Let’s start by reading in the data and check out what the top spacecraft used in orbit have been.

astronauts <- read\_csv("<https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-07-14/astronauts.csv>")

astronauts %>%

count(in\_orbit, sort = TRUE)

## # A tibble: 289 x 2

## in\_orbit n

##

## 1 ISS 174

## 2 Mir 71

## 3 Salyut 6 24

## 4 Salyut 7 24

## 5 STS-42 8

## 6 explosion 7

## 7 STS-103 7

## 8 STS-107 7

## 9 STS-109 7

## 10 STS-110 7

## # … with 279 more rows

How has the duration of missions changed over time?

astronauts %>%

mutate(

year\_of\_mission = 10 \* (year\_of\_mission %/% 10),

year\_of\_mission = factor(year\_of\_mission)

) %>%

ggplot(aes(year\_of\_mission, hours\_mission,

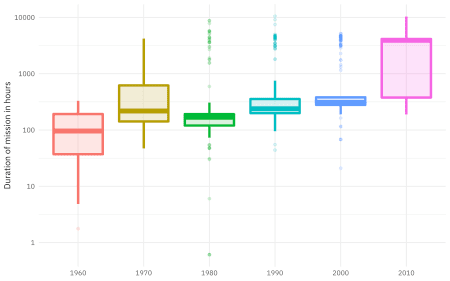
fill = year\_of\_mission, color = year\_of\_mission

)) +

geom\_boxplot(alpha = 0.2, size = 1.5, show.legend = FALSE) +

scale\_y\_log10() +

labs(x = NULL, y = "Duration of mission in hours")



This duration is what we want to build a model to predict, using the other information in this per-astronaut-per-mission dataset. Let’s get ready for modeling next, by bucketing some of the spacecraft together (such as all the space shuttle missions) and taking the logarithm of the mission length.

astronauts\_df <- astronauts %>%

select(

name, mission\_title, hours\_mission,

military\_civilian, occupation, year\_of\_mission, in\_orbit

) %>%

mutate(in\_orbit = case\_when(

str\_detect(in\_orbit, "^Salyut") ~ "Salyut",

str\_detect(in\_orbit, "^STS") ~ "STS",

TRUE ~ in\_orbit

)) %>%

filter(hours\_mission > 0) %>%

mutate(hours\_mission = log(hours\_mission)) %>%

na.omit()

It may make more sense to perform transformations like taking the logarithm of the outcome during data cleaning, *before* feature engineering and using any tidymodels packages like  
[recipes](https://recipes.tidymodels.org/). This kind of transformation is deterministic and can cause problems for tuning and resampling.

**Build a model**

We can start by loading the tidymodels metapackage, and splitting our data into training and testing sets.

library(tidymodels)

set.seed(123)

astro\_split <- initial\_split(astronauts\_df, strata = hours\_mission)

astro\_train <- training(astro\_split)

astro\_test <- testing(astro\_split)

Next, let’s **preprocess** our data to get it ready for modeling.

astro\_recipe <- recipe(hours\_mission ~ ., data = astro\_train) %>%

update\_role(name, mission\_title, new\_role = "id") %>%

step\_other(occupation, in\_orbit,

threshold = 0.005, other = "Other"

) %>%

step\_dummy(all\_nominal(), -has\_role("id"))

Let’s walk through the steps in this recipe.

* First, we must tell the recipe() what our model is going to be (using a formula here) and what data we are using.
* Next, update the role for the two columns that are not predictors or outcome. This way, we can keep them in the data for identification later.
* There are a lot of different occupations and spacecraft in this dataset, so let’s collapse some of the less frequently occurring levels into an “Other” category, for each predictor.
* Finally, we can create indicator variables.

We’re going to use this recipe in a workflow() so we don’t need to stress about whether to prep() or not.

astro\_wf <- workflow() %>%

add\_recipe(astro\_recipe)

astro\_wf

## ══ Workflow ════════════════════════════════════════════════════════════════════════════════════

## Preprocessor: Recipe

## Model: None

##

## ── Preprocessor ────────────────────────────────────────────────────────────────────────────────

## 2 Recipe Steps

##

## ● step\_other()

## ● step\_dummy()

For this analysis, we are going to build a  
[bagging](https://link.springer.com/content/pdf/10.1007/BF00058655.pdf), i.e. bootstrap aggregating, model. This is an ensembling and model averaging method that:

* improves accuracy and stability
* reduces overfitting and variance

In tidymodels, you can create bagging ensemble models with baguette a  
[parsnip](https://parsnip.tidymodels.org/)-adjacent package. The baguette functions create new bootstrap training sets by sampling with replacement and then fit a model to each new training set. These models are combined by averaging the predictions for the regression case, like what we have here (by voting, for classification).

Let’s make two bagged models, one with decision trees and one with  
[MARS](https://en.wikipedia.org/wiki/Multivariate_adaptive_regression_spline) models.

library(baguette)

tree\_spec <- bag\_tree() %>%

set\_engine("rpart", times = 25) %>%

set\_mode("regression")

tree\_spec

## Bagged Decision Tree Model Specification (regression)

##

## Main Arguments:

## cost\_complexity = 0

## min\_n = 2

##

## Engine-Specific Arguments:

## times = 25

##

## Computational engine: rpart

mars\_spec <- bag\_mars() %>%

set\_engine("earth", times = 25) %>%

set\_mode("regression")

mars\_spec

## Bagged MARS Model Specification (regression)

##

## Engine-Specific Arguments:

## times = 25

##

## Computational engine: earth

Let’s fit these models to the training data.

tree\_rs <- astro\_wf %>%

add\_model(tree\_spec) %>%

fit(astro\_train)

tree\_rs

## ══ Workflow [trained] ══════════════════════════════════════════════════════════════════════════

## Preprocessor: Recipe

## Model: bag\_tree()

##

## ── Preprocessor ────────────────────────────────────────────────────────────────────────────────

## 2 Recipe Steps

##

## ● step\_other()

## ● step\_dummy()

##

## ── Model ───────────────────────────────────────────────────────────────────────────────────────

## Bagged CART (regression with 25 members)

##

## Variable importance scores include:

##

## # A tibble: 11 x 4

## term value std.error used

##

## 1 year\_of\_mission 890. 18.5 25

## 2 in\_orbit\_Other 689. 55.6 25

## 3 in\_orbit\_STS 386. 19.4 25

## 4 occupation\_flight.engineer 190. 14.9 25

## 5 occupation\_pilot 189. 20.4 25

## 6 in\_orbit\_Mir 124. 20.7 25

## 7 in\_orbit\_Salyut 100. 9.61 25

## 8 occupation\_MSP 96.3 9.89 25

## 9 occupation\_Other 54.7 4.09 25

## 10 military\_civilian\_military 39.8 4.77 25

## 11 occupation\_PSP 34.4 6.24 25

mars\_rs <- astro\_wf %>%

add\_model(mars\_spec) %>%

fit(astro\_train)

mars\_rs

## ══ Workflow [trained] ══════════════════════════════════════════════════════════════════════════

## Preprocessor: Recipe

## Model: bag\_mars()

##

## ── Preprocessor ────────────────────────────────────────────────────────────────────────────────

## 2 Recipe Steps

##

## ● step\_other()

## ● step\_dummy()

##

## ── Model ───────────────────────────────────────────────────────────────────────────────────────

## Bagged MARS (regression with 25 members)

##

## Variable importance scores include:

##

## # A tibble: 10 x 4

## term value std.error used

##

## 1 in\_orbit\_STS 100 0 25

## 2 in\_orbit\_Other 91.7 1.78 25

## 3 year\_of\_mission 62.6 4.46 25

## 4 in\_orbit\_Salyut 31.7 2.41 25

## 5 in\_orbit\_Mir 1.08 0.914 4

## 6 military\_civilian\_military 0.699 1.43 2

## 7 occupation\_Other 0.698 0.186 3

## 8 occupation\_PSP 0.542 0.924 2

## 9 occupation\_pilot 0.436 0.710 2

## 10 occupation\_flight.engineer 0.215 0 1

The models return aggregated variable importance scores, and we can see that the spacecraft and year are importance in both models.

**Evaluate model**

Let’s evaluate how well these two models did by evaluating performance on the test data.

test\_rs <- astro\_test %>%

bind\_cols(predict(tree\_rs, astro\_test)) %>%

rename(.pred\_tree = .pred) %>%

bind\_cols(predict(mars\_rs, astro\_test)) %>%

rename(.pred\_mars = .pred)

test\_rs

## # A tibble: 316 x 9

## name mission\_title hours\_mission military\_civili… occupation year\_of\_mission

##

## 1 Carp… Mercury-Atla… 1.61 military Pilot 1962

## 2 Schi… Mercury-Atla… 2.22 military pilot 1962

## 3 Tere… Vostok 6 4.26 military pilot 1963

## 4 Koma… Voskhod 1 3.19 military commander 1964

## 5 Feok… Voskhod 1 3.19 civilian MSP 1964

## 6 Youn… Gemini 10 4.26 military pilot 1966

## 7 Youn… Apollo 16 5.58 military commander 1972

## 8 Youn… STS-9 5.48 military commander 1983

## 9 McDi… Gemini 4 4.57 military commander 1965

## 10 Whit… Gemini 4 4.58 military pilot 1965

## # … with 306 more rows, and 3 more variables: in\_orbit , .pred\_tree ,

## # .pred\_mars

We can use the metrics() function from yardstick for both sets of predictions.

test\_rs %>%

metrics(hours\_mission, .pred\_tree)

## # A tibble: 3 x 3

## .metric .estimator .estimate

##

## 1 rmse standard 0.640

## 2 rsq standard 0.798

## 3 mae standard 0.357

test\_rs %>%

metrics(hours\_mission, .pred\_mars)

## # A tibble: 3 x 3

## .metric .estimator .estimate

##

## 1 rmse standard 0.640

## 2 rsq standard 0.795

## 3 mae standard 0.351

Both models performed pretty similarly.

Let’s make some “new” astronauts to understand the kinds of predictions our bagged tree model is making.

new\_astronauts <- crossing(

in\_orbit = fct\_inorder(c("ISS", "STS", "Mir", "Other")),

military\_civilian = "civilian",

occupation = "Other",

year\_of\_mission = seq(1960, 2020, by = 10),

name = "id", mission\_title = "id"

) %>%

filter(

!(in\_orbit == "ISS" & year\_of\_mission < 2000),

!(in\_orbit == "Mir" & year\_of\_mission < 1990),

!(in\_orbit == "STS" & year\_of\_mission > 2010),

!(in\_orbit == "STS" & year\_of\_mission < 1980)

)

new\_astronauts

## # A tibble: 18 x 6

## in\_orbit military\_civilian occupation year\_of\_mission name mission\_title

##

## 1 ISS civilian Other 2000 id id

## 2 ISS civilian Other 2010 id id

## 3 ISS civilian Other 2020 id id

## 4 STS civilian Other 1980 id id

## 5 STS civilian Other 1990 id id

## 6 STS civilian Other 2000 id id

## 7 STS civilian Other 2010 id id

## 8 Mir civilian Other 1990 id id

## 9 Mir civilian Other 2000 id id

## 10 Mir civilian Other 2010 id id

## 11 Mir civilian Other 2020 id id

## 12 Other civilian Other 1960 id id

## 13 Other civilian Other 1970 id id

## 14 Other civilian Other 1980 id id

## 15 Other civilian Other 1990 id id

## 16 Other civilian Other 2000 id id

## 17 Other civilian Other 2010 id id

## 18 Other civilian Other 2020 id id

Let’s start with the decision tree model.

new\_astronauts %>%

bind\_cols(predict(tree\_rs, new\_astronauts)) %>%

ggplot(aes(year\_of\_mission, .pred, color = in\_orbit)) +

geom\_line(size = 1.5, alpha = 0.7) +

geom\_point(size = 2) +

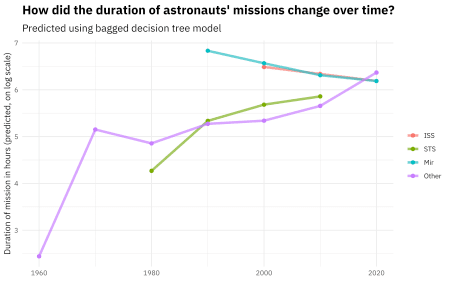
labs(

x = NULL, y = "Duration of mission in hours (predicted, on log scale)",

color = NULL, title = "How did the duration of astronauts' missions change over time?",

subtitle = "Predicted using bagged decision tree model"

)



What about the MARS model?

new\_astronauts %>%

bind\_cols(predict(mars\_rs, new\_astronauts)) %>%

ggplot(aes(year\_of\_mission, .pred, color = in\_orbit)) +

geom\_line(size = 1.5, alpha = 0.7) +

geom\_point(size = 2) +

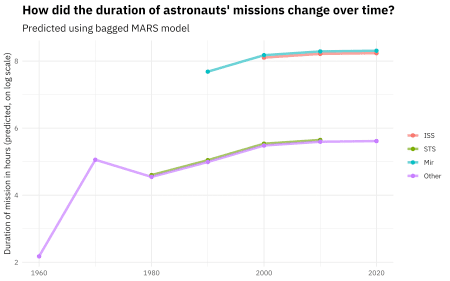
labs(

x = NULL, y = "Duration of mission in hours (predicted, on log scale)",

color = NULL, title = "How did the duration of astronauts' missions change over time?",

subtitle = "Predicted using bagged MARS model"

)



You can really get a sense of how these two kinds of models work from the differences in these plots (tree vs. splines with knots), but from both, we can see that missions to space stations are longer, and missions in that “Other” category change characteristics over time pretty dramatically.