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)
```

[338]:
```
# 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
```

```
#groupA = groupA.loc[(groupA['ParentArea'] == "WellingtonRegion")] #Only␣
↪Wellington
#groupA = groupA.loc[(groupA['ParentArea'] == "OtagoRegion")] #Only Otago
groupA = groupA.loc[(groupA['ParentArea'] == "BayOfPlentyRegion")] #Only␣
↪BayOfPlenty

#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/

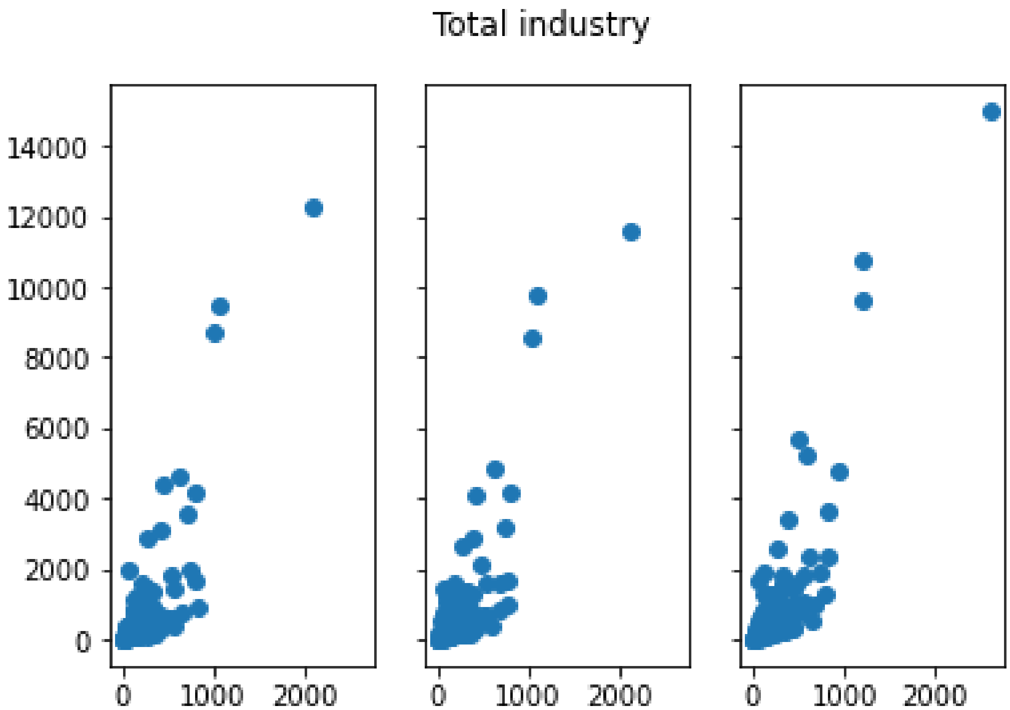
1.1 Total industry

1.1.1 Scatterplot

[339]:
```
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())
```

[340]:
```
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)
```

[340]:
```
<matplotlib.collections.PathCollection at 0x12313aaf0>
```



1.1.2 Correlation tests

[341]:
```
# 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
[[  63424.69230769  336971.23834499]
 [ 336971.23834499 2606019.04657148]]
Pearsons correlation: 0.829
Spearmans correlation: 0.712
2013
[[  63164.76748252  326496.78700466]
 [ 326496.78700466 2467933.1677836 ]]
Pearsons correlation: 0.827
Spearmans correlation: 0.673
2018
[[  85213.7583042   465564.83173077]
 [ 465564.83173077 3486023.02093046]]
Pearsons correlation: 0.854
Spearmans correlation: 0.687
```

1.1.3 Linear regression

regressor - # Facilities ; predictor - employee count

[342]:
```
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

[343]:
```
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)
```

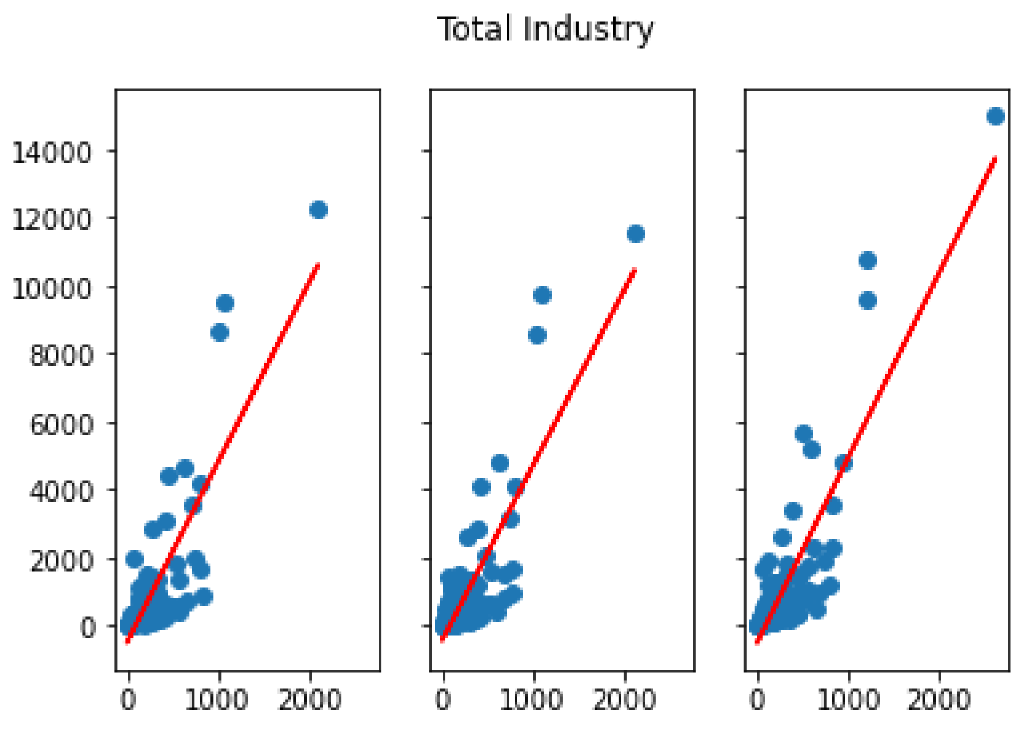
[344]:
```
print("2006")
r_sq = model_2006.score(totInd_Fac_2006, totInd_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.6869888894269253
intercept: -487.3271208925605
slope: [5.31293454]
2013
coefficient of determination: 0.6838321690024922
intercept: -432.05629255731344
slope: [5.16896998]
2018
coefficient of determination: 0.7296597039896453
intercept: -505.5000282019081
slope: [5.4634937]
```

[345]:
```
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')
```

[345]: [<matplotlib.lines.Line2D at 0x122f2aee0>]



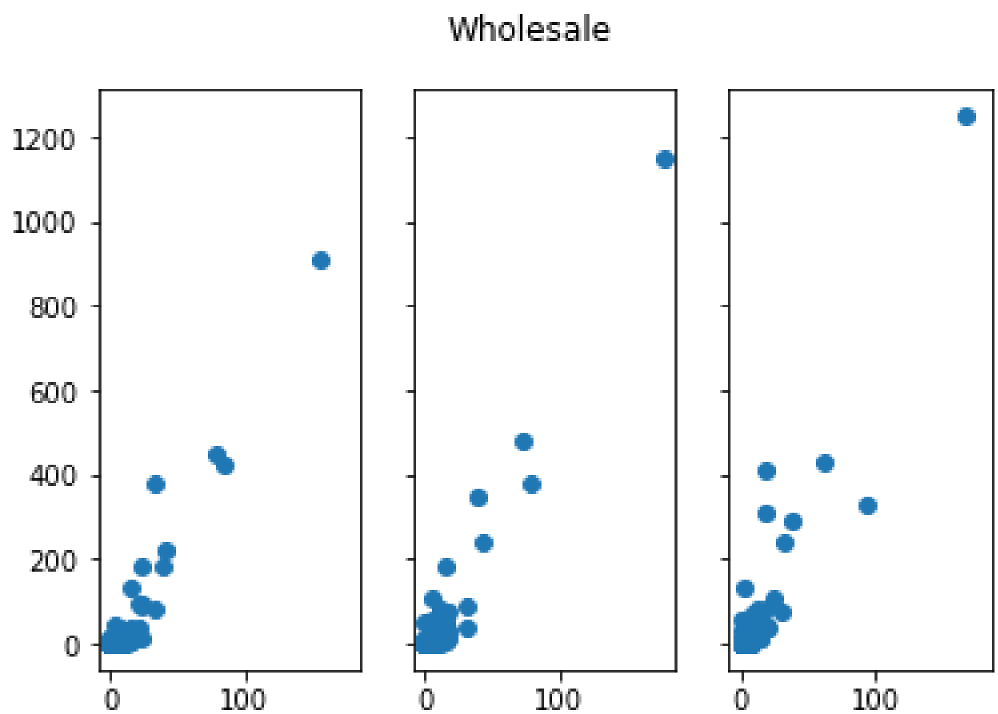
1.2 Wholesale (F)

1.2.1 Scatterplot

[346]:
```
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())
```

[347]:
```python
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)
```

[347]: <matplotlib.collections.PathCollection at 0x12302d9d0>



1.2.2 Correlation tests

[348]:
```python
# 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)
```

```
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
[[ 289.47858392 1607.78001166]
 [1607.78001166 9916.90889666]]
Pearsons correlation: 0.949
Spearmans correlation: 0.738
2013
[[  318.83916084  1940.4965035 ]
 [ 1940.4965035  12913.5506993 ]]
Pearsons correlation: 0.956
Spearmans correlation: 0.686
2018
[[  300.18181818  1959.13519814]
 [ 1959.13519814 15221.08605284]]
Pearsons correlation: 0.917
Spearmans correlation: 0.702
```

1.2.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.2.4 Create model

[349]:
```
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))
```

[350]:
```
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)
```

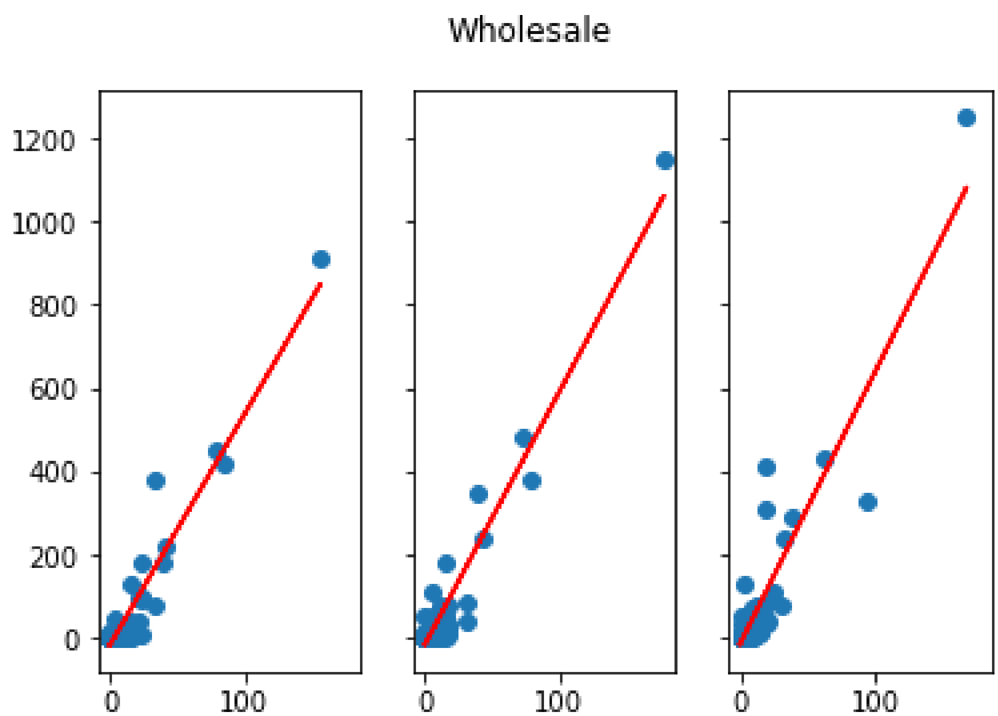
[351]:
```
print("2006")
r_sq = model_2006.score(whole_Fac_2006, whole_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.9004518344918482
intercept: -17.691356384384154
slope: [5.55405512]
2013
coefficient of determination: 0.9145519913386501
intercept: -15.854306487695734
slope: [6.08612975]
2018
coefficient of determination: 0.8400377239609632
intercept: -14.967018823090903
slope: [6.52649521]
```

[352]:
```
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')
```

[352]: [<matplotlib.lines.Line2D at 0x122ea50d0>]



1.3 Retail (G)

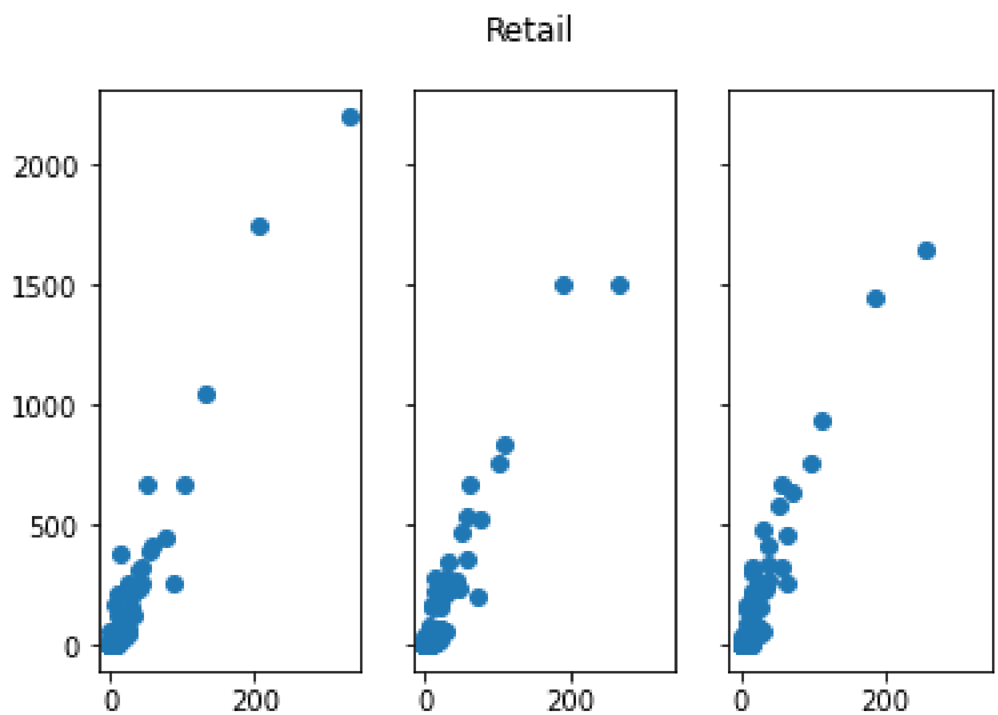
1.3.1 Scatterplot

[353]:
```
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())
```

[354]:
```
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)
```

[354]: <matplotlib.collections.PathCollection at 0x123385f70>



1.3.2 Correlation tests

[355]:
```
# Covariance

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

```python
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
[[ 1316.05550699  9379.85693473]
 [ 9379.85693473 71227.00213675]]
Pearsons correlation: 0.969
Spearmans correlation: 0.810
2013
[[  989.90166084  6612.86276224]
 [ 6612.86276224 48469.31915307]]
Pearsons correlation: 0.955
Spearmans correlation: 0.821
2018
[[  926.41783217  6785.28933566]
 [ 6785.28933566 54583.03103147]]
```

```
Pearsons correlation: 0.954
Spearmans correlation: 0.840
```

1.3.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.3.4 Create model

[356]:
```
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))
```

[357]:
```
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)
```

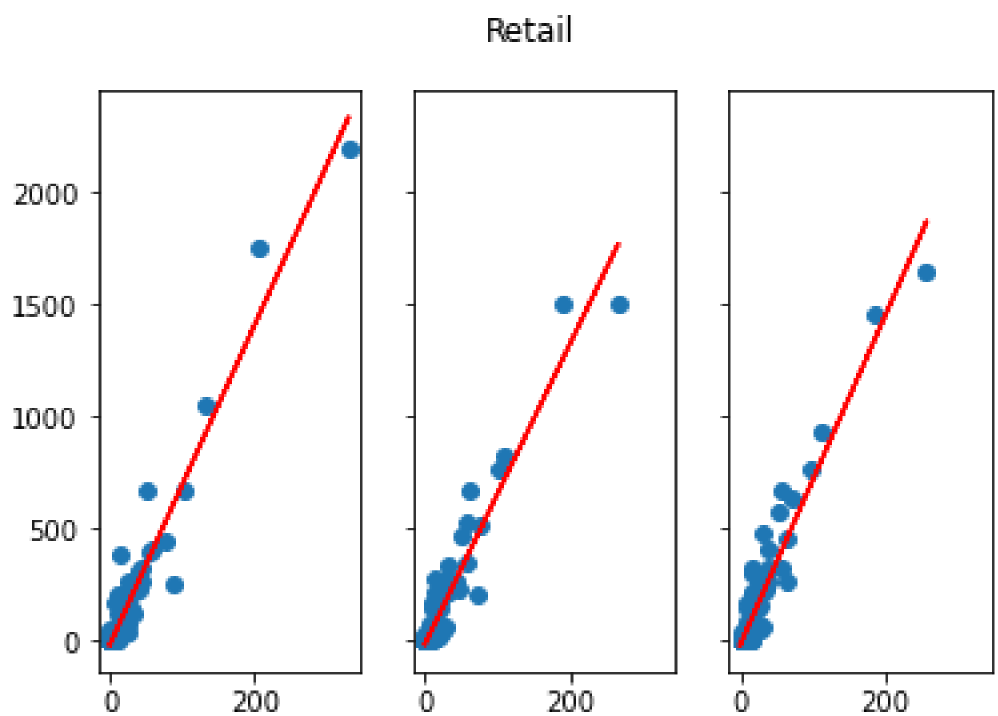
[358]:
```
print("2006")
r_sq = model_2006.score(retail_Fac_2006, retail_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.9385848853833145
intercept: -17.87869989223333
slope: [7.12725025]
2013
coefficient of determination: 0.911423133396744
intercept: -15.605061161775694
slope: [6.68032293]
2018
```

```
coefficient of determination: 0.910483727154958
intercept: -19.878459779094953
slope: [7.32422143]
```

[359]:
```python
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')
```

[359]:
```
[<matplotlib.lines.Line2D at 0x1235549d0>]
```



1.4 TransPostWare

[360]:
```python
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

[361]:
```
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())
```

[362]:
```
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)
```

[362]: <matplotlib.collections.PathCollection at 0x1236b1b80>

1.4.2 Correlation tests

[363]:
```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
[[  40.64160839  429.26311189]
 [ 429.26311189 6927.45682789]]
Pearsons correlation: 0.809
Spearmans correlation: 0.560
```

```
2013
[[  40.7618007   454.15311772]
 [ 454.15311772 7586.67244561]]
Pearsons correlation: 0.817
Spearmans correlation: 0.571
2018
[[  41.1451049   377.98164336]
 [ 377.98164336 6248.40438034]]
Pearsons correlation: 0.745
Spearmans correlation: 0.514
```

1.4.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.4.4 Create model

[364]:
```
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))
```

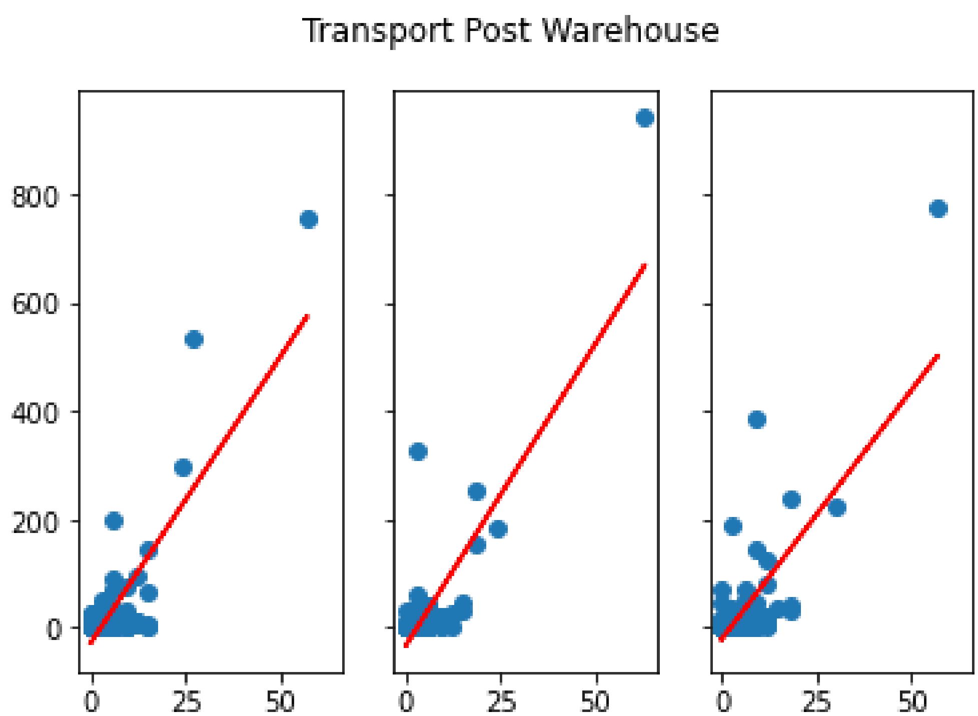
[365]:
```
model_2006 = LinearRegression().fit(tpw_Fac_2006, tpw_Emp_2006)
model_2013 = LinearRegression().fit(tpw_Fac_2013, tpw_Emp_2013)
model_2018 = LinearRegression().fit(tpw_Fac_2018, tpw_Emp_2018)
```

[366]:
```
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: 0.6544891095742873
intercept: -27.19026110896028
slope: [10.56215856]
2013
coefficient of determination: 0.6669601786452468
intercept: -31.68579179310122
slope: [11.1416353]
2018
coefficient of determination: 0.5557175543032851
intercept: -21.31916294879965
slope: [9.18655194]
```

[367]:
```
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')
```

[367]:
```
[<matplotlib.lines.Line2D at 0x12382b430>]
```



1.5 Transport

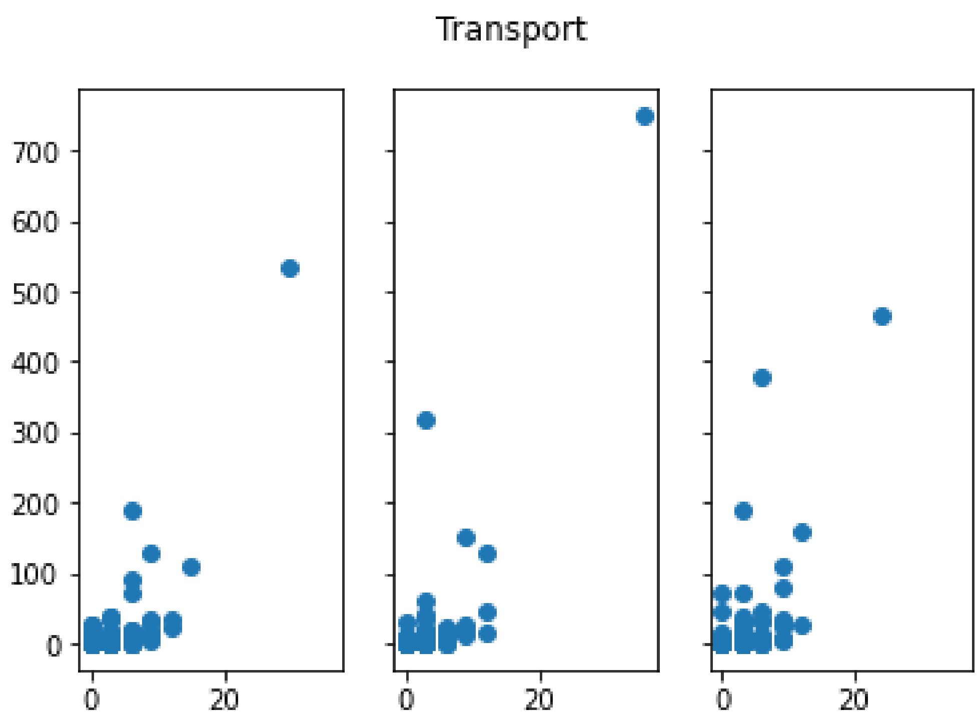
[368]:
```
groupA['Transport_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I481_G
groupA['Transport_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']
```

1.5.1 Scatterplot

[369]:
```
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())
```

[370]:
```
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)
```

[370]:
```
<matplotlib.collections.PathCollection at 0x123991430>
```



1.5.2 Correlation tests

[371]:
```
# 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)
```

```
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
[[  15.04152098  131.5259324 ]
 [ 131.5259324  2463.50097125]]
Pearsons correlation: 0.683
Spearmans correlation: 0.587
2013
[[  15.39685315  198.87470862]
 [ 198.87470862 4827.48232323]]
Pearsons correlation: 0.729
Spearmans correlation: 0.562
2018
[[  12.38767483  104.17555361]
 [ 104.17555361 3009.89039433]]
Pearsons correlation: 0.540
Spearmans correlation: 0.542
```

1.5.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.5.4 Create model

[372]:
```
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))
```

[373]:
```
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)
```

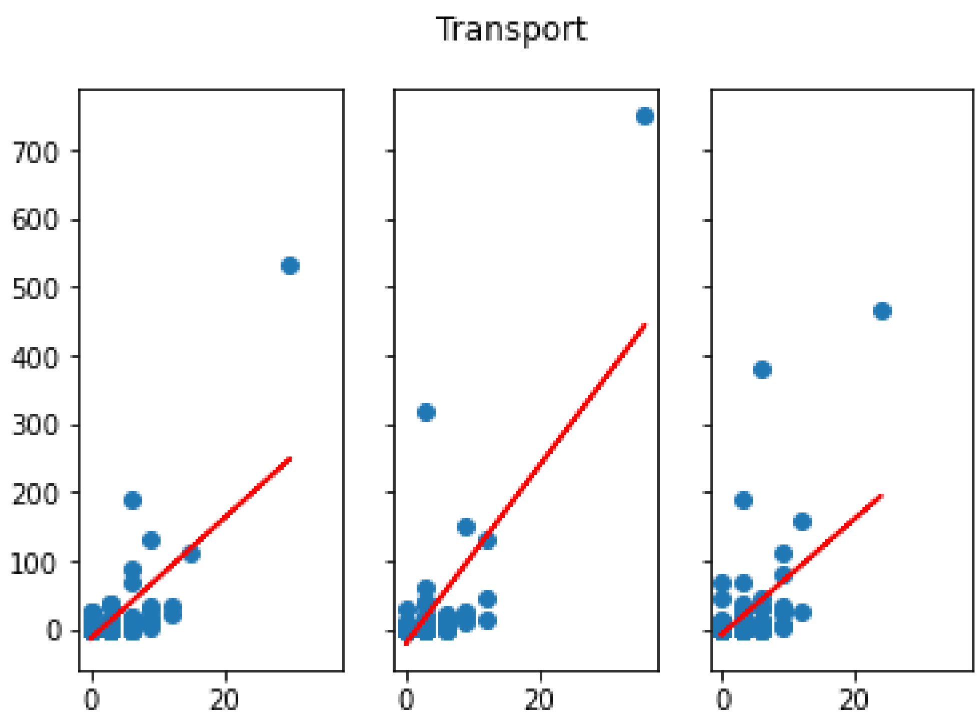
[374]:
```
print("2006")
r_sq = model_2006.score(trans_Fac_2006, trans_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.4668509929768683
intercept: -13.817521429609204
slope: [8.744191]
2013
coefficient of determination: 0.5321161656659776
intercept: -20.718519359600297
slope: [12.91658151]
2018
coefficient of determination: 0.29106578481180934
intercept: -8.01256042056239
slope: [8.40961319]
```

[375]:
```
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')
```

[375]: [<matplotlib.lines.Line2D at 0x123b28970>]



1.6 Post and storage

[376]:
```
groupA['Storage_GeogUnits']=groupA['I51_GeogUnits']+groupA['I53_GeogUnits']
groupA['Storage_EmpCo']=groupA['I51_EmpCo']+groupA['I53_EmpCo']
```

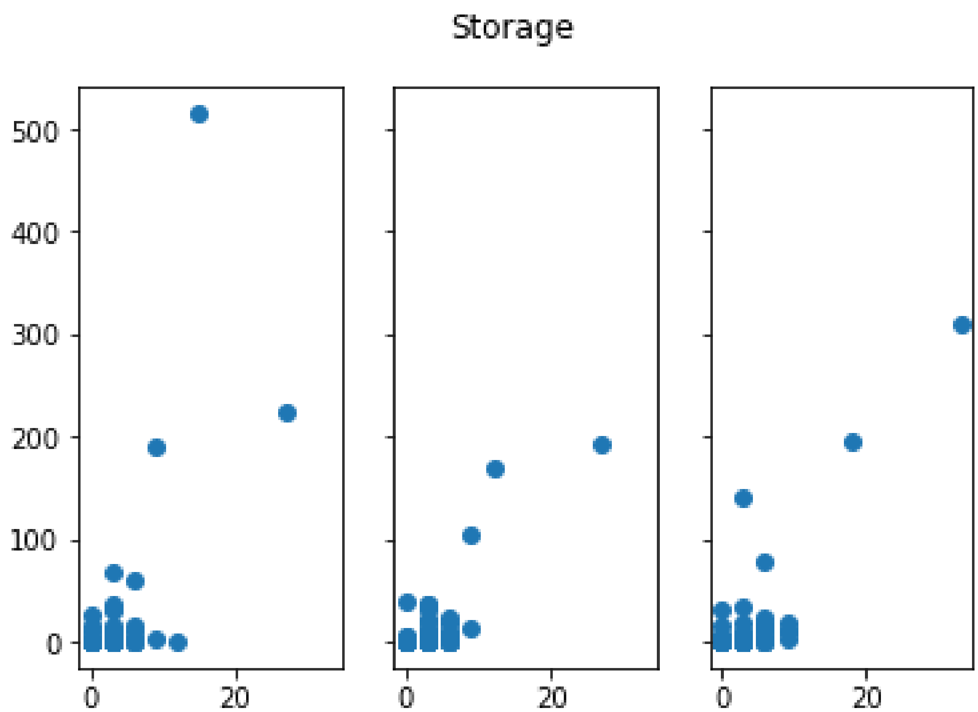
1.6.1 Scatterplot

[377]:
```
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())
```

[378]:
```
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)
```

[378]: <matplotlib.collections.PathCollection at 0x123bdf6a0>



1.6.2 Correlation tests

[379]:
```
# Covariance

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

```python
# 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
[[  10.81424825  102.08784965]
 [ 102.08784965 2463.62543706]]
Pearsons correlation: 0.625
Spearmans correlation: 0.488
2013
[[  9.75131119  54.3868007 ]
 [ 54.3868007  555.99956294]]
Pearsons correlation: 0.739
Spearmans correlation: 0.565
2018
[[  13.8666958    96.21241259]
 [  96.21241259 1099.59265734]]
Pearsons correlation: 0.779
Spearmans correlation: 0.518
```

1.6.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.6.4 Create model

[380]:
```
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))
```

[381]:
```
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)
```

[382]:
```
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_)
```

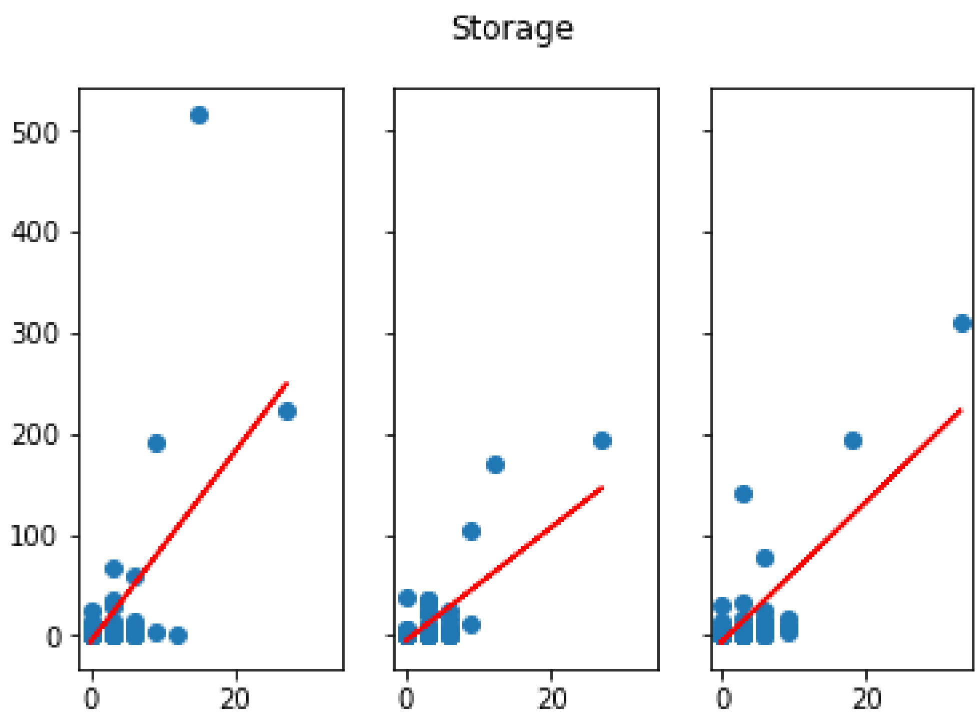
```
2006
coefficient of determination: 0.3911804100010018
intercept: -6.08103301943984
slope: [9.44012448]
2013
coefficient of determination: 0.5455688405008482
intercept: -4.993680247411586
slope: [5.57738335]
2018
coefficient of determination: 0.6070960265033266
intercept: -6.607211523308223
slope: [6.93838056]
```

[383]:

```python
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')
```

[383]:

```
[<matplotlib.lines.Line2D at 0x123f4bfa0>]
```



[ ]: