Algorithm

```
library(tidyverse)
library(haven)
library(palmerpenguins)
library(gtsummary)
```

Import dataset

```
#PR <- read_spss("BDPR7RFL.SAV")
hr <- read_spss("BDHR7RFL.SAV")

#PR_df <-PR |>
# select(HV226, HV206, HV208, HV243A, HV221, HV209, HV242, HV025, HV220, HV219, HV106,
# rename(fuel= HV226, Electricity = HV206,
      #   Television = HV208, Mobile.phone = HV243A, Landline = HV221,
       #  Refrigerator = HV209, separate.kitchen = HV242, residence = HV025, age = HV220,
        # sex = HV219, education = HV106, marital.status = HV115, work.status = SH13,
         #mutate(Cooking.fuel = cut(fuel,
         #                         breaks = c(1,5,10),
         #                         labels = c("Clean Fuel", "Not Clean"),
         #                        right = TRUE))


hr_df <- hr |>
  select(HV226, HV206, HV208, HV243A, HV221, HV209, HV242,HV241, HV025, HV220, HV219, `HV1
  ## Renaming Variable
  rename(fuel= HV226, Electricity = HV206, Television = HV208,
         Mobile.phone = HV243A, Landline = HV221, Refrigerator = HV209,
         separate.kitchen = HV242,  Kitchen = HV241, residence = HV025, age = HV220,
         sex = HV219, education = `HV106$01`, marital.status = `HV115$01`,
         work.status = `SH13$01`, Wealth.index = HV270, Family.size = HV009) |>

        mutate(cooking.fuel = case_when(fuel <= 5 ~ 1,  ## Categories fuel into two catego
                                        fuel == 6 ~ 0, ## 1= Clean, 0 = Unclean
                                        fuel == 7 ~ 0,
                                        fuel == 8 ~ 0,
                                        fuel == 9 ~ 0,
                                        fuel == 10 ~ 0,
                                        fuel == 11 ~ 0,
                                        TRUE ~ NA),
               sex = case_when(sex == 2 ~ 0,
                               sex == 1 ~ 1),
               residence = case_when(residence == 1 ~ 1,
                                     residence ==2 ~ 0),
               marital.status = case_when(marital.status == 1 ~ 1,
                                          marital.status == 2 ~ 1,# 1 = Yes
                                          marital.status == 0 ~ 0,
                                          marital.status == 3 ~ 0,
                                          marital.status == 4 ~ 0,
                                          marital.status == 5 ~ 0) # 0 = No



        )
```

Multidimentional Energy Poverty Index:

```
hr_mp <- hr_df |>
  select(cooking.fuel, Electricity, Television, Mobile.phone, Landline, Refrigerator, sepa
         Kitchen)

Y <- as.matrix(hr_mp[c(-8)])

head(Y)
```

```
     cooking.fuel Electricity Television Mobile.phone Landline Refrigerator
[1,]            0           0          0             1        0            0
[2,]            0           0          0             1        0            0
[3,]            0           0          0             1        0            0
```

```
[4,]            0           0          0            1        0            0
[5,]            0           0          0            1        0            0
[6,]            0           0          0            1        0            0
     separate.kitchen
[1,]               NA
[2,]               NA
[3,]               NA
[4,]               NA
[5,]               NA
[6,]               NA
```

```
names(hr_mp)
```

```
[1] "cooking.fuel"     "Electricity"      "Television"       "Mobile.phone"    
[5] "Landline"         "Refrigerator"     "separate.kitchen" "Kitchen"         
```

```
M = hr_mp |>
  select(Kitchen,separate.kitchen) |>
  mutate(Kitchen = case_when(Kitchen == 2 ~ "Build",
                             Kitchen == 3 ~ "outdoor",
                             Kitchen == 6 ~ "outdoor",
                             Kitchen == 1 ~ "Indoor"),
         sp = if_else(separate.kitchen == 0, 0,1, missing = 2),

         sp.kit = case_when( sp == 0 & Kitchen == "outdoor" ~ 1,
                             sp == 2 & Kitchen == "outdoor" ~ 1,
                             sp == 1 & Kitchen == "Indoor" ~ 0,
                             sp == 2 & Kitchen == "Build" ~ 0)

         )



table(M$sp.kit)
```

```

    0     1 
14571  4487 
```

Univariate Analysis

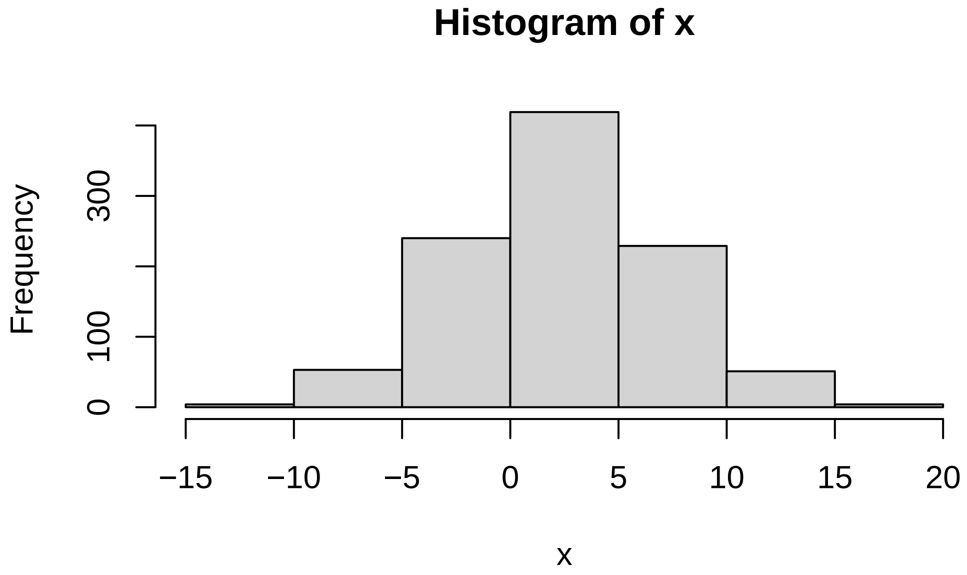
```
hr_a <- hr_df |>
  select(cooking.fuel,Electricity, Television, Mobile.phone, Landline, Refrigerator, separ
  mutate_all(as.numeric, as.factor) |>
  mutate(across(1:7,as.factor)) |>
  tbl_summary()


hr_a
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 19,457**
cooking.fuel	
0	15,435 (79%)
1	3,983 (21%)
Unknown	39
Electricity	
0	3,643 (19%)
1	15,814 (81%)
Television	
0	10,223 (53%)
1	9,234 (47%)
Mobile.phone	
0	1,049 (5.4%)
1	18,408 (95%)
Landline	
0	19,341 (99%)
1	116 (0.6%)
Refrigerator	
0	13,711 (70%)
1	5,746 (30%)
separate.kitchen	
0	387 (67%)
1	194 (33%)
Unknown	18,876

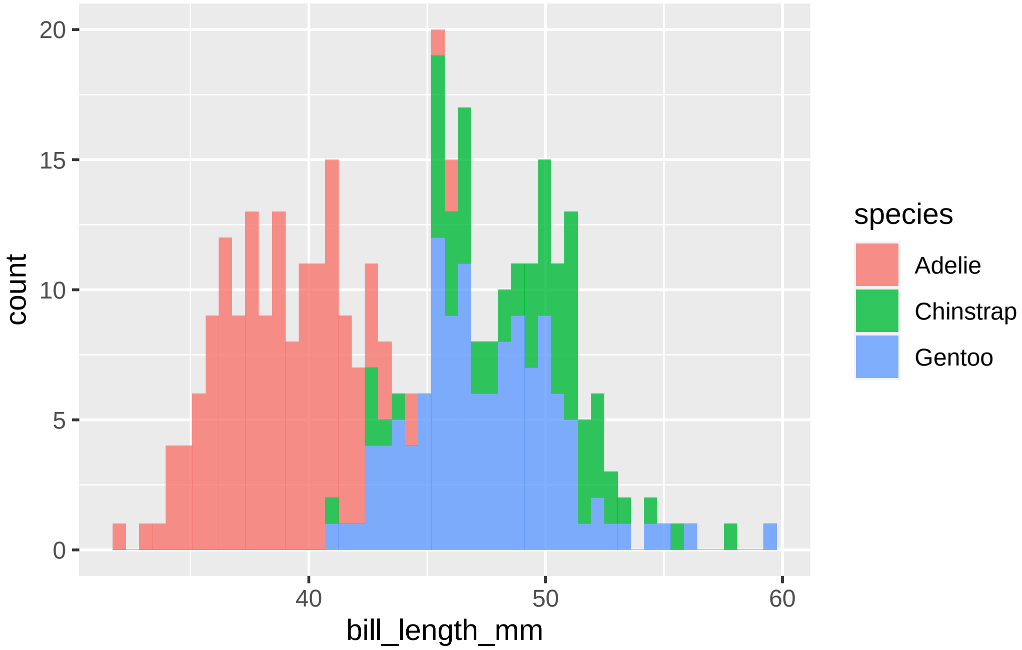
Generate data from Normal Distribution

```
x <- rnorm(1000,2,5)
hist(x)
```



```
penguins |>
  ggplot(aes(x= bill_length_mm, fill = species))+
  geom_histogram(bins = 50, alpha=0.8)
```

```
Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

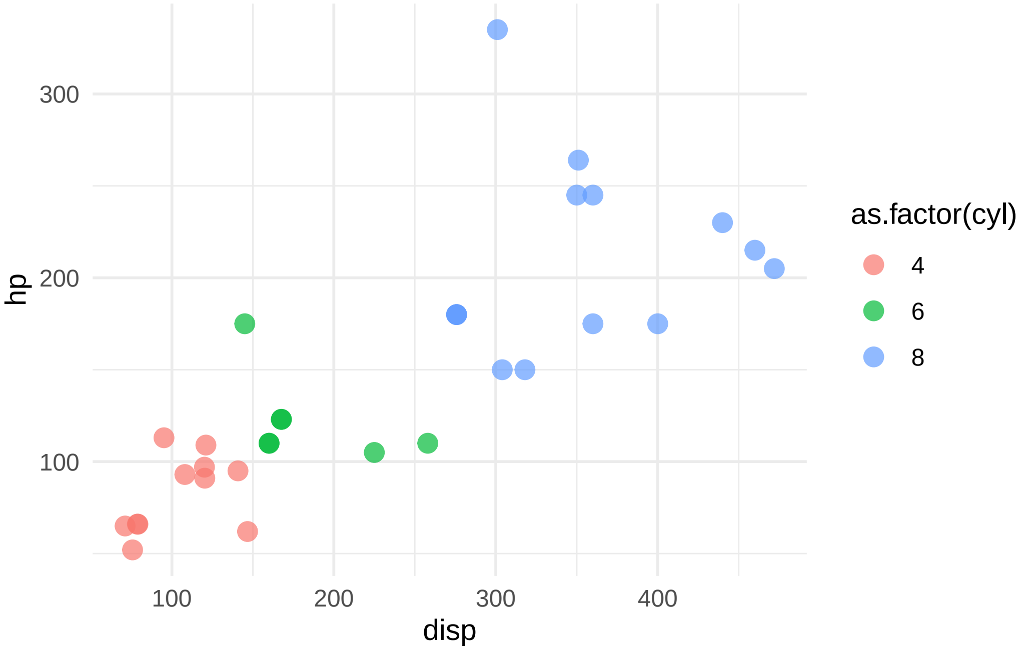


Data Cleaning

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

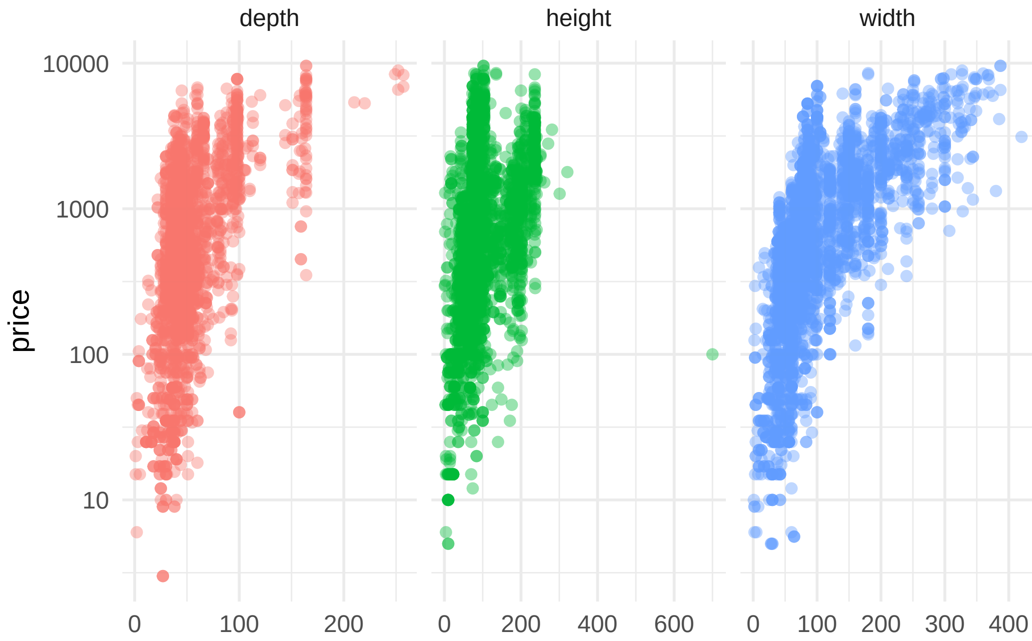
```
ggplot(mtcars,aes(x= disp,y=hp,col=as.factor(cyl)))+
  geom_point(alpha=0.7,size=3)+
  theme_minimal()
```



```r
library(tidyverse)
ikea <- read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/dat
```

```
New names:
Rows: 3694 Columns: 14
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(7): name, category, old_price, link, other_colors, short_description, d... dbl
(6): ...1, item_id, price, depth, height, width lgl (1): sellable_online
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

```r
ikea <- rename(ikea, id = ...1)

ikea %>%
  select(id, price, depth:width) %>%
  pivot_longer(depth:width, names_to = "dim") %>%
  ggplot(aes(value, price, color = dim)) +
  geom_point(alpha = 0.4, show.legend = FALSE) +
  scale_y_log10() +
  facet_wrap(~dim, scales = "free_x") +
  labs(x = NULL) +
  theme_minimal()
```



```
ikea_df <- ikea %>%
  select(price, name, category, depth, height, width) %>%
  mutate(price = log10(price)) %>%
  mutate_if(is.character, factor)

ikea_df
```

```
# A tibble: 3,694 x 6
   price name                    category      depth height width
   <dbl> <fct>                   <fct>         <dbl>  <dbl> <dbl>
 1  2.42 FREKVENS                Bar furniture    NA     99    51
 2  3.00 NORDVIKEN               Bar furniture    NA    105    80
 3  3.32 NORDVIKEN / NORDVIKEN Bar furniture    NA     NA    NA
 4  1.84 STIG                    Bar furniture    50    100    60
 5  2.35 NORBERG                 Bar furniture    60     43    74
 6  2.54 INGOLF                Bar furniture    45      91     40
 7  2.11 FRANKLIN              Bar furniture    44      95     50
 8  2.29 DALFRED               Bar furniture    50      NA     50
 9  2.11 FRANKLIN              Bar furniture    44      95     50
10  3.34 EKEDALEN / EKEDALEN   Bar furniture    NA      NA     NA
# i 3,684 more rows
```

#Building Model

```
## Build Model


library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.1.1 --

v broom        1.0.5     v rsample      1.2.0
v dials        1.2.0     v tune         1.1.2
v infer        1.0.5     v workflows    1.1.3
v modeldata    1.2.0     v workflowsets 1.0.1
v parsnip      1.1.1     v yardstick    1.2.0
v recipes      1.0.8

-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x recipes::all_double()  masks gtsummary::all_double()
x recipes::all_factor()  masks gtsummary::all_factor()
x recipes::all_integer() masks gtsummary::all_integer()
x recipes::all_logical() masks gtsummary::all_logical()
x recipes::all_numeric() masks gtsummary::all_numeric()
x scales::discard()      masks purrr::discard()
x dplyr::filter()        masks stats::filter()
x recipes::fixed()       masks stringr::fixed()
x dplyr::lag()           masks stats::lag()
x yardstick::spec()      masks readr::spec()
x recipes::step()        masks stats::step()
* Use tidymodels_prefer() to resolve common conflicts.
```

```
set.seed(123)
ikea_split <- initial_split(ikea_df, strata = price)
ikea_train <- training(ikea_split)
```

```r
ikea_test <- testing(ikea_split)

set.seed(234)
ikea_folds <- bootstraps(ikea_train, strata = price)
ikea_folds
```

```
# Bootstrap sampling using stratification
# A tibble: 25 x 2
   splits             id
   <list>             <chr>
 1 <split [2770/994]>  Bootstrap01
 2 <split [2770/1003]> Bootstrap02
 3 <split [2770/1037]> Bootstrap03
 4 <split [2770/1010]> Bootstrap04
 5 <split [2770/1014]> Bootstrap05
 6 <split [2770/1007]> Bootstrap06
 7 <split [2770/1036]> Bootstrap07
 8 <split [2770/1016]> Bootstrap08
 9 <split [2770/1021]> Bootstrap09
10 <split [2770/1043]> Bootstrap10
# i 15 more rows
```

```r
library(usemodels)
use_ranger(price ~ ., data = ikea_train)
```

```
ranger_recipe <-
  recipe(formula = price ~ ., data = ikea_train)

ranger_spec <-
  rand_forest(mtry = tune(), min_n = tune(), trees = 1000) %>%
  set_mode("classification") %>%
  set_engine("ranger")

ranger_workflow <-
  workflow() %>%
  add_recipe(ranger_recipe) %>%
  add_model(ranger_spec)

set.seed(67013)
ranger_tune <-
```

```
  tune_grid(ranger_workflow, resamples = stop("add your rsample object"), grid = stop("add nu
```

```
## lots of options, like use_xgboost, use_glmnet, etc


library(textrecipes)
ranger_recipe <-
  recipe(formula = price ~ ., data = ikea_train) %>%
  step_other(name, category, threshold = 0.01) %>%
  step_clean_levels(name, category) %>%
  step_impute_knn(depth, height, width)

ranger_spec <-
  rand_forest(mtry = tune(), min_n = tune(), trees = 1000) %>%
  set_mode("regression") %>%
  set_engine("ranger")

ranger_workflow <-
  workflow() %>%
  add_recipe(ranger_recipe) %>%
  add_model(ranger_spec)

set.seed(8577)
doParallel::registerDoParallel()
ranger_tune <-
  tune_grid(ranger_workflow,
    resamples = ikea_folds,
    grid = 11
  )
```

```
i Creating pre-processing data to finalize unknown parameter: mtry
```

```
show_best(ranger_tune, metric = "rmse")
```

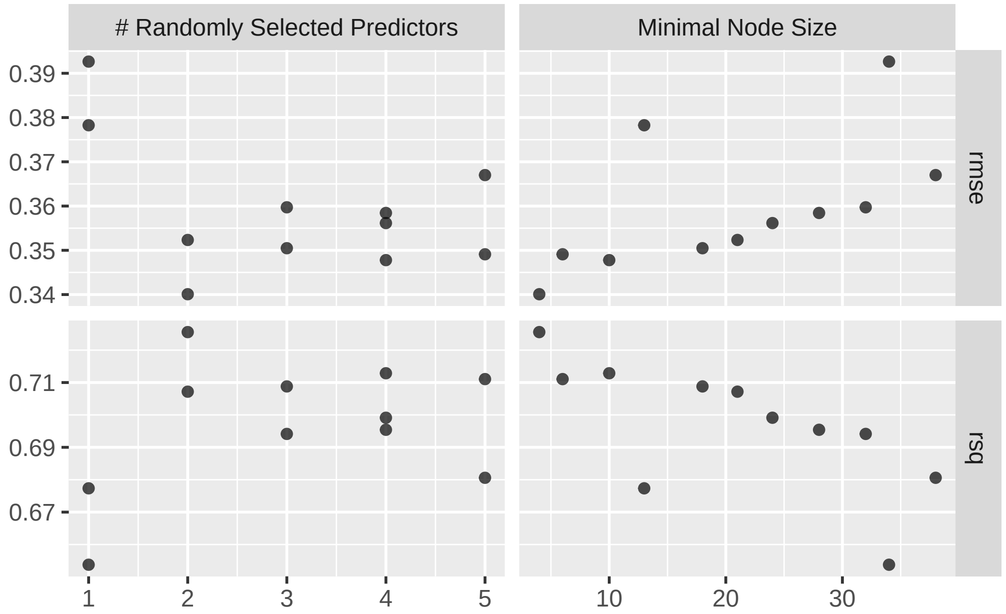
```
# A tibble: 5 x 8
   mtry min_n .metric .estimator  mean     n std_err .config
  <int> <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
1     2     4 rmse    standard   0.340    25 0.00203 Preprocessor1_Model10
2     4    10 rmse    standard   0.348    25 0.00226 Preprocessor1_Model05
```

```
3     5     6 rmse    standard   0.349    25 0.00235 Preprocessor1_Model06
4     3    18 rmse    standard   0.350    25 0.00218 Preprocessor1_Model01
5     2    21 rmse    standard   0.352    25 0.00200 Preprocessor1_Model08
```

```
show_best(ranger_tune, metric = "rsq")
```

```
# A tibble: 5 x 8
   mtry min_n .metric .estimator  mean     n std_err .config
  <int> <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
1     2     4 rsq     standard   0.726    25 0.00332 Preprocessor1_Model10
2     4    10 rsq     standard   0.713    25 0.00372 Preprocessor1_Model05
3     5     6 rsq     standard   0.711    25 0.00385 Preprocessor1_Model06
4     3    18 rsq     standard   0.709    25 0.00368 Preprocessor1_Model01
5     2    21 rsq     standard   0.707    25 0.00347 Preprocessor1_Model08
```

```
autoplot(ranger_tune)
```



```
final_rf <- ranger_workflow %>%
  finalize_workflow(select_best(ranger_tune))
```

```
Warning: No value of `metric` was given; metric 'rmse' will be used.
```

```
final_rf
```

```
== Workflow ====================================================================
Preprocessor: Recipe
Model: rand_forest()

-- Preprocessor ----------------------------------------------------------------
3 Recipe Steps

* step_other()
* step_clean_levels()
* step_impute_knn()

-- Model -----------------------------------------------------------------------
Random Forest Model Specification (regression)

Main Arguments:
  mtry = 2
  trees = 1000
  min_n = 4

Computational engine: ranger
```

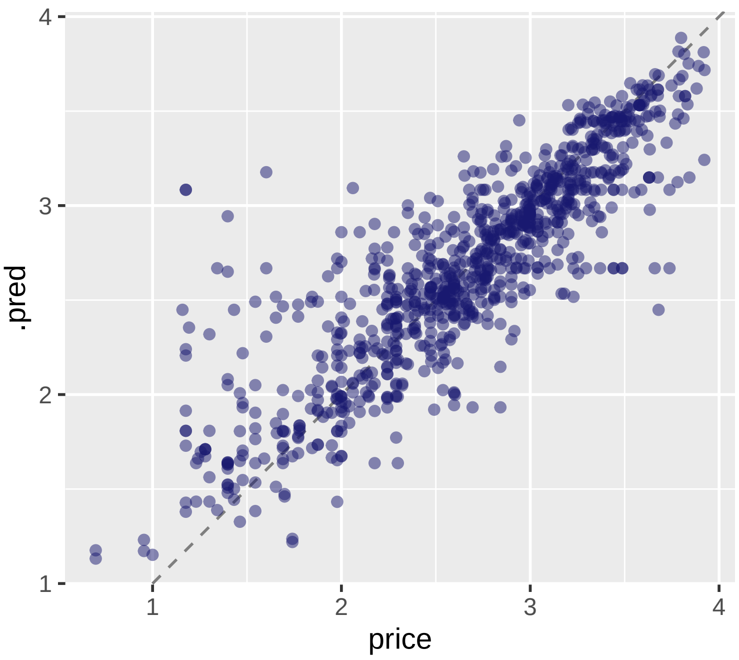
```
ikea_fit <- last_fit(final_rf, ikea_split)
ikea_fit
```

```
# Resampling results
# Manual resampling
# A tibble: 1 x 6
  splits             id               .metrics .notes   .predictions .workflow
  <list>             <chr>            <list>   <list>   <list>       <list>
1 <split [2770/924]> train/test split <tibble> <tibble> <tibble>     <workflow>
```

```
collect_metrics(ikea_fit)
```

```
# A tibble: 2 x 4
  .metric .estimator .estimate .config
  <chr>   <chr>          <dbl> <chr>
1 rmse    standard       0.318 Preprocessor1_Model1
2 rsq     standard       0.753 Preprocessor1_Model1
```

```
collect_predictions(ikea_fit) %>%
  ggplot(aes(price, .pred)) +
  geom_abline(lty = 2, color = "gray50") +
  geom_point(alpha = 0.5, color = "midnightblue") +
  coord_fixed()
```



```
predict(ikea_fit$.workflow[[1]], ikea_test[15, ])
```

```
# A tibble: 1 x 1
  .pred
  <dbl>
1  2.42
```

```r
library(vip)
```

```
Attaching package: 'vip'
```

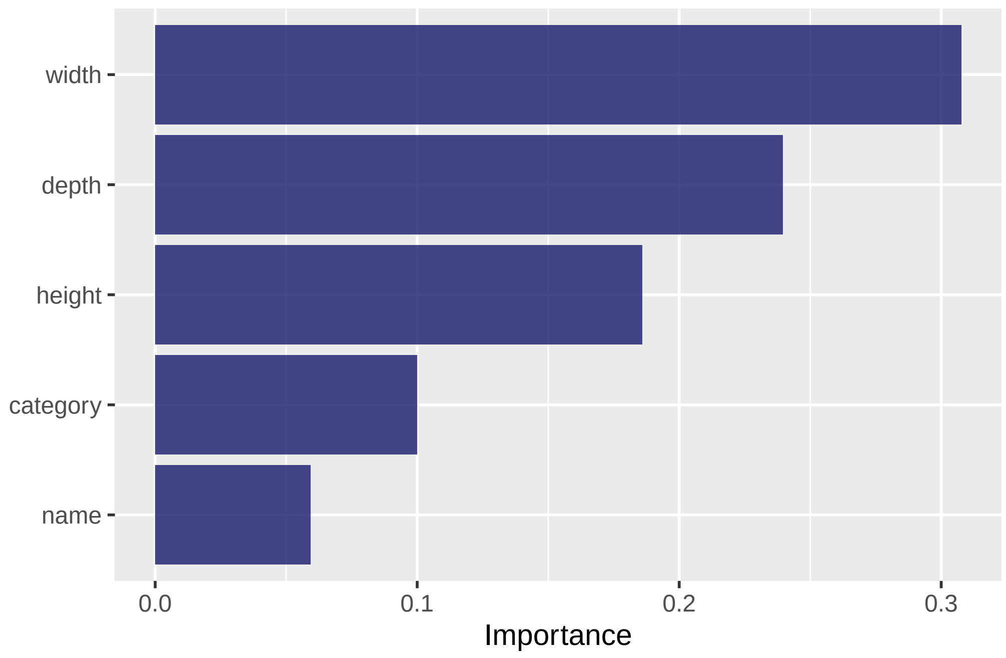
```
The following object is masked from 'package:utils':

    vi
```

```r
imp_spec <- ranger_spec %>%
  finalize_model(select_best(ranger_tune)) %>%
  set_engine("ranger", importance = "permutation")
```

```
Warning: No value of `metric` was given; metric 'rmse' will be used.
```

```r
workflow() %>%
  add_recipe(ranger_recipe) %>%
  add_model(imp_spec) %>%
  fit(ikea_train) %>%
  pull_workflow_fit() %>%
  vip(aesthetics = list(alpha = 0.8, fill = "midnightblue"))
```

```
Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
i Please use `extract_fit_parsnip()` instead.
```


width
depth
height
category
name
0.0
0.1
0.2
0.3
Importance