TidyTuesday Pumpkin Weight: Workflowsets

Andrew vanderWilden

10/22/2021

```
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(tidymodels))
options(tidymodels.dark = TRUE)
```

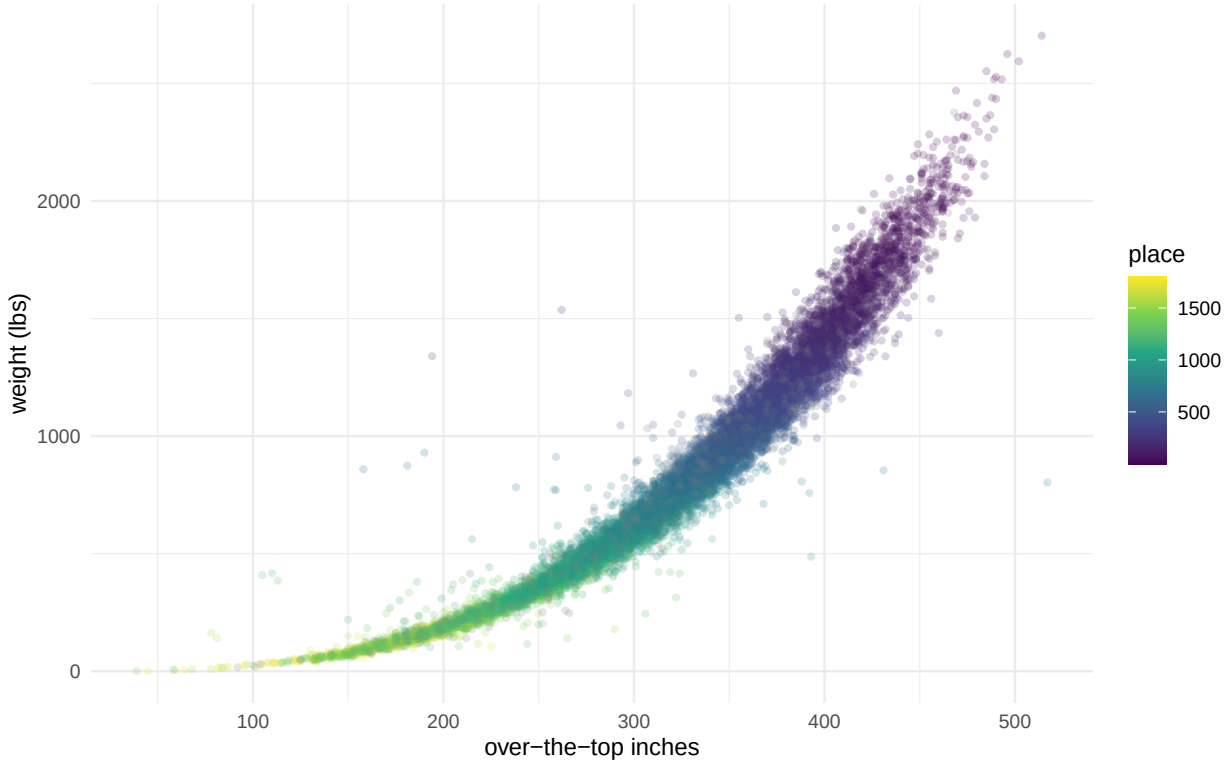
```
# pumpkins_raw <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/

pumpkins_raw <- read_csv('pumpkins.csv')
```

Explore Data

```
pumpkins <-
  pumpkins_raw %>%
  separate(id, into = c('year', 'type')) %>%
  mutate(across(c(year,weight_lbs, ott, place), parse_number)) %>%
  filter(type == 'P') %>%
  select(weight_lbs, year, place, ott, gpc_site, country)
```

OTT is "over-the-top inches" (size of pumpkin).

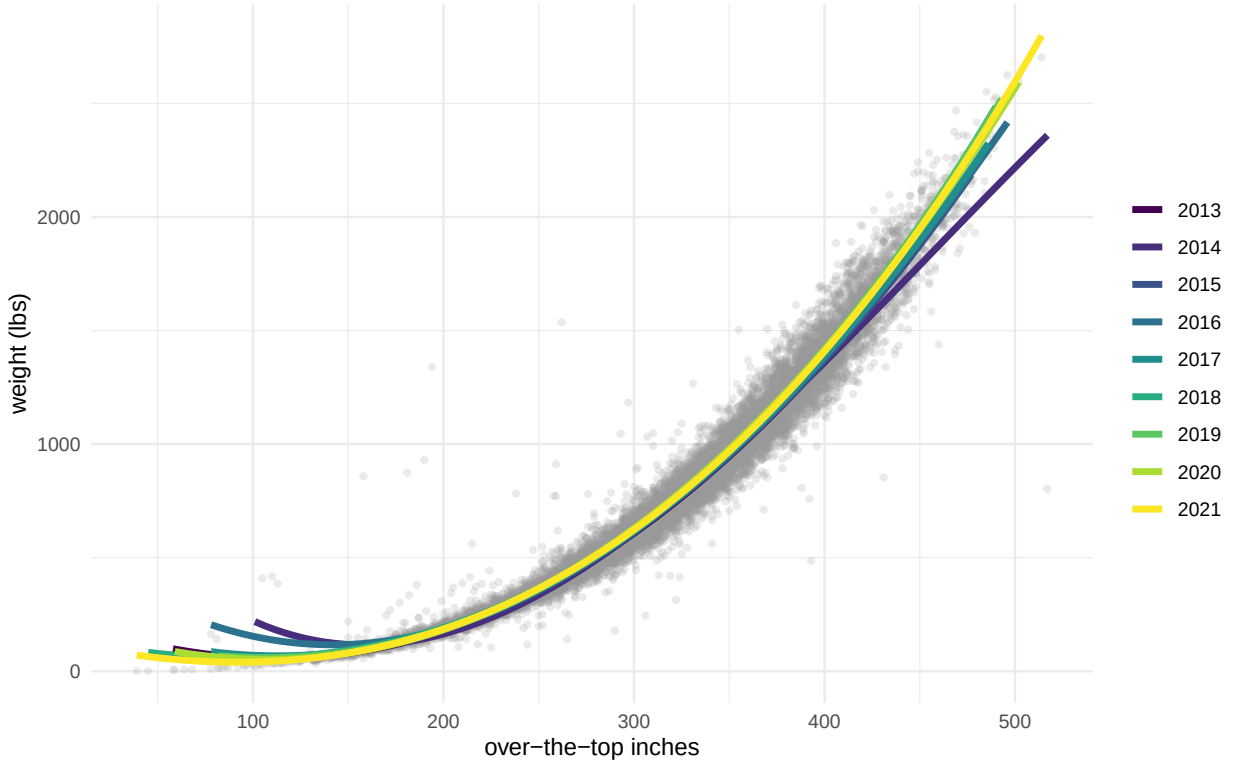
```
pumpkins %>%
  filter(ott > 20, ott < 1e3) %>%
  ggplot(aes(ott, weight_lbs, color = place)) +
  geom_point(alpha = 0.2, size = 1.1) +
  labs(x = 'over-the-top inches', y = 'weight (lbs)') +
  scale_color_viridis_c() +
  theme_minimal()
```



Big Heavy pumpkins tend to win! (As we would expect)

Has there been a change over time?

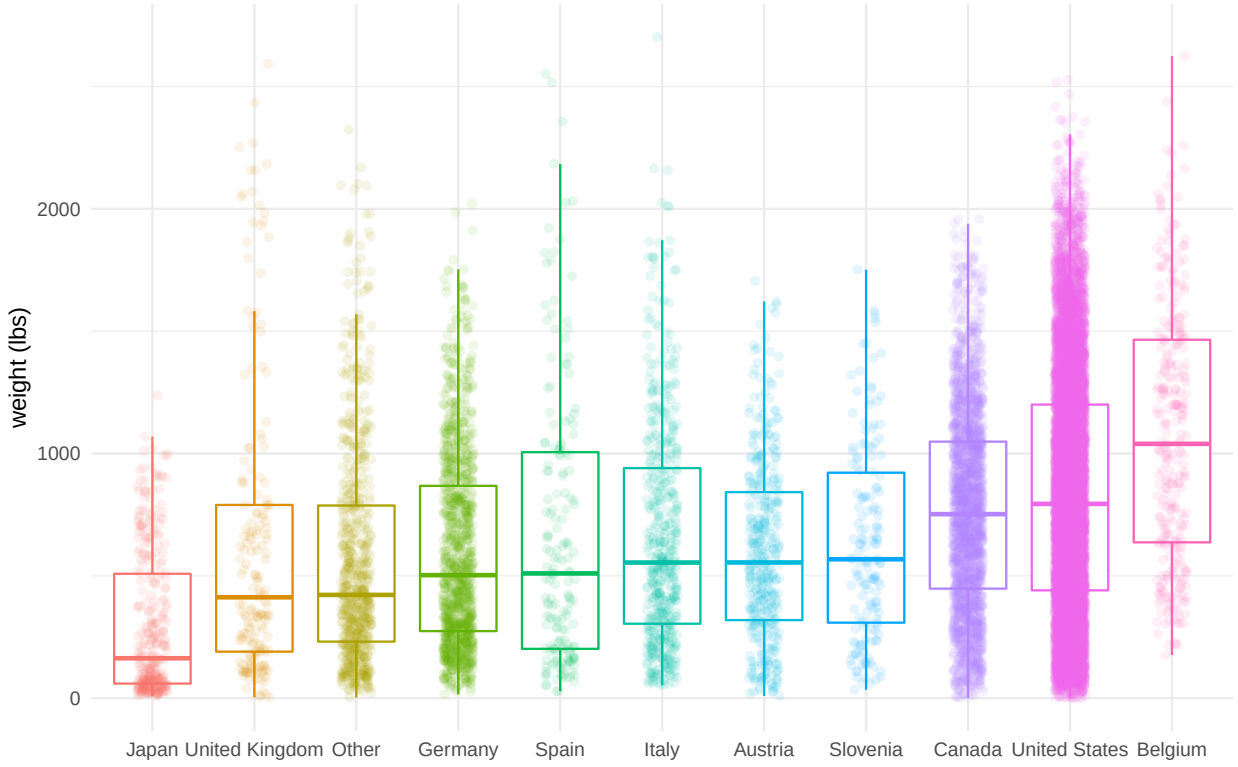
```
pumpkins %>%
  filter(ott > 20, ott < 1e3) %>%
  ggplot(aes(ott, weight_lbs)) +
  geom_point(alpha = 0.2, size = 1.1, color = 'gray 60') +
  geom_smooth(aes(color = factor(year)),
              method = 'lm', formula = y~splines::bs(x,3),
              se = F, size = 1.5, alpha = 0.6) +
  labs(x = 'over-the-top inches', y = 'weight (lbs)', color = NULL) +
  scale_color_viridis_d() +
  theme_minimal()
```



Hard to draw any definitive conclusions.

Which countries produced the largest pumpkins?

```
pumpkins %>%
  mutate(
    country = fct_lump(country, n = 10),
    country = fct_reorder(country, weight_lbs)
  ) %>%
  ggplot(aes(country, weight_lbs, color = country)) +
  geom_boxplot(outlier.colour = NA) +
  geom_jitter(alpha = 0.1, width = 0.15) +
  labs(x = NULL, y = 'weight (lbs)') +
  theme_minimal() +
  theme(legend.position = 'none')
```



Build a workflow set

```
set.seed(9171)

pumpkin_split <-
  pumpkins %>%
  filter(ott > 20, ott < 1e3) %>%
  initial_split(strata = weight_lbs)

pumpkin_train <- training(pumpkin_split)
pumpkin_test <- testing(pumpkin_split)

set.seed(269)

pumpkin_folds <-
  vfold_cv(pumpkin_train, strata = weight_lbs)
```

We will create three recipes, each building on the previous:

1. Pool infrequent factor levels together
2. Create indicator (dummy) variables
3. Create spline terms for OTT

```
base_rec <- recipe(weight_lbs~ott + year + country + gpc_site,
                   data = pumpkin_train) %>%
  step_other(country, gpc_site, threshold = 0.02)

ind_rec <- base_rec %>%
  step_dummy(all_nominal_predictors())

spline_rec <- ind_rec %>%
  step_bs(ott)
```

We will create three types of models:

1. Random Forest
2. MARS
3. Linear Regression

```
rf_spec <- rand_forest(trees = 1e3) %>%
  set_engine('ranger') %>%
  set_mode('regression')

mars_spec <- mars() %>%
  set_engine('earth') %>%
  set_mode('regression')

lm_spec <- linear_reg()
```

We put then put them together in a workflow set:

```
pumpkin_set <-
  workflow_set(
    list(base_rec, ind_rec,spline_rec),
    list(rf_spec,mars_spec,lm_spec),
    cross = FALSE
  )

pumpkin_set
```

```
## # A workflow set/tibble: 3 x 4
##   wflow_id             info             option    result
##   <chr>                <list>           <list>    <list>
## 1 recipe_1_rand_forest <tibble [1 x 4]> <opts[0]> <list [0]>
## 2 recipe_2_mars        <tibble [1 x 4]> <opts[0]> <list [0]>
## 3 recipe_3_linear_reg  <tibble [1 x 4]> <opts[0]> <list [0]>
```

We use `cross = FALSE` because we only want the 3 models, if we said `cross = TRUE` it would fit 9 models, each recipe to each model.

We then fit the models:

```
doParallel::registerDoParallel()
set.seed(4736)
```

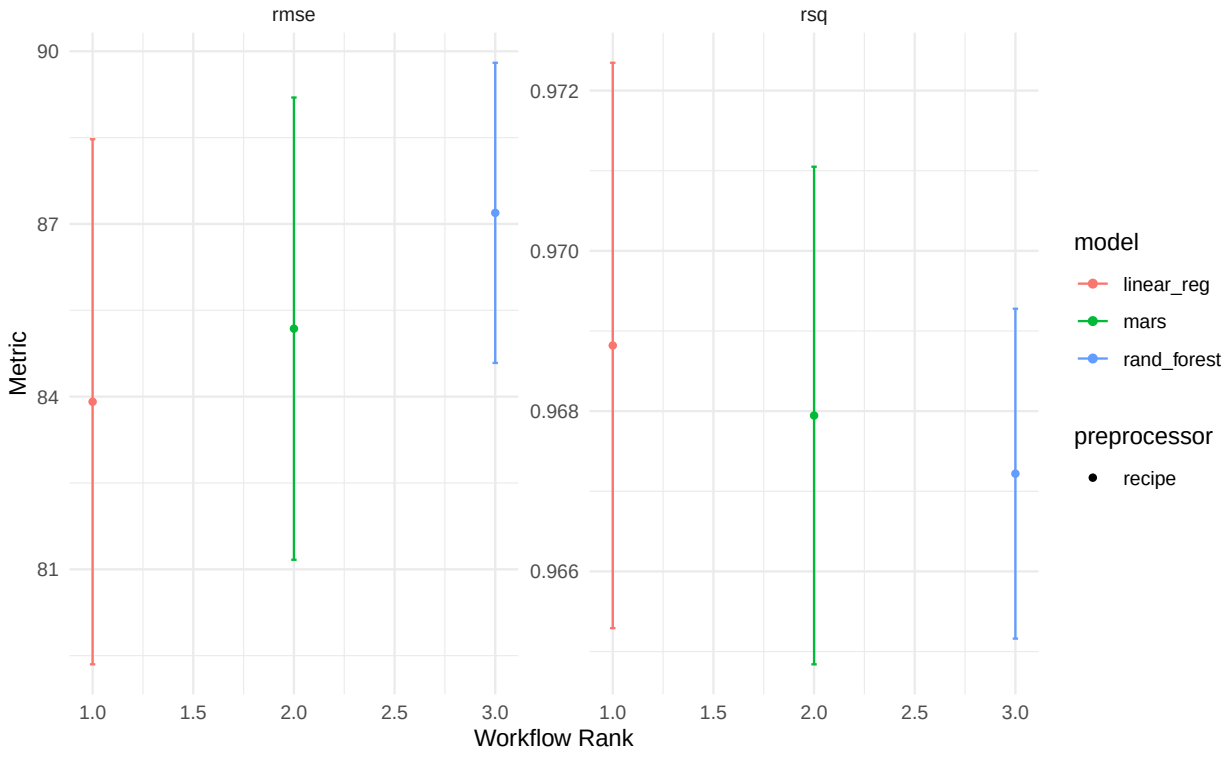
```
pumpkin_rs <-
  workflow_map(
    pumpkin_set,
    'fit_resamples',
    resamples = pumpkin_folds
  )

pumpkin_rs
```

```
## # A workflow set/tibble: 3 x 4
##   wflow_id             info             option    result
##   <chr>                <list>           <list>    <list>
## 1 recipe_1_rand_forest <tibble [1 x 4]> <opts[1]> <rsmp[+]>
## 2 recipe_2_mars        <tibble [1 x 4]> <opts[1]> <rsmp[+]>
## 3 recipe_3_linear_reg  <tibble [1 x 4]> <opts[1]> <rsmp[+]>
```

Evaluate the models

How did they do?

```
autoplot(pumpkin_rs) +
  theme_minimal()
```



Not a ton of difference but the linear model with splines appears to perform best. This is also good news because it is the simplest of the models.

We can then extract the model and fit we want to use:

```
final_fit <-
  extract_workflow(pumpkin_rs, 'recipe_3_linear_reg') %>%
  fit(pumpkin_train)
```

```
tidy(final_fit) %>%
  arrange(-abs(estimate))
```

```
## # A tibble: 16 x 5
##    term                                  estimate std.error statistic   p.value
##    <chr>                                    <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                          -9.04e+3   687.      -13.2   3.20e- 39
##  2 ott_bs_3                              2.54e+3    28.0      90.9   0
##  3 ott_bs_2                              4.15e+2    12.8      32.4   2.64e-219
##  4 ott_bs_1                             -4.06e+2    39.8     -10.2   2.57e- 24
##  5 gpc_site_Ohio.Valley.Giant.Pumpkin.Gr~  2.31e+1     7.85      2.94  3.30e-  3
##  6 country_Germany                      -1.98e+1     6.93     -2.86  4.25e-  3
##  7 country_other                        -1.65e+1     6.59     -2.50  1.25e-  2
##  8 gpc_site_Stillwater.Harvestfest       1.36e+1     7.98      1.71  8.76e-  2
##  9 gpc_site_Elk.Grove.Giant.Pumpkin.Fest~ -1.21e+1     7.61     -1.59  1.11e-  1
## 10 country_United.States                 5.36e+0     5.91      0.907 3.65e-  1
## 11 year                                  4.57e+0     0.340    13.4   8.48e- 41
## 12 country_Canada                        4.01e+0     6.35      0.632 5.28e-  1
## 13 gpc_site_PGPGA.Great.Pumpkin.Weigh.off -3.02e+0     8.03     -0.376 7.07e-  1
## 14 country_Italy                         2.38e+0     7.33      0.325 7.45e-  1
## 15 gpc_site_Wiegemeisterschaft.Berlin.Br~ -1.26e+0     8.09     -0.155 8.77e-  1
## 16 gpc_site_other                        -8.16e-1     5.55     -0.147 8.83e-  1
```

```
last_fit <-
  extract_workflow(pumpkin_rs, 'recipe_3_linear_reg') %>%
  last_fit(pumpkin_split)

collect_metrics(last_fit)
```

```
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard      79.0   Preprocessor1_Model1
## 2 rsq     standard       0.974 Preprocessor1_Model1
```