Introduction to Interpretable Machine Learning

Dean Allsopp

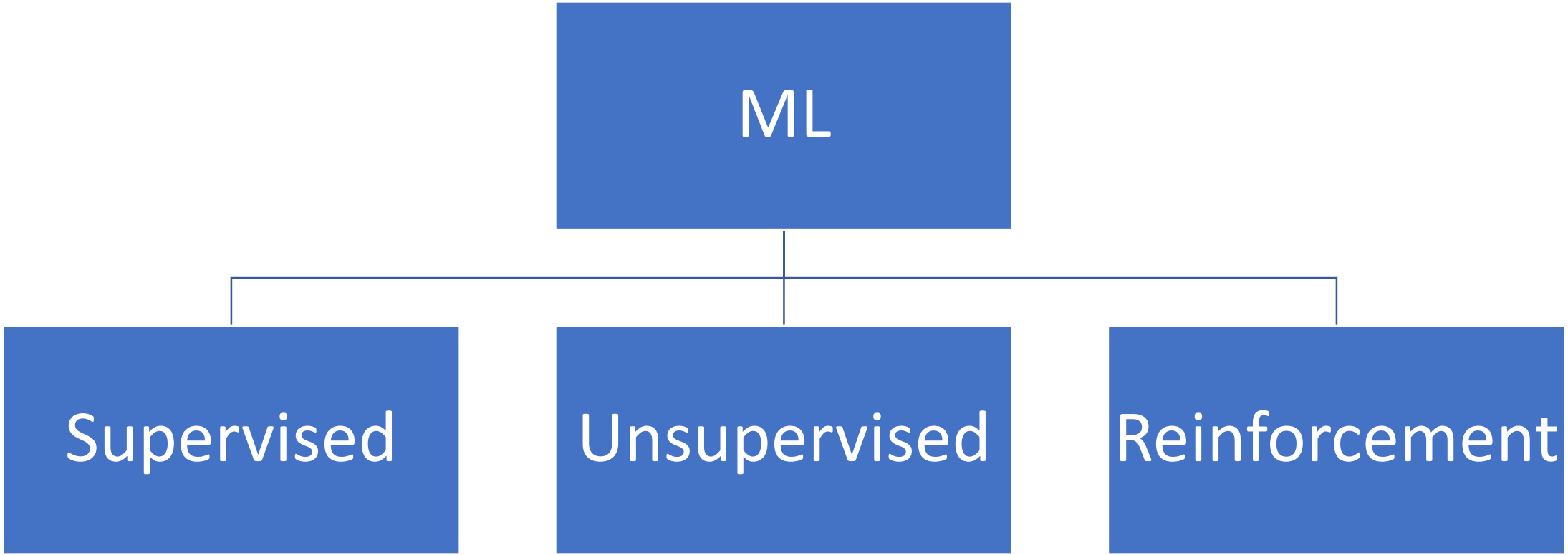
Global AI Bootcamp

December 15, 2018

Machine Learning?

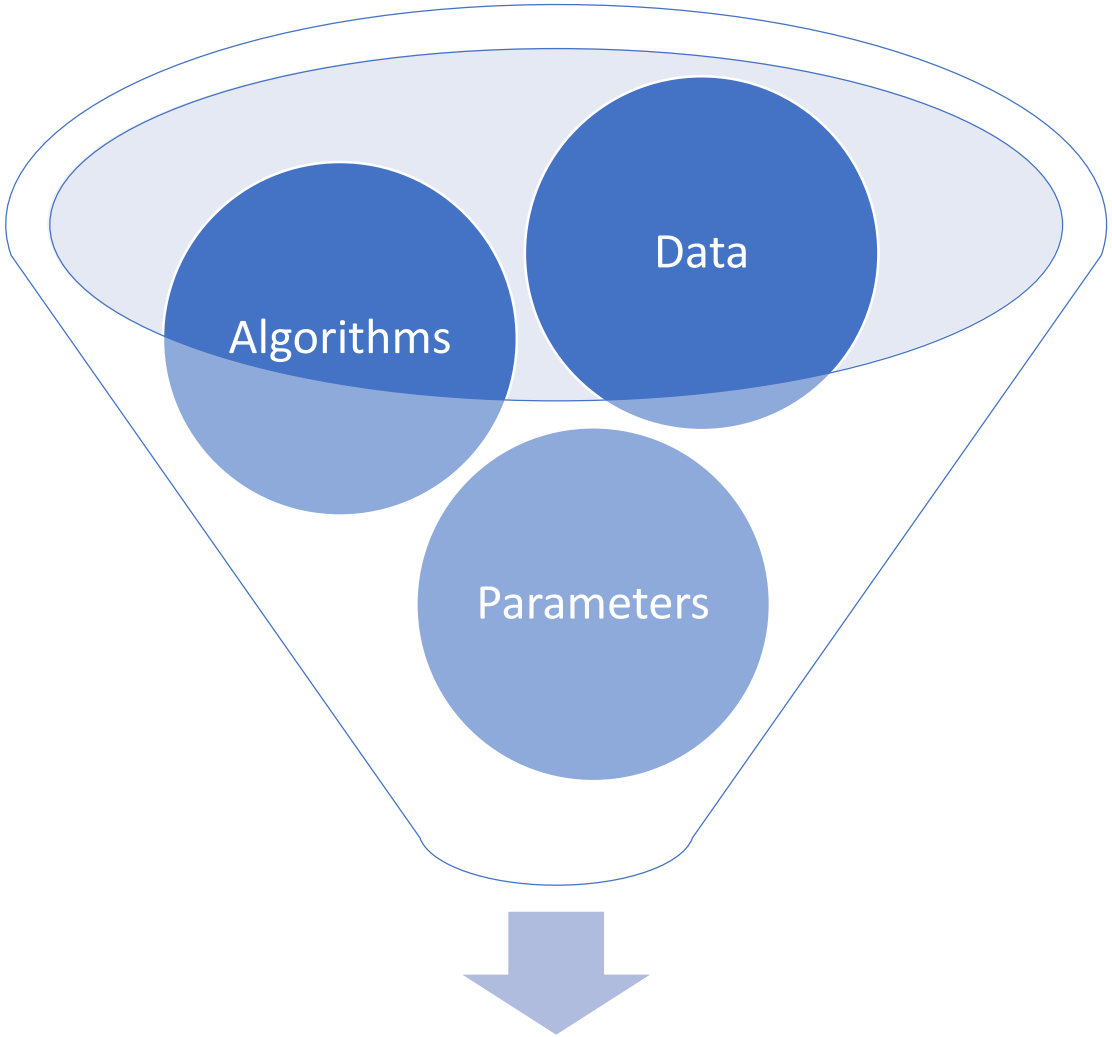
Iteratively derive structure from data

(Patterns, Rules, Clusters, Policies)


ML
Supervised
Unsupervised
Reinforcement

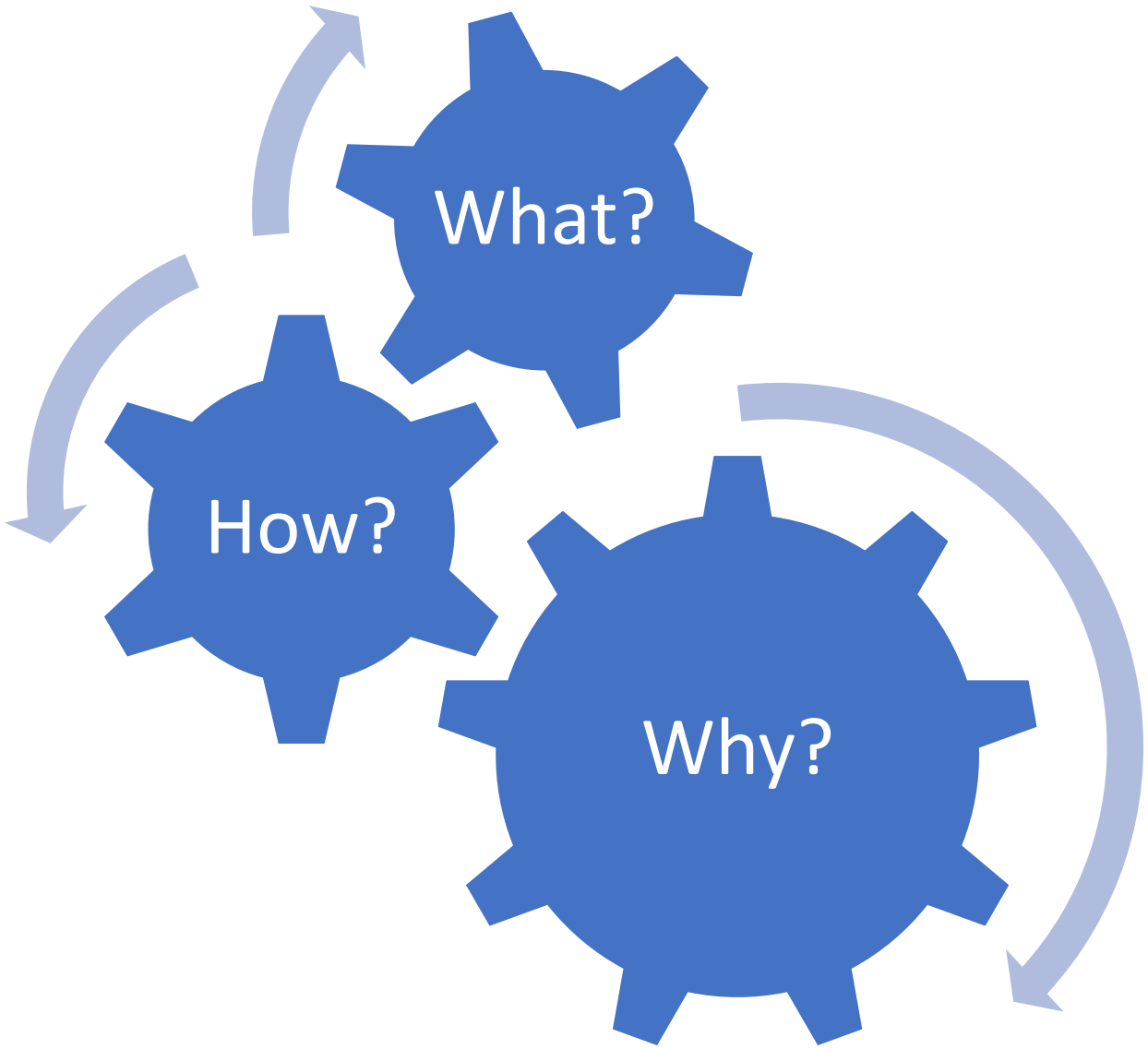
Supervised Machine Learning?

(We have labels)



Predictions

Interpretable Machine Learning?

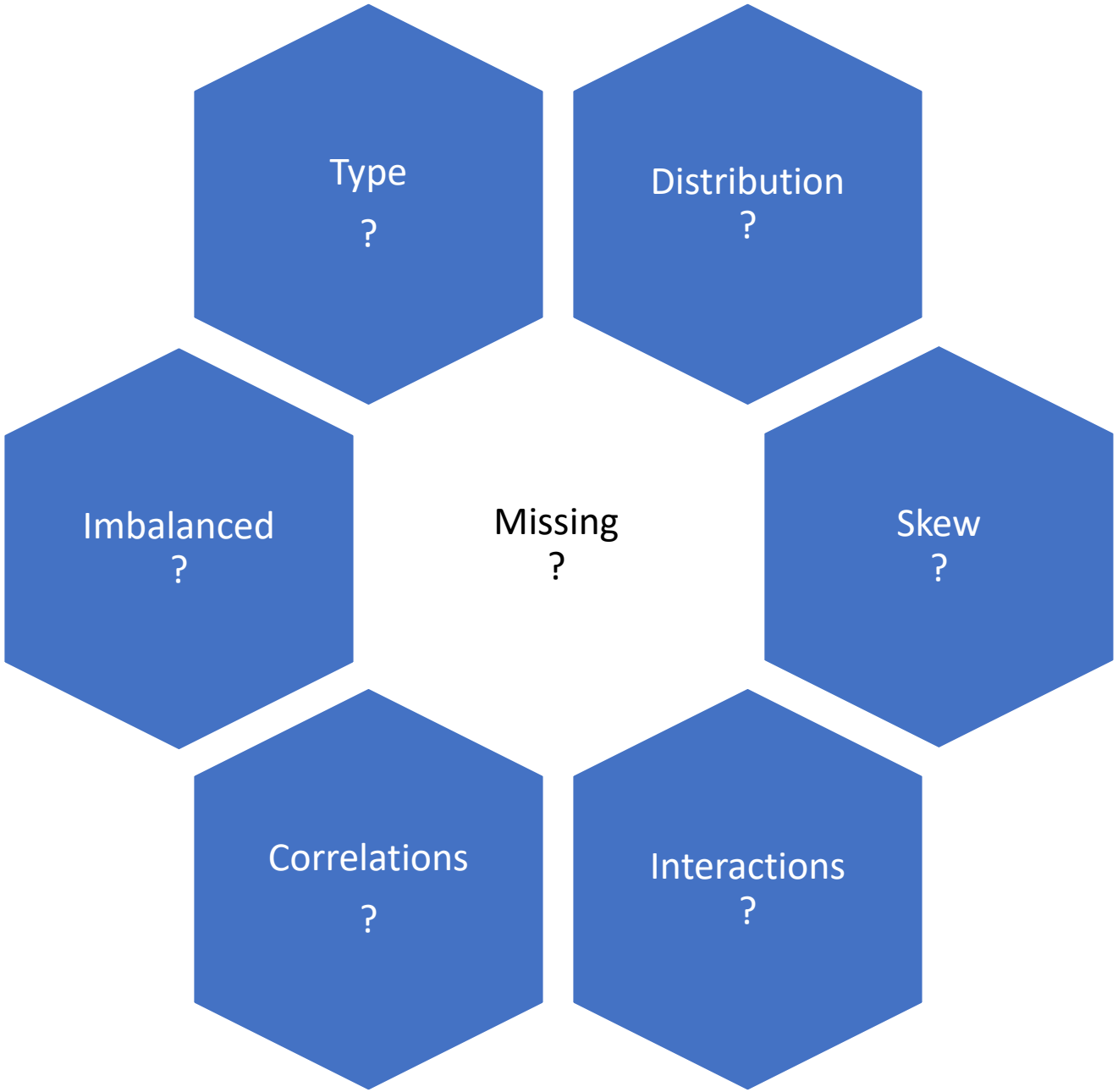

What?
How?
Why?

What? – Data (Features)

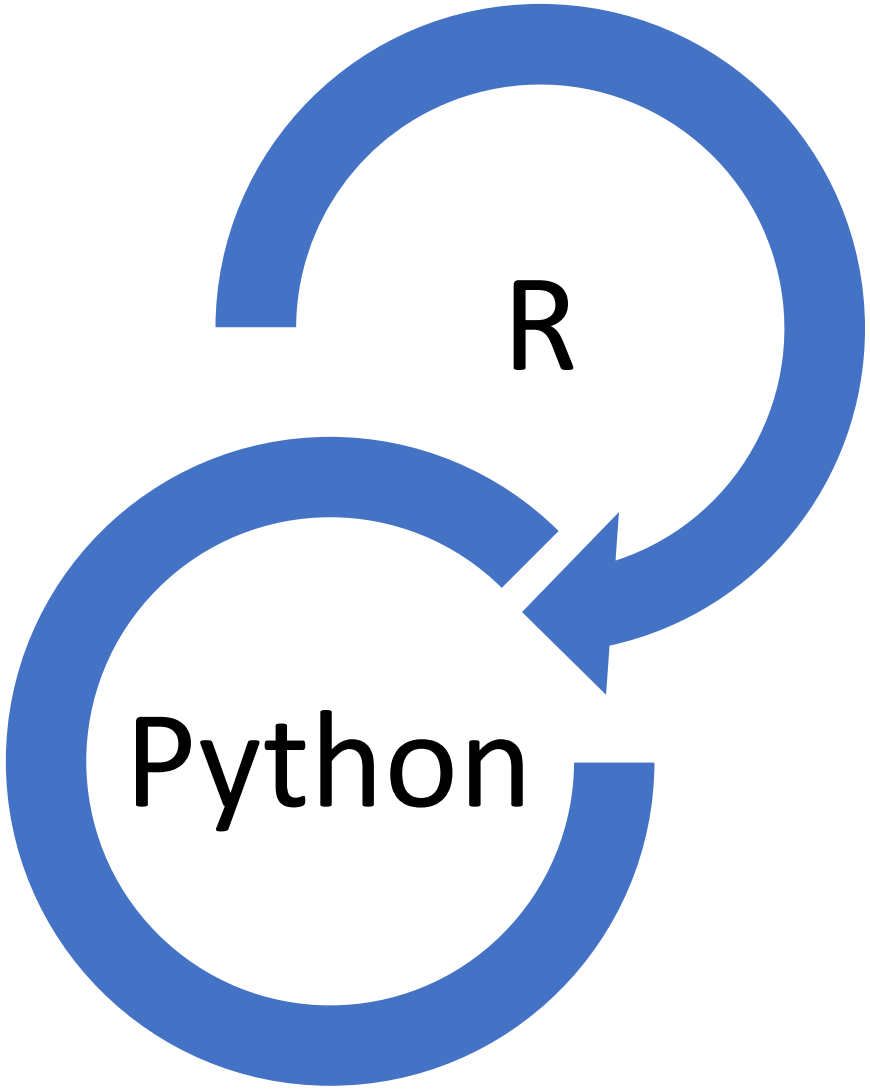
How? – Model (Global)

Why? – Prediction (Local)

How well do you understand the data?


Type
?
Distribution
?
Imbalanced
?
Missing
?
Skew
?
Correlations
?
Interactions
?

Tools?


R
Python

EDA

Data Explorer (R)

Pandas Profiling (Py)

create_report([dataset])

Data Profiling Report

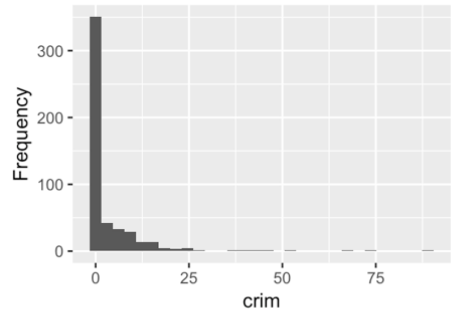
Basic Statistics

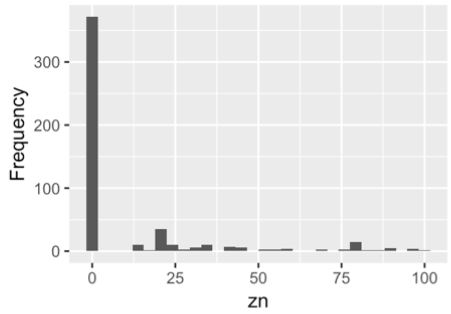
Raw Counts

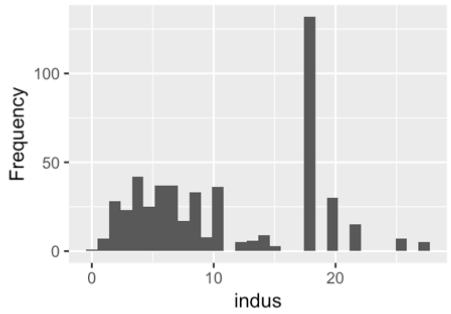
Name	Value
Rows	506
Columns	14
Discrete columns	1
Continuous columns	13
All missing columns	0
Missing observations	0
Complete Rows	506
Total observations	7,084
Memory allocation	57.4 Kb

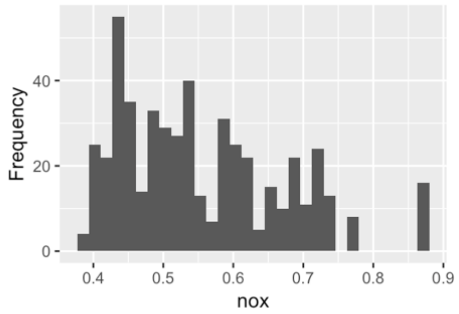
Univariate Distribution

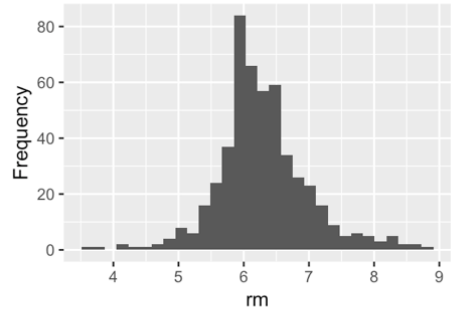
Histogram

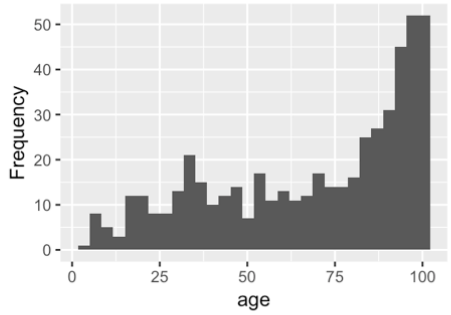


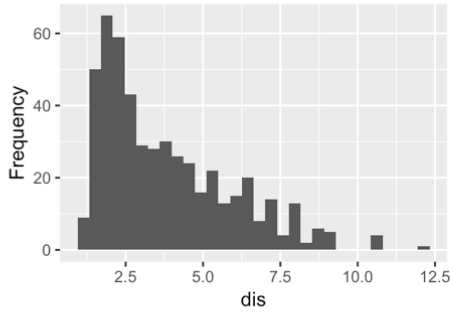


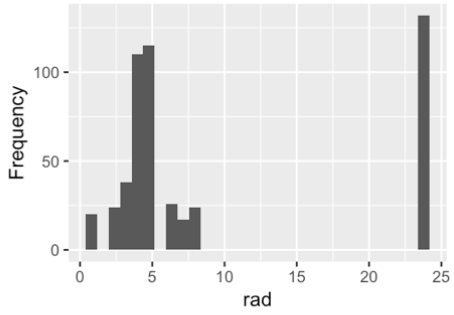


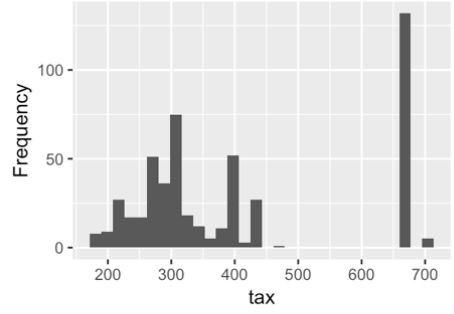


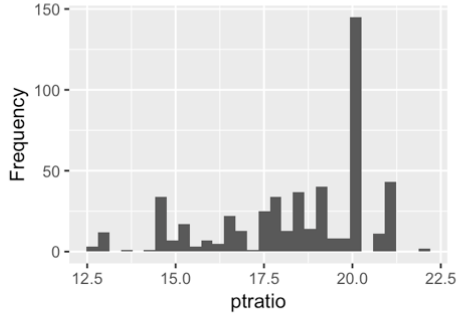


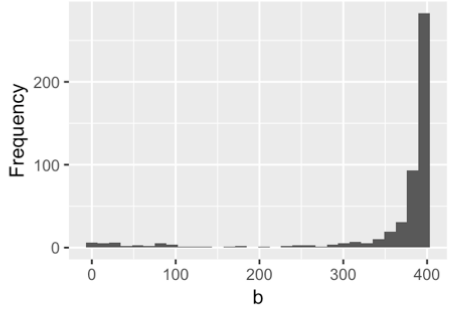


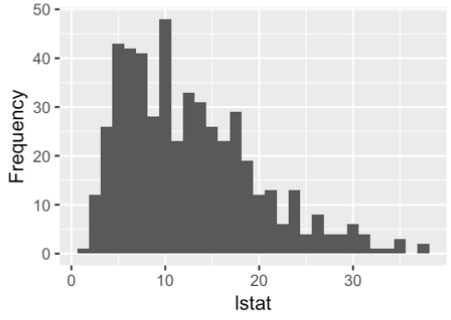


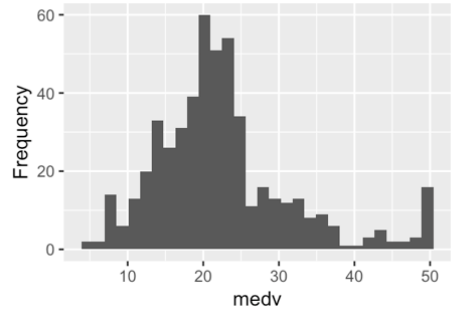




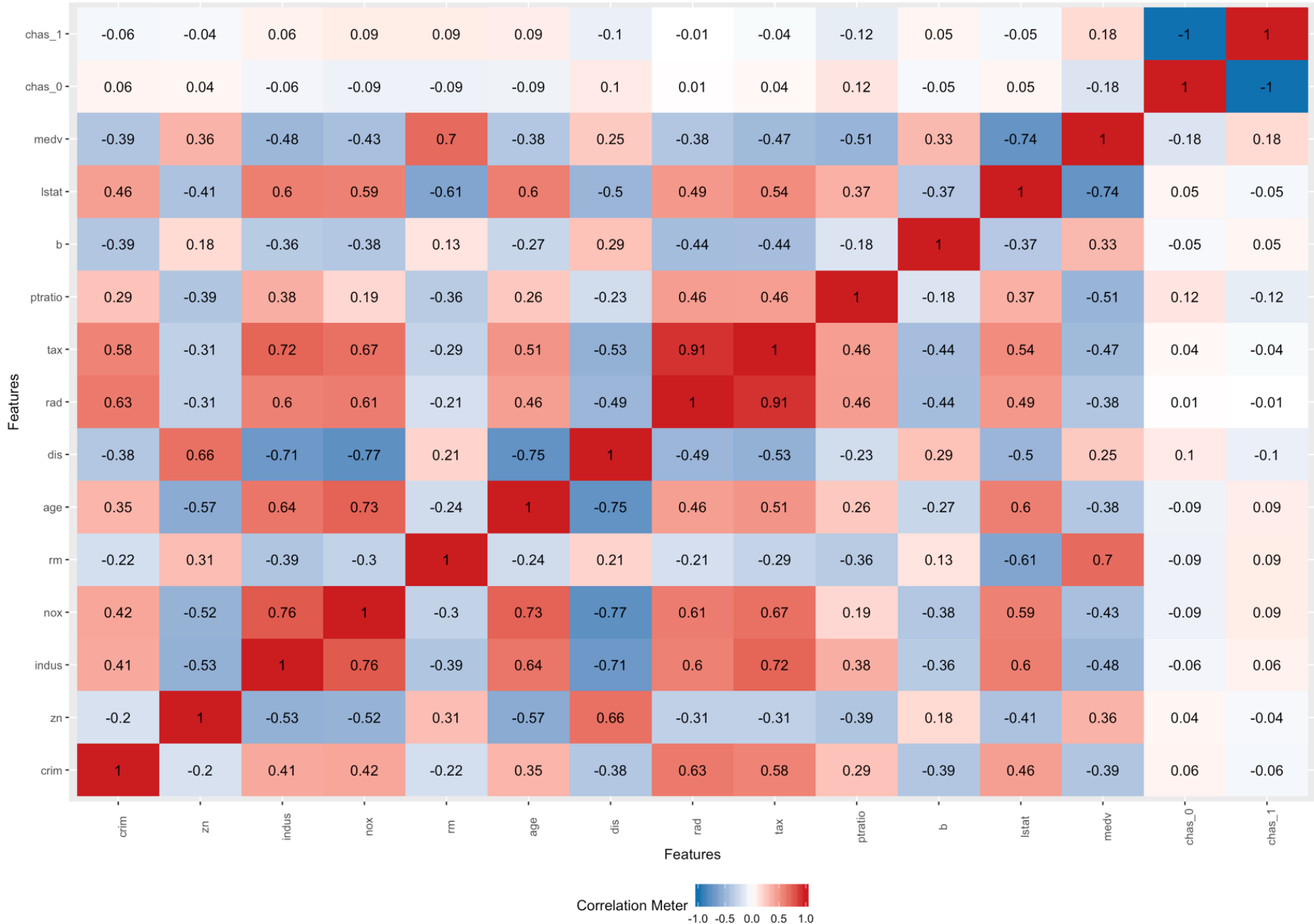








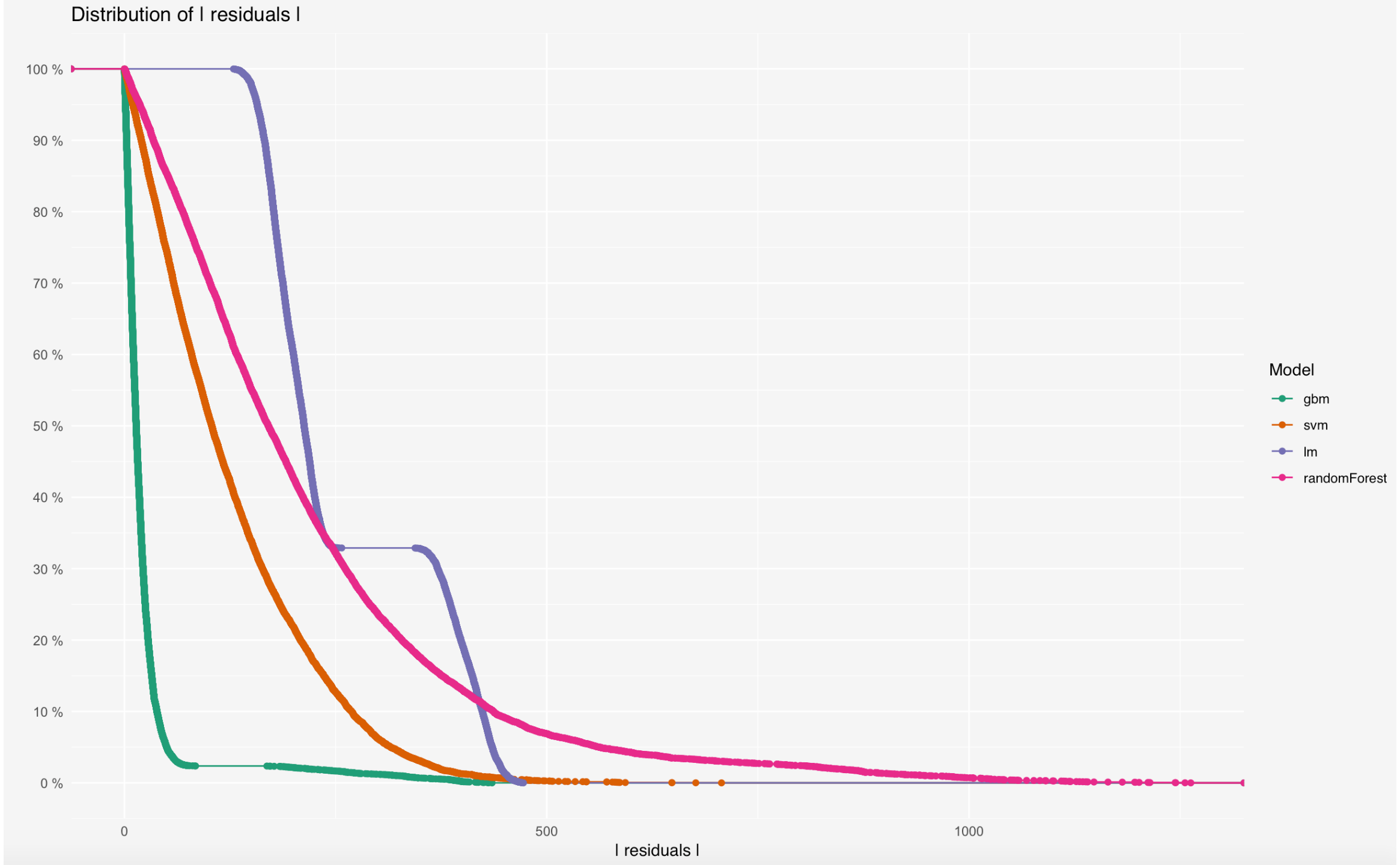
Correlation Analysis



Model Residuals Plot

DALEX (R)

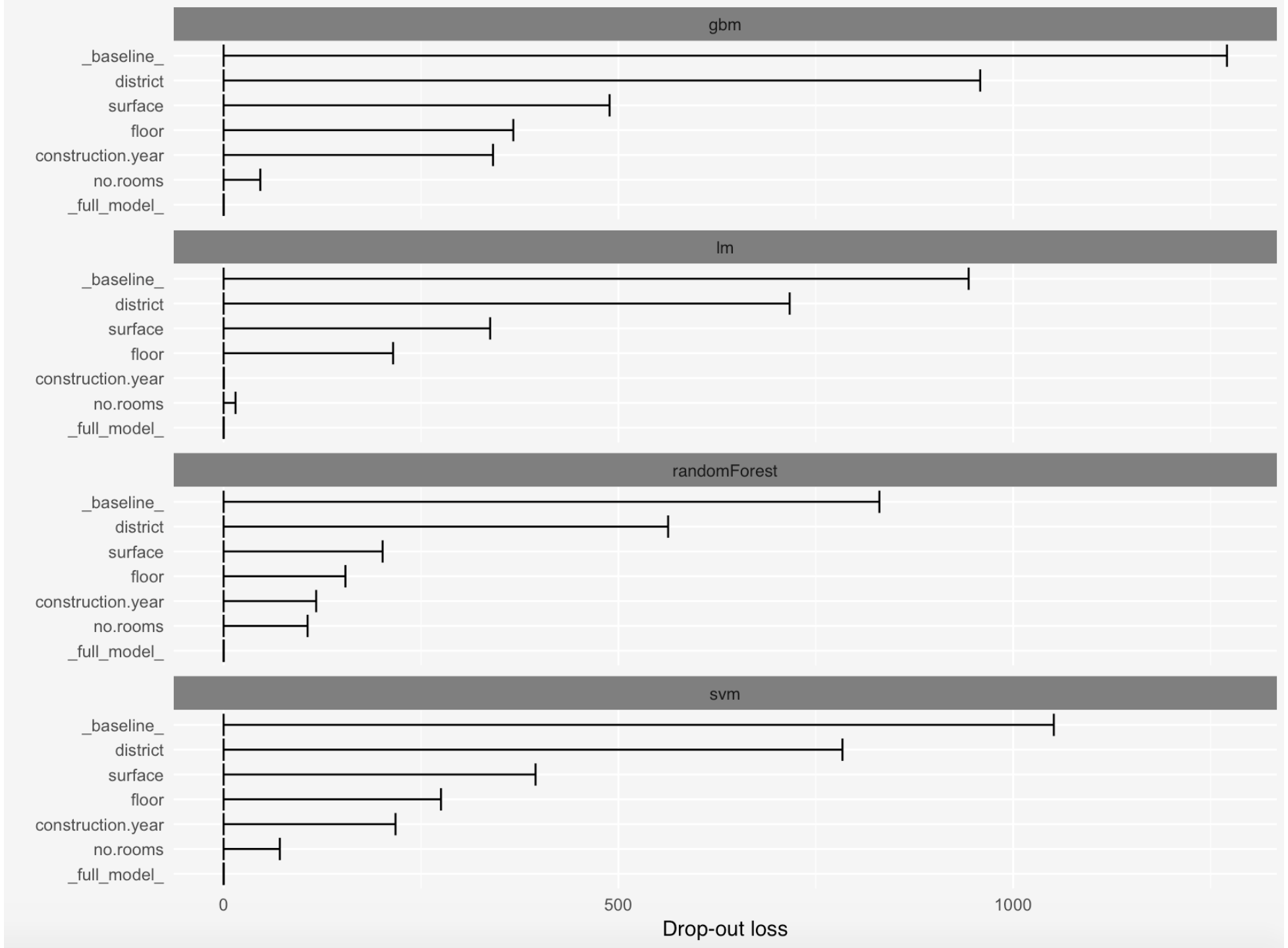
YELLOWBRICK (Py)


Distribution of | residuals |
100 %
90 %
80 %
70 %
60 %
50 %
40 %
30 %
20 %
10 %
0 %
0
500
1000
| residuals |
Model
gbm
svm
lm
randomForest

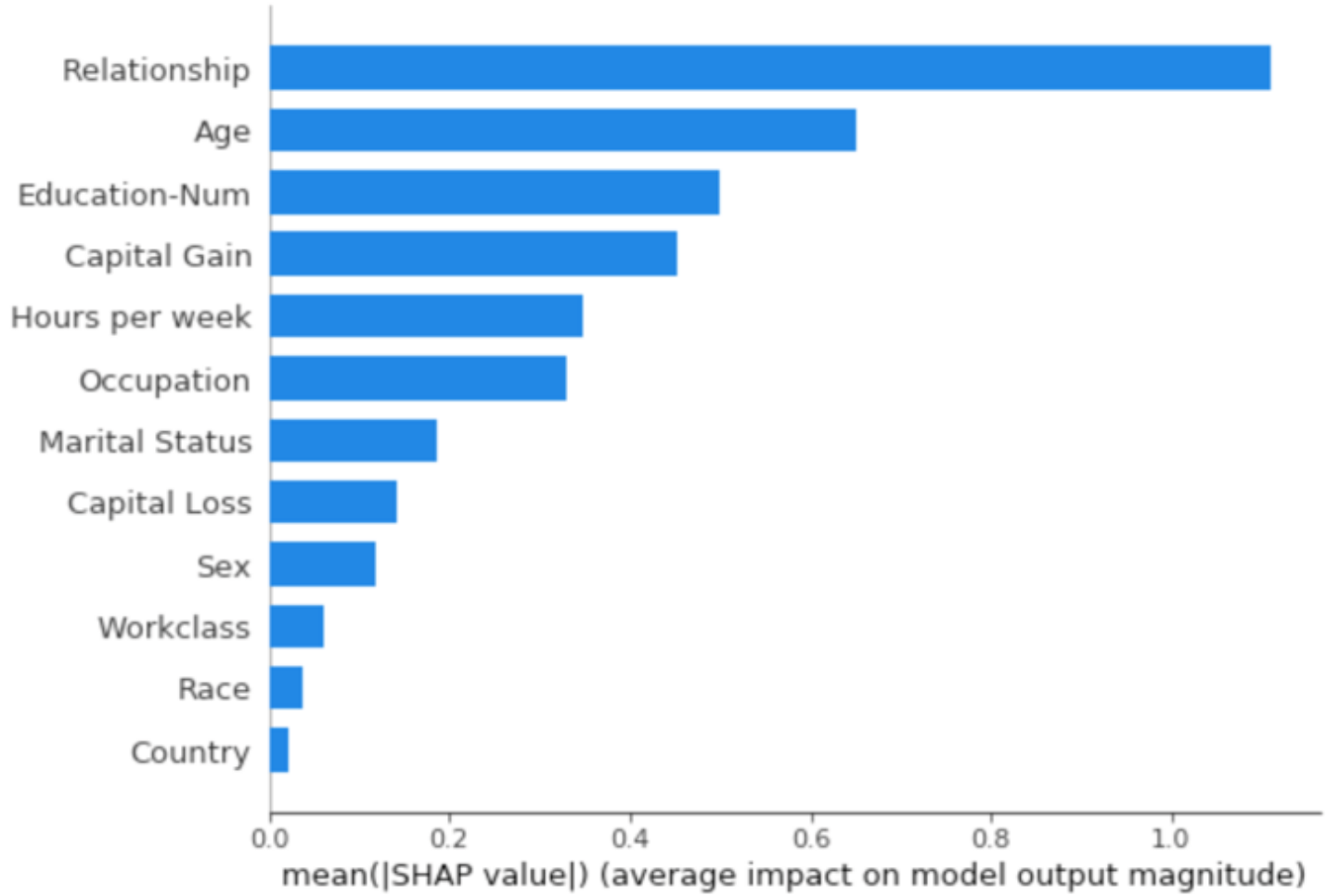
Variable Importance Plot

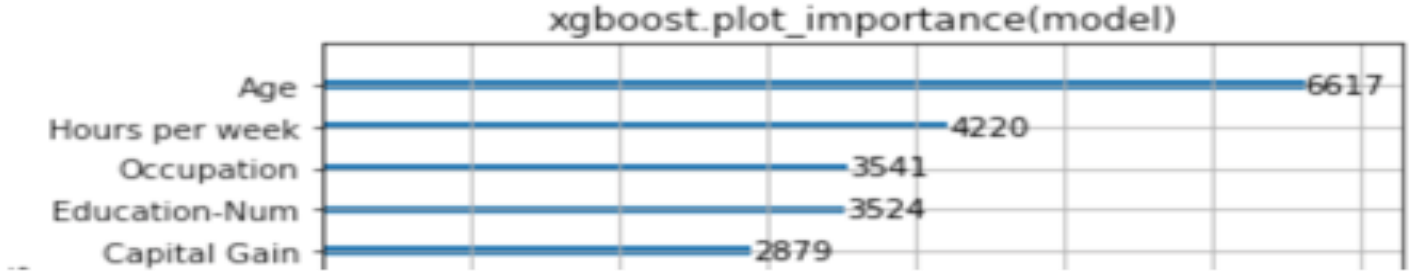
DALEX (R)

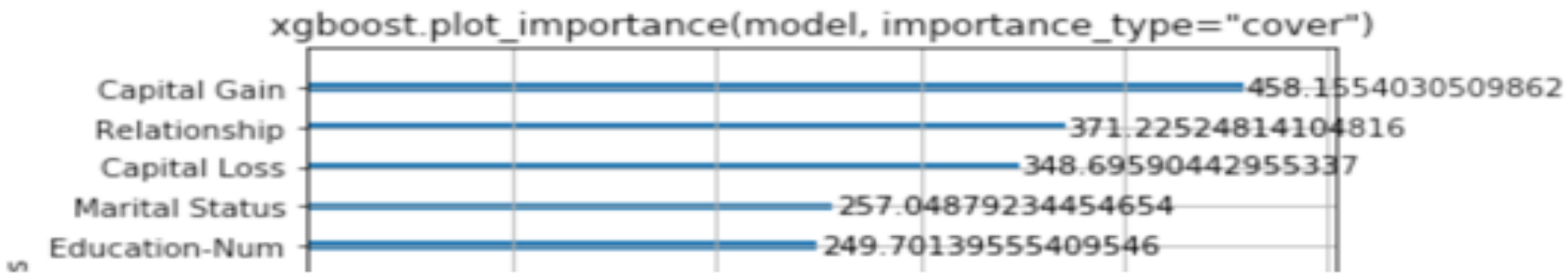
SHAP (PY)

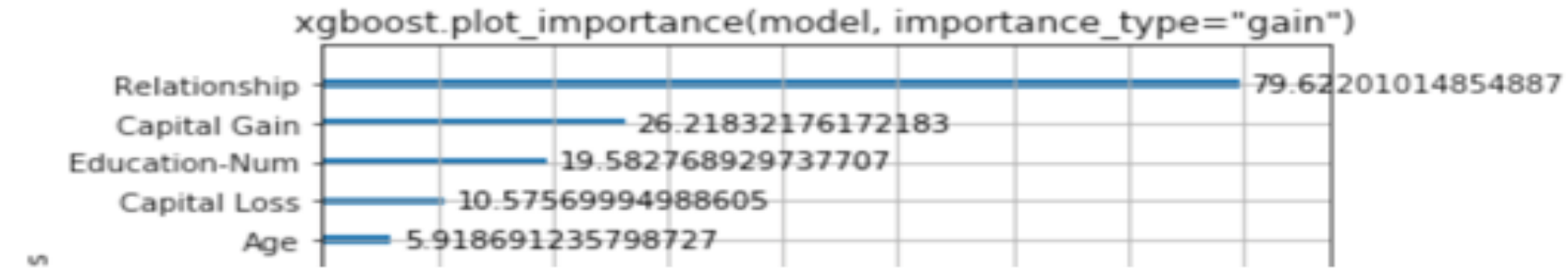

gbm
baseline
district
surface
floor
construction.year
no.rooms
_full_model_
lm
baseline
district
surface
floor
construction.year
no.rooms
_full_model_
randomForest
baseline
district
surface
floor
construction.year
no.rooms
_full_model_
svm
baseline
district
surface
floor
construction.year
no.rooms
_full_model_
0
500
1000
Drop-out loss

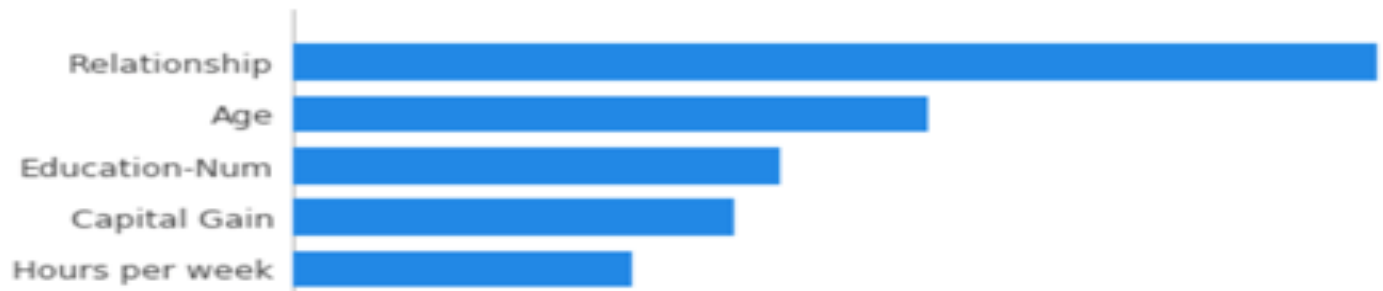
In [11]:
```
shap.summary_plot(shap_values, X_display, plot_type="bar")
```




xgboost.plot_importance(model)
Age
Hours per week
Occupation
Education-Num
Capital Gain
6617
4220
3541
3524
2879


xgboost.plot_importance(model, importance_type="cover")
Capital Gain
Relationship
Capital Loss
Marital Status
Education-Num
458.1554030509862
371.225248141048l6
348.69590442955337
257.04879234454654
249.70139555409546

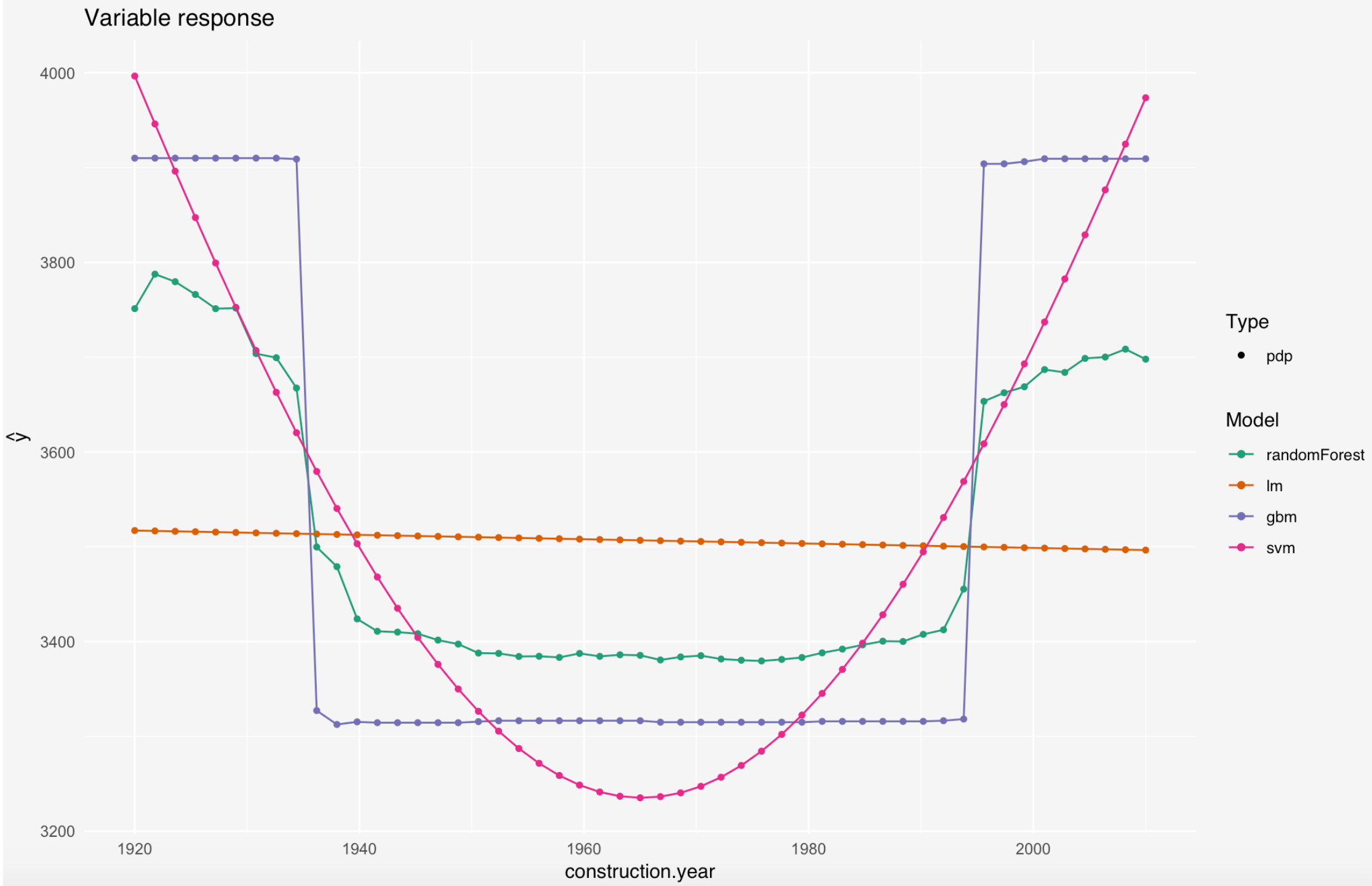

xgboost.plot_importance(model, importance_type="gain")
Relationship
Capital Gain
Education-Num
Capital Loss
Age
79.62201014854887
26.21832176172183
19.582768929737707
10.57569994988605
5.918691235798727


Relationship
Age
Education-Num
Capital Gain
Hours per week

Partial Dependence & Individual Conditional Plots

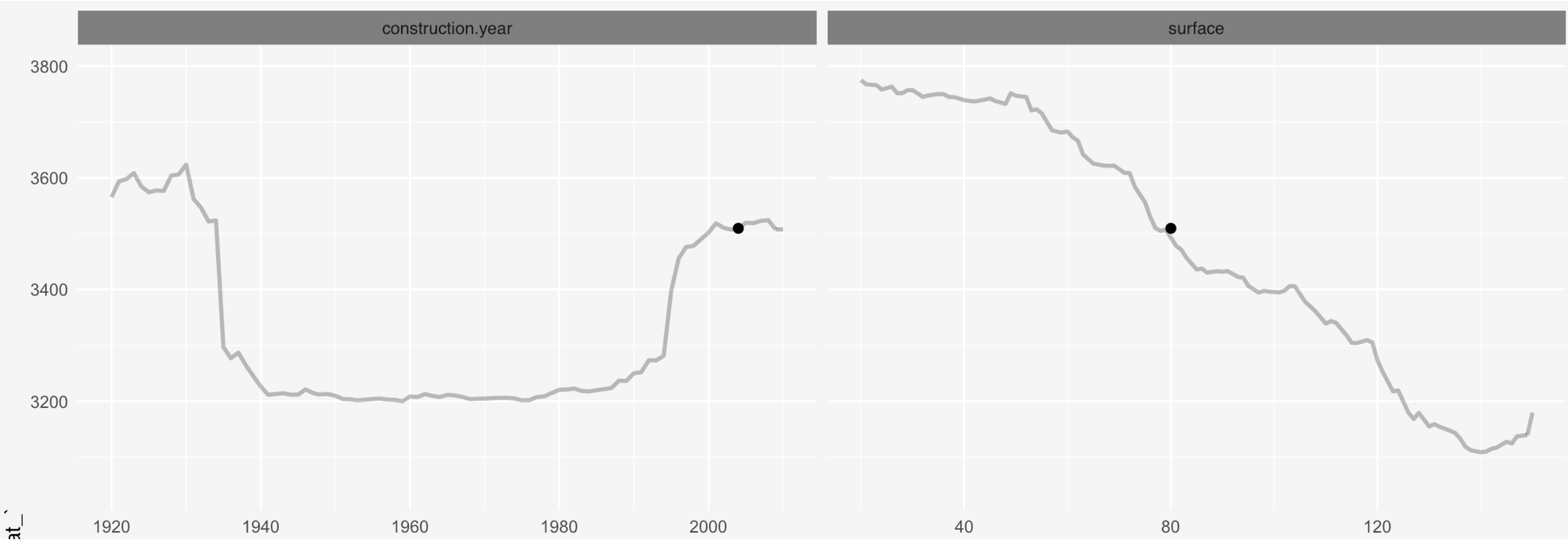
DALEX/IML (R)

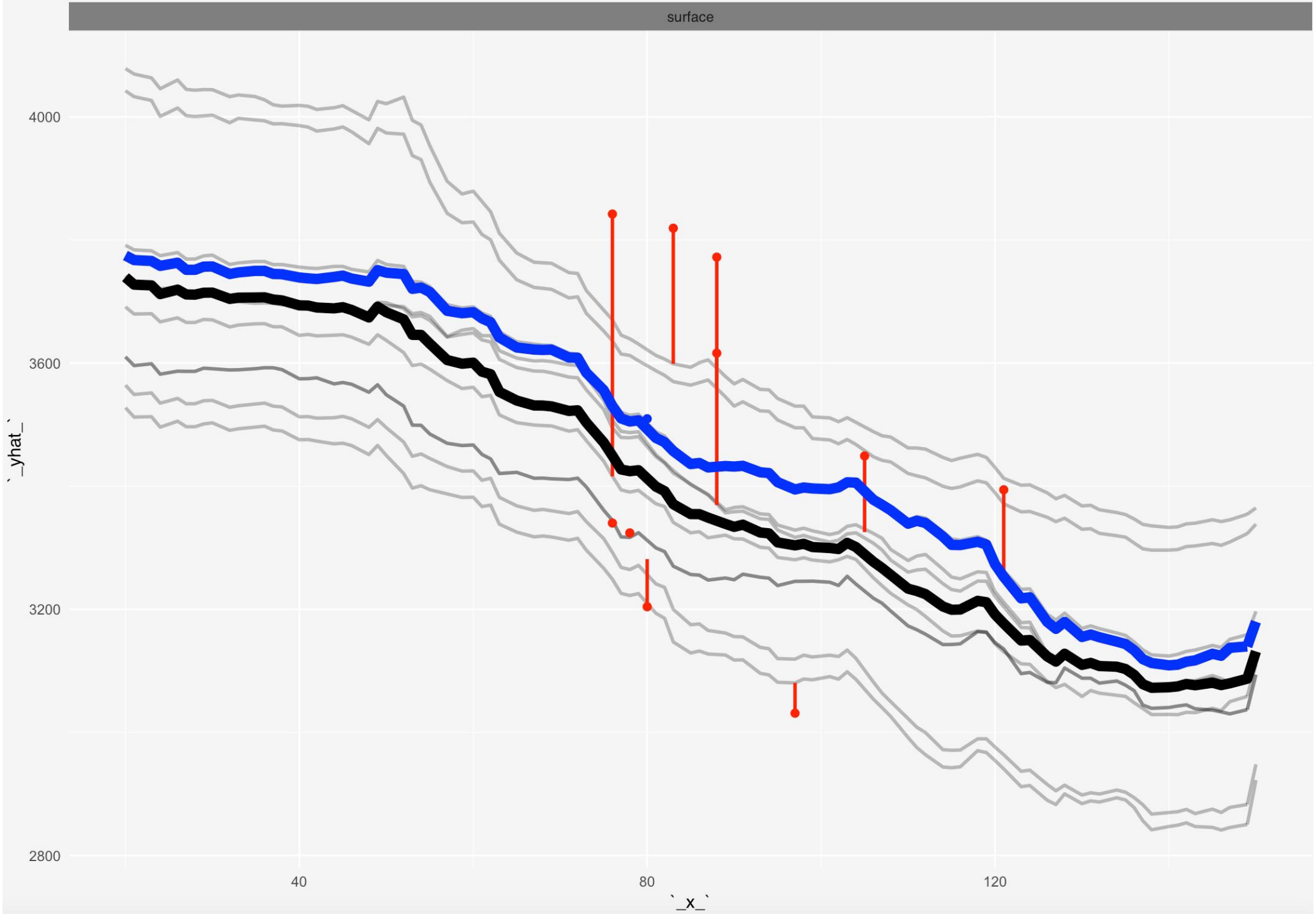
PDPBOX (Py)

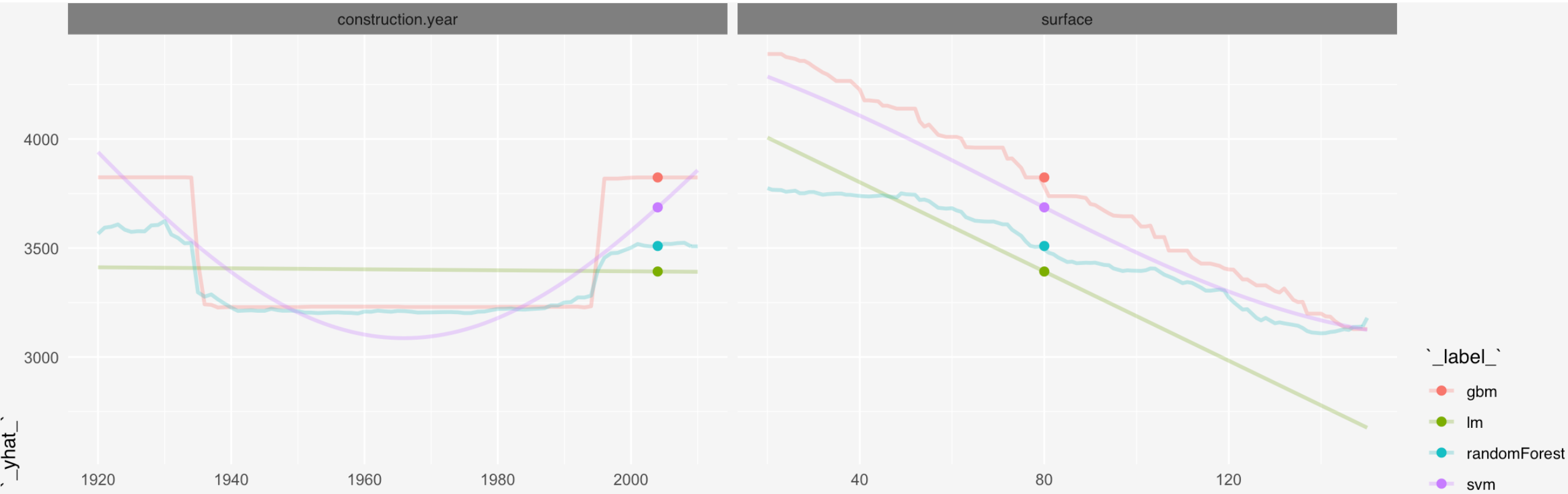

Variable response
4000
3800
3600
3400
3200
<y>
1920
1940
1960
1980
2000
construction.year
Type
pdp
Model
randomForest
lm
gbm
svm

Ceteris Paribus Plot

DALEX (R)


construction.year
surface
3800
3600
3400
3200
1920
1940
1960
1980
2000
40
80
120


surface
4000
3600
3200
2800
`_yhat_`
40
80
120
`_x_`


construction.year
surface
4000
3500
3000
`_yhat_`
1920
1940
1960
1980
2000
40
80
120
`_label_`
gbm
lm
randomForest
svm

LIME (R)

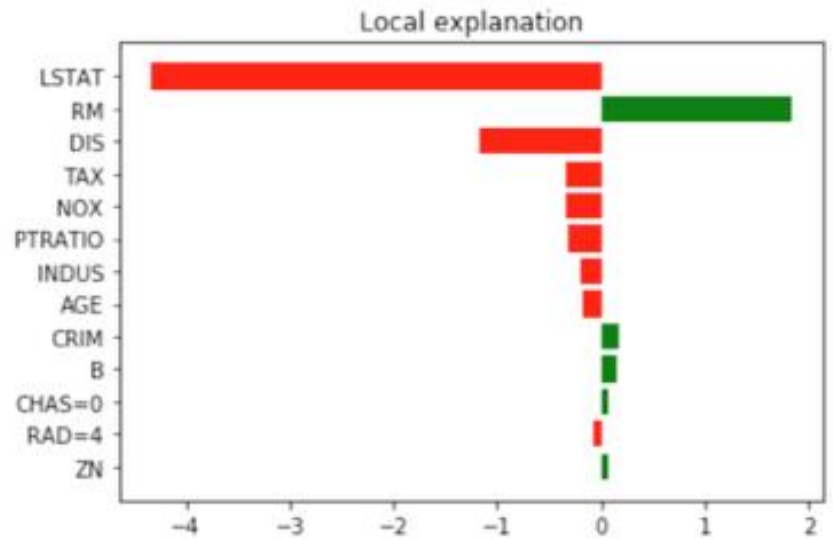
LIME (Py)

In [2]:
```
# import lime tools
import lime
import lime.lime_tabular

# generate an "explainer" object
categorical_features  = np.argwhere(np.array([len(set(boston.data[:,x])) for x in range(boston.data.shape[1])]) <= 10).
explainer = lime.lime_tabular.LimeTabularExplainer(train, feature_names=boston.feature_names, class_names=['price'], ca
```

In [3]:
```
#generate an explanation
i = 13
exp = explainer.explain_instance(test[i], rf.predict, num_features=14)
```

In [4]:
```
%matplotlib inline
fig = exp.as_pyplot_figure();
```



Generate data points based on training data

Compute complex model predictions from the generated data to find the 'most useful features'

Fit a local linear model for the 'most useful features' and use the feature coefficients as reason codes

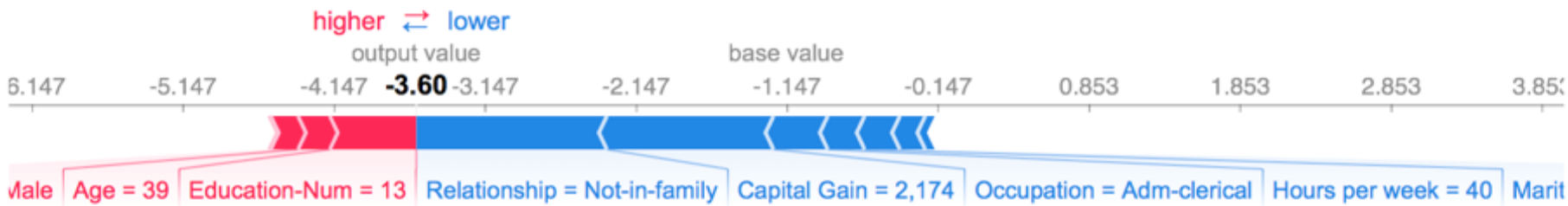
Shapley Values (Coalition Attribution)

IML (R)

SHAP (Py)

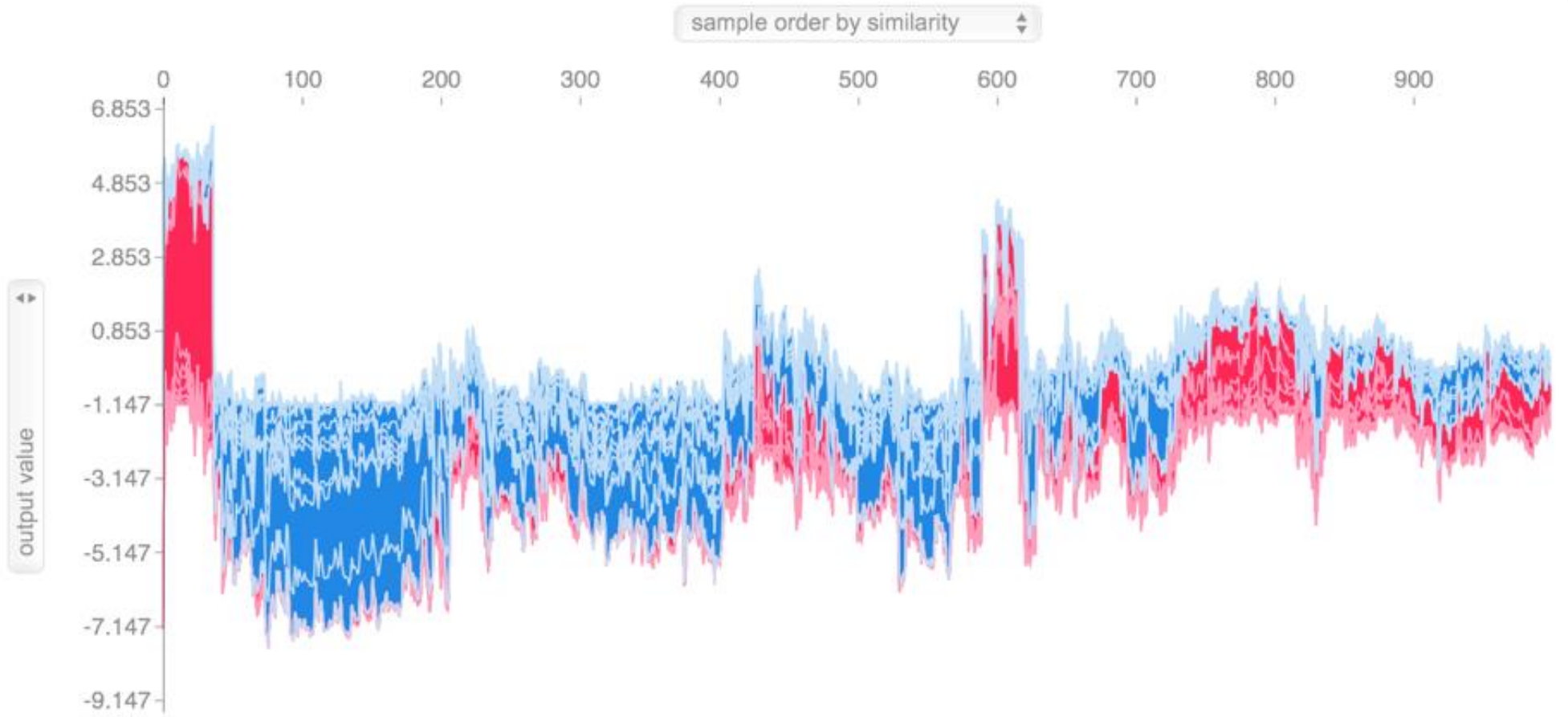
In [9]:
```
shap.force_plot(explainer.expected_value, shap_values[0,:], X_display.iloc[0,:])
```

Out[9]:



In [10]:
```
shap.force_plot(explainer.expected_value, shap_values[:1000,:], X_display.iloc[:1000,:])
```

Out[10]:



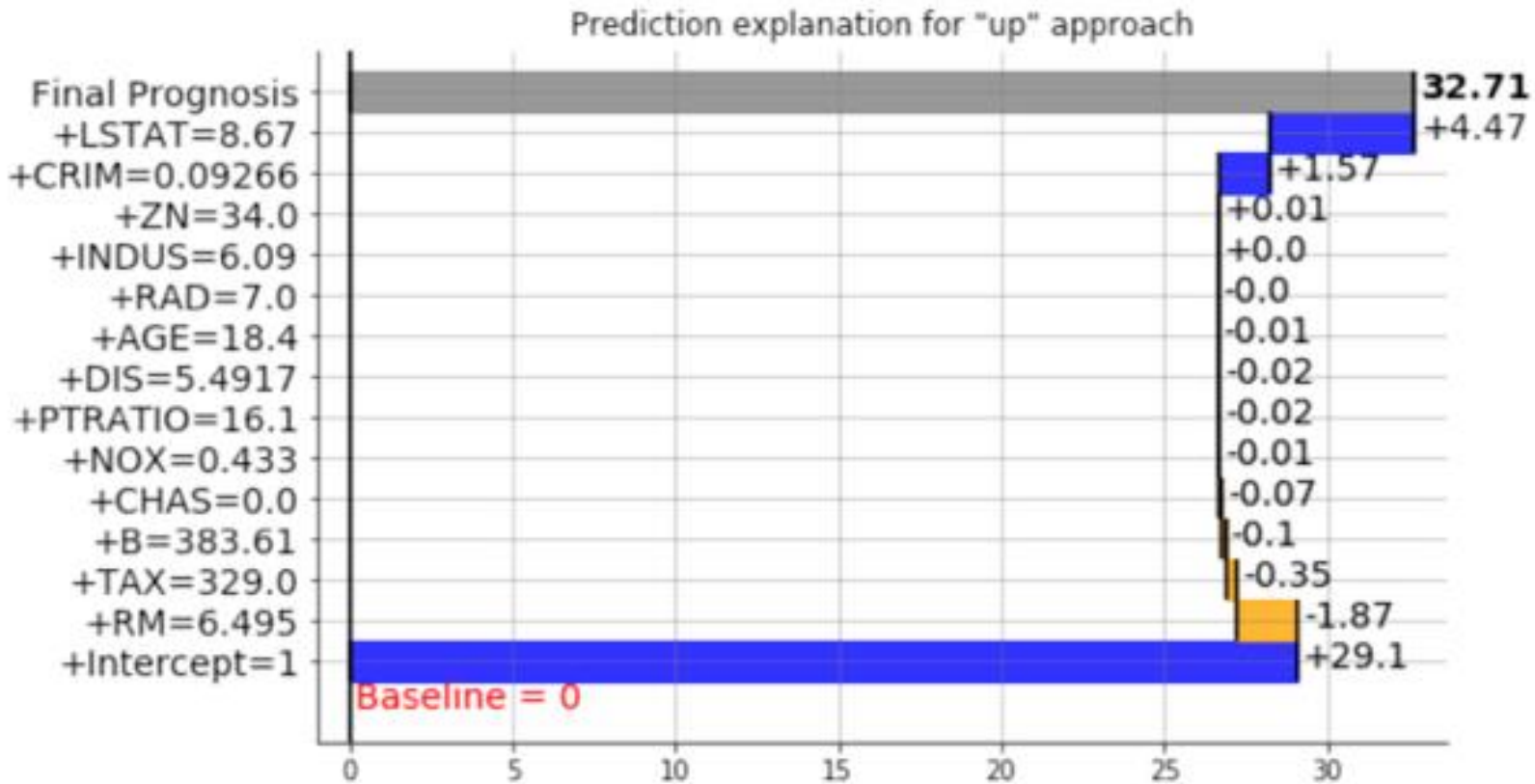
BreakDown

DALEX (R)

PyBreakdown (Py)

In [7]:
```
# visualisation

explanation.visualize(figsize=(8,5),dpi=100)
```



Surrogate Trees

IML (R)

SKATER (Py)

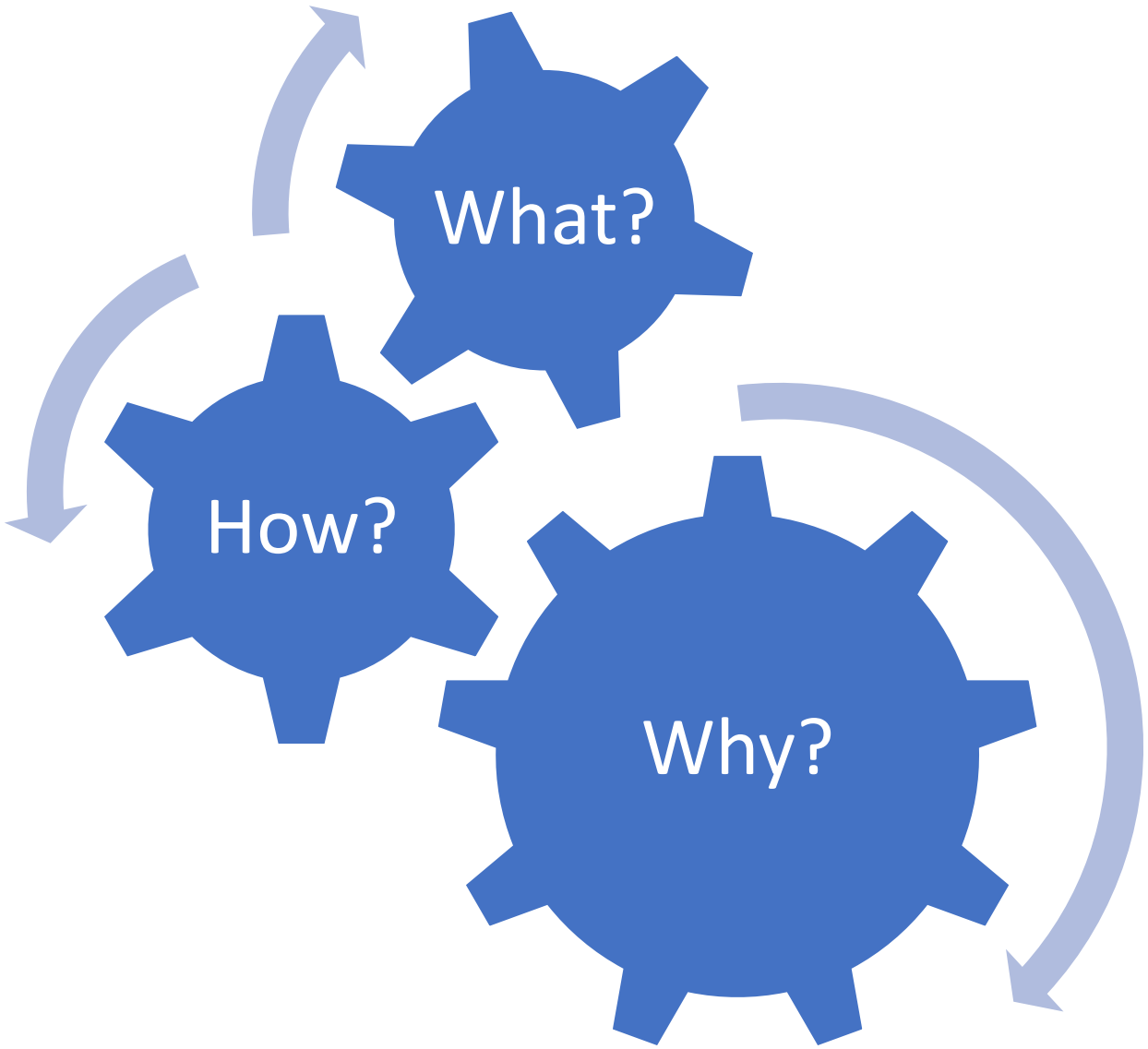
In [49]:
```
surrogate_explainer_reg.decisions_as_txt('local', X_test.iloc[sample_index])
```

```
As RM[5.914] <= 6.837500095367432 then,
  As LSTAT[18.33] > 14.399999618530273 then,
    As CRIM[0.31827] <= 7.084139823913574 then,
      As CRIM[0.31827] <= 0.6147900223731995 then,
        As DIS[3.9986] > 1.979900024795532 then,
          As AGE[83.2] > 73.30000305175781 then,
            As CRIM[0.31827] > 0.1712700128555297 9 then,
              As CHAS[0.0] <= 0.5 then,
                As B[390.7] <= 391.875 then,
                  As NOX[0.544] > 0.4864999499320984 then,
                    As B[390.7] > 389.96002197265625 then,
                      Value: [[19.36]]
```

Conclusion

Multiple ML Models

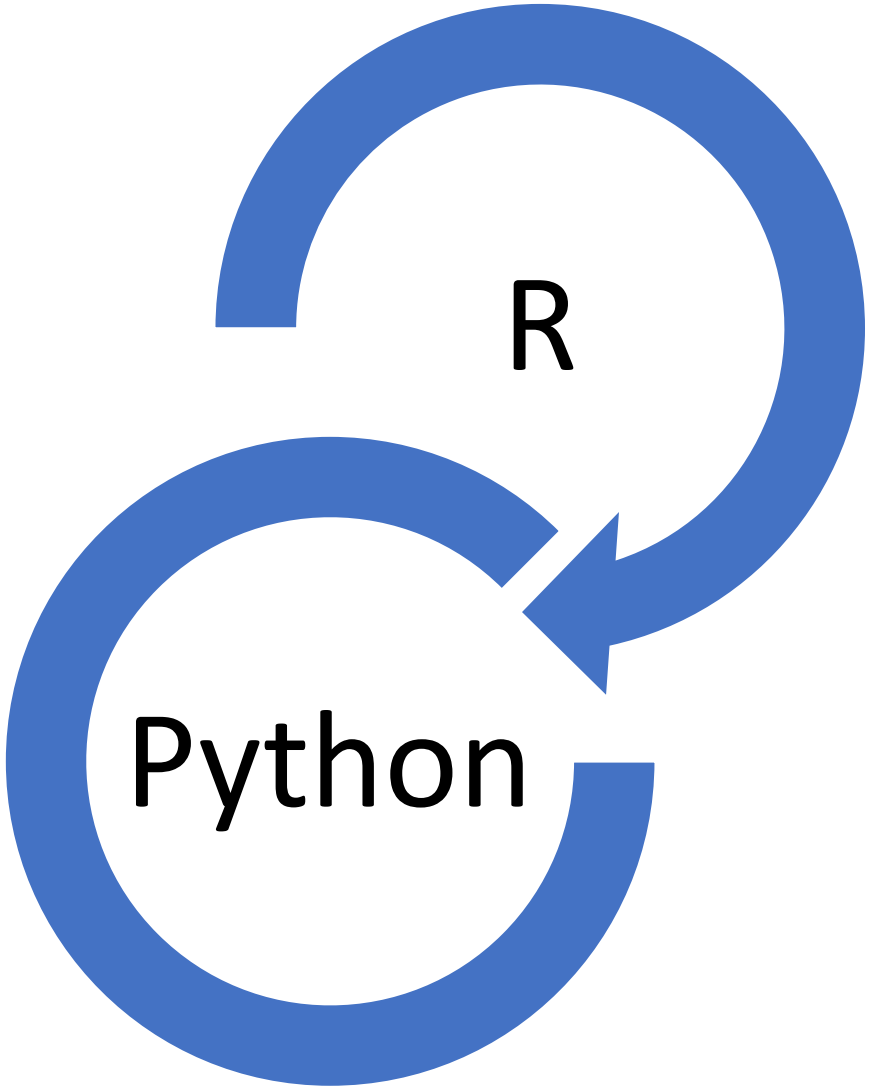
Multiple Interpretation Approaches


What?
How?
Why?

Data Understanding

Consider Local and Global

Combine Multiple Perspectives


R
Python

Recommendations (Python):

Local - [SHAP]
Global - Surrogate Trees [SKATER]
Data -[Pandas Profiling]
Visual - [Yellowbrick], [PDPBox]

Recommendations (R):

Local - SHAP [IML]
Global - Surrogate Trees [IML], Variable Importance [DALEX]
Data - [Data Explorer]
Visual - Breakdown, ALE, Ceteris Paribus [DALEX]
Visual - PDP, ICE [IML]

Thanks for Listening!

dean_allsopp@hotmail.com