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)
```

[153]:
```
# 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

#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/

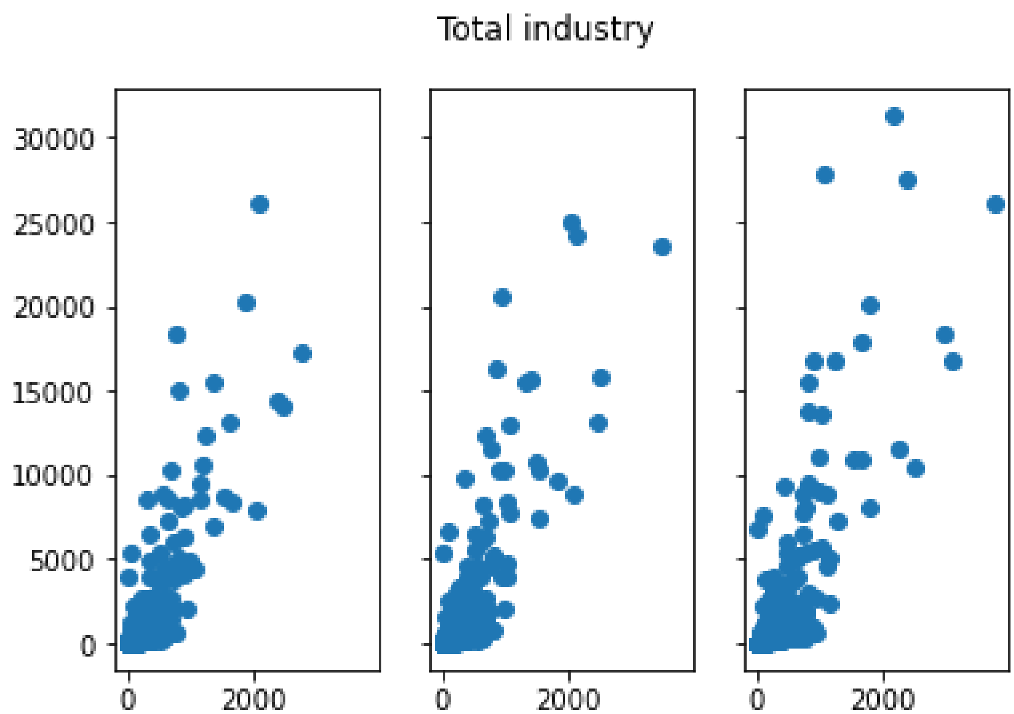
1.1 Total industry

1.1.1 Scatterplot

[154]:
```
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())
```

[155]:
```
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)
```

[155]:
```
<matplotlib.collections.PathCollection at 0x11e47a310>
```



1.1.2 Correlation tests

[156]:
```
# 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
[[  95173.15059765  669185.56599432]
 [ 669185.56599432 7108362.02322332]]
Pearsons correlation: 0.814
Spearmans correlation: 0.691
2013
[[ 106107.26085472  764844.44989665]
 [ 764844.44989665 8538721.69634583]]
Pearsons correlation: 0.804
Spearmans correlation: 0.666
2018
[[  135381.66068911   995110.51146312]
 [  995110.51146312 11979410.30211184]]
Pearsons correlation: 0.781
Spearmans correlation: 0.654
```

1.1.3 Linear regression

regressor - # Facilities ; predictor - employee count

[157]:
```
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

[158]:
```
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)
```

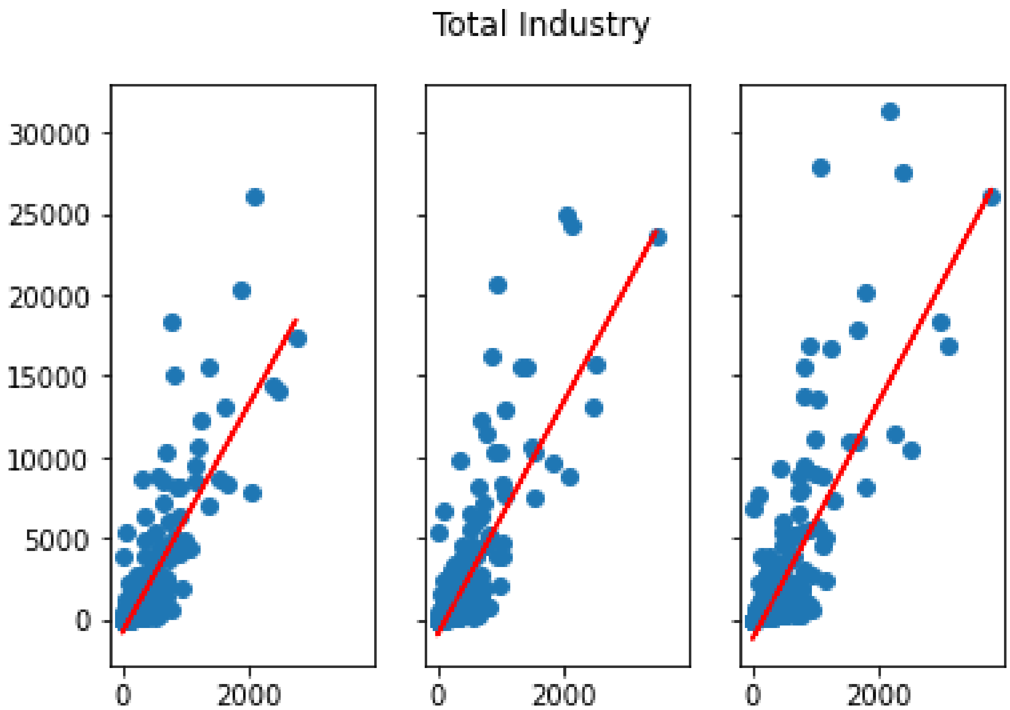
[160]:
```
print("2006")
r_sq = model_2006.score(totInd_Fac_2006, totInd_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.661925550672053
intercept: -832.7415775723923
slope: [7.03124318]
2013
coefficient of determination: 0.6456665342738417
intercept: -958.2647734074387
slope: [7.20821972]
2018
coefficient of determination: 0.6105867143755628
intercept: -1183.460647365669
slope: [7.35040851]
```

[161]:
```
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')
```

[161]: [<matplotlib.lines.Line2D at 0x11e4fb6d0>]



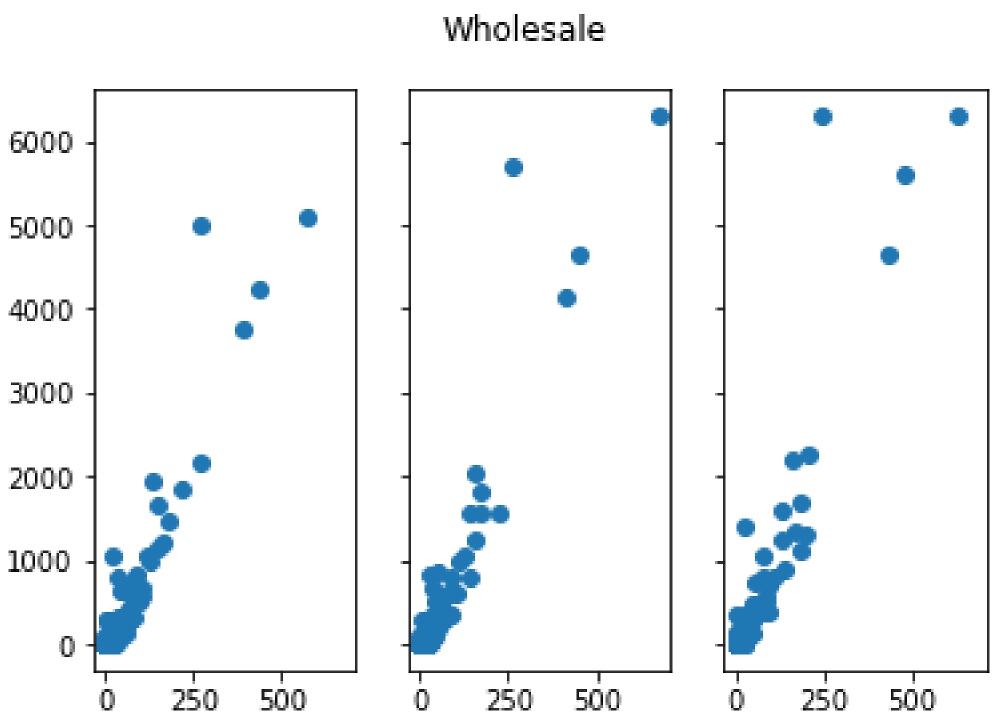
1.2 Wholesale (F)

1.2.1 Scatterplot

[162]:
```
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())
```

[163]:
```
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)
```

[163]: <matplotlib.collections.PathCollection at 0x11b88a100>



1.2.2 Correlation tests

[164]:
```
# 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
[[  1834.67818562  17687.40127242]
 [ 17687.40127242 191735.24810528]]
Pearsons correlation: 0.943
Spearmans correlation: 0.736
2013
[[  2002.83692471  20354.43748222]
 [ 20354.43748222 234031.31458948]]
Pearsons correlation: 0.940
Spearmans correlation: 0.727
2018
[[  2034.1732268   22179.10503593]
 [ 22179.10503593 280603.40844358]]
Pearsons correlation: 0.928
Spearmans correlation: 0.754
```

1.2.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.2.4 Create model

[165]:
```
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))
```

[166]:
```
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)
```

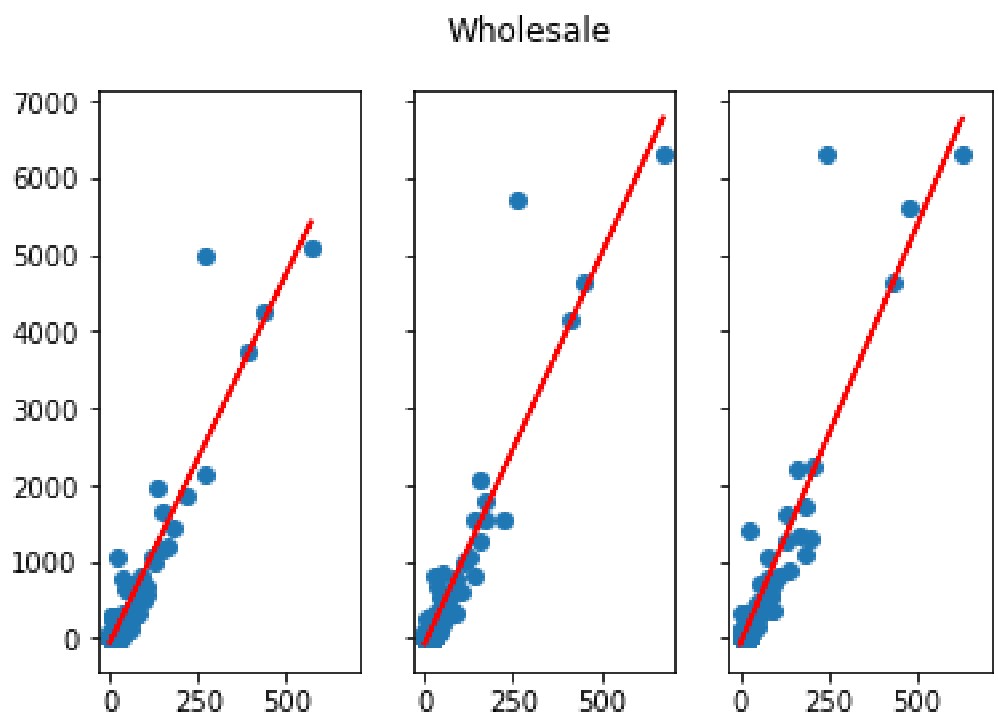
[167]:
```
print("2006")
r_sq = model_2006.score(whole_Fac_2006, whole_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.8893366710733217
intercept: -67.5810013228765
slope: [9.6406015]
2013
coefficient of determination: 0.8838908694086772
intercept: -75.42897566869821
slope: [10.16280319]
2018
coefficient of determination: 0.8618013286861013
intercept: -87.80228888822448
slope: [10.90325285]
```

[168]:
```
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')
```

[168]: [<matplotlib.lines.Line2D at 0x11d7584c0>]



1.3 Retail (G)

1.3.1 Scatterplot

[169]:
```
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())
```

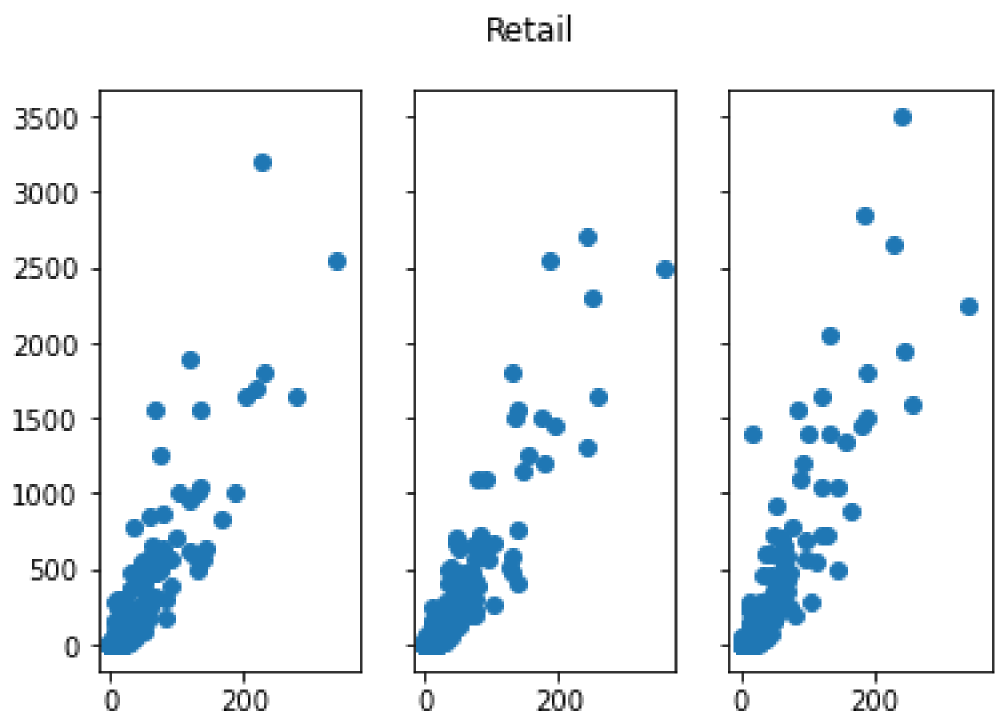
```
retail_Emp_2013 = np.array(groupA.loc[(groupA['Year'] == 2013)].G_EmpCo.
 ↪tolist())

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())
```

[170]:

```
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)
```

[170]:

```
<matplotlib.collections.PathCollection at 0x11dfecc40>
```



1.3.2 Correlation tests

[171]:

```
# 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
[[ 1182.09310822  9209.06652845]
 [ 9209.06652845 88971.83167829]]
Pearsons correlation: 0.898
Spearmans correlation: 0.806
2013
[[ 1295.2700012  10205.46999425]
 [10205.46999425 95258.89797286]]
Pearsons correlation: 0.919
Spearmans correlation: 0.804
2018
[[  1263.5558997   11336.70162702]
 [ 11336.70162702 128215.62436237]]
```

```
Pearsons correlation: 0.891
Spearmans correlation: 0.794
```

1.3.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.3.4 Create model

[172]:
```
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))
```

[173]:
```
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)
```

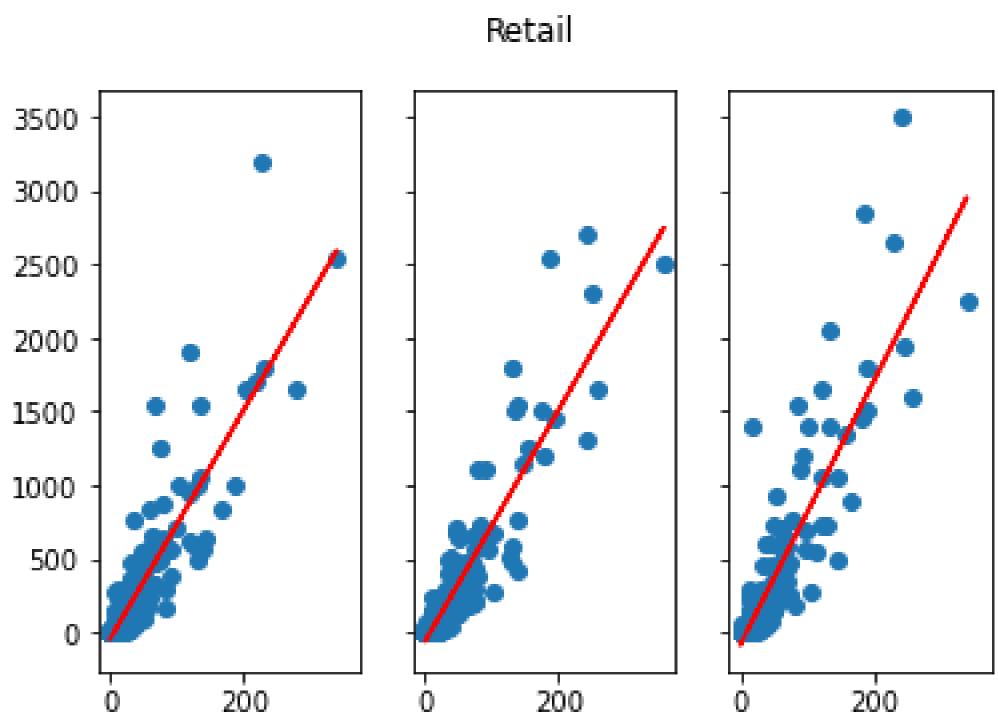
[174]:
```
print("2006")
r_sq = model_2006.score(retail_Fac_2006, retail_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.8063563381547219
intercept: -49.034226345452126
slope: [7.79047476]
2013
coefficient of determination: 0.844112166726288
intercept: -64.04877492489598
slope: [7.87902907]
2018
```

```
coefficient of determination: 0.7933010362505684
intercept: -88.09156000519746
slope: [8.97206181]
```

[175]:
```
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')
```

[175]:
```
[<matplotlib.lines.Line2D at 0x11ebbb520>]
```



1.4 TransPostWare

[176]:
```
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

[177]:
```
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())
```

[178]:
```
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)
```

[178]: <matplotlib.collections.PathCollection at 0x11ed2e250>

1.4.2 Correlation tests

[179]:

```
# 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
[[   39.72918339   682.8054272 ]
 [  682.8054272  22471.84084373]]
Pearsons correlation: 0.723
Spearmans correlation: 0.478
```

```
2013
[[   37.63100573   637.88355467]
 [  637.88355467 20024.05161723]]
Pearsons correlation: 0.735
Spearmans correlation: 0.524
2018
[[   45.27329444   778.2653047 ]
 [  778.2653047  27900.08616145]]
Pearsons correlation: 0.692
Spearmans correlation: 0.619
```

1.4.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.4.4 Create model

[180]:
```
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))
```

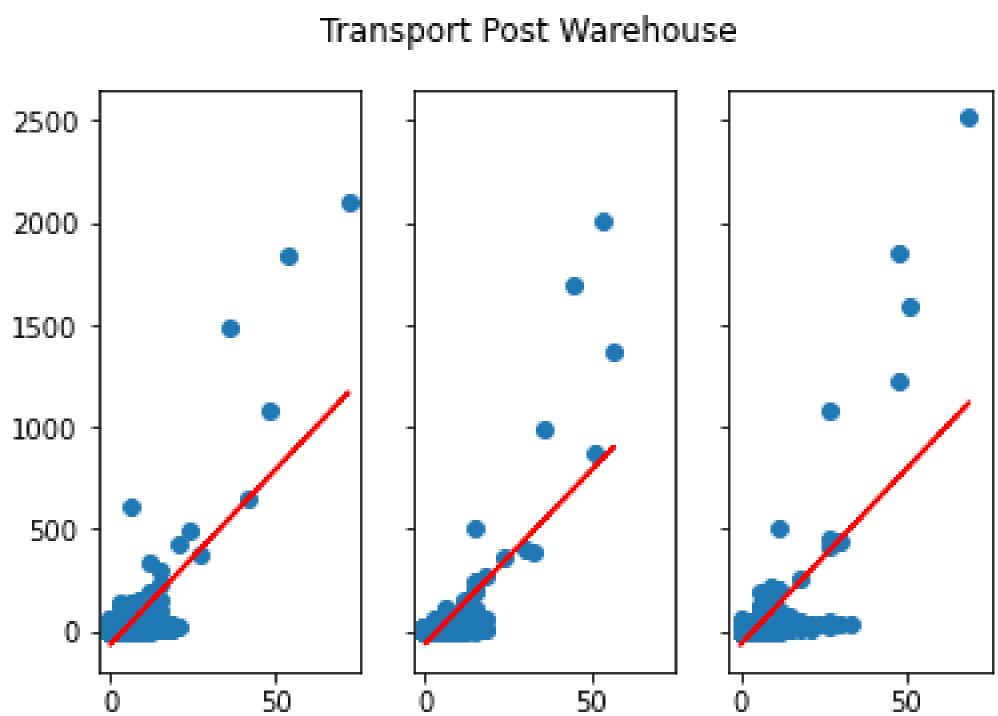
[181]:
```
model_2006 = LinearRegression().fit(tpw_Fac_2006, tpw_Emp_2006)
model_2013 = LinearRegression().fit(tpw_Fac_2013, tpw_Emp_2013)
model_2018 = LinearRegression().fit(tpw_Fac_2018, tpw_Emp_2018)
```

[182]:
```
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: 0.5222105482431673
intercept: -69.79126644557851
slope: [17.18649539]
2013
coefficient of determination: 0.539989145247184
intercept: -64.82415848050286
slope: [16.95101]
2018
coefficient of determination: 0.4795210725990203
intercept: -69.90633799896706
slope: [17.19038374]
```

[183]:
```
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')
```

[183]:
```
[<matplotlib.lines.Line2D at 0x11ee9bbe0>]
```



1.5 Transport

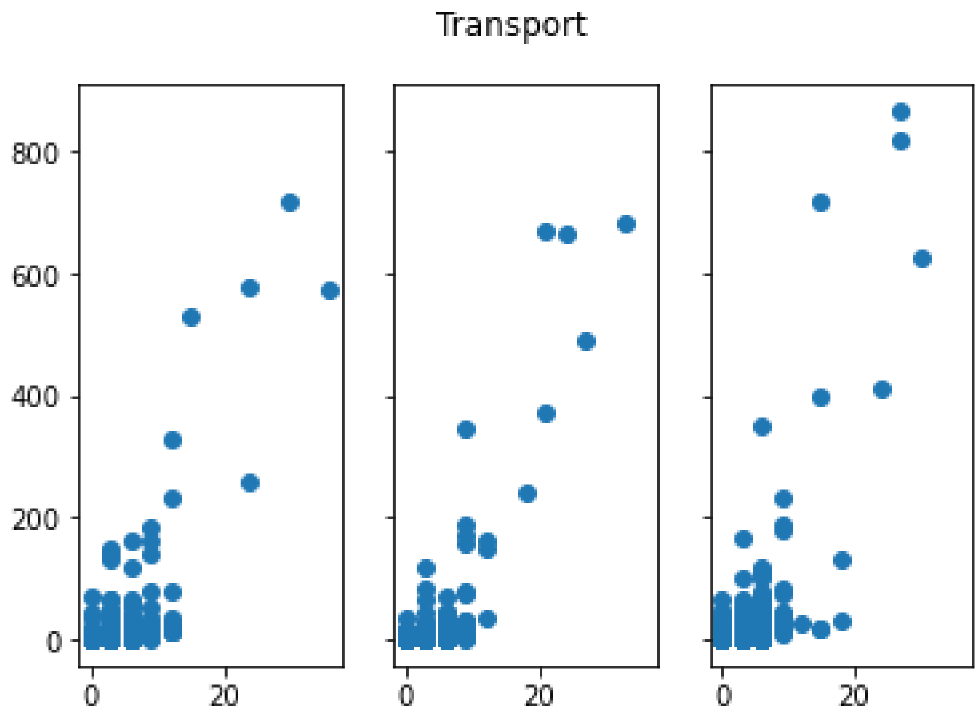
[184]:
```
groupA['Transport_GeogUnits']=groupA['I461_GeogUnits']+groupA['I471_GeogUnits']+groupA['I481_G
groupA['Transport_EmpCo']=groupA['I461_EmpCo']+groupA['I471_EmpCo']+groupA['I481_EmpCo']
```

1.5.1 Scatterplot

[185]:
```
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())
```

[186]:
```
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)
```

[186]:
```
<matplotlib.collections.PathCollection at 0x11f041af0>
```



1.5.2 Correlation tests

[187]:
```
# 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
[[  13.82492115  136.89676239]
 [ 136.89676239 3308.8738962 ]]
Pearsons correlation: 0.640
Spearmans correlation: 0.601
2013
[[  12.02336239  149.74588661]
 [ 149.74588661 3638.54750542]]
Pearsons correlation: 0.716
Spearmans correlation: 0.617
2018
[[  12.66979134  169.65592309]
 [ 169.65592309 5253.82375808]]
Pearsons correlation: 0.658
Spearmans correlation: 0.608
```

1.5.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.5.4 Create model

[188]:
```
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))
```

[189]:
```
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)
```

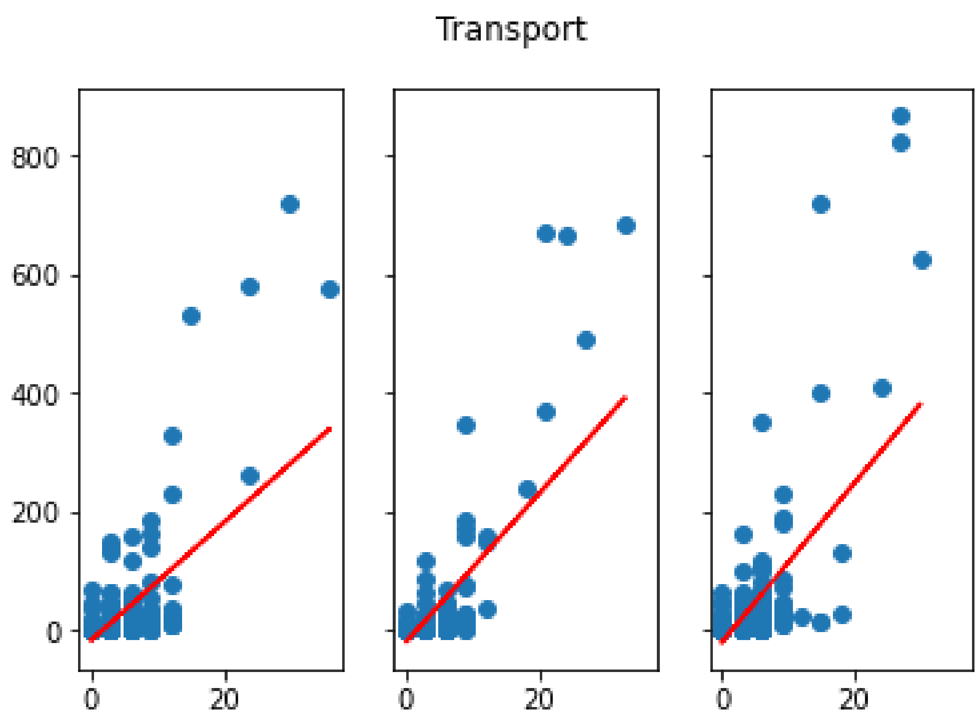
[190]:
```
print("2006")
r_sq = model_2006.score(trans_Fac_2006, trans_Emp_2006)
print('coefficient of determination:', r_sq)
print('intercept:', model_2006.intercept_)
print('slope:', model_2006.coef_)


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: 0.40967878547346004
intercept: -17.378154113309407
slope: [9.90217311]
2013
coefficient of determination: 0.5125731070043044
intercept: -19.649362662528173
slope: [12.4545765]
2018
coefficient of determination: 0.43240737271797314
intercept: -20.9623952117318
slope: [13.39058541]
```

[191]:
```
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')
```

[191]: [<matplotlib.lines.Line2D at 0x11e3f1190>]



1.6 Post and storage

[192]:
```
groupA['Storage_GeogUnits']=groupA['I51_GeogUnits']+groupA['I53_GeogUnits']
groupA['Storage_EmpCo']=groupA['I51_EmpCo']+groupA['I53_EmpCo']
```

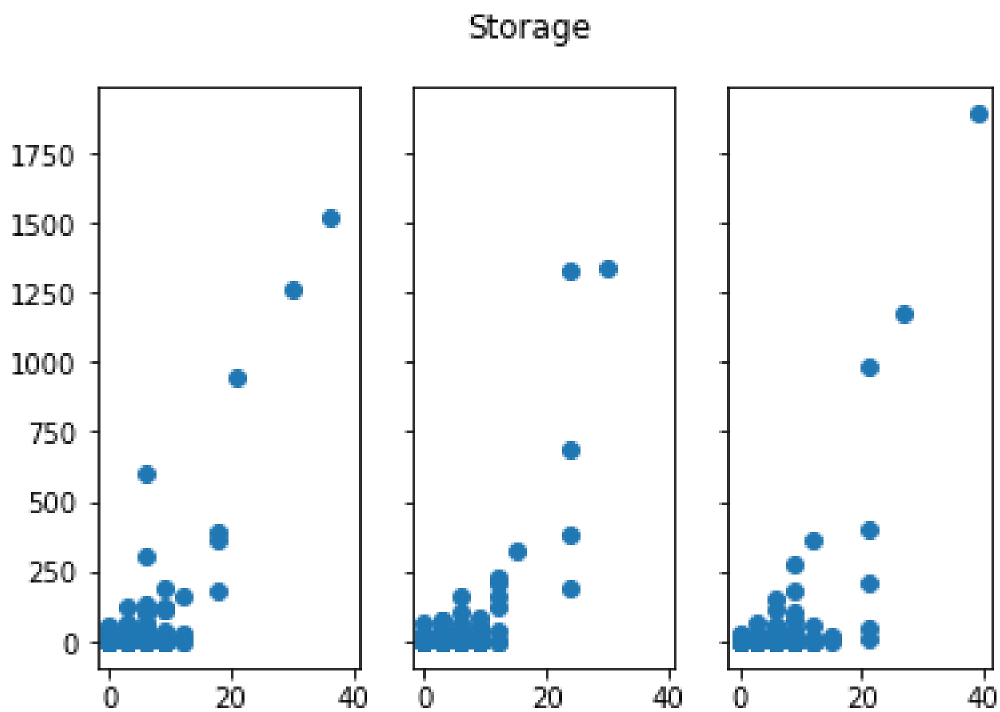
1.6.1 Scatterplot

[193]:
```
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())
```

[194]:
```
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)
```

[194]: <matplotlib.collections.PathCollection at 0x11f2a7ee0>



1.6.2 Correlation tests

[195]:
```
# 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
[[   12.38176267   227.71738526]
 [  227.71738526 10066.25200533]]
Pearsons correlation: 0.645
Spearmans correlation: 0.333
2013
[[  12.82862525  188.81458948]
 [ 188.81458948 7934.97351504]]
Pearsons correlation: 0.592
Spearmans correlation: 0.425
2018
[[   15.96479839   243.90210679]
 [  243.90210679 11302.88724613]]
Pearsons correlation: 0.574
Spearmans correlation: 0.484
```

1.6.3 Linear regression

regressor - # Facilities ; predictor - employee count

1.6.4 Create model

[196]:
```
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))
```

[197]:
```
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)
```

[198]:
```
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: 0.4160467203225553
intercept: -33.000365268690715
slope: [18.39135439]
2013
coefficient of determination: 0.3502236720462094
intercept: -27.27142432694909
slope: [14.71822474]
2018
coefficient of determination: 0.329669120612715
intercept: -32.54443023044497
slope: [15.2774937]
```

[199]:

```
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')
```

[199]:

```
[<matplotlib.lines.Line2D at 0x11f4931f0>]
```