11 Comparing models with resampling

Once we create two or more models, the next step is to compare them. In some cases, comparisons might be *within-model*, where the same model might be evaluated with different features or preprocessing methods. Alternatively, *between-model* comparisons, such as when we compared linear regression and random forest models in Chapter 10, are the more common scenario.

In either case, the result is a collection of resampled summary statistics (e.g. RMSE, accuracy, etc.) for each model. The first section of this chapter demonstrates how workflow sets can be used to fit multiple models. The second section discusses important aspects of resampling statistics. Two additional sections follow describing how to formally compare models.

11.1 CREATING MULTIPLE MODELS

Previously, Section 7.5 describes the idea of a workflow set where different preprocessors and/or models can be combinatorially generated. In the last chapter, we used a recipe for the Ames data that included an interaction term as well as spline functions for longitude and latitude. To demonstrate more with workflow sets, let’s create different linear models that add these preprocessing steps incrementally. We’ll create three recipes then combine them into a workflow set:

```r
library(tidymodels)
tidymodels_prefer()

basic_rec <-
  recipe(Sale_Price ~ Neighborhood + Gr_Liv_Area + Year_Built + Bldg_Type +
           Latitude + Longitude, data = ames_train) %>%
  step_log(Gr_Liv_Area, base = 10) %>%
  step_other(Neighborhood, threshold = 0.01) %>%
  step_dummy(all_nominal_predictors())

interaction_rec <-
  basic_rec %>%
  step_interact( ~ Gr_Liv_Area:starts_with("Bldg_Type_") )

spline_rec <-
  interaction_rec %>%
  step_ns(Latitude, Longitude, deg_free = 50)

preproc <-
  list(basic = basic_rec,
       interact = interaction_rec,
       splines = spline_rec
  )

lm_models <- workflow_set(preproc, list(lm = lm_model), cross = FALSE)
lm_models
#> # A workflow set/tibble: 3 × 4
#>   wflow_id    info             option    result
#>   <chr>       <list>           <list>    <list>
#> 1 basic_lm    <tibble [1 × 4]> <opts[0]> <list [0]>
#> 2 interact_lm <tibble [1 × 4]> <opts[0]> <list [0]>
#> 3 splines_lm  <tibble [1 × 4]> <opts[0]> <list [0]>
```

We'd like to resample each of these models in turn. To do so, a **purrr**-like function called `workflow_map()` is used. This function takes an initial argument of the function to apply to the workflows, followed by an options to that function. We also set a `verbose` argument that will print the progress as well as a `seed` argument that makes sure that each model uses the same random number stream as the others.

```
lm_models <- 
  lm_models %>% 
  workflow_map("fit_resamples", 
               # Options to `workflow_map()`: 
               seed = 1101, verbose = TRUE,
               # Options to `fit_resamples()`: 
               resamples = ames_folds, control = keep_pred)
#> i 1 of 3 resampling: basic_lm
#> ✓ 1 of 3 resampling: basic_lm (2.9s)
#> i 2 of 3 resampling: interact_lm
#> ✓ 2 of 3 resampling: interact_lm (3.5s)
#> i 3 of 3 resampling: splines_lm
#> ✓ 3 of 3 resampling: splines_lm (3.7s)
lm_models
#> # A workflow set/tibble: 3 × 4
#>   wflow_id    info             option    result   
#>   <chr>       <list>           <list>    <list>   
#> 1 basic_lm    <tibble [1 × 4]> <opts[2]> <rsmp[+]>
#> 2 interact_lm <tibble [1 × 4]> <opts[2]> <rsmp[+]>
#> 3 splines_lm  <tibble [1 × 4]> <opts[2]> <rsmp[+]>
```

Notice that the `option` and `result` columns are now populated. The former includes the options to `fit_resamples()` that were given (for reproducibility) and the latter column contains the results produced by `fit_resamples()` .

There are few convenience functions for workflow sets, including one to collate the performance statistics:

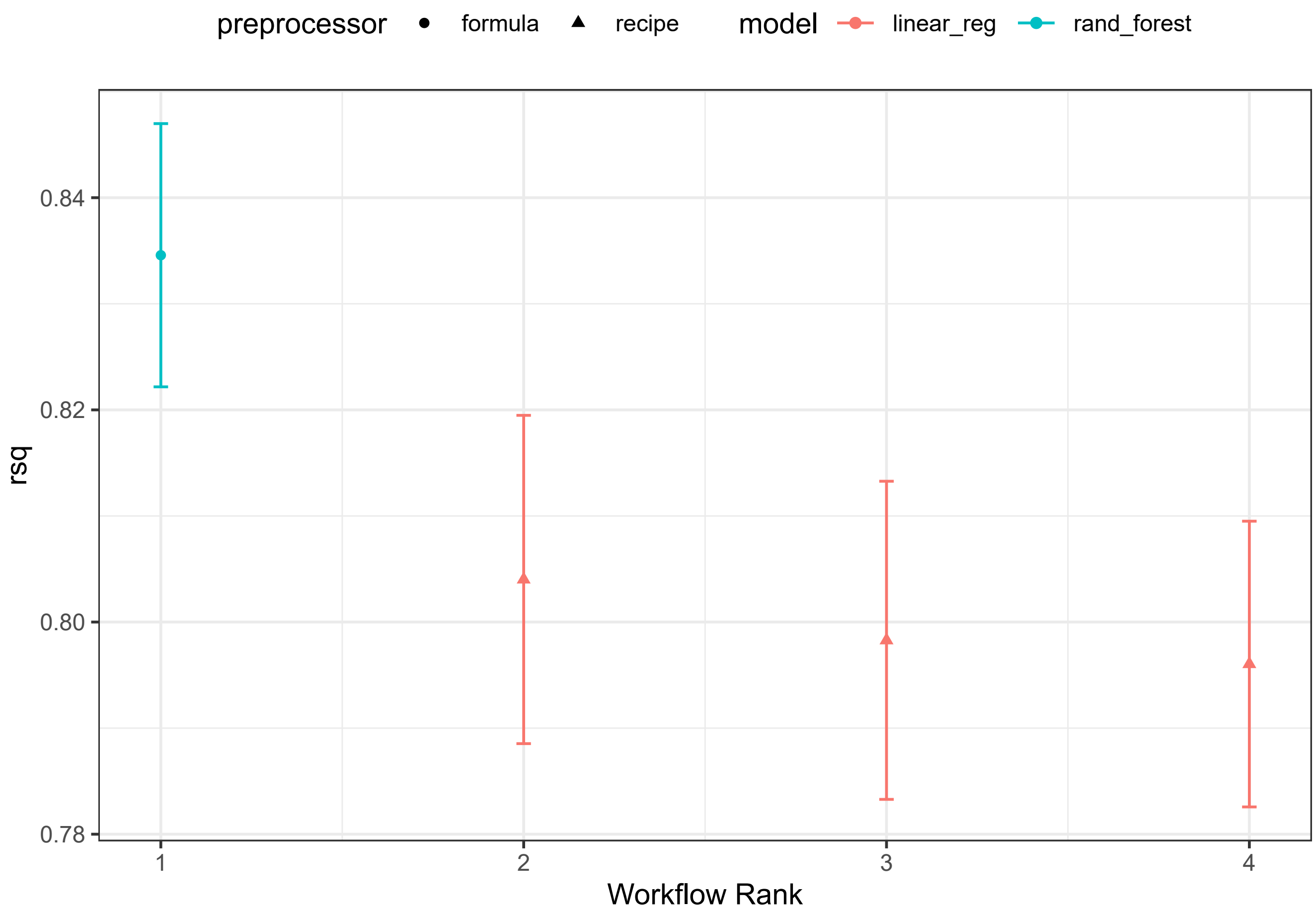
```
collect_metrics(lm_models) %>%
  filter(.metric == "rmse")
#> # A tibble: 3 × 9
#>   wflow_id     .config         preproc model    .metric .estimator   mean     n std_err
#>   <chr>        <chr>           <chr>   <chr>    <chr>   <chr>       <dbl> <int>   <dbl>
#> 1 basic_lm     Preprocessor1… recipe  linear… rmse    standard   0.0799    10 0.00237
#> 2 interact_lm  Preprocessor1… recipe  linear… rmse    standard   0.0794    10 0.00257
#> 3 splines_lm   Preprocessor1… recipe  linear… rmse    standard   0.0783    10 0.00257
```

What about the random forest model from the previous chapter? We can add it to the set by first converting it to its own workflow set then binding rows. This requires that, when the model was resampled, the `save_workflow = TRUE` option was set in the control function.

```
four_models <-
  as_workflow_set(random_forest = rf_res) %>%
  bind_rows(lm_models)
four_models
#> # A workflow set/tibble: 4 × 4
#>   wflow_id      info             option    result
#>   <chr>         <list>           <list>    <list>
#> 1 random_forest <tibble [1 × 4]> <opts[0]> <rsmp[+]>
#> 2 basic_lm      <tibble [1 × 4]> <opts[2]> <rsmp[+]>
#> 3 interact_lm   <tibble [1 × 4]> <opts[2]> <rsmp[+]>
#> 4 splines_lm    <tibble [1 × 4]> <opts[2]> <rsmp[+]>
```

The `autoplot()` method shows confidence intervals for each model in order of best-to-worst. In this chapter, we'll focus on the coefficient of determination (a.k.a. R^2):

```
autoplot(four_models, metric = "rsq")
```



The random forest method is doing the best job and there are minor improvements in the linear models as we add more recipe steps.

Now that we have 10 resampled performance estimates for each of four models, these summary statistics can be used to make between-model comparisons.

11.2 RESAMPLED PERFORMANCE STATISTICS

Considering these results, it appears that the additional terms do not profoundly improve the *mean* RMSE or R^2 statistics for the linear models. The difference is small, but it might be larger than the experimental noise in the system, i.e., considered statistically significant. We can formally **test the hypothesis that the additional terms increase R^2**.

> Before comparing models, it is important to discuss the within-resample correlation for resampling statistics. Each model was measured with the same cross-validation folds, and *results for the same resample tend to be similar.*

In other words, there are some resamples where performance across models tends to be low and others where it tends to be high. In statistics, this is called a resample-to-resample component of variation.

To illustrate, let's gather the individual resampling statistics for the linear models and the random forest. We will focus on the R^2 statistic for each model, which measures correlation between the observed and predicted sale prices for each house.

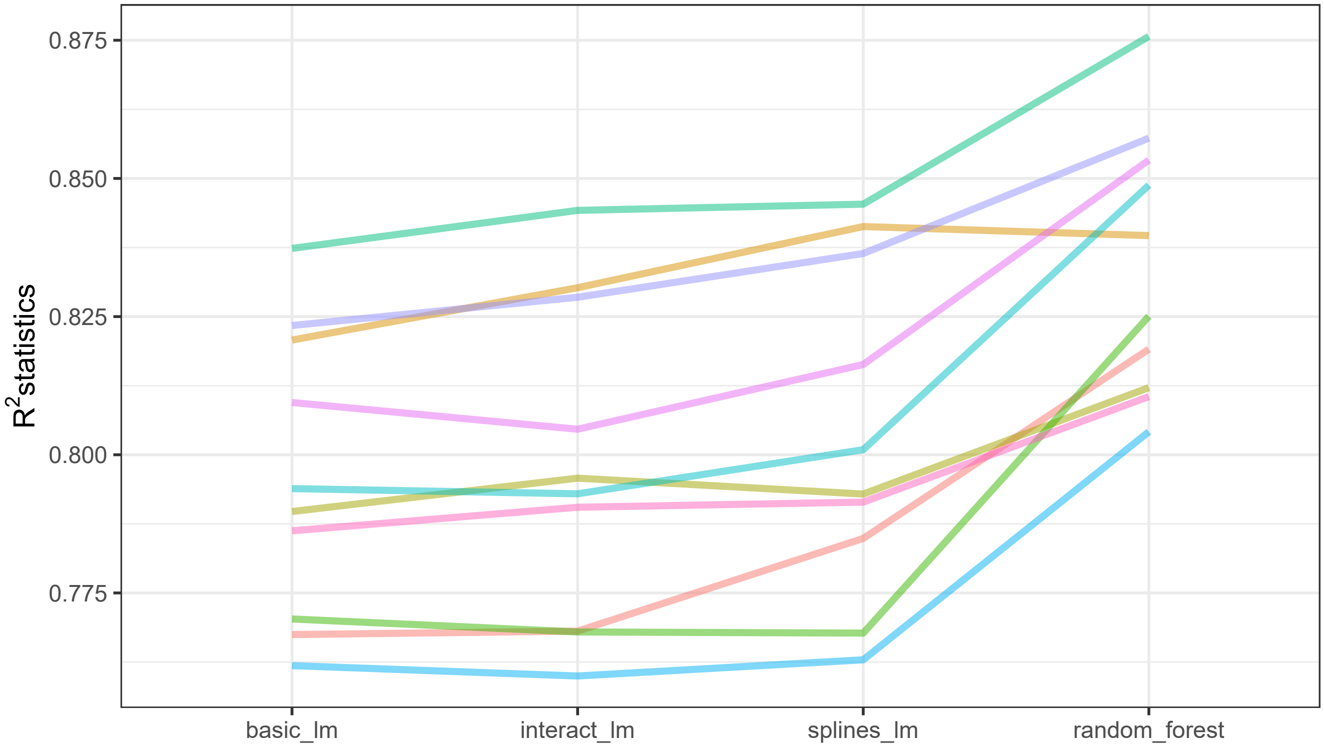
```
rsq_indiv_estimates <-
  collect_metrics(four_models, summarize = FALSE) %>%
  filter(.metric == "rsq")

rsq_wider <-
  rsq_indiv_estimates %>%
  select(wflow_id, .estimate, id) %>%
  pivot_wider(id_cols = "id", names_from = "wflow_id", values_from = ".estimate")

corrr::correlate(rsq_wider %>% select(-id), quiet = TRUE)
#> # A tibble: 4 × 5
#>   term          random_forest basic_lm interact_lm splines_lm
#>   <chr>                 <dbl>    <dbl>       <dbl>      <dbl>
#> 1 random_forest        NA        0.858       0.799      0.823
#> 2 basic_lm              0.858   NA           0.991      0.976
#> 3 interact_lm           0.799    0.991      NA          0.977
#> 4 splines_lm            0.823    0.976       0.977     NA
```

These high correlations indicate that, across models, there are large within-resample correlations. To see this visually, the R^2 statistics are shown for each model with lines connecting the resamples:

```
rsq_indiv_estimates %>%
  mutate(wflow_id = reorder(wflow_id, .estimate)) %>%
  ggplot(aes(x = wflow_id, y = .estimate, group = id, col = id)) +
  geom_line(alpha = .5, lwd = 1.25) +
  theme(legend.position = "none") +
  labs(x = NULL, y = expression(paste(R^2, "statistics")))
```



If the resample-to-resample effect was not real, there would not be any parallel lines. A statistical test for the correlations evaluates whether the magnitudes of these correlations are not simply noise. For the linear models:

```
rsq_wider %>%
  with( cor.test(basic_lm, splines_lm) ) %>%
  tidy() %>%
  select(estimate, starts_with("conf"))
#> # A tibble: 1 × 3
#>   estimate conf.low conf.high
#>      <dbl>    <dbl>     <dbl>
#> 1    0.976    0.899     0.995
```

The within-resample correlation appears to be real.

What effect does the extra correlation have on our analysis? Consider the variance of a difference of two variables:

$$Var[X - Y] = Var[X] + Var[Y] - 2Cov[X, Y]$$

The last term is the covariance between two items. If there is a significant positive covariance, then any statistical test of this difference would be critically under-powered comparing the difference in two models. In other words, ignoring the resample-to-resample effect would bias our model comparisons towards finding no differences between models.

> This characteristic of resampling statistics will come into play in the next two sections.

Before making model comparisons or looking at the resampling results, it can be helpful to define a relevant *practical effect size*. Since these analyses focus on the R^2 statistics, the practical effect size is the change in R^2 that we would consider to be a realistic difference that matters. For example, we might think that two models are not practically different if their R^2 values are within ± 2%. If this were the case, differences smaller than 2% are not deemed important even if they are statistically significant.

Practical significance is subjective; two people can have very different ideas on the threshold for importance. However, as shown later, this consideration can be very helpful when deciding between models.

11.3 SIMPLE HYPOTHESIS TESTING METHODS

Consider the familiar linear statistical model:

$$y_{ij} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_{ij}$$

This versatile model is used to create regression models as well as being the basis for the popular analysis of variance (ANOVA) technique for comparing groups. With the ANOVA model, the predictors (x_{ij}) are binary dummy variables for different groups. From this, the β parameters estimate whether two or more groups are different from one another using hypothesis testing techniques.

In our specific situation, the ANOVA can also make model comparisons. Suppose the individual resampled R^2 statistics serve as the *outcome data* (i.e., the y_{ij}) and the models as the *predictors* in the ANOVA model. A sampling of this data structure would be:

Y = rsq	model	X1	X2	X3	id
0.7675	basic_lm	0	0	0	Fold01
0.7681	interact_lm	1	0	0	Fold01
0.8191	random_forest	0	1	0	Fold01
0.7849	splines_lm	0	0	1	Fold01
0.8208	basic_lm	0	0	0	Fold02
0.8302	interact_lm	1	0	0	Fold02

The `X1` , `X2` , and `X3` columns in the table are indicators for the values in the `model` column. Their order was defined in the same way that R would define them, alphabetically ordered by `model` .

For our model comparison, the specific ANOVA model is:

$$y_{ij} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_{ij}$$

where

- β_0 is the estimate of the mean R^2 statistic for the basic linear models (i.e., without splines or interactions),
- β_1 is the *change* in mean R^2 when interactions are added to the basic linear model,
- β_2 is the *change* in mean R^2 between the basic linear model and the random forest model.

- β_3 is the *change* in mean R^2 between the basic linear model and one with interactions and splines.

From these model parameters, hypothesis tests and p-values are generated to statistically compare models, but we must contend with how to handle the resample-to-resample effect. Historically, the resample groups would be considered a *block effect* and an appropriate term was added to the model. Alternatively, the resample effect could be considered a *random effect* where these particular resamples were drawn at random from a larger population of possible resamples. However, we aren't really interested in these effects; we only want to adjust for them in the model so that the variances of the interesting differences are properly estimated.

Treating the resamples as random effects is theoretically appealing. Methods for fitting an ANOVA model with this type of random effect could include the linear mixed model (Faraway 2016) or a Bayesian hierarchical model (shown in the next section).

A simple and fast method for comparing two models at a time is to use the *differences in R^2 values* as the outcome data in the ANOVA model. Since the outcomes are matched by resample, the differences *do not contain the resample-to-resample effect* and, for this reason, the standard ANOVA model is appropriate. To illustrate, this call to `lm()` tests the difference between two of the linear regression models:

```
compare_lm <- 
  rsq_wider %>% 
  mutate(difference = splines_lm - basic_lm)

lm(difference ~ 1, data = compare_lm) %>% 
  tidy(conf.int = TRUE) %>% 
  select(estimate, p.value, starts_with("conf"))
#> # A tibble: 1 × 4
#>   estimate p.value conf.low conf.high
#>      <dbl>   <dbl>    <dbl>     <dbl>
#> 1  0.00797 0.00660  0.00284    0.0131

# Alternatively, a paired t-test could also be used: 
rsq_wider %>% 
  with( t.test(splines_lm, basic_lm, paired = TRUE) ) %>%
  tidy() %>% 
  select(estimate, p.value, starts_with("conf"))
#> # A tibble: 1 × 4
#>   estimate p.value conf.low conf.high
#>      <dbl>   <dbl>    <dbl>     <dbl>
#> 1  0.00797 0.00660  0.00284    0.0131
```

We could evaluate each pair-wise difference in this way. Note that the p-value indicates a *statistically significant signal*; the collection of spline terms for longitude and latitude do appear to have an effect. However, the difference in R^2 is estimated at 0.8%. If our practical effect size were 2%, we might not consider these terms worth including in the model.

> What's a p-value? From Wasserstein and Lazar (2016): "Informally, a p-value is the probability under a specified statistical model that a statistical summary of the data (e.g., the sample mean difference between two compared groups) would be equal to or more extreme than its observed value."
>
> In other words, if this analysis were repeated a large number of times under the null hypothesis of no differences, the p-value reflects how extreme our *observed* results would be in comparison.

The next section takes a more general approach to making formal comparisons between models using random effects and Bayesian statistics (McElreath 2020). While the model is more complex than the ANOVA method shown above, the interpretation is more simple and straight-forward than the p-value approach.

11.4 BAYESIAN METHODS

The previous ANOVA model for these data had the form:

$$y_{ij} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_{ij}$$

where the residuals ϵ_{ij} are assumed to be independent and follow a Gaussian distribution with zero mean and constant standard deviation of σ. From this assumption, statistical theory shows that the estimated regression parameters follow a multivariate Gaussian distribution and, from this, p-values and confidence intervals are derived.

A Bayesian linear model makes additional assumptions. In addition to specifying a distribution for the residuals, we require *prior distribution* specifications for the model parameters (β_j and σ). These are distributions for the parameters that the model assumes before being exposed to the observed data. For example, a simple set of prior distributions for our model might be:

$$\begin{aligned} \epsilon_{ij} &\sim N(0, \sigma) \\ \beta_j &\sim N(0, 10) \\ \sigma &\sim \text{exponential}(1) \end{aligned}$$

These priors set the possible/probable ranges of the model parameters and have no unknown parameters. For example, the prior on σ indicates that values must be larger than zero, are very right-skewed, and have values that are usually less than 3 or 4.

Note that the regression parameters have a pretty wide prior distribution, with a standard deviation of 10. In many cases, we might not have a strong opinion about the prior beyond it being symmetric and bell shaped. The large standard deviation implies a fairly uninformative prior; it is not overly restrictive in terms of the possible values that the parameters might take on. This allows the data to have more of an influence during parameter estimation.

Given the observed data and the prior distribution specifications, the model parameters can then be estimated. The final distributions of the model parameters are combinations of the priors and the maximum likelihood estimates. These *posterior distributions* of the parameters are the key distributions of interest. They are a full probabilistic description of the model's estimated parameters.

To adapt our Bayesian ANOVA model so that the resamples are adequately modeled, we consider a *random intercept model*. Here, we assume that the resamples impact the model only by changing the intercept. Note that this constrains the resamples from having a differential impact on the regression parameters β_j; these are assumed to have the same relationship across resamples. This model equation is:

$$y_{ij} = (\beta_0 + b_i) + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_{ij}$$

This is not an unreasonable model for resampled statistics which, when plotted across models (as above), tend to have fairly parallel effects across models (i.e., little cross-over of lines).

For this model configuration, an additional assumption is made for the prior distribution of random effects. A reasonable assumption for this distribution is another symmetric distribution, such as another bell-shaped curve. Given the effective sample size of 10 in our summary statistic data, let's use a prior that is wider than a standard normal distribution. We'll use a t-distribution with a single degree of freedom (i.e. $b_i \sim t(1)$), which has heavier tails than an analogous Gaussian distribution.

The **tidyposterior** package has functions to fit such Bayesian models for the purpose of comparing resampled models. The main function is called `perf_mod()` and it is configured to "just work" for different types of objects:

- For workflow sets, it creates an ANOVA model where the groups correspond to the workflows. Our existing models did not optimize any tuning parameters (see the next three chapters). If one of the workflows in the set had data on tuning parameters, the best tuning parameters set for each workflow is used in the Bayesian analysis. In other words, despite the presence of tuning parameters, `perf_mod()` focuses on making *between-workflow comparisons*.
- For objects that contain a single model that has been tuned using resampling, `perf_mod()` makes *within-model comparisons*. In this situation, the grouping variables tested in the Bayesian ANOVA model are the submodels defined by the tuning parameters.

- The `perf_mod()` function can also take a data frame produced by **rsample** that has columns of performance metrics associated with two or more model/workflow results. These could have been generated by non-standard means.

From any of these types of objects, the `perf_mod()` function determines an appropriate Bayesian model and fits it with the resampling statistics. For our example, it will model the four sets of R^2 statistics associated with the workflows.

The **tidyposterior** package uses the Stan software for specifying and fitting the models via the **rstanarm** package. The functions within that package have default priors (see `?priors` for more details). The model below uses the default priors for all parameters except for the random intercepts (which follow a *t*-distribution). The estimation process uses random numbers so the seed is set within the function call. The estimation process is iterative and is replicated several times in collections called *chains*. The `iter` parameter tells the function how long to run the estimation process in each chain. When several chains are used, their results are combined (assume that this is validated by diagnostic assessments).

```r
library(tidyposterior)
library(rstanarm)

# The rstanarm package creates copious amounts of output; those results
# are not shown here but are worth inspecting for potential issues. The
# option `refresh = 0` can be used to eliminate the logging.
rsq_anova <-
  perf_mod(
    four_models,
    metric = "rsq",
    prior_intercept = rstanarm::student_t(df = 1),
    chains = 4,
    iter = 5000,
    seed = 1102
  )
```

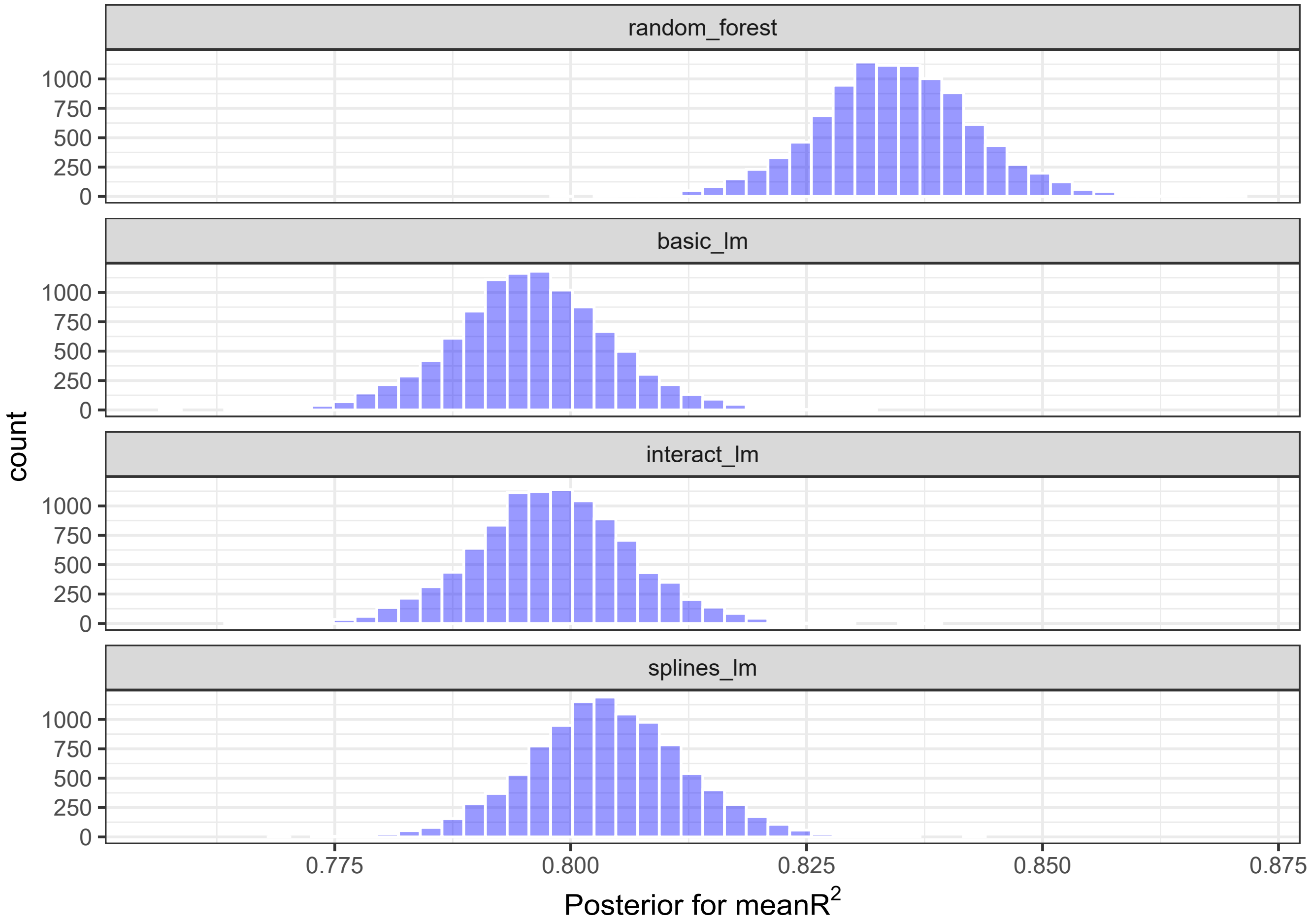
The resulting object has information on the resampling process as well as the Stan object embedded within (in an element called `stan`). We are most interested in the posterior distributions of the regression parameters. The **tidyposterior** package has a `tidy()` method that extracts these posterior distributions into a tibble:

```
model_post <-
  rsq_anova %>%
  # Take a random sample from the posterior distribution
  # so set the seed again to be reproducible.
  tidy(seed = 1103)

glimpse(model_post)
#> Rows: 40,000
#> Columns: 2
#> $ model     <chr> "random_forest", "random_forest", "random_forest", "random_fores…
#> $ posterior <dbl> 0.8290, 0.8367, 0.8372, 0.8387, 0.8333, 0.8218, 0.8271, 0.8235, …
```

