# Cross-validation in complex surveys

Thomas Lumley
Amaia Iparragirre

2025–11–27

# Cross-validation

With an **exchangeable** set of observations

- Randomly divide data into training and test sets
- Fit to training set
- Evaluate loss in test set
- Repeat so that every observation is in test set the same number of times

Honest estimate of generalisation error for a model-fitting **strategy**

# Complex Surveys

- Strata: generalisation is to same strata
- Clusters: generalisation is between clusters
- Weights: relevant losses are weighted

*The Flainian Pobble Bead is exchangeable only for other Flainian Pobble Beads*

— *HhGttG*

# NHANES

In a two-year period:

- About 15 strata: geographic $\times$ rurality
- Two clusters per stratum: city/county
- Weights: depend on cluster size, stratum size, neighbourhood demographics

# Big problem: weights

If you don't use sampling weights you are optimising the sample predictive accuracy, not the population

Case–control sample: prevalence in sample is 50%

Many machine-learning methods can handle case weights that multiply the loss for each observation

# Smaller problem: clusters, strata

Information leakage

- Stratum information **should** leak from test to training set (same strata in population)
- Cluster information **should not** leak from test to training set (different clusters in population)

A stratum **should** be split between test and training, a cluster **should not** be split

# Replicate weights

Survey statisticians have developed analogues of bootstrap and jackknife for complex samples

These follow the same principles

- Generalisation is to the **same strata**, so each stratum should be in all resamples
- Generalisation is to **different clusters**, so clusters should not be split

# Replicate weights

For reproducibility (and because they don't trust you) survey designs publish sets of resampling weights that everyone uses

$r_{ik}$ is the weight for observation $i$ in resample $k$

- jackknife weights: zero for one cluster, increased for other clusters
- bootstrap weights: 0, 1, 2, 3 times sampling weight
- split-half weights: 0 or 2 times sampling weight
- ...

# Cross-validation

**Test set** : observations with weight (approximately) zero in this replicate

**Training set** : observations with weight not approximately zero in this replicate

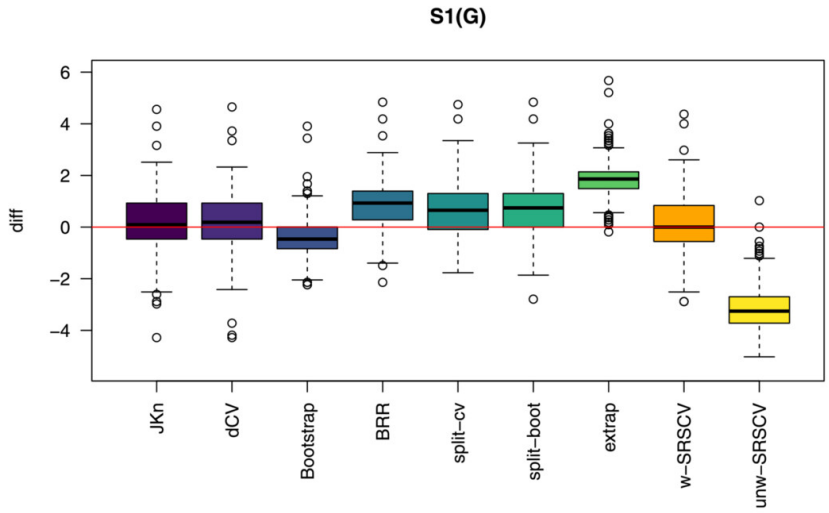
Simple example: cluster jackknife leaves out one cluster at a time

