Importance measures

```r
set.seed(42)

library(rcompanion) # KW effect size calculation
library(rstatix) # Wilcox effect size calculation
```

```
##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

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

```r
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(QuantPsyc) # for the multivariate normality test
```

```
## Loading required package: boot

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: purrr

##
## Attaching package: 'purrr'
## The following objects are masked from 'package:igraph':
##
##     compose, simplify
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## The following object is masked from 'package:rstatix':
##
##     select
```

```
##
## Attaching package: 'QuantPsyc'
```

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

```r
library(dunn.test)
library(nFactors) # for the scree plot
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
##
## Attaching package: 'nFactors'
```

```
## The following object is masked from 'package:lattice':
##
##     parallel
```

```r
library(psych) # for PA FA
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
## The following object is masked from 'package:rcompanion':
##
##     phi
```

```r
library(caret) # highly correlated features removal
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.2.0 --

## v broom        1.0.5     v tibble       3.2.1
## v dials        1.3.0     v tidyr        1.3.1
## v infer        1.0.7     v tune         1.2.1
## v modeldata    1.4.0     v workflows    1.1.4
## v parsnip      1.2.1     v workflowsets 1.1.0
## v recipes      1.1.0     v yardstick    1.3.2
## v rsample      1.2.1

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x ggplot2::%+%()           masks psych::%+%()
## x yardstick::accuracy()    masks rcompanion::accuracy()
## x scales::alpha()          masks ggplot2::alpha(), psych::alpha()
## x tibble::as_data_frame()  masks dplyr::as_data_frame(), igraph::as_data_frame()
## x infer::chisq_test()      masks rstatix::chisq_test()
## x purrr::compose()         masks igraph::compose()
## x tidyr::crossing()        masks igraph::crossing()
## x dials::degree()          masks igraph::degree()
## x scales::discard()        masks purrr::discard()
## x dplyr::filter()          masks rstatix::filter(), stats::filter()
## x dials::get_n()           masks rstatix::get_n()
## x dplyr::lag()             masks stats::lag()
## x caret::lift()            masks purrr::lift()
## x dials::neighbors()       masks igraph::neighbors()
## x yardstick::precision()   masks caret::precision()
## x infer::prop_test()       masks rstatix::prop_test()
## x yardstick::recall()      masks caret::recall()
## x MASS::select()           masks dplyr::select(), rstatix::select()
## x yardstick::sensitivity() masks caret::sensitivity()
## x purrr::simplify()        masks igraph::simplify()
## x yardstick::specificity() masks caret::specificity()
## x recipes::step()          masks stats::step()
## x infer::t_test()          masks rstatix::t_test()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```r
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.3     v stringr   1.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::%--%()       masks igraph::%--%()
## x ggplot2::%+%()          masks psych::%+%()
## x scales::alpha()         masks ggplot2::alpha(), psych::alpha()
## x tibble::as_data_frame() masks dplyr::as_data_frame(), igraph::as_data_frame()
## x readr::col_factor()     masks scales::col_factor()
## x purrr::compose()        masks igraph::compose()
## x tidyr::crossing()       masks igraph::crossing()
## x scales::discard()       masks purrr::discard()
## x dplyr::filter()         masks rstatix::filter(), stats::filter()
## x stringr::fixed()        masks recipes::fixed()
## x dplyr::lag()            masks stats::lag()
## x caret::lift()           masks purrr::lift()
## x MASS::select()          masks dplyr::select(), rstatix::select()
## x purrr::simplify()       masks igraph::simplify()
## x readr::spec()           masks yardstick::spec()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(paletteer) # color palettes

library(conflicted) # to resolve QuantPsyc x dplyr conflicts
conflict_prefer("select", "dplyr")
```

