AppendixSyntax.R

# Using R-version 3.3.0
rm(list = ls())

############################################################################
############################# Data preparation ################################
############################################################################

# Comment 1

# Using data from PS Shu, Chan YM, and Huang SL "Higher body mass index and lower intake of dairy products predict poor glycaemic control among Type 2 Diabetes patients in Malaysia" Plos One 2017; a cross-sectional study.

dt1 <- read.csv("ShuPlosOne.csv")[-72,] # removing the 72th patient which has a very long history with T2DM (100 years!)

# Preparing data
dt2 <- dt1[,c("AGE", "SEX", "MARITAL.STATUS", "EDUC_YEARS", "DURATION._.yeAR", "bmi", "HBA1c_average")]
# using a subset of the predictors

colnames(dt2) <- c("age", "sex", "married", "edu_y", "t2dm_y", "bmi", "hba1c")

defult <- c(4,7, 1:3,5:6]
# reordering columns

# Recoding
dt2$sex <- as.numeric(dt2$sex) -1 # zero = female
dt2$married <- ifelse(dt2$married == "Married", 1, 0)

dt1 <- dt2
rm(dt2)

############################################################################
############################# Data exploration ################################
############################################################################

dt1[1:6,] # first 6 observations

## edu_y hba1c age sex married t2dm_y bmi
## 1 11 7.0 60 1 1 28 17.13
## 2 14 11.4 55 1 1 5 25.70
## 3 16 6.5 57 0 1 5 36.78
## 4 5 12.3 52 0 1 7 28.39
# exploring the central locations and min/max

```
##      edu_y           hba1c             age             sex
##  Min.   : 0.00   Min.   : 4.800   Min.   :25.00   Min.   :0.0000
##  1st Qu.: 8.25   1st Qu.: 7.225   1st Qu.:46.25   1st Qu.:0.0000
##  Median :11.00   Median : 8.900   Median :56.00   Median :0.0000
##  Mean   :10.53   Mean   : 9.023   Mean   :52.88   Mean   :0.4675
##  3rd Qu.:13.00   3rd Qu.:10.375   3rd Qu.:60.00   3rd Qu.:1.0000
##  Max.   :22.00   Max.   :16.800   Max.   :65.00   Max.   :1.0000
```

### Note extreme maxima for t2dm_y and bmi.
### Sadly we don't have access to primary data source
### so we can't double check these values. For now we will keep them.

```r
par(mfrow = c(1,1))
breaks1 <- 20
xlab1 = "HbA1C"
hist(dt1$hba1c, freq = FALSE, breaks = breaks1,
     xlab = xlab1, main = "", ylab = "")
curve(dnorm(x, mean(dt1$hba1c), sd(dt1$hba1c)),
      col = 2, add = TRUE) # expected normal density
```
There is a tail towards larger HbA1c values.

#exploring pairwise distributions and relations#
pairs(dt1,
   panel = function(x,y,...){
     points(x,y,...)
     abline(lm(y ~ x), col = "red", lwd = 1.2)
   }, pch = ".", cex = 1.7
)
# Time since T2DM diagnosis and BMI seem related to HbA1c, while most other variables show a flat relation.
# Time since T2DM diagnosis itself is strongly related to age.
# BMI seems to be independent of most other variables except, perhaps, marital status.

# Pairwise linear model regressing HbA1c on times since T2DM diagnosis

# For illustrative purposes we first focus on a simple pairwise linear model

fit.0 <- lm(hba1c ~ t2dm_y, data=dt1)

summary(fit.0) # Model estimates
## Call:
## lm(formula = hba1c ~ t2dm_y, data = dt1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0175 -1.7462 -0.2527  1.3364  7.3545
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.60346    0.28754  29.921   <2e-16 ***
## t2dm_y       0.04280    0.02284   1.874   0.0629 .
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.236 on 152 degrees of freedom
## Multiple R-squared:  0.02258,   Adjusted R-squared:  0.01615
## F-statistic: 3.512 on 1 and 152 DF,  p-value: 0.06285

cbind(fit.0$coef, confint(fit.0))

##             2.5 %     97.5 %
## (Intercept) 8.6034615  8.035372873 9.17155021
## t2dm_y      0.0428048 -0.002323315 0.08793291

# distribution of the residuals #

par(bg = 'white')
par(mfrow = c(1,1))
par(cex.axis = 0.6, cex.lab = 0.6, cex.main = 0.7)
par(mgp = c(1.0, 0.35, 0), oma = c(0,0,0,0), mar = c(2.2, 1.7, 0.2, 0.4))
par(tcl = -0.25)

xlab1 <- "Fitted values"
ylab1 <- "Residuals"
lwd1 <- 2
cex2 <- 0.7

res <- summary(fit.0)$res # extracting residuals
qqnorm(res, main = "", ylab = "Residuals", pch = 16, bty = "l", cex = cex2)
qqline(res, col = "red", lwd =lwd1) # line indicating perfect fit
# The residual plot indicates tail-area deviations from the normal distribution

# figure resolutions and dimensions
res1 <- 600
h <- 6 * 0.393701
w <- 6 * 0.393701
#
# exploring non-linearity
#
# tiff("Figure1.tiff", width = w, height = h, res = res1, units = "in", compr
session = "lzw", type = "cairo"
par(bg = 'white')
par(mfrow = c(1,1))
par(cex.axis = 0.6, cex.lab = 0.6, cex.main = 0.7)
par(mgp = c(1.0, 0.35, 0), oma = c(0,0,0,0), mar = c(2.2, 1.7, 0.2, 0.4))  
# mar (b,l, t,r)
par(tcl = -0.25)

d # predict(fit.0) extracts the fitted values
plot(x = mat[,1], y = mat[,2], ylab = ylab1, xlab = xlab1, bty = "l", pch = 1 6, cex = cex2)

# loess curve
loess.fit <- loess(mat[,2]~ mat[,1])
xloess <- seq(min(mat[,1]), max(mat[,1]), length.out = 100)
yloess <- predict(loess.fit, newdata = xloess)

lines(x = xloess, yloess, col = "red", lwd = lwd1)  # add the loess curve to the graph

# line of perfect fit
abline(h = 0, lwd = lwd1, col = "grey", lty = 2)  # if there is a perfect line
# relation the residuals should cluster along the grey line

# dev.off()

# The figure indicates that time since T2DM diagnosis is
# initially related with an increased HbA1c, which switches to a negative
# relation as subjects report a longer experience with T2DM.
# Potentially this is a true effect related to increased disease
# management skills.

# Modeling the non-linear effect 

# Non-linear trends can be modelled in many different ways here we use restricted cubic splines.
# Loading the rms package for the restricted cubic spline function
ev <- try(require("rms", quietly = T), silent = T)

##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
## format.pval, round.POSIXt, trunc.POSIXt, units

##
## Attaching package: 'SparseM'
##
## The following object is masked from 'package:base':
##
## backsolve

if(ev == FALSE){
  options(repos=c(CRAN ="http://cran-mirror.cs.uu.nl/"))
  install.packages("rms", dependencies = TRUE)
  library(rms)
}

fit.1 <- lm(hba1c ~ rcs(t2dm_y, 3), data=dt1)
anova(fit.0, fit.1) # The non-linear model fits the data better

## Analysis of Variance Table
##
## Model 1: hba1c ~ t2dm_y
## Model 2: hba1c ~ rcs(t2dm_y, 3)
## Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    152 760.02
## 2    151 711.53  1    48.485 10.289 0.001634 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Plotting the non-linear association
X <- seq(min(dt1$t2dm_y), max(dt1$t2dm_y), length.out = 100)

kn1 <- rcspline.eval(dt1$t2dm_y, nk = 3, knots.only = T)  # extracting the cubic spline transformation of the independent variable
designX <- cbind(1, rcspline.eval(X, knots = kn1, inclx = T))  # design matrix storing the predictor values
cf <- coef(fit.1)
pred1 <- matrix(cf, ncol = 3) %*% t(designX)

par(mfrow = c(1, 2))
par(mgp = c(1.5, 0.8, 0), oma = c(0, 0, 0, 0), mar = c(3, 3, 0.5, 0.5))

plot(y = dt1$hba1c, x = dt1$t2dm_y, pch = 16, xlab = "Time since T2DM diagnosis", ylab = "Average HbA1c")
lines(x = X, pred1, col = "red", lwd = lwd1)

# Plotting the new residuals ~ fitted values plot
mat <- matrix(cbind(predict(fit.1), res), ncol = 2)
plot(x = mat[, 1], y = mat[, 2], ylab = ylab1, xlab = xlab1, bty = "l", pch = 16, cex = cex2)

abline(h = 0, lwd = lwd1, col = "grey", lty = 2)  # expected line
The left graph shows that the restricted cubic splines adequately model the non-linear trend observed. The right graph shows an absence of trend between the residuals and fitted value confirming the improved model fit.

Corrections for heteroscedastic or correlated errors

It is likely this non-linear trend is a true reflection of the population association (i.e., T2DM management skills improve with time). However, an alternative explanation of the trend between the residuals and fitted values (of the linear model)
# could be heteroscedasticity or correlation between errors.
# As an illustration of how to correct for these issues, we will replace the naive standard errors
# by heteroscedastic robust standard errors.

# loading the sandwich package for the "robust" standard error function
ev <- try(require("sandwich", quietly = T), silent = T)
if(ev == FALSE){
  options(repos=c(CRAN ="http://cran-mirror.cs.uu.nl/"))
  install.packages('sandwich', dependencies = TRUE)
  library(sandwich)
}

# original model
summary(fit.0)

##
## Call:
## lm(formula = hba1c ~ t2dm_y, data = dt1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0175 -1.7462 -0.2527  1.3364  7.3545
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.60346    0.28754  29.921   <2e-16 ***
## t2dm_y       0.04280    0.02284   1.874   0.0629 .
##
## Residual standard error: 2.236 on 152 degrees of freedom
## Multiple R-squared:  0.02258,    Adjusted R-squared:  0.01615
## F-statistic: 3.512 on 1 and 152 DF,  p-value: 0.06285

# new robust standard errors compared to the naive standard errors.
(se.r <- sqrt(diag(vcovHC(fit.0))))

## (Intercept)      t2dm_y
##  0.28648195  0.02249715

summary(fit.0)$coef[,"Std. Error"]

## (Intercept)      t2dm_y
##  0.28753880  0.02284165

se.r/summary(fit.0)$coef[,"Std. Error"] # ratio between new and old standard errors

## (Intercept)      t2dm_y
##   0.9963245   0.9849177
# new robust confidence intervals compared to the naive CI's

df1 <- dim(dt1)[1] - 1 # degrees of freedom
a1 <- 0.05 # alpha, the significant cut-off value

(ci.r <- summary(fit.0)$coef[, "Estimate"] + matrix(c(-1, 1), 2, 2, byrow = T) * qt(1-a1/2, df = df1) * se.r)

##              [,1]       [,2]
## [1,]  8.037490579 9.16943250
## [2,] -0.001640349 0.08724995

confint(fit.0)

##          2.5 %     97.5 %
## (Intercept)  8.035372873 9.17155021
## t2dm_y      -0.002323315 0.08793291

# new robust p-values compared to the naive p-values
t.r <- summary(fit.0)$coef[, "Estimate"]/se.r # t-values

(pval.r <- pt(abs(t.r), df = df1, lower.tail = F) * 2)

## (Intercept) t2dm_y
## 4.966966e-66 5.896134e-02

summary(fit.0)$coef[, "Pr(>|t|)"]

## (Intercept) t2dm_y
## 1.379007e-65 6.285256e-02

# Comparing the robust standard errors to the naive standard errors
# showed little difference (the quotient was close to 1),
# confirming what we already suspected ~ the observed trend is
# most likely explained by the non-linear association of
# time since diagnosis. More generally it is often difficult to determine if
# the errors are correlated or not, especially in the case of non-time series
# data. As such external, prior knowledge is key here.
# When the rows in the data can be ordered in an informative way (e.g, as in
time-series),
# a possible data driven grafic to check dependency between errors is to simp
ly
# plot the residuals against the ordered observations (e.g., time).
# As we saw before the relation between HbA1c seemed
# to be related to duration of T2DM, and BMI.
# Furthermore, these variables themselves seemed related
# to age and marital status. To further explore the relation between HbA1c
# and duration of T2DM we will next include these variables
# in a multivariable linear model.

```r
fit.full <- lm(hba1c ~ rcs(t2dm_y, 3) + bmi + age + married, data=dt1)
anova(fit.full)
```

## Analysis of Variance Table
## Response: hba1c
## Df  Sum Sq Mean Sq  F value    Pr(>F)
## rcs(t2dm_y, 3) 2 66.04 33.022  7.1977 0.00104 **
## bmi             1 30.18 30.180  6.5782 0.01132 *
## age             1  2.07  2.069  0.4511 0.50287
## married         1  0.28  0.276  0.0602 0.80648
## Residuals      148 679.01   4.588
## ---
## Signif. codes:  *** 0.001 ** 0.01 * 0.05 . 0.1  1

# conditional on covariables, both time since diagnosis
# and bmi are significantly associated with HbA1c.
# Let's check how well this "full" model,
# fits the collected data and if
# there are potential model assumption violations.
xlab1 <- "Fitted values"
ylab1 <- "Residuals"
lwd1 <- 2
cex2 <- 0.7

res <- summary(fit.full)$res # extracting residuals
qqnorm(res, main = "", ylab = "Residuals", pch = 16, bty = "l", cex = cex2)
qqline(res, col = "red", lwd = lwd1) # line indicating perfect fit

# as before the more outlying residuals # deviate from the normal distribution

mat <- matrix(cbind(predict(fit.full), res), ncol = 2) # predict(fit.0) extracts the fitted values
plot(x = mat[,1], y = mat[,2], ylab = ylab1, xlab = xlab1, bty = "l", pch = 16, cex = cex2)
# Loess curve

loess.fit <- loess(mat[,2] ~ mat[,1])
xloess <- seq(min(mat[,1]), max(mat[,1]), length.out = 100)
yloess <- predict(loess.fit, newdata = xloess)

lines(x = xloess, yloess, col = "red", lwd = lwd1)  # add the loess curve to the graph
abline(h = 0, lwd = lwd1, col = "grey", lty = 2)  # if there is a perfect line or relation the

# residuals should cluster along the grey line

# No clear trend can be observed so we don't have to worry # about heteroscedasticity or correlated errors

# Multivariable outliers #

rm(res, mat)  # removing unstandardized residuals
sres <- rstudent(fit.full)  # extracting Studentized residuals
mat <- matrix(cbind(predict(fit.full), sres), ncol = 2)  # predict(fit.0) extract the fitted values
plot(x = mat[,1], y = mat[,2], ylab = "Studentized residuals", xlab = xlab1, bty = "l", pch = 16, cex = cex2, ylim = c(-max(abs(sres)),max(abs(sres)) ))
abline(h = c(-3,0,3), lwd = lwd1, col = "grey", lty = 2)

# extracting outlying observations
out1 <- as.numeric(as.character(rownames(dt1)))[abs(sres) > 3]  # outlying observations
yout1 <- sres[abs(sres) > 3]
xout1 <- predict(fit.full)[abs(sres) > 3]
text(y = yout1, xout1-0.1, cex = 0.7, labels = as.character(out1), font = 3)

# by using Studentized residuals we can more easily identify
# outlying HbA1c values. For example we would not expect many
# Studentized residuals larger or smaller than 3. In this graph
# we find 3 outlying values which deserve further consideration.

# Leverage #
rm(mat)
sres <- rstudent(fit.full)  # extracting Studentized residuals
leverage <- hatvalues(fit.full)

mat <- matrix(cbind(leverage, sres), ncol = 2)
plot(x = mat[,1], y = mat[,2], ylab = "Studentized residuals", xlab = "Leverage", bty = "l", pch = 16, cex = cex2, ylim = c(-max(abs(sres)), max(abs(sres))))

# adding points with outlying HbA1c values
xout1 <- leverage[abs(sres) > 3]
text(y = yout1, xout1 - 0.005, cex = 0.7, labels = as.character(out1), font = 3)

# adding points with high leverage only
cut <- 3*mean(leverage)  # heuristic rule to define 3 times the mean as extreme
lev1 <- as.numeric(as.character(rownames(dt1)))[leverage > cut]  # outlying observations
ylev1 <- sres[leverage > cut]
xlev1 <- leverage[leverage > cut]
text(y = ylev1, xlev1-0.005, cex = 0.7, labels = as.character(lev1), font = 3, col = "red")
# It turns out that from the previously defined outlying values 150 also has a slightly high leverage.
# Furthermore, we identified 3 observations with high leverage without outlying HbA1c values.

dt1[rownames(dt1) %in% c(lev1,out1),]

<table>
<thead>
<tr>
<th></th>
<th>edu_y</th>
<th>hba1c</th>
<th>age</th>
<th>sex</th>
<th>married</th>
<th>t2dm_y</th>
<th>bmi</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>6</td>
<td>16.6</td>
<td>59</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>15 28.22</td>
</tr>
<tr>
<td>35</td>
<td>6</td>
<td>16.8</td>
<td>59</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>20 30.08</td>
</tr>
<tr>
<td>80</td>
<td>5</td>
<td>10.5</td>
<td>54</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15 49.98</td>
</tr>
<tr>
<td>85</td>
<td>6</td>
<td>7.5</td>
<td>65</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>35 34.41</td>
</tr>
<tr>
<td>97</td>
<td>12</td>
<td>15.4</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3   50.59</td>
</tr>
<tr>
<td>150</td>
<td>6</td>
<td>15.8</td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3   41.19</td>
</tr>
</tbody>
</table>
# It turns out that the observations with high leverage (80, 85, 97)
# are related to a long history with T2DM and/or a high BMI value.
# The outlying observations are (not surprisingly) related to a
# relatively high HbA1c values (which are naturally difficult to model)

# In itself this having extreme outcome or predictor values
# does not mandate removal. In fact every analysis is expected to have a few
# unusual observations. Clustering of such observations would be more worrisome,
# especially if these unusual observations would be outliers and have high leverage.
# Where possible it is however important to check whether these observations
# are due to errors
# in data collection.

################################
# Short-cut for diagnostic plots#
################################

# Manually creating these plots is great fun and a useful skill
# if one would want to include these in a publication. However,
# to quickly check the modelling assumptions one can simply use
# the plot() function

diagnostic plots
par(mfrow = c(2, 2))
plot(fit.full)
# note that the bottom left most plot compares the Studentized residuals
# against leverage (as done above as well), and additionally provides
# Cook's distance which indicates observations with both high
# leverage and high residual (outliers). The Cook's distance
# indicates that there is little reason to worry about the
# observations previously highlighted as "unusual".