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)
```

[246]:
```
# 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/

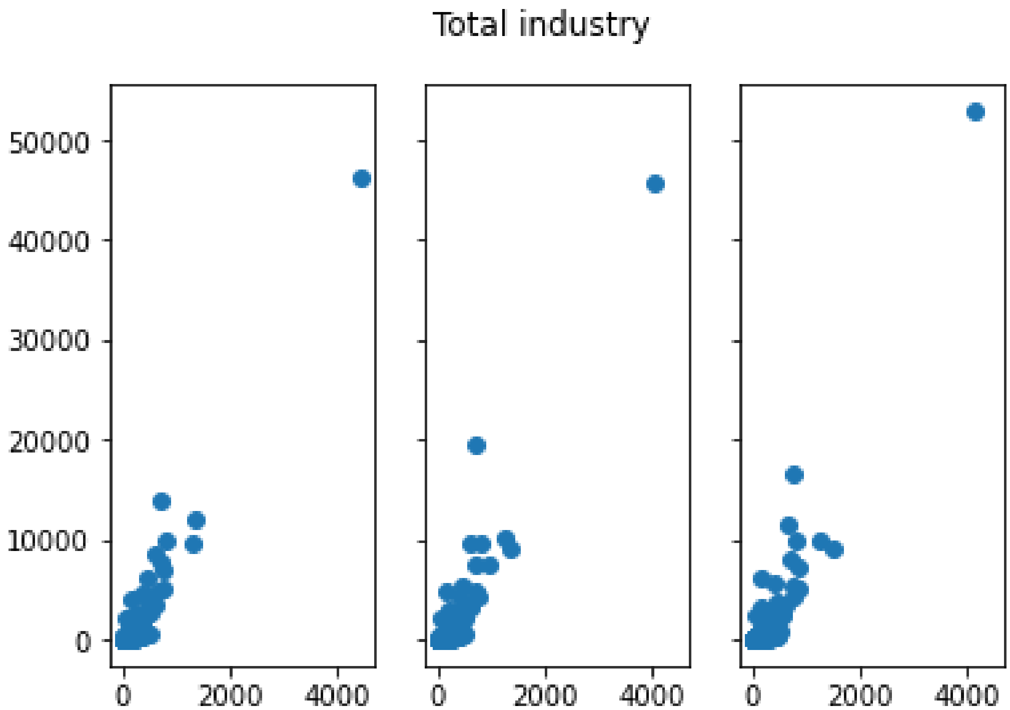
1.1 Total industry

1.1.1 Scatterplot

[247]:
```
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())
```

[248]:
```
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)
```

[248]:
```
<matplotlib.collections.PathCollection at 0x11e54c3a0>
```



1.1.2 Correlation tests

[249]:
```
# 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
[[  114193.47266558  1164740.14528088]
 [ 1164740.14528088 13102541.76741653]]
Pearsons correlation: 0.952
Spearmans correlation: 0.709
2013
[[  100610.21389971  1063953.59928365]
 [ 1063953.59928365 13444006.21113485]]
Pearsons correlation: 0.915
Spearmans correlation: 0.681
2018
[[  105686.13359306  1201853.79414352]
 [ 1201853.79414352 16424420.03267563]]
Pearsons correlation: 0.912
Spearmans correlation: 0.685
```

1.1.3 Linear regression

regressor - # Facilities ; predictor - employee count

[250]:
```
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

[251]:
```
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)
```

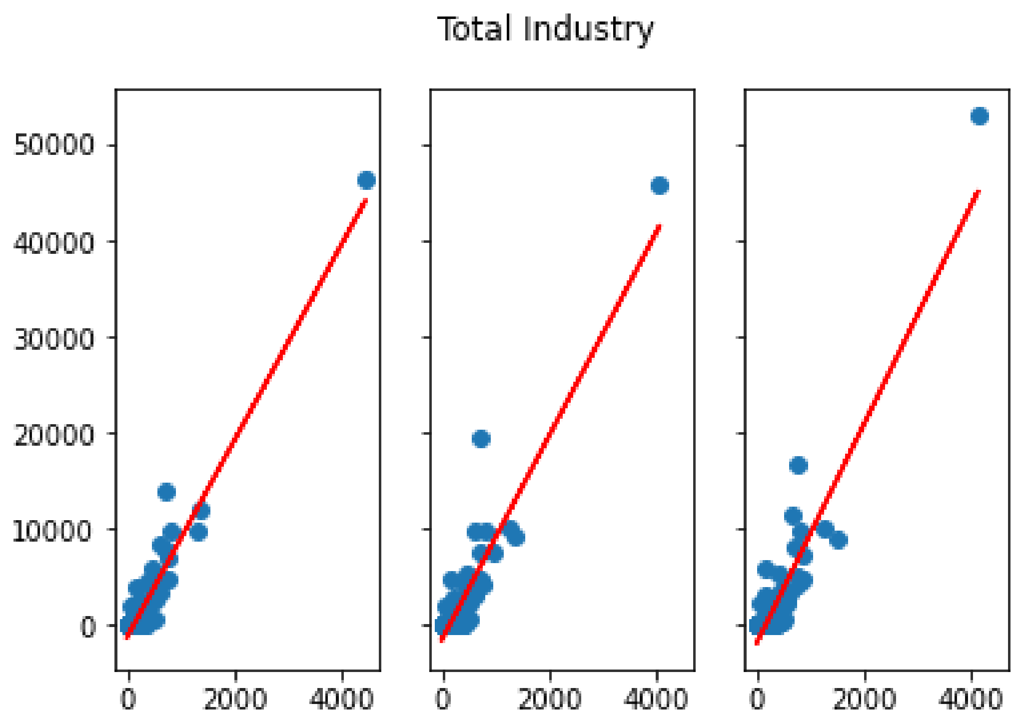
[252]:
```
print("2006")
r_sq = model_2006.score(totInd_Fac_2006, totInd_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.9066950739926828
intercept: -1248.4688285348286
slope: [10.1997086]
2013
coefficient of determination: 0.8369019878721475
intercept: -1450.717496607781
slope: [10.57500584]
2018
coefficient of determination: 0.8321377507129403
intercept: -1780.2826853575054
slope: [11.37191563]
```

[253]:
```
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')
```

[253]: [<matplotlib.lines.Line2D at 0x11e64f8b0>]



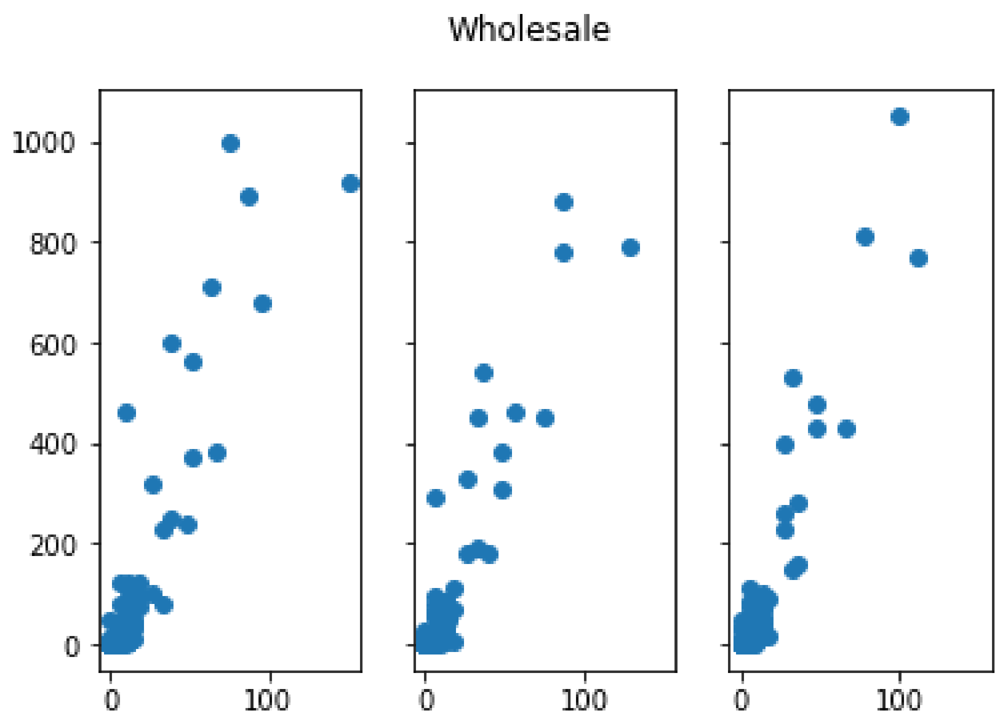
1.2 Wholesale (F)

1.2.1 Scatterplot

[254]:
```
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())
```

[255]:
```
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)
```

[255]: <matplotlib.collections.PathCollection at 0x11e72b3a0>



1.2.2 Correlation tests

[256]:
```
# 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
[[  274.23400779  2204.84014076]
 [ 2204.84014076 21742.59017218]]
Pearsons correlation: 0.903
Spearmans correlation: 0.668
2013
[[  217.5397763   1661.89587784]
 [ 1661.89587784 14859.71815173]]
Pearsons correlation: 0.924
Spearmans correlation: 0.743
2018
[[  185.83184617  1604.77623476]
 [ 1604.77623476 15830.29324285]]
Pearsons correlation: 0.936
Spearmans correlation: 0.687
```

1.2.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.2.4 Create model

[257]:
```
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))
```

[258]:
```
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)
```

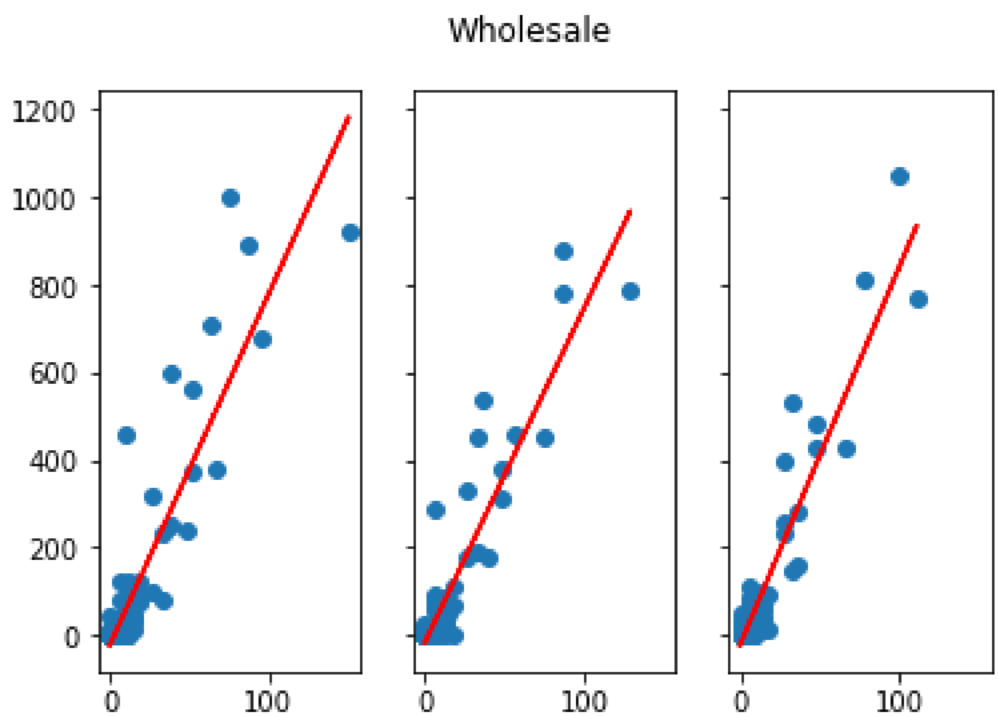
[259]:
```
print("2006")
r_sq = model_2006.score(whole_Fac_2006, whole_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.8153078490150397
intercept: -23.103522327320718
slope: [8.03999533]
2013
coefficient of determination: 0.8543943573396555
intercept: -16.948061311500418
slope: [7.63950348]
2018
coefficient of determination: 0.8754268590355829
intercept: -22.95800364382984
slope: [8.63563629]
```

[260]:
```
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')
```

[260]: [<matplotlib.lines.Line2D at 0x120d46850>]



1.3 Retail (G)

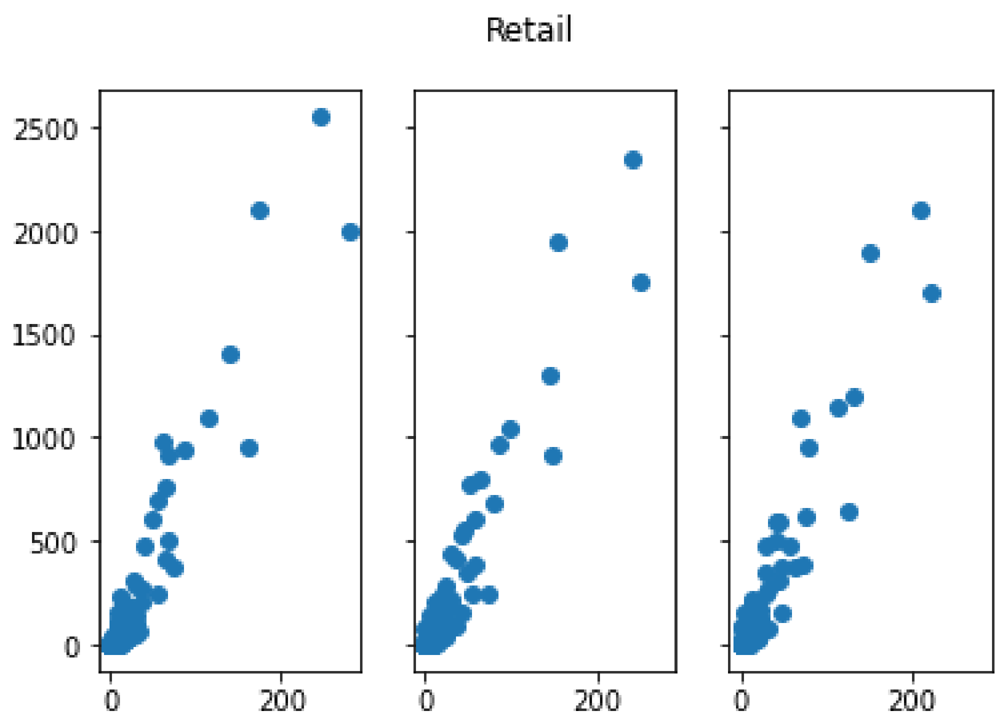
1.3.1 Scatterplot

[261]:
```
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())
```

[262]:
```
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)
```

[262]: <matplotlib.collections.PathCollection at 0x120e64280>



1.3.2 Correlation tests

[263]:
```
# 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
[[  1117.84479075  10185.07427422]
 [ 10185.07427422 102393.43986427]]
Pearsons correlation: 0.952
Spearmans correlation: 0.852
2013
[[  983.58376272  8747.57653638]
 [ 8747.57653638 86304.89447447]]
Pearsons correlation: 0.949
Spearmans correlation: 0.799
2018
[[  813.6695991   7568.0397763 ]
 [ 7568.0397763  78552.66385991]]
```

```
Pearsons correlation: 0.947
Spearmans correlation: 0.828
```

1.3.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.3.4 Create model

[264]:
```python
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))
```

[265]:
```python
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)
```

[266]:
```python
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: 0.9063058021675727
intercept: -38.54712693223405
slope: [9.11134923]
2013
coefficient of determination: 0.9014231877752285
intercept: -36.86781966519368
slope: [8.89357558]
2018
```

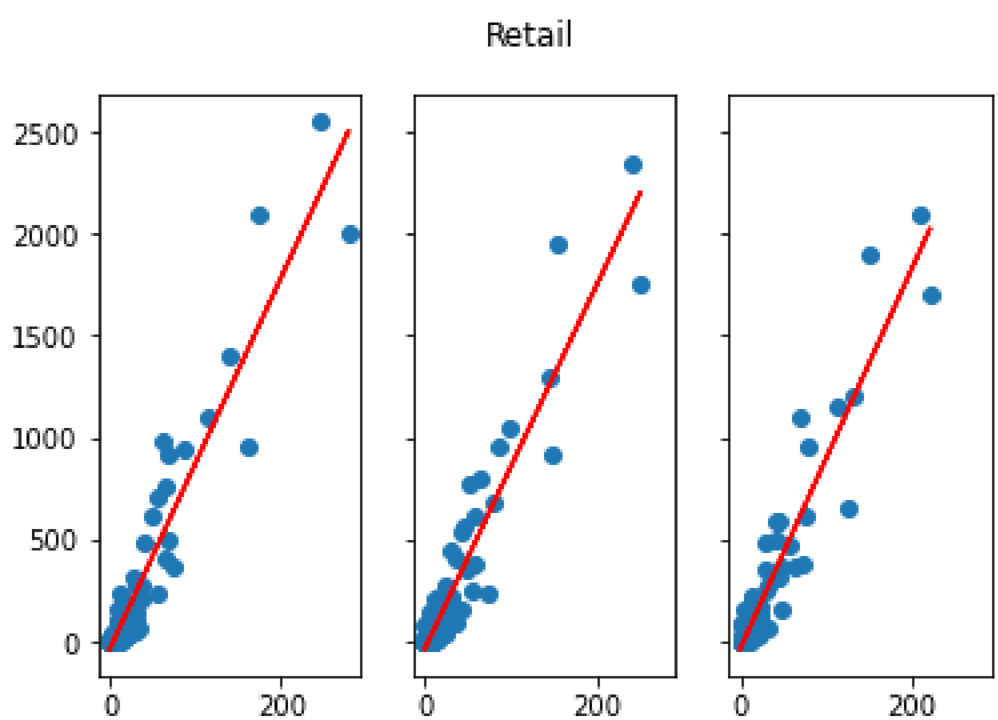
```
coefficient of determination: 0.8961027389506979
intercept: -39.035010052717084
slope: [9.30112147]
```

[267]:

