Smoothing splines and modelling: tricep skin thickness and cancer mortality case studies

Cimmaron Yeoman

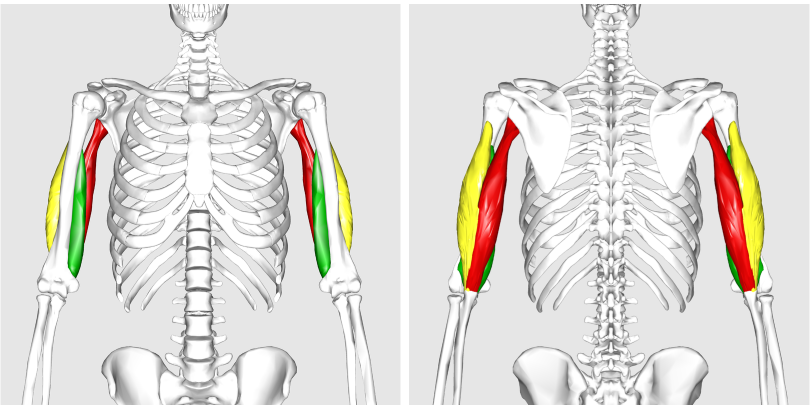
2023-03-17

Part 1: Smoothing and Triceps Data

The first section of this report evaluated the tricep skinfold thickness of 892 Gambian women in three West African villages, over 50 years. The *triceps brachii* is a muscle situated on the back of the the arm, and includes a lateral, long, and medial head (see Figure 1). The data set includes the tricep skinfold thickness measurements of the Gambian women, the log of the tricep skinfold thickness measurements, and the age of the women. The tricep skinfold thickness was predicted as a function of age, using a smoothing model.

Figure 1

Triceps brachii anterior and posterior view



Note. This figure highlights the heads of the *triceps brachii* on the human body. The long head was highlighted in red, the lateral head in yellow, and the medial head in green. These images provide an anterior and posterior view of the *triceps brachii* muscles on the human skeleton.

Triceps brachii data

Data set and missing observations

The triceps data set (**triceps2**) was evaluated for any missing variables. There were no missing observations and all cases were complete. The data set was renamed to **triceps2**. This will be the working data set for Part 1 of this report.

```
triceps2 <- triceps_ds
triceps2 <- triceps2[complete.cases(triceps2), ]
```

Visualizing the data set: scatter plots

Figure 2

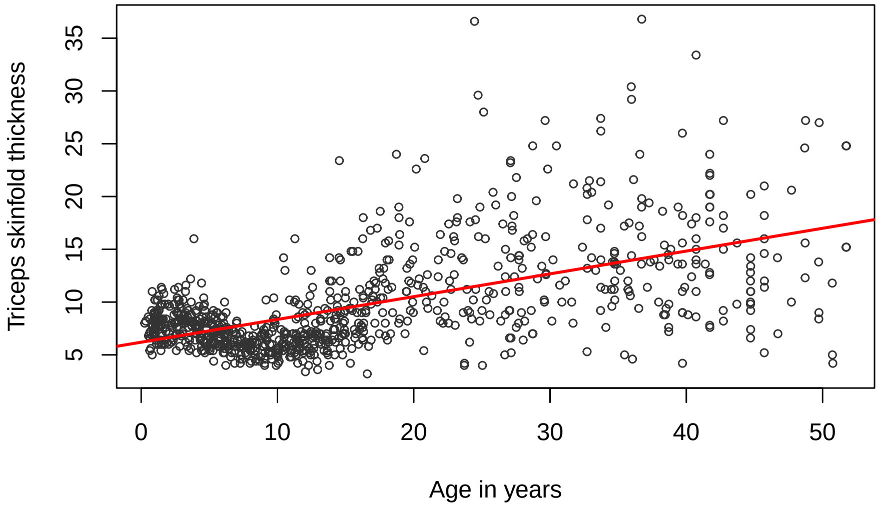
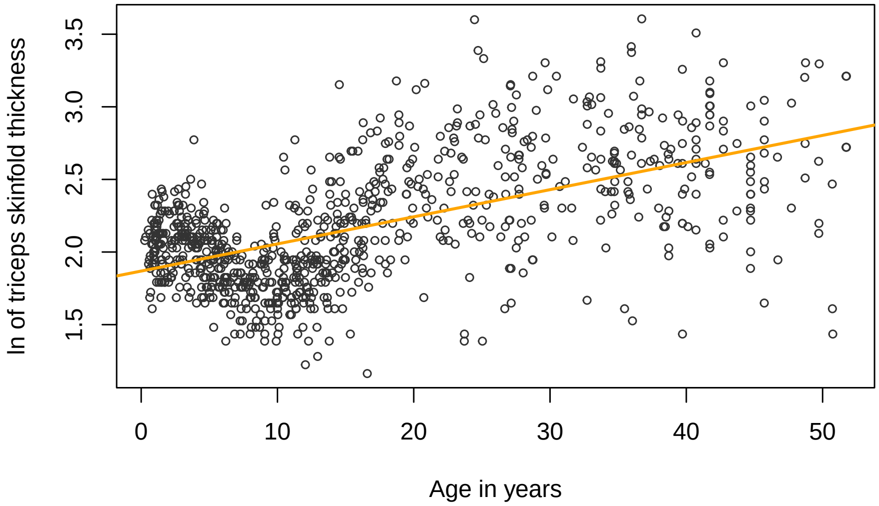


Figure 3



Both the **triceps** and **lntriceps** (log of tricep skinfold thickness) did not fit a simple linear regression well. When plotted with **age** on the x-axis, both variables showed distinct curves and a non-linear pattern.

Smoothing spline models: tricep skin thickness and age

A model was made using the smooth.spline function for both the tricep skinfold thickness and the log of tricep skinfold thickness. Cross-validation was used to determine the equivalent degrees of freedom.

```
library(splines)
mod_ss <- smooth.spline(triceps2$age, triceps2$triceps, cv = TRUE)
mod_ss_ln <- smooth.spline(triceps2$age, triceps2$lntriceps, cv = TRUE)
```

Smooth spline plots: tricep skin thickness or log of tricep thickness and age

Figure 4

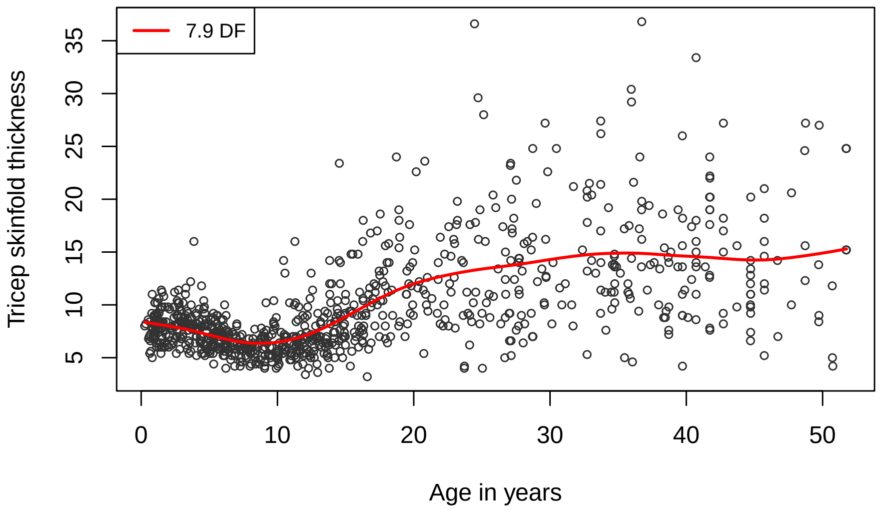
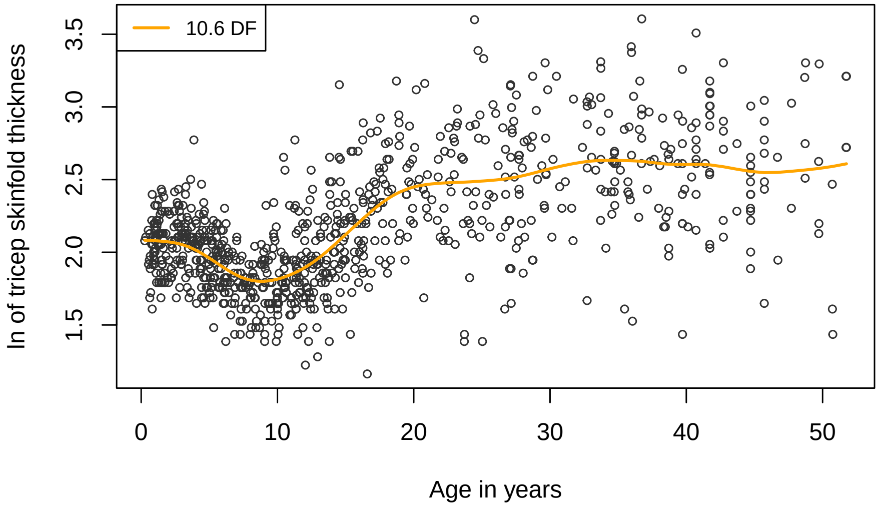


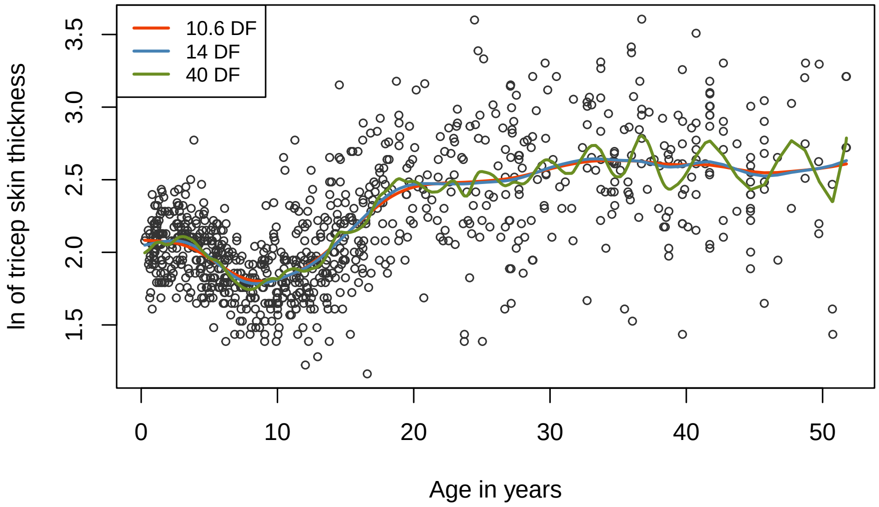
Figure 5



Using the smooth.spline function with cross-validation included, the smoothing parameter for the **triceps** ~ **age** model had an equivalent degrees of freedom of **7.9**, and the **lntriceps** ~ **age** model had an equivalent degrees of freedom of **10.6**. Lambda was about **0.005** for the first model and **0.001** for the second model; the curve for the first model was slightly smoother. Neither of the smoothing splines appeared to over or underfit the data.

Smooth splines with different *df* (log of tricep skin thickness)

Figure 6



Using the same smooth.spline function, different degrees of freedom were tested with the **lntriceps** ~ **age** model. As the degrees of freedom increased, the smoothing spline became more bumpy and the lambda values decreased to much smaller values. The Figure 6 *df* = **14** smoothing spline looked visually similar to the *df* = **10.6** smoothing spline in Figure 5. The *df* = **40** smoothing spline of Figure 6 was very rough and clearly overfitting the data.

Smoothing splines with different *df* (tricep skin thickness)

Figure 7

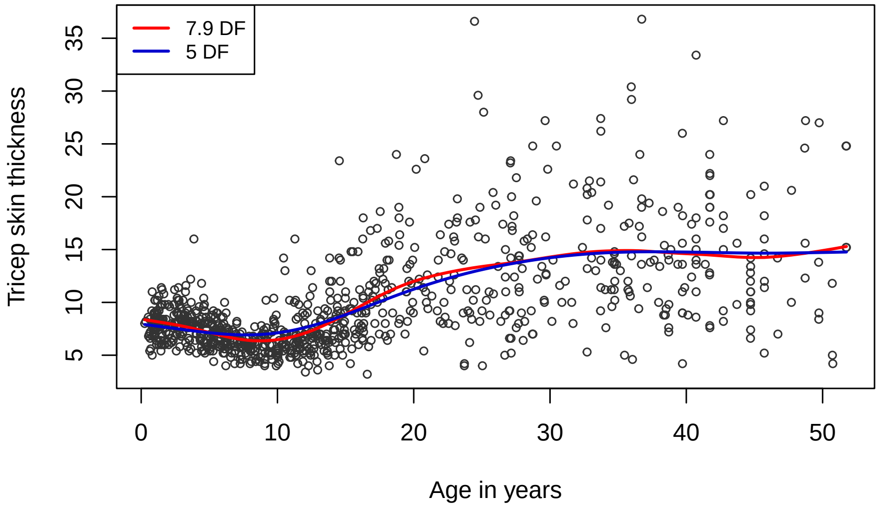
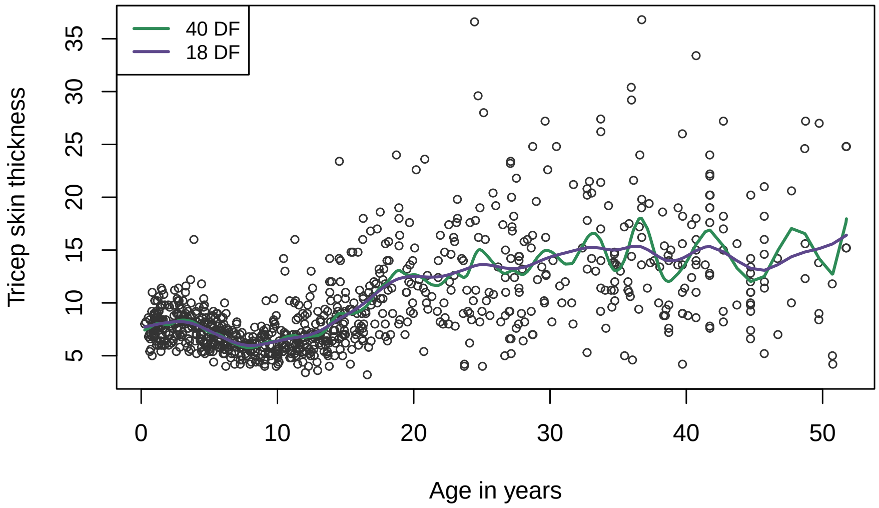


Figure 8



Similar to the plots with **lntriceps**, the higher the *df*, the rougher the smoothing splines for **triceps** ~ **age** model appeared, with both splines in Figure 8 overfitting the data. In Figure 7, the blue *df* = 5 smoothing splines appeared to slightly underfit the data while the *df* = 7.9 smoothing spline had a better fit.

Conclusion

The smoothing splines selected that fit the **triceps** ~ **age** or **lntriceps** ~ **age** models showed the practicality of fitting splines to data sets, and the potential for their use in predictive models. Very high **df** smoothing splines seem to overfit data sets in most cases, even when they have clear s-shaped curves and bends.

Part 2: Cancer data set and lasso regression

Cancer is one of the leading causes of death in the United States and across the globe. The disease exists in various forms, and millions of people, both young and old, receive a cancer diagnosis each year. This report contains a United States focused data set, consisting of health and socioeconomic factors from 3047 counties. The response variable or variable of interest was **TARGET_deathRate**, the mean per capita (100,000) cancer mortalities.

The goal of this analysis was to identify relevant predictor variables that could be candidates for a smooth relationship. A model with smoothing was compared to a basic model. Both natural cubic splines and the smoothing.spline package were used. Variable selection was performed using the lasso regression method.

```
cancer_L <- read_csv("C:/Users/cimmy/Documents/2023WI-3561H/Assignment4/cancer_reg.csv",
    col_types = cols(Geography = col_skip()))
cancer_L$binnedInc <- as.factor(cancer_L$binnedInc)
```

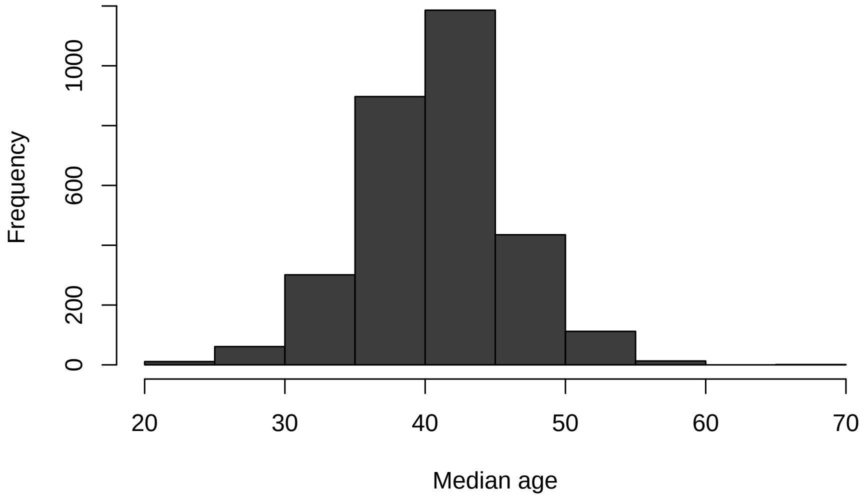
The data set was imported with the **Geography** or county name variable removed and the **binnedInc** or binned income variable included as a factor.

Removing outliers

The **MedianAge** variable had extreme values in the hundreds; to avoid some level of error in the data analysis, any values over 100 were removed. A median age of 100 would still be considered extreme and reviewing variable observations in future research is advised.

```
cancer_L$MedianAge[cancer_L$MedianAge > 100] <- NA
```

Figure 9



Note. This histogram shows the corrected distribution of **MedianAge** values after outlier values were removed.

Working data set

The final step before performing lasso regression was evaluation for missing observations. There were no missing observations detected, all cases were complete. The data set was renamed to **can_L**. This will be the working data set for Part 2 of this report.

```
can_L <- cancer_L[complete.cases(cancer_L), ]
```

Lasso regression

A test and train set were created from the **can_L** data set.

```
set.seed(56)
trainC <- sample(x = 1:584, size = 292, replace = FALSE)
testC <- (-trainC)
x <- model.matrix ( TARGET_deathRate ~ ., data = can_L) [, -1]
y <- can_L$TARGET_deathRate
```

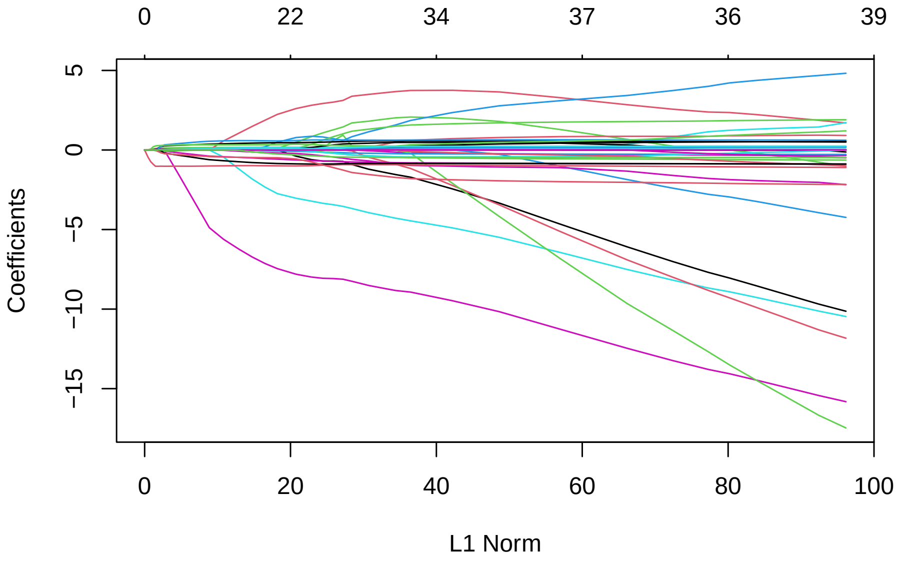
Testing a simple linear regression model

```
modC1 <- lm(TARGET_deathRate ~ ., data = can_L, subset = trainC)
summary(modC1)
modC1 <- summary(modC1)$adj.r.squared * 100
```

A basic model using the **trainC** subset had an adjusted R^2 of 50.4% variation explained with 32 predictor variables included. A majority of the variables did not appear to have a significant relationship with the response variable. Lasso regression will eliminate the less relevant variables before the smooth relationships are explored.

Creating a grid of lambdas

Figure 10



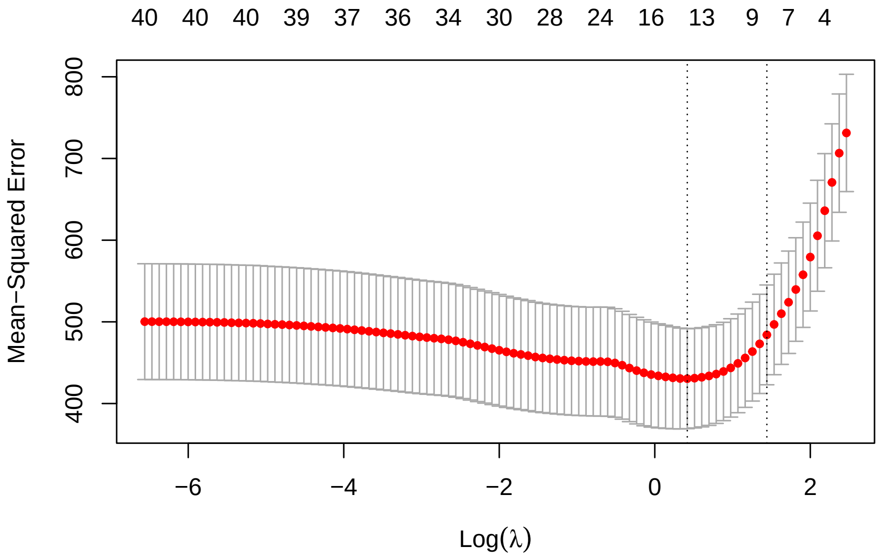
As the lambda values on the x-axis increase, the larger penalty forces the coefficients of the less relevant predictor to zero. This is the variable selection action of lasso, which will help identify the predictors variables for a model under a particular penalty value.

Selecting lambda

```
set.seed(76)
cv_outL <- cv.glmnet(x[trainC, ],
                     y[trainC],
                     alpha = 1)
lambda <- cv_outL$lambda.min
lambda
```

The minimum lambda was **1.52** after rounding.

Figure 11



The lambda fell between the the two dashed lines on the plot above: this is where the MSE of prediction was minimized.

Testing a model and calculating the MSE

```
modC_lasso <- glmnet(x[trainC, ], y[trainC], alpha = 1,
                     lambda = grid, thresh = 1e-10)
best_predL <- predict(modC_lasso, s = lambda, newx = x[testC, ])
mean((best_predL - y[testC])^2) #MSE calculation
```

When lambda was at **1.52** the MSE was **492.1**.

Eliminating variables

```
modC_lasso$lambda
coefficients(modC_lasso)[, 73]
```

The 73rd term of **1.519** was the most similar to the minimum lambda value of **1.52**. Many of the variables were zeroed and therefore could be eliminated from the subset selected for modelling.

Variable subset

```
which(abs(coefficients(modC_lasso)[, 73]) >= 1e-10)
```

The lasso regression identified 13 variables for the subset which could be used in modelling. From this list, **PctBlack**, **PctPrivateCoverage**, and **medIncome** were selected for the smooths and models.

Selecting variables to smooth

A scatter plot matrix showed that many of the predictor variables had a significant positive or negative linear relationship with the **TARGET_deathRate** response variable. Smoothing did not appear to be necessary for many variables, and doing so would likely overfit the data. The variables **PctBlack**, **PctPrivateCoverage**, and **medIncome** were selected, as the pattern they formed when plotted against the response variable showed slight bends or curves, potentially suitable for smoothing.

```
A <- seq(from = 0, to = 85)
x1 <- canF$PctBlack
y1 <- canF$TARGET_deathRate
mod_SSA <- smooth.spline(canF$PctBlack, canF$TARGET_deathRate, cv = TRUE)
smooth1 <- glm(y1 ~ ns(x1, df = 5))
```

This code chunk provides an example of the formulas used to create the models and splines. For each variable, a natural cubic spline and smoothing spline model was made.

Plotted natural cubic splines and smoothing splines

Figure 12

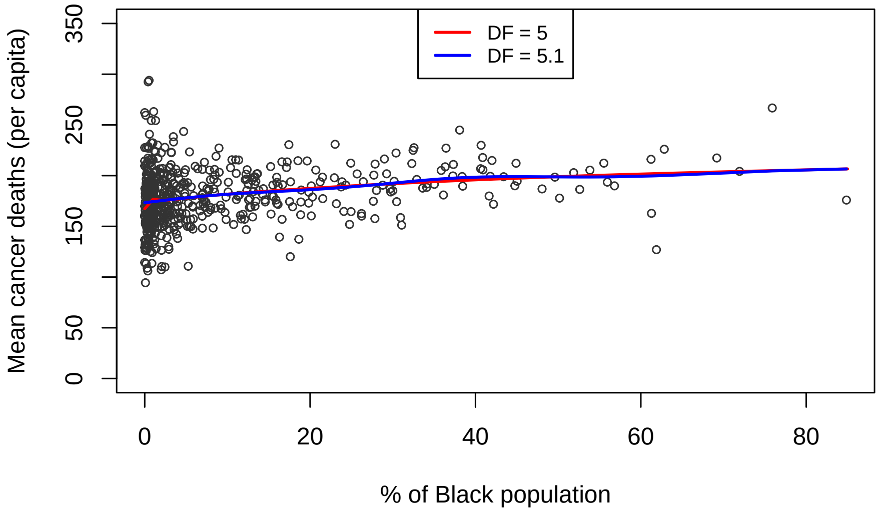


Figure 12 visually had a lot of scatter and a curved, s-shaped pattern that continued along the x-axis. A majority of the data was grouped between 0-10% on the x-axis. Fitting a spline with a higher degrees of freedom or specified knots along the s-curve would look more aesthetically pleasing, but this would overfit the data. The s-shaped data points did not appear dense enough to justify a rough smoothing spline. The grouping of the data was somewhat unusual; perhaps there were some missing observations for the **PctBlack** variable in this data set.

Figure 13

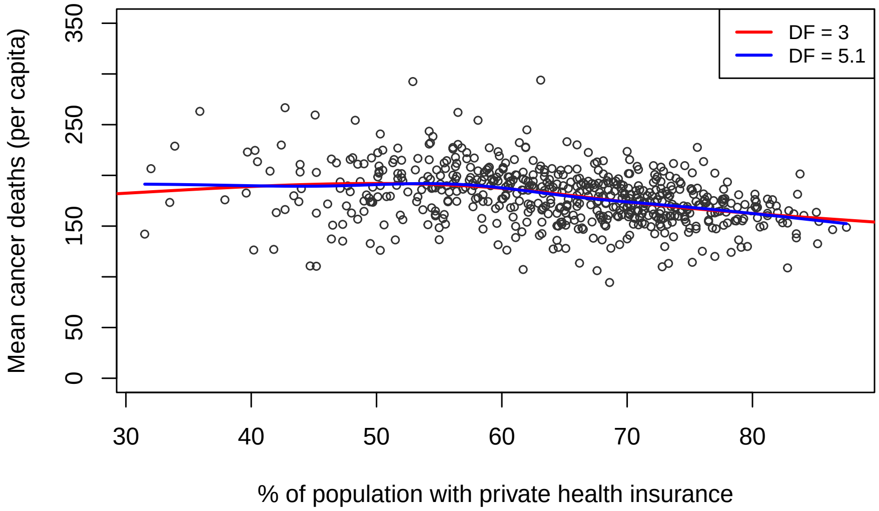
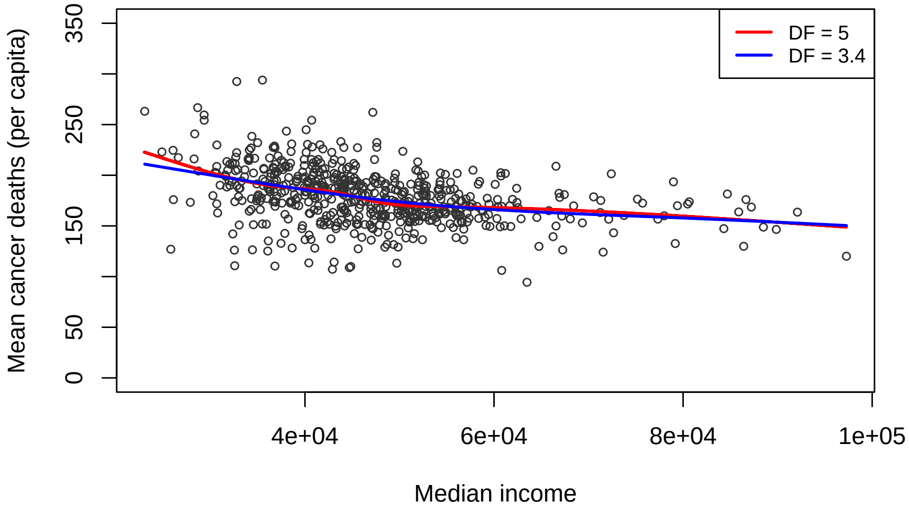


Figure 13 appeared to have an l-shaped bend where the cancer deaths peaked when the x-value percentages were between the 55-65% range. After this maximum point, the cancer deaths droped as more of the population reported having private health insurance. The natural cubic spline selected a *df* of 3, and the smoothing spline had a *df* of 5. Either of the degrees of freedom seemed suitable, neither under or overfitting the data.

Figure 14



In the case of Figure 14, I found the lower degrees of freedom selected by the cross-validated smoothing spline underfit the data, considering the grouping of the data points in the bottom left corner of the scatter plot. The $df = 5$ natural cubic spline reflected the slight curved shape of the data.

With and without smoothing

```
Nsmooth <- glm(TARGET_deathRate ~ PctBlack + PctPrivateCoverage + medIncome, data = canF)
Ysmooth <- glm(TARGET_deathRate ~ ns(PctBlack, df = 5) + ns(PctPrivateCoverage, df = 3)
               + ns(medIncome, df = 5), data = canF)
```

The main differences between these models were the coefficient values and the standard error values. The model without smoothing had smaller coefficients (mostly under 0.5), with standard error values only slightly higher. The model with smoothing had a greater range in coefficient values with some in the positive and negative tens, and one in the low hundreds. The smoothing spline values were not too ridiculous, but the predictor variables may be correlated - a scatter plot matrix of all the variables showed that many predictors were correlated with each other.

Conclusion

The smoothing was unnecessary to some degree. If you were to compare the splines fitting the tricep skin thickness models in Part 1 of this report, you would see how much better the splines fit the data. Splines could place too much emphasis on outliers or fitting every single data point if used improperly. This would result in a poor predictive model.