IE 5374 Project

Walmart Sales Analysis

Ritwik Achanti | Mahitha Guduri | Siddhartha Putti

Business Problem

- It used hierarchical sales data, generously made available by Walmart, starting at the item level and aggregating to that of departments, product categories and stores in three geographical areas of the US: California, Texas, and Wisconsin.
- Besides the time series data, it also included explanatory variables such as price, promotions, day of the week, and special events (e.g. Super Bowl, Valentine's Day, and Orthodox Easter) that affect sales which are used to improve forecasting accuracy.
- The majority of the more than 42,840 time series display intermittency (sporadic sales including zeros).

Credits: https://mofc.unic.ac.cy/m5-competition/

Objectives covered under this analysis:

- Aggregate time series over all items, stores, categories, departments and sales.
- Sales per state on a monthly aggregate level.
- Sales per store and category.
- Sales per Department.
- Seasonality - global effect on sales.
- Sales volume during days with special events vs non-events for all categories.
- Daily sales percentage for SNAP vs other.
- Analyse how item prices are subjected to change relatively.

The data has *7 departments*, **3049** products from *3 categories* sold in *10 stores* in *3 states*.

The data comes in 3 separate files:

Importing data

```
train <- read.csv('sales.csv')
prices <- read.csv('sell_prices.csv')
calendar <-read.csv('calendar.csv')
```

quick look at the data

```
train %>%
  select(seq(1,8,1)) %>%
  head(3)
```

```
##                              id       item_id   dept_id  cat_id store_id
## 1 HOBBIES_1_001_CA_1_validation HOBBIES_1_001 HOBBIES_1 HOBBIES     CA_1
## 2 HOBBIES_1_002_CA_1_validation HOBBIES_1_002 HOBBIES_1 HOBBIES     CA_1
## 3 HOBBIES_1_003_CA_1_validation HOBBIES_1_003 HOBBIES_1 HOBBIES     CA_1
##   state_id d_1 d_2
## 1       CA   0   0
## 2       CA   0   0
## 3       CA   0   0
```

- The sales per date are encoded as columns starting with the prefix `d_`. Those are the number of units sold per day.

```
  c(ncol(train),nrow(train))
```

```
## [1]  1919 30490
```

- There are in total 1919 columns for which the days are represented in a column fashion, we can pivot the dates to single column by a helper function without changing the original dataframe.
- There are 30490 rows that have different items in 3 States with 10 different Walmart stores.

The other data file we have is sales_price.

```
prices %>%
  head(3)
```

```
##   store_id       item_id wm_yr_wk sell_price
## 1     CA_1 HOBBIES_1_001    11325       9.58
## 2     CA_1 HOBBIES_1_001    11326       9.58
## 3     CA_1 HOBBIES_1_001    11327       8.26
```

This shows the change in prices with respective week numbers.

The third data file is calendar, lets have a quick look at this data even.

```
calendar %>%
  head(3)
```

```
##         date wm_yr_wk  weekday wday month year   d event_name_1 event_type_1
## 1 2011-01-29    11101 Saturday    1     1 2011 d_1
## 2 2011-01-30    11101   Sunday    2     1 2011 d_2
## 3 2011-01-31    11101   Monday    3     1 2011 d_3
##   event_name_2 event_type_2 snap_CA snap_TX snap_WI
## 1                                0       0       0
## 2                                0       0       0
## 3                                0       0       0
```

- we have event name and type of events on a particular date and SNAP binary values to corresponding states.

we have many columns representing the dates in sales data, this is short helper function to convert the long columns into dates format.

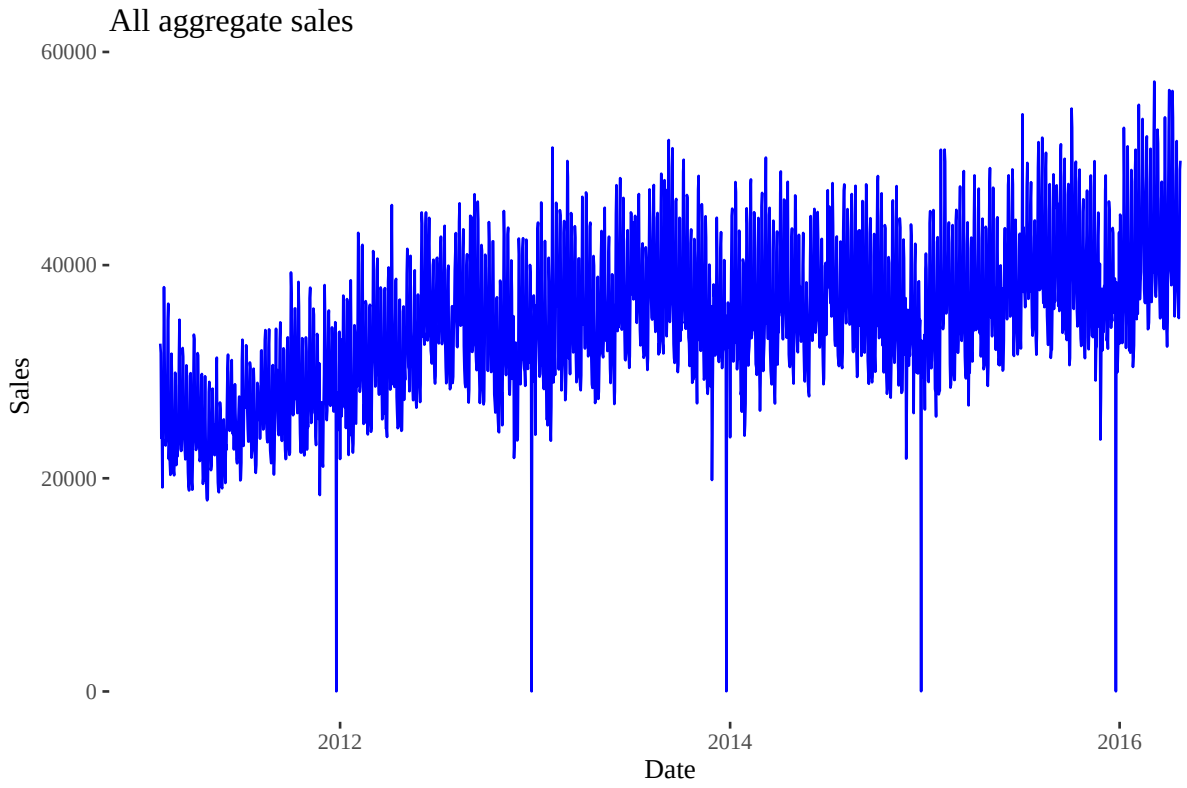
- we know that the start date is from 2011-01-29 which is given as day 1 in our dataset. lets calculate our dates accordingly.

