Intro

This tutorial is a good first step for someone looking to learn the steps needed for exploring data, cleaning data, and training/evaluating some basic machine learning algorithms.

Resources:

Main

Additional 1

Additional 2

(***Advanced)

Step 1: Load in the data.

```
library(tidyverse)
library(reshape2)
housing = read.csv('../input/housing.csv')
head(housing)
summary(housing)
```

Output Picture 1

```
par(mfrow=c(2,5))
colnames(housing)
ggplot(data = melt(housing), mapping = aes(x =
value)) +
geom_histogram(bins = 30) + facet_wrap(~variable,
scales = 'free_x')
```

Output Picture 2

(***Advanced)

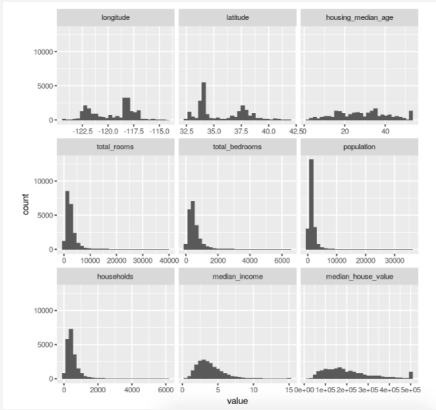
Output Picture 1

```
   longitude         latitude     housing_median_age  total_rooms
 Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
 Median :-118.5   Median :34.26   Median :29.00      Median : 2127
 Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
 3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
 Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320

 total_bedrooms     population      households     median_income
 Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
 1st Qu.: 296.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
 Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
 Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
 3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
 Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
 NA's   :207
 median_house_value   ocean_proximity
 Min.   : 14999     <1H OCEAN :9136
 1st Qu.:119600     INLAND    :6551
 Median :179700     ISLAND    :   5
 Mean   :206856     NEAR BAY  :2290
 3rd Qu.:264725     NEAR OCEAN:2658
 Max.   :500001
```

(***Advanced)

Output Picture 2



(***Advanced)

Step 2: Clean the data

Impute missing values

```
housing$total_bedrooms[
is.na(housing$total_bedrooms)] =
median(housing$total_bedrooms , na.rm = TRUE)
```

Fix the total columns - make them means

```
housing$mean_bedrooms =
housing$total_bedrooms/housing$households
housing$mean_rooms =
housing$total_rooms/housing$households
drops = c('total_bedrooms', 'total_rooms')
housing = housing[ , !(names(housing) %in% drops)]
```

Turn categoricals into booleans

```
categories = unique(housing$ocean_proximity)
#split the categories off
cat_housing = data.frame(ocean_proximity =
housing$ocean_proximity)
for(cat in categories){
cat_housing[,cat] = rep(0, times=
nrow(cat_housing))
}
for(i in 1:length(cat_housing$ocean_proximity)){
cat = as.character(cat_housing$ocean_proximity[i])
cat_housing[,cat][i] = 1
}
cat_columns = names(cat_housing)
keep_columns = cat_columns[cat_columns !=
'ocean_proximity']
cat_housing =
select(cat_housing,one_of(keep_columns))
```

Step 2: Clean the data (cont)

Scale the numerical variables

```
drops = c('ocean_proximity','median_house_value')
housing_num = housing[ , !(names(housing) %in% drops)]
scaled_housing_num = scale(housing_num)
```

Merge the altered numerical and categorical dataframes

```
cleaned_housing = cbind(cat_housing, scaled_housing_num, median_house_value=housing$median_house_value)
head(cleaned_housing)
```

Output Picture 3

Output Picture 4

(***Advanced)

Output Picture 3

NEAR BAY	<1H OCEAN	INLAND	NEAR OCEAN	ISLAND	longitude	latitude	housing_median_age	population
1	0	0	0	0	-1.327803	1.052523	0.9821189	-0.9744050
1	0	0	0	0	-1.322812	1.043159	-0.6070042	0.8614180
1	0	0	0	0	-1.332794	1.038478	1.8561366	-0.8207575
1	0	0	0	0	-1.337785	1.038478	1.8561366	-0.7660095
1	0	0	0	0	-1.337785	1.038478	1.8561366	-0.7598283
1	0	0	0	0	-1.337785	1.038478	1.8561366	-0.8940491

(***Advanced)

Output Picture 4

households	median_income	mean_bedrooms	mean_rooms	median_house_value
-0.9770092	2.34470896	-0.148510661	0.6285442	452600
1.6699206	2.33218146	-0.248535936	0.3270334	358500
-0.8436165	1.78265622	-0.052900657	1.1555925	352100
-0.7337637	0.93294491	-0.053646030	0.1569623	341300
-0.6291419	-0.01288068	-0.038194658	0.3447024	342200
-0.8017678	0.08744452	0.005232996	-0.2697231	269700

(***Advanced)

Step 3: Create a test set of data

```
set.seed(1738)
sample = sample.int(n = nrow(cleaned_housing), size = floor(.8*nrow(cleaned_housing)), replace = F)
train = cleaned_housing[sample, ] #just the samples
test = cleaned_housing[-sample, ] #everything but the samples
nrow(train) + nrow(test) == nrow(cleaned_housing)
```

TRUE

(***Advanced)

Step 4: Test some predictive models.

```
library('boot')
?cv.glm
glm_house = glm(median_house_value~median_income+ mean_rooms+ population, data= cleaned_housing)
k_fold_cv_error = cv.glm(cleaned_housing , glm_house, K=5)
k_fold_cv_error$delta
```

6946162248.89155
6942675168.18876

```
glm_cv_rmse = sqrt(k_fold_cv_error$delta)[1]
glm_cv_rmse
```

83343.6395227107

```
glm_house$coefficients
```

Output Picture 5

Random forest model

```
library('randomForest')
?randomForest
set.seed(1738)
train_y = train[,'median_house_value']
train_x = train[, names(train) != 'median_house_value']
rf_model = randomForest(train_x, y = train_y , ntree = 500, importance = TRUE)
rf_model$importance
```

Output Picture 6

The out-of-bag (oob) error estimate

```
oob_prediction = predict(rf_model)
#leaving out a data source forces OOB predictions
train_mse = mean(as.numeric ((oob_prediction - train_y)^2))
oob_rmse = sqrt(train_mse)
oob_rmse
```

48976.2521584537

```
test_y = test[,'median_house_value']
test_x = test[, names(test) !='median_house_value']
y_pred = predict(rf_model , test_x)
test_mse = mean(((y_pred - test_y)^2))
test_rmse = sqrt(test_mse)
```

Step 4: Test some predictive models. (cont)

```
test_rmse
```

48354.9021429439

(***Advanced)

Output Picture 5

(Intercept)	206855.816908914
median_income	82608.9593842245
mean_rooms	-9755.44247542691
population	-3948.29342917358

(***Advanced)

Output Picture 6

	%IncMSE	IncNodePurity
NEAR BAY	471570753	1.386134e+12
<1H OCEAN	1589525273	3.934732e+12
INLAND	4041972795	3.010652e+13
NEAR OCEAN	505892227	2.133809e+12
ISLAND	1608344	6.900350e+10
longitude	6800888175	2.573286e+13
latitude	5404868569	2.224088e+13
housing_median_age	1101454757	9.896973e+12
population	1049908169	7.480374e+12
households	1156877777	7.922140e+12
median_income	8288648272	7.354195e+13
mean_bedrooms	462307992	7.767073e+12
mean_rooms	1788332497	2.056660e+13

(***Advanced)