The four posterior distributions are:

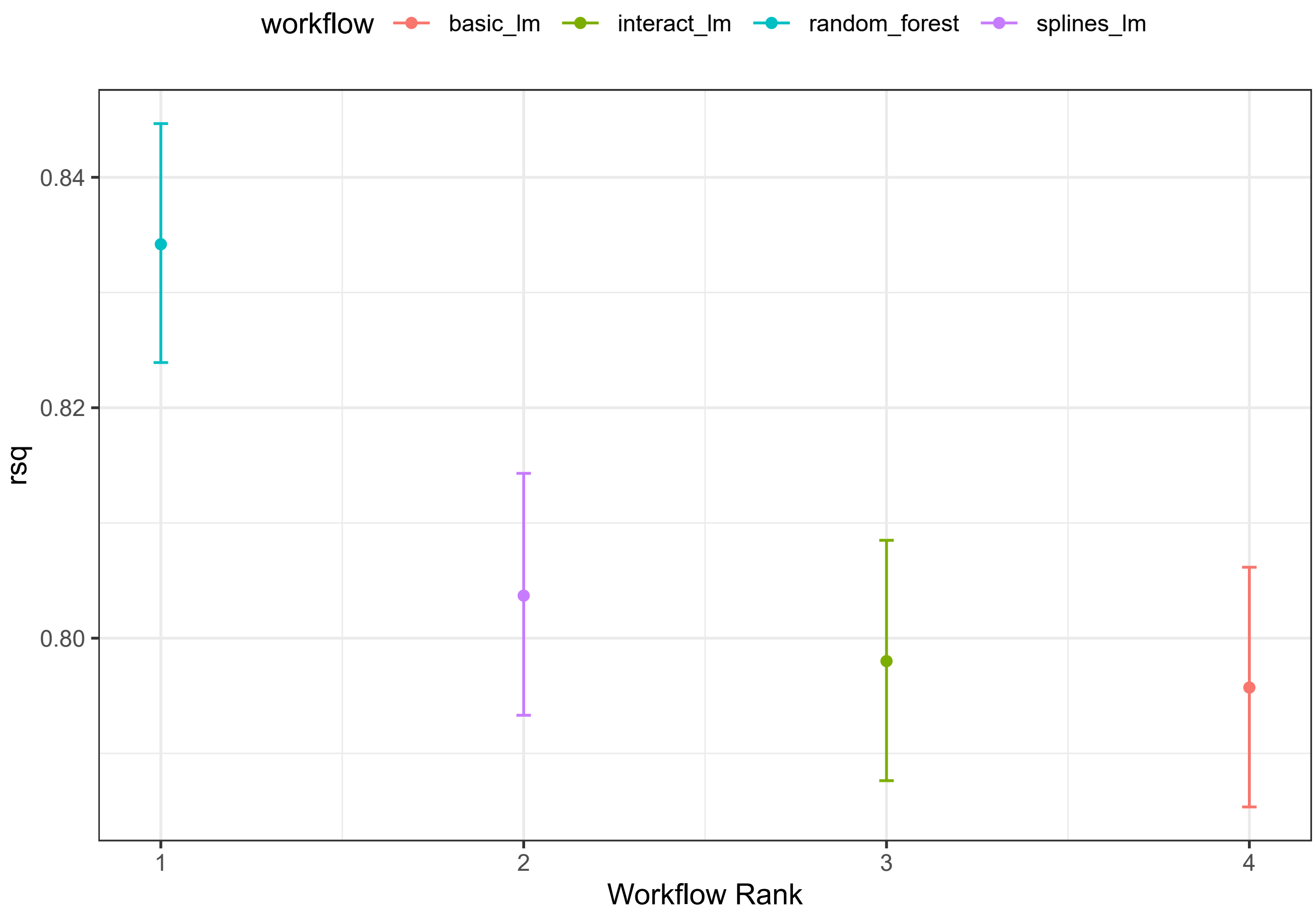
```
model_post %>%
  mutate(model = forcats::fct_inorder(model)) %>%
  ggplot(aes(x = posterior)) +
  geom_histogram(bins = 50, col = "white", fill = "blue", alpha = 0.4) +
  facet_wrap(~ model, ncol = 1) +
  labs(x = expression(paste("Posterior for mean ", R^2)))
```



These histograms describe the estimated probability distributions of the mean R^2 value for each model. There is some overlap, especially for the two linear models.

There is also a basic `autoplot()` method for the model results as well as the tidied object that shows overlaid density plots. For example:

```
autoplot(rsq_anova)
```

