Weekly Sales Forecasting

San Diego Data Science and R Users Group | June 2014

Kevin Davenport
http://kldavenport.com
kldavenportjr@gmail.com
@KevinLDavenport

Thank you to our sponsors:

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

REVOLUTION ANALYTICS

ansir innovation center
startup accelerator

The competition

Use historical markdown data to predict store sales. Predict weekly sales for a set of store & department pairs in 2013 given the weekly sales from 2010-2012.

The Data

1. train/test: weekly sales for each combination of store, department, and date.

2. features: markdowns, CPI, unemployment rate, temperature, fuel price, and holiday (T/F)

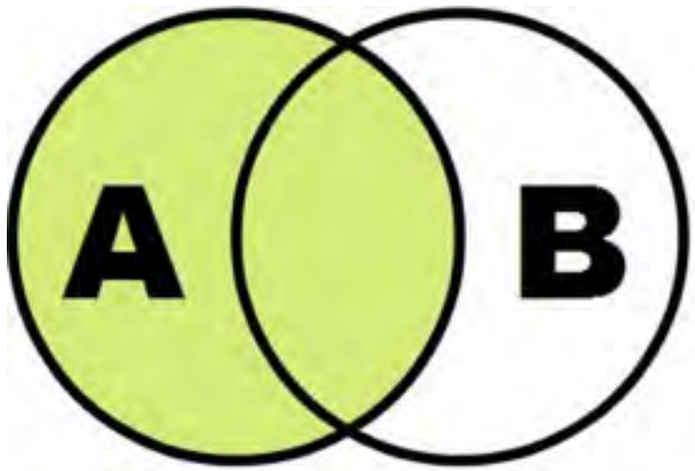
3. stores: size and type of each store

Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
1	2010-02-05	42.31	2.572	NA	NA	NA	NA	NA	211.0963582	8.106	FALSE
1	2010-02-12	38.51	2.548	NA	NA	NA	NA	NA	211.2421698	8.106	TRUE
1	2010-02-19	39.93	2.514	NA	NA	NA	NA	NA	211.2891429	8.106	FALSE
1	2010-02-26	46.63	2.561	NA	NA	NA	NA	NA	211.3196429	8.106	FALSE
1	2010-03-05	46.5	2.625	NA	NA	NA	NA	NA	211.3501429	8.106	FALSE
1	2010-03-12	57.79	2.667	NA	NA	NA	NA	NA	211.3806429	8.106	FALSE
1	2010-03-19	54.58	2.72	NA	NA	NA	NA	NA	211.215635	8.106	FALSE
1	2010-03-26	51.45	2.732	NA	NA	NA	NA	NA	211.0180424	8.106	FALSE
1	2010-04-02	62.27	2.719	NA	NA	NA	NA	NA	210.8204499	7.808	FALSE

Store	Dept	Date	Weekly_Sales	IsHoliday
1	1	2010-02-05	24924.5	FALSE
1	1	2010-02-12	46039.49	TRUE
1	1	2010-02-19	41595.55	FALSE
1	1	2010-02-26	19403.54	FALSE
1	1	2010-03-05	21827.9	FALSE
1	1	2010-03-12	21043.39	FALSE
1	1	2010-03-19	22136.64	FALSE
1	1	2010-03-26	26229.21	FALSE
1	1	2010-04-02	57258.43	FALSE

Store	Type	Size
1	A	151315
2	A	202307
3	B	37392
4	A	205863

Joining the datasets




```
SELECT *
FROM A
LEFT JOIN B
ON A.id = B.id
```

```
Select *
FROM Train_Xdf A
LEFT JOIN addtl_features B
ON A.Store = B.Store
AND A.Date = B.Date
```

```
X_traindf = pd.merge(X_traindf, X_addtl_feat, how='left', on=['Store', 'Date'])
X_traindf = pd.merge(X_traindf, stores, how='left', on='Store')

X_testdf = pd.merge(X_testdf, X_addtl_feat, how='left', on=['Store', 'Date'])
X_testdf = pd.merge(X_testdf, stores, how='left', on='Store')
```