Helper function for date.

```
extract_dates <- function(df){

  min_date <- as.Date("2011-01-29")

  df %>%
    select(id, starts_with("d_")) %>%
    pivot_longer(starts_with("d_"), names_to = "dates", values_to = "sales") %>%
    mutate(dates = as.integer(str_remove(dates, "d_"))) %>%
    mutate(dates = min_date + dates - 1) %>%
    mutate(id = str_remove(id, "_validation"))
}
```

First off, here we plot the aggregate time series over all items, stores, categories, departments and sales.

```
df_dates <- train %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  mutate(id = 1)

df_extract_dates <- extract_dates(df_dates)

gg <- df_extract_dates %>%
  ggplot(aes(dates, sales)) +
  geom_line(col = "blue") +
  theme_tufte() +
  labs(x = "Date", y = "Sales", title = "All aggregate sales")
gg
```

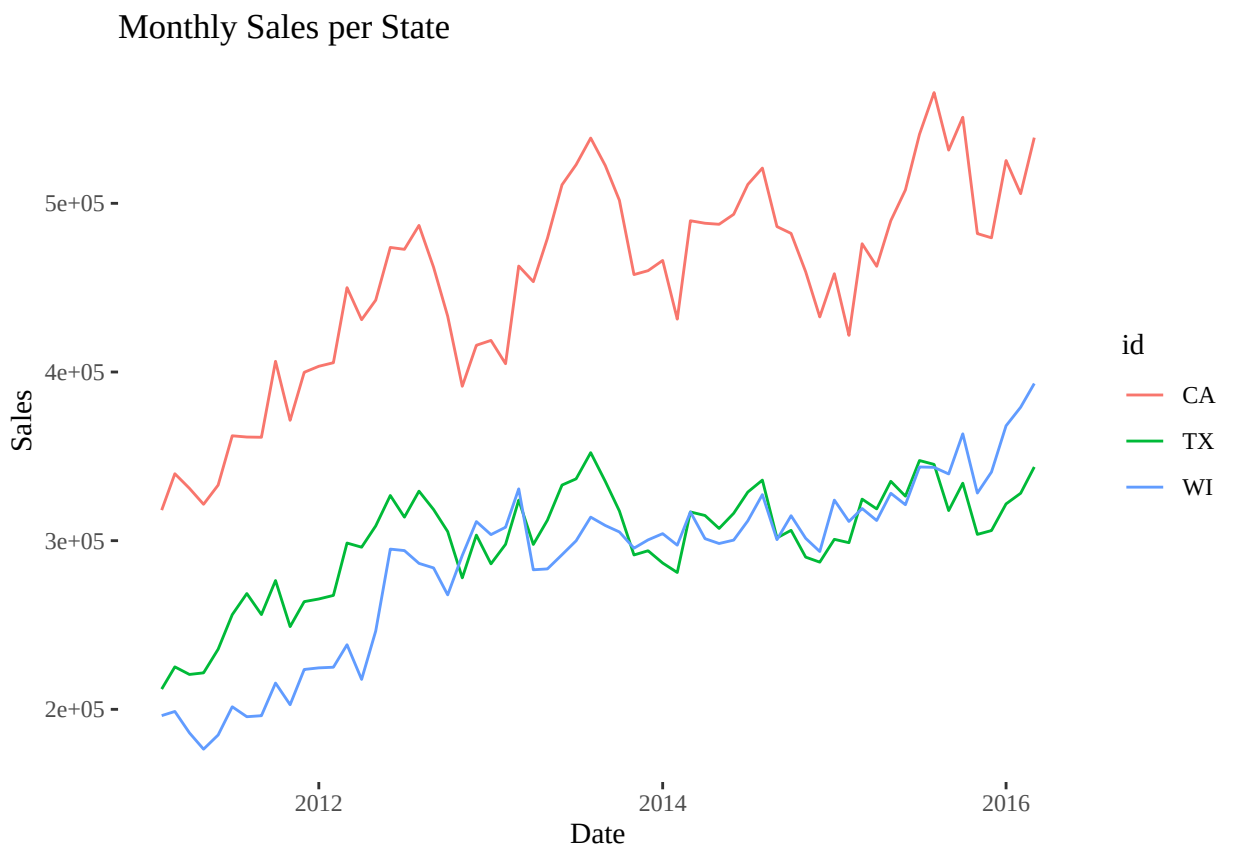


We find:

- The sales are generally going up, We can observe some yearly dips, and a dip at Christmas, which is the only day of the when the stores are closed.
- we can see strong weekly seasonality plus possibly some additional overlaying patterns with shorter periods than yearly.
- 2016 sales numbers appear to grow a bit faster than in previous years.

Now we will look at the sales per state on a monthly aggregate level.

```
df_groupBy_SID <- train %>%
  group_by(state_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  rename(id = state_id)

df_extract_dates <- extract_dates(df_groupBy_SID) %>%
  mutate(month = month(dates),
         year = year(dates)) %>%
  group_by(month, year, id) %>%
  summarise(sales = sum(sales),
            dates = min(dates)) %>%
  ungroup() %>%
  filter(str_detect(as.character(dates), "..-..-01")) %>%
  filter(dates != max(dates))

gg <- df_extract_dates %>%
  ggplot(aes(dates, sales, col = id)) +
  geom_line() +
  theme_tufte() +
  labs(x = "Date", y = "Sales", title = "Monthly Sales per State")
gg
```



```
#ggplotly(gg, dynamicTicks = TRUE)
```

We find:

- CA sells more items, while WI is catching up to TX and eventually crosses TX.
- Except in 2016 all the states appears to have pronounced dips in every other year. These dips and peaks don't appear to always occur (see 2012) but they might primarily reflect the yearly seasonality we noticed already. i.e end of the year.

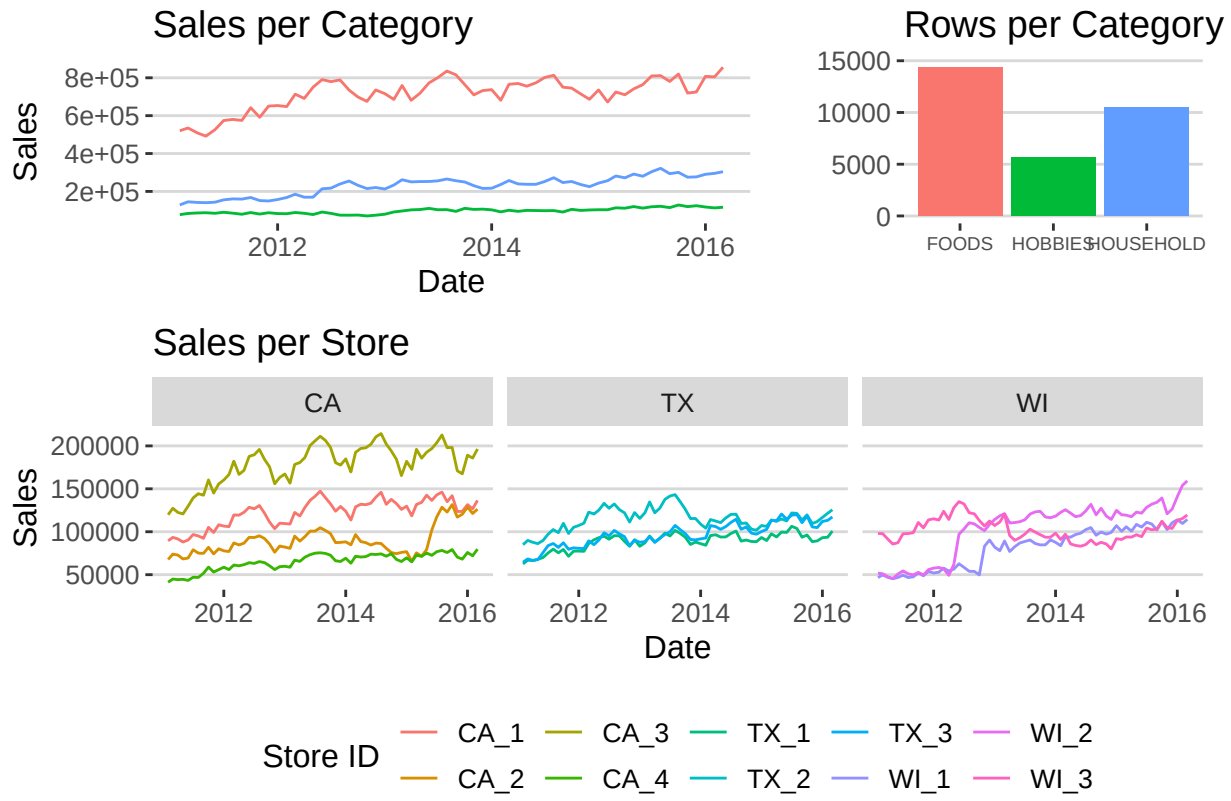
As we got overview of sales per state. lets analyse

Sales per store and category.

we know that there are 10 stores- 4 in CA and 3 in TX and WI, with 3 categories namely Foods, Hobbies, and Household.

```
# grouping data by category
foo <- train %>%
  group_by(cat_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  rename(id = cat_id)

#grouping data by store
bar <- train %>%
  group_by(store_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  rename(id = store_id)

# simple plot in the sales by category per month
plot1 <- extract_dates(foo) %>%
  mutate(month = month(dates),
         year = year(dates)) %>%
  group_by(month, year, id) %>%
  summarise(sales = sum(sales),
            dates = min(dates)) %>%
  ungroup() %>%
  filter(str_detect(as.character(dates), "..-..-01")) %>%
  filter(dates != max(dates)) %>%
  ggplot(aes(dates, sales, col = id)) +
  geom_line() +
  theme_hc() +
  theme(legend.position = "none") +
  labs(title = "Sales per Category", x = "Date", y = "Sales")

# finding the total number of row per category on the dataset
plot2 <- train %>%
  count(cat_id) %>%
  ggplot(aes(cat_id, n, fill = cat_id)) +
  geom_col() +
  theme_hc() +
```

```
  theme(legend.position = "none") +
  theme(axis.text.x = element_text(size = 7)) +
  labs(x = "", y = "", title = "Rows per Category")

# simple plot for the sales in each store by month
plot3 <- extract_dates(bar) %>%
  mutate(month = month(dates),
         year = year(dates)) %>%
  group_by(month, year, id) %>%
  summarise(sales = sum(sales),
            dates = min(dates)) %>%
  ungroup() %>%
  filter(str_detect(as.character(dates), "..-..-01")) %>%
  filter(dates != max(dates)) %>%
  mutate(state_id = str_sub(id, 1, 2)) %>%
  ggplot(aes(dates, sales, col = id)) +
  geom_line() +
  theme_hc() +
  theme(legend.position = "bottom") +
  labs(title = "Sales per Store", x = "Date", y = "Sales", col = "Store ID") +
  facet_wrap(~state_id)
#lets make all the visualization on the same plot
layout <- "
AAB
CCC
"

plot1 + plot2 + plot3 + plot_layout(design = layout)
```



We find:

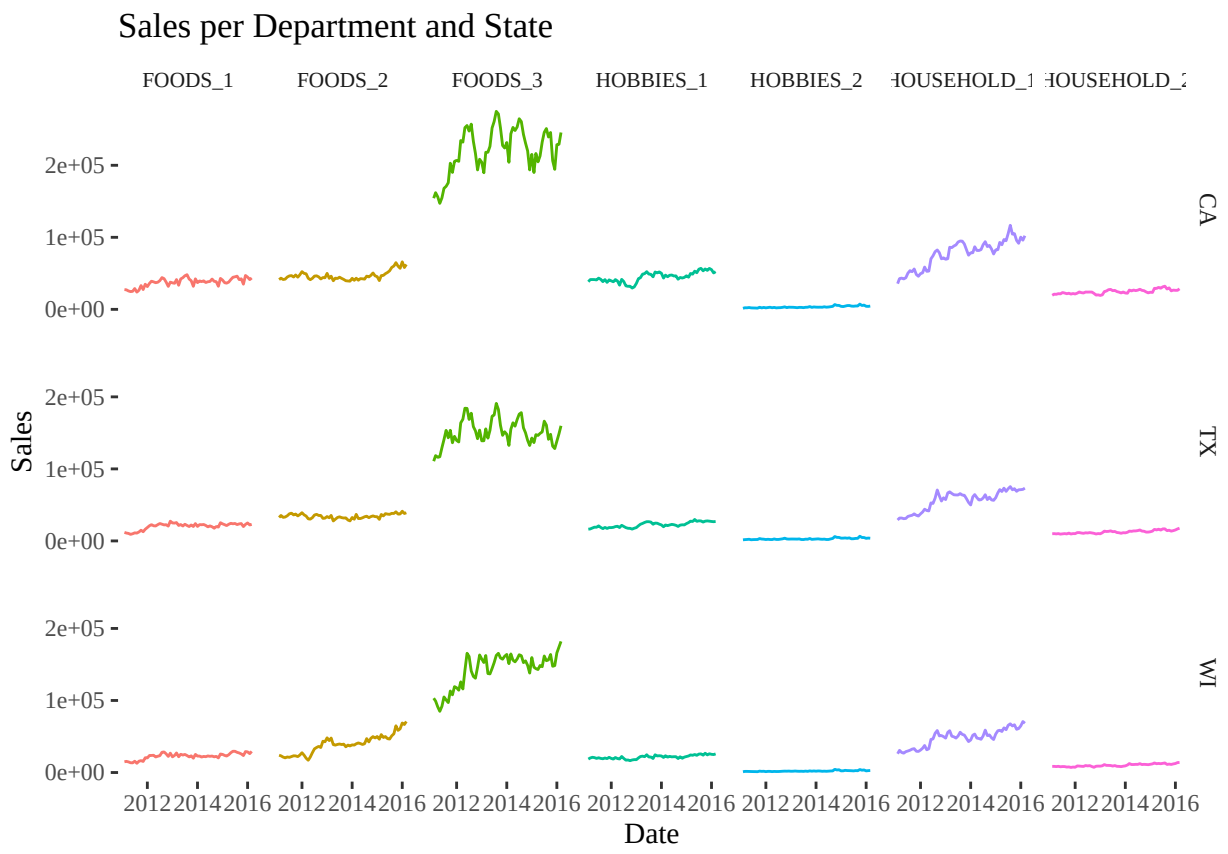
- "Foods" are the most common category. The number of "Household" rows is closer to the number of "Foods" rows than the corresponding sales figures, indicating that more "Foods" units are sold than "Household" ones.
- In terms of stores, we see that the Texas stores are quite close together in sales; with "TX_3" rising from the levels of "TX_1" to the level of "TX_2" over the time. The WI stores "WI_1" and "WI_2" show a jump in sales in 2012, while "WI_3" shows a long dip over many year.
- The CA stores are relatively well separated in store volume. also "CA_2", which declines to the "CA_4" level in 2015, and jump up to "CA_1" sales later in the year.

Sales per Department and state.

Our data has 7 departments, 3 for "FOODS" and 2 each for "HOBBIES" and "HOUSEHOLD". Together with the 3 states those are 21 different plots.

```
min_date <- as.Date("2011-01-29")

df_groupBy_SID_DepID <- train %>%
  group_by(dept_id, state_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  ungroup() %>%
  select(ends_with("id"), starts_with("d_")) %>%
```

```
  pivot_longer(starts_with("d_"), names_to = "dates", values_to = "sales") %>%
  mutate(dates = as.integer(str_remove(dates, "d_"))) %>%
  mutate(dates = min_date + dates - 1)

df_groupBy_SID_DepID %>%
  mutate(month = month(dates),
         year = year(dates)) %>%
  group_by(month, year, dept_id, state_id) %>%
  summarise(sales = sum(sales),
            dates = min(dates)) %>%
  ungroup() %>%
  filter(str_detect(as.character(dates), "..-..-01")) %>%
  filter(dates != max(dates)) %>%
  ggplot(aes(dates, sales, col = dept_id)) +
  geom_line() +
  facet_grid(state_id ~ dept_id) +
  theme_tufte() +
  theme(legend.position = "none", strip.text.x = element_text(size = 8)) +
  labs(title = "Sales per Department and State", x = "Date", y = "Sales")
```



We find:

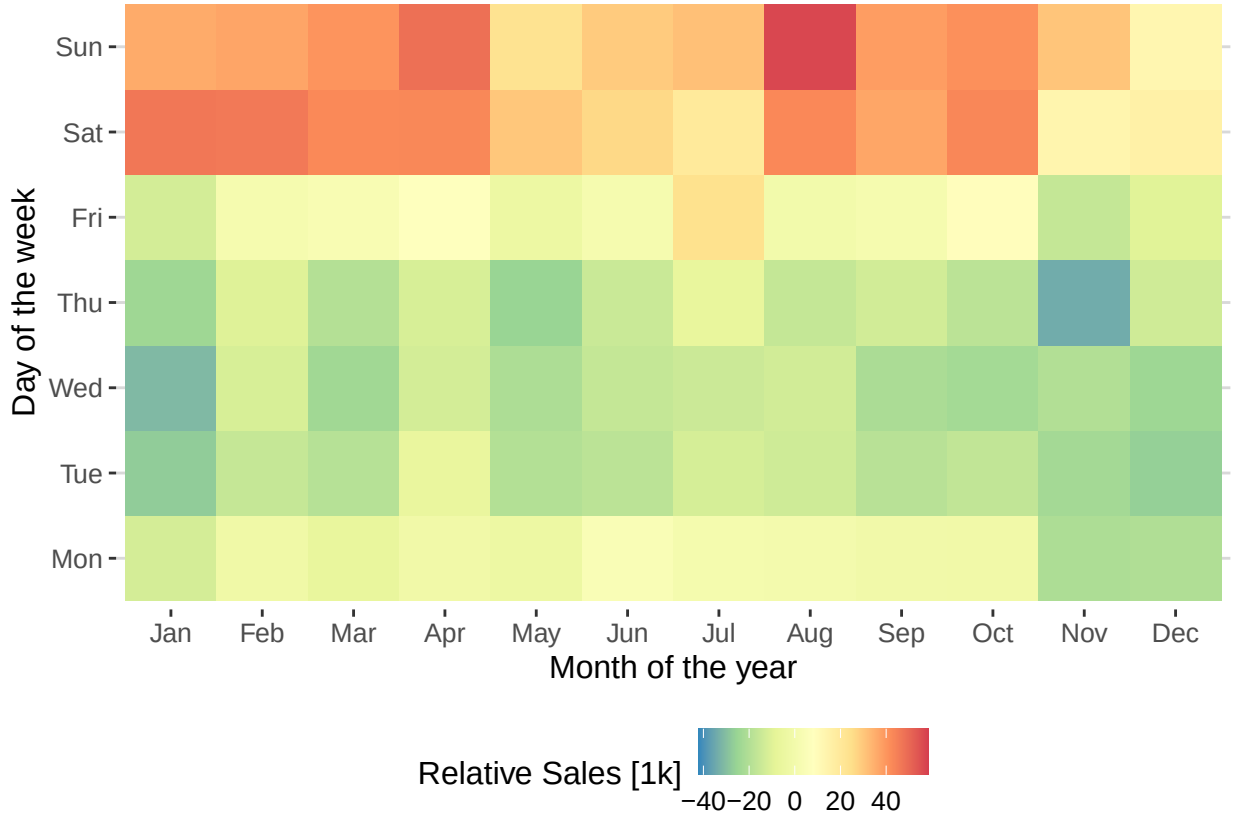
- "FOODS_3" is clearly driving the majority of "FOODS" category sales in all states. "FOODS_2" is picking up a bit towards the end of the time range, especially in "WI".
- Similarly, "HOUSEHOLD_1" is clearly outselling "HOUSEHOLD_2".
- "HOBBIES_1" is on a higher average sales level than "HOBBIES_2", but both are not showing much development over time.

Seasonalities - global

Now, we aim to model this trend using a smoothed (LOESS) fit which we then subtract from the data of sales we have, this gives the relative change in sales.

Note, that we are removing the Christmas(dec-25th) dips because they would be distracting for the purpose of this smoothing and relative change in our plot:

```
#getting only date columns
dates <- train %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  mutate(id = 1)

#deleting dec-25th dips from the data we have
rm_dips <- extract_dates(dates) %>%
  filter(!str_detect(as.character(dates), "-12-25"))

#smoothing the sales as per the neighbor values,
#returns the array of dimensions of input-
#this shows only the observed values are smoothed and we are not predicting any outcomes
loess_all <- predict(loess(rm_dips$sales ~ as.integer(rm_dips$dates -
                                  min(rm_dips$dates))+1 , span =1/2, degree=1))

mutate_loess <- rm_dips %>%
  mutate(loess = loess_all) %>%
  mutate(sales_rel = sales - loess)

p1 <- mutate_loess %>%
  mutate(wday = wday(dates, label = TRUE, week_start = 1),
         month = month(dates, label = TRUE),
         year = year(dates)) %>%
  group_by(wday, month, year) %>%
  summarise(sales = sum(sales_rel)/1e3) %>%
  ggplot(aes(month, wday, fill = sales)) +
  geom_tile() +
  labs(x = "Month of the year", y = "Day of the week",
       fill = "Relative Sales [1k]") +
       scale_fill_distiller(palette = "Spectral") +
  theme_hc()
p1
```



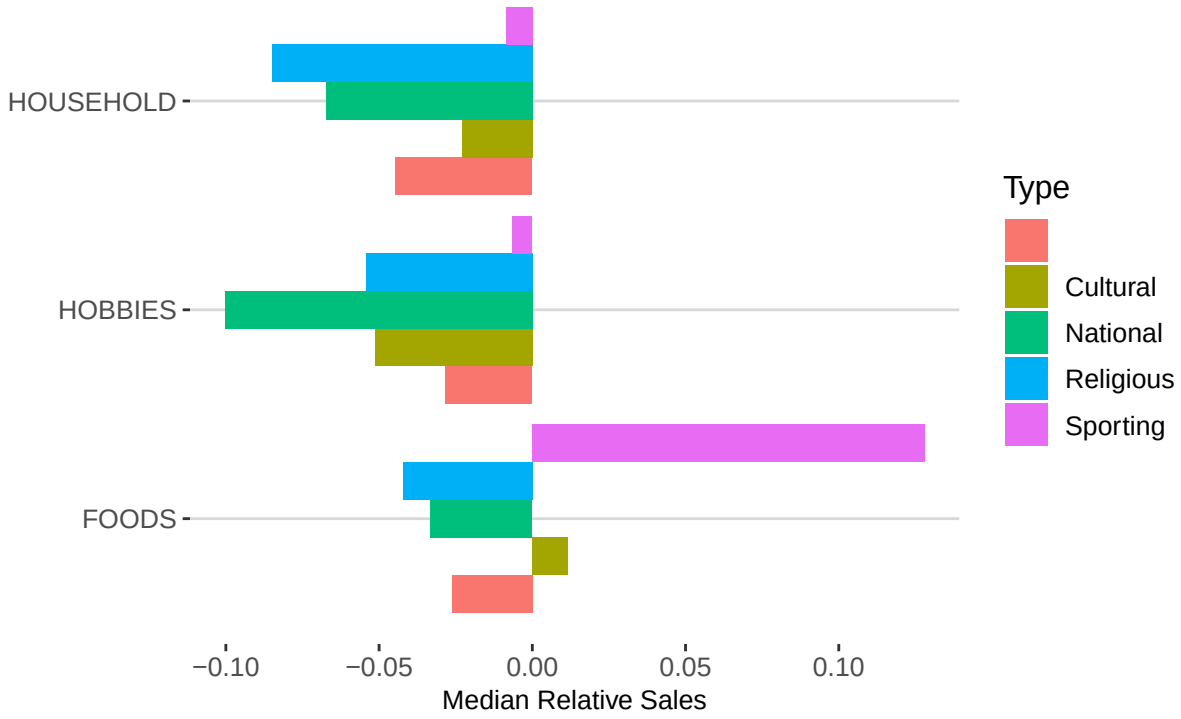
We find:

- we see that sales on Sat and Sun standing out prominently. Also Monday seems to benefit a bit from the effect of saturday and sunday.
- The months of Nov and Dec show clear dips, while the summer months May, Jun, and Jul suggest a milder secondary dip. Certain holidays, like the 4th of July, might somewhat influence these patterns; but over 5 years they should average out well.

