Assignment 10

Will Doyle

Complete the following steps:

1. Using the counties dataset (`pd.Rdata`), create a model that predicts median household income (`median_hh_inc`).

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## Registered S3 method overwritten by 'tune':
##   method                   from
##   required_pkgs.model_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --

## v broom        0.7.9      v rsample      0.1.0
## v dials        0.0.10     v tune         0.1.6
## v infer        1.0.0      v workflows    0.2.4
## v modeldata    0.1.1      v workflowsets 0.1.0
## v parsnip      0.1.7      v yardstick    0.0.8
## v recipes      0.1.17

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## 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.
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

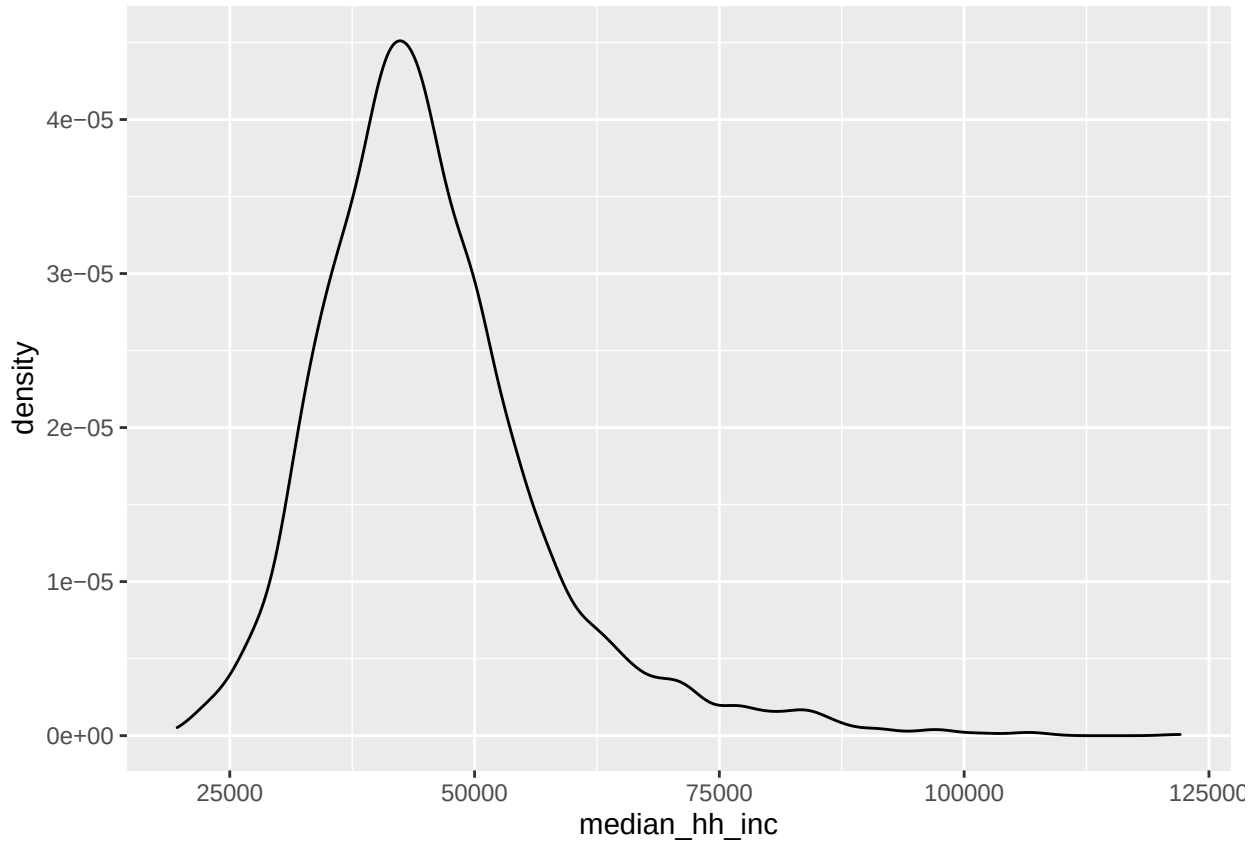
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-2
```

```
load("pd.RData")

pd<-pd%>%
  select(median_home_val,median_hh_inc,coll_grad_pc,homeown_rate,per_capita_inc,pop65p,retail_percap)%>%
  mutate_all(.funs=list(as.numeric))
```

```
pd%>%
  ggplot(aes(x=median_hh_inc))+
  geom_density()
```



```
lm_fit <-
  linear_reg() %>%
  set_engine("lm")
```

```
lm_formula<-as.formula("median_hh_inc~.")
```

```
lm_rec <- recipe(lm_formula, data = pd) %>%
  step_log(all_outcomes(),offset=1)%>%
  step_zv(all_numeric()) %>% # drop any zero variance
  step_naomit(all_predictors())
```

```
lm_workflow<-workflow()%>%
  add_recipe(lm_rec)%>%
  add_model(lm_fit)
```

2. Provide the results of a 10-fold cross validation of your model. Describe what the results mean in a few clear sentences. Plot the results.

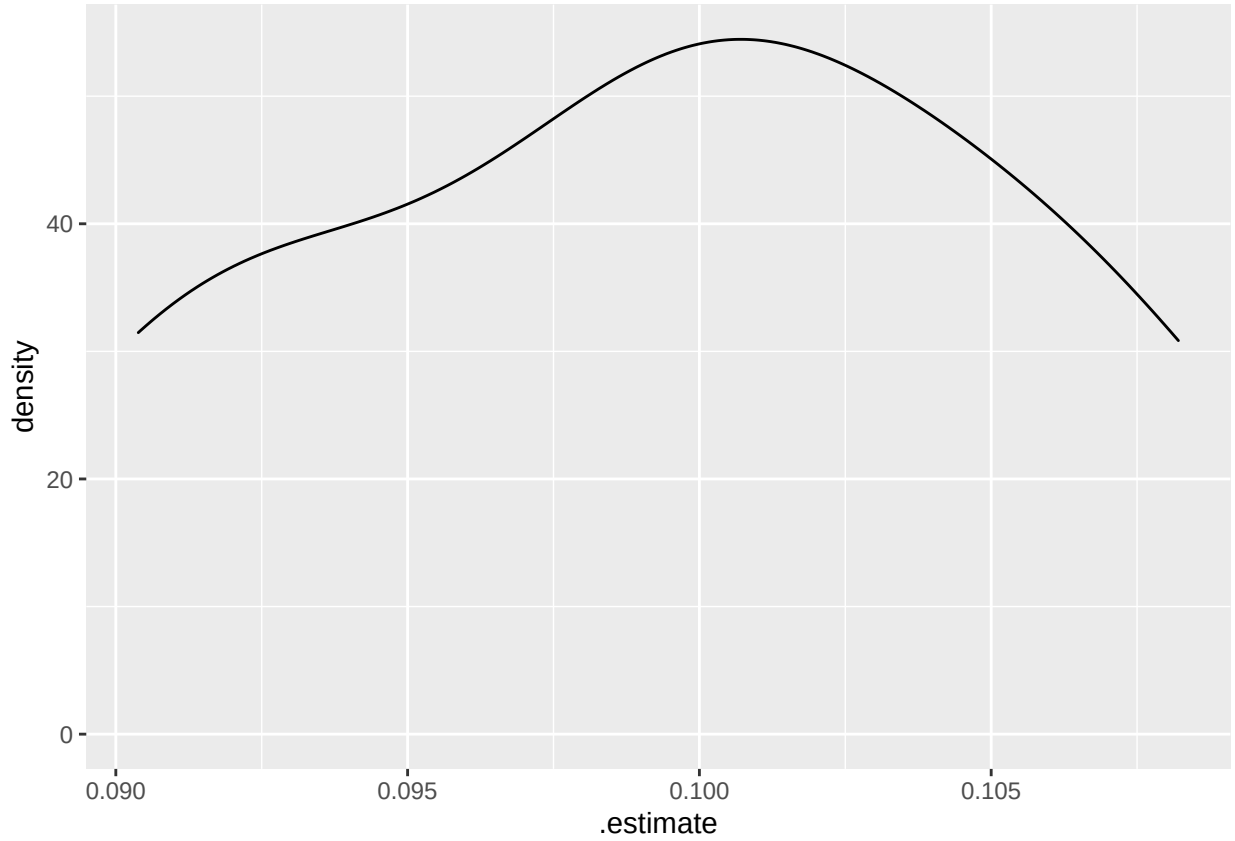
```
pd_kfold_rs<-vfold_cv(pd,v=10)
```

```
lm_mc_results<-
  fit_resamples(
    lm_workflow,
    pd_kfold_rs, ##resampling plan
    control=control_resamples(save_pred = TRUE)
  )
```

```
lm_mc_results%>%
  collect_metrics()
```

```
## # A tibble: 2 x 6
##   .metric .estimator   mean     n std_err .config
##   <chr>   <chr>       <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   0.0994    10 0.00193 Preprocessor1_Model1
## 2 rsq     standard   0.830     10 0.00747 Preprocessor1_Model1
```

```
lm_mc_results%>%
  unnest(.metrics)%>%
  filter(.metric=="rmse")%>%
  ggplot(aes(x=.estimate))+
  geom_density()
```



3. Provide the results of a 1000 repetition monte carlo validation with 10 percent (`prop=.9`) leave out rate for the random partition. Describe what the results mean in a few clear sentences. Plot the results.

```
pd_mc_rs<-mc_cv(pd,times = 100,prop = .9) ##1000 is usual minimum
```

Fit Monte Carlo Resampling

```
lm_mc_results<-
  fit_resamples(
    lm_workflow,
    pd_mc_rs, ##resampling plan
    control=control_resamples(save_pred = TRUE)
  )
```

```
lm_mc_results%>%
  collect_metrics()
```

```
## # A tibble: 2 x 6
##   .metric .estimator   mean     n  std_err .config
##   <chr>   <chr>       <dbl> <int>    <dbl> <chr>
## 1 rmse    standard   0.0992   100 0.000735 Preprocessor1_Model1
## 2 rsq     standard   0.832    100 0.00234  Preprocessor1_Model1
```

```
lm_mc_results%>%
  unnest(.metrics)%>%
  filter(.metric=="rmse")%>%
  ggplot(aes(x=.estimate))+
  geom_density()
```