In [0]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
```

In [0]:
```python
#df=spark.read.format("csv").option("header","true").load("/FileStore/tables/Sales/Car_Sales.csv")
df=spark.read.csv("/FileStore/tables/Sales/Car_Sales.csv",header=True,inferSchema=True)
df.show(5)
```

```
+-----------+---------------+---------+---------------+-------+---------------+-------+------+--------+-------+-----------+----+-----------+-----
---------------+----------------+--------------------+------------+-------------+-----+----------+-------+---------+---------------+-------------
---+--------+
ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|      ORDERDATE| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|
CUSTOMERNAME|           PHONE|        ADDRESSLINE1|ADDRESSLINE2|         CITY|STATE|POSTALCODE|COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME
|DEALSIZE|
+-----------+---------------+---------+---------------+-------+---------------+-------+------+--------+-------+-----------+----+-----------+-----
---------------+----------------+--------------------+------------+-------------+-----+----------+-------+---------+---------------+-------------
---+--------+
      10107|             30|     95.7|              2| 2871.0| 2/24/2003 0:00|Shipped|     1|       2|   2003|Motorcycles|  95|   S10_1678|   Lan
d of Toys Inc.|      2125557818|897 Long Airport ...|        null|          NYC|   NY|     10022|    USA|       NA|             Yu|           Kw
ai|   Small|
      10121|             34|    81.35|              5| 2765.9|  5/7/2003 0:00|Shipped|     2|       5|   2003|Motorcycles|  95|   S10_1678|  Reim
s Collectables|      26.47.1555|  59 rue de l'Abbaye|        null|        Reims| null|     51100| France|     EMEA|        Henriot|           Pa
ul|   Small|
      10134|             41|    94.74|              2|3884.34|  7/1/2003 0:00|Shipped|     3|       7|   2003|Motorcycles|  95|   S10_1678|     L
yon Souveniers|+33 1 46 62 7555|27 rue du Colonel...|        null|        Paris| null|     75508| France|     EMEA|       Da Cunha|         Dani
el|  Medium|
      10145|             45|    83.26|              6| 3746.7| 8/25/2003 0:00|Shipped|     3|       8|   2003|Motorcycles|  95|   S10_1678|   Toy
s4GrownUps.com|      6265557265|  78934 Hillside Dr.|        null|     Pasadena|   CA|     90003|    USA|       NA|          Young|          Jul
ie|  Medium|
      10159|             49|    100.0|             14|5205.27|10/10/2003 0:00|Shipped|     4|      10|   2003|Motorcycles|  95|   S10_1678|Corpor
ate Gift Id...|      6505551386|     7734 Strong St.|        null|San Francisco|   CA|      null|    USA|       NA|          Brown|          Jul
ie|  Medium|
+-----------+---------------+---------+---------------+-------+---------------+-------+------+--------+-------+-----------+----+-----------+-----
---------------+----------------+--------------------+------------+-------------+-----+----------+-------+---------+---------------+-------------
---+--------+
only showing top 5 rows
```

In [0]:
```python
df.printSchema()
```

```
root
-- ORDERNUMBER: integer (nullable = true)
-- QUANTITYORDERED: integer (nullable = true)
-- PRICEEACH: double (nullable = true)
-- ORDERLINENUMBER: integer (nullable = true)
-- SALES: double (nullable = true)
-- ORDERDATE: string (nullable = true)
-- STATUS: string (nullable = true)
-- QTR_ID: integer (nullable = true)
-- MONTH_ID: integer (nullable = true)
-- YEAR_ID: integer (nullable = true)
-- PRODUCTLINE: string (nullable = true)
-- MSRP: integer (nullable = true)
-- PRODUCTCODE: string (nullable = true)
-- CUSTOMERNAME: string (nullable = true)
-- PHONE: string (nullable = true)
-- ADDRESSLINE1: string (nullable = true)
-- ADDRESSLINE2: string (nullable = true)
-- CITY: string (nullable = true)
-- STATE: string (nullable = true)
-- POSTALCODE: string (nullable = true)
-- COUNTRY: string (nullable = true)
-- TERRITORY: string (nullable = true)
-- CONTACTLASTNAME: string (nullable = true)
-- CONTACTFIRSTNAME: string (nullable = true)
-- DEALSIZE: string (nullable = true)
```

In [0]:
```python
df.select(["SALES","CITY"]).show(5)
```

```
+-------+-------------+
  SALES|         CITY|
+-------+-------------+
 2871.0|          NYC|
 2765.9|        Reims|
3884.34|        Paris|
 3746.7|     Pasadena|
5205.27|San Francisco|
+-------+-------------+
only showing top 5 rows
```

In [0]:
```python
type(df)
```

Out[88]: pyspark.sql.dataframe.DataFrame

In [0]:
```python
df.columns
```

```
Out[89]: ['ORDERNUMBER',
 'QUANTITYORDERED',
 'PRICEEACH',
 'ORDERLINENUMBER',
 'SALES',
 'ORDERDATE',
 'STATUS',
 'QTR_ID',
 'MONTH_ID',
 'YEAR_ID',
 'PRODUCTLINE',
 'MSRP',
 'PRODUCTCODE',
 'CUSTOMERNAME',
 'PHONE',
 'ADDRESSLINE1',
 'ADDRESSLINE2',
 'CITY',
 'STATE',
 'POSTALCODE',
 'COUNTRY',
 'TERRITORY',
 'CONTACTLASTNAME',
 'CONTACTFIRSTNAME',
 'DEALSIZE']
```

In [0]:
```python
df_sel=df.select(['PRICEEACH','MSRP','SALES'])
```

In [0]:
```python
df_sel.describe().show()
```

```
+-------+------------------+------------------+------------------+
summary|         PRICEEACH|              MSRP|             SALES|
+-------+------------------+------------------+------------------+
  count|              2823|              2823|              2823|
   mean| 83.65854410201929|100.71555083244775|  3553.88907190932|
 stddev|20.174276527840536| 40.18791167720266|1841.8651057401842|
    min|             26.88|                33|            482.13|
    max|             100.0|               214|           14082.8|
+-------+------------------+------------------+------------------+
```

In [0]:
```python
df=df.na.drop(how='any',subset=['SALES'])
```

In [0]:
```python
from pyspark.sql.functions import isnan, when, count, col
df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

```
+-----------+---------------+---------+---------------+-----+---------+------+------+--------+-------+-----------+----+-----------+------------+-
----+------------+------------+----+-----+----------+-------+---------+---------------+----------------+--------+
ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|SALES|ORDERDATE|STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|CUSTOMERNAME|PH
ONE|ADDRESSLINE1|ADDRESSLINE2|CITY|STATE|POSTALCODE|COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
+-----------+---------------+---------+---------------+-----+---------+------+------+--------+-------+-----------+----+-----------+------------+-
----+------------+------------+----+-----+----------+-------+---------+---------------+----------------+--------+
          0|              0|        0|              0|    0|        0|     0|     0|       0|      0|          0|   0|          0|           0|
0|           0|        2521|   0| 1486|        76|      0|        0|              0|               0|       0|
+-----------+---------------+---------+---------------+-----+---------+------+------+--------+-------+-----------+----+-----------+------------+-
----+------------+------------+----+-----+----------+-------+---------+---------------+----------------+--------+
```

In [0]:
```python
bounds = {
    c: dict(
        zip(["q1", "q3"], df_sel.approxQuantile(c, [0.25, 0.75], 0))
    )
    for c in df_sel.columns
}
for c in bounds:
    iqr = bounds[c]['q3'] - bounds[c]['q1']
    bounds[c]['lower'] = bounds[c]['q1'] - (iqr * 1.5)
    bounds[c]['upper'] = bounds[c]['q3'] + (iqr * 1.5)
print(bounds)
```

```
{'PRICEEACH': {'q1': 68.8, 'q3': 100.0, 'lower': 21.999999999999993, 'upper': 146.8}, 'MSRP': {'q1': 68.0, 'q3': 124.0, 'lower': -16.0, 'upper':
208.0}, 'SALES': {'q1': 2203.11, 'q3': 4508.0, 'lower': -1254.225, 'upper': 7965.335}}
```

In [0]:
```python
df.describe().toPandas()
```

Out[95]:

	summary	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID
0	count	2823	2823	2823	2823	2823	2823	2823	2823	2823
1	mean	10258.725115125753	35.09280906836698	83.65854410201929	6.466170740347148	3553.88907190932	None	None	2.7176762309599716	7.0924548352816155
2	stddev	92.0854775957196	9.74144273706958	20.174276527840536	4.22584096469094	1841.8651057401842	None	None	1.203878088001756	3.656633307661765
3	min	10100	6	26.88	1	482.13	1/10/2003 0:00	Cancelled	1	1
4	max	10425	97	100.0	18	14082.8	9/9/2004 0:00	Shipped	4	12

In [0]:
```python
df.toPandas().isnull().sum()
```

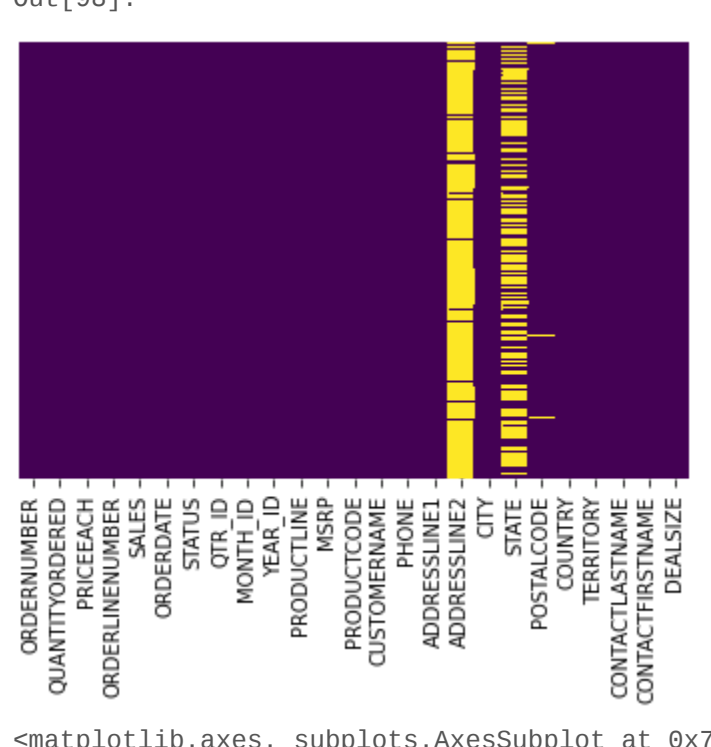
```
Out[96]: ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY              0
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

In [0]:
```python
df.toPandas().shape
```

Out[97]: (2823, 25)

In [0]:
```python
sns.heatmap(df.toPandas().isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[98]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7ee0190>

In [0]:
```python
df=df.drop('STATE','ADDRESSLINE2')
```

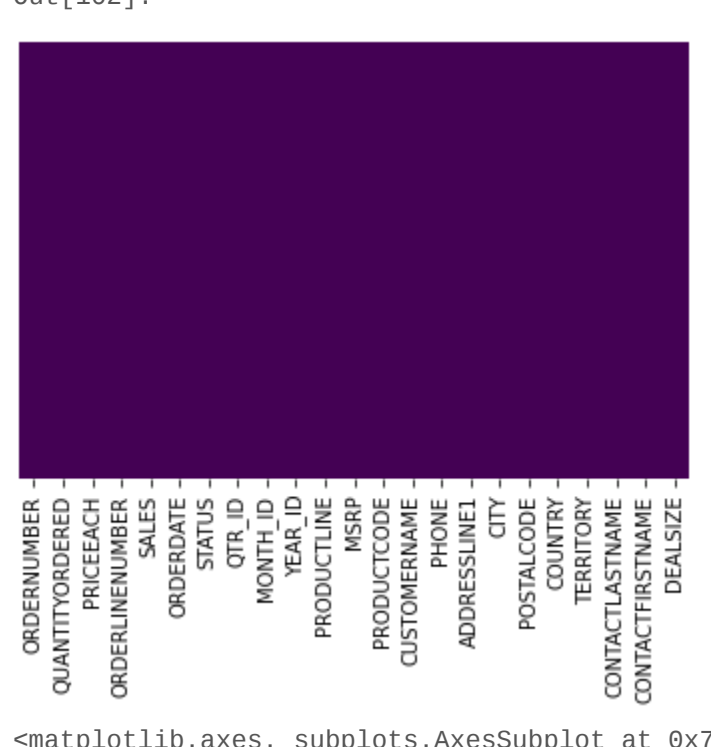
In [0]:
```python
df=df.na.drop(how='any',subset=['POSTALCODE'])
```

In [0]:
```python
df.toPandas().isnull().sum()
```

```
Out[101]: ORDERNUMBER         0
QUANTITYORDERED     0
PRICEEACH           0
ORDERLINENUMBER     0
SALES               0
ORDERDATE           0
STATUS              0
QTR_ID              0
MONTH_ID            0
YEAR_ID             0
PRODUCTLINE         0
MSRP                0
PRODUCTCODE         0
CUSTOMERNAME        0
PHONE               0
ADDRESSLINE1        0
CITY                0
POSTALCODE          0
COUNTRY             0
TERRITORY           0
CONTACTLASTNAME     0
CONTACTFIRSTNAME    0
DEALSIZE            0
dtype: int64
```

In [0]:
```python
sns.heatmap(df.toPandas().isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[102]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7ee00d0>

In [0]:
```python
min_thres=df.select('PRICEEACH').toPandas().quantile(0.05)
min_thres
```

```
Out[103]: PRICEEACH    42.76
Name: 0.05, dtype: float64
```

In [0]:
```python
df.select('PRICEEACH').where(df.PRICEEACH < 21.86250000000001).show()
```

```
+---------+
PRICEEACH|
+---------+
+---------+
```

In [0]:
```python
max_thres=df.select('PRICEEACH').toPandas().quantile(0.95)
max_thres
```

```
Out[105]: PRICEEACH    100.0
Name: 0.95, dtype: float64
```

In [0]:
```python
df.select('PRICEEACH').where(df.PRICEEACH>100).show()
```

```
+---------+
PRICEEACH|
+---------+
+---------+
```

In [0]:
```python
Q1 = np.percentile(df.select('PRICEEACH').toPandas(), 25, interpolation = 'midpoint')
Q2 = np.percentile(df.select('PRICEEACH').toPandas(), 50, interpolation = 'midpoint')
Q3 = np.percentile(df.select('PRICEEACH').toPandas(), 75, interpolation = 'midpoint')
IQR=Q3-Q1
low_lim = Q1 - 1.5 * IQR
up_lim = Q3 + 1.5 * IQR
df.where(df.PRICEEACH < low_lim).toPandas().shape
```

Out[107]: (0, 23)

In [0]:
```python
df.where(df.PRICEEACH > up_lim).toPandas().shape
```

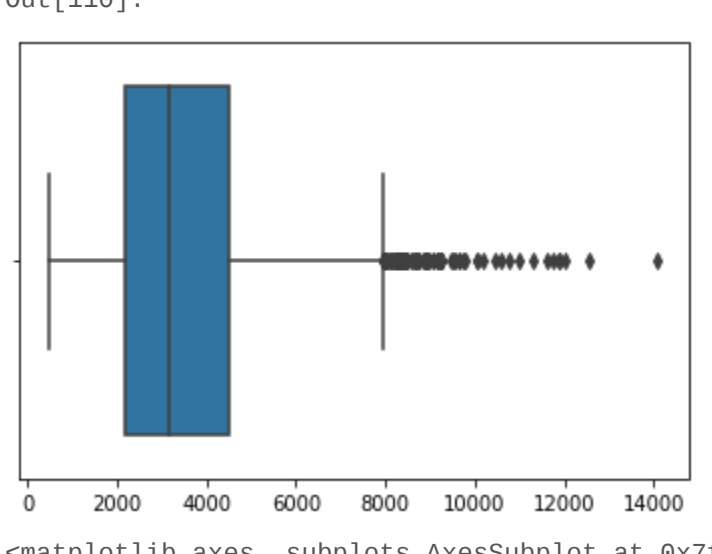
Out[108]: (0, 23)

In [0]:
```python
Q4 = np.percentile(df.select('SALES').toPandas(), 25, interpolation = 'midpoint')
Q5 = np.percentile(df.select('SALES').toPandas(), 50, interpolation = 'midpoint')
Q6 = np.percentile(df.select('SALES').toPandas(), 75, interpolation = 'midpoint')
IQR1=Q6-Q4
low_lim1 = Q4 - 1.5 * IQR1
up_lim1 = Q6 + 1.5 * IQR1
df.where(df.SALES > up_lim1).toPandas().shape
```

Out[109]: (79, 23)

In [0]:
```python
sns.boxplot(df.select('SALES').toPandas())
```

Out[110]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7ded430>

In [0]:
```python
df.where(df.SALES<low_lim1).toPandas().shape
```

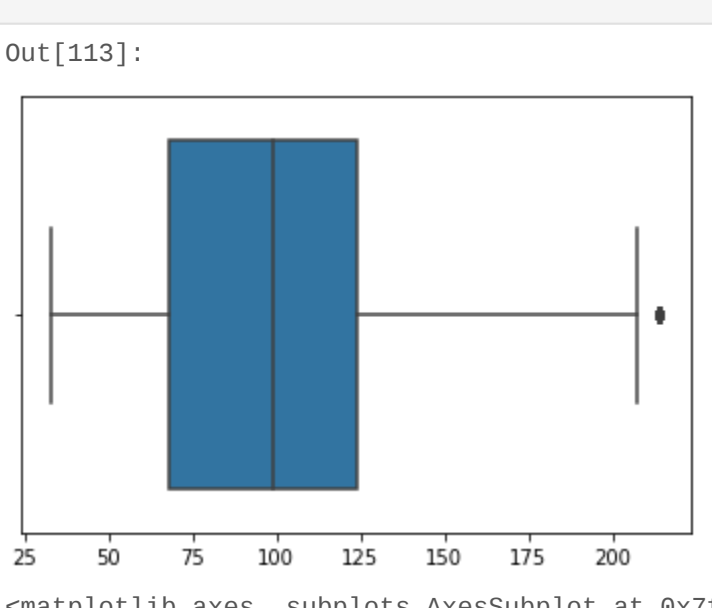
Out[111]: (0, 23)

In [0]:
```python
Q7 = np.percentile(df.select('MSRP').toPandas(), 25, interpolation = 'midpoint')
Q8 = np.percentile(df.select('MSRP').toPandas(), 50, interpolation = 'midpoint')
Q9 = np.percentile(df.select('MSRP').toPandas(), 75, interpolation = 'midpoint')
IQR2=Q9-Q7
low_lim2 = Q7 - 1.5 * IQR2
up_lim2 = Q9 + 1.5 * IQR2
df.where(df.MSRP > up_lim2).toPandas().shape
```

Out[112]: (26, 23)

In [0]:
```python
sns.boxplot(df.select('MSRP').toPandas())
```

Out[113]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7eff790>

In [0]:
```python
df.where(df.MSRP<low_lim2).toPandas().shape
```

Out[114]: (0, 23)

In [0]:
```python
(df.count(), len(df.columns))
```

Out[115]: (2747, 23)

In [0]:
```python
df2 = df.where(df.SALES<up_lim1)
(df2.count(),len(df2.columns))
```

Out[116]: (2668, 23)

In [0]:
```python
df2 = df2.where(df.MSRP<up_lim2)
(df2.count(),len(df2.columns))
```

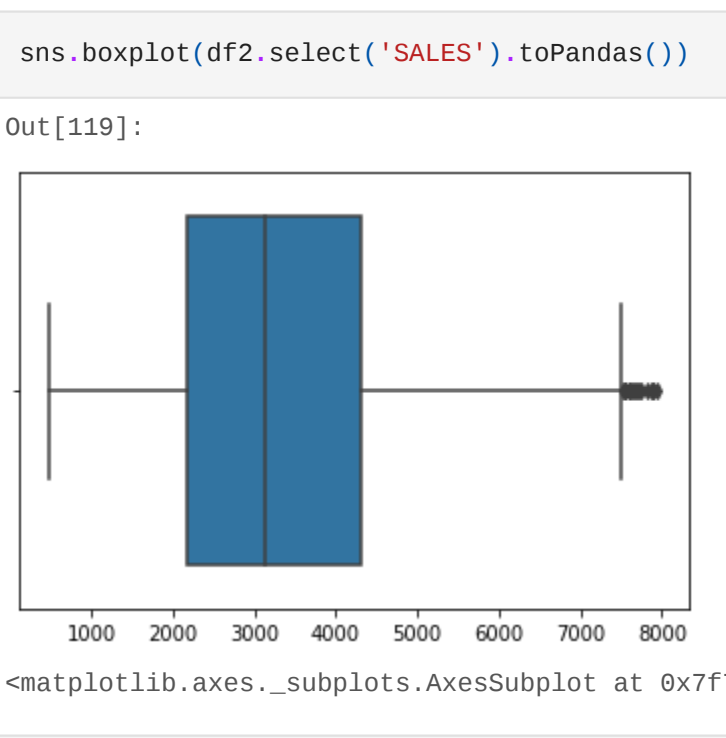
Out[117]: (2649, 23)

In [0]:
```python
Q10 = np.percentile(df2.select('SALES').toPandas(), 25, interpolation = 'midpoint')
Q11 = np.percentile(df2.select('SALES').toPandas(), 50, interpolation = 'midpoint')
Q12 = np.percentile(df2.select('SALES').toPandas(), 75, interpolation = 'midpoint')
IQR3=Q12-Q10
low_lim3 = Q10 - 1.5 * IQR3
up_lim3 = Q12 + 1.5 * IQR3
df2.where(df2.SALES > up_lim3).toPandas().shape
```

Out[118]: (17, 23)

In [0]:
```python
sns.boxplot(df2.select('SALES').toPandas())
```

Out[119]:



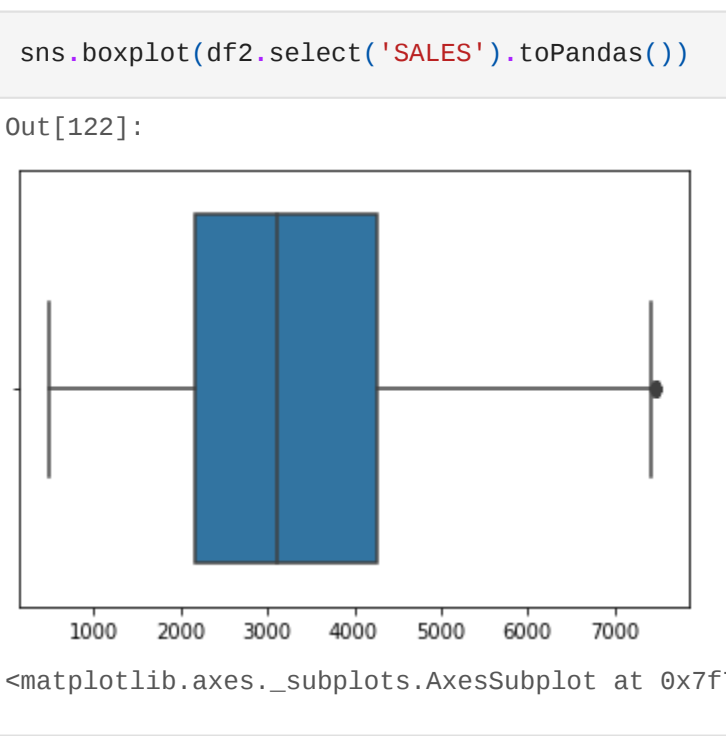
<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7e3d730>

In [0]:
```python
df2 = df2.where(df2.SALES<up_lim3)
(df2.count(),len(df2.columns))
```

Out[120]: (2632, 23)

In [0]:
```python
sns.boxplot(df2.select('SALES').toPandas())
```

Out[122]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7a43880>

In [0]:
```python
Q13 = np.percentile(df2.select('SALES').toPandas(), 25, interpolation = 'midpoint')
Q14 = np.percentile(df2.select('SALES').toPandas(), 50, interpolation = 'midpoint')
Q15 = np.percentile(df2.select('SALES').toPandas(), 75, interpolation = 'midpoint')
IQR4=Q15-Q13
low_lim4 = Q13 - 1.5 * IQR4
up_lim4 = Q15 + 1.5 * IQR4
df2.where(df2.SALES > up_lim4).toPandas().shape
```

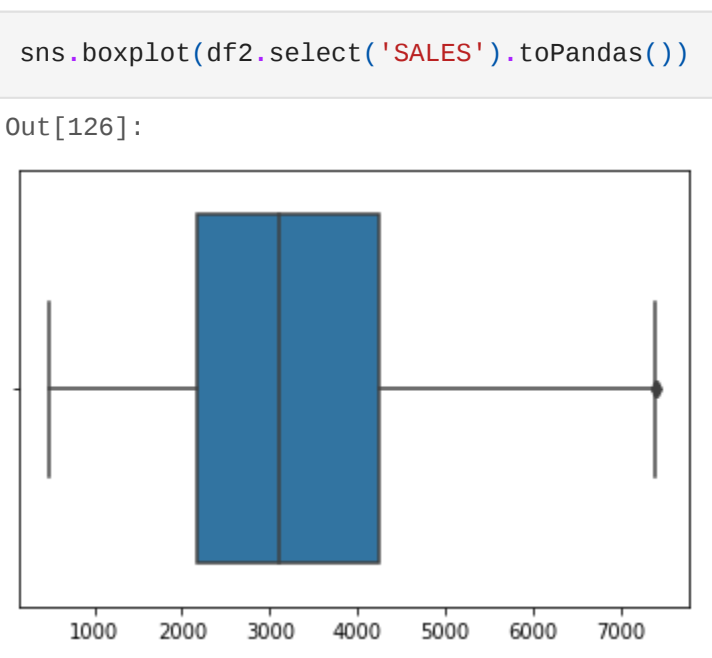
Out[124]: (7, 23)

In [0]:
```python
df2 = df2.where(df2.SALES<up_lim4)
(df2.count(),len(df2.columns))
```

Out[125]: (2625, 23)

In [0]:
```python
sns.boxplot(df2.select('SALES').toPandas())
```

Out[126]:



<matplotlib.axes._subplots.AxesSubplot at 0x7f71a7ee8880>