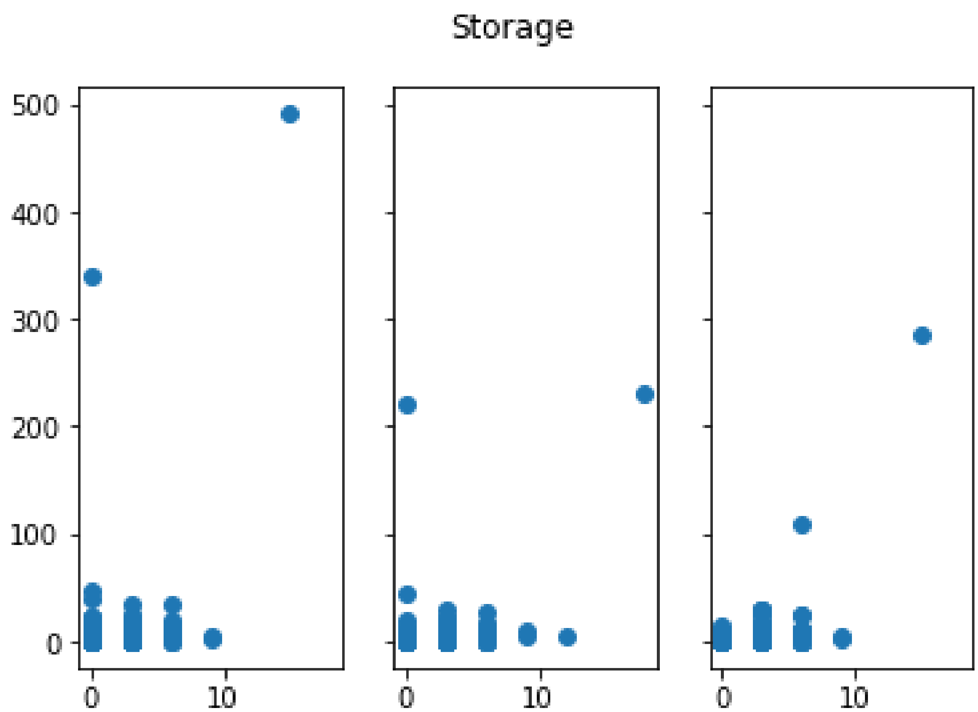
1.6.1 Scatterplot

[239]:
```
stor_Fac_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].Storage_GeogUnits.
↪tolist())
stor_Emp_2006 = np.array(groupA.loc[(groupA['Year'] == 2006)].Storage_EmpCo.
↪tolist())

stor_Fac_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].Storage_GeogUnits.
↪tolist())
stor_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].Storage_EmpCo.
↪tolist())
```

```
stor_Fac_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].Storage_GeogUnits.
 ↪tolist())
stor_Emp_2018 = np.array(groupA.loc[(groupA['Year'] == 2018)].Storage_EmpCo.
 ↪tolist())
```

[240]:
```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Storage')
ax1.scatter(stor_Fac_2006, stor_Emp_2006)
ax2.scatter(stor_Fac_2013, stor_Emp_2013)
ax3.scatter(stor_Fac_2018, stor_Emp_2018)
```

[240]:
```
<matplotlib.collections.PathCollection at 0x1209521f0>
```



1.6.2 Correlation tests

[241]:
```
# Covariance

print("2006")
covariance = np.cov([stor_Fac_2006], [stor_Emp_2006])
print(covariance)
```

```
# Pearson's correlation
corrP, _ = pearsonr(stor_Fac_2006, stor_Emp_2006)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(stor_Fac_2006, stor_Emp_2006)
print('Spearmans correlation: %.3f' % corrS)

print("2013")
covariance = np.cov([stor_Fac_2013], [stor_Emp_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(stor_Fac_2013, stor_Emp_2013)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(stor_Fac_2013, stor_Emp_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([stor_Fac_2018], [stor_Emp_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(stor_Fac_2018, stor_Emp_2018)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(stor_Fac_2018, stor_Emp_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2006
[[   5.01637216   29.39805265]
 [  29.39805265 1532.43245582]]
Pearsons correlation: 0.335
Spearmans correlation: 0.522
2013
[[  5.46036783  17.96646232]
 [ 17.96646232 453.76047602]]
Pearsons correlation: 0.361
Spearmans correlation: 0.474
2018
[[  5.30376848  22.34264335]
 [ 22.34264335 416.96507393]]
Pearsons correlation: 0.475
Spearmans correlation: 0.492
```

1.6.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.6.4 Create model

[242]:
```
stor_Fac_2006 = stor_Fac_2006.reshape((-1, 1))
stor_Fac_2013 = stor_Fac_2013.reshape((-1, 1))
stor_Fac_2018 = stor_Fac_2018.reshape((-1, 1))
```

[243]:
```
model_2006 = LinearRegression().fit(stor_Fac_2006, stor_Emp_2006)
model_2013 = LinearRegression().fit(stor_Fac_2013, stor_Emp_2013)
model_2018 = LinearRegression().fit(stor_Fac_2018, stor_Emp_2018)
```

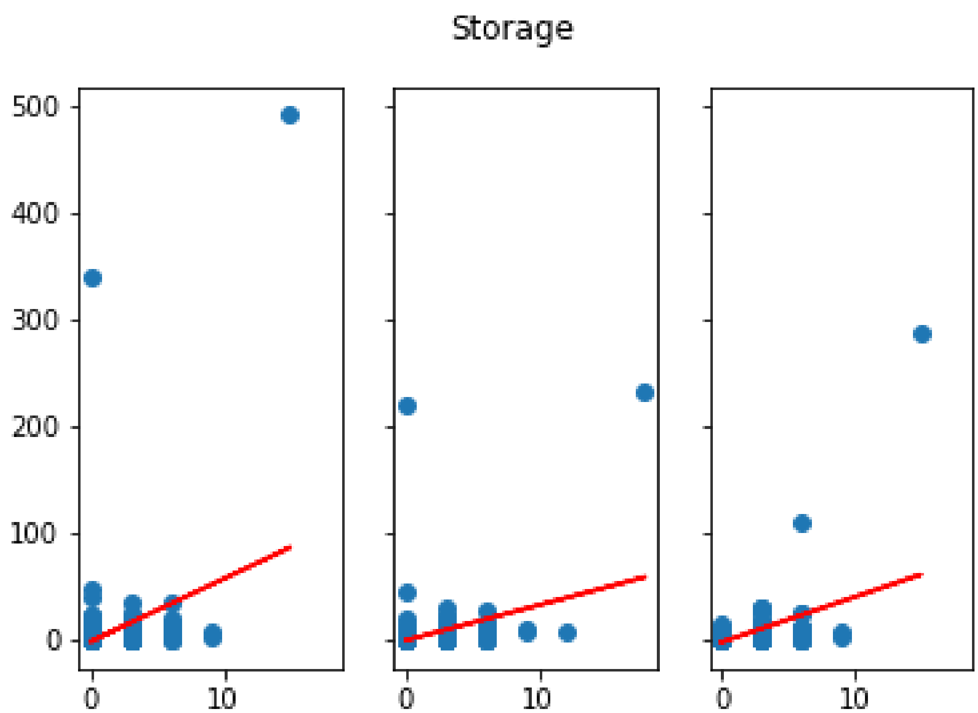
[244]:
```
print("2006")
r_sq = model_2006.score(stor_Fac_2006, stor_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


print("2013")
r_sq = model_2013.score(stor_Fac_2013, stor_Emp_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(stor_Fac_2018, stor_Emp_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2006
coefficient of determination: 0.11242581287632247
intercept: -2.2179089026915104
slope: [5.86042098]
2013
coefficient of determination: 0.1302796482478844
intercept: -1.091060389919165
slope: [3.2903392]
2018
coefficient of determination: 0.22572770648585505
intercept: -2.8782853471223584
slope: [4.21259778]
```

[245]:

```
fig, (ax1, ax2,ax3) = plt.subplots(1, 3,sharex=True,sharey=True)
fig.suptitle('Storage')
ax1.scatter(stor_Fac_2006, stor_Emp_2006)
ax1.plot(stor_Fac_2006,model_2006.coef_*stor_Fac_2006+model_2006.intercept_,'r')
ax2.scatter(stor_Fac_2013, stor_Emp_2013)
ax2.plot(stor_Fac_2013,model_2013.coef_*stor_Fac_2013+model_2013.intercept_,'r')
ax3.scatter(stor_Fac_2018, stor_Emp_2018)
ax3.plot(stor_Fac_2018,model_2018.coef_*stor_Fac_2018+model_2018.intercept_,'r')
```

[245]: [<matplotlib.lines.Line2D at 0x11f832b20>]



[ ]: