Data wrangling

S01-import.py code

Import dataframes

Here we'll import the dataframes in the directory and show the head of each.

- Note that the directory /data/raw may contain a couple of other files like 'test.csv' or 'sample_submission.csv', but they don't add any new information for our current purposes.

train.csv

This is the most important file - information by store by date.

- Sales is our target variable.
- Customers is the number of customers on that date.
- Open and Promo are as they sound, and we don't have more information than that.
- StateHoliday and SchoolHoliday are as they sound. Note that StateHoliday can be a (public holiday), b (Easter), c (Christmas), or 0 (none).

	Store	DayOfWeek	Date	Sales	Customers	Ope
1	1	5	2015-07-31	5263	555	1
2	2	5	2015-07-31	6064	625	1
3	3	5	2015-07-31	8314	821	1
4	4	5	2015-07-31	13995	1498	1
5	5	5	2015-07-31	4822	559	1

store.csv

This file has information about each particular store.

- Store type: one of 'a', 'b', 'c', 'd'
- Assortment: 'a' = basic, 'b' = extra, 'c' = extended
- CompetitionDistance is in meters. The closest competitor to any given store is 20 meters, while the furthest 'closest competitor' is nearly 50 miles from a Rossmann store.
- CompetitionOpenSinceMonth and Year are as they sound. Most of the competitors have been opened relatively recently.
- Promo2: according to data/raw/description.txt, Promo2 is a continuing and consecutive promotion for some stores. 0 = not participating, 1 = participating.
- Promo2SinceWeek and Year are as they sound. Note that if Promo2 = 0, a NaN value is meaningful here.
- PromoInterval: relative to Promo2. Options are 'Jan,Apr,Jul,Oct', 'Feb,May,Aug,Nov', 'Mar,Jun,Sept,Dec' - note that 'Sept' here has 4 characters.

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth

weather.csv

This file has weather by state and date.

- file: This is the name of the state rather than its abbreviation, which necessitates the use of the store_states csv later.
- Other than Date, the rest of the file is various weather measurements.

	file	Date	Max_TemperatureC	Mean_TemperatureC	Min_Temperatur
1	NordrheinWestfalen	2013-01-01	8	4	2
2	NordrheinWestfalen	2013-01-02	7	4	1
3	NordrheinWestfalen	2013-01-03	11	8	6
4	NordrheinWestfalen	2013-01-04	9	9	8
5	NordrheinWestfalen	2013-01-05	8	8	7

googletrend.csv

This file has google search trends by state and date.

- file: This is the state abbreviation, along with some other characters that we'll strip out.
- week: This is the week of the measurement.
- trend: this is the trend, which we'll concatenate to our dataframe.

	file	w
1	Rossmann_DE_SN	2012-12-02 - 2012-12-08
2	Rossmann_DE_SN	2012-12-09 - 2012-12-15
3	Rossmann_DE_SN	2012-12-16 - 2012-12-22
4	Rossmann_DE_SN	2012-12-23 - 2012-12-29
5	Rossmann_DE_SN	2012-12-30 - 2013-01-05

store_states.csv

This file lists the state that each store is in, so we can merge the dataframes together.

	Store	
1	1	HE
2	2	TH
3	3	NW

4	4	BE

state_names.csv

This file lists state names and abbreviations, so we can merge the dataframes together.

	StateName
1	BadenWuerttemberg
2	Bayern
3	Berlin
4	Brandenburg
5	Bremen

S02.py code

Merge dataframes

In order to merge the dataframes, we did the following:

- Cleaned each dataframe individually
    - For train.csv, store_states.csv, and state_names.csv, it was just making column names consistent
    - For googletrend.csv, it was fixing 'file' to be legitimate state names and changing the 'week' format into actual dates
    - For store.csv, it was replacing NaNs with the mean of the column
    - For weather.csv, there were a few mistyped column names, and some NaNs that had to be replaced

We end up with a dataframe with 1,050,330 rows: there are 942 stores, and there are 942 days from 2013-01-01 to 2015-07-31, so we have 942 * 1115 = 1,050,330 rows.

Our table has 43 columns.

	store	state	date	max_temperature_c	mean_temperature_c	min_temperature_
1	1	HE	2013-01-01	8	6	3
2	56	HE	2013-01-01	8	6	3
3	69	HE	2013-01-01	8	6	3
4	77	HE	2013-01-01	8	6	3
5	111	HE	2013-01-01	8	6	3