MODEL AGNOSTIC INTERPRETABILITY

Dr. Aric LaBarr

Institute for Advanced Analytics

MODEL PERFORMANCE

Comparison

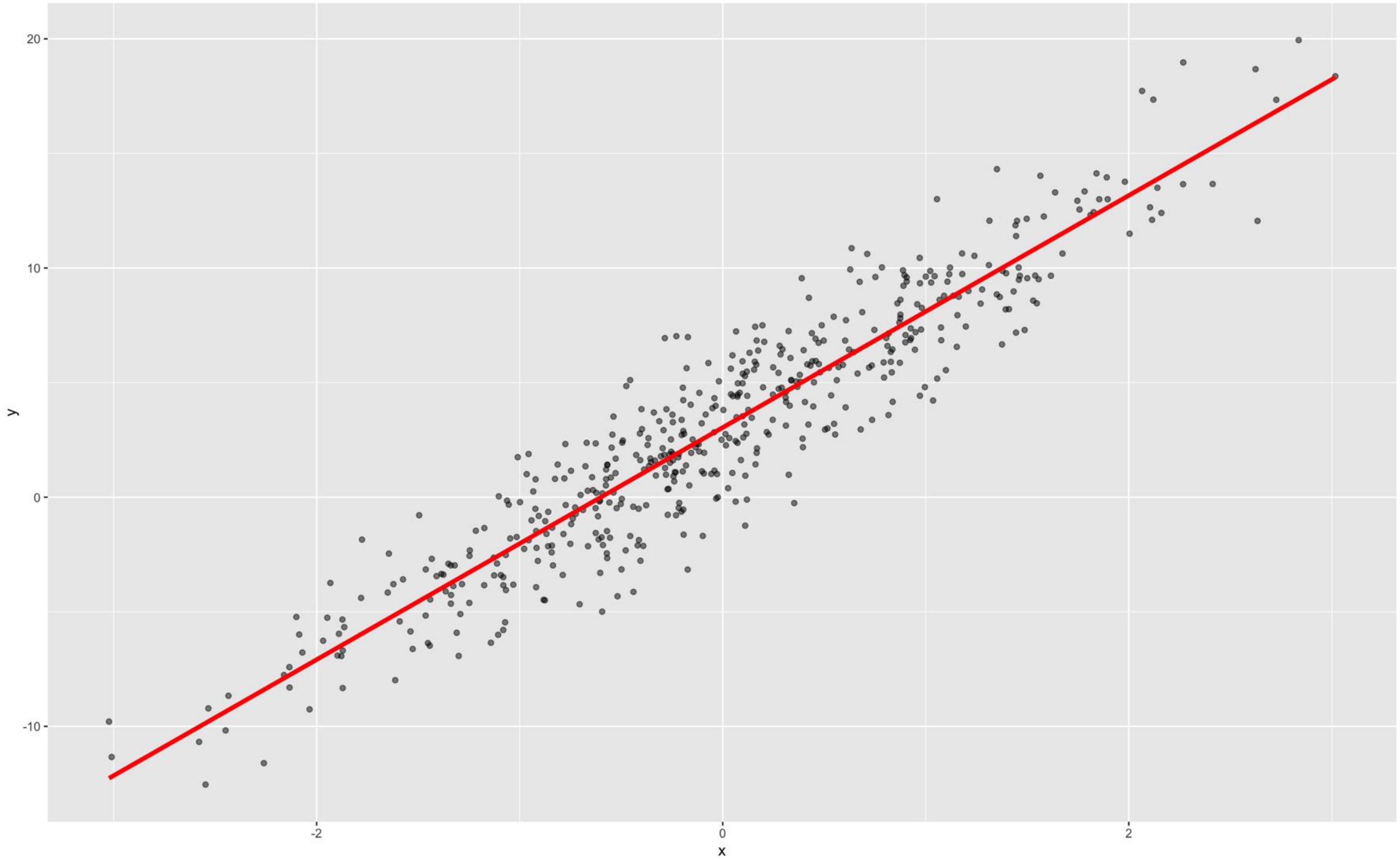
- Let's compare the Ames housing dataset across all these different models on the test dataset we set aside:

Modeling Technique	Mean Absolute Error (MAE)	Mean Absolute % Error (MAPE)
Linear Regression	$27,638	16.2%
Elastic Net	$31,255	18.6%
MARS / EARTH	$23,369	14.1%
GAM	$23,799	14.4%
Random Forest	$19,205	11.6%
XGBoost	$19,576	11.7%
Neural Network	$20,654	12.7%

“INTERPRETABILITY”

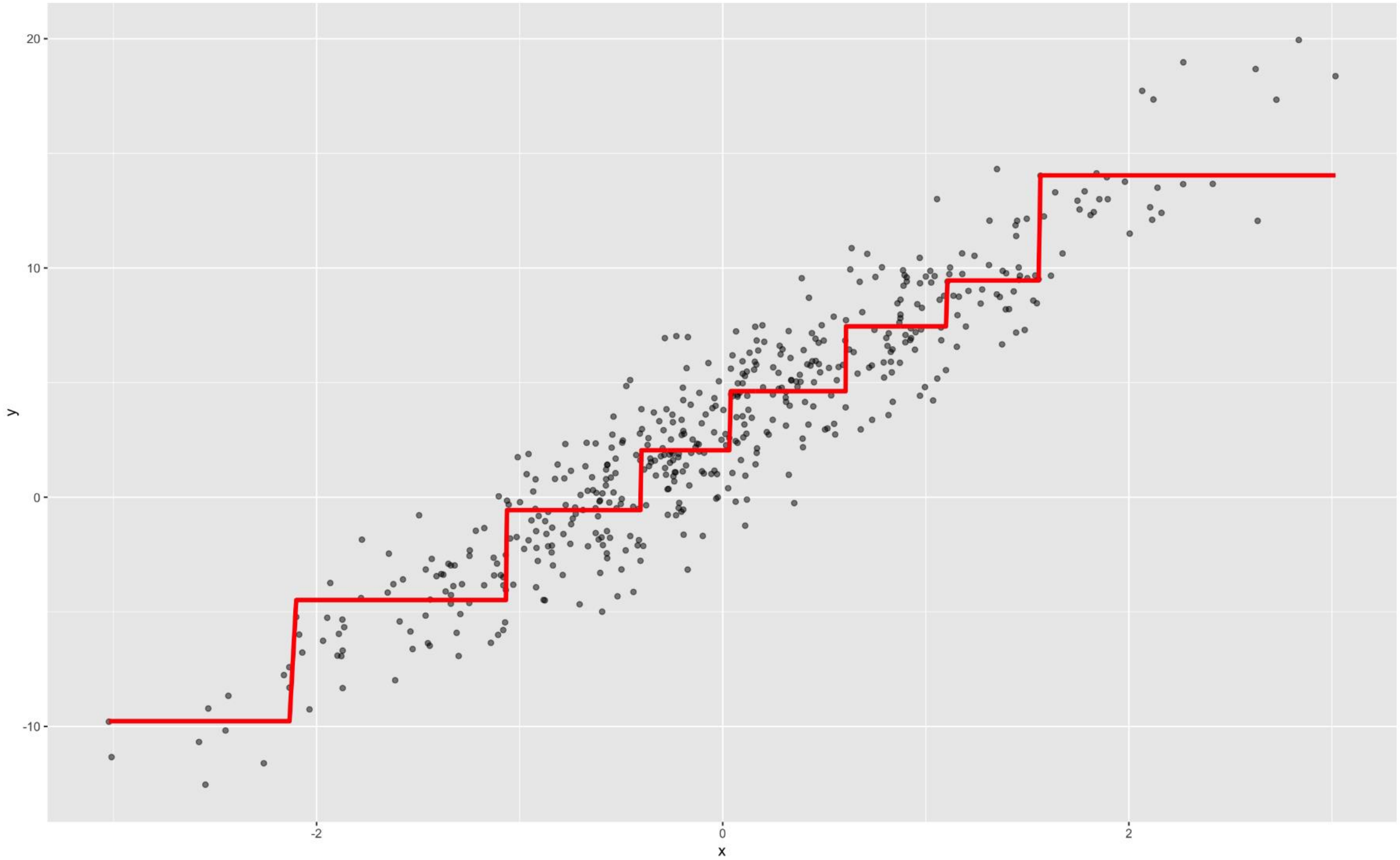
Generalized Linear Models

- If x increases by 1 unit, then on average y increases by β units…



Decision Trees

- If x is between A and B, then y is … If x is between C and D then y is …



“Interpretable”

- Most machine learning models are **not** interpretable in the classical sense – as one predictor variable increase, the target variable always does…BLAH.
- This is because the relationships are **not linear**.
- The relationships are more complicated than a linear relationship, so the interpretations are as well.

Still Want Interpretability

- People (especially clients) want to interpret and understand model behavior.
- Questions drive this need for interpretability:
  - Why was someone’s loan rejected?
  - Why is this symptom occurring in this patient?
  - Why is the stock price expected to decrease?

- Interpretations can be model and context dependent:
  - **Model** – variable importance in regression has different implication than variable importance in tree-based models.
  - **Context** – the effects of a change in a single variable on a target variable.

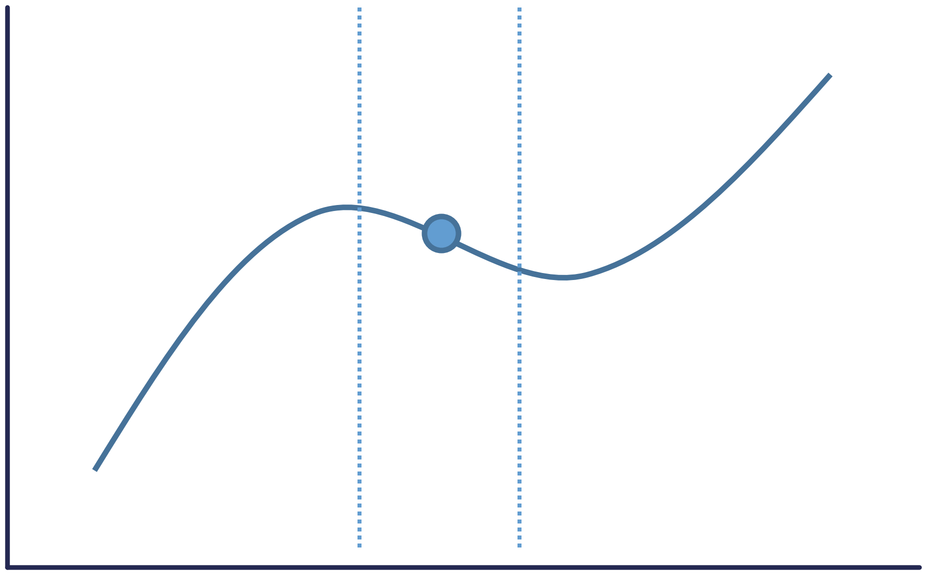
Importance of Interpretability

- Fairness/Transparency:
  - Understanding model decisions improves client/customer trust.
  - Interpretations reveal model behavior on different (potentially marginalized) groups of people.
- Model Robustness and Integrity:
  - Reveal odd model behavior or overfitting problems (does this conclusion make intuitive sense?).
- Adverse Action Requirements:
  - Equal Credit Opportunity Act (ECOA)
  - Fair Credit Reporting Act (FCRA)
  - More on the horizon…

Types of Model Interpretability

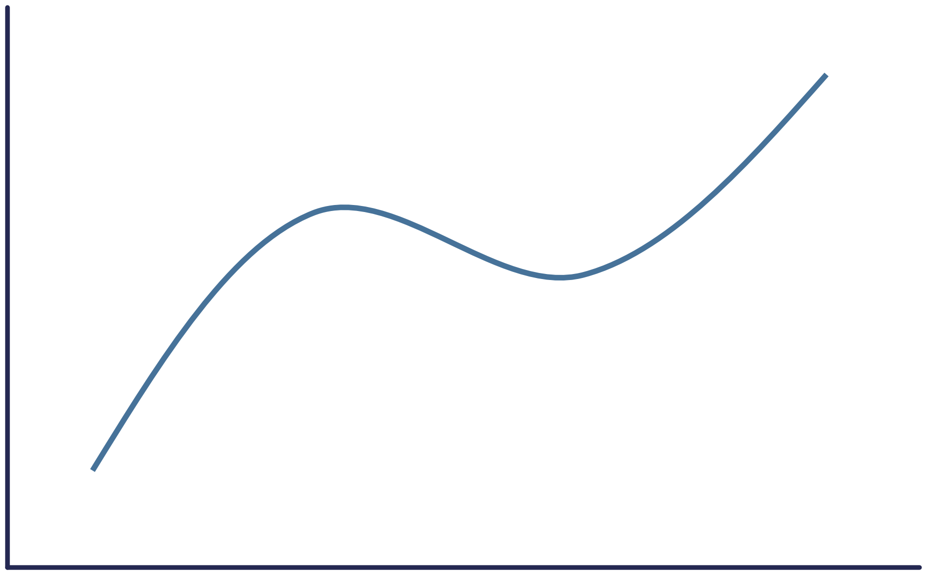
Local

- Example:
  - When x = 10, y decreases as x increases…



Global

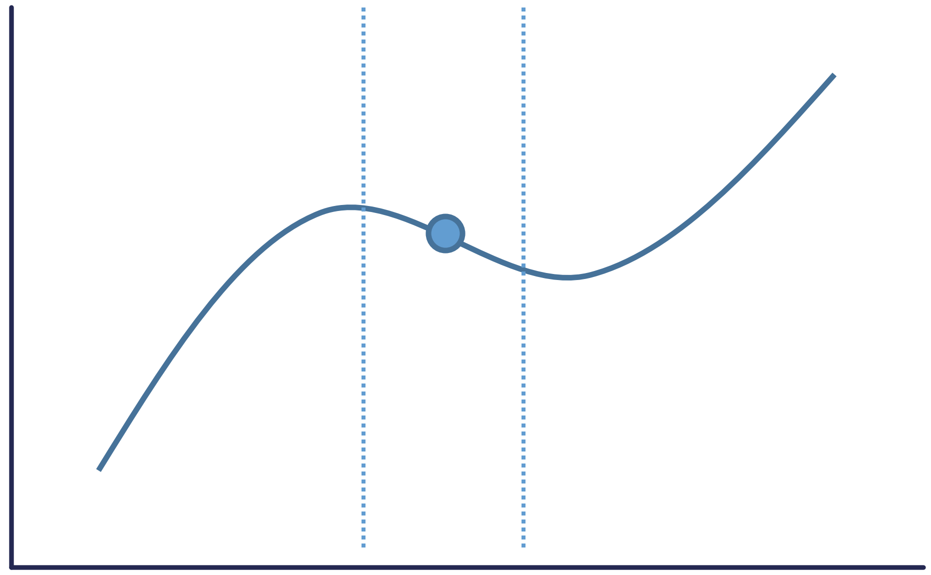
- Example:
  - As x increases, y **tends** to increase…



Model Agnostic Interpretability

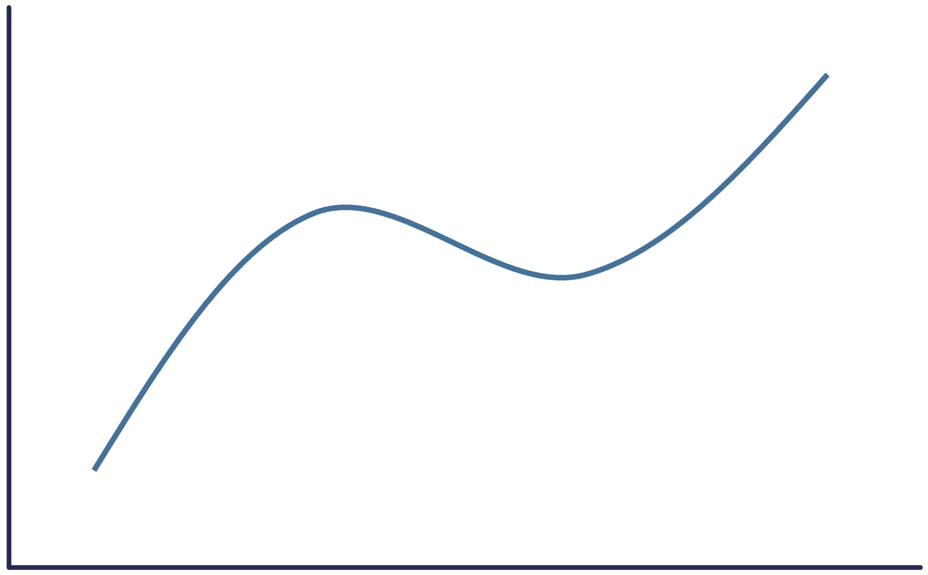
Local

- ICE
- LIME
- Shapley Values



Global

- Permutation Importance
- Partial Dependence
- ALE

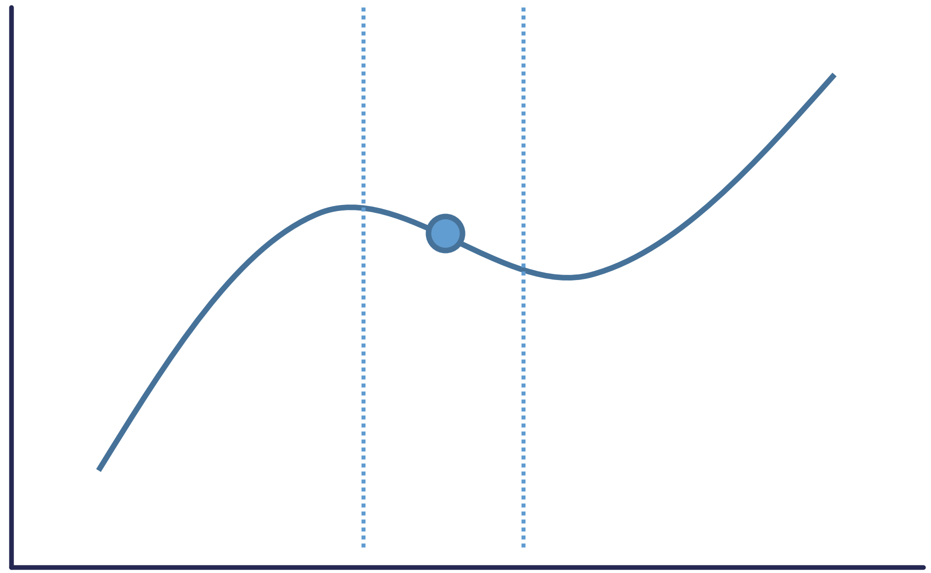


PERMUTATION IMPORTANCE

Model Agnostic Interpretability

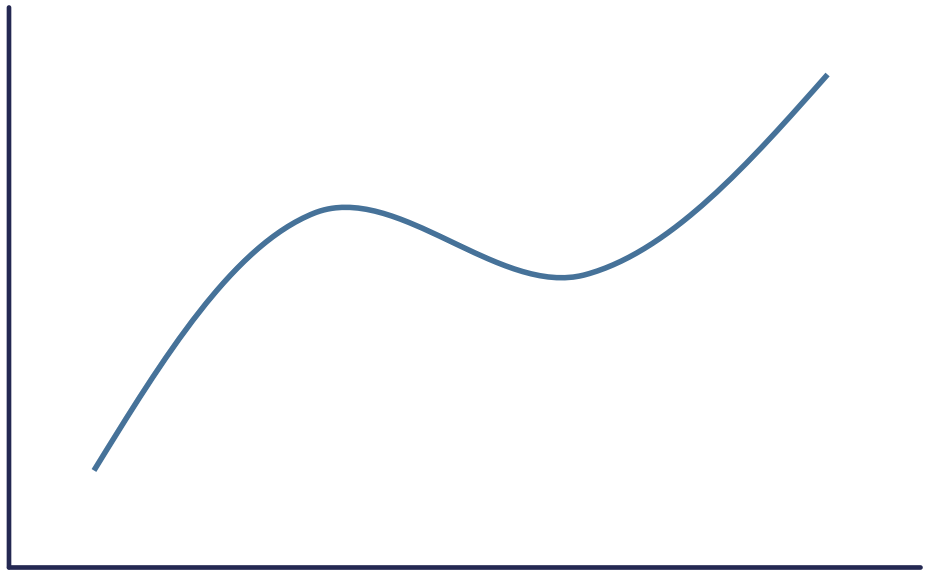
Local

- ICE
- LIME
- Shapley Values



Global

- **Permutation Importance**
- Partial Dependence
- ALE



Permutation Importance

- General Idea:
  - “Let me show you how much worse the predictions of our model get if we input randomly shuffled data values for each variable.”

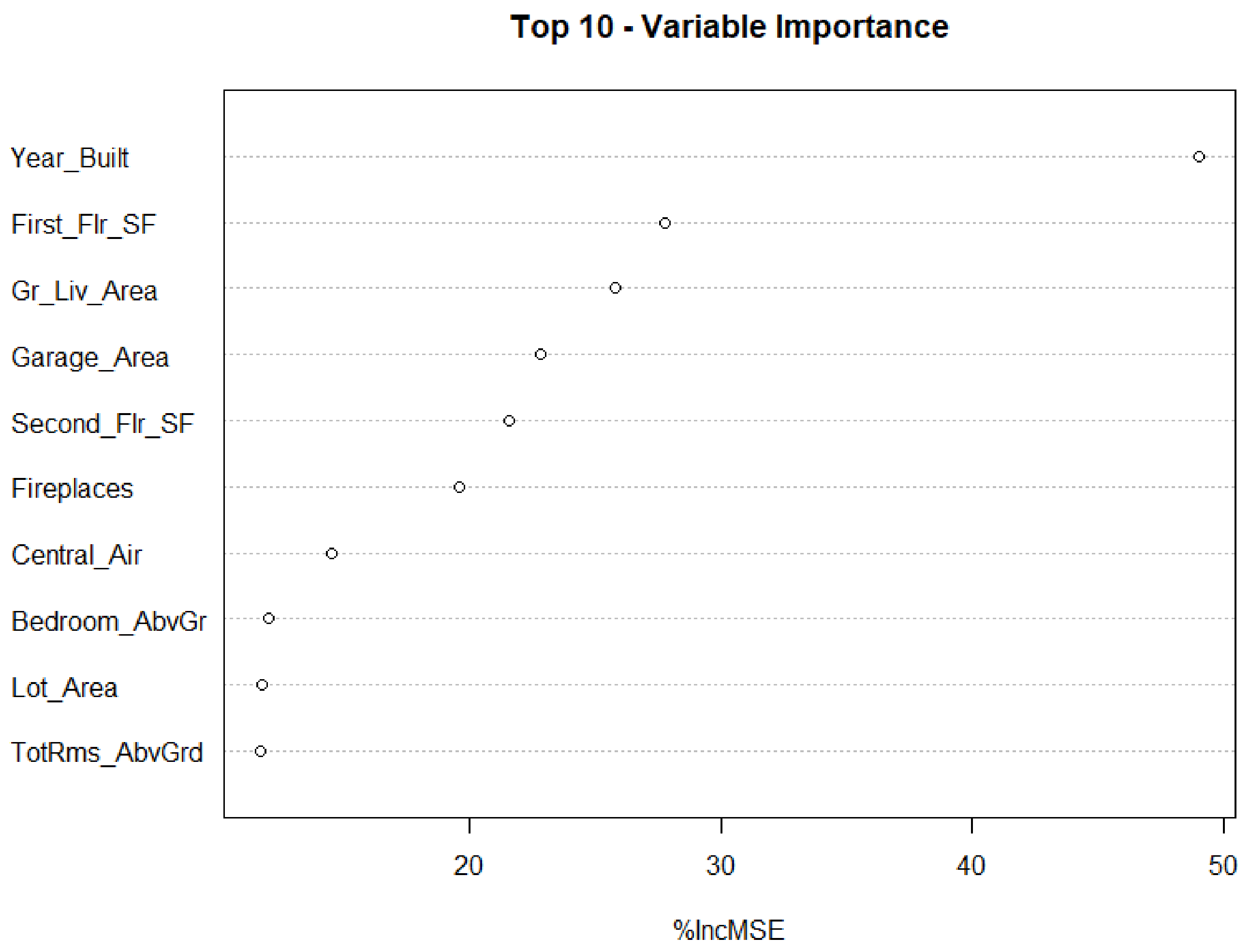
Permutation Importance

- If a variable is important, the model should get worse when that variable is removed.
- To make a direct comparison, rather than **remove** a variable from the model, we will **remove the signal** from the variable.
- By randomly shuffling (permuting) the values in that variable, we will break the true relationship between the variable and the target.

- Ex: How much worse does the model get when we take the average impact of 5 random permutations?

Permutation Importance (Random Forest)

- These were already given to us by default for random forests!



Permutation Importance (General)

```
library(iml)

lm.ames <- lm(Sale_Price ~ ., data = training)

linear_pred <- Predictor$new(lm.ames, data = training[,-1],
                             y = training$Sale_Price, type = "response")

plot(FeatureImp$new(linear_pred, loss = "mse"))
```

Build a linear regression just for example other model

Permutation Importance (General)

```
library(iml)

lm.ames <- lm(Sale_Price ~ ., data = training)

linear_pred <- Predictor$new(lm.ames, data = training[,-1],
                             y = training$Sale_Price, type = "response")

plot(FeatureImp$new(linear_pred, loss = "mse"))
```

Predictor variables

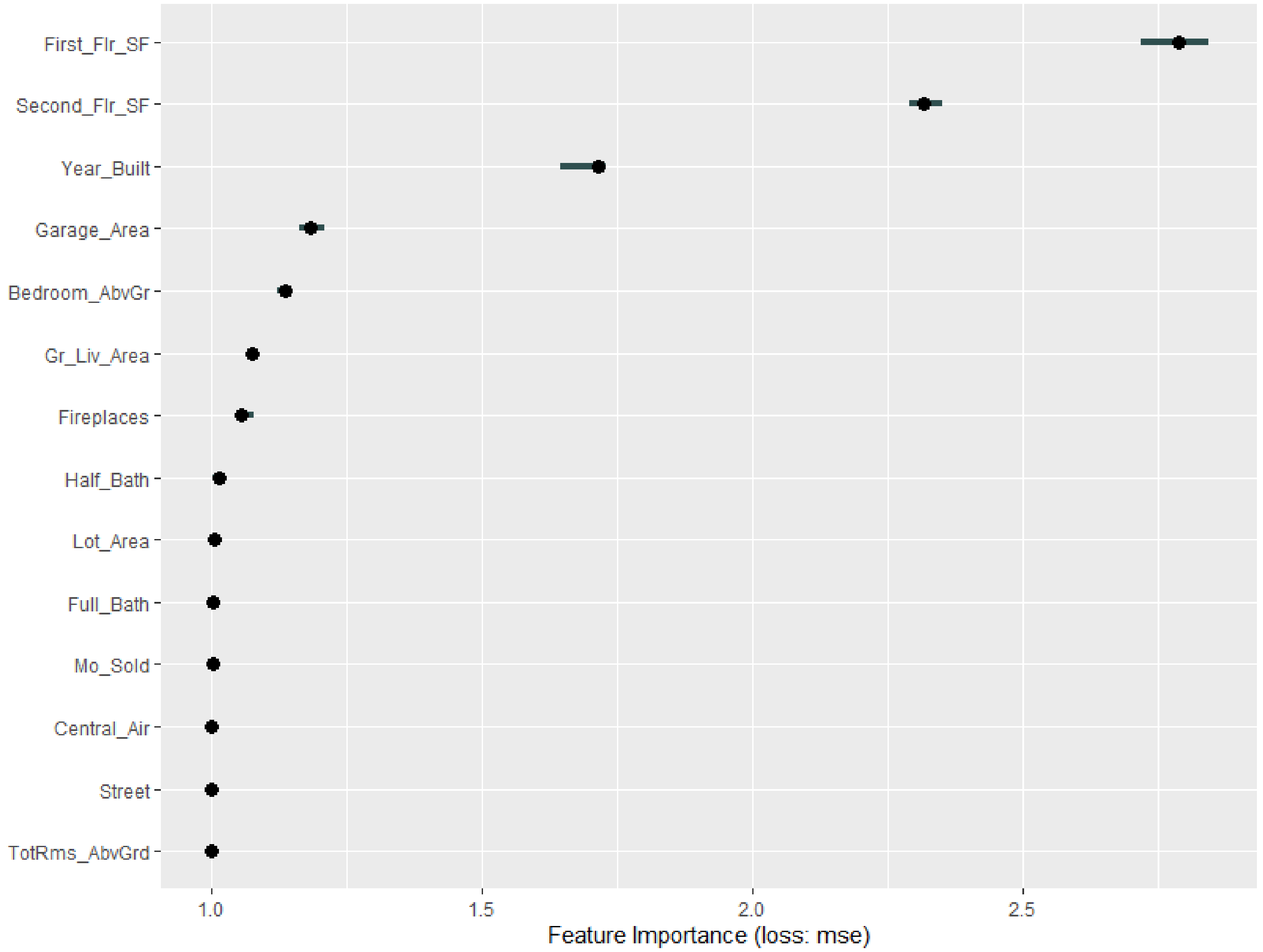
Target variable

Permutation Importance (General)

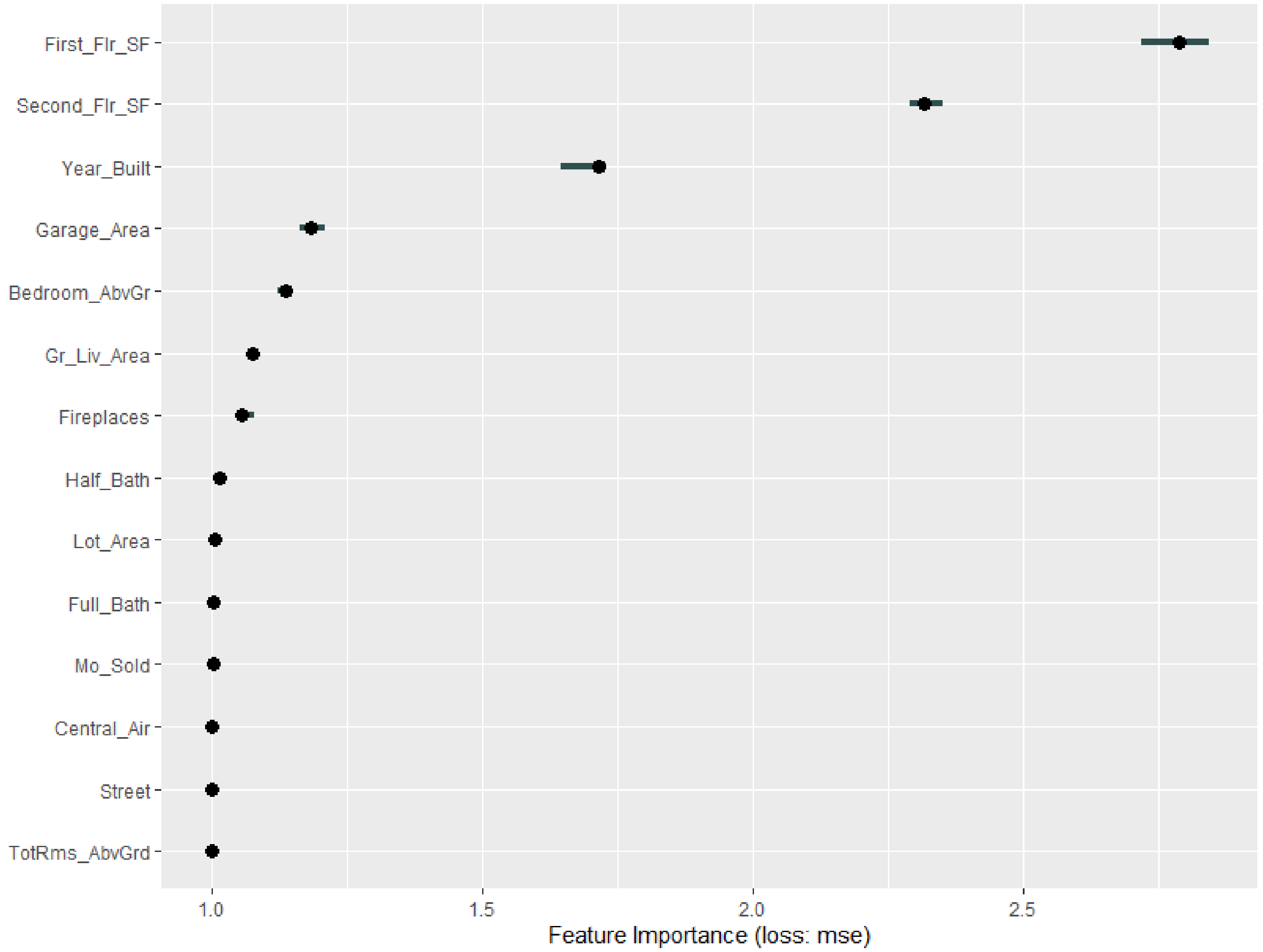
```
library(iml)

lm.ames <- lm(Sale_Price ~ ., data = training)

linear_pred <- Predictor$new(lm.ames, data = training[,-1],
                             y = training$Sale_Price, type = "response")

plot(FeatureImp$new(linear_pred, loss = "mse"))
```

Permutation based on MSE

Permutation Importance (General)



Permutation Importance (General)



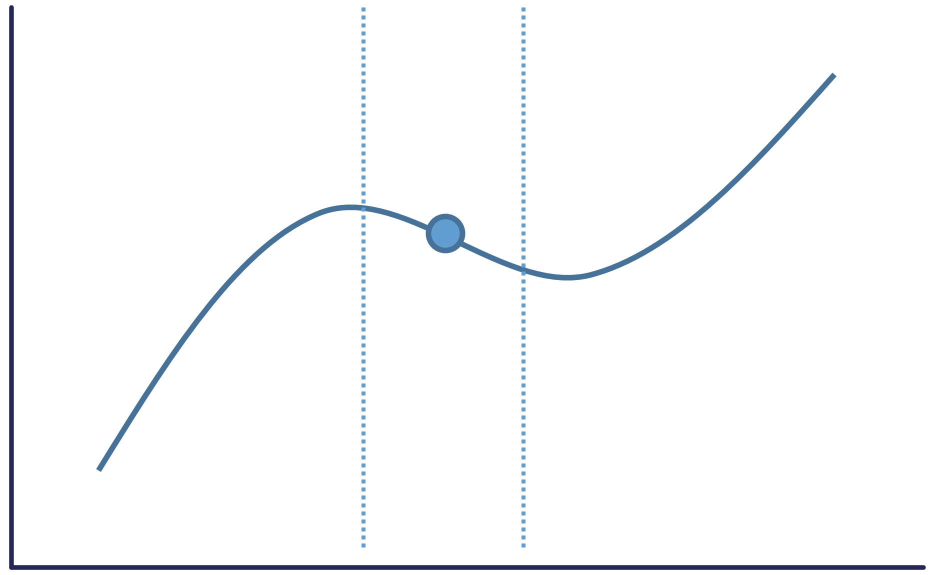

```
## Coefficients:
##                  Estimate   Pr(>|t|)
## (Intercept)   -1.513e+06    < 2e-16 ***
## Bedroom_AbvGr -1.251e+04   2.39e-16 ***
## Year_Built     7.700e+02    < 2e-16 ***
## Mo_Sold       -5.963e+02    0.06467 .
## Lot_Area       3.471e-01    0.00297 **
## StreetPave     2.173e+04    0.15425
## Central_AirY   6.043e+03    0.11900
## First_Flr_SF   9.762e+01   2.26e-06 ***
## Second_Flr_SF  7.404e+01    0.00033 ***
## Full_Bath     -3.270e+03    0.18077
## Half_Bath     -6.372e+03    0.01139 *
## Fireplaces     1.056e+04   2.61e-11 ***
## Garage_Area    5.816e+01    < 2e-16 ***
## Gr_Liv_Area    1.601e+01     .43115
## TotRms_AbvGrd -5.310e+02    0.62539
```

INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

Model Agnostic Interpretability

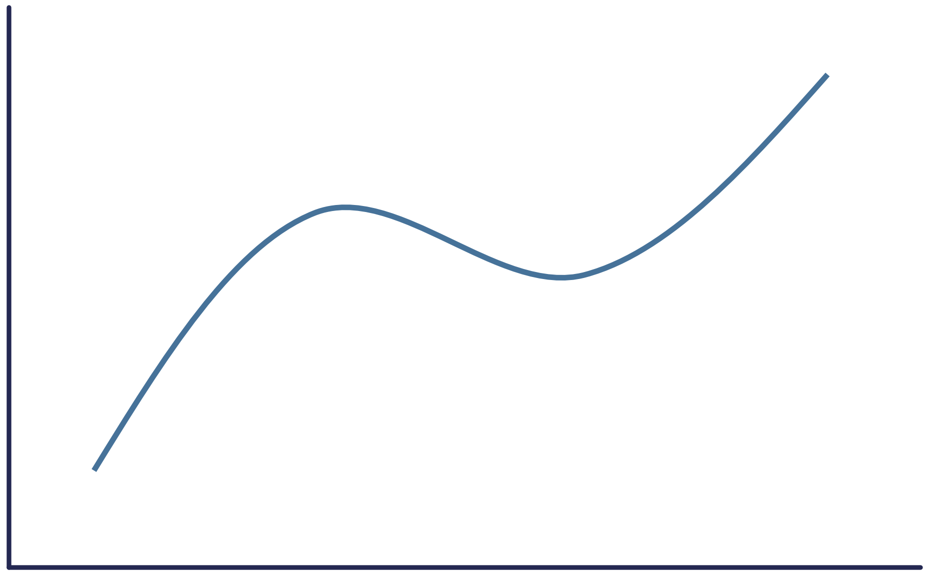
Local

- **ICE**
- LIME
- Shapley Values



Global

- Permutation Importance
- Partial Dependence
- ALE



Individual **C**onditional **E**xpectation

- General Idea:
  - “Let me show you how the predictions for each observation change if we vary the feature of interest.”

Individual Conditional Expectation

- Local method → visualizes the dependence of an **individual prediction** on a given predictor variable.
- Fix all other variables for a single observation while changing the variable of interest.
- Plot the resulting predictions vs. the variable of interest.

ICE Example

- Choose a **variable of interest** and a **single observation**.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	725	1947	…	274,000	248,756.99
1922	Y	100	816	…	75,200	92,225.59
2005	Y	784	2358	…	329,900	331,746.07
1926	Y	506	1285	…	145,400	122,841.74

ICE Example

- Choose a **variable of interest** and a **single observation**.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	725	1947	…	274,000	248,756.99
1922	Y	100	816	…	75,200	92,225.59
2005	Y	784	2358	…	329,900	331,746.07
1926	Y	506	1285	…	145,400	122,841.74

ICE Example

- Replicate single observation, holding **all other variables constant.**

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y		1947	…	274,000	
2001	Y		1947	…	274,000	
2001	Y		1947	…	274,000	
…	…	…	…	…	…	…
2001	Y		1947	…	274,000	

ICE Example

- Fill in values for variable of interest across the entire range of the variable.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	
2001	Y	1	1947	…	274,000	
2001	Y	2	1947	…	274,000	
…	…	…	…	…	…	…
2001	Y	1488	1947	…	274,000	

ICE Example

- Use the model to predict each of these simulated observations.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	213,198.60
2001	Y	1	1947	…	274,000	213,198.60
2001	Y	2	1947	…	274,000	213,198.60
…	…	…	…	…	…	…
2001	Y	1488	1947	…	274,000	268,953.00

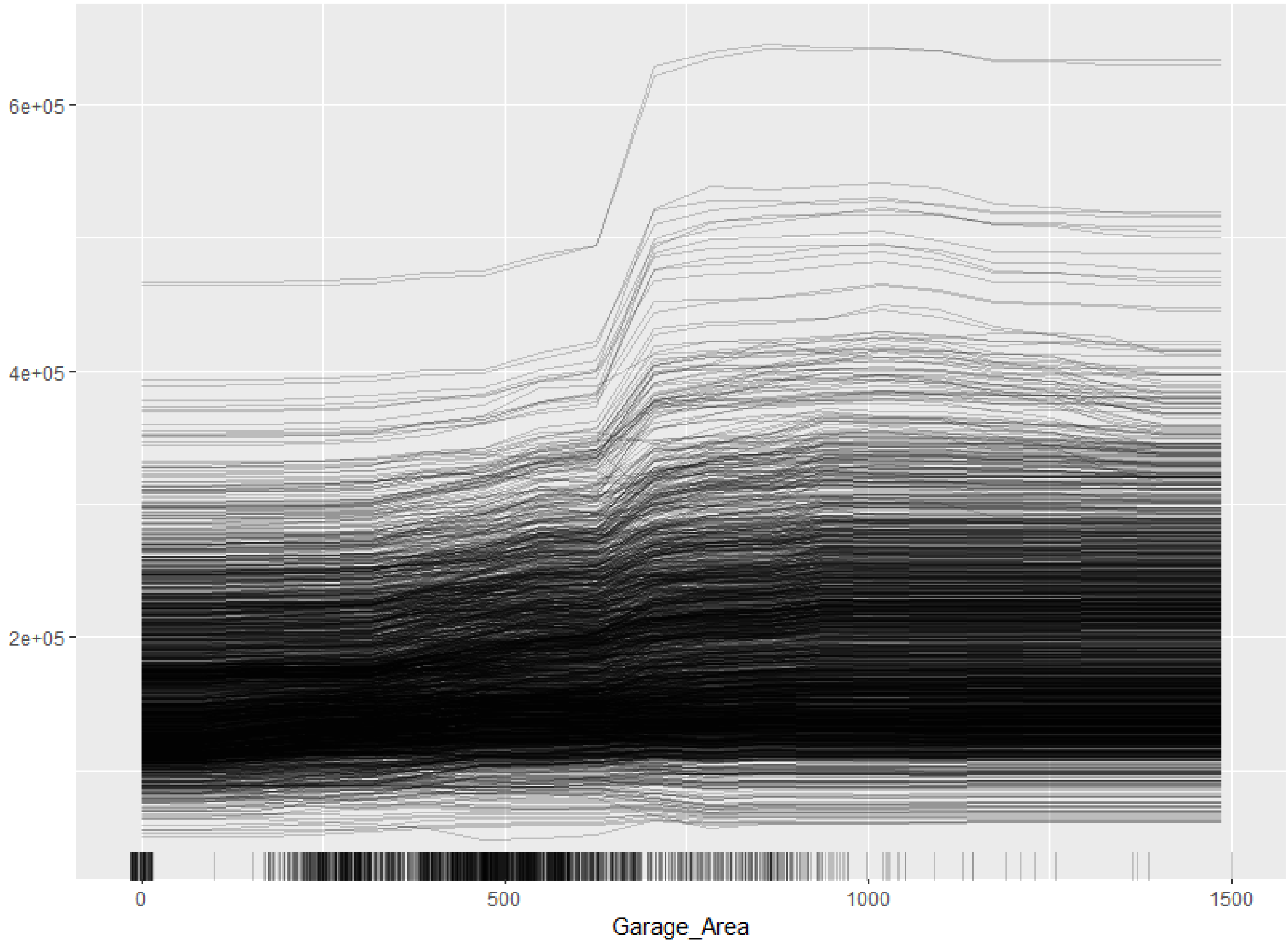
ICE Example

- REPEAT FOR ALL OBSERVATIONS (or large sample)!

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	725	1947	…	274,000	248,756.99
1922	Y	100	816	…	75,200	92,225.59
2005	Y	784	2358	…	329,900	331,746.07
1926	Y	506	1285	…	145,400	122,841.74

ICE Plot

- One line for every observation in the dataset.



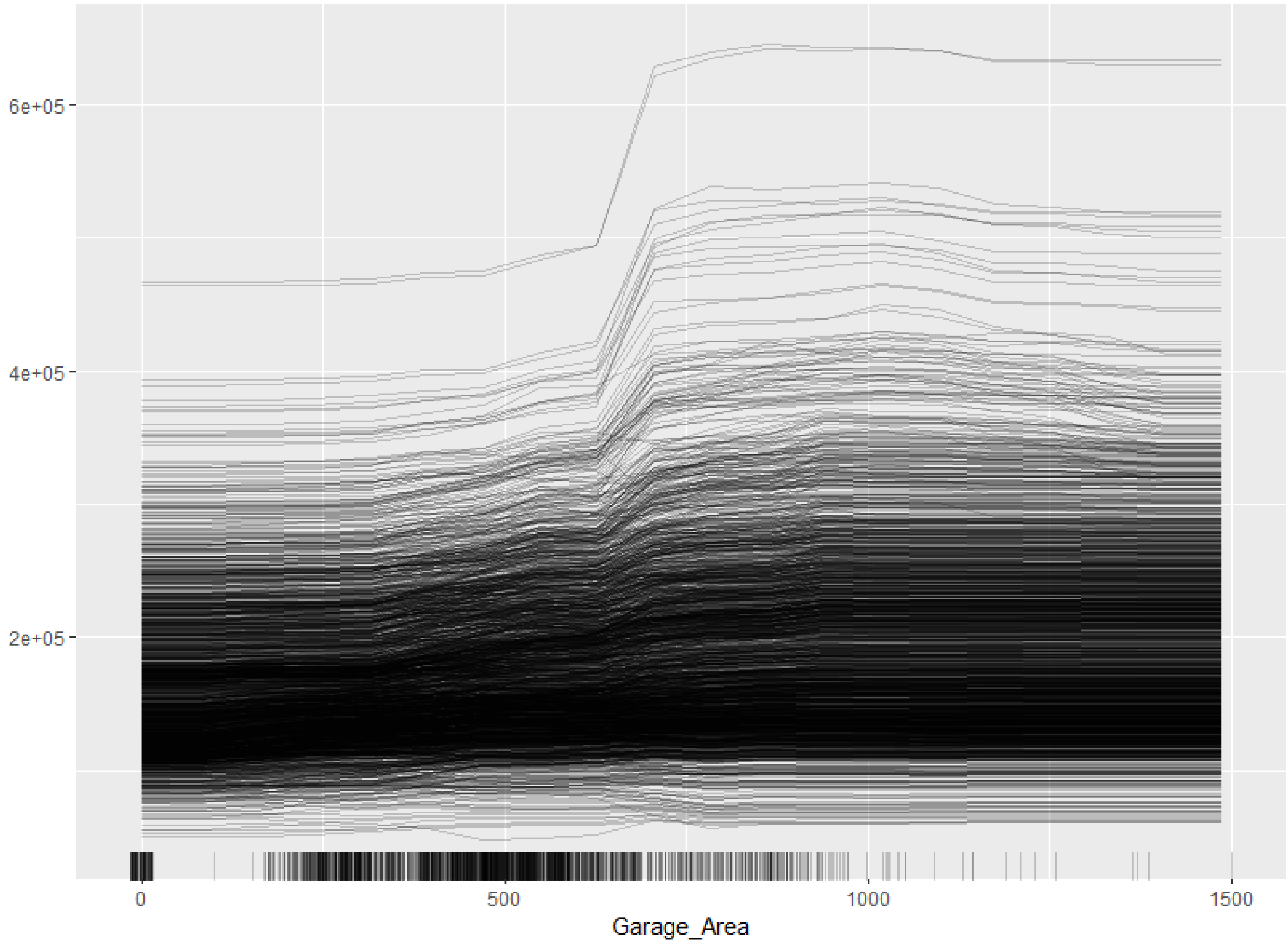
Build Model for ICE

```
set.seed(12345)
rf.ames <- randomForest(Sale_Price ~ ., data = training.df,
                        ntree = 500, importance = TRUE)
```

Let’s build a random forest model

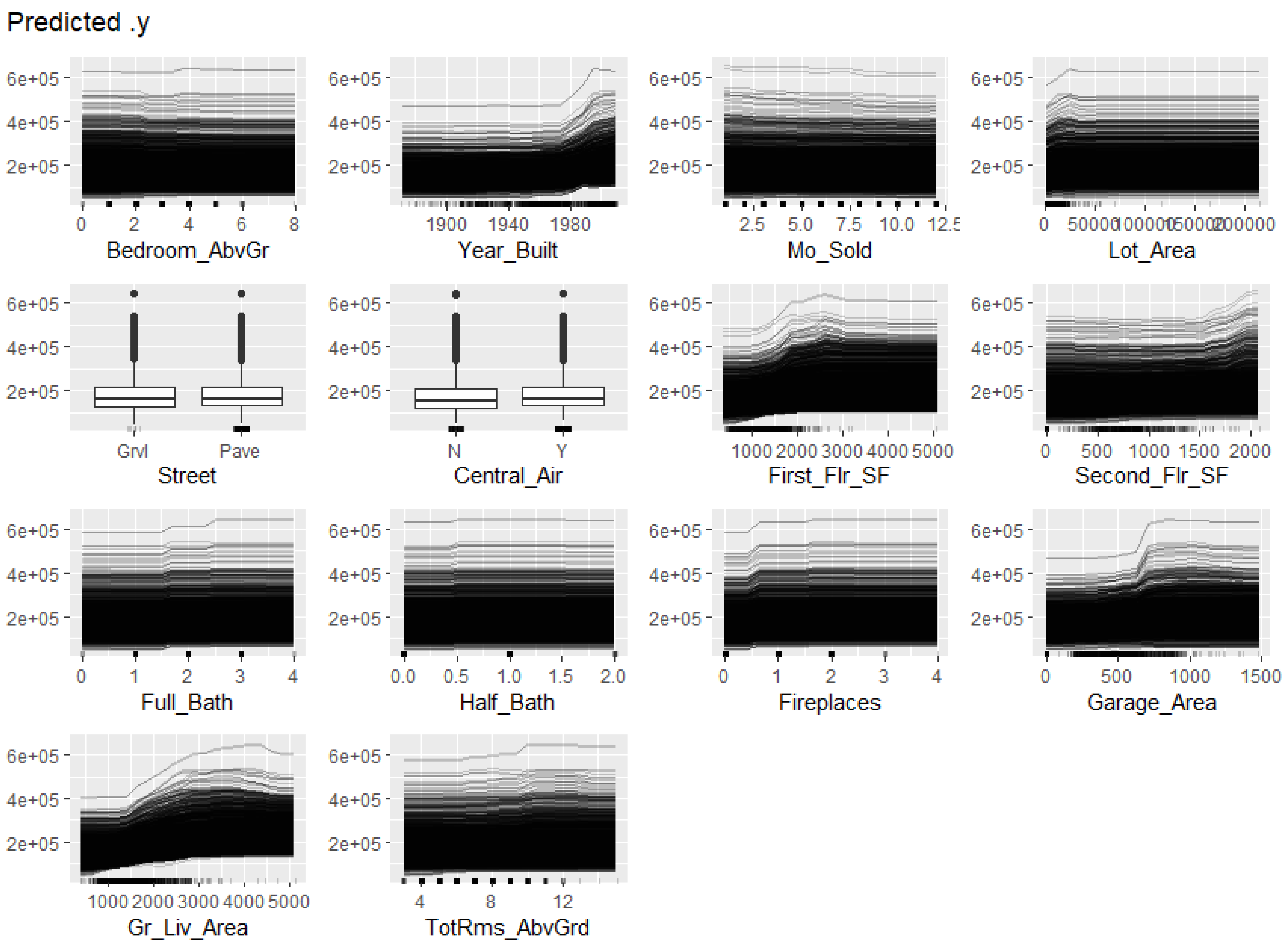
ICE Plot

```r
set.seed(12345)
forest_pred <- Predictor$new(rf.ames,
                    data = training[,-1],
                    y = training$Sale_Price,
                    type = "response")

ice_plot <- FeatureEffects$new(forest_pred,
                 method = "ice")

ice_plot$plot(c("Garage_Area"))
```



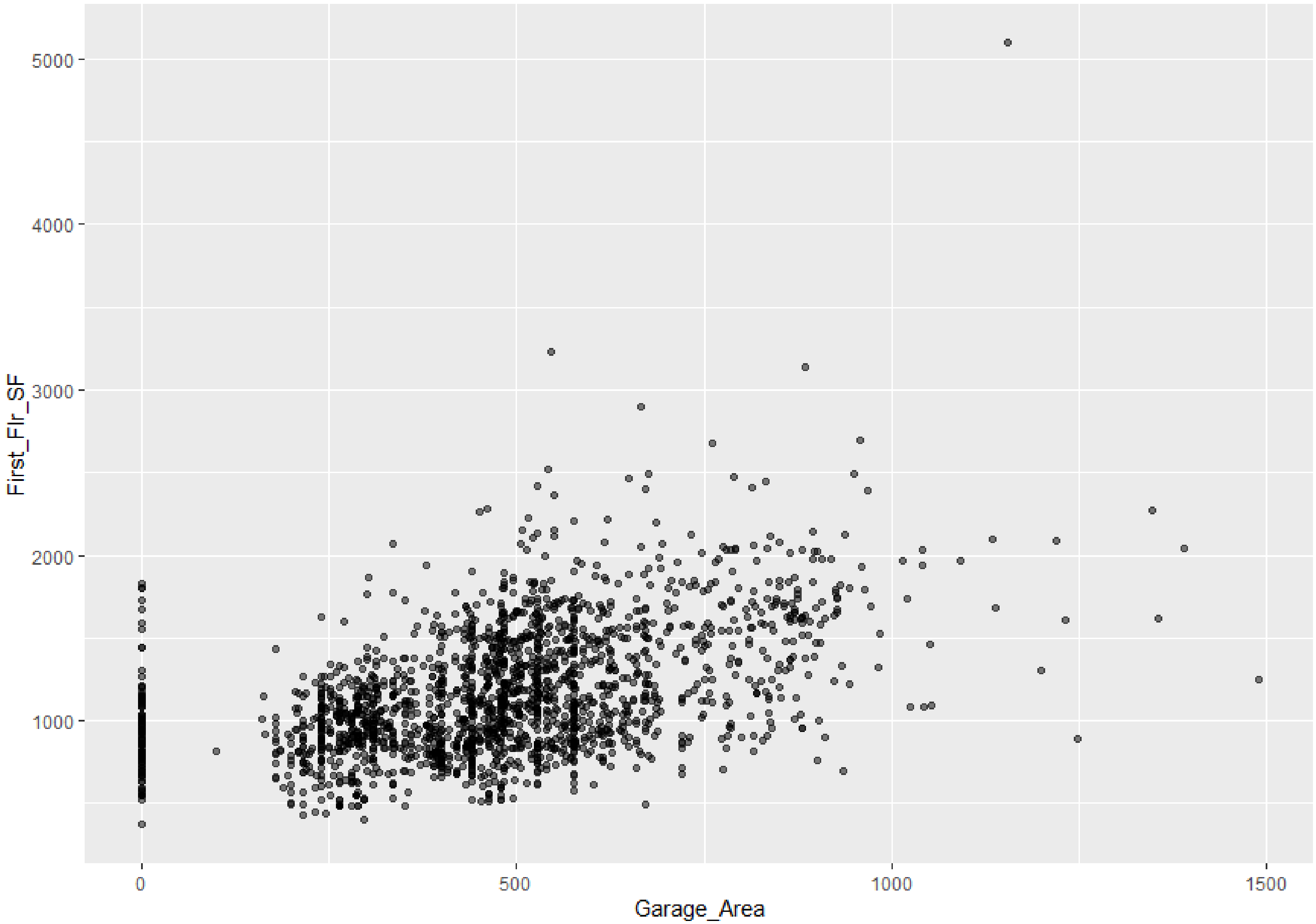
ICE Plot

```
ice_plot$plot()
```



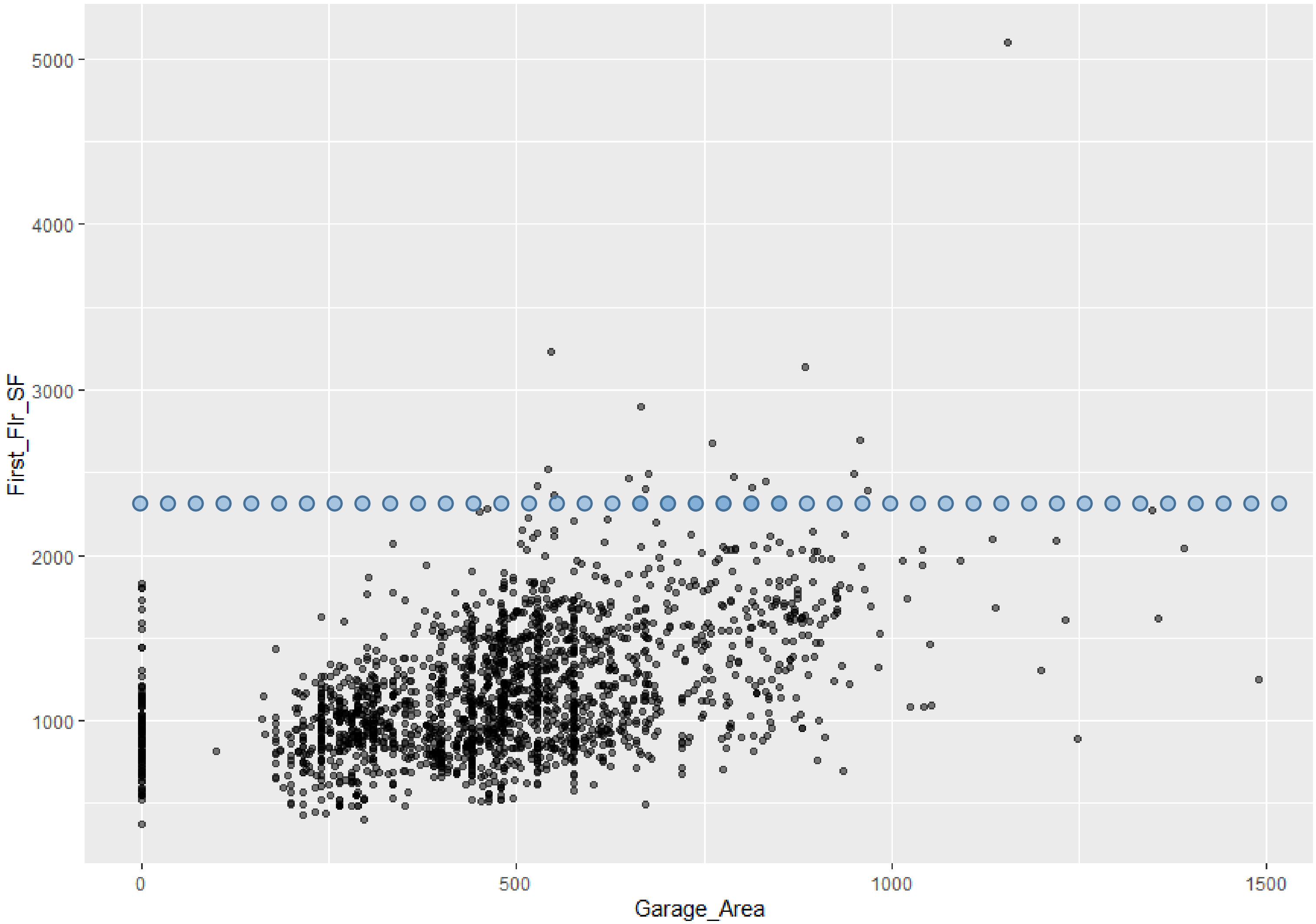
Multicollinearity Problems

- If the variable of interest is correlated with other inputs, some of the simulated data may be invalid.



Multicollinearity Problems

- If the variable of interest is correlated with other inputs, some of the simulated data may be invalid.
- Keep all other variables constant while replicating across all values of garage area.



Summary

Advantages

- Intuitive → one line represents predictions for one observation if we change the variable of interest.
- Capable of showing changing relationships (different impact of variable across different observations).

Disadvantages

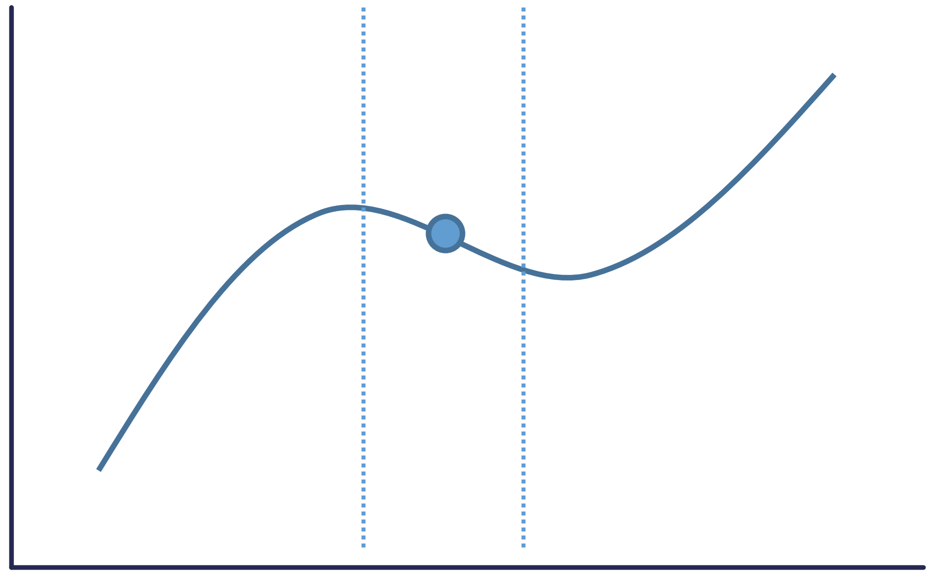
- Computationally expensive.
- Hard to visualize with many observations.
- Multicollinearity problems.
- One variable at a time.

PARTIAL DEPENDENCE

Model Agnostic Interpretability

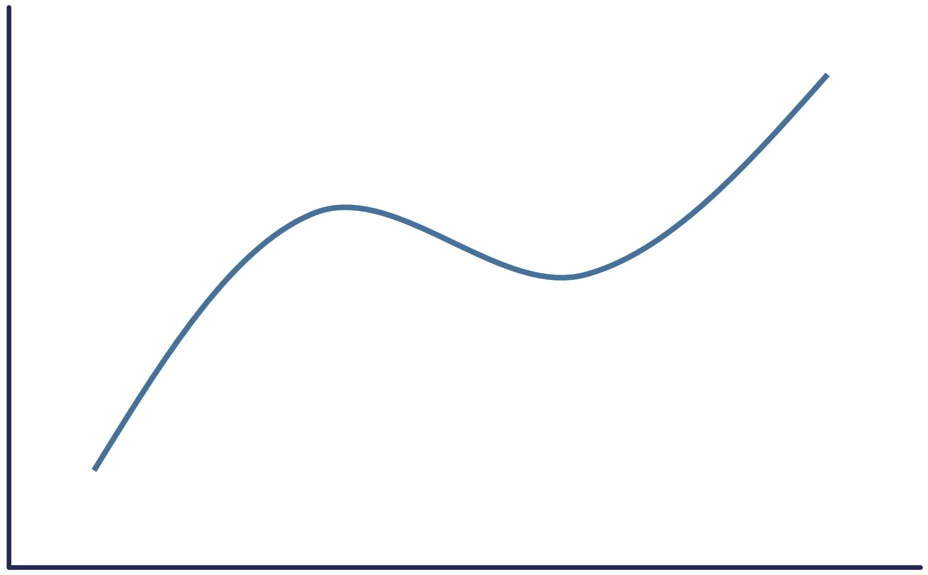
Local

- ICE
- LIME
- Shapley Values



Global

- Permutation Importance
- **Partial Dependence**
- ALE



Partial Dependence

- General Idea:
  - “Let me show you what the model predicts on average when each observation has the value *k* for that feature. We’ll ignore whether the value *k* makes sense for all data instances.”

Partial Dependence Plots (PDP)

- Attempts to show **marginal** effect of inputs on the target variable.
- **Marginal effects** are essentially **averaged** effects over all possible values of a single variable.

- **Think average of all the lines in the ICE plot!**

PDP Example

- Choose a **variable of interest**.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	725	1947	…	274,000	248,756.99
1922	Y	100	816	…	75,200	92,225.59
2005	Y	784	2358	…	329,900	331,746.07
1926	Y	506	1285	…	145,400	122,841.74

PDP Example

- Replicate your dataset, holding all variables constant **except** the variable of interest.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y		1947	…	274,000	
1922	Y		816	…	75,200	
2005	Y		2358	…	329,900	
1926	Y		1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y		1947	…	274,000	
1922	Y		816	…	75,200	
2005	Y		2358	…	329,900	
1926	Y		1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y		1947	…	274,000	
1922	Y		816	…	75,200	
2005	Y		2358	…	329,900	
1926	Y		1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y		1947	…	274,000	
1922	Y		816	…	75,200	
2005	Y		2358	…	329,900	
1926	Y		1285	…	145,400	

⋮

PDP Example

- Replicate your dataset, holding all variables constant **except** the variable of interest.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	
1922	Y	0	816	…	75,200	
2005	Y	0	2358	…	329,900	
1926	Y	0	1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1487	1947	…	274,000	
1922	Y	1487	816	…	75,200	
2005	Y	1487	2358	…	329,900	
1926	Y	1487	1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1	1947	…	274,000	
1922	Y	1	816	…	75,200	
2005	Y	1	2358	…	329,900	
1926	Y	1	1285	…	145,400	

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1488	1947	…	274,000	
1922	Y	1488	816	…	75,200	
2005	Y	1488	2358	…	329,900	
1926	Y	1488	1285	…	145,400	

⋮

PDP Example

- Use model to generate predictions for all this simulated data.

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	213,198.63
1922	Y	0	816	…	75,200	83,683.52
2005	Y	0	2358	…	329,900	271,165.62
1926	Y	0	1285	…	145,400	113,025.91

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1487	1947	…	274,000	268,952.96
1922	Y	1487	816	…	75,200	97,340.47
2005	Y	1487	2358	…	329,900	329,710.53
1926	Y	1487	1285	…	145,400	133,129.74

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1	1947	…	274,000	213,198.63
1922	Y	1	816	…	75,200	83,683.52
2005	Y	1	2358	…	329,900	271,165.62
1926	Y	1	1285	…	145,400	113,025.91

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1488	1947	…	274,000	268,952.96
1922	Y	1488	816	…	75,200	97,340.47
2005	Y	1488	2358	…	329,900	329,710.53
1926	Y	1488	1285	…	145,400	133,129.74

⋮

PDP Example

- Use model to generate predictions for all this simulated data.
- **Take average of each prediction column corresponding to variable value.**

$\bar{x}_0$

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	213,198.63
1922	Y	0	816	…	75,200	83,683.52
2005	Y	0	2358	…	329,900	271,165.62
1926	Y	0	1285	…	145,400	113,025.91

$\bar{x}_{1487}$

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1487	1947	…	274,000	268,952.96
1922	Y	1487	816	…	75,200	97,340.47
2005	Y	1487	2358	…	329,900	329,710.53
1926	Y	1487	1285	…	145,400	133,129.74

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1	1947	…	274,000	213,198.63
1922	Y	1	816	…	75,200	83,683.52
2005	Y	1	2358	…	329,900	271,165.62
1926	Y	1	1285	…	145,400	113,025.91

$\bar{x}_1$

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	1488	1947	…	274,000	268,952.96
1922	Y	1488	816	…	75,200	97,340.47
2005	Y	1488	2358	…	329,900	329,710.53
1926	Y	1488	1285	…	145,400	133,129.74

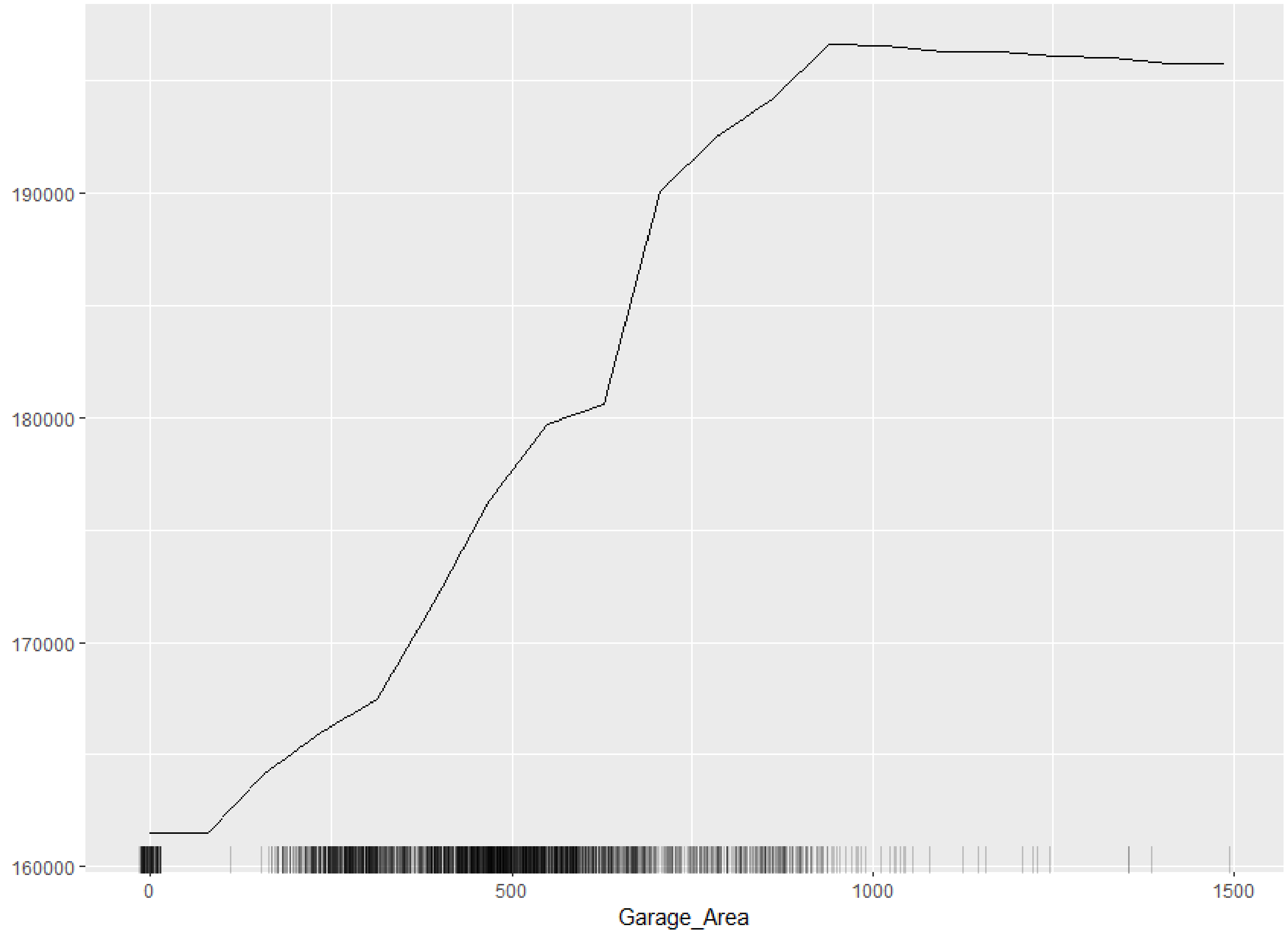
$\bar{x}_{1488}$

⋮

Partial Dependence Plot

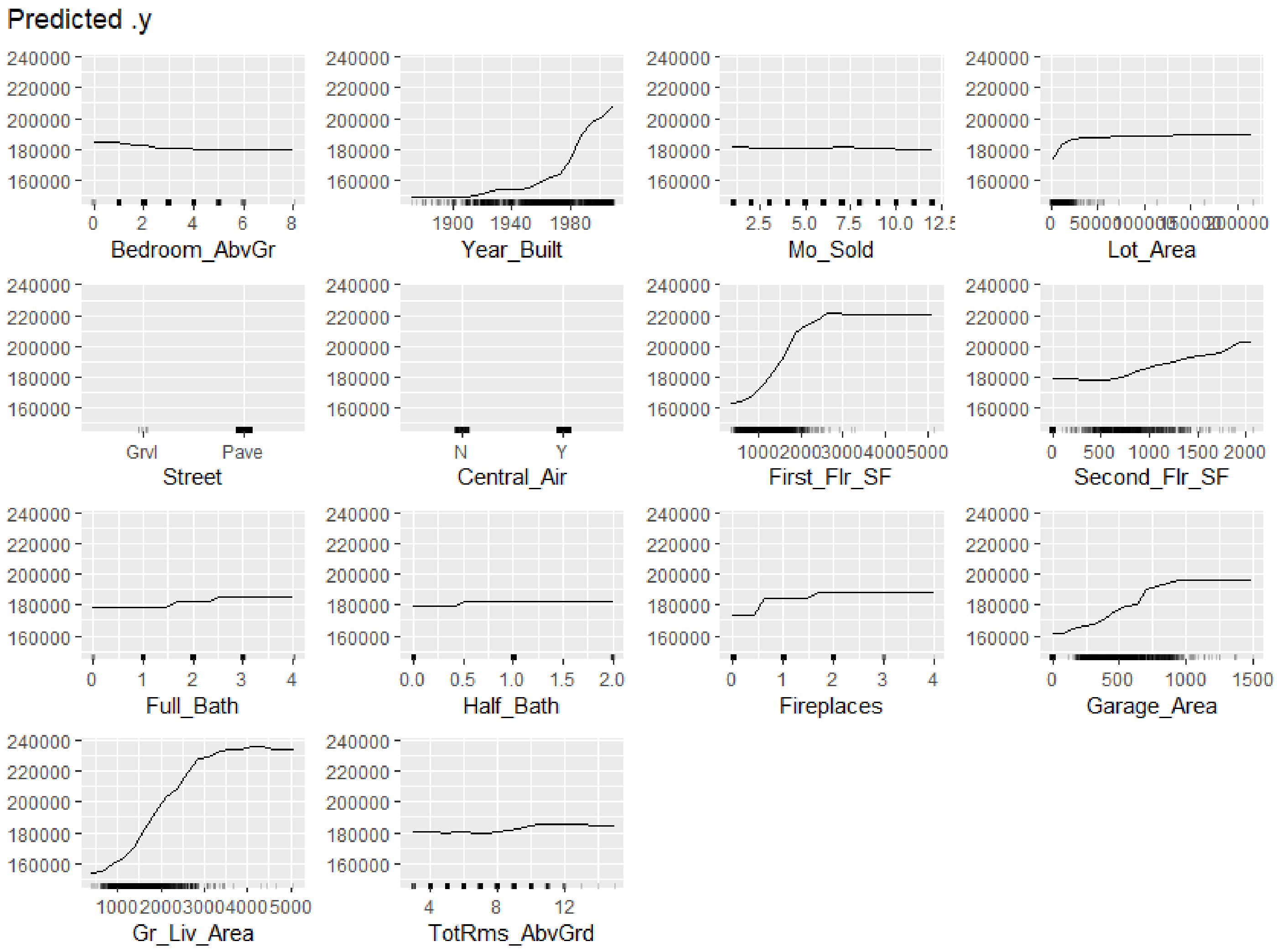
Same thing as ICE

```
set.seed(12345)
forest_pred <- Predictor$new(rf.ames,
                  data = training[,-1],
                  y = training$Sale_Price,
                  type = "response")


pd_plot <- FeatureEffects$new(forest_pred,
               method = "pdp")

pd_plot$plot(c("Garage_Area"))
```



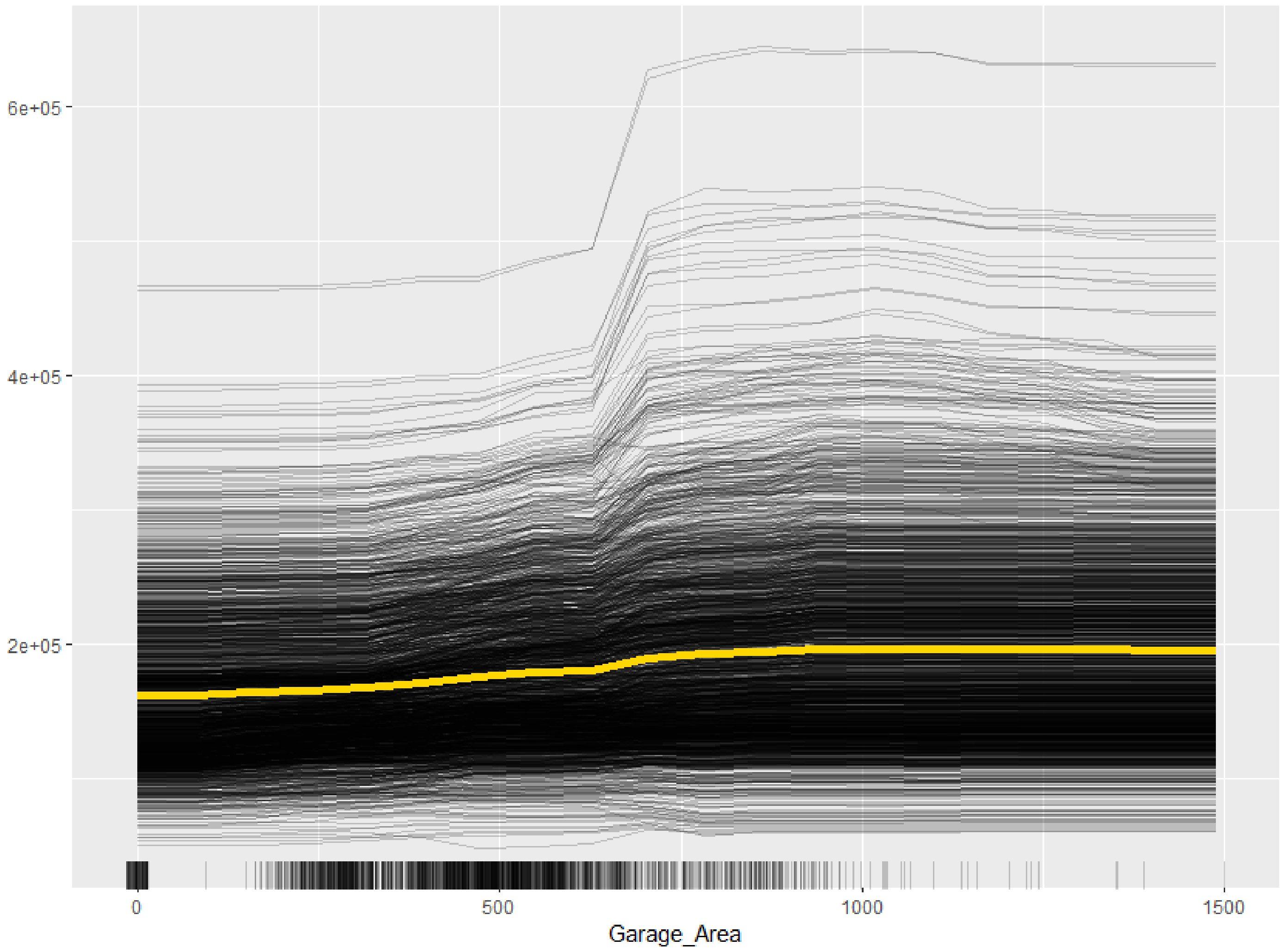
Partial Dependence Plot

```
pd_plot <- FeatureEffects$new(forest_pred,
           method = "pdp")

pd_plot$plot()
```

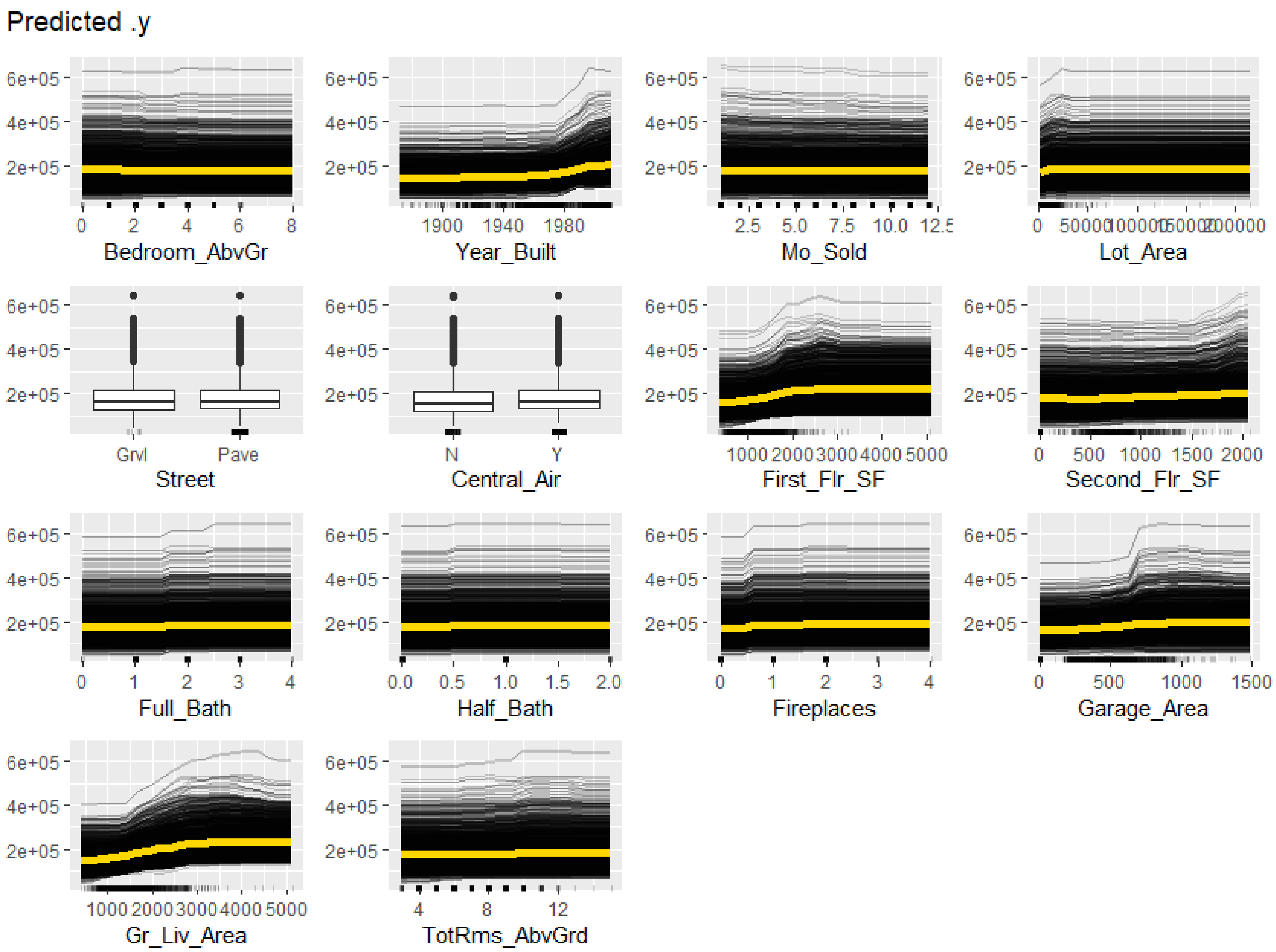


Partial Dependence Plot with ICE

```
pd_plot <- FeatureEffects$new(forest_pred,
            method = "pdp+ice")

pd_plot$plot(c("Garage_Area"))
```

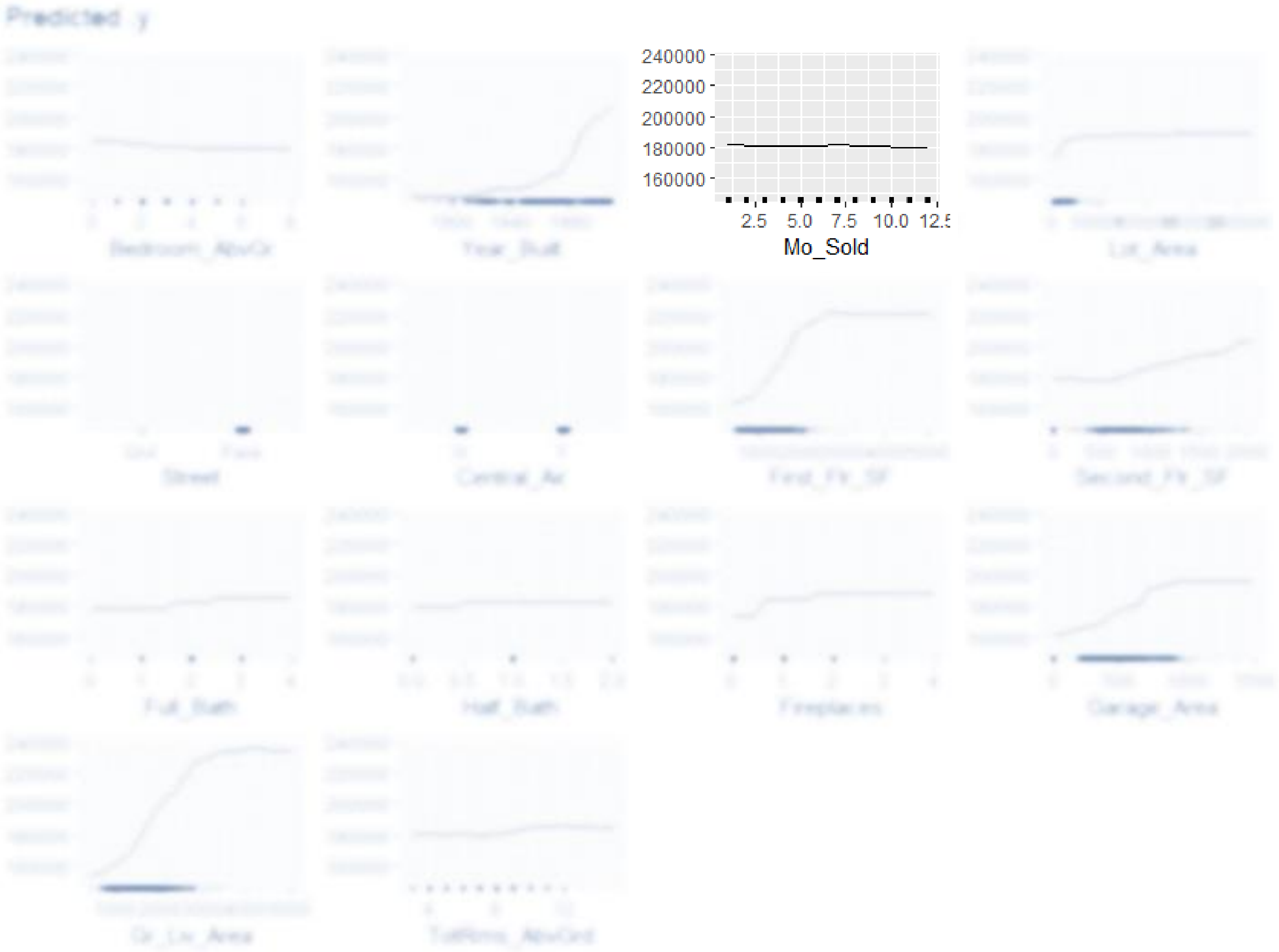


Partial Dependence Plot Interpretation



Watch-out for Scale!

Watch-out for Scale!

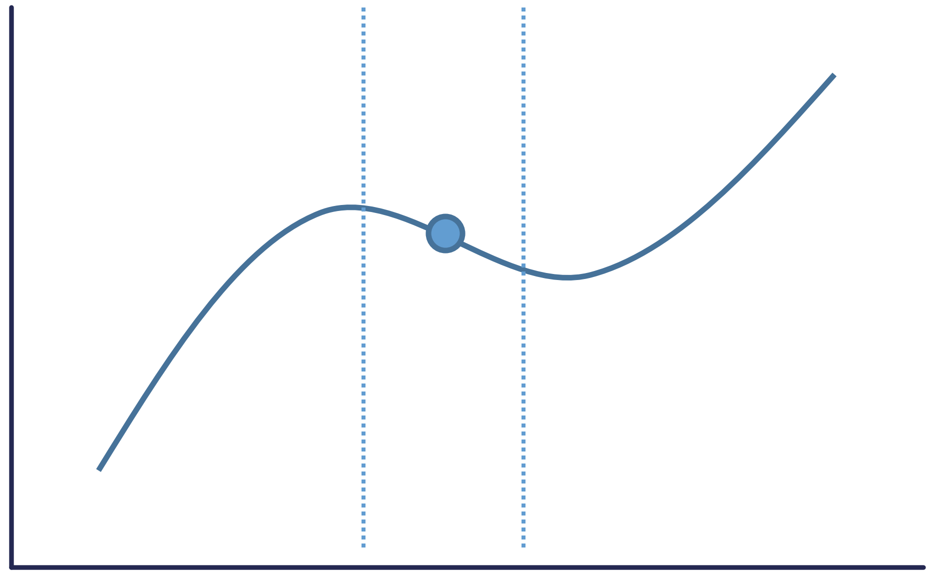


ACCUMULATED LOCAL EFFECTS (ALE)

Model Agnostic Interpretability

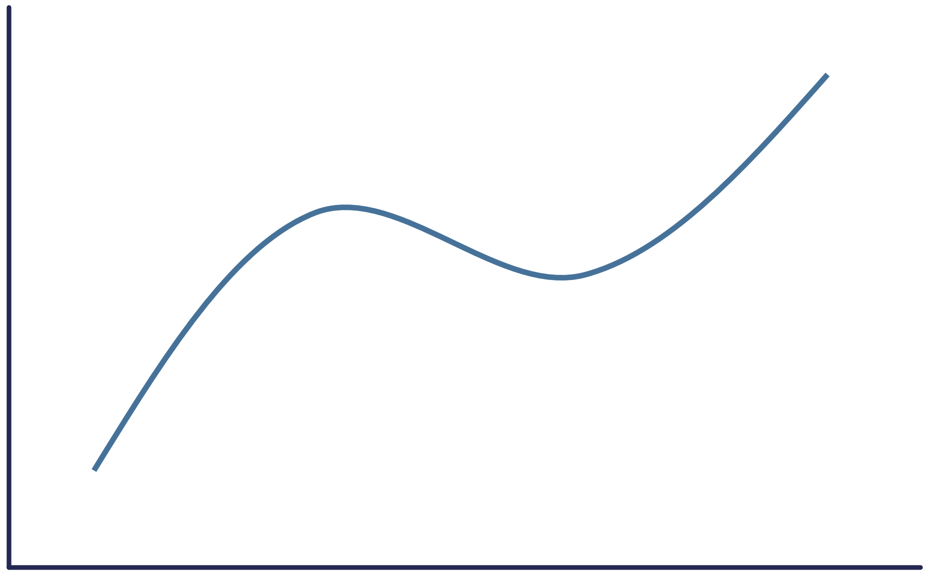
Local

- ICE
- LIME
- Shapley Values



Global

- Permutation Importance
- Partial Dependence
- **ALE**

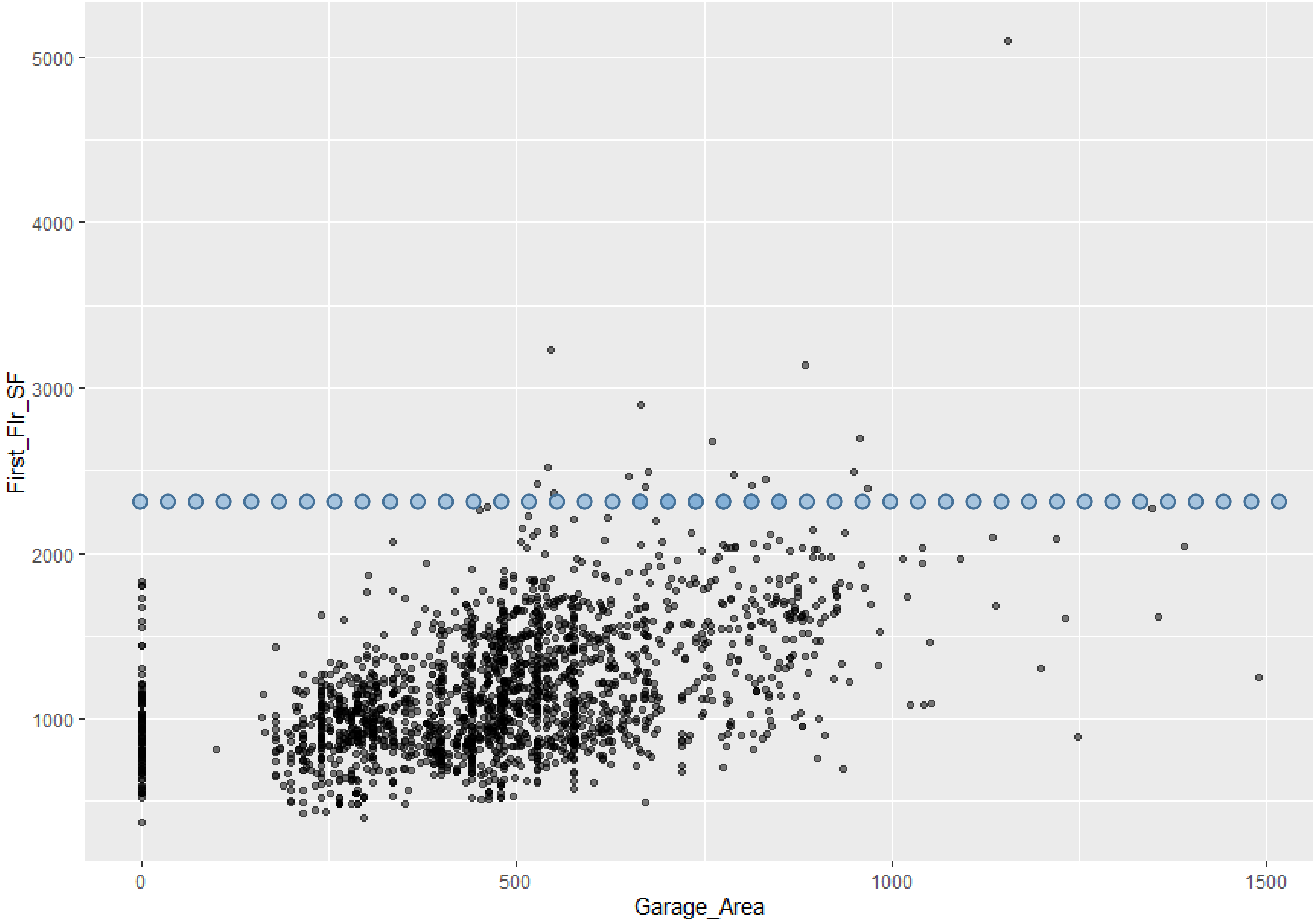


Accumulated **L**ocal **E**ffects

- General Idea:
  - “Let me show you how the model predictions change when I change the variable the interest to values within a small interval around their current values.”

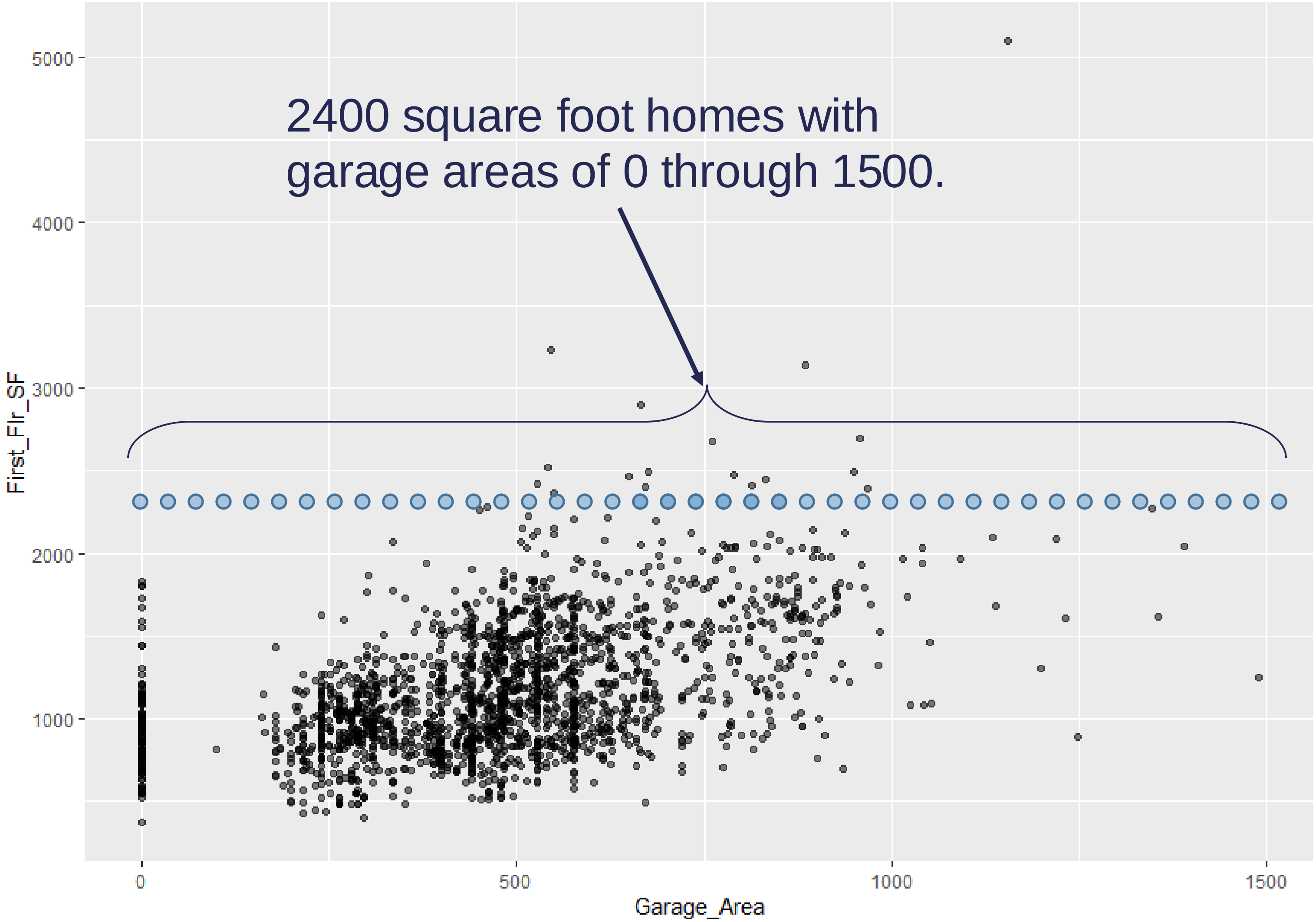
Partial Dependence Plot Problem

- Multicollinearity still a problem in the predictor variables.
- Conclusions are now based on simulated data with possibly some impossible data values.



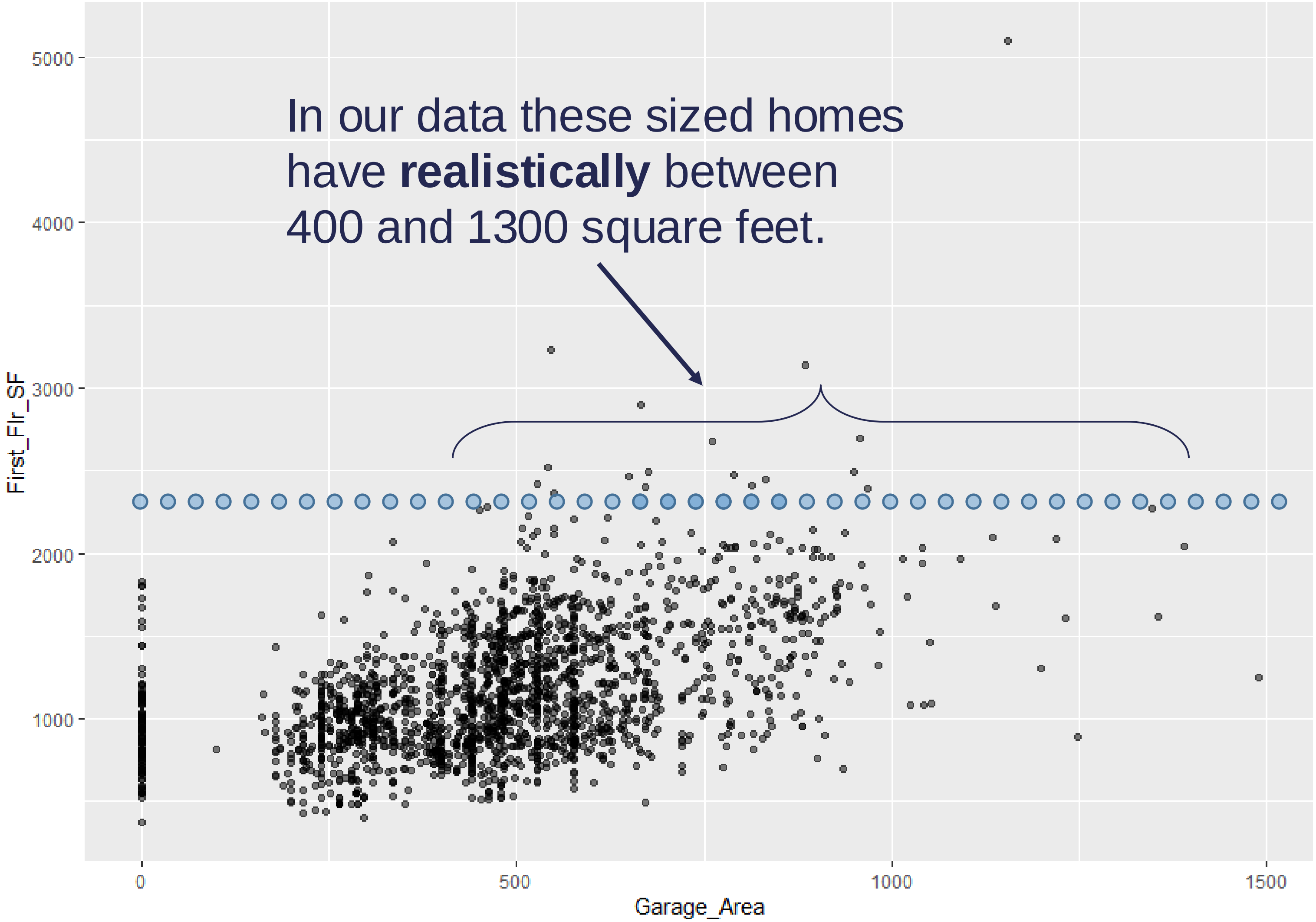
Partial Dependence Plot Problem

- Multicollinearity still a problem in the predictor variables.
- Conclusions are now based on simulated data with possibly some impossible data values.



Partial Dependence Plot Problem

- Multicollinearity still a problem in the predictor variables.
- Conclusions are now based on simulated data with possibly some impossible data values.



ALE vs. PD

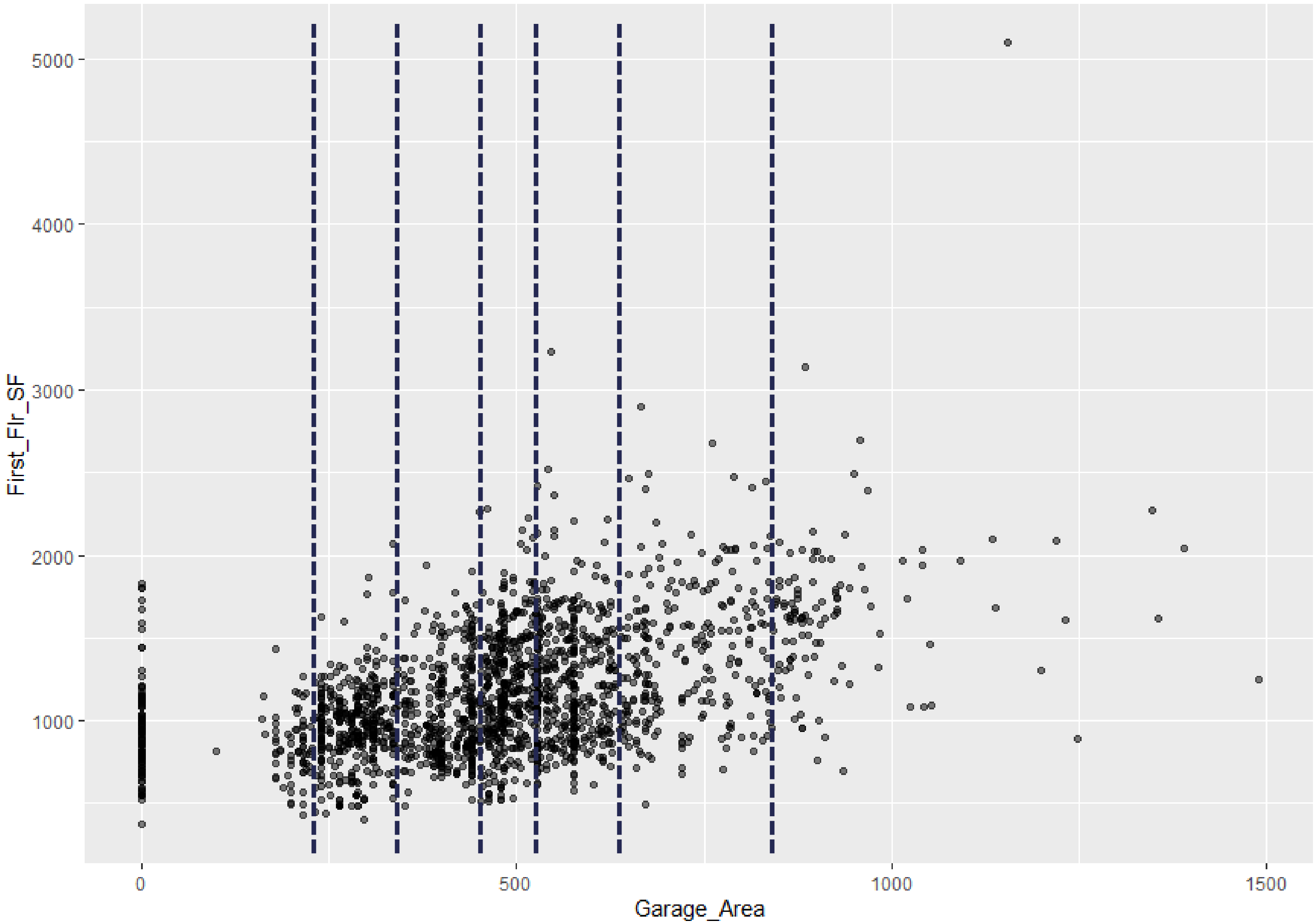
- The accumulated local effects (ALE) approach uses only reasonably contrived data to get a clearer picture of the relationship between a variable of interest and the target.

Previously used all possible values in range of the data…

Year Built	Central Air	Garage Area	Greater Living Area	…	Sale Price	Predicted Sale Price
2001	Y	0	1947	…	274,000	
2001	Y	1	1947	…	274,000	
2001	Y	2	1947	…	274,000	
…	…	…	…	…	…	…
2001	Y	1488	1947	…	274,000	

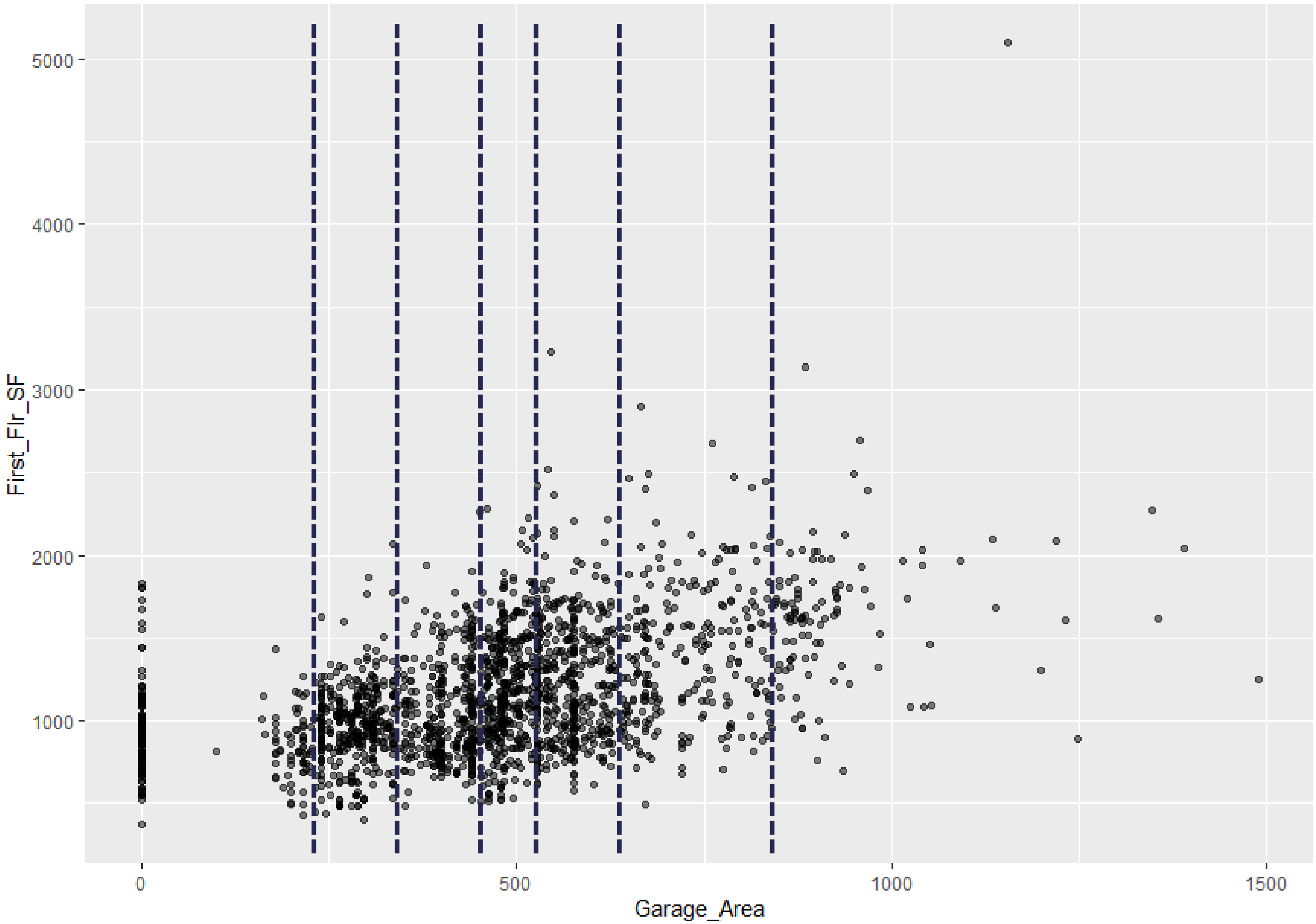
ALE vs. PD

- ALE uses quantiles of your data (by default) to define reasonable range.
- Uses quantiles to get approximately same number of observations in each group.



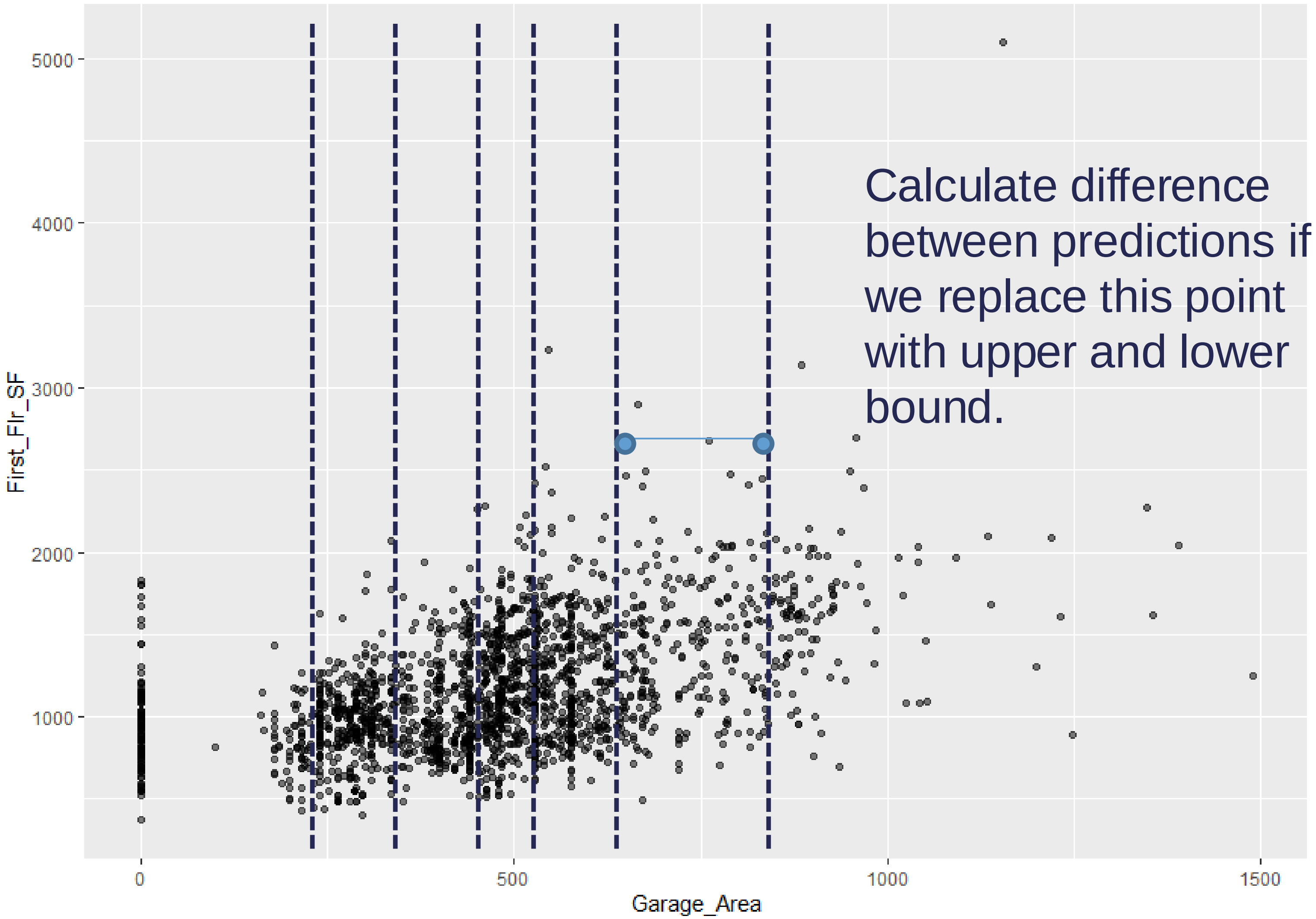
ALE vs. PD

- For observations in each interval, determine how much their prediction would change if we replace the feature of interest with the upper and lower bounds of the interval (keeping all other variables constant).



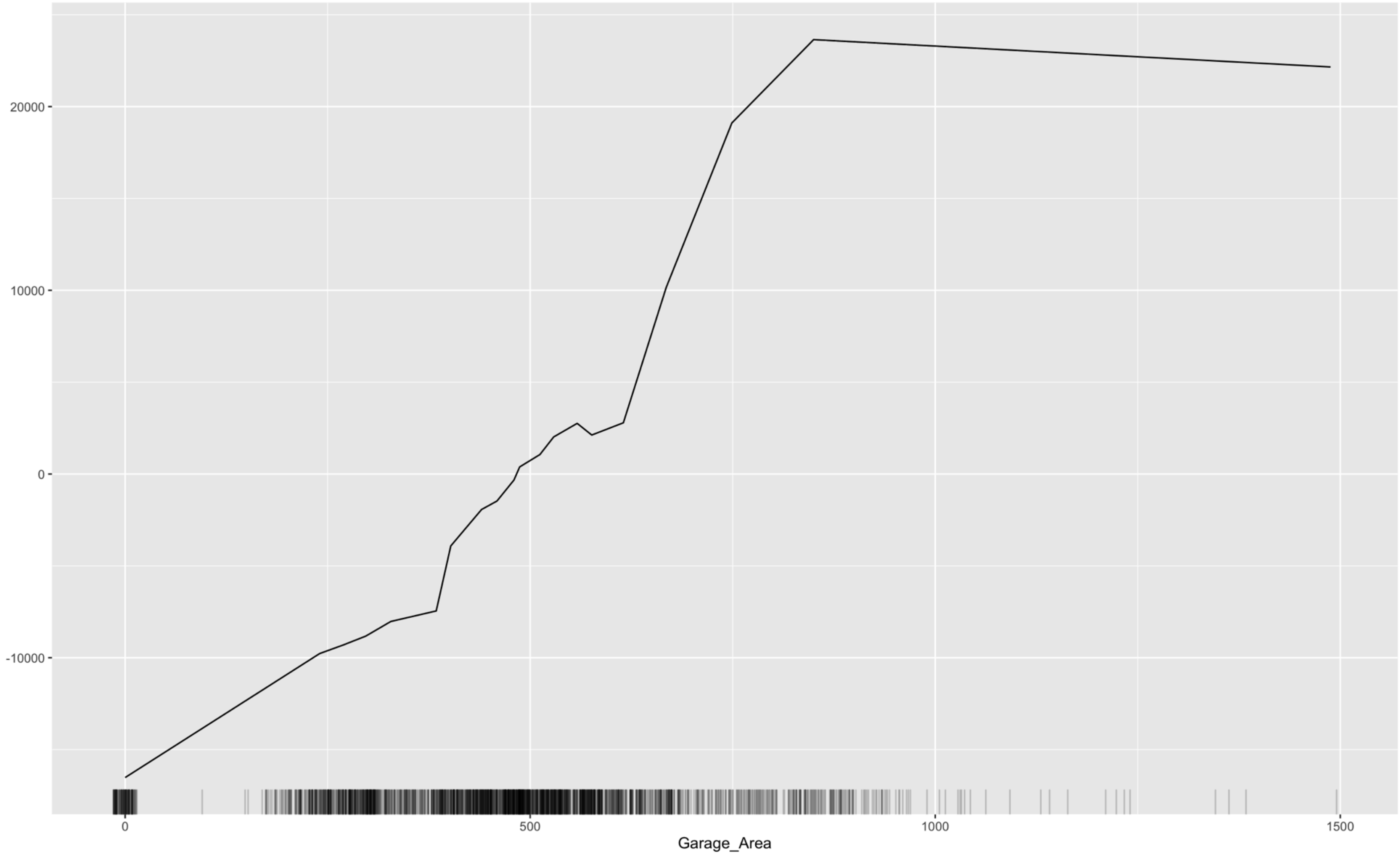
ALE vs. PD

- For observations in each interval, determine the difference in prediction if the variable of interest is set at the upper vs. lower bound of this interval (keeping all other variables constant).



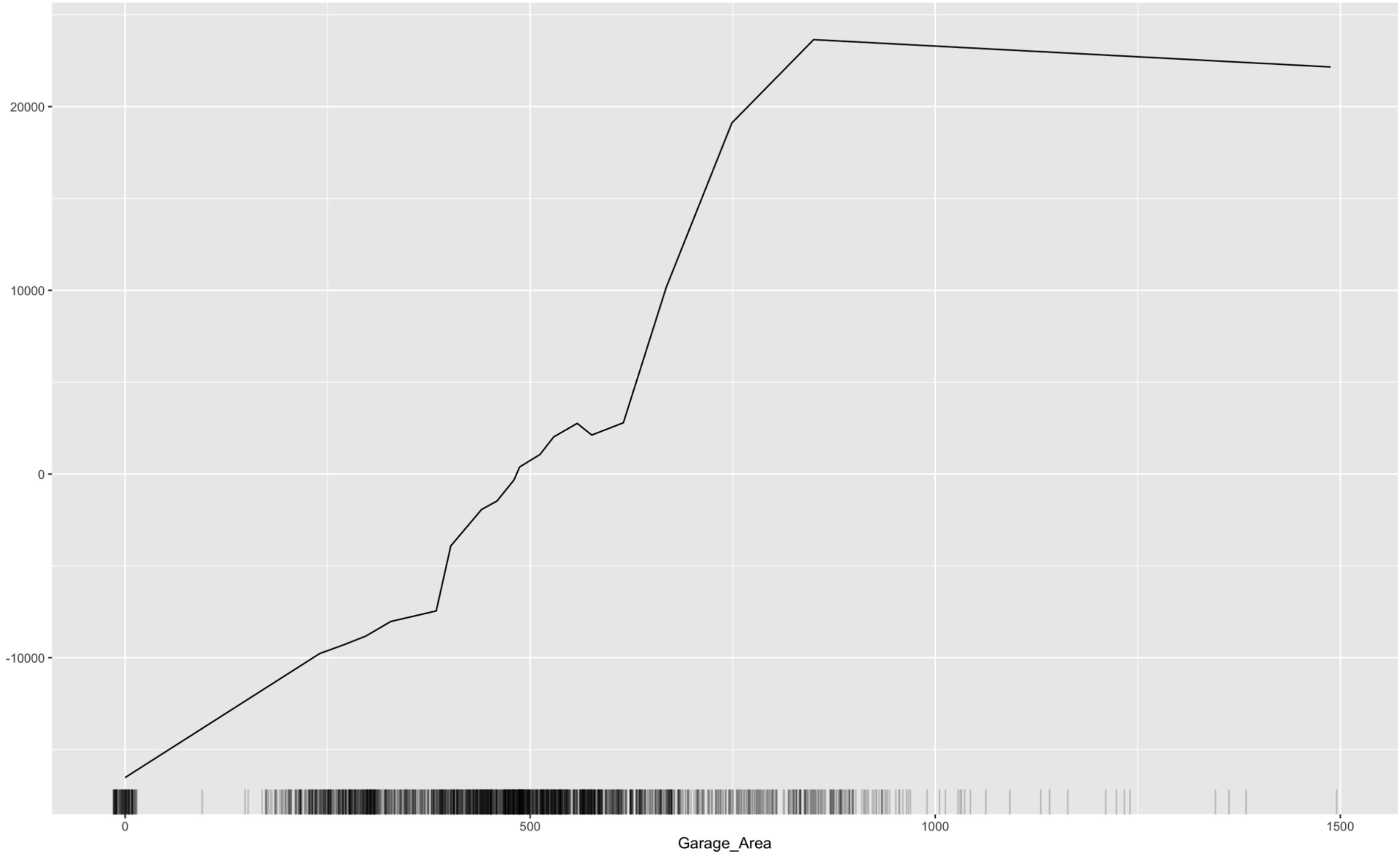
ALE vs. PD

- Do this for all observations in the interval.
- Accumulate these changes and center them to form the ALE curve.



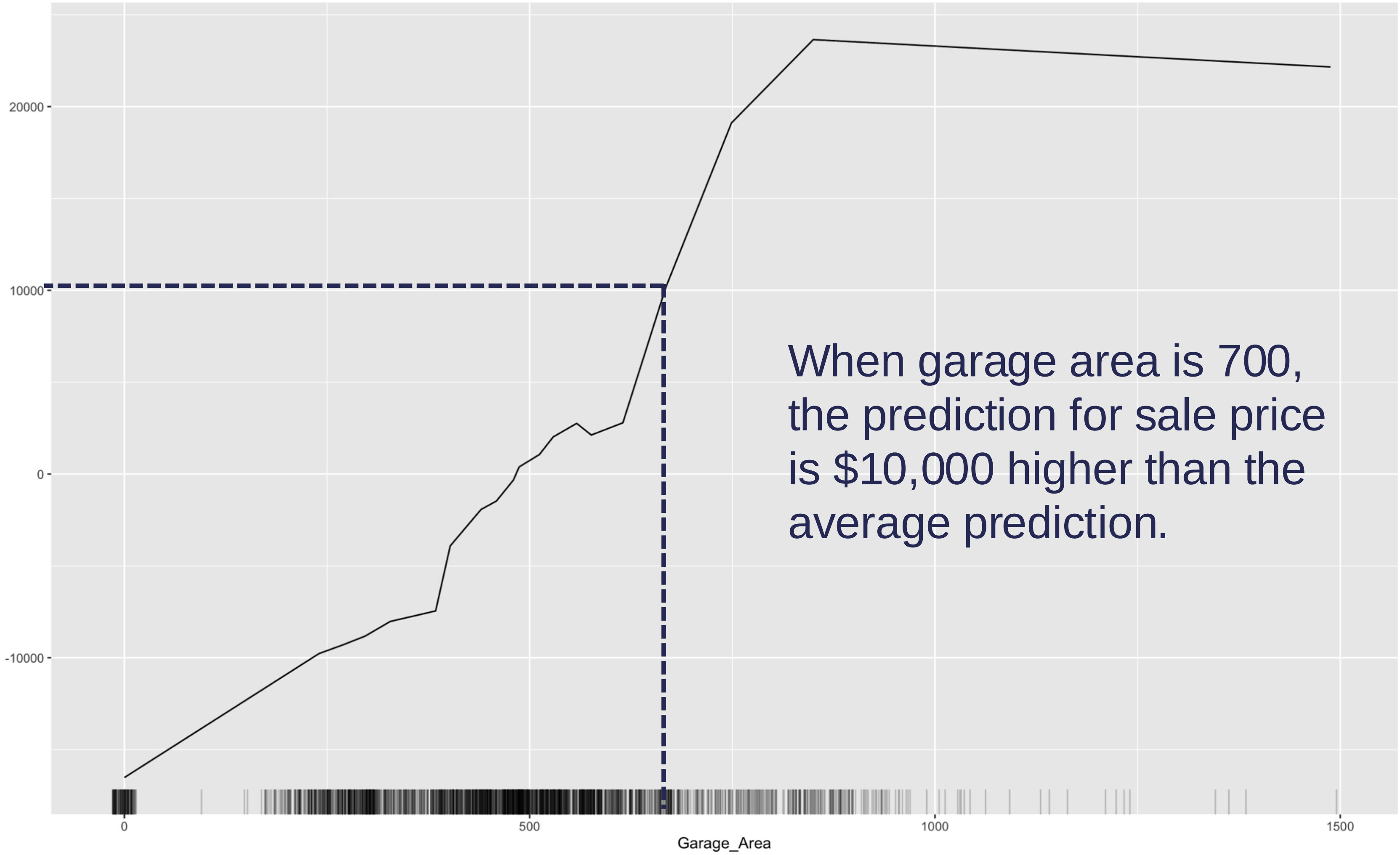
ALE Curve

- ALE curve describes the main effect of the input variable **compared to the data's average prediction**.



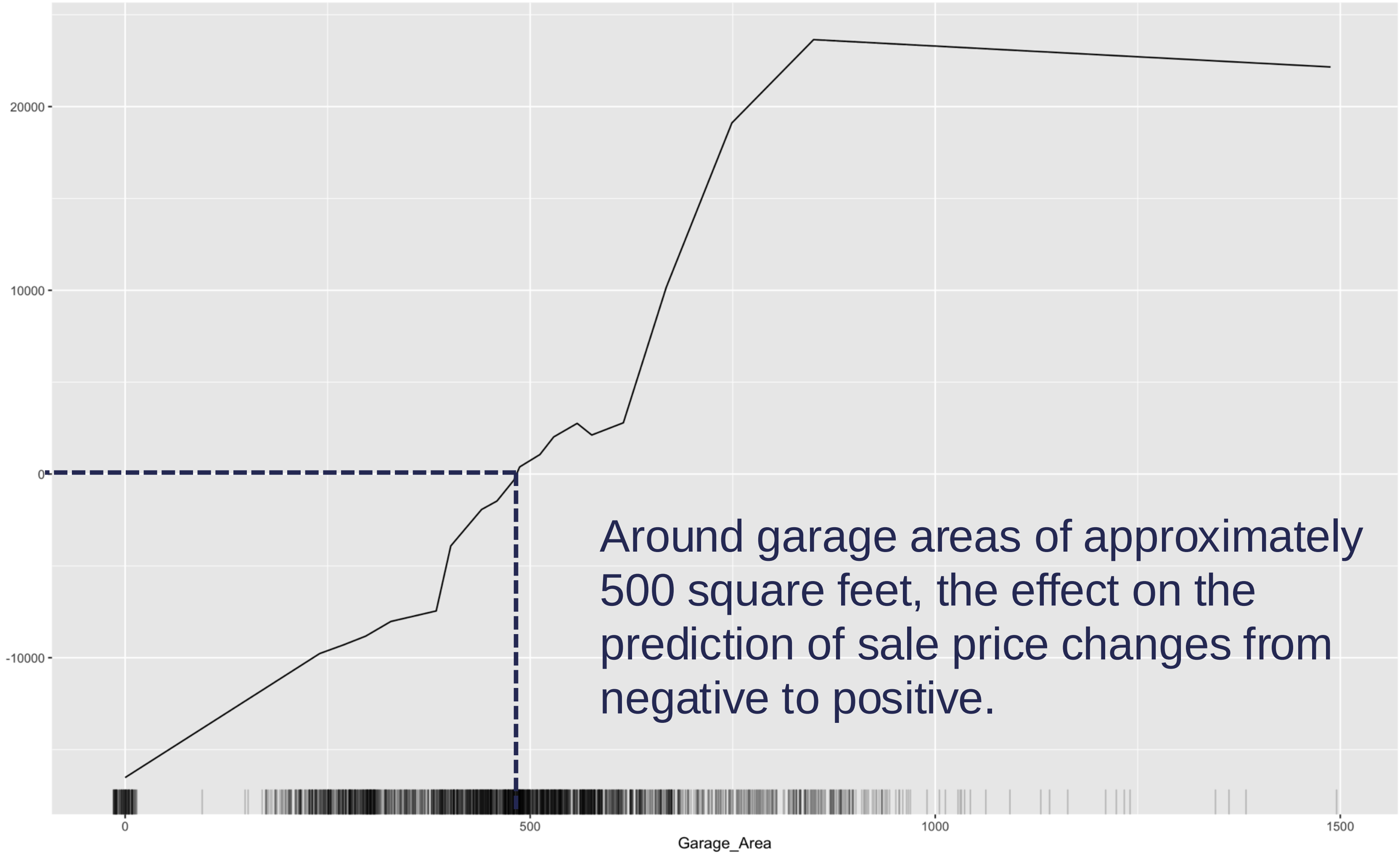
ALE Curve

- ALE curve describes the main effect of the input variable **compared to the data's average prediction**.



ALE Curve

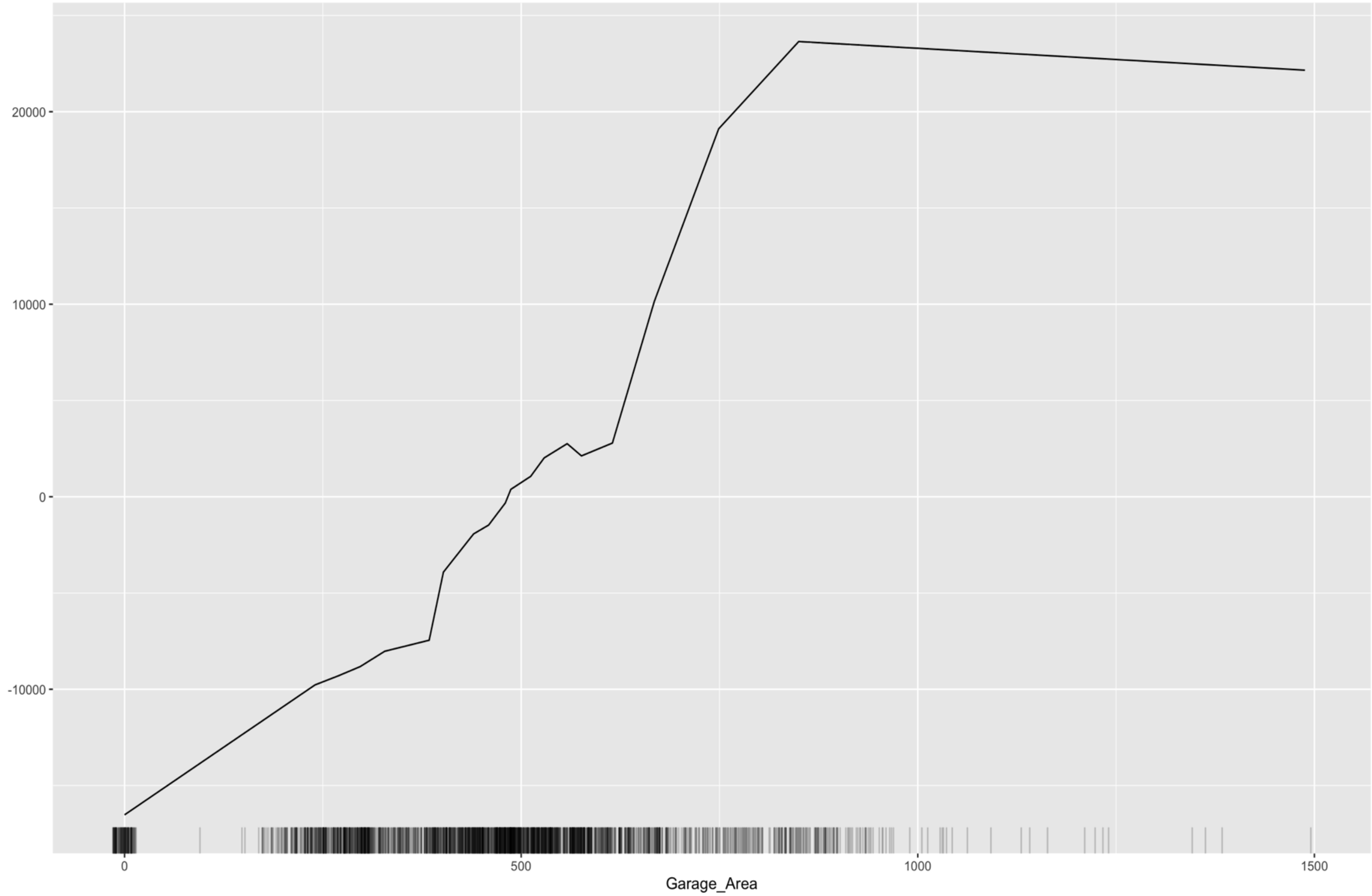
- ALE curve describes the main effect of the input variable **compared to the data's average prediction**.
- Changes from negative to positive (or vice versa) are also important.



ALE Curve

```
set.seed(12345)
forest_pred <- Predictor$new(rf.ames,
                data = training[,-1],
                y = training$Sale_Price,
                type = "response")
```

```
ale_plot <- FeatureEffects$new(
             forest_pred,
             method = "ale")

ale_plot$plot(c("Garage_Area"))
```

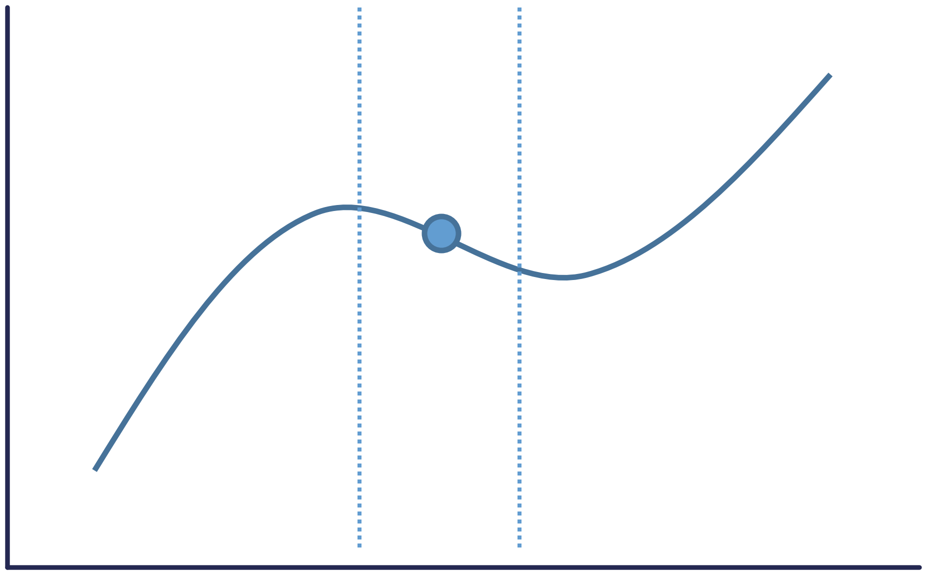


LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME)

Model Agnostic Interpretability

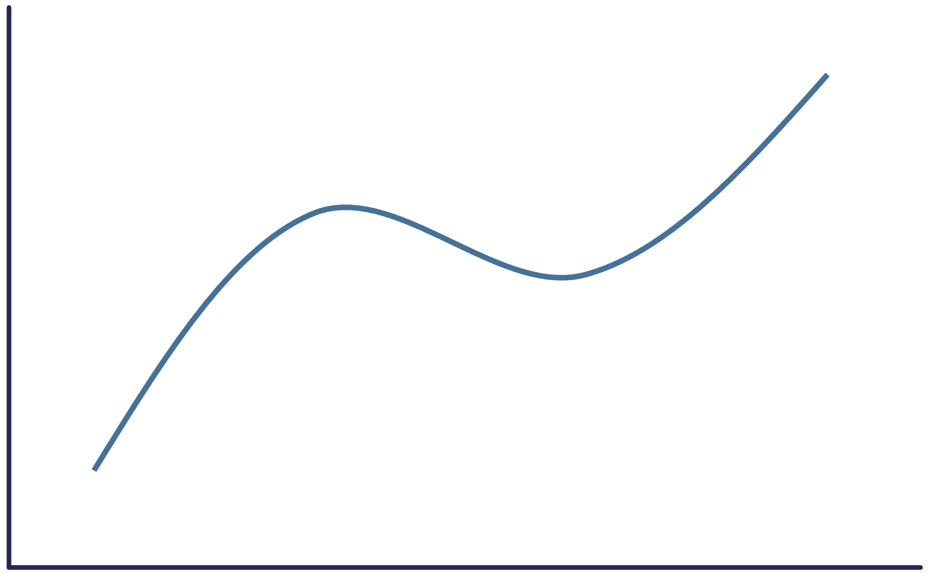
Local

- ICE
- **LIME**
- Shapley Values



Global

- Permutation Importance
- Partial Dependence
- ALE

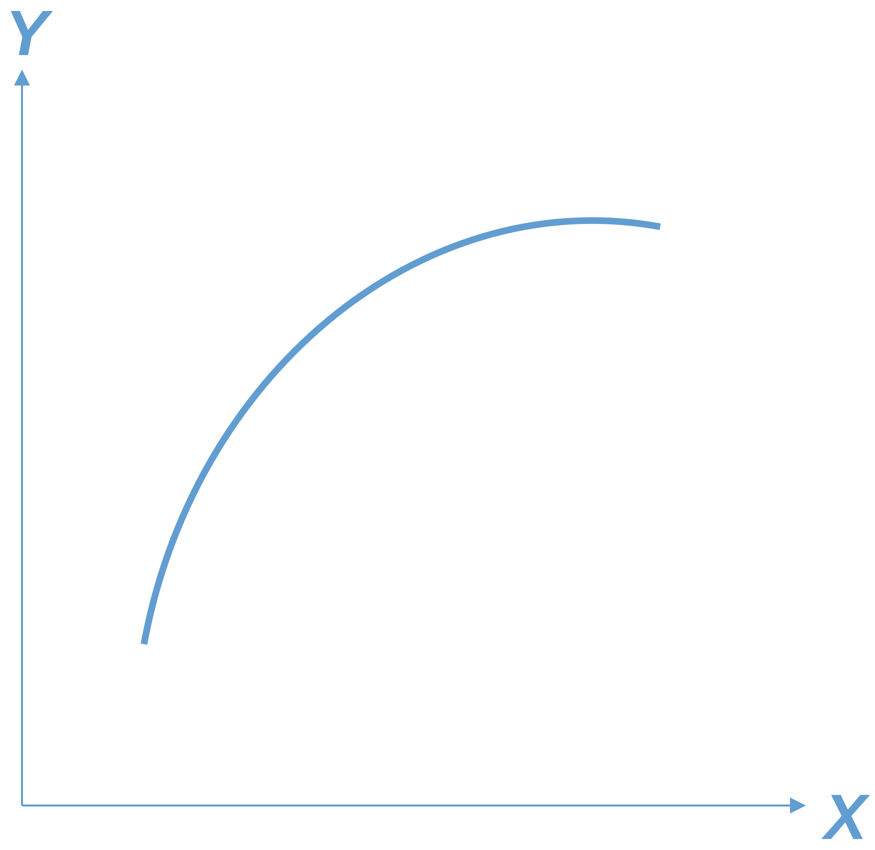


Local Interpretable Model-Agnostic Explanations

- General Idea:
  - "Let me show you a linear model that could explain the exact orientation of the predictive model at a specific point."

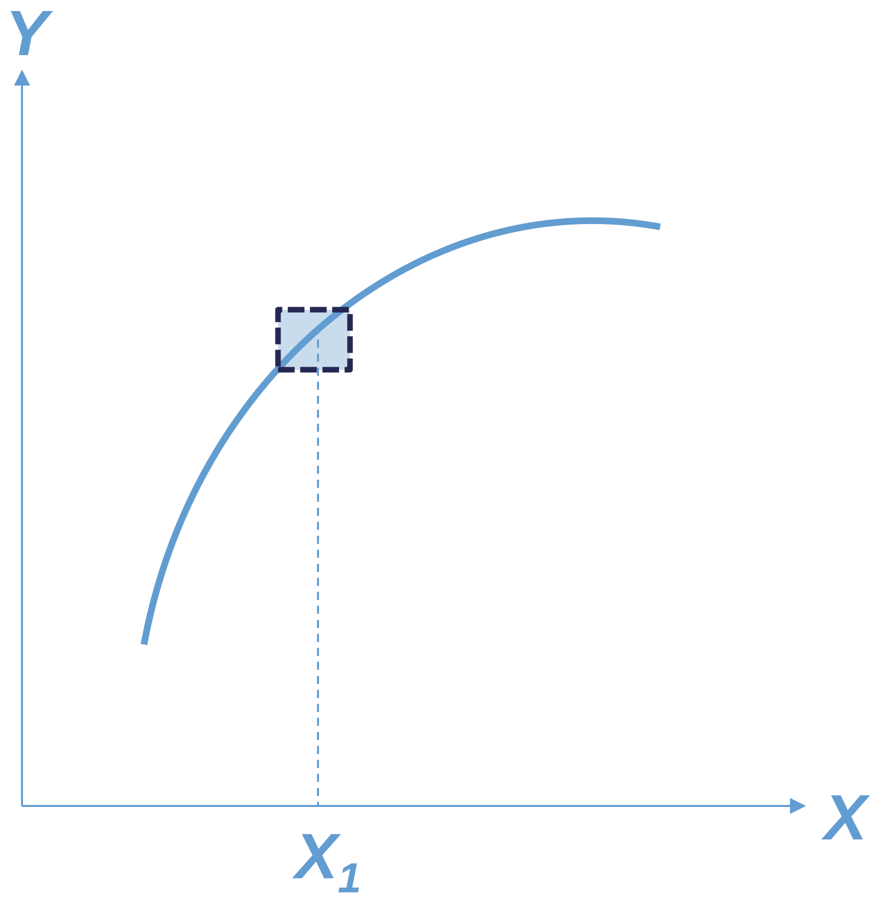
LIME Intuition

- Imagine that you had a nonlinear relationship between a target and a predictor variable.



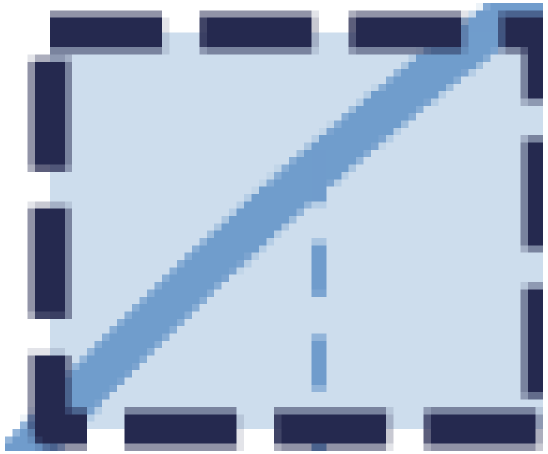
LIME Intuition

- Imagine that you had a nonlinear relationship between a target and a predictor variable.
- Zoom in **REALLY** close to a specific point of interest…



LIME Intuition

- Imagine that you had a nonlinear relationship between a target and a predictor variable.
- Zoom in **REALLY** close to a specific point of interest…
- Looks (approximately) like a straight line!
- We can model straight lines with linear regression → we can understand the predictive model **at that exact point**.



LIME Process

- We can model straight lines with linear regression → we can understand the predictive model **at that exact point**.
- But how…?
- Here are the basic steps to what LIME is doing:

1. Randomly generate values (Normally distributed) for each variable in the model.
2. Weight more heavily the fake observations that are **near the real observation of interest**.
3. Build a **weighted** linear regression model based on fake observations and observation of interest.
4. “Interpret” coefficients of variables from linear regression model.

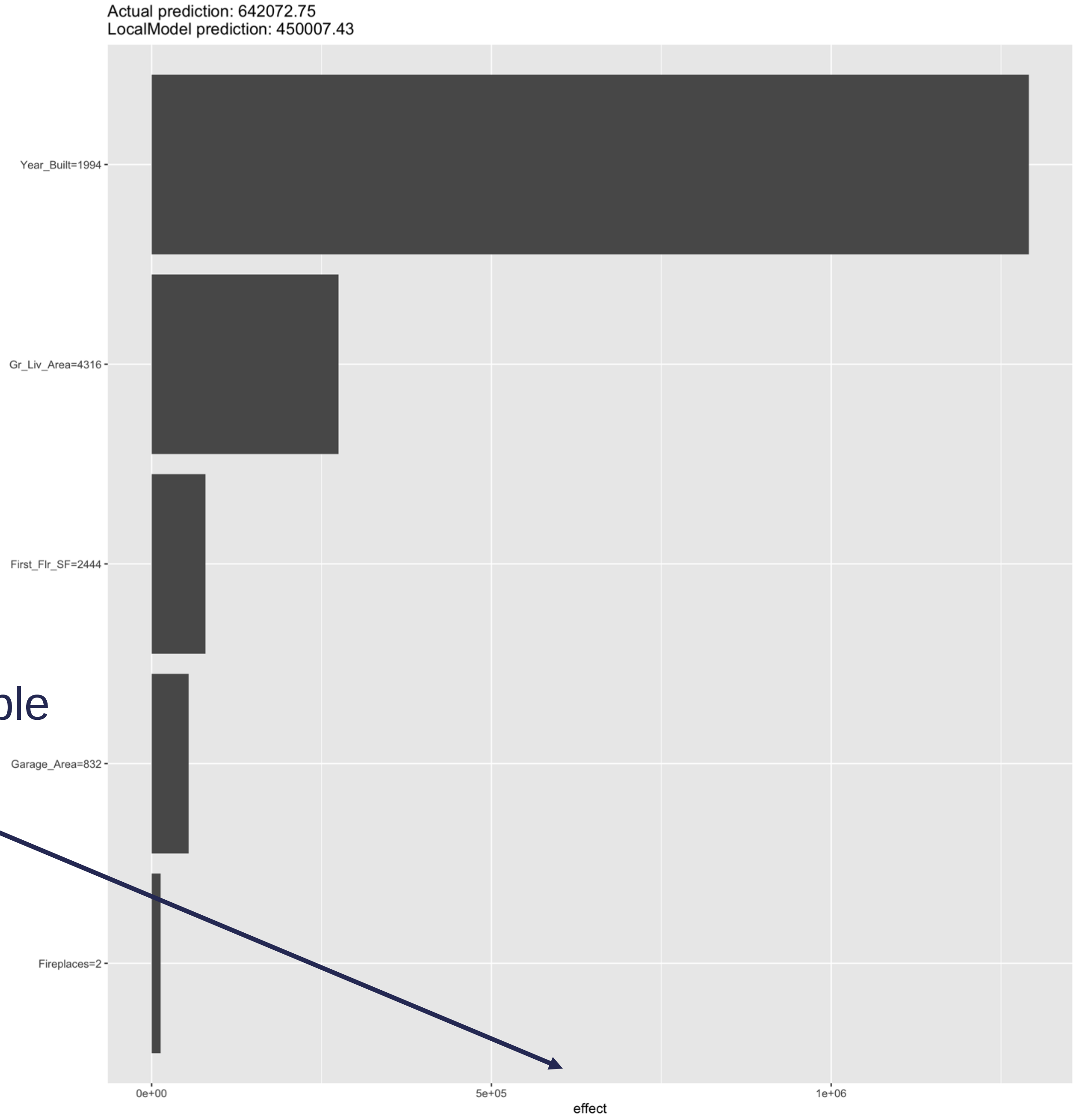
LIME Details

- LIME not actually limited to linear regression… could use any interpretable model (decision tree for example).
- **Choices** → how much explanation do you want your model to have (aka, number of variables to use in explainable model)?
- LIME commonly used for small local models (not a lot of variables).

- The definition of “near the points of interest” is a **very big and unsolved problem** in the world of LIME.

LIME

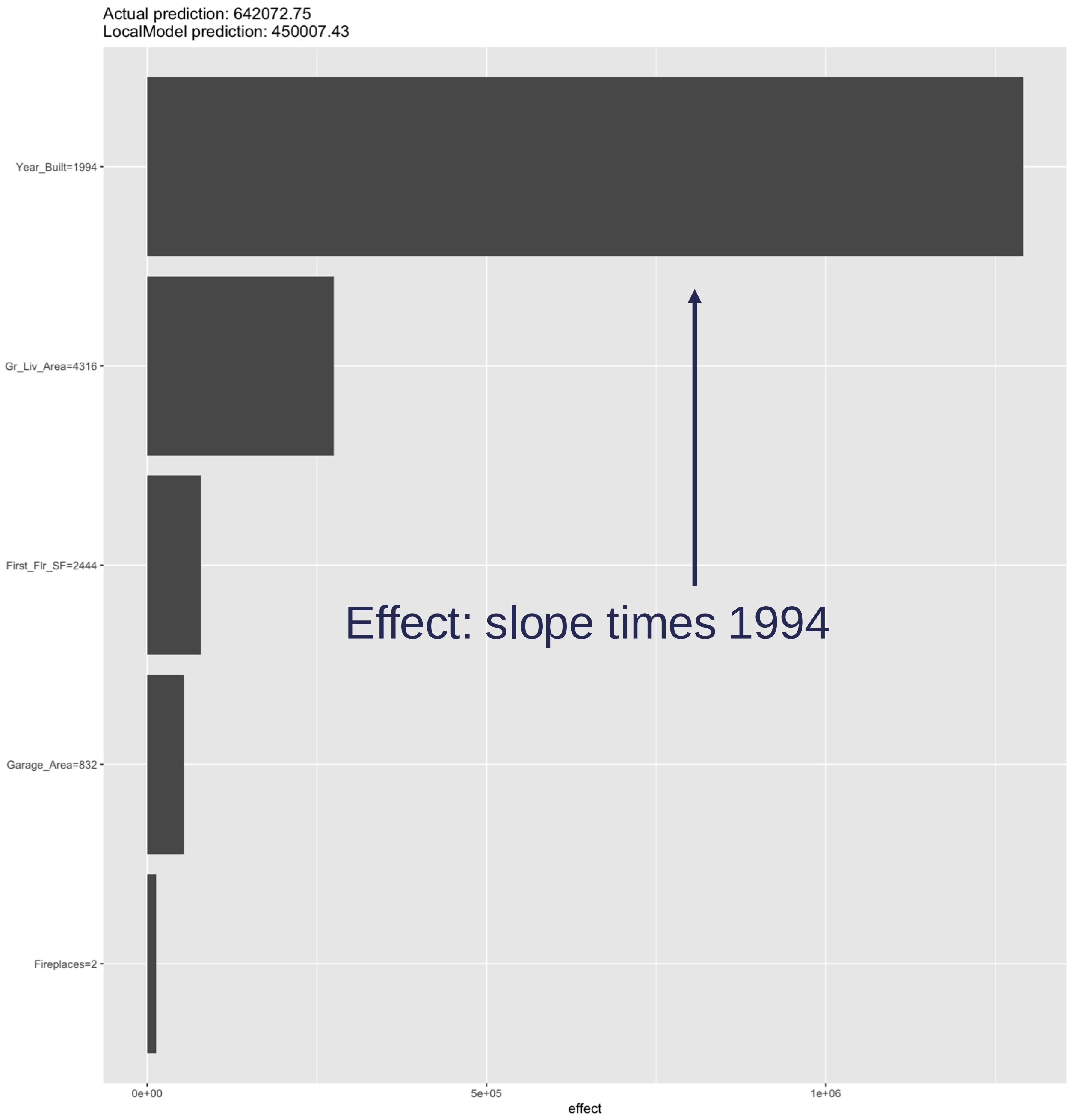
```
point <- 1328

lime.explain <- LocalModel$new(forest_pred,
                 x.interest = training[point,-1],
                 k = 5)

plot(lime.explain)
```



Effect: Total impact of that variable calculated as slope multiplied by the value of the variable.

LIME

```
point <- 1328

lime.explain <- LocalModel$new(forest_pred,
                  x.interest = training[point,-1],
                  k = 5)

plot(lime.explain)
```



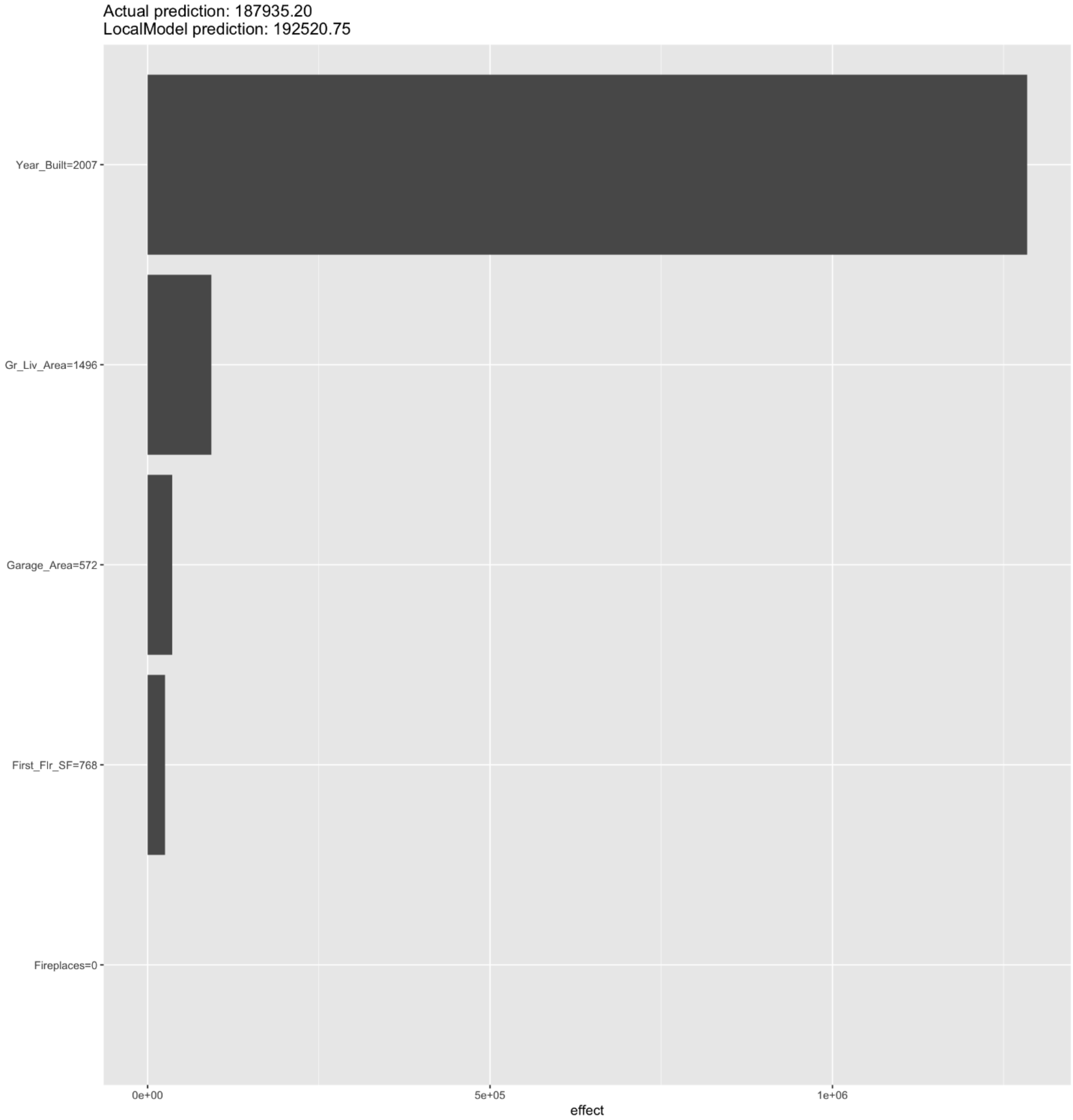
LIME

```r
point <- 1328

lime.explain <- LocalModel$new(forest_pred,
                x.interest = training[point,-1],
                k = 5)

lime.explain$results
```

```
                   beta x.recoded     effect x.original      feature     feature.value
Year_Built    647.21726      1994 1290551.21       1994   Year_Built   Year_Built=1994
First_Flr_SF   32.40725      2444   79203.32       2444 First_Flr_SF First_Flr_SF=2444
Fireplaces   6563.12201         2   13126.24          2   Fireplaces      Fireplaces=2
Garage_Area    65.42648       832   54434.83        832  Garage_Area   Garage_Area=832
Gr_Liv_Area    63.81149      4316  275410.39       4316  Gr_Liv_Area  Gr_Liv_Area=4316
```

LIME

```
point <- 1000

lime.explain <- LocalModel$new(forest_pred,
                               x.interest = training[point,-1],
                               k = 5)

plot(lime.explain)
```

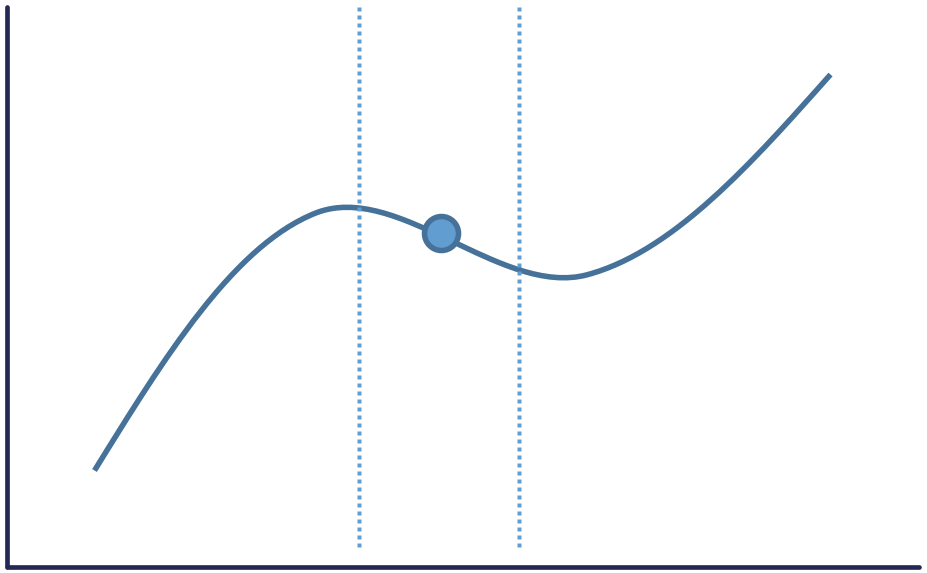


SHAPLEY VALUES

Model Agnostic Interpretability

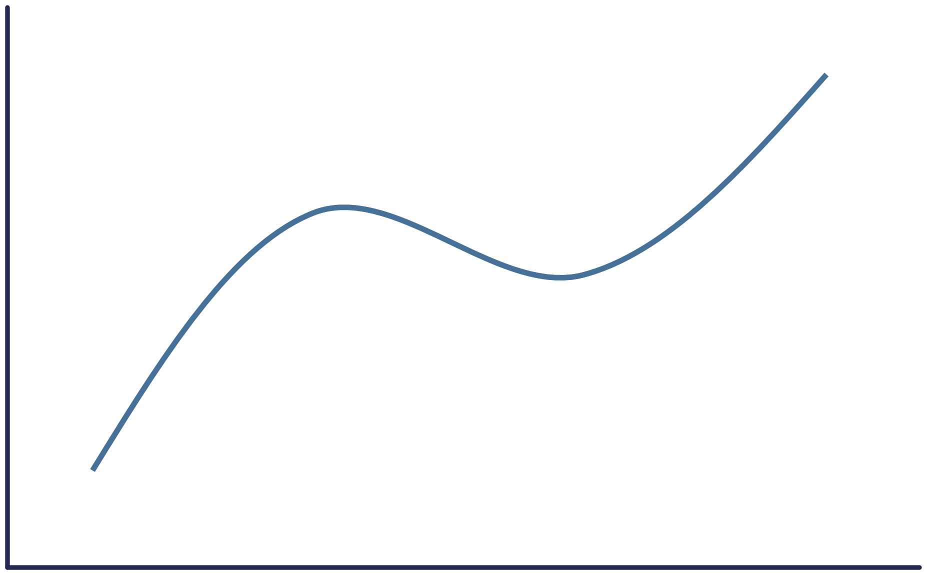
Local

- ICE
- LIME
- **Shapley Values**



Global

- Permutation Importance
- Partial Dependence
- ALE

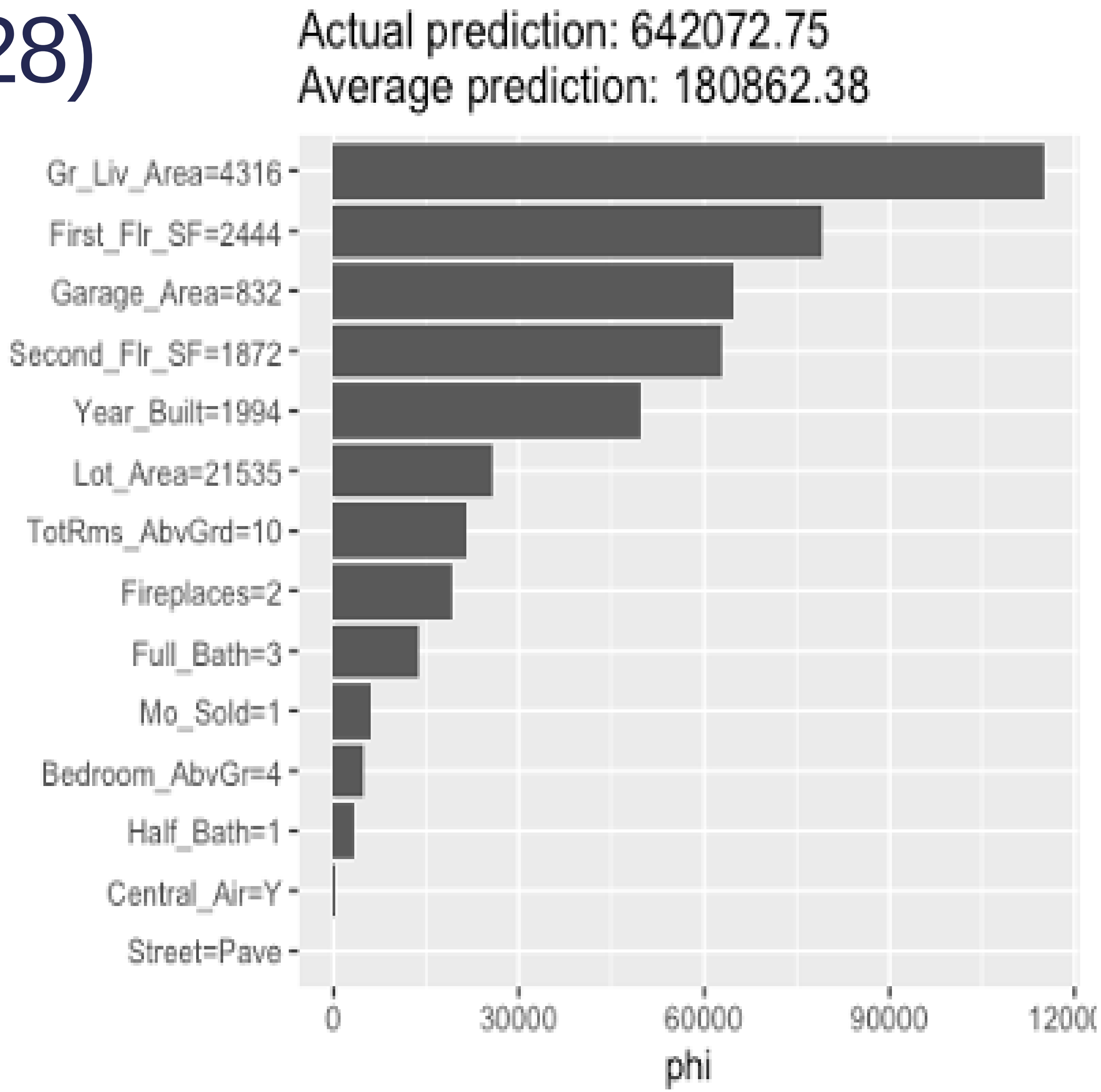


Shapley Values

- General Idea:
  - “The value of the j^{th} feature contributed … to the prediction of this particular instance compared to the average prediction for the dataset.”

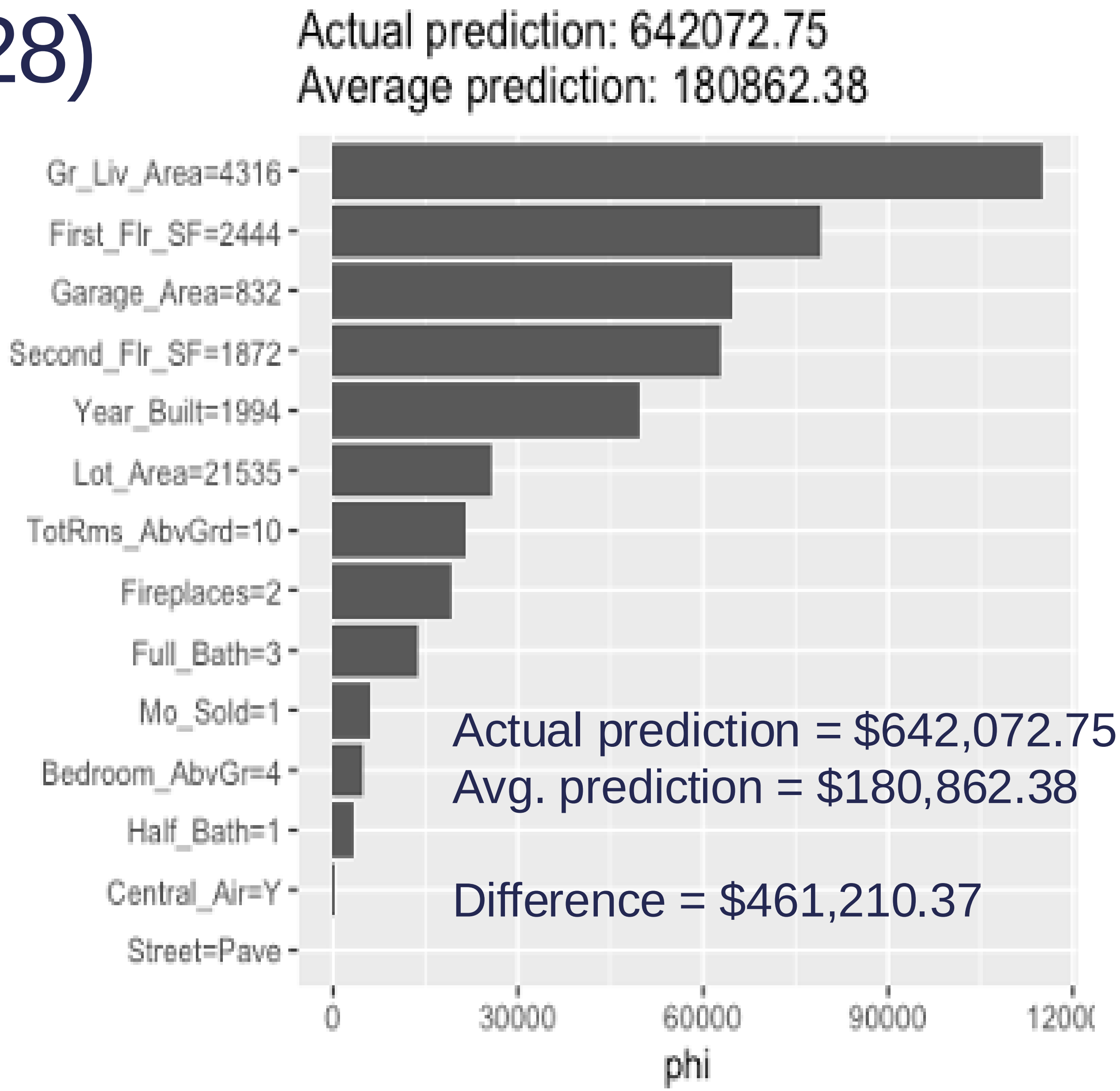
Shapley Values (obs. 1328)

- General Idea:
  - "The value of the j^{th} feature contributed … to the prediction of this particular instance compared to the average prediction for the dataset."
  - In other words… how do I get from the average prediction to the actual prediction and **WHY!**



Shapley Values (obs. 1328)

- General Idea:
  - "The value of the j^{th} feature contributed … to the prediction of this particular instance compared to the average prediction for the dataset."
  - In other words… how do I get from the average prediction to the actual prediction and **WHY!**



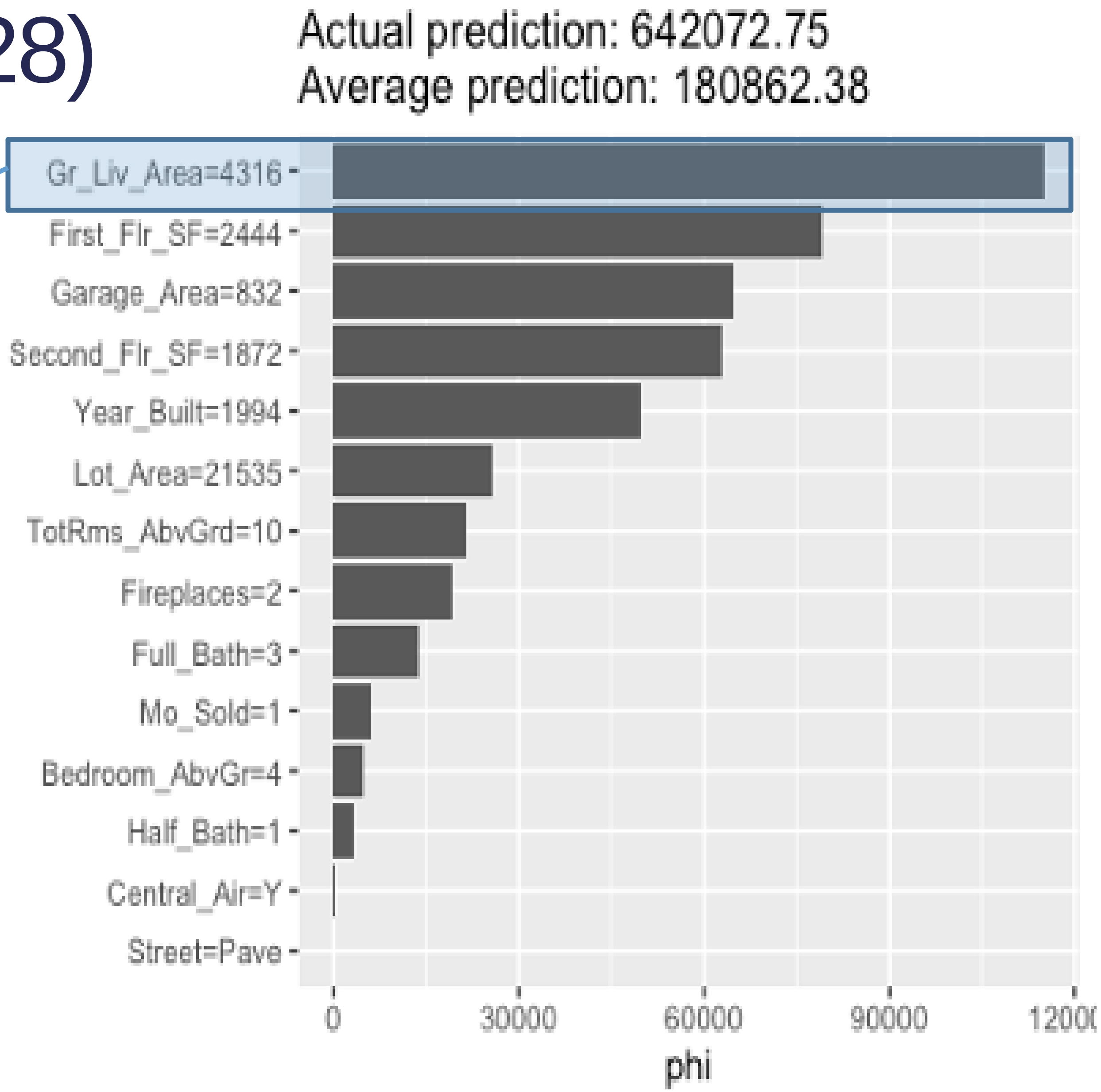
Actual prediction = $642,072.75
Avg. prediction = $180,862.38

Difference = $461,210.37

Shapley Values (obs. 1328)

- **Difference = $461.210.37**

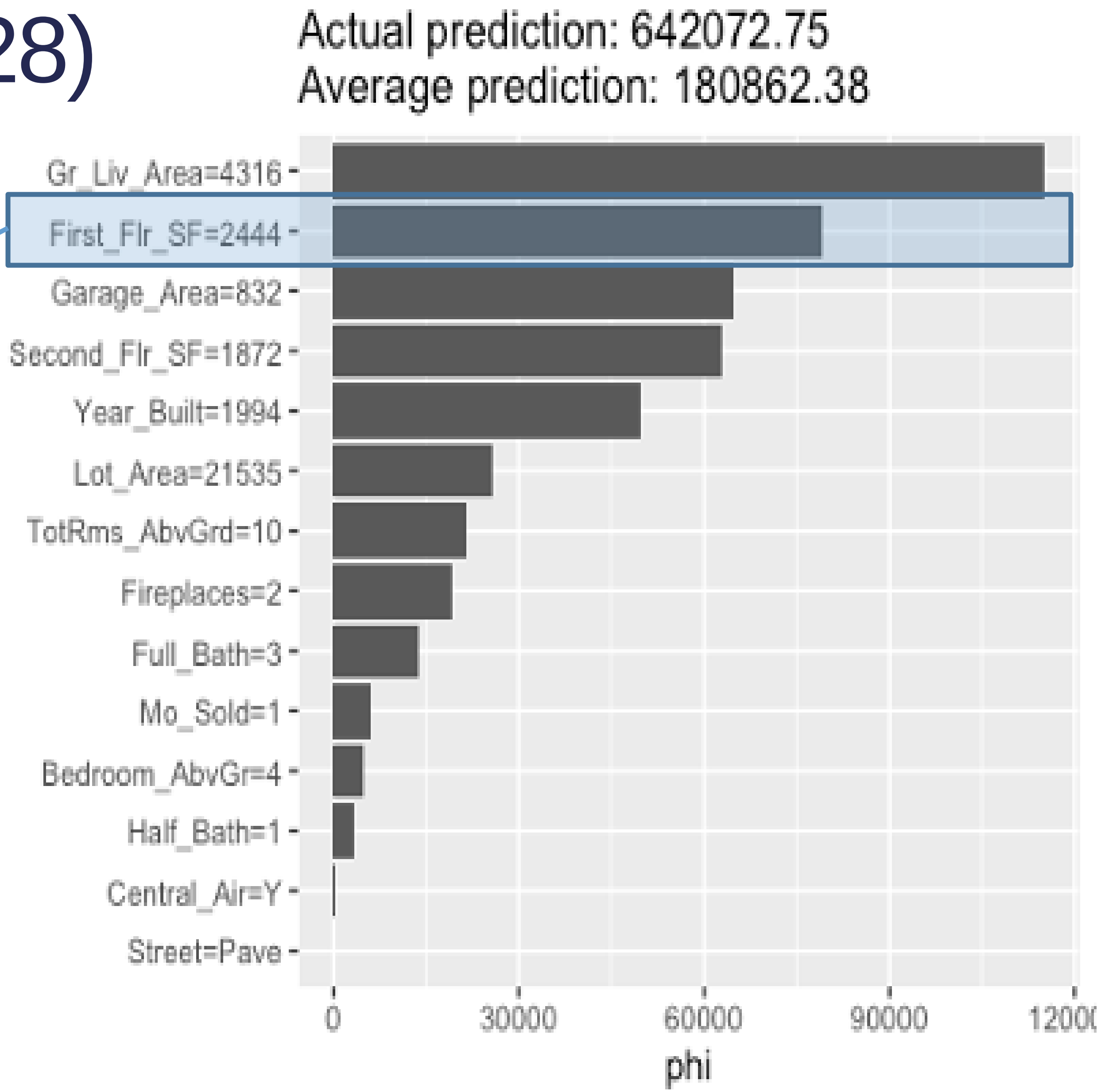
= $114,137.72 +



Shapley Values (obs. 1328)

- **Difference = $461.210.37**

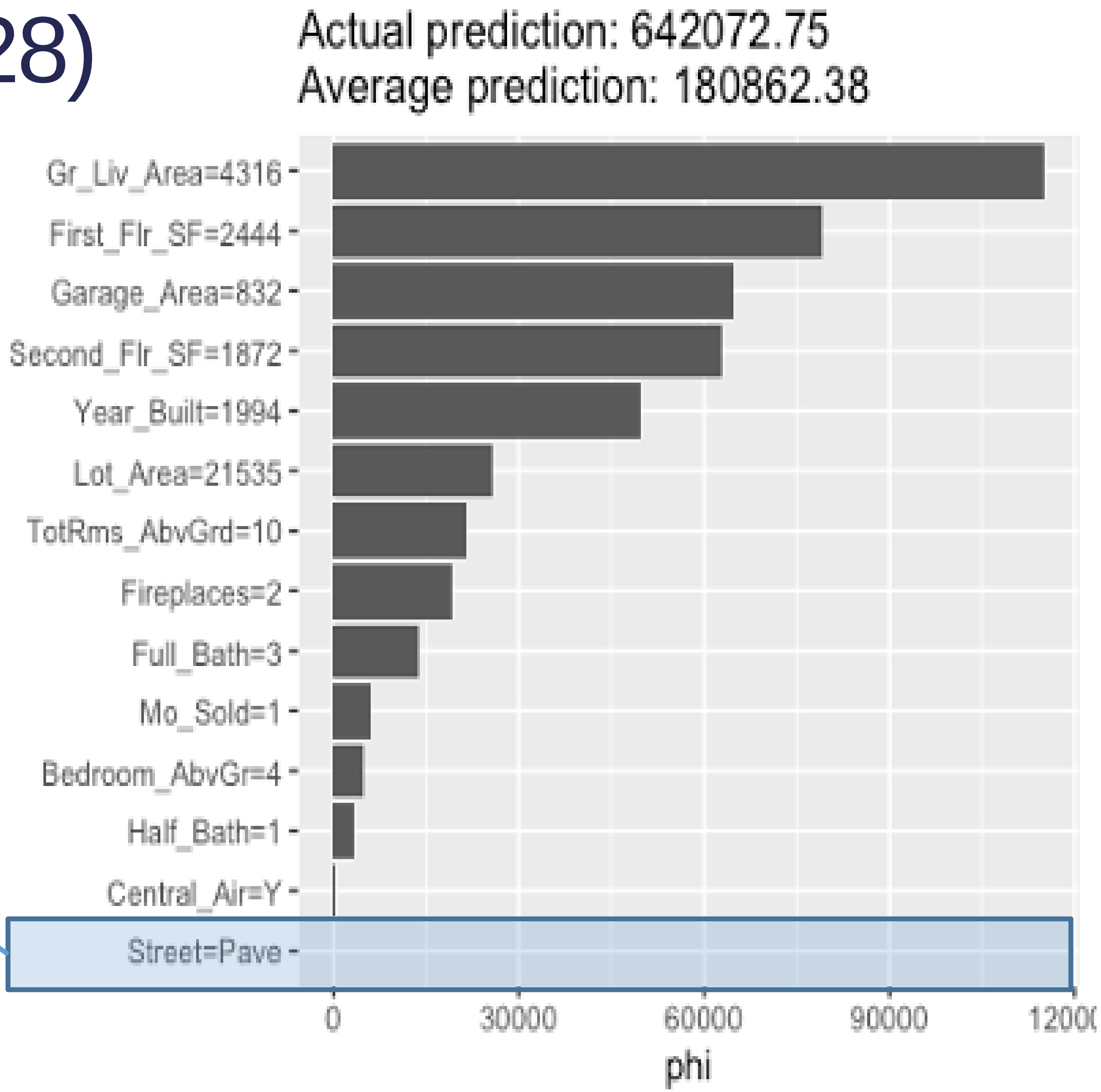
  = $114,137.72 +
  $78,440.11 +



Shapley Values (obs. 1328)

- **Difference = $461.210.37**

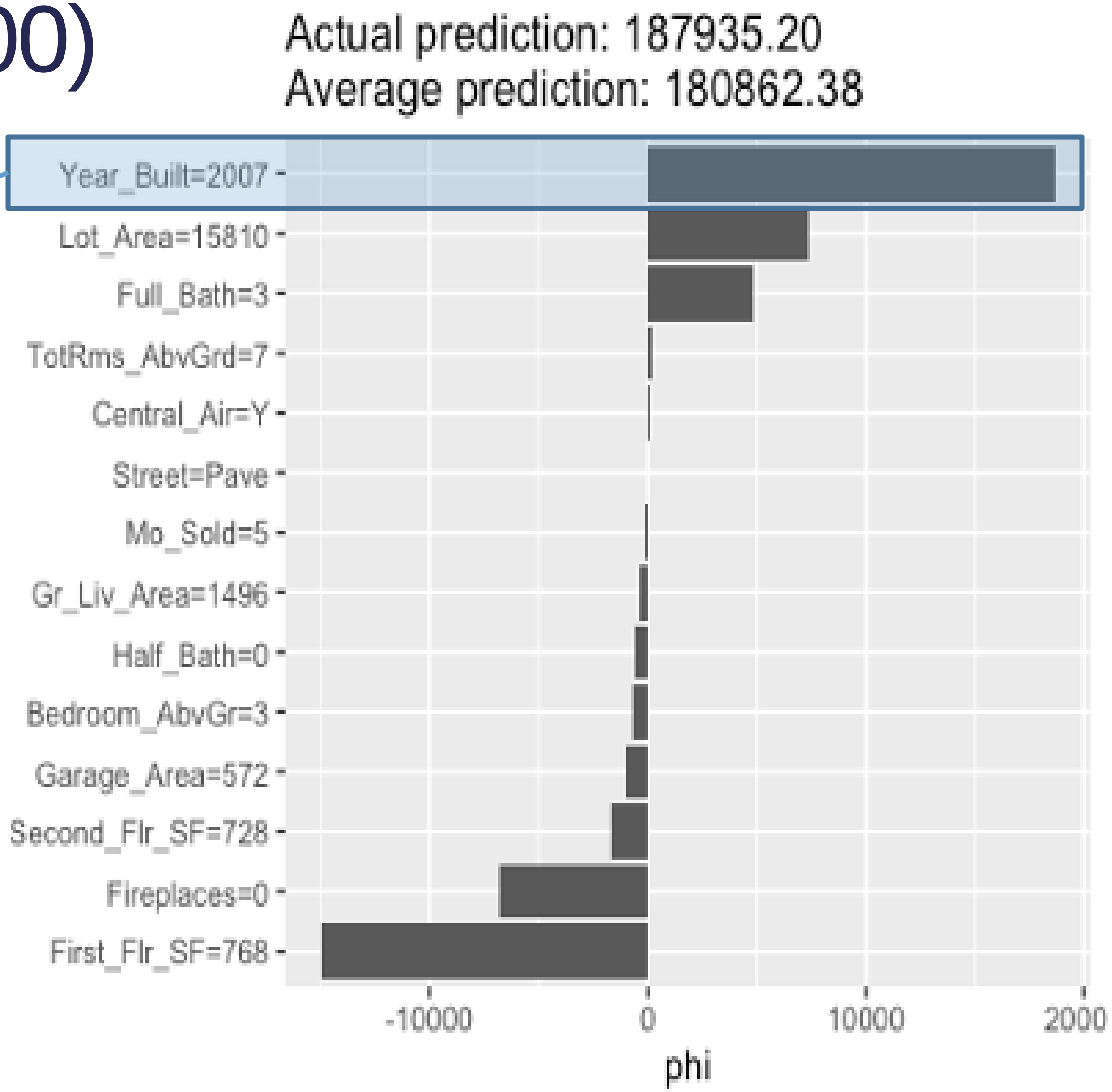
  = $114,137.72 +
  $78,440.11 +
  …
  $0



Shapley Values (obs. 1000)

- **Difference = $7,072.82**

  = $21,206.09 +
  $6,081.23 +
  …
  -$14,590.45



Computation

- Neat idea, but how does it work? **Game theory!**
- Shapley (1953) assigned a payout value for players depending on their contribution to the total payout across the coalition (think team).
  - In other words, you are a team of players (say basketball)…
  - You win some money at a local tournament…
  - Do you split it evenly? Maybe…
  - Or you split it based on contribution! Star player gets most, next best player gets second highest, and so on…
- That is the idea of a Shapley value in game theory!

Computation

- Neat idea, but how does it work? **Game theory!**
- Shapley (1953) assigned a payout value for players depending on their contribution to the total payout across the coalition (think team).
- That is the idea of a Shapley value in game theory!
- The Shapley value in machine learning is the average marginal contribution of a feature (teammate) across all possible coalitions of variables (collections of teammates).

Computation

- Neat idea, **but how does it work?**
- Need to compute the average change in the prediction that a set of variables experiences when the variable of interest is added.

Computation

- Neat idea, **but how does it work?**
- Need to compute the average change (**across all observations**) in the prediction that a set of variables (**across all combinations**) experiences when the variable of interest is added.

- VERY TIME CONSUMING (ALMOST IMPOSSIBLE) TO DO WITH LARGE NUMBER OF VARIABLES!

Computation

- Neat idea, **but how does it work?**
- Need to compute the average change (**across all observations**) in the prediction that a set of variables (**across all combinations**) experiences when the variable of interest is added.

- VERY TIME CONSUMING (ALMOST IMPOSSIBLE) TO DO WITH LARGE NUMBER OF VARIABLES!

- Lot's of research into shortcuts for this computation → sampling… subsets of variables… etc.

Advantages

- **Efficiency** – variable contributions must sum to the difference of prediction for point of interest compared to the average.
- **Symmetry** – contributions of two variables (j and k) should be the same if they contribute equally to all possible combinations of variables (coalitions).
- **Dummy** – a variable that does not change the predicted value, for any combination of variables, should have a Shapley value of 0.
- **Additivity** – for a forest of trees, the Shapley value of the forest for a given observation should be the average of the Shapley values for each tree at that given point.

Disadvantages

- Some people look at distributions of all Shapley values across a variable to measure “overall impact” of a variable.
- BE VERY CAREFUL OF THIS!
- Shapley values were never designed for this as they are created for local interpretations, not global ones.
- Only thing you **might be able to do** is to see if all Shapley values are positive or negative.

Shapley Values (obs. 1328)

```r
point <- 1328

shap <- Shapley$new(forest_pred,
         x.interest = training[point,-1])

shap$plot()
```