Variable Coding in Regression

Ashley I Naimi

Spring 2024

Contents

Learning Objectives

- Describe how a regression model can be conceptually divided into a "nuisance function" and a function of interest, or a "target function".
- Outline how considerations about coding variables in the nuisance function differ from considerations about coding variables in the function of interest.
- Describe the "table 2 fallacy."
- Identify the benefits and tradeoffs of categorizing a continuous exposure or outcome variable.
- Communicate the difference between quantile regression and standard linear regression for a continuous outcome.
- Describe what the `asis` function is in R, and why it's important to use it.
- Identify problems with transforming a continuous exposure using z-scores.
- Be able to deploy regression splines in R using the `ns()` and/or `bs()` functions.

1 Variable Coding

Variable coding is one of the more important considerations when fitting a regression model. The way we code our data an enter them into our regression functions can have a fundamental effect on how we can interpret the results of interest. However, it's important to recognize that, in other fields (e.g., statistics), coding considerations are often made with respect to the properties of an estimator, such as information loss, efficiency, bias, and power (Altman and Royston, 2006). While these considerations are important, they do not always outweigh considerations such as how the coding strategy affects our interpretation.

One important tool in navigating these ideas is the concept of a "nuisance function". Consider the following logistic regression model, with an outcome of interest Y, an exposure of interest X, and a set of confounders required for identifiability:

$$\text{logit}\left[P(Y = 1 \mid X, C)\right] = \alpha_0 + \alpha_1 X + \alpha_2 C$$

Mathematically, we can depict the right hand side of the above equation as a combination of two functions: the part relating the exposure to the outcome, which captures the effect of interest, and a second part that captures the relationship between the confounders and the outcome of interest:

$$\text{logit}\left[P(Y = 1 \mid X, C)\right] = \mu(X) + \eta(C)$$

where $\mu(X)$ represents the effect of the treatment of interest, or the **target function**, and the remaining function $\eta(C)$ is a **nuisance function** that enables us to adjust for confounding.

The distinction between the nuisance function $\eta(C)$ and the target function $\mu(X)$ helps us understand when we need to consider coding with respect to interpretation, versus when we need to consider coding with respect to information loss, efficiency, and bias. Roughly speaking, we should focus on coding the variables of interest with respect to the interpretation that results, and we should focus on coding the nuisance function variables with respect to efficiency and bias.

Context Note:

In 2013, Westreich and Greenland coined the "Table 2 Fallacy" (Westreich and Greenland, 2013). In a scientific manuscript, Table 1 usually contains descriptive statistics on the data being used to answer the question at hand. Table 2 usually contains the estimates and standard errors for all of the variables included in the regression models used to analyze the data.

Often, researchers will interpret the coefficients in Table 2 in roughly the same manner; for example, as the "effect" of the exposure and the "effect" of the confounder on the outcome. However, one often collects confounding information to achieve identifiability with respect to the exposure. One typically does not collect additional information to achieve identifiability with respect to the confounders. Interpreting regression coefficients for the confounding (i.e., nuisance) variables leads to the Table 2 Fallacy.

One way to avoid the Table 2 Fallacy is to understand that the nuisance function is there to assist us in estimating the exposure effect. It is not something we are substantively interested in. If we were interested in the effect of a confounding variable, we would conduct a separate analysis, with a different set of confounding variables.

Conceptually separating the nuisance function from the function of interest helps us to avoid some of these traps.

2 Coding the Outcome Variable (left hand side)

We'll start with a consideration of coding variables on the *left hand side* of a regression equation: the "outcome" (or dependent) variable. Let's consider the outcome we've been working with most in this class: weight change between 1971 and 1982 in the NHEFS data:

```
#' Define where the data are
file_loc <- url("https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/1268/20/nhefs.csv")

#' This begins the process of cleaning and formatting the data
nhefs <- read_csv(file_loc) %>%
  select(qsmk,wt82_71,wt82, wt71, exercise,sex,age,
         race,income, marital,school,
         asthma,bronch,
         starts_with("alcohol"),-alcoholpy,
         starts_with("price"),
         starts_with("tax"),
         starts_with("smoke"),
         smkintensity82_71) %>%
  na.omit(.)

factor_names <- c("exercise","sex","race","asthma","bronch")
nhefs[,factor_names] <- lapply(nhefs[,factor_names] , factor)

#' Define outcome
nhefs <- nhefs %>% mutate(id = row_number(),
                          wt_delta = as.numeric(wt82_71>median(wt82_71)),
                          .before = qsmk)

#' Quick summary of data
nhefs %>% print(n=5)
```

```
## # A tibble: 1,055 x 27
##      id wt_delta  qsmk wt82_71  wt82  wt71 exercise sex     age race  income
##   <int>    <dbl> <dbl>   <dbl> <dbl> <dbl> <fct>    <fct> <dbl> <fct>  <dbl>
```

```
## 1     1        0     0  -10.1   68.9  79.0 2        0        42 1         19
## 2     2        0     0    2.60  61.2  58.6 0        0        36 0         18
## 3     3        1     0    4.99  64.4  59.4 2        0        68 1         15
## 4     4        1     0    4.99  92.1  87.1 1        0        40 0         18
## 5     5        1     0    4.42 103.   99   1        1        43 1         11
## # i 1,050 more rows
## # i 16 more variables: marital <dbl>, school <dbl>, asthma <fct>, bronch <fct>,
## #   alcoholfreq <dbl>, alcoholtype <dbl>, alcoholhowmuch <dbl>, price71 <dbl>,
## #   price82 <dbl>, price71_82 <dbl>, tax71 <dbl>, tax82 <dbl>, tax71_82 <dbl>,
## #   smokeintensity <dbl>, smokeyrs <dbl>, smkintensity82_71 <dbl>
```

Let's generate a histogram with overlaid density plot of the continuous weight change variable:

```
wt_plot <- ggplot(nhefs) +
  geom_histogram(aes(x=wt82_71, after_stat(density))) +
  geom_density(aes(x=wt82_71),
               kernel = "epanechnikov",
               bw = "ucv",
               size=1,
               color="blue") +
  geom_vline(aes(xintercept = mean(wt82_71)),
             color="red") +
  geom_vline(aes(xintercept = median(wt82_71)),
             color="magenta",
             linetype="dashed") +
    geom_vline(aes(xintercept = quantile(wt82_71, probs=c(0.25))),
             color="green",
             linetype="dashed") +
      geom_vline(aes(xintercept = quantile(wt82_71, probs=c(0.75))),
             color="green",
             linetype="dashed") +
  facet_wrap(~qsmk) +
  scale_x_continuous(expand=c(0,0)) +
  scale_y_continuous(expand=c(0,0)) +
  ylab("Density") + xlab("Weight Change (kg), 1971-1982")
```

```
ggsave(here("figures","wt_change_density.png"))
```

This figure shows the distribution of the continuous weight change variable stratified by `qsmk` status, with lines representing the mean (red) and median (magenta) weight changes.

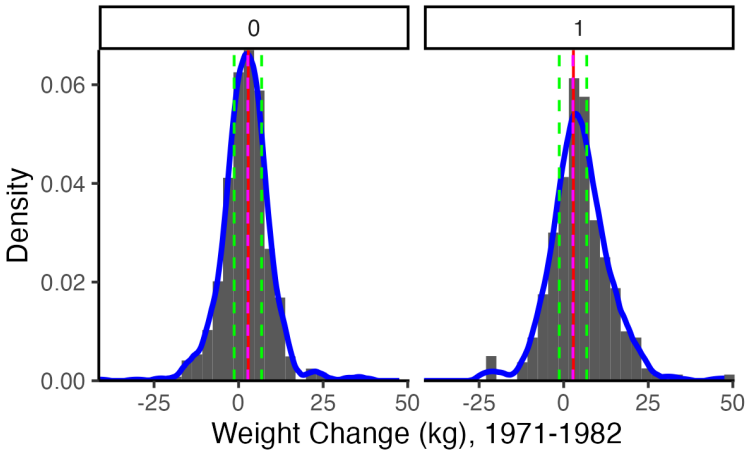


Figure 1: Distribution of weight change in kilograms between 1971 and 1982 in the NHEFS data. Blue line represents Epanechnikov kernel density estimator. Solid red line represents mean weight change. Dashed magenta line represents median weight change.

If we were to leave weight change coded as a continuous variable, and use linear regression to model this variable (which includes ordinary least squares or maximum likelihood with a Gaussian distribution and identity link function), we would be comparing the **mean** weight change between those with `qsmk = 1` versus `qsmk = 0`:

```
summary(lm(wt82_71 ~ qsmk, data = nhefs))$coefficients
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 2.234612  0.2695599 8.289852 3.427324e-16
## qsmk        2.569969  0.5419526 4.742055 2.406345e-06
```

```
summary(glm(wt82_71 ~ qsmk, data = nhefs, family = gaussian("identity")))$coefficients
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 2.234612  0.2695599 8.289852 3.427324e-16
## qsmk        2.569969  0.5419526 4.742055 2.406345e-06
```

The coefficient from these models can be interpreted as a difference in the mean weight change among those who quit smoking versus those who didn't quit smoking.

We may instead be interested in modeling the **median** instead of the mean. We could do this with quantile regression using the `quantreg` package:

```
install.packages("quantreg", repos='http://lib.stat.cmu.edu/R/CRAN', dependencies=T)
```

```
##
## The downloaded binary packages are in
##  /var/folders/zm/rqfqp5xs0fs86qs2mcxk6q0r0000gr/T//Rtmp6DhqfG/downloaded_packages
```

```
library(quantreg)

summary(rq(wt82_71 ~ qsmk, tau=.5, data = nhefs))
```

```
##
## Call: rq(formula = wt82_71 ~ qsmk, tau = 0.5, data = nhefs)
##
## tau: [1] 0.5
##
## Coefficients:
##             Value   Std. Error t value Pr(>|t|)
## (Intercept) 2.38293 0.26332    9.04955 0.00000
## qsmk        2.03922 0.61142    3.33523 0.00088
```

The coefficient for `qsmk` in this example can be interpreted as the difference in the **median** weight change among those who quit versus those who didn't quit smoking.

With a continuous outcome and quantile regression, we can actually model much more than just the median. For example, we can look at the difference in the 25th, median (50th), and 75th percentiles of weight change among those who quit versus those who didn't quit smoking:

```
summary(rq(wt82_71 ~ qsmk, tau=c(.25,.5,.75), data = nhefs))
```

```
##
## Call: rq(formula = wt82_71 ~ qsmk, tau = c(0.25, 0.5, 0.75), data = nhefs)
##
## tau: [1] 0.25
##
## Coefficients:
##             Value    Std. Error t value  Pr(>|t|)
## (Intercept) -1.70085  0.27640   -6.15357  0.00000
## qsmk         1.81359  0.72190    2.51226  0.01214
##
## Call: rq(formula = wt82_71 ~ qsmk, tau = c(0.25, 0.5, 0.75), data = nhefs)
##
## tau: [1] 0.5
##
## Coefficients:
##             Value   Std. Error t value Pr(>|t|)
## (Intercept) 2.38293 0.26332    9.04955 0.00000
## qsmk        2.03922 0.61142    3.33523 0.00088
##
## Call: rq(formula = wt82_71 ~ qsmk, tau = c(0.25, 0.5, 0.75), data = nhefs)
##
## tau: [1] 0.75
##
## Coefficients:
##             Value    Std. Error t value  Pr(>|t|)
## (Intercept)  6.23236  0.27621   22.56400  0.00000
## qsmk         3.40368  0.78658    4.32719  0.00002
```

These results suggest that the association between quitting smoking is greater for the 75th percentile of the distribution of weight change than it is for the median and 25th percentiles[1].

Overall, it's important to recognize that if the outcome variable is continuous, we end up quantifying mean differences for a GLM (or ratios, if we use

[1] Note that we are not formally comparing these, which would require carrying out a statistical test (z-test, t-test) to compare the point estimates, or which would require constructing confidence intervals, which can be done using (among others) the standard Wald equation.

the appropriate link function in a GLM) or quantile differences for a quantile regression models.

If we choose to categorize our outcome using, for example, binary indicator coding, this changes what we end up estimating with the GLM approach.[2] For example, dichotomizing weight change as greater than median versus less than or equal to the median enables us to quantify how the probability of greater than median weight gain changes for those who quit smoking versus for those who don't.

[2] quantile regression cannot be used for a binary dependent variable.

When choosing to categorize a continuous outcome variable, it is essential to evaluate **how it will affect the interpretation of the results.** For the outcome variable, which is closely tied to the function of substantive interest (i.e., not the nuisance function), this should always be the essential consideration in determining how to code the variable.

For example, in this class, we've looked at the outcome variable coded as a **greater than median weight gain**. In the NHEFS data, the median weight gain value is 2.73. Here are several questions that we should ask before we opt to use the median as a cutpoint (or any cutpoint):

- Is the median value meaningful? For example, if you were interested in a positive weight change, and the median value was zero or negative, it may not be the best choice. Or perhaps the median value is too small to be of any interest. All of these considerations come into play when choosing a threshold.
- Is the median stable? The median in the NHEFS data may be very different from the median in another dataset that includes data on weight change. This may make the results of the analysis with NHEFS data non-generalizable. This is also true of other descriptive summaries such as other percentiles of the distribution, or the mean.

Even if the median, or mean, or some other quantile of the distribution of the outcome variable represents a meaningful threshold, it is still essential that you interpret your results in terms of actual values, rather that statistical summaries. For example, instead of stating that: "the association between quitting smoking and greater than median weight change was …" one should instead state that: "the association between quitting smoking and weight change greater than 2.7 kg was …".

3 Coding the Exposure Variable

Many of the same considerations come into play when coding the exposure variable. Imagine we're interested in quantifying the relationship between smoking intensity and weight change.

```
nhefs <- read_csv(here("data","nhefs_data.csv"))
```

Change in smoking intensity is distributed as follows:

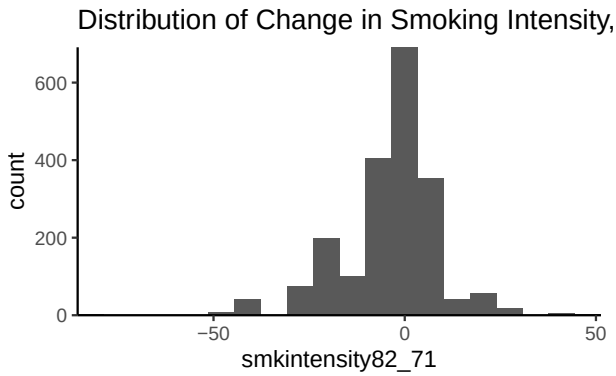


Figure 2: Distribution of smoking intensity in the NHEFS data.

This variable represents the change in the number of cigarettes smoked per day between 1971 and 1982. If we carry out a simple regression, we obtain the following results:

```
summary(lm(wt82_71 ~ smkintensity82_71, data = nhefs))$coefficients
```

```
##                      Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)        2.40617607 0.17759026 13.549032 4.466751e-40
## smkintensity82_71 -0.03733264 0.01287615 -2.899364 3.780015e-03
```

This model suggests that smoking one more cigarette per day is associated with a 0.04 kg reduction in weight between 1971 and 1982.

This relationship can be seen in the figure below, showing the regression line from the above model fit to the data:

A 0.04 kg reduction in weight over ten years, though statistically significant, is not likely a clinically meaningful difference. However, arguably, neither is increasing the number of cigarettes smoked per day by 1.

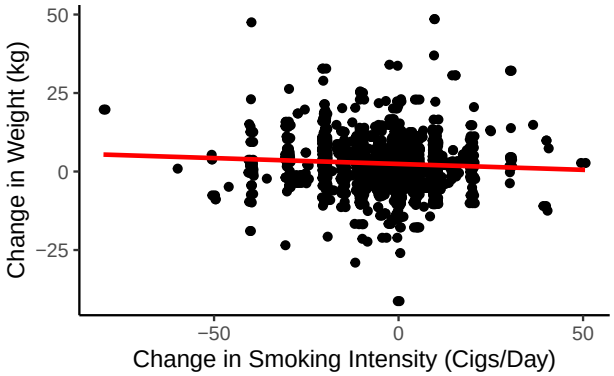


Figure 3: Distribution of smoking intensity in the NHEFS data.

This speaks to an important relationship between statistical significance, clinical and/or public health (more broadly, substantive) significance, and the scale of the exposure and outcome variables. For example, an estimate may be highly statistically significant, but of low substantive significance. This may be because the exposure and/or outcome variables are scaled inappropriately. For instance:

- a 1 mm Hg change in diastolic blood pressure
- an increase in low intensity exercise of 1 minute per day
- a one kg/m^2 increase in BMI

Generally, for variables of primary interest such as the exposure and outcome variables, it's important to consider what scale they are on before making judgements about the importance or lack of importance for a given set of results.

For an exposure like the number of cigarettes smoked per day, we can rescale the exposure variable to give us a more meaningful contrast. Suppose, for example, we were interested in the impact of increasing the number of cigarettes by 5. We can rescale[3] the exposure as follows:

[3] Generally, to rescale an exposure so one might interpret the coefficient as an a unit change, when the original scale of the variable is one single unit, one can rescale the variable as x/a.

```
summary(lm(wt82_71 ~ I(smkintensity82_71/5),
    data = nhefs))$coefficients
```

```
##                          Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)             2.4061761 0.17759026 13.549032 4.466751e-40
## I(smkintensity82_71/5) -0.1866632 0.06438073 -2.899364 3.780015e-03
```

In the above regression equation, we divide smoking intensity by 5 to place

it on a "five per day" scale rather than a "one per day" scale. Additionally, we use the `asis` function, denoted `I()`, which, when used in a regression equation, prevents operators such as "+", "-", "*","^" from being interpreted as a regression formula operator, and interprets them instead as arithmetic operators.

3.1 Creating a z-score for the exposure

One commonly used transformation is to standardize a variable to generate a coefficient that can be interpreted as the change in the mean of the outcome for one standard deviation change in the continuous exposure of interest. Standardizing the exposure is usually accomplished by subtracting the mean from the variable and dividing by its standard deviation[4]:

$$X' = \frac{X - E(X)}{SD(X)}$$

which can be accomplished in R using the `scale` function:

[4] There are other "standardizing" transformations, including scaling the range of the variable to lie between 0 and 1.

```
smk_stdz <- scale(nhefs$smkintensity82_71)

str(smk_stdz)
```

```
##  num [1:2000, 1] 1.051 2.563 0.522 0.296 -0.989 ...
##  - attr(*, "scaled:center")= num -3.91
##  - attr(*, "scaled:scale")= num 13.2
```

However, standardizing the exposure can create several problems. These problems stem from the fact that the standard deviation of any variable is very sensitive to arbitrary features of the study. Features that can affect the standard deviation in a particular study include inclusion criteria (age, geographic location, underlying risk, history), study design (nested case-cohort, nested case-control, stratified sampling designs, etc). After standardizing, these features can essentially confound the true relation between the exposure and outcome under study (Greenland et al., 1991).

In effect, standardizing the exposure is often not a good idea, because the interpretation of the effect of interest becomes dependent upon the magnitude of the standard deviation of the variable, which can be rather arbitrary.

In some settings, it is important to standardize the exposure of interest, such as when the exposure is highly correlated with another variable that is also associated with the outcome. This is often the case in studies of the effect of gestational weight gain on pregnancy outcomes. GWG is highly correlated with gestational age, also a marker of adverse pregnancy outcomes. However, when transforming GWG on the z-score scale, it is often customary to use the standard deviation of GWG in the target population, instead of just the sample standard deviation. This is one potential solution to the problems that can be introduced by standardizing exposure variables.

3.2 Categorizing the Exposure

It may sometimes be useful to categorize a continuous exposure. There are a lot of perspectives against the practice of categorizing continuous exposures (Altman and Royston, 2006, Bennette and Vickers (2012), Schellingerhout et al. (2009)). Often, these perspectives are motivated as follows:

Step 1: simulate a continuous exposure and a continuous or binary outcome, such that there is a relationship between the exposure and the outcome

Step 2: estimate the association between the continuous exposure and the outcome, demonstrating that the estimate is accurate, and the p value is low (or other summary statistic denotes the presence of a strong signal with low noise)

Step 3: categorize the exposure at some arbitrary threshold, and estimate the relation between the outcome and this categorized exposure.

Step 4: demonstrate that the p value for this categorized exposure is higher than in the previous analysis (or that the other summary statistic denotes the presence of a weaker signal with potentially more noise)

The problem with these arguments (again) that they prioritize statistical considerations over substantive considerations. Consider the following simulated example, with an exposure x related to an outcome of interest y. Imagine further that there is an important threshold in x determined *a priori*.

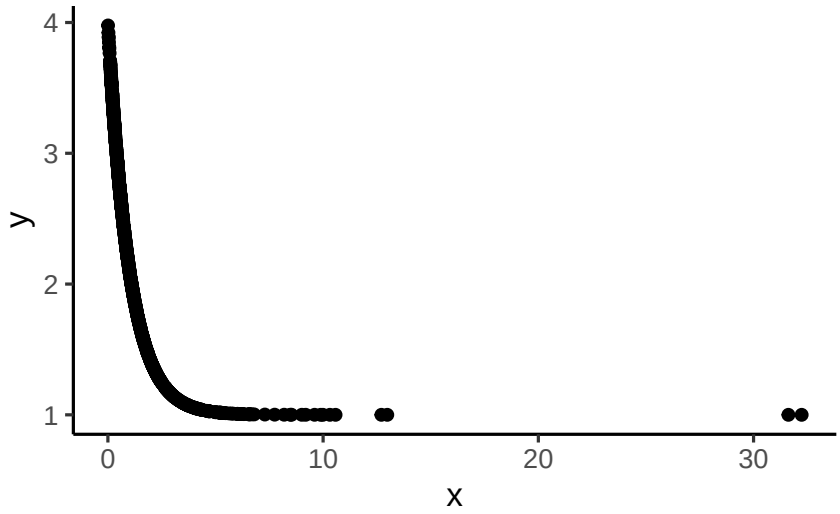
For instance, x may be the number of cups of vegetables consumed per day. In this case, 2 cups of vegetables per day represents a natural threshold, in that it is a recommended number of cups of vegetables one needs to achieve a healthy diet pattern.

```
n <- 1000
x <- exp(rnorm(n))
y <- 1 + 3 * exp(-x)
plot_dat <- tibble(x, y, x_cat = as.numeric(x >
    2))

plot_dat %>%
    group_by(x_cat) %>%
    summarise(meanY = mean(y))
```

```
## # A tibble: 2 x 2
##   x_cat meanY
##   <dbl> <dbl>
## 1     0  2.49
## 2     1  1.17
```

```
ggplot(plot_dat) + geom_point(aes(x = x,
    y = y)) + theme_classic()
```



In this case, it would likely make much less sense to keep x as a continuous variable, since the interpretation of a categorized x would be much more relevant. These are largely the same considerations as discussed above with regards to categorizing the outcome.

4 Regression Splines

When interest does lie in the continuous relationship between an exposure and outcome of interest, the preferred approach is to use splines to model this relation. The word spline is an engineering/architectural term. It refers to a flexible piece of material that individuals would use to draw up blue-prints that incorporated flexible curves:

In the 1970s and 80s, statisticians began translating some of these engineering concepts to curve fitting. The basic idea was to create functions of a continuous variable that would yield the appropriate degree of flexibility between that value and the conditional outcome expectation.

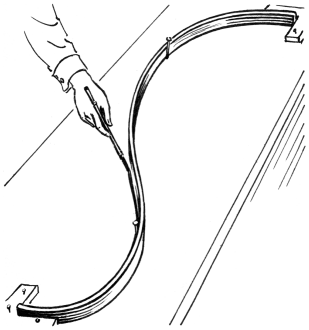


Figure 4: An illustration of a engineering/architectural spline use to draw flexible curves for blueprint diagrams. Source: Wikipedia

When it comes to implementation, there are a great many number of different options one can use to fit splines. Among these include natural cubic splines (the `ns()` option in R), B-splines (the `bs()` option in R), generalized additive models (or GAMs, implemented in the `gam` package or the `mgcv` package in R), penalized smoothing splines (implemented via the `smooth.spline()` function in R), or restricted quadratic splines (Howe et al., 2011).

Of all these, restricted quadratic splines are the easiest to understand. They do not share some of the ideal mathematical properties of the other implementations (properties that we will not discuss, but that relate to the derivatives of the spline functions). However, here we will walk through the steps to create restricted quadratic splines to demonstrate how splines work in principle.

4.1 Restricted Quadratic Splines

When using splines, the basic question is about how to code the relation between the conditional expectation of the outcome and a continuous covariate. For example, suppose we had the following exposure (x) and outcome (y) data:

```
# load package needed to generate
# laplace distribution
install.packages("rmutil", repos = "http://lib.stat.cmu.edu/R/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/zm/rqfqp5xs0fs86qs2mcxk6q0r0000gr/T//Rtmp6DhqfG/downloaded_packages
```

```
library(rmutil)
```

```
##
## Attaching package: 'rmutil'
## The following object is masked from 'package:Hmisc':
##
##     units
## The following object is masked from 'package:tidyr':
##
##     nesting
## The following object is masked from 'package:stats':
##
##     nobs
## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

```
# set the seed for reproducibility
set.seed(12345)

## generate the observed data
n = 1000
# uniform random variable bounded by 0
# and 8
x = runif(n, 0, 8)
# continuous outcome as a complex
# function of x
y = 5 + 4 * sqrt(9 * x) * as.numeric(x <
    2) + as.numeric(x >= 2) * (abs(x - 6)^(2)) +
    rlaplace(n)

a <- data.frame(x = x, y = y)
head(a)
```

```
##          x          y
## 1 5.767231  3.1931580
## 2 7.006186  7.1753275
## 3 6.087859  0.7120441
## 4 7.088997  5.8326162
## 5 3.651848 10.9792288
## 6 1.330974 18.1672097
```

We can create a scatter plot of these data to see how they relate:

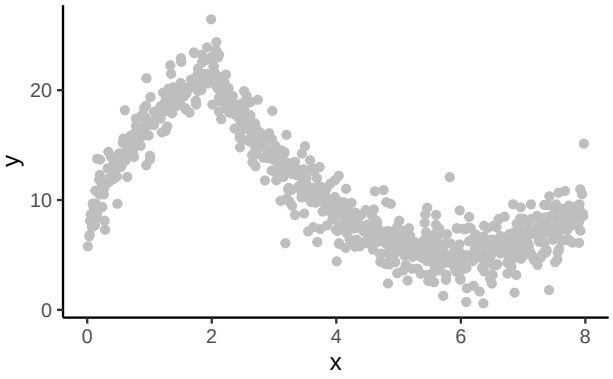


Figure 5: Scatterplot of the relation between a simulated exposure and simulated outcome with a complex curvilinear relation

Obviously, the relation between X and Y is not a straight line. But suppose we assume linearity, fitting the following regression model to these data:

$$E(Y \mid X) = \beta_0 + \beta_1 X$$

```
model1 <- lm(y ~ x)
a$y_pred1 <- predict(model1)
```

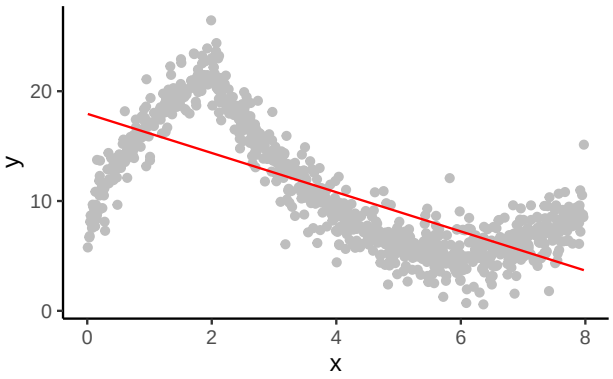


Figure 6: Scatterplot of the relation between a simulated exposure and simulated outcome with a complex curvilinear relation and a linear fit

If X were a confounder and we assumed such a linear fit, there would be an important degree of residual confounding left over in our estimate. If X were our exposure of interest, such a linear fit would seriously mis-represent the true functional relation between the exposure and the outcome. Splines are meant to solve these problems.

Splines are essentially a function that take the exposure as an argument, and return a set of **basis functions** that account for the curvilinear relation. Any spline starts by selecting knots, which are the points along the variable's distribution where we will create categories. In our example, we will use three knots chosen at $x = 1$, $x = 4$, and $x = 6$. We will denote these χ_1, χ_2, and χ_3, respectively.[5]

[5] Knots can be chosen as *a priori* cutpoints, or by selecting percentile's of the distribution (e.g., the 25th, 50th, and 75th percentile values).

Restricted quadratic spline basis functions for a three knot spline can then be defined as follows:

$$f(x) = \left[(x-\chi_1)_+^2 - (x-\chi_3)_+^2\right] \left[(x-\chi_2)_+^2 - (x-\chi_3)_+^2\right]$$

The parentheses with a subscripted plus sign refers to the *positive part function* returns the value of the difference if it is positive, and zero otherwise[6]

[6] formally, $(x-\chi_1)_+ = x-\chi_1$ if $(x-\chi_1) > 0; 0$ otherwise

With these equations, we can create the spline basis functions we need to fit restricted quadratic splines with out simulated data:

```
basis_1 <- as.numeric((x - 1) > 0) * (x -
    1)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2
basis_2 <- as.numeric((x - 3) > 0) * (x -
    3)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2
```

We can now fit our regression model using these spline basis functions:

```
model2 <- lm(y ~ x + basis_1 + basis_2)
a$y_pred2 <- predict(model2)
```

Clearly, using splines gives us a much better fit.

A natural question that arises from this illustration is, how do we interpret

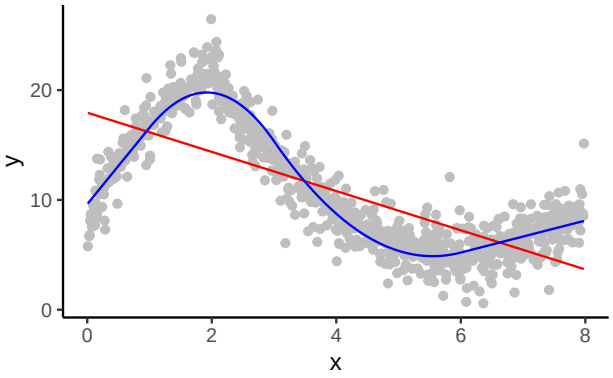


Figure 7: Scatterplot of the relation between a simulated exposure and simulated outcome with a complex curvilinear relation and a linear fit

the spline results? For example, if we look at a summary of the estimates from `model2`, we get:

```
summary(model2)$coefficients[, -3]
```

```
##              Estimate Std. Error      Pr(>|t|)
## (Intercept)  9.604306 0.20346470 2.641625e-256
## x            6.974522 0.15343656 3.773854e-245
## basis_1     -3.796962 0.05921589  0.000000e+00
## basis_2      5.408630 0.08260659  0.000000e+00
```

Can we interpret the 7, -3.8, and 5.4 that we estimated for the linear and spline terms? The answer is **no.**

An analyst will encounter using splines in two settings: 1) adjusting for a continuous confounder; and 2) accounting for a curvilinear relation between an exposure and an outcome. When adjusting for confounding, interest often lies primarily in the exposure-outcome relation.

The confounder-outcome relation is usually not of particular interest. Again, it's a "nuisance" function, and not of direct interest. As a result, even though spline estimates do not have an interpretation, it does not matter as long as they are appropriately adjusting for the confounder-outcome relation.

If splines are being used to model a continuous expsoure-outcome relation, then the interpretation of the estimates will not matter as long as there is a curvilinear relation. Consider, for example, the use of quadratic and cubic terms:

$$E(Y \mid X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$$

The objective of fitting these quadratic and cubic terms is to account for any curvilinear relation. One would not typically interpret the coefficients of the squared and cubic terms. One could, however, predict the outcome under different values of X from this model, and compare these predicted outcomes.

The same principles apply to splines. There is no interesting way to interpret the coefficients for the spline terms. However we could obtain estimates of the effect of changing X from one level to another. Suppose we were interested in comparing the average outcome under $X = 1$ versus $X = 2$ and $X = 2$ versus $X = 4$. We could easily do this using the splines we fit above:

```
x = 1
basis_1 <- as.numeric((x - 1) > 0) * (x -
    1)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2
basis_2 <- as.numeric((x - 3) > 0) * (x -
    3)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2

nd1 <- data.frame(x, basis_1, basis_2)
mu1 <- predict(model2, newdata = nd1)

x = 2
basis_1 <- as.numeric((x - 1) > 0) * (x -
    1)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2
basis_2 <- as.numeric((x - 3) > 0) * (x -
    3)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2

nd2 <- data.frame(x, basis_1, basis_2)
mu2 <- predict(model2, newdata = nd2)

x = 6
basis_1 <- as.numeric((x - 1) > 0) * (x -
    1)^2 - as.numeric((x - 6) > 0) * (x -
```

```
    6)^2
basis_2 <- as.numeric((x - 3) > 0) * (x -
    3)^2 - as.numeric((x - 6) > 0) * (x -
    6)^2

nd6 <- data.frame(x, basis_1, basis_2)
mu6 <- predict(model2, newdata = nd6)

mu2 - mu1
```

```
##       1
## 3.17756
```

```
mu6 - mu2
```

```
##         1
## -14.55133
```

Here, we see that the effect of going from $X = 1$ to $X = 2$ is 3.18 while the effect of going from $X = 6$ to $X = 2$ is -14.55. Notably, these estimates account for the curvilinear relation between X and Y. Confidence intervals can be obtained using the bootstrap.

5 Coding the Nuisance Variables

In this set of notes, we will cover some considerations and techniques in specifying the right hand side of a regression model. This is sometimes referred to as "coding" the variables in the model, or "feature engineering" (in the machine learning literature). To illustrate some of the issues, we will use the NHEFS data where we seek to estimate the association between sex and smoke intensity adjusted for age and marital status:

The age variable is distributed as follows:

Additionally, marital status and sex are distributed as:

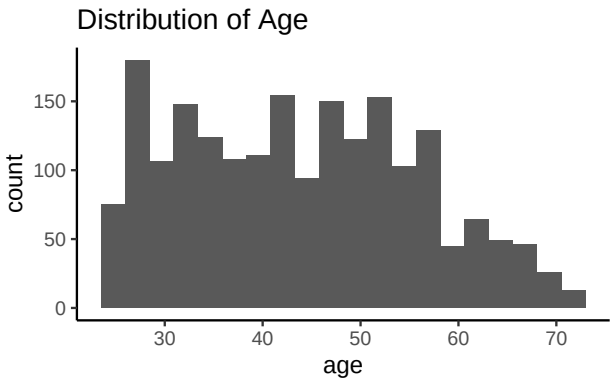


Figure 8: Distribution of age in the NHEFS data.

```
table(nhefs$marital)
```

```
##
##    2    3    4    5    6    8
## 1583  102  130  126   58    1
```

```
table(nhefs$sex)
```

```
##
##    0    1
##  994 1006
```

Using the NHEFS codebook, we can determine that the marital status categories are:

Category	Marital Status in 1971	N
2	Married	1583
3	Widowed	102
4	Never Married	130
5	Divorced	126
6	Separated	58
8	Unknown	1

First, we will deal with marital status. The first thing we want to do is reduce the number of categories as much as possible. This reduction will be based entirely on substantive (background) knowledge. Let's assume, for our purposes, that we do not expect the average outcome between separated and divorced individuals to differ. We can thus combine their category:

```
nhefs$marital <- ifelse(nhefs$marital ==
    6, 5, nhefs$marital)

table(nhefs$marital)
```

```
##
##    2    3    4    5    8
## 1583  102  130  184    1
```

Next we have to deal with the last (unknown) category, which has a single observation. For our purposes, let's assume the person in this category is among those with the most common value: married

```
nhefs$marital <- ifelse(nhefs$marital ==
    8, 2, nhefs$marital)

table(nhefs$marital)
```

```
##
##    2    3    4    5
## 1584  102  130  184
```

Finally, we will examine the status of the marital variable in R:

```
class(nhefs$marital)
```

```
## [1] "numeric"
```

Because `marital` is a numeric variable, if we include it in a regression model as is, then we will be estimating the association between marital and smoking intensity assuming a linear relation between all the categories. This assumption is untenable for a variable like marital status.

To resolve this, **we must change the class of the marital variable.** We can do this in two ways: first by changing the class in the data object itself; second by changing the class in the model itself:

```
model_2 <- glm(smokeintensity ~ sex + age +
    factor(marital), data = nhefs, family = poisson(link = "log"))

summary(model_2)$coefficients[, 1:2]
```

```
##                     Estimate   Std. Error
## (Intercept)       3.30111415 0.0198670209
## sex              -0.24142834 0.0102123462
## age              -0.00377753 0.0004250036
## factor(marital)3  0.04046523 0.0244502187
## factor(marital)4 -0.10657459 0.0210181383
## factor(marital)5 -0.06804850 0.0181707534
```

We can tell from the output that the referent category for the marital variable is category 2: married. Finally, with the age variable, we must also account for the fact that the relation between age and smoking intensity is potentially nonlinear. We can do this with splines.

In R, splines are easily implemented using the `splines` package. For our particular example, we use b-splines:

```
library(splines)

model_2a <- glm(smokeintensity ~ sex + age +
    factor(marital), data = nhefs, family = poisson(link = "log"))

summary(model_2a)$coefficients[, 1:2]
```

```
##                     Estimate   Std. Error
## (Intercept)       3.30111415 0.0198670209
## sex              -0.24142834 0.0102123462
## age              -0.00377753 0.0004250036
## factor(marital)3  0.04046523 0.0244502187
## factor(marital)4 -0.10657459 0.0210181383
## factor(marital)5 -0.06804850 0.0181707534
```

```
model_2b <- glm(smokeintensity ~ sex + bs(age,
    df = 3, degree = 3) + factor(marital),
    data = nhefs, family = poisson(link = "log"))

summary(model_2b)$coefficients[, 1:2]
```

```
##                                  Estimate Std. Error
## (Intercept)                   3.108889493 0.01666093
## sex                          -0.243571695 0.01021521
## bs(age, df = 3, degree = 3)1  0.335596239 0.05164451
## bs(age, df = 3, degree = 3)2 -0.290153284 0.04323100
## bs(age, df = 3, degree = 3)3 -0.007047172 0.04103466
## factor(marital)3              0.049850244 0.02459812
## factor(marital)4             -0.094997166 0.02106360
## factor(marital)5             -0.062969924 0.01819567
```

But we can also use natural splines:

```
library(splines)

model_2a_ns <- glm(smokeintensity ~ sex +
    age + factor(marital), data = nhefs,
    family = poisson(link = "log"))

summary(model_2a)$coefficients[, 1:2]
```

```
##                      Estimate   Std. Error
## (Intercept)       3.30111415 0.0198670209
## sex              -0.24142834 0.0102123462
## age              -0.00377753 0.0004250036
## factor(marital)3  0.04046523 0.0244502187
## factor(marital)4 -0.10657459 0.0210181383
## factor(marital)5 -0.06804850 0.0181707534
```

```
model_2b_ns <- glm(smokeintensity ~ sex +
    ns(age, df = 3) + factor(marital), data = nhefs,
    family = poisson(link = "log"))

summary(model_2b)$coefficients[, 1:2]
```

```
##                                   Estimate Std. Error
## (Intercept)                   3.108889493 0.01666093
## sex                          -0.243571695 0.01021521
## bs(age, df = 3, degree = 3)1  0.335596239 0.05164451
## bs(age, df = 3, degree = 3)2 -0.290153284 0.04323100
## bs(age, df = 3, degree = 3)3 -0.007047172 0.04103466
## factor(marital)3              0.049850244 0.02459812
## factor(marital)4             -0.094997166 0.02106360
## factor(marital)5             -0.062969924 0.01819567
```

6 Takeaways

- A regression model can be conceptually divided into a "nuisance function" and a function of interest. The function of interest typically represents the estimand targeted by the regression model. The nuisance function represents the portion of the regression model required for identifying the estimand of interest.
- Nuisance functions are not typically subject to interpretation from a substantive perspective. Therefore, when coding variables in the nuisance function, one should seek to optimize the statistical properties of the estimator. However, the target function is often subject to nuanced interpretation substantively. Therefore, when coding variables in the target function, one should seek to balance optimizing the interpretation of the function with it's statistical properties.
- The "Table 2 Fallacy" occurs when one interprets coefficients from a regression model representing the nuisance function in the same capacity as the coefficients representing the target function. The problem occurs because identification often focuses exclusively on the target function. That

is, variables in the nuisance function are meant to support identification of the target function. However, the effects of variables in the nuisance function are often not identified (i.e. are subject to confounding, selection, information bias).

- Categorizing continuous expsoure and / or outcome variables can lead to a reduction in the statistical performance of the estimation approach. However, it may better align with the subject matter of interest, and lead to better interpretability of the effects of interest.
- For a continuous outcome, quantile regression can be used to estimate quantile difference of interest. For example, the median difference in the outcome for a one unit change in the exposure. In contrast, standard linear regression quantifies the difference in outcome means for a unit change in the exposure.
- When used in a regression function, the `as.is()` function in R prevents operators such as "+", "-", "*" and "^" to be interpreted in the regression context, and instead interprets them as arithmetic operators. For example, fitting the following model `lm(y ~ .^2, data = a)` treats "^" as an interaction operator (the regression context), whereas `lm(y ~ I(.^2), data = a)` will square all variables in the dataset and include these squared terms in a regression model (the arithmetic context).
- Using z-scores can lead to efficiency gains for variables in the nuisance function. However, when use to transform a continuous expsoure, z-scores can lead to a confounding of the expsoure effect of interest by the standard deviation of the exposure itself.

References

Douglas G Altman and Patrick Royston. The cost of dichotomising continuous variables. *BMJ*, 332(7549):1080, May 2006.

Caroline Bennette and Andrew Vickers. Against quantiles: categorization of continuous variables in epidemiologic research, and its discontents. *BMC Medical Research Methodology*, 12(1):21, 2012.

Sander Greenland, Malcolm Maclure, James J. Schlesselman, Charles Poole, and Hal Morgenstern. Standardized regression coefficients: A further critique and review of some alternatives. *Epidemiology*, 2(5), 1991.

CJ Howe, SR Cole, DJ Westreich, S Greenland, S Napravnik, and JJ Eron. Splines for trend analysis and continuous confounder control. *Epidemiol*, 22(6):874–5, 2011.

Jasper M. Schellingerhout, Martijn W. Heymans, Henrica C. W. de Vet, Bart W. Koes, and Arianne P. Verhagen. Categorizing continuous variables resulted in different predictors in a prognostic model for nonspecific neck pain. *Journal of Clinical Epidemiology*, 62(8):868–874, 2009.

Daniel Westreich and Sander Greenland. The table 2 fallacy: Presenting and interpreting confounder and modifier coefficients. *American Journal of Epidemiology*, 177(4):292–298, 2013.