Import some libraries. Remember, we're about to using Spearman and Chi-Square to testing non-parameteric correlation data, so import spearmanr and chi2_contingency from scipy.stats

In [29]:
```
import numpy as np
import seaborn as sb

import pandas as pd
from pandas import Series, DataFrame

import matplotlib.pyplot as plt
from pylab import rcParams

import scipy
from scipy.stats import spearmanr
from scipy.stats import chi2_contingency
```

Configuration canvas

In [2]:
```
%matplotlib inline
rcParams['figure.figsize'] =  5,4
sb.set_style('whitegrid')
```

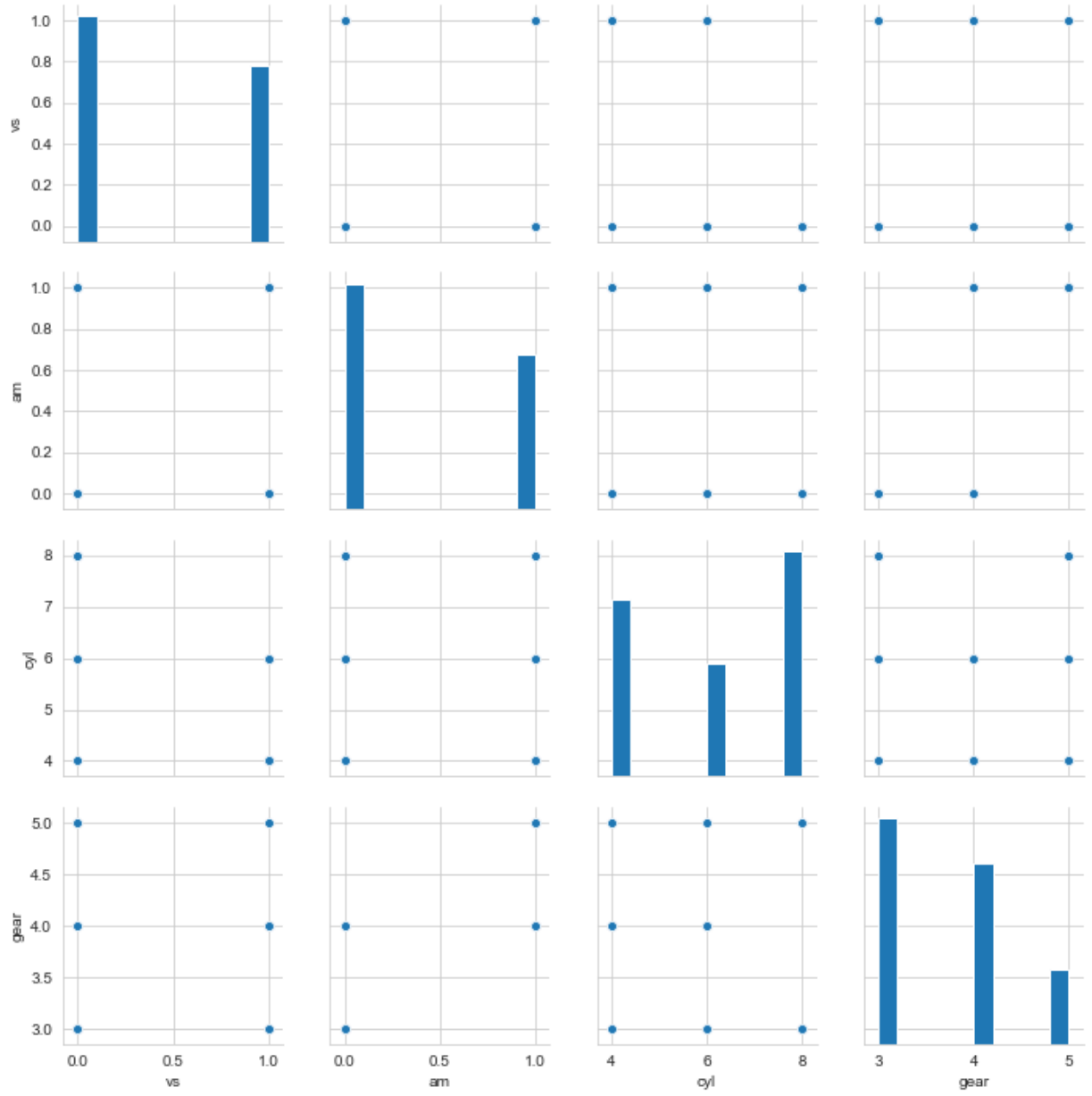
Read csv document from local computer

In [3]:
```
address = 'C:/Users/muham/mtcars.csv'
cars = pd.read_csv(address)
cars.columns = ['car_names', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', '
```

We're about to see the graphic data using pairplot from Seaborn to analysis some not normal data distribution from vs, am, cyl and gear

In [4]:
```
x = cars[['vs', 'am', 'cyl', 'gear']]
sb.pairplot(x)
```

Out[4]: <seaborn.axisgrid.PairGrid at 0x3056130>



Right now, we're about to count the Spearman Rank Coefficient to see how strong correlation between those data

In [47]:
```
cyl = cars['cyl']
vs = cars['vs']
am = cars['am']
gear = cars['gear']

spearmanr_coefficient, p_value = spearmanr(cyl, vs)
print ('Spearman Coefficient between Cyl and Vs is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Cyl and Vs is
-0.814
```

In [46]:
```
spearmanr_coefficient, p_value = spearmanr(cyl, am)
print ('Spearman Coefficient between Cyl and Am is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Cyl and Am is
-0.522
```

In [45]:
```
spearmanr_coefficient, p_value = spearmanr(cyl, gear)
print ('Spearman Coefficient between Cyl and Gear is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Cyl and Gear is
-0.564
```

In [44]:
```
spearmanr_coefficient, p_value = spearmanr(vs, am)
print ('Spearman Coefficient between Vs and Am is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Vs and Am is
0.168
```

In [43]:
```
spearmanr_coefficient, p_value = spearmanr(vs, gear)
print ('Spearman Coefficient between Vs and Gear is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Vs and Gear is
0.283
```

In [42]:
```
spearmanr_coefficient, p_value = spearmanr(am, gear)
print ('Spearman Coefficient between Am and Gear is')
print ('%.3f' %spearmanr_coefficient)
```

```
Spearman Coefficient between Am and Gear is
0.808
```

Right now, we're about to count the Chi-2 Square to see how strong data to be independence or not. We need p>0.05 to conclude that data is strongly independence

In [40]:
```
table = pd.crosstab(cyl,am)

chi2, p, dof, expected = chi2_contingency(table.values)
print ('Chi-Square Test for independency data between Cyl and Am')
print('%.3f' % p)
```

```
Chi-Square Test for independency data between Cyl and Am
0.013
```

In [39]:
```
table = pd.crosstab(cyl,vs)

chi2, p, dof, expected = chi2_contingency(table.values)
print ('Chi-Square Test for independency data between Cyl and Am')
print('%.3f' % p)
```

```
Chi-Square Test for independency data between Cyl and Am
0.000
```

In [38]:
```
table = pd.crosstab(cyl,gear)

chi2, p, dof, expected = chi2_contingency(table.values)
print ('Chi-Square Test for independency data between Cyl and Am')
print('%.3f' % p)
```

```
Chi-Square Test for independency data between Cyl and Am
0.001
```