W203_Final_1

Natarajan Shankar

August 15, 2016

Data Analysis: Perform administrative and basic setup tasks

```
library(MASS); library(gvlma); library(sandwich) ; library(stats)
suppressMessages(library(lmtest));

# Set a seed for repeatable results, when needed
set.seed(123456)

# Read in the dating csv file and look at contents
datingDF <- read.csv("Dating.csv", header=TRUE)
```

```
#
#
#
#
#
#
#
```

Ali, thank you for a great term. Cory, thank you for your support - shankar

Question 15, Part a : Clean up and reverse scale on life_quality variable

```
# Start by recording the original number of rows - 2252
nrow(datingDF)
```

```
## [1] 2252
```

```
# Check the construct of the life_quality variable
str(datingDF$life_quality)
```

```
##  Factor w/ 7 levels "1","2","3","4",..: 2 2 3 5 3 4 3 5 2 2 ...
```

```
class(datingDF$life_quality)
```

```
## [1] "factor"
```

```
levels(datingDF$life_quality)
```

```
## [1] "1"          "2"          "3"          "4"          "5"
## [6] "Don't know" "Refused"
```

```
# life_quality attribute is a factor with 7 levels including "Don't know" and "Refused"
# Convert the "Don't know" and "refused" factors into <NA>
datingDF$life_quality[datingDF$life_quality == "Don't know"] <- NA # There are 8 instances
datingDF$life_quality[datingDF$life_quality == "Refused"] <- NA # there are 12 instances

# Now that two factors have been dropped, ensure that factor levels is reset
datingDF$life_quality <- droplevels(datingDF$life_quality)

# convert life_quality data into numeric form
datingDF$life_quality <- as.numeric(datingDF$life_quality)

# reverse the ranking within life_quality so that 5 = excellent and 1 = poor
datingDF$life_quality <- (max(datingDF$life_quality, na.rm = T) +
                     min(datingDF$life_quality, na.rm = T) - datingDF$life_quality)

# What is the mean quality_of_life in the sample
meanQoL <- mean(datingDF$life_quality, na.rm = TRUE)
meanQoL
```

```
## [1] 3.392921
```

Answer to Question 15, Part a :

The mean quality_of_life in the sample is : `3.39292`

Question 15, Part b : Clean up years_in_relationship variable

```r
# Check the construct of the years_in_relationship variable
class(datingDF$years_in_relationship)
```

```
## [1] "factor"
```

```r
levels(datingDF$years_in_relationship)
```

```
##  [1] " "       "0"       "1"       "10"      "11"      "12"      "13"
##  [8] "14"      "15"      "16"      "17"      "18"      "19"      "2"
## [15] "20"      "21"      "22"      "23"      "24"      "25"      "26"
## [22] "27"      "28"      "29"      "3"       "30"      "31"      "32"
## [29] "33"      "34"      "35"      "36"      "37"      "38"      "39"
## [36] "4"       "40"      "41"      "42"      "43"      "44"      "45"
## [43] "46"      "47"      "48"      "49"      "5"       "50"      "51"
## [50] "52"      "53"      "54"      "55"      "56"      "57"      "58"
## [57] "59"      "6"       "60"      "61"      "62"      "63"      "65"
## [64] "66"      "67"      "7"       "8"       "86"      "9"       "97"
## [71] "Refused"
```

The number "97" looks suspicious but the Pew site does not say anything suspicious other than "97 or more". Leave as is

```r
# The years_in_relationship attribute is a factor with 7 levels including " " and "Refused"
# Convert the "Don't know" and "refused" into <NA>
datingDF$years_in_relationship[datingDF$years_in_relationship == " "] <- NA
datingDF$years_in_relationship[datingDF$years_in_relationship == "Refused"] <- NA

# Now that two factors have been removed, redo the factor levels
datingDF$years_in_relationship <- droplevels(datingDF$years_in_relationship)

# Recode the years_in_relationship values as numeric
datingDF$years_in_relationship <- as.numeric(as.character(datingDF$years_in_relationship))
nrow(datingDF)
```

```
## [1] 2252
```

```r
# What is the mean of years_in_relationship in the sample?
meanYiR <- mean(datingDF$years_in_relationship, na.rm = TRUE)
meanYiR
```

```
## [1] 13.47697
```

Answer to Question 15, Part b :

The mean years_in_relationship in the sample is : `13.47697`

Question 15, Part c : First step with preparing for Nested Regression

```r
# Question 15, Part c : Start from the original data set
nrow(datingDF)
```

```
## [1] 2252
```

```r
# Study the use_internet variable before using it
str(datingDF$use_internet)
```

```
##  Factor w/ 5 levels " ","Don't know",..: 5 5 1 1 1 5 1 5 5 1 ...
```

```r
class(datingDF$use_internet)
```

```
## [1] "factor"
```

```r
levels(datingDF$use_internet)
```

```
## [1] " "          "Don't know" "No"         "Refused"    "Yes"
```

```r
# The years_in_relationship attribute is a factor with 5 levels
# including " ", "Don't know" and "Refused"
# Convert the "Don't know" and "refused" into NA
datingDF$use_internet[datingDF$use_internet == " "] <- NA
datingDF$use_internet[datingDF$use_internet == "Don't know"] <- NA
datingDF$use_internet[datingDF$use_internet == "Refused"] <- NA

# drop the unused factor levels
datingDF$use_internet <- droplevels(datingDF$use_internet)

# Convert into numeric
datingDF$use_internet <- as.numeric(datingDF$use_internet)

# Find the number of rows that have no missing values for
# life_quality, years_in_relatinship and use_internet
# Note : data in datingDF is being copied into completeDF
completeDF <- datingDF[complete.cases(datingDF$years_in_relationship,
                            datingDF$life_quality, datingDF$use_internet),]

# Count the number of complete rows : 1090
nrow(completeDF)
```

```
## [1] 1090
```

Answer to Question 15, Part c :

Number of cases with complete values for life__quality, years__in__relatinship and use__internet is : 1090

Question 15, Part d : Fit an OLS Model

```r
# Fit an OLS model to the data in the previpus step that predicts life_quality
# as a linear function of years_in_relationship
modelYiR <- lm(life_quality ~ years_in_relationship, data = completeDF)
summary(modelYiR)
```

```
##
## Call:
## lm(formula = life_quality ~ years_in_relationship, data = completeDF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6296 -0.4799 -0.3302  0.6698  1.6698
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.33022    0.04170  79.853   <2e-16 ***
## years_in_relationship  0.00499    0.00197   2.533   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.093 on 1088 degrees of freedom
## Multiple R-squared:  0.005861,   Adjusted R-squared:  0.004947
## F-statistic: 6.414 on 1 and 1088 DF,  p-value: 0.01146
```

```r
# Look for practical significance through R^2, P-value and F value
anova_modelYiR <- anova(modelYiR)
anova_modelYiR
```

```
## Analysis of Variance Table
##
## Response: life_quality
##                         Df  Sum Sq Mean Sq F value  Pr(>F)
## years_in_relationship    1    7.66  7.6565  6.4143 0.01146 *
## Residuals             1088 1298.71  1.1937
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answers to Question 15, Part d :

The slope coefficient for life_quality vs. years_in_relationship is : `0.00499`

Without any predictors, the model predicts that a rating of `3.33022` **is the comparison score, at a high level of statistical significance (as ascertained from the corresponding P-value)**

For the slope, the associated p-value is `0.01146` **and because the value is less than 0.05, this slope value is statistically significant. The F value in the associated model is** `6.41428` **which is greater than 1, and at a p-value of 0.01146, and so the model and hence the slope is statistically significant.**

However, the corresponding R^2 value is : `0.00586`**. The corresponding Pearson coefficient is :** `0.07656`**. This value indicates a low correlation between the predictor and the outcome and is of the order of 7.7%**

Question 15, Part e : Fit a second OLS model that also includes use_internet

```r
# Fit an OLS model to the data that predicts life_quality
# as a linear function of years_in_relationship AND use_internet
modelYiR_UI <- lm(life_quality ~ years_in_relationship + use_internet, data = completeDF)
summary(modelYiR_UI)
```

```
##
## Call:
## lm(formula = life_quality ~ years_in_relationship + use_internet,
##     data = completeDF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.61852 -0.53523 -0.01881  0.60195  2.00568
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.590578   0.167005  15.512  < 2e-16 ***
## years_in_relationship 0.004899   0.001952   2.509   0.0122 *
## use_internet          0.403738   0.088325   4.571 5.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.083 on 1087 degrees of freedom
## Multiple R-squared:  0.02461,    Adjusted R-squared:  0.02282
## F-statistic: 13.71 on 2 and 1087 DF,  p-value: 1.314e-06
```

```r
# Look for practical significance through F value and R^2 value
anova_modelYiR_UI <- anova(modelYiR_UI)
anova_modelYiR_UI
```

```
## Analysis of Variance Table
##
## Response: life_quality
##                         Df  Sum Sq Mean Sq F value   Pr(>F)
## years_in_relationship    1    7.66  7.6565  6.5316  0.01073 *
## use_internet             1   24.49 24.4930 20.8943 5.41e-06 ***
## Residuals             1087 1274.22  1.1722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer to Question 15, Part e :

The slope coefficient for use_internet is : `0.40374`.

Given the associated p-value of `5.41\times 10^{-6}` there is strong evidence that this value is statistically significant.

Also, given that the F value in the associated model is `20.8943`, a much bigger F compared to the F value for years_in_relationship alone, the new predictor and its slope are statistically and practically significant. The use_internet variable adds high practical significance.

The corresponding R^2 value is : `0.02461` and the corresponding Pearson coefficient is : `0.15688`. This value indicates a correlation between the predictor and the outcome and is of the order of 16% i.e. 16% of the outcome variable can be explainefd by this predictor.

Question 15, Part f : Assess improvement from first model to the second

```r
# Compute the F ratio and associated p-value between the two regression models
# Assess the improvement from the first model to the second
modelComparisonData <- anova(modelYiR, modelYiR_UI)
modelComparisonData
```

```
## Analysis of Variance Table
##
## Model 1: life_quality ~ years_in_relationship
## Model 2: life_quality ~ years_in_relationship + use_internet
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   1088 1298.7
## 2   1087 1274.2  1    24.493 20.894 5.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# extract the F value
modelComparisonData$F[2]
```

```
## [1] 20.8943
```

```r
# extract the p-value
modelComparisonData$Pr[2]
```

```
## [1] 5.409549e-06
```

Answer Question 15, Part f :

The significance (akin to comparing R^2) is tested by examining the F-ratio and the associated p-value

Inclusion of the use_internet variable significantly improved the fit of the new model with F(1, 1087) = `20.8943`, with p-value of `5.41\times 10^{-6}`

F Factor improvement from first model to second model is a factor of : `3.25747`

Question 16 Part a : Logistic Regression

This is question 16, start again with original data set from Pew rather than using data from Question 15

```
# start from original data set
datingDF <- read.csv("Dating.csv", header=TRUE)

# Study the flirted_online variable before using it
str(datingDF$flirted_online)
```

```
##  Factor w/ 5 levels " ","Don't know",..: 3 3 3 1 3 3 3 5 3 3 ...
```

```
class(datingDF$flirted_online)
```

```
## [1] "factor"
```

```
levels(datingDF$flirted_online)
```

```
## [1] " "          "Don't know" "No"         "Refused"    "Yes"
```

```
# The flirted_online attribute is a factor with 5 levels
# including " ", "Don't know" and "Refused"
# Convert the " ", Don't know" and "refused" into NA
datingDF$flirted_online[datingDF$flirted_online == " "] <- NA
datingDF$flirted_online[datingDF$flirted_online == "Don't know"] <- NA
datingDF$flirted_online[datingDF$flirted_online == "Refused"] <- NA

# drop the unused factor levels
datingDF$flirted_online <- droplevels(datingDF$flirted_online)

# Now look for complete cases across all of flirted_online
# Note : data in datingDF is being copied into completeDF
completeDF <- datingDF[complete.cases(datingDF$flirted_online),]

# Compute new number of valid rows
nrow(completeDF)
```

```
## [1] 1887
```

```
# What are the odds that a respondent in the sample has flirted online at some point
odds_flirted <- nrow(completeDF[completeDF$flirted_online =="Yes",])/
                        nrow(completeDF[completeDF$flirted_online =="No",])
odds_flirted
```

```
## [1] 0.2613636
```

Answer to Question 16 Part a :

The odds that a respondent in the sample has flirted online is : `0.26136`. The number is reached by dividing the number that reported "yes" by the number that reported "no", from the original data set with NA rows removed.

