Stats 101A Project

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

3/23/2018

Data Cleanup

Consulted codebook to decide which codes could be converted to NAs. Changed "not answered" to NA, as that is information we do not have. Also converted variables coded to denote "_ or more" to NAs, as that is information we do not have and cannot create. 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.

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

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

First Model

We start with the full model with everything except *Health* and *WorkHrs* predictors. *Health* and *WorkHrs* predictors throw errors due to large number of NAs (811 and 1898 NAs respectively) and few categories. R^2 currently at 0.2811438 and R^2_{adj} at 0.2069141.

```r
attach(happiness_data)
JobSat.f <- factor(JobSat)
OwnHome.f <- factor(OwnHome)
Marital.f <- factor(Marital)
Instagram.f <- factor(Instagram)
Health.f <- factor(Health)
Household.f <- factor(Household)
Children.f <- factor(Children)
Sex.f <- factor(Sex)

full_model <- lm(Happy ~ Household.f + OwnHome.f + Instagram.f +
    Marital.f + Children.f + Education + JobSat.f + Income +
    Age + Sex.f)

sum(is.na(Health))
```

```
## [1] 811
```

```r
sum(is.na(WorkHrs))
```

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

```r
summary(full_model)$r.squared
```

```
## [1] 0.317573
```

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

```
## [1] 0.2038351
```

Transformation

Transformation of numerical variables *Education*, *Income*, and *Age* using powertransform. Understanding of the effects of wealth lead us to use log transformation of *Income* predictor which proved more effective than the estimated transformation parameter. Inverse rseponse plot suggested lambda close to 0. As such, we took log(*Happy*) for a simpler model. R^2 currently at 0.3518696 and R^2_{adj} at 0.2438479.

```
# Power transformation
powerTransform(cbind(Household.f, OwnHome.f, Instagram.f, Marital.f,
    Children.f, Education, JobSat.f, Income, Age, Sex.f) ~ 1)
```

```
## Estimated transformation parameters
## Household.f   OwnHome.f Instagram.f   Marital.f  Children.f   Education
##  -0.2139927  -1.9783865   6.3174335   0.2855726  -0.1489214   0.7943619
##    JobSat.f      Income         Age       Sex.f
##   0.1400369   0.2140292   0.3192108  -1.2017747
```

```
Education_transformed <- Education^0.7943619
Income_transformed <- Income^0.2140292
Income_log <- log(Income)
Age_transformed <- Age^0.3192108

full_model_transform_log <- lm(Happy ~ Household.f + OwnHome.f +
    Instagram.f + Marital.f + Children.f + Education_transformed +
    JobSat.f + Income_log + Age_transformed + Sex.f)

summary(full_model_transform_log)$r.squared
```

```
## [1] 0.3211593
```

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

```
## [1] 0.2080192
```

```
# Inverse response plot
par(mfrow = c(2, 1))
inverseResponsePlot(full_model_transform_log, key = TRUE)
```

```
##       lambda      RSS
## 1 -0.1711872 15.37896
## 2 -1.0000000 15.61674
## 3  0.0000000 15.39022
## 4  1.0000000 15.86280
```

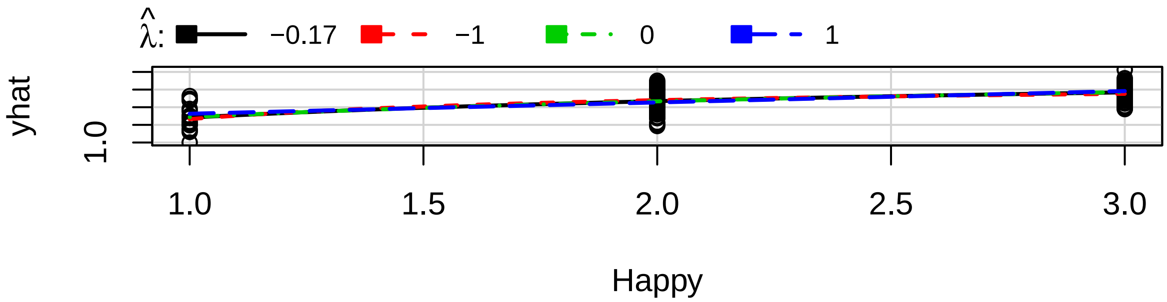
```
full_model_transform_log_inverse_response <- lm(log(Happy) ~
    Household.f + OwnHome.f + Instagram.f + Marital.f + Children.f +
        Education_transformed + JobSat.f + Income_log + Age_transformed +
        Sex.f)

summary(full_model_transform_log_inverse_response)$r.squared
```

```
## [1] 0.3518696
```

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

```
## [1] 0.2438479
```



Cursory Variable Selection

We look at number of NAs in our predictors. *OwnHome*, *JobSat*, and *Income* all have a high number of NAs (812, 1612, and 1039 respectively). From summary, predictors showing p-values over 0.05 are *OwnHome*, *Instagram*, *Marital*, *Children*, *Education*, and *Age*. These may need to be removed.

```
df_NA_count <- data.frame(c(sum(is.na(Household.f)), sum(is.na(OwnHome.f)),
    sum(is.na(Instagram.f)), sum(is.na(Marital.f)), sum(is.na(Children.f)),
    sum(is.na(Education_transformed)), sum(is.na(JobSat.f)),
    sum(is.na(Income_log)), sum(is.na(Age_transformed)), sum(is.na(Sex.f))))
```

Partial F-test - Further Variable Selection

We start with manual F-tests based on backward selection (removing the least significant variables first each iteration). We remove all insigificant variables (*Instagram*, *Children*, *OwnHome*, *Sex*, and *Age*). Our R^2 and R^2_{adj} values dropped significantly to 0.1718834 and 0.1467889 respectively, but we do this in order to avoid overfitting the data.

```
full <- drop1(full_model_transform_log_inverse_response, test = "F")
reduced_1 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f), test = "F")
reduced_2 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f), test = "F")
reduced_3 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f - OwnHome.f), test = "F")
reduced_4 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f - OwnHome.f - Sex.f), test = "F")
reduced_5 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f - OwnHome.f - Sex.f - Age_transformed),
    test = "F")
reduced_6 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f - OwnHome.f - Sex.f - Age_transformed -
        Education_transformed), test = "F")
```

```
updated_model <- full_model_transform_log_inverse_response <- lm(log(Happy) ~
    Household.f + Marital.f + Education_transformed + JobSat.f +
        Income_log)

summary(updated_model)$r.squared
```

```
## [1] 0.1718834
```

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

```
## [1] 0.1467889
```

Interaction Terms

We tested a new model with all the possible interaction terms. Based on the summary, we elminated all the insignificant interaction terms and finally arrived at a new model with R^2 value of 0.3555 and an adjusted R^2 of 0.265.

```
# #Adding interaction terms
# full_interaction_terms <- lm(log(Happy) ~ Household.f + OwnHome.f + Marital.f + Children.f + Educatio
#
# lm(log(Happy) ~ JobSat.f + OwnHome.f + Marital.f + Household.f + log(Income) + Age_transformed + JobS
# summary(m_new1)
#
# #Deleting insignificant interaction terms
# final_model <- lm(log(Happy) ~ JobSat.f + OwnHome.f + Marital.f + Household.f + log(Income) + Age_tra
# summary(m_new2)
# plot(m_new2)
```

```
om1 <- lm(Happy ~ Household.f)
om2 <- lm(Happy ~ Household.f + Marital.f)
om3 <- lm(Happy ~ Household.f + Marital.f + Education_transformed)
om4 <- lm(Happy ~ Household.f + Marital.f + Education_transformed +
    JobSat.f)
om5 <- lm(Happy ~ Household.f + Marital.f + Education_transformed +
    JobSat.f + Income_log)
n = length(om1$residuals)

p <- 1
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
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
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
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
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
```

```r
AIC <- c(AIC1, AIC2, AIC3, AIC4, AIC5)
AICc <- c(AICc1, AICc2, AICc3, AICc4, AICc5)
BIC <- c(BIC1, BIC2, BIC3, BIC4, BIC5)

opmodel <- data.frame(Size = 1:5, AIC = AIC, AICc = AICc, BIC = BIC)
opmodel
```

```
##   Size        AIC       AICc        BIC
## 1    1 -2099.9811 -2099.9709 -2059.6162
## 2    2 -2151.4770 -2151.4600 -2088.0464
## 3    3 -2159.6305 -2159.6051 -2090.4335
## 4    4  -759.4022  -759.3665  -655.6067
## 5    5  -638.0411  -637.9936  -528.4792
```

Leverages, Outliers, and Influential Points

Now we see how many bad leverage points we have. We have 9 bad leverage points. Amongst 600 observations, this is reasonable. We conclude that our final model is accurate.

```r
StanRes1 <- rstandard(updated_model); leverage1 <- hatvalues(updated_model); cookd1 <- cooks.distance(up

a <- which(StanRes1 > 2 | StanRes1 < - 2); b <- which(leverage1 > 2*(p+1)/n)
intersect(a, b)
```

```
## [1]  30 128 165 437 472 489 521 544 588
```

Final Model

```r
summary(updated_model)
```

```
##
## Call:
## lm(formula = log(Happy) ~ Household.f + Marital.f + Education_transformed +
##     JobSat.f + Income_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91017 -0.13748  0.02525  0.20392  0.84607
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.440587   0.141815   3.107 0.001982 **
## Household.f2           0.097269   0.034371   2.830 0.004812 **
## Household.f3           0.046892   0.045818   1.023 0.306519
## Household.f4          -0.080698   0.081543  -0.990 0.322755
## Household.f5          -0.052852   0.129571  -0.408 0.683494
## Household.f6           0.041831   0.180345   0.232 0.816656
## Household.f8           0.548283   0.317432   1.727 0.084643 .
## Marital.f2            -0.175336   0.069511  -2.522 0.011915 *
## Marital.f3            -0.100056   0.040677  -2.460 0.014187 *
## Marital.f4            -0.173264   0.069700  -2.486 0.013198 *
## Marital.f5            -0.062741   0.033929  -1.849 0.064930 .
```

```
## Education_transformed  0.010691   0.009245   1.156 0.247971    
## JobSat.f2             -0.015031   0.037600  -0.400 0.689469    
## JobSat.f3             -0.129969   0.039212  -3.314 0.000974 ***
## JobSat.f4             -0.192838   0.064196  -3.004 0.002778 ** 
## JobSat.f5             -0.193365   0.060460  -3.198 0.001456 ** 
## JobSat.f6             -0.356865   0.088709  -4.023 6.49e-05 ***
## JobSat.f7             -0.305760   0.181588  -1.684 0.092743 .  
## Income_log             0.028051   0.013041   2.151 0.031886 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3075 on 594 degrees of freedom
##   (1754 observations deleted due to missingness)
## Multiple R-squared:  0.1719, Adjusted R-squared:  0.1468 
## F-statistic: 6.849 on 18 and 594 DF,  p-value: 6.962e-16
```