Stats 101A Project

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

3/23/2018

Introduction

- In this study, we are looking to create a model to predict *Happiness*. We have variables *Household*, *Health*, *OwnHome*, *Instagram*, *Marital*, *Sex*, *Age*, *Children*, *Education*, *JobSat*, *Income*, and *WorkHrs*. By finding a viable model, we could be able to use it for increasing happiness in a variety of fields and applications from academic settings to social welfare.
- We will first clean the data, take a high-level look at it, transform our predictors, and then remove insignificant variables. We expect to encounter problems with the data along the way, as this is a social sciences data set, and observations are not being dialed in by a computer and could be susceptible to human error.

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.

```
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
```

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

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

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

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

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

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

```
#pairs(happiness_data, gap=0.4, cex.labels=1.5)
```

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
```

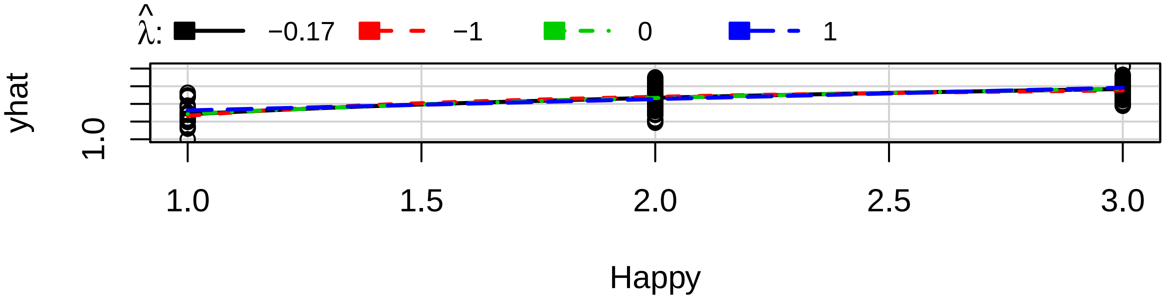
```
full_model_transform_log_inverse_response <- lm(log(Happy) ~
    Household.f + OwnHome.f + Instagram.f + Marital.f + Children.f +
        Education_transformed + JobSat.f + Income_log + Age_transformed +
        Sex.f)

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))))
```

Single 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*, *Age*, and *Education*). Our R^2 and R^2_{adj} values dropped significantly to 0.170019 and 0.1463052 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")
```

```r
reduced_6 <- drop1(update(full_model_transform_log_inverse_response,
    ~. - Instagram.f - Children.f - OwnHome.f - Sex.f - Age_transformed -
        Education_transformed), test = "F")
```

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

summary(updated_model)$r.squared
```

```
## [1] 0.170019
```

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

```
## [1] 0.1463052
```

AIC, AICc, BIC

AIC, AICc, and BIC subsets show a subset of size 2 to be optimal. However, doing so drops our R^2 and R^2_{adj} significantly and cuts our number of predictors drastically. We choose not to implement the subset of size 2.

```
##   Size        AIC       AICc        BIC
## 1    1 -2099.9811 -2099.9709 -2059.6162
## 2    2 -2151.4770 -2151.4600 -2088.0464
## 3    3  -758.0509  -758.0255  -660.0218
## 4    4  -639.0840  -639.0483  -535.2885
```

Leverages, Outliers, and Influential Points

Now we see how many bad leverage points we have. We have 22 bad leverage points. Amongst 600 social sciences observations, this is reasonable due to human error. We conclude that our final model is accurate.

```r
StanRes1 <- rstandard(updated_model)
leverage1 <- hatvalues(updated_model)
cookd1 <- cooks.distance(updated_model)
p <- 4
n <- 600

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

```
##  [1]  30  84 128 141 158 166 171 226 252 307 313 343 377 396 437 472 489
## [18] 493 507 521 571 588
```

Final Model and Conclusion

- Steps we took: data clean up, look at full pairs() plots, transformation of predictors, cursory variable selection where we looked at our predictors from a broad view, F-testing / variable selection where we looked at p-values to select our predictors, AIC, AICc, BIC looking at all possible sets, and finally looking at any bad leverage points.
- Our final model contains 4 of the 12 possible predictors we began with, 3 categorical and 1 numerical. For each categorical value included, at least one level is significant, justifying all their inclusions. The R^2 and R^2_{adj} values we obtained are quite low, as is typical in social science data. Our model thus states

that the key determining factors for an individual's happiness are their marital status, job satisfaction, income and to an extent, the number of members in their household, particularly for a household of only two members.

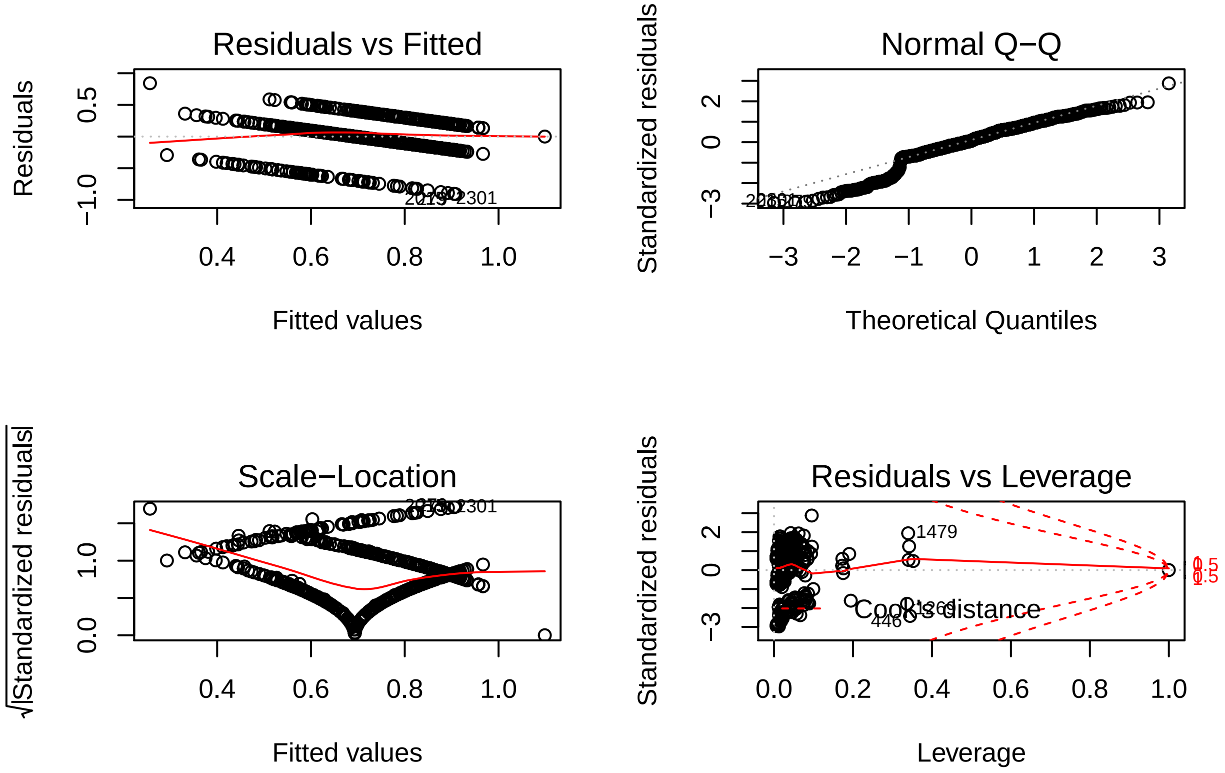
- We found that the R^2 value is not everything in data analysis and regression. There are so many other factors to look out for, and sometimes sacrificing a high R^2 value for a more statistically sound model is the correct way to go.
- The model makes sense in the real world. For example one of the predictors we found to be very accurate is *Income* after we applied a log transformation to it to increase its predicting power. We see through numerous studies and articles that wealth has diminishing returns on happiness (for example: https://www.cnbc.com/2015/12/14/money-can-buy-happiness-but-only-to-a-point.html). The fact that *Income* responded better to our model after a log transformation was very interesting. We found a large income difference between those at the low and middle happiness levels, but a much smaller one between those at the middle and high happiness levels, implying the effect of income on happiness plateaus. A well-publicized study, conducted by Princeton University professors and Economics Nobel Prize winners Angus Deaton and Daniel Kahneman, found that "while life evaluation rose steadily with annual income, the quality of the respondents' everyday experiences did not improve beyond approximately $75,000 a year." At incomes lower than this value, respondents reported increased sadness and stress, and lower happiness levels (http://wws.princeton.edu/news-and-events/news/item/two-wws-professors-release-new-study-income%E2%80%99s-influence-happiness).
- One of the big limitation of the analysis we believe is caused by the vast amount of missing data which ultimately reduced what we had to work with. Variables like *Health* and *WorkHrs* did not work with the model, because of how many NAs they had (811 and 1898 NAs respectively). We felt it was not right to tamper with the original data. However, in the future, a way to salvage some of our variables is to input the mean value of the column in position of all the NA entries. This will make the variable still usable but might breach ethical practices of the social science data.
- Our analysis is limited in that it only uses 4 predictors. Happiness is a human emotion; it is thus complicated and certainly cannot be definitively predicted by 4 factors, or even 12, for that matter. This is part of why R^2 and R^2_{adj} values are low in social sciences; human behavior is immensely difficult to predict.

```r
summary(updated_model)
```

```
##
## Call:
## lm(formula = log(Happy) ~ Household.f + Marital.f + JobSat.f +
##     Income_log)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9085 -0.1385  0.0232  0.2066  0.8419
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.493849   0.134165   3.681 0.000254 ***
## Household.f2   0.095511   0.034347   2.781 0.005594 **
## Household.f3   0.044931   0.045800   0.981 0.326981
## Household.f4  -0.083052   0.081541  -1.019 0.308836
## Household.f5  -0.053529   0.129606  -0.413 0.679743
## Household.f6   0.039566   0.180385   0.219 0.826460
## Household.f8   0.501450   0.314927   1.592 0.111855
## Marital.f2    -0.177191   0.069512  -2.549 0.011051 *
## Marital.f3    -0.103372   0.040588  -2.547 0.011119 *
## Marital.f4    -0.176811   0.069652  -2.538 0.011387 *
## Marital.f5    -0.064764   0.033894  -1.911 0.056511 .
## JobSat.f2     -0.009829   0.037340  -0.263 0.792470
## JobSat.f3     -0.124555   0.038943  -3.198 0.001455 **
## JobSat.f4     -0.185987   0.063941  -2.909 0.003764 **
## JobSat.f5     -0.186692   0.060201  -3.101 0.002019 **
## JobSat.f6     -0.348749   0.088456  -3.943 9.02e-05 ***
## JobSat.f7     -0.290503   0.181159  -1.604 0.109337
## Income_log     0.031429   0.012714   2.472 0.013711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3076 on 595 degrees of freedom
##   (1754 observations deleted due to missingness)
## Multiple R-squared:   0.17,  Adjusted R-squared:  0.1463
## F-statistic:  7.17 on 17 and 595 DF,  p-value: 4.543e-16
```

Appendix

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


Residuals vs Fitted
Residuals
0.5
−1.0
0.4
0.6
0.8
1.0
Fitted values
2019
2301
Normal Q−Q
Standardized residuals
2
0
−3
−3
−2
−1
0
1
2
3
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
1.0
0.0
0.4
0.6
0.8
1.0
Fitted values
2301
Residuals vs Leverage
Standardized residuals
2
0
−3
0.0
0.2
0.4
0.6
0.8
1.0
Leverage
1479
1269
446
Cook's distance
0.5
0.5