Question 16, Part b : Conduct a Logistic Regression to predict flirted_online versus usr

```r
# Study the usr variable (where the respondent lives) before using the variable
str(datingDF$usr)
```

```
##  Factor w/ 4 levels " ","Rural","Suburban",..: 2 3 3 3 4 2 3 4 3 3 ...
```

```r
class(datingDF$usr)
```

```
## [1] "factor"
```

```r
levels(datingDF$usr)
```

```
## [1] " "        "Rural"    "Suburban" "Urban"
```

```r
# Convert the usr value of " " to NA prior to using it
datingDF$usr[datingDF$usr == " "] <- NA

# drop the unused factor levels
datingDF$usr <- droplevels(datingDF$usr)

# Find the number of rows that have no missing values for
# flirted_online and usr
completeGLMDF <- datingDF[complete.cases(datingDF$flirted_online, datingDF$usr),]

# Establish the number of complete rows
nrow(completeGLMDF)
```

```
## [1] 1885
```

```r
# Look at the creation of dummy variable combinations for the 3 tiers in usr
contrasts(completeGLMDF$usr)
```

```
##          Suburban Urban
## Rural           0     0
## Suburban        1     0
## Urban           0     1
```

```r
# verify levels of each categorivcal variable prior to processing
levels(completeGLMDF$usr) # "rural" is baseline
```

```
## [1] "Rural"    "Suburban" "Urban"
```

```r
levels(completeGLMDF$flirted_online) # "No" is baseline
```

```
## [1] "No"  "Yes"
```

```r
# relevel towards "Suburban"" so that we can directly see odds for focus
# groups "Rural"" and "Urban" and the odds can be directly compared
completeGLMDF$usr <- relevel(completeGLMDF$usr, "Suburban")

# Re-look at the creation of dummy variable combinations for the 3 tiers in usr
# Suburban needs to be the baseline group
contrasts(completeGLMDF$usr)
```

```
##          Rural Urban
## Suburban     0     0
## Rural        1     0
## Urban        0     1
```

```r
# Run the Logistical Regression
glmModel <- glm(flirted_online ~ usr, data = completeGLMDF, family = binomial(), na.rm = TRUE)
```

```
## Error in glm.control(na.rm = TRUE): unused argument (na.rm = TRUE)
```

```r
# Look at the data from Logistical Regression
summary(glmModel)
```

```
##
## Call:
## glm(formula = flirted_online ~ usr, family = binomial(), data = completeGLMDF)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7592  -0.7592  -0.6731  -0.5432   1.9934
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.36949    0.08347 -16.406  < 2e-16 ***
## usrRural    -0.46974    0.17639  -2.663  0.00774 **
## usrUrban     0.27293    0.12330   2.214  0.02686 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1922.0  on 1884  degrees of freedom
## Residual deviance: 1903.4  on 1882  degrees of freedom
## AIC: 1909.4
##
## Number of Fisher Scoring iterations: 4
```

```r
glmModel$aic
```

```
## [1] 1909.359
```

```r
# get a feel for the overall fit of the model using analysis of deviance
glmModel_AoD <- anova(glmModel, test = "Chisq")
glmModel_AoD
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: flirted_online
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                  1884     1922.0
## usr   2   18.646      1882     1903.4 8.934e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Deviance analysis shows that improvement over the mean is significant

# Check confidence intervals of the coefficients
exp(confint(glmModel))
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2152716 0.2986639
## usrRural    0.4385490 0.8767810
## usrUrban    1.0314044 1.6728453
```

```r
# also compute the betas to understand impact of predictors on outcome
lm.beta(glmModel)
```

```
## Error in eval(expr, envir, enclos): could not find function "lm.beta"
```

Answer to Question 16 b :

The AIC value for the Logistic Regression model is : `1909.35897`

Question 16, Part c : Odds comparison

```r
glmModel$coefficients
```

```
## (Intercept)    usrRural    usrUrban
##  -1.3694872  -0.4697388   0.2729347
```

```r
# Look at the exponent of the coefficient to get a feel for increase in odds
odds <- exp(glmModel$coefficients)
odds
```

```
## (Intercept)    usrRural    usrUrban
##   0.2542373   0.6251656   1.3138144
```

```r
# How much bigger are the odds that an urban responden has flirted online
# than the odds that a rural respondent has flirted online
# Index 3 is Urban, Index 2 is Rural
odds[3]/odds[2]
```

```
## usrUrban
## 2.101546
```

Answer to Question 16 part c :

The odds multiple of a urban resident having flirted online versus a rural resident having flirted online is : `2.10155`. This effect is significant because it is greater than 1.

The p-value 8.934e-05 also shows that the effect of the associated model is significant