Math 423
Linear Regression

Final Project

Frédéric Boileau

Prof. David A. Stephens

20th December 2016

Introduction

The course MATH423 could be said to culminate in a concise introduction to different approches to effective statistical model building, generally restricted to multiple linear regression under standard assumptions. In this final project we use data from a large ongoing survey in the United States which provides real publicly available data. Using this data, we have attempted to build a suitable model to represent the variation in blood pressure in terms of the other available potential predictors. Moreover we have focussed on data from the 2011-2012 year. The data on blood pressure was stored in two different variables. They are named "bpdia" and "bpsys" which stands for diastolic and systolic blood pressure. They measure in mm Hg the blood pressure when the heart muscle is contracting and when it is relaxed respectively.

```
library(pander)
library(xtable)
library("NHANES")
library(car)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

## Error:  RStudio not running

survey = read.csv("nhanes-sub.csv",header = TRUE)
```

1 Data inspection and pre-processing

We first start by inspecting the data and pre-process it so we can apply the regular techniques of model building and standard regression functions available in R. We get rid of the identifiers for the regression as they clearly by definition not predictors. Moreover we remove the column "year" from the dataframe as the sample is only concerned with one of those years; a factor variable with only one variable beeing completely useless in regression analysis as well. We also decide to scale the data because of the large different in scale between different potential predictors. Moreover the intercept makes more sense for scaled data when dealing with obviously non-negative data which is clearly in a certain range as it indicated conditional expected value for $x_i = \mu_i$ and not $x_i = 0$. The latter beeing quite meaningless for height or blood pressure.

```
isFact <- sapply(survey, is.factor)
survey[isFact] = lapply(survey[isFact],factor)
survey = survey[, names(survey) != "year"]
unwantedCols = c("bpdia","X","id")
isNum <- sapply(survey, is.numeric)
survey[isNum] <- lapply(survey[isNum], scale)
```

There are two variables dealing with race which differ only in the levels available, the race3 one having more levels. If we were to select one as a predictor, it is obvious the other one should not be found in the model.

2 Diastolic blood pressure

```
coln = colnames(survey)
f <- as.formula(paste("bpdia ~", paste(coln[!coln %in% unwantedCols], collapse = "+")))
fit.fullAdd = lm(f, survey)
fit.fpredictors_T = as.data.frame(summary(fit.fullAdd)$coefficients)
```

```
library(xtable)
xtable(fit.fpredictors_T[fit.fpredictors_T$`Pr(>|t|)`>0.5,])
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.17	0.25	0.67	0.50
race1Mexican	0.06	0.10	0.61	0.54
hhinc10000-14999	-0.11	0.18	-0.59	0.55
hhinc35000-44999	-0.11	0.18	-0.64	0.52
hhinc75000-99999	-0.03	0.19	-0.14	0.89
hhincmore 99999	-0.08	0.20	-0.39	0.70
homeownOwn	-0.07	0.13	-0.53	0.60
homeownRent	-0.08	0.13	-0.56	0.57
dirchol	-0.01	0.02	-0.64	0.53
physactYes	-0.00	0.04	-0.00	1.00

```
xtable(fit.fpredictors_T[fit.fpredictors_T$`Pr(>|t|)`<0.05,])
```

	Estimate	Std. Error	t value	Pr(>|t|)
age	-0.16	0.03	-5.58	0.00
race3Other	-0.54	0.16	-3.29	0.00
marriedLivePartner	-0.22	0.09	-2.35	0.02
marriedMarried	-0.22	0.07	-3.13	0.00
marriedNeverMarried	-0.32	0.08	-3.90	0.00
marriedWidowed	-0.57	0.12	-4.89	0.00
hhinc15000-19999	-0.40	0.18	-2.17	0.03
hhinc20000-24999	-0.37	0.18	-2.01	0.04
hhinc45000-54999	-0.36	0.18	-2.00	0.05
pulse	0.11	0.02	5.55	0.00
bpsys	0.46	0.02	21.32	0.00
totchol	0.16	0.02	7.97	0.00

```r
fit.1 = lm(bpdia ~ bpsys+totchol,survey)
fit.2 = lm(bpdia ~ bpsys*totchol,survey)
anova(fit.1,fit.2)

## Analysis of Variance Table
##
## Model 1: bpdia ~ bpsys + totchol
## Model 2: bpdia ~ bpsys * totchol
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   2101 1679.9
## 2   2100 1679.3  1   0.63706 0.7967 0.3722

fit.3 = lm(bpdia ~ bpsys + totchol + pulse + age, survey)
anova(fit.1,fit.3)

## Analysis of Variance Table
##
## Model 1: bpdia ~ bpsys + totchol
## Model 2: bpdia ~ bpsys + totchol + pulse + age
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2101 1679.9
## 2   2099 1614.5  2    65.347 42.477 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fit.4 = lm(bpdia ~ bpsys*totchol*pulse*age,survey)
anova(fit.3,fit.4)

## Analysis of Variance Table
##
## Model 1: bpdia ~ bpsys + totchol + pulse + age
## Model 2: bpdia ~ bpsys * totchol * pulse * age
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2099 1614.5
## 2   2088 1516.5 11    98.035 12.271 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fit.5 = update(fit.4, ~. + weight + bmi + hhinc + poverty)
```

Before adding any more variables we look at plausible multicollinearity.

```
library(car)
inflation = as.data.frame(vif(fit.5))
inflation[inflation[,"GVIF"] > 2,]
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## weight        4.696054  1        2.167038
## bmi           4.574553  1        2.138821
## hhinc         5.436117 11        1.079996
## poverty       4.490268  1        2.119025
## bpsys:totchol 2.054289  1        1.433279
```

```
fit.rem_1_1 = update(fit.5, ~. - weight)
fit.rem_1_2 = update(fit.5, ~. - bmi)
fit.rem_1_3 = update(fit.5, ~.  - hhinc)
fit.rem_1_4 = update(fit.5, ~. - poverty)
predictors_1_1 = as.data.frame(summary(fit.rem_1_1)$coefficients)
predictors_1_2 = as.data.frame(summary(fit.rem_1_2)$coefficients)
predictors_1_3 = as.data.frame(summary(fit.rem_1_3)$coefficients)
predictors_1_4 = as.data.frame(summary(fit.rem_1_4)$coefficients)
predictors_1_1["bmi",]
```

```
##       Estimate Std. Error t value  Pr(>|t|)
## bmi 0.02576644 0.01931112 1.33428 0.1822586
```

```
predictors_1_2["weight",]
```

```
##          Estimate Std. Error  t value    Pr(>|t|)
## weight 0.05680843 0.01953452 2.908105 0.003674894
```

```
predictors_1_3["poverty",]
```

```
##          Estimate Std. Error  t value     Pr(>|t|)
## poverty 0.0750112 0.01966544 3.814367 0.0001405028
```

```
predictors_1_4[grep('^hhinc',rownames(predictors_1_4)),]
```

```
##                       Estimate Std. Error     t value   Pr(>|t|)
## hhinc 5000-9999  -0.101516220  0.2034493 -0.49897555 0.61784950
## hhinc10000-14999 -0.076978345  0.1727306 -0.44565563 0.65589251
## hhinc15000-19999 -0.360332117  0.1756688 -2.05120206 0.04037258
## hhinc20000-24999 -0.330930392  0.1723662 -1.91992598 0.05500421
## hhinc25000-34999 -0.108672152  0.1623298 -0.66945287 0.50328112
## hhinc35000-44999 -0.037409304  0.1621899 -0.23065131 0.81760844
## hhinc45000-54999 -0.256570857  0.1635162 -1.56908559 0.11678047
## hhinc55000-64999 -0.124529197  0.1673558 -0.74409863 0.45690106
## hhinc65000-74999 -0.255095749  0.1658689 -1.53793575 0.12421679
## hhinc75000-99999  0.069877024  0.1595372  0.43799827 0.66143313
## hhincmore 99999   0.006631136  0.1543159  0.04297117 0.96572865
```

We have looked at the full additive model first. Then we chose the six predictors which were the most significant according to the t test values. We inspected the coefficients with both high and low p values and found that hhinc had levels in the two extreme categories. Moreover we supposed there should be some multicollinearity between house hold income and the poverty index and bmi and weight. We confirmed this by fitting both and then looking at the VIF. At the same time we actually checked multicollinearity in the rest of the model too. The test reveled that there was indeed heavy multicollinearity between hhinc and poverty but not in the rest. Moreover hhinc is a binned predictor whereas poverty is a continuous variable. As we don't have a solid grasp of binning we were naturally inclined to choose poverty over hhinc. Finally poverty seemed by a simple anova to have slightly more predictive power. The thus chosen main variables were our basis for foward selection. We then added interactions iteratively and ended up including all of them.

```
fit.4_int = lm(bpdia ~weight*poverty*bpsys*totchol*pulse*age,survey)
s = step(fit.4_int,direction="backward")

## Start:  AIC=-682.89
## bpdia ~ weight * poverty * bpsys * totchol * pulse * age
##
##                                           Df Sum of Sq    RSS     AIC
## <none>                                                  1431.1 -682.89
## - weight:poverty:bpsys:totchol:pulse:age  1    7.1022 1438.2 -674.47

tail(s$coefficients,2)

##        poverty:bpsys:totchol:pulse:age
##                             -0.03008054
## weight:poverty:bpsys:totchol:pulse:age
##                             -0.09732629

fit.5 = lm(bpdia ~ weight*poverty*bpsys*totchol*pulse*age,survey)
```

Since we had only 6 variables at this point we were able to use the function step to find the best model according to AIC values amongst all the models with those 6 predictors. This model turns out to have high order interactions and so we refrained from trying to add them "manually" one by one and concluded that the model with all interactions was the best one achievable "manually" for those 6 variables.

3 Systolic Blood Pressure

```
coln = colnames(survey)
unwantedCols = c("bpsys", "X", "id")
f <- as.formula(paste("bpsys ~", paste(coln[!coln %in% unwantedCols], collapse = "+")))
fit.fullAdd = lm(f, survey)
fit.fpredictors_T = as.data.frame(summary(fit.fullAdd)$coefficients)
```

```
library(xtable)
xtable(fit.fpredictors_T[fit.fpredictors_T$`Pr(>|t|)`>0.5,])
```

	Estimate	Std. Error	t value	Pr(>|t|)
marriedSeparated	0.09	0.13	0.66	0.51
hhinc 5000-9999	-0.00	0.19	-0.02	0.98
hhinc10000-14999	0.08	0.16	0.48	0.63
hhinc15000-19999	0.00	0.17	0.01	0.99
hhinc20000-24999	0.05	0.17	0.30	0.76
pulse	0.01	0.02	0.48	0.63

```
xtable(fit.fpredictors_T[fit.fpredictors_T$`Pr(>|t|)`<0.02,])
```

	Estimate	Std. Error	t value	Pr(>|t|)
gendermale	0.22	0.05	4.23	0.00
age	0.43	0.02	17.71	0.00
race1Mexican	-0.26	0.09	-2.80	0.01
race1Other	-0.28	0.11	-2.49	0.01
race1White	-0.23	0.06	-3.63	0.00
marriedMarried	0.15	0.07	2.33	0.02
marriedNeverMarried	0.35	0.07	4.64	0.00
marriedWidowed	0.42	0.11	3.90	0.00
hhinc45000-54999	0.44	0.17	2.65	0.01
hhinc65000-74999	0.51	0.18	2.86	0.00
poverty	-0.18	0.04	-4.78	0.00
bmi	0.18	0.05	3.70	0.00
bpdia	0.39	0.02	21.32	0.00
alcday	0.07	0.02	3.43	0.00
alcyear	0.07	0.02	3.52	0.00

```
fit.11 = lm(bpsys ~ bpdia+age+gender,survey)
fit.21 = lm(bpsys ~ bpdia*age*gender,survey)
anova(fit.11,fit.21)

## Analysis of Variance Table
##
## Model 1: bpsys ~ bpdia + age + gender
## Model 2: bpsys ~ bpdia * age * gender
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2100 1423.8
## 2   2096 1386.1  4    37.689 14.248 1.771e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fit.31 = update(fit.21, ~. + alcyear + alcday + bmi)
```

Before adding any more variables we look at plausible multicollinearity.

```
library(car)
inflation = as.data.frame(vif(fit.11))
inflation

##        vif(fit.11)
## bpdia     1.013722
## age       1.002918
## gender    1.011390

inflation = as.data.frame((vif(lm(bpsys ~ alcyear + alcday,survey))))
inflation

##         (vif(lm(bpsys ~ alcyear + alcday, survey)))
## alcyear                                    1.003205
## alcday                                     1.003205

fit.41 = lm(bpsys ~ bpdia*age*gender*alcyear*alcday*bmi,survey)
anova(fit.31,fit.41)

## Analysis of Variance Table
##
## Model 1: bpsys ~ bpdia + age + gender + alcyear + alcday + bmi + bpdia:age +
##     bpdia:gender + age:gender + bpdia:age:gender
## Model 2: bpsys ~ bpdia * age * gender * alcyear * alcday * bmi
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2093 1342.1
## 2   2040 1198.9 53    143.19 4.5969 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
s = step(fit.41,direction = "backward")

## Start:  AIC=-1055.34
## bpsys ~ bpdia * age * gender * alcyear * alcday * bmi
##
##                                         Df Sum of Sq    RSS     AIC
## <none>                                               1198.9 -1055.3
## - bpdia:age:gender:alcyear:alcday:bmi  1    11.774 1210.7 -1036.8

tail(s$coefficients,1)

## bpdia:age:gendermale:alcyear:alcday:bmi
##                                0.726011
```

The procedure was the same as the one outlined for diastolic pressure but multicollinearity was not a potential issue this time. Once again the best model was found to be the one with all the possible interactions of the predictors with the biggest main effect.

Conclusion

We have tried to find by model building techniques seen in class the "best" linear model to describe the response variables diastolic and systolic blood pressure with the data from "NHANES" in the 2010-2011 year. In both cases we started by adding the main effects of the most powerful predictors and then looking at possible interactions. We ended up adding all the interactions between the 6 most powerful predictors. Throughout the procedure we checked for multicollinearity. It was detected between the pairs bmi and weight and poverty and household income which sounded very reasonable. In this respect we only included one of those when they were both powerful. However we did not find multicollinearity between alcohol consumption as measured by year or by day to be a problem and added both to the second model. We confirmed the validity of those two models by automatic backward elimination with the function "step". This latter finds the best possible model according to AIC values between all possible models (i.e. linear regressions based on all possible subsets of the set of all interactions between main effects.) This indicated that including the interactions was a good choice as the AIC includes a penalty for over fitting. However this is an approach closer to data mining than actual statistical model building. To build a good model one should be informed by theoretical knowledge in order to explore possible interactions. We did not have that knowledge and therefore chose to include all the terms which might violate the principle of parsimony. We did not find this to be a major issue for the number of predictors we had (6), the resulting models weren't complicated to an absurd extent.