EDA OF NOSHOW APPOINTMENTS IN MAY 2016.

In [12]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [4]:
```
df = pd.read_csv(r'C:\Users\Sherry\Downloads\noshowappointments-kagglev2-may-2016.cs
df.head()
```

Out[4]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	S
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	
1	5.589980e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	
2	4.262960e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	
3	8.679510e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	
4	8.841190e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	

In [4]:
```
df.shape
```

Out[4]:
```
(110527, 14)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64  
 2   Gender          110527 non-null  object 
 3   ScheduledDay    110527 non-null  object 
 4   AppointmentDay  110527 non-null  object 
 5   Age             110527 non-null  int64  
 6   Neighbourhood   110527 non-null  object 
 7   Scholarship     110527 non-null  int64  
 8   Hipertension    110527 non-null  int64  
 9   Diabetes        110527 non-null  int64  
 10  Alcoholism      110527 non-null  int64  
 11  Handcap         110527 non-null  int64  
 12  SMS_received    110527 non-null  int64  
 13  No-show         110527 non-null  object 
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

In [7]:
```
df.ScheduledDay = pd.to_datetime(df.ScheduledDay)
df.ScheduledDay.head(5)
```

Out[7]:
```
0   2016-04-29 18:38:08+00:00
1   2016-04-29 16:08:27+00:00
2   2016-04-29 16:19:04+00:00
3   2016-04-29 17:29:31+00:00
4   2016-04-29 16:07:23+00:00
Name: ScheduledDay, dtype: datetime64[ns, UTC]
```

In [8]:
```
df.AppointmentDay = pd.to_datetime(df.AppointmentDay)
df.AppointmentDay.head(5)
```

Out[8]:
```
0   2016-04-29 00:00:00+00:00
1   2016-04-29 00:00:00+00:00
2   2016-04-29 00:00:00+00:00
3   2016-04-29 00:00:00+00:00
4   2016-04-29 00:00:00+00:00
Name: AppointmentDay, dtype: datetime64[ns, UTC]
```

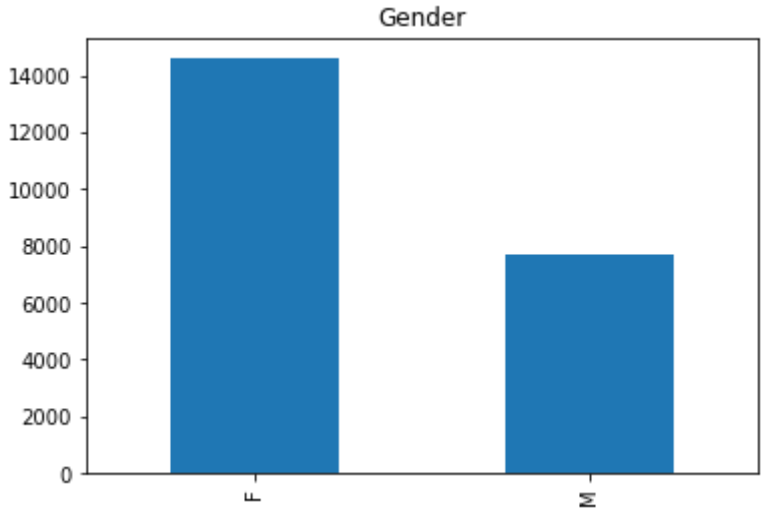
In [10]:
```
# creating a new df with only No show Appointments.
df1 = df.loc[["Yes" in title for title in df["No-show"]], :]
df1.head()
```

Out[10]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood
6	7.336880e+14	5630279	F	2016-04-27T15:05:12Z	2016-04-29T00:00:00Z	23	GOIABEIRAS
7	3.449830e+12	5630575	F	2016-04-27T15:39:58Z	2016-04-29T00:00:00Z	39	GOIABEIRAS
11	7.542950e+12	5620163	M	2016-04-26T08:44:12Z	2016-04-29T00:00:00Z	29	NOVA PALESTINA
17	1.479500e+13	5633460	F	2016-04-28T09:28:57Z	2016-04-29T00:00:00Z	40	CONQUISTA
20	6.222570e+14	5626083	F	2016-04-27T07:51:14Z	2016-04-29T00:00:00Z	30	NOVA PALESTINA

In [9]:
```
print(round(df1.Gender.value_counts()/len(df)*100))
#
df1.Gender.value_counts().plot(kind="bar")
plt.title("Gender");
```

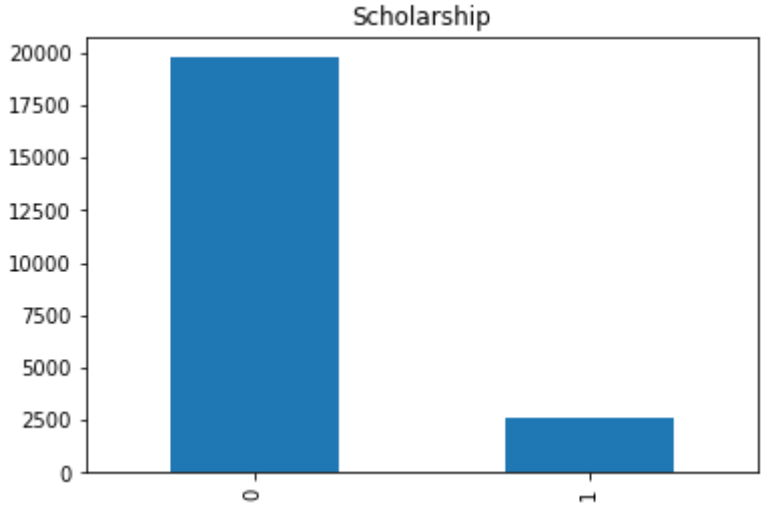
```
F    13.0
M     7.0
Name: Gender, dtype: float64
```



The number of females missing the appointments was greater than their male counterparts.

In [17]:
```
print(round(df1.Scholarship.value_counts()/len(df)*100))
#
df1.Scholarship.value_counts().plot(kind="bar")
plt.title("Scholarship");
```

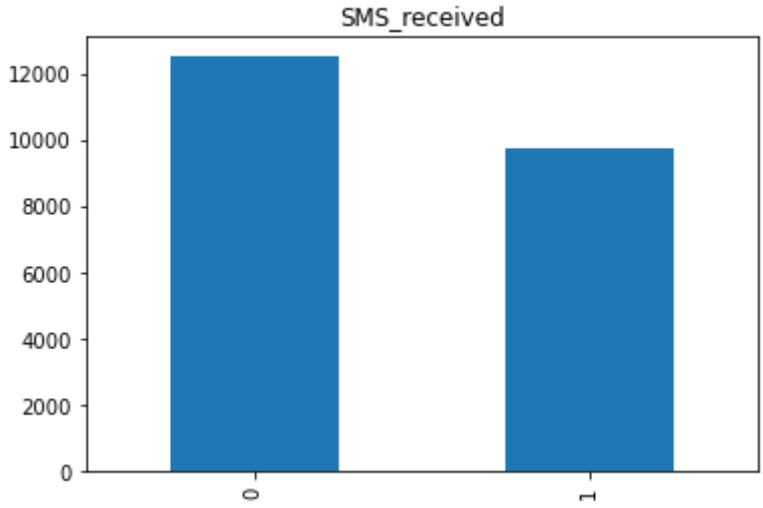
```
0    18.0
1     2.0
Name: Scholarship, dtype: float64
```



The number of patients who were under scholarship was way lesser than that of the self sponsored, for the no shows.

In [20]:
```
print(round(df1.SMS_received.value_counts()/len(df)*100))
#
df1.SMS_received.value_counts().plot(kind="bar")
plt.title("SMS_received");
```

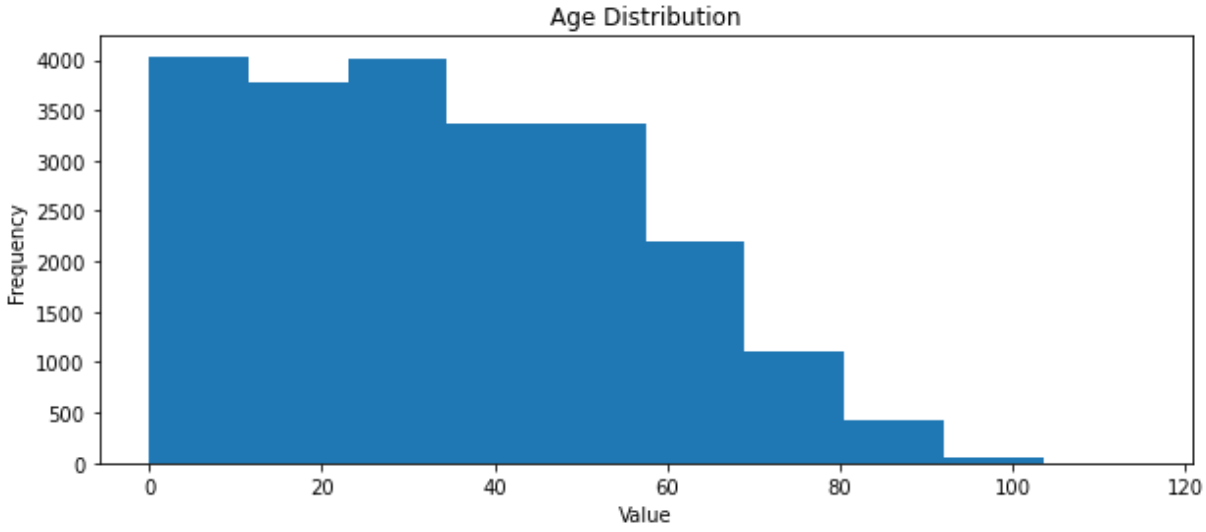
```
0    11.0
1     9.0
Name: SMS_received, dtype: float64
```



From the no shows, most of them had not received an SMS with the appointment details.

In [21]:
```
# Get the variable to examine
var_data = df1['Age']

# Create a Figure
fig = plt.figure(figsize=(10,4))

# Plot a histogram
plt.hist(var_data)

# Add titles and labels
plt.title('Age Distribution')
plt.xlabel('Value')
plt.ylabel('Frequency')

# Show the figure
fig.show()
```

```
C:\Users\Sherry\AppData\Local\Temp/ipykernel_11540/1249021034.py:16: UserWarning: Ma
tplotlib is currently using module://matplotlib_inline.backend_inline, which is a no
n-GUI backend, so cannot show the figure.
  fig.show()
```



The most No shows were aged between 1-10 and 20s-30s