In [7]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = pd.read_csv(r"C:\Users\DELL\Downloads\sales+transactions+dataset+weekly\Sales_T
print(data.describe())
print(data.head())

data[' W1'].hist(bins=20)
plt.title('Store sales distribution')
plt(show)
```

```
              W0          W1          W2          W3          W4          W5  \
count  811.000000  811.000000  811.000000  811.000000  811.000000  811.000000
mean     8.902589    9.129470    9.389642    9.717633    9.574599    9.466091
std     12.067163   12.564766   13.045073   13.553294   13.095765   12.823195
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
50%      3.000000    3.000000    3.000000    4.000000    4.000000    3.000000
75%     12.000000   12.000000   12.000000   13.000000   13.000000   12.500000
max     54.000000   53.000000   56.000000   59.000000   61.000000   52.000000

              W6          W7          W8          W9  ...  Normalized 42  \
count  811.000000  811.000000  811.000000  811.000000  ...     811.000000
mean     9.720099    9.585697    9.784217    9.681874  ...       0.299149
std     13.347375   13.049138   13.550237   13.137916  ...       0.266993
min      0.000000    0.000000    0.000000    0.000000  ...       0.000000
25%      0.000000    0.000000    0.000000    0.000000  ...       0.000000
50%      4.000000    4.000000    4.000000    4.000000  ...       0.280000
75%     13.000000   12.500000   13.000000   13.000000  ...       0.490000
max     56.000000   62.000000   63.000000   52.000000  ...       1.000000

       Normalized 43  Normalized 44  Normalized 45  Normalized 46  \
count     811.000000     811.000000     811.000000     811.000000
mean        0.287571       0.304846       0.316017       0.334760
std         0.256630       0.263396       0.262226       0.275203
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.020000       0.085000
50%         0.270000       0.300000       0.310000       0.330000
75%         0.450000       0.500000       0.500000       0.500000
max         1.000000       1.000000       1.000000       1.000000

       Normalized 47  Normalized 48  Normalized 49  Normalized 50  \
count     811.000000      811.00000     811.000000     811.000000
mean        0.314636        0.33815       0.358903       0.373009
std         0.266029        0.27569       0.286665       0.295197
min         0.000000        0.00000       0.000000       0.000000
25%         0.000000        0.10500       0.100000       0.110000
50%         0.310000        0.33000       0.330000       0.350000
75%         0.500000        0.50000       0.550000       0.560000
max         1.000000        1.00000       1.000000       1.000000

       Normalized 51
count     811.000000
mean        0.427941
std         0.342360
min         0.000000
25%         0.090000
50%         0.430000
75%         0.670000
max         1.000000

[8 rows x 106 columns]
  Product_Code  W0  W1  W2  W3  W4  W5  W6  W7  W8  ...  Normalized 42  \
0           P1  11  12  10   8  13  12  14  21   6  ...          0.06
1           P2   7   6   3   2   7   1   6   3   3  ...          0.20
2           P3   7  11   8   9  10   8   7  13  12  ...          0.27
3           P4  12   8  13   5   9   6   9  13  13  ...          0.41
4           P5   8   5  13  11   6   7   9  14   9  ...          0.27

   Normalized 43  Normalized 44  Normalized 45  Normalized 46  Normalized 47  \
0           0.22           0.28           0.39           0.50           0.00
```

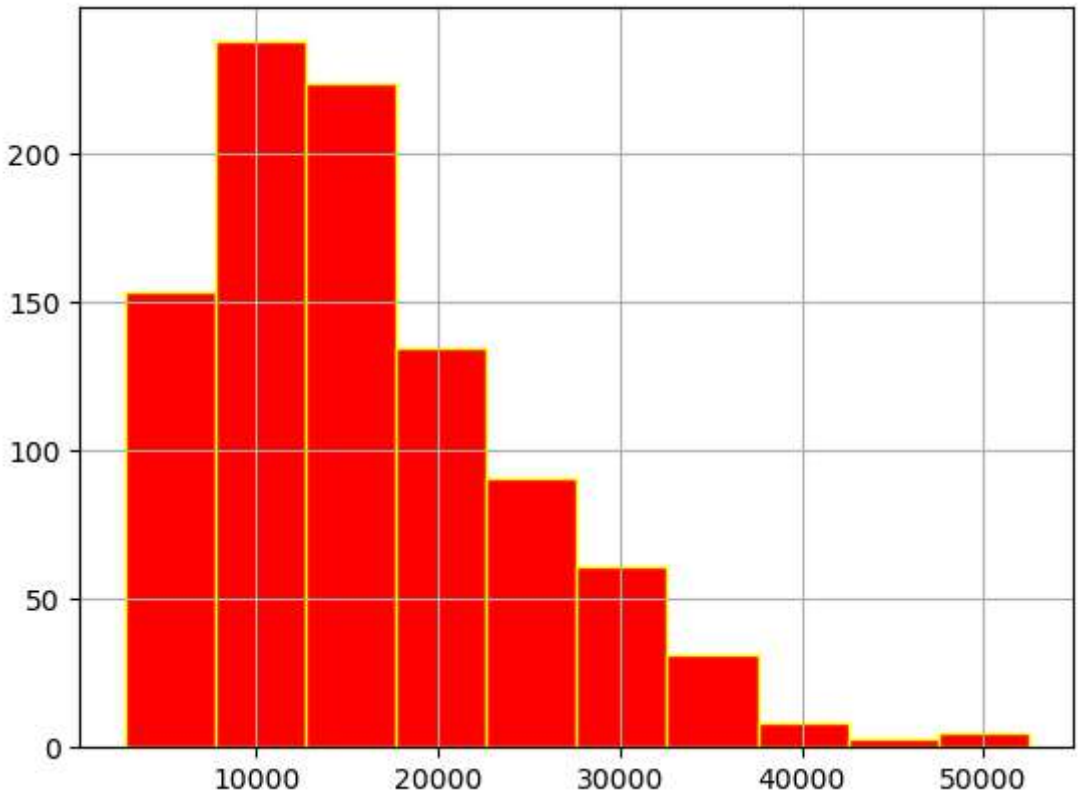
```
1           0.40           0.50           0.10           0.10           0.40
2           1.00           0.18           0.18           0.36           0.45
3           0.47           0.06           0.12           0.24           0.35
4           0.53           0.27           0.60           0.20           0.20

