BZAN 535 - Homework 1

Kevin Gardner

August 30, 2016

Problem 1

Determine the number of orders, the first and last shipping date, as well as totals for price and units sold. List the 20 products with the highest total dollar sales, along with their first and last order dates and number of orders and number of units. How many of these top 20 selling products are we still selling in 2016 (the last year in the data orders)?

```
select productid
, count(*) as units_sold
, sum(numunits*unitprice) as total_sales
, max(shipdate) as last_date
, min(shipdate) as first_date
from orderline group by productid
order by total_sales desc limit 20;
```

```
##     productid units_sold total_sales  last_date first_date
##  1:     10361       1349   240831.94 2016-07-08 2010-03-22
##  2:     11168       9546   232272.25 2013-02-13 2010-10-07
##  3:     12826       2172   229873.40 2016-09-20 2014-09-03
##  4:     10640       1105   200338.90 2016-08-22 2010-07-08
##  5:     12139       7208   183098.50 2016-09-15 2014-05-05
##  6:     12825       1653   140144.09 2016-09-19 2014-09-02
##  7:     13629       3686   123258.07 2016-09-19 2015-02-13
##  8:     10991       1176   121106.98 2016-08-26 2010-12-17
##  9:     11196       4768    97500.72 2016-08-09 2009-11-17
## 10:     11023        544    97183.46 2016-05-09 2011-09-08
## 11:     11058       2837    97178.74 2016-09-09 2013-04-15
## 12:     11009       5874    96800.66 2012-01-30 2010-11-18
## 13:     12830       1096    94815.67 2016-09-20 2014-10-08
## 14:     11199        262    93577.48 2016-08-16 2013-07-03
## 15:     12005       2787    79989.57 2016-09-02 2013-11-11
## 16:     10992        757    77871.87 2015-01-28 2010-11-08
## 17:     10786        449    77853.89 2016-07-08 2011-02-24
## 18:     10641        237    76835.50 2016-07-15 2012-11-06
## 19:     11088       3782    76777.87 2016-01-13 2011-05-23
## 20:     11006       2205    70363.04 2016-09-16 2013-07-19
```

The following 17 products are still selling as of 2016.

```
##    productid units_sold total_sales  last_date first_date
## 1      12826       2172   229873.40 2016-09-20 2014-09-03
## 2      12830       1096    94815.67 2016-09-20 2014-10-08
## 3      12825       1653   140144.09 2016-09-19 2014-09-02
## 4      13629       3686   123258.07 2016-09-19 2015-02-13
## 5      11006       2205    70363.04 2016-09-16 2013-07-19
## 6      12139       7208   183098.50 2016-09-15 2014-05-05
```

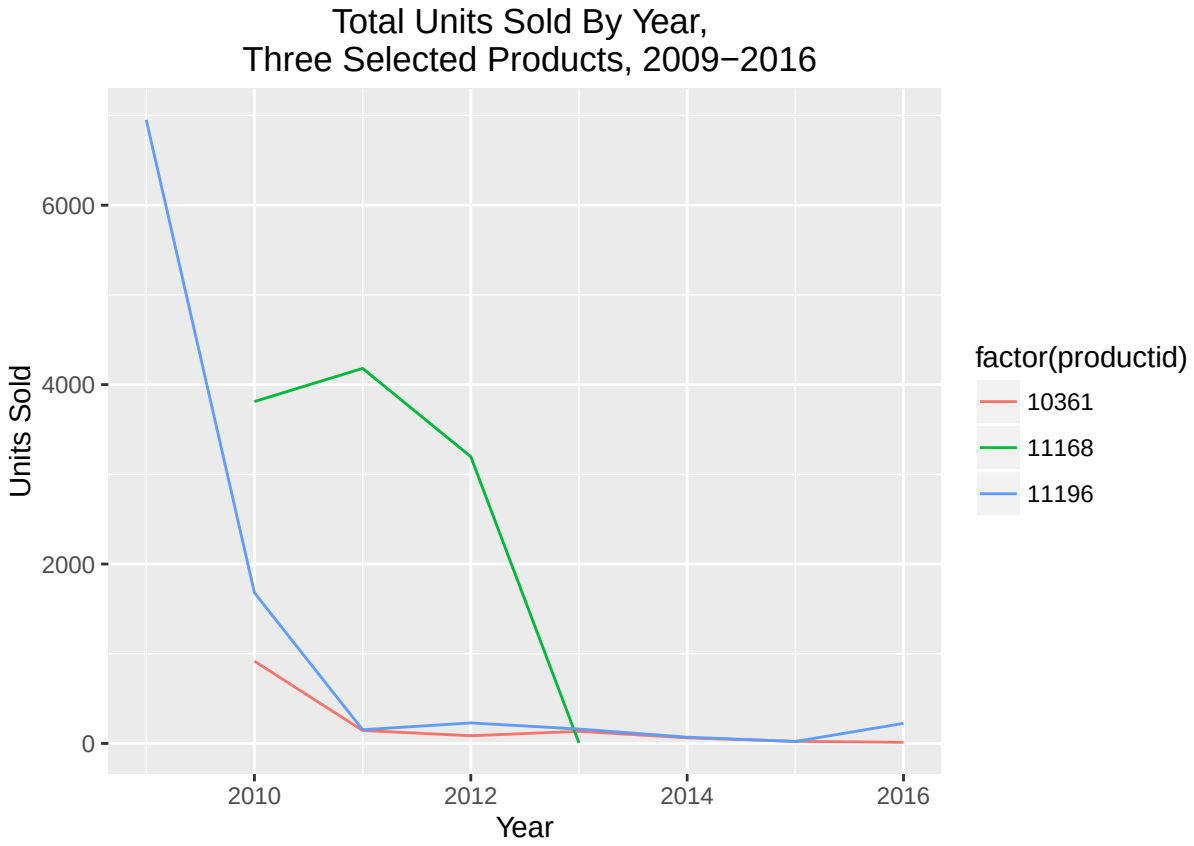
```
## 7      11058       2837    97178.74 2016-09-09 2013-04-15
## 8      12005       2787    79989.57 2016-09-02 2013-11-11
## 9      10991       1176   121106.98 2016-08-26 2010-12-17
## 10     10640       1105   200338.90 2016-08-22 2010-07-08
## 11     11199        262    93577.48 2016-08-16 2013-07-03
## 12     11196       4768    97500.72 2016-08-09 2009-11-17
## 13     10641        237    76835.50 2016-07-15 2012-11-06
## 14     10361       1349   240831.94 2016-07-08 2010-03-22
## 15     10786        449    77853.89 2016-07-08 2011-02-24
## 16     11023        544    97183.46 2016-05-09 2011-09-08
## 17     11088       3782    76777.87 2016-01-13 2011-05-23
```

Problem 2

Product `10361` was the best seller, product `11168` was the second-best seller, while product `11196` has had a very long life cycle. For each of these three products, report by year the number of units sold and the average per unit price. Comment on what this reveals about the pattern of sales for popular items.

```
select productid
, sum(numunits) as total_units
, extract(year from shipdate) as year_sold
, avg(unitprice) as mean_unitprice
from orderline
where productid IN ('10361','11168','11196')
group by year_sold, productid
order by productid, year_sold;
```

```
##     productid total_units year_sold mean_unitprice
##  1:     10361         917      2010     171.674752
##  2:     10361         145      2011     170.600699
##  3:     10361          86      2012     180.461176
##  4:     10361         135      2013     183.632222
##  5:     10361          63      2014     176.618852
##  6:     10361          24      2015     180.163043
##  7:     10361          13      2016     184.500000
##  8:     11168        3811      2010      22.214930
##  9:     11168        4181      2011      20.713397
## 10:     11168        3197      2012      20.552742
## 11:     11168           6      2013      21.260000
## 12:     11196        6951      2009      14.556298
## 13:     11196        1680      2010      13.770806
## 14:     11196         153      2011      13.301985
## 15:     11196         229      2012      13.179733
## 16:     11196         161      2013      13.788182
## 17:     11196          70      2014      10.195455
## 18:     11196          22      2015       6.228333
## 19:     11196         224      2016       6.730000
```



Plotting the data we see that popular items lose popularity over time. Product `11169`, which experienced a shorter lifecycle than the other two products, grew in popularity the first year and then shrunk in popularity at an increasing rate in the remaining two years. This suggests that for popular items there may be a sharp difference bewteen shorter and longer lifecycle products with respect to the pattern of sales over time.

Problem 3

Compute monthly sales totals for this business over the eight years. Append to this the best selling product each month and what % of monthly sales are from this one item. Discuss what you learn. Show output only for `2009` and `2016`, but you should examine all `8` years.

```
select t1.billdate
, t1.sales_month
, t1.sales_year
, t1.total_sales
, t2.productid
, t2.maxsales
, (t2.maxsales/t1.total_sales)*100 as percent_sales

# Monthly sales totals for the business
from (
        select billdate
        , month(billdate) as sales_month
        , year(billdate) as sales_year
        , sum(totalprice) as total_sales
        from orderline
```

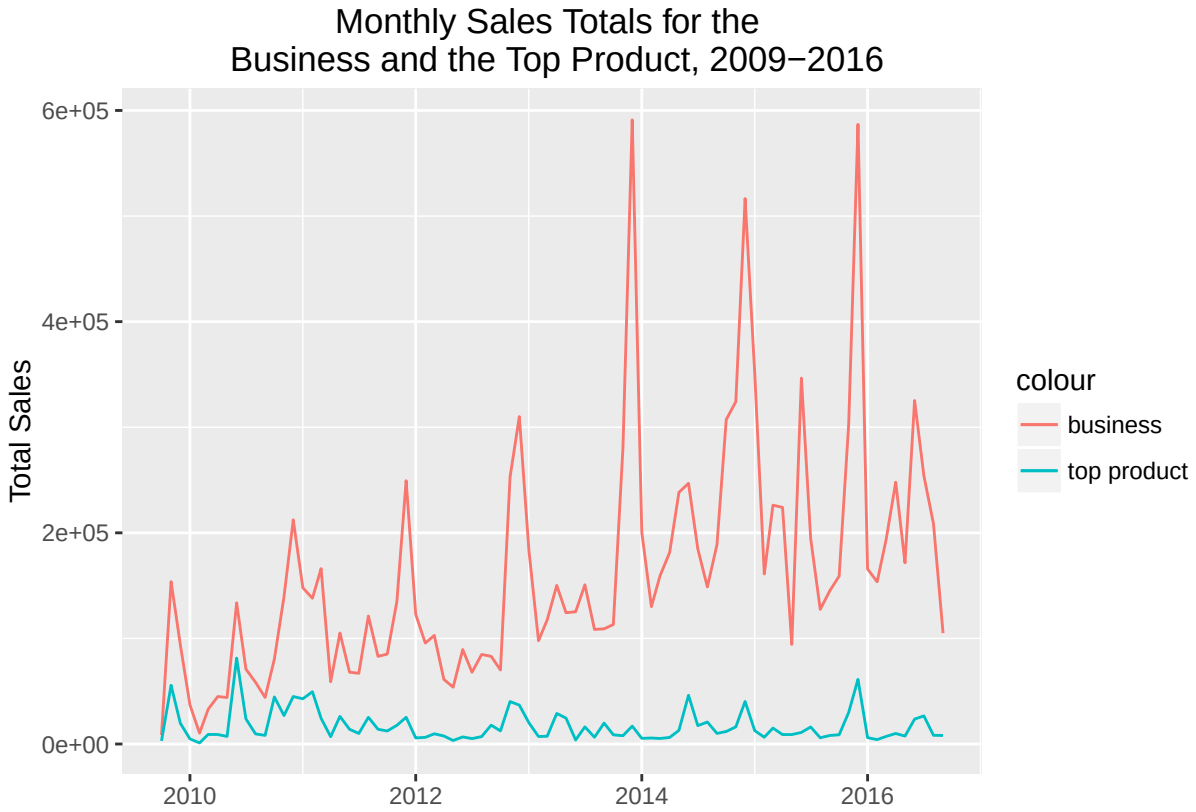
```
        group by month(billdate), year(billdate)
) as t1

# Monthly sales totals for best selling product
, (
        select billdate
        , month(billdate) as sales_month
        , year(billdate) as sales_year
        , productid
        , max(totalprice_sum) as maxsales
        from (
                select billdate
                , month(billdate) as sales_month
                , year(billdate) as sales_year
                , productid
                , sum(totalprice) as totalprice_sum
                from orderline
                group by sales_month, sales_year, productid
                order by  sales_year, sales_month, totalprice_sum
        ) as a group by sales_month, sales_year
) as t2

where t1.sales_month = t2.sales_month and t1.sales_year = t2.sales_year
order by t1.sales_year,t1.sales_month;
```

Below are the data for 2009 and 1016.

```
##    sales_year sales_month total_sales productid maxsales percent_sales
## 1        2009          10     8497.68     11070  2947.96     34.691351
## 2        2009          11   153762.23     10799 55576.14     36.144208
## 3        2009          12    93931.24     11049 19612.50     20.879635
## 4        2016           1   165687.74     13487  6039.38      3.645037
## 5        2016           2   153651.43     13149  4162.20      2.708859
## 6        2016           3   193762.48     12394  7184.93      3.708112
## 7        2016           4   247939.74     13429  9967.70      4.020211
## 8        2016           5   171625.53     11069  7520.79      4.382093
## 9        2016           6   325295.10     13982 23616.00      7.259870
## 10       2016           7   254145.97     13941 26655.75     10.488362
## 11       2016           8   208698.05     13805  8266.50      3.960986
## 12       2016           9   105060.47     11974  8118.00      7.726979
```



Plotting the data we see there are seasonal fluctuations with highest sales occuring in December. Also, the trend for overall sales is more positive than the trend for top products sales.

Problem 4

Order date appears in the orders table while shipping date appears in orderline. We should be concerned about delays in shipping. Join these tables and identify the largest shipping delays. Show the 10 worst cases.

```
select t1.orderid, t1.orderdate, t2.shipdate
, datediff(t2.shipdate, t1.orderdate) as delay
from orders as t1 join orderline as t2
on t1.orderid = t2.orderid
order by delay desc
limit 10;
```

```
##     orderid  orderdate   shipdate delay
##  1: 1030574 2009-12-01 2011-08-18   625
##  2: 1029008 2009-12-01 2011-08-18   625
##  3: 1030930 2009-12-01 2011-08-18   625
##  4: 1030573 2009-12-01 2011-08-18   625
##  5: 1033456 2009-12-08 2011-08-18   618
##  6: 1032217 2009-12-08 2011-08-18   618
##  7: 1036520 2010-01-19 2011-08-18   576
##  8:  999994 2010-01-28 2011-08-18   567
##  9: 1001121 2010-03-05 2011-08-18   531
## 10: 1087808 2011-12-29 2013-04-02   460
```