Housing Price Prediction and Fraud Detection

1 Introduction

Detecting housing fraud is an important component of monitoring housing markets. Detecting all housing fraud cases manually would be difficult, it's a great choice to make use of data modeling and prediction tools to help with detection.

This report constructs a model to predict the expected cost of houses in TWD at the time of scale with detailed methods and steps, tests the accuracy of the model and explains how different factors can influence housing price based on the model. Our analyses conclude that a generalized additive model with splines can fit the housing data well. The influences of transaction date, house age, distance to nearest MRT station, and longitude and latitude on the housing price are wiggly. The influence of number of stores within living circle on foot on the housing price is roughly linear. A integrated function that predicts house prices is provided, and examples and documentation are provided after the main body of this report.

2 Methods

There are two main goals for our analyses. First, to construct a model in the form of computer program to help our client predict the expected housing price using past housing data, with detailed description of how model is constructed; Second, to explain how different factors influence housing price to help our client understand the model we provided better.

To address these three goals, a generalized additive model (GAM) with splines was constructed to explore the association between the housing price and other factors and then predict the expected housing price. The house price of unit area was used as response, and transaction date, house age, distance to the nearest MRT station, number of convenience stores in the living circle on foot and geographic coordinate (latitude, longitude) were used as predictors.

Based on exploratory analyses, it was found that there were non-linear relationships between housing price and predictors. And the price varied differently over different region of these predictors. So, we introduced splines which were piece-wise polynomials fitting separate low-degree polynomials over different regions of predictors to GAM to make the prediction more smooth and more accurate.

We constructed the GAM in R using the mgcv package. The function gam() in this package would help us select optimal effective degrees of freedom (A summary statistic that reflects the degree of non-linearity for each predictor in GAM. The higher the effective degrees of freedom is, the more wiggly the curve is.) and smoothing parameter estimation automatically based on generalized cross validation.

The data set we used was clean and there was no missing values. All variables including transaction date, house age, distance to the nearest MRT station, number of convenience stores in the living circle on foot and geographic coordinate (latitude, longitude) made sense in the prediction of housing price. No variables were excluded. One observation with extremely high housing price was removed as outlier. Since the goal is to construct a model to predict housing prices which would be used to compare with the actual housing prices to judge whether the actual prices are reasonable, the data used in model construction should be as normal as possible and including outliers would impact the accuracy and stability of our model.

As a note, the transaction date was in numerical format in the data set and we kept the format in our model. There are two main reasons. Firstly, the order of transaction date mattered in our purpose. If we split the date into year and month and treated the year and month as categorical variables, we could not depict the intrinsic ordering of transaction date; Secondly, categorical variables do not accept values beyond its original levels. However, our data

were limited and could not capture all possible years and months. To satisfy the most important purpose that was predicting expected housing price better, we treated the transaction date as a numerical variable to accept new values beyond its original values in the date set.

We used three different metrics to measure the performance of our models. First was the correlation coefficient between predicted prices and the actual prices. If the actual prices were close to the predicted prices, the correlation coefficient should be close to 1. And closer the predicted prices were to the actual prices, closer the correlation coefficient was to 1. Second was the scatter plot of actual prices vs. predicted prices. If the actual prices were close to the predicted prices, the points in scatter plot should be close to a straight line with slope of 1 and intercept 0. Third was the box plot and histogram of difference between actual prices and predicted prices which reflected the distribution of the differences. If the actual prices were close to the predicted prices, the distribution of difference should concentrate on 0.

One complexity was about the interaction between longitude and latitude. Since longitude and latitude were used together to measure the geographic coordinate of a house, it might not work well and be less meaningful to treat longitude and latitude as independent variables and fit the model over them separately. To tackle this problem, we regarded longitude and latitude as a whole in GAM and used the spline over these two variables instead of including two splines (one over longitude and one over latitude).

There are two main limitations in our analyses.

One was the interpretation and inference on how different factors influenced housing price. The coefficients of GAM with splines were hard to interpret. So, we made use of graphics to show the partial effects of different factors on housing price. Although it was difficult to quantify the effects, the clear trends of change in housing price over different predictors were shown to help understand the association between the price and the predictors. Furthermore, since the P values in GAM with splines were invalid for inference, we were limited to make inference on whether a predictor influence the housing price significantly or not.

The other limitation was that not all data was used to fit the GAM. To examine the accuracy of our model on predicting housing prices for data that was not included in the data set we obtained, we split 80% the data set as training data and the rest 20% as test data. The test data set was used to mimic the data that was not included and to measure the performance of our model. Only the training data was used to fit the GAM. We used the trained GAM to predict housing price for samples in the test data set, and then compare the provided actual housing price in test data set with the predicted housing price.

3 Results

Each row of the data represents one housing transaction. The initial data set contained 414 observations and 6 variables. After removing one outlier with extremely high housing price, 413 observations were remained in total. 331 observations (80% of total) were used as training data set, and the rest 72 (20% of total) were used as test data set. Table 1 shows the summary on statistics for training data set. (Summary on statistics for test data set was omitted because we used test data to mimic the data not shown to us and no baseline information on those data should be included.)

Variable	Mean	Standard Derivation	Median (IOR)
Transaction Date	2013.158	0.277	2013.167 (2012.917, 2013.417)
House Age (in Years)	17.605	11.393	16.200 (8.950, 27.400)
Distance to the Nearest MRT Station (in Meters)	1065.339	1265.177	488.819 (289.325, 1448.504)
Number of Stores within Living Circle on Foot (Integer)	4.227	2.958	5 (1, 6.5)
Latitude (in Degree)	24.969	0.012	24.971 (24.963, 24.977)
Longitude (in Degree)	121.534	0.015	121.539 (121.530, 121.543)
House Price (in 10000 New Taiwan Dollar/Ping)	37.944	13.292	38.400 (27.700, 45.950)

Table 1: Distribution Metrics.

As a note, the transaction date can be translated into year and month by using the integer part of transaction date as year and multiplying the decimal part with 12 as month. For example, 2013.250 is Mar 2013 ($0.250 \times 12 = 3$) and 2013.500 is Jun 2013 ($0.500 \times 12 = 6$).

Figure 1 and 2 shows the partial effect of different factors on housing price given by the GAM. It provided the modeled expected cost of houses for each factor while holding all other variables constant.

