Data

```
pacman::p_load(dplyr, magrittr, data.table)
x =
read.csv("C:\\Users\\jaime\\OneDrive\\Desktop\\housing_data_2016_2017.csv",
header = TRUE)
xhat = x %>%
  select(coop_condo, full_address_or_zip_code, approx_year_built,
common_charges, maintenance_cost, parking_charges,
listing_price_to_nearest_1000, total_taxes, date_of_sale, fuel_type,
garage_exists, kitchen_type, num_total_rooms, num_bedrooms,
num_floors_in_building, sale_price)
```

Regression Tree Modeling

```
pacman::p_load(rpart)
rpart(sale_price~coop_condo, data = xhat)

## n=528 (1702 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 528 517 $155,000  (0.0019 0.0019 0.0019 0.0019 0.0038 0.0019
0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019 0.0095 0.0019 0.0038
0.0076 0.0019 0.0019 0.0019 0.0019 0.0019 0.013 0.0038 0.0057 0.0019 0.0038
0.0038 0.0019 0.0019 0.0019 0.0019 0.0038 0.0038 0.0019 0.011 0.0038 0.0019
0.0095 0.0019 0.0019 0.0019 0.0019 0.0019 0.021 0.0019 0.0038 0.0057 0.0038
0.0019 0.0076 0.0038 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019
0.0038 0.0057 0.0095 0.0019 0.0019 0.0038 0.019 0.0019 0.0019 0.0019 0.0019
0.0038 0.0095 0.0019 0.0019 0.0095 0.0019 0.0019 0.0038 0.0057 0.0019 0.0019
0.0019 0.0019 0.0019 0.0019 0.0076 0.0019 0.0038 0.0019 0.0095 0.0038 0.0038
0.0019 0.0019 0.0019 0.0019 0.0038 0.013 0.0038 0.0019 0.0019 0.0076 0.0019
0.0038 0.0057 0.0019 0.0038 0.0019 0.013 0.0019 0.0019 0.0019 0.0019 0.0019
0.0076 0.0019 0.0019 0.0057 0.0019 0.0038 0.0019 0.0019 0.0076 0.0019 0.0019
0.0057 0.0019 0.0019 0.0019 0.0019 0.0057 0.0038 0.0019 0.0076 0.0019 0.0019
0.0019 0.0057 0.0019 0.0019 0.011 0.0019 0.0019 0.0095 0.0019 0.0019 0.0076
0.0019 0.0019 0.0019 0.0019 0.0076 0.0019 0.0019 0.0019 0.0019 0.0038 0.0038
0.0019 0.0038 0.0019 0.0076 0.0057 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019
0.0038 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0076 0.0019 0.0038 0.0019
0.0057 0.0038 0.0019 0.0038 0.0038 0.0076 0.0019 0.0019 0.0019 0.0038 0.0019
0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0038 0.0019 0.0057 0.0057 0.0019
0.0076 0.0038 0.0019 0.0019 0.0057 0.0019 0.0038 0.0019 0.0038 0.0019 0.0038
0.0038 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0038
0.0019 0.0057 0.0019 0.0019 0.0057 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019
0.0019 0.0057 0.0038 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019
0.0019 0.0038 0.0038 0.0019 0.0038 0.0038 0.0038 0.0038 0.0019 0.0019 0.0019
0.0038 0.0019 0.0038 0.0038 0.0019 0.0019 0.0019 0.0019 0.0057 0.0019 0.0019
0.0019 0.0019 0.0019 0.0057 0.0038 0.0038 0.0019 0.0019 0.0019 0.0038 0.0019
0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0038 0.0019
```

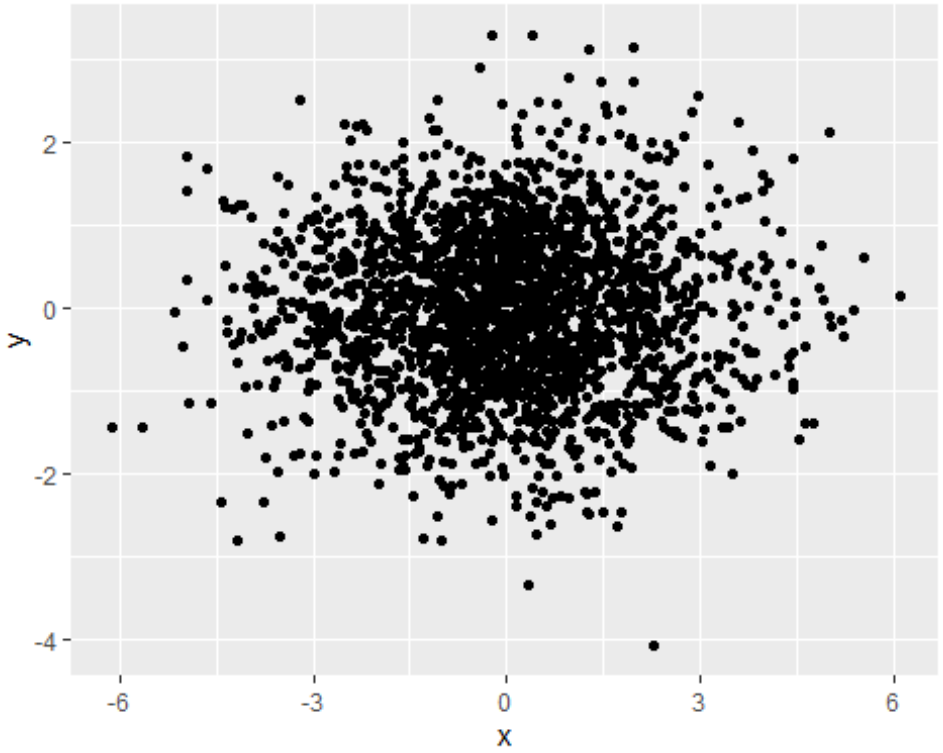
```
0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019
0.0019 0.0019 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019 0.0019
0.0019) *
```

Linear Modeling

```
y = rnorm(xhat$sale_price)
x0 = rnorm(xhat$coop_condo)
x1 = rnorm(xhat$num_total_rooms)
x2 = rnorm(xhat$parking_charges)
data_set = lm(y ~ x0 + x1 + x2)
summary(data_set)

##
## Call:
## lm(formula = y ~ x0 + x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0746 -0.6439  0.0062  0.6375  3.3461
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.034478   0.020185  -1.708   0.0878 .
## x0           0.003352   0.020036   0.167   0.8671
## x1           0.014030   0.019827   0.708   0.4792
## x2          -0.011799   0.020244  -0.583   0.5601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9529 on 2226 degrees of freedom
## Multiple R-squared:  0.0003774,  Adjusted R-squared:  -0.0009698
## F-statistic: 0.2801 on 3 and 2226 DF,  p-value: 0.8398

pacman::p_load(ggplot2)
ggplot(data.frame(x = x0 + x1 + x2, y = y)) +
  geom_point(aes(x = x, y = y))
```



Random Forest

```
pacman::p_load(randomForest)

n = 500
sigma = 0.5
x_min = min(y)
x_max = max(y)
f_x = function(x){sin(x)}
y_x = function(x, sigma){f_x(x) + rnorm(n, 0, sigma)}
x_train = runif(n, x_min, x_max)
y_train = y_x(x_train, sigma)
x_test = runif(n, x_min, x_max)
y_test = y_x(x_test, sigma)
node_sizes = 1:500

se_by_node_sizes = array(NA, length(node_sizes))
for (i in 1:length(node_sizes)) {
  rf_mod = randomForest(x = data.frame(x = x_train), y = y_train, ntree = 1,
replace = FALSE, sampsize = n, nodesize = node_sizes[i])
  y_hat_test = predict(rf_mod,data.frame(x = x_test))
  se_by_node_sizes[i] = sd(y_test - y_hat_test)
}
```

Finding R^2 and RMSE

```
  # x0 = coop_condo
  x_bar0 = sum(x0)/n
  y_bar = sum(y)/n
  b_1 = (sum(x0*y)-n*x_bar0*y_bar) / (sum(x0^2)-n*x_bar0^2)
  b_0 = y_bar - b_1*x_bar0
  yhat = b_0 + b_1 *x0
  e0 = y -yhat
  SSE0 = sum(e0^2)
  SST0 = sum((y-y_bar)^2)
  MSE0 = SSE0 /(n-2)
  RMSE0 = sqrt(MSE0)
  Rsq0 = 1 - SSE0 / SST0

  # x1 = num_total_rooms
  x_bar1 = sum(x1)/n
  b_1 = (sum(x1*y)-n*x_bar1*y_bar) / (sum(x1^2)-n*x_bar1^2)
  b_0 = y_bar - b_1*x_bar1
  yhat = b_0 + b_1 *x1
  e1 = y -yhat
  SSE1 = sum(e1^2)
  SST1 = sum((y-y_bar)^2)
  MSE1 = SSE1 /(n-2)
  RMSE1 = sqrt(MSE1)
  Rsq1 = 1 - SSE1 / SST1

  # x2 = parking_charges
  x_bar2 = sum(x2)/n
  b_1 = (sum(x2*y)-n*x_bar2*y_bar) / (sum(x2^2)-n*x_bar2^2)
  b_0 = y_bar - b_1*x_bar2
  yhat = b_0 + b_1 *x2
  e2 = y -yhat
  SSE2 = sum(e2^2)
  SST2 = sum((y-y_bar)^2)
  MSE2 = SSE2 /(n-2)
  RMSE2 = sqrt(MSE2)
  Rsq2 = 1 - SSE2 / SST2
```

oobRsq and oobRMSE

```
nstar = 500
ystar = rnorm(xhat$sale_price)
x0_star = rnorm(xhat$coop_condo)
x1_xtar = rnorm(xhat$num_total_rooms)
x2_xtar = rnorm(xhat$parking_charges)
# x0 = coop_condo
  x_bar0 = sum(x0_star)/n
  y_bar = sum(ystar)/n
  b_1 = (sum(x0_star*ystar)-n*x_bar0*y_bar) / (sum(x0_star^2)-n*x_bar0^2)
  b_0 = y_bar - b_1*x_bar0
  yhat = b_0 + b_1 *x0_star
```

```
oobe0 = ystar -yhat
oobSSE0 = sum(oobe0^2)
oobSST0 = sum((ystar-y_bar)^2)
oobMSE0 = oobSSE0 /(n-2)
oobRMSE0 = sqrt(oobMSE0)
oobRsq0 = 1 - oobSSE0 / oobSST0

# x1 = num_total_rooms
x_bar1 = sum(x1_xtar)/n
b_1 = (sum(x1_xtar*ystar)-n*x_bar1*y_bar) / (sum(x1_xtar^2)-n*x_bar1^2)
b_0 = y_bar - b_1*x_bar1
yhat = b_0 + b_1 *x1_xtar
oobe1 = ystar -yhat
oobSSE1 = sum(oobe1^2)
oobSST1 = sum((ystar-y_bar)^2)
oobMSE1 = oobSSE1 /(n-2)
oobRMSE1 = sqrt(oobMSE1)
oobRsq1 = 1 - oobSSE1 / oobSST1

# x2 = parking_charges
x_bar2 = sum(x2_xtar)/n
b_1 = (sum(x2_xtar*ystar)-n*x_bar2*y_bar) / (sum(x2_xtar^2)-n*x_bar2^2)
b_0 = y_bar - b_1*x_bar2
yhat = b_0 + b_1 *x2_xtar
oobe2 = ystar -yhat
oobSSE2 = sum(oobe2^2)
oobSST2 = sum((ystar-y_bar)^2)
oobMSE2 = oobSSE2 /(n-2)
oobRMSE2 = sqrt(oobMSE2)
oobRsq2 = 1 - oobSSE2 / oobSST2
```