Importing libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

Loading the datasets

In [2]:
```
behaviour =pd.read_csv(r'G:\ajay\Ajay\Education\1.1 IT sector\Practice works\
transaction=pd.read_csv(r'G:\ajay\Ajay\Education\1.1 IT sector\Practice works
```

In [3]:
```
# Checking the imported data
transaction
```

Out[3]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	T
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	
...	...	...	...	...	...	...	...	
264831	43533	272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	
264832	43325	272	272358	270154	74	Tostitos Splash Of Lime 175g	1	
264833	43410	272	272379	270187	51	Doritos Mexicana 170g	2	
264834	43461	272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g	2	
264835	43365	272	272380	270189	74	Tostitos Splash Of Lime 175g	2	

264836 rows × 8 columns

In [4]:
```
behaviour
```

Out[4]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream
...	...	...	...
72632	2370651	MIDAGE SINGLES/COUPLES	Mainstream
72633	2370701	YOUNG FAMILIES	Mainstream
72634	2370751	YOUNG FAMILIES	Premium
72635	2370961	OLDER FAMILIES	Budget
72636	2373711	YOUNG SINGLES/COUPLES	Mainstream

72637 rows × 3 columns

Data Cleaning

In [5]:
```
# shape of dataset
print('Transaction Dataset:', transaction.shape)
print('Behaviour Dataset:',behaviour.shape)
```

```
Transaction Dataset: (264836, 8)
Behaviour Dataset: (72637, 3)
```

Transaction Table

In [6]:
```
# Viewing the information of the dataset
transaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   DATE            264836 non-null  int64
 1   STORE_NBR       264836 non-null  int64
 2   LYLTY_CARD_NBR  264836 non-null  int64
 3   TXN_ID          264836 non-null  int64
 4   PROD_NBR        264836 non-null  int64
 5   PROD_NAME       264836 non-null  object
 6   PROD_QTY        264836 non-null  int64
 7   TOT_SALES       264836 non-null  float64
dtypes: float64(1), int64(6), object(1)
memory usage: 16.2+ MB
```

In [8]:
```
# Converting the Date column to date format
transaction['DATE'] = pd.to_datetime(transaction['DATE'],origin='2020-01-01')
```

In [9]:
```
pd.DatetimeIndex(transaction.DATE).year.value_counts()
```

Out[9]:
```
2020    264836
Name: DATE, dtype: int64
```

In [10]:
```
# Viewing the statistical information of the dataset#
transaction.describe()
```

Out[10]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SAL
count	264836.00000	2.648360e+05	2.648360e+05	264836.000000	264836.000000	264836.000(
mean	135.08011	1.355495e+05	1.351583e+05	56.583157	1.907309	7.3042
std	76.78418	8.057998e+04	7.813303e+04	32.826638	0.643654	3.0832
min	1.00000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.500(
25%	70.00000	7.002100e+04	6.760150e+04	28.000000	2.000000	5.400(
50%	130.00000	1.303575e+05	1.351375e+05	56.000000	2.000000	7.400(
75%	203.00000	2.030942e+05	2.027012e+05	85.000000	2.000000	9.200(
max	272.00000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.000(

In [12]:
```
# Extracting first name from product name as the brand name
transaction['BRAND_NAME']=transaction['PROD_NAME'].apply(lambda x: x.split("
```

In [14]:
```
# extracting the last word from product namewhich is the pkg details
transaction['PROD_PKG']=transaction['PROD_NAME'].apply(lambda x: x.split(" ")
```

In [15]:
```
# removing the first word from product name
transaction['PROD_DESC'] = transaction['PROD_NAME'].str.split(n=1).str[1]
```

In [16]:
```
# also removing the last word further to get the product description
transaction['PROD_DESC']=transaction['PROD_DESC'].str.rsplit(' ',1).str[0]
```

In [17]:
```
transaction
```

Out[17]:

_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NAMI
1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	Natura
1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	CCs
1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	Smiths
2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	Smiths
2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	Kettle
...	...	...	...	...	...	...	..
272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	10.8	Kettle
272	272358	270154	74	Tostitos Splash Of Lime 175g	1	4.4	Tostitos
272	272379	270187	51	Doritos Mexicana 170g	2	8.8	Doritos
272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g	2	7.8	Doritos
272	272380	270189	74	Tostitos Splash Of Lime 175g	2	8.8	Tostitos

In [18]:
```
#extracting only numeric characters
transaction['PROD_PKG']= transaction.PROD_PKG.str.extract('(\d+)')
```

In [19]:
```
transaction
```

Out[19]:

E_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NAM
1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	Natur
1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	CC
1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	Smith
2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	Smith
2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	Kett
...	...	...	...	...	...	...	
272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	10.8	Kett
272	272358	270154	74	Tostitos Splash Of Lime 175g	1	4.4	Tostito
272	272379	270187	51	Doritos Mexicana 170g	2	8.8	Dorito
272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g	2	7.8	Dorito
272	272380	270189	74	Tostitos Splash Of Lime 175g	2	8.8	Tostito

In [20]:
```
transaction.PROD_PKG.value_counts()
```

Out[20]:
```
175    66390
150    43131
134    25102
110    22387
170    19983
165    15297
300    15166
330    12540
380     6418
270     6285
210     6272
200     4473
250     3169
90      3008
190     2995
160     2970
220     1564
70      1507
180     1468
125     1454
Name: PROD_PKG, dtype: int64
```

You can see that around 3257 observations are missing in product pkg. As observed earlier the product name

In [21]:
```
transaction["PROD_PKG"].fillna("No Value", inplace = True)
```

In [22]:
```
transaction.PROD_PKG.value_counts()
```

Out[22]:
```
175         66390
150         43131
134         25102
110         22387
170         19983
165         15297
300         15166
330         12540
380          6418
270          6285
210          6272
200          4473
No Value     3257
250          3169
90           3008
190          2995
160          2970
220          1564
70           1507
180          1468
125          1454
Name: PROD_PKG, dtype: int64
```

In [23]:
```
transaction[transaction['PROD_PKG'] == 'No Value']
```

Out[23]:

TORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NA
83	83008	82099	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
208	208139	206906	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
237	237227	241132	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
243	243070	246706	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
7	7077	6604	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
...	...	...	...	...	...	...	
260	260240	259480	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
261	261035	259860	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
266	266413	264246	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
269	269133	265839	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
272	272156	269855	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke

In [25]:
```
transaction["PROD_PKG"].replace({"No Value": "135"}, inplace=True)
transaction[transaction['PROD_PKG'] == '135']
```

Out[25]:

TORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NA
83	83008	82099	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
208	208139	206906	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
237	237227	241132	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
243	243070	246706	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
7	7077	6604	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
...	...	...	...	...	...	...	
260	260240	259480	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
261	261035	259860	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
266	266413	264246	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Ke
269	269133	265839	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke
272	272156	269855	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Ke

In [27]:
```
transaction.BRAND_NAME.value_counts()
```

Out[27]:
```
Kettle        41288
Smiths        28860
Pringles      25102
Doritos       24962
Thins         14075
RRD           11894
Infuzions     11057
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Old            9324
Tyrrells       6442
Grain          6272
Natural        6050
Red            5885
Cheezels       4603
CCs            4551
Woolworths     4437
Dorito         3185
Infzns         3144
Smith          2963
Cheetos        2927
Snbts          1576
Burger         1564
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: BRAND_NAME, dtype: int64
```

There are Brand name that are duplicated , for example RRD is same as RED, SNBTS is SUNBITE etc.So, we need to replace them.

In [30]:
```
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Red','RRD')
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Snbts','Sunbit
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Dorito','Dorit
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Grain','GrnWve
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Infzns','Infuz
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('WW','Woolworth
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('Smith','Smiths
transaction['BRAND_NAME'] = transaction['BRAND_NAME'].replace('NCC','Natural'
```

In [31]:
```
transaction.BRAND_NAME.value_counts()
```

Out[31]:
```
Kettle        41288
Smiths        31823
Doritos       28147
Pringles      25102
RRD           17779
Woolworths    14757
Infuzions     14201
Thins         14075
Cobs           9693
Tostitos       9471
Twisties       9454
Old            9324
GrnWves        7740
Natural        7469
Tyrrells       6442
Cheezels       4603
CCs            4551
Sunbites       3008
Cheetos        2927
Burger         1564
French         1418
Name: BRAND_NAME, dtype: int64
```

Behaviour table

In [33]:
```
behaviour.head()
```

Out[33]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

In [34]:
```
# Merging tables
merged = transaction.merge(behaviour, on='LYLTY_CARD_NBR', how='left')
```

In [35]:
```
merged
```

Out[35]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME
0	2020-01-01 00:00:00.000043390	1	1000	1	5	Natural Chip Compny SeaSalt175g
1	2020-01-01 00:00:00.000043599	1	1307	348	66	CCs Nacho Cheese 175g
2	2020-01-01 00:00:00.000043605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g
3	2020-01-01 00:00:00.000043329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g
4	2020-01-01 00:00:00.000043330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g
...	...	...	...	...	...	...
264831	2020-01-01 00:00:00.000043533	272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g
264832	2020-01-01 00:00:00.000043325	272	272358	270154	74	Tostitos Splash Of Lime 175g
264833	2020-01-01 00:00:00.000043410	272	272379	270187	51	Doritos Mexicana 170g
264834	2020-01-01 00:00:00.000043461	272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g
264835	2020-01-01 00:00:00.000043365	272	272380	270189	74	Tostitos Splash Of Lime 175g

