Lab 6: House Price Prediction

- Public domain data set from kaggle.com
  - Sales of houses in the Seattle area from May, 2014 - May, 2015.
- Inputs to the models you will develop
  - id#, prop_id – identify each property – *not useful inputs*
  - number of bedrooms
  - number of bathrooms
  - size in square feet
  - lot size in square feet
  - number of floors
  - condition – small integer value
  - location – zip code; latitude and longitude
  - etc.
- Output – predicted price

Example house_data.csv

```
house_data <- read.csv("house_data.csv")
head(house_data)
```

	id	date	price	bedrooms	bathrooms	sqft_living	…
1	7129300520	20141013T000000	221900	3	1.00	1180	…
2	6414100192	20141209T000000	538000	3	2.25	2570	…
3	5631500400	20150225T000000	180000	2	1.00	770	…
4	2487200875	20141209T000000	604000	4	3.00	1960	…
5	1954400510	20150218T000000	510000	3	2.00	1680	…
6	7237550310	20140512T000000	1230000	4	4.50	5420	

…

Goal: Predict House Prices

- Divide data into training and testing sets
- Training set
  - All inputs plus price
- Testing set
  - Predict the price using the model you developed with the training set
- Compute error between your predicted price and the actual priced provided in data set.
  - Measure of overall error = RMSE

RMSE - Root Mean Square Error

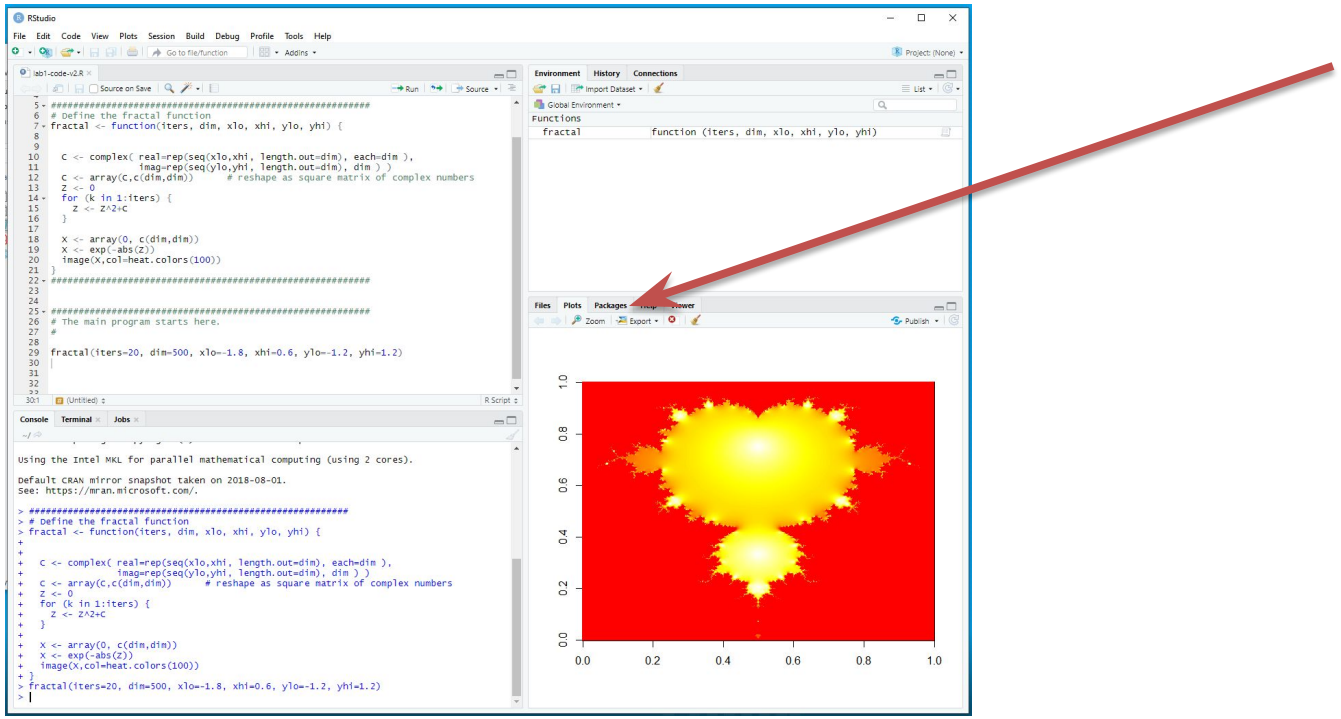
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_{predicted} - y_{actual})^2}{n}}$$

Models to Develop

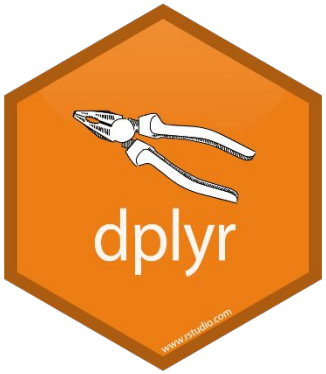
- Backward elimination using all data
- Segment data by zip code
  - Train and test models for each zip code using specified predictors

Packages in R

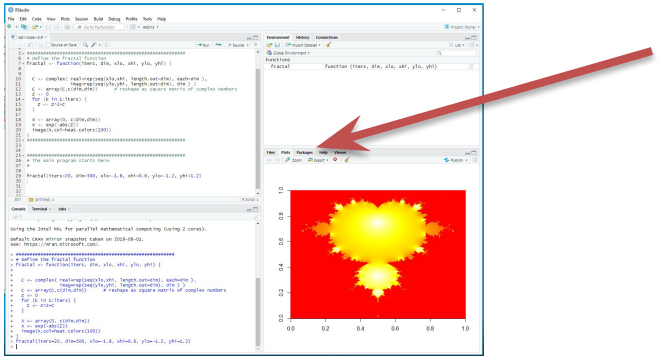
- Package = library of functions developed by someone else for you to use
- In RStudio
  - Window in lower right quadrant
  - “Packages” tab □ “install” tab □ type “package name” in box



dplyr package



- dplyr package
  - Set of functions to easily access and change dataframes
  - group, filter, select, summarize, ….
  - Much more functionality than we will be using
  - We will focus on segmentation and summarizing
  - “Packages” tab □ “install” tab □ type “dplyr” in box



head(house_data)

house_data <- read.csv("house_data.csv")

	id	date	price	bedrooms	bathrooms	sqft_living	…
1	7129300520	20141013T000000	221900	3	1.00	1180	…
2	6414100192	20141209T000000	538000	3	2.25	2570	…
3	5631500400	20150225T000000	180000	2	1.00	770	…
4	2487200875	20141209T000000	604000	4	3.00	1960	…
5	1954400510	20150218T000000	510000	3	2.00	1680	…
6	7237550310	20140512T000000	1230000	4	4.50	5420	

…

Segmentation by zipcode

```
house_data <- read.csv("house_data.csv")
data_by_zipcode <- house_data %>%
  group_by(zipcode) %>%
  summarize(
    count = n(),
    med_price = median(price),
    med_yr_built = median(yr_built),
)
```

Pipeline operation %>%

New columns that we are generating using summarize()

Data flow of the pipeline operations:

house_data □ group_by □ summarize □ data_by_zipcode

head(data_by_zipcode)

Zipcode	count	med_price	med_yr_built
98001	362	260000.0	1981
98002	199	235000.0	1966
98003	280	267475.0	1975
98004	317	1150000.0	1965
98005	168	765475.0	1967
98006	498	760184.5	1978

Adding Your Predictions

```
house_data <- read.csv("house_data.csv")
data_by_zipcode <- house_data %>%
  group_by(zipcode) %>%
  summarize(
    count = n(),
    med_price = median(price),
    med_yr_built = median(yr_built),
    error = price_prediction_error(price, bedrooms,
sqft_living, …..)
)
```

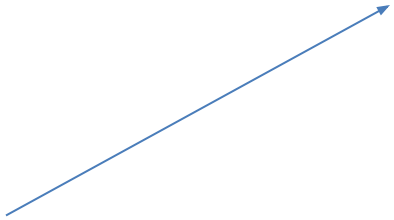
Pipeline operation %>%

Column names from house_data

Within your function, divide into training and testing sets and return rmse of your predictions.

head(data_by_zipcode)

Zipcode	count	med_price	med_yr_built	error
98001	362	260000.0	1981	10238.2
98002	199	235000.0	1966	9896.7
98003	280	267475.0	1975	4524.4
98004	317	1150000.0	1965	….
98005	168	765475.0	1967	….
98006	498	760184.5	1978	….



Your computed rmse values for each zip code.

Your price prediction function

```
price_prediction_error <- function(price, bedrooms, bathroom, sqft_living, sqft_lot, grade,
yr_built) {

    # Create a new data frame for the variables to be used in the price prediction
     house_info <- data.frame(price, bedrooms, bathroom, sqft_living, sqft_lot, grade, yr_built)
     # Separate the data into training and testing sets
        This is the same as last lab, but using house_info
     # Compute the linear model
        This is the same as last lab, but using house_info, and the parameters above
     # Use the model to predict the prices
        This is the same as last lab, but using house_info
     # Compute the rmse of the predicted - actual values
     rmse <- compute rmse of (predicted – actual)
     return(rmse)
}
```

To do

- Download and complete Lab 6