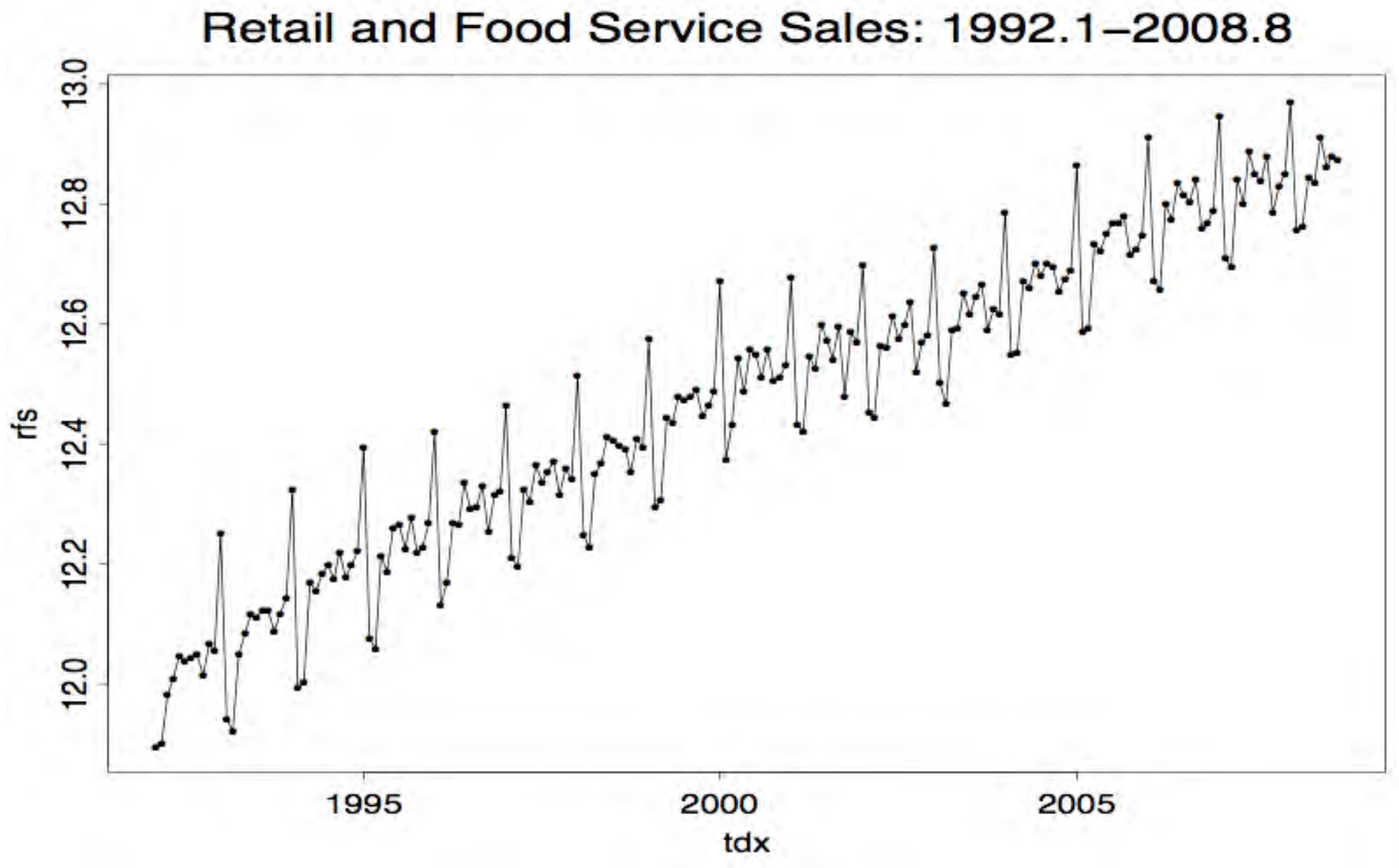
http://pandas.pydata.org/pandas-docs/stable/comparison_with_sql.html#compare-with-sql-join

Time Series and Forecasting

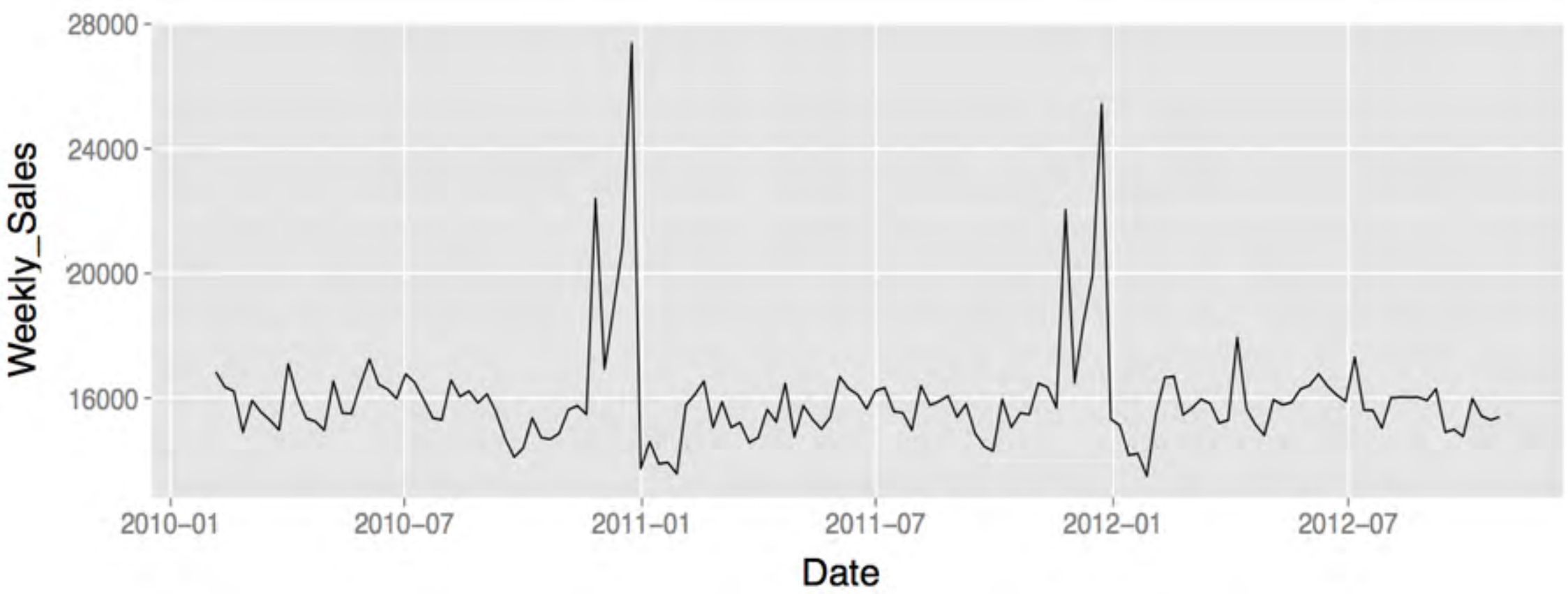
Let's assume that a relationship between all observations or measurements can be expressed by a linear or non-linear function.

Consider a measurement of 100 RPM @ 11:00am, it's highly probable that the observation before or after it (10:59am and 11:01am) are close to 100 RPM.

Seasonality vs Trend



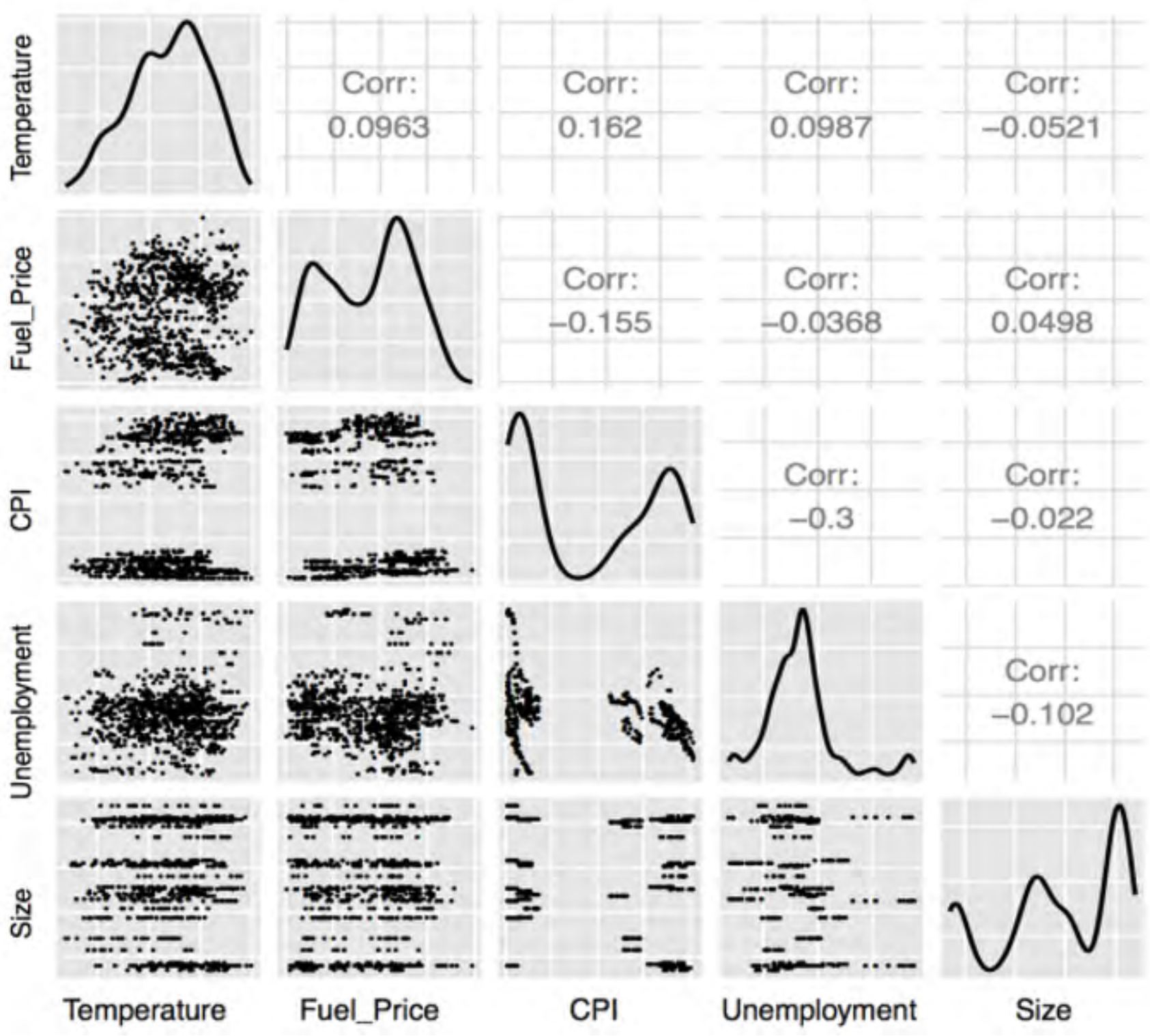
Seasonality vs Trend



Average sales of all stores/departments per week.