```
## [conflicted] Will prefer dplyr::select over any other package.
```

```r
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

Load and tidy data

```r
pretty_names <- read_csv("../feat_name_mapping.csv")
```

```
## Rows: 85 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): name_orig, name_pretty
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
prettify_feat_name <- function(x) {
  name <- pull(pretty_names %>%
    filter(name_orig == x), name_pretty)
  if (length(name) == 1) {
    return(name)
  } else {
    return(x)
  }
}

prettify_feat_name_vector <- function(x) {
  map(
    x,
    prettify_feat_name
  ) %>% unlist()
}


data <- read_csv("../measurements/measurements.csv")
```

```
## Rows: 753 Columns: 108
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (20): fpath, KUK_ID, FileName, FileFormat, FolderPath, subcorpus, Source...
## dbl (85): RuleAbstractNouns, RuleAmbiguousRegards, RuleAnaphoricReferences, ...
## lgl  (3): ClarityPursuit, SyllogismBased, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
.firstnonmetacolumn <- 17

data_no_nas <- data %>%
  select(!c(
    fpath,
    # KUK_ID,
    # FileName,
    FolderPath,
    # subcorpus,
    DocumentTitle,
    ClarityPursuit,
    Readability,
    SyllogismBased,
    SourceDB
  )) %>%
  # replace -1s in variation coefficients with NAs
  mutate(across(c(
    `RuleDoubleAdpos.max_allowable_distance.v`,
    `RuleTooManyNegations.max_negation_frac.v`,
    `RuleTooManyNegations.max_allowable_negations.v`,
    `RuleTooManyNominalConstructions.max_noun_frac.v`,
    `RuleTooManyNominalConstructions.max_allowable_nouns.v`,
    `RuleCaseRepetition.max_repetition_count.v`,
    `RuleCaseRepetition.max_repetition_frac.v`,
    `RulePredSubjDistance.max_distance.v`,
    `RulePredObjDistance.max_distance.v`,
    `RuleInfVerbDistance.max_distance.v`,
    `RuleMultiPartVerbs.max_distance.v`,
    `RuleLongSentences.max_length.v`,
    `RulePredAtClauseBeginning.max_order.v`,
    `mattr.v`,
    `maentropy.v`
```

```
), ~ na_if(.x, -1))) %>%
# replace NAs with 0s
replace_na(list(
  RuleGPcoordovs = 0,
  RuleGPdeverbaddr = 0,
  RuleGPpatinstr = 0,
  RuleGPdeverbsubj = 0,
  RuleGPadjective = 0,
  RuleGPpatbenperson = 0,
  RuleGPwordorder = 0,
  RuleDoubleAdpos = 0,
  RuleDoubleAdpos.max_allowable_distance.v = 0,
  RuleAmbiguousRegards = 0,
  RuleReflexivePassWithAnimSubj = 0,
  RuleTooManyNegations = 0,
  RuleTooManyNegations.max_negation_frac.v = 0,
  RuleTooManyNegations.max_allowable_negations.v = 0,
  RuleTooManyNominalConstructions.max_noun_frac.v = 0,
  RuleTooManyNominalConstructions.max_allowable_nouns.v = 0,
  RuleFunctionWordRepetition = 0,
  RuleCaseRepetition.max_repetition_count.v = 0,
  RuleCaseRepetition.max_repetition_frac.v = 0,
  RuleWeakMeaningWords = 0,
  RuleAbstractNouns = 0,
  RuleRelativisticExpressions = 0,
  RuleConfirmationExpressions = 0,
  RuleRedundantExpressions = 0,
  RuleTooLongExpressions = 0,
  RuleAnaphoricReferences = 0,
  RuleLiteraryStyle = 0,
  RulePassive = 0,
  RulePredSubjDistance = 0,
  RulePredSubjDistance.max_distance.v = 0,
  RulePredObjDistance = 0,
  RulePredObjDistance.max_distance.v = 0,
  RuleInfVerbDistance = 0,
  RuleInfVerbDistance.max_distance.v = 0,
  RuleMultiPartVerbs = 0,
  RuleMultiPartVerbs.max_distance.v = 0,
  RuleLongSentences.max_length.v = 0,
  RulePredAtClauseBeginning.max_order.v = 0,
  RuleVerbalNouns = 0,
  RuleDoubleComparison = 0,
  RuleWrongValencyCase = 0,
  RuleWrongVerbonominalCase = 0,
  RuleIncompleteConjunction = 0
)) %>%
# replace NAs with medians
mutate(across(c(
  RuleDoubleAdpos.max_allowable_distance,
  RuleTooManyNegations.max_negation_frac,
  RuleTooManyNegations.max_allowable_negations,
  RulePredSubjDistance.max_distance,
    RulePredObjDistance.max_distance,
    RuleInfVerbDistance.max_distance,
    RuleMultiPartVerbs.max_distance
  ), ~ coalesce(., median(., na.rm = TRUE)))) %>%
  # merge GPs
  mutate(
    GPs = RuleGPcoordovs +
      RuleGPdeverbaddr +
      RuleGPpatinstr +
      RuleGPdeverbsubj +
      RuleGPadjective +
      RuleGPpatbenperson +
      RuleGPwordorder
  ) %>%
  select(!c(
    RuleGPcoordovs,
    RuleGPdeverbaddr,
    RuleGPpatinstr,
    RuleGPdeverbsubj,
    RuleGPadjective,
    RuleGPpatbenperson,
    RuleGPwordorder
  ))

data_clean <- data_no_nas %>%
  # norm data expected to correlate with text length
  mutate(across(c(
    GPs,
    RuleDoubleAdpos,
    RuleAmbiguousRegards,
    RuleFunctionWordRepetition,
    RuleWeakMeaningWords,
    RuleAbstractNouns,
    RuleRelativisticExpressions,
    RuleConfirmationExpressions,
    RuleRedundantExpressions,
    RuleTooLongExpressions,
    RuleAnaphoricReferences,
    RuleLiteraryStyle,
    RulePassive,
    RuleVerbalNouns,
    RuleDoubleComparison,
    RuleWrongValencyCase,
    RuleWrongVerbonominalCase,
    RuleIncompleteConjunction,
    num_hapax,
    RuleReflexivePassWithAnimSubj,
    RuleTooManyNominalConstructions,
    RulePredSubjDistance,
    RuleMultiPartVerbs,
    RulePredAtClauseBeginning
  ), ~ .x / word_count)) %>%
  mutate(across(c(
    RuleTooFewVerbs,
    RuleTooManyNegations,
    RuleCaseRepetition,
    RuleLongSentences,
    RulePredObjDistance,
    RuleInfVerbDistance
  ), ~ .x / sent_count)) %>%
  # remove variables identified as text-length dependent
  select(!c(
    RuleTooFewVerbs,
    RuleTooManyNegations,
    RuleTooManyNominalConstructions,
    RuleCaseRepetition,
    RuleLongSentences,
    RulePredAtClauseBeginning,
    syllab_count,
    char_count
  )) %>%
  # remove variables identified as unreliable
  select(!c(
    RuleAmbiguousRegards,
    RuleFunctionWordRepetition,
    RuleDoubleComparison,
    RuleWrongValencyCase,
    RuleWrongVerbonominalCase
  )) %>%
  # remove further variables belonging to the 'acceptability' category
  select(!c(RuleIncompleteConjunction)) %>%
  # remove artificially limited variables
  select(!c(
    RuleCaseRepetition.max_repetition_frac,
    RuleCaseRepetition.max_repetition_frac.v
  )) %>%
  # remove variables with too many NAs
  select(!c(
    RuleDoubleAdpos.max_allowable_distance,
    RuleDoubleAdpos.max_allowable_distance.v
  )) %>%
  mutate(across(c(
    class,
    FileFormat,
    subcorpus,
    DocumentVersion,
    LegalActType,
    Objectivity,
    AuthorType,
    RecipientType,
    RecipientIndividuation,
    Anonymized
  ), ~ as.factor(.x)))

# no NAs should be present now
data_clean[!complete.cases(data_clean[.firstnonmetacolumn:ncol(data_clean)]), ]
```

```
## # A tibble: 0 x 77
## # i 77 variables: KUK_ID <chr>, FileName <chr>, FileFormat <fct>,
## #   subcorpus <fct>, SourceID <chr>, DocumentVersion <fct>,
## #   ParentDocumentID <chr>, LegalActType <fct>, Objectivity <fct>,
## #   Bindingness <lgl>, AuthorType <fct>, RecipientType <fct>,
## #   RecipientIndividuation <fct>, Anonymized <fct>, Recipient Type <chr>,
## #   class <fct>, RuleAbstractNouns <dbl>, RuleAnaphoricReferences <dbl>,
## #   RuleCaseRepetition.max_repetition_count <dbl>, ...
```

```
colnames(data_clean) <- prettify_feat_name_vector(colnames(data_clean))

data_clean_scaled <- data_clean %>%
  mutate(across(class, ~ .x == "good")) %>%
  mutate(across(.firstnonmetacolumn:ncol(data_clean), ~ scale(.x)))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(.firstnonmetacolumn:ncol(data_clean), ~scale(.x))`.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(.firstnonmetacolumn)
##
##   # Now:
##   data %>% select(all_of(.firstnonmetacolumn))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

Important features identification

Regularized regression

split the data

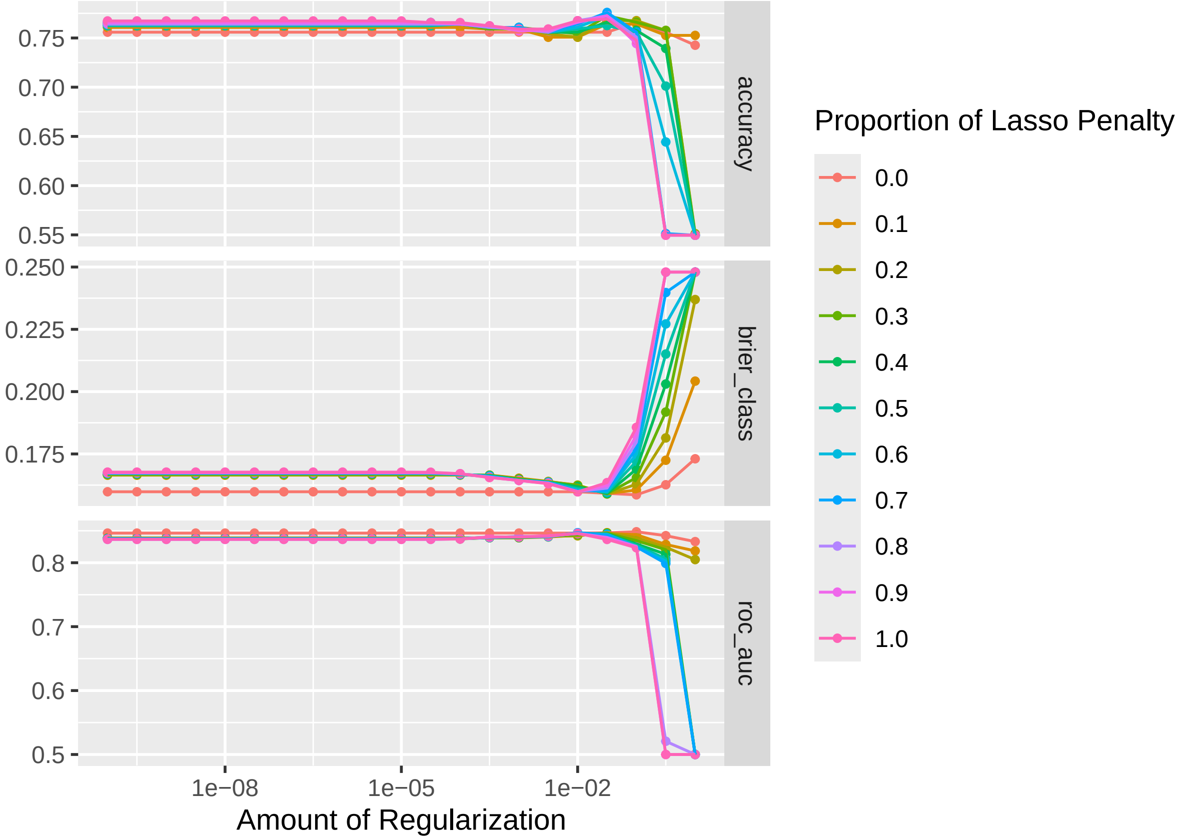
```
.no_folds <- 10
.split_prop <- 4 / 5

data_split <- initial_split(data_clean, strata = class, prop = .split_prop)
training_set <- training(data_split)
testing_set <- testing(data_split)

folds <- vfold_cv(training_set, .no_folds)
```

recipe

```
lin_formula <- reformulate(colnames(data_clean)[17:77], "class")
lin_rec <- recipe(lin_formula, data = training_set) %>%
  # step_corr(all_predictors()) %>%
  step_normalize(all_predictors())

lin_wf_base <- workflow() %>% add_recipe(lin_rec)
```

tuning

```
lin_wf <- lin_wf_base %>%
  add_model(logistic_reg(
    mode = "classification", engine = "glmnet",
    penalty = tune(), mixture = tune()
  ))

tune_grid <- grid_regular(
  penalty(), mixture(),
  levels = c(penalty = 21, mixture = 11)
)

tune_rs <- tune_grid(
  lin_wf, folds,
  grid = tune_grid,
  metrics = metric_set(yardstick::accuracy, brier_class, roc_auc)
)

autoplot(tune_rs)
```



```
choose_roc_auc <- tune_rs %>%
  select_by_one_std_err(metric = "roc_auc", -mixture, penalty)
choose_roc_auc
```

```
## # A tibble: 1 x 3
##         penalty mixture .config
##           <dbl>   <dbl> <chr>
## 1 0.0000000001       1 Preprocessor1_Model211
```

final

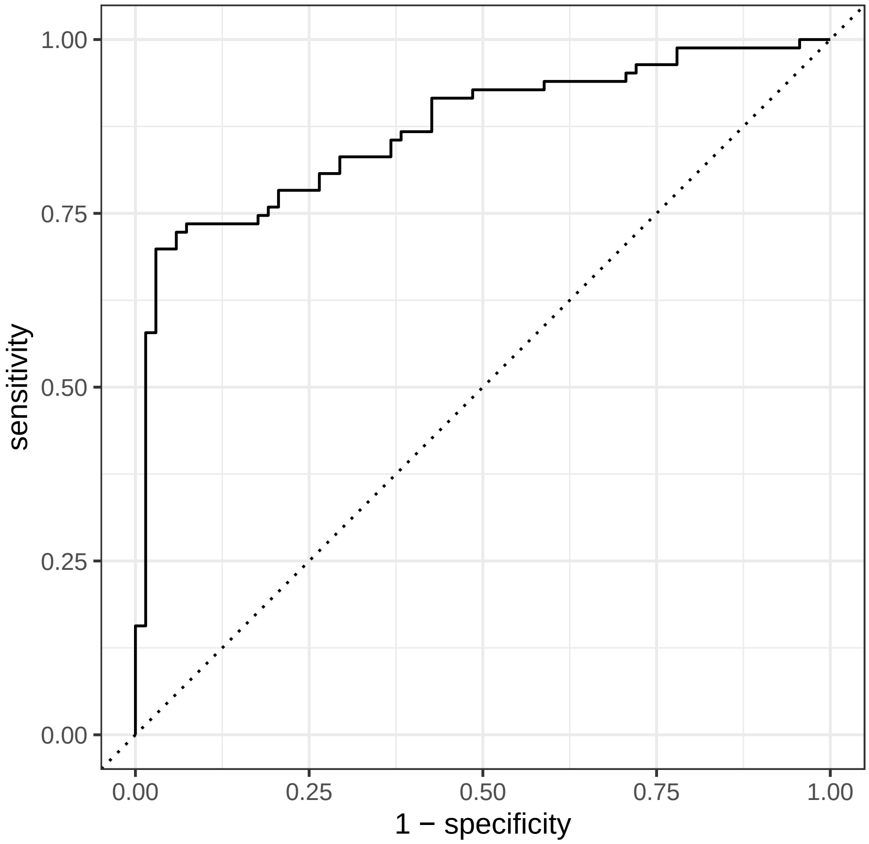
```r
lin_final_wf <- finalize_workflow(lin_wf, choose_roc_auc)
lin_final_wf
```

```
## == Workflow ====================================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 1 Recipe Step
##
## * step_normalize()
##
## -- Model -----------------------------------------------------------------------
## Logistic Regression Model Specification (classification)
##
## Main Arguments:
##   penalty = 1e-10
##   mixture = 1
##
## Computational engine: glmnet
```

```r
lin_final_fitted <- last_fit(lin_final_wf, data_split)

collect_predictions(lin_final_fitted) %>%
  conf_mat(truth = class, estimate = .pred_class)
```

```
##           Truth
## Prediction bad good
##       bad   64   14
##       good  19   54
```

```r
collect_predictions(lin_final_fitted) %>%
  roc_curve(truth = class, .pred_bad) %>%
  autoplot()
```



```
extract_fit_parsnip(lin_final_fitted) %>%
  vip::vi(lambda = choose_roc_auc$penalty) %>%
  print(n = 80)
```

```
## # A tibble: 61 x 3
##    Variable            Importance Sign
##    <chr>                    <dbl> <chr>
##  1 sentlen.m             2.99     POS
##  2 ari                   2.64     NEG
##  3 gf                    1.96     NEG
##  4 sentcount             1.86     POS
##  5 atl                   1.41     POS
##  6 activity              1.37     POS
##  7 VERBfrac.m            1.32     NEG
##  8 smog                  1.17     POS
##  9 hpoint                1.13     NEG
## 10 wordcount             1.05     NEG
## 11 ttr                   0.886    NEG
## 12 fre                   0.806    NEG
## 13 entropy.v             0.720    POS
## 14 entropy               0.693    NEG
## 15 sentlen.v             0.580    POS
## 16 ttr.v                 0.541    NEG
## 17 predsubjdist.m        0.493    NEG
## 18 anaphoricrefs         0.447    POS
## 19 cli                   0.430    NEG
## 20 extrcaseexprs         0.411    POS
## 21 compoundVERBs         0.410    POS
## 22 passives              0.402    NEG
## 23 mattr                 0.347    NEG
## 24 caserepcount.v        0.339    NEG
## 25 predobjdist.m         0.321    NEG
## 26 literary              0.314    NEG
## 27 verbdist              0.308    POS
## 28 caserepcount.m        0.307    POS
## 29 maentropy             0.285    POS
## 30 predorder.m           0.267    NEG
## 31 hapaxes               0.263    POS
## 32 VERBcomp              0.247    POS
## 33 NOUNcount.v           0.227    NEG
## 34 subj                  0.223    POS
## 35 NOUNcount.m           0.212    POS
## 36 VERBcompdist.v        0.208    NEG
## 37 predobjdist.v         0.203    POS
## 38 rfpass_animsubj       0.197    NEG
## 39 NEGcount.m            0.188    POS
## 40 NOUNfrac.m            0.184    NEG
## 41 longexprs             0.179    POS
## 42 redundexprs           0.177    NEG
## 43 compoundVERBsdist.m   0.175    NEG
## 44 doubleADPs            0.168    NEG
## 45 VERBfrac.v            0.157    POS
## 46 relativisticexprs     0.157    NEG
## 47 NEGcount.v            0.145    NEG
## 48 compoundVERBsdist.v   0.139    POS
## 49 NEGfrac.v             0.126    POS
## 50 VERBcompdist.m        0.126    POS
## 51 GPs                   0.105    NEG
## 52 predsubjdist.v        0.0944   NEG
## 53 mamr                  0.0940   NEG
## 54 NOUNfrac.v            0.0857   POS
## 55 obj                   0.0766   POS
## 56 weakmeaning           0.0758   NEG
## 57 predorder.v           0.0467   POS
## 58 verbalNOUNs           0.0348   NEG
## 59 abstractNOUNs         0.00983  POS
## 60 NEGfrac.m             0.000988 POS
## 61 fkgl                  0        NEG
```

```
lin_final_fitted %>%
  extract_fit_parsnip() %>%
  tidy() %>%
  arrange(estimate) %>%
  print(n = 80)
```

```
## # A tibble: 62 x 3
##    term                 estimate      penalty
##    <chr>                   <dbl>        <dbl>
##  1 ari                   -2.64     0.0000000001
##  2 gf                    -1.96     0.0000000001
##  3 VERBfrac.m            -1.32     0.0000000001
##  4 hpoint                -1.13     0.0000000001
##  5 wordcount           -1.05     0.0000000001
##  6 ttr                 -0.886    0.0000000001
##  7 fre                 -0.806    0.0000000001
##  8 entropy             -0.693    0.0000000001
##  9 (Intercept)         -0.542    0.0000000001
## 10 ttr.v               -0.541    0.0000000001
## 11 predsubjdist.m      -0.493    0.0000000001
## 12 cli                 -0.430    0.0000000001
## 13 passives            -0.402    0.0000000001
## 14 mattr               -0.347    0.0000000001
## 15 caserepcount.v      -0.339    0.0000000001
## 16 predobjdist.m       -0.321    0.0000000001
## 17 literary            -0.314    0.0000000001
## 18 predorder.m         -0.267    0.0000000001
## 19 NOUNcount.v         -0.227    0.0000000001
## 20 VERBcompdist.v      -0.208    0.0000000001
## 21 rfpass_animsubj     -0.197    0.0000000001
## 22 NOUNfrac.m          -0.184    0.0000000001
## 23 redundexprs         -0.177    0.0000000001
## 24 compoundVERBsdist.m -0.175    0.0000000001
## 25 doubleADPs          -0.168    0.0000000001
## 26 relativisticexprs   -0.157    0.0000000001
## 27 NEGcount.v          -0.145    0.0000000001
## 28 GPs                 -0.105    0.0000000001
## 29 predsubjdist.v      -0.0944   0.0000000001
## 30 mamr                -0.0940   0.0000000001
## 31 weakmeaning         -0.0758   0.0000000001
## 32 verbalNOUNs         -0.0348   0.0000000001
## 33 fkgl                 0        0.0000000001
## 34 NEGfrac.m            0.000988 0.0000000001
## 35 abstractNOUNs        0.00983  0.0000000001
## 36 predorder.v          0.0467   0.0000000001
## 37 obj                  0.0766   0.0000000001
## 38 NOUNfrac.v           0.0857   0.0000000001
## 39 VERBcompdist.m       0.126    0.0000000001
## 40 NEGfrac.v            0.126    0.0000000001
## 41 compoundVERBsdist.v  0.139    0.0000000001
## 42 VERBfrac.v           0.157    0.0000000001
## 43 longexprs            0.179    0.0000000001
## 44 NEGcount.m           0.188    0.0000000001
## 45 predobjdist.v        0.203    0.0000000001
## 46 NOUNcount.m          0.212    0.0000000001
## 47 subj                 0.223    0.0000000001
## 48 VERBcomp             0.247    0.0000000001
## 49 hapaxes              0.263    0.0000000001
## 50 maentropy            0.285    0.0000000001
## 51 caserepcount.m       0.307    0.0000000001
## 52 verbdist             0.308    0.0000000001
## 53 compoundVERBs        0.410    0.0000000001
## 54 extrcaseexprs        0.411    0.0000000001
## 55 anaphoricrefs        0.447    0.0000000001
## 56 sentlen.v            0.580    0.0000000001
## 57 entropy.v            0.720    0.0000000001
## 58 smog                 1.17     0.0000000001
## 59 activity             1.37     0.0000000001
## 60 atl                  1.41     0.0000000001
## 61 sentcount            1.86     0.0000000001
## 62 sentlen.m            2.99     0.0000000001
```

Individual regressions

```r
data_scaled <- data_clean %>%
  mutate(across(all_of(.firstnonmetacolumn:ncol(data_clean)), ~ scale(.x)[, 1]))

feature_importances <- tibble(
  feat_name = character(),
  p_value = numeric(),
  estimate = numeric(),
  wilcox_p = numeric(),
  wilcox_r = numeric(),
  kw_p = numeric(),
  kw_chi2 = numeric(),
  kw_epsilon2 = numeric(),
  kw_epsilon2_lci = numeric(),
  kw_epsilon2_uci = numeric(),
  med_sign = numeric(),
  mean_sign = numeric()
)

for (i in .firstnonmetacolumn:ncol(data_scaled)) {
  fname <- names(data_scaled)[i]
  message(fname)

  formula_single <- reformulate(fname, "class")
  formula_single_reversed <- reformulate("class", fname)

  glm_model <- glm(formula_single, data_scaled, family = "binomial")
  glm_coefficients <- summary(glm_model)$coefficients
  row_index <- which(rownames(glm_coefficients) == fname)
  p_value <- glm_coefficients[row_index, 4]
  beta <- glm_coefficients[row_index, 1]

  wilcox_p <- wilcox.test(formula_single_reversed, data_scaled)$p.value
  wilcox_r <- wilcox_effsize(data_scaled, formula_single_reversed)$effsize[[1]]

  kw <- kruskal.test(data_scaled[[fname]], data_scaled$class)
  kw_p <- kw$p.value
  kw_chi2 <- kw$statistic[[1]]
  kw_epsilon2_t <- epsilonSquared(
    data_scaled[[fname]], data_scaled$class,
    ci = TRUE
  )
  kw_epsilon2 <- kw_epsilon2_t[[1]]
  kw_epsilon2_lci <- kw_epsilon2_t[[2]]
  kw_epsilon2_uci <- kw_epsilon2_t[[3]]

  med_good <- filter(data_scaled, class == "good")[[fname]] %>% median()
  med_bad <- filter(data_scaled, class == "bad")[[fname]] %>% median()
  med_sign <- sign(med_good - med_bad)

  mean_good <- filter(data_scaled, class == "good")[[fname]] %>% mean()
  mean_bad <- filter(data_scaled, class == "bad")[[fname]] %>% mean()
  mean_sign <- sign(mean_good - mean_bad)

  feature_importances <- feature_importances %>%
    add_row(
      feat_name = fname,
      p_value = p_value,
      estimate = beta,
      wilcox_p = wilcox_p,
      wilcox_r = wilcox_r,
      kw_p = kw_p,
      kw_chi2 = kw_chi2,
      kw_epsilon2 = kw_epsilon2,
      kw_epsilon2_uci = kw_epsilon2_uci,
      kw_epsilon2_lci = kw_epsilon2_lci,
      med_sign = med_sign,
      mean_sign = mean_sign,
    )
}
```

```
## abstractNOUNs

## anaphoricrefs

## caserepcount.m

## caserepcount.v

## extrcaseexprs

## doubleADPs

## VERBcomp

## VERBcompdist.m

## VERBcompdist.v

## literary

## sentlen.m

## sentlen.v

## compoundVERBs

## compoundVERBsdist.m

## compoundVERBsdist.v

## passives

## predorder.m

## predorder.v

## obj

## predobjdist.m
## predobjdist.v

## subj

## predsubjdist.m

## predsubjdist.v

## redundexprs

## rfpass_animsubj

## relativisticexprs

## VERBfrac.m

## VERBfrac.v

## longexprs

## NEGcount.m

## NEGcount.v

## NEGfrac.m

## NEGfrac.v

## NOUNcount.m

## NOUNcount.v

## NOUNfrac.m

## NOUNfrac.v

## verbalNOUNs

## weakmeaning

## activity

## ari

## atl

## cli

## entropy

## fkgl

## fre

## gf

## hpoint

## maentropy

## entropy.v

## mamr

## mattr

## ttr.v

## hapaxes

## sentcount
## smog
```

```
## ttr
```

```
## verbdist
```

```
## wordcount
```

```
## GPs
```

```
feature_importances
```

```
## # A tibble: 61 x 12
##    feat_name     p_value estimate wilcox_p wilcox_r     kw_p kw_chi2 kw_epsilon2
##    <chr>           <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>       <dbl>
##  1 abstractNOU~ 2.20e- 3   0.232  6.39e- 3   0.0994 6.39e- 3    7.44     0.00989
##  2 anaphoricre~ 6.73e- 1   0.0308 9.80e- 3   0.0941 9.79e- 3    6.67     0.00887
##  3 caserepcoun~ 6.59e- 2  -0.137  7.61e- 2   0.0647 7.60e- 2    3.15     0.00419
##  4 caserepcoun~ 4.54e- 3  -0.215  9.43e- 4   0.121  9.43e- 4   10.9      0.0145
##  5 extrcaseexp~ 1.08e- 1  -0.123  1.34e- 3   0.117  1.34e- 3   10.3      0.0137
##  6 doubleADPs   2.71e- 1  -0.0816 3.02e- 1   0.0376 3.02e- 1    1.06     0.00141
##  7 VERBcomp     5.24e-15   0.659  1.36e-16   0.301  1.36e-16   68.4      0.0909
##  8 VERBcompdis~ 5.48e- 2  -0.191  1.73e- 2   0.0868 1.73e- 2    5.67     0.00754
##  9 VERBcompdis~ 6.58e- 2  -0.137  7.90e- 2   0.0640 7.89e- 2    3.09     0.0041
## 10 literary     7.00e-21  -0.918  1.44e-26   0.389  1.44e-26  114.        0.151
## # i 51 more rows
## # i 4 more variables: kw_epsilon2_lci <dbl>, kw_epsilon2_uci <dbl>,
## #   med_sign <dbl>, mean_sign <dbl>
```

```
selected_features <- feature_importances %>%
  mutate(
    selected = p_value <= 0.05,
    wilcox_sel = wilcox_p < 0.05,
    kw_sel = kw_p < 0.05
  )

selected_features %>%
  select(selected, kw_sel) %>%
  table()
```

```
##          kw_sel
## selected FALSE TRUE
##    FALSE     8    4
##    TRUE      4   45
```

```
cor(-log(selected_features$p_value), selected_features$kw_epsilon2)
```

```
## [1] 0.952316
```

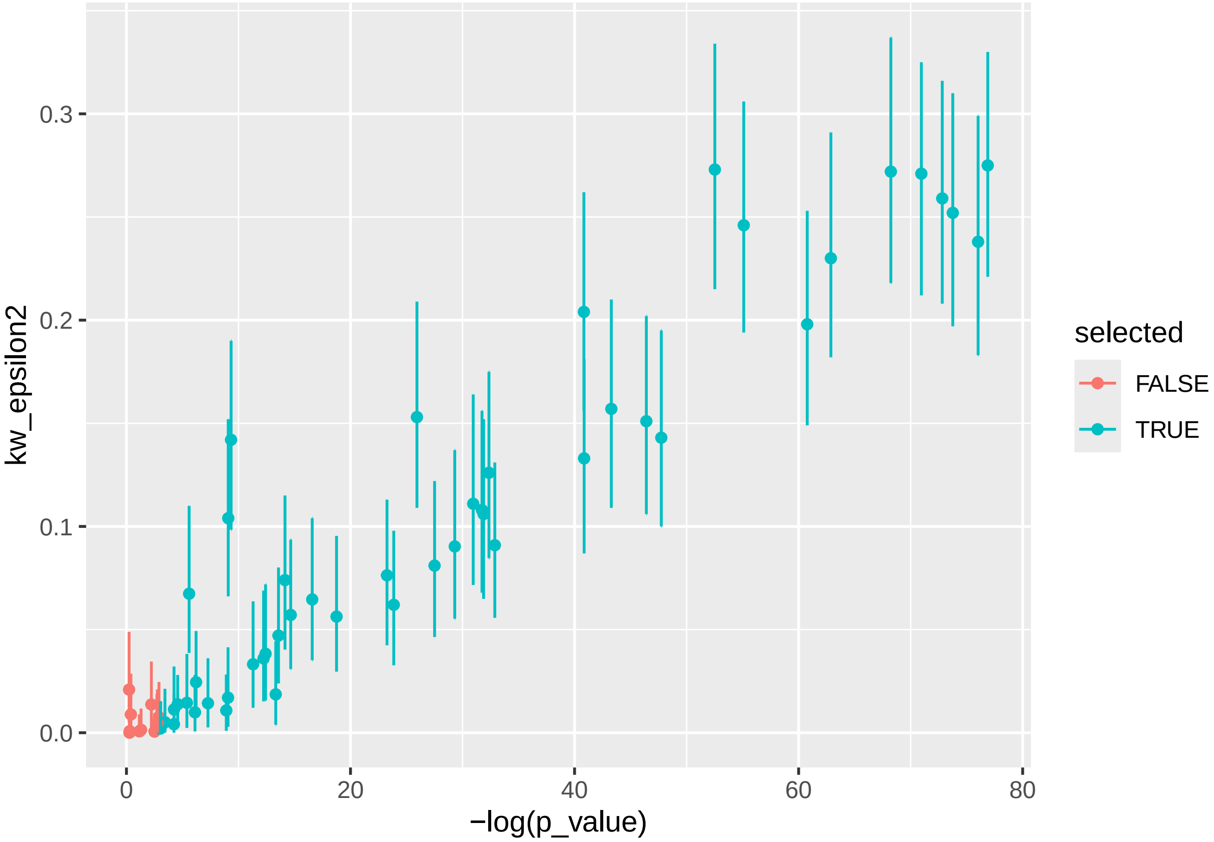
```
cor(-log(selected_features$p_value), -log(selected_features$kw_p))
```

```
## [1] 0.9524106
```

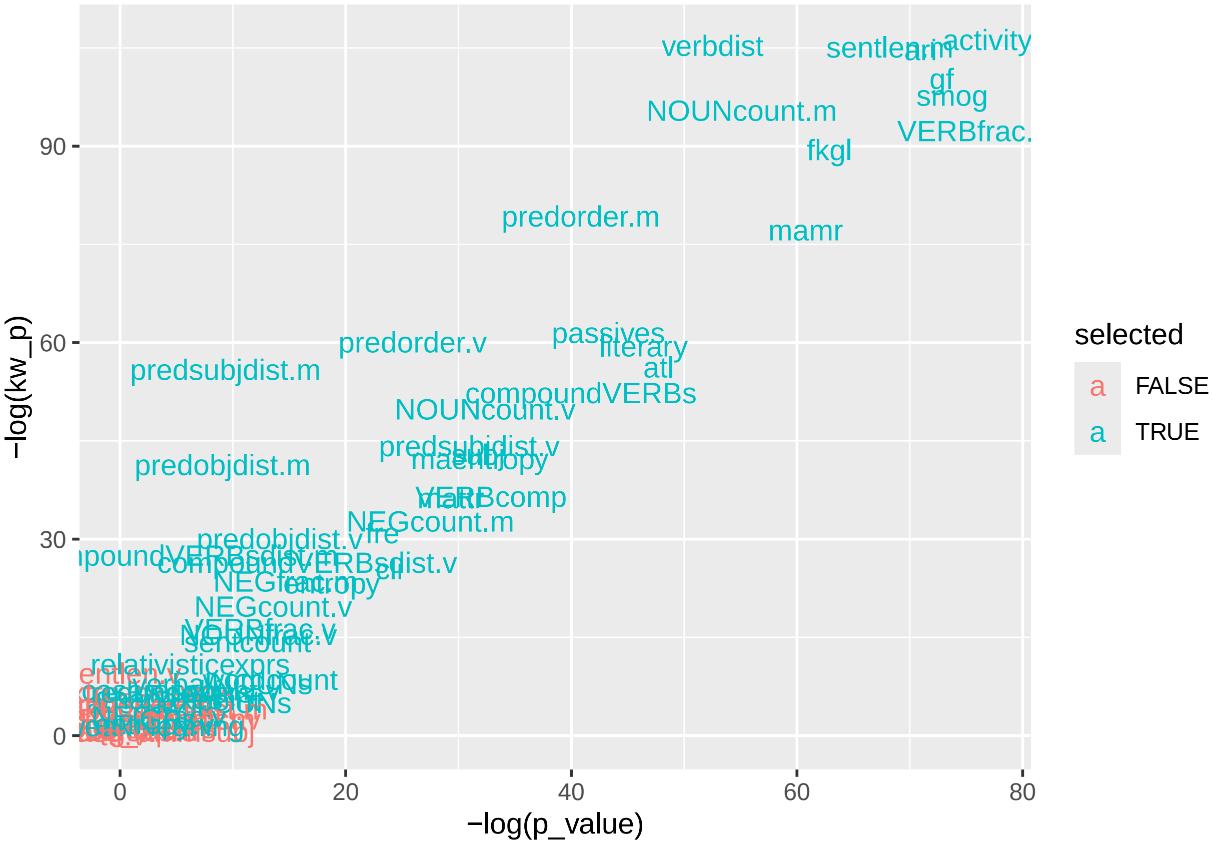
```
cor(selected_features$estimate, selected_features$kw_epsilon2)
```

```
## [1] -0.3662002
```

```
selected_features %>%
  ggplot(aes(
    x = -log(p_value), y = kw_epsilon2,
    ymin = kw_epsilon2_lci, ymax = kw_epsilon2_uci, color = selected
  )) +
  geom_point() +
  geom_errorbar()
```



```
selected_features %>%
  ggplot(aes(
    x = -log(p_value), y = -log(kw_p), color = selected, label = feat_name
  )) +
  # geom_point() +
  geom_text()
```



```
selected_features_names <- selected_features %>%
  filter(selected) %>%
  pull(feat_name)
```

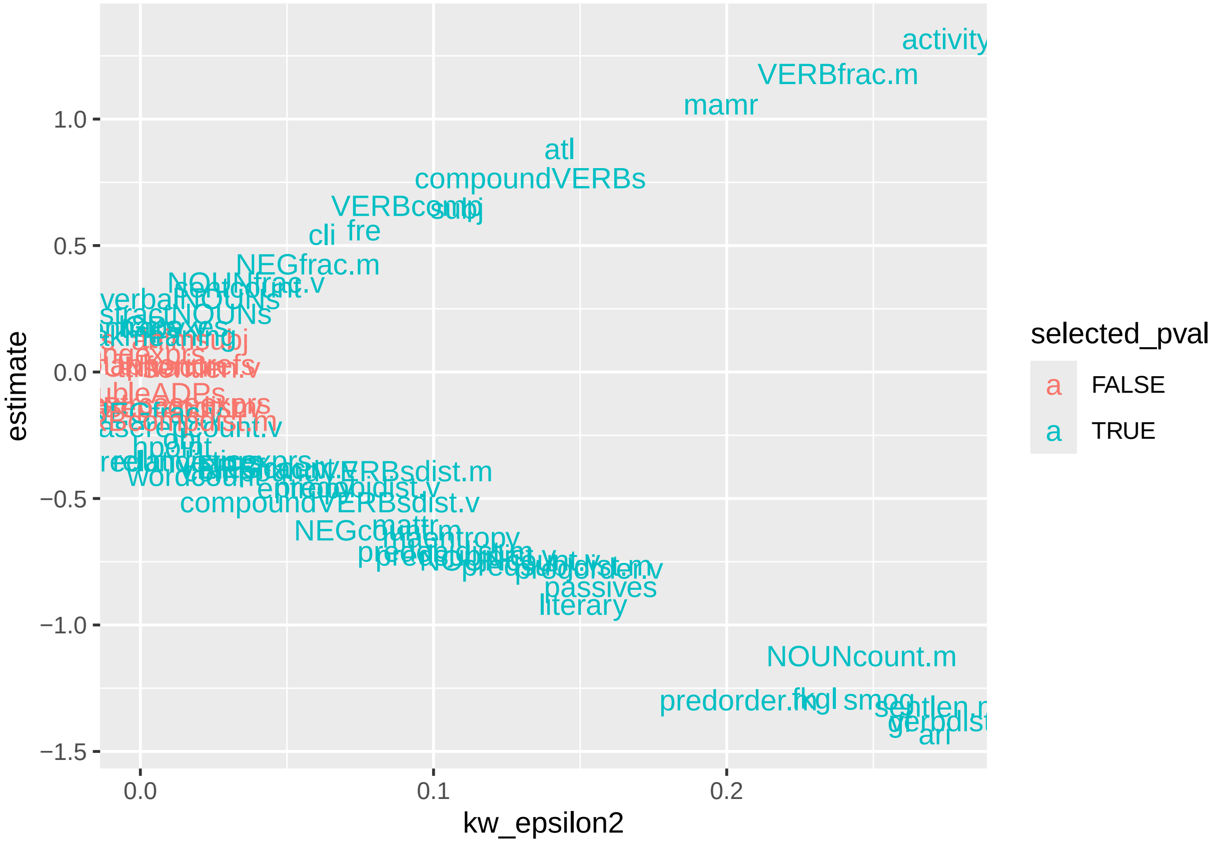
Compare the two

```
featcomp <- extract_fit_parsnip(lin_final_fitted) %>%
  vip::vi(lambda = choose_roc_auc$penalty) %>%
  full_join(
    selected_features %>% rename(Variable = feat_name),
    by = "Variable"
  ) %>%
  rename(selected_pval = selected) %>%
  mutate(
    log_p = -log(p_value),
    log_wilcox_p = -log(wilcox_p),
    log_kw_p = -log(kw_p),
    selected_reg = Importance > 0
  )

featcomp %>% write_csv("featcomp.csv")

featcomp %>%
  filter(!is.na(Importance)) %>%
  select(Importance, kw_epsilon2, log_p, log_kw_p) %>%
  cor() %>%
```

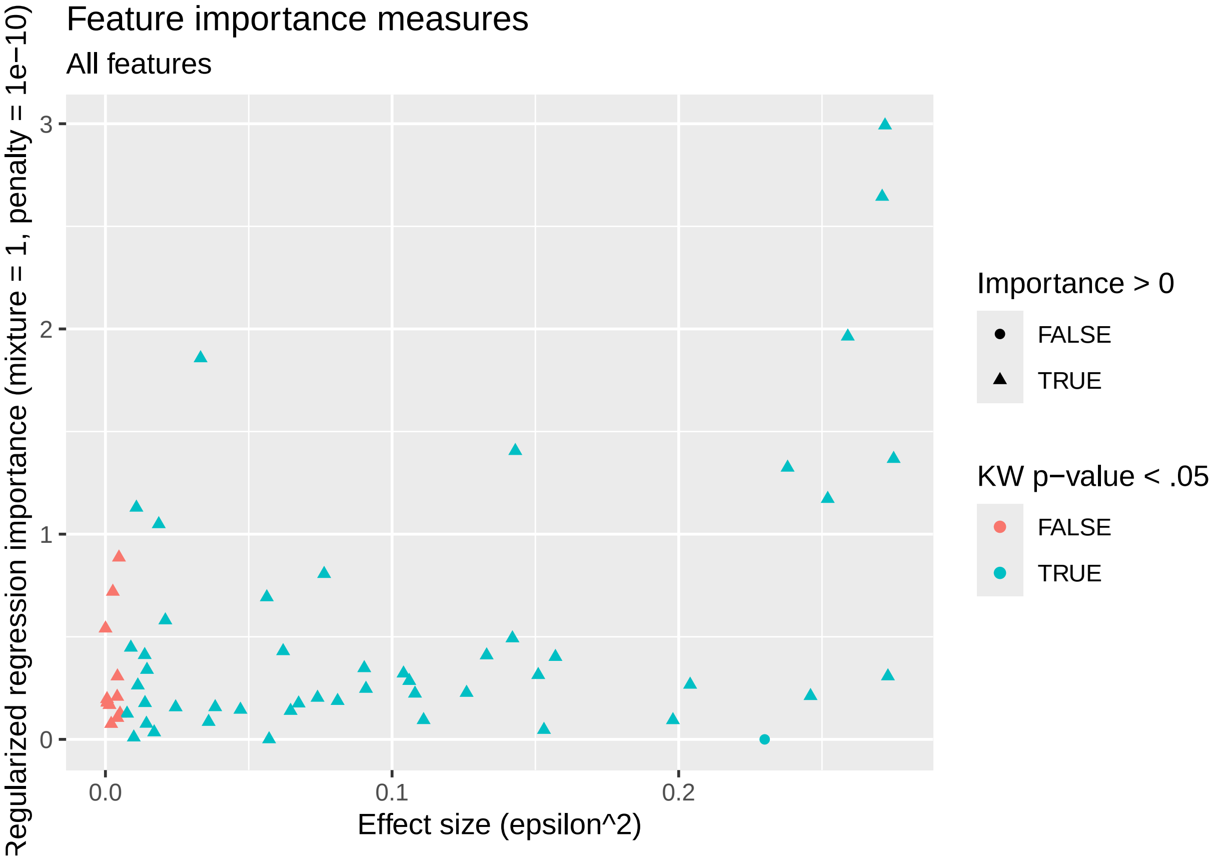
```r
  round(2)
```

```
##              Importance kw_epsilon2 log_p log_kw_p
## Importance         1.00        0.47  0.51     0.47
## kw_epsilon2        0.47        1.00  0.95     1.00
## log_p              0.51        0.95  1.00     0.95
## log_kw_p           0.47        1.00  0.95     1.00
```

```r
featcomp %>%
  ggplot(aes(
    x = kw_epsilon2, y = estimate, color = selected_pval, label = Variable
  )) +
  geom_text()
```



```r
featcomp_plot <- featcomp %>% ggplot(aes(
  x = kw_epsilon2,
  y = Importance,
  # size = log_p,
  color = kw_sel,
  shape = selected_reg
)) +
  geom_point() +
  labs(
    title = "Feature importance measures",
    subtitle = "All features",
    # subtitle = "Features with |r| < 0.90",
    x = "Effect size (epsilon^2)",
    y = paste0(c(
      "Regularized regression importance (mixture = ",
      choose_roc_auc$mixture[1], ", penalty = ",
      choose_roc_auc$penalty[1], ")"
    ), collapse = ""),
    # size = "-log(p-value)",
    color = "KW p-value < .05",
    shape = "Importance > 0"
  )
print(featcomp_plot)
```



```
ggsave("featcomp_all.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# ggsave("featcomp_nocorr.png")
```

Results

```
featcomp %>%
  filter(!kw_sel) %>%
  select(Variable, kw_chi2, kw_p) %>%
  arrange(Variable) %>%
  as.data.frame() %>%
  print(digits = 2)
```

```
##             Variable kw_chi2  kw_p
## 1              GPs   3.116 0.078
## 2        NEGfrac.v   3.835 0.050
## 3       NOUNfrac.m   0.582 0.446
## 4   VERBcompdist.v   3.087 0.079
## 5   caserepcount.m   3.148 0.076
## 6       doubleADPs   1.064 0.302
## 7        entropy.v   1.937 0.164
## 8        longexprs   0.513 0.474
## 9  rfpass_animsubj   0.414 0.520
## 10             ttr   3.550 0.060
## 11           ttr.v   0.022 0.882
## 12     weakmeaning   1.504 0.220
```

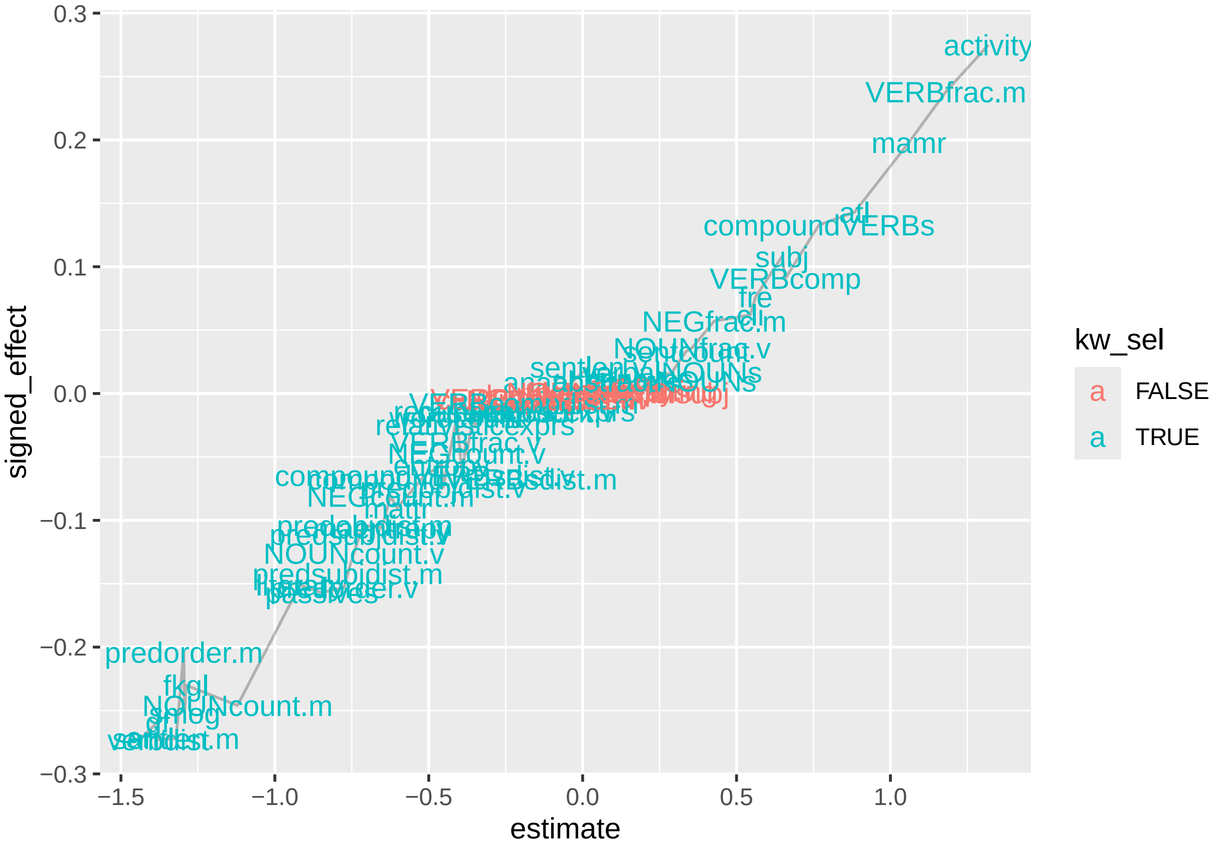
```r
featcomp %>%
  filter(kw_sel) %>%
  mutate(signed_effect = kw_epsilon2 * mean_sign) %>%
  select(Variable, kw_epsilon2, kw_p, signed_effect) %>%
  arrange(-kw_epsilon2) %>%
  as.data.frame() %>%
  print(digits = 2)
```

```
##               Variable kw_epsilon2    kw_p signed_effect
## 1             activity      0.2750 6.9e-47        0.2750
## 2             verbdist      0.2730 1.7e-46       -0.2730
## 3           sentlen.m      0.2720 2.2e-46       -0.2720
## 4                  ari      0.2710 3.2e-46       -0.2710
## 5                   gf      0.2590 2.7e-44       -0.2590
## 6                 smog      0.2520 3.4e-43       -0.2520
## 7          NOUNcount.m      0.2460 3.4e-42       -0.2460
## 8           VERBfrac.m      0.2380 7.7e-41        0.2380
## 9                 fkgl      0.2300 1.4e-39       -0.2300
## 10         predorder.m      0.2040 3.5e-35       -0.2040
## 11                mamr      0.1980 2.9e-34        0.1980
## 12            passives      0.1570 1.9e-27       -0.1570
## 13         predorder.v      0.1530 7.8e-27       -0.1530
## 14            literary      0.1510 1.4e-26       -0.1510
## 15                 atl      0.1430 3.6e-25        0.1430
## 16      predsubjdist.m      0.1420 5.2e-25       -0.1420
## 17       compoundVERBs      0.1330 1.8e-23        0.1330
## 18         NOUNcount.v      0.1260 2.2e-22       -0.1260
## 19      predsubjdist.v      0.1110 6.0e-20       -0.1110
## 20                subj      0.1080 2.2e-19        0.1080
## 21           maentropy      0.1060 4.3e-19       -0.1060
## 22       predobjdist.m      0.1040 1.1e-18       -0.1040
## 23            VERBcomp      0.0909 1.4e-16        0.0909
## 24               mattr      0.0903 1.7e-16       -0.0903
## 25          NEGcount.m      0.0810 5.9e-15       -0.0810
## 26                 fre      0.0763 3.6e-14        0.0763
## 27       predobjdist.v      0.0740 8.6e-14       -0.0740
## 28 compoundVERBsdist.m      0.0674 1.1e-12       -0.0674
## 29 compoundVERBsdist.v      0.0646 3.2e-12       -0.0646
## 30                 cli      0.0620 8.5e-12        0.0620
## 31          NEGfrac.m      0.0571 5.7e-11        0.0571
## 32             entropy      0.0563 7.6e-11       -0.0563
## 33          NEGcount.v      0.0471 2.6e-09       -0.0471
## 34          VERBfrac.v      0.0383 8.1e-08       -0.0383
## 35          NOUNfrac.v      0.0360 2.0e-07        0.0360
## 36           sentcount      0.0332 5.9e-07        0.0332
## 37   relativisticexprs      0.0245 1.8e-05       -0.0245
## 38           sentlen.v      0.0209 7.2e-05        0.0209
## 39           wordcount      0.0186 1.8e-04       -0.0186
## 40         verbalNOUNs      0.0170 3.6e-04        0.0170
## 41      caserepcount.v      0.0145 9.4e-04       -0.0145
## 42                 obj      0.0143 1.0e-03       -0.0143
## 43         redundexprs      0.0138 1.3e-03       -0.0138
## 44       extrcaseexprs      0.0137 1.3e-03       -0.0137
## 45             hapaxes      0.0113 3.5e-03        0.0113
## 46              hpoint      0.0108 4.4e-03       -0.0108
## 47       abstractNOUNs      0.0099 6.4e-03        0.0099
## 48       anaphoricrefs      0.0089 9.8e-03        0.0089
## 49      VERBcompdist.m      0.0075 1.7e-02       -0.0075
```

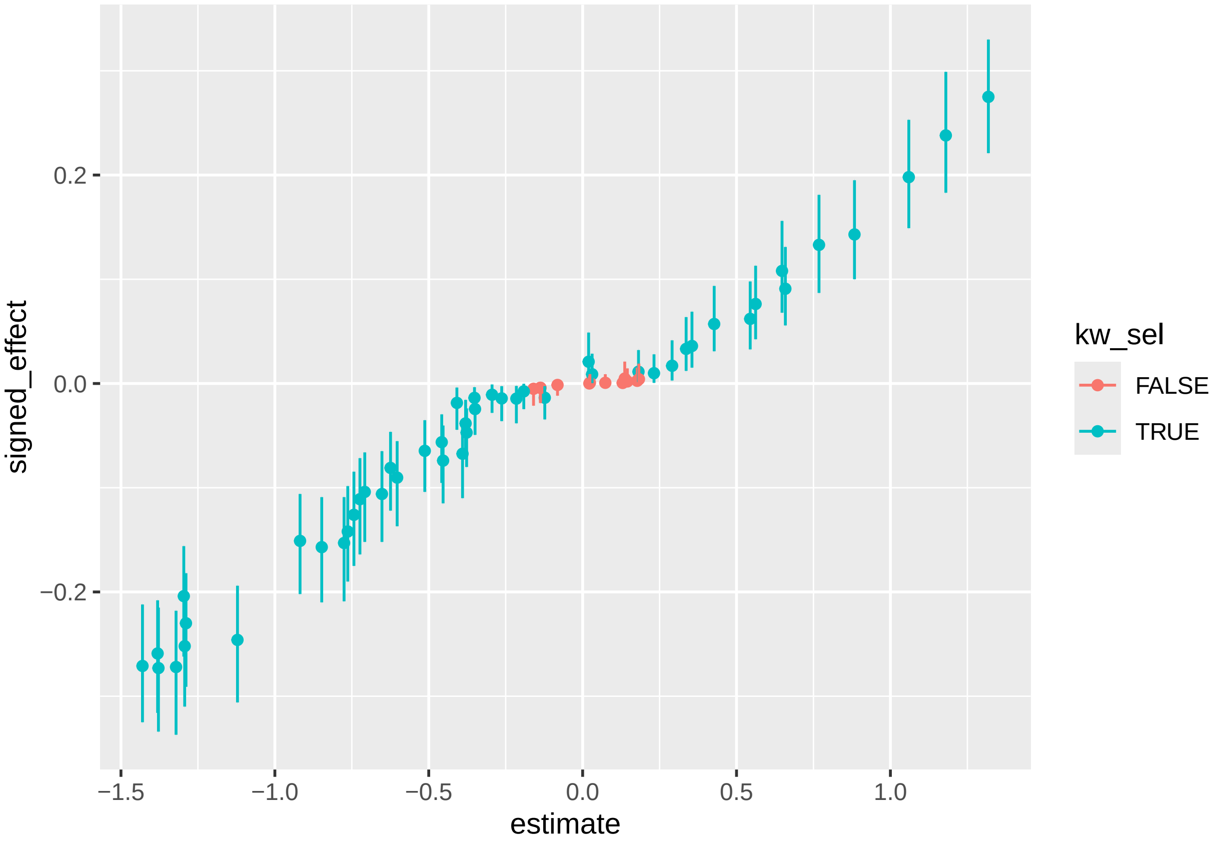
```
featcomp %>%
  filter(kw_sel) %>%
  select(
    Variable,
    kw_chi2,
    kw_p,
    kw_epsilon2_lci,
    kw_epsilon2,
    kw_epsilon2_uci,
    mean_sign
  ) %>%
  arrange(-kw_epsilon2) %>%
  print(n = 100)
```

```
## # A tibble: 49 x 7
##    Variable         kw_chi2     kw_p kw_epsilon2_lci kw_epsilon2 kw_epsilon2_uci
##    <chr>              <dbl>    <dbl>           <dbl>       <dbl>           <dbl>
##  1 activity          207.   6.94e-47        0.221        0.275            0.33
##  2 verbdist          205.   1.70e-46        0.215        0.273            0.334
##  3 sentlen.m         205.   2.17e-46        0.218        0.272            0.337
##  4 ari               204.   3.23e-46        0.212        0.271            0.325
##  5 gf                195.   2.68e-44        0.208        0.259            0.316
##  6 smog              190.   3.42e-43        0.197        0.252            0.31
##  7 NOUNcount.m       185.   3.41e-42        0.194        0.246            0.306
##  8 VERBfrac.m        179.   7.72e-41        0.183        0.238            0.299
##  9 fkgl              173.   1.40e-39        0.182        0.23             0.291
## 10 predorder.m       153.   3.50e-35        0.156        0.204            0.262
## 11 mamr              149.   2.90e-34        0.149        0.198            0.253
## 12 passives          118.   1.87e-27        0.109        0.157            0.21
## 13 predorder.v       115.   7.80e-27        0.109        0.153            0.209
## 14 literary          114.   1.44e-26        0.106        0.151            0.202
## 15 atl               107.   3.57e-25        0.1          0.143            0.195
## 16 predsubjdist.m    107.   5.16e-25        0.0984       0.142            0.19
## 17 compoundVERBs      99.6  1.83e-23        0.0869       0.133            0.181
## 18 NOUNcount.v        94.7  2.18e-22        0.0846       0.126            0.175
## 19 predsubjdist.v     83.6  5.96e-20        0.0716       0.111            0.164
## 20 subj               81.0  2.20e-19        0.0679       0.108            0.156
## 21 maentropy          79.7  4.28e-19        0.0649       0.106            0.152
## 22 predobjdist.m      77.9  1.07e-18        0.0661       0.104            0.152
## 23 VERBcomp           68.4  1.36e-16        0.0557       0.0909           0.131
## 24 mattr              67.9  1.70e-16        0.0553       0.0903           0.137
## 25 NEGcount.m         60.9  5.91e-15        0.0464       0.081            0.122
## 26 fre                57.4  3.55e-14        0.0424       0.0763           0.113
## 27 predobjdist.v      55.7  8.58e-14        0.0403       0.074            0.115
## 28 compoundVERBsdi~   50.7  1.08e-12        0.0387       0.0674           0.11
## 29 compoundVERBsdi~   48.5  3.22e-12        0.0352       0.0646           0.104
## 30 cli                46.6  8.51e-12        0.0327       0.062            0.0979
## 31 NEGfrac.m          42.9  5.68e-11        0.0309       0.0571           0.0936
## 32 entropy            42.4  7.56e-11        0.0296       0.0563           0.0954
## 33 NEGcount.v         35.4  2.62e- 9        0.0239       0.0471           0.0801
## 34 VERBfrac.v         28.8  8.05e- 8        0.0157       0.0383           0.0719
## 35 NOUNfrac.v         27.1  1.95e- 7        0.0152       0.036            0.0689
## 36 sentcount          25.0  5.87e- 7        0.0121       0.0332           0.0637
## 37 relativisticexp~   18.4  1.78e- 5        0.00828      0.0245           0.0493
## 38 sentlen.v          15.8  7.22e- 5        0.00497      0.0209           0.0489
## 39 wordcount          14.0  1.84e- 4        0.00386      0.0186           0.0444
## 40 verbalNOUNs        12.8  3.56e- 4        0.00287      0.017            0.0414
## 41 caserepcount.v     10.9  9.43e- 4        0.00234      0.0145           0.0382
## 42 obj                10.8  1.03e- 3        0.00258      0.0143           0.0361
## 43 redundexprs        10.4  1.29e- 3        0.00351      0.0138           0.028
## 44 extrcaseexprs      10.3  1.34e- 3        0.00258      0.0137           0.0345
## 45 hapaxes             8.53 3.50e- 3        0.00135      0.0113           0.0321
## 46 hpoint              8.12 4.38e- 3        0.000932     0.0108           0.0282
## 47 abstractNOUNs       7.44 6.39e- 3        0.000641     0.00989          0.028
## 48 anaphoricrefs       6.67 9.79e- 3        0.00037      0.00887          0.0286
## 49 VERBcompdist.m      5.67 1.73e- 2        0.000255     0.00754          0.0246
## # i 1 more variable: mean_sign <dbl>
```

```
featcomp %>%
  mutate(signed_effect = kw_epsilon2 * mean_sign) %>%
  ggplot(aes(x = estimate, y = signed_effect, label = Variable)) +
  geom_line(alpha = 0.25) +
  geom_text(aes(color = kw_sel))
```



```
featcomp %>%
  mutate(
    signed_effect = kw_epsilon2 * mean_sign,
    signedlci = kw_epsilon2_lci * mean_sign,
    signeduci = kw_epsilon2_uci * mean_sign
  ) %>%
  ggplot(aes(
    x = estimate, y = signed_effect,
    color = kw_sel, ymin = signedlci, ymax = signeduci
  )) +
  geom_point() +
  geom_errorbar()
```


0.2
0.0
−0.2
signed_effect
−1.5
−1.0
−0.5
0.0
0.5
1.0
estimate
kw_sel
FALSE
TRUE