Sales volume during days with special events vs non-events for all categories.

```
calendar$date = as.Date(calendar$date)
foo <- train %>%
  group_by(cat_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  rename(id = cat_id) %>%
  extract_dates() %>%
  rename(cat_id = id) %>%
  left_join(calendar %>% select(date, event_type_1), by = c("dates" = "date")) %>%
  filter(!str_detect(as.character(dates), "-12-25")) %>%
  group_by(cat_id) %>%
  mutate(loess = predict(loess(sales ~ as.integer(dates - min(dates)) + 1,
```

```
                span = 1/2, degree = 1)),mean_sales = mean(sales)) %>%
  mutate(sales_rel = (sales - loess)/mean_sales) %>%
  mutate(is_event = !is.na(event_type_1)) %>%
  ungroup()


plt <- foo %>%
  filter(is_event == TRUE) %>%
  group_by(cat_id, event_type_1) %>%
  summarise(sales = median(sales_rel)) %>%
  ggplot(aes(cat_id, sales, fill = event_type_1)) +
  geom_col(position = "dodge") +
  coord_flip() +
  theme_hc() +
  theme(legend.position = "right", axis.title.x = element_text(size = 10)) +
  labs(x = "", y = "Median Relative Sales", fill = "Type")

plt
```



We find:

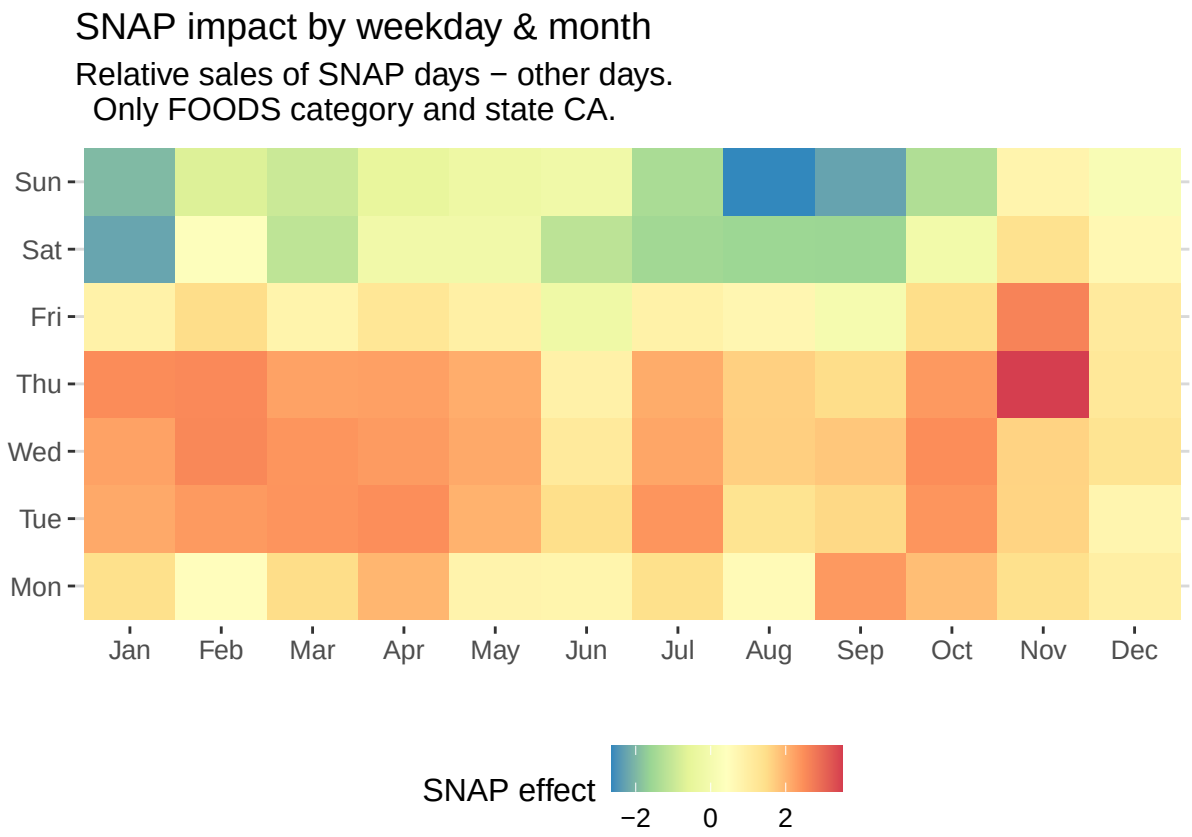
- We break out the events by type and look at the medians of the relative sales. The first thing we see is that FOODS sales are notably higher during "Sporting" events. This makes sense, given the food culture associated with big events like the Superbowl. FOODs also have slightly positive net sales during "Cultural" events.
- In general, "National" and "Religious" events both lead to relative decline in sales volume. "National" events are more depressing for the HOBBIES category, while the other two categories are slightly more affected by "Religious" events. HOBBIES also sees lower sales from "Cultural" events, while for FOODS

and HOUSEHOLD the differences are smaller. “Sporting” has a minor impact on HOUSEHOLD and HOBBIES.

We now look at the daily sales percentage for SNAP vs other.

i.e differences between the sum of relative sales on SNAP days minus the sum of relative sales on other days.

```
min_date <- min(calendar$date)

bar <- calendar %>%
  select(date, starts_with("snap")) %>%
  pivot_longer(starts_with("snap"), names_to = "state_id", values_to = "snap") %>%
  mutate(state_id = str_replace(state_id, "snap_", ""))

foo <- train %>%
  group_by(state_id, cat_id) %>%
  summarise_at(vars(starts_with("d_")), sum) %>%
  pivot_longer(starts_with("d_"), names_to = "dates", values_to = "sales") %>%
  mutate(dates = as.integer(str_remove(dates, "d_"))) %>%
  mutate(dates = min_date + dates - 1) %>%
  left_join(bar, by = c("dates" = "date", "state_id")) %>%
  filter(!str_detect(as.character(dates), "-12-25")) %>%
  mutate(snap = as.logical(snap)) %>%
  group_by(state_id, cat_id) %>%
  mutate(loess = predict(loess(sales ~ as.integer(dates - min(dates)) + 1, span = 1/2,
                               degree = 1)), mean_sales = mean(sales)) %>%
  mutate(sales_rel = (sales - loess)/mean_sales) %>%
  ungroup()

plt <- foo %>%
  filter(state_id == "CA" & cat_id == "FOODS") %>%
    mutate(wday = wday(dates, label = TRUE, week_start = 1),
         month = month(dates, label = TRUE),
         year = year(dates)) %>%
  group_by(wday, month, snap) %>%
  summarise(sales = sum(sales_rel)) %>%
  pivot_wider(names_from = "snap", values_from = "sales", names_prefix = "snap") %>%
  mutate(snap_effect = snapTRUE - snapFALSE) %>%
  ggplot(aes(month, wday, fill = snap_effect)) +
  geom_tile() +
  labs(x = "Month of the year", y = "Day of the week", fill = "SNAP effect") +
  scale_fill_distiller(palette = "Spectral") +
  theme_hc() +
  theme(legend.position = "bottom") +
  labs(x = "", y = "", title = "SNAP impact by weekday & month",
  subtitle = "Relative sales of SNAP days - other days.
  Only FOODS category and state CA.")

plt
```



We find:

- The heatmap focusses on FOODS and CA (because CA has the overall largest sales numbers). We see that overall the work days Mon-Fri show stronger benefits from SNAP purchases than the weekend Sat/Sun.
- Thursdays in November stand out. Thanksgiving is celebrated on the 4th Thursday in November every year. Here, this holiday most likely cuts into the "other" purchases and leads to the SNAP effect appearing artificially high.

Item Prices

We have item price information for each item ID, which includes the `category` and `department` IDs, and its `store` ID, which includes the `state` ID.

Here is a facet grid with overlapping density plots for price distributions within the groups of `category`, `department`, and `state`. Note the logarithmic scale on the x-axes:

```
id_changes <- prices %>%
  mutate(cat_id = str_sub(item_id, 1, -7)) %>%
  mutate(dept_id = str_sub(item_id, -5, -5)) %>%
  mutate(state = str_sub(store_id, 1, 2))

id_changes %>%
  ggplot(aes(sell_price, fill = dept_id)) +
```

```
  geom_density(bw = 0.1, alpha = 0.5) +
  scale_x_log10(breaks = c(0.5, 1, 5, 10, 50)) +
  coord_cartesian(xlim = c(0.3, 60)) +
  # facet_wrap(~ cat_id, nrow = 3) +
  facet_grid(cat_id ~ state) +
  theme_hc() +
  theme(legend.position = "bottom") +
  labs(x = "Average Sales Price [$]", y = "", fill = "Department",
       title = "Item Prices vary by Category and Department",
       subtitle = "But distributions are almost identical from State to State")
```

We find:

- The distributions are almost identical between the 3 states. There are some minute differences in the "FOODS" category, but we can treat the price distributions as equal.
- There are notable differences between the categories: FOODs are on average cheaper than HOUSEHOLD items. And HOBBIES items span a wider range of prices than the other two.
  - Among the three food categories, department 3 does not contain a high-price tail.
  - The HOBBIES category is the most diverse one, almost all of the items above $10.
  - The HOUSEHOLD price distributions are quite similar, but "HOUSEHOLD_2" peaks at clearly higher prices than "HOUSEHOLD_1".

Note: Every member of our team contributed equally on this analysis.

References: StackOverFlow, R Documentation some other interactive plots on kaggle.