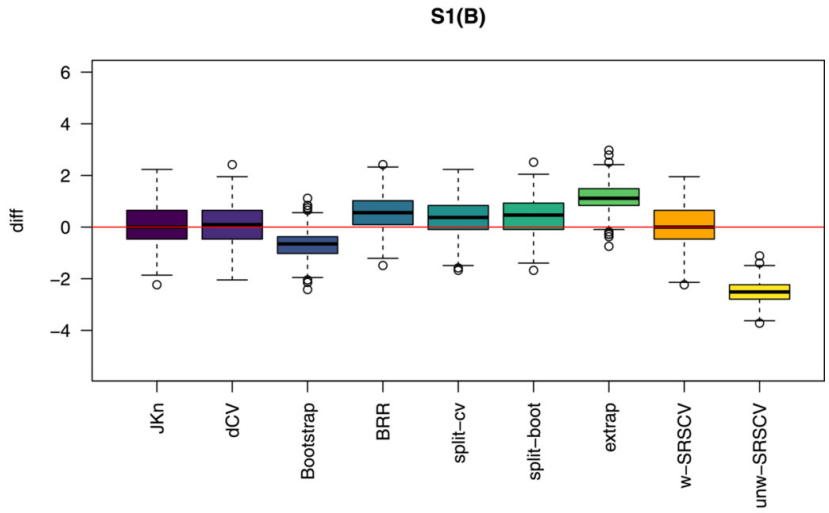
# Simulations

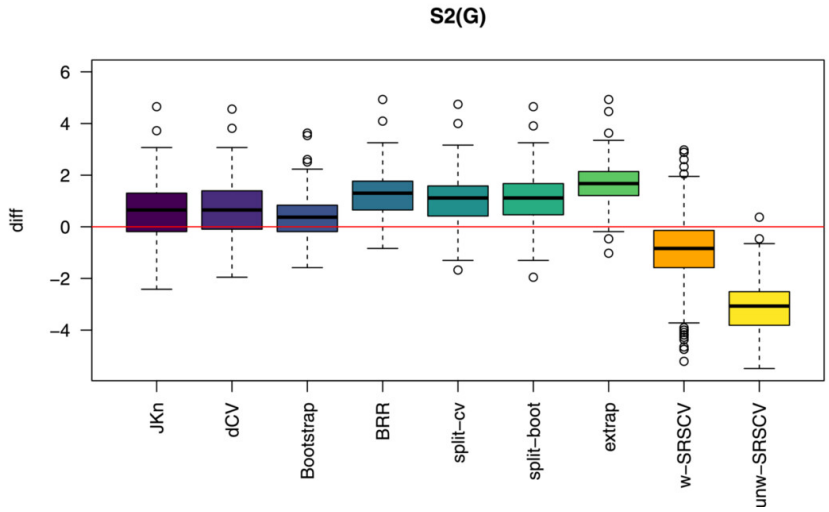
- logistic or normal lasso with 50 observed variables, population size $10^5$
- stratified cluster sample with 5 strata, 4 clusters per sample, total of 330 or 500 units
- six replicate-weight approaches, plus using one stratum as test, plus design-agnostic CV with and without weights
- How close is the penalty to the optimal penalty?
- How close is the number of variables selected to the optimal number?

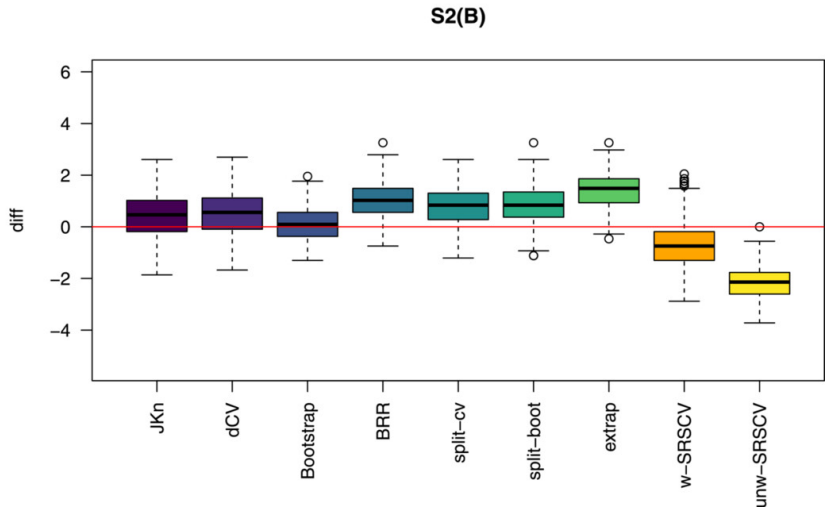
Iparragirre, A., Lumley, T., Barrio, I., & Arostegui, I. (2023). Variable selection with LASSO regression for complex survey data. Stat, 12(1), e578.

# Simulations

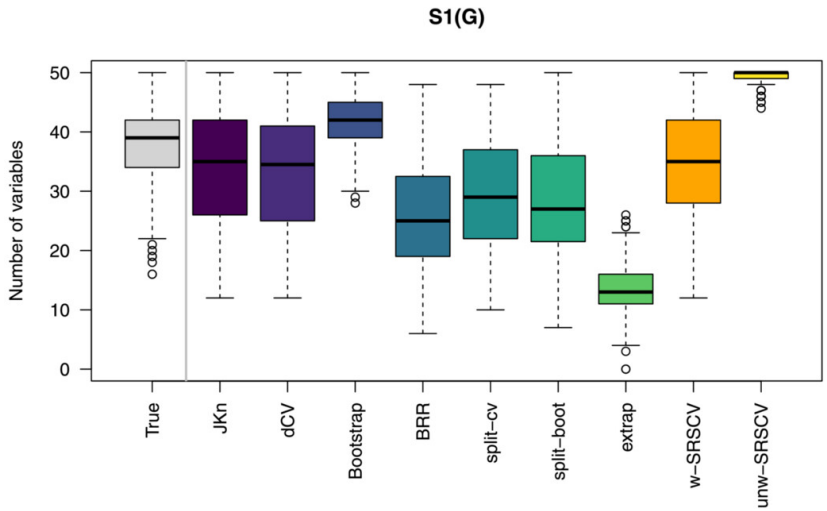


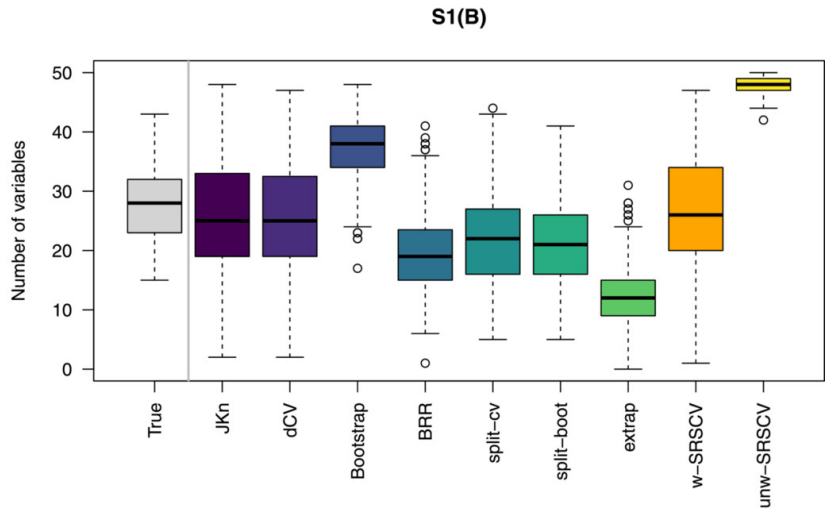



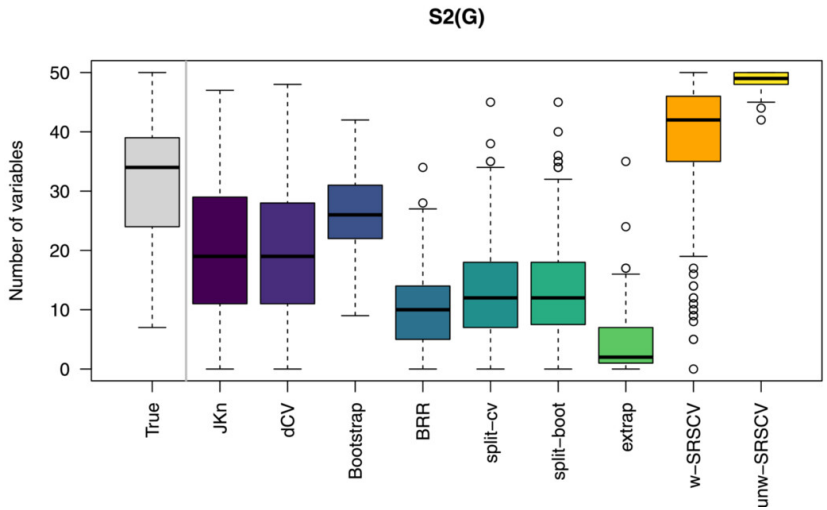



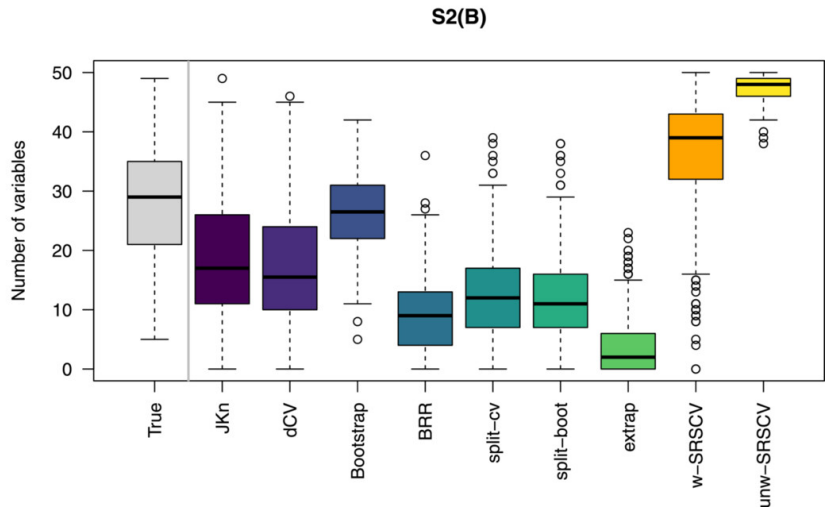



# Simulations



# User interface

```
withCrossval(rclus1,
    api00~api99+ell+stype+mobility+enroll,
    trainfun=ftrain,
    testfun=ftest,
    intercept=FALSE, loss="MSE",
    tuning=0:3)
```

# User interface

```
ftrain=function(X,y,w,tuning) {
  m<-glmnet(X,y,weights=w)
  lambda<-m$lambda[min(which(m$df>=tuning))]
  list(m,lambda)
}
```

```
ftest=function(X, trainfit, tuning){
   predict(trainfit[[1]], newx=X, s=trainfit[[2]])
}
```

# Summary

- Weights matter for cross-validation
- Design matters for cross-validation
- ... a bit
- `survey::withCrossval` coming soon for survey 4.5
- Should basically work for any prediction technique where cross-validation works