

TREE BASED MODELS

Decision Trees

INTRODUCTION

- Decision Trees can be used for
 - Classification
 - Regression
- Decision Trees can be used as standalone algorithms (this is what we do here)
- Decision Trees can be used as components for other models such as:
 - Random Forest Models
 - Boosted Tree Models

INTRODUCING THE IDEA OF DECISION TREES WITH THE TITANIC DATASET

► Code

	Survived	Sex	Class	Age	Fare
5	0	male	3	35	8.0500
7	0	male	1	54	51.8625
8	0	male	3	2	21.0750
13	0	male	3	20	8.0500
14	0	male	3	39	31.2750

GENERATING A DECISION TREE WITH `tidymodels`

Note, `rpart` package needs to be installed. `tidymodels` loads the `rpart` package automatically. Therefore `library(rpart)` is not needed.

► Code

```
== Workflow [trained] ==
Preprocessor: Recipe
Model: decision_tree()

— Preprocessor —
0 Recipe Steps

— Model —
n= 664

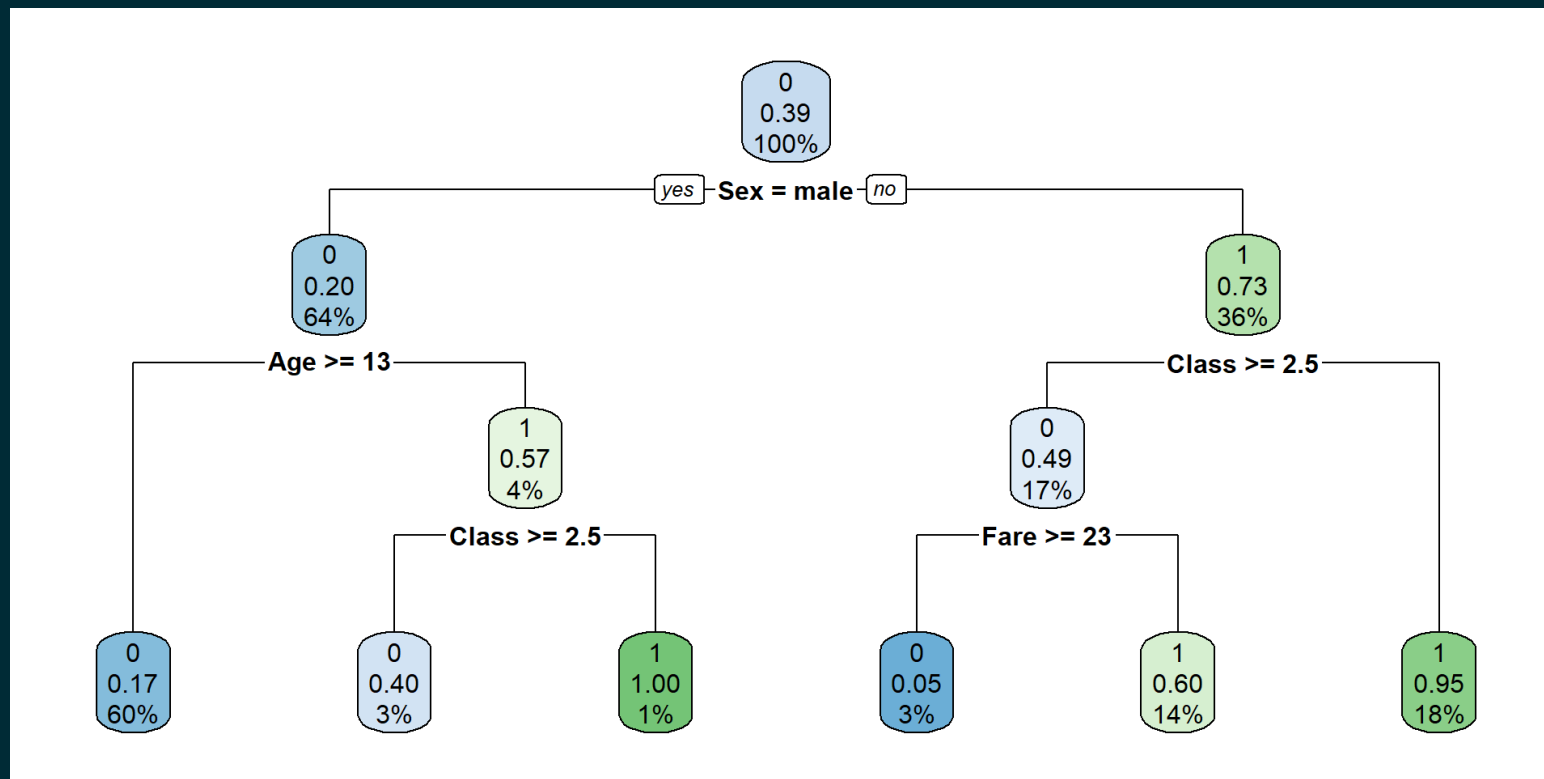
node), split, n, loss, yval, (yprob)
  * denotes terminal node

1) root 664 256 0 (0.61445783 0.38554217)
 2) Sex=male 428 84 0 (0.80373832 0.19626168)
 4) Age>=13 400 68 0 (0.83000000 0.17000000) *
 5) Age< 13 28 12 1 (0.42857143 0.57142857)
 10) class>=2 5 20 8 0 (0.60000000 0.40000000) *
```

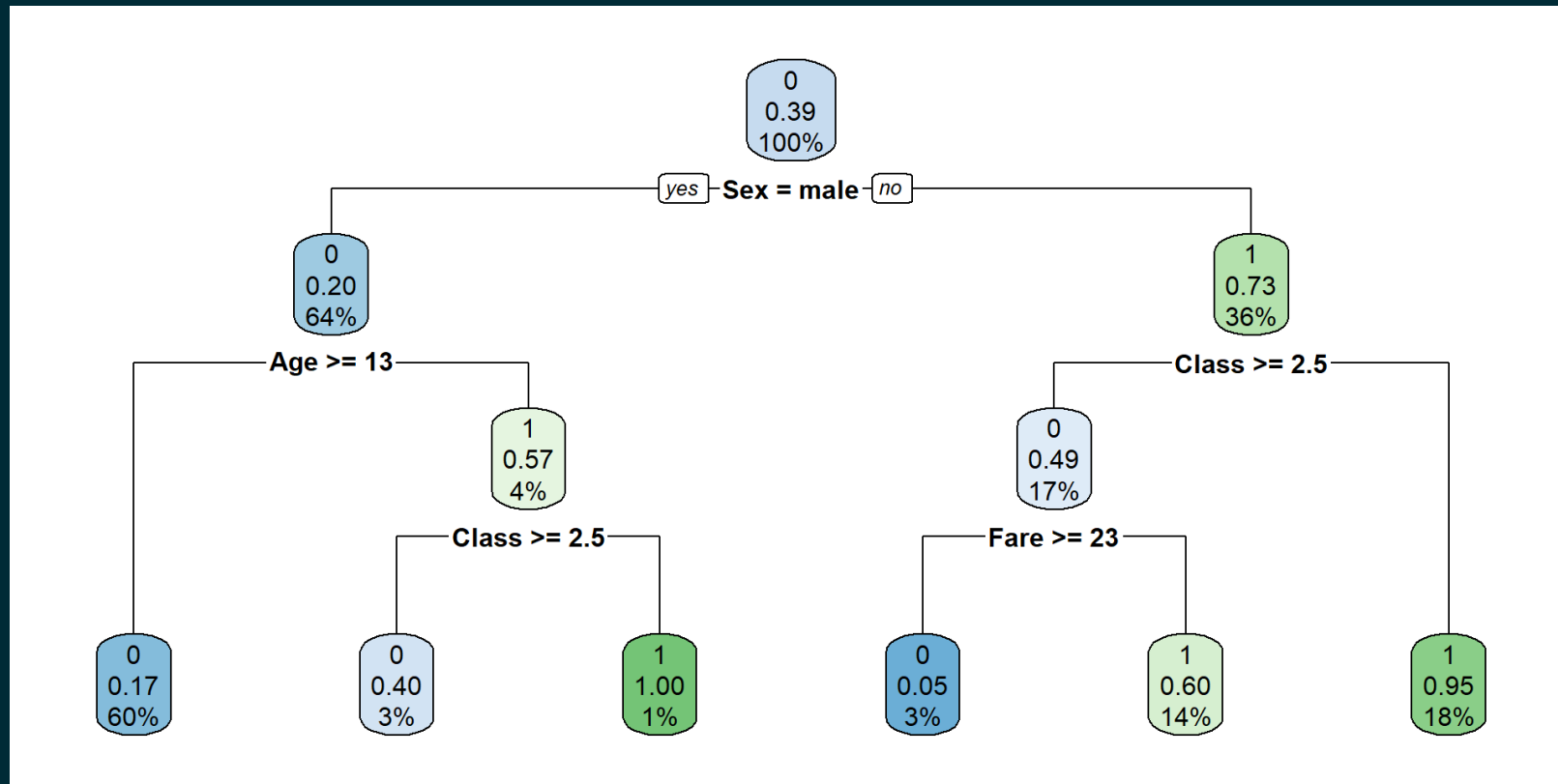
DISPLAYING THE DECISION TREE WITH `rpart.plot`

Note, `rpart.plot` package needs to be installed and loaded with `library(rpart.plot)`.

► Code



NODES IN THE DECISION TREE (IGNORE DECISION RULES FOR NOW)



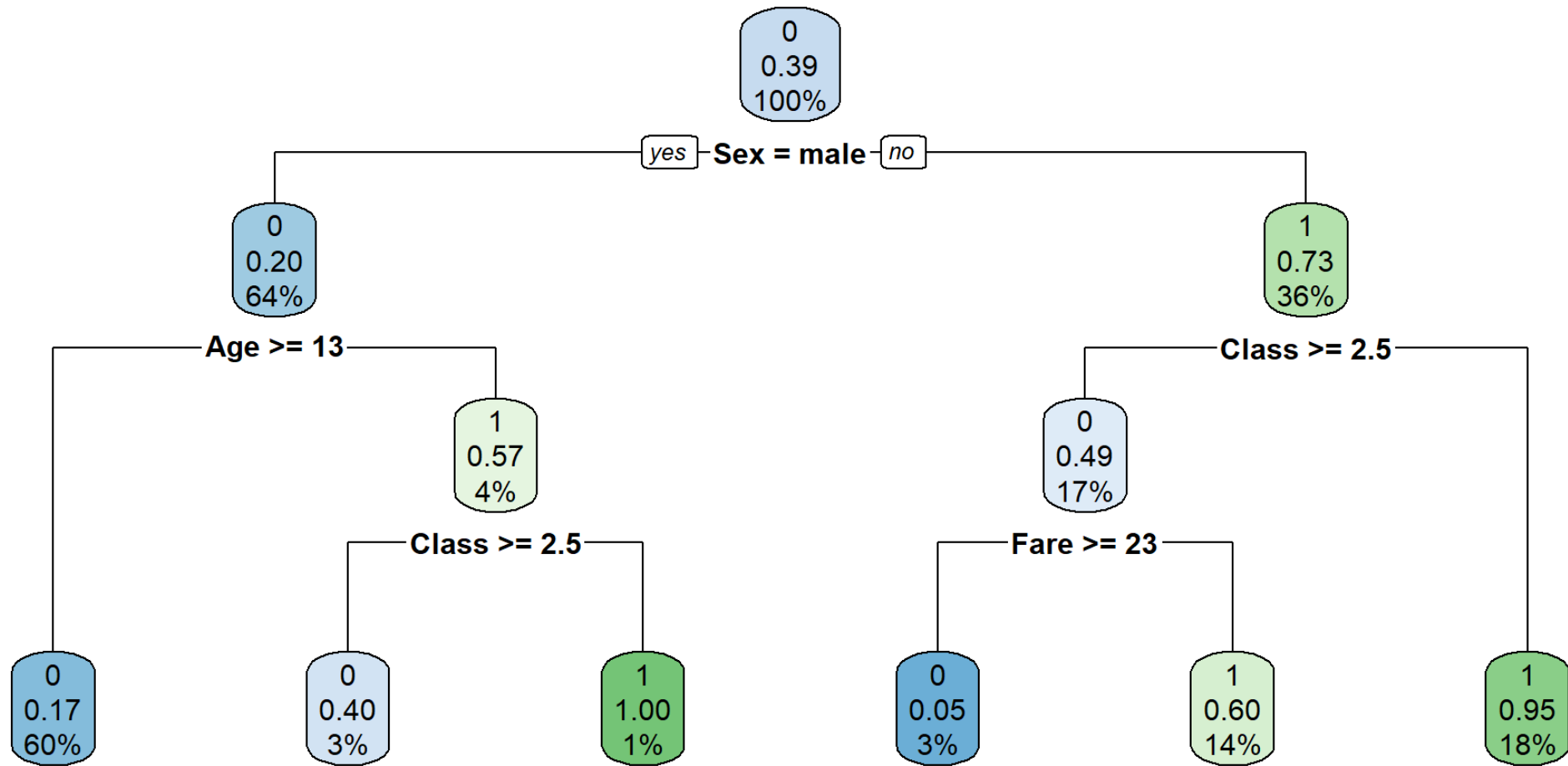
- Nodes are like containers holding all or some of the training data
 - *root* node holds all training observations.
 - moving down the tree *parent nodes* get split into *child nodes*
- **RPart** nodes show three types of information.»

NODES IN THE DECISION TREE – THE OPTIMIZER CREATED DECISION RULES

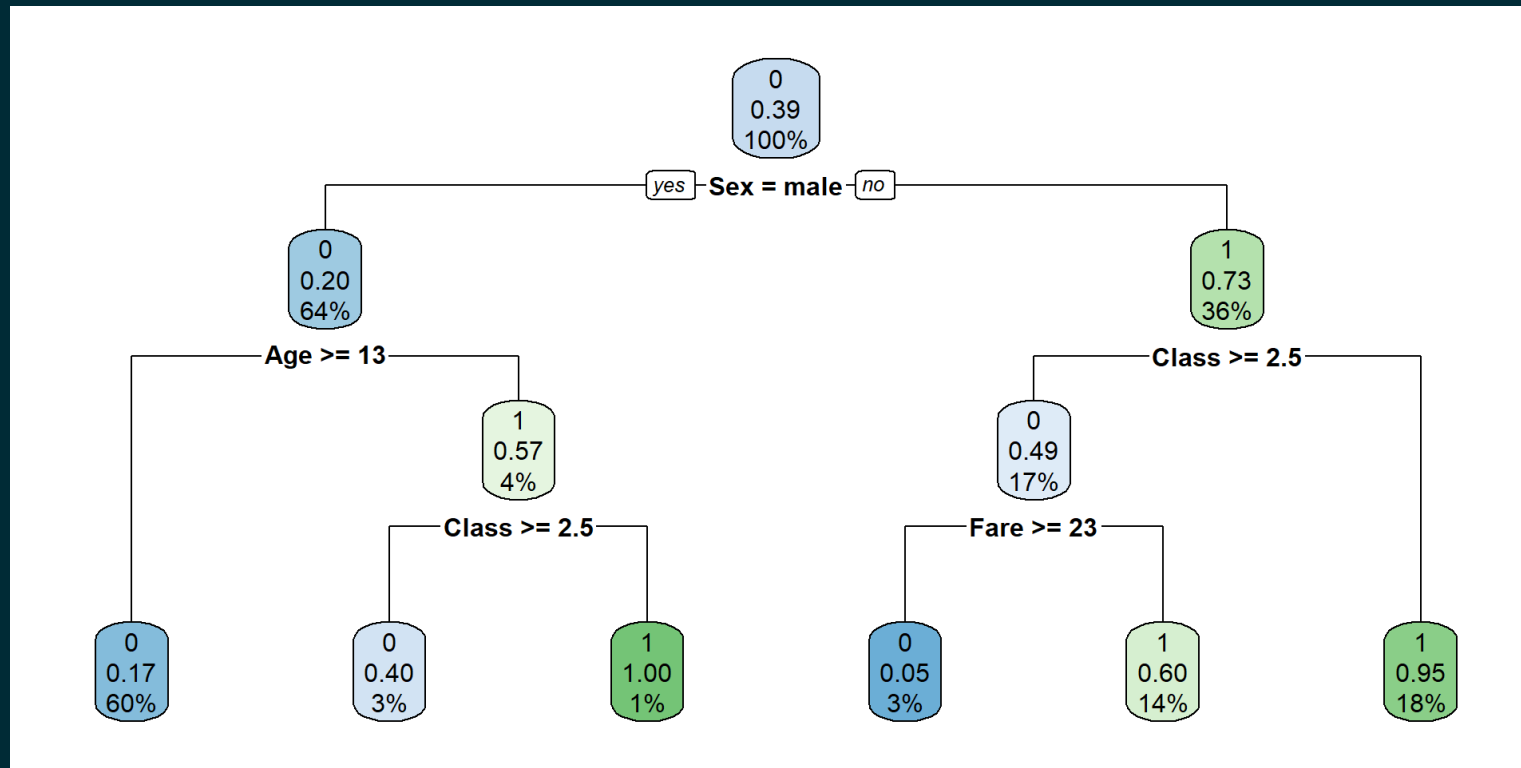
Let us follow the last observation in **DataTrain**

	Survived	Sex	Class	Age	Fare
886	1	male	1	26	30

NODES IN THE DECISION TREE – INTERPRETING TERMINAL NODES

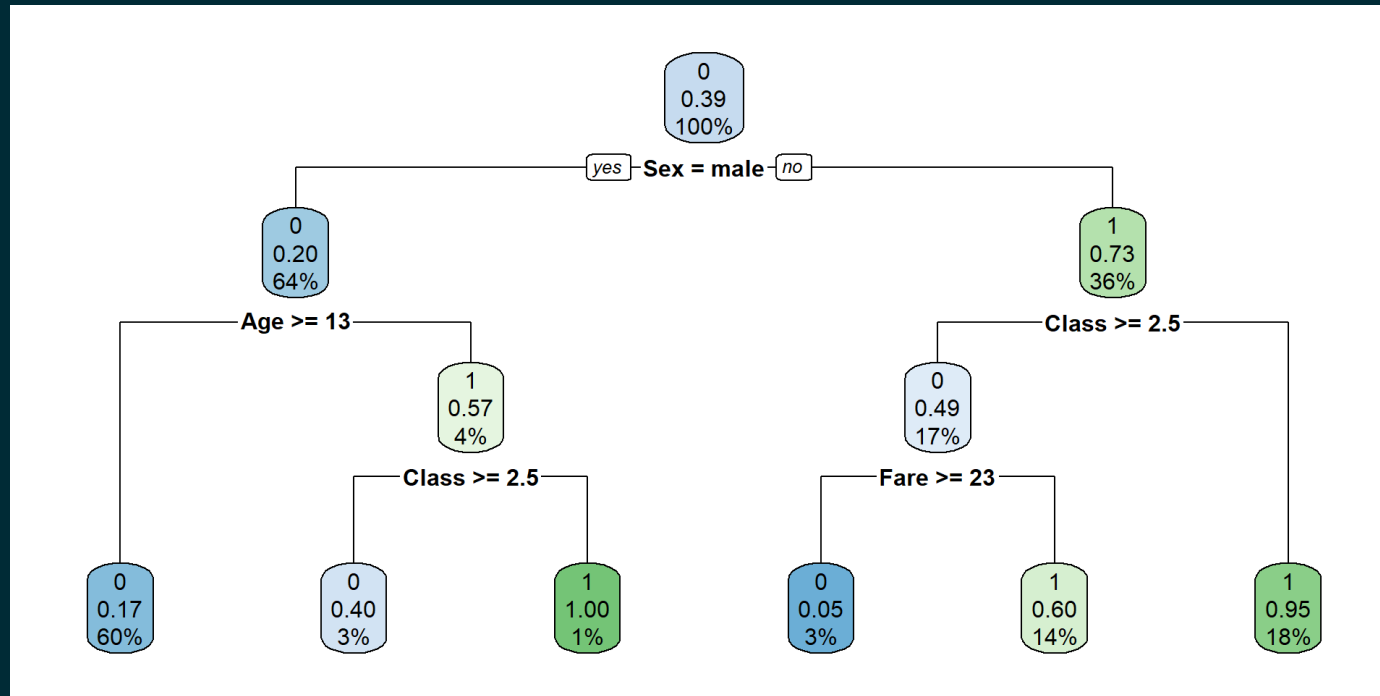


NODES IN THE DECISION TREE – STYLIZED FACTS



1. Adult male passengers, regardless of the class and fare, had only a survival rate of 17%.
2. Female passengers, regardless of age and not considering the class or the fare, had a survival chance of 73%.
3. Considering the class female passengers traveled in (regardless of age), the survival rate was 95% for First or Second Class.

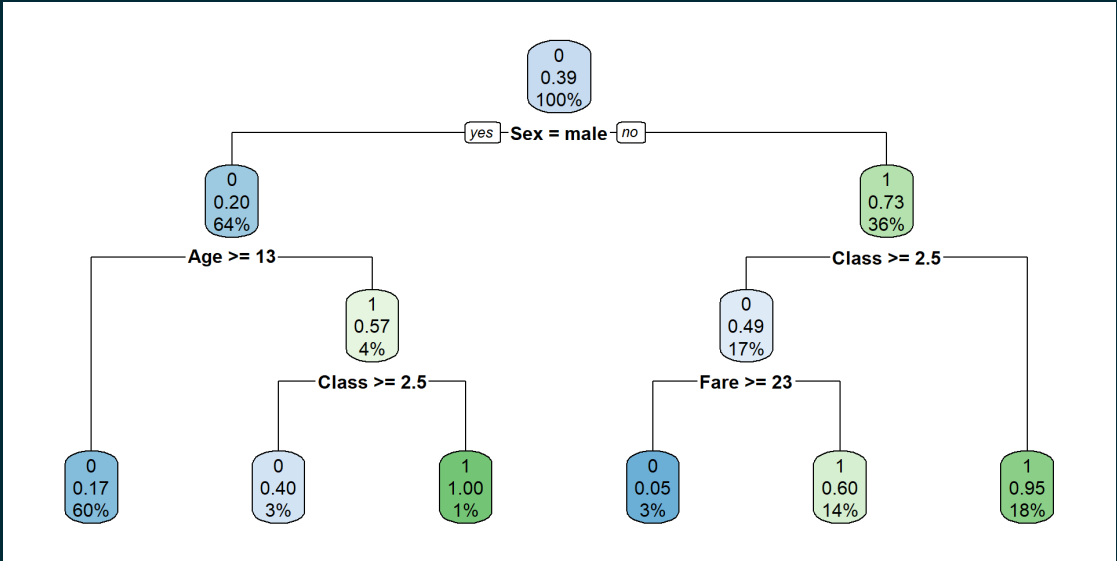
NODES IN THE DECISION TREE – NOT ALL DECISION RULES MAKE SENSE



For example:

- Females traveling in Third Class have a survival rate of 49% (**this makes sense**)
- Next split **does not make much sense**:
 - Fare greater or equal to 23 British Pounds survival rate only 5%.
 - In contrast, lower fare had a survival rate of 60%.»

PREDICTING TESTING DATA WITH A DECISION TREE



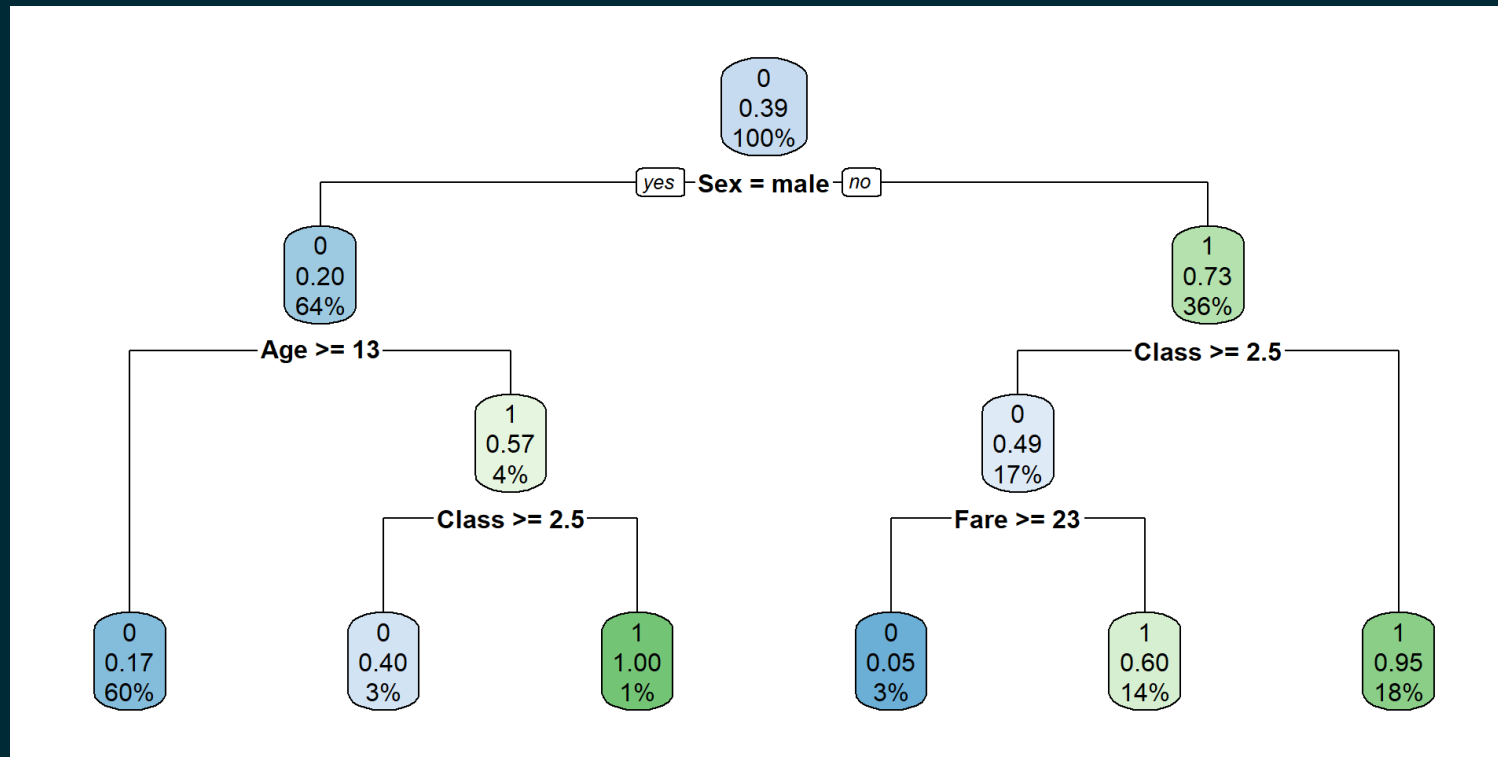
Predicting Survival (9-year Old Boy in Third Class)

Survived	Sex	Class	Age	Fare
1	male	3	9	15.9

Prediction: **Not Survived**

Observation is a **false positive** (0=:positive class)»

PREDICTING ALL TESTING DATA WITH A DECISION TREE – PREDICTION



► Code

```
# A tibble: 6 × 8
  Survived Sex    Class   Age   Fare .pred_class .pred_0 .pred_1
  <fct>    <chr>   <int> <dbl> <dbl> <fct>      <dbl>   <dbl>
1 0      male     3    22  7.25  0          0.83    0.17
2 1    female     1    35 53.1  1          0.0492  0.951
3 0      male     3    27  8.46  0          0.83    0.17
4 1    female     3    27 11.1  1          0.402   0.598
5 0    female     3    31  18    1          0.402   0.598
6 0      male     2    35  26    0          0.83    0.17
```

PREDICTING ALL TESTING DATA WITH A DECISION TREE – METRICS

```
1 DataTestWithPred=augment(WfModelTitanic, new_data = DataTest)
```

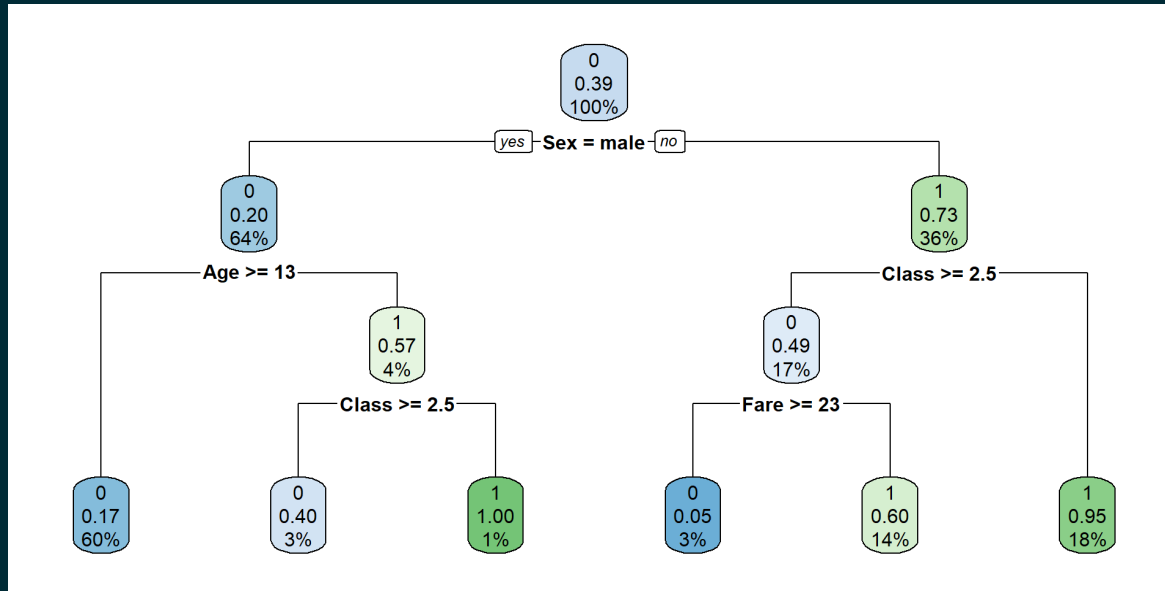
```
1 conf_mat(DataTestWithPred, truth = Survived, estimate = .pred_class)
```

	Truth	
Prediction	0	1
0	123	23
1	14	63

```
1 metricSetTitanic=metric_set(accuracy, sensitivity, specificity)
2 metricSetTitanic(DataTestWithPred, truth = Survived, estimate = .pred_class)
```

```
# A tibble: 3 × 3
  .metric      .estimator .estimate
  <chr>        <chr>         <dbl>
1 accuracy    binary         0.834
2 sensitivity binary         0.898
3 specificity binary         0.733
```

HOW ARE THE DECISION RULES DETERMINED?



The short answer: **by the Optimizer.**

- Decision rules are determined from the top down to the bottom.
- Regardless of decision rule on next level.
 - No turning back reversing decision rule on higher level.
 - *greedy algorithm*
- Decision rules consists of two components:
 - i. the **splitting variable**,
 - ii. the **splitting value** (e.g., **Age** for splitting (here: **Age>=13** for **yes**))

Optimizer compares all *splitting variables* and all possible *splitting values* to find **best decision rule.**»

CRITERIA TO QUANTIFY QUALITY OF DECISION RULES

How can we determine if a decision rule is good?

Common criteria for categorical outcomes:

- *Information Gain*
- *Chi-Square*
- **Gini Impurity** used by **RPart** »

HOW ARE THE DECISION RULES DETERMINED? – GINI IMPURITY CRITERIUM

Gini Impurity is calculated for an individual node and estimates ” (...) the probability that two entities taken at random from the dataset of interest (with replacement) represent (...) different types.”
(Wikipedia contributors. 2022. “Diversity Index – Wikipedia, the Free Encyclopedia.”)

CRITERIA TO ASSESS DECISION RULES – GINI IMPURITY

$$G^{Imp} = 1 - \overbrace{\left(P_{Surv.}^2 + \underbrace{P_{NotSurv.}^2}_{(1-P_{Surv.})^2} \right)}^{\text{Prob. for 2 identical outcomes}}$$

$P_{Surv.} :=$ Proportion Surv.

and

$P_{NotSurv.} :=$ Proportion Not Surv.

CRITERIA TO ASSESS DECISION RULES – GINI IMPURITY

$$G^{Imp} = 1 - (P_{Surv.}^2 + (1 - P_{Surv.})^2)$$

$$G^{Imp} = 1 - P_{Surv.}^2 - 1 + 2P_{Surv.} - P_{Surv.}^2$$

$$G^{Imp} = 2P_{Surv.} - 2P_{Surv.}^2$$

$$G^{Imp} = 2P_{Surv.}(1 - P_{Surv.})$$

QUANTIFYING QUALITY OF DECISION RULES – GINI IMPURITY

$$G^{Imp} = 2P_{Surv.}(1 - P_{Surv.})$$

Purest Possible Node:

- Only *Survived* observations: $P_{Surv.} = 1$ and $(1 - P_{Surv.}) = 0$
- or
- only *Not Survived* observations: $P_{Surv.} = 0$ and $(1 - P_{Surv.}) = 1$
- **In any case:** $G^{Imp} = 0$
(probability of drawing two different outcomes = 0)

QUANTIFYING QUALITY OF DECISION RULES – GINI IMPURITY

$$G^{Imp} = 2P_{Surv.}(1 - P_{Surv.})$$

Impurest Possible Node:

- Equal amount of *Survived* and *Not Survived* observations:
 $P_{Surv.} = 0.5$ and $(1 - P_{Surv.}) = 0.5$
- $G^{Imp} = 2 \cdot 0.25 \cdot 0.25 = 0.5$
(Note, $G^{Imp} = 0.5$ is maximum for Gini Impurity for 2 categories)

Probability of drawing two different outcomes = 0.5

DETERMINING IMPURITY FOR ROOT'S PARENT AND CHILD NODES

REAL WORLD DATA WITH A DECISION TREE

Predicting vaccination rates in the U.S. based on data from September 2021:

- Outcome variable: Percentage of fully vaccinated (two shots) people (*PercVacFull*).
- Data from 2,630 continental U.S. counties.»

REAL WORLD DATA WITH A DECISION TREE – PREDICTOR VARIABLES

- Race/Ethnicity:
 - Counties' proportion African Americans (*PercBlack*),
 - Counties' proportion Asian Americans (*PercAsian*), and
 - Counties' proportion Hispanics (*PercHisp*)
- Political Affiliation (Presidential election 2020):
 - Counties' proportion Republican votes (*PercRep*)
- Age Groups in Counties:
 - Counties' proportion young adults (20-25 years); *PercYoung25*
 - Counties' proportion older adults (65 years and older); *PercOld65*
- Income related:
 - Proportion of households receiving food stamps (*PercFoodSt*)»

LOADING THE DATA AND ASSIGNING TRAINING AND TESTING DATA

► Code

County	State	PercVacFull	PercRep	PercAsian	PercBlack	PercHisp	PercYoung2
Baldwin	AL	0.504	0.7506689	0.0092	0.0917	0.0456	0.063418
Barbour	AL	0.416	0.4911710	0.0048	0.4744	0.0436	0.073464
Chambers	AL	0.315	0.7063935	0.0112	0.3956	0.0238	0.064653
Cherokee	AL	0.318	0.8319460	0.0025	0.0460	0.0159	0.052545
Choctaw	AL	0.648	0.7445596	0.0013	0.4255	0.0041	0.061241
Cleburne	AL	0.319	0.8402859	0.0002	0.0275	0.0246	0.059486

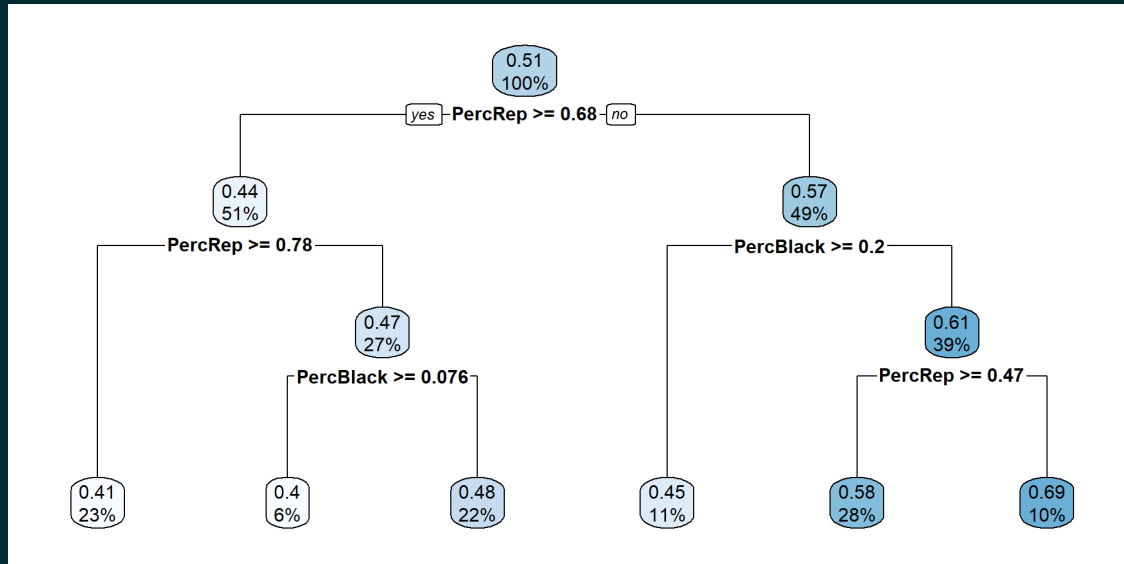
CREATING MODEL DESIGN, RECIPE, AND FITTED WORKFLOW

► Code

```
== Workflow [trained] ==  
Preprocessor: Recipe  
Model: decision_tree()  
  
— Preprocessor —  
0 Recipe Steps  
  
— Model —  
n= 2234  
  
node), split, n, deviance, yval  
  * denotes terminal node  
  
1) root 2234 47.539570 0.5068938  
  2) PercRep>=0.683455 1132 14.614060 0.4410887  
    4) PercRep>=0.7847773 522 5.763314 0.4119579 *  
    5) PercRep< 0.7847773 610 8.028707 0.4660172  
      10) PercBlack>=0 0755 129 2 231454 0 3983725 *
```

DECISION TREE FOR THE VACCINATION MODEL

► Code



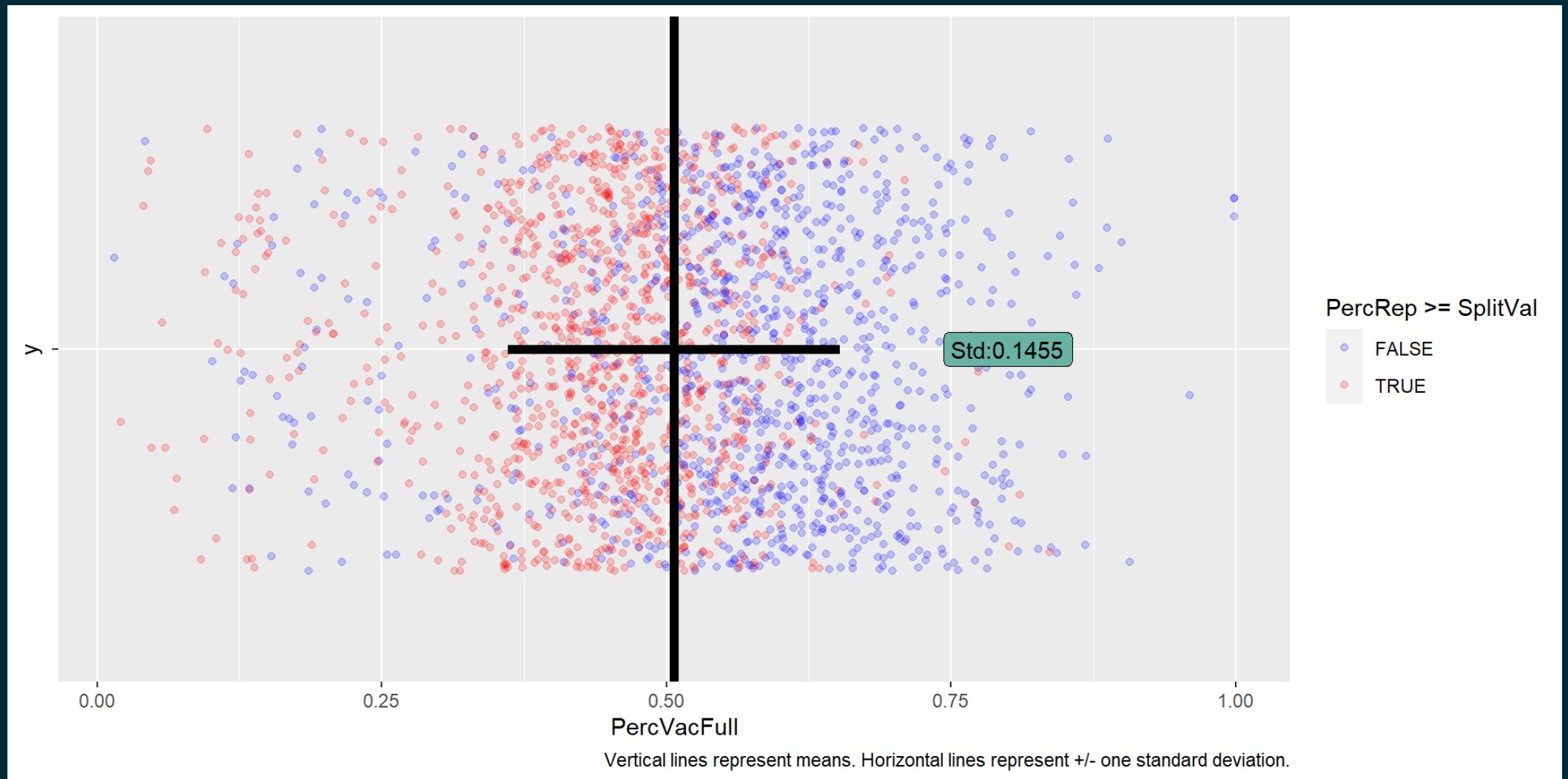
What is difference compared to a classification model?

- Terminal node estimates now continuous variable.
- Variance instead of Gini Impurity

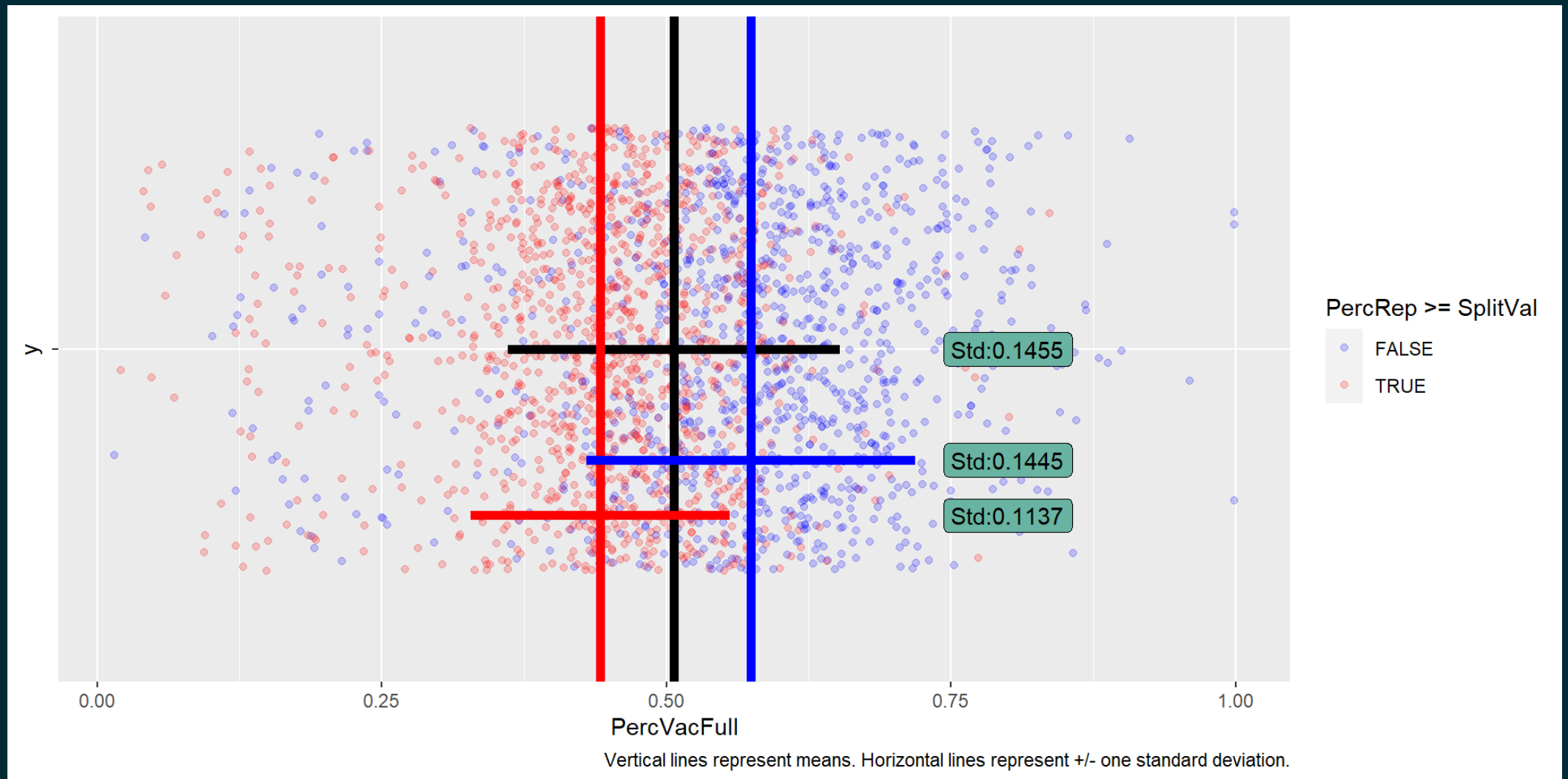
VARIANCE REDUCTION EXPLAINED



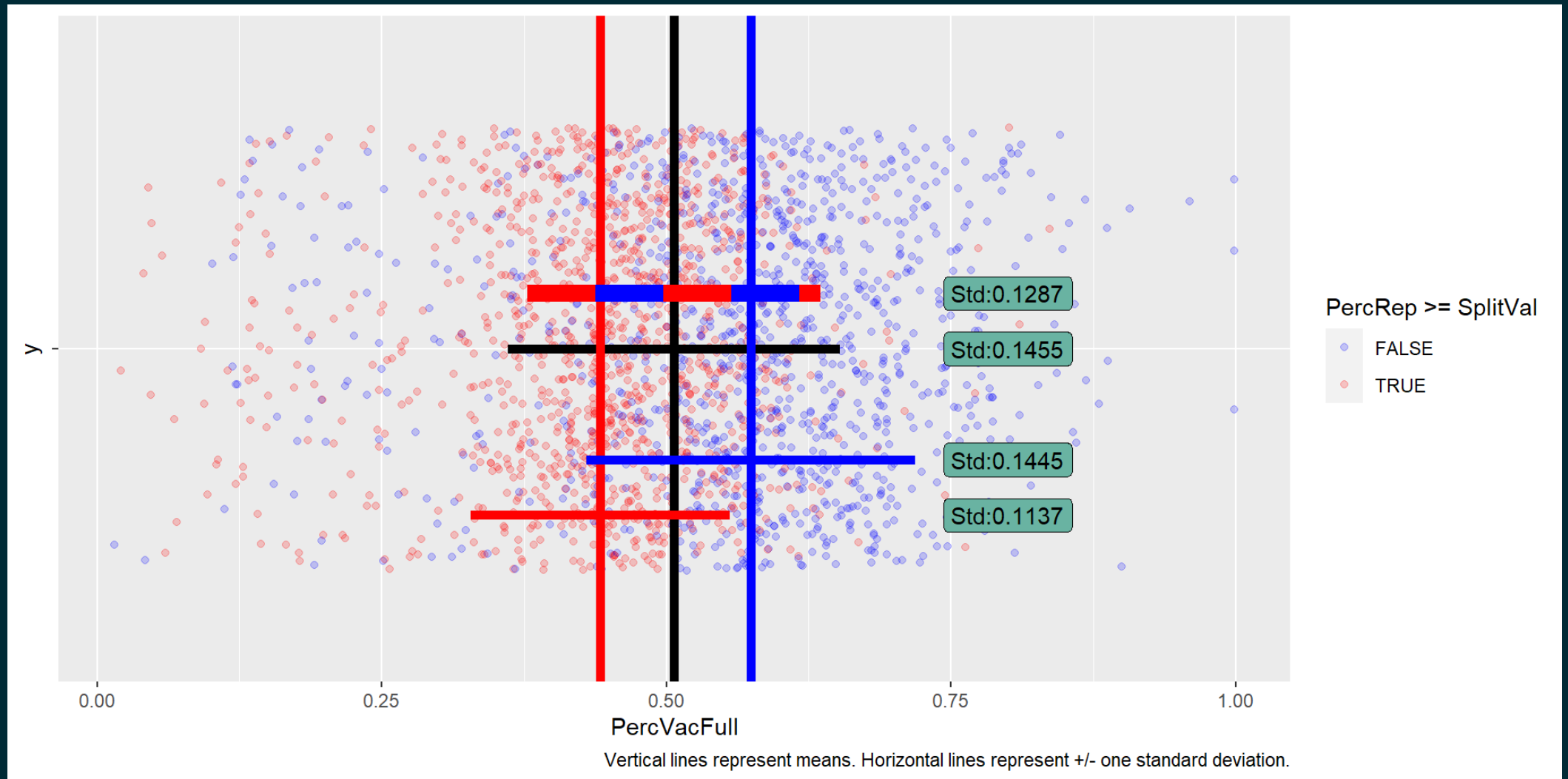
VARIANCE REDUCTION EXPLAINED (PERCREP<0.68)



VARIANCE REDUCTION EXPLAINED (PERCREP<0.68)



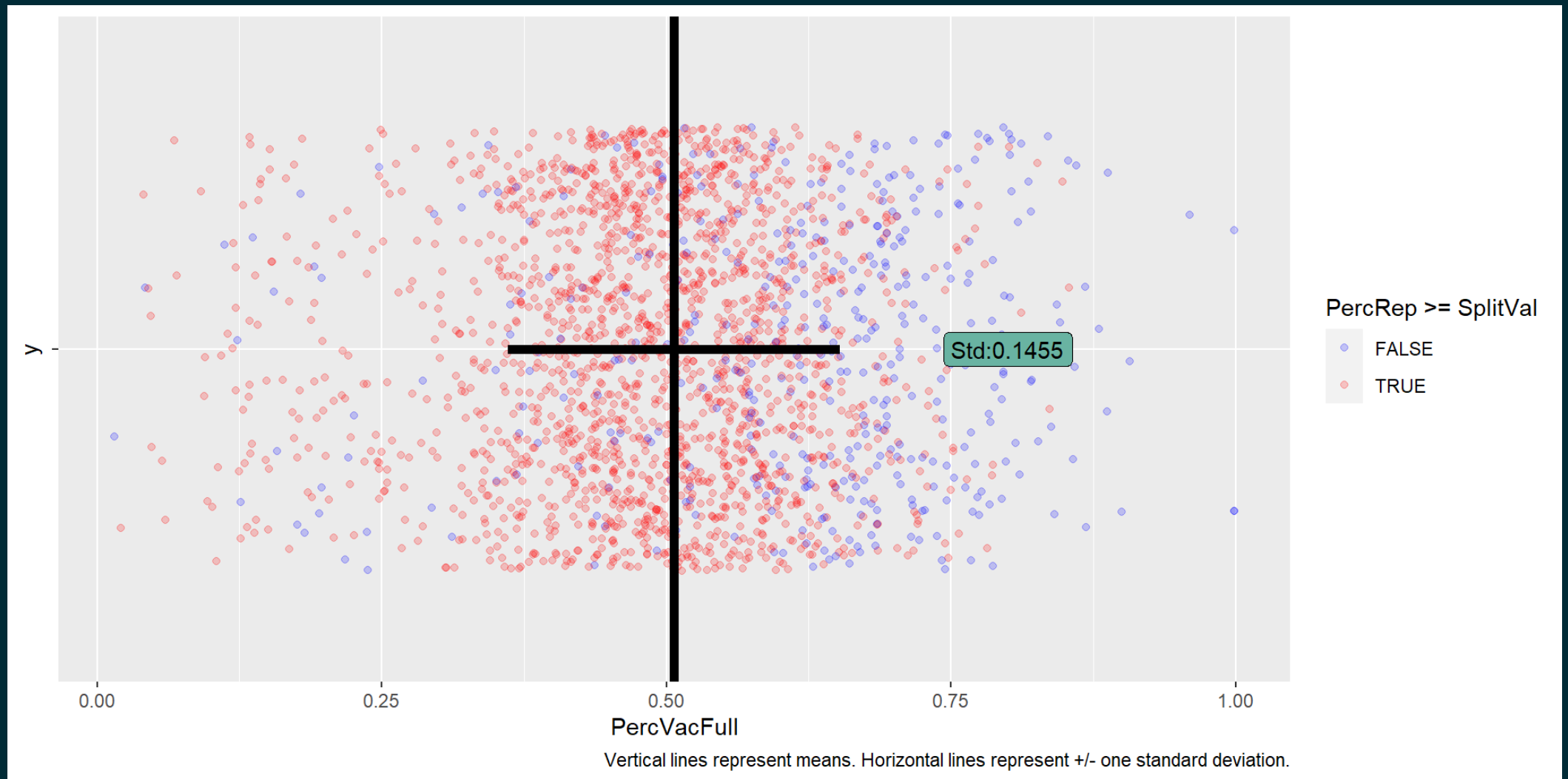
VARIANCE REDUCTION EXPLAINED (PERCREP<0.68)



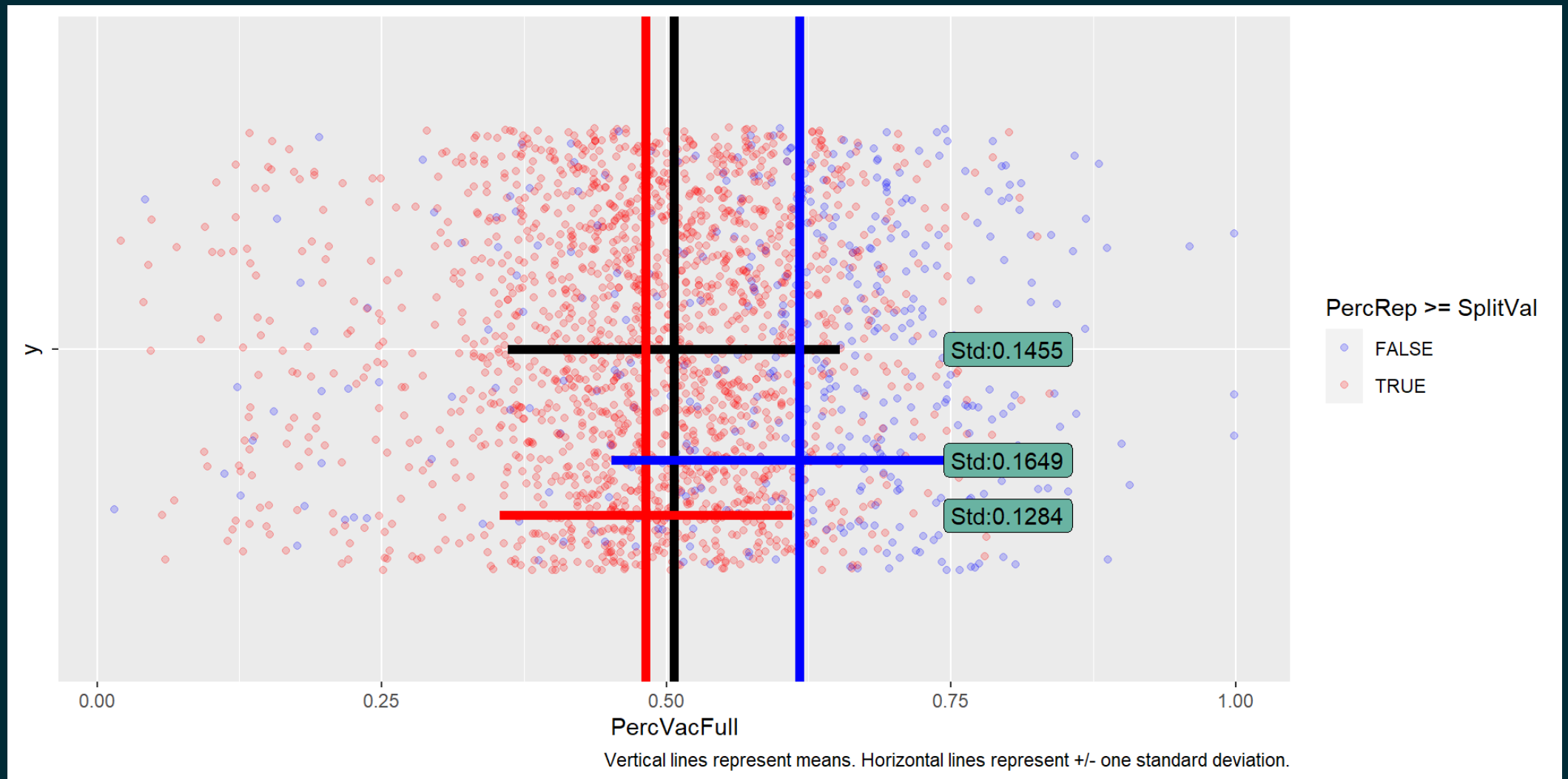
VARIANCE REDUCTION EXPLAINED



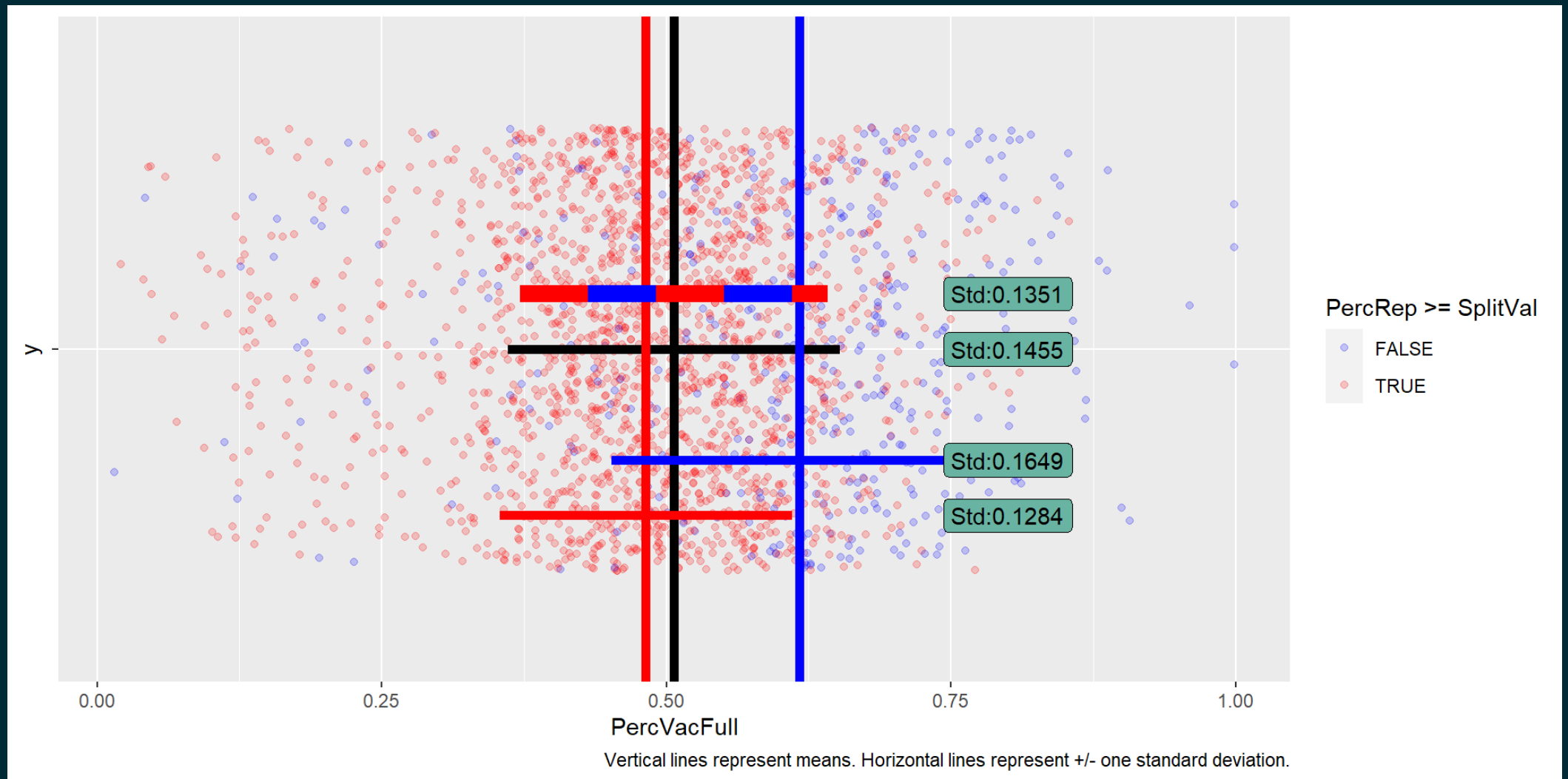
VARIANCE REDUCTION EXPLAINED (PERCREP<0.5)



VARIANCE REDUCTION EXPLAINED (PERCREP<0.5)



VARIANCE REDUCTION EXPLAINED (PERCREP<0.5)



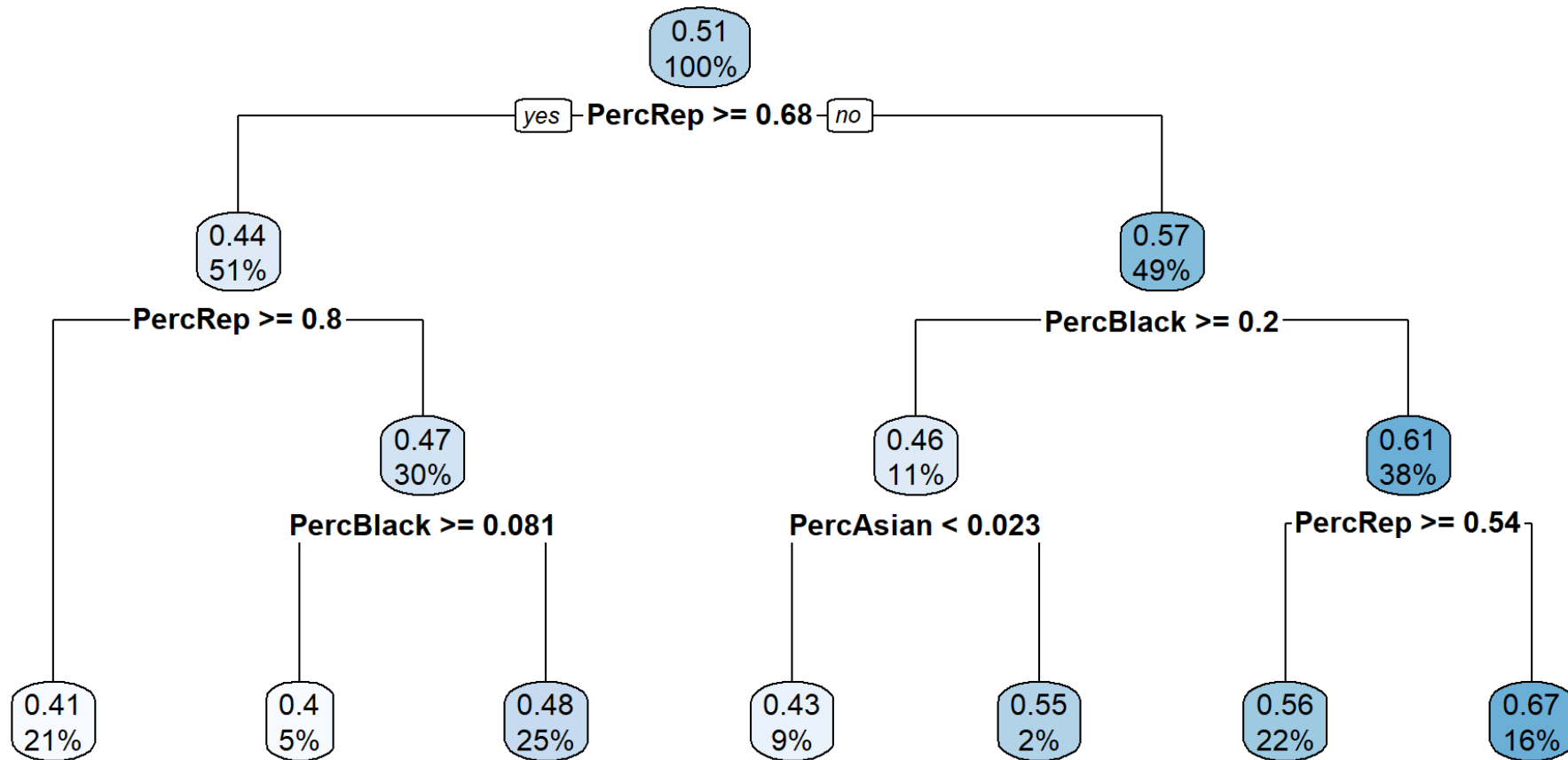
METRICS FOR THE DECISION TREE VACCINATION MODEL

► Code

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse     standard      0.120
2 rsq      standard      0.341
3 mae      standard      0.0852
```

INSTABILITY OF DECISION TREES

► Code



METRICS FOR THE (SLIGHTLY) CHANGED DECISION TREE VACCINATION MODEL

► Code

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse     standard      0.129
2 rsq      standard      0.323
3 mae      standard      0.0918
```

WHEN AND WHEN NOT TO USE DECISION TREES

- As a standalone model *Decision Trees* should not be used for predictions. Here is why:
 - *Decision Trees* respond very sensitively to a change in hyperparameters such as the tree depth. Therefore, the tree's structure can change, which may also change the predictions.
 - *Decision Trees* respond very sensitively to a change in the training data. Therefore, when choosing different training data (e.g., by changing the `set.seed()`), the tree's structure can change, which may also change the predictions.
- Decision Trees have an educational value because the graphical representation provides an easy way to see which variables influenced the predictions.
- Using Decision Trees as components of more advanced models like *Random Forest* and *Boosted Trees* often leads to excellent predictive results.»