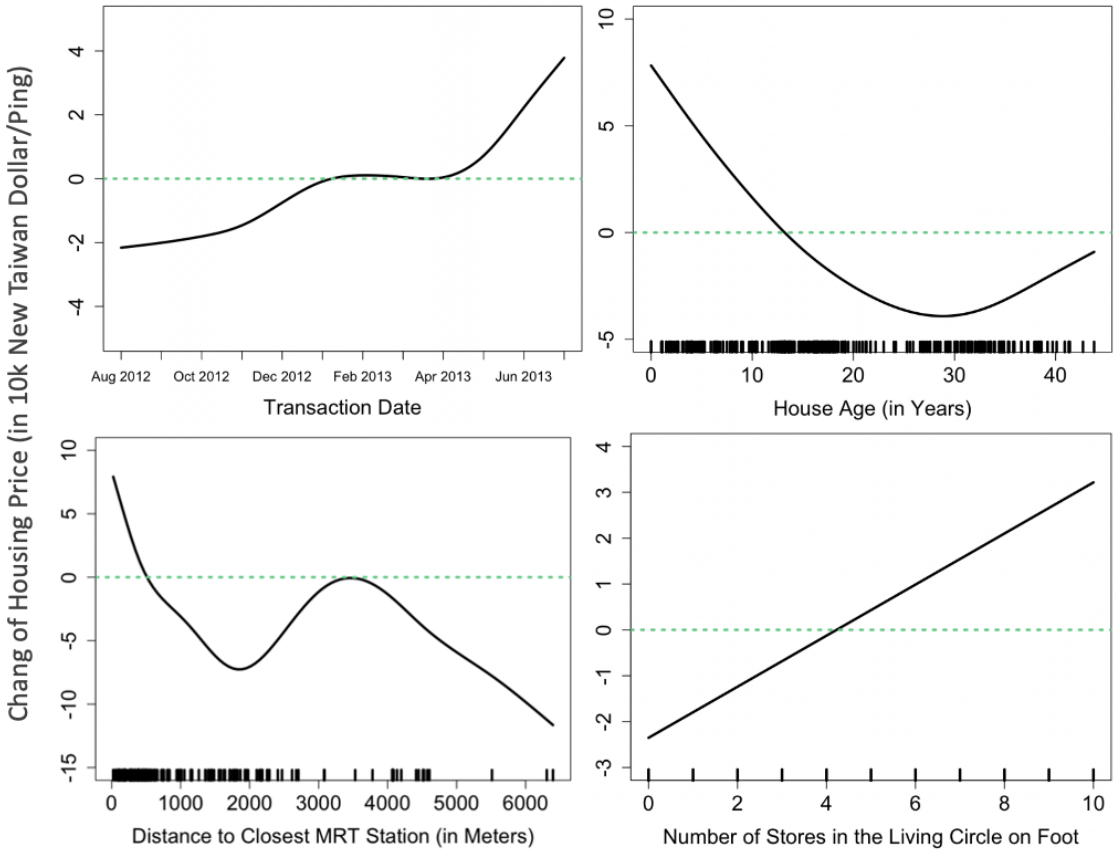


Figure 1: Partial Effects of Different Factors on Housing Price.

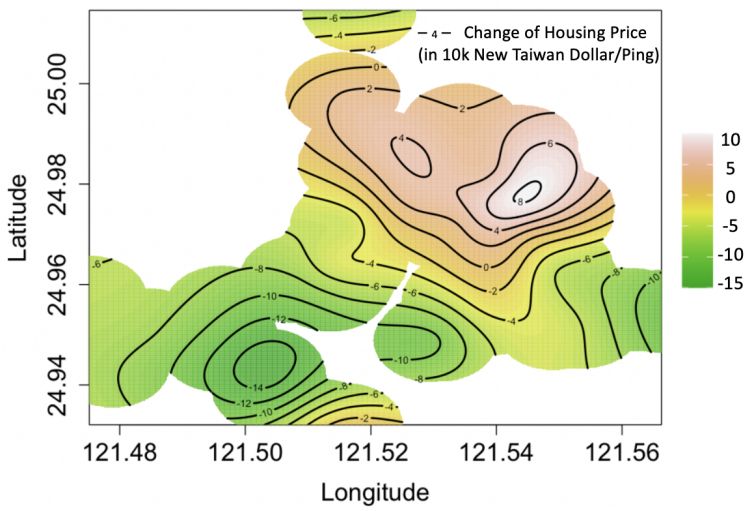


Figure 2: Partial Effects of Different Factors on Housing Price.

We could see that the influence of transaction date, house age, distance to closest MRT station and (longitude, latitude) were wiggly, and the influence of number of stores in the living circle on foot was roughly linear. In Figure 1, the y-axis stood for the change of housing price. In Figure 2, the contour stood for the change of housing price. Negative value meant housing prices decreased and positive value meant housing prices increased.

The plots for house age (top-left in Figure 1), number of stores in the living circle on foot (bottom-left in Figure 2) and (longitude, latitude) (Figure 2) were taken as examples to explain how to understand the partial effect plots when keeping other factors not changing in Figure 1 and 2. The house age increased the housing prices when smaller than (around) 12, , but as the house increased, the increment of housing prices decreased. For example, the increment of housing price when house age was 5 was approximately 50k TWD per Ping; but the increment at house age of 10 was 20k TWD per Ping. When the house age was larger than (around) 12, the house age decreased the housing prices. The decrement of housing prices increased as the house age increased from 12 to 28 and then decreased as house ages increased when house age was larger than 28. For number of stores in the living circle on foot, the change of housing price over it was roughly linear. When smaller than 4, the number of stores decreased the housing prices; when larger than 4, the number of stores increased the housing prices. For geographic coordinates, we considered longitude and latitude together, so a contour plot was used to show the partial effect of them. We could see that the being at southwestern of Taiwan decreased the housing price and being at northeastern of Taiwan increased the housing price. For example, being at (24.94°N, 121.50°E) decreased the housing price by around 140k TWD per Ping; and being at (24.98°N, 121.54°E) increased the housing price by around 40k TWD per Ping.

We applied this GAM model to the test data set to predict housing price for transactions in test data set and then compared the difference between the actual transaction price and the predicted price.

The left plot in Figure 3 was the scatter plot comparing difference between actual housing price and predicted price for each transaction. The green line had slope of 1 and intercept of 0. Being close to the green line meant the actual price was close to the predicted price. The right plot in Figure 3 showed the distribution (histogram) and box-plot of the actual housing price minus predicted price value. If the actual price was close to predicted price, the value should be close to zero.

The correlation coefficient between the actual housing prices and the predicted housing price was 0.922 which was pretty close to 1. And in the scatter plot (left figure in Figure 3), the points were concentrated around the straight line with slope 1 and intercept 0 (green line), which meant the predicted housing price was close to the predicted price. In the box-plot and histogram, we could see that most of the difference concentrated from -5 to 5. From

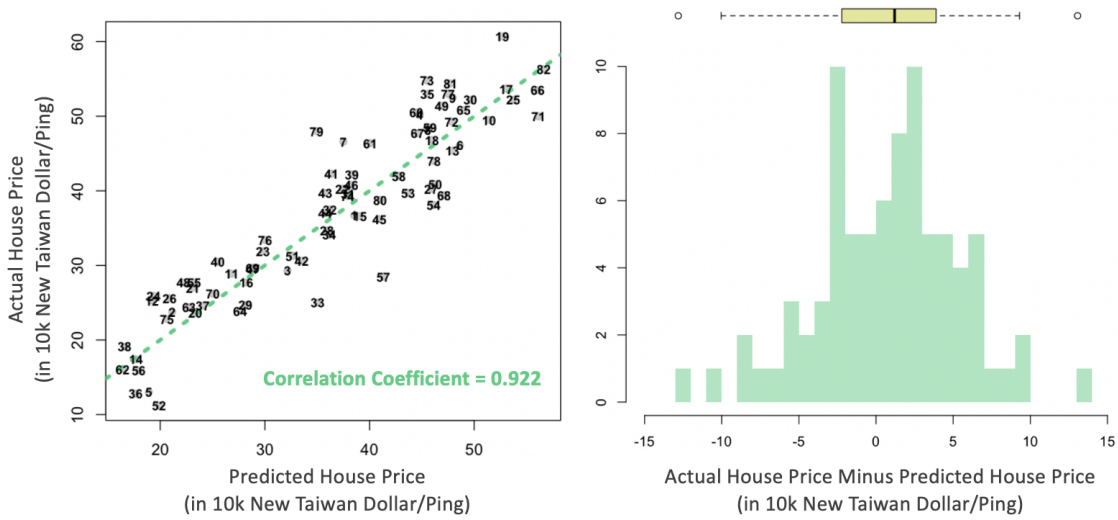


Figure 3: Difference between Actual Housing Prices and Predicted Prices.

Figure 3, it was indicated that the GAM model fitted the data well and had a good accuracy performance for data not included to train the model.

Furthermore, according to Figure 3, two abnormal housing prices were detected, whose actual price was over 12k TWD per Ping larger or smaller than the predicted price. They might be suspected of housing fraud based on the GAM.

4 Conclusion

In conclusion, we accomplished the goals of constructing a model in the form of computer program to predict the expected housing price using past housing data, with detailed description of how model is constructed, and explaining how different factors influence housing price.

Generally speaking, based on the GAM, the influences of transaction date, house age, distance to nearest MRT station, and longitude and latitude on the housing price were wiggly. The influence of number of stores within living circle on foot on the housing price was roughly linear.

