OASIS - EDA and ML Prediction

Data and Library Setup

In [2]:

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
import os
print(os.listdir("F:/Dementia Prediction/data"))
```

```
['oasis_cross-sectional.csv', 'oasis_longitudinal.csv']
```

In [4]:

```
df = pd.read_csv('F:/Dementia Prediction/data/oasis_longitudinal.csv')
```

Exploratory Data Analysis

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 373 entries, 0 to 372
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Subject ID  373 non-null    object 
 1   MRI ID      373 non-null    object 
 2   Group       373 non-null    object 
 3   Visit       373 non-null    int64  
 4   MR Delay    373 non-null    int64  
 5   M/F         373 non-null    object 
 6   Hand        373 non-null    object 
 7   Age         373 non-null    int64  
 8   EDUC        373 non-null    int64  
 9   SES         354 non-null    float64
 10  MMSE        371 non-null    float64
 11  CDR         373 non-null    float64
 12  eTIV        373 non-null    int64  
 13  nWBV        373 non-null    float64
 14  ASF         373 non-null    float64
dtypes: float64(5), int64(5), object(5)
memory usage: 43.8+ KB
```

In [6]:

```
print("Tota Rows and Columns (Rows,Columns) : ",df.shape)
#print first five rows of the dataset
df.head(5)
```

```
Tota Rows and Columns (Rows,Columns) :  (373, 15)
```

	Subject ID	MRI ID	Group	Visit	MR Delay	M/F	Hand	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
0	OAS2_0001	OAS2_0001_MR1	Nondemented	1	0	M	R	87	14	2.0	27.0	0.0	1987	0.696	0.883
1	OAS2_0001	OAS2_0001_MR2	Nondemented	2	457	M	R	88	14	2.0	30.0	0.0	2004	0.681	0.876
2	OAS2_0002	OAS2_0002_MR1	Demented	1	0	M	R	75	12	NaN	23.0	0.5	1678	0.736	1.046
3	OAS2_0002	OAS2_0002_MR2	Demented	2	560	M	R	76	12	NaN	28.0	0.5	1738	0.713	1.010
4	OAS2_0002	OAS2_0002_MR3	Demented	3	1895	M	R	80	12	NaN	22.0	0.5	1698	0.701	1.034

In [7]:
```
df.describe()
```

	Visit	MR Delay	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
count	373.000000	373.000000	373.000000	373.000000	354.000000	371.000000	373.000000	373.000000	373.000000	373.000000
mean	1.882038	595.104558	77.013405	14.597855	2.460452	27.342318	0.290885	1488.128686	0.729568	1.195461
std	0.922843	635.485118	7.640957	2.876339	1.134005	3.683244	0.374557	176.139286	0.037135	0.138092
min	1.000000	0.000000	60.000000	6.000000	1.000000	4.000000	0.000000	1106.000000	0.644000	0.876000
25%	1.000000	0.000000	71.000000	12.000000	2.000000	27.000000	0.000000	1357.000000	0.700000	1.099000
50%	2.000000	552.000000	77.000000	15.000000	2.000000	29.000000	0.000000	1470.000000	0.729000	1.194000
75%	2.000000	873.000000	82.000000	16.000000	3.000000	30.000000	0.500000	1597.000000	0.756000	1.293000
max	5.000000	2639.000000	98.000000	23.000000	5.000000	30.000000	2.000000	2004.000000	0.837000	1.587000

In [8]:
```
df.isna().sum()
```

```
Subject ID     0
MRI ID         0
Group          0
Visit          0
MR Delay       0
M/F            0
Hand           0
Age            0
EDUC           0
SES           19
MMSE           2
CDR            0
eTIV           0
nWBV           0
ASF            0
dtype: int64
```

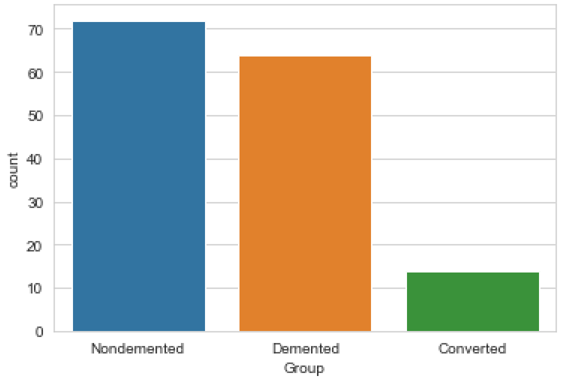
In [9]:
```
sum(df.duplicated())
```

```
0
```

In [10]:
```
df["SES"].fillna(df["SES"].median(), inplace=True)
df["MMSE"].fillna(df["MMSE"].mean(), inplace=True)
```

In [11]:
```
sns.set_style("whitegrid")
ex_df = df.loc[df['Visit'] == 1]
sns.countplot(x='Group', data=ex_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2551d4d8ca0>
```

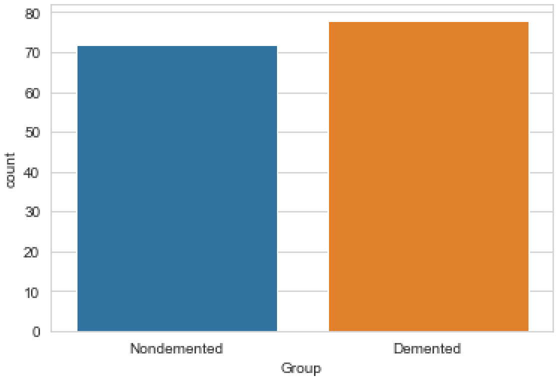


In [12]:

```
ex_df['Group'] = ex_df['Group'].replace(['Converted'], ['Demented'])
df['Group'] = df['Group'].replace(['Converted'], ['Demented'])
sns.countplot(x='Group', data=ex_df)
```

```
<ipython-input-12-f052a051643f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  ex_df['Group'] = ex_df['Group'].replace(['Converted'], ['Demented'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2551d4fae80>
```

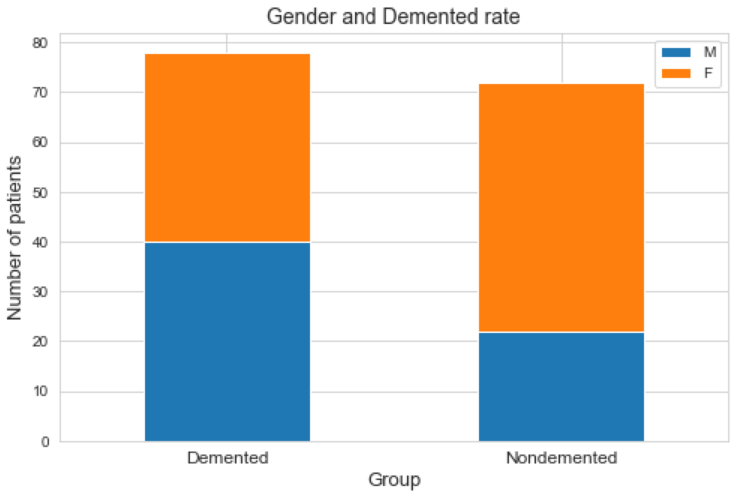


In [13]:

```
def bar_chart(feature):
    Demented = ex_df[ex_df['Group']=='Demented'][feature].value_counts()
    Nondemented = ex_df[ex_df['Group']=='Nondemented'][feature].value_counts()
    df_bar = pd.DataFrame([Demented,Nondemented])
    df_bar.index = ['Demented','Nondemented']
    df_bar.plot(kind='bar',stacked=True, figsize=(8,5))
    print(df_bar)


# Gender  and  Group ( Female=0, Male=1)
bar_chart('M/F')
plt.xlabel('Group',fontsize=13)
plt.xticks(rotation=0,fontsize=12)
plt.ylabel('Number of patients',fontsize=13)
plt.legend()
plt.title('Gender and Demented rate',fontsize=14)
```

```
              M   F
Demented     40  38
Nondemented  22  50
```

```
Text(0.5, 1.0, 'Gender and Demented rate')
```

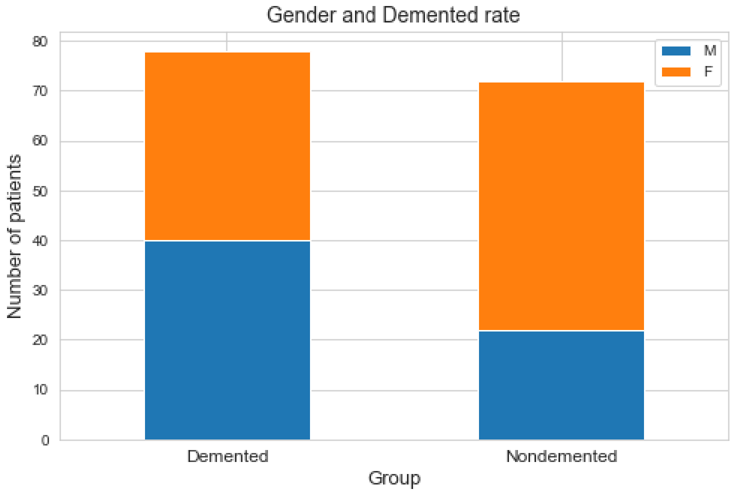


In [14]:

```
def bar_chart(feature):
    Demented = ex_df[ex_df['Group']=='Demented'][feature].value_counts()
    Nondemented = ex_df[ex_df['Group']=='Nondemented'][feature].value_counts()
    df_bar = pd.DataFrame([Demented,Nondemented])
    df_bar.index = ['Demented','Nondemented']
    df_bar.plot(kind='bar',stacked=True, figsize=(8,5))
    print(df_bar)


# Gender  and  Group ( Female=0, Male=1)
bar_chart('M/F')
plt.xlabel('Group',fontsize=13)
plt.xticks(rotation=0,fontsize=12)
plt.ylabel('Number of patients',fontsize=13)
plt.legend()
plt.title('Gender and Demented rate',fontsize=14)
```

```
              M   F
Demented     40  38
Nondemented  22  50
```

```
Text(0.5, 1.0, 'Gender and Demented rate')
```

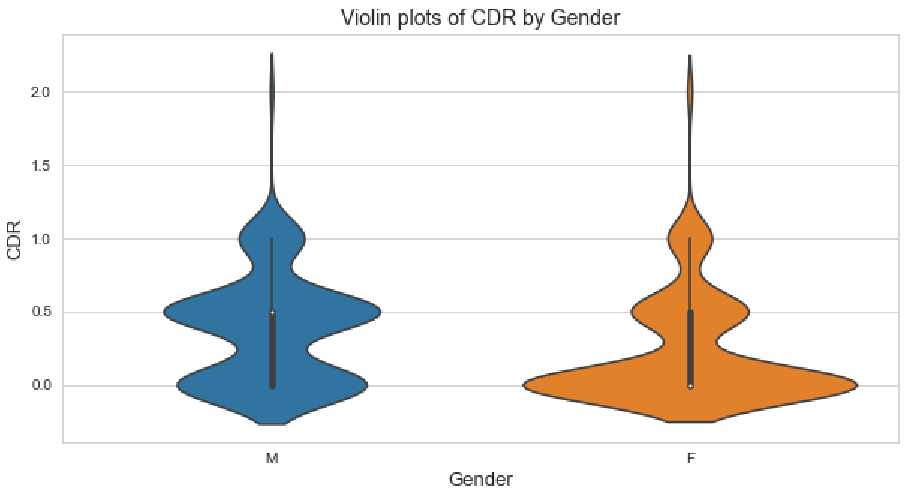


CDR (Clinical Dementia Rating) : Ratings are assigned on a 0–5 point scale, (0 = absent; 0.5 = questionable; 1= present, but mild; 2 = moderate; 3 = severe; 4 = profound; 5 = terminal). A global summary score is obtained, leading to the use of the CDR for grouping patients on severity of dementia.

CDR By Gender

In [15]:

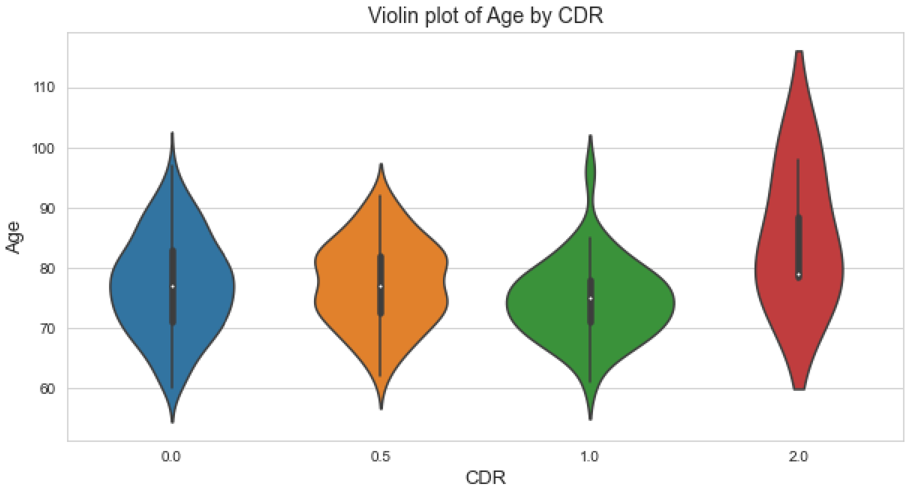
```
plt.figure(figsize=(10,5))
sns.violinplot(x='M/F', y='CDR', data=df)
plt.title('Violin plots of CDR by Gender',fontsize=14)
plt.xlabel('Gender',fontsize=13)
plt.ylabel('CDR',fontsize=13)
plt.show()
```



CDR By Age

In [16]:

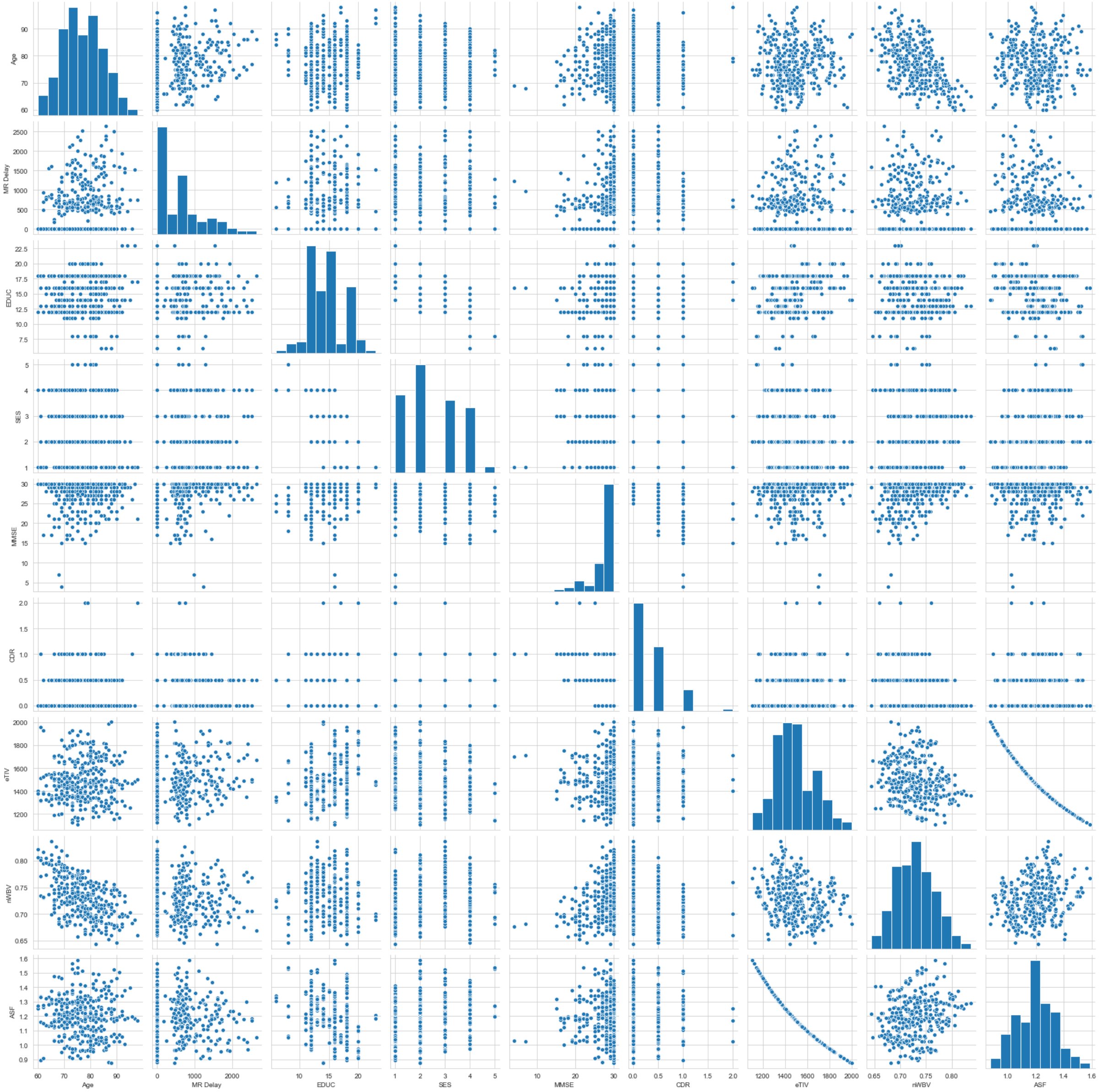
```
plt.figure(figsize=(10,5))
sns.violinplot(x='CDR', y='Age', data=df)
plt.title('Violin plot of Age by CDR',fontsize=14)
plt.xlabel('CDR',fontsize=13)
plt.ylabel('Age',fontsize=13)
plt.show()
```



In [17]:
```
def outliers_iqr(ys):
    quartile_1, quartile_3 = np.percentile(ys, [25, 75])
    iqr = quartile_3 - quartile_1
    lower_bound = quartile_1 - (iqr * 1.5)
    upper_bound = quartile_3 + (iqr * 1.5)
    return np.where((ys > upper_bound) | (ys < lower_bound))

list_atributes = ['MR Delay','EDUC', "SES", "MMSE", 'eTIV', "nWBV", "ASF"]
print("Outliers: \n")
for item in list_atributes:
    print(item,': ',outliers_iqr(df[item]))
```

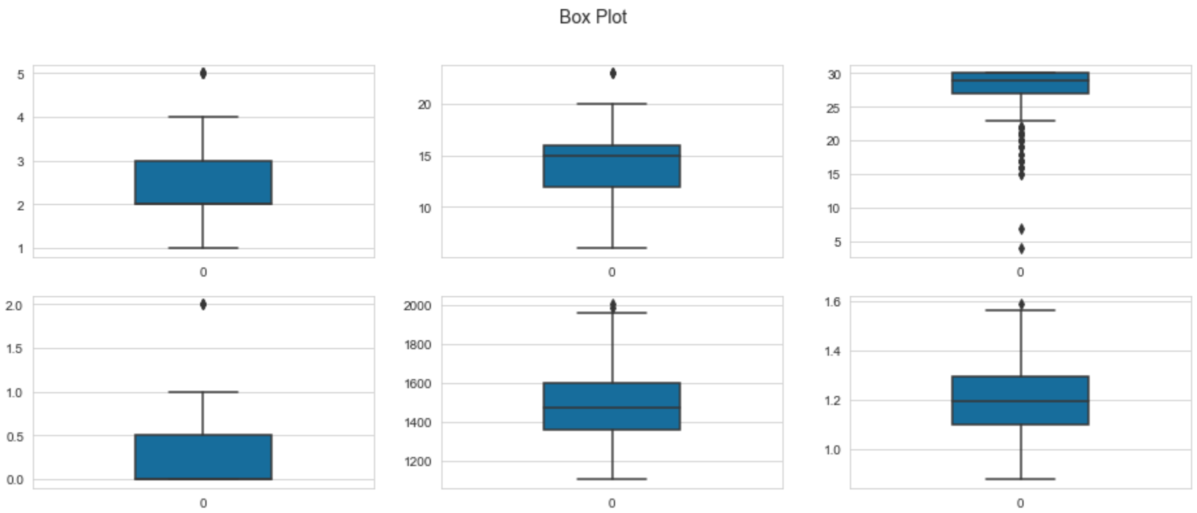
```
Outliers: 

MR Delay :  (array([ 32,  71,  75, 153, 159, 160, 265, 369], dtype=int64),)
EDUC :  (array([107, 108, 109], dtype=int64),)
SES :  (array([136, 137, 138, 161, 162, 179, 180], dtype=int64),)
MMSE :  (array([  4,  25,  26,  43,  44,  51,  52,  60,  88,  89,  90,  93,  94,
        97,  98,  99, 100, 101, 105, 106, 138, 162, 172, 173, 184, 185,
       186, 222, 225, 226, 231, 232, 234, 251, 299, 300, 316, 317, 328,
       332, 360, 366], dtype=int64),)
eTIV :  (array([0, 1], dtype=int64),)
nWBV :  (array([], dtype=int64),)
ASF :  (array([282], dtype=int64),)
```

```
In [18]: from pylab import rcParams
         rcParams['figure.figsize'] = 8,5
         cols = ['Age','MR Delay', 'EDUC', 'SES', 'MMSE', 'CDR','eTIV','nWBV','ASF']
         x=df.fillna('')
         sns_plot = sns.pairplot(x[cols])
```



In [19]:

```
fig, axes = plt.subplots(2,3,figsize = (16,6))
fig.suptitle("Box Plot",fontsize=14)
sns.set_style("whitegrid")
sns.boxplot(data=df['SES'], orient="v",width=0.4, palette="colorblind",ax = axes[0][0]);
sns.boxplot(data=df['EDUC'], orient="v",width=0.4, palette="colorblind",ax = axes[0][1]);
sns.boxplot(data=df['MMSE'], orient="v",width=0.4, palette="colorblind",ax = axes[0][2]);
sns.boxplot(data=df['CDR'], orient="v",width=0.4, palette="colorblind",ax = axes[1][0]);
sns.boxplot(data=df['eTIV'], orient="v",width=0.4, palette="colorblind",ax = axes[1][1]);
sns.boxplot(data=df['ASF'], orient="v",width=0.4, palette="colorblind",ax = axes[1][2]);
```



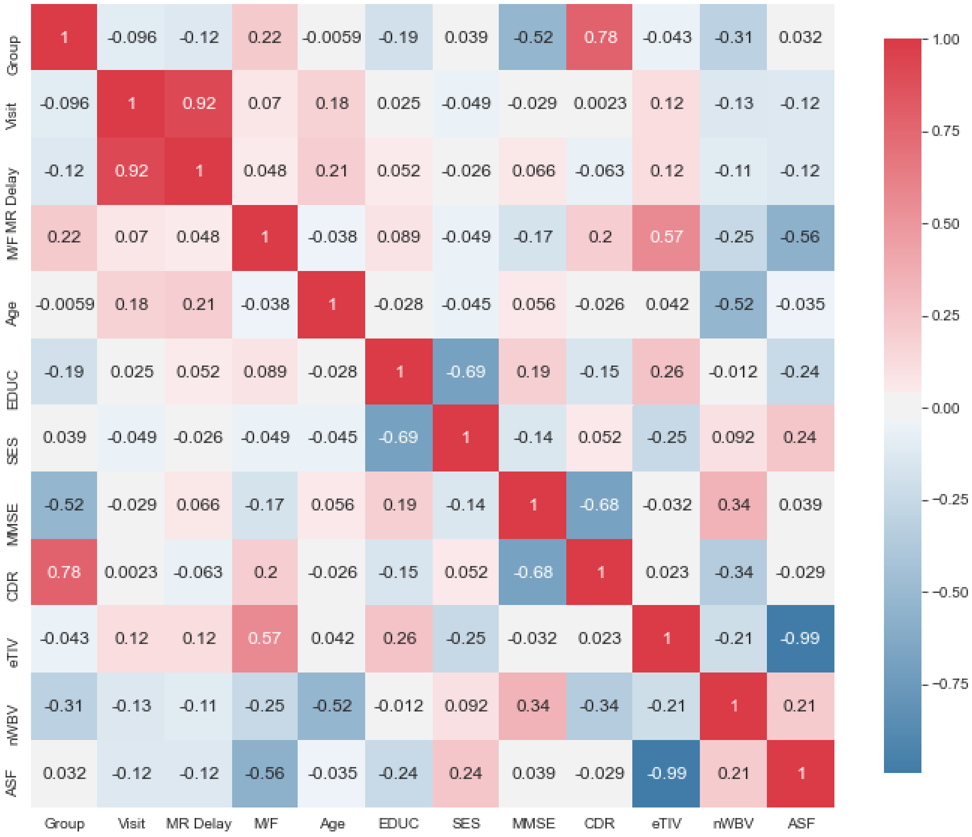
In [20]:

```
#convet the charecter data into numeric
group_map = {"Demented": 1, "Nondemented": 0}

df['Group'] = df['Group'].map(group_map)
df['M/F'] = df['M/F'].replace(['F','M'], [0,1])
```

In [21]:

```
def plot_correlation_map( df ):
    corr = df.corr()
    _ , ax = plt.subplots( figsize =( 12 , 10 ) )
    cmap = sns.diverging_palette( 240 , 10 , as_cmap = True )
    _ = sns.heatmap(corr,cmap = cmap,square=True, cbar_kws={ 'shrink' : .9 }, ax=ax, annot = True,
annot_kws = { 'fontsize' : 12 })
```

```
In [22]: plot_correlation_map(df)
```



	Group	Visit	MR Delay	M/F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
Group	1	-0.096	-0.12	0.22	-0.0059	-0.19	0.039	-0.52	0.78	-0.043	-0.31	0.032
Visit	-0.096	1	0.92	0.07	0.18	0.025	-0.049	-0.029	0.0023	0.12	-0.13	-0.12
MR Delay	-0.12	0.92	1	0.048	0.21	0.052	-0.026	0.066	-0.063	0.12	-0.11	-0.12
M/F	0.22	0.07	0.048	1	-0.038	0.089	-0.049	-0.17	0.2	0.57	-0.25	-0.56
Age	-0.0059	0.18	0.21	-0.038	1	-0.028	-0.045	0.056	-0.026	0.042	-0.52	-0.035
EDUC	-0.19	0.025	0.052	0.089	-0.028	1	-0.69	0.19	-0.15	0.26	-0.012	-0.24
SES	0.039	-0.049	-0.026	-0.049	-0.045	-0.69	1	-0.14	0.052	-0.25	0.092	0.24
MMSE	-0.52	-0.029	0.066	-0.17	0.056	0.19	-0.14	1	-0.68	-0.032	0.34	0.039
CDR	0.78	0.0023	-0.063	0.2	-0.026	-0.15	0.052	-0.68	1	0.023	-0.34	-0.029
eTIV	-0.043	0.12	0.12	0.57	0.042	0.26	-0.25	-0.032	0.023	1	-0.21	-0.99
nWBV	-0.31	-0.13	-0.11	-0.25	-0.52	-0.012	0.092	0.34	-0.34	-0.21	1	0.21
ASF	0.032	-0.12	-0.12	-0.56	-0.035	-0.24	0.24	0.039	-0.029	-0.99	0.21	1

Prediction task using Machine Learning (TBD)

```
In [ ]:
```