Stats 101A Project: Final First Draft

Simran Vatsa, Naren Akurati, Sohom Paul, Ashwin Ayyasamy, Jeremy Phan

3/16/2018

For each variable, our first step was consulting the codebook to decide which codes could be converted to NAs. We decided that "not answered" could in every case be classified as NA. We also converted the variables coded to denote "_ or more" to NAs as we felt those might skew our analysis. We did not convert 8 (8 or more) in Household or Children, and we converted 0 (Inapplicable) and 8 (Don't know) to 2 (No) in Instagram. With these new conditions, we repeated plotting the full model, carrying out more complete analyses on it this time. Its R squared value was 0.2811, with an adjusted R squared of 0.2069.

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 3.4.3
```

The first conclusion we came to in our model selection process was that WorkHrs could be excluded, as there were 1898 missing values, most of which were -1 (Inapplicable). Its sheer amount of missing values made it ineligible for model fitting - every attempt to include it resulted in an error being thrown.

```r
# Finding the number of NAs
sum(is.na(happiness_data$WorkHrs))
```

```
## [1] 1898
```

```r
# Insta = 10, Marital = 1; #Household = 1; #Health = 811;
# #OwnHome = 812; #JobSat = 1612; #WorkHrs = 1898; #Income =
# 1039
```

We plotted the full model (without WorkHrs). This factored in around 190 observations, as all the rest had NAs under some variables. We found the Residuals vs Fitted plot showed a linear trend, a result of some of the predictor variables being categorical. The standardized residual plot also showed a pattern that skewed the plot much more than it did in the Residuals vs. Fitted plot.

```r
attach(happiness_data)
# Factoring Categorical Variables
JobSat.f <- factor(JobSat)
OwnHome.f <- factor(OwnHome)
Marital.f <- factor(Marital)
Instagram.f <- factor(Instagram)
Health.f <- factor(Health)
# Couldn't include Health as it throws an error
full_model <- lm(Happy ~ Household + OwnHome.f + Instagram.f +
    Marital.f + Children + Education + JobSat.f + Income + Age +
    Sex)
```

We decided the best way to proceed would be to test each variable's significance individually. We created models with individual variables and Happy, finding that Instagram and Sex did not have statistically significant linear relationships to Happy.

```r
insta <- lm(Happy ~ Instagram)
# summary(insta); plot(insta); Instagram is insignificant
marital <- lm(Happy ~ Marital.f)
# summary(marital); plot(marital); Marital is significant
Job <- lm(Happy ~ JobSat.f)
# summary(Job); plot(Job); Job is significant
```

```
House <- lm(happiness_data$Happy ~ happiness_data$Household)
# summary(House); plot(Household); Household is significant
sex <- lm(Happy ~ Sex)
# summary(sex); plot(sex); Sex is insignificant
```

We then used partial F-tests to verify these findings, as well as potentially weed out other variables. To do so, we created models that each excluded one variable and then tested them against our full model. This method found Children and Education to be insignificant in addition to Instagram and Sex. OwnHome, JobSat, Income and Age threw errors in partial F-testing, so we have not included the code for those.

```
noMarital <- lm(Happy ~ Sex + JobSat.f + OwnHome.f + Household +
    Education + Age + Income + Children + Instagram.f)
anova(full_model, noMarital)$`Pr(>F)`
```

```
## [1]          NA 0.003016323
```

```
# Marital is significant
noHousehold <- lm(Happy ~ Sex + JobSat.f + OwnHome.f + Marital.f +
    Education + Age + Income + Children + Instagram.f)
anova(full_model, noHousehold)$`Pr(>F)`
```

```
## [1]          NA 0.003873214
```

```
# Household is significant
noSex <- lm(Happy ~ JobSat.f + OwnHome.f + Household + Marital.f +
    Education + Age + Income + Children + Instagram.f)
anova(full_model, noSex)$`Pr(>F)`
```

```
## [1]        NA 0.2419433
```

```
# Sex is insignificant
noInstagram <- lm(Happy ~ Sex + JobSat.f + OwnHome.f + Household +
    Marital.f + Education + Age + Income + Children)
anova(full_model, noInstagram)$`Pr(>F)`
```

```
## [1]        NA 0.4980311
```

```
# Instagram is insignificant
noChildren <- lm(Happy ~ Sex + JobSat.f + OwnHome.f + Household +
    Marital.f + Education + Age + Income + Instagram.f)
anova(full_model, noChildren)$`Pr(>F)`
```

```
## [1]        NA 0.4352332
```

```
# Children is insignificant
noEducation <- lm(Happy ~ Sex + JobSat.f + OwnHome.f + Household +
    Marital.f + Age + Income + Children + Instagram.f)
anova(full_model, noEducation)$`Pr(>F)`
```

```
## [1]        NA 0.2048761
```

```
# Education is insignificant
```

After eliminating the four insignificant variables, we obtained AIC, AICc and BIC values, which were lowest when all six variables were included. Performing forward selection showed OwnHome to be insignificant and performing backward selection showed Age to be insignificant. Since the forward and backward selections were not in agreement, this did not seem like strong enough evidence to exclude the variables to us. We found including all 6 variables gave us the lowest values for each, so we did not choose to omit any variables from the model in this process.

```r
# Eliminating education, instagram, children, sex
new_model <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Income + Age)

Rad <- summary(new_model)$adj.r.squared
om1 <- lm(Happy ~ JobSat.f)
om2 <- lm(Happy ~ JobSat.f + OwnHome.f)
om3 <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f)
om4 <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household)
om5 <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Age)
om6 <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Age + Income)
n = length(Happy)

p <- 1
oms1 <- summary(om1)
AIC1 <- extractAIC(om1, k = 2)[2]  # AIC
AICc1 <- extractAIC(om1, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC1 <- extractAIC(om1, k = log(n))[2]  # BIC
p <- 2
oms2 <- summary(om2)
AIC2 <- extractAIC(om2, k = 2)[2]  # AIC
AICc2 <- extractAIC(om2, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC2 <- extractAIC(om2, k = log(n))[2]  # BIC
p <- 3
oms3 <- summary(om3)
AIC3 <- extractAIC(om3, k = 2)[2]  # AIC
AICc3 <- extractAIC(om3, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC3 <- extractAIC(om3, k = log(n))[2]  # BIC
p <- 4
oms4 <- summary(om4)
AIC4 <- extractAIC(om4, k = 2)[2]  # AIC
AICc4 <- extractAIC(om4, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC4 <- extractAIC(om4, k = log(n))[2]  # BIC
p <- 5
oms5 <- summary(om5)
AIC5 <- extractAIC(om5, k = 2)[2]  # AIC
AICc5 <- extractAIC(om5, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC5 <- extractAIC(om5, k = log(n))[2]  # BIC
p <- 6
oms6 <- summary(om6)
AIC6 <- extractAIC(om6, k = 2)[2]  # AIC
AICc6 <- extractAIC(om6, k = 2)[2] + 2 * (p + 2) * (p + 3)/(n -
    p - 1)  # AICc
BIC6 <- extractAIC(om6, k = log(n))[2]  # BIC

AIC <- c(AIC1, AIC2, AIC3, AIC4, AIC5, AIC6)
```

```
AICc <- c(AICc1, AICc2, AICc3, AICc4, AICc5, AICc6)
BIC <- c(BIC1, BIC2, BIC3, BIC4, BIC5, BIC6)

opmodel <- data.frame(Size = 1:6, Radj2 = Rad, AIC = AIC, AICc = AICc,
    BIC = BIC)
opmodel
```

```
##   Size     Radj2       AIC      AICc       BIC
## 1    1 0.2013563 -721.0678 -721.0577 -680.6822
## 2    2 0.2013563 -286.4328 -286.4159 -234.5084
## 3    3 0.2013563 -293.8557 -293.8303 -218.8538
## 4    4 0.2013563 -298.4634 -298.4278 -217.6921
## 5    5 0.2013563 -296.6381 -296.5906 -210.0974
## 6    6 0.2013563 -239.8424 -239.7814 -147.5323
```

```
# Lowest AIC, AICc, BIC values occur when size = 6. Thus, we
# are retaining all variables

# Checking Forward Selection
add1(lm(Happy ~ 1), Happy ~ JobSat.f + OwnHome.f + Marital.f +
    Household + Age + Income, test = "F")$`Pr(>F)`  #prints p-value
```

```
## Warning in add1.lm(lm(Happy ~ 1), Happy ~ JobSat.f + OwnHome.f + Marital.f
## + : using the 204/2361 rows from a combined fit
```

```
## [1]           NA 1.728212e-73 6.417063e-05 2.519737e-43 3.824285e-08
## [6] 4.212786e-01 5.915053e-32
```

```
# Age seems to be insignificant

# Performing another forward selection to see if age is
# actually insignificant
add1(lm(Happy ~ JobSat.f), Happy ~ JobSat.f + OwnHome.f + Marital.f +
    Household + Age + Income, test = "F")$`Pr(>F)`  #prints p-value
```

```
## Warning in add1.lm(lm(Happy ~ JobSat.f), Happy ~ JobSat.f + OwnHome.f + :
## using the 204/754 rows from a combined fit
```

```
## [1]           NA 4.392858e-02 1.009862e-09 1.578902e-03 5.276440e-01
## [6] 6.963063e-07
```

```
# Backward Selection to see check the significance of
# variables
drop1(lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Age + Income), test = "F")$`Pr(>F)`  #prints p-value
```

```
## [1]           NA 0.0008187357 0.1868556335 0.0037590269 0.0034175331
## [6] 0.0239082085 0.0385886570
```

```
# Ownhome seems to be insignificant
drop1(lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Income), test = "F")$`Pr(>F)`  #prints p-value
```

```
## [1]           NA 0.0009888392 0.3642221333 0.0139487313 0.0153868191
## [6] 0.0637063446
```

Our untransformed model thus contained JobSat, OwnHome, Marital and Household in factor form, as well as the numerical variables Income and Age. Its R squared value was 0.2844, and its adjusted R squared value improved to 0.2105.

```
Household.f <- factor(Household)

Final_model <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f +
    Household.f + Income + Age)
summary(Final_model)$r.squared
```

```
## [1] 0.2844382
```

```
summary(Final_model)$adj.r.squared
```

```
## [1] 0.2105487
```

We used both the Box Cox and inverse response plot methods in transforming our model. Per Box Cox, we raised Age to the power of 0.226. Intuitively, looking at the relationship between Income and Happy, we decided on a logarithmic transformation on Income, as well. The adjusted R squared value reduced a little from this transformation, to 0.2099, and R squared dropped to 0.2838. The inverse response plot suggested a lambda of -0.1130549, but the RSS for a lambda of 0 was very similar, so we took the log transformation of our response variable instead, as it it represented a better representation of real world data. We ended with an R^2 value of 0.3092 and an adjusted R^2 of 0.2378.

```
# Box Cox transformation
powerTransform(cbind(JobSat.f, OwnHome.f, Marital.f, Household.f,
    Income, Age) ~ 1)
```

```
## Estimated transformation parameters
##    JobSat.f   OwnHome.f   Marital.f Household.f      Income         Age
##   0.1523107  -1.9515759   0.2229948  -0.2117924   0.2160991   0.4521715
```

```
Age_transformed <- Age^0.226
m_new <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household.f +
    log(Income) + Age_transformed)
# summary(m_new)

# Inverse response plot
par(mfrow = c(2, 2))
m_new <- lm(log(Happy) ~ JobSat.f + OwnHome.f + Marital.f + Household.f +
    log(Income) + Age_transformed)
summary(m_new)
```

```
##
## Call:
## lm(formula = log(Happy) ~ JobSat.f + OwnHome.f + Marital.f +
##     Household.f + log(Income) + Age_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85310 -0.15026 -0.00986  0.17885  0.54401
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.26028    0.39631   3.180 0.001728 **
## JobSat.f2       -0.06340    0.05695  -1.113 0.267085
## JobSat.f3       -0.16091    0.06045  -2.662 0.008453 **
## JobSat.f4       -0.12654    0.10126  -1.250 0.213043
## JobSat.f5       -0.34313    0.08838  -3.882 0.000144 ***
## JobSat.f6       -0.48979    0.13130  -3.730 0.000255 ***
## JobSat.f7       -0.13356    0.27702  -0.482 0.630291
## OwnHome.f2       0.02446    0.04382   0.558 0.577330
## OwnHome.f3       0.24536    0.20878   1.175 0.241428
## Marital.f2      -0.40775    0.10880  -3.748 0.000239 ***
## Marital.f3      -0.15417    0.06445  -2.392 0.017759 *
## Marital.f4      -0.13232    0.12918  -1.024 0.307024
## Marital.f5      -0.18972    0.06156  -3.082 0.002374 **
## Household.f2    -0.09814    0.05651  -1.736 0.084157 .
## Household.f3    -0.03689    0.07957  -0.464 0.643503
## Household.f4    -0.45965    0.11490  -4.000 9.16e-05 ***
## Household.f5    -0.23737    0.20499  -1.158 0.248370
## Household.f6    -0.40293    0.19714  -2.044 0.042389 *
## log(Income)      0.03511    0.01981   1.772 0.077969 .
## Age_transformed -0.23907    0.14813  -1.614 0.108244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2675 on 184 degrees of freedom
##   (2163 observations deleted due to missingness)
## Multiple R-squared:  0.3092, Adjusted R-squared:  0.2378
## F-statistic: 4.334 on 19 and 184 DF,  p-value: 6.092e-08
```

Now we see how many bad leverage points we have. We only have two bad leverage points. We conclude that our final model is accurate.

```
StanRes1 <- rstandard(m_new); leverage1 <- hatvalues(m_new); cookd1 <- cooks.distance(m_new); p <- 7; n
a <- which(StanRes1 > 2 | StanRes1 < -2); b <- which(leverage1 > 2*(p+1)/n)
intersect(a, b)
```

```
## [1]  44  62  72 108 145 170 172
```