The prediction accuracy of our GAM model was measured using correlation coefficient, scatter plot (actual price vs. predicted price), box plot and histogram (of difference between actual prices and predicted prices). The GAM achieved a high correlation coefficient of 0.922, and based on three plots, the predicted housing prices were pretty close to the actual housing prices. It suggested that our model performed well in predicting housing prices.

One possible limitation was that our analyses did not include inferences on GAM. Due to skill limit on it, for each predictor, we only included the rough changing trend of housing prices without confidence intervals; and only the predicted housing prices were provided by our model, but the prediction intervals were omitted, which might be far too complex for our purposes. Inferences on GAM can be taken into consideration in the future analysis to give more information on the influences of different factors on the housing prices and more information on the housing price prediction.

Documentation

We provided a function `predict_house_price` to predict the housing price.

Input:

- `train_data`: a data set that you plan to use to train the GAM. It should be in the format of data.frame, which must contain 7 columns named transaction_date (transaction date), house_age (house age), dist_MRT (distance to the closest MRT station), num_stores (number of stores in the living circle on foot), lat (latitude), long(longitude) and house_price (housing price).
- `test_data`: a data set that contains transaction you want to predict the housing price for. It should be in the format of data.frame and must contain 6 columns named transaction_date (transaction date), house_age (house age), dist_MRT (distance to the closest MRT station), num_stores (number of stores in the living circle on foot), lat (latitude) and long(longitude).

Output:

A data.frame that contains all 6 columns provided in `test_data` and an additional column named pred_price which contains the predicted housing price.

Examples:

In [30]:
```
# see the format and content of train_data
head(train_data)
```

A tibble: 6 × 7

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2012.916667	32.0	84.87882	10	24.98298	121.54024	37.9
2012.916667	19.5	306.59470	9	24.98034	121.53951	42.2
2013.583333	13.3	561.98450	5	24.98746	121.54391	47.3
2012.833333	5.0	390.56840	5	24.97937	121.54245	43.1
2012.666667	7.1	2175.03000	3	24.96305	121.51254	32.1
2013.416667	20.3	287.60250	6	24.98042	121.54228	46.7

In [34]:
```
# see the format and content of test_data
head(test_data)
```

A tibble: 6 × 6

transaction_date	house_age	dist_MRT	num_stores	lat	long
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2013.500000	13.1	1144.4360	4	24.99176	121.53456
2012.666667	20.4	2469.6450	4	24.96108	121.51046
2013.500000	15.2	3771.8950	0	24.93363	121.51158
2013.500000	14.4	169.9803	1	24.97369	121.52979
2013.000000	13.6	4197.3490	0	24.93885	121.50383
2013.166667	33.2	121.7262	10	24.98178	121.54059

In [35]:
```
# see how to use predict_house_price and see the format of output
pred_data = predict_house_price(train_data, test_data)
head(pred_data)
```

A tibble: 6 × 7

transaction_date	house_age	dist_MRT	num_stores	lat	long	pred_price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2013.500000	13.1	1144.4360	4	24.99176	121.53456	38.53726754
2012.666667	20.4	2469.6450	4	24.96108	121.51046	21.14236057
2013.500000	15.2	3771.8950	0	24.93363	121.51158	32.16139795
2013.500000	14.4	169.9803	1	24.97369	121.52979	44.79241870
2013.000000	13.6	4197.3490	0	24.93885	121.50383	18.90182992
2013.166667	33.2	121.7262	10	24.98178	121.54059	48.64404126

Appendix

In [1]:
```
# load required library
library(tidyverse)
library(ggplot2)
library(rmarkdown)
library(readxl)
library(mgcv)
library(AICcmodavg)
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.7     ✔ dplyr   1.0.9
✔ tidyr   1.2.0     ✔ stringr 1.4.0
✔ readr   2.1.2     ✔ forcats 0.5.1

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()

Loading required package: nlme


Attaching package: ‘nlme’


The following object is masked from ‘package:dplyr’:

    collapse


This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
```

In [2]:
```
# load data
data = read_excel("Real estate valuation data set.xlsx")
data = data[, -1]
colnames(data) = c("transaction_date", "house_age", "dist_MRT", "num_stores", '
head(data)
```

A tibble: 6 × 7

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2012.917	32.0	84.87882	10	24.98298	121.5402	37.9
2012.917	19.5	306.59470	9	24.98034	121.5395	42.2
2013.583	13.3	561.98450	5	24.98746	121.5439	47.3
2013.500	13.3	561.98450	5	24.98746	121.5439	54.8
2012.833	5.0	390.56840	5	24.97937	121.5425	43.1
2012.667	7.1	2175.03000	3	24.96305	121.5125	32.1

In [3]:
```
# check variable type
str(data)
```

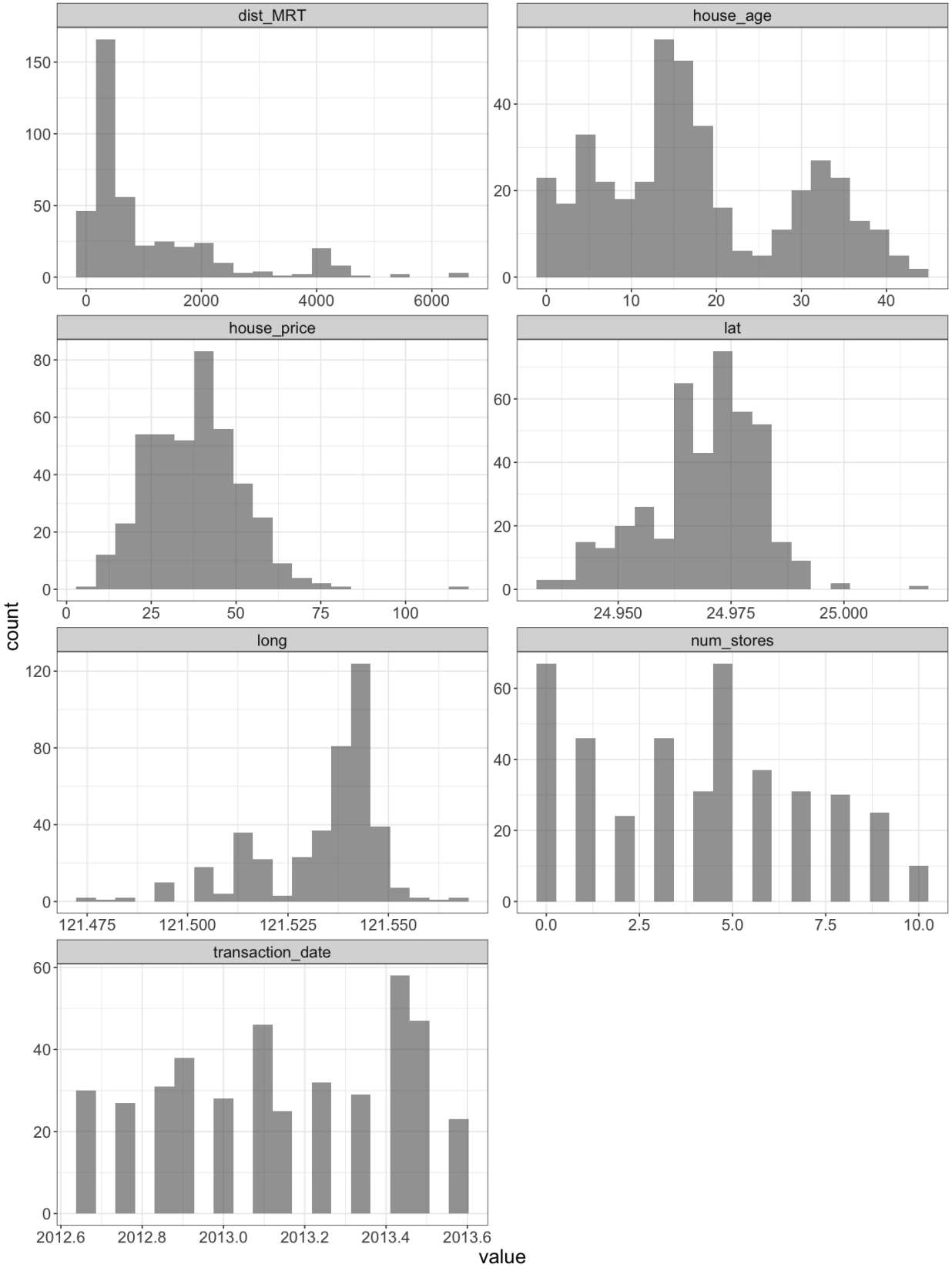
```
tibble [414 × 7] (S3: tbl_df/tbl/data.frame)
 $ transaction_date: num [1:414] 2013 2013 2014 2014 2013 ...
 $ house_age       : num [1:414] 32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9
...
 $ dist_MRT        : num [1:414] 84.9 306.6 562 562 390.6 ...
 $ num_stores      : num [1:414] 10 9 5 5 5 3 7 6 1 3 ...
 $ lat             : num [1:414] 25 25 25 25 25 ...
 $ long            : num [1:414] 122 122 122 122 122 ...
 $ house_price     : num [1:414] 37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8
22.1 ...
```

In [4]:
```
# summary on data
options(digits = 10)
summary(data)
```

```
 transaction_date     house_age          dist_MRT            num_stores
 Min.   :2012.667   Min.   : 0.00000   Min.   :  23.38284   Min.   : 0.000000
 1st Qu.:2012.917   1st Qu.: 9.02500   1st Qu.: 289.32480   1st Qu.: 1.000000
 Median :2013.167   Median :16.10000   Median : 492.23130   Median : 4.000000
 Mean   :2013.149   Mean   :17.71256   Mean   :1083.88569   Mean   : 4.094203
 3rd Qu.:2013.417   3rd Qu.:28.15000   3rd Qu.:1454.27900   3rd Qu.: 6.000000
 Max.   :2013.583   Max.   :43.80000   Max.   :6488.02100   Max.   :10.000000
      lat              long           house_price
 Min.   :24.93207   Min.   :121.4735   Min.   :  7.60000
 1st Qu.:24.96300   1st Qu.:121.5281   1st Qu.: 27.70000
 Median :24.97110   Median :121.5386   Median : 38.45000
 Mean   :24.96903   Mean   :121.5334   Mean   : 37.98019
 3rd Qu.:24.97745   3rd Qu.:121.5433   3rd Qu.: 46.60000
 Max.   :25.01459   Max.   :121.5663   Max.   :117.50000
```

Exploraroty Analysis

In [5]:
```
cat.vars = c('transaction_year', 'transaction_month')
num.vars = colnames(data)[which(!colnames(data) %in% cat.vars)]
num.hist = data[, num.vars] %>% gather(key = "variable", value = "value")
options(repr.plot.width = 12, repr.plot.height = 16)
# histogram for numerical variables
num.hist %>% ggplot() +
    geom_histogram(aes(x = value), bins = 20, alpha=0.6, position = "identity")
    facet_wrap(~variable, scales = 'free', ncol = 2) + theme_bw() +
    theme(text = element_text(size = 18))
```



In [6]:
```
data %>% filter(house_price > 100)
```

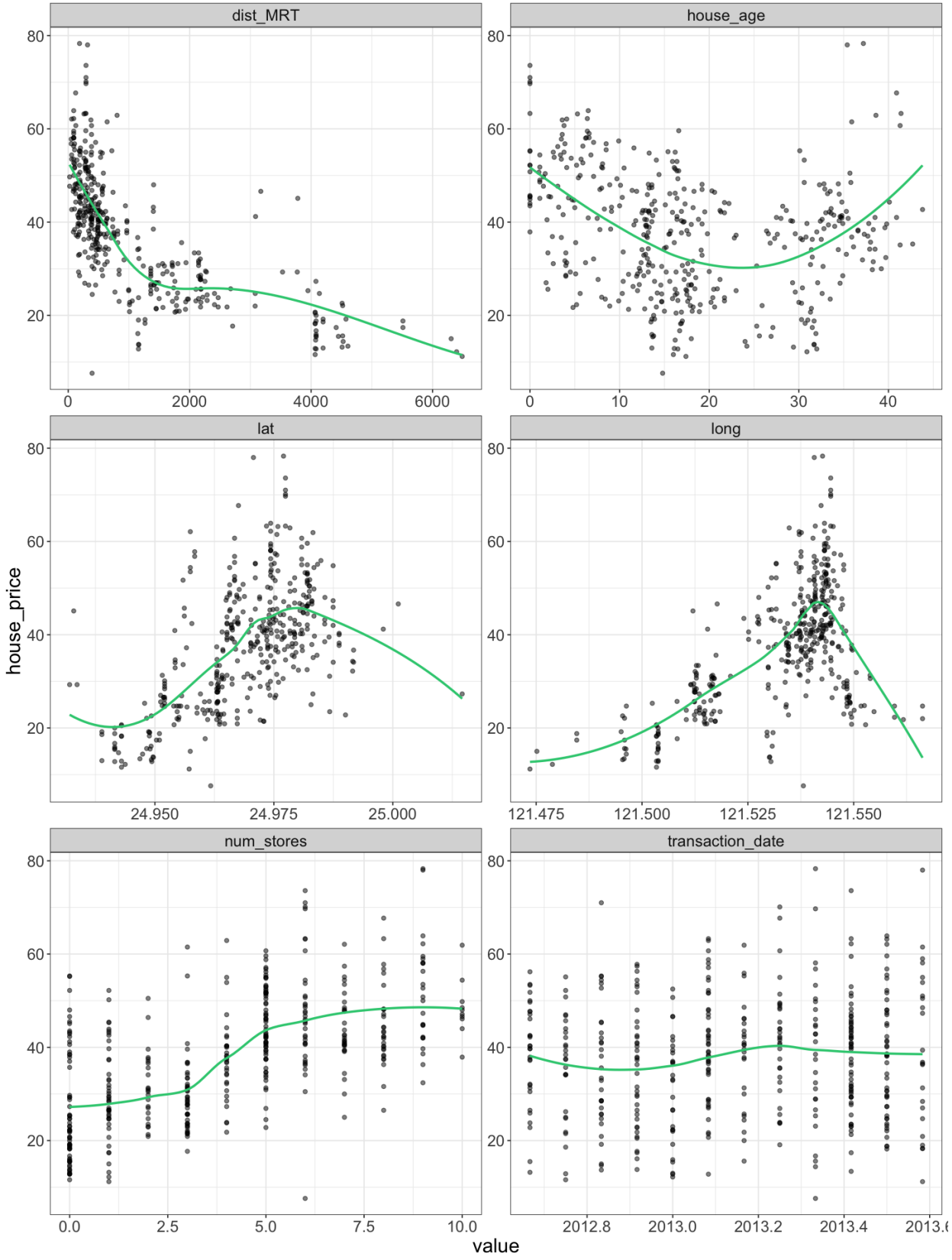
A tibble: 1 × 7

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2013.333333	10.8	252.5822	1	24.9746	121.53046	117.5

In [7]:
```
# remove the house price outlier
data = data %>% filter(house_price <= 100)
```

In [8]:
```
# rough trend of house price over different variables
num.points = data[, num.vars] %>% gather(key = "variable", value = "value", -c(
options(repr.plot.width = 12, repr.plot.height = 16)
# scatter plots for numerical variables
num.points %>% ggplot() +
    geom_point(aes(x = value, y = house_price), alpha = 0.5) +
    geom_smooth(aes(x = value, y = house_price), color = rgb(0.2,0.8,0.5), se=F
    facet_wrap(~variable, scales = 'free', ncol = 2) + theme_bw() +
    theme(text = element_text(size = 18))
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



In [9]:
```
# separate training and test set
set.seed(123)
test_index = sample(1:nrow(data), size = nrow(data)*0.2)
test_data = data[test_index, ]
train_data = data[-test_index, ]
```

In [10]:
```
summary(train_data)
```

```
 transaction_date     house_age          dist_MRT           num_stores
 Min.   :2012.667   Min.   : 0.00000   Min.   :  23.38284   Min.   : 0.000000
 1st Qu.:2012.917   1st Qu.: 8.95000   1st Qu.: 289.32480   1st Qu.: 1.000000
 Median :2013.167   Median :16.20000   Median : 488.81930   Median : 5.000000
 Mean   :2013.158   Mean   :17.60514   Mean   :1065.33899   Mean   : 4.226586
 3rd Qu.:2013.417   3rd Qu.:27.40000   3rd Qu.:1448.50400   3rd Qu.: 6.500000
 Max.   :2013.583   Max.   :43.80000   Max.   :6396.28300   Max.   :10.000000
      lat              long           house_price
 Min.   :24.93207   Min.   :121.4752   Min.   : 7.60000
 1st Qu.:24.96299   1st Qu.:121.5298   1st Qu.:27.70000
 Median :24.97073   Median :121.5389   Median :38.40000
 Mean   :24.96891   Mean   :121.5336   Mean   :37.94381
 3rd Qu.:24.97744   3rd Qu.:121.5433   3rd Qu.:45.95000
 Max.   :25.01459   Max.   :121.5663   Max.   :78.30000
```

In [11]:
```
sqrt(diag(var(train_data[, num.vars])))
```

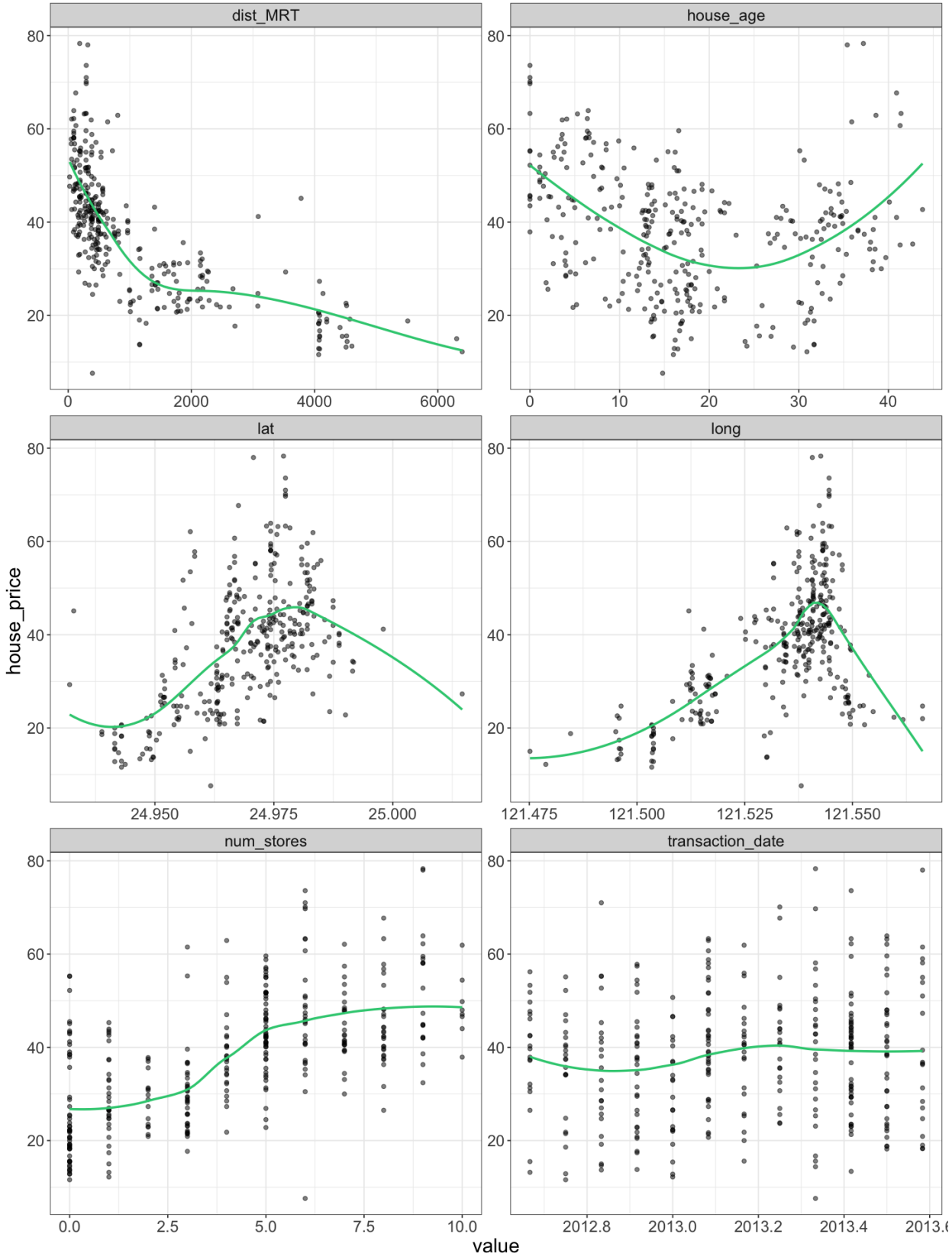
transaction_date: 0.277316820155285 **house_age:** 11.3930123811846 **dist_MRT:** 1265.17722784331 **num_stores:** 2.95829908249711 **lat:** 0.0123609536712329 **long:** 0.0152255362767727 **house_price:** 13.291869787025

In [12]:
```
summary(test_data)
```

```
 transaction_date     house_age          dist_MRT           num_stores
 Min.   :2012.667   Min.   : 0.00000   Min.   :  56.47425   Min.   : 0.000000
 1st Qu.:2012.833   1st Qu.: 9.67500   1st Qu.: 326.62307   1st Qu.: 1.000000
 Median :2013.083   Median :15.90000   Median : 620.59325   Median : 3.000000
 Mean   :2013.110   Mean   :18.23049   Mean   :1168.88888   Mean   : 3.597561
 3rd Qu.:2013.417   3rd Qu.:30.37500   3rd Qu.:1495.25175   3rd Qu.: 5.000000
 Max.   :2013.583   Max.   :39.80000   Max.   :6488.02100   Max.   :10.000000
      lat              long           house_price
 Min.   :24.93363   Min.   :121.4735   Min.   :11.20000
 1st Qu.:24.96304   1st Qu.:121.5202   1st Qu.:27.70000
 Median :24.97238   Median :121.5375   Median :38.45000
 Mean   :24.96946   Mean   :121.5326   Mean   :37.15732
 3rd Qu.:24.97937   3rd Qu.:121.5434   3rd Qu.:47.92500
 Max.   :25.00115   Max.   :121.5596   Max.   :60.70000
```

In [13]:
```
# rough trend of house price over different variables for training data
num.points = train_data[, num.vars] %>%
  gather(key = "variable", value = "value", -c("house_price"))
options(repr.plot.width = 12, repr.plot.height = 16)
# scatter plots for numerical variables
num.points %>% ggplot() +
    geom_point(aes(x = value, y = house_price), alpha = 0.5) +
    geom_smooth(aes(x = value, y = house_price), color = rgb(0.2,0.8,0.5), se=F
    facet_wrap(~variable, scales = 'free', ncol = 2) + theme_bw() +
    theme(text = element_text(size = 18))
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Modeling

In [14]:
```
# splines with interaction between long and lat
gam_fit = gam(house_price~s(transaction_date)+s(house_age)+s(dist_MRT)+s(num_st
              data=train_data)
summary(gam_fit)
```

