Dichotomizing continuous predictors

Anna Gudjonsdottir, Ali Charlson, and Megan Aadland

May 8, 2019

Introduction

A very common problem that biostatisticians run into is when medical researchers want to transform continuous variables into categorical variables. In clinical settings such as doctor's appointments or research studies using patient participation, many patient related variables are recorded as continuous. For example, age, height, and weight are all continuous variables when directly recorded from patients in a clinician's office. Once this patient data is collected, clinicians like to convert many of the continuous variables into categorical variables. While these decisions do not always make sense in the statistical setting, they do make sense in a clinical setting.

Medical decision making is entirely based on yes/no decision making. All of their patient level decisions are made with the baseline question of "do I treat this person or do I not treat this person?" This results in the reliance on categorical data in the medical community. Many medical researchers may be inexperienced with continuous variables because of how common categorical variables are used, so this results in more and more medical researchers continuing to use categorical variables even when the settings may not be completely appropriate in the statistical sense.

While it is very common for medical researchers to use these dichotomization methods, statisticians do not like to arbitrarily alter their data when not necessary. It is a very unnatural process from a statistical standpoint to divide variables up into categories with no real mathematical or statistical explanations behind the decisions. Additionally, dividing data up like this can result in statistical analysis being incorrect or uninterpretable. From a technical perspective, there are a couple advantages of this transformation, but overall the disadvantages far outweigh the advantages.

The two advantages of this continuous to categorical process are that most statistical tests for categorical data are somewhat easier to perform, and the interpretation of these tests is much simpler. However, despite these advantages, the disadvantages are numerous. The most important and influential disadvantage that results from data dichotomization is the significant loss of information that comes from using continuous variables as categorical. It results in an unrealistic characterization of the data that can overall lead to lower power, increased probability of type I error rates, issues detecting non-linearity, residual confounding, spurious associations, and increased bias. Despite these many disadvantages, continuous variables are still frequently divided into categorical variables by medical researchers.

There are a variety of methods used to divide up continuous data. All of them are used very subjectively. While a paper examining a certain type of medical research might use one method, another similar paper might use a completely different method for no real reason. It is difficult to determine when any of the most common data dichotomization methods are supposed to be used. The most frequently used methods are: setting the data cutpoint categories by the sample median, setting the categories in multiples of five or ten, running an optimization on the data to determine the optimal cut point (either minimum or maximum) and then basing the rest of the data dichotomization around that cutpoint, and using previous scientific knowledge and research to determine clinically relevant cut points. In addition to being arbitrarily used, all of these methods result in significant problems. The median cut point method, optimal cut point method, and multiples of five or ten cut point method are all performed subjectively, thus if cut points are chosen incorrectly the rest of the analysis will be flawed. Using statistics such as the median, minimum, or maximum will complicate the study and the results will not be able to be compared across other studies and larger populations. And finally, the main issue with all of the current methods is that researchers commonly derive these cutpoints and data dichotomization methods using univariate analyses, and then try to generalize those

cutpoint results to multivariate analyses. This results in incorrect information being used as a basis for the entire data set, and again results in flawed statistical analysis.

The statistical issues with converting continuous variables into categorical variables are lengthy and relevant, however this does not stop medical researchers from continuing to use this process. Dichotomous decision making is a required part of medical treatments, so regardless of the statistical concerns, it is important to examine these methods to determine how the dichotomization methods compare and contrast, and under what settings these methods could be used. The goal of this analysis is to analyze and discuss the median cutpoint, optimal cutpoint, and natural spline methods, then compare those methods with a true continuous model, and try do determine the conditions and analyses these methods should be used for. We will analyze these methods through simulation, model building, and diagnostic analysis.

Methods

To analyze possible methods to find a statistically accurate way to dichotomize continuous variables, we compared various predicted models to true models based on simulated data. Our first step is to perform three separate simulations that generate logistic regression data based on a true model of our specification. The three simulations were repeated 1,000 times on a true continuous age predictor, a median cutpoint indicator predictor for age, and a non-central cutpoint indicator for predictor age. Within each simulation, we will fit four models to be evaluated at two different signal strengths: one using the continuous predictors, one using the median cutpoint, one using the optimal cutpoint, and one using the natural spline. From these models, we will generate a variety of diagnostic criteria that can be used to evaluate the final model fits and choose the best method.

The equations for the four models are as follows:

1) Keeping continuous variables

$$logit(Y) = \alpha + \beta_1(sex) + \beta_2(age)$$

2) Dichotomizing by median cutpoint

$$logit(Y) = \alpha + \beta_1(sex) + beta_2 * I(age < median(age))$$

3) Dichotomizing by optimal univariate cutpoint

$$logit(Y) = \alpha + \beta_1(sex) + \beta_2 * I(age < cutpoint(age))$$

4) Dichotomizing with splines

$$logit(Y) = \alpha + \beta_1(sex) + \beta_2 * (spline_1) + \beta_3(spline_2)$$

These four models are evaluated at a $\beta_1 = 0.1$ and $\beta_2 = 0.5$ at the strong signal and $\beta_2 = 0$ at the weak signal level. Because the natural spline has an extra parameter, we evaluated the deviance to determine of the anova table between the natural spline model and the null model.

For each model, we looked at the area under the ROC curve, pseudo R^2 and the percentage of the time that the age parameters were significance. The ROC curve is the true positive rate against the false positive rate. The area under that curve measures how well the parameter distinguishes between two groups. Pseudo R^2 explains the proportion of variance that the covariate explains of the latent variable, with higher values indicating better model fits. Finally, we evaluated what percentage of the time did the model correctly detect the effect of the variable. This is power under the strong signal and type I error under the weak signal.

Results

Creating a function to fit the models and format the results

```
fit_models <- function(data, mort, plot = TRUE, table = TRUE){

    ##Model 1: Continuous Linear
    f1 <- glm(mort ~ sex + age, family = binomial, data = data)
    summary(f1)

    ##Model 2: Median Cutpoint
    data$age_median <- ifelse(data$age < median(data$age), 0, 1)

    f2 <- glm(mort ~ age_median + sex, family = binomial, data = data)
    summary(f2)

    #Model 3: Optimal Cutpoint
    cp <- cutpointr(data, age, mort, method = minimize_metric, metric = p_chisquared)
    summary(cp)
    multi_cp <- cp$optimal_cutpoint[[1]]
    cut <- multi_cp[1]

    data$age_optimal <- ifelse(age < cut, 0, 1)
    f3 <- glm(mort ~ age_optimal + sex, family = binomial, data = data)
    summary(f3)

    ##Model 4: Natural Spline
    f4 <- glm(mort ~ ns(age, 2) + sex, family = binomial, data = data)
    summary(f4)

    f5 <- glm(mort ~ sex, family = binomial, data = data)
    avar <- anova(f5,f4)
    m3p <- pchisq(avar$Deviance[2], 2, lower.tail = FALSE)

    # Calculate Sensitivity and Specificity for Model1
    fity_ypos1 <- f1$fitted[mort == 1]
    fity_yneg1 <- f1$fitted[mort == 0]

    sort_fity1 <- sort(f1$fitted.values)

    sens1 <- 0
    spec_c1 <- 0

    for (i in length(sort_fity1):1){
        sens1 <- c(sens1, mean(fity_ypos1 >= sort_fity1[i]))
        spec_c1 <- c(spec_c1, mean(fity_yneg1 >= sort_fity1[i]))

    }

    # Calculate Sensitivity and Specificity for Model2
    fity_ypos2 <- f2$fitted[mort == 1]
    fity_yneg2 <- f2$fitted[mort == 0]
```

```r
sort_fity2 <- sort(f2$fitted.values)

sens2 <- 0
spec_c2 <- 0

for (i in length(sort_fity2):1){
    sens2 <- c(sens2, mean(fity_ypos2 >= sort_fity2[i]))
    spec_c2 <- c(spec_c2, mean(fity_yneg2 >= sort_fity2[i]))

}

# Calculate Sensitivity and Specificity for Model3
fity_ypos3 <- f3$fitted[mort == 1]
fity_yneg3 <- f3$fitted[mort == 0]

sort_fity3 <- sort(f3$fitted.values)

sens3 <- 0
spec_c3 <- 0

for (i in length(sort_fity3):1){
    sens3 <- c(sens3, mean(fity_ypos3 >= sort_fity3[i]))
    spec_c3 <- c(spec_c3, mean(fity_yneg3 >= sort_fity3[i]))

}

# Calculate Sensitivity and Specificity for Model4
fity_ypos4 <- f4$fitted[mort == 1]
fity_yneg4 <- f4$fitted[mort == 0]

sort_fity4 <- sort(f4$fitted.values)

sens4 <- 0
spec_c4 <- 0

for (i in length(sort_fity4):1){
    sens4 <- c(sens4, mean(fity_ypos4 >= sort_fity4[i]))
    spec_c4 <- c(spec_c4, mean(fity_yneg4 >= sort_fity4[i]))

}
# plot ROC curves
if (plot == TRUE){
  plot(spec_c1, sens1, xlim = c(0, 1), ylim = c(0, 1), type = "l",
       xlab = "False Positive Rate", ylab = "True Positive Rate", col = 'blue')
  abline(0, 1, col= "black")
  lines(spec_c2, sens2, col='green')
  lines(spec_c3, sens3, col='red')
  lines(spec_c4, sens4, col='orange')
  legend("topleft", legend = c("Continuous","Median", "Optimal", "Spline") , pch = 15, bty = 'n',
         col = c("blue","green", "red", "orange"))
}

# Model 1
```

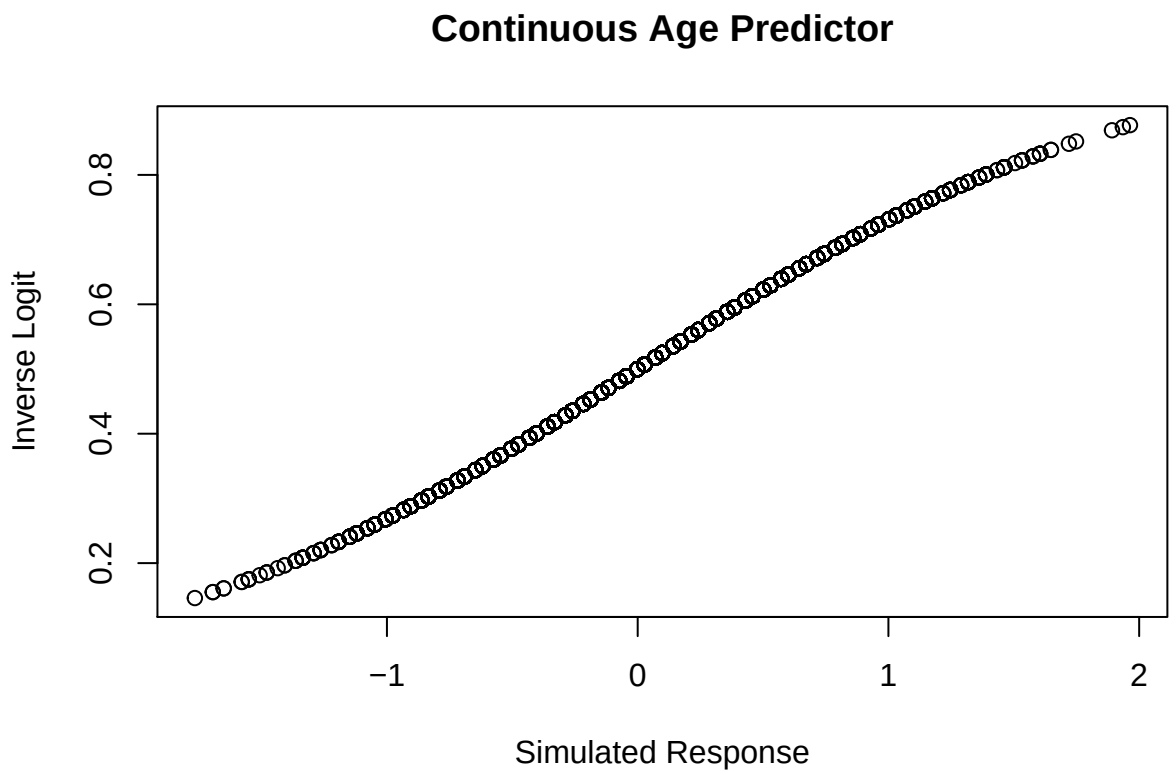
```
npoints1 <- length(sens1)
(area1 <- sum(0.5 * (sens1[-1] + sens1[-npoints1]) * (spec_c1[-1] -
        spec_c1[-npoints1])))
# Discrete approximation area under the curve, using Trapezoidal Rule

# Model 2
npoints2 <- length(sens2)
(area2 <- sum(0.5 * (sens2[-1] + sens2[-npoints2]) * (spec_c2[-1] -
        spec_c2[-npoints2])))

# Model 3
npoints3 <- length(sens3)
(area3 <- sum(0.5 * (sens3[-1] + sens3[-npoints3]) * (spec_c3[-1] -
        spec_c3[-npoints3])))

# Model 4
npoints4 <- length(sens4)
(area4 <- sum(0.5 * (sens4[-1] + sens4[-npoints1]) * (spec_c4[-1] -
        spec_c4[-npoints4])))


#Pseudo R^2
#model.1
FullcovModelRed <- glm(mort~1, family=binomial(link=logit))
r1 <- 1-(logLik(f1))/(logLik(FullcovModelRed))

#model.2
FullcovModelRed <- glm(mort~1, family=binomial(link=logit))
r2 <- 1-(logLik(f2))/(logLik(FullcovModelRed))

#model.3
FullcovModelRed <- glm(mort~1, family=binomial(link=logit))
r3 <- 1-(logLik(f3))/(logLik(FullcovModelRed))

#model.4
FullcovModelRed <- glm(mort~1, family=binomial(link=logit))
r4 <- 1-(logLik(f4))/(logLik(FullcovModelRed))

#p-value checks
pval1 <- ifelse(summary(f1)$coefficients[3,4] < .05, 1, 0)
pval2 <- ifelse(summary(f2)$coefficients[2,4] < .05, 1, 0)
pval3 <- ifelse(summary(f3)$coefficients[2,4] < .05, 1, 0)
pval4 <- ifelse(pchisq(avar$Deviance[2], 2) < .05, 1, 0)
#pval4 <- ifelse(summary(f4)$coefficients[2, 4] < .05 &
#                  summary(f4)$coefficients[3, 4] < .05, 1, 0)



#Table


return(model_summary <- data.frame("Continuous" = c(round(area1,3), round(f1$deviance,3),
                                          round(f1$aic, 3), round(r1, 3), pval1),
```

```
                                        "Median" = c(round(area2, 3), round(f2$deviance, 3),
                                                     round(f2$aic, 3), round(r2, 3), pval2),
                                        "Optimal" = c(round(area3, 3), round(f3$deviance, 3),
                                                     round(f3$aic, 3), round(r3, 3), pval3),
                                        "Spline" = c(round(area4, 3), round(f4$deviance, 3),
                                                     round(f4$aic, 3), round(r4, 3), pval4)))

    rownames(model_summary) <- c("Area Under ROC Curve", "Deviance",
                                                "AIC","Pseudo R^2", "P-Value")

    if (table == TRUE){
      kable(model_summary,
            align = c("l", "c", "c"),
            col.names = c("Continuous", "Median", "Optimal", "Spline 4"))%>%
        kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                      full_width = FALSE)

    }
}


#Formatting the Results

format_simulation <- function(list, signal = TRUE){
  Sim_1_result <- Reduce(`+`, list) / length(list)

  if (signal == TRUE){
    rownames(Sim_1_result) <- c("Area Under ROC Curve", "Deviance",
                                "AIC","Pseudo R^2", "Power")
  }else{
    rownames(Sim_1_result) <- c("Area Under ROC Curve", "Deviance",
                               "AIC","Pseudo R^2", "Type I Error")
  }

  table1 <- Sim_1_result[-c(2:3), ]


  kable(table1,
        align = c("l", "c", "c"),
        col.names = c("Continuous", "Median", "Optimal", "Spline"))%>%
      kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                    full_width = FALSE)

}
```

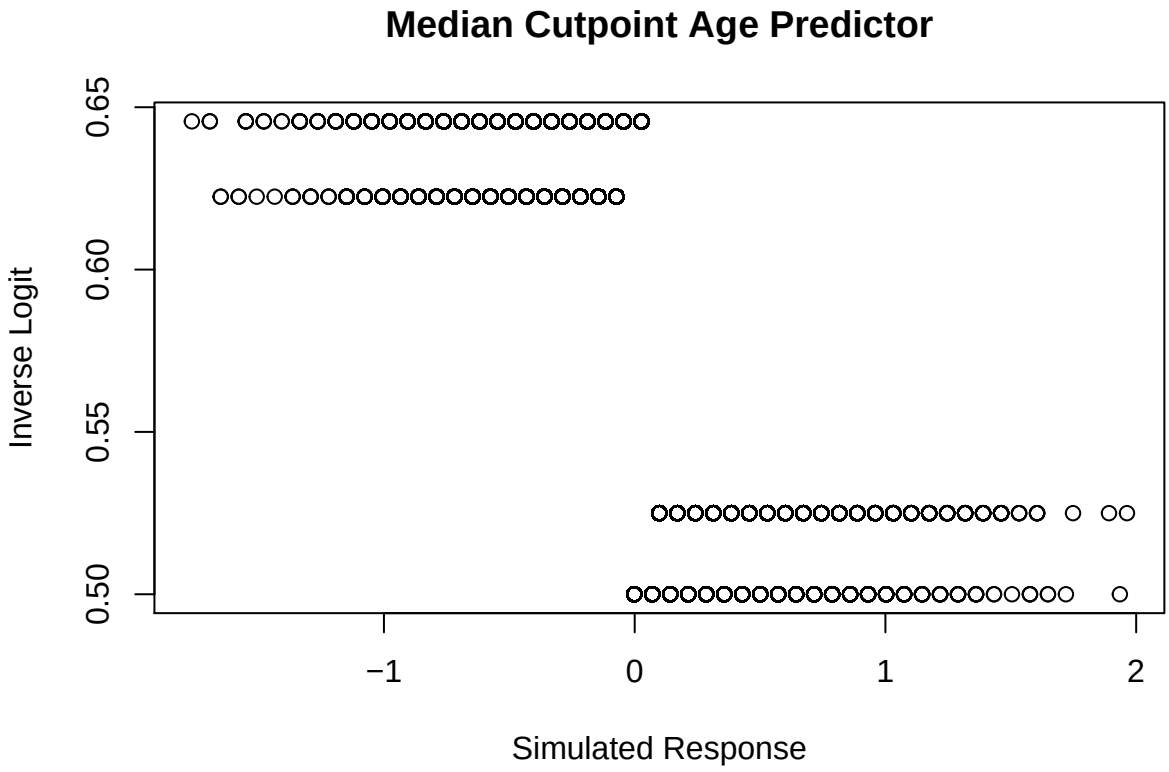
Data Visualization

```
set.seed(888)
options(scipen = 999)

a1 <- .5
b1 <- .1
```

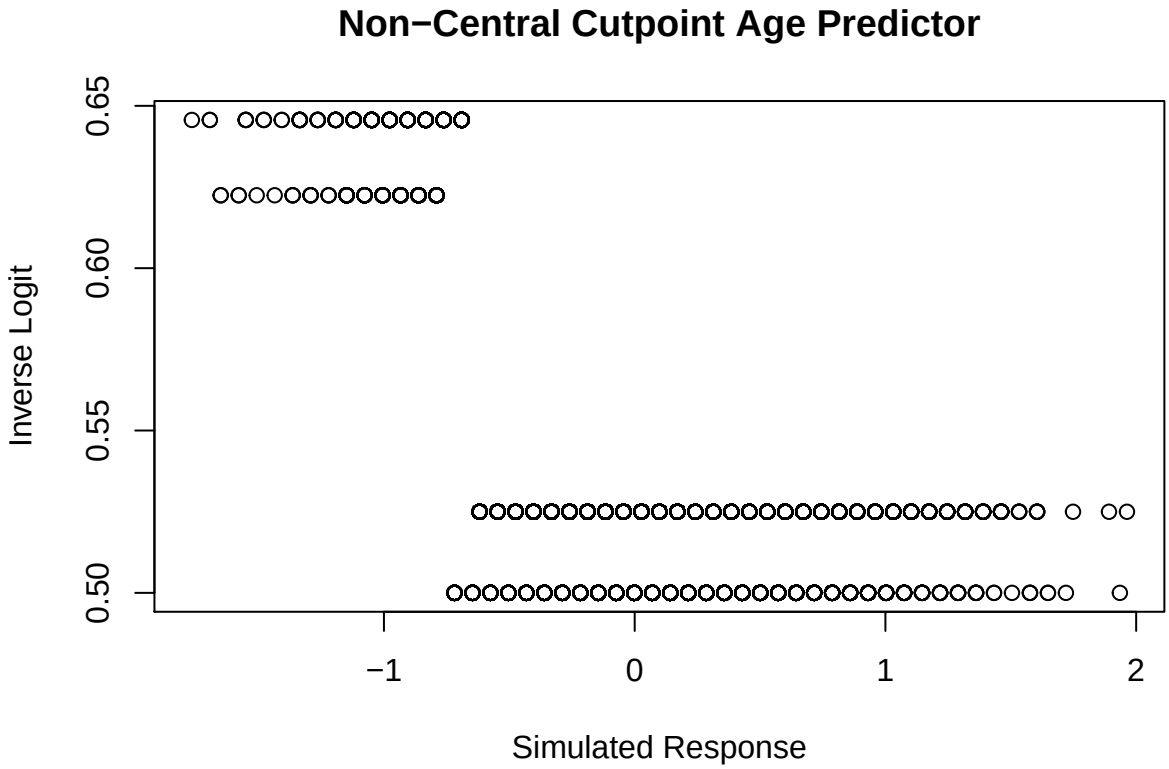
```
age <- abs(round(rnorm(n = 10000, mean = 45, sd = 7)))
age <- standardize(age)
sex <- rbinom(n = 10000, size = 1, prob = 0.7)
z <- a1 * age + b1* sex
z1 <- a1 * (age<median(age)) + b1 * sex
z2 <- a1 * (age<(median(age) - 1.5)) + b1 * sex
pr <- 1 / (1 + exp(-z))
pr1 <- 1 / (1 + exp(-z1))
pr2 <- 1 / (1 + exp(-z2))
mort <- rbinom(10000, 1, pr)
mort1 <- rbinom(10000, 1, pr1)
mort2 <- rbinom(10000, 1, pr2)
data <- data.frame(age, sex, mort)
data1 <- data.frame(age, sex, mort1)
data2 <- data.frame(age, sex, mort2)

y <- inv.logit(z)
plot(y ~ z, xlab = "Simulated Response", ylab = "Inverse Logit",
     main = "Continuous Age Predictor")
```




```
y <- inv.logit(z1)
plot(y ~ z, xlab = "Simulated Response", ylab = "Inverse Logit",
     main = "Median Cutpoint Age Predictor")
```




```
y <- inv.logit(z2)
plot(y ~ z, xlab = "Simulated Response", ylab = "Inverse Logit",
     main = "Non-Central Cutpoint Age Predictor")
```



These first three plots allow us to visualize the simulated data on a single simulation. The first simulation has

a continuous linear age predictor, while the second two simulations have a simulated median and non-central cutpoint.

Simulating the Data

```
#Strong Signal
set.seed(888)

a1 <- .5
b1 <- .1

n.rep = 1000

s1DF <- list(NA)
s2DF <- list(NA)
s3DF <- list(NA)


for (i in 1:n.rep){
  age <- abs(round(rnorm(n = 1000, mean = 45, sd = 7)))
  age <- standardize(age)
  sex <- rbinom(n = 1000, size = 1, prob = 0.7)
  z <- a1 * age + b1* sex
  z1 <- a1 * (age<median(age)) + b1 * sex
  z2 <- a1 * (age<(median(age) - 1.5)) + b1 * sex
  pr <- 1 / (1 + exp(-z))
  pr1 <- 1 / (1 + exp(-z1))
  pr2 <- 1 / (1 + exp(-z2))
  mort <- rbinom(1000, 1, pr)
  mort1 <- rbinom(1000, 1, pr1)
  mort2 <- rbinom(1000, 1, pr2)
  data <- data.frame(age, sex, mort)
  data1 <- data.frame(age, sex, mort1)
  data2 <- data.frame(age, sex, mort2)

  s1DF[[i]] <- fit_models(data = data, mort = mort, plot = FALSE, table = FALSE)
  s2DF[[i]] <- fit_models(data = data1, mort = mort1, plot = FALSE, table = FALSE)
  s3DF[[i]] <- fit_models(data = data2, mort = mort2, plot = FALSE, table = FALSE)

}

#Weak Signal

s4DF <- list(NA)
s5DF <- list(NA)
s6DF <- list(NA)

a2 <- 0

for (i in 1:n.rep){
  age <- abs(round(rnorm(n = 1000, mean = 45, sd = 7)))
  age <- standardize(age)
  sex <- rbinom(n = 1000, size = 1, prob = 0.7)
```

```
  z <- a2 * age + b1* sex
  z1 <- a2 * (age<median(age)) + b1 * sex
  z2 <- a2 * (age<(median(age) - 1.5)) + b1 * sex
  pr <- 1 / (1 + exp(-z))
  pr1 <- 1 / (1 + exp(-z1))
  pr2 <- 1 / (1 + exp(-z2))
  mort <- rbinom(1000, 1, pr)
  mort1 <- rbinom(1000, 1, pr1)
  mort2 <- rbinom(1000, 1, pr2)
  data <- data.frame(age, sex, mort)
  data1 <- data.frame(age, sex, mort1)
  data2 <- data.frame(age, sex, mort2)

  s4DF[[i]] <- fit_models(data = data, mort = mort, plot = FALSE, table = FALSE)
  s5DF[[i]] <- fit_models(data = data1, mort = mort1, plot = FALSE, table = FALSE)
  s6DF[[i]] <- fit_models(data = data2, mort = mort2, plot = FALSE, table = FALSE)
}
```

Simulation 1: Continuous

Strong Signal

```
format_simulation(s1DF)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.637	0.603	0.571	0.637
Pseudo R^2	0.044	0.029	0.022	0.044
Power	1.000	1.000	0.998	0.000

Weak Signal

```
format_simulation(s4DF, signal = FALSE)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.525	0.524	0.529	0.529
Pseudo R^2	0.002	0.002	0.005	0.003
Type I Error	0.053	0.049	0.431	0.039

The true model is based on a continuous, linear age predictor in this first simulation. We can see that when the true model is continuous and there is a true effect of age, the continuous predictor performed better in all metrics. When we had no effect of age, the type I error of all of the models was around 0.05 with the exception of the optimal cutpoint model. We see that the optimal cutpoint model has severely inflated type I error. In this case, it is the best to keep the predictor continuous.

Simulation 2: Median Cutpoint

Strong Signal

```
format_simulation(s2DF)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.565	0.569	0.569	0.567
Pseudo R^2	0.009	0.013	0.013	0.010
Power	0.864	0.968	0.985	0.000

Weak Signal

```
format_simulation(s5DF, signal = FALSE)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.525	0.524	0.529	0.529
Pseudo R^2	0.002	0.002	0.005	0.003
Type I Error	0.050	0.053	0.448	0.046

The second simulation has the true age entering as an indicator cutpoint chosen near the median. In this case, the optimal and median cutpoint models appear to perform the best on the simulated data by all metrics. The median cutpoint has slightly lower power than the optimal cutpoint; however, we see the same problem of a highly inflated type I error when there is a weak signal. The continuous model has a power above 80% and still performs fairly well when the true data have a median cutpoint. Therefore, a continuous model is recommended unless you know the true cutpoint of your dataset.

Simulation 3: Non-Central Cutpoint

Strong Signal

```
format_simulation(s3DF)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.528	0.525	0.533	0.534
Pseudo R^2	0.002	0.002	0.006	0.004
Power	0.177	0.098	0.731	0.020

Weak Signal

```
format_simulation(s6DF, signal = FALSE)
```

	Continuous	Median	Optimal	Spline
Area Under ROC Curve	0.525	0.524	0.529	0.529
Pseudo R^2	0.002	0.002	0.005	0.003
Type I Error	0.051	0.048	0.428	0.047

In our final simulation, the true age enters as an indicator farther from median than our second simulation. The optimal cutpoint model was found to perform the best with the strong signal simulated data in terms of the power and pseudo R^2 diagnostics. However, we again see the issue of type I error with the weak signal, indicating that we would see a false positive in over 40% of simulations. Based on these results, the continuous predictor could still be considered as one of the best options, although it has low power to detect the age effect.

Conclusions

The simulation results show that the method used, such as keeping the predictor continuous, using an optimal cutpoint, a median cutpoint, or using natural splines, can greatly impact the performance of the model. All metrics showed that keeping the predictors continuous is the best way to handle data that is naturally continuous. For the data simulated with a median cutpoint and a non-central cutpoint, the optimal cutpoint model performed the best. However, this method produced a high percentage of false positives. When the data was simulated so that age did not have an effect on the response, the optimal cutpoint model still identified age as a significant predictor. The "best" method will depend on the nature of the underlying data. However, keeping predictors continuous is the best method unless the data is known to have a true cutpoint.

The study in the paper by Royston et al. was set up differently than our case study. The study used a multivariable fractional polynomial as the continuous model, while our simulation study used logistic regression. The multivariable fractional polynomial is fit using an algorithm that determines important variables and their functional form. This model preserves the continuous nature of the predictors and works for non-linear relationships. The case study in the paper did not use any natural spline models, but the paper did mention the technique and it was included in our simulation study. Additionally, the study in the paper had several continuous variables while our models consisted of only one. The data used in the study is from a clinical trial regarding primary biliary cirrhosis of the liver, while our data is simulated.

Despite these differences, both case studies reached the same conclusion. In the case study from the paper, the median cutpoint model performed the worst, explaining the least variance in the data. The optimal cutpoint explained 6% more of the variance in the data and the multivariable fractional polynomial explained 31% more. Therefore, the model that left the predictors continuous performed the best.

The methods presented for finding cutpoints and handling continuous predictors are not exhaustive. Optimal cutpoints can be found by maximizing or minimizing a variety of metrics including Cohen's Kappa, an odds ratio, relative risk and a variety of metrics related to sensitivity and specificity. Both the simulation study and the case study in the paper used the p-value of the chi-squared test to find the optimal cutpoints. If using optimal cutpoints, the cutpoint should be found in the multivariable setting rather than the univariate setting in order to adjust for other factors, minimize bias, and improve power. Additionally, validation methods, in which the cutpoint is determined independently of the patients, should be implemented. Two popular validation techniques are the split sample and cross validation approaches. In the split sample approach the cutpoint is determined from half of the data and used to classify patients in the other half. However, a more powerful approach is two-fold cross validation, which allows all of the data to be used. In two-fold cross validation, an optimal cutpoint is found in each half of the data and used to classify patients in the other subset of the data.

So far, uncertainty in how to select cutpoints has led researchers to use median and optimal cutpoints. However, optimal cutpoints inflate type I error rates, bias the parameter estimates and deflate the variance estimates. Although validation techniques and finding the cutpoint in the multivariable setting could improve the use of cutpoints, none of the authors of any of the referenced papers actually recommends using optimal cutpoints. Overall, clinicians should avoid dichotomization during the model building process. Categorization should instead be done on the prognostic index.

References

Royston, P., Altman, D. and Sauerbrei, W. (2005). Dichotomizing continuous predictors in multiple regression: a bad idea. Statistics in Medicine, 25(1), pp.127-141.

Zhang, Z. (2016). Model building strategy for logistic regression: purposeful selection. Annals of Translational Medicine, 4(6), pp.111-111.