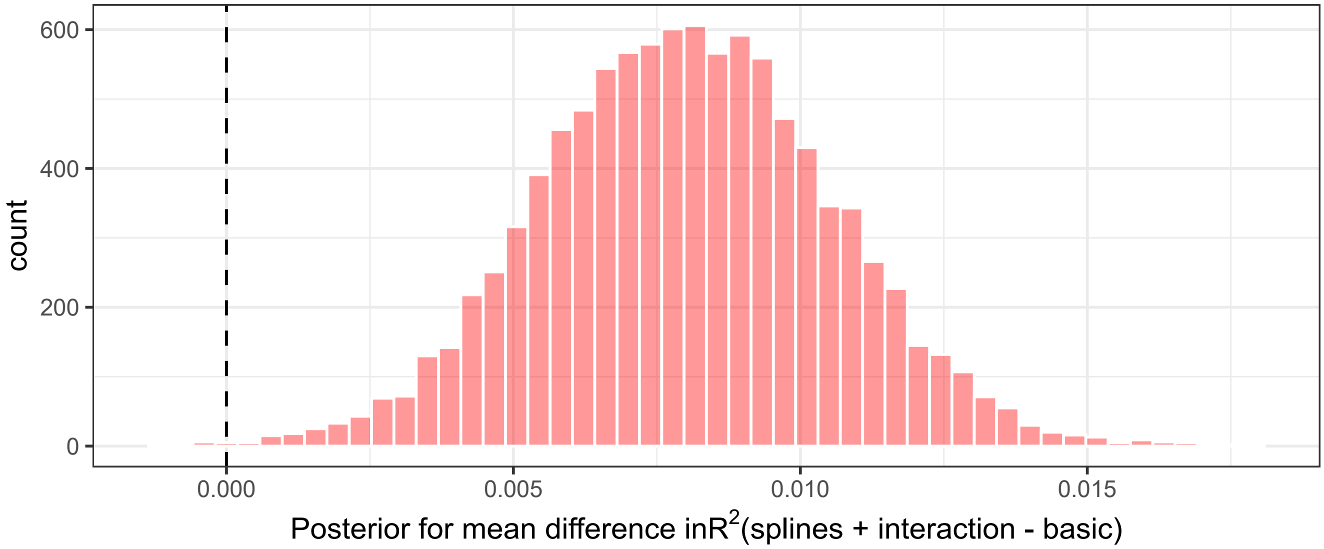


One wonderful aspect of using resampling with Bayesian models is that, once we have the posteriors for the parameters, it is trivial to get the posterior distributions for combinations of the parameters. For example, to compare the two linear regression models, we are interested in the difference in means. The posterior of this difference is computed by sampling from the individual posteriors and taking the differences. The `contrast_models()` function can do this. To specify the comparisons to make, the `list_1` and `list_2` parameters take character vectors and compute the differences between the models in those lists (parameterized as `list_1 - list_2`).

To compare two of the linear models:

```
rqs_diff <-
  contrast_models(rsq_anova,
                  list_1 = "splines_lm",
                  list_2 = "basic_lm",
                  seed = 1103)

rqs_diff %>%
  as_tibble() %>%
  ggplot(aes(x = difference)) +
  geom_vline(xintercept = 0, lty = 2) +
  geom_histogram(bins = 50, col = "white", fill = "red", alpha = 0.4) +
  labs(x = expression(paste("Posterior for mean difference in ", R^2,
                            " (splines + interaction - basic)")))
```



The posterior shows that the center of the distribution is greater than zero (indicating that the model with splines had larger values) but does overlap with zero to a degree. The `summary()` method for this object computes the mean of the distribution as well as *credible intervals*.

```
summary(rqs_diff) %>% 
  select(-starts_with("pract"))
#> # A tibble: 1 × 6
#>   contrast               probability    mean   lower  upper  size
#>   <chr>                        <dbl>   <dbl>   <dbl>  <dbl> <dbl>
#> 1 splines_lm vs basic_lm       0.999 0.00800 0.00394 0.0122     0
```

The `probability` column reflects the proportion of the posterior that is greater than zero. This is the probability that the positive difference is real. The value is not close to zero, providing a strong case for *statistical significance*.

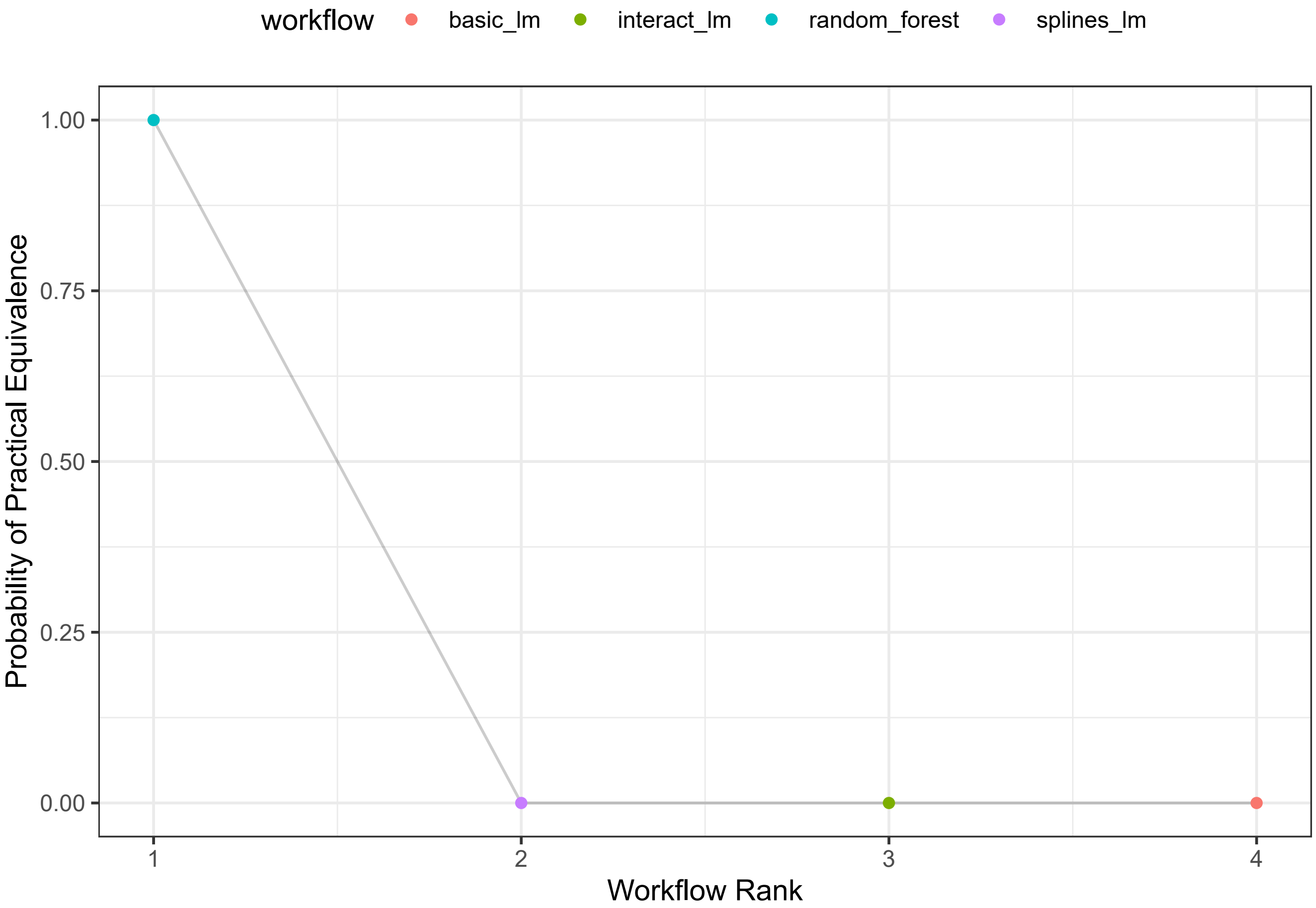
However, the estimate of the mean difference is fairly close to zero. Recall that the practical effect size we suggested previously is 2%. With a posterior distribution, the *probability of being practically significant* is also computed. In Bayesian analysis, this is a "ROPE estimate" (for Region Of Practical Equivalence, Kruschke and Liddell (2018)). To estimate this, the `size` option to the summary function is used:

```
summary(rqs_diff, size = 0.02) %>% 
  select(contrast, starts_with("pract"))
#> # A tibble: 1 × 4
#>   contrast               pract_neg pract_equiv pract_pos
#>   <chr>                      <dbl>       <dbl>     <dbl>
#> 1 splines_lm vs basic_lm         0           1         0
```

The `pract_equiv` column is the proportion of the posterior that is within `[-size, size]`. This large value indicates that, for our effect size, there is an overwhelming probability that the two models are practically the same.

The same process could be used to compare the random forest model to one or both of the linear regressions that were resampled. In fact, when `perf_mod()` is used with a workflow set, the `autoplot()` method can show the `pract_equiv` results that compare each workflow to the current best (the random forest model, in this case).

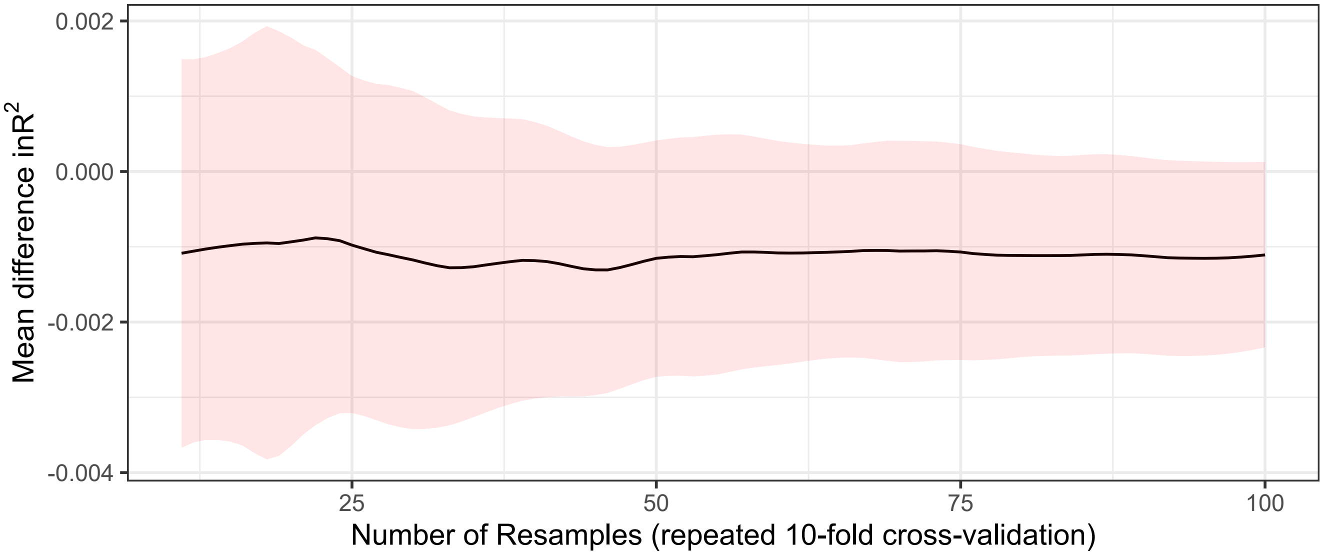
```
autoplot(rsq_anova, type = "ROPE", size = 0.02)
```



None of the linear models come close to the random forest model when a 2% practical effect size is used.

THE EFFECT OF THE AMOUNT OF RESAMPLING

How does the number of resamples affect this analysis? More resamples increases the precision of the overall resampling estimate; that precision propagates to this type of analysis. For illustration, additional resamples were added using repeated cross-validation. How did the posterior distribution change? The figure below shows the 90% credible intervals with up to 100 resamples (generated from 10 repeats of 10-fold cross-validation):



The width of the intervals decreases as more resamples are added. Clearly, going from ten resamples to thirty has a larger impact than going from eighty to 100. There are diminishing returns for using a “large” number of resamples (“large” will be different for different data sets).

11.5 CHAPTER SUMMARY

This chapter describes formal statistical methods for testing differences in performance between models. We demonstrated the within-resample effect, where results for the same resample tend to be similar; this aspect of resampled summary statistics requires appropriate analysis in order for valid model comparisons. Further, although statistical significance and practical significance are both important concepts for model comparisons, they are different.

REFERENCES

Faraway, J. 2016. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. CRC press.

Kruschke, J, and T Liddell. 2018. “The Bayesian New Statistics: Hypothesis Testing, Estimation, Meta-Analysis, and Power Analysis from a Bayesian Perspective.” *Psychonomic Bulletin and Review* 25 (1): 178–206.

McElreath, R. 2020. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC press.

Wasserstein, R, and N Lazar. 2016. “The ASA Statement on P-Values: Context, Process, and Purpose.” *The American Statistician* 70 (2): 129–33.