Report

Exercise 1:

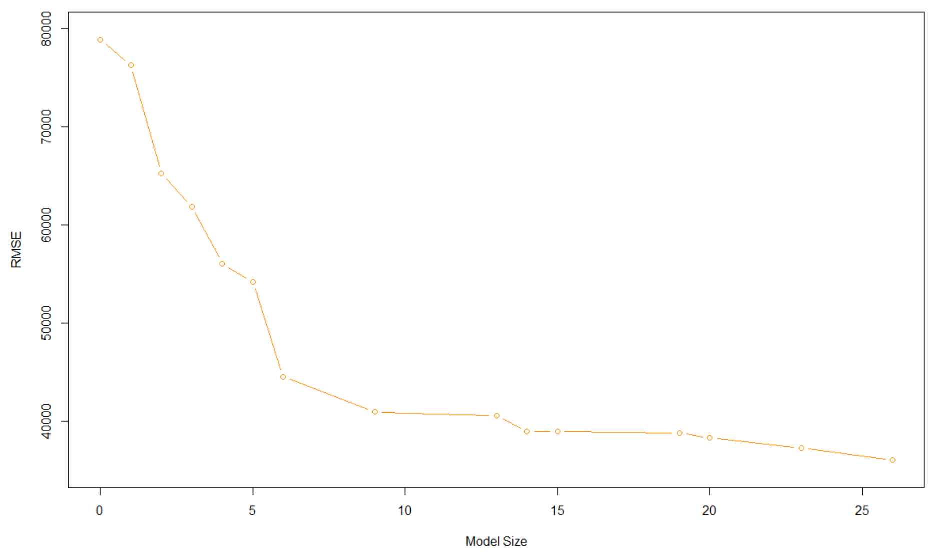
1. Use the following R code to load the Ames data, and drop the variables OverallCond and OverallQual.

```
#load the data file
rawdata=read.csv("ames.csv",header=TRUE)
#Drop the variables OverallCond and OverallQual
rawdata$OverallCond<-NULL
rawdata$OverallQual<-NULL
```

2. Using forward selection, create a series of models up to length 15. The fit_0 model is the model with no predictors. The R code of created 15 models and their related chosen variables are shown as below:

```
# a series of models up to complexity length 15
fit_0 = lm(SalePrice ~ 1, data = data2)
fit_1 = lm(SalePrice ~ LotArea, data=data2)
fit_2 = lm(SalePrice ~ LotArea + YearRemodAdd, data=data2)
fit_3 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1, data=data2)
fit_4 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF, data=data2)
fit_5 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF, data=data2)
fit_6 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF +
             X2ndFlrSF, data=data2)
fit_7 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
             KitchenQual, data=data2)
fit_8 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
             KitchenQual +GarageQual, data=data2)
fit_9 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
             KitchenQual +GarageQual + YearBuilt, data=data2)
fit_10 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
              KitchenQual +GarageQual + YearBuilt +Street, data=data2)
fit_11 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
              KitchenQual +GarageQual + YearBuilt +Street +MSZoning, data=data2)
fit_12 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
              KitchenQual +GarageQual + YearBuilt +Street+MSZoning +GarageArea , data=data2)
fit_13 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
              KitchenQual +GarageQual + YearBuilt +Street+MSZoning +GarageArea +ExterQual, data=data2)
fit_14 = lm(SalePrice ~ LotArea + YearRemodAdd + BsmtFinSF1 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
              KitchenQual +GarageQual + YearBuilt +Street+MSZoning +GarageArea +ExterQual+
              BsmtQual, data=data2)
```

3. The chart plotting the model complexity as the x-axis variable and RMSE as the y-axis variable is shown as below.



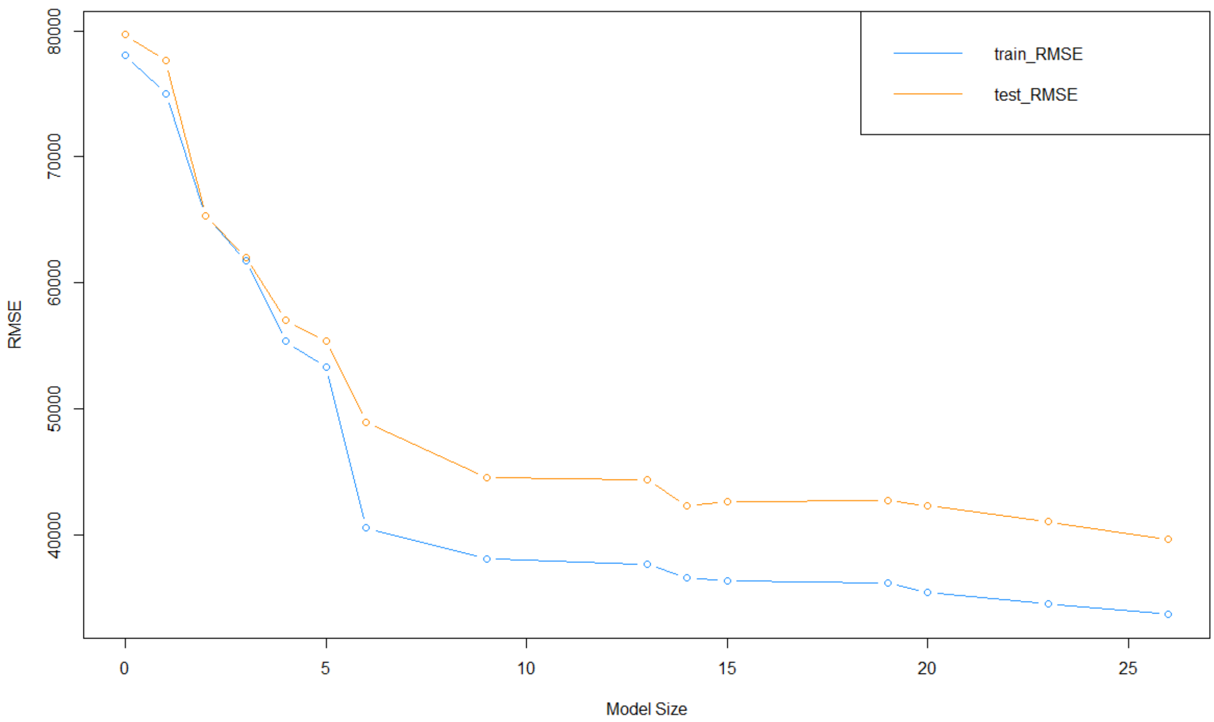
From the chart, it is observed that as model complexity rises, the RMSE of the dataset decreases. Thus, as the number of variables in the linear regression model increases, the prediction accuracy of the model improves. In conclusion, we should use the full-size model that includes all the variables. Because the full-size model has the highest prediction accuracy. The criterion we are using to make this statement is RMSE.

Exercise 2:

1. Use the following code to split the train dataset and test dataset. Use 50% of all the dataset as train dataset and the other 50% as test dataset.

```
set.seed(123)
num_obs = nrow(data2)
#50% of data2 as train dataset, and the other 50% as test dataset
train_index = sample(num_obs, size = trunc(0.50 * num_obs))
train_data = data2[train_index, ]
test_data = data2[-train_index, ]
```

The chart of Train and Test RMSE for the 15 models we fit in Exercise 1 is shown as below:



From the chart, it can be observed that as model complexity rises, the RMSE of the train dataset and test dataset all decrease.

2. The model we use to predict SalePrice is **Random Forest** model. We choose the following 69 variables as regressors.

```
 [1] "MSSubClass"    "MSZoning"      "LotFrontage"   "LotArea"       "Street"        "LotShape"
 [7] "LotConfig"     "Neighborhood"  "Condition1"    "Condition2"    "BldgType"      "HouseStyle"
[13] "YearBuilt"     "YearRemodAdd"  "RoofStyle"     "RoofMatl"      "Exterior1st"   "Exterior2nd"
[19] "MasVnrType"    "MasVnrArea"    "ExterQual"     "ExterCond"     "Foundation"    "BsmtQual"
[25] "BsmtCond"      "BsmtExposure"  "BsmtFinType1"  "BsmtFinSF1"    "BsmtFinType2"  "BsmtFinSF2"
[31] "BsmtUnfSF"     "TotalBsmtSF"   "Heating"       "HeatingQC"     "CentralAir"    "Electrical"
[37] "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath"  "BsmtHalfBath"
[43] "FullBath"      "HalfBath"      "BedroomAbvGr"  "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"
[49] "Functional"    "Fireplaces"    "GarageType"    "GarageYrBlt"   "GarageFinish"  "GarageCars"
[55] "GarageArea"    "GarageQual"    "GarageCond"    "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"
[61] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"      "MiscVal"       "MoSold"
[67] "YrSold"        "SaleType"      "SaleCondition"
```

According to our prediction results, the Train RMSE is 13127.19 and the Test RMSE is 31998.24, as is shown below:

```
> rf_train_rmse
[1] 13127.19

> rf_test_rmse
[1] 31998.24
```

3. The procedure to build the **Random Forest** model is shown as below:
   (1) Data Cleaning:

   Firstly, we drop the variables Id, Alley, LandContour, Utilities, LandSlope, PoolQC, Fence, MiscFeature, FireplaceQu and use the other 69 variables as regressors, as is discussed in problem 2.

   Secondly, we found that there are 18% missing values in the column LotFrontage, so we fill these missing values with 0.

   Thirdly, use na.omit() function to drop the rows that contain missing values. After doing this, there are 1338 rows left. And we name this dataset as data2.

   (2) Train-Test data split

   As is discussed in problem 1, we use 50% of data2 as train dataset and the other 50% as test dataset.

   (3) Build **Random Forest** model

   Use randomForest() function to build the model. According to the results, some parameters of Random Forest model is shown as below:

```
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 23

          Mean of squared residuals: 1034399522
                    % Var explained: 83.03
```

   And two kinds of importance scores of the 69 regressors are shown as below. The higher the score is, the more important the variable is.

	%IncMSE	IncNodePurity
GrLivArea	41.63509815	707867138974
Neighborhood	38.96564125	524472933551
X1stFlrSF	22.38672413	161925912834
TotalBsmtSF	17.60193772	204678103922
ExterQual	16.88226023	270488709774
FullBath	16.63740045	129226400529
GarageCars	14.46461917	578884350910
GarageArea	13.89242932	223322898277
KitchenQual	13.53909443	89474810663
YearBuilt	13.28958134	93073011291
BsmtFinSF1	11.58756907	70306125938
LotArea	10.93328613	75260611773
X2ndFlrSF	10.78985917	137165086707
GarageType	10.51974988	18619199669
YearRemodAdd	10.50333764	30888258418
BsmtFinType1	9.93105445	17620743946
MSZoning	9.89680155	7994949357
BsmtQual	9.65793723	89623738226
Fireplaces	9.18614212	19562288938
MasVnrArea	8.39810276	136431328667
GarageYrBlt	8.34854478	30928448786
HouseStyle	8.33567725	10154806989
MSSubClass	8.28245423	8740988333
OpenPorchSF	8.21216415	21386849157
GarageFinish	7.39885511	15792473084
BsmtUnfSF	7.19049491	22205811454
Exterior2nd	7.01399647	51624987344
Foundation	6.84616666	6707116881
HalfBath	6.58432145	5961383094
BsmtFullBath	6.48852499	6439097573
BedroomAbvGr	5.82073732	10407769936
BldgType	5.43368545	3505099586
BsmtExposure	5.36124517	14804098464
Exterior1st	5.31161516	43665319592
TotRmsAbvGrd	5.21440715	34473524055
HeatingQC	5.03050032	8359422461
CentralAir	4.07708524	3663757410
WoodDeckSF	3.96013963	15439328117
ScreenPorch	3.09798741	4726763767
RoofStyle	3.07068364	9278109571
MasVnrType	2.61681634	6630322115
BsmtFinSF2	2.52548514	2997954594
KitchenAbvGr	2.47203972	1149035560
SaleCondition	2.39887616	7249080934
Functional	2.26598562	3667722557
SaleType	2.02189162	3923647166
BsmtCond	1.82246827	2146451346
LotFrontage	1.78538240	16944289413
EnclosedPorch	1.45333704	3392179877
LotConfig	1.25767666	12700781647
Heating	1.25052494	976561971
YrSold	1.14669331	6466151554
PavedDrive	0.92307804	764312215
MiscVal	0.86960683	268091097
BsmtHalfBath	0.70634127	7242289193
GarageCond	0.66330197	703041617
ExterCond	0.59314937	2758468097
GarageQual	0.58809113	4030307485
Condition1	0.58446233	5336238774
RoofMatl	0.05018633	11097139607
Street	0.00000000	109367386
PoolArea	0.00000000	42922887
LotShape	-0.31010169	10144405756
X3SsnPorch	-0.50070934	104352716
LowQualFinSF	-0.55788930	2618222454
Electrical	-0.72318737	914373677
MoSold	-0.95372896	15403087913
BsmtFinType2	-1.39112732	2476316330
Condition2	-2.44126746	1341054044

(4) Predict and calculate the Train and Test RMSE

After building the Random Forest model, use the following code to predict the SalePrice on the train dataset and test dataset. And calculate the Train and Test RMSE.

```
pred_rf=predict(rf_fit,newdata = test_data)
pred_rf_train=predict(rf_fit,newdata = train_data)

rf_train_rmse=rmse(train_data$SalePrice,pred_rf_train)
rf_train_rmse

rf_test_rmse=rmse(test_data$SalePrice,pred_rf)
rf_test_rmse
```

As is discussed in Problem 2, the Train RMSE is 13127.19 and the Test RMSE is 31998.24, as is shown below:

```
> rf_train_rmse
[1] 13127.19

> rf_test_rmse
[1] 31998.24
```

The advantages of the Random Forest model:

A random forest is an ensemble learning approach to supervised learning. Multiple predictive models are developed, and the results are aggregated to improve classification rates.

The algorithm for a random forest involves sampling cases and variables to create a large number of decision trees. Each case is classified by each decision tree. The most common classification for that case is then used as the outcome.

Random forests tend to be very accurate compared with other classification methods. Additionally, they can handle large problems (many observations and variables), can handle large amounts of missing data in the training set, and can handle cases in which the number of variables is much greater than the number of observations. The provision of OOB error rates and measures of variable importance are also significant advantages.