   Normalized 48  Normalized 49  Normalized 50  Normalized 51
0           0.22           0.17           0.11           0.39
1           0.50           0.10           0.60           0.00
2           1.00           0.45           0.45           0.36
3           0.71           0.35           0.29           0.35
4           0.13           0.53           0.33           0.40

[5 rows x 107 columns]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3802, in Index.get_loc
(self, key, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:138, in pandas._libs.index.
IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:165, in pandas._libs.index.
IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashtable.PyObject
HashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashtable.PyObject
HashTable.get_item()

KeyError: ' W1'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[7], line 8
      5 print(data.describe())
      6 print(data.head())
----> 8 data[' W1'].hist(bins=20)
      9 plt.title('Store sales distribution')
     10 plt(show)

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3807, in DataFrame.__getitem_
_(self, key)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3804, in Index.get_loc
(self, key, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: ' W1'
```

In [6]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data=pd.read_csv(r"C:/Users/DELL/Downloads/carsalesextendedmissingdata.csv")
print(data.describe())
data['Price'].hist(color='red',edgecolor='yellow')
plt.show()
```

```
       Odometer (KM)       Doors         Price
count     950.000000  950.000000    950.000000
mean   131253.237895    4.011579  16042.814737
std     69094.857187    0.382539   8581.695036
min     10148.000000    3.000000   2796.000000
25%     70391.250000    4.000000   9529.250000
50%    131821.000000    4.000000  14297.000000
75%    192668.500000    4.000000  20806.250000
max    249860.000000    5.000000  52458.000000
```

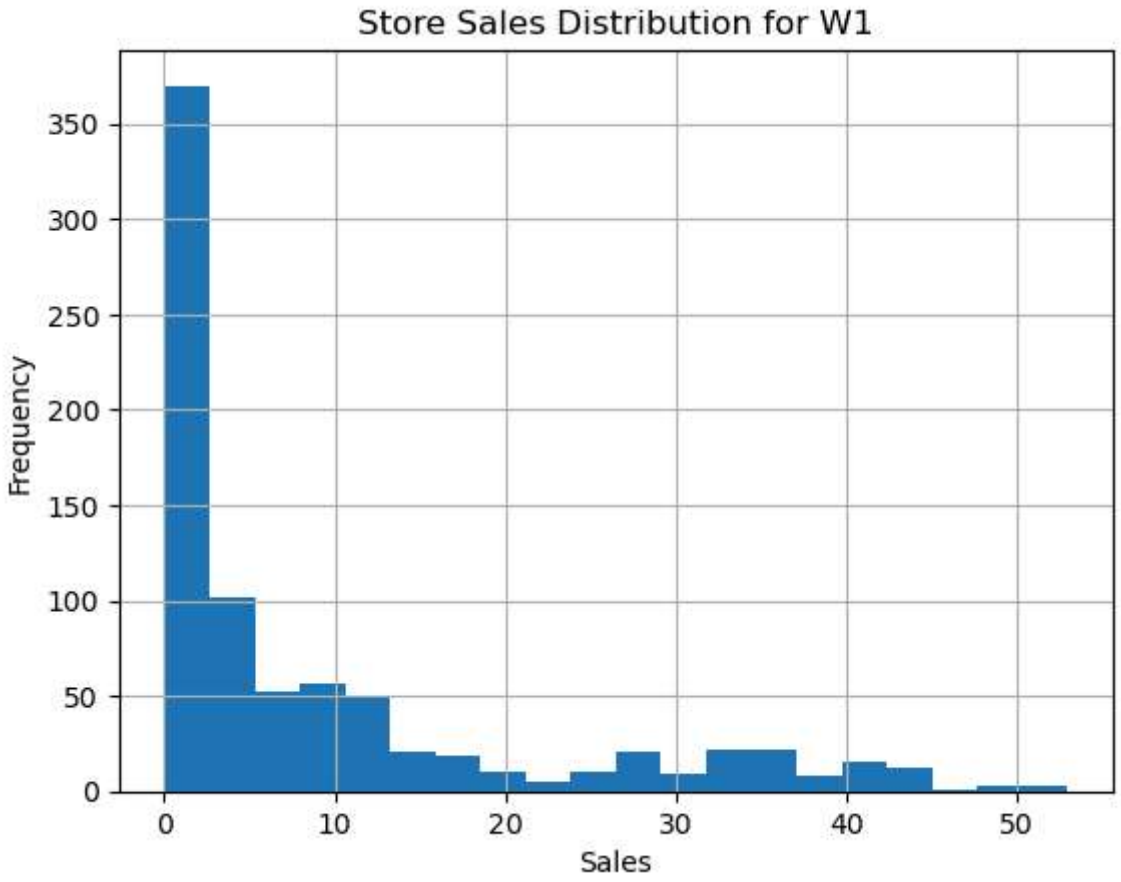


In [8]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data
data = pd.read_csv(r"C:\Users\DELL\Downloads\sales+transactions+dataset+weekly\Sales_T

# Print descriptive statistics and the first few rows of the dataset
print(data.describe())
print(data.head())

# Plot the histogram of the 'W1' column
data['W1'].hist(bins=20)
plt.title('Store Sales Distribution for W1')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```

```
              W0          W1          W2          W3          W4          W5  \
count  811.000000  811.000000  811.000000  811.000000  811.000000  811.000000
mean     8.902589    9.129470    9.389642    9.717633    9.574599    9.466091
std     12.067163   12.564766   13.045073   13.553294   13.095765   12.823195
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
50%      3.000000    3.000000    3.000000    4.000000    4.000000    3.000000
75%     12.000000   12.000000   12.000000   13.000000   13.000000   12.500000
max     54.000000   53.000000   56.000000   59.000000   61.000000   52.000000

              W6          W7          W8          W9  ...  Normalized 42  \
count  811.000000  811.000000  811.000000  811.000000  ...     811.000000
mean     9.720099    9.585697    9.784217    9.681874  ...       0.299149
std     13.347375   13.049138   13.550237   13.137916  ...       0.266993
min      0.000000    0.000000    0.000000    0.000000  ...       0.000000
25%      0.000000    0.000000    0.000000    0.000000  ...       0.000000
50%      4.000000    4.000000    4.000000    4.000000  ...       0.280000
75%     13.000000   12.500000   13.000000   13.000000  ...       0.490000
max     56.000000   62.000000   63.000000   52.000000  ...       1.000000

       Normalized 43  Normalized 44  Normalized 45  Normalized 46  \
count     811.000000     811.000000     811.000000     811.000000
mean        0.287571       0.304846       0.316017       0.334760
std         0.256630       0.263396       0.262226       0.275203
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.020000       0.085000
50%         0.270000       0.300000       0.310000       0.330000
75%         0.450000       0.500000       0.500000       0.500000
max         1.000000       1.000000       1.000000       1.000000

       Normalized 47  Normalized 48  Normalized 49  Normalized 50  \
count     811.000000      811.00000     811.000000     811.000000
mean        0.314636        0.33815       0.358903       0.373009
std         0.266029        0.27569       0.286665       0.295197
min         0.000000        0.00000       0.000000       0.000000
25%         0.000000        0.10500       0.100000       0.110000
50%         0.310000        0.33000       0.330000       0.350000
75%         0.500000        0.50000       0.550000       0.560000
max         1.000000        1.00000       1.000000       1.000000

       Normalized 51
count     811.000000
mean        0.427941
std         0.342360
min         0.000000
25%         0.090000
50%         0.430000
75%         0.670000
max         1.000000

[8 rows x 106 columns]
  Product_Code  W0  W1  W2  W3  W4  W5  W6  W7  W8  ...  Normalized 42  \
0           P1  11  12  10   8  13  12  14  21   6  ...          0.06
1           P2   7   6   3   2   7   1   6   3   3  ...          0.20
2           P3   7  11   8   9  10   8   7  13  12  ...          0.27
3           P4  12   8  13   5   9   6   9  13  13  ...          0.41
4           P5   8   5  13  11   6   7   9  14   9  ...          0.27

   Normalized 43  Normalized 44  Normalized 45  Normalized 46  Normalized 47  \
0           0.22           0.28           0.39           0.50           0.00
```

```
1           0.40           0.50           0.10           0.10           0.40
2           1.00           0.18           0.18           0.36           0.45
3           0.47           0.06           0.12           0.24           0.35
4           0.53           0.27           0.60           0.20           0.20

   Normalized 48  Normalized 49  Normalized 50  Normalized 51
0           0.22           0.17           0.11           0.39
1           0.50           0.10           0.60           0.00
2           1.00           0.45           0.45           0.36
3           0.71           0.35           0.29           0.35
4           0.13           0.53           0.33           0.40

[5 rows x 107 columns]
```



In [ ]: