

Case Study 3 – Machine Learning

DS501 Simon Chuang

1. Please select any data set from below links or any other suitable dataset

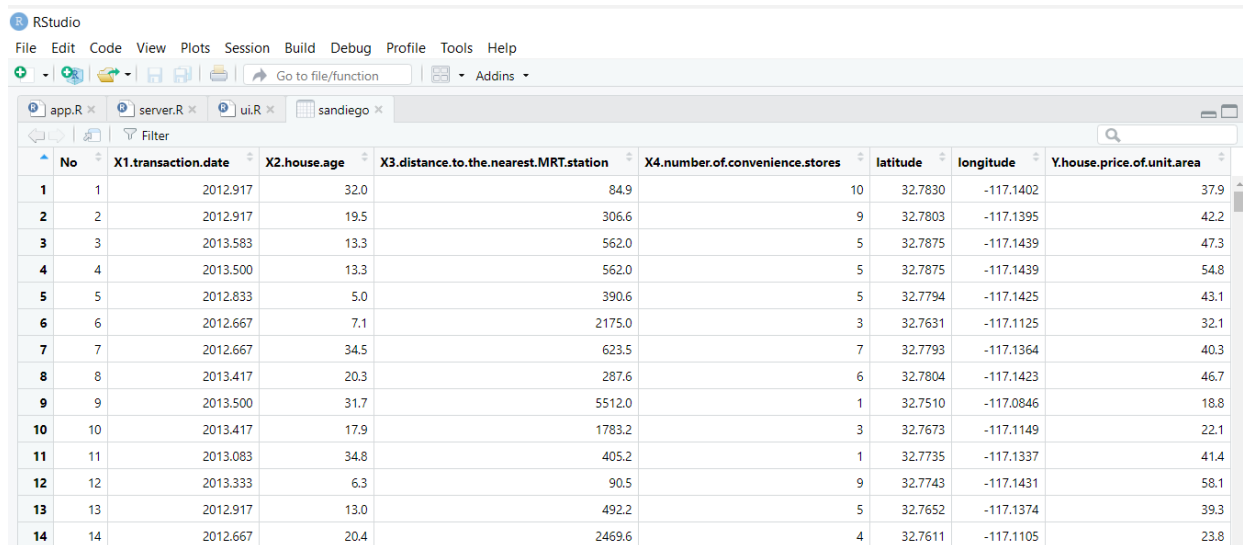
<http://archite.ics.uci.edu/ml/datasets.html> <https://www.kaggle.com/datasets>

Used Real Estate Valuation Data Set. Match the project requirements that it is a small data set of 414 rows with the following columns as multivariate data set. The following are the columns:

Variable	Description as Part of the Variable Name
Row number	Not a variable
X1	X1.Transaction.date
X2	X2.house.age
X3	X3.distance to the nearest MRT station
X4	number of convenience stores
X5	latitude
X6	longitude
Y	house price of unit area

Table. 1 Real Estate Valuation Data

Each of these variable names is words concatenated with “.”, which conveniences the data analyst. When brought into the RStudio and for a later data analysis, the following is a sample of first several rows:



No	X1.transaction.date	X2.house.age	X3.distance.to.the.nearest.MRT.station	X4.number.of.convenience.stores	latitude	longitude	Y.house.price.of.unit.area
1	2012.917	32.0	84.9	10	32.7830	-117.1402	37.9
2	2012.917	19.5	306.6	9	32.7803	-117.1395	42.2
3	2013.583	13.3	562.0	5	32.7875	-117.1439	47.3
4	2013.500	13.3	562.0	5	32.7875	-117.1439	54.8
5	2012.833	5.0	390.6	5	32.7794	-117.1425	43.1
6	2012.667	7.1	2175.0	3	32.7631	-117.1125	32.1
7	2012.667	34.5	623.5	7	32.7793	-117.1364	40.3
8	2013.417	20.3	287.6	6	32.7804	-117.1423	46.7
9	2013.500	31.7	5512.0	1	32.7510	-117.0846	18.8
10	2013.417	17.9	1783.2	3	32.7673	-117.1149	22.1
11	2013.083	34.8	405.2	1	32.7735	-117.1337	41.4
12	2013.333	6.3	90.5	9	32.7743	-117.1431	58.1
13	2012.917	13.0	492.2	5	32.7652	-117.1374	39.3
14	2012.667	20.4	2469.6	4	32.7611	-117.1105	23.8

Table 2 Sample of House Valuation Data set

2. Select an algorithm suitable for the above data set (classification/Regression/Clustering/Other)

Because this is a real estate valuation data set, it is suitable for building a Linear Model for price predictions. In the class, we have learned Linear Classification, Linear Regression, and Logistic Regression Algorithms. Certainly, there are many other ones such as **Lasso Regression** to be learned in the ML/DML courses, which can be used to optimize the weighting and error functions. Those algorithms have more advanced Optimization techniques used in the Weighting functions and Error correction functions for problems such as the overfitting problem. But as discussed in the class, they are beyond the scope of DS501.

3. Explain the mathematical / statistical details of the algorithm

Linear regression is a mathematical/statistical technique used to model the relationships between observed variables. The idea behind simple **linear regression** is to "fit" the observations of two variables into a **linear** relationship between them. Graphically, the algorithm is to determine the line that fits best (goes closest) to the points (x_i, y_i) . The class notes also say x_i, y_i are observations of the two variables, X and Y, which are expected to depend linearly on each other.

As the lecture notes and many articles or books such as “Pattern Recognition and Machine Learning” by Bishop, Springer. 2006. In it, Chap 4 specifically describes Linear Models for Classification problem. The mathematics involves, as the lecture notes described in the chapter for PCA, Linear Algebra to find Eigen Vectors and Eigen Values for the new coordinate system in the N-dimensional space such that the these variables, each representing a dimension, can be as orthogonal to each other as possible. It is also common to add an error term to optimize the error in the model, which requires optimization techniques commonly used in mathematical analysis and statistics. PCA is a very powerful tool in coming up with variables for the linear models. For the UCI Real Estate Valuation data set, they gave 6 components to consider. In the real world of real estate valuation, it can be more than several dozens of parameters to consider.

Use the data acquired from the UCI link, there can be 6 X variables as described in Table 1. Among them, the best single variable example of would be the `x3.distance.to.the.nearest.MRT.station` versus `Y.house.price.per.unit.area` in Fig.1 below:

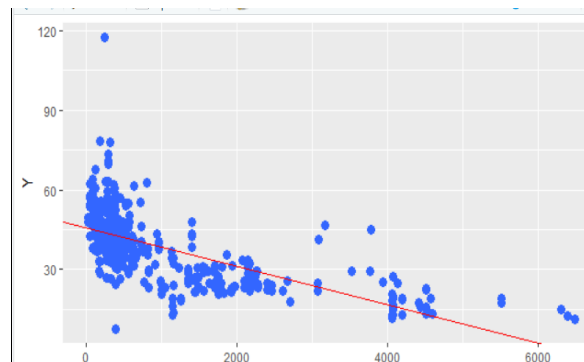


Fig. 1 Scatter plot to explain X_i, Y_i , observations and the best-fitting linear relationship between them

As we can easily see, the closer the property to the MRT station, the less time wasted in daily commute. Therefore, people are willing to pay for that. Regression is a common process used in many applications of statistics in the real world. There are at least two types of applications, we can imagine:

- **Predictions:** With a series of observations (of X-Y variables), regression analysis can model the relationship between the variables and come up with a statistical model. This model can then be used to generate predictions. In other words, with a series of two variables X_i and Y_i , the established model can be used to predict values of Y_j given future observations of X_j . Car sale prices, Stock prices, and Real Estate prices are just a few examples. This is why I chose to see if the UCI Real Estate Valuation Data Set is a good collection of info (high entropy) for Linear Model based Predictions.
- **Correlation:** The linear model produced by a regression analysis will often fit some kinds of data better than others. This linear model can then be used to analyze correlations between variables and to refine the statistical model. The purpose is to incorporate/include more variables or further inputs. As discussed in the class, the process can go the other way by reducing the number of variables after analyzing the data and the predictor outcome. As discussed in the class, there can be overfitting or under fitting cases. In other words, the initial model may describe a subset of the data points well, but predicts for other data points not as well. Then the process continues to examine/explain the differences between these subsets of data points. This process is common to Computer Software tests, e.g., software defect ratio versus software complexity (of number of function units, global variables, number of lines of code, and more.)

Although there can be quite a few measures of best fits, for most applications the best-fitting line is found using the method of Least Squares. In this way, we view Y as a linear function of X , the method finds the linear function lm which minimizes the sum of the squares of the errors in the approximations of the y_i by $lm(x_i)$. $lm()$ is exactly the function call to construct such model. In Case Study 3, we use, as discussed in the class: RMSEP or R2.

Though there are formulae that can be called directly, there are R codes manually written in the data analysis phase and in the server.R module to report the RMSEP and R2 values after one or more variable is selected for the Linear Model. Before we choose the X variables, as the class lecture 9 and lab notes describe, we need to choose X variables that are highly correlated. This is an iterative process. As described in the lecture preparing for Case Study 1, it is a scientific study with the following steps:

- Collect Data
- Examine the Data
- Simple Analysis
- Collect More Data and More Analysis

The aforementioned Real Estate Valuation data set collected by UCI is set, which saves a lot of time for data cleansing. It was also classified conveniently into variables of X_1, X_2, \dots, X_6 together with Y for observations. It is a great data set for pedagogical use.

Also, as described in the lecture for Case Study 2, a Data Science project has 6 or more phases, composed of the following:

- A. Discovery
- B. Initial Hypothesis
- C. Data Planning
- D. Model
- E. Results
- F. Business Impact

We can use R scripts and what we have learned from Lecture 8, 9 and previous lectures to perform A, B, C, D, and get E.

4. Create a shiny application giving end user options to change parameters and see how they affect the results. Please remember the Data Science life cycle lecture and follow the suggestions. Explain your data set and machine learning modeling methodology in the descriptive on your shiny application.

For Problem 4, it can be translated to the following Shiny App Designs:

- A. Allow the User to Choose and Visualize Correlation of one Variable X_i to Y_i , e.g. Scatter Plot.** We found Data Visualization very effective in Data Analysis and Data Planning Phases, which Case Study 2 already demonstrated that. Thus, this Shiny application should be effective in presenting data to the analyst via Data Visualization. For this Real Estate Valuation dataset, as Lecture 9 of Linear Regression Models discussed and showed. The Shiny Reactive Drop Down List (selected_var) is used:

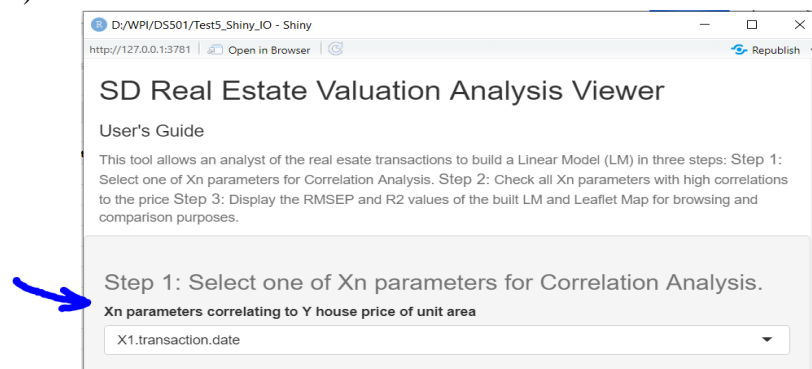


Fig. 2 Allow the user to select X_i for its Correlation to The Y house price of unit area

After selecting the X_i , its correlation value to the Y and scatter plot should be displayed as the following:

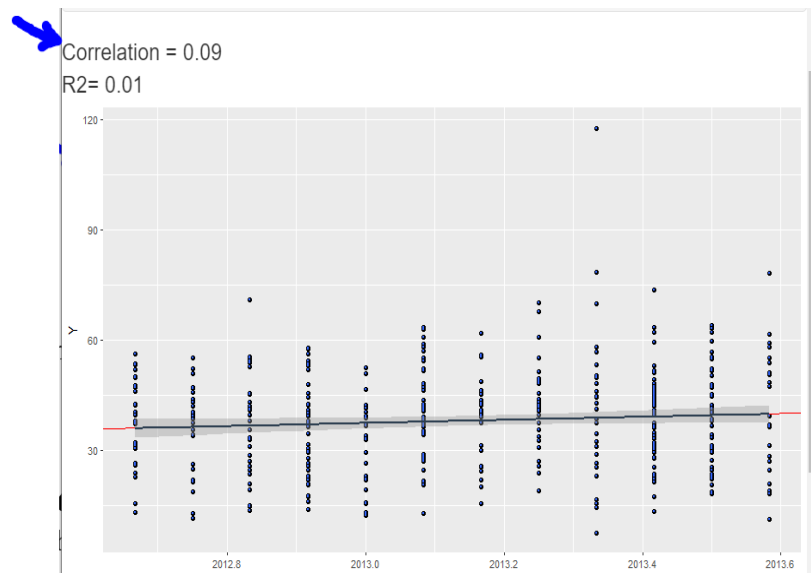


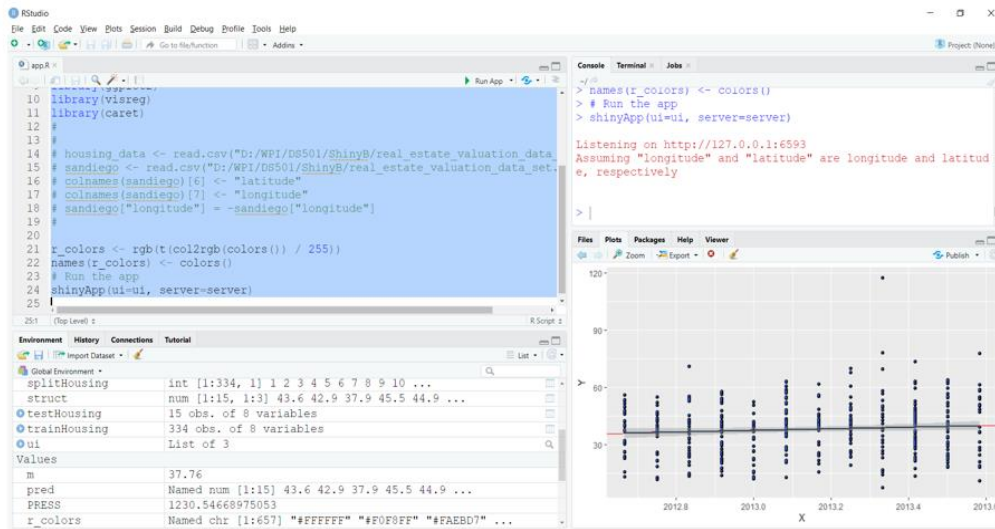
Fig. 3 Scatterplot of X_i and Y

Comparing the result of Fig. 3 to Fig. 1, we immediately can find that the X distance to the nearest MRT station correlates better than X transaction date to Y, House Price of Unit Area. The comparison of correlation values and scatter plots are complementary to each other. The visualization helps in quite a few ways in telling the data semantics and characteristics of data. For example, the closer the house location to the MRT station, the convenience it gets for commute and hence the price. Another one is that from this data set, the transaction date (seasonal effects) seems to uncorrelated to house price of unit area.

The implementation requires the following:

- The proficiency of R and RStudio to certain degree (learned in Case Study 1)
- A good understanding of Shiny framework (learned from Case Study 3)
- Data visualization and manipulation (learned in Case Study 2), e.g., Scatter Plot for front phases of

The following diagram shows RStudio is used to construct the Shiny based application for visualization using scatterplot.



For the data set, we found that the correlations of the X variables to Y were:

```

cor(x, y)
x <- housing_data[3:7]
y <- housing_data[8]
cor(x, y)

```

X Variable	Correlation
X1.house.transaction.date	0.0874906 (low)
X2.house.age	-0.2105670 (low)
X3.distance.to.the.nearest.MRT.station	-0.6736099
X4.number.of.convenience.stores	0.5710049
X5.latitude	0.5459911
X6.longitude	0.5235417

The higher correlation to 1.0, the better, when choosing X variables. Though these variables are good, especially the Distance to the nearest MRT station, anyone of them alone can hardly model the very complex real estate valuation and transactions. Since 11 years ago, Zillow, Red Fin has created more than \$2B annual business, which have outshined the Century 21 of \$1.5B, due to their data analytics and ML technologies. A reference can be found in the following:

<https://www.zillow.com/research/zestimate-forecast-methodology/>

B. Allow the User to Choose Any Combination of the X_i Parameters in the Data Set and See How They affect the (Prediction) Results.

This requirement translates best to Shiny designs of `checkGroup` to select any combination of X_i variables and see how they affect the predicted price in general, which can be reflected in the R^2 or RMSEP values. Thus, the design would require the following:

Fig. 5 is a RStudio screen capture whereas the requirement further asks for publish the app.R, ui.R, and server.R. Thus, the web page should provide the following for data analyst to browse the result:

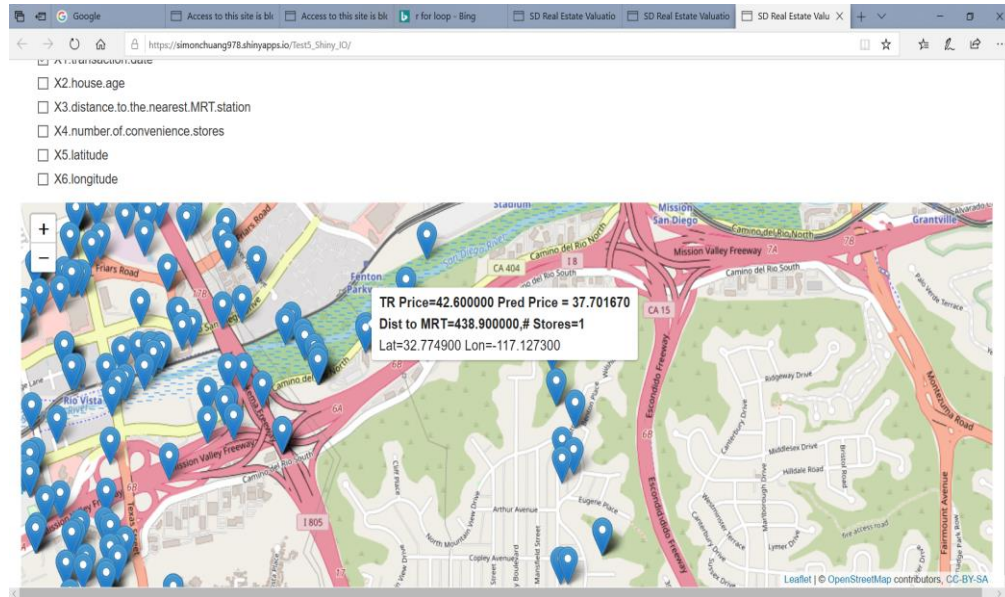


Fig.6 Published app.R by using the shiny.io tool

The implementation as a prototype data product, requires the following, adding upon what are required by Case Study 2 :

- I. A further proficiency of R and RStudio to handle 1) check box list and incorporate them to the formula for $Y \sim X$. 2) each the fitted values of the Linear Model for predicted Y – house price of unit area.
- II. A good understanding of Shiny framework (learned from Case Study 3) to construct a leaflet map with markers/labels of info
- III. Publish a webpage for this app.R on a designated shiny.io Linux server.
Resolve any issues when publishing the app.R/RStudio to the shiny.io/web server.

Item I can be implemented by using advanced R scripting composing of the “formula” object, i.e.,

```
# default the X variable
formula_string <- ""
if (length(input$checkGroup) == 0){
  formula_string = "X1.transaction.date"
} else {
  # compose the X part of the formula from check box
  for (i in 1:length(input$checkGroup)) {
    formula_string = paste(
      formula_string,
      toString(choice[input$checkGroup,"X.variable"][as.integer(i)]),
      "+"
    )
  }
  # rid of the extraneous + to complete the formula
  formula_string = substr(formula_string,1,nchar(formula_string)-1)
}

# compose the Y part of the formula string
```



```
formula_string = paste("Y.house.price.of.unit.area ~", formula_string)

# create the formula object
formula = as.formula(formula_string)
```

Subsequently, a Linear Model based on the X_i variables and Y can be constructed:

```
# creat the linear model based on the formula
housing_model <- lm(formula,
                    data=housing_data)
```

For the Linear Model to predict the price, train/test the model, and further tune the model would similar to what the Lectures in “Data Product engineering and its phases” described, would need to split the dataset into training and testing datasets to provide the R2 value, which hopefully would come close to 1.0 (100%) to indicate the accuracy of the model. In this process, the model has to be tuned to overcome any anomalies as Risk Factors.

Risk Factor 1: Data Planning and Data Modeling Related Issues

For this, we can imagine there could be a need to correct the model if the predicted Y value (house price of unit area) is negative. We also could run in to the dilemma of “Getting a positive Y value or getting a higher R2 value.” This dilemma forces us to include/exclude intercept, and add more X_i variables to explain the headscratchers. This also misleads the result, reduces the significance of our analysis and sabotages the entire predictability.

Lesson Learned 1: Model Tuning could take more time than expected

However, in this Real Estate Valuation Viewer project, it can be realized that there are times when we need to perform a regression analysis without the intercept i.e., when the model requires a process which has a zero-intercept. Thus, though regression analysis is a powerful statistical technique to make predictions, we have to use it wisely by trading manipulating the results for getting the most out of our data. To tune the model and predict better (no negative prices predicted), the following scripts were used in early stages of analyzing the model and data:

```
model_coef <- as.list(coef(housing_model))
housing_predictor <- function ( x2, x3, x4, x5, x6) {
  model_coef[["(Intercept)"]] +
    model_coef[["X2.house.age"]] * x2 +
    model_coef[["X3.distance.to.the.nearest.MRT.station"]] * x3 +
    model_coef[["X4.number.of.convenience.stores"]] * x4 +
    model_coef[["X5.latitude"]] * x5 +
    model_coef[["X6.longitude"]] * x6
}
```

Additionally, one of the many references of intercept, and 0 origin can be found. For example, [https://stats.libretexts.org/Bookshelves/Computing_and_Modeling/Supplemental_Modules_\(Computing_and_Modeling\)/Regression_Analysis/Simple_linear_regression/Regression_through_the_origin](https://stats.libretexts.org/Bookshelves/Computing_and_Modeling/Supplemental_Modules_(Computing_and_Modeling)/Regression_Analysis/Simple_linear_regression/Regression_through_the_origin)

Summarizing the Data Model construction and Tuning the Data Model and its associated the R2 value, which took a good amount of 2-3 calendar weeks yielding the following R script fragments, occupying and confirming the Data Planning, and Model Fine Tuning process is an iterative process and could take up to 50%-60% of the project life cycle. During the model prototyping the following code fragment is used to compute the RMSE and R2 values. In the Shiny GUI, the R2 values is shown:

```

# compute RMSE for output 7/20
# split housing_data as Training and Test sets
set.seed(2012)
splitHousing <- caret::createDataPartition (housing_data[,1], p = 0.8, list=F, times=1)

# Training data set
trainHousing <- housing_data[splitHousing, ]
head(trainHousing)

# Test data set
testHousing <- housing_data[!row.names(housing_data) %in% row.names(trainHousing),]
testHousing <- testHousing[-splitHousing,]

# library(Metrics)
# RMSEP Root - Mean Squared Error of Prediction
# Predicted Residual Sum of Squares
PRESS = sum((housing_data[, "Y.house.price.of.unit.area"] -
            housing_data[, "predict"])^2)
RMSEP = sqrt(PRESS/nrow(housing_data))
print("RMSEP=")
print(RMSEP)

# SST Sum of Squares of difference between individual mean
# Note: Word could make normal quote, apostrophe, and minus to ", ', - (dash)
#
m = mean(housing_data[, "Y.house.price.of.unit.area"])
housing_data[, "Mean"] = 0.0
housing_data[, "Mean"] = m
housing_data[, "St"] = 0.0
housing_data[, "St"] = housing_data[, "Y.house.price.of.unit.area"] - m
SST = sum((housing_data[, "St"])^2)
# print("SST=", SST)

# R2
R2 = 1- (PRESS/SST)
print ("R2= ")
print (R2)

output$checkGroup <- renderText({
  paste("R2= ", print(round(R2,2)))
})
# End of RMSEP computation

```

For other requirements such as Explain your data set and machine learning modeling methodology, they have been described in answers to problem 1, 2, 3, and prior paragraphs in this problem.

5. Deploy your application to <https://www.shiny.io>

Risk Factor 2: The shiny.io platform related issues

Internet is never a 100% reliable platform for IoT though marking hypes insist. Routers/switches, servers, carriers, and many other things are usually good but they could go wrong at times. This experience translates to Case Study 3 project too. (Also happened to Case Study 1, which Internet access provided by the carrier was very bad to ground everything of finishing the study to a halt.)

Though the app.R prototype in RStudio runs fine, publishing app.R to the shiny.io web server may not. In a 3 days publish / republish effort from RStudio to the shiny.io web server, the following platform errors were encountered repeatedly in the first day; the next day it went away luckily.

```
Attaching package: 'ggplot2'
The following object is masked from 'package:ggvis':
  resolution
```

```
Loading required package: lattice
Error in value[[3L]](cond) : object 'server' not found
Calls: local ... tryCatch -> tryCatchList -> tryCatchOne -> <Anonymous>
Execution halted
```

The second day, another error popped up on the webpage:

```
Preparing to deploy application...DONE
Uploading bundle for application: 2597979...DONE
Deploying bundle: 3418130 for application: 2597979 ...
Preparing to deploy application...DONE
Uploading bundle for application: 2597979...DONE
Deploying bundle: 3418175 for application: 2597979 ...
Waiting for task: 764877743
  building: Processing bundle: 3418175
  building: Parsing manifest
  building: Building image: 3826520
  building: Installing system dependencies
  building: Fetching packages
  building: Installing packages
  building: Installing files
  building: Pushing image: 3826520
  deploying: Starting instances
  rollforward: Activating new instances
  terminating: Stopping old instances
An error has occurred!
```

How could an app.R that run absolutely fine in RStudio/Windows 10 end up having a run time error like this after deployed on a Linux server? Could it be the shiny.io Linux server issue? Should we add more diagnostic code to debug this error? Can we look into the /var/log? Do we have the privilege to look into the server?

Lesson Learned 2: RStudio/Windows 10 and shiny.io/Linux are two different environments

This took another day worth of work to figure out what have gone wrong by incrementally growing and uploading the server.R piece from nothing to where the shiny.io server encounter the errors. It turned out that the RStudio is a too forgiving platform allowing the server.R to use the leaflet library without declaring it and also see a global data frame (sandiego) the app.R has. In other words, the RStudio on Windows implemented some scope rules deviating from the R language specification.

Moreover, the layout of the webpage in Shiny in RStudio looked nicer than the webpage on the shiny.io/Linux server, in which the latter could misalign or misplace objects/widgets. Of course, this requires another tuning of the layout design and implementation by trial and error. The following are examples. On RStudio, the shiny GUI looked pretty acceptable:

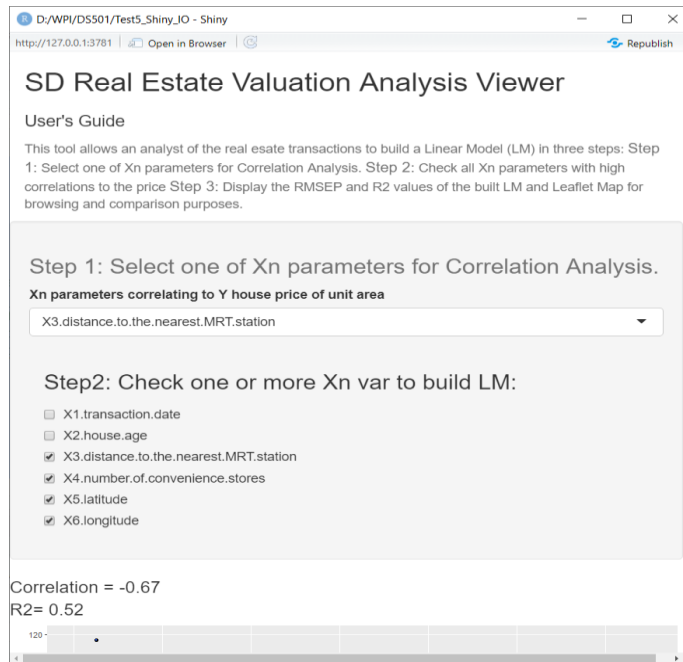


Fig. 7. Shiny generated layout well on RStudio/Windows10

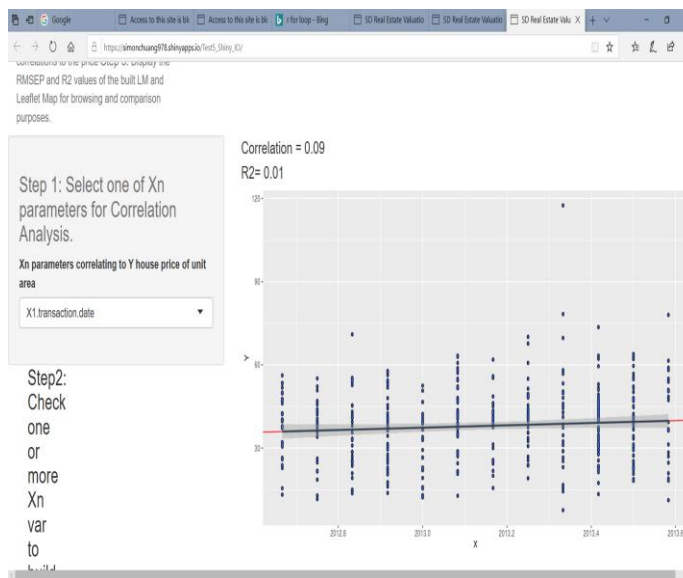


Fig. 8 Prototype layout shiny/Linux did not do as well on the left

This disparity brings back the memory of how difficult the X-windowing system was to code in the early 90's. Shiny made the design and implementation process easier. However, the intuitiveness and friendliness are forever moving targets. This is especially true in real-time programming, on which 10's of million dollars are often spent in GUI design and implementation per recent experience. A layout editor and GUI call-back (reactive) testing tool like JavaBeans would be very helpful. After debugging without much Leaflet documentation, the GUI text layout looks very fine now with color and tiles all show up:

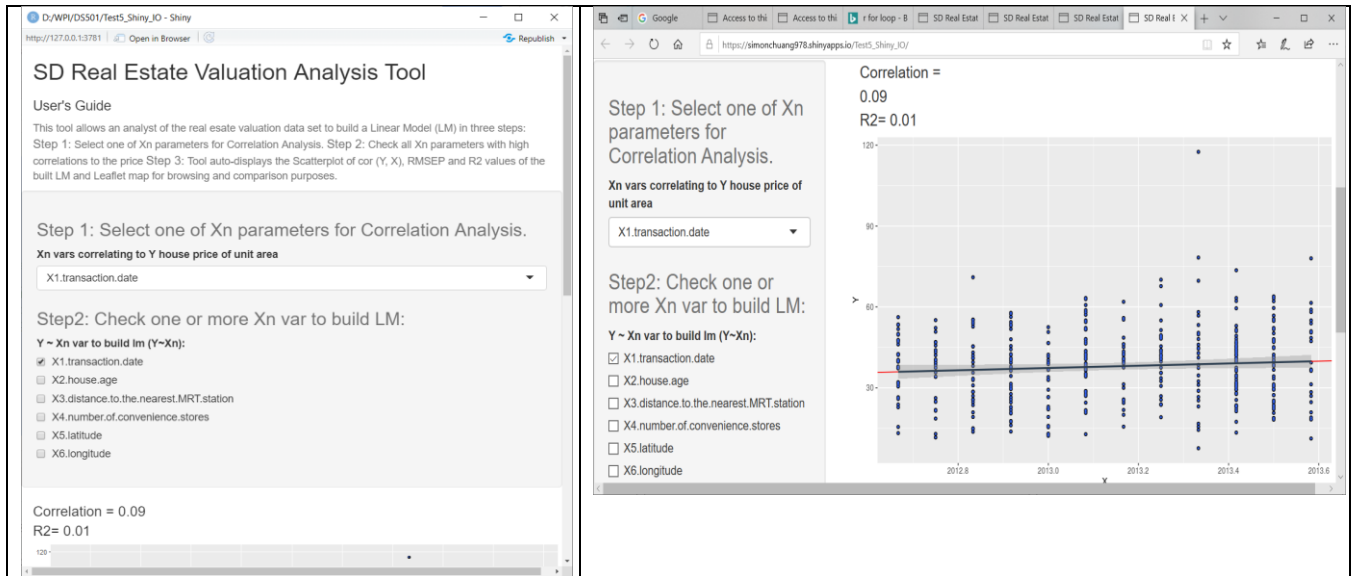


Fig. 8.1 SD Real Estate Valuation Tool layout aligned.

6. What do you need to submit

Submit the link of your R shiny application from <https://www.shinyapps.io> on Canvas Course Webpage.

(Submitted) It is https://simonchuang978.shinyapps.io/Test5_Shiny_IO/

- **What data you collected?** The data collected was from the UCI website, as described in the answer to Problem 1.

Selected <http://archive.ics.uci.edu/ml/index-php.html>

Real Estate Valuation Data Set. Match the project requirements that it is a small data set of 414 rows with the following columns as multivariate data set:

Used Real Estate Valuation Data Set. Match the project requirements that it is a small data set of 414 rows with the following columns as multivariate data set. The following are the columns:

Variable	Description as Part of the Variable Name
Row number	Not a variable
X1	X1.Transaction.date
X2	X2.house.age
X3	X3.distance to the nearest MRT station
X4	number of convenience stores
X5	Latitude
X6	Longitude
Y	house price of unit area

Table. 1 Real Estate Valuation Data

- Why is this topic interesting to you?** This is Real Estate Valuation Data Set allows us to learn, understand, and have a taste of the billion dollars revenue/business created and technologies used by Zillow, Red Fin, and the like in the past 10 years. Also, the data set is sufficient enough to use for a course project. Also with this data set, it enhances our understanding of the advanced R programming techniques and reminds us the Data Science Product Life Cycle activities in future business projects. For example, PCA in Data Planning / Modeling for Large Software System Quality Analysis, Prediction, and Improvement. In other words, they take time with certain risks to be determined.
Details of the reasons to this question can also be found from answers to Problem 1, 2, 3, and 4.

- How did you analyze the data?**
Before the Shiny based tool was created, raw R scripts and methodologies taught in the course were used to analyze the correlations for each X_i and its contribution to Y. The end results was, X_3 , X_4 , X_5 , and X_6 , contribute more than X_1 , and X_2 in the Linear Model. Please see Fig. 7 for some example of the correlation analysis and their scatterplots. Please also see Table 1 in the above. A sample of the raw R script was provided in the answer to Problem 4 and in the following as well.

After this data analysis and data planning phase, the prototyped data model for Y. house.price.per.unit.area has the following characteristics and can be visualized:

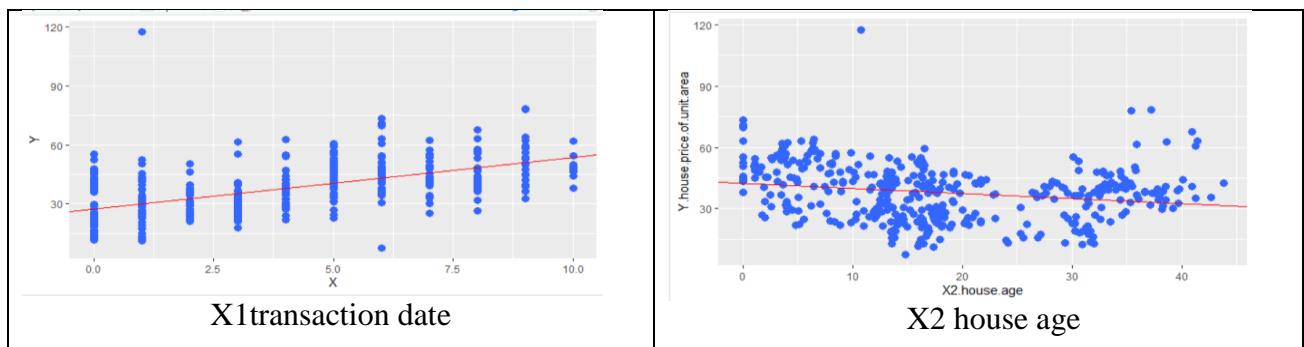


Fig 9.1 X_i variables with low correlation to Y house price per unit area

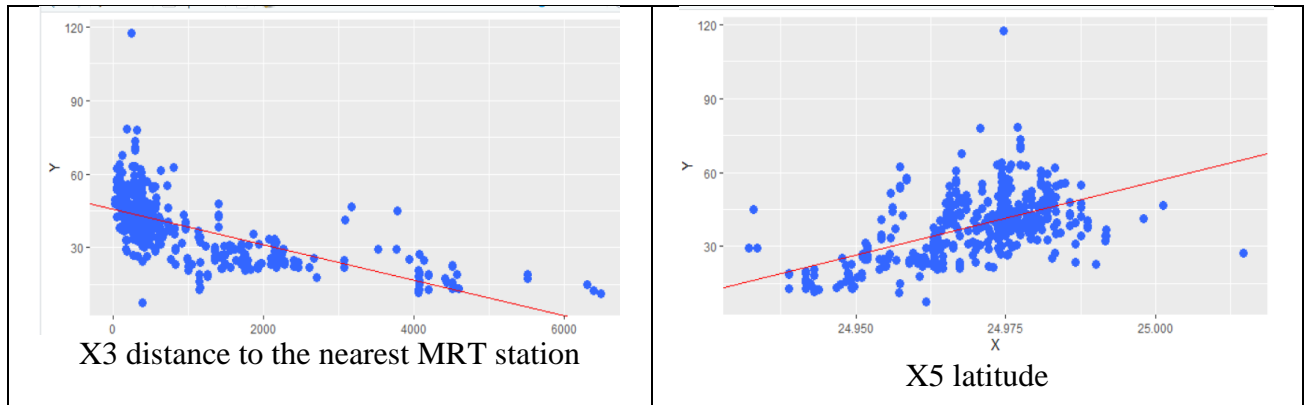


Fig 9.2 Xi variables with higher correlation to Y house price per unit area

Prior to establishing the Shiny based model, quite a few R scripts were used. Thus, from the shiny GUI, the user (data analyst) can also select the following for the final model that has the best R2 value after all experiments. Note that after taking checkGroup input, the R script has to compose the $Y \sim X$ formula and invoke the `lm()`. This has been discussed in Item I Problem 4.

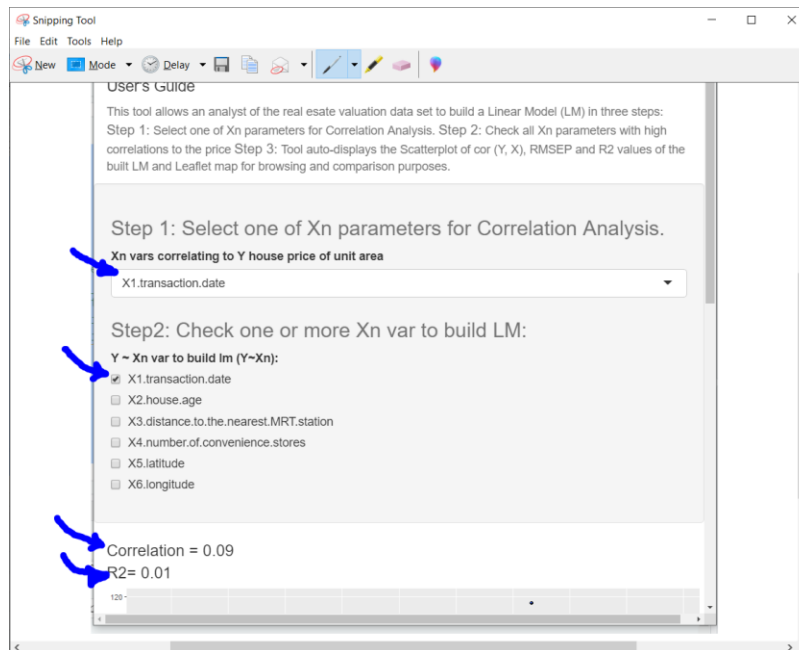


Fig. 10 The shiny GUI to check Xi vars to build the model

The following is one of the prior scripts before the shiny GUI was used to analyze the linear model:

```
housing_model <- lm(Y.house.price.of.unit.area ~
  X3.distance.to.the.nearest.MRT.station +
  X4.number.of.convenience.stores +
  X5.latitude +
  X6.longitude, data=trainHousing)

model_coef <- as.list(coef(housing_model))
housing_predictor <- function(x3, x4, x5, x6) {
  model_coef[["X3.distance.to.the.nearest.MRT.station"]] * x3 +
```



```

model_coef[["X4.number.of.convenience.stores"]] * x4 +
model_coef[["X5.latitude"]] * x5 +
model_coef[["X6.longitude"]] * x6
}

```

To prepare for Training and Testing, the following code is used:

```

#split housing_data as Training and Test sets
set.seed(2012)
splitHousing <- caret::createDataPartition (housing_data[,1], p = 0.8, list=F, times=1)
# Training data set
trainHousing <- housing_data[splitHousing, ]
head(trainHousing)

# Test data set
testHousing <- housing_data[!row.names(housing_data) %in% row.names(trainHousing),]
testHousing <- testHousing[-splitHousing,]

```

The following is a sample code to calculate the RMSEP and R2 value:

```

#install.packages("Metrics")
#library(Metrics)
# RMSEP Root - Mean Squared Error of Prediction
RMSEP = sqrt(PRESS/nrow(predPrice))
#RMSEP

# SST Sum of Squares of difference between individual mean
# Note: Word could make normal quote, apostrophe, and minus to `', '\', - (dash)
#
m = mean(predPrice$Reference)
predPrice[, 'Mean'] = m
predPrice[, 'St'] = 0
predPrice$St = predPrice$Reference - m
SST = sum((predPrice$St)^2)
SST
# R2
R2 = 1- (PRESS/SST)
#R2

```

• What did you find in the data?

There are many things found in the data. They are briefly discussed in the following bullets:

- a. This data set may have a quantity issue. Real Estate Valuation data set is a small data set of 414 transactions. I believe this data set is real. It records transactions happened during 2012-2013 in a small geographical region. To establish a linear data model, we might need more data in the same region, inflation adjusted. Practically speaking, it is a very good data set for Data Science projects such as the DS501 to realize the how important it is in the front phase to Collect Data Examine the Data, Perform Simple Analysis and Collect More Data and More Analysis as we have learned and discussed in answers to the Problem 3 of this case study.
- b. Principal Component wise, the data set needs many more components. The data set includes 6 key variables, listed in Table 1; they seem to be far from enough. For example, education quality in this town, township, safety, crime rate, average household income, lot size, location, education /degree of the household, house architecture style, professions of the household, and many more. Individual differences that can affect the price of the house also exist. For instance, architect who designed the house, material used, social status of the previous owner such as

celebrities, antique preservation status of the house. All these factors can be considered in the valuation and data model. Answer 4.A to Problem 4 discussed how large the industry is.

The attached Zillow Zestimate methodology implied that it is a very complex model used with large quantity of data across the country:

<https://www.zillow.com/research/zestimate-forecast-methodology/>

- c. After learning that the correlation and RMSE/R2 values are not so high as to 0.95. we can imagine that other models might have a better fit for predicting the price than the Linear Model, i.e., models can optimize the error and variance in their predictions. For example, Lasso Regression model or other models that can classify the data set and apply more than one distributions to the data model and back propagate the error info to optimize classification and predication in their multi-layered architecture. However, these will be the subjects in future in the Machine Learning and Deep Machine Learning courses.
- d. In the data set, the latitude and longitude seem not matching the leaflet and google map well, the longitude has to be negative. Therefore, the lng and lat coordinates are translated to San Diego, CA to fit in to the US map. Otherwise, all the houses would be in West Africa. Regardless, the original data characteristics are preserved after translation.
- e. The transaction date and house age are found to have very little information. Both the scatterplot of data pattern and the correlation reflect this fact. These are discussed in greater details in answers to Problem 4.
- f. The distance to the nearest MRT station, number of convenience stores, latitude and longitude all have pretty good correlation. Therefore, the final Linear model includes these 4 variables. (Also discussed in answers to Problem 4)
- g. The zero intercept issue of Linear Model was learned and described in Problem 4 Lesson learned 1. It was a headscratcher when seeing the predicted price of the house being negative. However, it was a very good witness of anomalies such as overfitting and under fitting, which open the doors to various research issues/resolutions meant to optimize the models and deal with situations when data sparsity exists.
- h. Otherwise, this data set is practical and good for a graduate level Data Science project, with which there is a lot learned and documented in answers to Problem 1, 2, 3, and 4, especially Lessons Learned.

7. Email the code (including data, take a small dataset.) Please compress all files in a zipped file. Send an email to ndingari@wpi.edu with the subject: "DS501 Case Study 3"

The shiny.io link provided https://simonchuang978.shinyapps.io/Test5_Shiny_IO/, the report (this document), and the source code + data set will be submitted on 7/25 too.
/ysc