May 2020

TEXAS
The University of Texas at Austin

What's in your cart?

Analytical database using 2017 Instacart order data and USDA ERS price database

Analalaura Rodriguez, Alexandra Smith
Elements of Databases, The University of Texas at Austin

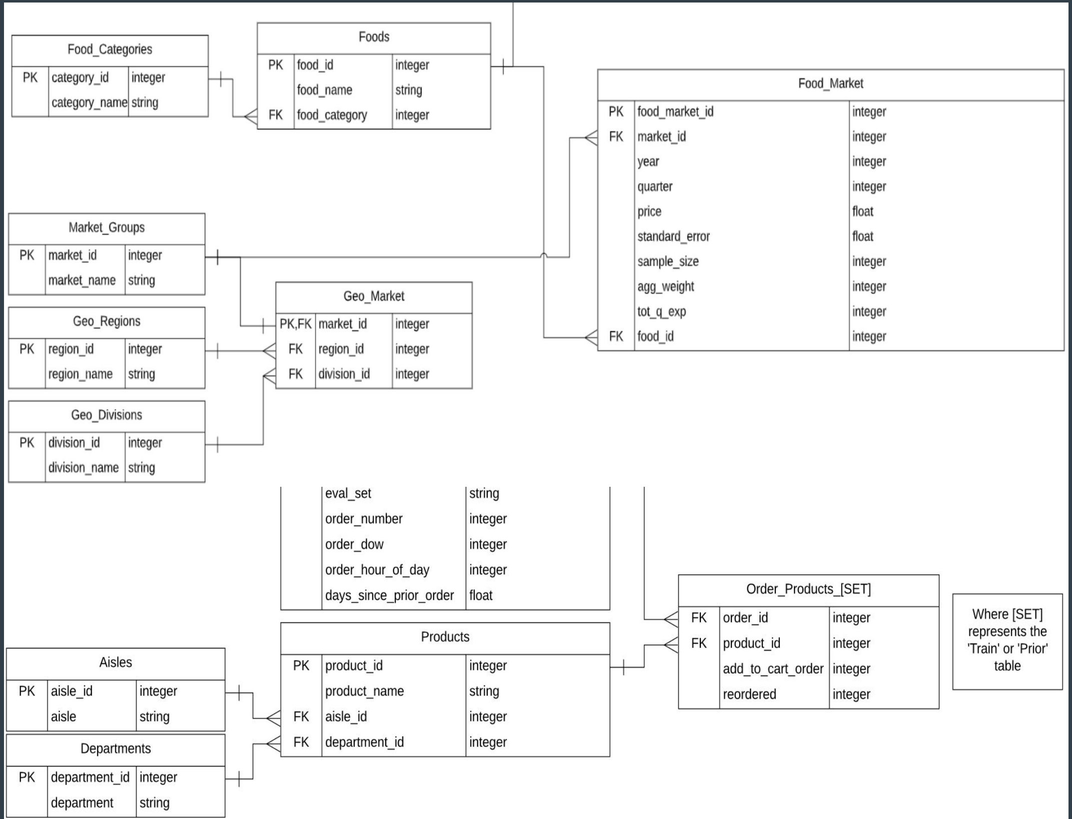
Mission

Analyze the spending trends of Instacart users to improve Instacart efficiency

Datasets

instacart

USDA



Food_Categories

PK	category_id	integer
	category_name	string

Foods

PK	food_id	integer
	food_name	string
FK	food_category	integer

Food_Market

PK	food_market_id	integer
FK	market_id	integer
	year	integer
	quarter	integer
	price	float
	standard_error	float
	sample_size	integer
	agg_weight	integer
	tot_q_exp	integer
FK	food_id	integer

Market_Groups

PK	market_id	integer
	market_name	string

Geo_Market

PK,FK	market_id	integer
FK	region_id	integer
FK	division_id	integer

Geo_Regions

PK	region_id	integer
	region_name	string

Geo_Divisions

PK	division_id	integer
	division_name	string

	eval_set	string
	order_number	integer
	order_dow	integer
	order_hour_of_day	integer
	days_since_prior_order	float

Order_Products_[SET]

FK	order_id	integer
FK	product_id	integer
	add_to_cart_order	integer
	reordered	integer

Where [SET] represents the 'Train' or 'Prior' table

Products

PK	product_id	integer
	product_name	string
FK	aisle_id	integer
FK	department_id	integer

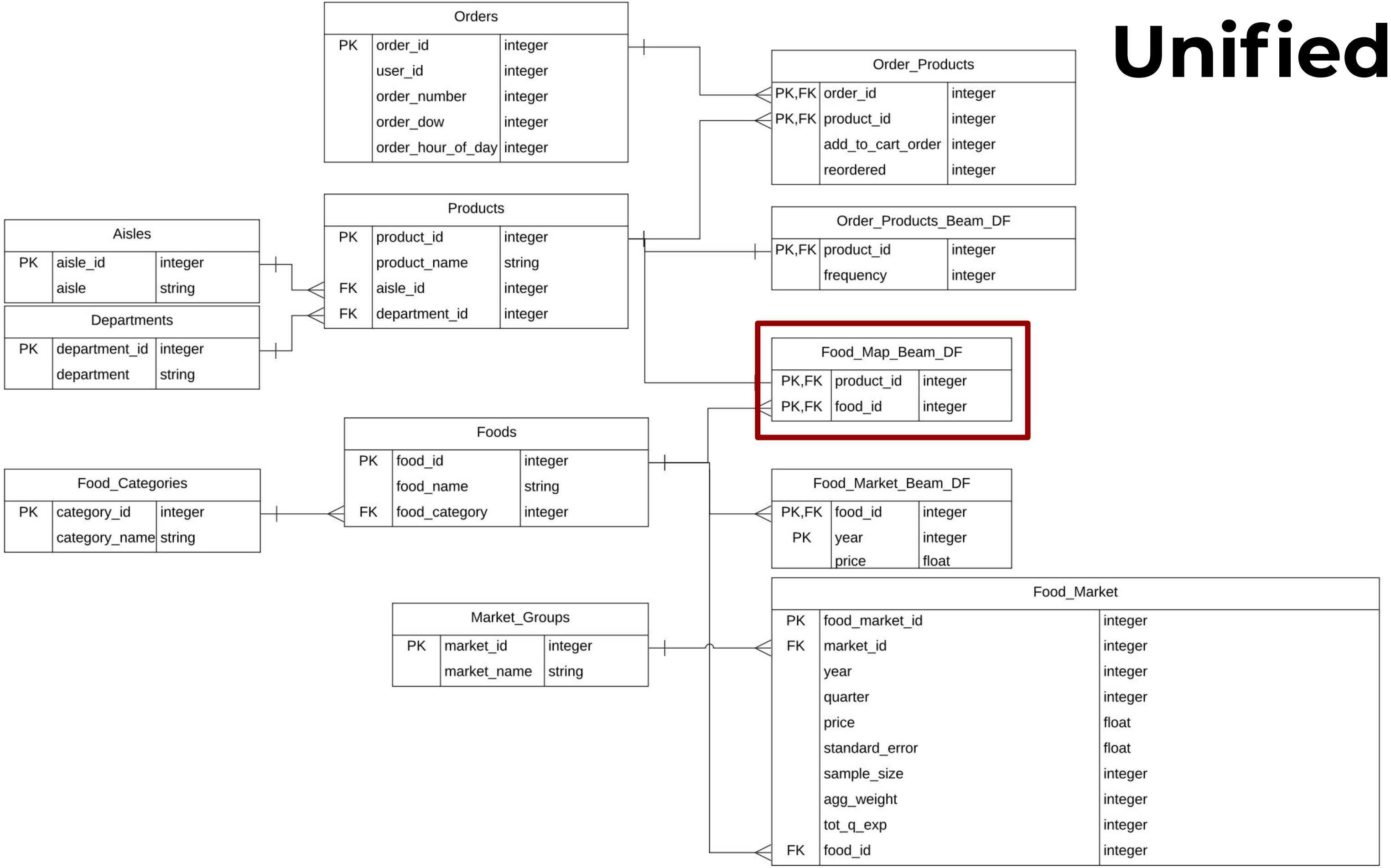
Aisles

PK	aisle_id	integer
	aisle	string

Departments

PK	department_id	integer
	department	string

Unified ERD



Orders		
PK	order_id	integer
	user_id	integer
	order_number	integer
	order_dow	integer
	order_hour_of_day	integer

Order_Products		
PK,FK	order_id	integer
PK,FK	product_id	integer
	add_to_cart_order	integer
	reordered	integer

Products		
PK	product_id	integer
	product_name	string
FK	aisle_id	integer
FK	department_id	integer

Aisles		
PK	aisle_id	integer
	aisle	string

Departments		
PK	department_id	integer
	department	string

Order_Products_Beam_DF		
PK,FK	product_id	integer
	frequency	integer

Food_Map_Beam_DF		
PK,FK	product_id	integer
PK,FK	food_id	integer

Foods		
PK	food_id	integer
	food_name	string
FK	food_category	integer

Food_Categories		
PK	category_id	integer
	category_name	string

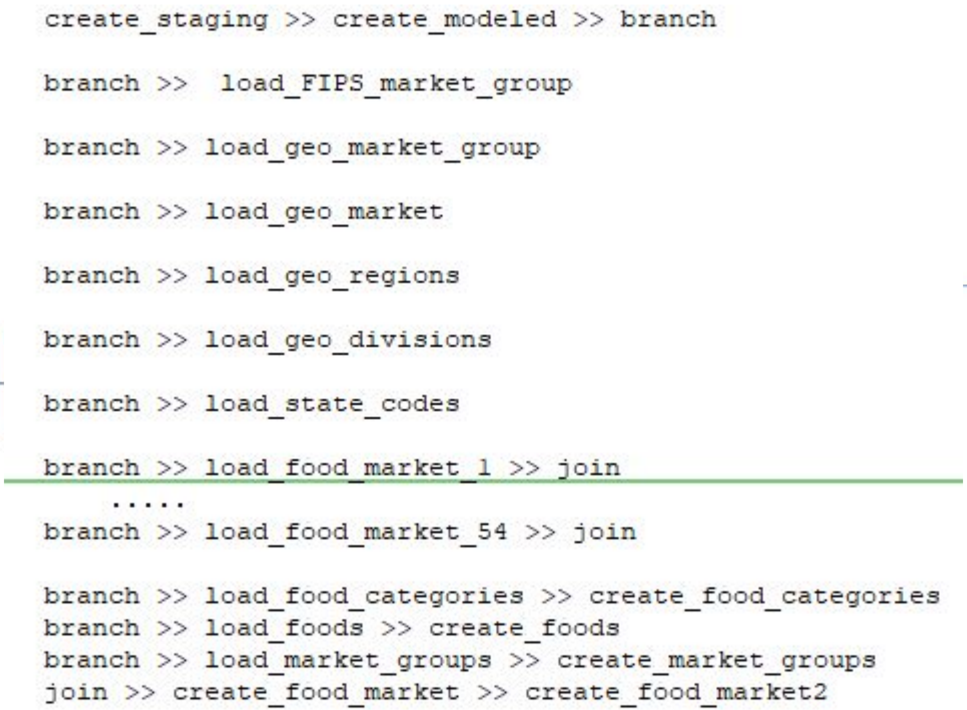
Food_Market_Beam_DF		
PK,FK	food_id	integer
PK	year	integer
	price	float

Market_Groups		
PK	market_id	integer
	market_name	string

Food_Market		
PK	food_market_id	integer
FK	market_id	integer
	year	integer
	quarter	integer
	price	float
	standard_error	float
	sample_size	integer
	agg_weight	integer
	tot_q_exp	integer
FK	food_id	integer

Airflow with Secondary Dataset

create_staging



```
create_staging >> create_modeled >> branch

branch >>  load_FIPS_market_group

branch >> load_geo_market_group

branch >> load_geo_market

branch >> load_geo_regions

branch >> load_geo_divisions

branch >> load_state_codes

branch >> load_food_market_1 >> join
    .....
branch >> load_food_market_54 >> join

branch >> load_food_categories >> create_food_categories
branch >> load_foods >> create_foods
branch >> load_market_groups >> create_market_groups
join >> create_food_market >> create_food_market2
```

Add price column based on linear regression

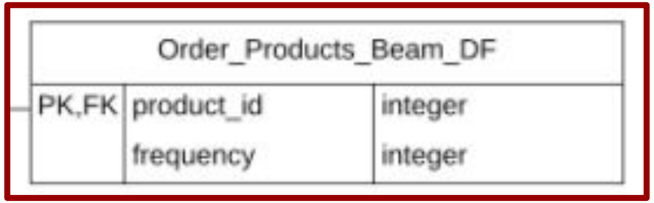
$$price = a\,(year) + b$$

Slope $a = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$

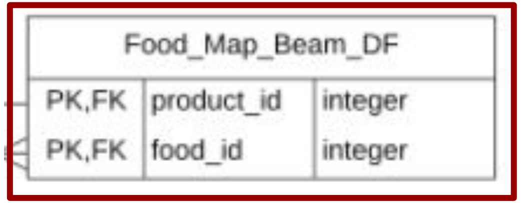
intercept $b = \bar{y} - a\bar{x}$

__where overbar denotes average

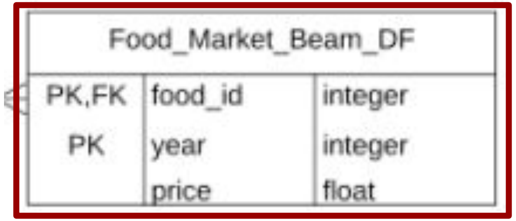
Beam Transforms



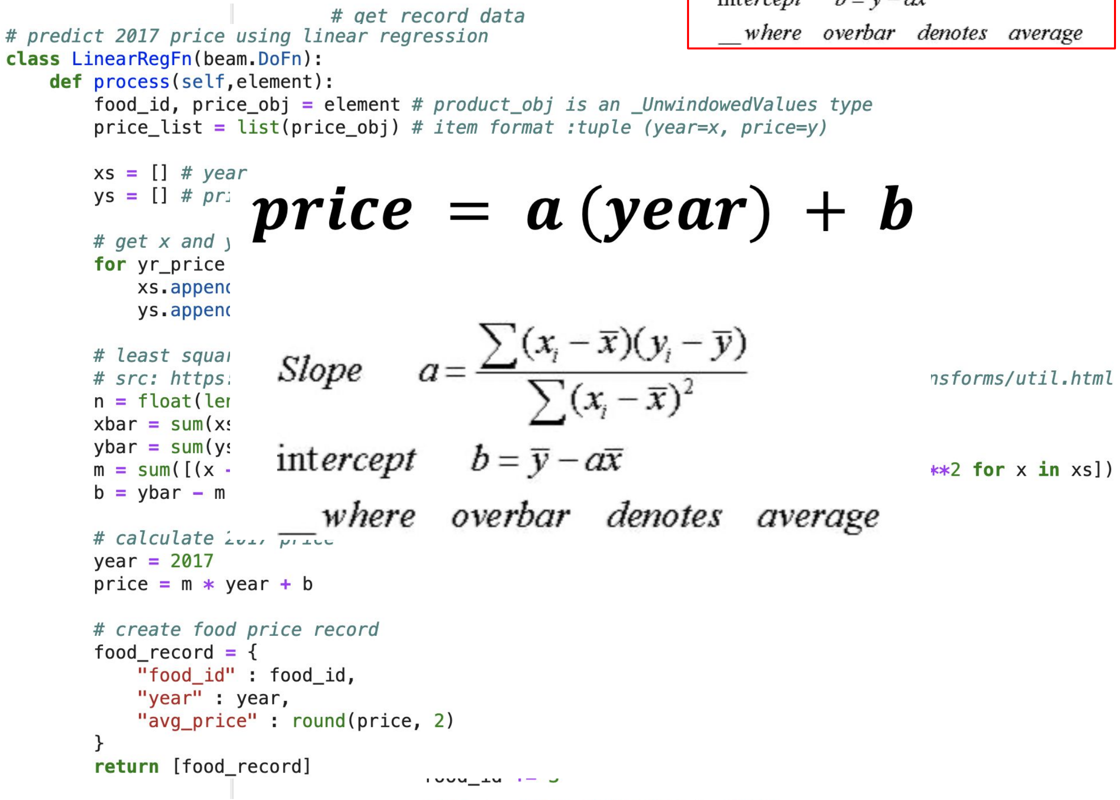
Order_Products_Beam_DF		
PK,FK	product_id	integer
	frequency	integer



Food_Map_Beam_DF		
PK,FK	product_id	integer
PK,FK	food_id	integer



Food_Market_Beam_DF		
PK,FK	food_id	integer
PK	year	integer
	price	float



```
               # get record data
# predict 2017 price using linear regression
class LinearRegFn(beam.DoFn):
    def process(self,element):
        food_id, price_obj = element # product_obj is an _UnwindowedValues type
        price_list = list(price_obj) # item format :tuple (year=x, price=y)

        xs = [] # year
        ys = [] # pri

        # get x and y
        for yr_price
            xs.append
            ys.append

        # least squa
        # src: https:                                                nsforms/util.html
        n = float(len
        xbar = sum(xs
        ybar = sum(ys
        m = sum([(x -                                                **2 for x in xs])
        b = ybar - m

        # calculate 2017 price
        year = 2017
        price = m * year + b

        # create food price record
        food_record = {
            "food_id" : food_id,
            "year" : year,
            "avg_price" : round(price, 2)
        }
        return [food_record]
```

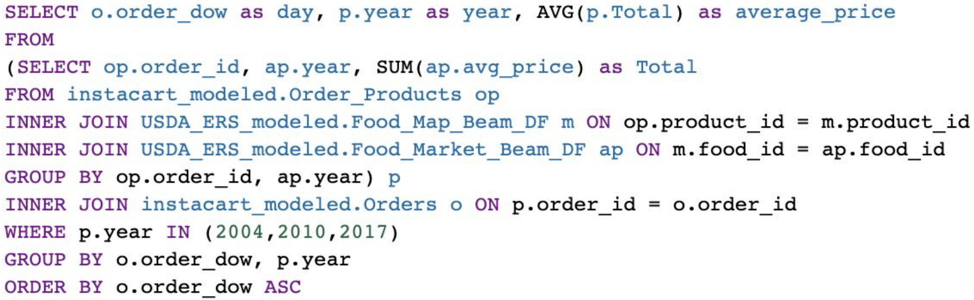
$$price = a\,(year) + b$$

Slope $a = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$

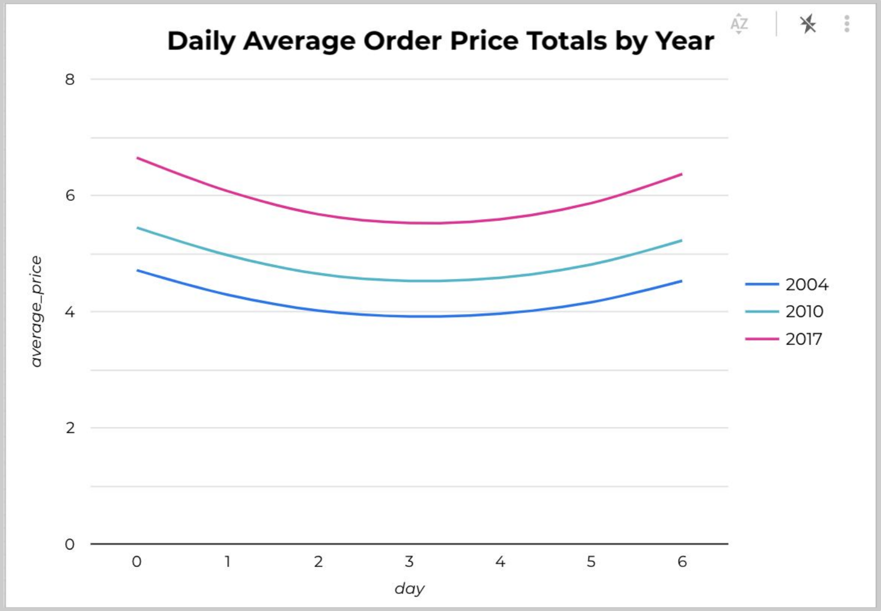
intercept $b = \bar{y} - a\bar{x}$

__where overbar denotes average

Daily Average Order Price Totals by Year



```
SELECT o.order_dow as day, p.year as year, AVG(p.Total) as average_price
FROM
(SELECT op.order_id, ap.year, SUM(ap.avg_price) as Total
FROM instacart_modeled.Order_Products op
INNER JOIN USDA_ERS_modeled.Food_Map_Beam_DF m ON op.product_id = m.product_id
INNER JOIN USDA_ERS_modeled.Food_Market_Beam_DF ap ON m.food_id = ap.food_id
GROUP BY op.order_id, ap.year) p
INNER JOIN instacart_modeled.Orders o ON p.order_id = o.order_id
WHERE p.year IN (2004,2010,2017)
GROUP BY o.order_dow, p.year
ORDER BY o.order_dow ASC
```



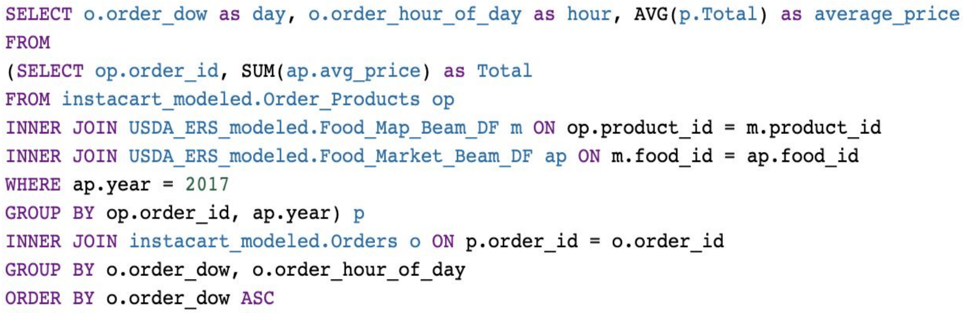
Results:

- Prices have increased dramatically since 2004
- Users spend more on days 0 and 6

Action:

- more shoppers on days 0 and 6

2017 Daily and Hourly User Spending



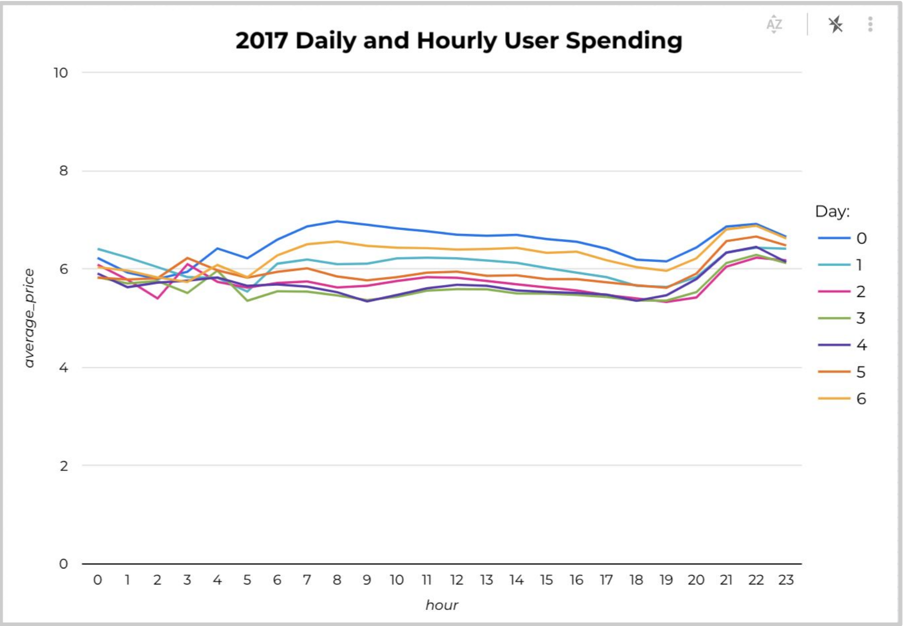
```
SELECT o.order_dow as day, o.order_hour_of_day as hour, AVG(p.Total) as average_price
FROM
(SELECT op.order_id, SUM(ap.avg_price) as Total
FROM instacart_modeled.Order_Products op
INNER JOIN USDA_ERS_modeled.Food_Map_Beam_DF m ON op.product_id = m.product_id
INNER JOIN USDA_ERS_modeled.Food_Market_Beam_DF ap ON m.food_id = ap.food_id
WHERE ap.year = 2017
GROUP BY op.order_id, ap.year) p
INNER JOIN instacart_modeled.Orders o ON p.order_id = o.order_id
GROUP BY o.order_dow, o.order_hour_of_day
ORDER BY o.order_dow ASC
```

Results

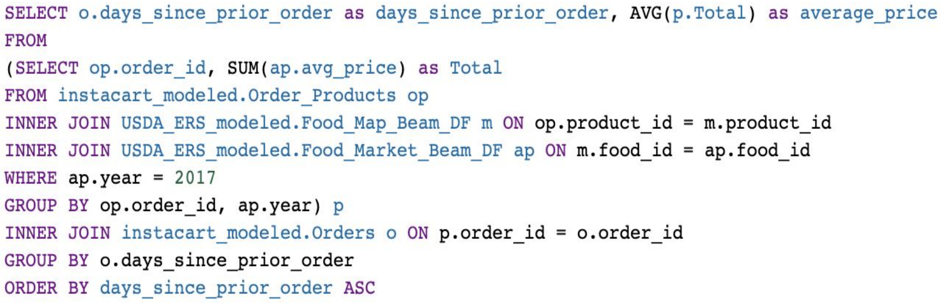
- Similar hourly sales activity across all days
  - 7- 17 hrs steady activity
  - peak at 21-22 hrs
- Day 0 and 1 most active

Action

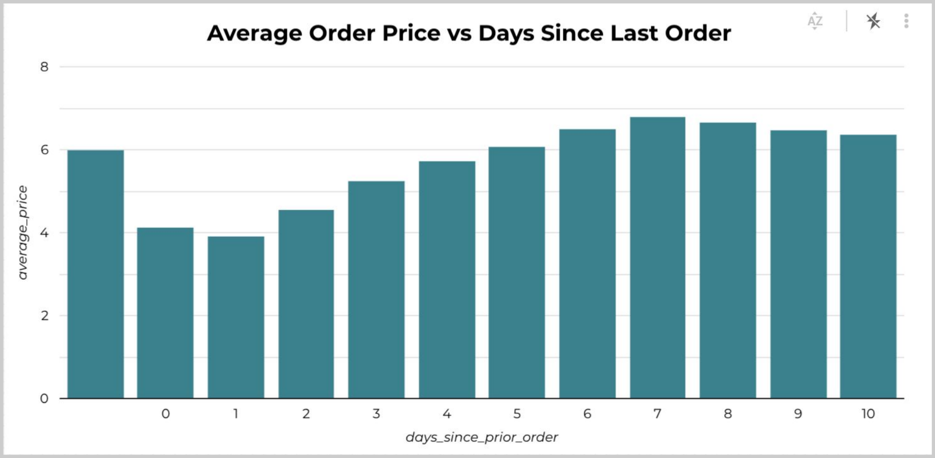
- increase shoppers on days/hours



2017 Daily and Hourly User Spending



```
SELECT o.days_since_prior_order as days_since_prior_order, AVG(p.Total) as average_price
FROM
(SELECT op.order_id, SUM(ap.avg_price) as Total
FROM instacart_modeled.Order_Products op
INNER JOIN USDA_ERS_modeled.Food_Map_Beam_DF m ON op.product_id = m.product_id
INNER JOIN USDA_ERS_modeled.Food_Market_Beam_DF ap ON m.food_id = ap.food_id
WHERE ap.year = 2017
GROUP BY op.order_id, ap.year) p
INNER JOIN instacart_modeled.Orders o ON p.order_id = o.order_id
GROUP BY o.days_since_prior_order
ORDER BY days_since_prior_order ASC
```



Results:

- ordering on a weekly basis is more profitable
- least profitable
  - `days_since_prior_order = 1,2, or 3`

Action:

- target them to raise sales
  - reminders, coupons, promotions

Improvements

- More sophisticated/automated algorithm to create Food-Product mapping
- Extend mapping to all products (incl. non-foods)
- Instacart order locations
    - Product sales fluctuation based on price