SoftwareX 35 (2026) 102917

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

kfre: A Python library for kidney failure risk estimation using the kidney failure risk equation

Leonid Shpaner

University of California Los Angeles, Los Angeles, CA, USA

ARTICLE INFO

Keywords:
Chronic kidney disease
Kidney failure risk
KFRE
Clinical risk prediction
Nephrology
Python

ABSTRACT

Chronic kidney disease (CKD) affects an estimated 11–13% of adults worldwide, and progression to kidney failure requiring dialysis or transplantation carries severe clinical and economic consequences. The Kidney Failure Risk Equation (KFRE), developed by Tangri et al. and validated in over 700,000 individuals across 30 countries, provides individualized 2- and 5-year probability estimates of kidney failure using 4, 6, or 8 routine clinical variables, and is endorsed by the KDIGO 2024 guidelines as a standard tool for estimating absolute kidney failure risk in CKD stages G3–G5. Despite this broad clinical adoption, no dedicated Python library for computing KFRE estimates existed prior to this work. The `kfre` library provides a complete, auditable implementation supporting batch and individual-patient predictions across all three model variants and both time horizons, with calibration for North American and non-North American populations. It includes unit conversion utilities, kidney-failure (ESKD) outcome labeling, CKD stage classification, model evaluation metrics (AUC-ROC, average precision, precision, sensitivity, specificity, and Brier score) with bootstrap confidence intervals, and publication-ready ROC and precision-recall curve plotting. It is intended as a research and validation tool for biostatisticians and data scientists working with patient cohorts, rather than a point-of-care clinical device. The library is available on PyPI at https://pypi.org/project/kfre/ and archived on Zenodo (DOI: 10.5281/zenodo.11100222).

Required metadata

Code metadata (mandatory).

Current code version.

Nr.	Code metadata description	Metadata
C1	Current code version	v0.1.18
C2	Permanent GitHub link to code/repository used for this code version	https://github.com/lshpaner/kfre
C3	Legal Code License	MIT
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	Python (≥ 3.7.4)
C6	Compilation requirements, operating environments & dependencies	numpy ≥ 1.18.5, pandas ≥ 1.0.5, matplotlib ≥ 3.2.2, seaborn ≥ 0.10.1, scikit-learn ≥ 0.23.1, tqdm ≥ 4.0
C7	If available, link to developer documentation/manual	https://lshpaner.github.io/kfre/
C8	Support email for questions	lshpaner@ucla.edu

Software metadata.

Current executable software version

Nr.	(Executable) software metadata description	Metadata
S1	Current software version	v0.1.18
S2	Permanent link to executables of this version	https://pypi.org/project/kfre/
S3	Permanent link to Reproducible Capsule	https://doi.org/10.5281/zenodo.11100222
S4	Legal Software License	MIT
S5	Computing platforms/Operating Systems	OS Independent (Linux, macOS, Microsoft Windows)
S6	Installation requirements & dependencies	Python ≥ 3.7.4; numpy, pandas, matplotlib, seaborn, scikit-learn, tqdm. Install via `pip install kfre`
S7	If available, link to user manual	https://lshpaner.github.io/kfre/

Email address: lshpaner@ucla.edu.

URL: https://lshpaner.github.io/kfre/.

https://doi.org/10.1016/j.softx.2026.102917
Received 13 June 2026; Received in revised form 16 July 2026; Accepted 29 July 2026

1. Motivation and significance

Chronic kidney disease (CKD) is a major global health burden [1], affecting an estimated 11–13% of adults worldwide [2]. For the subset of patients whose disease progresses to kidney failure requiring dialysis or transplantation, the clinical and economic consequences are severe. Identifying high-risk individuals early enough to intervene meaningfully is one of the central challenges of nephrology practice.

The Kidney Failure Risk Equation (KFRE) was developed by Tangri et al. [3] and subsequently validated by Tangri et al. [4] in over 700,000 individuals spanning more than 30 countries. The KFRE provides individualized 2- and 5-year probability estimates of kidney failure using either four, six, or eight routine clinical variables, with calibration for North American and non-North American populations. It is endorsed by the Kidney Disease: Improving Global Outcomes (KDIGO) 2024 clinical practice guidelines [5] as a standard tool for estimating absolute kidney failure risk in patients with CKD stages G3–G5.

Despite this widespread clinical adoption, no dedicated Python library for computing KFRE estimates existed prior to this work. Analysts were required to implement the equations manually from published coefficients, adapt scripts from other languages, or use spreadsheet-based calculators. These approaches are error-prone, difficult to audit, and do not integrate with standard Python data science workflows. General-purpose survival analysis libraries such as `lifelines` [11] exist in Python, but none implement the KFRE specifically, nor provide the KFRE-specific unit conversion, staging, or outcome labeling utilities that real-world clinical datasets require. No equivalent package exists on PyPI.

The `kfre` library [6] fills this gap with a well-documented, tested, and PyPI-distributed implementation that works naturally within pandas-based pipelines. A companion R package, also named `kfre`, is available on CRAN [7] and exposes the same public API (function names and `kfre_{n}var_{t}year` output columns), so that analyses can be expressed in either language.

2. Software description

2.1. Software architecture

`kfre` is designed around two complementary interfaces reflecting distinct workflows in clinical research.

The library is intended for researchers, biostatisticians, and data scientists who compute and validate KFRE estimates across patient cohorts in Python, rather than for point-of-care clinical use. Its purpose is to make the published equation reproducible and auditable within standard data science pipelines: it does not provide clinical decision support, does not store or transmit patient data, and is not a regulated medical device. Point-of-care risk estimation for an individual patient is already served by validated web-based calculators, but such calculators are inherently single-patient tools: they return a risk estimate for one set of inputs and cannot score an entire cohort, cannot be embedded in a reproducible analysis pipeline, and cannot be used to assess how the equation performs against observed outcomes. The complementary need this library addresses is precisely that: applying the Tangri KFRE to an investigator's own patient population at scale and externally validating it against that population's realized outcomes, with discrimination and calibration quantified through the accompanying evaluation utilities. This capability, external validation of a published equation on new data, is what distinguishes a programmatic implementation from a calculator, and it is a prerequisite for the cross-cohort and recalibration studies that sustain confidence in any risk model.

Single-patient queries are served by `kfre_person()`, which accepts individual scalar values and returns a risk probability. This interface is suited for interactive exploration, worked examples, and unit testing of the equation on individual cases.

Batch cohort processing is served by `add_kfre_risk_col()`, which accepts a pandas DataFrame and column name mappings and returns the same DataFrame augmented with one new column per requested model-and-horizon combination, following the pandas idiom of returning an enriched DataFrame.

Output columns follow a consistent naming convention (`kfre_{n}var_{t}year`, e.g., `kfre_4var_2year`), enforced across `add_kfre_risk_col()`, `eval_kfre_metrics()`, and `plot_kfre_metrics()`, so that the output of one function passes directly to the next without renaming. The library builds on the standard scientific Python stack (NumPy, pandas, Matplotlib, seaborn, and scikit-learn) and installs cleanly into existing analysis environments via pip. The underlying KFRE risk model, published coefficients, and the uPCR-to-uACR conversion are detailed in Section 3.1.

2.2. Software functionalities

The library provides the following core functions:

- `kfre_person()`: Single-patient risk prediction from scalar inputs, with North American or non-North American calibration.
- `add_kfre_risk_col()`: Appends KFRE risk columns to a DataFrame for all requested model variants (4, 6, or 8 variables) and time horizons (2 or 5 years).
- `perform_conversions()`: Handles laboratory unit conversions that vary across clinical systems: uPCR (mg/mmol to mg/g), calcium (mmol/L to mg/dL), phosphate (mmol/L to mg/dL), and albumin (g/L to g/dL). These conversions are a common source of error in manual implementations.
- `upcr_uacr()`: Derives uACR from uPCR using the sex-, diabetes-, and hypertension-stratified log-linear regression equations from [8].
- `class_ckd_stages()`: Assigns KDIGO CKD stage labels (G1–G5, with the G3a/G3b split) from eGFR.
- `class_esrd_outcome()`: Constructs binary outcome columns for a chosen time horizon from a user-supplied kidney-failure event indicator and follow-up duration. The KFRE endpoint is kidney failure, defined as the initiation of maintenance dialysis or kidney transplantation (kidney replacement therapy); the function applies a time window to the supplied event flag and does not derive the event from eGFR. An optional `censor_incomplete` argument labels non-event patients with follow-up shorter than the horizon as missing, so that right-censored patients can be excluded from fixed-horizon evaluation.
- `eval_kfre_metrics()`: Computes AUC-ROC, average precision, precision, sensitivity, specificity, and Brier score for each model variant and time horizon. These evaluation utilities let users independently validate KFRE discrimination and calibration on their own cohorts rather than relying solely on the originally published performance figures.
- `bootstrap_metric_ci()`: Computes bootstrap confidence intervals for any of the supported metrics, so that point estimates can be reported with a measure of their precision.
- `plot_kfre_metrics()`: Produces publication-quality ROC and precision-recall curves, with support for overlaying multiple model variants.

For model evaluation, the **Brier score** measures mean squared error between predicted probabilities $\hat{p}_i$ and binary outcomes $y_i \in \{0, 1\}$ over n patients, $\mathrm{BS} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2$; the **area under the ROC curve** (AUC-ROC) measures the probability that a randomly chosen event patient receives a higher predicted risk than a randomly chosen non-event patient, $\mathrm{AUC} = P(\hat{p}_i > \hat{p}_j \mid y_i = 1,\ y_j = 0)$; and **average precision** (AP) summarizes the precision-recall curve as $\mathrm{AP} = \sum_k (R_k - R_{k-1}) \cdot P_k$, which is particularly informative for the imbalanced datasets typical of rare kidney failure events.

2.3. Sample code snippets analysis

The following illustrates the single-patient and cohort prediction interfaces:

```
from kfre import kfre_person, add_kfre_risk_col

# Single-patient query
risk = kfre_person(
    age=57.28, is_male=False,
    eGFR=15.0, uACR=1762.0,
    is_north_american=False, years=2
) * 100  # convert to percentage

print(f"The 2-year risk of kidney failure for this patient
is {risk:.2f}%.")
# Output: The 2-year risk of kidney failure for this patient
is 44.66%.

# Batch cohort processing
df_kfre = add_kfre_risk_col(
    df=cohort, age_col="Age", sex_col="Sex",
    eGFR_col="eGFR", uACR_col="uACR",
    dm_col="Diabetes", htn_col="Hypertension",
    albumin_col="Albumin", phosphorous_col="Phosphorus",
    bicarbonate_col="Bicarbonate", calcium_col="Calcium",
    num_vars=[4, 6, 8], years=(2, 5),
    is_north_american=False, copy=True
)
```

3. Illustrative examples

To demonstrate the full `kfre` workflow, we apply the library to the publicly available CKD cohort from Ali et al. [9], a validation study of the KFRE in advanced CKD patients stratified by renal disease etiology at Salford Royal NHS Foundation Trust, UK. The dataset includes patient age, sex, eGFR, urine protein-to-creatinine ratio (uPCR), serum calcium, phosphate, bicarbonate, and albumin, along with clinical outcomes including ESKD and mortality with follow-up duration in years. All patients have eGFR values in the G4–G5 range (5–29 mL/min/1.73 m^2).

3.1. KFRE risk model

The KFRE models the probability of kidney failure within a given time horizon using a proportional hazards framework [3]. For a patient with covariate vector $\mathbf{x}$, the predicted probability of kidney failure within t years is:

$$P(\text{kidney failure within } t \text{ years} \mid \mathbf{x}) = 1 - S_0(t)^{\exp(\mathbf{x}^\top \boldsymbol{\beta} - c)}$$

where $S_0(t)$ is the baseline survival probability at horizon t, $\boldsymbol{\beta}$ is the vector of published log-hazard coefficients, and c is a centering constant equal to the mean linear predictor in the derivation cohort. The term $\exp(\mathbf{x}^\top \boldsymbol{\beta} - c)$ is the relative hazard for the individual patient relative to the average patient in the derivation sample.

The linear predictor for the 4-variable model is:

$$\eta_4 = \beta_{\text{age}} \cdot \text{age} + \beta_{\text{sex}} \cdot \mathbf{1}[\text{male}] + \beta_{\text{eGFR}} \cdot \text{eGFR} + \beta_{\text{uACR}} \cdot \ln(\text{uACR})$$

The 6-variable model extends this with diabetes and hypertension indicators, and the 8-variable model further adds serum albumin, phosphorus, bicarbonate, and calcium:

$$\eta_6 = \eta_4 + \beta_{\text{dm}} \cdot \mathbf{1}[\text{diabetes}] + \beta_{\text{htn}} \cdot \mathbf{1}[\text{hypertension}]$$

$$\eta_8 = \eta_6 + \beta_{\text{alb}} \cdot \text{albumin} + \beta_{\text{phos}} \cdot \text{phosphorus} + \beta_{\text{bicarb}} \cdot \text{bicarbonate} + \beta_{\text{ca}} \cdot \text{calcium}$$

uACR enters as its natural logarithm, reflecting the log-linear relationship between albuminuria and kidney failure risk observed across the derivation and validation cohorts. The published log-hazard coefficients are taken from the original derivation by Tangri et al. [3]; they are identical across the 2-year and 5-year horizons within each population stratum, and only the baseline survival probability $S_0(t)$ differs by horizon. The North American and non-North American baseline survival values, the latter obtained through the multinational recalibration by Tangri et al. [4], are applied directly within the library.

The non-North American calibration applies a multiplicative shift to the baseline survival while holding $\boldsymbol{\beta}$ constant, so the overall event rate is adjusted while the relative ranking of patients by risk is preserved.

3.2. Unit conversion and uACR derivation

Because the dataset records uPCR rather than uACR, and laboratory values are reported in SI units, we first apply `perform_conversions()` to convert to the units expected by the KFRE, then derive uACR using the sex-, diabetes-, and hypertension-stratified log-linear regression equations from [8]:

$$\widehat{\text{uACR}} = \exp\left(\alpha_k + \beta_k \cdot \ln(\text{uPCR})\right)$$

where k indexes the subgroup defined by sex, diabetes status, and hypertension status. Accounting for these characteristics is preferable to a single fixed ratio because the uPCR-to-uACR relationship varies systematically with patient factors. This conversion is nonetheless an approximation: the albumin fraction of total urinary protein differs between individuals and by the underlying cause of proteinuria, no single conversion has reached consensus, and estimated values therefore carry inherent measurement error. A directly measured uACR should be preferred when available, and this limitation should be reported wherever converted values are used. Accordingly, `upcr_uacr()` documents this caveat and emits a runtime warning whenever it produces estimated values. The conversion is applied via `upcr_uacr()`:

```
from kfre import perform_conversions, upcr_uacr

# Convert SI units: calcium mmol/L -> mg/dL, phosphate
mmol/L -> mg/dL,
# albumin g/L -> g/dL, uPCR mg/mmol -> mg/g
converted_df = perform_conversions(df, reverse=False,
convert_all=True)

# Derive uACR from uPCR using sex-, diabetes-, and
hypertension-stratified
# log-linear regression equations from Sumida et al. (2020)
converted_df["uACR"] = upcr_uacr(
    df=converted_df,
    sex_col="SEX",
    diabetes_col="Diabetes (1=yes; 0=no)",
    hypertension_col="Hypertension (1=yes; 0=no)",
    upcr_col="uPCR_mg_g",
    female_str="Female",
)
```

3.3. Outcome labeling and KFRE prediction

Kidney-failure outcomes at 2- and 5-year horizons are labeled via `class_esrd_outcome()` (which retains the legacy ESRD acronym in its API) from the cohort's recorded dialysis or transplantation event, consistent with the KFRE endpoint of kidney replacement therapy. The non-North American KFRE calibration is applied given the cohort's UK origin. KFRE predictions across all three model variants and both time horizons are then computed via `add_kfre_risk_col()`:

```
from kfre import class_esrd_outcome, add_kfre_risk_col

# Label binary ESRD outcomes
df = class_esrd_outcome(df, col="ESRD", years=2,
```

```
    duration_col="Follow-up YEARS", prefix="esrd",
    create_years_col=False)
df = class_esrd_outcome(df, col="ESRD", years=5,
    duration_col="Follow-up YEARS", prefix="esrd",
    create_years_col=False)

# Merge outcome labels into converted DataFrame and compute
predictions
converted_df = converted_df.merge(df, how="inner")
df = add_kfre_risk_col(
    df=converted_df,
    age_col="Age", sex_col="SEX",
    eGFR_col="eGFR-EPI", uACR_col="uACR",
    dm_col="Diabetes (1=yes; 0=no)",
    htn_col="Hypertension (1=yes; 0=no)",
    albumin_col="Albumin_g_dl",
    phosphorous_col="Phosphate_mg_dl",
    bicarbonate_col="Bicarbonate (mmol/L)",
    calcium_col="Calcium_mg_dl",
    num_vars=[4, 6, 8], years=(2, 5),
    is_north_american=False, copy=False,
)
```

3.4. Results

The merged cohort comprises 743 patients after inner joining on shared identifiers, all with eGFR values in the G4–G5 range. The median age is 68.5 years (IQR 56.9–77.1) and 62.2% are male. The most common specified renal disease etiologies are diabetic nephropathy (24.0%), hypertensive nephropathy (16.8%), glomerulonephritis (11.6%), and ADPKD (8.6%), with the remaining 39.0% classified as other or unspecified causes. The median eGFR is 16 mL/min/1.73 m^2 (IQR 13–18), and median derived uACR is 405 mg/g (IQR 81–1325), reflecting the high-risk, advanced CKD composition of the cohort.

Table 1
Sample KFRE predicted risk estimates for the first five patients from the Ali et al. [9] UK CKD cohort. Values represent probabilities of kidney failure within the specified horizon. uACR derived from uPCR via `upcr_uacr()`.

