fishing

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## -- Attaching packages --------------------------------------- tidymodels 0.1.3 --
## v broom        0.7.6      v rsample      0.0.9
## v dials        0.0.9      v tune         0.1.5
## v infer        0.5.4      v workflows    0.2.2
## v modeldata    0.1.0      v workflowsets 0.0.2
## v parsnip      0.1.5      v yardstick    0.0.8
## v recipes      0.1.16
## -- 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.
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last
## The following object is masked from 'package:purrr':
##
##     keep
## The following object is masked from 'package:stats':
##
##     nobs
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

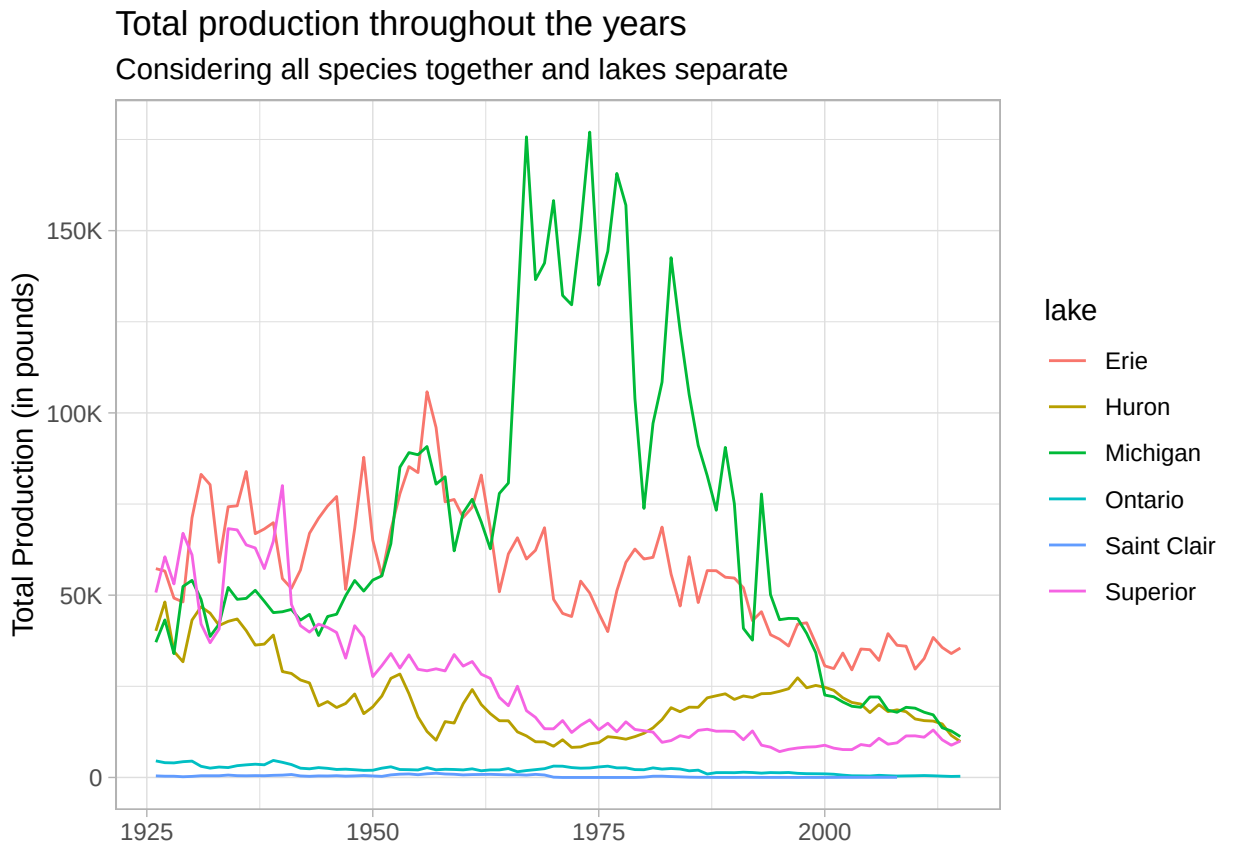
#Let's load the data

```
## Parsed with column specification:
## cols(
##   year = col_double(),
##   lake = col_character(),
##   species = col_character(),
##   grand_total = col_double(),
##   comments = col_character(),
##   region = col_character(),
##   values = col_double()
## )
```

Some data exploratory analysis

```
fishing %>%
  group_by(year, lake) %>%
  summarise(year_production = sum(values, na.rm = T)) %>%
  ggplot(aes(year, year_production)) +
  geom_line(aes(color = lake)) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL,
       y = "Total Production (in pounds)",
       title = "Total production throughout the years",
       subtitle = "Considering all species together and lakes separate")
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```



#It is not feasible to analyze the behaviour of each species on each lake.

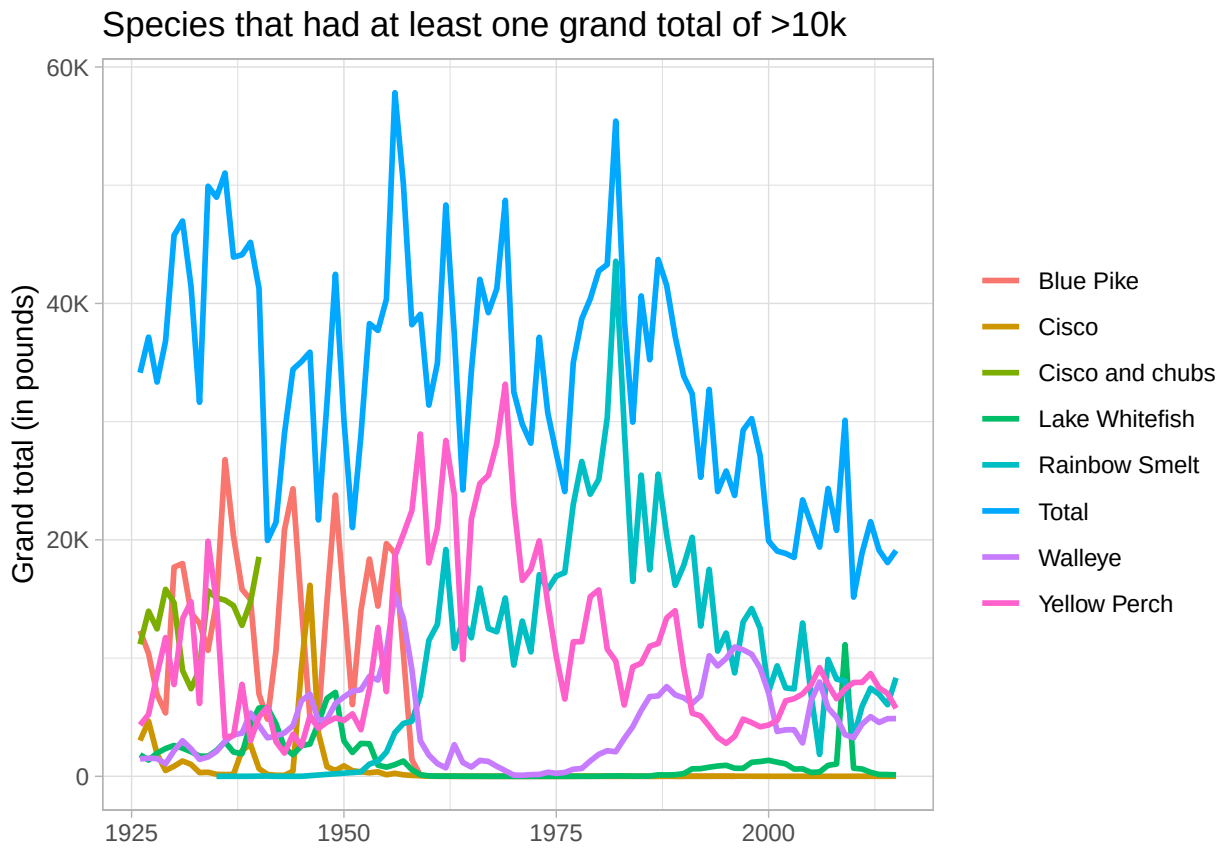
```
fishing %>%
  group_by(year, lake, species) %>%
  summarise(year_production = sum(values, na.rm = T))
```

```
## `summarise()` has grouped output by 'year', 'lake'. You can override using the `.groups` argument.
```

```
## # A tibble: 8,682 x 4
## # Groups:   year, lake [533]
##     year lake  species                       year_production
##    <dbl> <chr> <chr>                                   <dbl>
##  1  1926 Erie  Blue Pike                               21655
##  2  1926 Erie  Bullheads                                  20
##  3  1926 Erie  Burbot                                   564
##  4  1926 Erie  Carp                                    8603
##  5  1926 Erie  Channel Catfish                           10
##  6  1926 Erie  Channel Catfish and Bullheads            1452
##  7  1926 Erie  Cisco                                   4471
##  8  1926 Erie  Freshwater Drum                         2426
##  9  1926 Erie  Lake Sturgeon                            64
## 10  1926 Erie  Lake Whitefish                          2788
## # ... with 8,672 more rows
```

```
grand_total <- fishing %>%
  group_by(species, year) %>%
  slice_head(n = 1) %>%
  select(year, species, grand_total) %>%
```

```
  drop_na() %>%
  ungroup() %>%
  group_by(species) %>%
  mutate(species_max = max(grand_total)) %>%
         #species = ifelse(species_max <= 10000, "Other", species)) %>% View()
  filter(species_max > 10000) %>%
  select(year, species, grand_total)

year_total <- grand_total %>%
  ungroup() %>%
  group_by(year) %>%
  summarise(grand_total = sum(grand_total)) %>%
  mutate(species = "Total")

grand_total <- bind_rows(grand_total, year_total)

grand_total %>%
  ggplot(aes(year, grand_total)) +
  geom_line(aes(color = species), size = 1) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL,
       y = "Grand total (in pounds)",
       color = NULL,
       title = "Species that had at least one grand total of >10k")
```



Let's try to predict the U.S. total production based on the production of Ohio only.

These two plots represent the data used on the model.

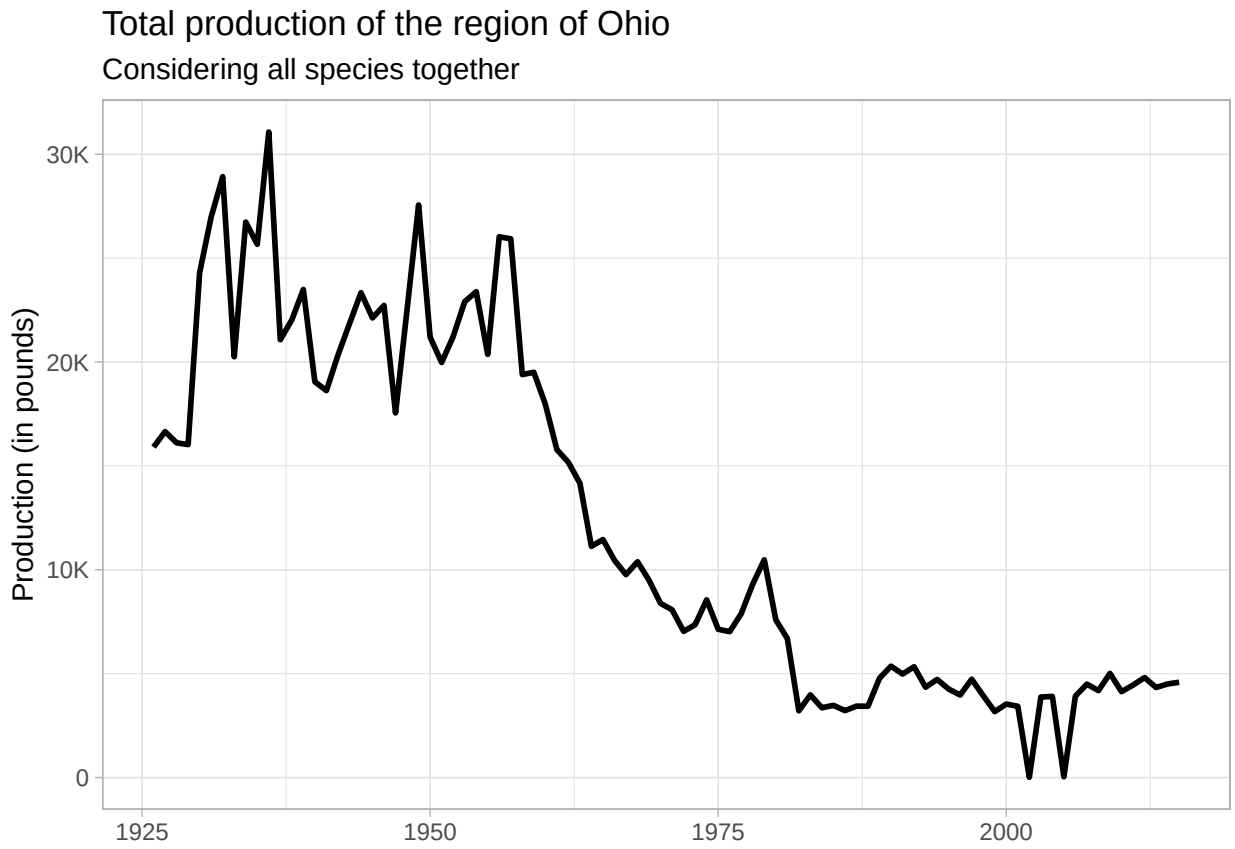
```
us_total_production <- fishing %>%
  filter(region == "U.S. Total") %>%
  group_by(year) %>%
  summarise(us_total_production = sum(values, na.rm = T))

us_total_production %>%
  ggplot(aes(year, us_total_production)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL,
       y = "Total Production (in pounds)",
       title = "U.S. total production throughout the years",
       subtitle = "Considering all species and all lakes together")
```


```
region_production <- fishing %>%
  group_by(year, region) %>%
  summarise(region_production = sum(values, na.rm = T)) %>%
  ungroup() %>%
  group_by(region) %>%
  mutate(region_max_production = max(region_production),
         region_min_production = min(region_production)) %>%
  # This prevents to keep data from regions that did not start fishing activities
  # by the year of 1925.
  filter(region_max_production > 10000, region_min_production > 0) %>%
  select(-region_min_production, -region_max_production) %>%
```

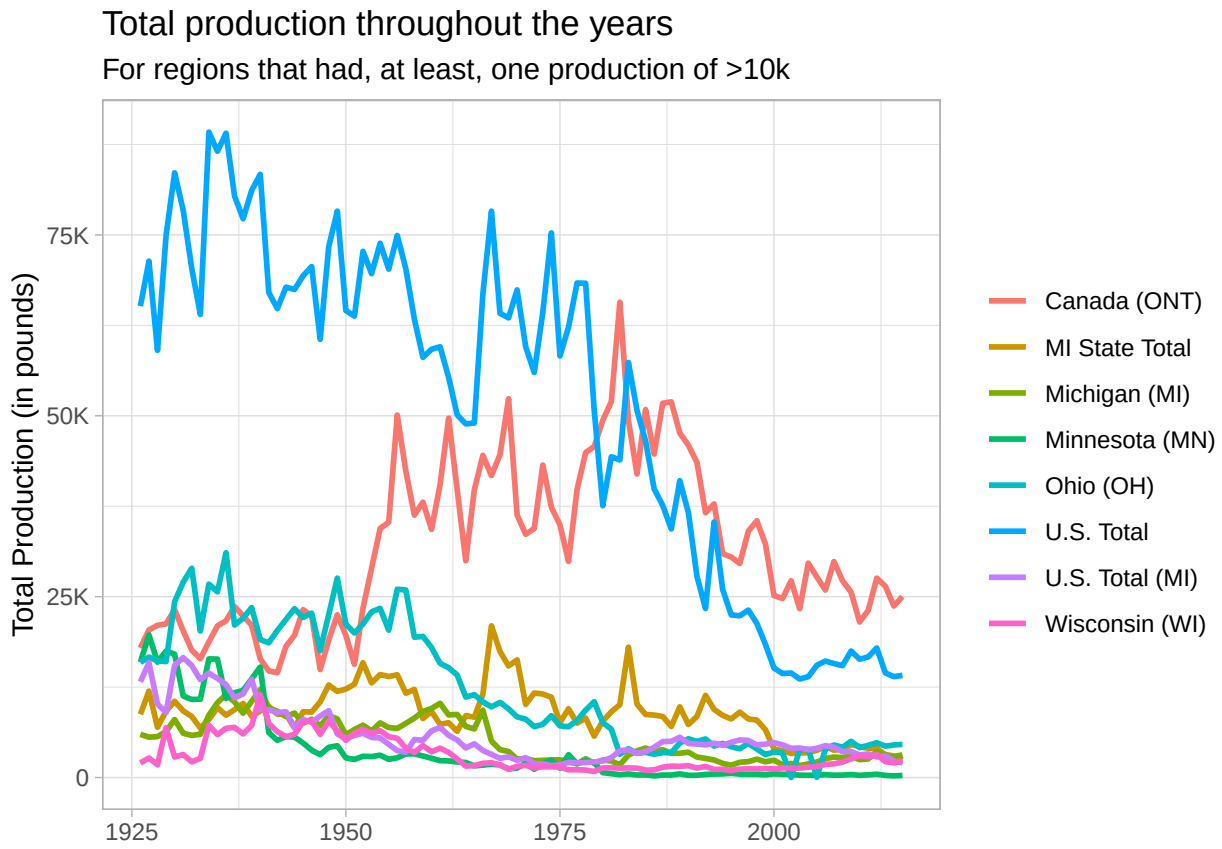
```
  pivot_wider(names_from = region, values_from = region_production)
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

```
# These are the regions that present at least one production of >10k,
#as shown on a previous plot.


region_production %>%
  ggplot(aes(year, `Ohio (OH)`)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL, y = "Production (in pounds)",
       title = "Total production of the region of Ohio",
       subtitle = "Considering all species together")
```




```
region_production %>%
  pivot_longer(cols = !starts_with("year"),
               names_to = "region", values_to = "year_production") %>%
  ggplot(aes(year, year_production)) +
  geom_line(aes(color = region), size = 1) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL,
       y = "Total Production (in pounds)",
       title = "Total production throughout the years",
       subtitle = "For regions that had, at least, one production of >10k",
       color = NULL)
```



Now, let's define training and testing data.

```
data <- initial_split(region_production)

train_production <- training(data)
test_production <- testing(data)
```

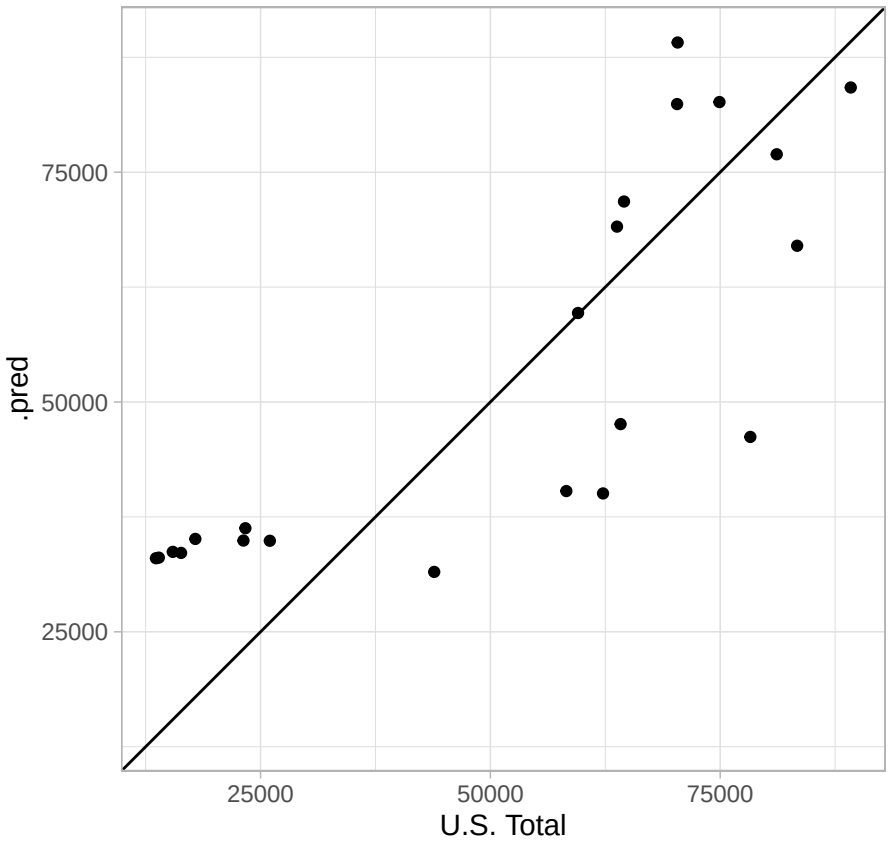
Now, it is possible to fit the model and make the predictions right away.

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

model_fit <-
  fit(lm_model, `U.S. Total` ~ `Ohio (OH)`, data = train_production)


prediction <- predict(model_fit, new_data = test_production)
```

We can apply some metrics to judge model effectiveness.

```
#Let's bind the real values and the predictions make on the test set.

prediction <- bind_cols(test_production, prediction) %>%
  ungroup() %>%
  select(-year)


prediction %>%
  ggplot(aes(`U.S. Total`, .pred)) +
```

```
  geom_abline() +
  geom_point() +
  coord_obs_pred()
```




```
pred_metrics <- metric_set(rmse, mae)

prediction %>%
  ungroup() %>%
  pred_metrics(truth = `U.S. Total`, estimate = .pred)
```

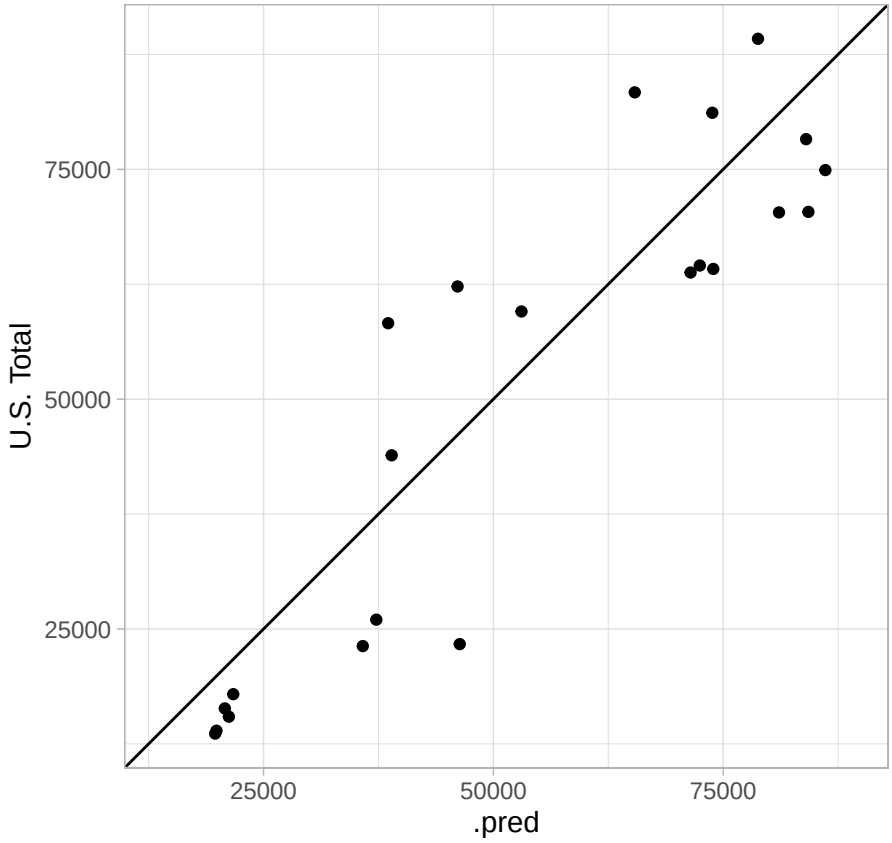
```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      15482.
## 2 mae     standard      13764.
```

This model produced a mean absolute error of 13k. Let's try adding more regions to the prediction.

Let's split the data again and fit the model.

```
model_fit <-
  fit(lm_model,
      `U.S. Total` ~ `Ohio (OH)` + `Minnesota (MN)` + `Wisconsin (WI)` +
        `Michigan (MI)` + `MI State Total`,
      data = train_production)


prediction <- predict(model_fit, new_data = test_production)
```

```r
#Let's bind the real values and the predictions make on the test set.

prediction <- bind_cols(test_production, prediction)

prediction %>%
  ggplot(aes(.pred, `U.S. Total`)) +
  geom_abline() +
  geom_point() +
  coord_obs_pred()
```




```r
prediction %>%
  ungroup() %>%
  pred_metrics(truth = `U.S. Total`, estimate = .pred)
```

```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      11365.
## 2 mae     standard      10134.
```

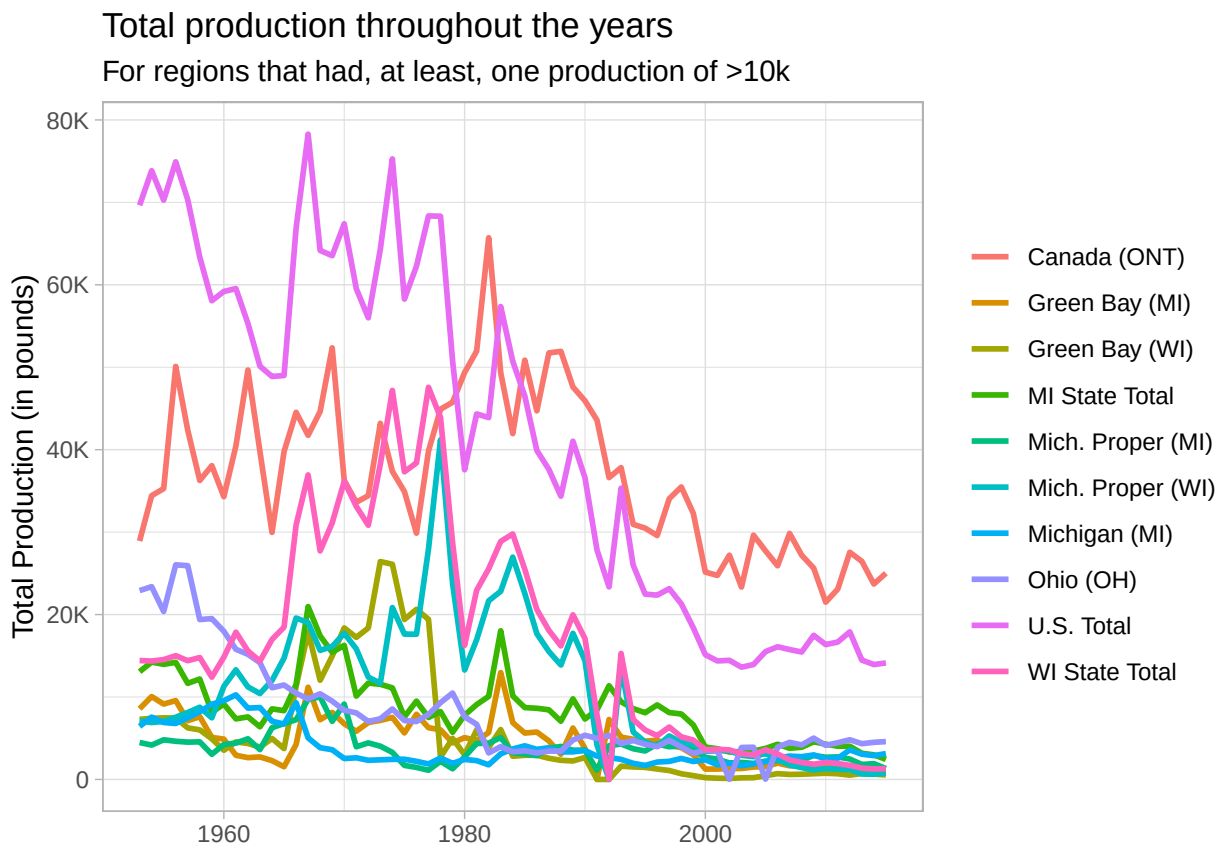
Some regions fit the requirement of having least one production of >10k, but present data starting only at 1953. Let's try using them on our model.

```r
region_production <- fishing %>%
  filter(year >= 1953) %>%
  group_by(year, region) %>%
```

```
  summarise(region_production = sum(values, na.rm = T)) %>%
  ungroup() %>%
  group_by(region) %>%
  mutate(region_max_production = max(region_production)) %>%
  filter(region_max_production > 10000) %>%
  select(-region_max_production) %>%
  pivot_wider(names_from = region, values_from = region_production)
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

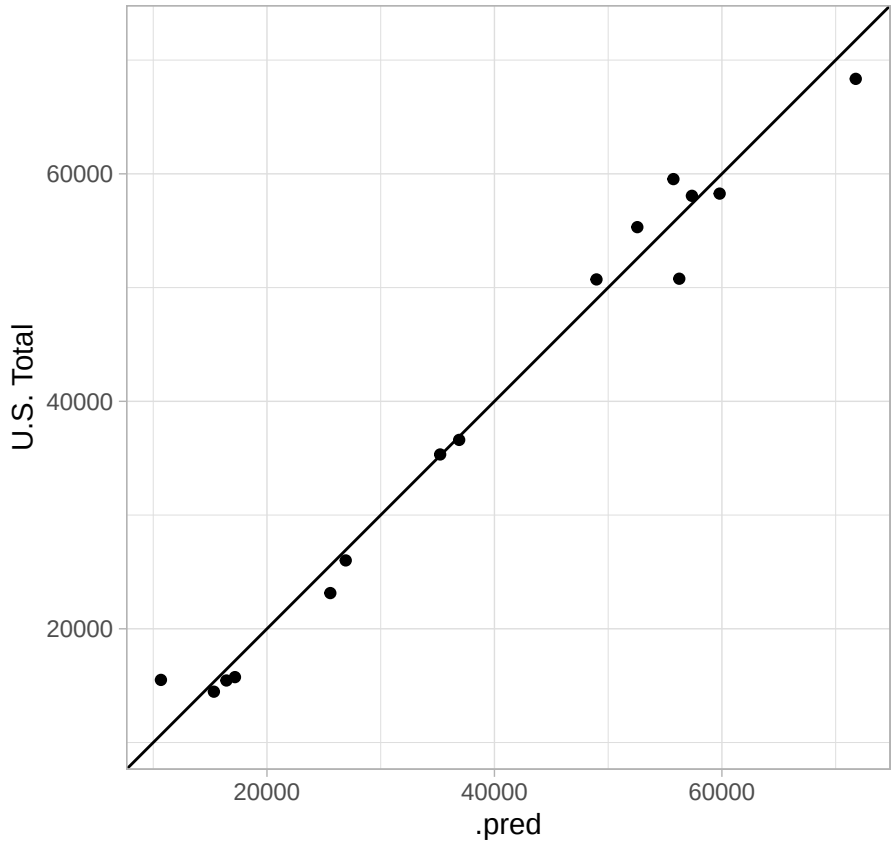
```
region_production %>%
  pivot_longer(cols = !starts_with("year"),
               names_to = "region", values_to = "year_production") %>%
  ggplot(aes(year, year_production)) +
  geom_line(aes(color = region), size = 1) +
  scale_y_continuous(labels = label_number_si()) +
  labs(x = NULL,
       y = "Total Production (in pounds)",
       title = "Total production throughout the years",
       subtitle = "For regions that had, at least, one production of >10k",
       color = NULL)
```




```
data <- initial_split(region_production)

train_production <- training(data)
test_production <- testing(data)
```

```
model_fit <-
  fit(lm_model,
      `U.S. Total` ~ `Ohio (OH)` + `Mich. Proper (MI)` + `Mich. Proper (WI)` +
        `Green Bay (MI)` + `Green Bay (WI)`,
      data = train_production)


prediction <- predict(model_fit, new_data = test_production)
```

```
prediction <- bind_cols(test_production, prediction)

prediction %>%
  ggplot(aes(.pred, `U.S. Total`)) +
  geom_abline() +
  geom_point() +
  coord_obs_pred()
```




```
prediction %>%
  ungroup() %>%
  pred_metrics(truth = `U.S. Total`, estimate = .pred)
```

```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2626.
## 2 mae     standard       2085.
```