Import required libraries

In [0]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [0]:

```
import pandas as pd
pd.set_option('display.max_rows', 999)
pd.set_option('max_colwidth', 40)
pd.set_option('display.max_columns', 50)
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style(style='darkgrid')
%matplotlib inline
```

Read data from csv

In [0]:

```
df_walmartAnalysis = pd.read_csv('/content/drive/My Drive/DS Course Assessments/walmart_case_study/Walmart_Store_sales.csv')
```

In [4]:

```
# view shape and top 5 rows of data
print('There are {} records and {} features.'.format(df_walmartAnalysis.shape[0], df_walmartAnalysis.shape[1]))
df_walmartAnalysis.head()
```

```
There are 6435 records and 8 features.
```

Out[4]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106

In [5]:

```
# see info of the data
# From info. we can see the data has no null values and all features are into their proper dtype.
# Date is an object converting it to DateTime
df_walmartAnalysis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Store         6435 non-null   int64  
 1   Date          6435 non-null   object 
 2   Weekly_Sales  6435 non-null   float64
```

```
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

In [0]:

```
# Converting Date to DateTime from object.
# df_walmartAnalysis['Date'] = pd.to_datetime(df_walmartAnalysis['Date'])
```

In [7]:

```
# describing df_walmartAnalysis
# From here we can see that there are 45 walmart stores with weekly sales in the range of $2,01,000 to $38,18,686.
# Min temperature being -2.06degC and Max being +100.14degC. Min. Fuel Price is $2.47 and Max. is $4.47.
# CPI is in the range of [126.06, 227.23] and Unemployment is in the range of [3.87, 14.31]

df_walmartAnalysis.describe()
```

Out[7]:

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
count	6435.000000	6.435000e+03	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000
mean	23.000000	1.046965e+06	0.069930	60.663782	3.358607	171.578394	7.999151
std	12.988182	5.643666e+05	0.255049	18.444933	0.459020	39.356712	1.875885
min	1.000000	2.099862e+05	0.000000	-2.060000	2.472000	126.064000	3.879000
25%	12.000000	5.533501e+05	0.000000	47.460000	2.933000	131.735000	6.891000
50%	23.000000	9.607460e+05	0.000000	62.670000	3.445000	182.616521	7.874000
75%	34.000000	1.420159e+06	0.000000	74.940000	3.735000	212.743293	8.622000
max	45.000000	3.818686e+06	1.000000	100.140000	4.468000	227.232807	14.313000

Which store with maximum sales?

In [8]:

```
# finding the store having maximum weekly sales

grpby_stores_sum = df_walmartAnalysis.groupby('Store')['Weekly_Sales'].sum()
print('Store {} has Max. Sales of ${}'.format(grpby_stores_sum.index[grpby_stores_sum == max(grpby_stores_sum)][0], max(grpby_stores_sum)))
```

```
Store 20 has Max. Sales of $301397792.46000004
```

Which store having maximum standard deviation? Also find coeficient of Mean to Std

In [9]:

```
grpby_stores_std = df_walmartAnalysis.groupby('Store')['Weekly_Sales'].std()
print('Store {} has Max. Standard Deviation of {}'.format(grpby_stores_std.index[grpby_stores_std == max(grpby_stores_std)][0], max(grpby_stores_std)))
```

```
Store 14 has Max. Standard Deviation of 317569.9494755081
```

In [10]:

```
mean_value = df_walmartAnalysis['Weekly_Sales'].mean()
std_value = df_walmartAnalysis['Weekly_Sales'].std()
print('Coefficient of Mean to Std: {}'.format(round(mean_value/std_value, 3)))
```

```
Coefficient of Mean to Std: 1.855
```

Which store/s has good quarterly growth rate in Q3'2012?

In [11]:

```
# extracting Year from Date and adding as a new column in dataframe
# extracting Month from Date and adding as a new column in dataframe
df_walmartQ3Analysis = df_walmartAnalysis.copy()
df1 = pd.DataFrame(df_walmartQ3Analysis['Date'].str.split('-').tolist(), columns=['Day',
'Month', 'Year'])
df1 = df1.astype(dtype=int)

df_walmartQ3Analysis = pd.concat([df_walmartQ3Analysis, df1], axis=1)
df_walmartQ3Analysis['Qtr'] = 'Q' + pd.to_datetime(df_walmartQ3Analysis['Month'], format
='%m').dt.quarter.astype(str)

df_walmartQ3Analysis.head()
```

Out[11]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year	Qtr
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106	5	2	2010	Q1
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106	12	2	2010	Q1
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106	19	2	2010	Q1
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106	26	2	2010	Q1
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106	5	3	2010	Q1

In [12]:

```
# extracting data for Q3 of 2012 i.e for the month of Aug, Sept and Oct ---- 7, 8, 9
growthRateQ3_2012= df_walmartQ3Analysis[(df_walmartQ3Analysis['Year'] == 2012) & (df_wal
martQ3Analysis['Qtr'] == 'Q3')]
growthRateQ3_2012.head()
```

Out[12]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year	Qtr
126	1	06-07-2012	1769854.16	0	81.57	3.227	221.883779	6.908	6	7	2012	Q3
127	1	13-07-2012	1527014.04	0	77.12	3.256	221.924158	6.908	13	7	2012	Q3
128	1	20-07-2012	1497954.76	0	80.42	3.311	221.932727	6.908	20	7	2012	Q3
129	1	27-07-2012	1439123.71	0	82.66	3.407	221.941295	6.908	27	7	2012	Q3
130	1	03-08-2012	1631135.79	0	86.11	3.417	221.949864	6.908	3	8	2012	Q3

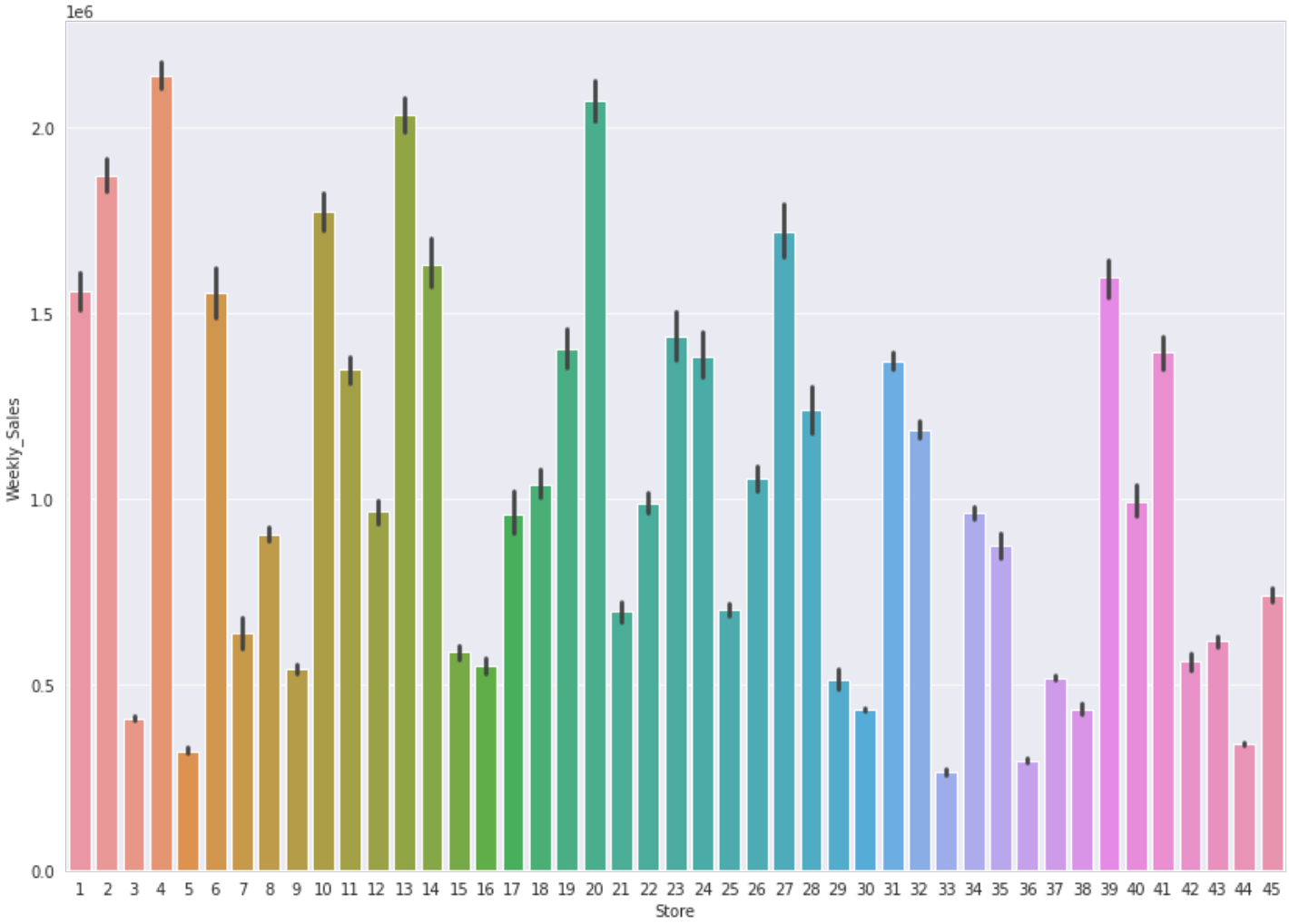
In [13]:

In [13]:

```
plt.figure(figsize=(14,10))
sns.barplot(x='Store', y='Weekly_Sales', data=growthRateQ3_2012)

# But from the plot below we can see that Store 4 procures highest growth rate in Q3 of 2012.
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b33e814a8>
```



Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together

In [14]:

```
# From this data, we can derive that out of total sales, around 5985 i.e. 93% sales occurs on Non Holiday week and remaining ~7% sales occurs on Holiday week
total_week_sales = df_walmartAnalysis.groupby('Holiday_Flag')
total_week_sales.size()
total_week_sales['Weekly_Sales'].describe()
```

Out[14]:

Holiday_Flag	count	mean	std	min	25%	50%	75%	max
0	5985.0	1.041256e+06	558957.436147	209986.25	551378.3900	956211.20	1414343.530	3818686.45
1	450.0	1.122888e+06	627684.931884	215359.21	575865.4825	1018538.04	1555213.175	3004702.33

In [15]:

```
# getting Non Holiday Sales
```

```
non_holiday_sales = total_week_sales.get_group(0)
print('Non Holiday Sales first 5 rows: \n')
non_holiday_sales.head()
```

```
Non Holiday Sales first 5 rows:
```

Out[15]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106

In [16]:

```
# getting Holiday Sales
holiday_sales = total_week_sales.get_group(1)
print('Holiday Sales first 5 rows: \n')
holiday_sales.head()
```

```
Holiday Sales first 5 rows:
```

Out[16]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
31	1	10-09-2010	1507460.69	1	78.69	2.565	211.495190	7.787
42	1	26-11-2010	1955624.11	1	64.52	2.735	211.748433	7.838
47	1	31-12-2010	1367320.01	1	48.43	2.943	211.404932	7.838
53	1	11-02-2011	1649614.93	1	36.39	3.022	212.936705	7.742

In [17]:

```
mean_non_holiday_sales = non_holiday_sales['Weekly_Sales'].mean()
print(mean_non_holiday_sales)
```

```
1041256.3802088564
```

In [0]:

```
df_whichHolidaySales = holiday_sales[holiday_sales['Weekly_Sales'] > mean_non_holiday_sal
es]
```

In [19]:

```
# Here is the required holiday sales data. Out of 450 holiday records, 220 records are th
ose where the sales is higher than the mean sales in non-holiday season for all stores to
gether.
df_whichHolidaySales
```

Out[19]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
31	1	10-09-2010	1507460.69	1	78.69	2.565	211.495190	7.787
42	1	26-11-2010	1955624.11	1	64.52	2.735	211.748433	7.838
47	1	31-12-2010	1367320.01	1	48.43	2.943	211.404932	7.838

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
53	1	11-02-2011	1649614.93	1	36.39	3.022	212.936705	7.742
83	1	09-09-2011	1540471.24	1	76.00	3.546	215.861056	7.962
94	1	25-11-2011	2033320.66	1	60.14	3.236	218.467621	7.866
99	1	30-12-2011	1497462.72	1	44.55	3.129	219.535990	7.866
105	1	10-02-2012	1802477.43	1	48.02	3.409	220.265178	7.348
135	1	07-09-2012	1661767.33	1	83.96	3.730	222.439015	6.908
144	2	12-02-2010	2137809.50	1	38.49	2.548	210.897994	8.324
174	2	10-09-2010	1839128.83	1	79.09	2.565	211.153210	8.099
185	2	26-11-2010	2658725.29	1	62.98	2.735	211.406287	8.163
190	2	31-12-2010	1750434.55	1	47.30	2.943	211.064774	8.163
196	2	11-02-2011	2168041.61	1	33.19	3.022	212.592862	8.028
226	2	09-09-2011	1748000.65	1	77.97	3.546	215.514829	7.852
237	2	25-11-2011	2614202.30	1	56.36	3.236	218.113027	7.441
242	2	30-12-2011	1874226.52	1	44.57	3.129	219.177306	7.441
248	2	10-02-2012	2103322.68	1	46.98	3.409	219.904907	7.057
278	2	07-09-2012	1898777.07	1	87.65	3.730	222.074763	6.565
430	4	12-02-2010	2188307.39	1	28.84	2.573	126.496258	8.623
460	4	10-09-2010	1865820.81	1	73.54	2.574	126.114581	7.372
471	4	26-11-2010	2789469.45	1	48.08	2.752	126.669267	7.127
476	4	31-12-2010	1794868.74	1	38.09	2.955	127.087677	7.127
482	4	11-02-2011	2187847.29	1	33.29	3.033	127.859129	6.510
512	4	09-09-2011	2093139.01	1	73.34	3.554	129.368613	5.644
523	4	25-11-2011	3004702.33	1	47.96	3.225	129.836400	5.143
528	4	30-12-2011	2007105.86	1	36.89	3.119	130.071032	5.143
534	4	10-02-2012	2374660.64	1	33.00	3.411	130.384903	4.607
564	4	07-09-2012	2125104.72	1	82.09	3.709	130.932548	4.077
716	6	12-02-2010	1606283.86	1	40.57	2.548	212.770042	7.259
746	6	10-09-2010	1424225.44	1	78.78	2.565	213.013312	6.973
757	6	26-11-2010	2267452.40	1	65.79	2.735	213.267296	7.007
762	6	31-12-2010	1464050.02	1	49.14	2.943	212.914967	7.007
768	6	11-02-2011	1486920.17	1	39.38	3.022	214.463094	6.858
798	6	09-09-2011	1483574.38	1	80.21	3.546	217.398030	6.925
809	6	25-11-2011	2249811.55	1	62.78	3.236	220.041741	6.551
814	6	30-12-2011	1598080.52	1	46.80	3.129	221.128263	6.551
820	6	10-02-2012	1620603.92	1	48.58	3.409	221.864499	6.132
850	6	07-09-2012	1608077.01	1	86.33	3.730	224.056008	5.668
1043	8	26-11-2010	1261693.16	1	51.07	2.735	215.107755	6.433
1095	8	25-11-2011	1235163.86	1	49.61	3.236	221.949157	6.123
1288	10	12-02-2010	2176028.52	1	49.96	2.828	126.496258	9.765
1318	10	10-09-2010	1720530.23	1	84.04	2.961	126.114581	9.199
1329	10	26-11-2010	2939946.38	1	55.33	3.162	126.669267	9.003
1334	10	31-12-2010	1707298.14	1	49.67	3.148	127.087677	9.003
1340	10	11-02-2011	2115408.31	1	51.51	3.381	127.859129	8.744
1370	10	09-09-2011	1670579.82	1	89.06	3.771	129.368613	8.257
1381	10	25-11-2011	2950198.64	1	60.68	3.760	129.836400	7.874

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
1386	10	30-12-2011	1930690.37	1	48.92	3.428	130.071032	7.874
1392	10	10-02-2012	2218595.80	1	55.73	3.722	130.384903	7.545
1422	10	07-09-2012	1708283.28	1	83.07	4.124	130.932548	7.170
1431	11	12-02-2010	1574684.08	1	48.01	2.548	214.574792	7.368
1461	11	10-09-2010	1231428.46	1	81.93	2.565	214.806543	7.346
1472	11	26-11-2010	1757242.51	1	69.90	2.735	215.061402	7.564
1477	11	31-12-2010	1172003.10	1	55.03	2.943	214.698647	7.564
1483	11	11-02-2011	1419236.90	1	44.61	3.022	216.266091	7.551
1513	11	09-09-2011	1249439.95	1	84.91	3.546	219.213531	7.567
1524	11	25-11-2011	1848953.48	1	70.03	3.236	221.901118	7.197
1529	11	30-12-2011	1352084.21	1	48.86	3.129	223.009084	7.197
1535	11	10-02-2012	1574287.76	1	52.23	3.409	223.753643	6.833
1565	11	07-09-2012	1304584.40	1	85.17	3.730	225.966026	6.334
1574	12	12-02-2010	1117863.33	1	47.87	2.946	126.496258	13.975
1615	12	26-11-2010	1601377.41	1	47.66	3.162	126.669267	14.313
1626	12	11-02-2011	1086421.57	1	51.30	3.381	127.859129	14.021
1667	12	25-11-2011	1591920.42	1	53.25	3.622	129.836400	12.890
1672	12	30-12-2011	1111638.07	1	44.64	3.428	130.071032	12.890
1678	12	10-02-2012	1199330.85	1	52.27	3.722	130.384903	12.187
1717	13	12-02-2010	2030933.46	1	33.16	2.671	126.496258	8.316
1747	13	10-09-2010	1772143.94	1	65.74	2.870	126.114581	7.951
1758	13	26-11-2010	2766400.05	1	28.22	2.830	126.669267	7.795
1763	13	31-12-2010	1675292.00	1	26.79	2.868	127.087677	7.795
1769	13	11-02-2011	1944438.90	1	30.83	3.034	127.859129	7.470
1799	13	09-09-2011	1872921.31	1	70.19	3.619	129.368613	6.877
1810	13	25-11-2011	2864170.61	1	38.89	3.445	129.836400	6.392
1815	13	30-12-2011	1969056.91	1	31.53	3.119	130.071032	6.392
1821	13	10-02-2012	2069284.57	1	33.73	3.116	130.384903	6.104
1851	13	07-09-2012	2165796.31	1	70.65	3.689	130.932548	5.765
1860	14	12-02-2010	1704218.84	1	27.73	2.773	181.982317	8.992
1890	14	10-09-2010	2191767.76	1	70.87	2.699	182.598178	8.743
1901	14	26-11-2010	2921709.71	1	46.15	3.039	182.783277	8.724
1906	14	31-12-2010	1623716.46	1	29.67	3.179	182.571448	8.724
1912	14	11-02-2011	1980405.03	1	30.30	3.239	183.701613	8.549
1942	14	09-09-2011	2202742.90	1	71.48	3.738	186.673738	8.625
1953	14	25-11-2011	2685351.81	1	48.71	3.492	188.350400	8.523
1958	14	30-12-2011	1914148.89	1	37.79	3.389	189.062016	8.523
1964	14	10-02-2012	2077256.24	1	37.00	3.640	189.707605	8.424
1994	14	07-09-2012	1904512.34	1	75.70	3.911	191.577676	8.684
2044	15	26-11-2010	1120018.92	1	40.71	3.186	132.836933	8.067
2096	15	25-11-2011	1066478.10	1	41.10	3.689	136.478800	7.866
2319	17	10-09-2010	1200888.28	1	56.28	2.870	126.114581	6.697
2371	17	09-09-2011	1161900.18	1	61.94	3.619	129.368613	6.745
2382	17	25-11-2011	1225700.28	1	32.81	3.445	129.836400	6.617
2423	17	07-09-2012	1255633.29	1	61.99	3.689	130.932548	5.936

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
2432	18	12-02-2010	1187880.70	1	26.41	2.771	131.586613	9.202
2473	18	26-11-2010	1653759.36	1	40.81	3.070	132.836933	9.331
2484	18	11-02-2011	1122053.58	1	24.30	3.255	133.260871	9.131
2525	18	25-11-2011	1624170.99	1	41.97	3.536	136.478800	8.471
2536	18	10-02-2012	1161615.51	1	32.83	3.655	137.166677	8.075
2566	18	07-09-2012	1083521.24	1	71.85	3.921	138.472936	8.535
2575	19	12-02-2010	1536549.95	1	23.22	2.940	131.586613	8.350
2605	19	10-09-2010	1591453.39	1	63.36	2.837	132.756452	8.099
2616	19	26-11-2010	1993367.83	1	42.62	3.186	132.836933	8.067
2621	19	31-12-2010	1275146.94	1	28.65	3.336	132.815032	8.067
2627	19	11-02-2011	1430851.11	1	21.79	3.416	133.260871	7.771
2657	19	09-09-2011	1566712.79	1	68.28	3.930	136.274581	7.806
2668	19	25-11-2011	1974646.78	1	42.75	3.689	136.478800	7.866
2673	19	30-12-2011	1405168.06	1	31.65	3.566	136.643258	7.866
2679	19	10-02-2012	1499496.67	1	32.61	3.826	137.166677	7.943
2709	19	07-09-2012	1497073.82	1	72.20	4.076	138.472936	8.193
2718	20	12-02-2010	2109107.90	1	22.12	2.773	204.385747	8.187
2748	20	10-09-2010	2014954.79	1	65.02	2.699	204.726683	7.527
2759	20	26-11-2010	2811634.04	1	46.66	3.039	204.962100	7.484
2764	20	31-12-2010	1799737.79	1	28.85	3.179	204.643227	7.484
2770	20	11-02-2011	2211388.14	1	25.38	3.239	206.076386	7.343
2800	20	09-09-2011	2050542.56	1	68.74	3.738	209.022556	7.274
2811	20	25-11-2011	2906233.25	1	46.38	3.492	211.412076	7.082
2816	20	30-12-2011	2043245.00	1	36.35	3.389	212.403576	7.082
2822	20	10-02-2012	2462978.28	1	33.47	3.640	213.118614	6.961
2852	20	07-09-2012	2080529.06	1	76.36	3.911	215.218957	7.280
2902	21	26-11-2010	1245628.61	1	62.96	2.735	211.406287	8.163
2954	21	25-11-2011	1219263.40	1	56.43	3.236	218.113027	7.441
3045	22	26-11-2010	1564502.26	1	44.61	3.070	136.689571	8.572
3097	22	25-11-2011	1535857.49	1	46.28	3.536	140.421786	7.706
3147	23	12-02-2010	1380892.08	1	18.75	2.771	131.586613	5.892
3177	23	10-09-2010	1272842.85	1	63.21	2.717	132.756452	5.326
3188	23	26-11-2010	2072685.05	1	34.95	3.070	132.836933	5.287
3193	23	31-12-2010	1169773.85	1	19.05	3.177	132.815032	5.287
3199	23	11-02-2011	1249786.40	1	21.52	3.255	133.260871	5.114
3229	23	09-09-2011	1423289.90	1	66.04	3.809	136.274581	4.584
3240	23	25-11-2011	2057059.53	1	35.23	3.536	136.478800	4.420
3245	23	30-12-2011	1213486.95	1	22.30	3.402	136.643258	4.420
3251	23	10-02-2012	1358444.07	1	26.60	3.655	137.166677	4.261
3281	23	07-09-2012	1427162.26	1	66.74	3.921	138.472936	4.156
3290	24	12-02-2010	1414107.10	1	25.94	2.940	131.586613	8.326
3320	24	10-09-2010	1474498.59	1	67.11	2.837	132.756452	8.117
3331	24	26-11-2010	1779276.51	1	41.92	3.186	132.836933	8.275
3336	24	31-12-2010	1208600.05	1	25.90	3.336	132.815032	8.275
3342	24	11-02-2011	1341240.62	1	26.51	3.416	133.260871	8.252

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
3372	24	09-09-2011	1527455.19	1	68.32	3.930	136.274581	8.358
3383	24	25-11-2011	1761235.67	1	41.83	3.689	136.478800	8.454
3388	24	30-12-2011	1363973.16	1	33.45	3.566	136.643258	8.454
3394	24	10-02-2012	1403460.87	1	33.82	3.826	137.166677	8.659
3424	24	07-09-2012	1477134.75	1	72.81	4.076	138.472936	8.953
3474	25	26-11-2010	1115240.61	1	43.43	3.039	204.962100	7.484
3526	25	25-11-2011	1116211.39	1	43.49	3.492	211.412076	7.082
3606	26	10-09-2010	1042226.30	1	54.82	2.717	132.756452	8.445
3617	26	26-11-2010	1286833.62	1	28.11	3.070	132.836933	8.149
3658	26	09-09-2011	1069710.97	1	60.98	3.809	136.274581	7.767
3669	26	25-11-2011	1282320.05	1	31.07	3.536	136.478800	7.598
3680	26	10-02-2012	1081005.64	1	23.89	3.655	137.166677	7.467
3710	26	07-09-2012	1081874.03	1	61.58	3.921	138.472936	7.405
3719	27	12-02-2010	1745362.72	1	29.81	2.940	135.411308	8.237
3749	27	10-09-2010	1913494.81	1	70.38	2.837	136.621208	7.982
3760	27	26-11-2010	2627910.75	1	46.67	3.186	136.689571	8.021
3765	27	31-12-2010	1440963.00	1	29.59	3.336	136.665265	8.021
3771	27	11-02-2011	1636224.77	1	30.45	3.416	137.137832	7.827
3801	27	09-09-2011	1911470.84	1	70.93	3.930	140.231017	7.850
3812	27	25-11-2011	2504400.71	1	47.88	3.689	140.421786	7.906
3817	27	30-12-2011	1650604.60	1	37.85	3.566	140.587450	7.906
3823	27	10-02-2012	1651605.35	1	37.86	3.826	141.119983	8.009
3853	27	07-09-2012	1840955.23	1	76.00	4.076	142.500303	8.239
3862	28	12-02-2010	1558968.49	1	47.87	2.946	126.496258	13.975
3892	28	10-09-2010	1246062.17	1	83.63	3.044	126.114581	14.180
3903	28	26-11-2010	1937033.50	1	47.66	3.162	126.669267	14.313
3908	28	31-12-2010	1090558.57	1	45.64	3.148	127.087677	14.313
3914	28	11-02-2011	1397301.38	1	51.30	3.381	127.859129	14.021
3944	28	09-09-2011	1310087.00	1	88.00	3.913	129.368613	13.503
3955	28	25-11-2011	1929738.27	1	53.25	3.622	129.836400	12.890
3960	28	30-12-2011	1270036.53	1	44.64	3.428	130.071032	12.890
3966	28	10-02-2012	1572966.15	1	52.27	3.722	130.384903	12.187
3996	28	07-09-2012	1469693.99	1	88.52	4.124	130.932548	10.926
4291	31	12-02-2010	1543947.23	1	37.77	2.548	210.897994	8.324
4321	31	10-09-2010	1308179.02	1	79.30	2.565	211.153210	8.099
4332	31	26-11-2010	1858856.06	1	62.96	2.735	211.406287	8.163
4337	31	31-12-2010	1198071.60	1	47.19	2.943	211.064774	8.163
4343	31	11-02-2011	1539230.32	1	34.61	3.022	212.592862	8.028
4373	31	09-09-2011	1376670.27	1	78.87	3.546	215.514829	7.852
4384	31	25-11-2011	1934099.65	1	56.43	3.236	218.113027	7.441
4389	31	30-12-2011	1355405.95	1	45.16	3.129	219.177306	7.441
4395	31	10-02-2012	1527688.58	1	46.52	3.409	219.904907	7.057
4425	31	07-09-2012	1358111.62	1	88.40	3.730	222.074763	6.565
4434	32	12-02-2010	1123566.12	1	28.09	2.572	189.464273	9.014
4475	32	26-11-2010	1634635.86	1	29.97	2.742	191.012180	9.137

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
4486	32	11-02-2011	1124357.20	1	18.51	3.037	191.857288	8.818
4516	32	09-09-2011	1128237.30	1	61.24	3.566	194.638785	8.622
4527	32	25-11-2011	1684468.66	1	40.22	3.424	195.770400	8.513
4532	32	30-12-2011	1102367.65	1	32.99	3.119	196.358610	8.513
4538	32	10-02-2012	1129422.86	1	23.34	3.103	196.919506	8.256
4568	32	07-09-2012	1126685.95	1	72.56	3.596	198.095048	7.872
4761	34	26-11-2010	1309476.68	1	41.13	2.752	126.669267	10.210
4813	34	25-11-2011	1345595.82	1	45.99	3.225	129.836400	10.148
4824	34	10-02-2012	1047658.09	1	36.70	3.411	130.384903	9.653
4863	35	12-02-2010	1168815.31	1	29.81	2.773	135.411308	9.262
4904	35	26-11-2010	1781866.98	1	46.67	3.039	136.689571	8.763
4956	35	25-11-2011	1733822.40	1	47.88	3.492	140.421786	8.745
5435	39	12-02-2010	1266229.07	1	44.58	2.548	209.997021	8.554
5465	39	10-09-2010	1279666.47	1	79.94	2.565	210.264116	8.360
5476	39	26-11-2010	2149355.20	1	67.75	2.735	210.515277	8.476
5481	39	31-12-2010	1230012.16	1	52.45	2.943	210.182398	8.476
5487	39	11-02-2011	1227893.89	1	40.34	3.022	211.698509	8.395
5517	39	09-09-2011	1429345.86	1	79.15	3.546	214.615538	8.177
5528	39	25-11-2011	2338832.40	1	66.36	3.236	217.181253	7.716
5533	39	30-12-2011	1537139.56	1	47.60	3.129	218.230236	7.716
5539	39	10-02-2012	1442988.44	1	52.89	3.409	218.955100	7.244
5569	39	07-09-2012	1609811.75	1	83.71	3.730	221.118114	6.623
5619	40	26-11-2010	1166142.85	1	32.94	3.070	132.836933	5.287
5671	40	25-11-2011	1230011.95	1	32.76	3.536	136.478800	4.420
5712	40	07-09-2012	1088248.40	1	65.06	3.921	138.472936	4.156
5721	41	12-02-2010	1075656.34	1	23.04	2.572	189.464273	7.541
5751	41	10-09-2010	1172672.27	1	63.30	2.780	190.395829	7.335
5762	41	26-11-2010	1866681.57	1	25.30	2.742	191.012180	7.508
5773	41	11-02-2011	1150003.36	1	16.81	3.037	191.857288	7.241
5803	41	09-09-2011	1280958.97	1	58.31	3.566	194.638785	6.901
5814	41	25-11-2011	1906713.35	1	36.37	3.424	195.770400	6.759
5819	41	30-12-2011	1264014.16	1	34.12	3.119	196.358610	6.759
5825	41	10-02-2012	1238844.56	1	22.00	3.103	196.919506	6.589
5855	41	07-09-2012	1392143.82	1	67.41	3.596	198.095048	6.432
6334	45	26-11-2010	1182500.16	1	46.15	3.039	182.783277	8.724
6386	45	25-11-2011	1170672.94	1	48.71	3.492	188.350400	8.523

In [20]:

```
# Holiday weeks ..
df_whichHolidaySales[['Date']]
```

Out[20]:

	Date
1	12-02-2010
31	10-09-2010

	Date
42	26-11-2010
47	31-12-2010
53	11-02-2011
83	09-09-2011
94	25-11-2011
99	30-12-2011
105	10-02-2012
135	07-09-2012
144	12-02-2010
174	10-09-2010
185	26-11-2010
190	31-12-2010
196	11-02-2011
226	09-09-2011
237	25-11-2011
242	30-12-2011
248	10-02-2012
278	07-09-2012
430	12-02-2010
460	10-09-2010
471	26-11-2010
476	31-12-2010
482	11-02-2011
512	09-09-2011
523	25-11-2011
528	30-12-2011
534	10-02-2012
564	07-09-2012
716	12-02-2010
746	10-09-2010
757	26-11-2010
762	31-12-2010
768	11-02-2011
798	09-09-2011
809	25-11-2011
814	30-12-2011
820	10-02-2012
850	07-09-2012
1043	26-11-2010
1095	25-11-2011
1288	12-02-2010
1318	10-09-2010
1329	26-11-2010
1334	31-12-2010
1340	11-02-2011

	Date
1370	09-09-2011
1381	25-11-2011
1386	30-12-2011
1392	10-02-2012
1422	07-09-2012
1431	12-02-2010
1461	10-09-2010
1472	26-11-2010
1477	31-12-2010
1483	11-02-2011
1513	09-09-2011
1524	25-11-2011
1529	30-12-2011
1535	10-02-2012
1565	07-09-2012
1574	12-02-2010
1615	26-11-2010
1626	11-02-2011
1667	25-11-2011
1672	30-12-2011
1678	10-02-2012
1717	12-02-2010
1747	10-09-2010
1758	26-11-2010
1763	31-12-2010
1769	11-02-2011
1799	09-09-2011
1810	25-11-2011
1815	30-12-2011
1821	10-02-2012
1851	07-09-2012
1860	12-02-2010
1890	10-09-2010
1901	26-11-2010
1906	31-12-2010
1912	11-02-2011
1942	09-09-2011
1953	25-11-2011
1958	30-12-2011
1964	10-02-2012
1994	07-09-2012
2044	26-11-2010
2096	25-11-2011
2319	10-09-2010
2371	09-09-2011
2383	25-11-2011

	Date
2382	25-11-2011
2423	07-09-2012
2432	12-02-2010
2473	26-11-2010
2484	11-02-2011
2525	25-11-2011
2536	10-02-2012
2566	07-09-2012
2575	12-02-2010
2605	10-09-2010
2616	26-11-2010
2621	31-12-2010
2627	11-02-2011
2657	09-09-2011
2668	25-11-2011
2673	30-12-2011
2679	10-02-2012
2709	07-09-2012
2718	12-02-2010
2748	10-09-2010
2759	26-11-2010
2764	31-12-2010
2770	11-02-2011
2800	09-09-2011
2811	25-11-2011
2816	30-12-2011
2822	10-02-2012
2852	07-09-2012
2902	26-11-2010
2954	25-11-2011
3045	26-11-2010
3097	25-11-2011
3147	12-02-2010
3177	10-09-2010
3188	26-11-2010
3193	31-12-2010
3199	11-02-2011
3229	09-09-2011
3240	25-11-2011
3245	30-12-2011
3251	10-02-2012
3281	07-09-2012
3290	12-02-2010
3320	10-09-2010
3331	26-11-2010
3338	31-12-2010

	Date
3336	31-12-2010
3342	11-02-2011
3372	09-09-2011
3383	25-11-2011
3388	30-12-2011
3394	10-02-2012
3424	07-09-2012
3474	26-11-2010
3526	25-11-2011
3606	10-09-2010
3617	26-11-2010
3658	09-09-2011
3669	25-11-2011
3680	10-02-2012
3710	07-09-2012
3719	12-02-2010
3749	10-09-2010
3760	26-11-2010
3765	31-12-2010
3771	11-02-2011
3801	09-09-2011
3812	25-11-2011
3817	30-12-2011
3823	10-02-2012
3853	07-09-2012
3862	12-02-2010
3892	10-09-2010
3903	26-11-2010
3908	31-12-2010
3914	11-02-2011
3944	09-09-2011
3955	25-11-2011
3960	30-12-2011
3966	10-02-2012
3996	07-09-2012
4291	12-02-2010
4321	10-09-2010
4332	26-11-2010
4337	31-12-2010
4343	11-02-2011
4373	09-09-2011
4384	25-11-2011
4389	30-12-2011
4395	10-02-2012
4425	07-09-2012
4431	12-02-2010

	Date
4434	12-02-2010
4475	26-11-2010
4486	11-02-2011
4516	09-09-2011
4527	25-11-2011
4532	30-12-2011
4538	10-02-2012
4568	07-09-2012
4761	26-11-2010
4813	25-11-2011
4824	10-02-2012
4863	12-02-2010
4904	26-11-2010
4956	25-11-2011
5435	12-02-2010
5465	10-09-2010
5476	26-11-2010
5481	31-12-2010
5487	11-02-2011
5517	09-09-2011
5528	25-11-2011
5533	30-12-2011
5539	10-02-2012
5569	07-09-2012
5619	26-11-2010
5671	25-11-2011
5712	07-09-2012
5721	12-02-2010
5751	10-09-2010
5762	26-11-2010
5773	11-02-2011
5803	09-09-2011
5814	25-11-2011
5819	30-12-2011
5825	10-02-2012
5855	07-09-2012
6334	26-11-2010
6386	25-11-2011

Provide a monthly and semester view of sales in units and give insights

In [0]:

```
# extracting Month from Date and adding as a new column in dataframe
df_walmartSalesAnalysis = df_walmartAnalysis.copy()
df1 = pd.DataFrame(df_walmartSalesAnalysis['Date'].str.split('-').tolist(), columns=['Da
```

```
y', 'Month', 'Year'])
df1 = df1.astype(dtype=int)

df_walmartSalesAnalysis = pd.concat([df_walmartSalesAnalysis, df1], axis=1)
```

In [22]:

```
df_walmartSalesAnalysis.sort_values(by=['Store', 'Year'], inplace=True)
df_walmartSalesAnalysis.head(10)
```

Out[22]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106	5	2	2010
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106	12	2	2010
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106	19	2	2010
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106	26	2	2010
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106	5	3	2010
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106	12	3	2010
6	1	19-03-2010	1472515.79	0	54.58	2.720	211.215635	8.106	19	3	2010
7	1	26-03-2010	1404429.92	0	51.45	2.732	211.018042	8.106	26	3	2010
8	1	02-04-2010	1594968.28	0	62.27	2.719	210.820450	7.808	2	4	2010
9	1	09-04-2010	1545418.53	0	65.86	2.770	210.622857	7.808	9	4	2010

In [0]:

```
# Function to return monthwise Sales Growth Results

def month_wise_sales(year):
  all_stores = df_walmartSalesAnalysis['Store'].unique().tolist()
  number_of_columns= 4
  number_of_rows = len(all_stores)/(number_of_columns)
  fig=plt.figure(figsize=(5*number_of_columns, 5*number_of_rows+3))
  fig.suptitle('Sales Growth Result for Year {}'.format(year), fontsize=22, color='blue'
)


  for i, store in enumerate(all_stores):
    all_months = df_walmartSalesAnalysis[(df_walmartSalesAnalysis['Year'] == year) & (df
_walmartSalesAnalysis['Store'] == store)]['Month'].unique()
    salesGrowthRate_2010 = df_walmartSalesAnalysis[(df_walmartSalesAnalysis['Year'] == y
ear) & ((df_walmartSalesAnalysis.Month.isin(all_months))) & (df_walmartSalesAnalysis['Sto
re'] == store)]

    ax = plt.subplot(number_of_rows + 1, number_of_columns, i+1)
    ax.set_title('Store {} Sales'.format(store), fontsize=14, pad=10, color='red')
    sns.barplot(x='Month', y='Weekly_Sales', data=salesGrowthRate_2010, ax=ax)

  plt.tight_layout()
  fig.subplots_adjust(wspace=0.3, hspace=0.35, top=0.96, bottom=0.1)

  return ax
```

In [24]:

```
# Sales Growth Results for Year 2011

month_wise_sales(2010)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b2f7c96a0>
```


Store 1 Sales
Store 2 Sales
Store 3 Sales
Store 4 Sales
Store 5 Sales
Store 6 Sales
Store 7 Sales
Store 8 Sales
Store 9 Sales
Store 10 Sales
Store 11 Sales
Store 12 Sales
Store 13 Sales
Store 14 Sales
Store 15 Sales
Store 16 Sales
Store 17 Sales
Store 18 Sales
Store 19 Sales
Store 20 Sales
Store 21 Sales
Store 22 Sales
Store 23 Sales
Store 24 Sales
Store 25 Sales
Store 26 Sales
Store 27 Sales
Store 28 Sales
Store 29 Sales
Store 30 Sales
Store 31 Sales
Store 32 Sales
Weekly_Sales
Month

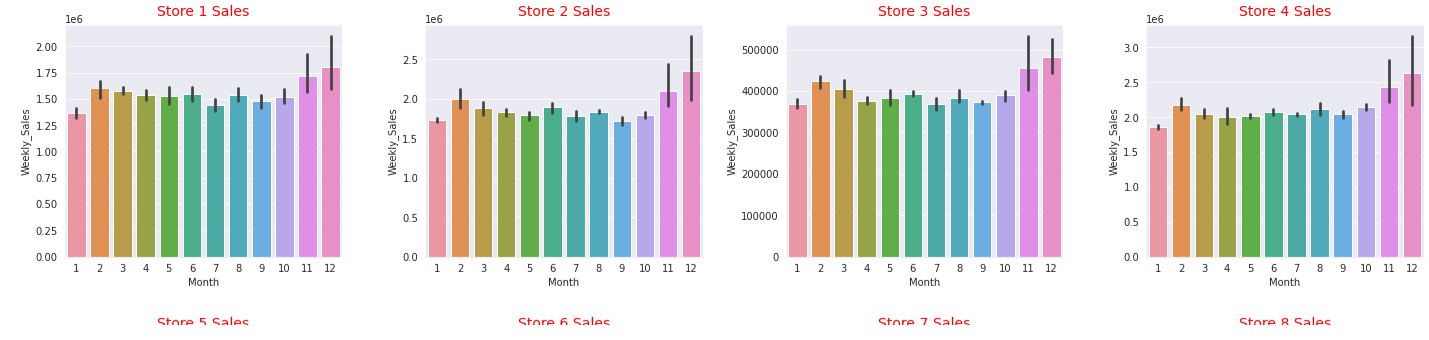
In [25]:

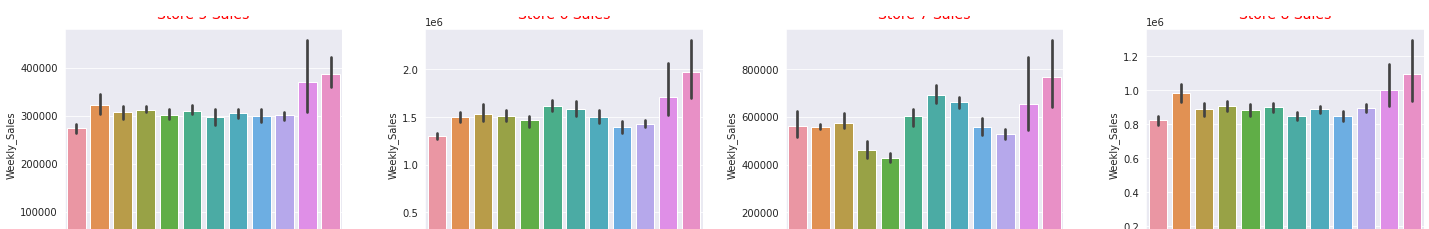
```
# Sales Growth Results for Year 2011

month_wise_sales(2011)
```

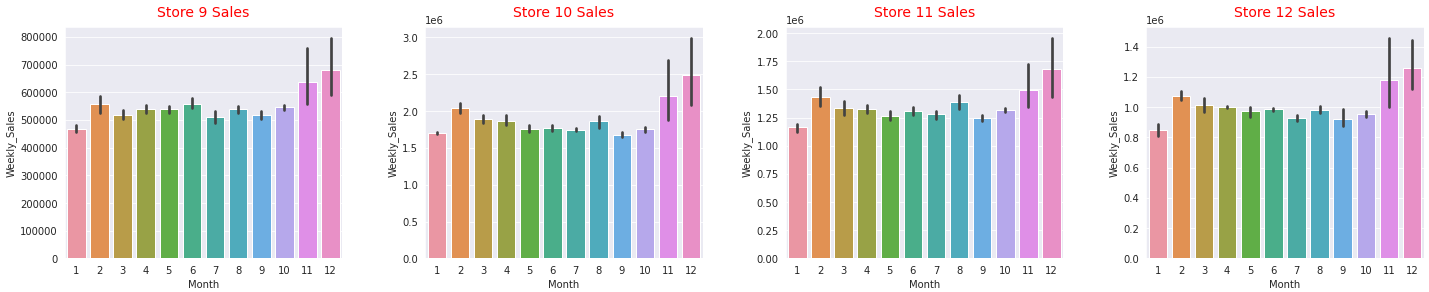
Out[25]:

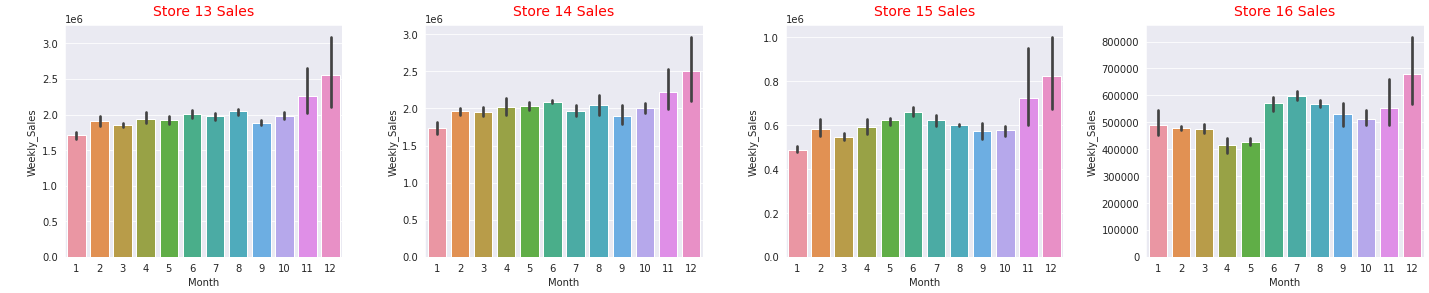
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b2b4542e8>
```

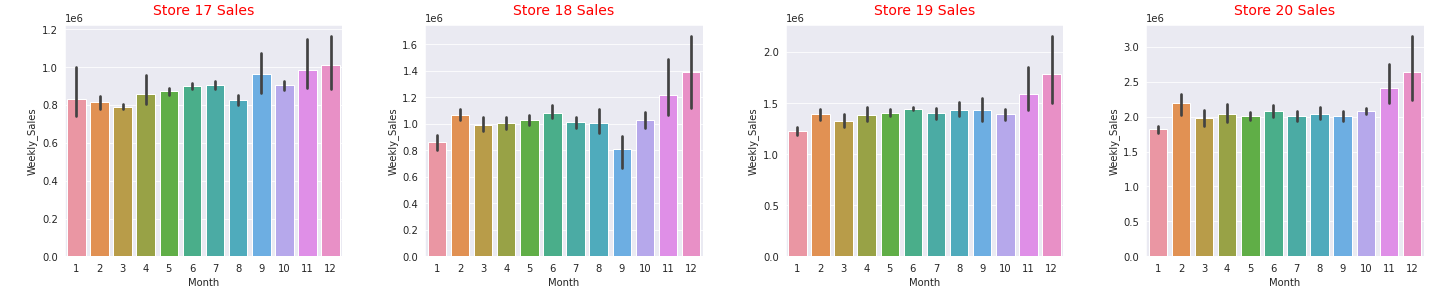
Sales Growth Result for Year 2011

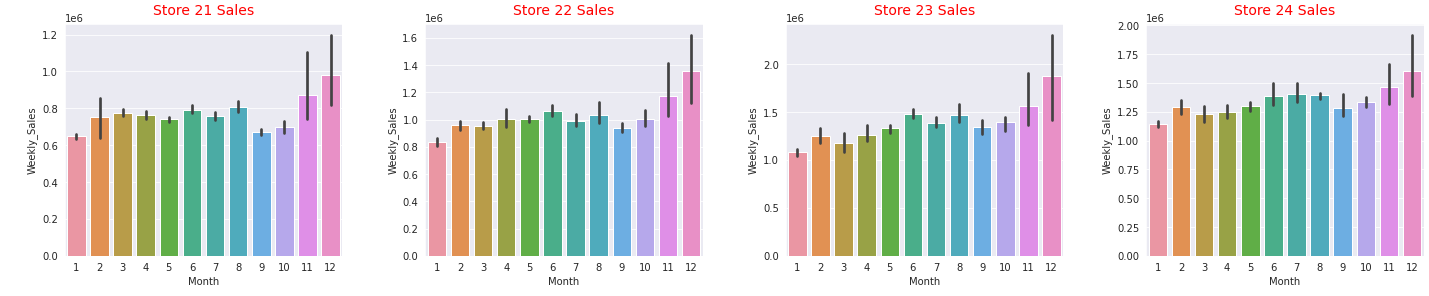


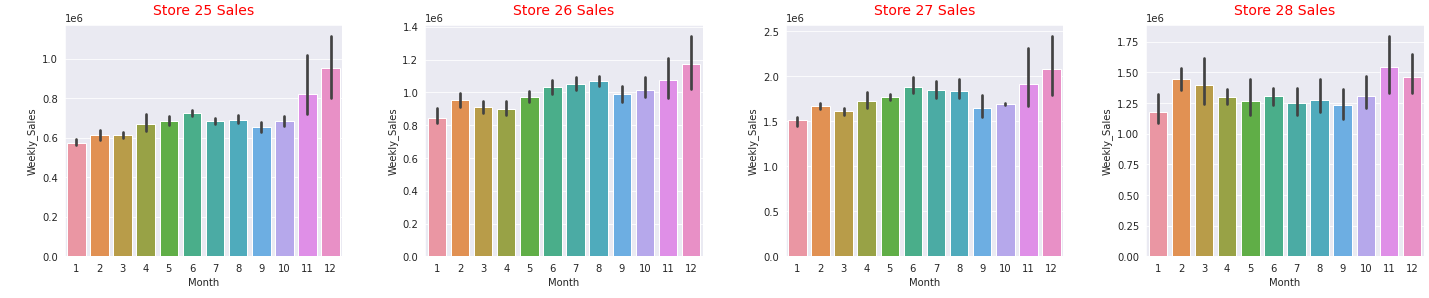

Store 5 Sales
Store 6 Sales
Store 7 Sales
Store 8 Sales

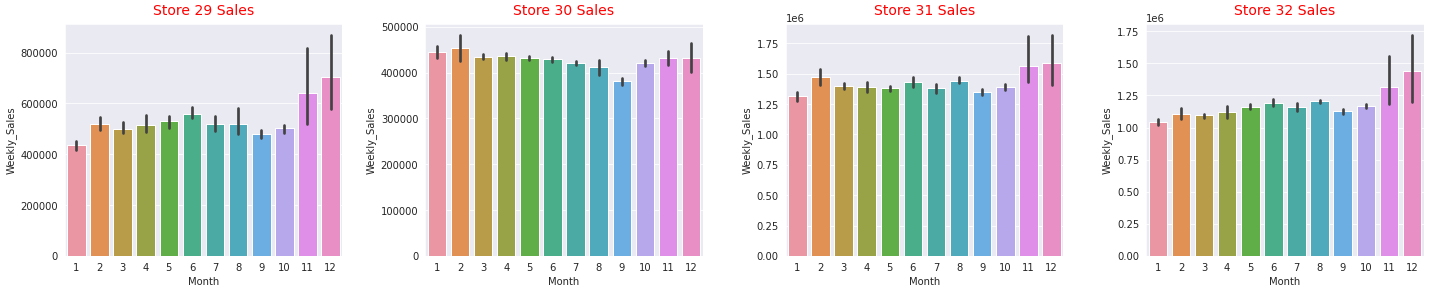

Store 9 Sales
Store 10 Sales
Store 11 Sales
Store 12 Sales


Store 13 Sales
Store 14 Sales
Store 15 Sales
Store 16 Sales

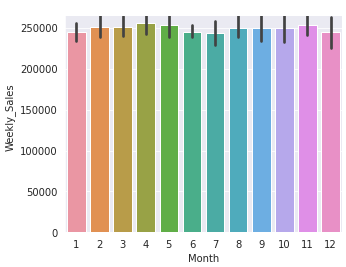
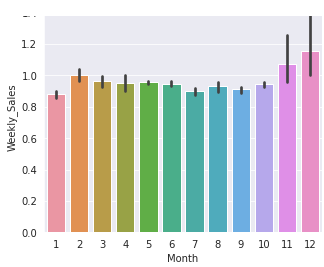
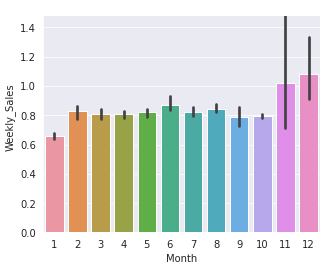
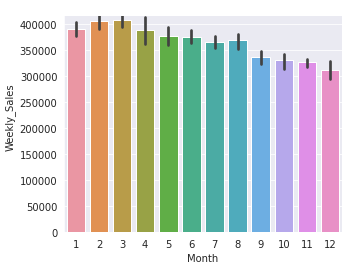
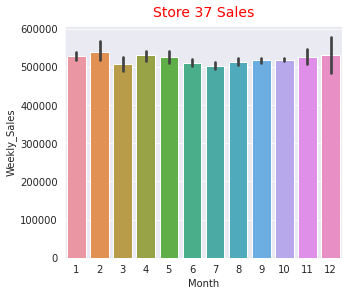

Store 17 Sales
Store 18 Sales
Store 19 Sales
Store 20 Sales

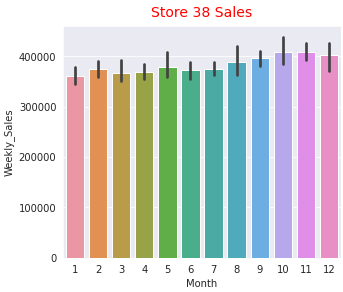

Store 21 Sales
Store 22 Sales
Store 23 Sales
Store 24 Sales

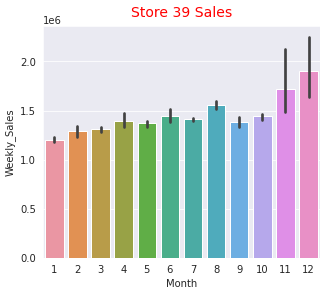

Store 25 Sales
Store 26 Sales
Store 27 Sales
Store 28 Sales

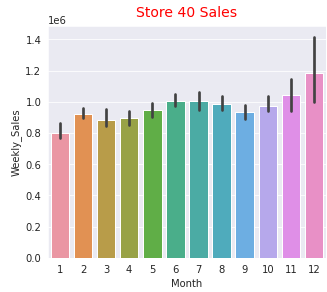

Store 29 Sales
Store 30 Sales
Store 31 Sales
Store 32 Sales

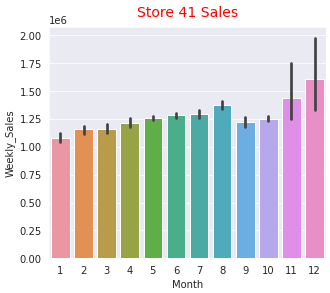

Store 33 Sales
Store 34 Sales
Store 35 Sales
Store 36 Sales

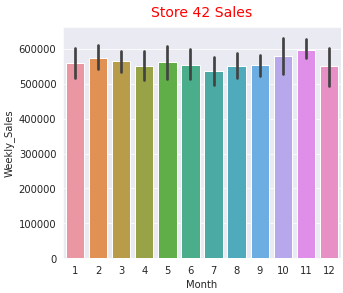






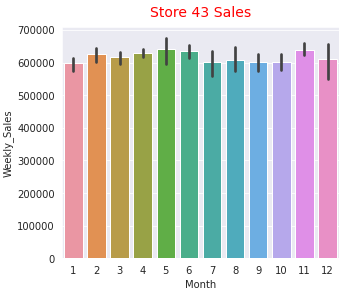


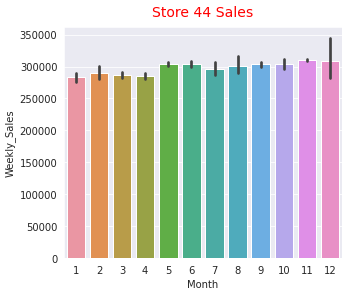


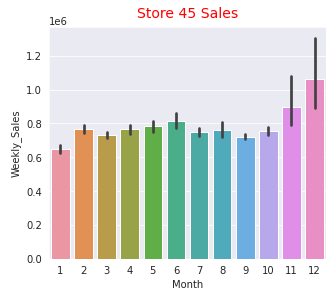










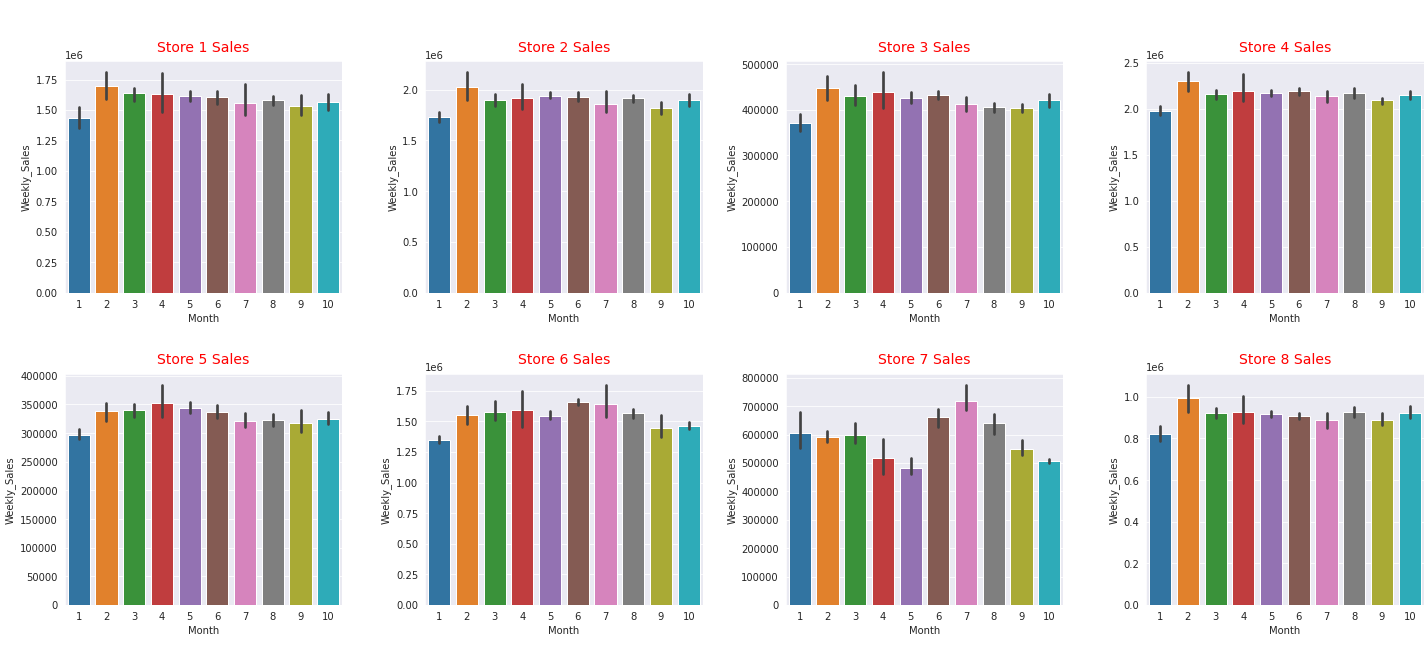


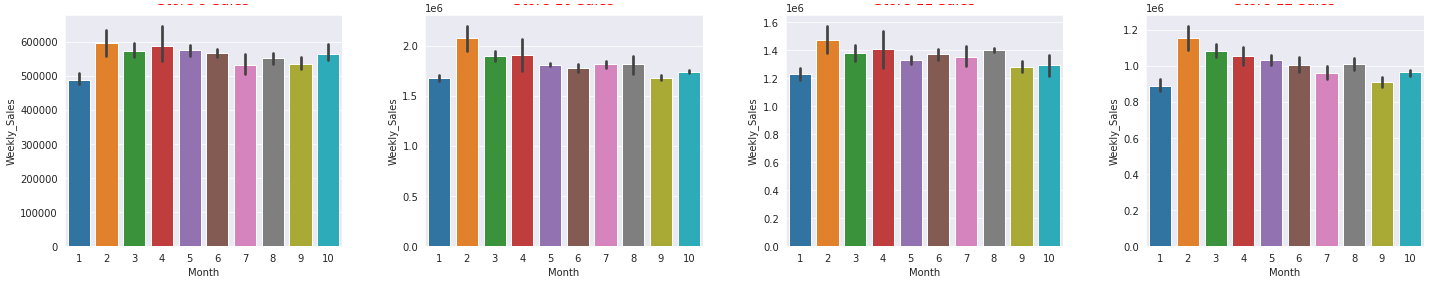
In [26]:

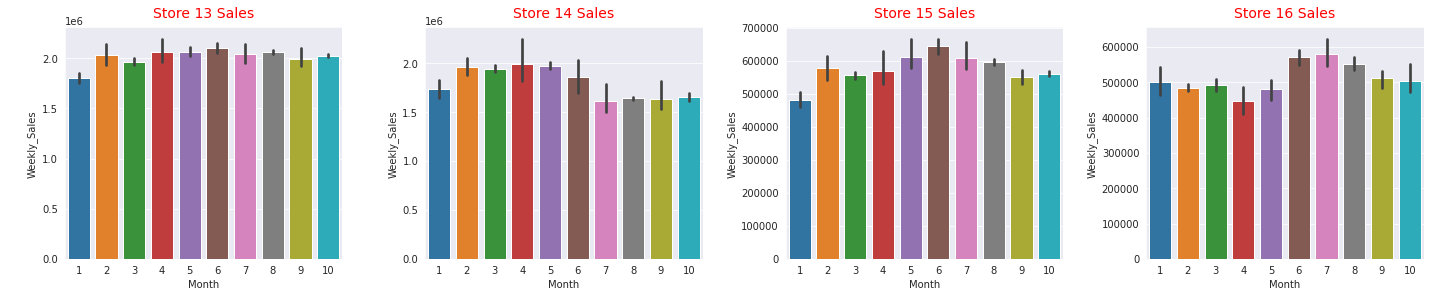
```
# Sales Growth Results for Year 2012

month_wise_sales(2012)
```

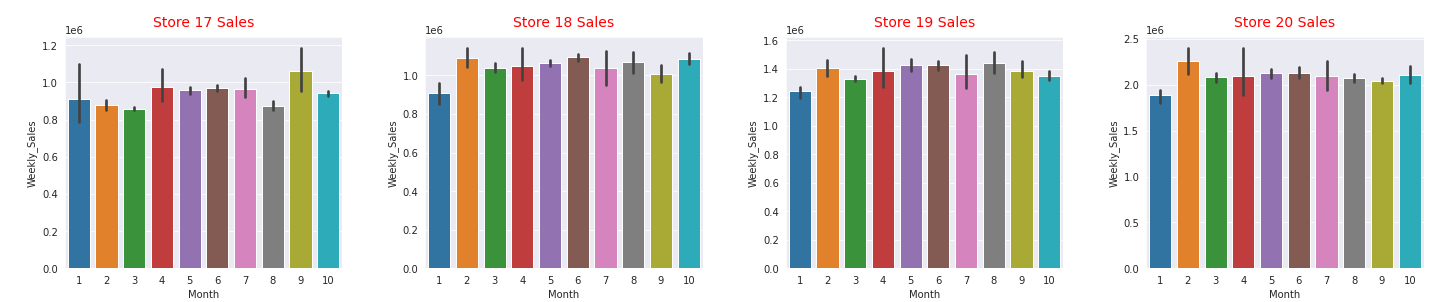
Out[26]:

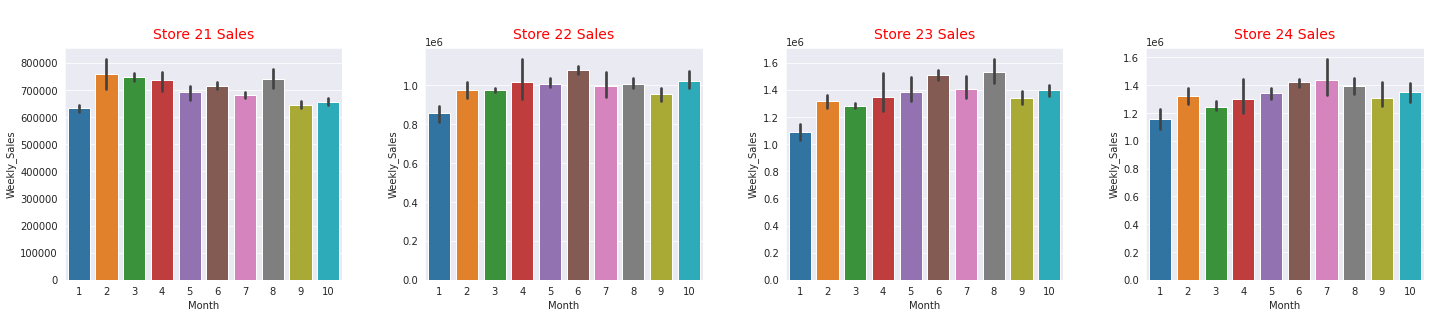
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b29b0f4e0>
```

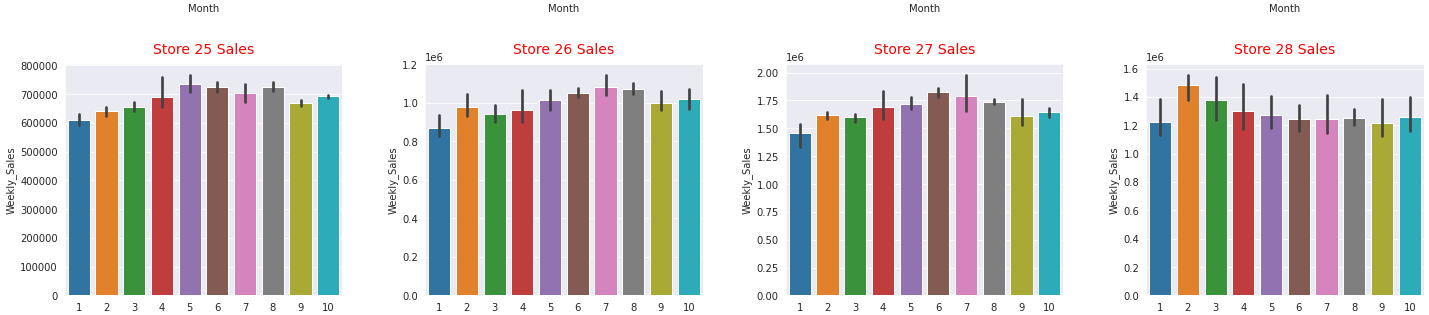


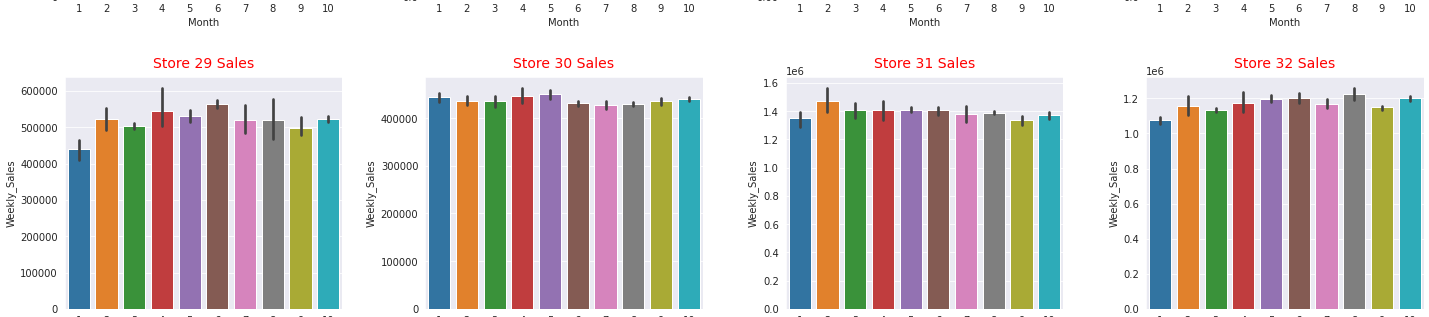

Store 13 Sales
Store 14 Sales
Store 15 Sales
Store 16 Sales
Weekly_Sales
Month

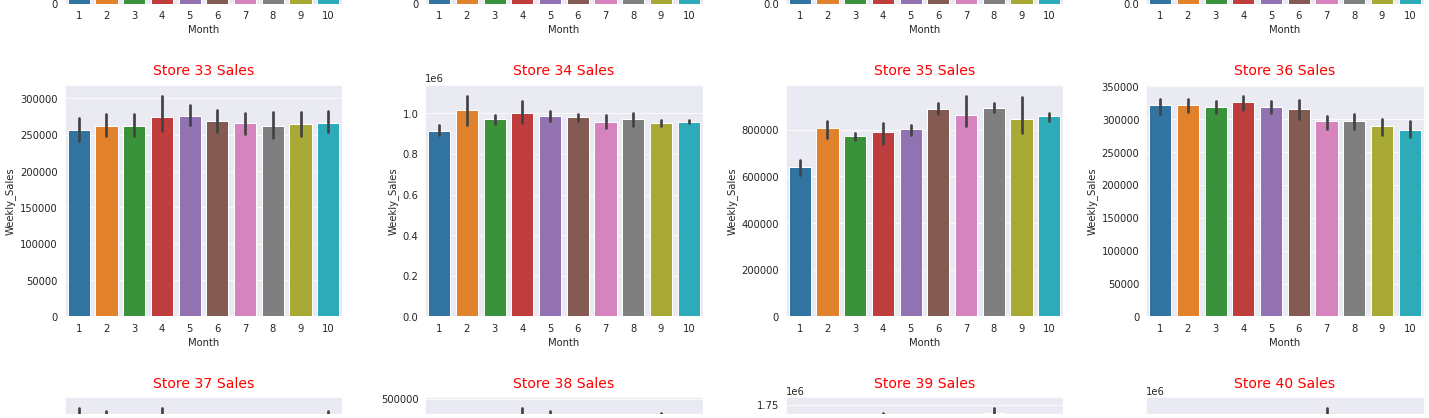

Store 17 Sales
Store 18 Sales
Store 19 Sales
Store 20 Sales
Weekly_Sales
Month

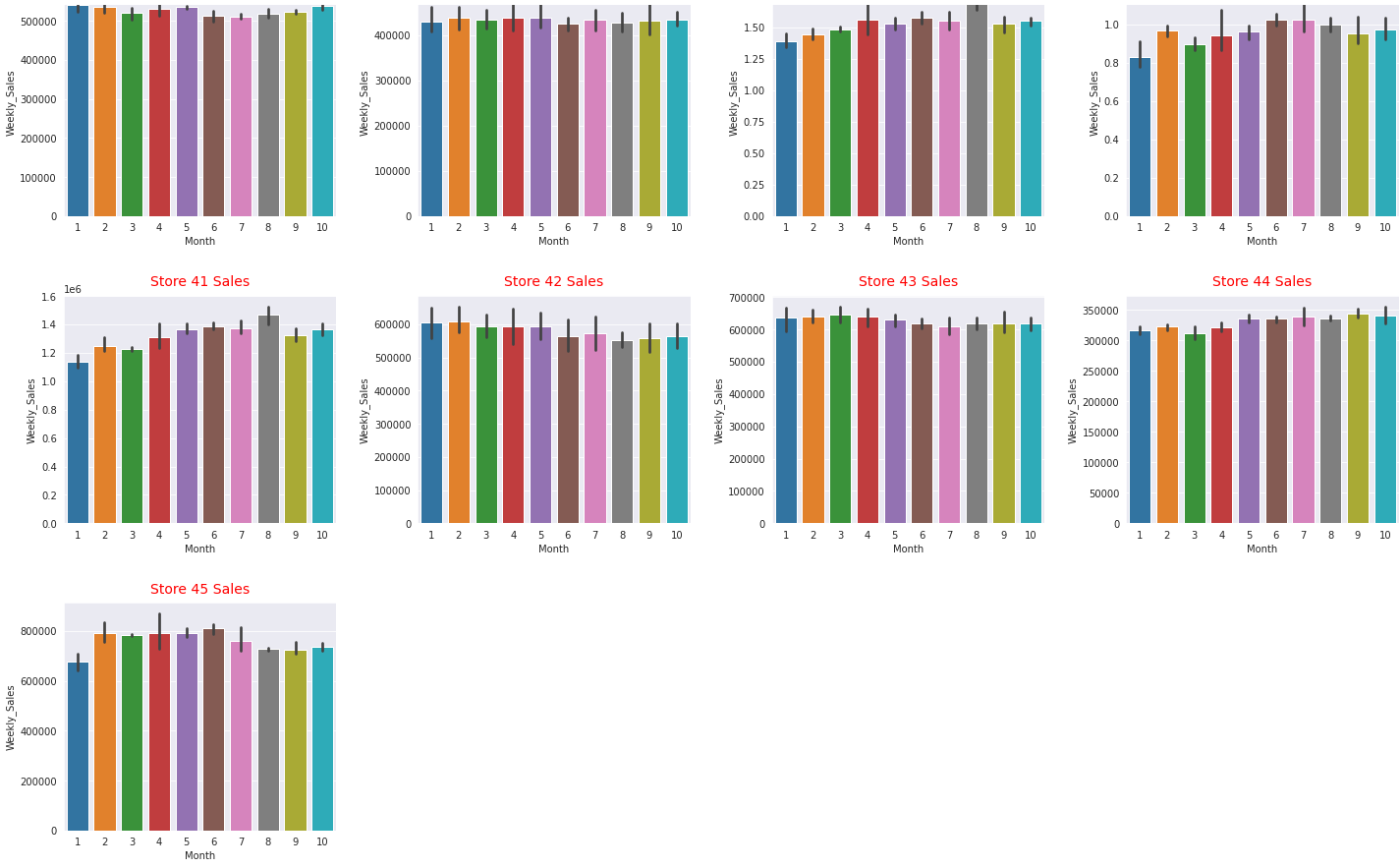

Store 21 Sales
Store 22 Sales
Store 23 Sales
Store 24 Sales
Weekly_Sales
Month


Store 25 Sales
Store 26 Sales
Store 27 Sales
Store 28 Sales
Weekly_Sales
Month


Store 29 Sales
Store 30 Sales
Store 31 Sales
Store 32 Sales
Weekly_Sales
Month


Store 33 Sales
Store 34 Sales
Store 35 Sales
Store 36 Sales
Weekly_Sales
Month


Store 37 Sales
Store 38 Sales
Store 39 Sales
Store 40 Sales
Weekly_Sales
Month



In [27]:

```
# Semester(Quarter) Wise Data....Converting months to semesters...

df_walmartSemesterSalesAnalysis = df_walmartSalesAnalysis.copy()
df_walmartSemesterSalesAnalysis['Semester'] = 'Q' + pd.to_datetime(df_walmartSalesAnalys
is['Month'], format='%m').dt.quarter.astype(str)

df_walmartSemesterSalesAnalysis.head()
```

Out[27]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year	Semest
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106	5	2	2010	Q
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106	12	2	2010	Q
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106	19	2	2010	Q
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106	26	2	2010	Q
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106	5	3	2010	Q

In [0]:

```
# Function to return Semester Wise Sales Growth Results for Year 2010, 2011, 2012

def semester_wise_sales(year):
  all_stores = df_walmartSemesterSalesAnalysis['Store'].unique().tolist()

  number_of_columns= 4
  number_of_rows = len(all_stores)/(number_of_columns)
```

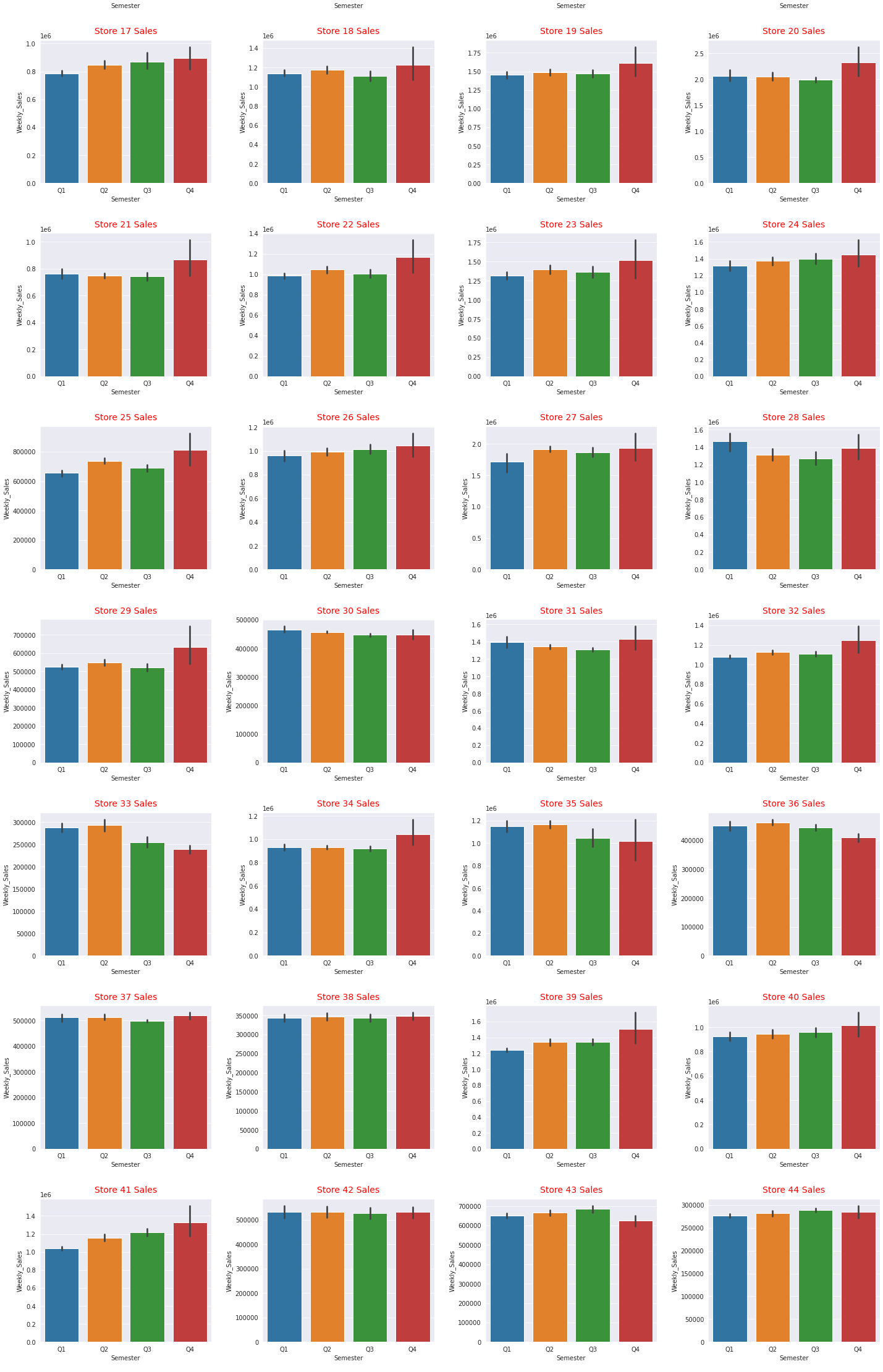
```
  fig=plt.figure(figsize=(5*number_of_columns, 5*number_of_rows+3))
  fig.suptitle('Semester wise Sales Growth Result for Year {}'.format(year), fontsize=22
, color='blue')


  for i, store in enumerate(all_stores):
    all_semesters = df_walmartSemesterSalesAnalysis[(df_walmartSemesterSalesAnalysis['Yea
r'] == year) & (df_walmartSemesterSalesAnalysis['Store'] == store)]['Semester'].unique()
    salesGrowthRate = df_walmartSemesterSalesAnalysis[(df_walmartSemesterSalesAnalysis['
Year'] == year) & ((df_walmartSemesterSalesAnalysis.Semester.isin(all_semesters))) & (df_
walmartSemesterSalesAnalysis['Store'] == store)]

    ax = plt.subplot(number_of_rows + 1, number_of_columns, i+1)
    ax.set_title('Store {} Sales'.format(store), fontsize=14, pad=10, color='red')
    sns.barplot(x='Semester', y='Weekly_Sales', data=salesGrowthRate, ax=ax)

  plt.tight_layout()
  fig.subplots_adjust(wspace=0.3, hspace=0.35, top=0.96, bottom=0.1)

  return ax
```

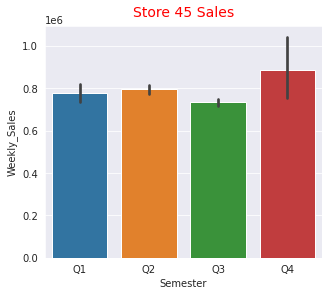
In [29]:

```
# Semester Wise Sales Growth Results for Year 2010

semester_wise_sales(2010)
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b288cfda0>
```


Store 17 Sales
Store 18 Sales
Store 19 Sales
Store 20 Sales
Store 21 Sales
Store 22 Sales
Store 23 Sales
Store 24 Sales
Store 25 Sales
Store 26 Sales
Store 27 Sales
Store 28 Sales
Store 29 Sales
Store 30 Sales
Store 31 Sales
Store 32 Sales
Store 33 Sales
Store 34 Sales
Store 35 Sales
Store 36 Sales
Store 37 Sales
Store 38 Sales
Store 39 Sales
Store 40 Sales
Store 41 Sales
Store 42 Sales
Store 43 Sales
Store 44 Sales
Weekly_Sales
Semester
Q1
Q2
Q3
Q4
1e6



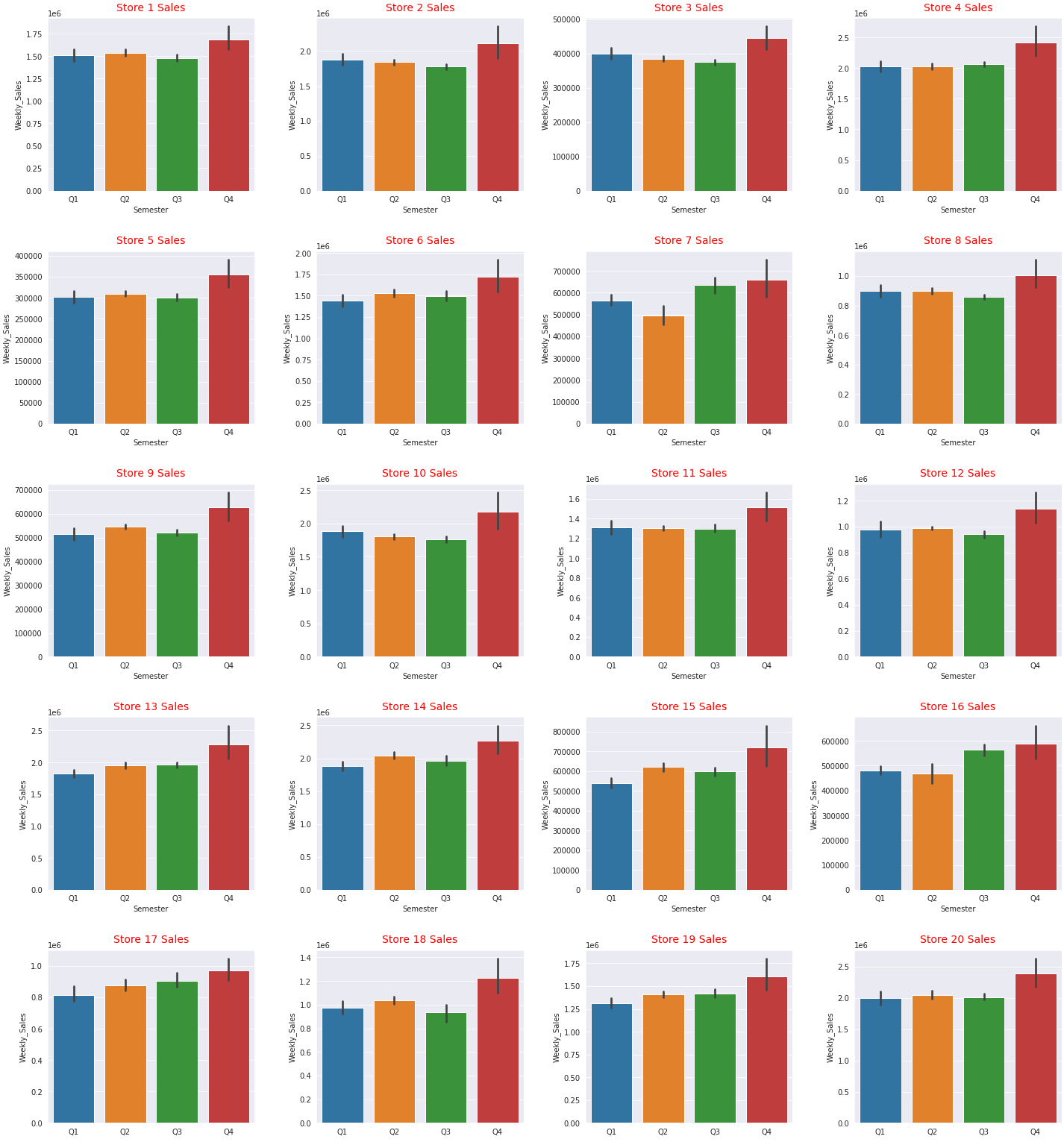
In [30]:

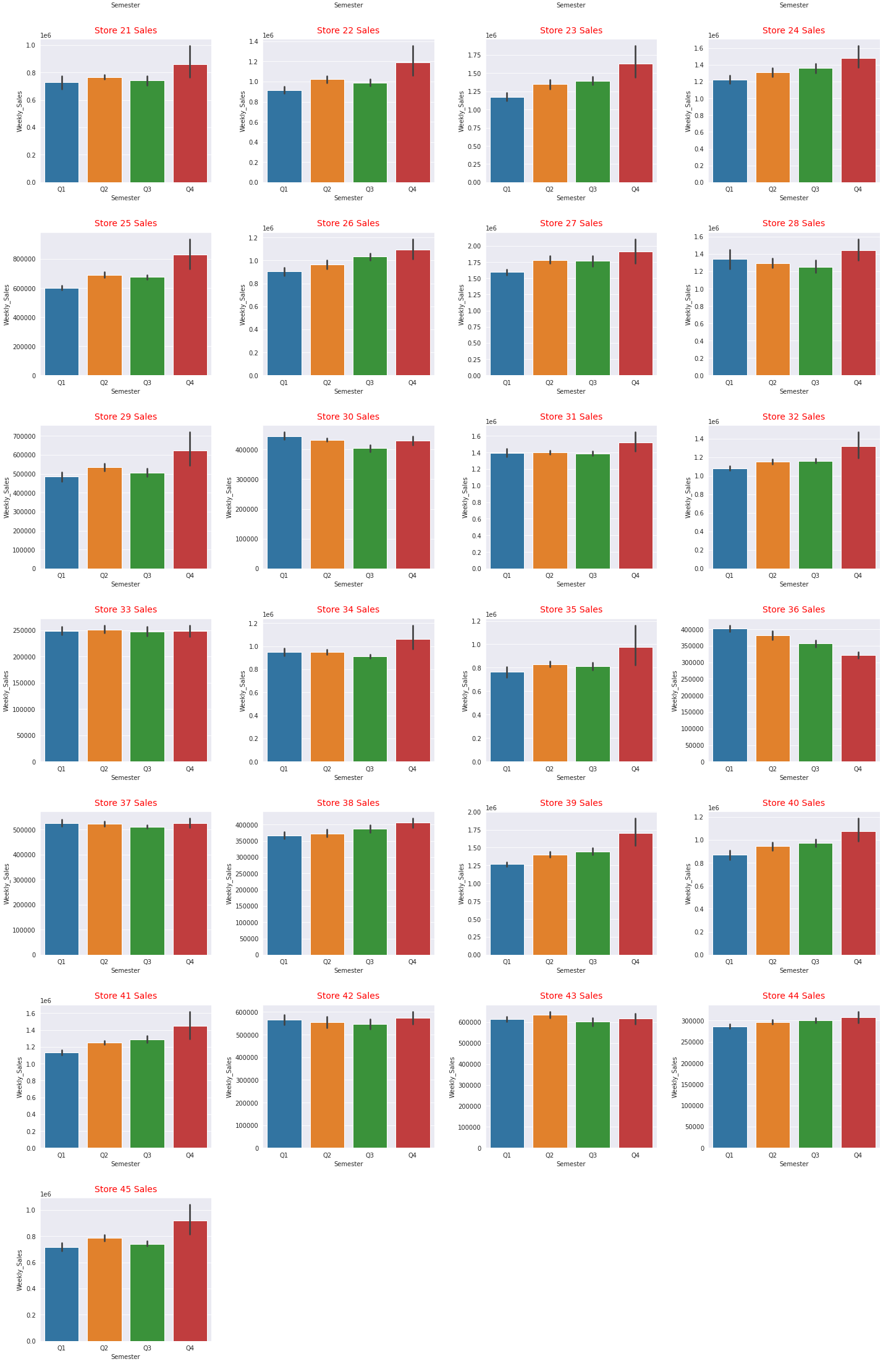
```
# Semester Wise Sales Growth Results for Year 2011

semester_wise_sales(2011)
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b2781abe0>
```

Semester wise Sales Growth Result for Year 2011



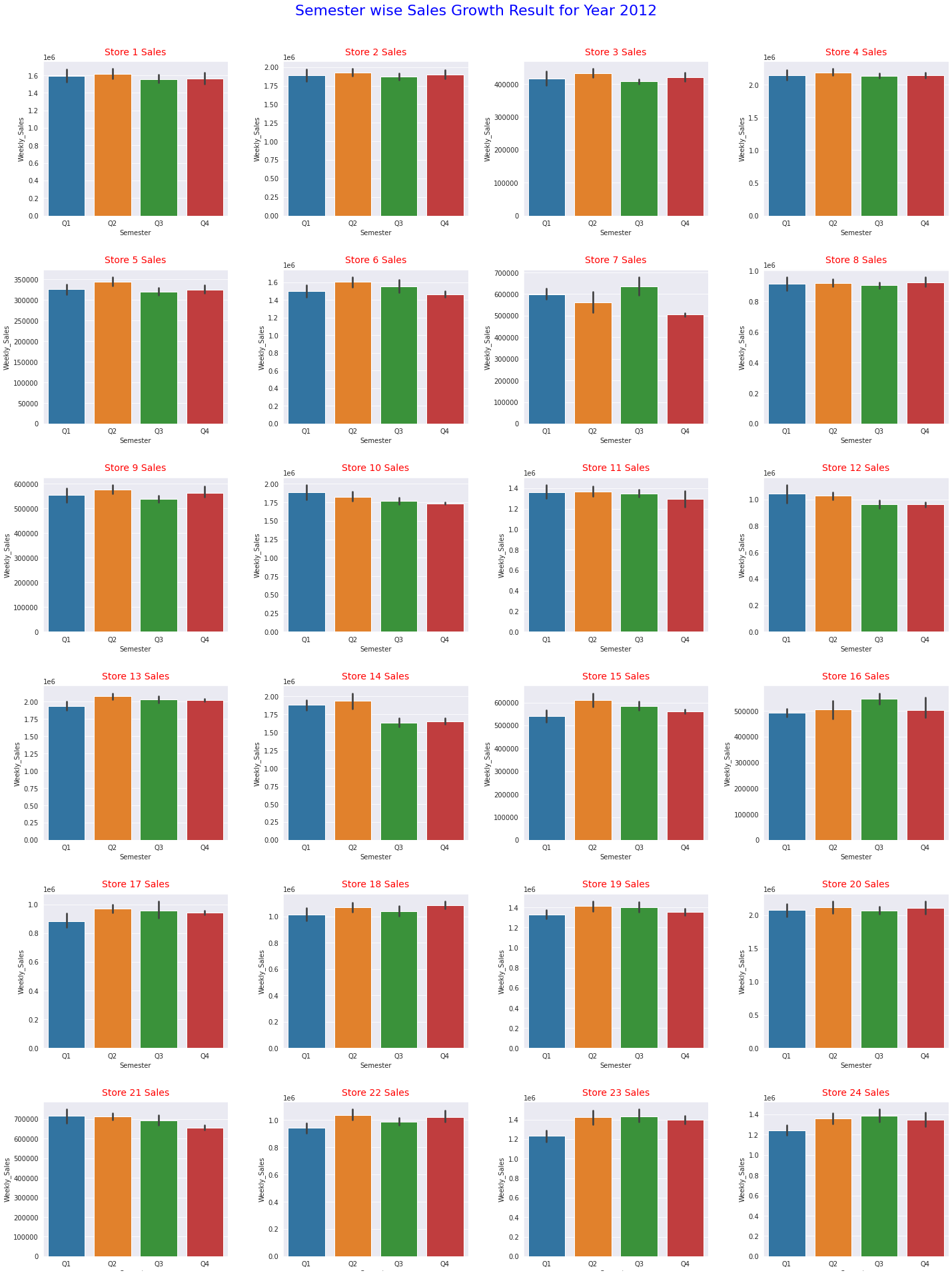

Store 21 Sales
Store 22 Sales
Store 23 Sales
Store 24 Sales
Store 25 Sales
Store 26 Sales
Store 27 Sales
Store 28 Sales
Store 29 Sales
Store 30 Sales
Store 31 Sales
Store 32 Sales
Store 33 Sales
Store 34 Sales
Store 35 Sales
Store 36 Sales
Store 37 Sales
Store 38 Sales
Store 39 Sales
Store 40 Sales
Store 41 Sales
Store 42 Sales
Store 43 Sales
Store 44 Sales
Store 45 Sales
Weekly_Sales
Semester
Q1
Q2
Q3
Q4
1e6

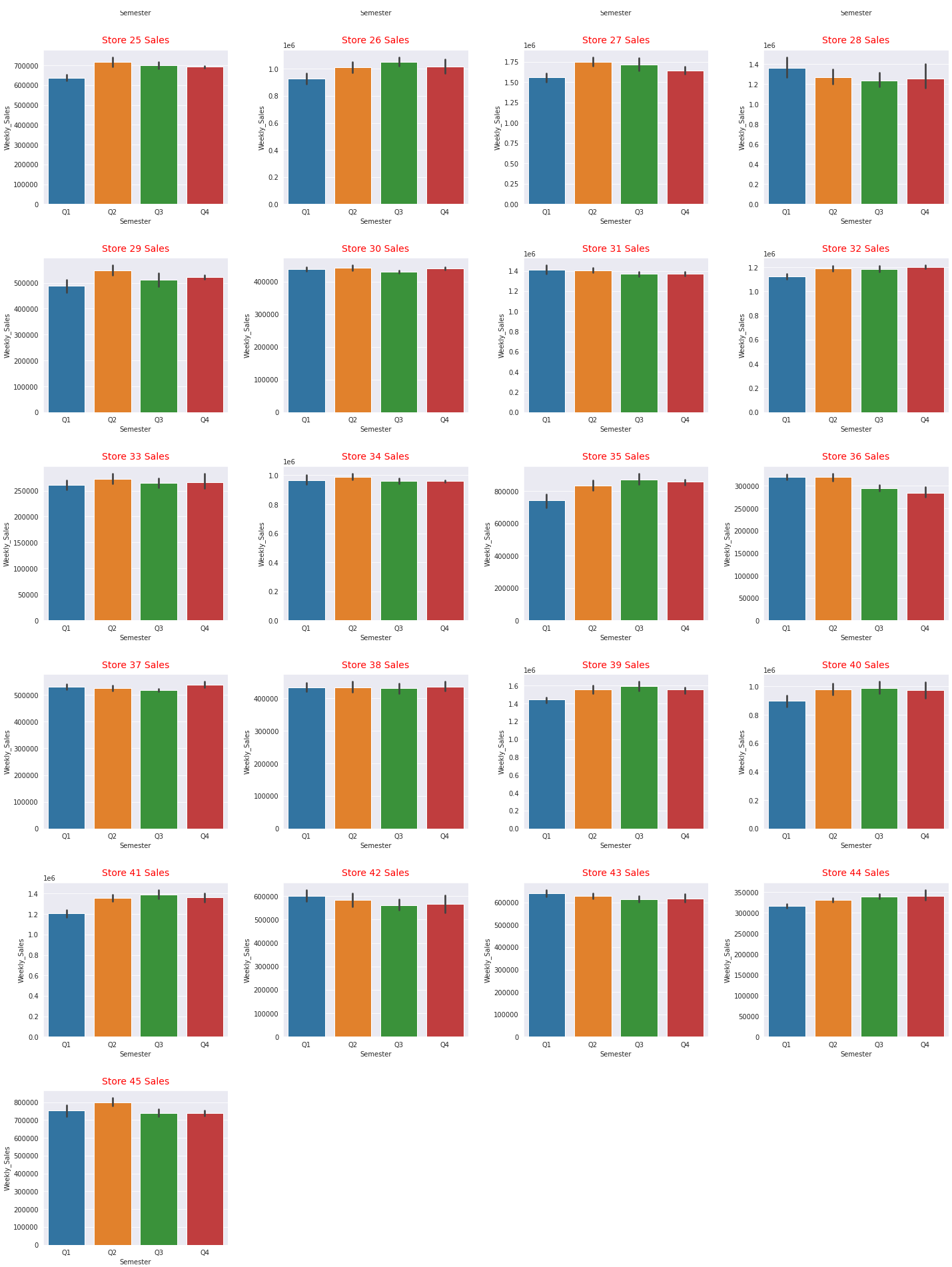
In [31]:

```
# Semester Wise Sales Growth Results for Year 2012

semester_wise_sales(2012)
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b266dd160>
```





Statistical Model

In [0]:

```
df_walmartAnalysisModel = df_walmartAnalysis.copy()

# change Dates into Days
```

```
df_walmartAnalysisModel['Total_Days'] = pd.Series(delta.days for delta in df_walmartAnal
ysisModel['Date'].apply(lambda x: pd.Timestamp(2020, 6, 4) - pd.Timestamp(x)).tolist())


df_dummyFeaturesModel = pd.get_dummies(df_walmartAnalysisModel, columns=['Store'])
```

In [91]:

```
from sklearn.model_selection import train_test_split

# Unemployment --- feature contribution to Weekly Sales is significant as it will proport
ionally affect the Store sales.
# On the other hand --- Fuel Price, CPI, Temperature don't contribute much to Weekly_Sale
s

# Segregate Data into predictors and target variable.

X_features = df_dummyFeaturesModel.drop(['Temperature', 'Fuel_Price', 'CPI', 'Date', 'We
ekly_Sales'], axis=1)
Y_target = df_dummyFeaturesModel[['Weekly_Sales']]

# split data into 70% training and 30% test

x_train, x_test, y_train, y_test = train_test_split(X_features, Y_target, test_size=0.3)

print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

```
(4504, 48) (1931, 48) (4504, 1) (1931, 1)
```

In [92]:

```
# Fit Linear Regression Model
from sklearn.linear_model import LinearRegression

salesLM = LinearRegression()
salesLM.fit(x_train, y_train)
```

Out[92]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [93]:

```
# fitting to linear regression model basically finds the best value for the intercept and
slope, which results in a line that best fits the data.

print('Intercept is- {}\n'.format(salesLM.intercept_[0]))

for idx, col_name in enumerate(x_train.columns):
    print("The coefficient for {} is {}".format(col_name, salesLM.coef_[0][idx]))
```

```
Intercept is- 1149270.7369196147

The coefficient for Holiday_Flag is 95088.92547989608
The coefficient for Unemployment is -29155.216609666262
The coefficient for Total_Days is 38.218131191138625
The coefficient for Store_1 is 496970.2636162093
The coefficient for Store_2 is 876782.6656928946
The coefficient for Store_3 is -671939.8070541833
The coefficient for Store_4 is 982995.8043979668
The coefficient for Store_5 is -781313.3238668651
The coefficient for Store_6 is 489270.3928597354
The coefficient for Store_7 is -452233.7835237351
The coefficient for Store_8 is -197405.67620568632
The coefficient for Store_9 is -559363.5818881978
The coefficient for Store_10 is 839503.6116042199
The coefficient for Store_11 is 293720.80489837716
The coefficient for Store_12 is 121780.3185847989
The coefficient for Store_13 is 919438.5991144078
The coefficient for Store_14 is 992676.4901073555
The coefficient for Store_15 is -433527.3062994573
```

```
The coefficient for Store_16 is -571731.1280186258
The coefficient for Store_17 is -195943.51475191972
The coefficient for Store_18 is 69324.43680621416
The coefficient for Store_19 is 398406.1299068503
The coefficient for Store_20 is 1048276.6158876342
The coefficient for Store_21 is -293105.50474674144
The coefficient for Store_22 is -22671.443768383633
The coefficient for Store_23 is 244798.39306208387
The coefficient for Store_24 is 329650.8446915614
The coefficient for Store_25 is -359595.08453827293
The coefficient for Store_26 is -49189.53638426766
The coefficient for Store_27 is 735863.5708401067
The coefficient for Store_28 is 425431.9777738704
The coefficient for Store_29 is -460457.4676602545
The coefficient for Store_30 is -621089.8092143002
The coefficient for Store_31 is 322359.9740788054
The coefficient for Store_32 is 149683.18102280522
The coefficient for Store_33 is -773714.3644214583
The coefficient for Store_34 is -25420.550370162728
The coefficient for Store_35 is -111392.33449597824
The coefficient for Store_36 is -676642.7154973809
The coefficient for Store_37 is -531061.9299633293
The coefficient for Store_38 is -509495.08079339756
The coefficient for Store_39 is 402123.40101348836
The coefficient for Store_40 is -168649.25461079925
The coefficient for Store_41 is 188369.18132425472
The coefficient for Store_42 is -480311.4213678524
The coefficient for Store_43 is -355765.2219469744
The coefficient for Store_44 is -783362.4647570027
The coefficient for Store_45 is -242044.35113841254
```

In [95]:

```
# model accuracy score

print('So in our model, {}% of the variability in Y can be explained using X'.format(roun
d(salesLM.score(x_test, y_test) * 100, 2)))
```

```
So in our model, 92.16% of the variability in Y can be explained using X
```

In [96]:

```
# Making Predictions using trained model

y_pred = salesLM.predict(x_test)

# Metric evaluation

from sklearn import metrics

mae = round(metrics.mean_absolute_error(y_test, y_pred), 2)
mse = round(metrics.mean_squared_error(y_test, y_pred), 2)
rmse = round(np.sqrt(metrics.mean_squared_error(y_test, y_pred)), 2)

print('Mean Absolute Error:', mae)
print('Mean Squared Error:', mse)
print('Root Mean Squared Error:', rmse)

print('\n An RMSE {} means that on average the model predicts ${} above or below values o
f the Weekly Sales Values. i.e {} % below Weekly Sales average.'.format(rmse, rmse, roun
d((rmse/df_walmartAnalysisModel['Weekly_Sales'].mean()) * 100, 2)))
```

```
Mean Absolute Error: 90036.62
Mean Squared Error: 24888564719.67
Root Mean Squared Error: 157761.1

 An RMSE 157761.1 means that on average the model predicts $157761.1 above or below value
s of the Weekly Sales Values. i.e 15.07 % below Weekly Sales average.
```

In [97]:

```
#Performance on the test data sets
```

```
plt.plot(y_test, y_pred,'ro')
plt.plot(y_test, y_test,'b-')
plt.show()
```