Stats 101A Project: Final First Draft

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

3/16/2018

We began by systematically cleaning the data. 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.

```r
# data cleanup
happiness_data <- read.table("Happiness.txt", header = TRUE)
# Household
happiness_data$Household[happiness_data$Household == 9] <- NA
# Health
happiness_data$Health[happiness_data$Health == 8 | happiness_data$Health ==
    9 | happiness_data$Health == 0] <- NA
happiness_data$Health[happiness_data$Health == 1] <- 400
happiness_data$Health[happiness_data$Health == 2] <- 300
happiness_data$Health[happiness_data$Health == 3] <- 2
happiness_data$Health[happiness_data$Health == 4] <- 1
happiness_data$Health[happiness_data$Health == 400] <- 4
happiness_data$Health[happiness_data$Health == 300] <- 3
# OwnHome
happiness_data$OwnHome[happiness_data$OwnHome == 0 | happiness_data$OwnHome ==
    8 | happiness_data$OwnHome == 9] <- NA
# Instagram - Set 'don't know' and 'inapplicable' to 'No'
happiness_data$Instagram[happiness_data$Instagram == 0 | happiness_data$Instagram ==
    8] <- 2
happiness_data$Instagram[happiness_data$Instagram == 9] <- NA
# Marital
happiness_data$Marital[happiness_data$Marital == 9] <- NA
# Age
happiness_data$Age[happiness_data$Age == 89 | happiness_data$Age ==
    98 | happiness_data$Age == 99] <- NA
# Children
happiness_data$Children[happiness_data$Children == 9] <- NA
# Education
happiness_data$Education[happiness_data$Education == 97 | happiness_data$Education ==
    98 | happiness_data$Education == 99] <- NA
# JobSat
happiness_data$JobSat[happiness_data$JobSat == 0 | happiness_data$JobSat ==
    8 | happiness_data$JobSat == 9] <- NA
# Income
happiness_data$Income[happiness_data$Income == 0 | happiness_data$Income ==
    999998 | happiness_data$Income == 999999] <- NA
# WorkHrs
happiness_data$WorkHrs[happiness_data$WorkHrs == -1 | happiness_data$WorkHrs ==
    998 | happiness_data$WorkHrs == 999] <- NA
```

```
# Happy
happiness_data$Happy[happiness_data$Happy == 0 | happiness_data$Happy ==
    8 | happiness_data$Happy == 9] <- NA
happiness_data$Happy[happiness_data$Happy == 1] <- 100
happiness_data$Happy[happiness_data$Happy == 3] <- 1
happiness_data$Happy[happiness_data$Happy == 100] <- 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). While we weren't sure whether "Inapplicable" could be considered a missing value per se, there did not seem to be any way around categorizing it so, since we had decided to treat WorkHrs as a numerical variable. Its sheer amount of missing values made it ineligible for model fitting - every attempt to include it resulted in an error being thrown.

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

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

```
# 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 the rest had NAs under some variables. We found the Residuals vs Fitted plot showed a decreasing 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.

```
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)
attach(happiness_data)
```

```
## The following objects are masked from happiness_data (pos = 3):
##
##     Age, Children, Education, Happy, Health, Household, Income,
##     Instagram, JobSat, Marital, OwnHome, Sex, WorkHrs
```

```
library(alr3)
```

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

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

```
# 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 correlated individual variables with Happy, finding that Instagram and Sex did not have statistically significant linear relationships to Happy.

```
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
```

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.

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

```
## [1] 0.2603694
```

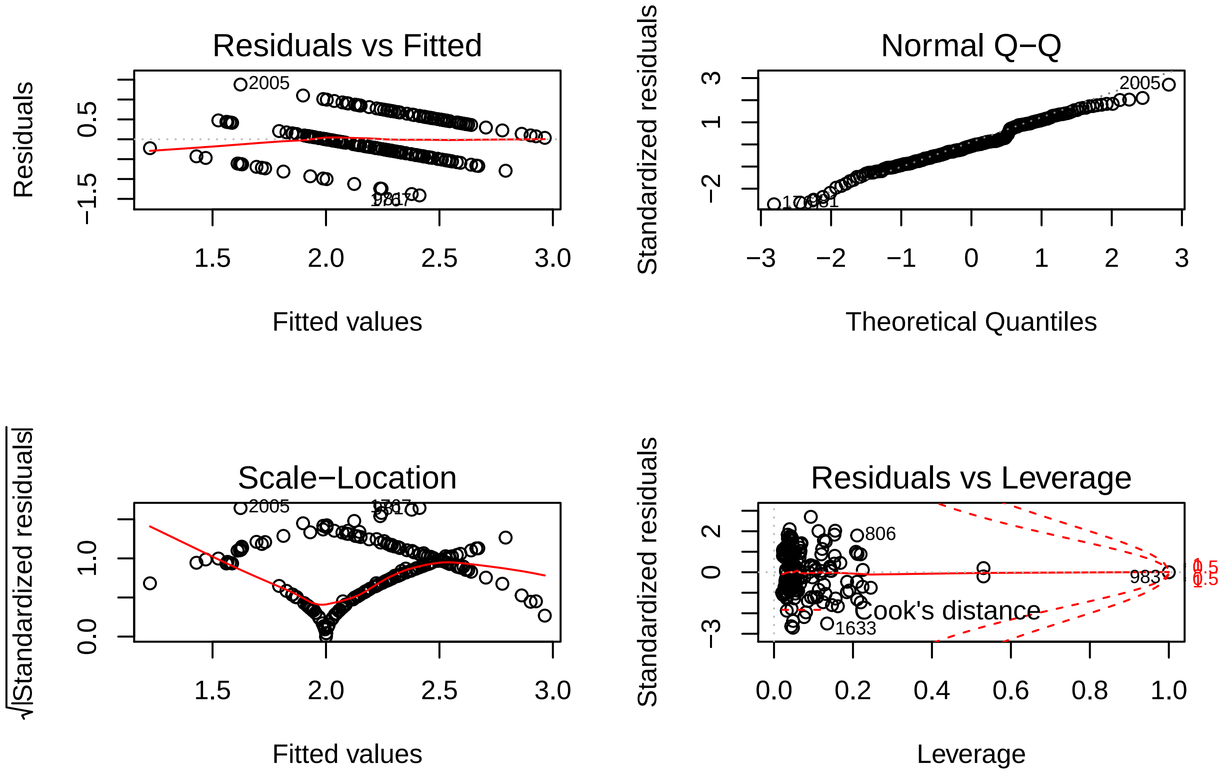
```
summary(new_model)$adj.r.squared
```

```
## [1] 0.2013563
```

```
par(mfrow = c(2, 2))
plot(new_model)
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



```
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)
#### subset size = 1 ####
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

#### subset size = 2 ####
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

#### subset size = 3 ####
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

#### subset size = 4 ####
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

#### subset size = 5 ####
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

#### subset size = 6 ####
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

## Answer
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")
```

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

```
## Single term additions
##
## Model:
## Happy ~ 1
##           Df Sum of Sq    RSS     AIC  F value    Pr(>F)
## <none>                 72.760 -208.31
## JobSat.f   6   10.1623 62.597 -227.00  63.6928 < 2.2e-16 ***
## OwnHome.f  2    0.5933 72.166 -205.99   9.6936 6.417e-05 ***
## Marital.f  4    6.0739 66.686 -218.10  53.6470 < 2.2e-16 ***
## Household  1    0.9268 71.833 -208.93  30.4377 3.824e-08 ***
## Age        1    0.0199 72.740 -206.37   0.6470    0.4213
## Income     1    4.1472 68.613 -218.29 142.5867 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 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")
```

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

```
## Single term additions
##
## Model:
## Happy ~ JobSat.f
##           Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                 62.597 -227.00
## OwnHome.f  2    0.5230 62.075 -224.72  3.1383  0.043929 *
## Marital.f  4    3.8999 58.698 -232.13 12.3415 1.010e-09 ***
## Household  1    0.8328 61.765 -227.74 10.0581  0.001579 **
## Age        1    0.0335 62.564 -225.11  0.3993  0.527644
## Income     1    2.0339 60.564 -231.74 25.0533 6.963e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## Single term deletions
##
## Model:
## Happy ~ JobSat.f + OwnHome.f + Marital.f + Household + Age +
##     Income
##           Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                 53.815 -239.84
## JobSat.f   6    6.9053 60.721 -227.22  4.0205 0.0008187 ***
## OwnHome.f  2    0.9689 54.784 -240.20  1.6925 0.1868556
## Marital.f  4    4.6015 58.417 -231.10  4.0188 0.0037590 **
## Household  1    2.5168 56.332 -232.52  8.7921 0.0034175 **
## Age        1    1.4842 55.300 -236.29  5.1850 0.0239082 *
## Income     1    1.2422 55.058 -237.19  4.3396 0.0385887 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Ownhome seems to be insignificant
drop1(lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household +
    Income), test = "F")
```

```
## Single term deletions
##
## Model:
## Happy ~ JobSat.f + OwnHome.f + Marital.f + Household + Income
##           Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                 55.621 -239.72
## JobSat.f   6    6.8734 62.494 -227.72  3.9338 0.0009888 ***
## OwnHome.f  2    0.5914 56.212 -241.54  1.0154 0.3642221
## Marital.f  4    3.7442 59.365 -234.30  3.2144 0.0139487 *
## Household  1    1.7410 57.362 -235.37  5.9786 0.0153868 *
## Income     1    1.0130 56.634 -238.00  3.4785 0.0637063 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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.

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

```
## [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.

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

```
## bcPower Transformations to Multinormality
##             Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
## JobSat.f       0.1523        0.00      -0.0976       0.4023
## OwnHome.f     -1.9516       -2.00      -2.6378      -1.2653
## Marital.f      0.2230        0.00      -0.0528       0.4988
## Household.f   -0.2118        0.00      -0.5046       0.0810
## Income         0.2161        0.22       0.1285       0.3037
## Age            0.4522        0.50       0.0432       0.8611
##
## Likelihood ratio tests about transformation parameters
##                                      LRT df         pval
## LR test, lambda = (0 0 0 0 0 0)  69.98314  6 4.121148e-13
## LR test, lambda = (1 1 1 1 1 1) 475.04275  6 0.000000e+00
```

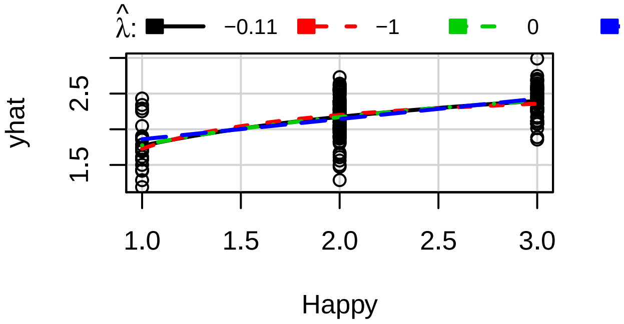
```r
Age_transformed <- Age^0.226
m_new <- lm(Happy ~ JobSat.f + OwnHome.f + Marital.f + Household.f +
    log(Income) + Age_transformed)
# summary(m_new)

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

```
##       lambda      RSS
## 1 -0.1130549 14.45109
## 2 -1.0000000 14.66344
## 3  0.0000000 14.45491
## 4  1.0000000 14.78982
```

```r
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
```


$\hat{\lambda}$:
yhat
Happy