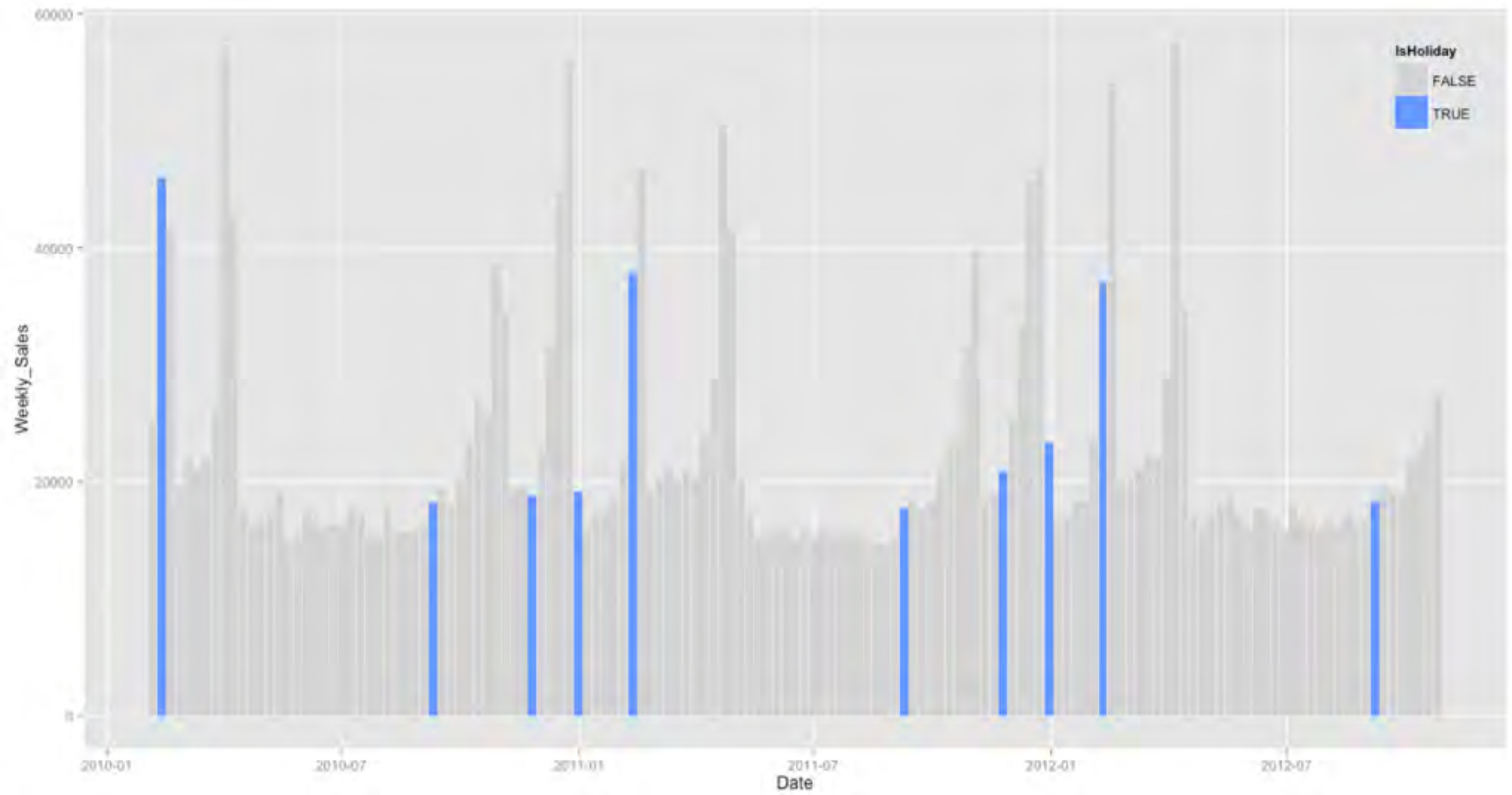
Two significant peaks at November and December.

Correlation and Distribution



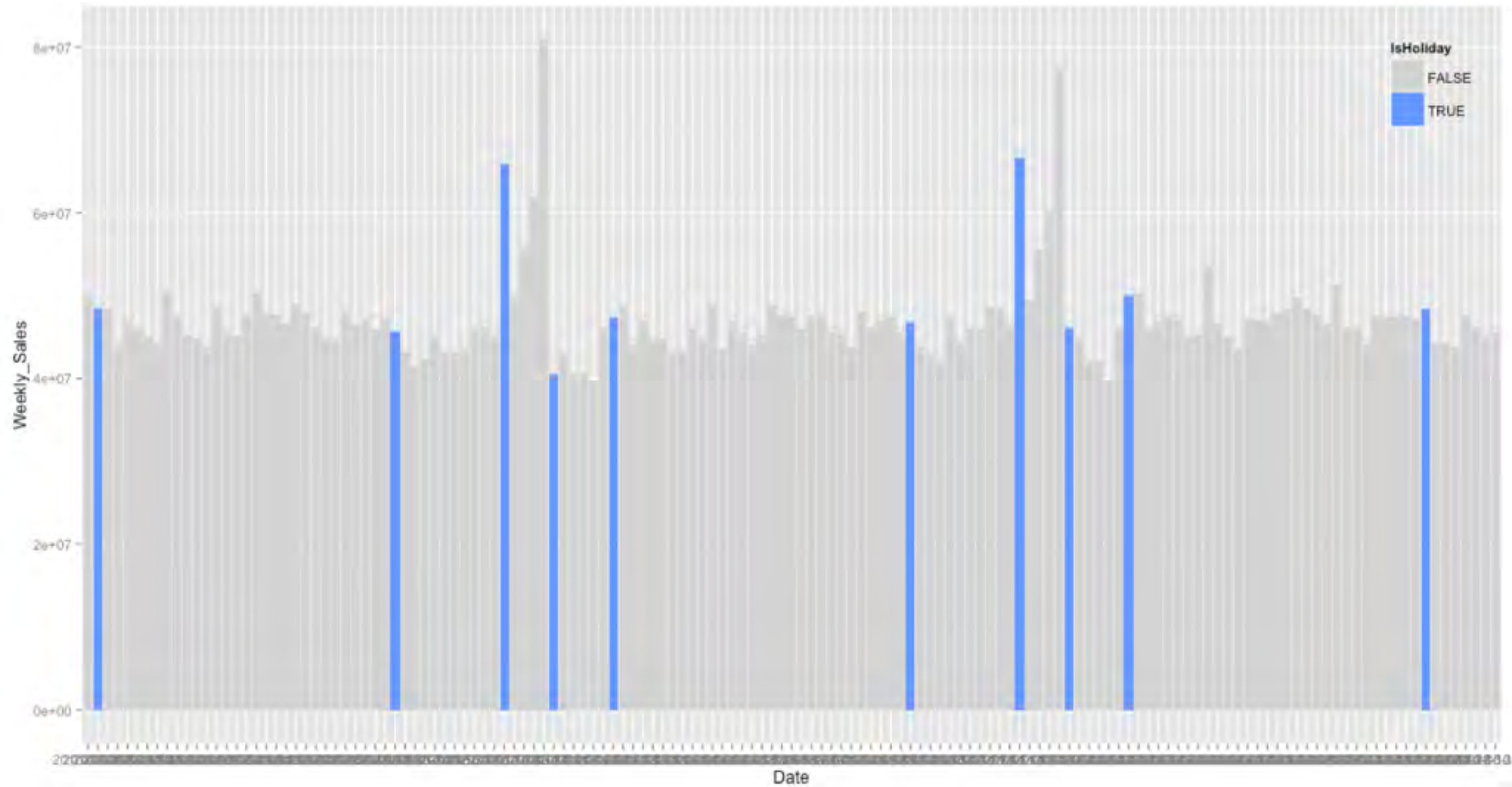
http://docs.ggplot2.org/0.9.3.1/plotmatrix.html#

Holidays



One randomly selected store/department

Holidays



All stores/departments

Feature Engineering

1. Naive Previous Year's Sales
2. Smoothed Sales
3. Additional features (features.csv)

Numerical Features

Scale

1. Consumer Price Index (CPI)
2. Fuel price
3. Temperature
4. Markdowns
5. Unemployment rate
6. Store size

Dummy encode

1. Department
2. Holiday
3. Store type
4. Date

Dummy encoding

```
>>> s1 = ['a', 'b', np.nan]
```

```
>>> get_dummies(s1)
   a  b
0  1  0
1  0  1
2  0  0
```

```
>>> get_dummies(s1, dummy_na=True)
   a  b  NaN
0  1  0    0
1  0  1    0
2  0  0    1
```

Date Encoding

The Date column can be binarily encoded to create the following features:

- Hour of the day (24 boolean features)

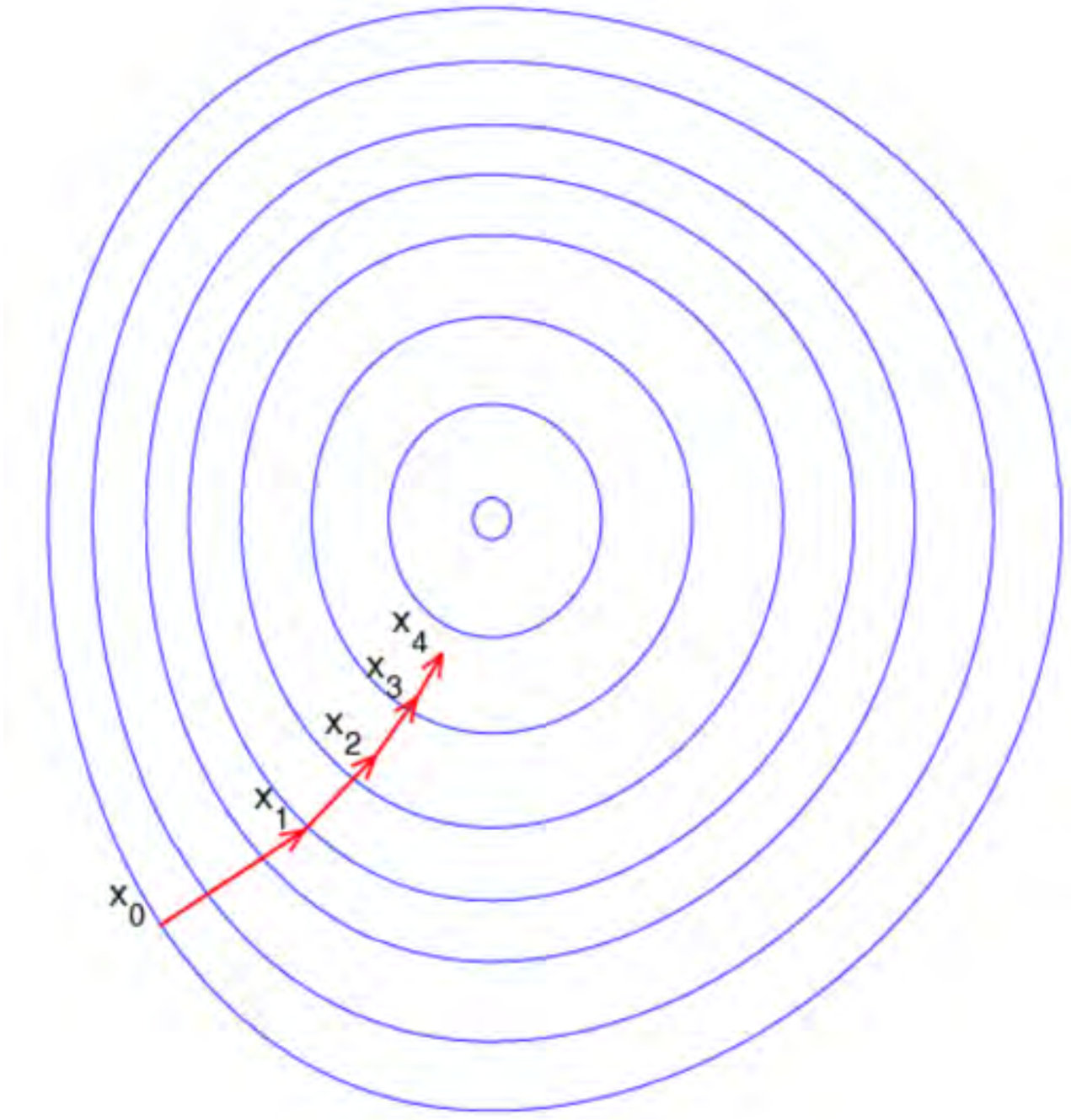
- Day of the week (7 boolean features)

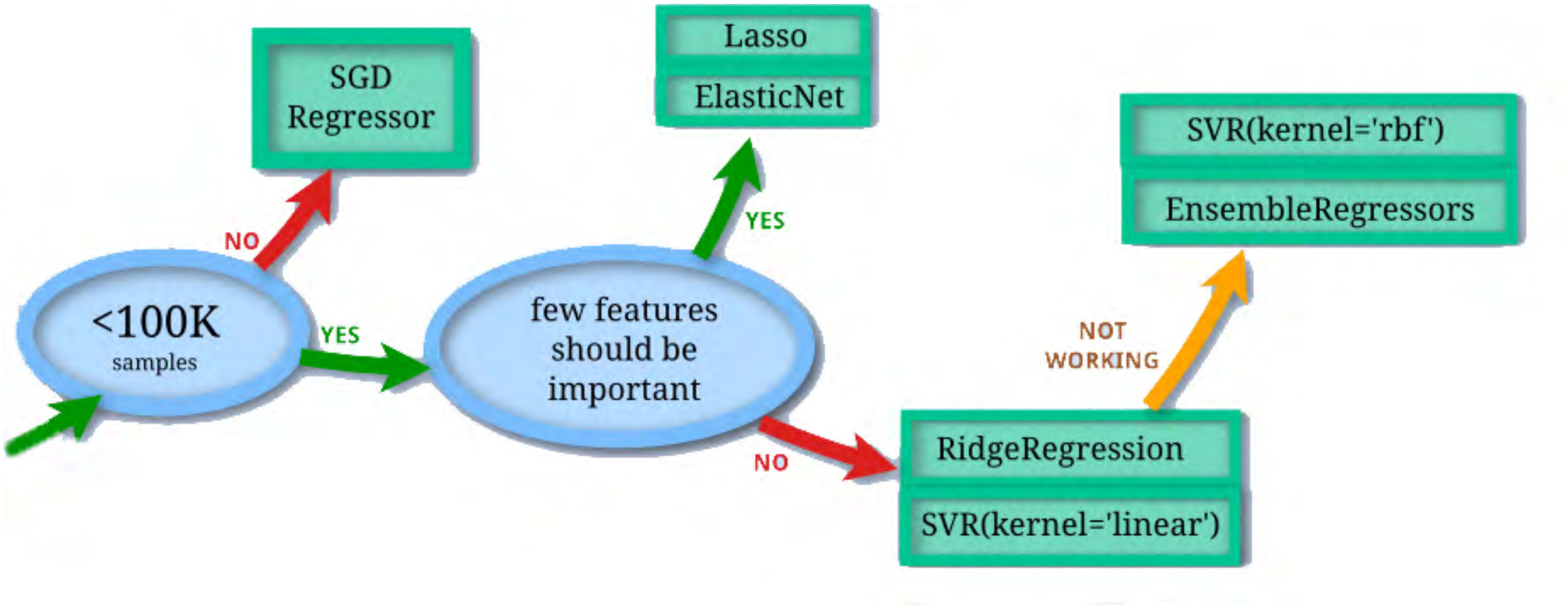
- Day of the month (up to 31 boolean features)

- Month of the year (12 boolean features)

- Year (as many boolean features as they are different years in your dataset)

Gradient Descent





the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning with the partial_fit method.

Gradient Descent

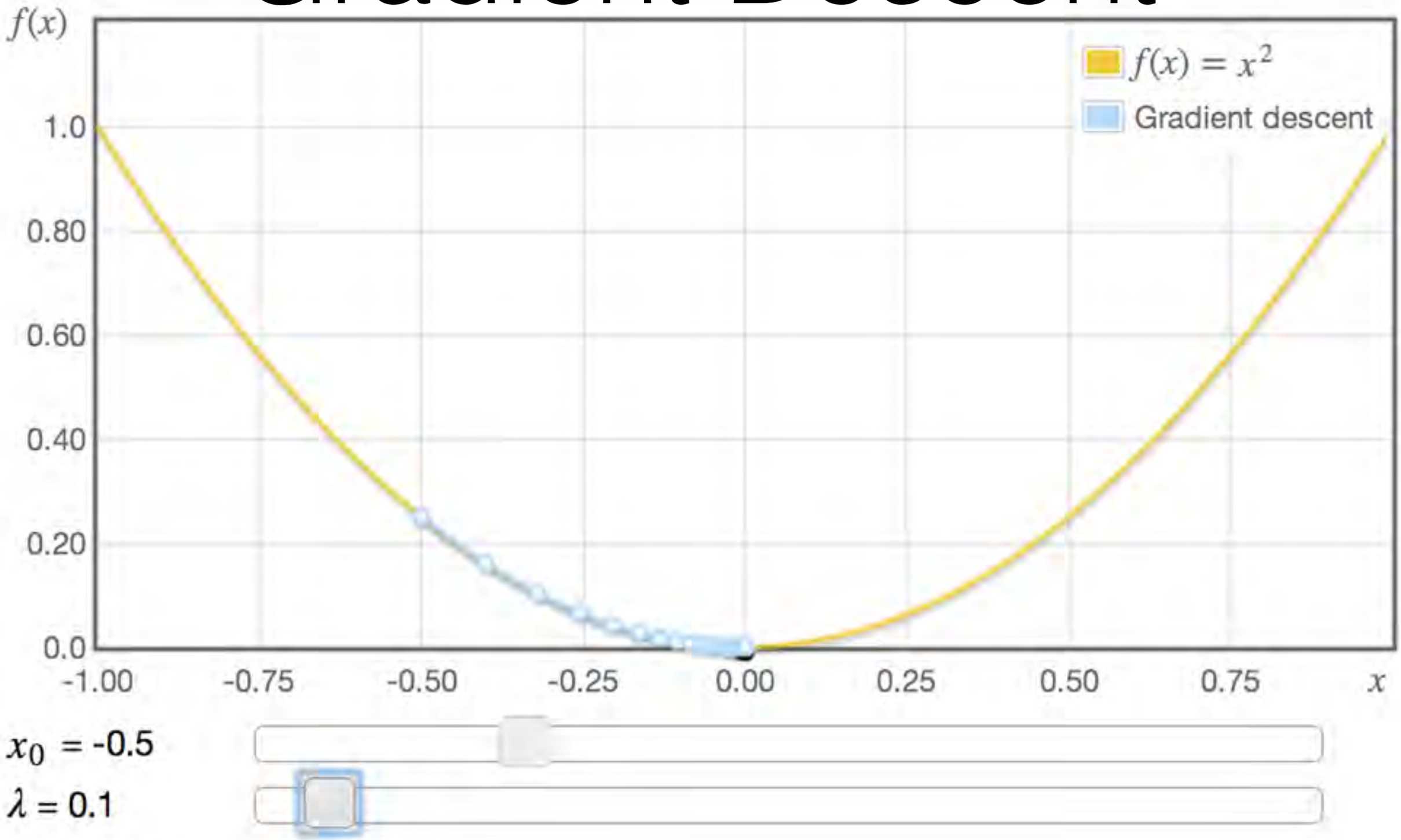
Gradient Descent - consider entire dataset

Stochastic Gradient Descent - Each step pick *one random data point* continue as if entire dataset was just the one point

Minibatch SGD - Each step pick *small subset* of data points continue as if entire dataset was just the this subset

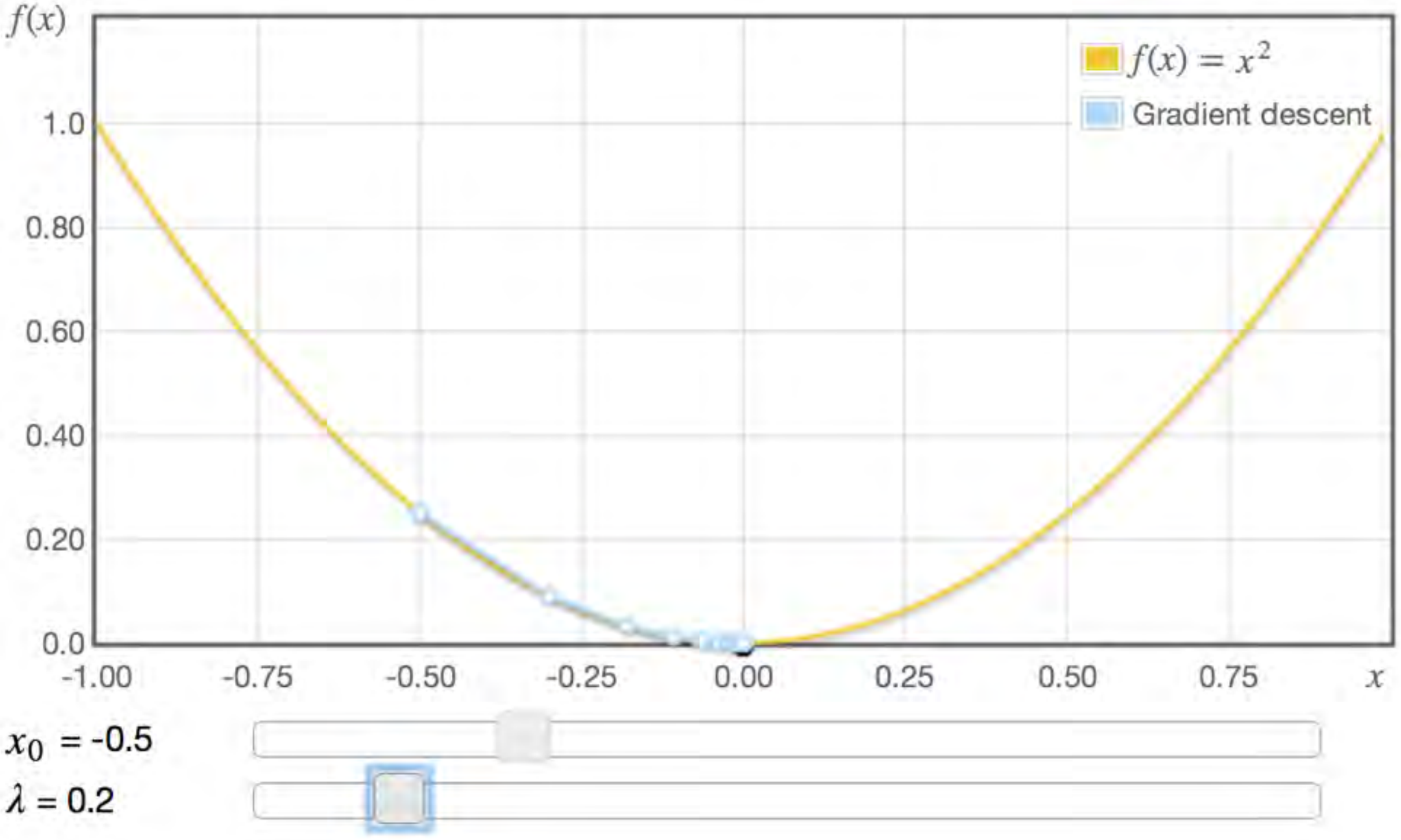
http://www.libfm.org/

Gradient Descent

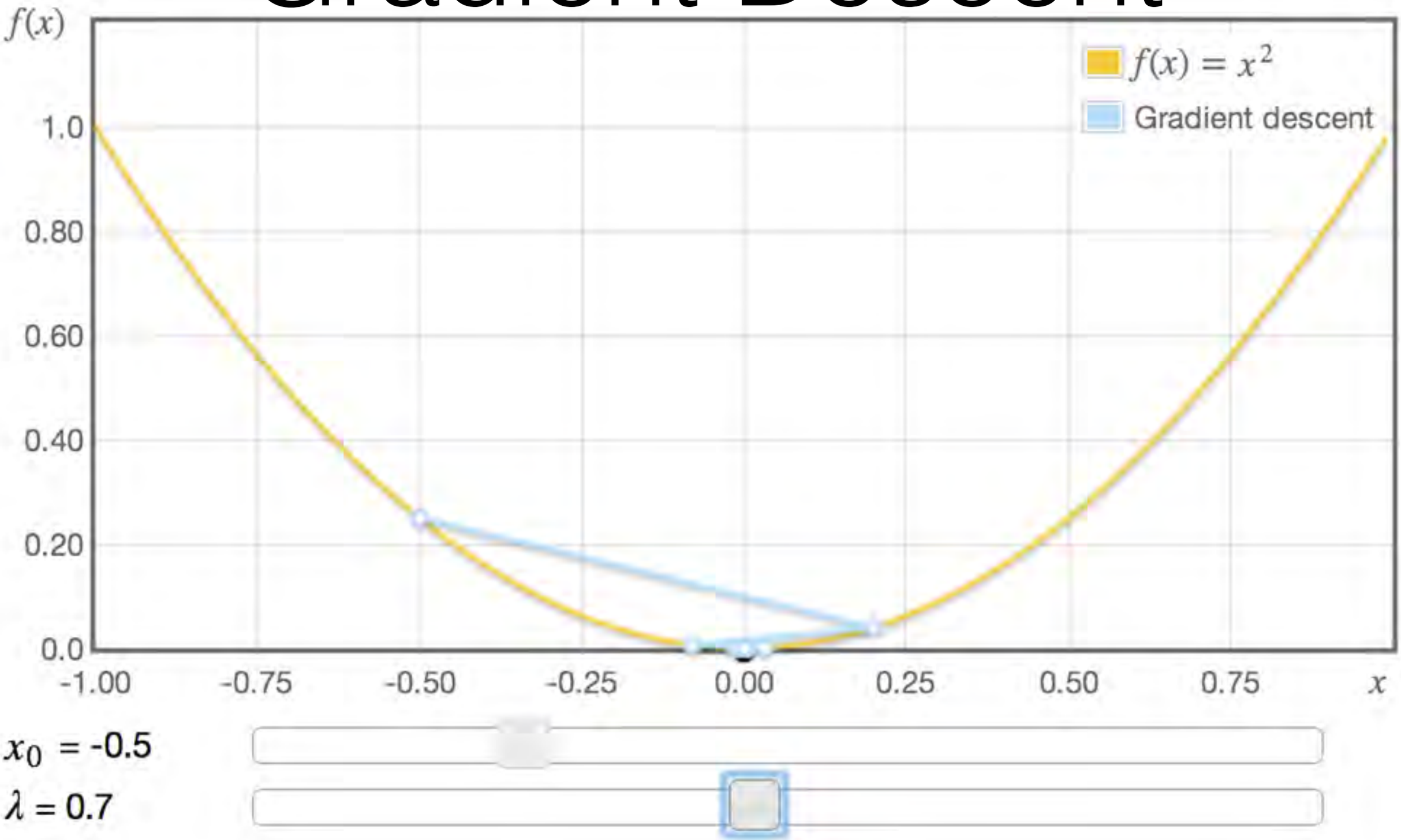


http://www.onmyphd.com/?p=gradient.descent

Gradient Descent



Gradient Descent



libFM

Factorization machines (FM) are a generic approach that allows to mimic most factorization models by feature engineering. This way, factorization machines combine the generality of feature engineering with the superiority of factorization models in estimating interactions between categorical variables of large domain

http://www.libfm.org/
haven't tried: https://github.com/coreylynch/pyFM

Evaluation

1. Smoothed last year's sales.
2. Ridge regression
3. Factorization machines

Please Donate to numfocus.org