Analysis on Data Set ACTG175

En Hu 18343043

2021/1/17

Abstract

In this article, we select several reasonable predictors at baseline, then make use of descriptive methods and **Non-Paramatric Test** to find the potential important predictors in determining `cd496`, CD4 cell count at 96 weeks after baseline. We first use these predictors and the response to build **GAM**, making use of **Smoothing Spline** and **Dummy Variable**. In studying parameters in smoothing spline, **Cross Validation** method is adopted to determine parameters λ and Effective Degree of Freedom, and **Bootstrap** method is applied to estimate the standard deviation of the method adopted above based on similar data set. Secondly, we study the association between response and predictors using **tree based method**. A best **Regression Tree** is built to study the association and importance of each predictor. To try to build a accurate predictive model, **Random Forest** model is built and **Bagging** is also easily built for comparison.

The first conclusion is about the importance and effect of predictors in determining CD4 cell count at 96 week after baseline. CD4/CD8 at baseline, which represents the status of the war between immune system and virus at the beginning of study, plays the most important role. Those who have higher Ratio48 at baseline tend to have higher CD4 count at 96 weeks. Then treatment taken(that is treatment with zidovudine only or with two nucleosides) and stratification of history of antiretroviral therapy play second roles. Treatment with two nucleosides performs better than with zidovudine only, and who are naive to antiretroviral therapy have higher CD496. Karnofsky score follows. Those who have Karnofsky score higher have higher CD496. Finally patient's weight is the least important.

The second conclusion is that it's difficult to predict `cd496` based on data at baseline.

Introduction

Based on the data set provided by ACTG175[1], this article aims at studying the relationship between the response variable, CD4 T cell count at 96±5 weeks, and several predictor variables and building a predictive model. Patients' age, weight, Karnofsky score, antiretroviral history, treatment taken(treatment with zidovudine only or two nucleosides) and the ratio of CD4/CD8 T cell count at baseline are considered as potential predictors. We mainly used non-parametric methods to achieve goals above.

In section 2, we first describe the data set and variables. In section 3, we study the relationship between the response variable and predictor variables, using descriptive measures, to obtain intuitive results. In section 4, we use non-parametric test to select potential predictors. In section 5, we make use of smoothing spline regression. In section 6, we apply bootstrap method to study the deviation of regression parameters. In section 7, a GAM is built based on the discussion above. In section 8, we make use of a new approach, tree based method and build a best regression tree. In section 9, we build random forest model to try to predict. In section 10, we make a conclusion.

Discription of the Data Set ACTG175

The data set, ACTG175, that our analysis based on, is provided by the study *A TRIAL COMPARING NUCLEOSIDE MONOTHERAPY WITH COMBINATION THERAPY IN HIV-INFECTED ADULTS WITH CD4 CELL COUNTS FROM 200 TO 500 PER CUBIC MILLIMETER*[1] and its corresponding R package. Aiming at evaluating treatment with either a single nucleoside or two nucleosides in adults infected with

HIV-1, whose CD4 cell counts were from 200 to 500 per cubic millimeter, the study followed 2467 patients with 4 different therapies and collected 2139 observations on 27 variables. 27 variables are described in table 1.

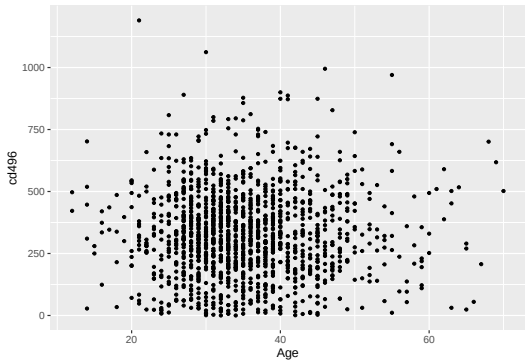
Table 1: The Variables in the Data and Corresponding Descriptions

Variables	Descriptions
pidnum	patient's ID number
age	age in years at baseline
wtkg	weight in kilogram at baseline
hemo	hemophilia (0=no, 1=yes)
homo	homosexual activity (0=no, 1=yes)
drugs	history of intravenous drug use (0=no, 1=yes)
karnof	Karnofsky score (on a scale of 0-100), which quantifies a patient's functional abilities and impact of treatments like chemotherapy on their basic functional capacities
oprior	non-zidovudine antiretroviral therapy prior to initiation of study treatment (0=no, 1=yes)
z30	zidovudine use in the 30 days prior to treatment initiation (0=no, 1=yes)
zprior	zidovudine use prior to treatment initiation (0=no, 1=yes)
preanti	number of days of previously received antiretroviral therapy
race	race (0=white, 1=non-white)
gender	gender (0=female, 1=male)
str2	antiretroviral history (0=naive, 1=experienced)
strat	antiretroviral history stratification (1='antiretroviral naive', 2='$>$ 1 but $\leq$ 52 weeks of prior antiretroviral therapy', 3='$>$ 52 weeks')
symptom	symptomatic indicator (0=asymptomatic, 1=symptomatic)
treat	treatment indicator (0=zidovudine only, 1=other therapies)
offtrt	indicator of off-treatment before 96$\pm$5 weeks (0=no,1=yes)
cd40	CD4 T cell count at baseline, which implies one's immunity
cd420	CD4 T cell count at 20$\pm$5 weeks
cd496	CD4 T cell count at 96$\pm$5 weeks (=NA if missing)
r	missing CD4 T cell count at 96$\pm$5 weeks (0=missing, 1=observed)
cd80	CD8 T cell count at baseline, which implies the intensity of the battle between immune system and virus.
cd820	CD8 T cell count at 20$\pm$5 weeks
days	number of days until the first occurrence of: (i) a decline in CD4 T cell count of at least 50%,(ii) an event indicating progression to AIDS,(iii) death, or (iv) missing
cens	indicator of observing the first 3 events in days
arms	treatment arm (0=zidovudine, 1=zidovudine and didanosine, 2=zidovudine and zalcitabine,3=didanosine).
Ratio48	cd40/cd80, which indicates the status of war between immuine system and virus.

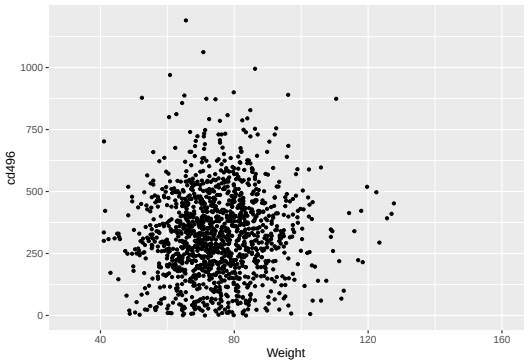
Descriptive Statistic

In this section, we perform scatter plots of cd496 against quantitative predictors(age, weight and cd40/cd80) and box-plots of cd496 against qualitative predictors(Karnofsky score, antiretroviral history, and treatment taken) respectively to give intuitive results. Plots is shown in figure 1.

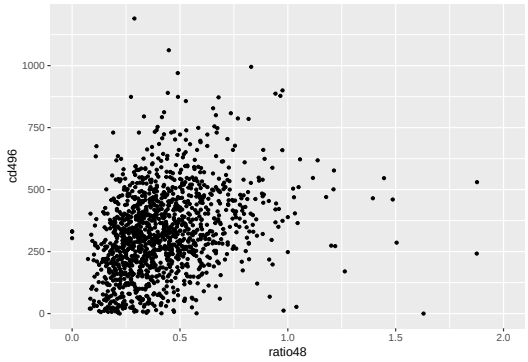
Figure 1: Descriptive Statistics



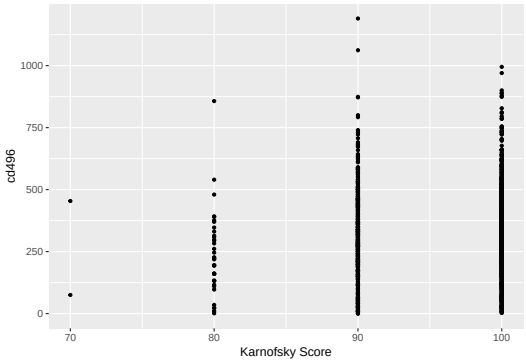
(a) Scatterplot against Age



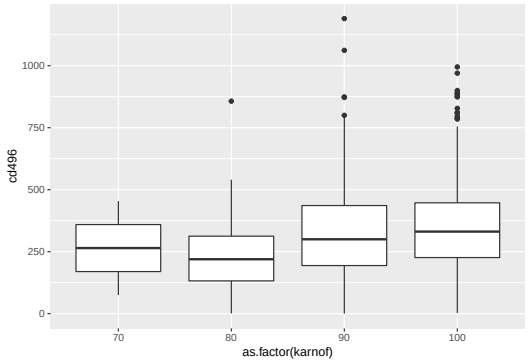
(b) Scatterplot against Weight



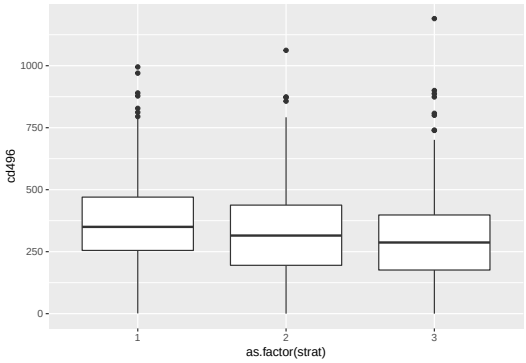
(c) Scatterplot against Ratio48



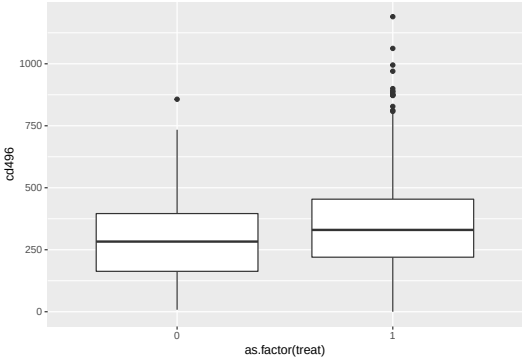
(d) Scatterplot against Karnofsky Score



(e) Boxplot against Karnofsky Score



(f) Scatterplot against Strat



(g) Scatterplot against Treatment

From scatter plot of cd496 against age, we do not see there is an association between cd496 and age. We will test their association in section ….. Robust nonparametric test will be applied since there are many outliers in the scatter plot.

The scatter plot of cd496 against weight have bell shape. This implies that we should make use of non-linear regression, such as KNN regression, smoothing splines and local regression since we prefer non-parametric method.

The scatter plot of cd496 against cd40/cd80 indicates a strong correlation. The possible reason is that the ratio of CD4 and CD8 represent the status of the war between immune system and virus. High ratio means that the immune system takes the advantage, while low ratio means that the virus takes the advantage. For convenience, we also adopt non-parametric non-linear regression methods as well.

Considering both scatter plot and box-plot of cd496 against Karnofsky score(omitting the '70' Karnofsky score reasonably), we see that cd496 increases as the score increasing reasonably since the score represents the health level of patients.

From box-plot of cd496 against stratification of antiretroviral history, we see that cd496 decreases as the history becoming longer. This may be the result of drug resistance.

From box-plot of cd496 against the treatment taken, we see that treatment with two nucleosides performs better than with a single nucleoside. We use step function to fit the qualitative predictors discussed above.

Non-Parametric Inference for Age and Ratio48

To test the association between CD496 and Age, we apply Spearman's ρ test. We perform this test in R using following codes(PLAIN CODES BLOCK IN THE END!!!!) and obtain the following result.

```
##
##  Spearman's rank correlation rho
##
## data:  Age and cd496
## S = 410670546, p-value = 0.4754
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.01950063
```

We see that the p-value is 0.4754, which is greater than 0.1. We also use this test to test if there is a quadratic association between CD496 and Age.

```
##
##  Spearman's rank correlation rho
##
## data:  Age2 and cd496
## S = 399372670, p-value = 0.7544
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## 0.008546647
```

The result is the same, since the p-value is 0.7544.

Therefore we conclude that there is no association between CD496 and Age, and thus in the following model we will not consider Age as a predictor.

Then we test the correlation between cd496 and ratio48(cd40/cd80). We perform Spearman's ρ test similarly.

```
##
```

```
##  Spearman's rank correlation rho
##
## data:  Ratio48 and cd496
## S = 264641620, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3430201
```

The p-value is very small, which indicates that the correlation is very highly significant.

Since the scatter plot of CD496 against weight appears a bell shape, we do not test the association here.

Select λ of Smoothing Spline for Weight and Ratio48 via Cross Validation(LOOCV)

The tuning parameter λ is needed when fitting a smoothing spline function g, which minimizes

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

To achieve this goal, cross validation method(Leave One Out CV is applied here) is applied on our data set ACTG175. We select the λ which minimizes the cross-validated RSS

$$RSS_{cv}(\lambda) = \sum_{i=1}^{n}[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}}]^2,$$

and then the resulted effective degree of freedom

$$df_\lambda = \sum_{i=1}^{n}\{S_\lambda\}_{ii}$$

The calculation and fitting is performed in R. The plot of smoothing spline is shown in figure 2.

```
## $SplineLambda
## [1] 1.565362
##
## $EffectDf
## [1] 2.453652
```

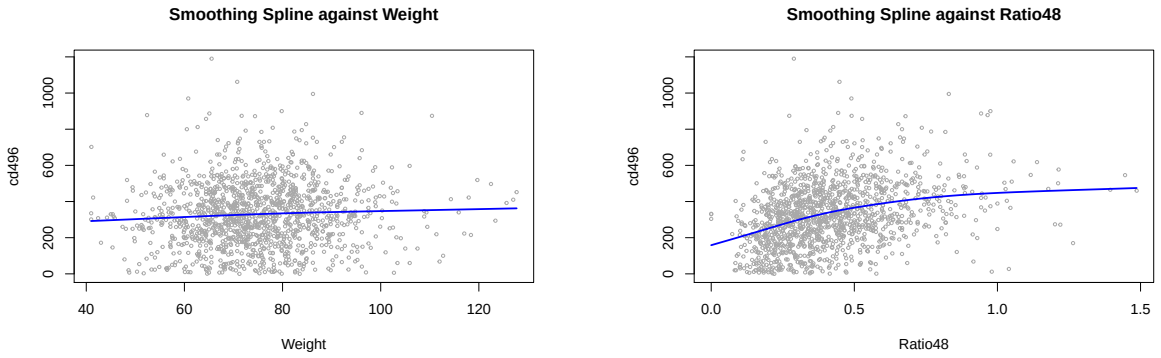
We obtain $\lambda = 1.57$ and desired effective degree of freedom $df_\lambda = 2.45$ and the corresponding smoothing spline model for CD496 and Weight. From the model plot we see that CD496 increases as Weight increasing, but it increases slowly when the Weight is large, that is the positive slope of CD496 becomes smaller as the Weight increasing.

We exclude the Ratio48 outliers which greater than 1.5 and then a similar procedure is performed between CD496 and Ratio48 in R. The plot of smoothing spline is also shown in figure 2.

```
## $SplineLambda2
## [1] 0.1070041
##
## $EffetDf2
## [1] 3.842323
```

We obtain $\lambda = 0.11$ and effective degree of freedom $df_{\lambda 2} = 3.84$ and the corresponding smoothing spline model for CD496 and Ratio48. From the model plot we see that CD496 increases as Ratio48 increasing, but it increases slowly when the Ratio48 is large.

Figure 2: Smoothing Spline



Estimate the standard deviation of estimated λ via Bootstrap

A bootstrap method can be used to estimate the deviation of our estimated λ, which quantify uncertainty, in previous section.

We bootstrap $B = 50$ sample from the original data set. Then estimate $\lambda_i, i = 1, 2, ..., B$ applying the same procedure via LOOCV in each sample. Finally we calculate the estimate of standard deviation

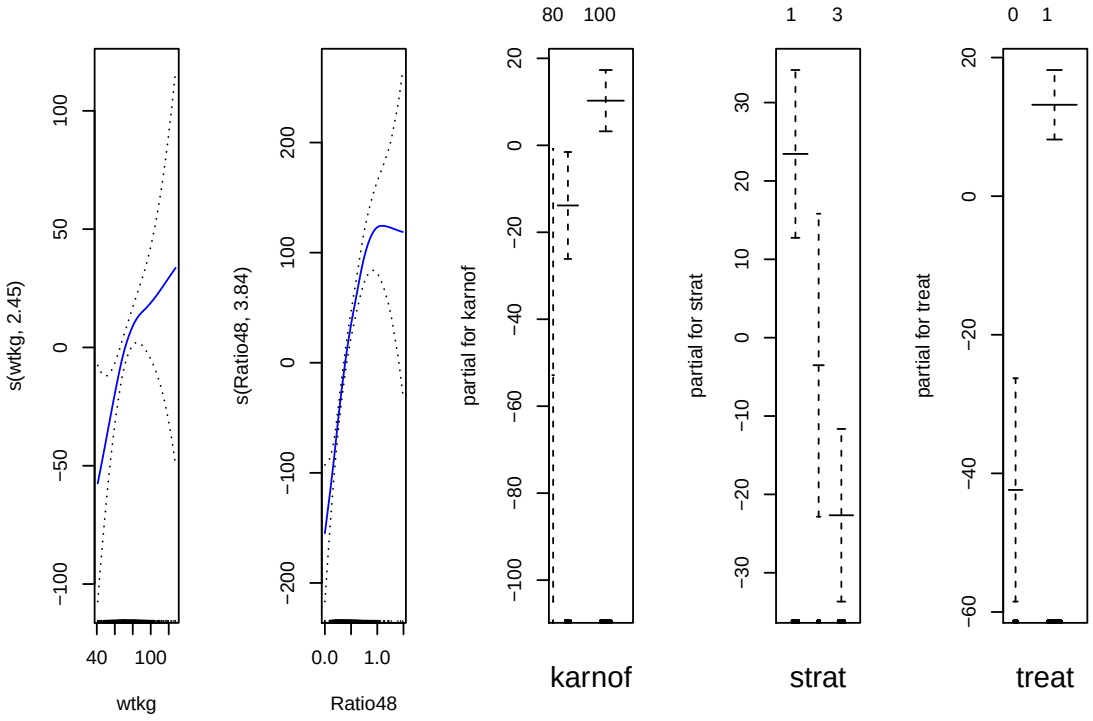
$$SE_B(\lambda) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\lambda_i - \frac{1}{B}\sum_{i'=1}^{B}\lambda_{i'})^2}$$

We perform this procedure in R.

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = actg, statistic = LambdaWt, R = 50)
##
##
## Bootstrap Statistics :
##     original    bias    std. error
## t1* 1.565362 28.225551   27.044426
## t2* 2.453652  0.432203    2.105956
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = actg, statistic = LambdaRat, R = 50)
##
##
## Bootstrap Statistics :
##       original     bias    std. error
## t1* 0.02265201 0.02861003  0.04673397
## t2* 4.44174372 2.54511159 12.34368339
```

Figure 3: GAM



The t1* represents the λ, and the t2* represents the Effective degree of freedom. From the result we see that the for CD496 against Weight, the standard deviation of λ is large, while of Effective DF is small. For CD496 against Ratio48, the standard deviation of λ is small, while of Effective DF is large. Thus for data of CD496 against Weight, Effective DF is recommended as a parameter of smoothing spline, while for data against Ratio48, λ is recommended as a parameter of smoothing spline.

GAM

We next consider both 4 predictors, wtkg(Weight), karnof(Karnofsky score), strat(antiretroviral history stratification), treat(zidovudine treatment only or other therapies)), Ratio48(the ratio between CD4 and CD8 at baseline) interpretive and predictive model. The wtkg and Ratio48 is quantitative variable, while other 3 are qualitative.

We apply Generalized Additive Model(GAM). We fit 3 qualitative variables using dummy variables, and a smoothing spline model for quantitative variables wtkg and Ratio48, applying the effective degree of freedom $df_\lambda = 2.45$ and $df_{\lambda 2} = 3.84$ obtained in the previous section for this model. We perform this procedure in R and plot main effect functions for each predictor in figure 3. For later use, we sample 9/10 of original data as a training set.

The first plot indicates that holding other variables fixed, the CD496 has a positive correlation with weight. Note that for the very low and high weight, where the sample size is small, the confidence band is wide.

The second plot indicates that holding other variables fixed, the CD496 has a positive correlation with Ratio48. For Ratio48 greater than 1 points, the slope becomes small and the confidence band is wide because of small sample size.

The third plot indicates that holding other variables fixed, CD496 increases with Karnofsky score. This is a

intuitive finding since the Karnofsky score implies the health level of patients.

The forth plot indicates that holding other variables fixed, CD496 decreases as the times of antiretroviral therapy increasing. The possible reason is drug resistance. Antiretroviral therapy may reduce effects of subsequent therapies.

The fifth plot indicates that holding other variables fixed, CD496 of those adopt zidovudine treatment only is less than those adopt treatment with two nucleosides. This finding is consistent with the conclusion of the referred study[1].

We also try to use GAM to predict CD496. We make use of the 1/10 observations that are not in the training set to perform a prediction. The MSE for test data, which indicates a standard deviation of approximate 150, is large. This implies that based on the values of predictors at baseline, it's hardly to predict CD4 cell count of a patient 96 weeks later.

```
## $MSE.GAM.Test
## [1] 23097.44
```

Regression Tree

Since there are 3 qualitative predictor variable in our model and tree based method can easily handle qualitative predictor, we also build regression tree model for our interpret and predict purpose.

We build a regression tree that each leaf region predicts the response using the mean of training observations and the branch nodes are divided by minimizing squared error

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$
$$R_1(j,s) = \{X | X_j < s\}, R_2(j,s) = \{X | X_j \geq s\},$$

recursively. Then we obtain a large tree. Next we tune the parameter α and minimize the adjusted sum of square error

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m(j,s)} (y_i - \hat{y}_{R_m})^2 + \alpha |T|,$$

to get a sequence of sub-trees. We use cross validation method to choose the best α and the corresponding sub-tree. We perform these procedure in R.

First we construct a default complexity parameter tree, shown in figure 4(a).

From the default tree we see that Ratio48(Ratio of CD4 and CD8 at baseline) is far more important than other factors in determining CD496. Patients who have higher Ratio48 have higher CD496. This result is consistent with our finding in section …… where we use scatter plot and GAM.

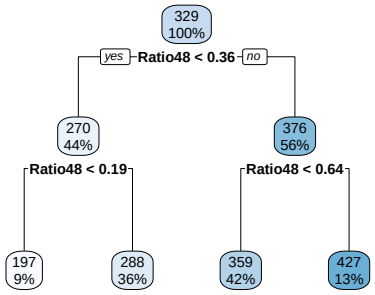
Other factors, such as weight and Karnofsky score, are omitted in the automatically generated tree. This implies that these 2 factors are less important.

We can set the complexity parameter cp= $\alpha = 0.004$ instead of default value 0.01 and obtain a more complex tree to study the influence of other factors. The tree is shown in figure 4(c).

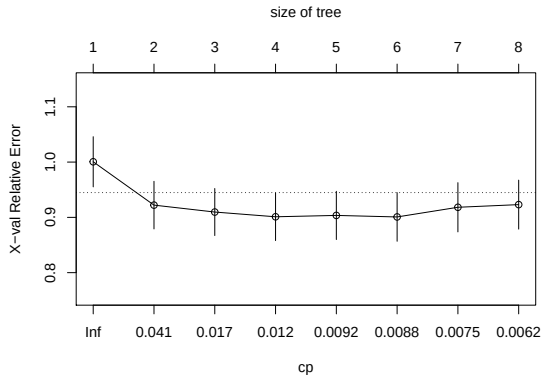
We can interpret this decision tree intuitively. For example, given that Ratio48 less than 0.36 and greater than 0.19, stratification of antiretroviral history plays a second role in his CD496. If he accept antiretroviral therapy before, the predicted CD496 would be 324, otherwise Treatment Choossing plays a third role in his CD496.

This cool tree may be meaningless, since a complex tree may cause overfitting. We then will choose a best cp based on 5 fold cross validation RSS_{cv_5} and prune the tree based on the cp we select.

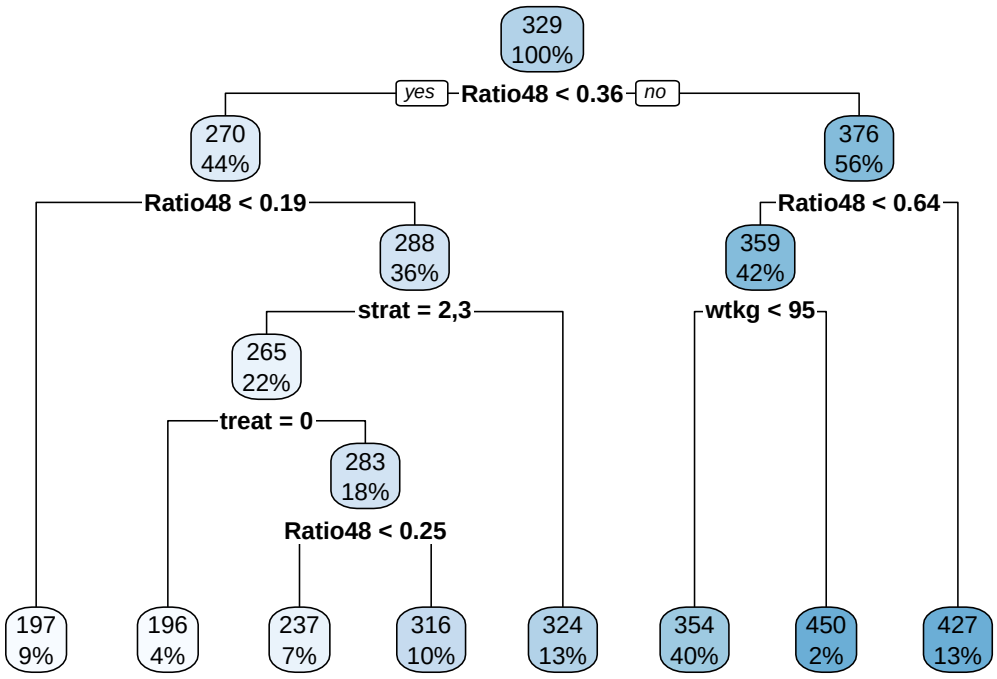
Figure 4: Finding a Best Tree



(a) Default Tree



(b) RSS against Complexity Parameter



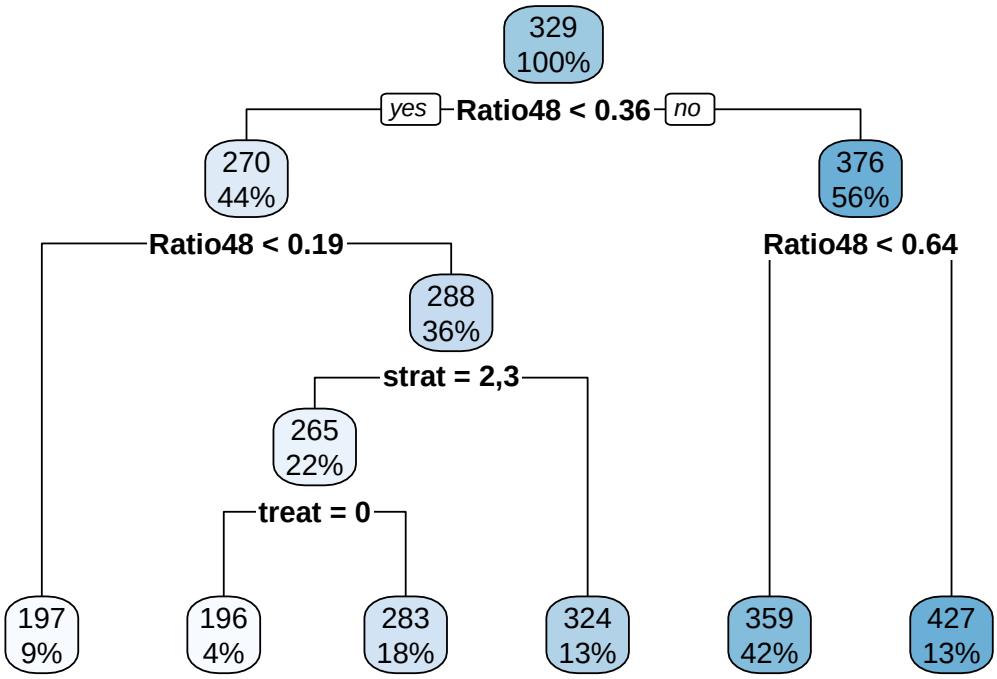
(c) Complex Tree

We plot RSS_{cv_5} against cp and find the cp we desired. The plot is shown in figure 4 (b).

```
## $min.cp
## [1] 0.008776257
```

Then prune the tree based on cp=0.0088. Note that some node are pruned. This is the best interpretive and predictive regression tree, shown in figure 5.

Figure 5: Best Regression Tree



Random Forest

To build a more efficient predictive model, we consider a random forest model rather than a single regression tree. The accuracy of prediction increases at the expense of interpretability.

For a random forest, we bootstrap `ntree` samples. The algorithem grow tree in each sample, selecting `mtry` predictors as split candidates whenever split a node. We try different values of `mtry`$\in \{1, 2, 3, 4, 5\}$ and for each try, we set the number of bootstrap `ntree` sufficiently large to stable the MSE and the `nodesize`, number of leaves, equals to 6, which we obtain in the previous section. Note that when \texttt= 5, the random forest is equal to bagging. Then we select the `mtry` which minimizes MSE. We select `mtry`=1 consequently.

This procedure is performed in R. We sample 9/10 of the original data as a training data, while other as a test data.

```
##
## Call:
##  randomForest(formula = CD496 ~ ., data = actg, ntree = 100, mtry = 1,importance = TRUE,
##  nodesize = 6, subset = train)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 26507.5
##                     % Var explained: 12.34
```

```
## $MSE.Test.RF1
## [1] 28908.45
```

The random forest's accuracy of prediction is worse than GAM's, having a standard deviation of 170 approximately. This implies again that based on the predictors at baseline, CD4 cell count of a patient is hard to predict 96 weeks later.

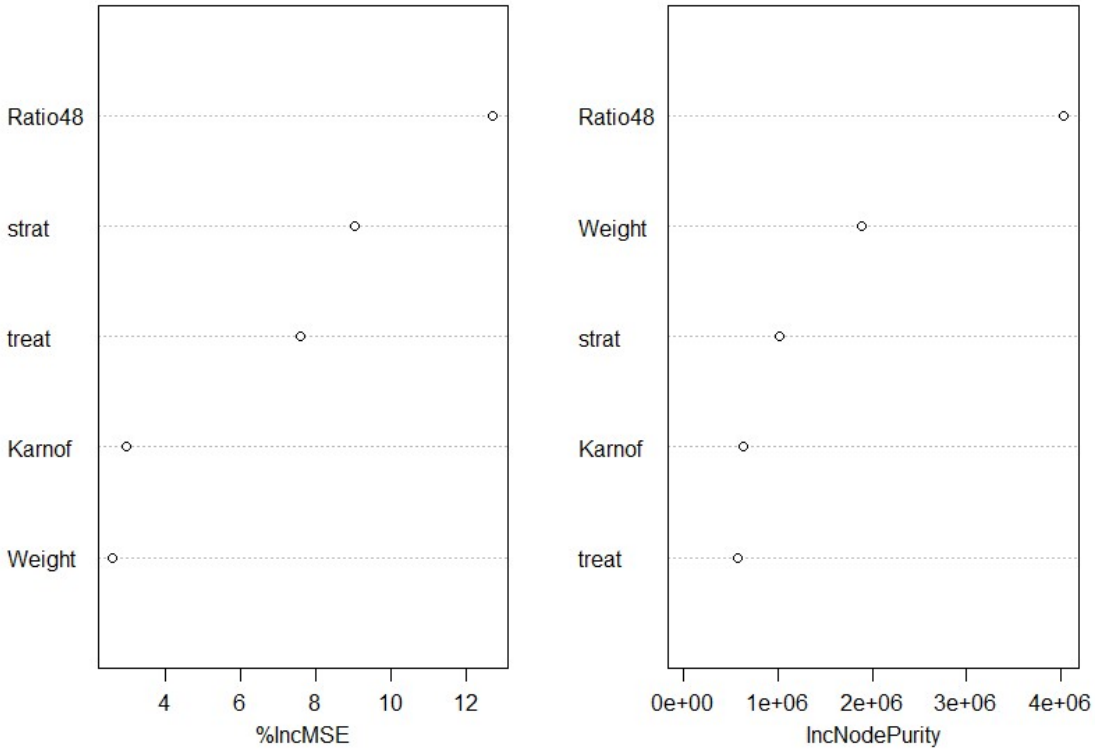
The performance of bagging is even worse than random forest, having a standard deviation of 180 approximately.

```
##
## Call:
##  randomForest(formula = CD496 ~ ., data = actg, ntree = 100, mtry = ncol(actg) - 1,
##  importance = TRUE, nodesize = 6, subset = train)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 29849.64
##                     % Var explained: 1.29
```

```
## $MSE.Test.Bagging
## [1] 31417.91
```

However, the advantage of random forest compared to GAM is that it can indicates the importance of different predictors in determining CD496. We illustrate the importance we discussed in figure 6.

Figure 6: Importance of Predictors in Determining CD496 in RandomForest1



From the plot we see that in decreasing the MSE, Ratio48 is the most important predictor, and `strat` and `treat` follow.

Conclusion

Our conclusions come from the 3 main model we built, that is GAM, Regression Tree, Random Forest.

The predictions of GAM and Random Forest perform bad. This implies that it's difficult to predict CD4 cell count at 96 week after baseline based on the data at baseline. We also try to apply same methods to predict CD4/CD8 at 20 weeks after baseline, which performs well, having a deviation of 0.3(not discussed in article). But this is meaningless, since CD4/CD8 at 20 weeks after baseline is mainly determined by CD4/CD8 at baseline but not other factors such as treatment, weight, or Karnofsky score.

Models also indicate the importance and effect of predictors in determining CD4 cell count at 96 week after baseline. CD4/CD8 at baseline, which represents the status of the war between immune system and virus at the beginning of study, plays the most important role. Those who have higher Ratio48 at baseline tend to have higher CD4 count at 96 weeks. Then treatment taken(that is treatment with zidovudine only or with two nucleosides) and stratification of history of antiretroviral therapy play second roles. Treatment with two nucleosides performs better than with zidovudine only, and who are naive to antiretroviral therapy have higher CD496. Karnofsky score follows. Those who have Karnofsky score higher have higher CD496. Finally patient's weight is the least important. All of these findings are reasonable.

References

[1] M. A. Fischl, D. D. Richman, and M. H. Grieco. A trial comparing nucleoside monotherapy with combination therapy in hiv-infected adults with cd4 cell counts from 200 to 500 per cubic millimeter — nejm. *Lancet*, 1996.

Appendix

Codes

Descriptive Statistics

```
#warning off
library(ggplot2)
library(speff2trial)
data("ACTG175")
# age
ggplot(ACTG175,aes(x=age,y=cd496))+geom_point(size=1)+labs(x='Age')
# weight
ggplot(ACTG175,aes(x=wtkg,y=cd496))+geom_point(size=1)+labs(x='Weight')
# cd4/cd8 at baseline
ratio48=ACTG175$cd40/ACTG175$cd80
cd496=ACTG175$cd496
datatemp=data.frame(ratio48,cd496)
ggplot(datatemp,aes(x=ratio48,y=cd496))+geom_point(size=1)+
  xlim(0,2)

# karnofsky score
ggplot(ACTG175,aes(x=karnof,y=cd496))+geom_point(size=1)+labs(x='Karnofsky Score')
ggplot(ACTG175,aes(x=as.factor(karnof),y=cd496))+geom_boxplot()
# stratification of antiretroviral history
ggplot(ACTG175,aes(x=as.factor(strat),y=cd496))+geom_boxplot()
# treatment taken
ggplot(ACTG175,aes(x=as.factor(treat),y=cd496))+geom_boxplot()
```

Non-Parametric Test

```
# warning off
library(speff2trial)
data("ACTG175")
actg=ACTG175
# Test Association between cd496 and Age
Age=actg$age;cd496=actg$cd496
TestAge=cor.test(Age,cd496,alternative = 'two.sided',
                 method = 'spearman',conf.level = 0.95)
print(TestAge)

Age2=(Age-mean(Age))^2
TestAge2=cor.test(Age2,cd496,alternative = 'two.sided',
                 method = 'spearman',conf.level = 0.95)
print(TestAge2)

# Error: bell shape can not use spearman rho
# Test Association between cd496 and Weight
# Weight=actg$wtkg
# TestWt=cor.test(Weight,cd496,alternative = 'two.sided',
#                  method = 'spearman',conf.level = 0.95)
# print(TestWt)

# Test Association between cd496 and Ratio48
```

```r
Ratio48=actg$cd40/actg$cd80
TestRat=cor.test(Ratio48,cd496,alternative = 'two.sided',
                 method = 'spearman',conf.level = 0.95)
print(TestRat)
```

Smoothing Spline

```r
# warning off

rm(list = ls())

library(speff2trial)
library(splines)
data("ACTG175")

# Clean data
actg=ACTG175
actg=actg[-which(is.na(actg$cd496)),]

# Against Weight
# Smoothing spline fit, using LOOCV to determine lambda
SplineCV=smooth.spline(actg$wtkg,actg$cd496,cv=TRUE)
print(list(SplineLambda=SplineCV$lambda,EffectDf=SplineCV$df))

# Plot results
wtlims=range(actg$wtkg)
par(mfrow=c(1,1))
plot(cd496~wtkg,data = actg,xlim=wtlims, cex=.5,col='darkgrey',
     main = 'Smoothing Spline against Weight',xlab='Weight')
lines(SplineCV,col='blue',lwd=2)

# Against Ratio48
# Smoothing spline fit, using LOOCV to determine lambda
Ratio48=actg$cd40/actg$cd80
# Clean the ratio48 outlier
cd496=actg[-which(Ratio48>1.5),'cd496']
Ratio48=Ratio48[-which(Ratio48>1.5)]

SplineCV2=smooth.spline(Ratio48,cd496,cv=TRUE)
print(list(SplineLambda2=SplineCV2$lambda,EffetDf2=SplineCV2$df))

# Plot results
Ratlims=range(Ratio48)
par(mfrow=c(1,1))
plot(Ratio48,cd496,xlim=Ratlims,cex=.5,col='darkgrey',
     main = 'Smoothing Spline against Ratio48')
lines(SplineCV2,col='blue',lwd=2)
```

Bootstrap

```r
# Bootstrap

# warning off
```

```r
rm(list = ls())

library(boot)
library(speff2trial)
library(splines)
data("ACTG175")

# Clean data
actg=ACTG175
actg=actg[-which(is.na(actg$cd496)),]
Ratio48=actg$cd40/actg$cd80
Weight=actg$wtkg
CD496=actg$cd496
actg=data.frame(CD496,Weight,Ratio48)

LambdaWt<-function(dat,indices){
  # Against Weight
  # Smoothing spline fit, using LOOCV to determine lambda
  SplineCV=smooth.spline(dat$Weight[indices],dat$CD496[indices],cv=TRUE)
  SplineLambda=SplineCV$lambda
  EffectDf=SplineCV$df
  c(SplineLambda,EffectDf)
}

LambdaRat<-function(dat,indices){
  # Against Ratio48
  # Smoothing spline fit, using LOOCV to determine lambda
  SplineCV2=smooth.spline(dat$Ratio48[indices],dat$CD496[indices],cv=TRUE)
  SplineLambda2=SplineCV2$lambda
  EffectDf2=SplineCV2$df
  c(SplineLambda2,EffectDf2)
}

set.seed(1)
BootWt=boot(actg,LambdaWt,R=50)
BootRat=boot(actg,LambdaRat,R=50)

print(BootWt)
print(BootRat)
```

GAM

```r
rm(list = ls())
set.seed(2)

library(speff2trial)
data("ACTG175")
library(gam)
#clean data
actg=ACTG175
Ratio48=actg$cd40/actg$cd80
actg=data.frame(actg,Ratio48)
```

```r
actg=actg[-which(is.na(actg$cd496)),]
actg=actg[-which(actg$Ratio48>1.5),]
actg=actg[-which(actg$karnof==70),] #remove the very few observation

actg$karnof=as.factor(actg$karnof)
actg$strat=as.factor(actg$strat)
actg$treat=as.factor(actg$treat)

train=sample(1:nrow(actg),nrow(actg)/10*9)

# GAM
gam.m1=gam(cd496~s(wtkg,2.45)+s(Ratio48,3.84)+karnof+strat+treat,
           data = actg, subset = train)
par(mfrow=c(1,5))
plot(gam.m1, se=TRUE,col="blue")

test=actg$cd496[-train]
preds=predict(gam.m1,newdata=actg[-train,])
print(list(MSE.GAM.Test=mean((preds-test)^2)))
```

Regression Tree

```r
# Regression Tree
# based on package rpart

rm(list=ls())

library(rpart)
library(rpart.plot)

library(speff2trial)
data("ACTG175")
#clean data
ACTG175c=ACTG175[-which(ACTG175$karnof==70),] #remove the very few observation
attach(ACTG175c)
karnof=as.factor(karnof)
strat=as.factor(strat)
treat=as.factor(treat)
Ratio48=cd40/cd80
actg2=data.frame(
  cd496,
  wtkg,
  karnof,
  strat,
  treat,
  Ratio48
)

actg2=actg2[-which(is.na(actg2$cd496)),]

# Use package rpart to construct a tree

# The rpart automatically set comlexity parameter as 0.01
```

```r
actg2.rpart=rpart(cd496~.,data = actg2)
# Plot the tree
par(mfrow=c(1,1))
rpart.plot(actg2.rpart)

# Set control parameter of rpart,
# lowest cp=0.004, 5 fold cross validation
contr=rpart.control(cp=0.006,xval=5)
# The rpart performs 5 fold cross validation and store results
set.seed(14)
actg2.rpart=rpart(cd496~.,data = actg2,control = contr)
# Plot the tree
rpart.plot(actg2.rpart)

# The cv result is stored as a form of
# different complexity parameter alpha and SSE
plotcp(actg2.rpart)
# Find the best cp
min.cp=actg2.rpart$cptable[which.min(actg2.rpart$cptable[,'xerror']),'CP']
print(list(min.cp=min.cp))

# plot pruned tree
actg2.rpart.pruned=prune(actg2.rpart,cp=min.cp)
rpart.plot(actg2.rpart.pruned)
```

Random Forest

```r
# Random Forest
# Based on package randomForest

rm(list=ls())

library(randomForest)
set.seed(1)

library(speff2trial)
data("ACTG175")
#clean data
actg=data.frame(
  ACTG175$cd496,
  ACTG175$wtkg,
  as.factor(ACTG175$karnof),
  as.factor(ACTG175$strat),
  as.factor(ACTG175$treat),
  ACTG175$cd40/ACTG175$cd80
)
colnames(actg)<-c('CD496','Weight','Karnof','strat','treat','Ratio48')
actg=actg[-which(is.na(actg$CD496)),]

train=sample(1:nrow(actg),nrow(actg)/10*9)
actg.test=actg[-train,'CD496']
#1
RF1.actg=randomForest(CD496~.,data = actg,subset = train, ntree=100,
```

```
                      mtry = 1, importance = TRUE,
                      nodesize = 6)
print(RF1.actg)
yhat.RF1=predict(RF1.actg,newdata = actg[-train,])
print(list(MSE.Test.RF1=mean((yhat.RF1-actg.test)^2)))

#Bagging
bag.actg=randomForest(CD496~.,data = actg,subset = train, ntree=100,
                      mtry = ncol(actg)-1, importance = TRUE,
                      nodesize = 6)
print(bag.actg)
yhat.bag=predict(bag.actg,newdata = actg[-train,])
print(list(MSE.Test.Bagging=mean((yhat.bag-actg.test)^2)))

# plot(yhat.RF1,actg.test)
# abline(0,1)

varImpPlot(RF1.actg,main = 'RandomForest1.ACTG')

# Test for other random forest for different values of mtry
# #2
# RF2.actg=randomForest(CD496~.,data = actg,subset = train, ntree=100,
#                       mtry = 2, importance = TRUE,
#                       nodesize = 6)
# print(RF2.actg)
# yhat.RF2=predict(RF2.actg,newdata = actg[-train,])
# mean((yhat.RF2-actg.test)^2)
# #3
# RF3.actg=randomForest(CD496~.,data = actg,subset = train, ntree=200,
#                       mtry = 3, importance = TRUE,
#                       nodesize = 6)
# print(RF3.actg)
# yhat.RF3=predict(RF3.actg,newdata = actg[-train,])
# mean((yhat.RF3-actg.test)^2)
# #4
# RF4.actg=randomForest(CD496~.,data = actg,subset = train, ntree=100,
#                       mtry = 4, importance = TRUE,
#                       nodesize = 6)
# print(RF4.actg)
# yhat.RF4=predict(RF4.actg,newdata = actg[-train,])
# mean((yhat.RF4-actg.test)^2)
```