

In [131]:

```
import numpy as np
```

In [132]:

```
import pandas as pd
```

In [133]:

```
import matplotlib.pyplot as plt
```

In [134]:

```
import seaborn as sns
```

In [135]:

```
%matplotlib inline
```

In [136]:

```
mt=pd.read_csv('C:/Users/user/Downloads/datasets_596958_1073629_Placement_Data_Full_Class.csv')
```

In [137]:

```
mt
```

Out[137]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed
...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed

215 rows × 15 columns



In [138]:

```
rt=mt.drop(['sl_no','salary'],axis=1)
```

In [139]:

```
rt
```

Out[139]:

gender ssc_p ssc_b hsc_p hsc_b hsc_s degree_p degree_t workex etest_p specialisation mba_p status

0	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed
...
210	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed
211	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed
212	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed
213	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed
214	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed

215 rows × 13 columns

In [140]:

```
rt.columns
```

Out[140]:

```
Index(['gender', 'ssc_p', 'ssc_b', 'hsc_p', 'hsc_b', 'hsc_s', 'degree_p',
       'degree_t', 'workex', 'etest_p', 'specialisation', 'mba_p', 'status'],
      dtype='object')
```

In [141]:

```
rt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   gender          215 non-null   object
1   ssc_p           215 non-null   float64
2   ssc_b           215 non-null   object
3   hsc_p           215 non-null   float64
4   hsc_b           215 non-null   object
5   hsc_s           215 non-null   object
6   degree_p        215 non-null   float64
7   degree_t        215 non-null   object
8   workex          215 non-null   object
9   etest_p         215 non-null   float64
10  specialisation  215 non-null   object
11  mba_p           215 non-null   float64
12  status          215 non-null   object
dtypes: float64(5), object(8)
memory usage: 22.0+ KB
```

Data information and Visualisation

In [142]:

```
rt.describe()
```

Out[142]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p
count	215.000000	215.000000	215.000000	215.000000	215.000000
mean	67.303395	66.333163	66.370186	72.100558	62.278186
std	10.827205	10.897509	7.358743	13.275956	5.833385
min	40.890000	37.000000	50.000000	50.000000	51.210000

	ssc_p	hsc_p	degree_p	etest_p	mba_p
25%	60.600000	60.900000	61.900000	60.000000	57.945000
50%	67.000000	65.000000	66.000000	71.000000	62.000000
75%	75.700000	73.000000	72.000000	83.500000	66.255000
max	89.400000	97.700000	91.000000	98.000000	77.890000

In [143]:

```
rt.isnull().sum()
```

Out[143]:

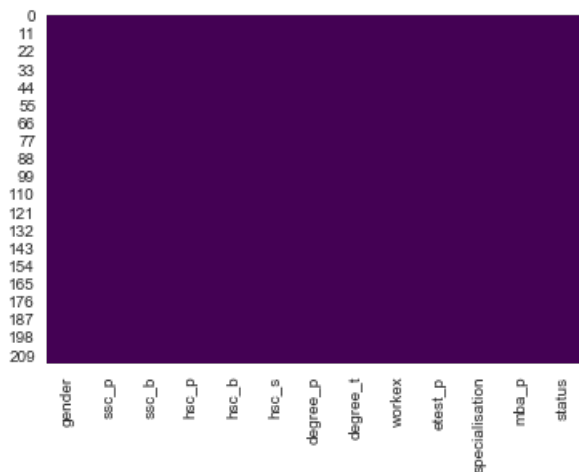
```
gender          0
ssc_p           0
ssc_b           0
hsc_p           0
hsc_b           0
hsc_s           0
degree_p        0
degree_t        0
workex          0
etest_p         0
specialisation  0
mba_p           0
status          0
dtype: int64
```

In [144]:

```
sns.heatmap(rt.isnull(), cbar=False, cmap='viridis')
```

Out[144]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be7d676ec8>

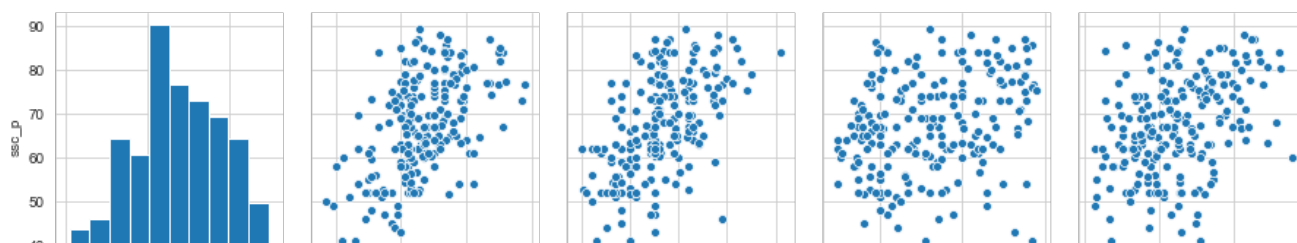


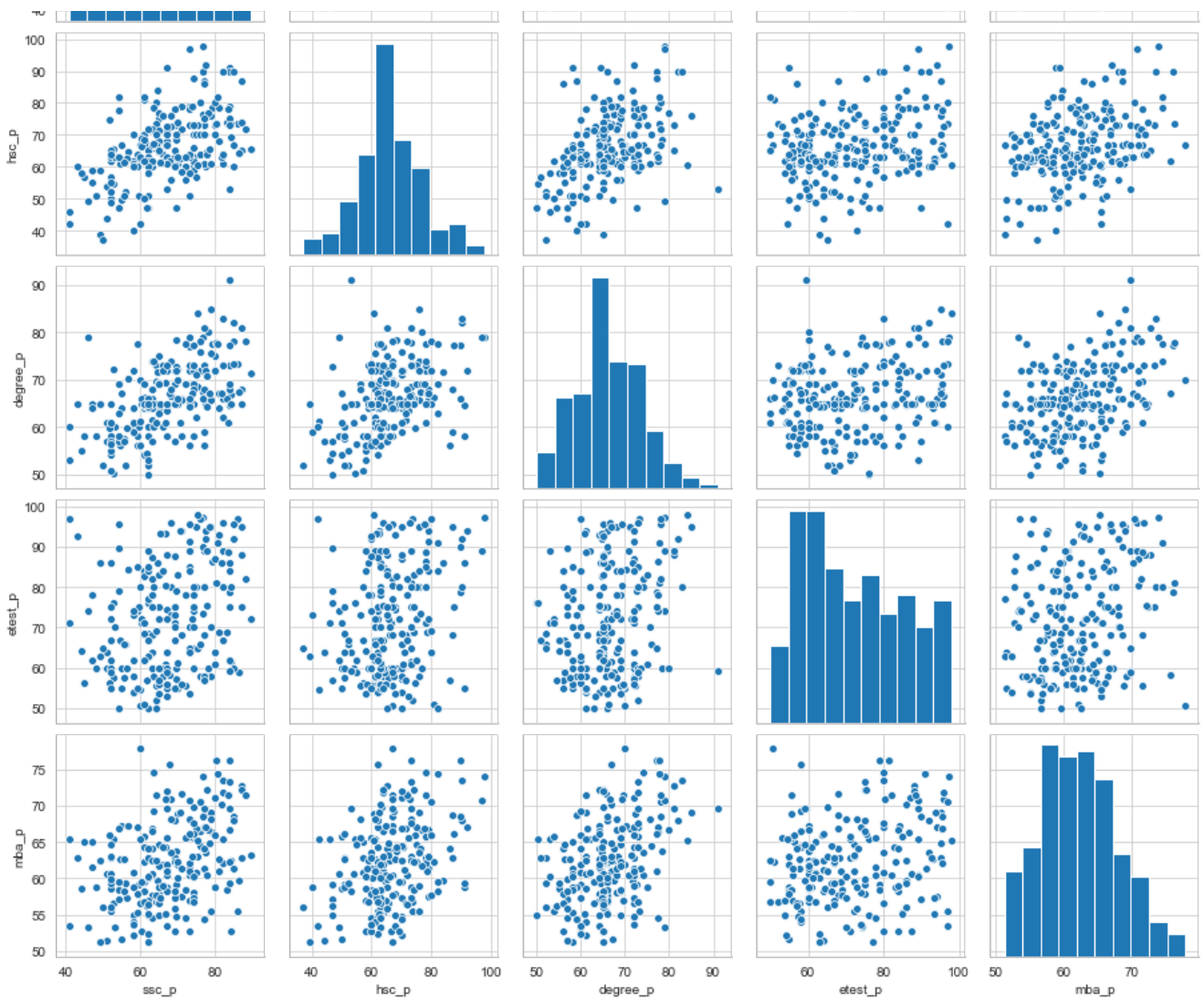
In [145]:

```
sns.pairplot(rt)
```

Out[145]:

<seaborn.axisgrid.PairGrid at 0x1be0194c6c8>





In [146]:

```
rt.corr()
```

Out[146]:

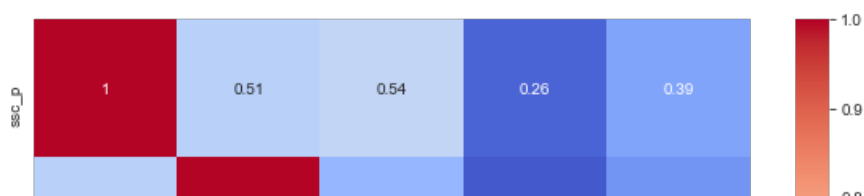
	ssc_p	hsc_p	degree_p	etest_p	mba_p
ssc_p	1.000000	0.511472	0.538404	0.261993	0.388478
hsc_p	0.511472	1.000000	0.434206	0.245113	0.354823
degree_p	0.538404	0.434206	1.000000	0.224470	0.402364
etest_p	0.261993	0.245113	0.224470	1.000000	0.218055
mba_p	0.388478	0.354823	0.402364	0.218055	1.000000

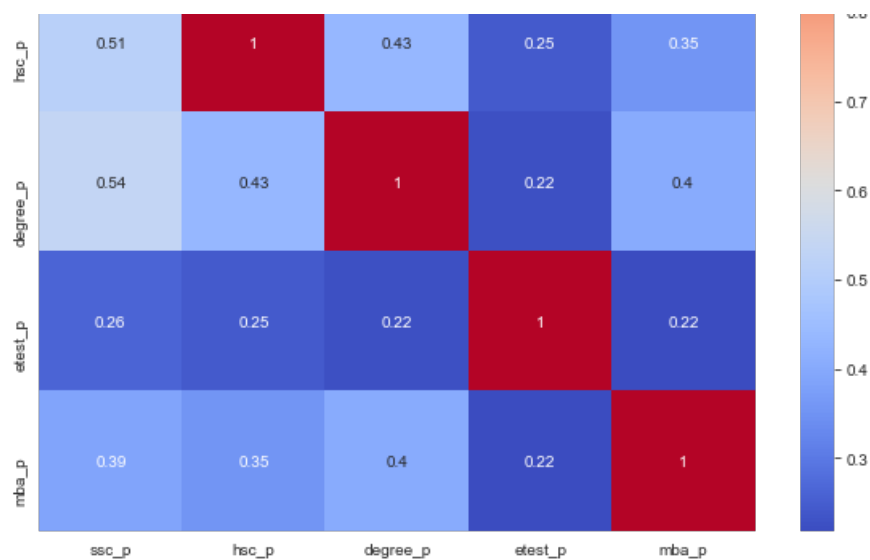
In [147]:

```
plt.figure(figsize=(10,8))
sns.heatmap(rt.corr(),cmap='coolwarm',annot=True)
```

Out[147]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be02542448>



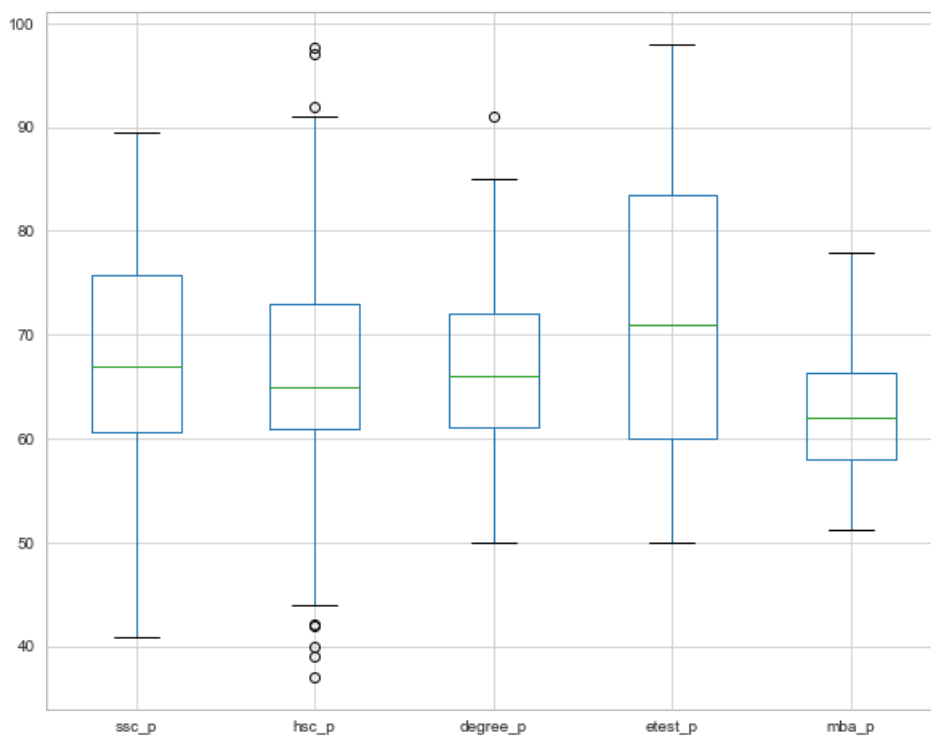


In [148]:

```
plt.figure(figsize=(10,8))
rt.boxplot()
```

Out[148]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03aa5b08>

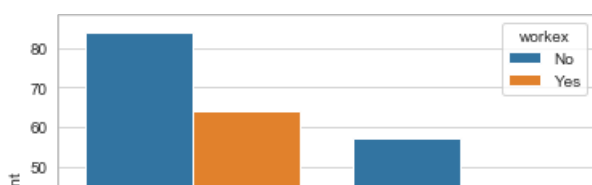


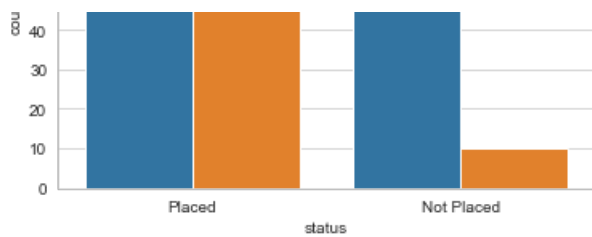
In [149]:

```
sns.countplot(x='status', data=rt, hue='workex')
```

Out[149]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be0399bfc8>



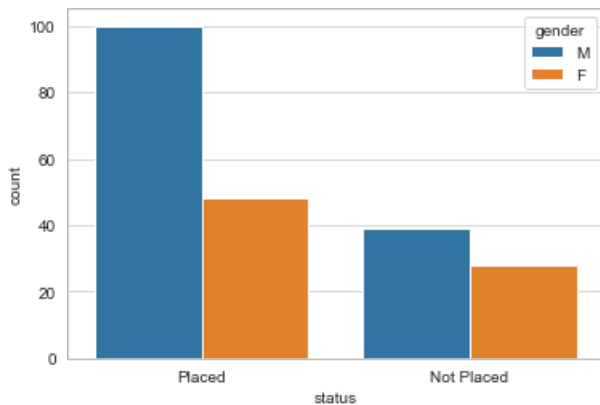


In [150]:

```
sns.countplot(x='status',data=rt,hue='gender')
```

Out[150]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be039f33c8>



In [151]:

```
rt.columns
```

Out[151]:

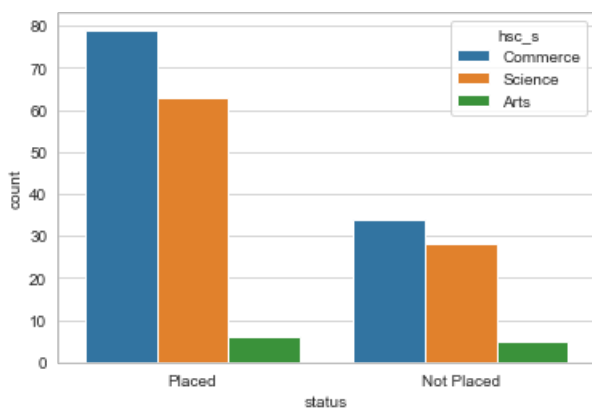
```
Index(['gender', 'ssc_p', 'ssc_b', 'hsc_p', 'hsc_b', 'hsc_s', 'degree_p',  
      'degree_t', 'workex', 'etest_p', 'specialisation', 'mba_p', 'status'],  
      dtype='object')
```

In [152]:

```
sns.countplot(x='status',data=rt,hue='hsc_s')
```

Out[152]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03acdec8>

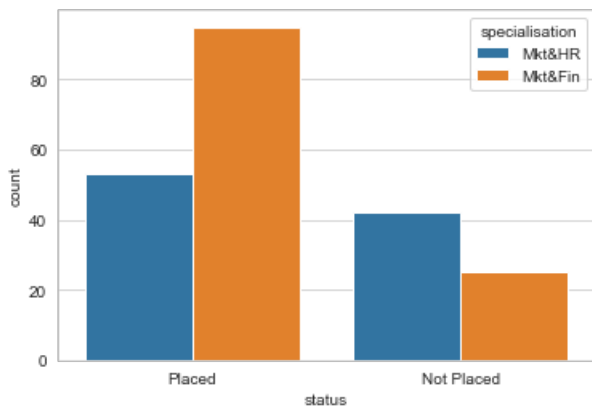


In [153]:

```
sns.countplot(x='status',data=rt,hue='specialisation')
```

Out[153]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03b36488>

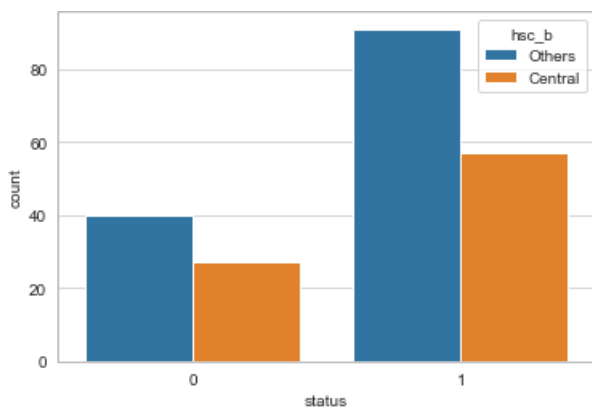


In [171]:

```
sns.countplot(x='status', data=rt, hue='hsc_b')
```

Out[171]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03ee42c8>

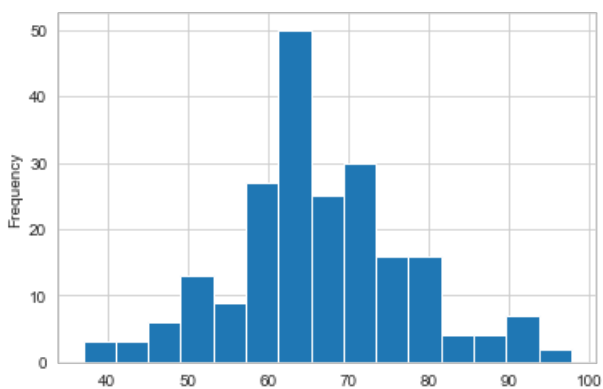


In [154]:

```
rt['hsc_p'].plot.hist(bins=15)
```

Out[154]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03b98bc8>



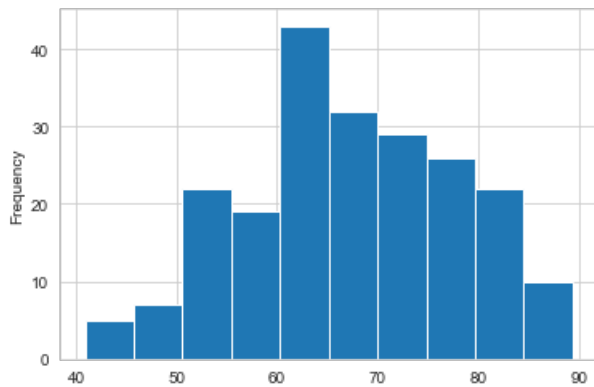
In [155]:

```
rt['ssc_p'].plot.hist(bins=10)
```

```
rt['ssc_p'].plot.hist(bins=10)
```

Out[155]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03c2b4c8>

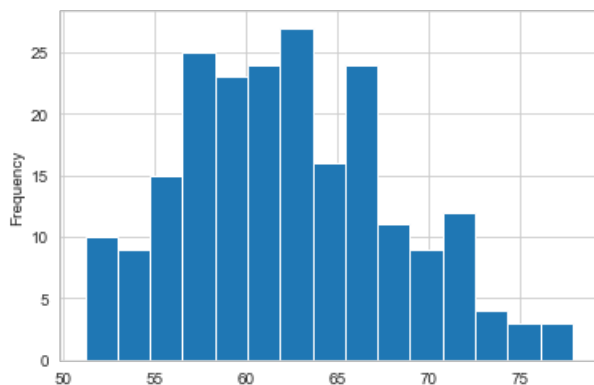


In [156]:

```
rt['mba_p'].plot.hist(bins=15)
```

Out[156]:

<matplotlib.axes._subplots.AxesSubplot at 0x1be03dd1688>



In []:

To check whether the student placed or not

we use 1 for placed and 0 for non placed students and the we use logistic regression model to solve the problem

Data cleaning

In [161]:

```
rt['status'].replace('Not Placed',0,inplace=True)
```

In [162]:

```
rt
```

Out[162]:

gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
--------	-------	-------	-------	-------	-------	----------	----------	--------	---------	----------------	-------	--------

0	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	Status
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	0
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed
...
210	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed
211	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed
212	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed
213	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed
214	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	0

215 rows × 13 columns

In [163]:

```
rt['status'].replace('Placed',1,inplace=True)
```

In [164]:

```
rt
```

Out[164]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
0	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	1
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	1
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	1
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	0
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	1
...
210	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	1
211	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	1
212	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	1
213	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	1
214	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	0

215 rows × 13 columns

In [174]:

```
gender=pd.get_dummies(rt['gender'],drop_first=True)
```

In [177]:

```
workex=pd.get_dummies(rt['workex'],drop_first=True)
```

In [190]:

```
ssc_b=pd.get_dummies(rt['ssc_b'],drop_first=True)
```

In [191]:

```
hsc_b=pd.get_dummies(rt['hsc_b'],drop_first=True)
```

In [195]:

```
rt=pd.concat([rt,gender,workex,ssc_b,hsc_b],axis=1)
```

In [196]:

```
rt
```

Out[196]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	M	Y
0	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	1	1	
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	1	1	
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	1	1	
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	0	1	
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	1	1	
...
210	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	1	1	
211	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	1	1	
212	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	1	1	
213	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	1	0	
214	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	0	1	

215 rows × 17 columns

Now we drop some columns of our data after getting their dummies

In [206]:

```
rt
```

Out[206]:

	ssc_p	hsc_p	hsc_s	degree_p	degree_t	etest_p	specialisation	mba_p	status	M	Yes	Others	Others
0	67.00	91.00	Commerce	58.00	Sci&Tech	55.0	Mkt&HR	58.80	1	1	0	1	1
1	79.33	78.33	Science	77.48	Sci&Tech	86.5	Mkt&Fin	66.28	1	1	1	0	1
2	65.00	68.00	Arts	64.00	Comm&Mgmt	75.0	Mkt&Fin	57.80	1	1	0	0	0
3	56.00	52.00	Science	52.00	Sci&Tech	66.0	Mkt&HR	59.43	0	1	0	0	0
4	85.80	73.60	Commerce	73.30	Comm&Mgmt	96.8	Mkt&Fin	55.50	1	1	0	0	0
...
210	80.60	82.00	Commerce	77.60	Comm&Mgmt	91.0	Mkt&Fin	74.49	1	1	0	1	1
211	58.00	60.00	Science	72.00	Sci&Tech	74.0	Mkt&Fin	53.62	1	1	0	1	1
212	67.00	67.00	Commerce	73.00	Comm&Mgmt	59.0	Mkt&Fin	69.72	1	1	1	1	1
213	74.00	66.00	Commerce	58.00	Comm&Mgmt	70.0	Mkt&HR	60.23	1	0	0	1	1
214	62.00	58.00	Science	53.00	Comm&Mgmt	89.0	Mkt&HR	60.22	0	1	0	0	1

215 rows × 13 columns

In [207]:

```
rt.drop(['degree_t','hsc_s'],axis=1,inplace=True)
```

In [209]:

```
rt.drop(['specialisation'],axis=1,inplace=True)
```

In [210]:

```
rt
```

Out[210]:

	ssc_p	hsc_p	degree_p	etest_p	mba_p	status	M	Yes	Others	Others
0	67.00	91.00	58.00	55.0	58.80	1	1	0	1	1
1	79.33	78.33	77.48	86.5	66.28	1	1	1	0	1
2	65.00	68.00	64.00	75.0	57.80	1	1	0	0	0
3	56.00	52.00	52.00	66.0	59.43	0	1	0	0	0
4	85.80	73.60	73.30	96.8	55.50	1	1	0	0	0
...
210	80.60	82.00	77.60	91.0	74.49	1	1	0	1	1
211	58.00	60.00	72.00	74.0	53.62	1	1	0	1	1
212	67.00	67.00	73.00	59.0	69.72	1	1	1	1	1
213	74.00	66.00	58.00	70.0	60.23	1	0	0	1	1
214	62.00	58.00	53.00	89.0	60.22	0	1	0	0	1

215 rows × 10 columns

Training Models

In [211]:

```
x=rt.drop(['status'],axis=1)
```

In [213]:

```
y=rt['status']
```

In [216]:

```
from sklearn.model_selection import train_test_split
```

In [256]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=101)
```

In [257]:

```
from sklearn.linear_model import LogisticRegression
```

In [258]:

```
logmodel=LogisticRegression()
```

In [259]:

```
logmodel.fit(X_train,y_train)
```

Out[259]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, l1_ratio=None, max_iter=100,
                    multi_class='auto', n_jobs=None, penalty='l2',
                    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False)
```

In [260]:

```
predictions=logmodel.predict(X_test)
```

In [261]:

```
predictions
```

Out[261]:

```
array([1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1],
      dtype=int64)
```

In [262]:

```
y_test
```

Out[262]:

```
151    1
66     1
117    1
96     1
109    0
..
61     1
4      1
15     1
1      1
43     1
Name: status, Length: 65, dtype: int64
```

checking model accuracy

In [263]:

```
from sklearn.metrics import classification_report
```

In [264]:

```
print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.84	0.80	0.82	20
1	0.91	0.93	0.92	45
accuracy			0.89	65
macro avg	0.88	0.87	0.87	65
weighted avg	0.89	0.89	0.89	65

In [265]:

```
from sklearn.metrics import confusion_matrix
```

In [266]:

```
confusion_matrix(y_test,predictions)
```

Out[266]:

```
array([[16,  4],
       [ 3, 42]], dtype=int64)
```

In []:

