|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | |  |
| April 2015 | |  |
| Walmart Recruiting - Sales in stormy weather | |  |
| Notes | |  |
|  | |  |
| Isaac | |  |
| Authored by: Gino Tesei | |  |
|  | |  |

Exploratory analysis

# Weather

Exploratory analysis on weather data has been performed and related results are shown in the Excel document [weather\_elab.xlsx](https://github.com/gtesei/fast-furious/blob/master/competitions/walmart-recruiting-sales-in-stormy-weather/weather_elab.xlsx)

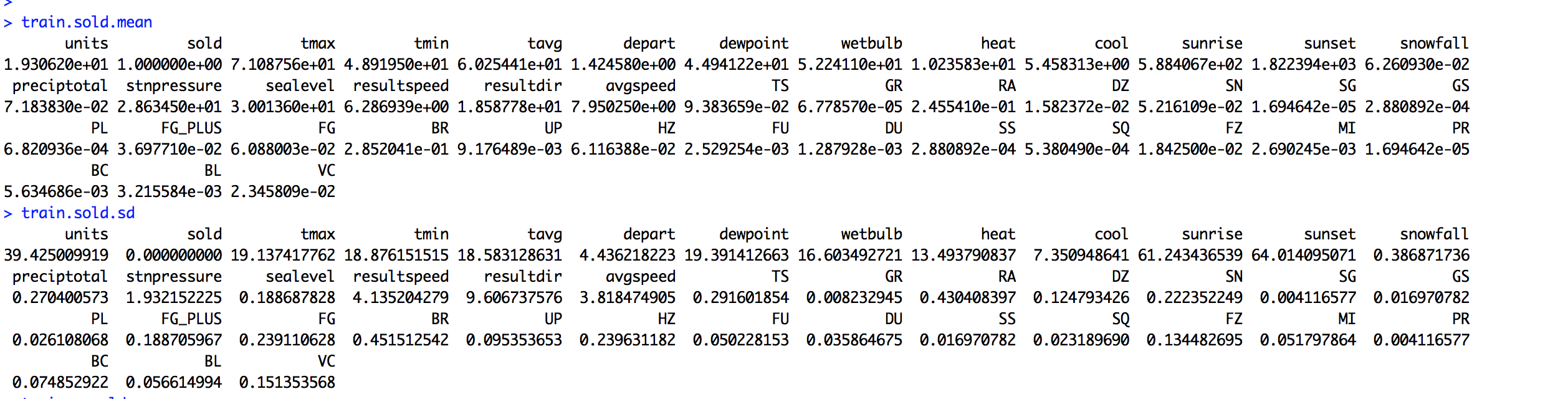
# Stores and products

In training data there’s only 2.5% of sold <units> with a value > 0, i.e. 97.5% of <date,store,item> in the training are 0s (=unsold).

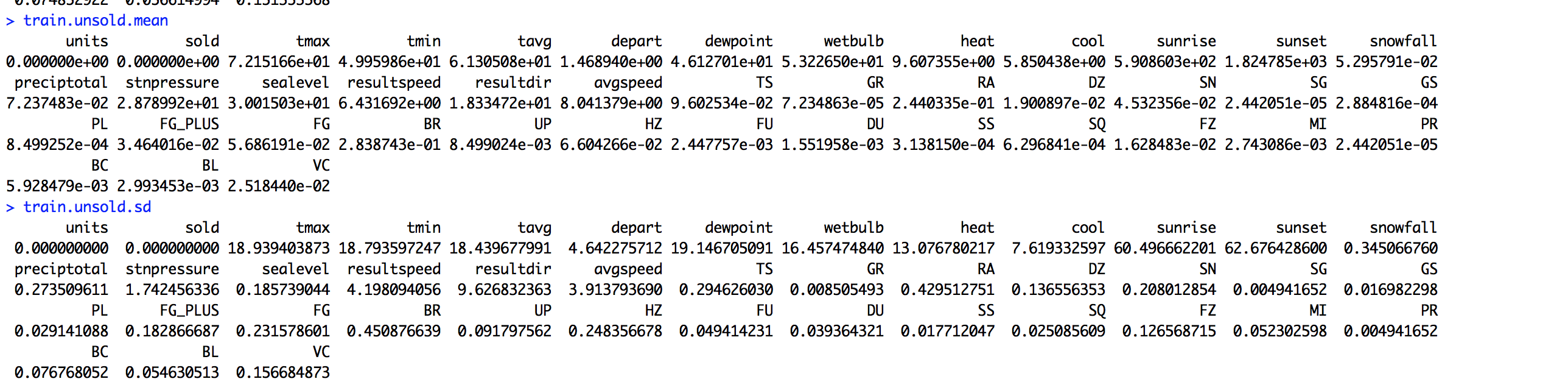
There are

* **20 weather stations** 
  + that are associated to **2.25 stores** on average (standard dev. = 1.51)
* **45 stores**
* **111 items**
* **4995 different combinations** of < store\_nbr, item\_nbr> (45 \* 111 = 4995), whose
  + **255 (5.1%)** combinations of < store\_nbr, item\_nbr> has at least one unit > 0 (=sold) one day in the training set
    - where the average units sold are 19.35 (standard dev.= 33.02)
    - corresponding to 236038 observations, i.e. 5.1% of total training observations
  + **4740 (94.9%)** combinations of < store\_nbr, item\_nbr> has units sold = 0 (=unsold) every day in the training set
    - corresponding to 4381562 observations, i.e. 94.9% of total training observations

# Predictors’ distribution in stations/products sold (5.1%)

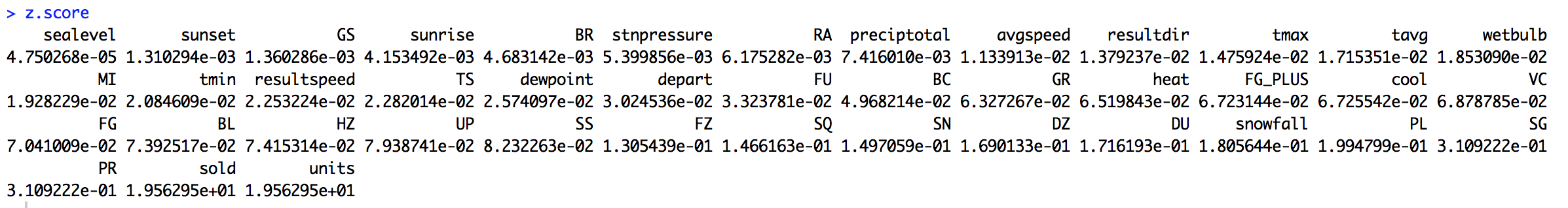


# Predictors’ distribution in stations/products unsold (94.9%)



# Z-score

Ordering by the difference of the predictors’ means in the two sets and dividing for the standard deviation, we have the following situation



So, the most promising predictors seem

* PR (z-score 0.31)
* SG (z-score 0.31)
* PL (z-score 0.19)
* Snowfall (z-score 0.18)
* DU (z-score 0.17)
* SN (z-score 0.15)

On the other hand, **z-score is** **always minor than 1**.

# Best selling combinations of stores / products

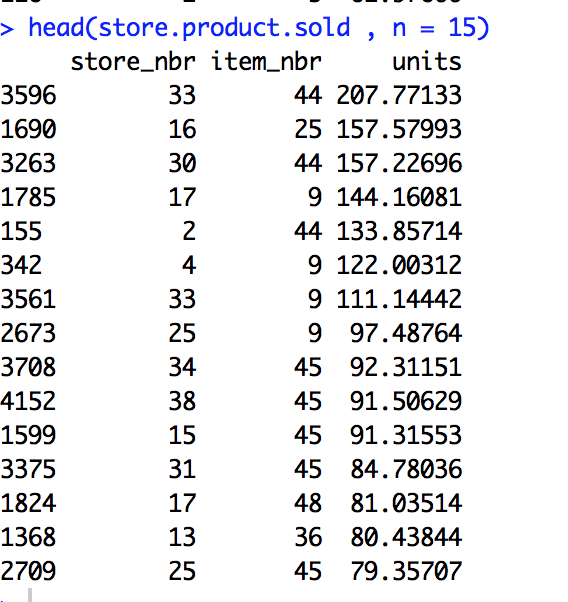


Figure 1 units is the average of sold units

# Most sold products

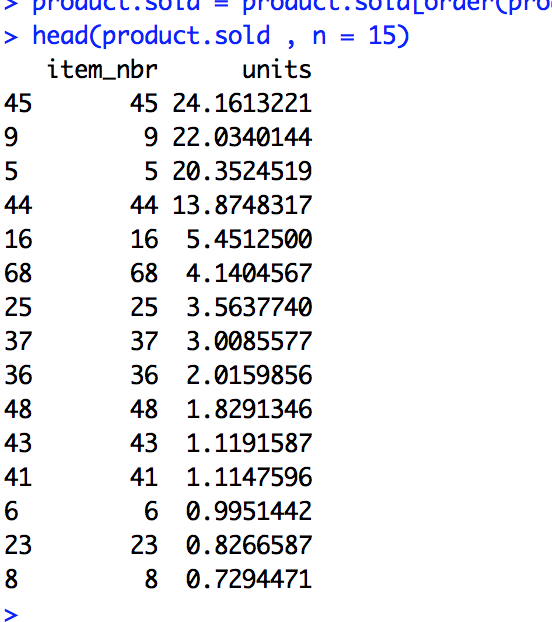


Figure 2 units is the average of sold units

# Best selling stores

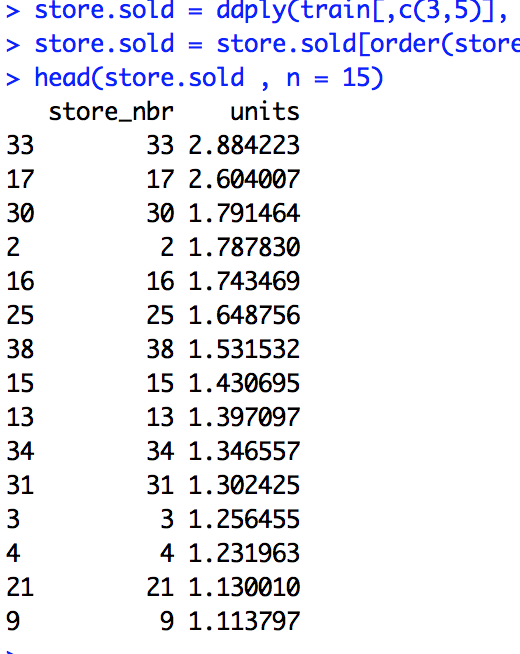


Figure 3 units is the average of sold units

# Correlation among sold products (cross selling?)

This analysis can be done at three levels, i.e. for each observed day in the training set

* among the products sold in each store (45 correlation matrices)
* among the products sold in the stores associates to the same weather station (20 correlation matrices)
* among the products sold in all stores (1 correlation matrix)

# Correlation among products sold in each store

Let’s focus only in a given store and let’s consider the correlation among products sold in such a store in the same day.

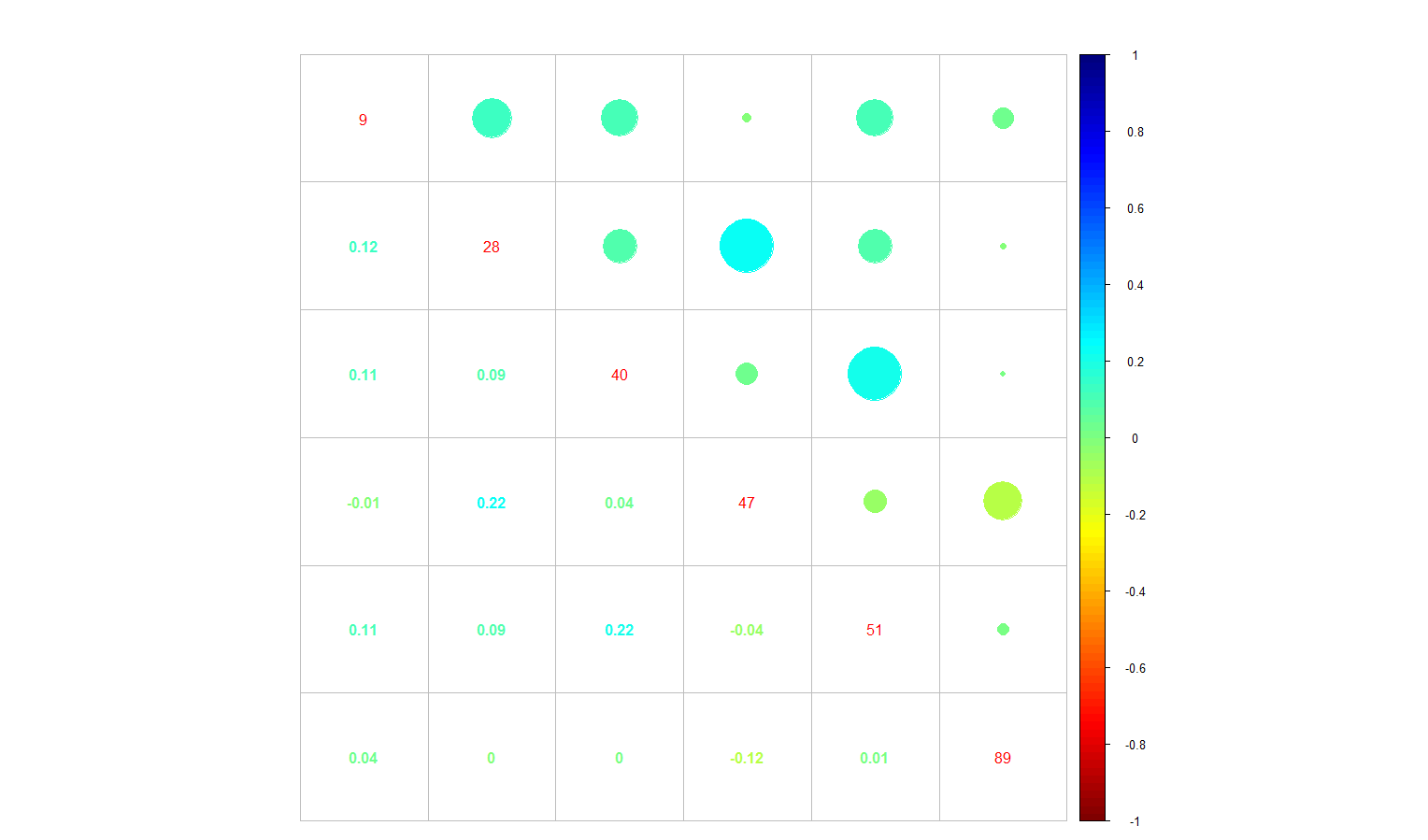


Figure 4 Correlation plot of sold products sold in store N.1 (unsold products are discarded)

For example, as shown by the previous correlation plot, in the store N.1 we can observe a direct correlation (0.22) among products (40,51) and (47,28) while an inverse correlation (-0.12) between products (47,89).

Let’s focus now on the stores associated to the same weather station, i.e. assuming that selling happened with the same weather. For example, let’s consider the **stores 14 and 45 associated to the same weather station N. 16**.

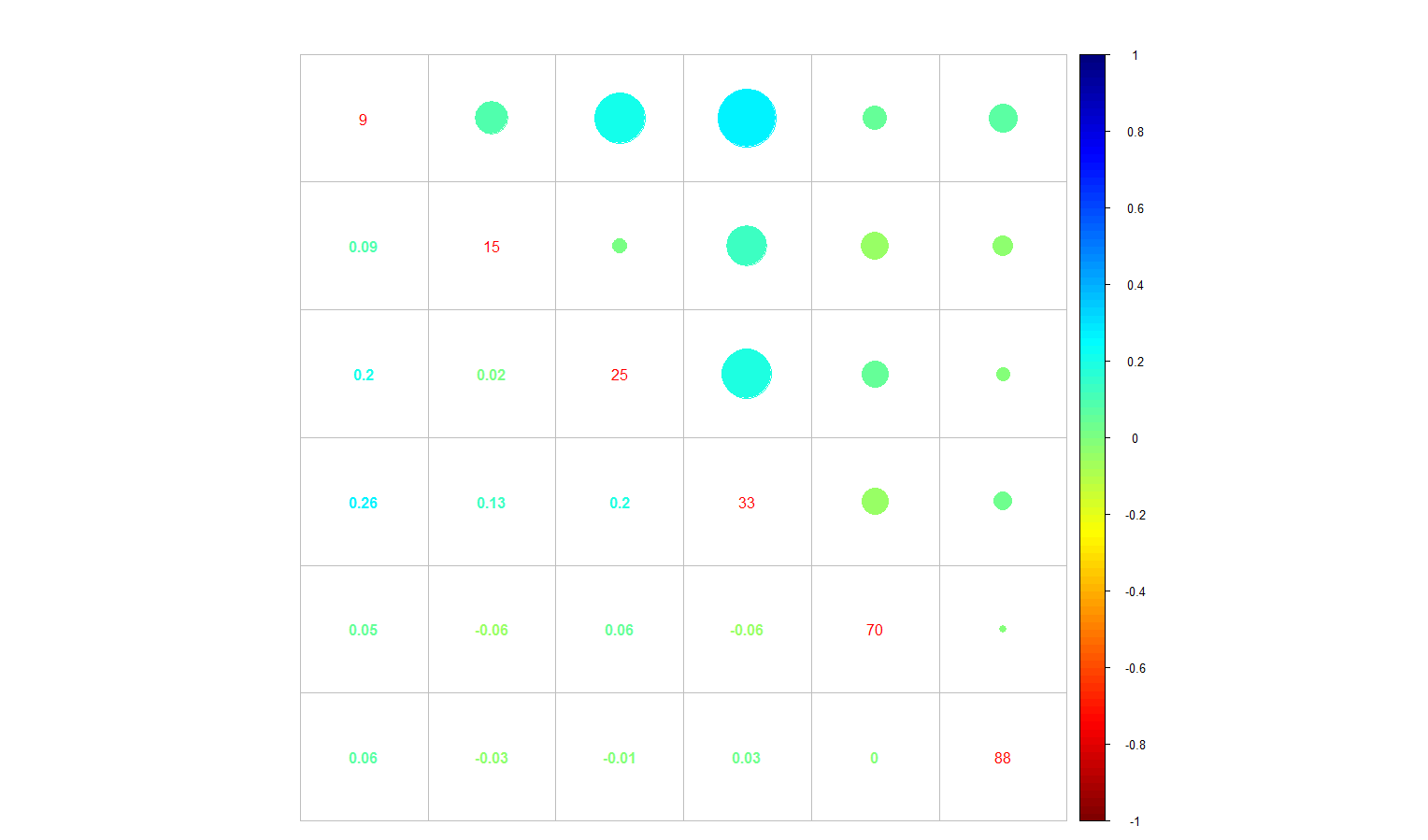


Figure 5 Correlation plot of sold products (unsold products are discarded) sold in store N.14 (station N. 16)

In the **store N. 14**, we can observe

* a **direct correlation (0.26)** between **products (33,9)**,
  + p-value: 4.441e-16
  + 95 percent confidence interval: 0.1998593 0.3183700
  + 99 percent confidence interval: 0.1805916 0.3362195
* a **direct correlation (0.2)** between **products (25,9)**,
  + p-value: 2.75e-10
  + 95 percent confidence interval: 0.1409988 0.2628716
  + 99 percent confidence interval: 0.1213516 0.2813823
* an inverse poor correlation (-0.06) between products (9,88)
  + **p-value: 0.05624**
  + **95 percent confidence interval: -0.001646013 0.124943908**
  + **99 percent confidence interval: -0.02163447 0.14457201**

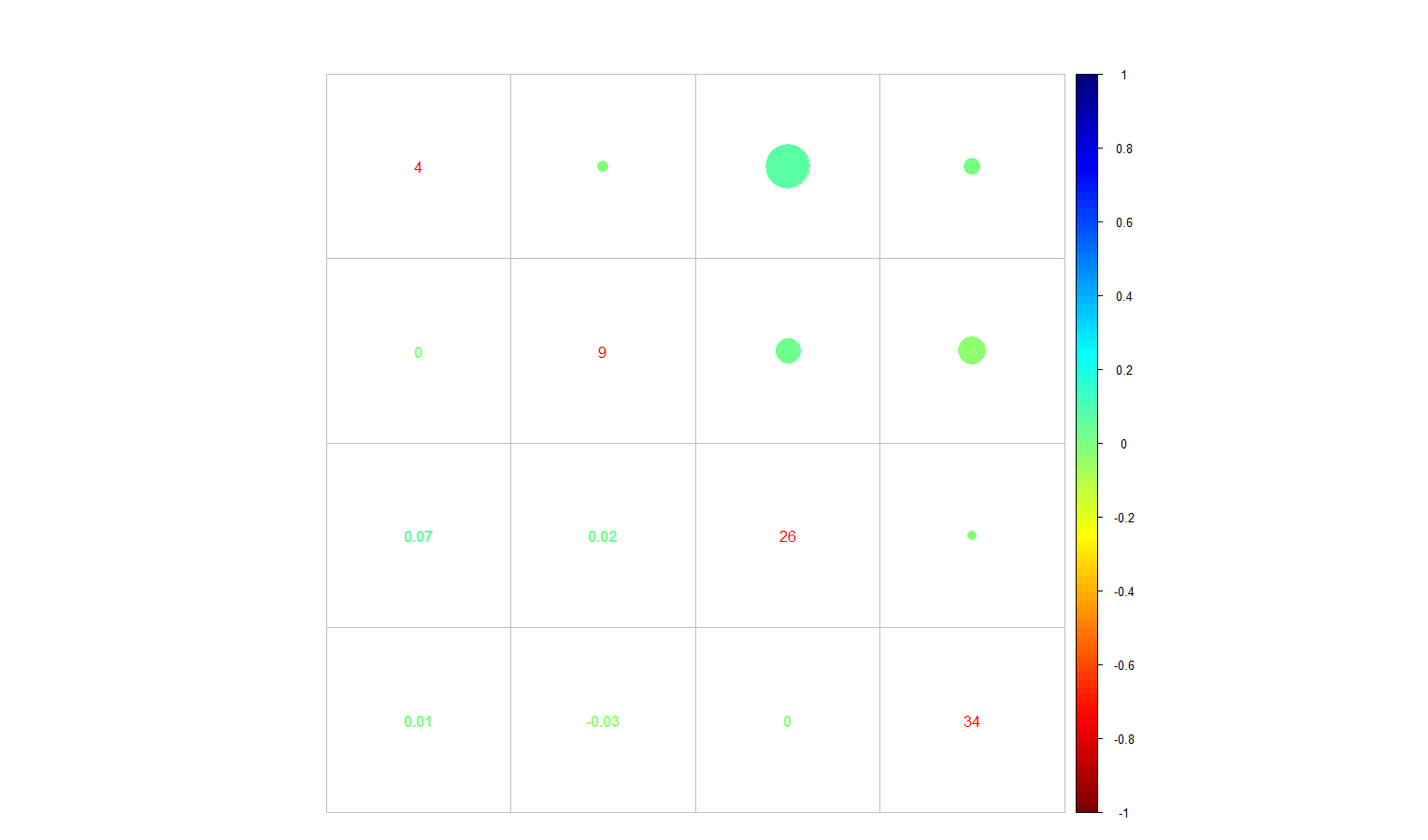


Figure 6 Correlation plot of sold products (unsold products are discarded) sold in store N.45 (station N. 16)

In the **store N. 45**, we can observe

* a direct **correlation (0.07)** between products **(26,4)**
  + p-value: 0.02467
  + 95 percent confidence interval: 0.00931338 0.13571730
  + **99 percent confidence interval: -0.01067778 0.15528520**
* an inverse poor correlation (-0.03) between products (34,4)
  + **p-value: 0.7548**
  + **95 percent confidence interval: -0.05343611 0.07362572**
  + **99 percent confidence interval: -0.07334693 0.09347732**

The question is why in store probably in the same area (i.e. associated to the same weather station) we have profiles so different? First of all, except for the product N.9, such stores has different sold products, i.e.

* Store N. 45: 9, 4, 26, 34
* Store N. 16: 9, 15, 25, 33, 70, 88

Moreover, the product correlation is different. We can make several hypotheses, e.g.

* different customers and/or different customer’s needs
* products out of stock in a store but not in the other
* etc.

In any case, we can conclude that **product correlation among the products sold in the stores associates to the same weather station is not very meaningful, in general**. For instance, referencing to the previous case, calculating the correlation between item N. 4 (sold in the store N. 45) and item N. 15 (sold in the store N. 16) is not very meaningful, as the previous item hasn’t been sold in the second store, and vice-versa.

# Correlation among products in the stores associates to the same weather station

Referencing to the previous example (i.e. stores 14 and 45 associated the weather station 16), different considerations can be done for the **item N.9 sold in both stores**. On the other hand, we observe a poor **direct correlation of 0.05579844**

* **p-value: 0.0853**
* **95 percent confidence interval: -0.007766513 0.118914267**
* **99 percent confidence interval: -0.02775138 0.13857405**

## Repeating same analysis for all store clusters

TODO

# Correlation among products sold in all stores

Referencing to the previous example (i.e. stores 14 and 45 associated the weather station 16), as the correlation observed for the only item sold in both stores is not statistic significant, it seems not very meaningful to consider sold units of item N. 9 in stores associated to other weather stations.

Weather imputation models

Discarded input variables

* station\_nbr
* date

All predictors are assumed as numeric (no factors)

**Models & Performance**

* Performed **basic** imputation with **BlackGuido[[1]](#footnote-1)** on Mode/Average/LineraReg and observed mean imputing performance (RMSE) **17.9**
* Performed **full** imputation with **BlackGuido** on Mode/Average/LineraReg/KNN\_Reg/PLS\_Reg/Ridge\_Reg/SVM\_Reg/Cubist\_Reg and observed mean imputing performance (RMSE) **17.8**

Predictive model #1 – basic

* For each date <d> and for each item <i> sold/predicted to be sold units in the store <s>, the related train/test set has been built with (imputed) weather data of the station <st> associated to the store <s> in the key.csv file, and the related output variable is the sold units for <d,s,i>
* Feature selection
  + Removing predictors that make ill-conditioned square matrix
  + Removing near zero variation predictors
  + Removing high correlated predictors
* Feature scaling
* Resampling: bootstrap + k-folds
* Models:
  + Average
  + Mode
  + LinearReg
  + RobustLinearReg
  + PLS\_Reg
  + Ridge\_Reg
  + Enet\_Reg
  + KNN\_Reg
  + BaggedTree\_Reg
* Leaderboard performance on dataset filled with basic imputation: **0.15193**

# Example (store n.1 / product n.9 / weather station n.1)

The output variable (units sold) has 929 observations with mean 29.48 and standard deviation 22.07. There are only 10 observations out of 929 (1%) with 0 units sold.

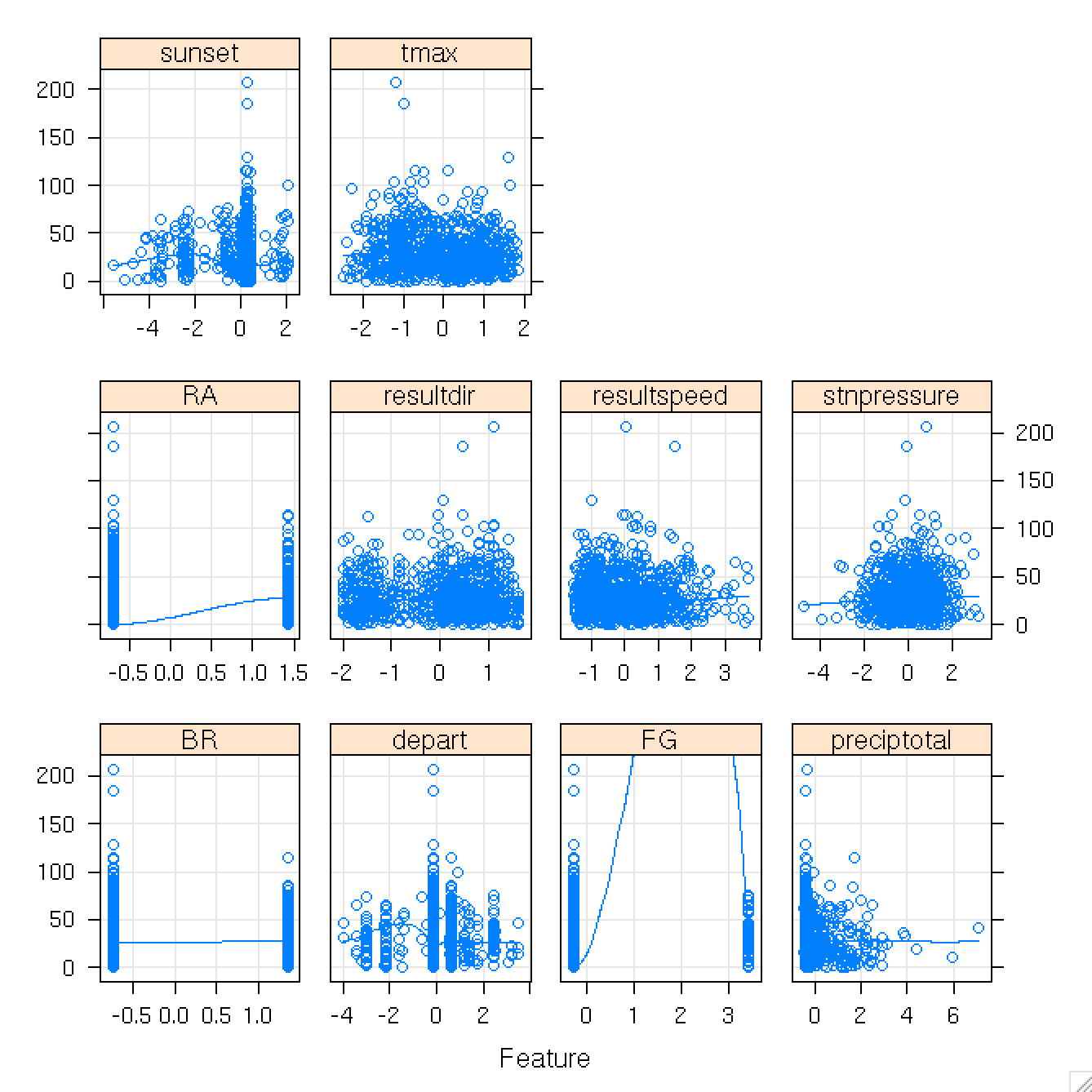
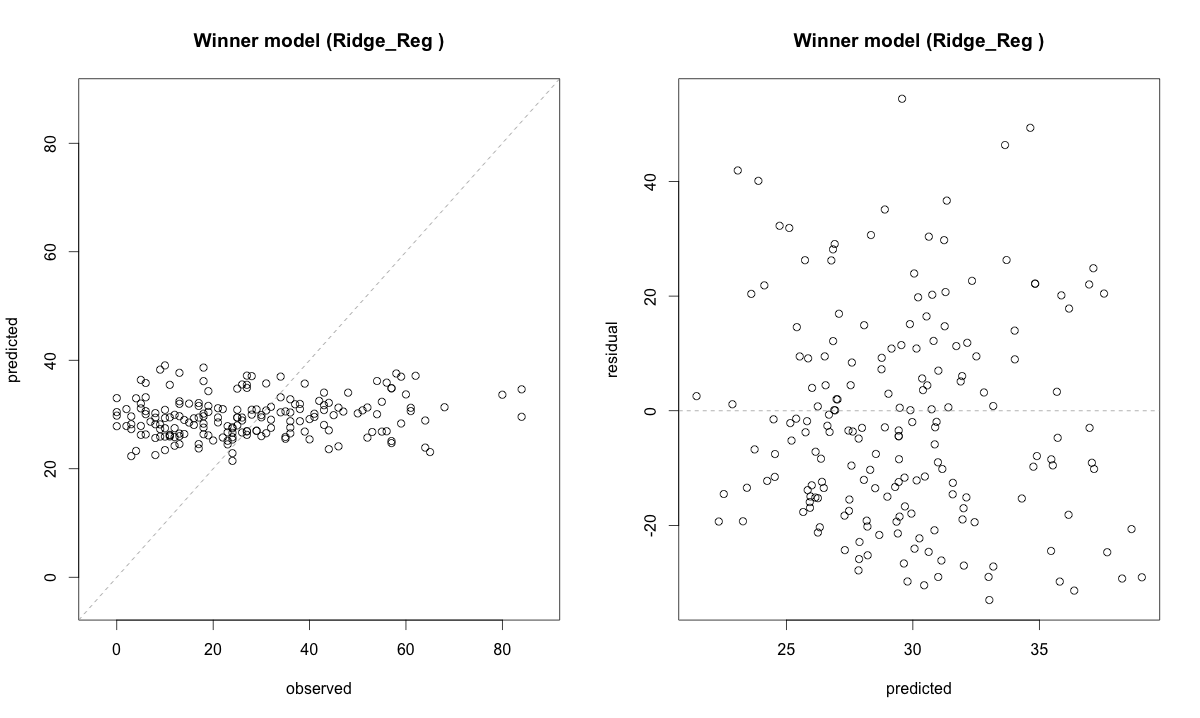


Figure 7 Scatter plots of predictors for store n.1 / product n.9 / weather station n.1 versus units sold

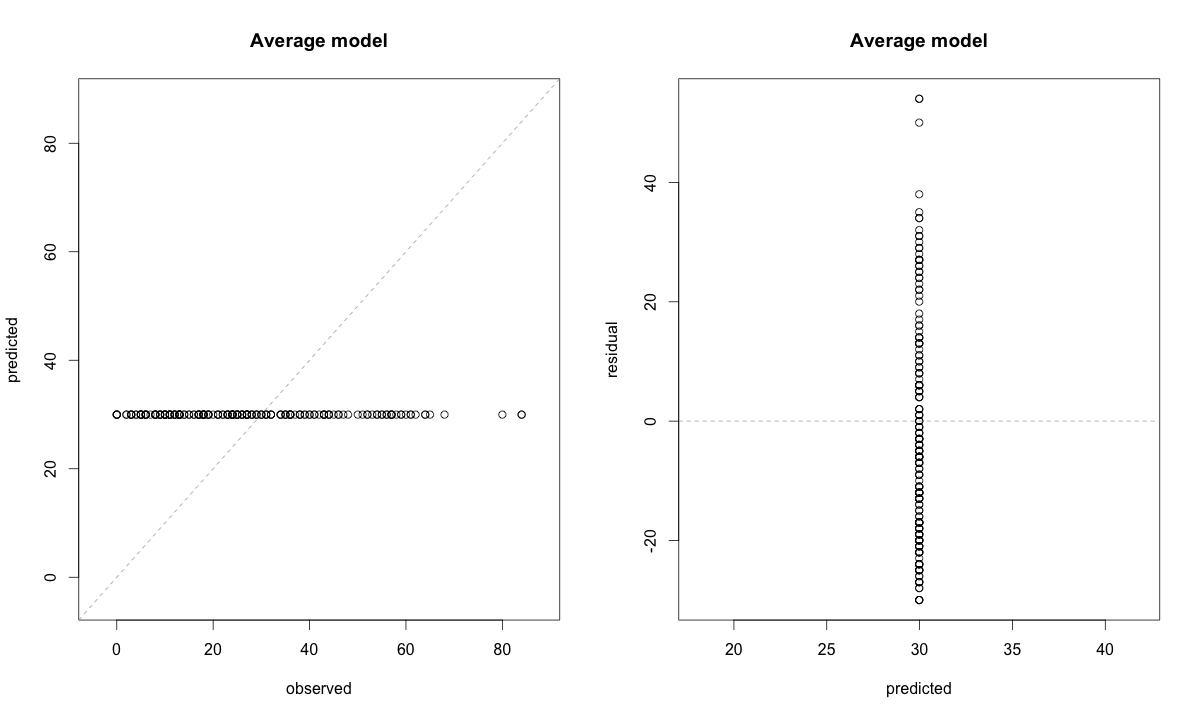
In the scatter plots, features are discarded according to the described feature selection process and scaled.

**Why does it look so bad?**

Previous scatter plots show a bad correlation between input and output variables. Moreover, looking at the console, the performances of winner models are not so better of the performance of the Average model. How is it possible? When the Average model is so good means that its residuals are very close to white noise. For example, in the previous case the **winner model is Ridge\_Reg** and in an ensemble its **RMSE is 18.45031** while **R2 is 0.024** (a very small piece of variability of output variable has been explained by predictors). Simple **correlation between** predicted and observed is **0.156** while **rank** **correlation 0.154**.



On the other hand, the **Average model has RMSE is 18.70546**.



Residuals of the Average model is not exactly white noise but its performance is pretty similar to the performance of the winner model and the predicted vs. observed plot shows undesired similarities to winner model’s plot. On the other hand, the residuals plot of winner model does not show any systematic pattern in the model predictions.

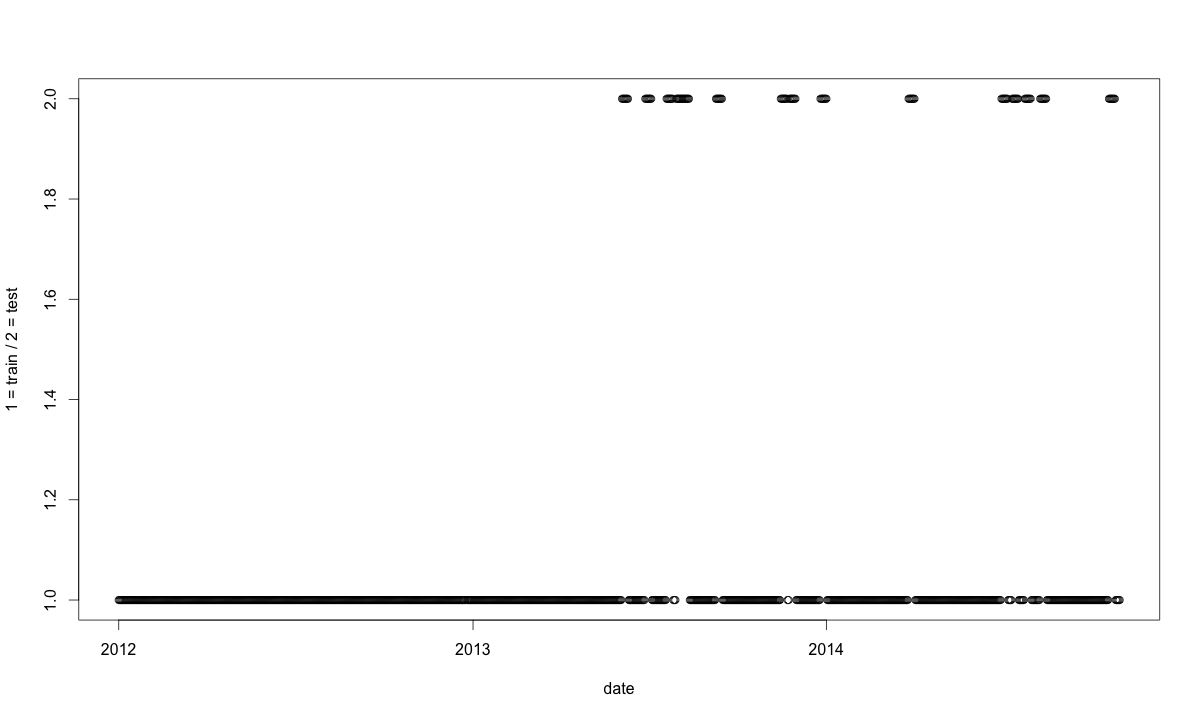
Why such performances? The simplest explanation is that such predictors alone cannot explain sold units variability. In fact, **the problem has been modeled so that the number of sold units is a strict function of weather condition indicators and this is a clear limit of the model**. **For example, if one day it’s raining and a customer bought a rain jacket, the same jacket can be used for the following raining days for a while**. **So, there is, at least, one important missing information in training data, i.e. potential/actual customers for each pair store/product**. Possible proxies to estimate it are

* Potential customer of the store estimated as the different customer served in a given period (very raw estimation);
* Potential customer of each pair store/product estimated as the different customers who bought such product in a given period (estimation requiring some CRM capabilities);
* Potential customer of each pair store/product estimated as the different customers who bought such product and the products highly correlated[[2]](#footnote-2) in a given period (estimation requiring some kind of basket analysis).

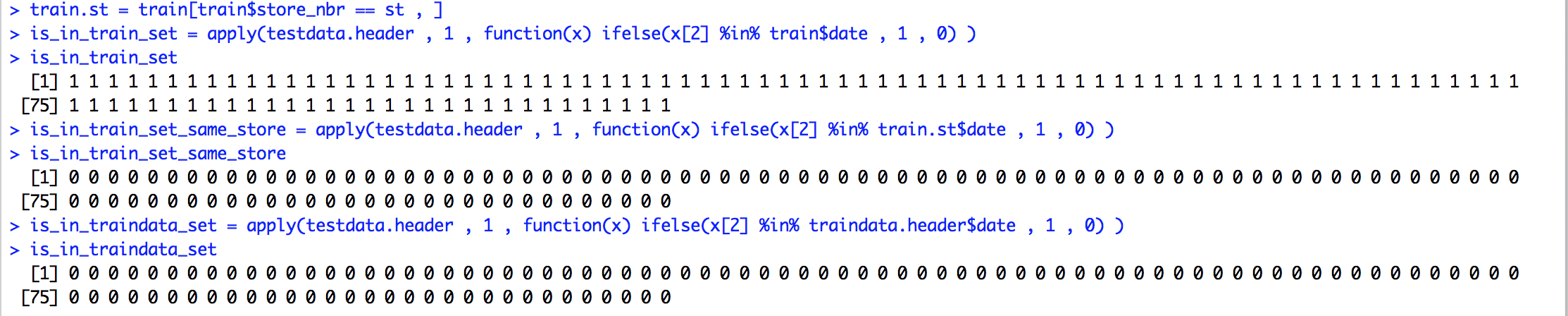
We can try to estimate this information inductively, but we have study better time periods in training and test data before.

**Training dates vs. test dates**

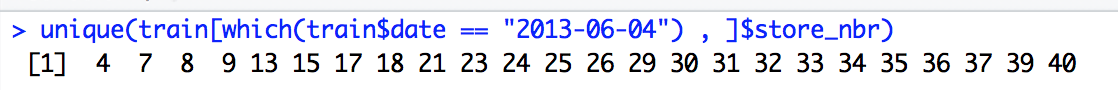
How are distributed training dates vs. test dates? For example, referencing at the previous example (store 1, product 9), the following plot shows it.



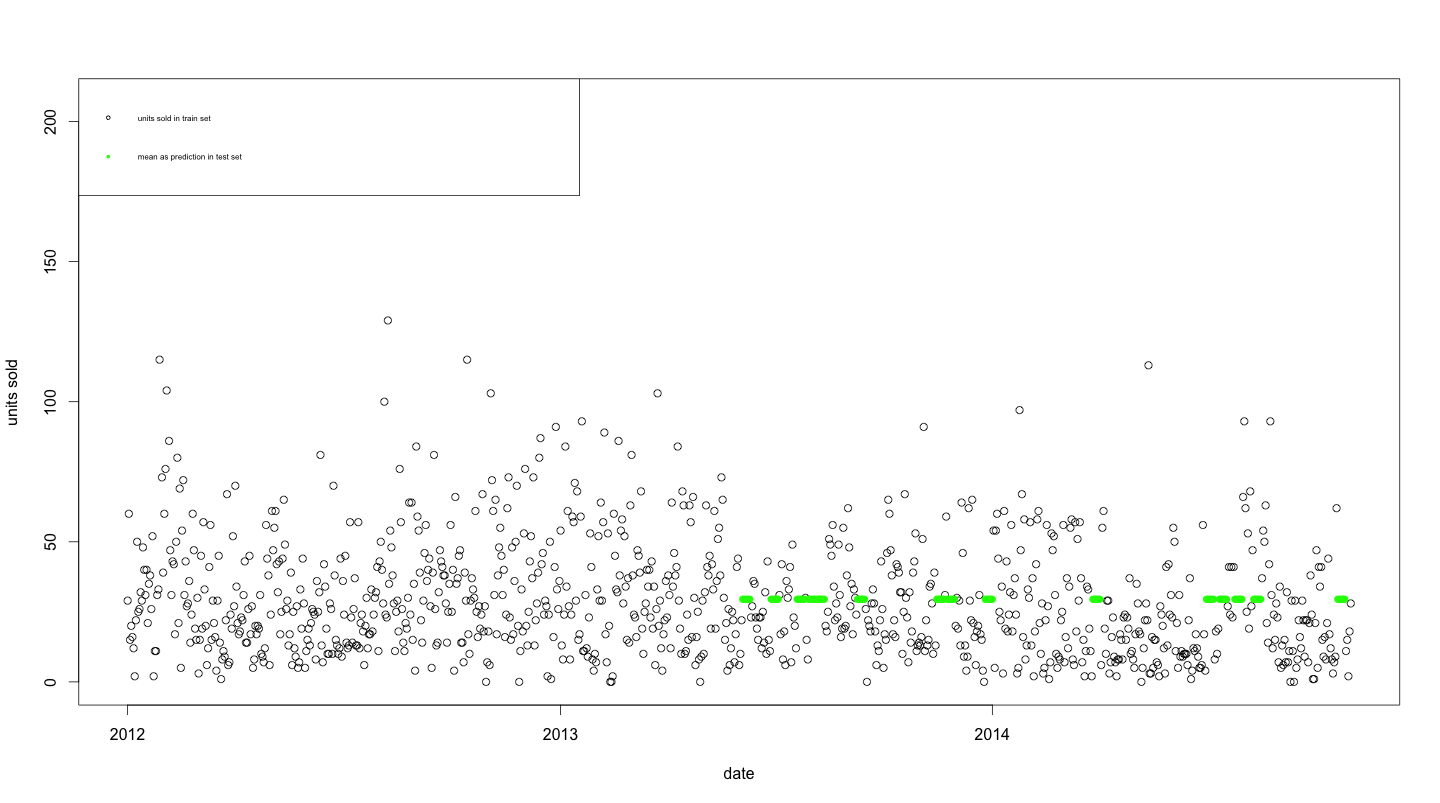
Moreover, for that pair store/product we find that test dates occur in train dates of some else store but don’t occur in train dates of the same store.



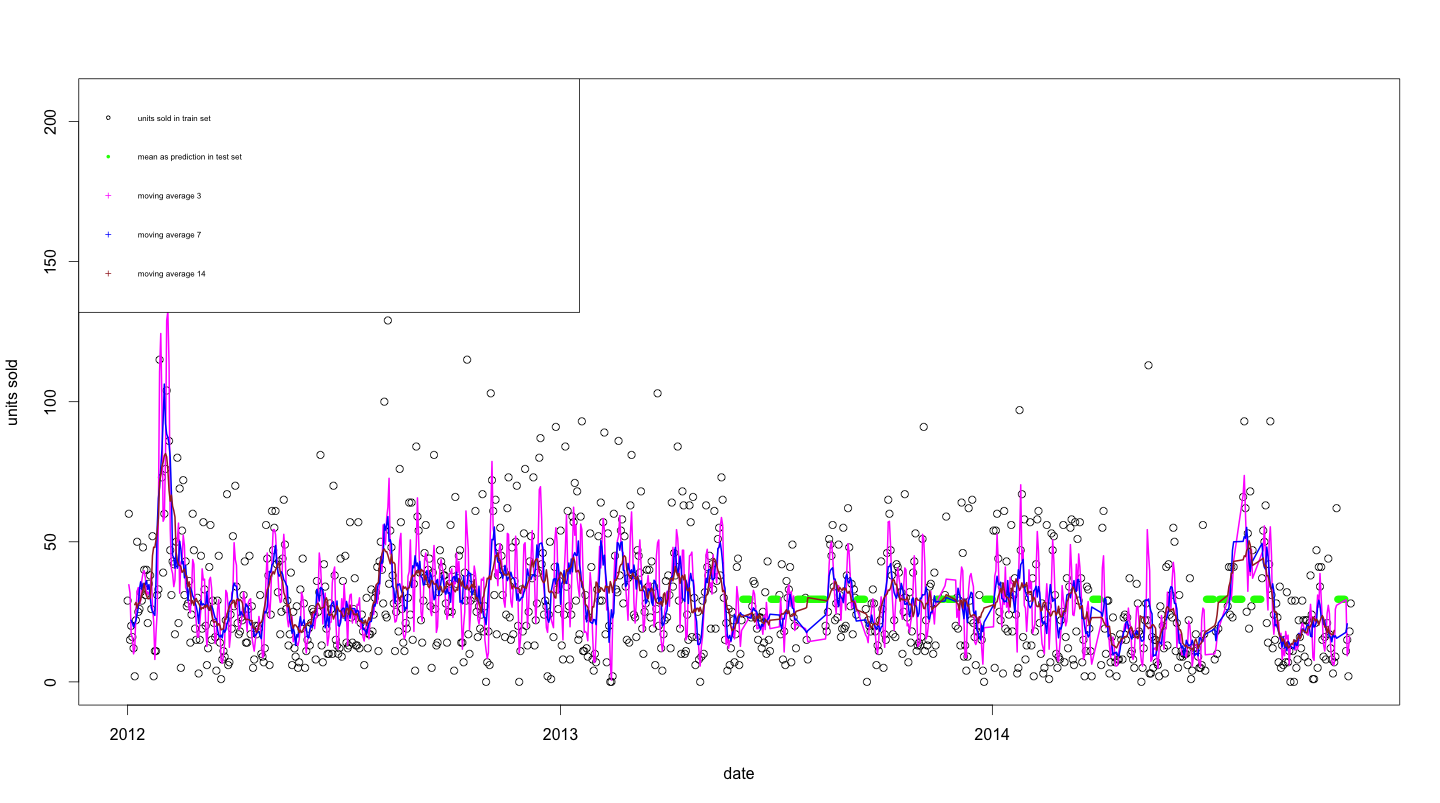
This fact can be used in case there’s a good correlation between product 9 and the pairs store/product were same test data occur. For instance, the first test data ("2013-06-04") occur in the following stores:



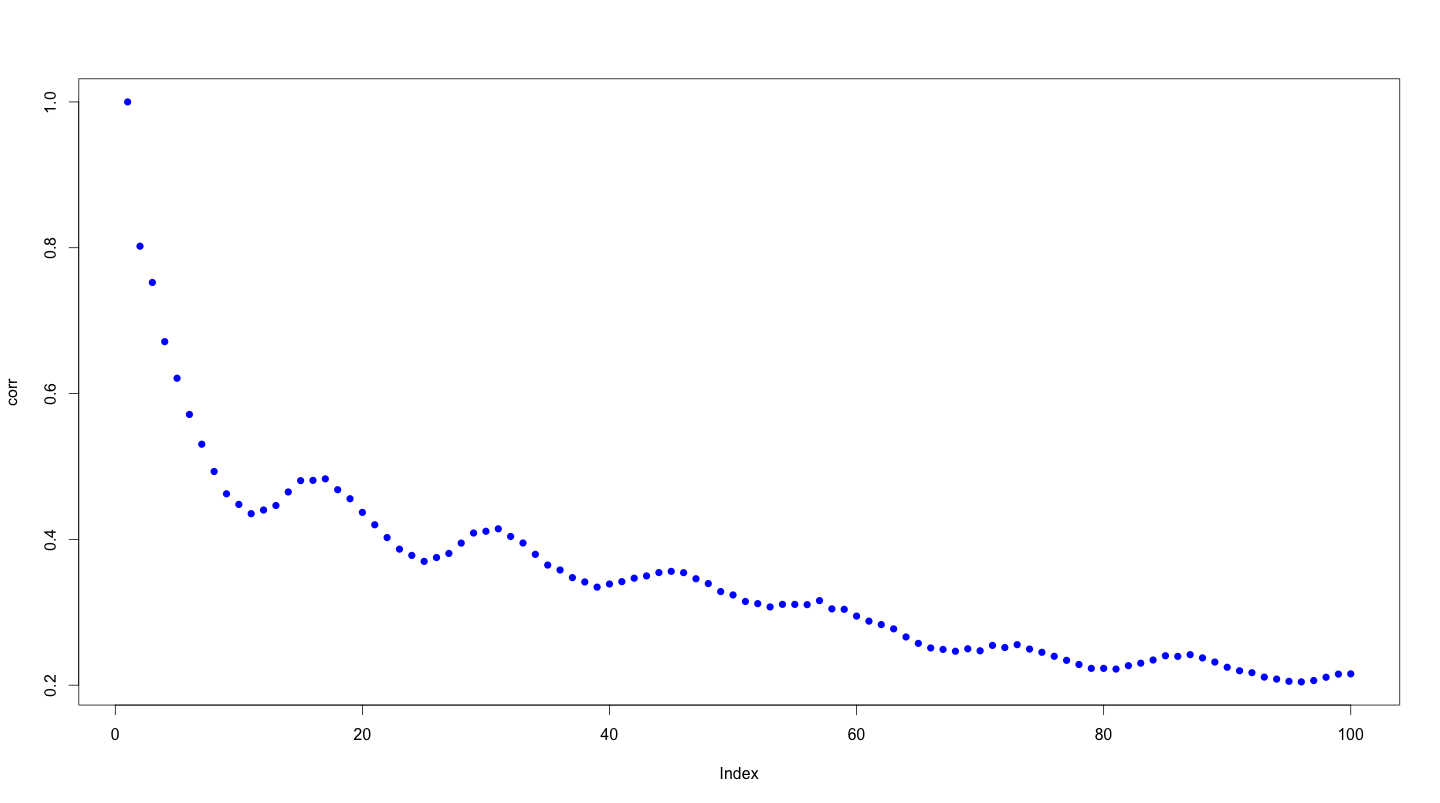
Coming back to train/test dates comparison, the following plot shows sold units in training set (black points) and its mean as prediction in the test set (green points).



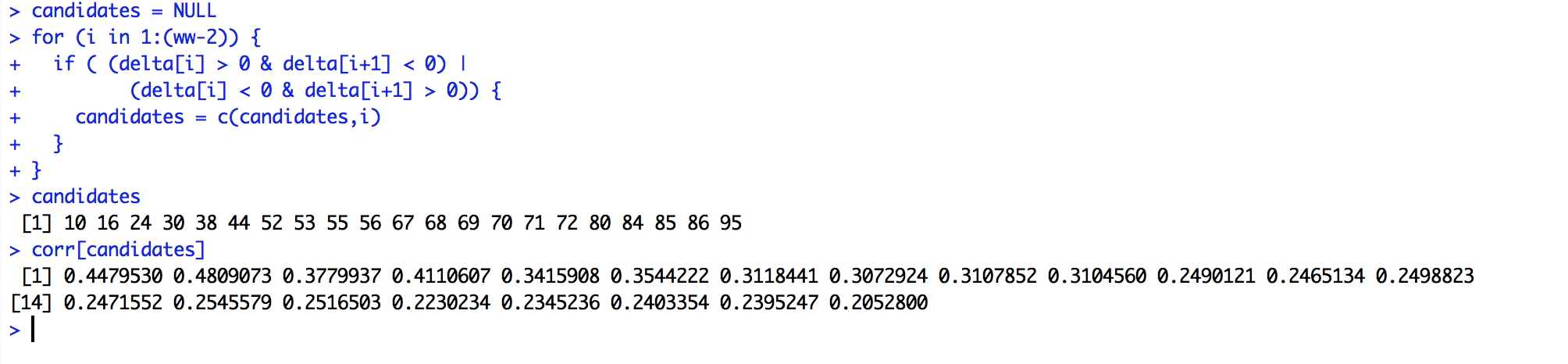
So, if this pattern holds also for other store/product pairs, there is plenty of training points around test points. Hence, in order to estimate potential/actual customer of such pair store/product, we can use for **moving average**.



We can estimate the correlation between moving average x and sold units in training data. The following plot shows such a correlation versus time window.



Local minima can be calculated approximately in the following way.



So, moving average 10 is the first min (correlation = 0.44) and moving average 16 is the first local max (correlation = 0.48).

So, at a given date t, let be

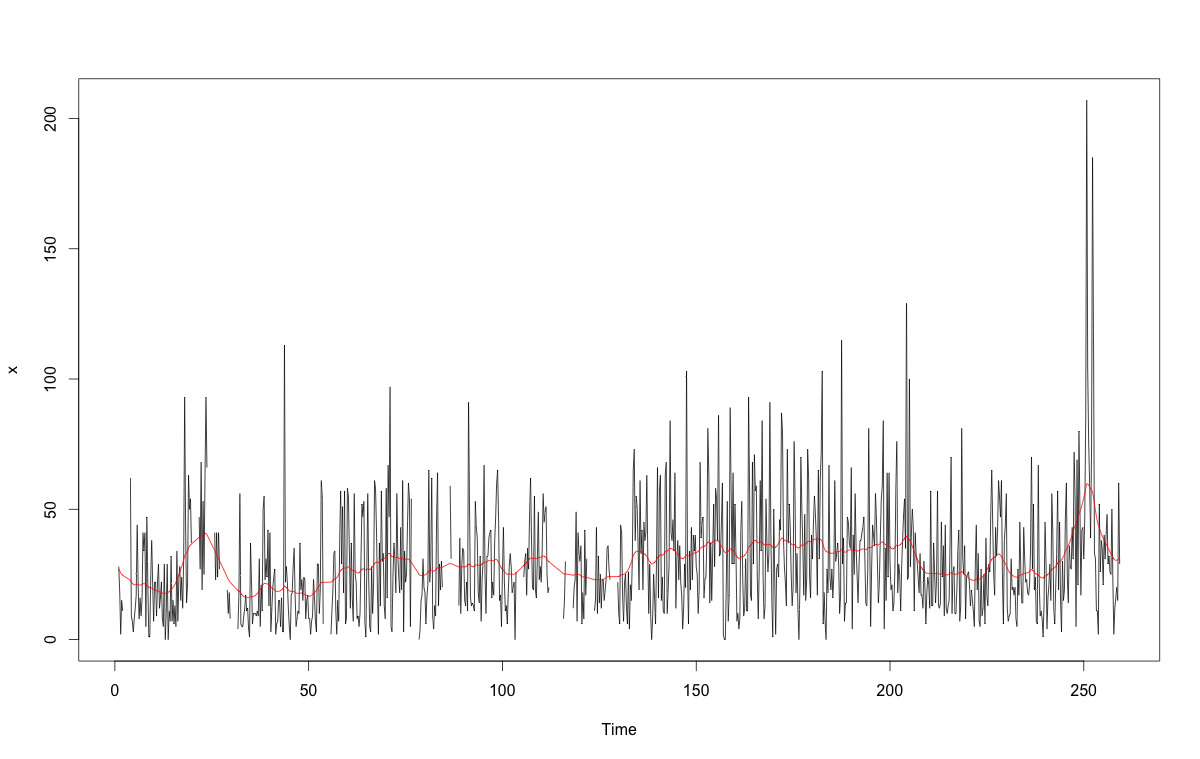
* the probability that a customer that does not have the product i can buy it
* C(t) the number of customers that does not have the product buy it;
* the units of product i sold at the time t;
  + notice that
  + notice that
* the number of potential customer that can buy it
* the mean time product life cycle
* the number of new potential customers or recurrent customers entering at the time t (net old customers)
  + notice that - *stationary hypothesis*

Hence,

Some observations:

* depends, besides , on and that we don’t know
* though we know , and could be undefined

Predictive model #2 – interpolation



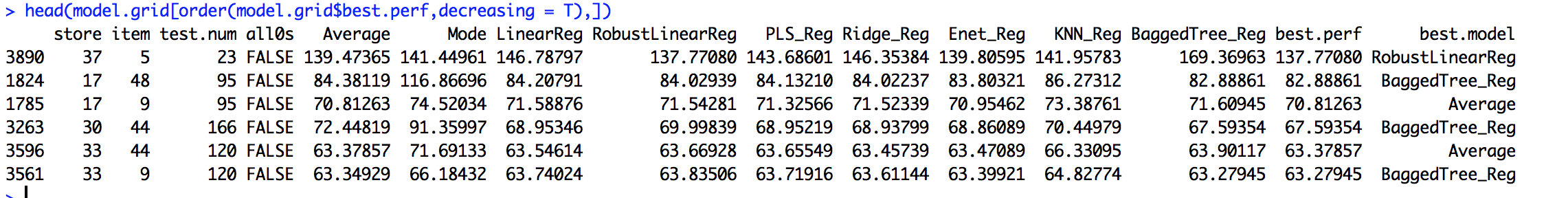
* Leaderboard performance: **0.10198**

Performance analysis

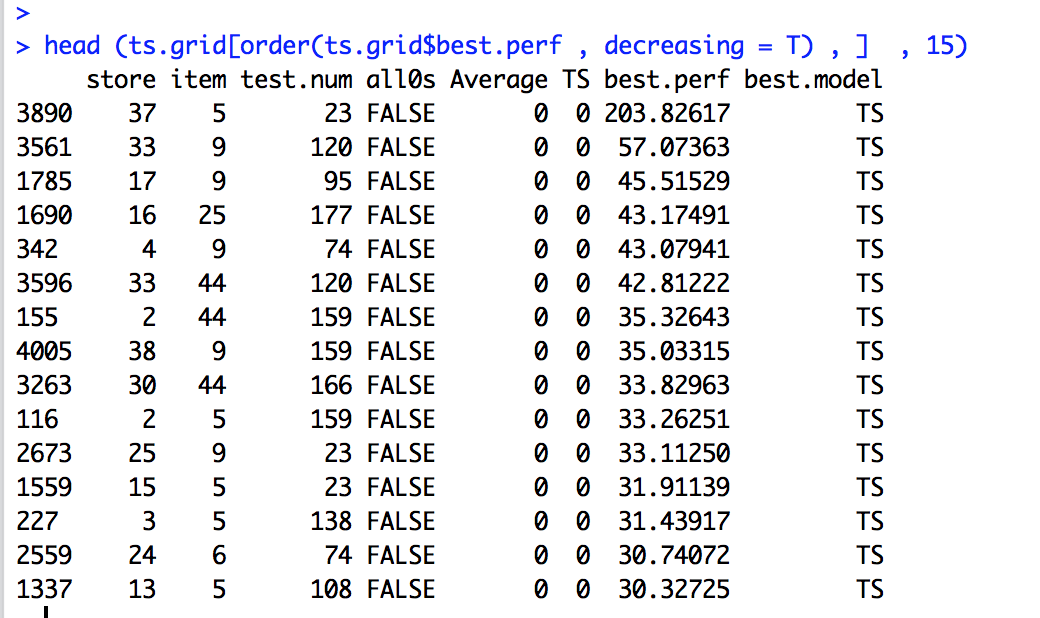
Let’s do a bit of performance analysis on above model.

**What are store/item combinations with highest RMSE?**

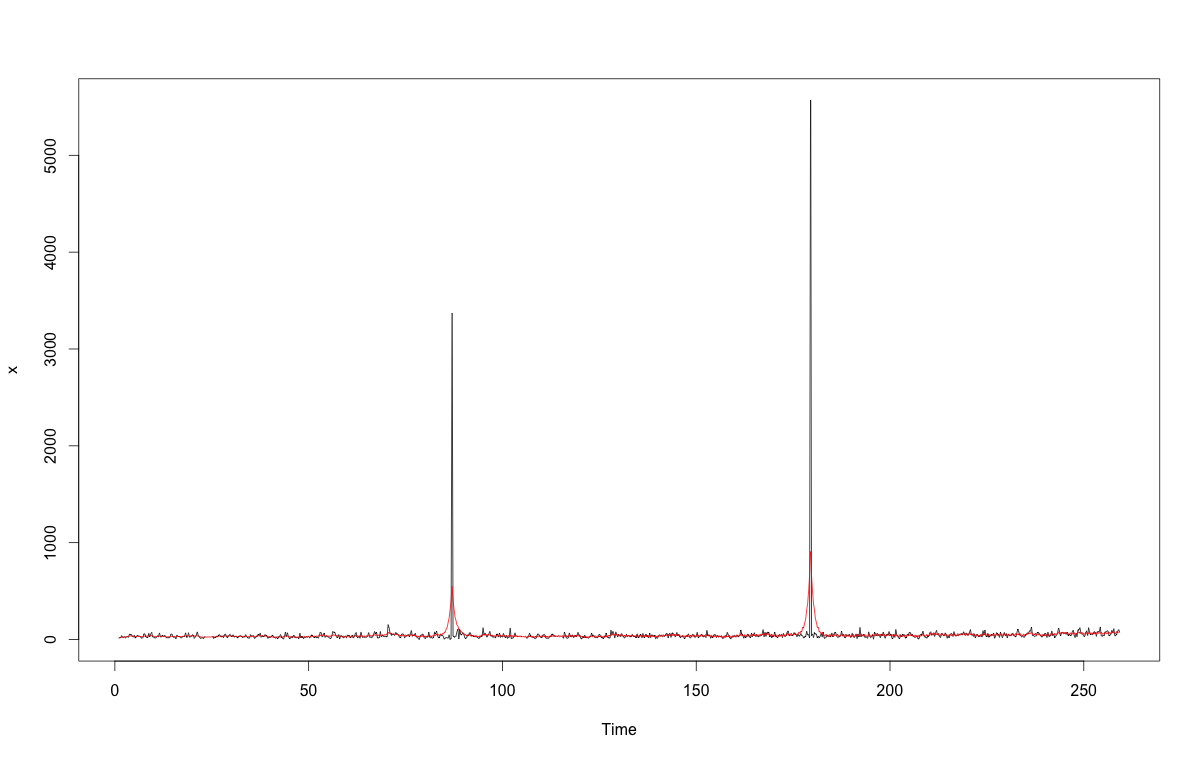
**Model # 1**



**Model # 2**

****

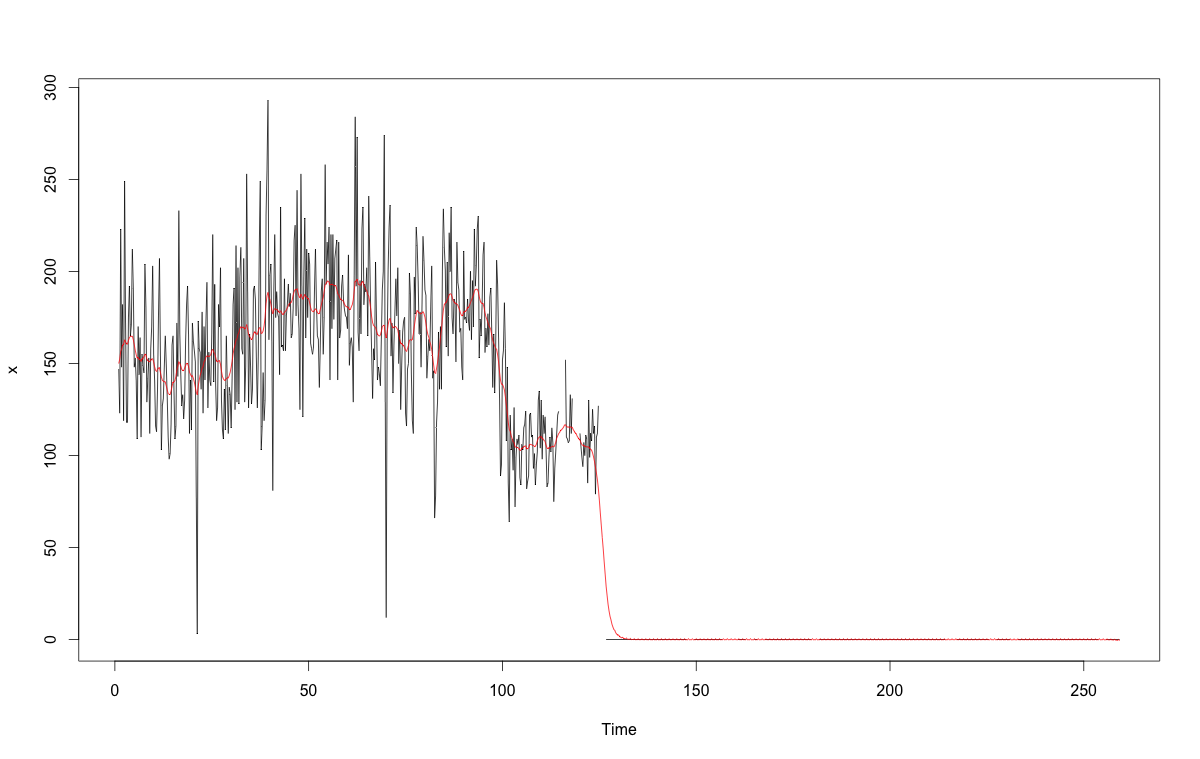
**Store n. 37 / item n. 5**



* Estimated RMSE of model # 1 is 137.77
* Estimated RMSE of model # 2 is 211.732

The 2 peeks are not related to weather conditions but they are 2 dates occurring exactly **1 week before thanksgiving days** of such years. Probably, Wal-Mart made some kind of promotions, so as 2014 test set last date is 2014-08-02, the safest choice is to replace such peeks in train set (e.g. with the mean). It’s not a surprise that in model # 1 the winner is *Robust Linear regression*.

**Store n. 17 / item n. 48**



Here, before 2013-05-19 sold units were something like 100, while after 2013-05-19 sold units were 0. Probably such an **item has been put out of shelves**. So, here the safer choice is setting up a prediction of 0 for test dates after 2013-05-19 and excluding such data from training set.

# Few more cases

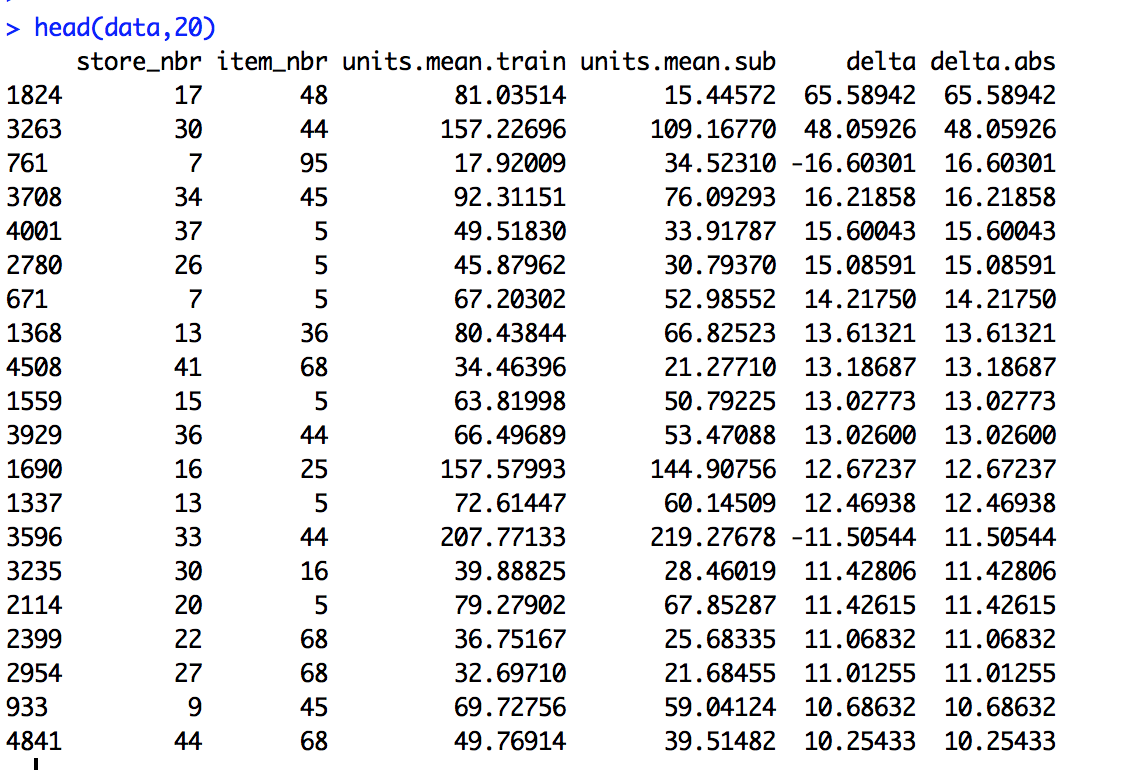
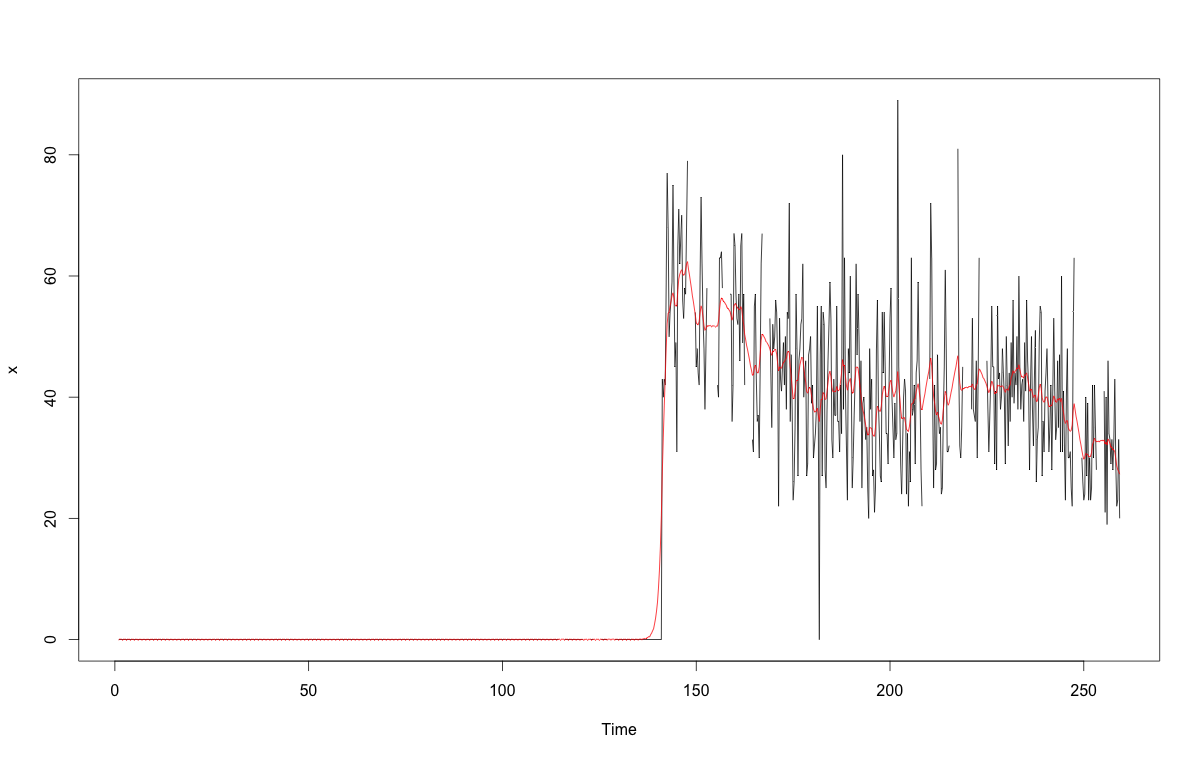


Figure 8 - Mean of submitted predictions vs. training data

We can note that in most cases the mean of submitted predictions is minor than the mean of training output variable. This comes from the fact the model, fitting a structural model for a time series by maximum likelihood, isn’t able to proper predict its oscillations.

**Store n. 7 / item n. 95**



Same dynamic as store n. 17 / item n. 48. Before 2013-07-15 sold units are 0.

All these adjustments give an **improvement of 0.00076 on the leaderboard**.

Predictive model #3 – model #1 improvement

Looking back at the equation

that implies

As a proxy of we use time series interpolation.

Performed as **0.10887**on leaderboard. **This would suggest that time series interpolation alone gets better performance (0.10198) than as feature together with other weather related features in model #1.**

Predictive model #4 – gradient boosting

**Gradient Boosting**

1. Compute the average response using time series interpolation, y, and use this as the initial predicted value for each sample
2. **Compute the** **residual**, the difference between the observed value and the current predicted value, for each sample
3. **Fit a regression model using the winner model of model #1 using the residuals as the response**
4. Predict each sample using the regression model in the previous step
5. Update the predicted value of each sample by adding the previous iteration’s predicted value to the predicted value generated in the previous step

Performed as xxx on leaderboard

Predictive model #5 – averaged interpolation

In this model we use the fact that typically in this problem the test set has been built as “holes” in the training set.

E.g.

Test set

Training set

So, for a given test set interval, the prediction is built as the mean of the k previous/following observations in the training set.

Performed as xxx on leaderboard

1. **BlackGuido** is part of the proprietary machine learning framework *fast-furious* [↑](#footnote-ref-1)
2. In this case bananas can be easily excluded from the analysis, but what about “bananas products” for this product category (e.g. item n. 9 in previous chapters)? [↑](#footnote-ref-2)