```
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')
```

[267]:

```
[<matplotlib.lines.Line2D at 0x120fded30>]
```



1.4 TransPostWare

[268]:

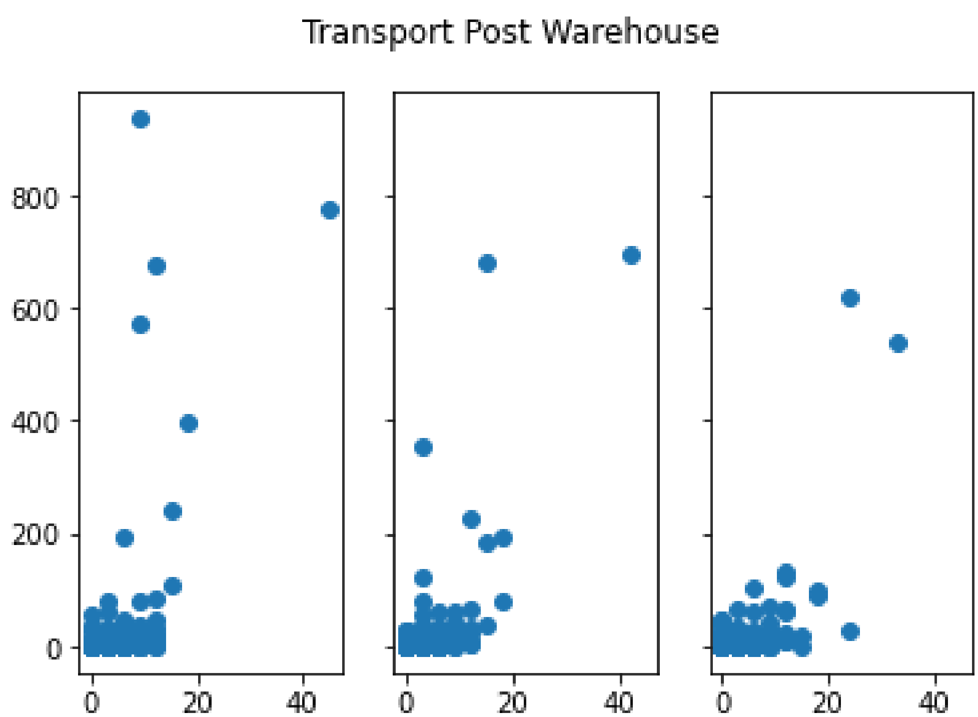
```
groupA['TransPostWare_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I4
groupA['TransPostWare_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']+g
```

1.4.1 Scatterplot

[269]:
```
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())
```

[270]:
```
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)
```

[270]: <matplotlib.collections.PathCollection at 0x121155f10>



1.4.2 Correlation tests

[271]:
```
# 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
[[   22.04511751   262.4725399 ]
 [  262.4725399  11200.39793054]]
Pearsons correlation: 0.528
Spearmans correlation: 0.494
```

```
2013
[[  20.88538394  209.27987935]
 [ 209.27987935 5450.84868669]]
Pearsons correlation: 0.620
Spearmans correlation: 0.550
2018
[[  22.57898706  170.76329018]
 [ 170.76329018 3389.27212936]]
Pearsons correlation: 0.617
Spearmans correlation: 0.548
```

1.4.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.4.4 Create model

[272]:
```
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))
```

[273]:
```
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)
```

[274]:
```
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: 0.27901135419713663
intercept: -22.12946588907321
slope: [11.90615291]
2013
coefficient of determination: 0.3847231879785534
intercept: -15.768210127267803
slope: [10.02039895]
2018
coefficient of determination: 0.3810466370119455
intercept: -12.74528695710256
slope: [7.56292963]
```

[275]:
```
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')
```

[275]:
```
[<matplotlib.lines.Line2D at 0x1212de7f0>]
```

1.5 Transport

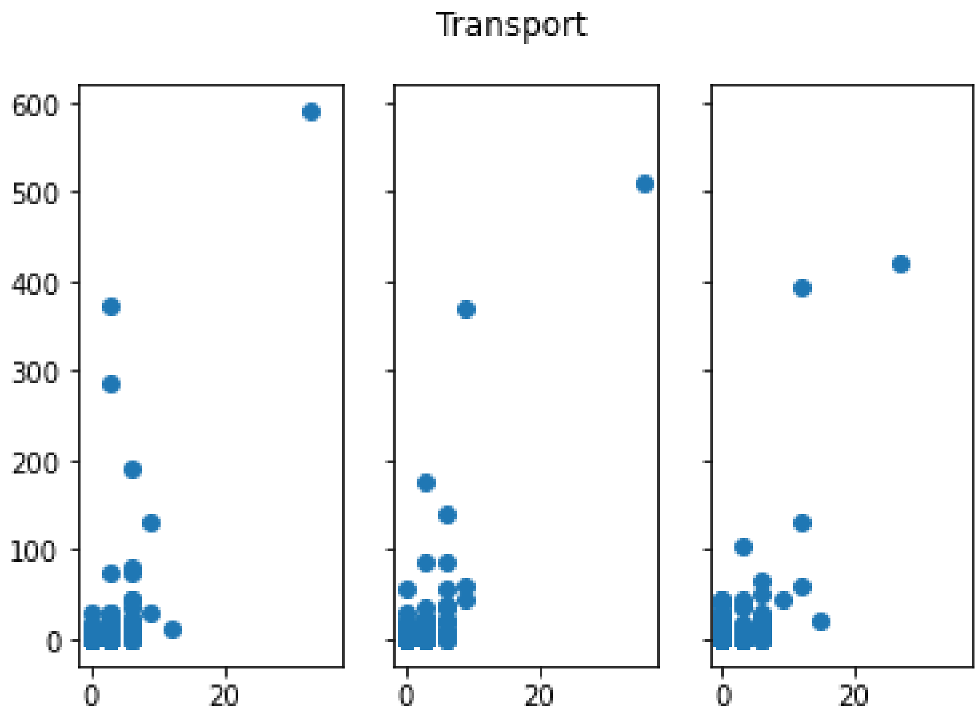
[276]:
```
groupA['Transport_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I481_G
groupA['Transport_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']
```

1.5.1 Scatterplot

[277]:
```
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())
```

[278]:
```
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)
```

[278]:
```
<matplotlib.collections.PathCollection at 0x1214527f0>
```



1.5.2 Correlation tests

[279]:
```
# 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
[[   9.70353148  104.82807591]
 [ 104.82807591 2862.68158016]]
Pearsons correlation: 0.629
Spearmans correlation: 0.490
2013
[[   9.99685811  108.11643836]
 [ 108.11643836 2121.45234804]]
Pearsons correlation: 0.742
Spearmans correlation: 0.579
2018
[[   9.73972603   85.4389217 ]
 [  85.4389217  1707.01374052]]
Pearsons correlation: 0.663
Spearmans correlation: 0.551
```

1.5.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.5.4 Create model

[280]:
```
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))
```

[281]:
```
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)
```

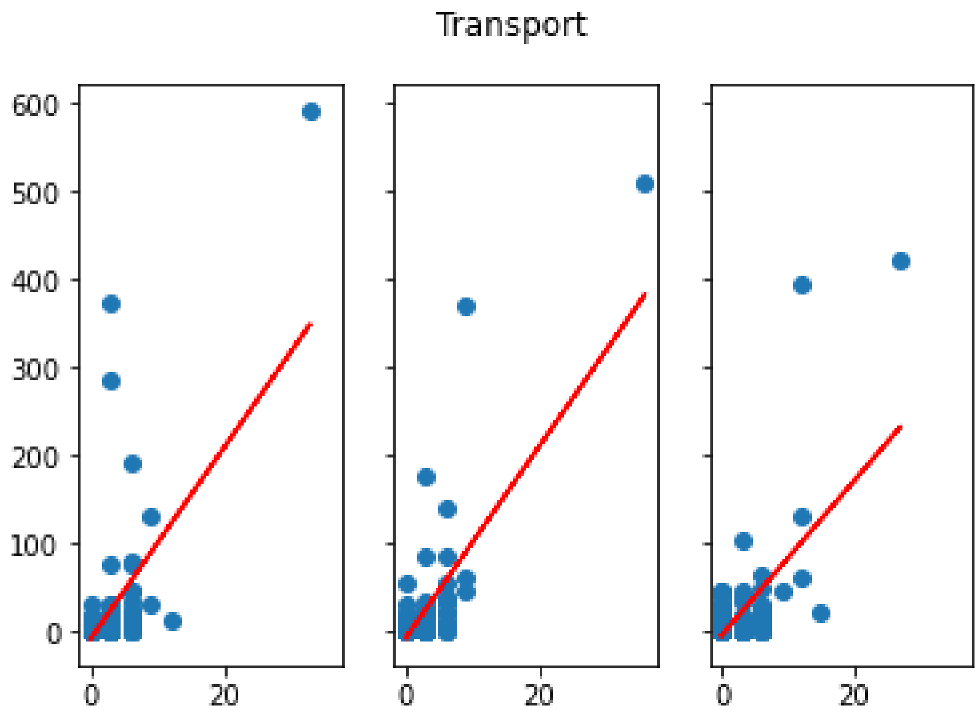
[282]:
```
print("2006")
r_sq = model_2006.score(trans_Fac_2006, trans_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.3955964323060728
intercept: -8.650580875781959
slope: [10.80308505]
2013
coefficient of determination: 0.5511713714411792
intercept: -7.401188006788601
slope: [10.8150418]
2018
coefficient of determination: 0.43906392930230775
intercept: -5.118472496419313
slope: [8.77220996]
```

[283]:
```
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')
```

[283]: [<matplotlib.lines.Line2D at 0x1215d3f70>]



1.6 Post and storage

[284]:
```
groupA['Storage_GeogUnits']=groupA['I51_GeogUnits']+groupA['I53_GeogUnits']
groupA['Storage_EmpCo']=groupA['I51_EmpCo']+groupA['I53_EmpCo']
```

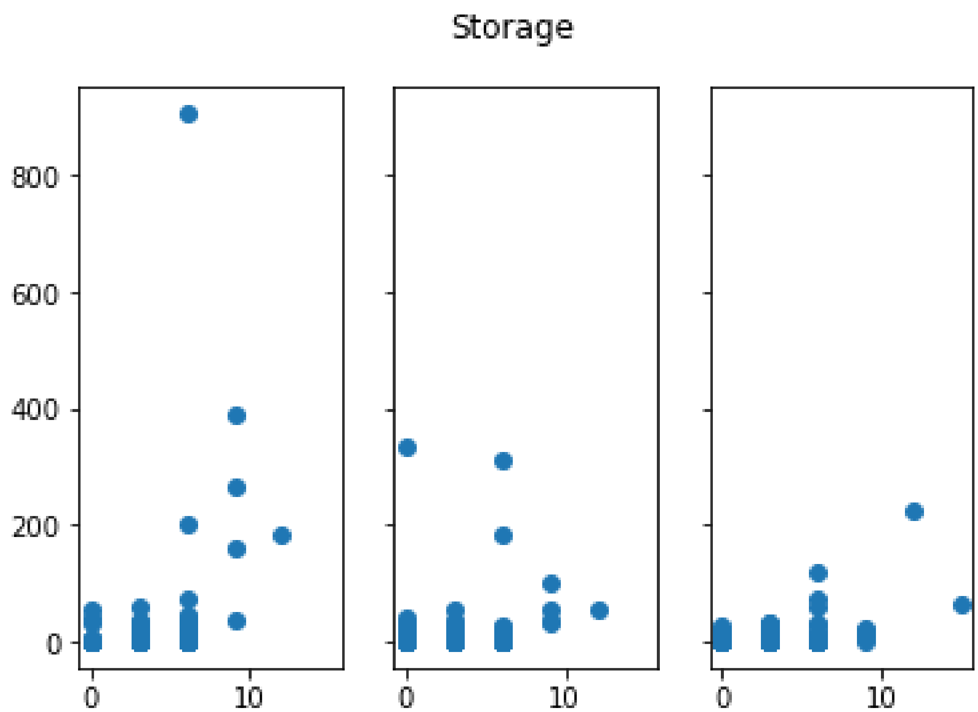
1.6.1 Scatterplot

[285]:
```
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())
```

[286]:
```
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)
```

[286]: <matplotlib.collections.PathCollection at 0x12174aeb0>



1.6.2 Correlation tests

[287]:
```
# 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
[[   6.04825939   60.68279502]
 [  60.68279502 5196.18549705]]
Pearsons correlation: 0.342
Spearmans correlation: 0.440
2013
[[   5.82882996   19.45337439]
 [  19.45337439 1203.51443174]]
Pearsons correlation: 0.232
Spearmans correlation: 0.479
2018
[[  7.25059696  23.6346613 ]
 [ 23.6346613  382.87168531]]
Pearsons correlation: 0.449
Spearmans correlation: 0.457
```

1.6.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.6.4 Create model

[288]:
```
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))
```

[289]:
```
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)
```

[290]:
```
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_)
```

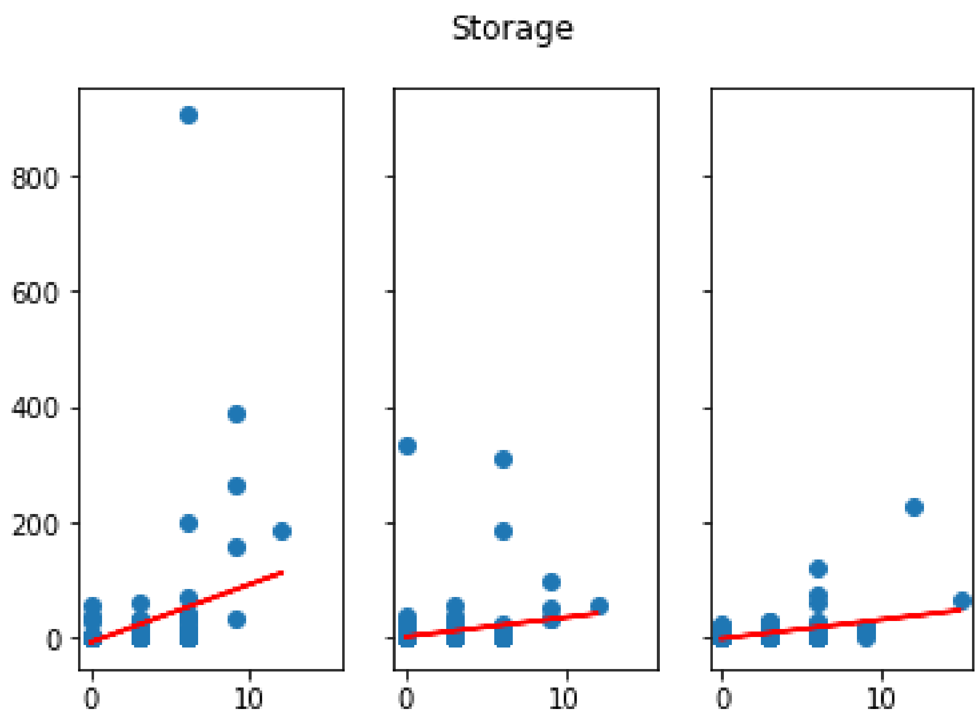
```
2006
coefficient of determination: 0.11716991015134426
intercept: -7.493579354195237
slope: [10.03310061]
2013
coefficient of determination: 0.053945745776386445
intercept: 2.033893919793014
slope: [3.33744071]
2018
coefficient of determination: 0.20122028121647828
intercept: -1.2347251832978046
slope: [3.25968488]
```

[291]:

```
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')
```

[291]:

```
[<matplotlib.lines.Line2D at 0x1218cb910>]
```



[ ]: