

**ANALYSIS OF KIDNEY VOLUME AND FUNCTIONAL OUTCOMES USING
SURVIVAL AND CLASSIFICATION TREE MODELS**

by

Xiaotian Gao

B.S, Biotechnology, Dalian Medical University, China, 2012

Submitted to the Graduate Faculty of
Graduate School of Public Health in partial fulfillment
of the requirements for the degree of
Master of Science

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

Graduate School of Public Health

This thesis was presented

by

Xiaotian Gao

It was defended on

June 3, 2015

and approved by

Stewart J. Anderson, PhD, Professor of Biostatistics, Graduate School of Public Health,
University of Pittsburgh

Ada O. Youk, PhD, Associate Professor of Biostatistics, Epidemiology and Clinical and
Translational Science, Graduate School of Public Health, University of Pittsburgh

Thesis Director: Douglas Landsittel, PhD, Professor of Medicine, Biostatistics and Clinical
and Translational Science, School of Medicine, University of Pittsburgh

Copyright © by Xiaotian Gao

2015

**ANALYSIS OF KIDNEY VOLUME AND FUNCTIONAL OUTCOMES USING
SURVIVAL AND CLASSIFICATION TREE MODELS**

Xiaotian Gao

University of Pittsburgh, 2015

ABSTRACT

Tree models have been widely used for clustering problems in areas like evidence-based decision-making, machine learning and data mining. The inherent properties of tree models, e.g. recursively dividing sample space, make it more flexible and superior in situation of nonlinear classifications and complex sample structure. In this thesis, they would be applied to the data from the Consortium for Radiologic Imaging Studies of Chronic Kidney Disease (CRISP) to explore the association between Total Kidney Volume (TKV) and Chronic Kidney Disease (CKD) stage 3.

In this current study, multivariable Cox survival models were used to adjust for baseline confounders and assess the relationship between TKV and time to CKD Stage 3. The same questions, were also analyzed using survival tree models, identifying the combination of variables associated with similar survival, and thus facilitating the identification of high and low risk sets. Variations of the tree modeling approach were employed to maximize model fit and generalizability, including pruning and bagging.

Classification tree models, and the same variations (pruning and bragging) were also fit to the development of the dichotomous outcome of CKD Stage 3 by a fixed time point. Receiver operator characteristic (ROC) curves with and without cross-validation are presented, and associated classification statistics (sensitivity, specificity and area under the curve) are calculated

to characterize the prognostic ability of the tree models. Findings are then compared to the standard logistic model.

Both tree models and regression models agreed on the significance of baseline total kidney volume and estimated glomerular filtration rate in predicting CKD prognosis. Cutoff values were also determined.

From the public health significance perspective, these cutoffs could be advisory to actual clinical decision and prognostics of CKD. Comparing with current continuously GFR monitoring for CKD progress, two baseline predictors measured in the early phase of the disease, makes early interventions more practical.

TABLE OF CONTENTS

PREFACE	X
1.0 INTRODUCTION	1
2.0 DATA SET AND METHODS	3
2.1 DATASET	3
2.2 MULTIVARIABLE LOGISTIC REGRESSION	4
2.3 COX PROPORTIONAL HAZARD MODEL	5
2.4 CLASSIFICATION AND REGRESSION TREES	5
2.4.1 Classification Trees	5
2.4.1.1 Splitting Criterion and Impurity measurement	6
2.4.1.2 Risk Function and Class assignment	7
2.4.1.3 Missing data and Surrogate Variables	8
2.4.1.4 Pruning and Optimal trees	8
2.4.2 Survival Trees	10
2.4.3 Bagging Trees and Importance of Variables	10
2.5 ROC CURVES BRIER'S SCORE AND INDEX C	13
3.0 RESULTS	15
3.1 ANALYSIS OF BINARY OUTCOMES FOR CHRONIC KIDNEY DISEASE	15

3.1.1	Multivariable Logistic Regression.....	16
3.1.2	Classification Trees.....	17
3.2	ANALYSIS OF TIME TO CHRONIC KIDNEY DISEASE.....	22
3.2.1	Multivariable Cox Model.....	22
3.2.2	Survival Tree.....	23
4.0	DISCUSSION.....	28
	APPENDIX R CODE.....	29
	BIBLIOGRAPHY.....	41

LIST OF TABLES

Table 1. Coefficients of full logistic regression.....	16
Table 2. Coefficients of stepwise logistic regression.....	16
Table 3. Importance of variables	20
Table 4. Area under ROC (AUC)	22
Table 5. Coefficients of full Cox regression model.....	22
Table 6. Coefficients of Cox regression	23

LIST OF FIGURES

Figure 1 ROC Curves for logistic regression models	17
Figure 2. The model without pruning classifying CKD	18
Figure 3. Cross-validation relative error for each complex parameter value	19
Figure 4. Pruned tree for complete tree for CKD categorization.....	19
Figure 5. ROC curves for classifying CKD	21
Figure 6. Complete grown tree for CKD survival	24
Figure 7. Cross-validated relative error for each complex parameter	25
Figure 8. Optimized tree for CKD survival time	25
Figure 9. Variable importance for bagged CKD survival tree.....	26
Figure 10. KM curves by eGFR and htTKV groups.....	27

PREFACE

I would like to express my sincere and genuine gratitude to Professor Douglas Landsittel, my academic advisor, who introduced the topic of my thesis and has always been encouraging and illuminating me through my whole study and research. Without his insights and guidance, finishing this thesis would be impossible. My gratitude would also go to Professor Stewart J. Anderson and Professor Ada O. Youk not only for sitting on my thesis committee but most importantly for sharing their valuable insights and inspiring comments on this thesis.

I would also like to thank Dr. Sally C. Morton, who has kindly mentored me since I first came to Pittsburgh.

Last but not least, I would like to thank my family, especially my parents Gao, Lan and Dong, Jing for supporting my back both financially and mentally while I worked on my degree. They have shared both my joy and frustration during the past two years. If any glory, it should be dedicated to them as well.

1.0 INTRODUCTION

Polycystic Kidney Disease (PKD) is a kidney disease induced by the multiplication of non-replaceable cystic renal cells. According to NIH most recent data and Grantham et. al's paper¹, PKD is a very severe and costly kidney disease: In the United States, approximately 600,000 people suffer from PKD, and it is the 4th leading cause to end stage renal failure.

Current clinical diagnostic standards for CKD are based measurement of glomerular filtration rate (GFR). $GFR \geq 90 \text{ mL/min}$ suggests healthy kidney or stage 1 chronic kidney disease (CKD) damage with high GFR. Stage 2 CKD is categorized by GFR between 60 mL/min and less than 90 mL/min, while stage 3 and above CKD, is categorized by GFR below 60 mL/min. Patients with GFR dropping below 15 mL/min most likely have renal failure, and have to either transplant another kidney or be on dialysis. As GFR is not most convenient and often impractical to measure on regular basis, estimated GFR⁴ (eGFR), using the CKD-EP1 equations, based on regular blood sample are presented.

Kidney filtration dysfunction by PKD stems from the gradual growth and multiplication of cystic cells, so GFR often only drops rapidly at a relative late age of the disease. This latent result narrows the valuable time window for intervention most effective at early age of the disease, necessitating developing of other measures, such as imaging and serum biomarkers.

Previous study from CRISP suggested potential connection between height adjusted total kidney volume (htTKV) and CKD. Gratham et al² (2006).suggested that higher rates of kidney

enlargement are associated with more rapid decrease in kidney function. Chapman et al³ (2012) established the correlation between baseline htTKV >600cc/m and higher risk of CKD. However, survival of CKD has not been analyzed and modern regression techniques have not yet been implemented. In this thesis, we use tree models to analyze both CKD binary outcomes and CKD free survival. We compared the results from tree models to regular multivariable logistic and Cox regression models. Pruned trees by cross-validation and bagged trees were also fit, and then compared to the standard methods. Results led to a relatively simple and accurate tree model with only baseline eGFR and htTKV for both outcomes. Results of the tree models complement standard methods, and yield an easily interpretable model, which should be validated in future studies.

2.0 DATA SET AND METHODS

2.1 DATASET

All variables were measured at baseline, except eGFR, which is used to categorize CKD stage. For simplicity, clinical variables are coded as follows through this thesis:

Estimated glomerular filtration rate (eGFR)⁴ (1999) , based on the CKD-EP1 equations is used as to categorize Chronic kidney disease(CKD) stage. Individuals with estimated glomerular filtration rate less than 60 mg/dL are categorized as CKD stage 3. Again for simplicity, we will label individuals with CKD stage 3 or higher as having CKD, while others are labeled as controls. The number of years since baseline CRISP I visit was used as the timeline in the survival analysis.

Other clinical variables and biomarkers were also analyzed. These included: estimated glomerular filtration rate measured at baseline (eGFR), height adjusted total kidney volume (htTKV), urinary monocyte chemo attractant protein (MCP), and blood Urea Nitrogen (BUN).

Demographic included gender (male and female), age, body mass index (BMI) and race. Race is categorized into 2 classes, namely, Caucasian/others and African American, and others. Caucasian/others was coded as baseline.

For Survival analyses, only those without any follow-up visits were excluded. For the binary outcome of CKD, we also excluded subjects who did not reach CKD and did not have at least 10 years of follow-up.

2.2 MULTIVARIABLE LOGISTIC REGRESSION

Multivariable logistic regression was used to predict the binary outcome of whether a patient reaches CKD stage 3 during the study (subsequently referred to as just 'CKD'). The multivariable logistic model was defined as:

$$\ln\left(\frac{p}{1-p}\right) = X\beta$$

where p is the probability of CKD, with covariate matrix X and parameter vector β . Wald's test was used to test the significance of variables in the model. Variable selection was based on forward stepwise selection under Akaike Information Criterion (AIC). The selection process could be summarized as following:

1. Start with no variables in the model.
2. Adding each individual variable not in the model, and select the model with lowest AIC less than the previous model in 1.
3. Adding until no new predictors could be added.

2.3 COX PROPORTIONAL HAZARD MODEL

The Cox proportional hazard model was used to predict CKD-free survival. The model was defined as:

$$h(t|X) = h_o(t)exp(X\beta)$$

where $h_o(t)$ is the baseline hazard function with covariate matrix X and parameter vector β .

Wald's test was used to test the significance of variables in the model. Variable selection was based on forward stepwise selection under AIC based on pseudolikelihood.

2.4 CLASSIFICATION AND REGRESSION TREES

2.4.1 Classification Trees

In defining the classification tree models, we used the following notation:

π_i $i = 1, 2, \dots, C$ Prior Probability of each classes

$L(i, j)$ $i, j = 1, 2, \dots, C$ Loss matrix for misclassifying class i as class j

A Some node in the tree

$P(A)$ Probability for future observations be classified in the node A , $P(A_L), P(A_R)$ denote the left and right node son under parent A

$I(A)$ Impurity measurement of node A .

$N_i(A)$ Number of observations of class i in node A

N_i Number of observations of class i in the whole learning dataset

$R(A)$ Risk of node A

2.4.1.1 Splitting Criterion and Impurity measurement

Consider the typical classification case, where we have C classes. The classification tree is grown under the splitting criterion that minimizes the impurity of the nodes in the tree. To be able to do that, impurity measurement functions f were introduced. $f(p_{iA})$ represents the impurity in the node A caused by class i . Intuitively, and most commonly, we need the node with p_{iA} , estimated by the frequency of class i in node A, to be as far from $1/C$ as possible. For instance, consider the simple two-class case, the worst scenario (most impure node) would be a node with a proportion of 0.5 in each class.

There are two common function forms for f , are the Gini Index and Information Index.

Gini index is defined by:

$$p_{iA} = p(1 - p)$$

Information index is defined by:

$$p_{iA} = -p \log(p)$$

To summarize the total impurity of node A, we sum the impurity measurement of each class in node A.

$$I(A) = \sum_{i=1}^C f(p_{iA})$$

An alternative way was called “twoing”, where the node has the minimum impurity if it could be partitioned into two sub-classes with least impurity. For two-class case, both $I(A)$ gives the same result. For all subsequent analysis, we will use Gini index in this thesis.

All possible splitter variables with all possible splitting values are first calculated for the node A. The best splitter is selected, so that the average impurity reduction by two son nodes is maximized.

$$\Delta I(A) := p(A)I(A) - p(A_L)I(A_L) - p(A_R)I(A_R)$$

$$\{A_L, A_R\} = \arg \max \Delta I(A)$$

The branches of the tree will continue splitting until either of the two following condition is met:

- a. The number of the observations in terminal node reaches the minimum predefined (20 in our cases).
- b. All the observations in the terminal node have same value for every predictor.

2.4.1.2 Risk Function and Class assignment

The next step is to assign a predicting class for each of the nodes in the defined tree. The node A would be labeled by the class i, so that the average risk in the node A will be minimized for future observations comparing with all other possible assignments. This process ensures the development of the tree with minimum expected decision cost to a dataset similar to the learning dataset, in the sense of the prior probability of each class identical. Formal criterion is given as:

Node A is labeled as class i, if

$$\frac{L(i, j)\pi_i N_i(A)}{L(j, i)\pi_j N_j(A)} > \frac{N_i}{N_j}, \forall j \neq i$$

The criteria above could also be written as:

Node A is labeled as class i, if

$$L(i, j)\pi_i \frac{N_i(A)}{N_i} > L(j, i)\pi_j \frac{N_j(A)}{N_j}, \forall j \neq i$$

where $\frac{N_i(A)}{N_i}$ is an estimate of the probability of future observations to be classified in the node A, given the true class of the observation is i. Then, multiplying the prior probability of class i, $\pi_i N_i(A)/N_i$ provides estimate the probability of future class i observations to be classified in the node A. As a result, the criteria restrain the assignment with minimum risk for every node in the tree. However, it worth noticing that the greedy algorithm (as described in section 2.4.1.1) only takes into account single node respectively while minimizing, and fails to balance the nodes across the whole tree for global minimum cost.

2.4.1.3 Missing data and Surrogate Variables

For each node, the primary splitting variable splits the node to achieve maximum purity. However, the primary splitting variable might be missing for some observations. In this case, surrogate variables are sought. Surrogate variables are the variables whose pattern is most similar to the primary splitting variable in the regard of predicting the outcome. They are determined by applying tree model to previous primary splitter as classification outcome. The procedure is done iteratively until the achievement of a surrogate splitter with no missing values. This approach provides a relative robust estimate comparing with regular regression methods.

2.4.1.4 Pruning and Optimal trees

As mentioned in section 2.4.1.1, the tree is grown until the stopping criteria are reached (section 2.4.1.1). This generally leads to over-fitting of the tree, and may not be able to be generalized to

other datasets. To produce a more generalizable tree, the process of pruning and an associated complex parameter (cp) is introduced.

Let T_1, T_2, \dots, T_k denote any sub-trees of the already-defined un-pruned tree T_0 . $\|T\|$ is the number of terminal nodes of tree T . Recall that the risk of the tree T was defined previously as the summation of all the risk of terminal nodes in tree T :

$$R(T) = \sum_{i=1}^m p(A_i)R(A_i)$$

Note $R(T)$ does not penalize for the value of $\|T\|$. When pruning the tree, new risk $R_{cp}(T)$ is introduced:

$$R_{cp}(T) := R(T) + cp\|T\|R(T_0) \quad cp > 0$$

Note that $R_{cp}(T)$ penalize for both the misclassification risk $R(T)$ and the tree size $\|T\|$ by complex parameter cp . Intuitively, the pruning process could be interpreted as following: Starting from the last level of the tree, the terminal nodes survive pruning only if the change of misclassification risk is higher than α times of the change of tree complexity measured by $\|T\|$.

Neither the estimate of the tree itself, nor the pruning process explained so far provides an accurate and stable estimation of the real case. This is due to the fact that both are based only on one learning dataset. To exploit the full variability of the learning dataset, cross-validation is introduced.

The total dataset is randomly divided into K parts. One of the partitions is selected as validation dataset, while the rest combined was used as learning dataset. Thus, there are K trees fitted. Pruning each of the K trees, K sets of pruning sub-trees are generated. Match pruned sub-trees across sets by number of terminals left ($\|T\|$). The average performance of the matched

trees with same number of terminal nodes left provides much more stable estimation of the original tree of same size. In our case, we implemented leave-one-out cross-validation where $K = N$. One standard error (1-SE) rule gives the optimized cp value. Namely: First, pick up the cp value corresponding to the lowest risk. Secondly, among all the sub-trees, whose cp is within one standard error of the original cp picked, pick the simplest tree.

2.4.2 Survival Trees

For regression trees, splitting criterion are the selection criterion for different regression models. For example, for regular linear regression model, the studentized residuals for model corresponding to split tree and un-split tree were calculated and compared to determine if the node should be split. For survival trees, logrank test statistics and likelihood ratio statistics are used for tree building and splitting. Details could be found in Ciampi^{5,6} et. al's paper(1986,1987).

2.4.3 Bagging Trees and Importance of Variables

The method of bagging (bootstrapping) trees was first introduced by Breiman^{7,8,9}(1996,1998) to stabilize the relatively unstable CART algorithm. Bragging differs when dealing with different outcomes, such as regression, classification and survival outcomes. For simplicity and illustration purpose, we introduce bragging with linear regression outcome, and then revisit bragging with classification and survival outcome sketchily.

The main motivation of bagging is that the “average” prediction of models from the same predefined model selection process gives better and more stable estimate of the outcome.

X The given predictor set(matrix with each observation predictor as one column)

- Y_X Continuous outcome corresponding to X
- ϕ Predefined model selection process (tree models in our case)
- $\phi(S)$ Based on random sample S with replacement, prediction of X by the best model purposed by ϕ
- μ_ϕ Expectation of prediction of X with respect to S.

Then by definition, we have

$$\mu_\phi = E_{S \in X}[\phi(S)]$$

This result could be interpreted as the “average” of predictions from models based on random sampling from X. The bootstrap estimate (μ_ϕ) is better than the prediction from the model solely based on sample X itself ($\phi(X)$).

$$E \left[(Y_X - \phi(X))^2 \right] \geq E \left[(Y_X - \mu_\phi)^2 \right]$$

This is the key idea and motivation for bootstrapping and bagging trees ¹³(1989).

For binary outcome, there is commonly two ways to estimate average of classifier function ϕ :

- a. Take the average of predicted probability across the bootstrap samples, and categorize based on the average. (Averaging)
- b. Categorize the observation into the class, which the observation has been classified into the most times in bootstrap samples. (Voting)

The accuracy of classifier is evaluated by out-of-bag estimation of misclassification error (OME). Hastiel¹⁰ et.al (2001) suggest voting is more favorable for large bootstrapping samples

(in our case we chose the bootstrap sample size as 150, which is not particularly large). This was confirmed in this thesis, with voting OME 0.2711 and averaging OME 0.259. Breiman⁷(1996a.) found that misclassification rate of CART decreased between 6% to 77% in the datasets he considered. The bagging of tree models for survival outcome was firstly introduced by Hothorn et.al¹¹(2003). The details and theory behind survival bagging are beyond the scope of this thesis.

The importance of variables (Imp) was measured by either improvement in decreasing risk or Gini index. For a given variable, the importance is calculated by summing the decreasing risk or Gini index across all the nodes, which were split by the variable. Two facts should be noticed: 1. The summation was weighted by the position of the node: a more ancestor nodes receive more “credit” in importance, as more data points were split by it. 2. Using the standard 0 and 1 cost, the decrease in risk equals to the decrease in the misclassification rate. Bagged Imp was calculated by summing Imp from each bootstrap samples, and calculating the average over all bootstrap samples. Incorporating with bagging, Imp gives a rather stabilized estimation of the variables in tree building process, especially in the situation where highly correlated variables with high Imp present. Imp measured in one single tree tends to suppress importance of all other correlated variables in favor of the one variable, which happens to be the primary splitter due to the data structure. While, on the other hand, bagged imp focuses equally on correlated variables when the bootstrap sample size is reasonable.

One should note that bagged tree models do not retain the same intuitive interpretation as single trees. Bagging trees cannot be presented by tree plots, and are not as interpretable in terms of the natural decision-making process. Therefore, the use of single trees may still be preferable in practice.

2.5 ROC CURVES BRIER'S SCORE AND INDEX C

This section introduces criteria to evaluate model prediction:

Receiver Operating Characteristic (ROC) curves plot 1-specificity by specificity of a binary classifier across different thresholds. Sensitivity and specificity can be defined as follows: Consider N data points, where the i th value belongs to one of the two possible categories, say $y_i = 0$ or 1. Under different cutoff values, the classifier categorizes every data point into one of the categories, say $\hat{y}_i = 0$ or 1. Thus the whole dataset is represented as $\{(y_i, \hat{y}_i) | i = 1, 2, \dots, N\}$

Then we have:

$$\text{sensitivity} = \frac{\sum_N I(y_i = \hat{y}_i = 1)}{\sum_N I(y_i = 1)}$$

$$\text{specificity} = \frac{\sum_N I(y_i = \hat{y}_i = 0)}{\sum_N I(y_i = 0)}$$

ROC curves are generated by plotting 1-specificity against sensitivity. It can be shown that the area under the curve (AUC) is the estimate of the probability of the classifier to rank a randomly chosen positive event higher than a randomly chosen negative event using normalized unit. The difference of two ROC curves can be tested by DeLong's test¹²(1988), where the difference of AUC is compared to zero.

Also, 10 folds cross-validation is used to compare ROC curves. In the cross-validation process, data points are randomly divided into 10 folds. For every single fold, all of the dataset except the fold is used to train the bagged tree model, two splits tree model and logistic regression model with covariates same as stepwise-selected regression model. And then, averaged 1-sensitivity was plot against specificity to construct cross-validation ROC curves.

Brier's score measures the accuracy of probabilistic prediction for binary outcome (in our case). Brier's score yields:

$$BS = \frac{1}{N_d} \sum_{t=1}^N (f_t - o_t)^2$$

Where N_d is the discrete event time instance, f_t is the prediction probability at instance t , and o_t is the 1, if the event actually happens at instance t , and 0 otherwise. Brier's score takes value from 0 to 1, where 0 reflects perfect prediction and 1 represents false prediction.

3.0 RESULTS

3.1 ANALYSIS OF BINARY OUTCOMES FOR CHRONIC KIDNEY DISEASE

The following section outlines the results of the analysis in the thesis. There were initially 241 patients in the study cohort. Eight patients were CKD positive at baseline (eGFR < 60 mg/dL), and 10 patients without CKD had only baseline visit data, and thus the 18 were censored for all the analysis in the thesis. In the cohort of the remaining 223 patients, and 205 had complete data for the covariates used. Moreover, the cohort (N=223) could be further categorized into the following three classes:

- (1) 89 reached stage 3 CKD during the study.
- (2) 85 did not reach CKD but did have eGFR measured after year 10.
- (3) 49 did not reach CKD and lost for follow-up before year 10. (i.e. no observed eGFR after year 10 and were only censored for CKD binary classification).

For variable race, Caucasian/other was coded as baseline comparing with African American. For variable genotype, PKD2/NMD was coded as baseline comparing with PKD1.

3.1.1 Multivariable Logistic Regression

As described in section 2.1, the outcome is binary classification: during the study, people either be classified into CKD (eGFR dropped below 60 mg/dL), or into control (eGFR did not drop below 60 mg/dL). The size of cohort used in for the binary outcome is 174. (See section 3.1)

The full model includes all variables is listed in Table1 below. All variables listed here were measured at baseline.

Table 1. Coefficients of full logistic regression

	OR	p-value	Adj. OR	Adj. p-value
age	1.089	<0.001*	1.031	0.330
htTKV	1.005	<0.001*	1.003	0.006*
PKD1	2.643	0.018*	3.812	0.034*
Gender(M)	0.953	0.878	0.662	0.418
Race(African American)	0.249	0.040*	1.185	0.877
BMI	1.073	0.030*	0.999	0.985
BUN	1.310	<0.001*	1.161	0.054
eGFR	0.915	<0.001*	0.916	<0.001*
MCP	1.002	<0.001*	1.000	0.712

*significant under 0.05

Stepwise variable selection under AIC gives model with covariates of baseline eGFR, height adjusted TKV genotype and BUN. The odds ratio and p-value of the model are given in Table2.

Table 2. Coefficients of stepwise logistic regression

	OR	P-value
PKD 1	2.769	0.068
htTKV	1.003	<0.001
eGFR	0.913	<0.001
BUN	1.135	0.064

As shown by the full model in table 1, baseline htTKV and eGFR are the most significantly associated with the outcome. As expected larger kidneys and lower eGFR at baseline were associated with higher odds of developing CKD.

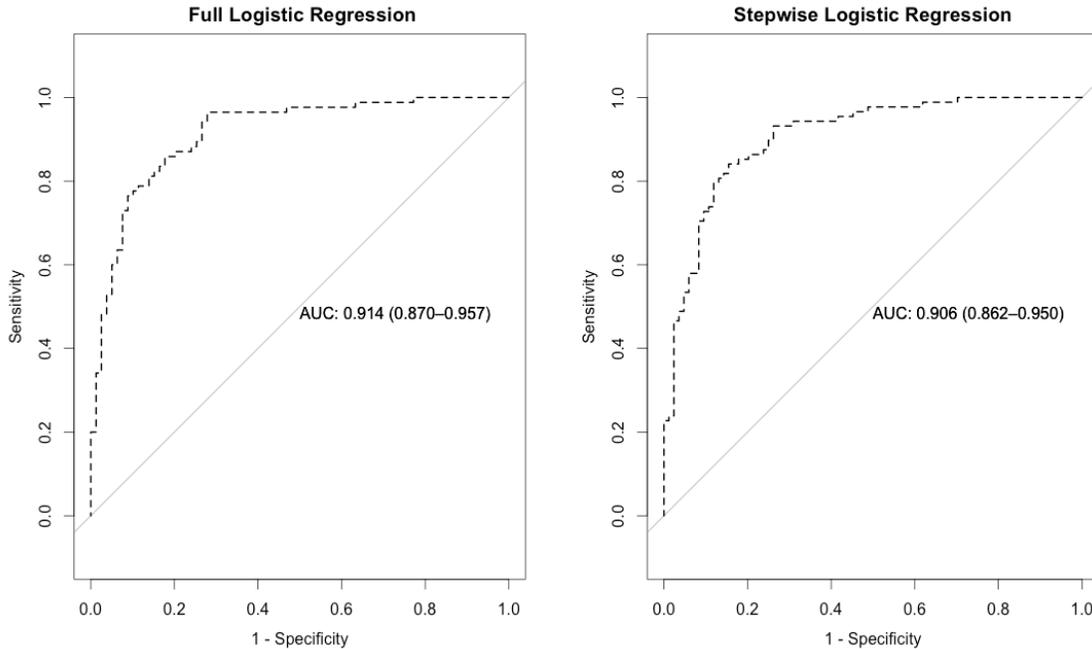
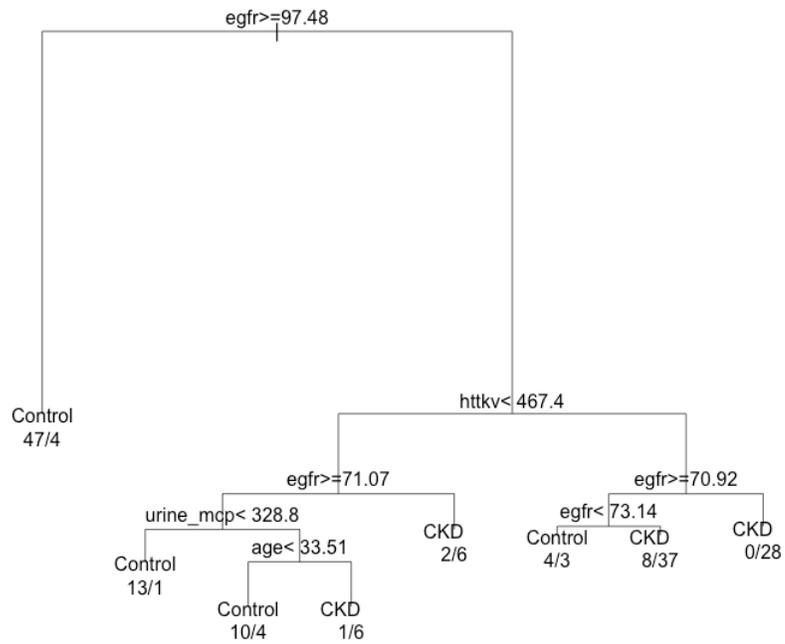


Figure 1 ROC Curves for logistic regression models

3.1.2 Classification Trees

Figure 2 shows the tree model without pruning. The numbers below each terminal node are the number of people normal/with CKD in the node. From figure 2, we can observe that only the baseline eGFR, htTKV, age and MCP contribute to the prognosis of CKD. The variable eGFR and htTKV account for the most splits. Figure 3 shows the relative error associated with each complexity parameter, leading to an optimal value of 0.07. The pruned tree is shown in Figure 4, and is only based on the variables of baseline eGFR and TKV, with misclassification rate at

18.95%. As shown in figure 5, the pruned tree model has an area under curve (AUC) of 0.857, suggesting strong discrimination ability, with an optimal threshold at 0.623.



Corresponding splitter represented by the number of people normal/with CKD in the node

Figure 2. The model without pruning classifying CKD

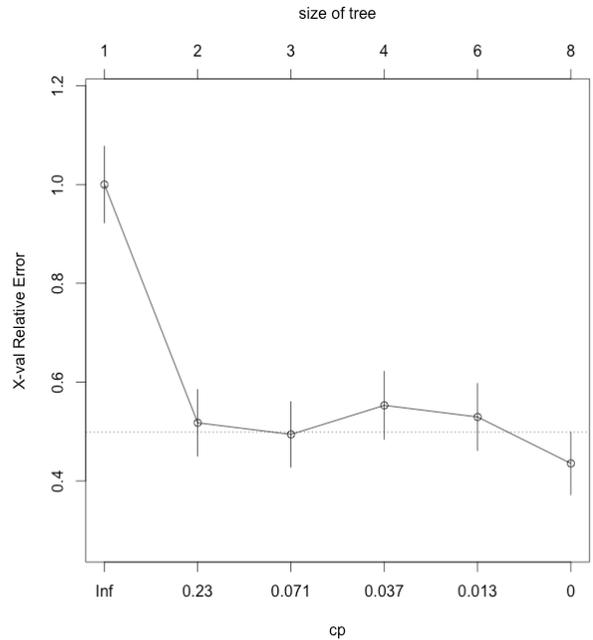
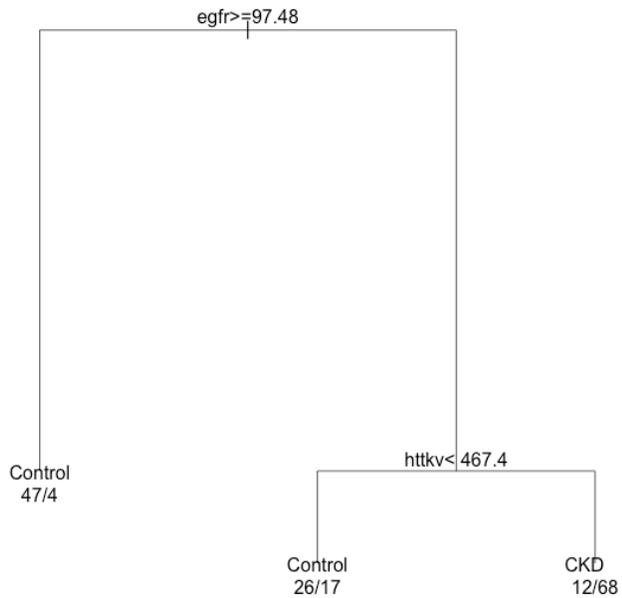


Figure 3. Cross-validation relative error for each complex parameter value



Corresponding splitter represented by the number of people normal/with CKD in the node

Figure 4. Pruned tree for complete tree for CKD categorization

Bagging trees are grown by 500 bootstrap replicate samples of size 150 with replacement. The out-of-bag estimation of misclassification error (OME) were 0.259 for averaging and 0.2831 for voting. As OME estimates were close, for calculation efficiency and interpretability, we use selected the voting model. Then, the predicted probability could be explained as the proportion of being categorized as CKD in all bootstrap replicates. (Bagged tree AUG 0.99). If omitting the 6 observations with missing data, the bagged model predicts even better with OME 21.08%. Importance of variables is listed in Table 3 below.

Table 3. Importance of variables

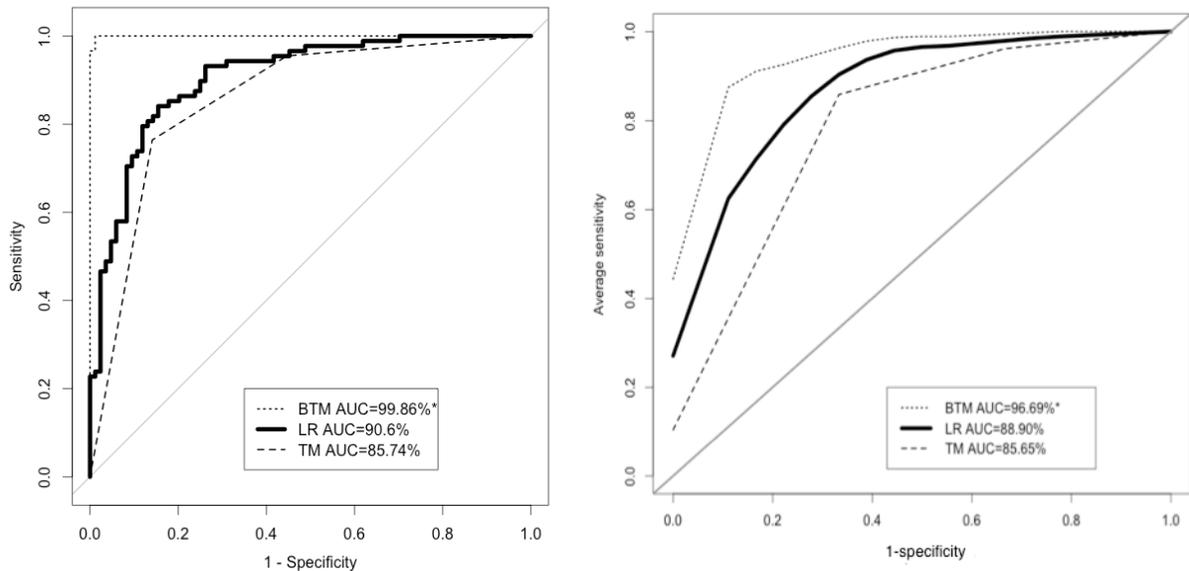
Variables	Mean Decrease Accuracy	Mean Decrease Gini
eGFR	36.33	28.81
htTKV	22.61	21.65
BUN	8.98	5.47
age	4.15	5.01
MCP	4.09	6.17
BMI	-0.16	6.04

- All other variables have both values less than 2.

Judging from both criterion in Table 3, we can observe that baseline eGFR and htTKV are much more predictive comparing with other potential predictors. This result agrees with the pruned tree model, in that only eGFR and htTKV end up in the model.

Figure 5 presents the ROC curves of all three models. (BTM for bagged tree model; LR for logistics regression model; TM for pruned tree model).

Using DeLong’s test for the AUC, BTM has the highest AUC in the learning dataset, significantly different with other two models, while LR and TM are not significantly different. From the results, we can see that BTM gives best in-bag prediction, but the model itself is hard to explain and present. TM yields a similar AUC as compared to regular logistic regression, but with less variables, and greater interpretability in decision-making process. All three models suggest baseline eGFR and htTKV highly predictive. Using 10-fold cross-validation, the different models yield results for the ROC curves: The BTM still has best AUC, which is significantly different than the AUCs from LR and TM. The AUC was not significantly different between LR and TM models. The classification accuracy described in more detail in table 4.



(Left regular ROC, Right Cross-validation ROC)

(BTM for bagged tree model; LR for logistics regression model; TM for pruned tree model)

Figure 5. ROC curves for classifying CKD

Table 4. Area under ROC (AUC)

Models	Regular ROC		Cross-Valid ROC	
	AUC	95% CI	AUC	95% CI
Pruned Tree	0.86	0.80~0.91	0.86	0.77~0.94
Bagged Tree	>0.99	0.99~1.00	0.97	0.94~0.99
Logistic Regression	0.91	0.86~0.95	0.89	0.84~0.94

3.2 ANALYSIS OF TIME TO CHRONIC KIDNEY DISEASE

3.2.1 Multivariable Cox Model

Full model including all variables is listed in Table 5 below:

Table 5. Coefficients of full Cox regression model

	RR	p-value	Adj. RR	Adj. p-value
age	1.068	<0.001*	1.032	0.07
htTKV	1.002	<0.001*	1.002	<0.001*
PKD 1	1.952	0.039*	1.880	0.09
Gender(M)	1.074	0.746	1.023	0.93
Race (African American)	0.249	0.040*	0.591	0.41
BMI	3.512	0.007*	1.007	0.76
BUN	1.173	<0.001*	1.051	0.14
eGFR	0.937	<0.001*	0.947	<0.001*
MCP	1.001	<0.001*	1.000	0.48

*significant under 0.05

Full model is significant ($p < 0.001$), with variables height adjusted TKV and baseline eGFR significant under 0.05 alpha level.

After forward selection using the AIC criterion, the Cox survival model retains the variables of htTKV, eGFR, age, PKD1 and BUN. The model is described below in table 6:

Table 6. Coefficients of Cox regression

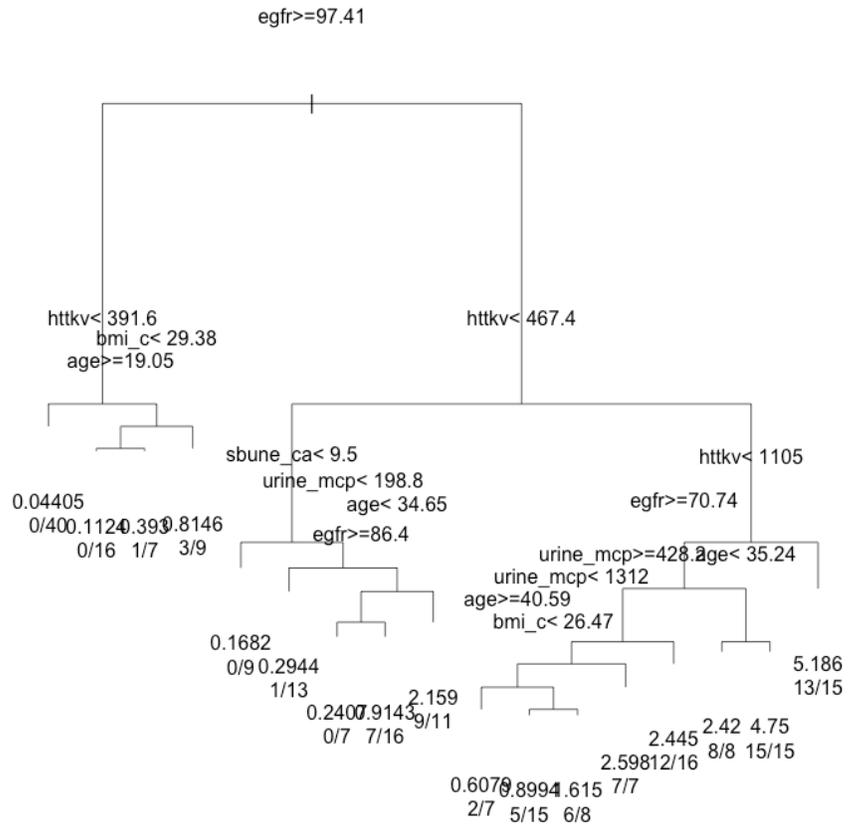
	RR	Z value	P-value
htTKV	0.002	4.84	<0.001*
eGFR	-0.0526	-6.35	<0.001*
age	0.030	1.73	0.084
PKD 1	0.654	1.79	0.074
BUN	0.050	1.62	0.110

*significant under 0.05

eGFR and htTKV are highly significant. Likelihood ratio test of the full model is highly significant ($p < 0.001$), and the C index of the model is 0.835.

3.2.2 Survival Tree

The full tree without pruning for the survival data is listed below in figure 6. (The number below the terminal nodes represent : number of people with CKD / total number of people in the node was listed below). Cross-validation with $N=219$ (our sample size in 219, see section 3.1), we pruned the tree with the optimized cp value of 0.037 (Figure 6). The pruned tree is shown in Figure 8, where only based on the variables of baseline eGFR and htTKV.



(Number of CKD/Number of total people in the node)

Figure 6. Complete grown tree for CKD survival

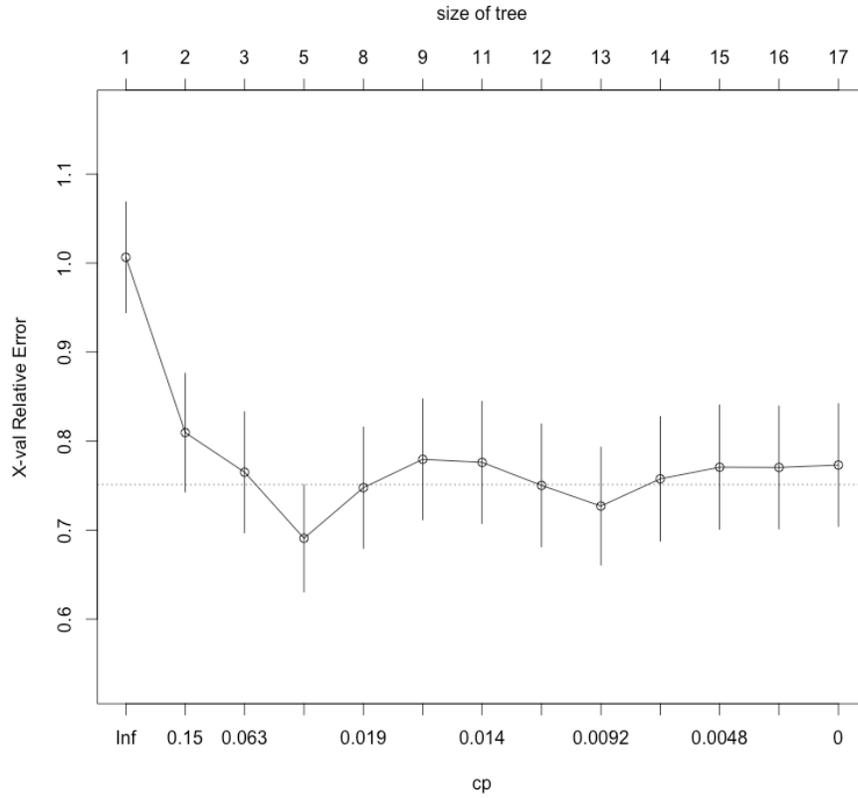
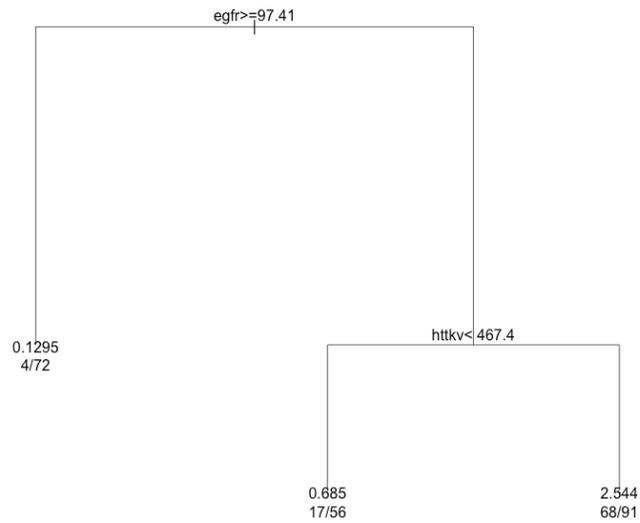


Figure 7. Cross-validated relative error for each complex parameter



(Number of CKD/Number of total people in the node)

Figure 8. Optimized tree for CKD survival time

The bagged survival tree is constructed using 500 bootstrapping sample of size 150. The model had an out-of-bag estimate of Brier's score of 0.1129, suggesting strong prediction ability. Also, as shown in figure 9, baseline eGFR and htTKV have high significance comparing with other possible variables. Finally, the KM estimator of the three groups categorized by pruned tree is presented in Figure 10. Trend log-rank test suggested strong trend in three groups. ($p < 0.001$)

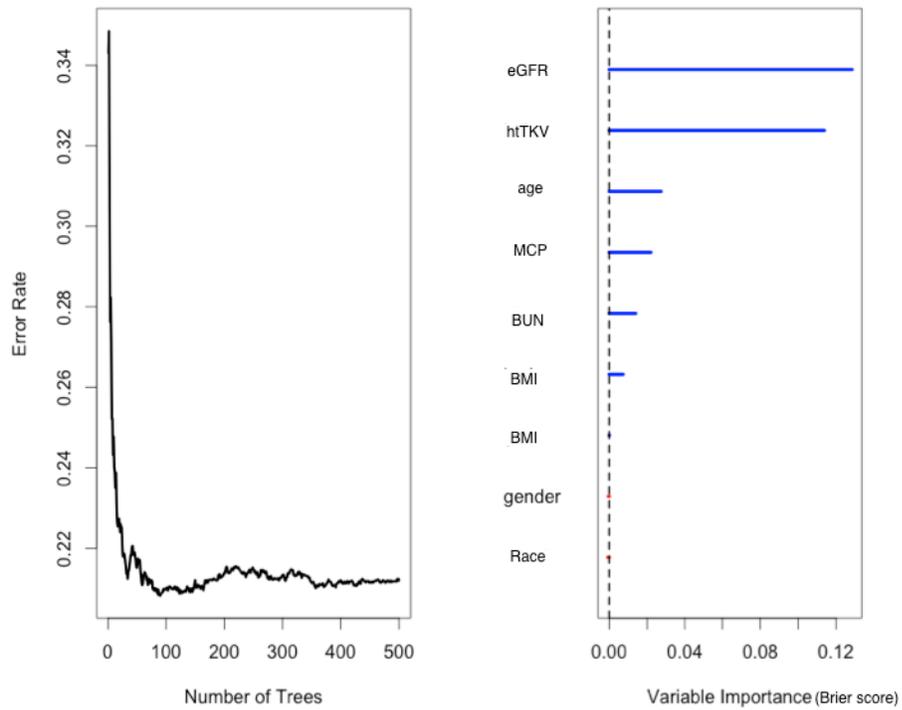


Figure 9. Variable importance for bagged CKD survival tree

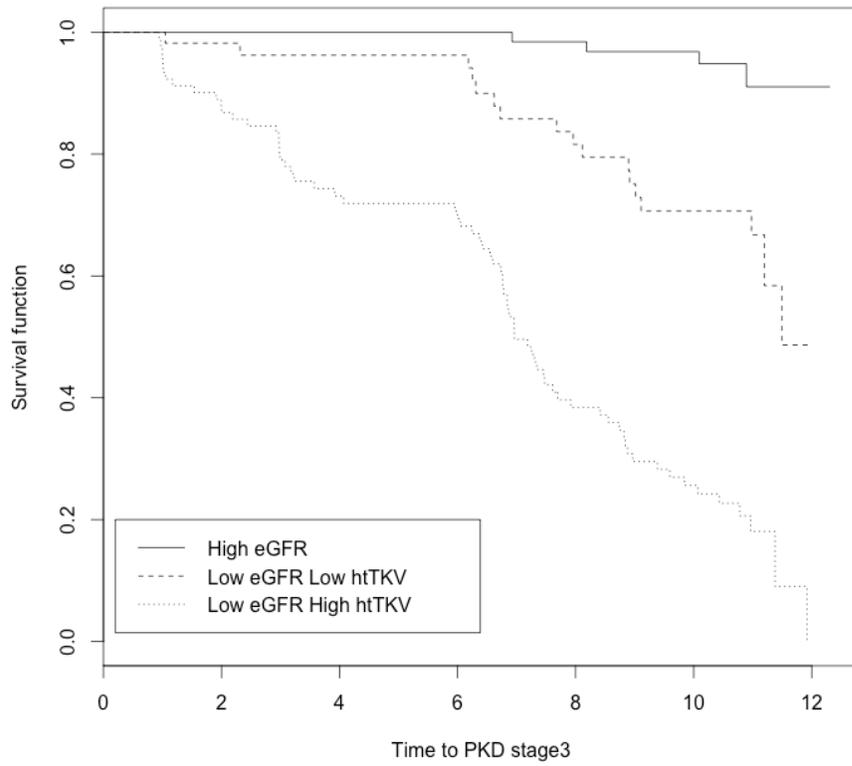


Figure 10. KM curves by eGFR and htTKV groups

4.0 DISCUSSION

For CKD classification, logistic regression suggests significance of baseline eGFR, htTKV and PKD 1 ($p < 0.05$) as significant predictors during the follow-up time in CRISP study. Optimal pruned tree depended only on eGFR and htTKV, with cutoff value at 97.48 and 467.4. Bagged tree model also suggests high importance of the two variables mentioned above. For CKD-free survival outcome, Cox model gives similar result as logistic regression model for binary outcome, namely eGFR and htTKV are significant ($p < 0.001$). Optimal survival pruned tree also depends eGFR and htTKV, with similar cutoff value at 97.41 and 467.4.

All the results in the thesis have pointed to the similar conclusion, that there is inherent connection and association between baseline eGFR, htTKV and CKD trajectory. Moreover, these two factors, even without any other variables, accurately predict CDK status longitudinally. Results show that higher baseline eGFR (above 97) and smaller kidneys (htTKV below 467) are strongly associated with better prognosis.

These cutoffs could be advisory to actual clinical decision and prognostics of CKD. Comparing with current continuously GFR monitoring for CKD progress, two baseline predictors measured in the early phase of the disease, makes early interventions more practical.

APPENDIX R CODE

```
### Libraries used

library("rpart")
library("survival")
library(ipred)
library(randomForest)
library(pROC)
library("mlogit")

### first line for survival outcome, second for binary

final<-read.csv("noLC(efgr+endpoint).csv")
final<-read.csv("BinaryFinal.csv")

### Survival time defining

survival<-function(data,critical){

  datas<-data
  r<-nrow(datas)
  c<-ncol(datas)
  # Reading in data

  k<-c(rep(0,r))
  timeub<-c(rep(0,r))
  timelb<-c(rep(0,r))
  legfr<-c(rep(0,r))
```

```

hegfr<-c(rep(0,r))
timec<-c(rep(0,r))
time<-c(rep(0,r))
indicator<-c(rep(0,r))
datas<-cbind(datas,k,indicator,timeub,timelb,hegfr,legfr,timec,time)

# k denotes the postion of first critical value
# indicator is 1 if observed, 0 if RC
# timeub is the LAST critical time when the ob stayed above critical value
# timelb is the FIRST critical time when the ob stayed under critical value
# time is the average of timeub and time lb

for (i in 1:r){

  j<-3

  repeat {
    if (is.na(datas[i,j])) {datas[i,c+1]=datas[i,c+1]+1
                          j<-j+2}
    else if (datas[i,j] > critical) {
      datas[i,c+5]=datas[i,j] #high egfr
      datas[i,c+3]=datas[i,j-1] #corresponding timeub
      datas[i,c+1]=datas[i,c+1]+1
      j<-j+2
    }
    else if (data[i,j] <= critical) {

      datas[i,c+2]=1
      datas[i,c+6]=data[i,j] #low egfr
      datas[i,c+4]=data[i,j-1] #corresponding timelb

      break()
    }
  }
}

```

```

        }

        if (j>c) break()
    }

}

##Final computation of indicator and survival time##

dataf<-datas

for (i in 1:r){
    if (dataf[i,c+2]==0) dataf[i,c+7]=dataf[i,c+3]
    else {
        beta1<-(dataf[i,c+5]-dataf[i,c+6])/(dataf[i,c+3]-dataf[i,c+4])
        beta2<-dataf[i,c+6]-(dataf[i,c+5]-dataf[i,c+6])/(dataf[i,c+3]-
dataf[i,c+4])*dataf[i,c+4]
        dataf[i,c+7]=(critical-beta2)/beta1
    }

}

##uncorrected event time
for (i in 1:r){
    if (dataf[i,c+2]==0) dataf[i,c+8]=dataf[i,c+3]
    else dataf[i,c+8]=dataf[i,c+4]
}

```

```

for (i in 1:r){
  if (dataf[i,c+2]==1 & dataf[i,c+3]==0 & dataf[i,c+4]==0 & dataf[i,c+5]==0){
    dataf[i,c+7]=0
    dataf[i,c+8]=0
    dataf[i,c+2]=2
  }
  # Corrected for already healthy people,use indicator 2 to denote LC
}
return(dataf)
}

### data cleaning

survstat <- factor(final$indicator, levels = 0:1, labels =
c("Control", "CKD"))

##collasing PKD2 & NMD into one category

finalx<-final[,c(1:6,8:18)]
racex<-c(final$race4=="African American")
racex[final$race4=="Hispanic"]<-NA
race4<-racex
final<-cbind(finalx,race4)

##collasing race category

genex<-c(final$genetype=="PKD1")
miss<-is.na(final$genetype)
final<-cbind(final,genex)

### for binary outcome

## logistic regression

```

```

final1<-na.omit(final) #Removing na for stepwise.
full<-
glm(indicator~age+httkv+genex+gender+as.factor(race4)+bmi_c+egfr+sbune_ca+
      urine_mcp,data=final1,family=binomial())
step(full,direction="backward")
stepw<-glm(formula = indicator ~ httkv + egfr + sbune_ca + genex, family =
binomial(),
      data = final)
exp(coef(stepw))
fullor<-exp(cbind(OR = coef(full), confint(full)))
exp(std(stepw))

```

Classification trees

```

dfit<-rpart(formula=survstat~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
+urine_mcp,data=final,method="class",control=rpart.control(cp=0,xval=174))
plot(dfit) #main="Complete Tree for binary outcome"
text(dfit,use.n=T,xpd = TRUE)
printcp(dfit)
plotcp(dfit)

fit3<-prune(dfit,cp=0.071)
plot(fit3) #,main="Optimized Tree for Binary outcome"
text(fit3,use.n=T,xpd = TRUE)
grpbin<-fit3$where

```

BAGGING

```

test1<-bagging(survstat~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
+urine_mcp,data=final,coob=T,nbag=500,ns=150,replace=T,

```

```

        aggregation="averaging")
test2<-bagging(survstat~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
+urine_mcp,data=final,coob=T,nbag=500,ns=150,replace=T,
        aggregation="majority")

test3<-
bagging(Surv(time,indicator)~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca+s
erumcreat
+urine_mcp,data=final,coob=T,nbag=500,ns=150,replace=T)

###random forest and importance

bag.final<-
randomForest(survstat~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
+urine_mcp,data=final,samplesize=150,ntree=500,
replace=T,mtry=9, importance=T,na.action = na.omit)
bag.final
importance(bag.final)

###prediction

#binary: pruned:fit3
prutree<-predict(fit3,type="prob",newdata=final)
outcome<-cbind(final$pkdid,final$indicator,prutree[,2])
colnames(outcome)<-c("pkdid","PKD","Pruned")
rocl<-roc(PKD ~ Pruned, outcome,ci=T)
plot(rocl,print.auc=T,lwd=6,legacy.axes=T,
      main="ROC Curve for Pruned Tree for Binary Outcome ")

plot(smooth(rocl),col="blue",lwd=4,add=T,lty=2)

```

```

legend(0.28,0.15, legend=c("Empirical", "Smoothed"),
      col=c(par("fg"), "blue"), lwd=4,lty=1:2)
#Binary Bagged:test2
voting<-predict(test2,,newdata=final,type="prob")
outcome<-cbind(outcome,voting[,2])
colnames(outcome)<-c("pkdid","PKD","Pruned","Bagged")
roc2<-roc(PKD ~ Bagged, outcome,ci=T)
plot(roc2,print.auc=T,print.thres=T,lwd=2,
     main="ROC Curve for Bagged Tree for Binary Outcome ")
#Binary Stepwise Logistic: stepw
stepwise<-predict(stepw,newdata=final,type="response")
outcome<-cbind(outcome,stepwise)
roc3<-roc(PKD~stepwise,outcome,ci=T)
plot(roc3,print.auc=T,print.thres=T,lwd=2,legacy.axes=T,
     main="ROC Curve for Stepwise Logistic Regression")

#Binary full model Logistic
full<-predict(full,newdata=final,type="response")
outcome<-cbind(outcome,full)
roc4<-roc(PKD~full,outcome,ci=T)
plot(roc4,print.auc=T,print.thres=T,lwd=2,legacy.axes=T,
     main="ROC Curve for Full Logistic Regression")
plot(roc1,lty=2,legacy.axes=T,ylim=c(0,1))
plot(roc3,lwd=6,add=T,lty=1,ylim=c(0,1))
plot(roc2,lty=3,add=T,ylim=c(0,1))
legend(0.65,0.2,legend=c("BTM          AUC=99.86%*", "LR          AUC=90.6%", "TM
AUC=85.74%"),lty=c(3,1,2),lwd=c(2,6,2))
roc.test(roc2,roc3)
roc.test(roc1,roc3)

```

CROSS-VALIDATION ROC

```
.cvFolds <- function(Y, V){ #Create CV folds in indices(stratify by outcome)
  Y0 <- split(sample(which(Y==0)), rep(1:V, length=length(which(Y==0))))
  Y1 <- split(sample(which(Y==1)), rep(1:V, length=length(which(Y==1))))
  folds <- vector("list", length=V)
  for (v in seq(V)) {folds[[v]] <- c(Y0[[v]], Y1[[v]])}
  return(folds)
}

#Train/test glm for each fold
#logistic regression
.doFitLR <- function(v, folds, data){
  fit <- glm(formula = indicator ~ httkv + egfr + sbune_ca,
            data=data[-folds[[v]],,], family=binomial)
  pred <- predict(fit, newdata=data[folds[[v]],,], type="response")
  return(pred)
}

#Pruned Tree model
.doFitPT <- function(v, folds, data){
  fit<-
rpart(formula=indicator~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
      +urine_mcp,data=data,method="class",
      control=rpart.control(cp=0,xval=174,maxdepth=2))

  pred2<- predict(fit, newdata=data[folds[[v]],,], type="prob")
  pred<-pred2[,2]
  return(pred)
}

#Bagged Tree model
.doFitBAG <- function(v, folds, data){
  fit<-bagging(indicator~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
```

```

+urine_mcp,data=data,coob=T,nbag=500,ns=150,replace=T,
      aggregation="majority")

  pred<- predict(fit, newdata=data[folds[[v]],], type="prob")

  return(pred)
}

#### Make predictions and outcome for crossvalidation samples with
#### Sample size V=10

folds <- .cvFolds(Y=final$indicator, V=10)
predictionLR <-sapply(seq(10), .doFitLR, folds=folds, data=final)
predictionPT <-sapply(seq(10), .doFitPT, folds=folds, data=final)
predictionBAG <-sapply(seq(10), .doFitBAG, folds=folds, data=final)

outcome<-list()
for (v in seq(10)) {outcome[[v]] <- final[folds[[v]],10]}

#### Plotting ROC for crossvalidations

install.packages("ROCR")
library(ROCR)
pred.BAG<-prediction(predictionBAG,outcome) ##bagged
pred.PT<-prediction(predictionPT,outcome) ##pruned
pred.LR<-prediction(predictionLR,outcome) ##logistic
perf.BAG<-performance(pred.BAG, "sens", "fpr")
perf.PT<-performance(pred.PT,"sens", "fpr")
perf.LR<-performance(pred.LR, "sens", "fpr")

####ROC CURVES PLOTTING#####
plot(perf.PT, avg="vertical",lty=2,lwd=2,legacy.axes=T)

```

```

plot(perf.LR, avg="vertical",lty=1,lwd=6,add=T)
plot(perf.BAG, avg="vertical",lty=3,lwd=2,add=T)
legend(0.43,0.2,legend=c("BTM AUC=96.69%*", "LR AUC=88.90%", "TM AUC=85.65%"),
      lty=c(3,1,2),lwd=c(2,6,2))

abline(a=0,b=1,lwd=3,col= "gray60",add=T)

#### Cross- Validation AUC calculation####
install.packages("cvAUC")
library(cvAUC)

ciPT<-ci.cvAUC(predictionPT,outcome, label.ordering = NULL, folds = NULL,
confidence = 0.95)
ciBAG<-ci.cvAUC(predictionBAG,outcome, label.ordering = NULL, folds = NULL,
confidence = 0.95)
ciLR<-ci.cvAUC(predictionLR,outcome, label.ordering = NULL, folds = NULL,
confidence = 0.95)

### Survival outcome

##COX and STEPWISE SELECTION

final1<-na.omit(final)
full<-coxph(Surv(time,indicator)~age+httkv+genex+gender+as.factor(race4)
            +bmi_c+egfr+sbune_ca+urine_mcp,data=final1,)
step(full,direction="backward")
stepcox<-coxph(Surv(time,indicator)~httkv+egfr+sbune_ca,data=final)
summary(stepcox)
predict(test1,type="class")

```

Survival regression trees

```
survr<-
rpart(formula=Surv(time,indicator)~age+httkv+gender+race4+genex+bmi_c+egfr+sbun
e_ca+urine_mcp,data=final,control=rpart.control(cp=0,xval=200))
plotcp(survr)
printcp(survr)
plot(survr) #main="Complete Tree for Survival Outcome"
text(survr,use.n=T,xpd = TRUE)
survp<-prune(survr,cp=0.0377609 )
plot(survp)#,main="Pruned Tree for survival outcome"
text(survp,use.n=T,xpd = TRUE)
newgrp<-survp$where
par(mai=c(1.5,1.5,1,1))
plot(survfit(Surv(time,indicator)~newgrp,data=final),mark.time=F,lty=1:3,
      main="K-M estimator for CART result")
title(xlab="Time to PKD stage3",ylab="Survival function")
legend(0.2,0.2,legend=c("High eGFR","Low eGFR Low TKV","Low eGFR High
TKV"),lty=1:3)
```

Bagging Survival tree and Importance of Variables

```
library(randomForestSRC)
bag.surv<-
rfsrc(Surv(time,indicator)~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
      +urine_mcp,data=final1,samplesize=150,ntree=500,
      replace=T,mtry=9, importance="random")

bag.surv<-
rfsrc(Surv(time,indicator)~age+httkv+genex+gender+race4+bmi_c+egfr+sbune_ca
      +urine_mcp,data=final1,samplesize=150,ntree=500,
      replace=T,mtry=9, importance="permute")
bag.surv
plot(bag.surv)
```

Unadjusted models and CI

```
m1<-glm(indicator~age,data=final1,family=binomial())
m2<-glm(indicator~httkv,data=final1,family=binomial())
m3<-glm(indicator~genex,data=final1,family=binomial())
m4<-glm(indicator~gender,data=final1,family=binomial())
m5<-glm(indicator~as.factor(race4),data=final1,family=binomial())
m6<-glm(indicator~bmi_c,data=final1,family=binomial())
m7<-glm(indicator~egfr,data=final1,family=binomial())
m8<-glm(indicator~sbune_ca,data=final1,family=binomial())
m9<-glm(indicator~urine_mcp,data=final1,family=binomial())

exp(rbind(coef(m1),coef(m2),coef(m3),coef(m4),coef(m6),coef(m7),coef(m8),coef(m
9)))
exp(coef(m5))

s1<-coxph(Surv(time,indicator)~age+httkv+genex+gender+as.factor(race4)
          +bmi_c+egfr+sbune_ca+urine_mcp,data=final1)
s1<-coxph(Surv(time,indicator)~age,data=final1)
s2<-coxph(Surv(time,indicator)~httkv,data=final1)
s3<-coxph(Surv(time,indicator)~genex,data=final1)
s4<-coxph(Surv(time,indicator)~gender,data=final1)
s5<-coxph(Surv(time,indicator)~as.factor(race4),data=final1)
s6<-coxph(Surv(time,indicator)~bmi_c,data=final1)
s7<-coxph(Surv(time,indicator)~egfr,data=final1)
s8<-coxph(Surv(time,indicator)~sbune_ca,data=final1)
s9<-coxph(Surv(time,indicator)~urine_mcp,data=final1)
```

BIBLIOGRAPHY

1. Grantham JJ, Nair V, Winklhofer F. Cystic diseases of the kidney. (2000) *Brenner BM, ed. Brenner & Rector's The Kidney*. Vol. 2. 6th ed. Philadelphia: WB Saunders Company: 1699-1730.
2. Grantham, Jared J., et al. "Volume progression in polycystic kidney disease." *New England Journal of Medicine* 354.20 (2006) : 2122-2130
3. Chapman, Arlene B., "Kidney volume and functional outcomes in autosomal dominant polycystic kidney disease." *Clinical Journal of the American Society of Nephrology* 7.3 (2012) : 479-486
4. Levey, A. S., Bosch, J. P., Lewis, J. B., Greene, T., Rogers, N., & Roth, D. (1999). A more accurate method to estimate glomerular filtration rate from serum creatinine: a new prediction equation. *Annals of internal medicine*,130(6), 461-470.
5. Ciampi, A., Thiffault, J., Nakache, J.-P. and Asselain, B. (1986). Stratification by Stepwise Regression, Correspondence Analysis and Recursive Partition: A Comparison of Three Methods of Analysis for Survival Data with Covariates. *Computational Statistics & Data Analysis* 4, 185–204.
6. Ciampi, A., Chang, C. H., Hogg, S. and McKinney, S. (1987). Recursive Partition: A Versatile Method for Exploratory Data Analysis in Biostatistics. *Biostatistics* 23–50
7. Leo Breiman (1996a), Bagging Predictors. *Machine Learning* 24(2), 123–140.
8. Leo Breiman (1996b), Out-Of-Bag Estimation. *Technical Report*
<ftp://ftp.stat.berkeley.edu/pub/users/breiman/OOBestimation.ps.Z>.
9. Leo Breiman (1998), Arcing Classifiers. *The Annals of Statistics* 26(3), 801–824.
10. Hastie, T., Tibshirani, R., Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, New York.
11. Hothorn, Torsten, and Berthold Lausen. "On the exact distribution of maximally selected rank statistics." *Computational Statistics & Data Analysis* 43.2 (2003): 121-137.
12. DeLong ER, DeLong DM, Clarke-Pearson DL (1988): Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* 44:837-845.
13. Noreen, E. (1989). *Computer-intensive methods for testing hypotheses: an Introduction*. Willey, ISBN: 978-0-471-61136-3.