264836 rows × 13 columns

In [38]:
```
merged.info()
```

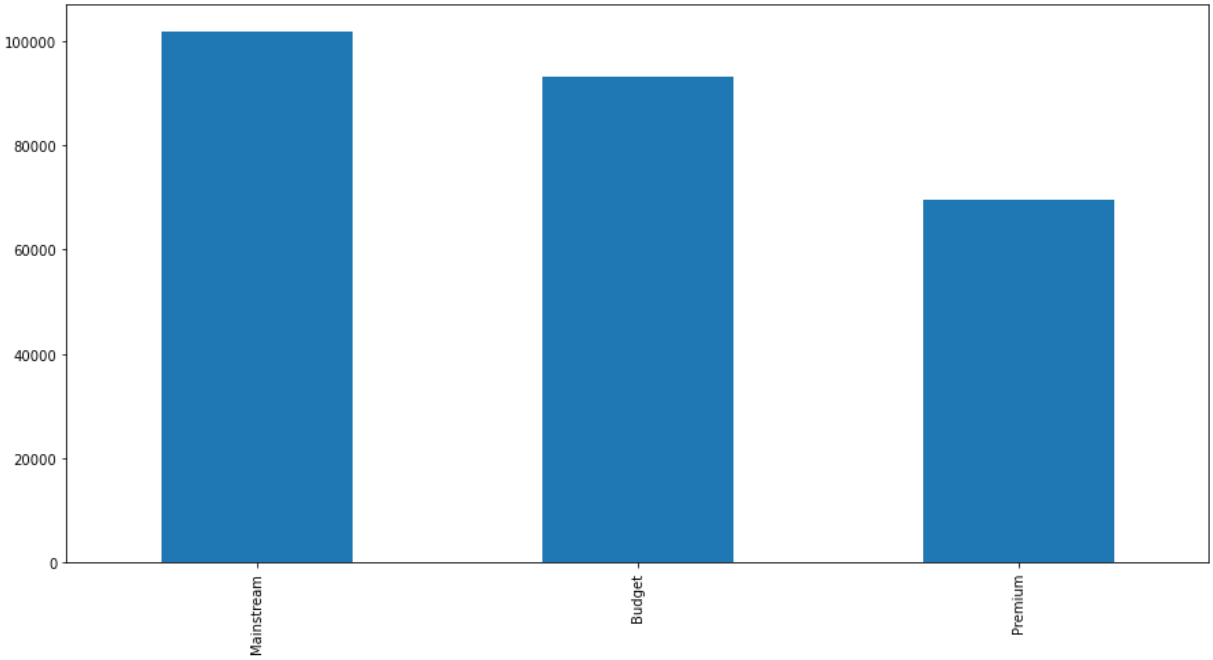
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 264836 entries, 0 to 264835
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   DATE              264836 non-null  datetime64[ns]
 1   STORE_NBR         264836 non-null  int64
 2   LYLTY_CARD_NBR    264836 non-null  int64
 3   TXN_ID            264836 non-null  int64
 4   PROD_NBR          264836 non-null  int64
 5   PROD_NAME         264836 non-null  object
 6   PROD_QTY          264836 non-null  int64
 7   TOT_SALES         264836 non-null  float64
 8   BRAND_NAME        264836 non-null  object
 9   PROD_PKG          264836 non-null  object
 10  PROD_DESC         264836 non-null  object
 11  LIFESTAGE         264836 non-null  object
 12  PREMIUM_CUSTOMER  264836 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(5), object(6)
memory usage: 28.3+ MB
```

In [40]:
```
# checking null values
merged.isnull().sum()
```

Out[40]:
```
DATE                0
STORE_NBR           0
LYLTY_CARD_NBR      0
TXN_ID              0
PROD_NBR            0
PROD_NAME           0
PROD_QTY            0
TOT_SALES           0
BRAND_NAME          0
PROD_PKG            0
PROD_DESC           0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

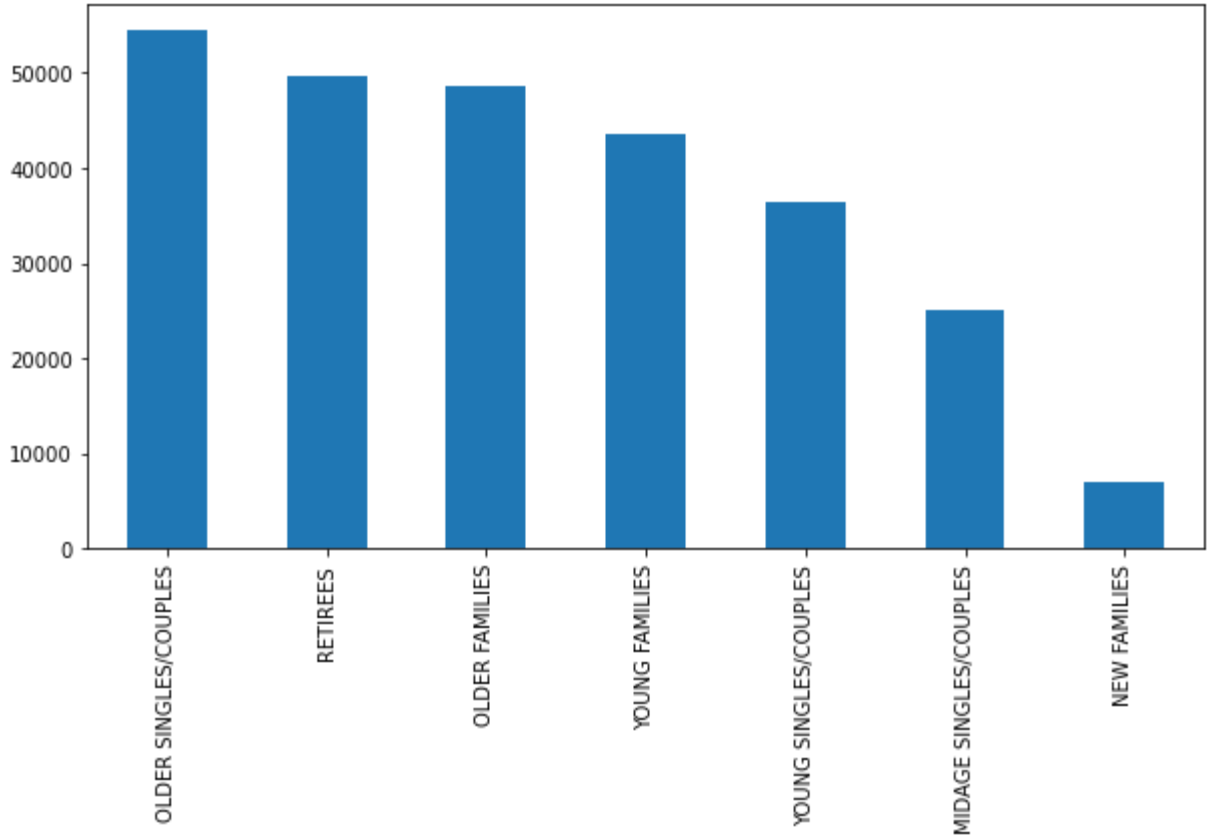
In [43]:
```
merged.PREMIUM_CUSTOMER.value_counts().plot(kind='bar',figsize=(15,7.5))
```

Out[43]: <AxesSubplot:>



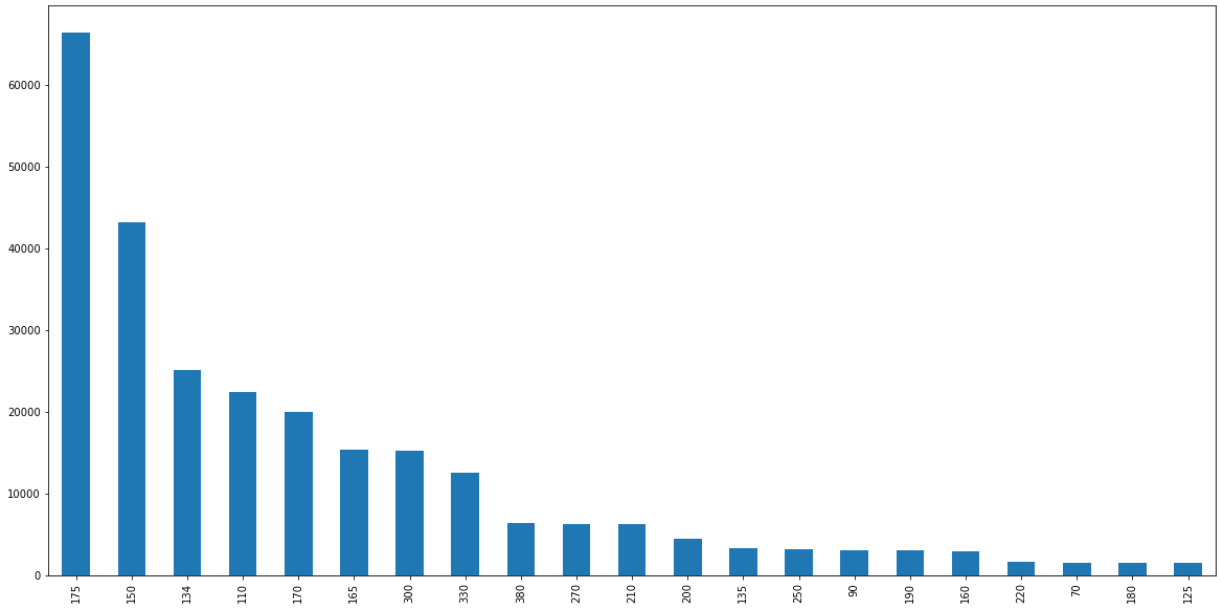
In [44]:
```
merged.LIFESTAGE.value_counts().plot(kind='bar',figsize=(10,5))
```

Out[44]: <AxesSubplot:>



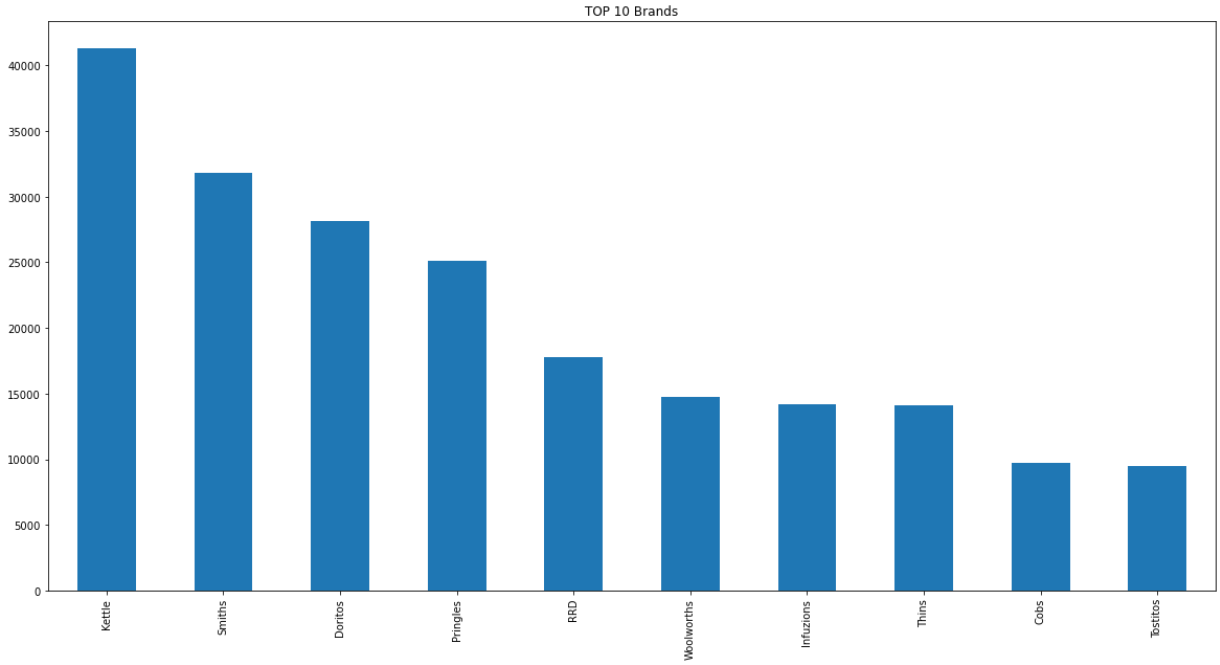
In [45]:
```
merged.PROD_PKG.value_counts().plot(kind='bar',figsize=(20,10))
```

Out[45]: <AxesSubplot:>



In [51]:
```
# top 10 values
merged.BRAND_NAME.value_counts()[0:10].plot(kind='bar',figsize=(20,10))
plt.title('TOP 10 Brands')
```

Out[51]: Text(0.5, 1.0, 'TOP 10 Brands')



In [53]:
```
merged.head()
```

Out[53]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_
0	2020-01-01 00:00:00.000043390	1	1000	1	5	Natural Chip Compny SeaSalt175g	
1	2020-01-01 00:00:00.000043599	1	1307	348	66	CCs Nacho Cheese 175g	
2	2020-01-01 00:00:00.000043605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	
3	2020-01-01 00:00:00.000043329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	
4	2020-01-01 00:00:00.000043330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	

In [ ]: