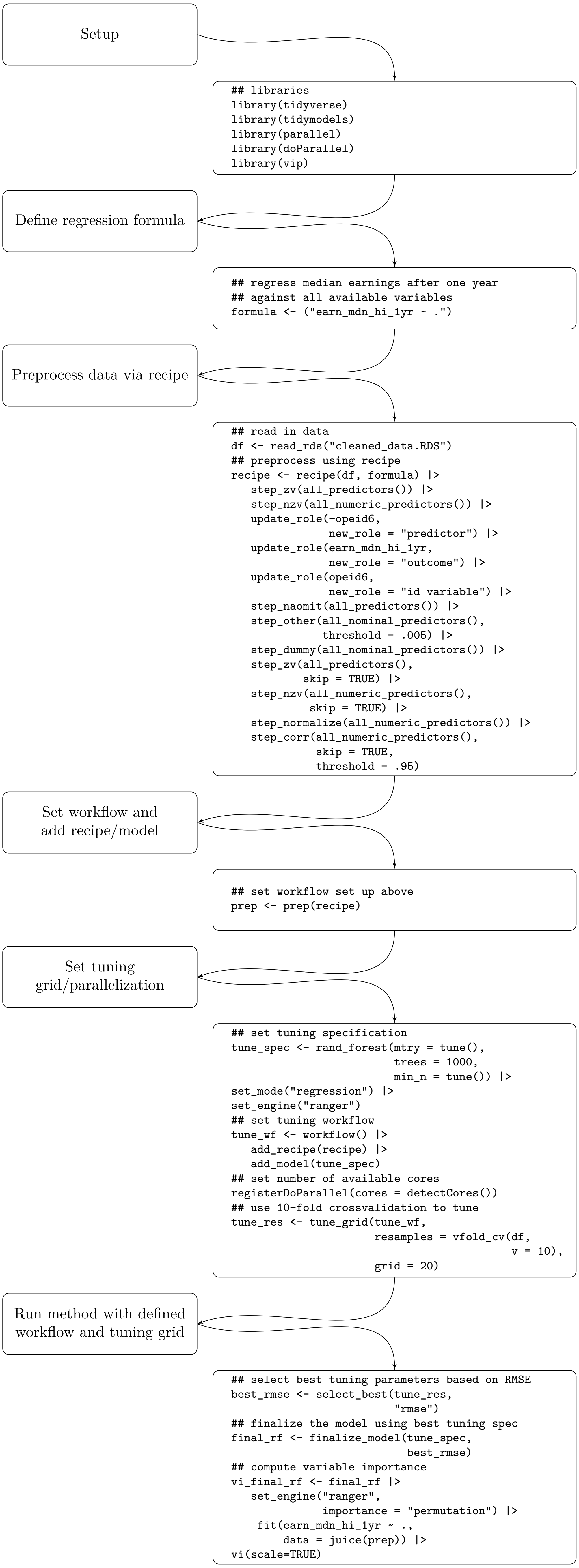

Setup

```
## libraries
library(tidyverse)
library(tidymodels)
library(parallel)
library(doParallel)
library(vip)
```

Define regression formula

```
## regress median earnings after one year
## against all available variables
formula <- ("earn_mdn_hi_1yr ~ .")
```

Preprocess data via recipe

```
## read in data
df <- read_rds("cleaned_data.RDS")
## preprocess using recipe
recipe <- recipe(df, formula) |>
   step_zv(all_predictors()) |>
   step_nzv(all_numeric_predictors()) |>
   update_role(-opeid6,
               new_role = "predictor") |>
   update_role(earn_mdn_hi_1yr,
               new_role = "outcome") |>
   update_role(opeid6,
               new_role = "id variable") |>
   step_naomit(all_predictors()) |>
   step_other(all_nominal_predictors(),
              threshold = .005) |>
   step_dummy(all_nominal_predictors()) |>
   step_zv(all_predictors(),
           skip = TRUE) |>
   step_nzv(all_numeric_predictors(),
            skip = TRUE) |>
   step_normalize(all_numeric_predictors()) |>
   step_corr(all_numeric_predictors(),
             skip = TRUE,
             threshold = .95)
```

Set workflow and add recipe/model

```
## set workflow set up above
prep <- prep(recipe)
```

Set tuning grid/parallelization

```
## set tuning specification
tune_spec <- rand_forest(mtry = tune(),
                         trees = 1000,
                         min_n = tune()) |>
set_mode("regression") |>
set_engine("ranger")
## set tuning workflow
tune_wf <- workflow() |>
   add_recipe(recipe) |>
   add_model(tune_spec)
## set number of available cores
registerDoParallel(cores = detectCores())
## use 10-fold crossvalidation to tune
tune_res <- tune_grid(tune_wf,
                      resamples = vfold_cv(df,
                                           v = 10),
                      grid = 20)
```

Run method with defined workflow and tuning grid

```
## select best tuning parameters based on RMSE
best_rmse <- select_best(tune_res,
                         "rmse")
## finalize the model using best tuning spec
final_rf <- finalize_model(tune_spec,
                           best_rmse)
## compute variable importance
vi_final_rf <- final_rf |>
   set_engine("ranger",
              importance = "permutation") |>
    fit(earn_mdn_hi_1yr ~ .,
        data = juice(prep)) |>
vi(scale=TRUE)
```