Age	Sex	eGFR	uACR	4v-2yr	4v-5yr	6v-2yr	6v-5yr	8v-2yr
87.24	Male	19	102.44	0.0672	0.2361	0.0640	0.2221	0.0291
56.88	Female	15	1762.04	0.4495	0.9009	0.4855	0.9197	0.4136
66.53	Female	17	659.14	0.2196	0.6172	0.2400	0.6471	0.1573
69.92	Male	12	1145.25	0.4828	0.9222	0.5209	0.9387	0.5997
81.14	Female	15	980.94	0.2356	0.6467	0.2261	0.6219	0.1475

Table 1 presents predicted risk estimates for the first five patients, demonstrating the output structure of `add_kfre_risk_col()` applied to real-world clinical data. The second patient (female, age 56.88, eGFR 15, uACR 1762.04) carries a predicted 5-year kidney failure risk of 90.1% under the 4-variable model, illustrating the clinical severity captured by the KFRE in advanced CKD.

As a first check of face validity, Fig. 1 shows predicted KFRE risk stratified by CKD stage. Predicted risk increases monotonically with advancing stage, with G5 patients carrying markedly higher risk than those in stage G4, consistent with the expected clinical gradient and confirming that the implementation reproduces the intended behavior of the equation on real cohort data.

Fig. 2 shows the correlation between 4-variable and 8-variable 2-year risk estimates across the full cohort. The two models are closely correlated, with systematic divergence at higher risk levels reflecting the additional prognostic information contributed by albumin, phosphorus, bicarbonate, and calcium in the extended model. For most patients in the low-to-moderate risk range, the 4-variable model is a sufficient approximation; for patients with more complex metabolic profiles, the 8-variable model provides added discrimination.

Fig. 3 reports discrimination and calibration for each model variant and time horizon, computed via `eval_kfre_metrics()` and rendered via `plot_kfre_metrics()`. The panels present ROC and precision-recall curves alongside the corresponding AUC-ROC and average precision summaries, with the precision-recall view being the more informative of the two given the low prevalence of kidney failure events in the cohort.

Table 2 reports the corresponding point estimates for all six metrics with 95% bootstrap confidence intervals (1000 resamples), computed via `bootstrap_metric_ci()`. Reporting interval estimates alongside point estimates conveys the precision of each metric given the cohort size and the low prevalence of kidney failure events.

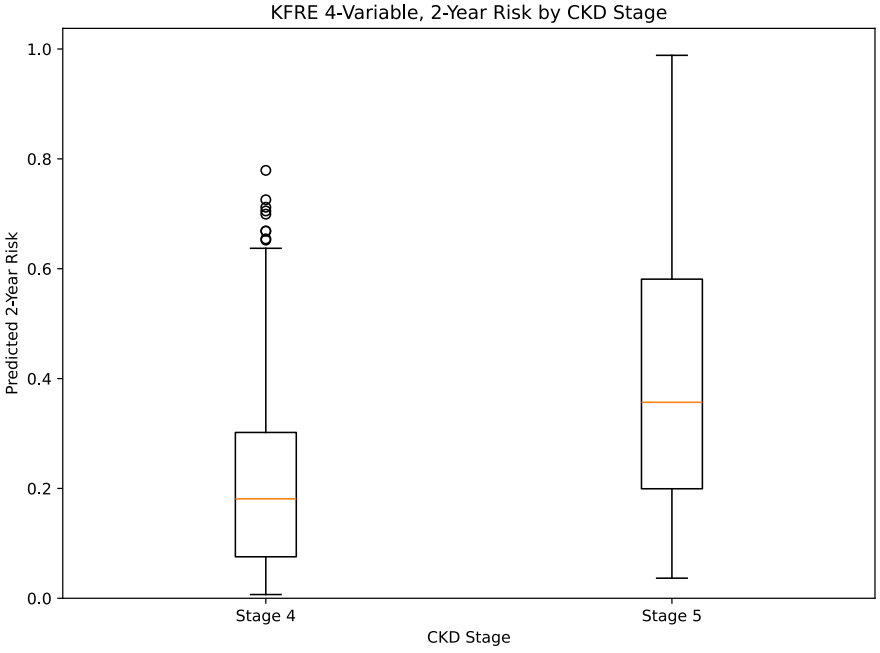

Fig. 1. Predicted KFRE risk by CKD stage in the UK CKD cohort of Ali et al. [9]. Risk rises monotonically with advancing stage, providing face validity for the implementation.

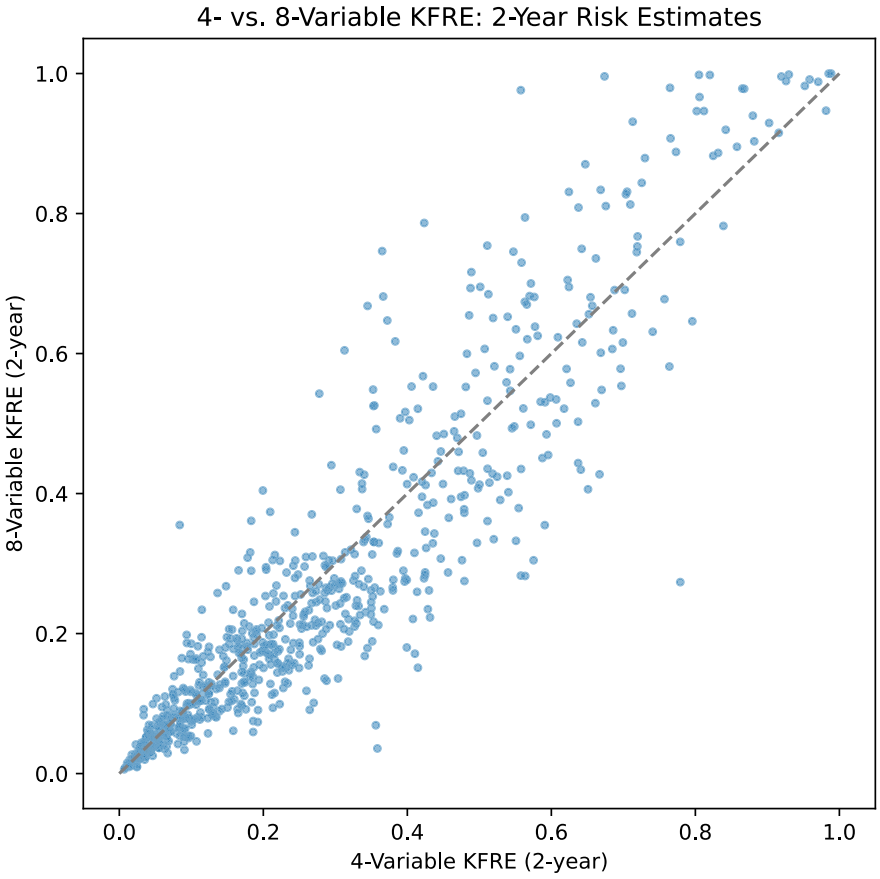

Fig. 2. 4-variable vs. 8-variable 2-year KFRE risk estimates in the UK CKD cohort of Ali et al. [9]. The dashed line indicates perfect agreement. Divergence at higher risk levels reflects the contribution of the extended laboratory variables.

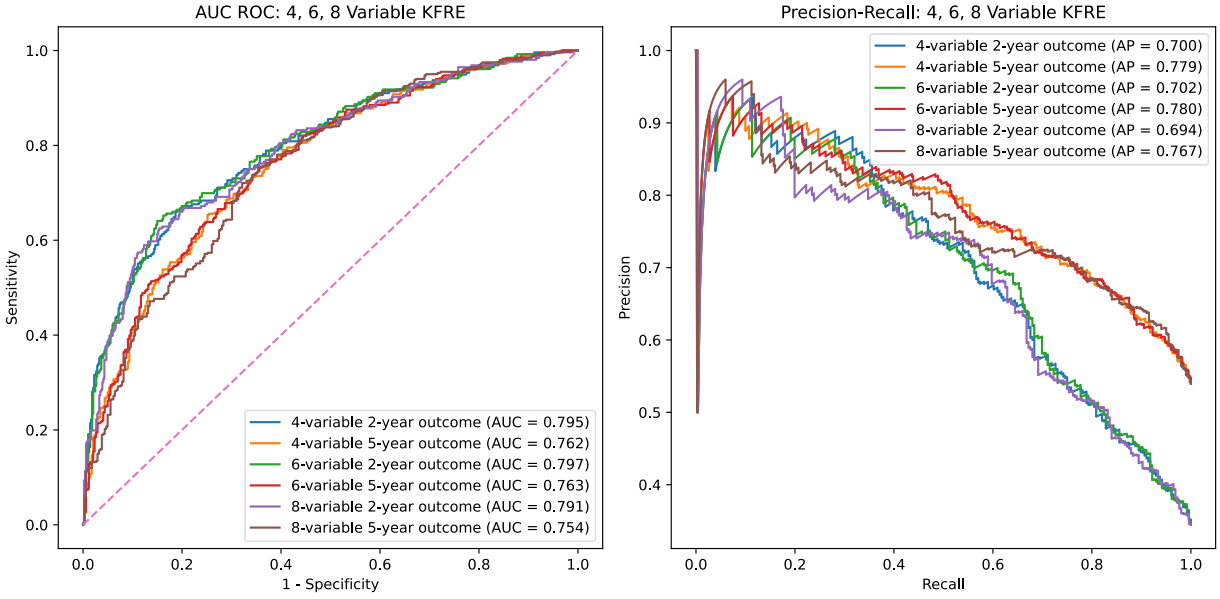

Fig. 3. KFRE model performance on the UK CKD cohort of Ali et al. [9]: ROC and precision-recall curves with AUC-ROC and average precision for the 4-, 6-, and 8-variable models across both time horizons.

The outcome labels above use a fixed-horizon rule in which every patient without a recorded event within the horizon is treated as a non-event. Because the cohort has substantial early mortality and incomplete follow-up, this rule classifies some right-censored patients (who died or were lost to follow-up before the horizon) as non-events. In this cohort, 104 patients (14%) at the 2-year horizon and 224 (30%) at the 5-year horizon had follow-up shorter than the horizon without a recorded event. To assess the sensitivity of the results to this labeling,

Table 2
KFRE discrimination and calibration metrics with 95% bootstrap confidence intervals (1000 resamples, seed 42) on the UK CKD cohort of Ali et al. [9], computed via `eval_kfre_metrics()` and `bootstrap_metric_ci()`. Point estimates for AUC-ROC and average precision correspond to the values shown in Fig. 3. Columns denote model variant (4/6/8 variables) and time horizon (2/5 years). Precision/PPV, sensitivity, and specificity are computed at a predicted-probability threshold of 0.5.

Metric	4v-2yr	4v-5yr	6v-2yr	6v-5yr	8v-2yr	8v-5yr
Precision/PPV	0.782 (0.716–0.854)	0.676 (0.634–0.718)	0.788 (0.725–0.857)	0.684 (0.643–0.727)	0.792 (0.721–0.862)	0.677 (0.637–0.718)
Average Precision	0.700 (0.638–0.759)	0.779 (0.734–0.823)	0.702 (0.640–0.761)	0.780 (0.735–0.825)	0.694 (0.634–0.750)	0.767 (0.722–0.812)
Sensitivity	0.406 (0.347–0.466)	0.810 (0.771–0.846)	0.422 (0.364–0.482)	0.820 (0.782–0.857)	0.402 (0.341–0.462)	0.810 (0.771–0.845)
Specificity	0.940 (0.920–0.961)	0.544 (0.490–0.595)	0.940 (0.920–0.961)	0.556 (0.506–0.611)	0.945 (0.924–0.965)	0.547 (0.493–0.599)
AUC-ROC	0.795 (0.760–0.829)	0.762 (0.728–0.795)	0.797 (0.763–0.830)	0.763 (0.730–0.796)	0.791 (0.756–0.824)	0.754 (0.720–0.788)
Brier Score	0.170 (0.155–0.186)	0.207 (0.190–0.225)	0.169 (0.154–0.184)	0.207 (0.190–0.226)	0.173 (0.157–0.191)	0.210 (0.192–0.228)

we re-evaluated performance after excluding these right-censored patients using the `censor_incomplete` option of `class_esrd_outcome()`. Discrimination at 2 years was essentially unchanged (for example, 4-variable AUC-ROC 0.795 vs. 0.808), while at 5 years it improved (4-variable AUC-ROC 0.762 vs. 0.836; average precision 0.779 vs. 0.934), indicating that the primary, fixed-horizon estimates are conservative rather than optimistic. We report the fixed-horizon results as primary because they use the complete observed cohort; complete-case exclusion is a pragmatic guard rather than a full competing-risks analysis, and the KFRE itself was derived using censoring-aware Cox models [3,4].

4. Impact

`kfre` is the first dedicated Python library on PyPI implementing all three KFRE model variants (4-, 6-, and 8-variable) across both time horizons, with integrated utilities for laboratory unit conversion, uPCR-to-uACR derivation, CKD staging, ESKD outcome labeling, model evaluation, and visualization. It enables nephrology researchers, biostatisticians, and data scientists working in Python to compute and validate a guideline-endorsed risk prediction tool within standard analytical workflows, without manual coefficient transcription or ad hoc scripts. The `perform_conversions()` and `upcr_uacr()` utilities specifically address common data quality barriers in real-world datasets, where laboratory values are reported in heterogeneous units and direct uACR measurement is frequently unavailable.

The library reproduces the KFRE as established and externally validated in peer-reviewed studies, including validation in advanced CKD cohorts stratified by disease etiology [9] and in an independent Asian population [10], and it is benchmarked here against the publicly available cohort from one such study [9]. It has been maintained through successive versioned releases since its initial 2024 publication and is archived with a permanent DOI on Zenodo [6].

New research questions enabled by `kfre` include systematic cross-cohort comparisons of KFRE performance using a standardized implementation, evaluation of the 4- versus 8-variable model trade-off in resource-limited settings, and integration of KFRE predictions into broader clinical risk prediction pipelines alongside other biomarker-based models.

5. Conclusions

`kfre` fills a practical and significant gap in the Python ecosystem for clinical kidney disease research. By providing a consistent, auditable implementation of the KFRE alongside utilities for unit conversion, staging, outcome labeling, and model evaluation, the library consolidates a complete analytical workflow into a single, cohesive tool. The two-interface architecture serves both interactive single-patient queries and vectorized cohort processing. Future extensions include support for updated coefficient sets as they are published and additional survival-analysis-aware outcome handling for evaluation on cohorts with substantial censoring.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author acknowledges Navdeep Tangri and colleagues for developing and validating the Kidney Failure Risk Equation. Special thanks to Panayiotis Petousis, PhD, Obidiugwu Duru, MD, MS, Kenn B. Daratha, PhD, Keith C. Norris, MD, PhD, Katherine R. Tuttle, MD, FASN, FACP, FNKF, Susanne B. Nicholas, MD, MPH, PhD, and Alex Bui, PhD. Their exceptional work on end-stage kidney disease has greatly inspired the creation of the `kfre` Python library.

References

[1] Bikbov B, et al. Global, regional, and national burden of chronic kidney disease, 1990–2017: a systematic analysis for the global burden of disease study 2017. Lancet 2020;395(10225):709–33. https://doi.org/10.1016/S0140-6736(20)30045-3
[2] Hill NR, Fatoba ST, Oke JL, Hirst JA, O'Callaghan CA, Lasserson DS, Hobbs FDR. Global prevalence of chronic kidney disease: a systematic review and meta-analysis. PLoS One 2016;11(7):e0158765. https://doi.org/10.1371/journal.pone.0158765
[3] Tangri N, Stevens LA, Griffith J, Tighiouart H, Djurdjev O, Naimark D, Levin A, Levey AS. A predictive model for progression of chronic kidney disease to kidney failure. JAMA 2011;305(15):1553–9. https://doi.org/10.1001/jama.2011.451
[4] Tangri N, et al. Multinational assessment of accuracy of equations for predicting risk of kidney failure: a meta-analysis. JAMA 2016;315(2):164–74. https://doi.org/10.1001/jama.2015.18202
[5] Kidney disease: improving global outcomes (KDIGO) CKD work group, KDIGO 2024 clinical practice guideline for the evaluation and management of chronic kidney disease. Kidney Int 2024;105(4S):S117–S314.
[6] Shpaner L. kfre: a Python library for reproducing kidney failure risk equations (KFRE). Zenodo 2024. Version 0.1.18. https://doi.org/10.5281/zenodo.11100222
[7] Shpaner L. kfre: kidney failure risk equation (KFRE) tools, R package, comprehensive R archive network (CRAN); 2025. https://CRAN.R-project.org/package=kfre.
[8] Sumida K, et al. Conversion of urine protein-creatinine ratio or urine dipstick protein to urine albumin-creatinine ratio for use in chronic kidney disease screening and prognosis. Ann Intern Med 2020;173(6):426–35. https://doi.org/10.7326/M20-0529
[9] Ali I, Donne RL, Kalra PA. A validation study of the kidney failure risk equation in advanced chronic kidney disease according to disease aetiology with evaluation of discrimination, calibration and clinical utility. BMC Nephrology 2021;22(1):194. https://doi.org/10.1186/s12882-021-02402-1
[10] Kang MW, Tangri N, Kim YC, An JN, Lee J, Li L, Oh YK, Kim DK, Joo KW, Kim YS, Lim CS, Lee JP. An independent validation of the kidney failure risk equation in an asian population. Sci Rep 2020;10:12920. https://doi.org/10.1038/s41598-020-69715-3
[11] Davidson-Pilon C. lifelines: survival analysis in Python. J Open Source Softw 2019;4(40):1317. https://doi.org/10.21105/joss.01317