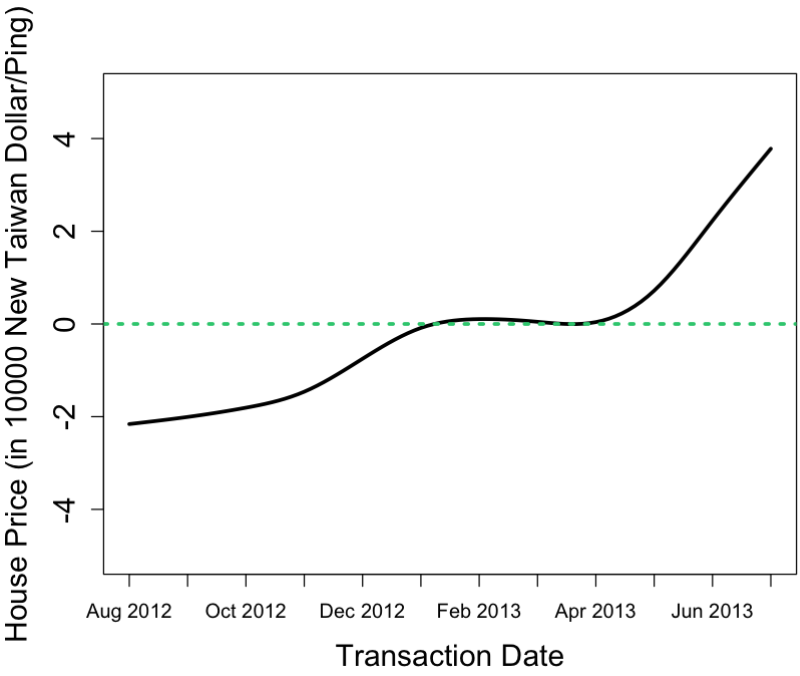
```
Family: gaussian
Link function: identity

Formula:
house_price ~ s(transaction_date) + s(house_age) + s(dist_MRT) +
    s(num_stores) + s(long, lat)

Parametric coefficients:
             Estimate Std. Error  t value   Pr(>|t|)
(Intercept) 37.9438066  0.3662655 103.5965 < 2.22e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                         edf    Ref.df        F    p-value
s(transaction_date)  3.717528  4.592078  3.88813  0.0028673 **
s(house_age)         2.766434  3.427270 19.41146 < 2.22e-16 ***
s(dist_MRT)          5.675779  6.578058  3.54893  0.0097016 **
s(num_stores)        1.000000  1.000000  3.00804  0.0838828 .
s(long,lat)         17.428087 21.426075  3.82184 < 2.22e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.749   Deviance explained = 77.2%
GCV = 49.088  Scale est. = 44.404    n = 331
```

Partial Effect of Covariates

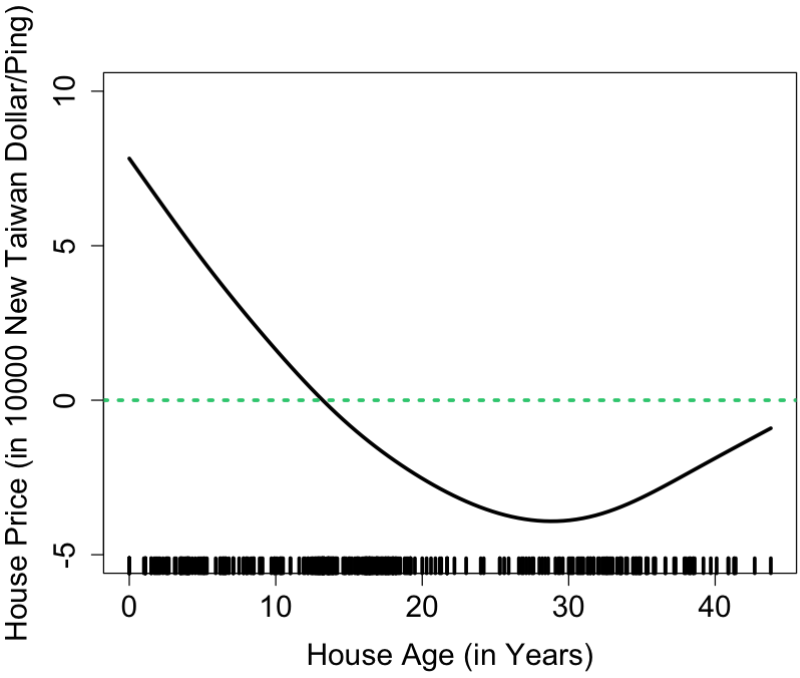
In [15]:
```
# base house price
coef(gam_fit)[1]
```

(Intercept): 37.9438066465274

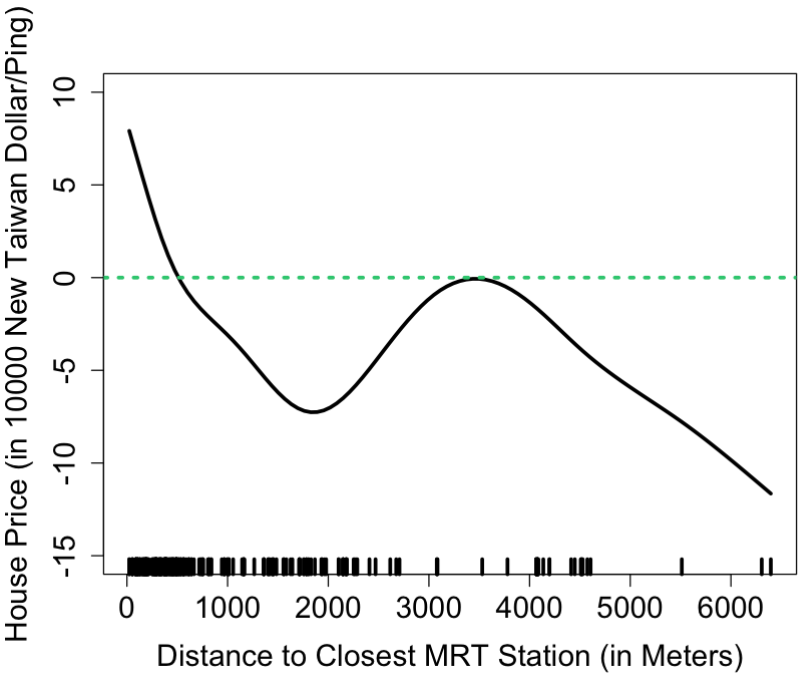
In [16]:
```
options(repr.plot.width = 7, repr.plot.height = 6)
plot(gam_fit, shade=F, se=F, select=1, lwd=3, ylim=c(-5,5), rug=T, cex.lab=1.5,
     shade.col=, xlab="Transaction Date",
     ylab="House Price (in 10000 New Taiwan Dollar/Ping)", xaxt="n")
axis(1, at = sort(unique(train_data$transaction_date)), las=1,
     labels = c("Aug 2012", "Sep 2012", "Oct 2012", "Nov 2012", "Dec 2012", "Ja
                "Feb 2013", "Mar 2013", "Apr 2013", "May 2013", "Jun 2013", "Ju
abline(h=0, lty=3, col=rgb(0.2,0.8,0.5), lwd=3)
```



In [17]:
```
#jpeg(file="ha.jpeg", height = 960, width = 1120)
plot(gam_fit, shade=F, se=F, select=2, lwd=3, ylim=c(-5, 10), rug=T, cex.lab=1.
    xlab="House Age (in Years)",
     ylab="House Price (in 10000 New Taiwan Dollar/Ping)")
abline(h=0, lty=3, col=rgb(0.2,0.8,0.5), lwd=3)
#dev.off()
```



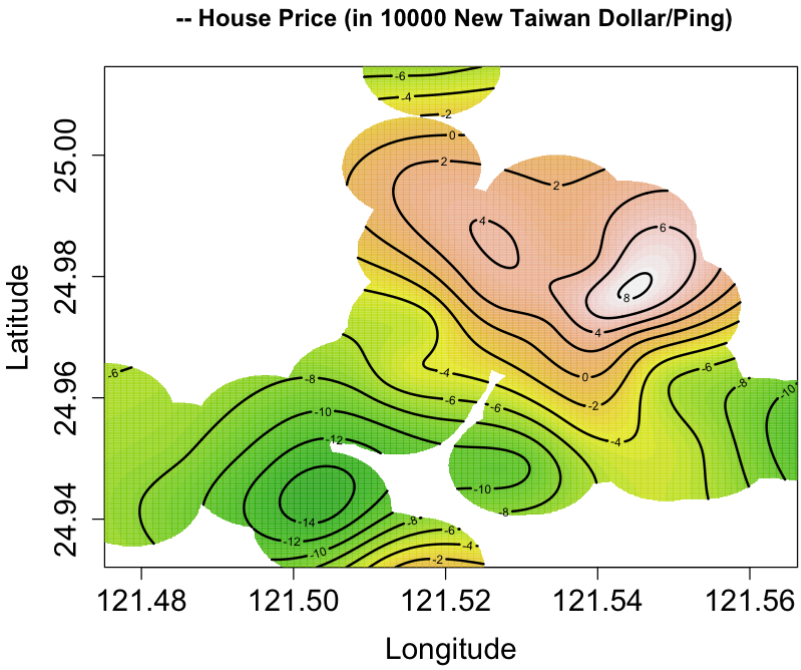
In [18]:
```
#jpeg(file="dm.jpeg", height = 960, width = 1120)
plot(gam_fit, shade=F, se=F, select=3, lwd=3, ylim=c(-15, 10), rug=T, cex.lab=1
     xlab="Distance to Closest MRT Station (in Meters)",
     ylab="House Price (in 10000 New Taiwan Dollar/Ping)")
abline(h=0, lty=3, col=rgb(0.2,0.8,0.5), lwd=3)
#dev.off()
```



In [19]:
```
#jpeg(file="ns.jpeg", height = 960, width = 1120)
plot(gam_fit, shade=F, se=F, select=4, lwd=3, ylim=c(-3, 4), rug=T, cex.lab=1.5
     xlab="Number of Stores in the Living Circle on Foot",
     ylab="House Price (in 10000 New Taiwan Dollar/Ping)")
abline(h=0, lty=3, col=rgb(0.2,0.8,0.5), lwd=3)
#dev.off()
```

In [20]:
```
#jpeg(file="ll.jpeg", height = 960, width = 1120)
plot(gam_fit, se=F, select=5, lwd=2, rug=FALSE, n2=800, scheme=2, cex.lab=1.5,
     hcolors=terrain.colors(50, alpha=0.7), contour.col="black",
     xlab="Longitude", ylab="Latitude",
     main=" -- House Price (in 10000 New Taiwan Dollar/Ping)")
# legend(121.52, 25.01, legend=c("House Price (in 10000\nNew Taiwan Dollar/Ping
```

```
#        col=c("black"), lty=1, cex=1.2, box.lty=0)
# dev.off()
```



Model Performance on Test Data

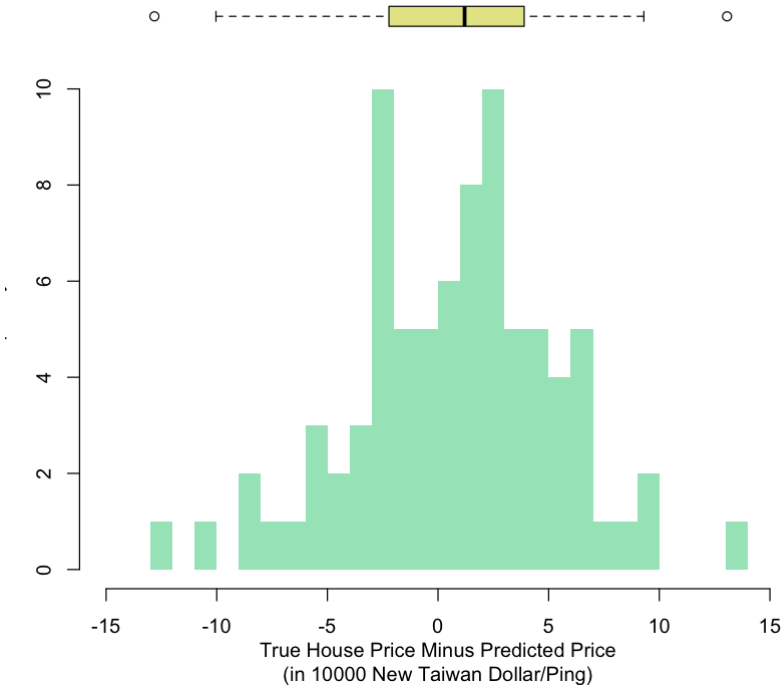
In [21]:
```
# use our model to predict the house price of test data set
pred_price = predict.gam(gam_fit, newdata = test_data, type = "response")
pred_price
```

1: 38.5372675389232 **2:** 21.142360566413 **3:** 32.1613979522419 **4:** 44.7924187031164 **5:** 18.9018299208571 **6:** 48.6440412570529 **7:** 37.5020601193266 **8:** 45.587711503003 **9:** 47.9398817332093 **10:** 51.4295832340915 **11:** 26.8217474465483 **12:** 19.2115832266446 **13:** 47.9164507348782 **14:** 17.6887433038821 **15:** 39.069717043831 **16:** 28.2314218624073 **17:** 53.0622304233257 **18:** 45.9758056477055 **19:** 52.6842121271546 **20:** 23.3567444128397 **21:** 23.1132160118901 **22:** 37.4100566935106 **23:** 29.8144935084017 **24:** 19.3802340517216 **25:** 53.7346221774069 **26:** 20.9044241897862 **27:** 45.9020592886491 **28:** 35.9294073432726 **29:** 28.1473349085998 **30:** 49.6378821635893 **31:** 37.938146729748 **32:** 36.2020591831386 **33:** 35.0356421694873 **34:** 36.1496269554694 **35:** 45.5113530795534 **36:** 17.6536106753682 **37:** 24.0603266299563 **38:** 16.6014257410708 **39:** 38.306397587743 **40:** 25.5044864176539 **41:** 36.3628112581597 **42:** 33.5177203632963 **43:** 35.7711224244148 **44:** 35.765044647348 **45:** 40.9449367935838 **46:** 38.2664099303135 **47:** 28.8598611770305 **48:** 22.2279808191173 **49:** 46.9044240740998 **50:** 46.2680557494721 **51:** 32.6273156870358 **52:** 19.8834152062862 **53:** 43.6913023357919 **54:** 46.1309621766637 **55:** 23.2445126165594 **56:** 17.9455236195335 **57:** 41.3180474280862 **58:** 42.8011339071557 **59:** 45.7704657403356 **60:** 44.4682572409543 **61:** 40.0682341441901 **62:** 16.4168630795502 **63:** 22.7577469082926 **64:** 27.6391944356499 **65:** 48.9807588842218 **66:** 56.0698852129701 **67:** 44.5878019735863 **68:** 47.1291743433358 **69:** 28.8274729801937 **70:** 25.0298719172222 **71:** 56.1310914195819 **72:** 47.8478086803834 **73:** 45.4987092020741 **74:** 37.8946438980667 **75:** 20.6342442014215 **76:** 30.0213284143811 **77:** 47.5007895250126 **78:** 46.1637242693831 **79:** 34.9381239115754 **80:** 41.0177670023668 **81:** 47.7318136974406 **82:** 56.6462480142444

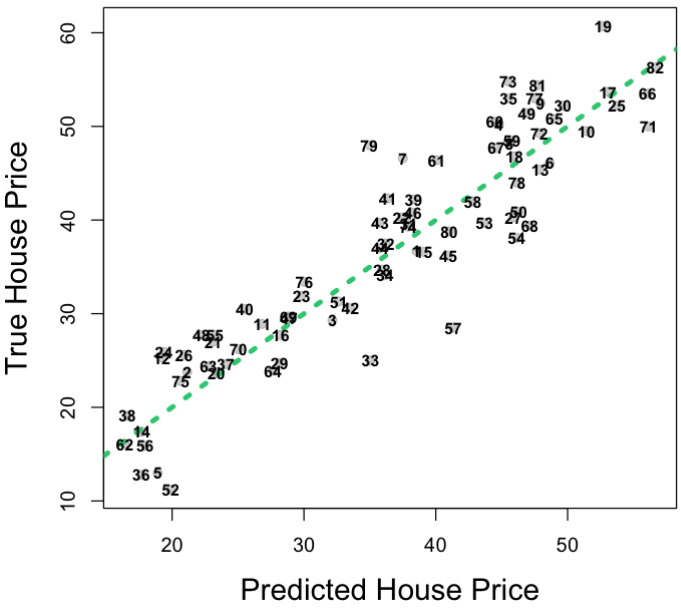
In [22]:
```
# correlation between predicted price and actual price
cor(pred_price, test_data$house_price)
```

0.922336778397262

In [23]:
```
# anomaly suggested by our model
options(repr.plot.width=7, repr.plot.height=6)
diff = test_data$house_price - pred_price
# Layout to split the screen
layout(mat = matrix(c(1,2),2,1, byrow=TRUE),  height = c(1,8))

# Draw the boxplot and the histogram
par(mar=c(0, 3.1, 1.1, 2.1))
boxplot(diff, horizontal=TRUE , ylim=c(-15,15), xaxt="n", col=rgb(0.8,0.8,0,0.5
par(mar=c(4, 3.1, 1.1, 2.1))
hist(diff, breaks=19, col=rgb(0.2,0.8,0.5,0.5), border=F , main="" ,
     xlab="True House Price Minus Predicted Price\n(in 10000 New Taiwan Dollar/
```



In [24]:
```
options(repr.plot.width=6, repr.plot.height=6)
plot(x=pred_price, y=test_data$house_price, pch=19, cex.lab=1.5,
     xlab="Predicted House Price",
     ylab="True House Price", col=rgb(0,0,0,0.2)
    )
abline(a=0,b=1, col=rgb(0.2,0.8,0.5), lwd=4, lty=3)
text(pred_price, test_data$house_price, labels = c(1:length(pred_price)), cex=0
```



In [25]:
```
test_data$pred_price = pred_price
```

In [26]:
```
test_data %>% filter(abs(house_price-pred_price)>12)
```

A tibble: 2 × 8

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price	pred
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
2012.833333	12.7	187.4823	1	24.97388	121.52981	28.5	41.318
2012.916667	8.9	1406.4300	0	24.98573	121.52758	48.0	34.938

In [27]:
```
which(abs(test_data$house_price-pred_price)>12)
```

57: 57 **79:** 79

Integrated Function

In [28]:
```
predict_house_price = function(train_data, test_data) {
    model = gam(house_price~s(transaction_date)+s(house_age)+s(dist_MRT)+s(num_
              data=train_data)
    pred_price = predict.gam(model, newdata = test_data, type = "response")
    res = test_data
    res$pred_price = pred_price
    return (res)
}
```

In [29]:
```
predict_house_price(train_data, test_data)
```

A tibble: 82 × 8

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price	pr
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
2013.500000	13.1	1144.4360	4	24.99176	121.53456	36.7	38.5
2012.666667	20.4	2469.6450	4	24.96108	121.51046	23.8	21.1
2013.500000	15.2	3771.8950	0	24.93363	121.51158	29.3	32.1
2013.500000	14.4	169.9803	1	24.97369	121.52979	50.2	44.7
2013.000000	13.6	4197.3490	0	24.93885	121.50383	13.0	18.9
2013.166667	33.2	121.7262	10	24.98178	121.54059	46.1	48.6
2013.416667	11.9	3171.3290	0	25.00115	121.51776	46.6	37.5
2013.416667	32.8	204.1705	8	24.98236	121.53923	48.2	45.5
2013.000000	8.1	104.8101	5	24.96674	121.54067	52.5	47.9
2013.250000	5.4	390.5684	5	24.97937	121.54245	49.5	51.4
2013.333333	12.0	1360.1390	1	24.95204	121.54842	28.9	26.8
2013.500000	23.0	3947.9450	0	24.94783	121.50243	25.3	19.2
2012.833333	0.0	274.0144	1	24.97480	121.53059	45.4	47.9
2013.416667	31.5	5512.0380	1	24.95095	121.48458	17.4	17.6
2013.000000	22.8	707.9067	2	24.98100	121.54713	36.6	39.0
2013.083333	7.6	2175.0300	3	24.96305	121.51254	27.7	28.2
2012.833333	4.6	259.6607	6	24.97585	121.54516	53.7	53.0
2012.750000	11.4	390.5684	5	24.97937	121.54245	46.8	45.9
2013.250000	3.8	383.8624	5	24.98085	121.54391	60.7	52.6
2012.833333	15.9	1497.7130	3	24.97003	121.51696	23.6	23.3
2013.083333	29.3	1487.8680	2	24.97542	121.51726	27.0	23.1
2012.666667	34.5	623.4731	7	24.97933	121.53642	40.3	37.4
2012.666667	37.1	918.6357	1	24.97198	121.55063	31.9	29.8
2012.666667	30.4	1735.5950	2	24.96464	121.51623	25.9	19.3
2013.500000	5.2	390.5684	5	24.97937	121.54245	52.2	53.7
2012.833333	20.5	2185.1280	3	24.96322	121.51237	25.6	20.9
2013.500000	11.8	533.4762	4	24.97445	121.54765	40.3	45.9
2013.416667	36.1	519.4617	5	24.96305	121.53758	34.7	35.9
2013.500000	5.6	2408.9930	0	24.95505	121.55964	24.7	28.1
2013.083333	0.0	274.0144	1	24.97480	121.53059	52.2	49.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
2012.666667	30.9	161.94200	9	24.98353	121.53966	39.7	43.6
2013.500000	26.4	335.52730	6	24.97960	121.54140	38.1	46.1

transaction_date	house_age	dist_MRT	num_stores	lat	long	house_price	pre
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
2013.416667	21.2	2185.12800	3	24.96322	121.51237	27.7	23.2
2013.000000	13.6	4082.01500	0	24.94155	121.50381	15.9	17.9
2012.833333	12.7	187.48230	1	24.97388	121.52981	28.5	41.3
2013.250000	13.3	250.63100	7	24.96606	121.54297	42.0	42.8
2013.583333	30.6	431.11140	10	24.98123	121.53743	48.5	45.7
2013.583333	35.7	579.20830	2	24.98240	121.54619	50.5	44.4
2013.083333	20.6	737.91610	2	24.98092	121.54739	46.4	40.0
2012.916667	31.9	1146.32900	0	24.94920	121.53076	16.1	16.4
2013.166667	16.2	2288.01100	3	24.95885	121.51359	24.4	22.7
2013.250000	8.0	2216.61200	4	24.96007	121.51361	23.9	27.6
2012.750000	0.0	338.96790	9	24.96853	121.54413	50.8	48.9
2012.666667	5.7	90.45606	9	24.97433	121.54310	53.5	56.0
2012.666667	3.1	577.96150	6	24.97201	121.54722	47.7	44.5
2013.000000	11.6	390.56840	5	24.97937	121.54245	39.4	47.1
2013.500000	18.1	837.72330	0	24.96334	121.54767	29.7	28.8
2012.750000	11.5	1360.13900	1	24.95204	121.54842	26.2	25.0
2012.666667	5.6	90.45606	9	24.97433	121.54310	50.0	56.1
2013.250000	16.5	323.65500	6	24.97841	121.54281	49.3	47.8
2013.500000	13.3	561.98450	5	24.98746	121.54391	54.8	45.4
2012.916667	13.0	492.23130	5	24.96515	121.53737	39.3	37.8
2012.666667	20.2	2185.12800	3	24.96322	121.51237	22.8	20.6
2012.833333	2.0	2077.39000	3	24.96357	121.51329	33.4	30.0
2013.416667	16.4	289.32480	5	24.98203	121.54348	53.0	47.5
2012.750000	0.0	208.39050	6	24.95618	121.53844	44.0	46.1
2012.916667	8.9	1406.43000	0	24.98573	121.52758	48.0	34.9
2013.000000	39.6	480.69770	4	24.97353	121.53885	38.8	41.0
2012.833333	3.4	56.47425	7	24.95744	121.53711	54.4	47.7
2012.916667	5.9	90.45606	9	24.97433	121.54310	56.3	56.6

In [ ]: